Arithmetic of pairings on algebraic curves for cryptography

Aurore Guillevic

Thales Communications & Security
Laboratoire Chiffre

École Normale Supérieure
Équipe Crypto

École doctorale Sciences Mathématiques de Paris Centre – ED 386

Thèse de doctorat

Étude de l'arithmétique des couplages sur les courbes algébriques pour la cryptographie

Spécialité : Informatique

présentée et soutenue publiquement le 20 décembre 2013 par

Aurore Guillevic

pour obtenir le grade de

Docteur de l'École Normale Supérieure

devant le jury composé de

Directeurs de thèse :	
Phong NGUYEN	(Inria et école normale supérieure, Paris)
Damien VERGNAUD	(École normale supérieure, Paris)
Encadrant industriel :	
Renaud DUBOIS	(Thales communications & security, Gennevilliers)
Rapporteurs :	
Pierrick GAUDRY	(Inria et CNRS, Loria, Nancy)
Marc JOYE	(Technicolor, Cesson-Sévigné)
Reynald LERCIER	(DGA-MI et université de Rennes I, Bruz)
Examinateurs :	
Antoine JOUX	(Chaire de cryptologie de la fondation de l'UPMC – LIP6, Paris)
Fabien LAGUILLAUMIE	(Université de Lyon I, Lyon)
David POINTCHEVAL	(Inria et école normale supérieure, Paris)
Benjamin SMITH	(Inria et école polytechnique, Palaiseau)

Remerciements

Je remercie Pierrick Gaudry, Marc Joye et Reynald Lercier qui ont accepté de rapporter cette thèse. Ils ont investi beaucoup de temps pour lire ce mémoire et leurs commentaires m'ont été très utiles pour l'améliorer. Je les remercie de leur patience et de leur minutie. Je remercie les examinateurs de ce jury qui ont accepté de se déplacer un vendredi 20 décembre, veille de vacances de Noël pour beaucoup : Antoine Joux, Fabien Laguillaumie, David Pointcheval et Benjamin Smith.

Je remercie profondément mon directeur de thèse Damien Vergnaud, qui s'est lancé dans l'encadrement de doctorants il y a trois ans. Il a su m'orienter vers des sujets intéressants et porteurs. Il s'est toujours montré très patient et pédagogue. Je le remercie beaucoup car sans lui cette thèse ne se serait pas aussi bien passée.

Je remercie tout autant Renaud Dubois avec qui j'ai commencé un stage au LCH il y a bientôt quatre ans et qui m'a suivie en thèse par la suite. Il m'a beaucoup aidée pour la programmation, par ses relectures d'articles en soumission, je que lui remettais parfois seulement à la dernière minute. Je remercie aussi David Lefranc et Sylvain Lachartre du même bureau, nous avons aussi partagé nos problèmes de programmation et compilation ce qui m'a beaucoup appris. Et merci de m'avoir remonté le moral aux moments nécessaires ! Merci pour votre aide et vos conseils.

Je remercie enfin tous les membres du LCH, en particulier Eric Garrido, Philippe Painchault qui m'a appris à jouer au squash, Olivier Orcière qui a toujours un fait historique à nous raconter, ou comment faisaient les Mayas pour multiplier de grands chiffres à la ficelle, Sonia Belaid, Emeline Hufschmitt et Alexandre Anzala Yamajako parce qu'ils sont toujours de bonne humeur. Merci à tous mes super collègues de bureau passés et présents : Ange, Vincent, Matthieu, Gaétan, Frédéric, Marine, Romain, Margaux, Brandon, Thomas et Christopher avec qui j'ai beaucoup apprécié discuter.

J'ai eu la chance d'être intégrée à la fois au LCH et à l'équipe crypto de David Pointcheval. Je remercie tous les membres de l'équipe, en particulier Phong Nguyen qui a supervisé le lancement de ma thèse et m'a suivie de loin. Je le remercie pour ses conseils avisés et pour ses relectures de ce mémoire. Pierre-Alain Fouque pour ses conseils qui arrivèrent au bon moment. Sorina ma co-auteur car j'ai beaucoup appris avec elle. Sonia et Sylvain pour leurs conseils. Les membres du projet ANR Best pour les discussions intéressantes que nous avons eues. Enfin merci particulièrement Liz et Ben pour leur grande aide pour réécrire certaines pages de ce mémoire.

Je garde le souvenir de moments bien agréables avec Liz, Miriam, Léo, Tancrède, Thomas et Mario. J'ai eu la chance de faire la connaissance dans l'open space crypto d'Olivier, Charles, Mehdi, Roch, Siamak, Fabrice, Angelo, Dario, Yuanmi et tous ceux qui étaient de passage. Merci à Charles et Pascal pour les mangas. Je remercie Mike et Barbara pour leur hospitalité lors des crues à Calgary juste avant ACNS en juin dernier. Je remercie Monique Crépin pour son aide à trouver un logement en Ile de France. Je remercie Claudie, Ludovic et Jacques du service informatique du DI. Je remercie pour leur aide indispensable Lydie Pezant, Nathalie Gaudechoux, Régine Guittard, Joëlle Isnard, Michelle Angely, Lise-Marie Bivard et Valérie Mongiat.

Je remercie l'ANRT, le LCH et le département d'informatique de l'ENS pour le financement de cette bourse Cifre et les moyens très appréciables déployés pour participer à de nombreuses conférences.

Je remercie beaucoup Monique Martineau pour tous ses conseils très avisés, sa présence et les romans policiers vraiment chouettes. Merci à Laura car nous avons partagé de bons moments. Je remercie énormément ma soeur Myriam qui est restée très proche et m'a soutenue même depuis la Scandinavie. Je ne parle pas encore le danois mais je connais déjà Copenhague aussi bien que le quartier Latin. Enfin je remercie mes parents qui sont là aujourd'hui.

Introduction à la cryptographie bilinéaire

La cryptographie asymétrique

Jusqu'au milieu du XXe siècle, la cryptographie consistait à chiffrer des données sensibles pour un archivage sûr ou pour des transmissions via des réseaux de communication publics. De nos jours, la cryptographie se doit aussi d'assurer l'intégrité des données et l'authentification des émetteurs et dépositaires sans recourir à une étape humaine.

Les débuts de la cryptographie moderne remontent aux prémices de la seconde guerre mondiale avec la conception de la machine Enigma, puis sa cryptanalyse par les Britanniques. On pourra consulter le chapitre 1 du livre [Ver12] à ce sujet. La cryptanalyse moderne et le premier calculateur sont nés à Bletchley Park en Angleterre. Ce site fut dédié au décryptage des communications adverses, chiffrées notamment avec Enigma. Une automatisation progressive d'une attaque par force brute de la machine Enigma y fut conçue et mise en œuvre.

La cryptographie moderne asymétrique a communément pour point de départ l'année 1976. Cette année-là, Diffie et Hellman publient leur article fondateur [DH76]. Merkle est aussi lié à l'histoire et apparaît comme troisième inventeur du brevet correspondant. Ce cryptosystème schématisé dans la Fig. 1 permet à deux participants de s'accorder sur une donnée secrète via un canal de transmission public (non sûr).

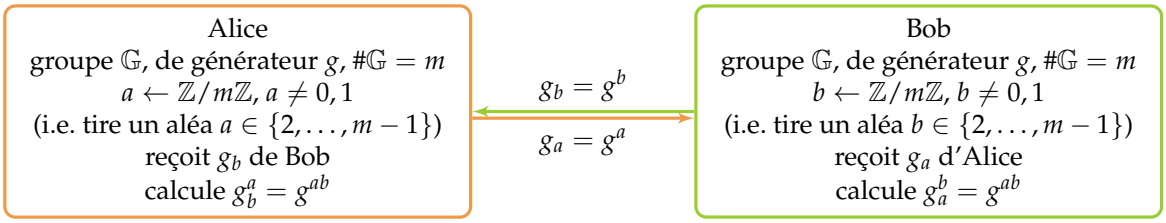


FIGURE 1 – Échange de clé de Diffie-Hellman. Alice et Bob connaissent l'élément g^{ab}.

Dans ce schéma, les éléments g^a, g^b qui transitent sur le canal public appartiennent à un groupe cyclique dans lequel il est facile de calculer g^a à partir de g et a mais difficile (impossible en temps et moyens informatiques raisonnables) de calculer le secret g^{ab} à partir des éléments g, g^a, g^b qui transitent sur le canal.

Le schéma basé sur la factorisation, proposé par Rivest, Shamir et Adleman (RSA) est quant à lui publié en 1978 [RSA78].

Le problème du logarithme discret

Le protocole d'échange de clés de Diffie-Hellman repose sur la difficulté de calculer l'élément g^{ab} à partir des trois éléments g, g^a et g^b. On pourra consulter [MvV97, §3.6 et 12.6] sur ce sujet. Ce calcul difficile est appelé le *problème Diffie-Hellman* ou DHP pour l'abréviation anglaise. Ce problème et ces variantes servent de point de départ à de nombreux protocoles utilisés couramment. Plus généralement, le problème du logarithme discret (DLP dans ce qui suit) est très étudié. Le DLP dans un groupe cyclique $\mathbb{G}$ d'ordre m (noté multiplicativement) est défini de la façon suivante : étant donnés un générateur g du groupe $\mathbb{G}$ et un élément aléatoire g_a du groupe $\mathbb{G}$, il s'agit de calculer l'entier $a \in \{2, \ldots, m-1\}$ tel que $g^a = g_a$. On peut voir aisément que s'il est facile de calculer le logarithme discret de n'importe quel

élément d'un groupe $\mathbb{G}$ alors il est facile de résoudre le problème Diffie-Hellman dans ce groupe. En effet, il suffit de calculer le logarithme discret a de g^a puis de calculer g^{ab} comme $g^{ab} = (g^b)^a$. La relation entre problème de Diffie-Hellman et problème de logarithme discret a été étudiée dans [MW99].

Le calcul de logarithmes discrets est supposé difficile dans certains groupes bien choisis. Une première proposition fut d'utiliser le groupe multiplicatif d'un corps fini, noté $\mathbb{F}_q^*$. L'identification de groupes appropriés, où le calcul de logarithmes discrets est très difficile, mais la multiplication très rapide, est toujours un domaine en activité en cryptographie. Pour assurer un bon niveau de sécurité à un protocole basé sur le DLP, on étudie la complexité en temps et en mémoire des attaques possibles dans ce groupe. Les attaques principales sur les groupes les plus répandus sont listées ci-après. La complexité de l'attaque est exprimée en nombre d'opérations (d'exponentiation $(g, a) \mapsto g^a$) dans le groupe $\mathbb{G}$, en fonction de l'ordre m du groupe $\mathbb{G}$. Les complexités sont données en bits, autrement dit une complexité de ℓ bits correspond à une attaque qui requiert 2^ℓ opérations. Cette notation *logarithmique* vient de la comparaison avec la cryptographie symétrique. En effet, étant donné un message chiffré avec une clé secrète de ℓ bits, une attaque par force brute pour retrouver la clé secrète et le clair correspondant va énumérer toutes les clés secrètes possibles. Il y a 2^ℓ clés secrètes possibles.

Une fois que l'on connaît le temps nécessaire pour chacune des attaques existantes, on dimensionne la taille du groupe en conséquence, afin de s'assurer que toutes les attaques connues nécessitent un temps de calcul conséquent.

1. Les attaques Baby-step Giant-step, (petit poucet et bottes de sept lieues) et ρ de Pollard calculent un logarithme discret dans un groupe $\mathbb{G}$ d'ordre m en temps $O(\sqrt{m})$ [MvV97, §3.6.2 et 3.6.3]. Ces attaques génériques sont possibles pour tout groupe $\mathbb{G}$. Pour obtenir une sécurité équivalente à ℓ bits, on choisit un groupe d'ordre au moins $m \geqslant 2^{2\ell}$, autrement dit $\log m \geqslant 2\ell$.
2. L'attaque de Pohlig-Hellman décompose le calcul du logarithme discret dans chaque sous-groupe premier de $\mathbb{G}$. Si l'on écrit $m = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$ alors l'attaque a pour complexité $O(\sum_{i=1}^{k} e_i(\log m + \sqrt{p_i}))$ [MvV97, §3.6.4]. Le terme prépondérant de cette complexité est $\sqrt{p_i}$ avec p_i le plus grand facteur premier de m. Une parade à cette attaque est de choisir un groupe d'ordre premier.
3. Dans les corps finis, des attaques spécifiques plus efficaces existent. Il s'agit des attaques de type *index calculus* ou calcul d'indice. Trois variantes existent pour trois cas différents de corps finis : grande, moyenne et petite caractéristique, la caractéristique d'un corps fini $\mathbb{F}_q$ étant le nombre premier p tel que q soit une puissance de p. Voici les trois principales complexités, en reprenant la classification de [JL07]. De plus récemment, des améliorations très importantes sont apparues, leur impact est également indiqué. Les complexités sont parfois exprimées avec la fonction L_Q dans ce contexte. Cette fonction vaut

$$L_Q(\alpha, c) = \exp\Big((c + o(1)) \ln^{\alpha}(Q) \ln^{1-\alpha}(\ln Q)\Big)$$

avec $0 \leqslant \alpha \leqslant 1$ et $c > 0$. Dès lors que $\alpha < 1$, la complexité exprimée avec cette fonction est *sous-exponentielle* en $\ln Q$. Lorsque $\alpha = 0$, la complexité correspondante est polynomiale en $\ln Q$.

   a) La première famille d'attaques concerne les corps finis en grande caractéristique, par exemple $\mathbb{F}_p$ avec p un grand nombre premier (de plus de mille bits) ou $\mathbb{F}_{p^{12}}$ avec p un nombre premier de 256 bits, ou plus généralement, les corps $\mathbb{F}_{p^e}$ avec e négligeable devant $(\ln^{2/3} q \ln^{1/3} \ln q)$. La complexité du *Number Field Sieve* (NFS), ou crible algébrique, est $\exp\Big(\big(\sqrt[3]{64/9} + o(1)\big) \ln^{1/3} q \ln^{2/3} \ln q\Big)$, autrement dit, $L_q(1/3, \sqrt[3]{64/9})$ avec $\sqrt[3]{64/9} \approx 1.923$. Une attaque avec la méthode NFS est a priori asymptotiquement plus efficace qu'une attaque générique. Plus précisément, cette attaque avec NFS dépend de la taille *totale* du corps fini et non pas de la taille du sous-groupe multiplicatif considéré. Ainsi, pour atteindre un même niveau de sécurité face à des attaques de deux types, génériques et avec NFS, on construira un corps fini de grande taille (par exemple, 3072 bits sont considérés comme apportant une sécurité de 128 bits) contenant un sous-groupe d'ordre premier de taille bien plus petite, 256 bits suffisent alors. Cette astuce ne permet pas de compresser les éléments du corps fini (les g^a d'un échange de clé Diffie-Hellman) mais permet d'avoir une taille réduite pour les exposants (les a, b) et ainsi avoir des exponentiations ($(g, a) \mapsto g^a$) moins coûteuses.

b) La deuxième famille d'attaques concerne les corps de petite caractéristique. Les plus utilisés en cryptographie sont de la forme $\mathbb{F}_{2^\ell}$ et $\mathbb{F}_{3^\ell}$. Les corps de la forme $\mathbb{F}_{p^e}$ avec p premier et e prépondérant devant $\ln^{2/3} q \ln^{1/3} \ln q$ aussi sont concernés. L'attaque a pour complexité $\exp\left(\left(\sqrt[3]{32/9} + o(1)\right) q^{1/3} \ln^{2/3} q\right)$ [JL07] avec $q = 2^\ell$ ou 3^ℓ. Mais depuis le début de l'année 2013 et la première publication [Jou13b], de nouvelles améliorations assez époustouflantes ont montré la vulnérabilité de ces corps de petite caractéristique [BBD+13, GGMZ13b, BGJT13, AMORH13], notamment lorsqu'ils apparaissent comme corps de plongement de courbes supersingulières. Il s'agit par exemple du corps $\mathbb{F}_{3^{6\cdot 97}}$, déjà attaqué en 2012 [HSST12].

c) Enfin entre ces deux possibilités, lorsque e est compris entre les frontières $O(\ln^{1/3} q \ln^{2/3} \ln q)$ et $O(\ln^{2/3} q \ln^{1/3} \ln q)$, ces corps de moyenne caractéristique connaissent aussi des attaques spécifiques. Lorsque le degré de l'extension est premier, il existe une attaque connue depuis 2006 en $\exp\left(\left(\sqrt[3]{128/9} + o(1)\right) \ln^{1/3} q \ln^{2/3} \ln q\right)$ [JL07]. En décembre 2012 [Jou13a], Joux a proposé une nouvelle amélioration de cette méthode de calcul de logarithme discret dans ces corps, en $L_q(1/4, c)$.

d) Lorsque le degré de l'extension e est friable (c'est-à-dire e est composé de petits nombres premiers), en petite et moyenne caractéristique, depuis très récemment il existe de prodigieux algorithmes pour calculer des logarithmes discrets, par exemple [Jou13a, BGJT13]. De plus, ces algorithmes s'appliquent d'une certaine façon aux corps de petite caractéristique et de degré d'extension premier. En quelque sorte, il s'agit de construire une petite extension $\mathbb{F}_{2^{\ell\cdot e}}$ puis de changer la représentation de ce corps pour en exploiter la structure plus riche afin d'appliquer des variantes des algorithmes pour les corps de moyenne caractéristique. Cette nouvelle méthode pour l'instant est plus efficace que précédemment lorsque ℓ est suffisamment petit, par exemple sur $\mathbb{F}_{3^{6\cdot 97}}$ où $\ell = 97$. Par contre lorsque par exemple $\ell = 1000$, il faut prendre $e = 10$ ce qui donne des paramètres trop grands pour être intéressants.

Depuis 2013, les corps de petite et moyenne caractéristique sont remis en cause pour de sérieuses raisons. Les attaques ne s'appliquent pas encore à tous les corps mais au vu des avancées majeures de ces derniers mois, il est préférable d'éviter d'utiliser des corps finis de petite et moyenne caractéristique. En particulier, cela remet en cause l'utilisation de courbes supersingulières en caractéristique 2 et 3, jusque-là très populaires dans le contexte des applications bilinéaires.

L'introduction des courbes elliptiques et hyperelliptiques en cryptographie

Pour instancier un protocole reposant sur l'hypothèse Diffie-Hellman, avec des paramètres de tailles les plus petites possibles pour un niveau de sécurité donné, on s'intéresse aux groupes dans lesquels seules les attaques génériques sont applicables. Ainsi, pour un niveau de 128 bits de sécurité, il est suffisant de construire un groupe d'ordre premier de 256 bits. D'ailleurs, lorsqu'on utilise le groupe multiplicatif d'un corps fini, on considère un sous-groupe d'ordre premier de 256 bits. La taille des exposants (a, b dans le schéma en Fig. 1) est ainsi optimale. Mais la taille totale du corps fini n'est pas optimale. Puisque les attaques par calcul d'indice s'appliquent, il faut un corps fini de taille bien plus grande pour contrebalancer ces attaques par calcul d'indice.

On s'intéresse donc aux groupes où seules les attaques génériques sont possibles. La loi de groupe doit bien sûr rester très efficace. Dans les années 70, les attaques par calcul d'indice n'étaient pas encore très développées. De plus l'arithmétique des corps finis était bien connue et efficace. C'est pourquoi le groupe multiplicatif d'un corps fini était très utilisé. De plus la multiplication y est très rapide. Cependant ces groupes ne sont plus optimaux depuis l'émergence des attaques sous-exponentielles exposées ci-dessus.

En 1985, Koblitz et Miller proposent indépendamment d'utiliser en cryptographie asymétrique le groupe de points d'une courbe elliptique définie sur un corps fini. Si la courbe est bien choisie, seules les attaques génériques s'appliquent. En effet jusqu'à maintenant, les tentatives pour adapter les attaques par calcul d'indice aux courbes elliptiques sont infructueuses. Il est de plus très facile d'identifier les courbes particulières à éviter. La dernière difficulté était de pouvoir construire des courbes avec un groupe d'ordre premier, ou bien contenant un très gros sous-groupe d'ordre premier. Pour cela, les algorithmes dits de *comptage de points* se sont beaucoup développés. De tels algorithmes sont aussi importants que les tests de

primalité pour construire de bons modules RSA. Finalement, la combinaison de méthodes dues à Schoof, Elkies et Atkin, appelée SEA, permet de déterminer l'ordre d'une courbe elliptique de taille cryptographique en quelques secondes sur un PC. Ainsi, il est devenu assez simple d'obtenir un bon exemple de courbe elliptique sur laquelle le logarithme discret est difficile. On définit un corps premier $\mathbb{F}_p$ de 256 bits, puis une courbe elliptique sur ce corps. On calcule son ordre grâce à la méthode SEA et on choisit de nouveaux paramètres pour la courbe elliptique tant que l'ordre calculé n'est pas premier. Il est possible de trouver une courbe convenable en moins d'une minute.

Il existe une seconde méthode pour construire une courbe elliptique appropriée. Il s'agit de choisir d'abord son ordre premier m, puis de construire un corps fini $\mathbb{F}_p$ et des paramètres qui détermineront une courbe elliptique d'ordre m sur ce corps. Cette méthode repose sur le calcul de polynômes de classes, polynômes de Hilbert ou polynômes de Weber par exemple. Là encore, de grandes avancées ont permis de pouvoir effectuer ces calculs pour de très grands nombres.

Et bien sûr la loi de groupe sur les courbes elliptiques est efficace. Elle est plus complexe que la simple multiplication dans un corps fini. Mais puisque sur une courbe elliptique, les éléments du groupe considéré sont de taille bien plus petite que pour un corps fini présentant un niveau de sécurité équivalent, la complexité de la loi de groupe est compensée par la rapidité obtenue grâce aux tailles bien plus petites des éléments manipulés.

Utilisation des couplages en cryptographie et cryptanalyse

Les *accouplements de Weil* apparaissent à la fin des années 40 en mathématiques. Le mathématicien André Weil les définit pour ses travaux en géométrie algébrique. Il introduisit ce qu'il nomma alors les *accouplements*. Après un passage en anglais (*pairings*), ils furent retraduits par *couplages* dans la communauté cryptographique. Le terme accouplement est toujours utilisé en mathématiques. Pour l'anecdote historique, Weil introduisit aussi la notation ø pour l'ensemble vide. Cette lettre est empruntée aux langues scandinaves. Elle se prononce /e/ et est l'abréviation d'est (le point cardinal de géographie) en danois.

Un couplage est une application bilinéaire $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Les trois groupes $\mathbb{G}_1, \mathbb{G}_2$ et $\mathbb{G}_T$ sont de même ordre m. L'application est bilinéaire à gauche et à droite, et non-dégénérée. Cela s'écrit, avec $\mathbb{G}_1$ et $\mathbb{G}_2$ notés additivement et $\mathbb{G}_T$ multiplicativement, étant donnés des éléments $g, g_1, g_2 \in \mathbb{G}_1$, $h, h_1, h_2 \in \mathbb{G}_2$, on a $e(g, h_1 + h_2) = e(g, h_1)e(g, h_2)$ et de même à gauche : $e(g_1 + g_2, h) = e(g_1, h)e(g_2, h)$. De plus l'application est non-dégénérée, c'est à dire que pour tout $g \in \mathbb{G}_1$ non nul, il existe un élément $h \in \mathbb{G}_2$ tel que $e(g, h)$ ne soit pas l'élément neutre de $\mathbb{G}_T$ (et de même à droite, étant donné un élément non nul $h \in \mathbb{G}_2$). Dans les détails, les deux groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ sont deux sous-groupes distincts, de même ordre, d'une courbe elliptique E définie sur un corps et $\mathbb{G}_T$ est une extension de degré fixé du corps sur lequel est définie la courbe elliptique.

Afin d'exploiter cette application bilinéaire en cryptographie, elle doit être facilement calculable (au sens calculable en temps polynomial en la taille des entrées) mais difficilement inversible. S'il est facilement calculable, le couplage permet de faire le lien entre le problème du logarithme discret dans le groupe de points d'une courbe elliptique et dans une extension d'un corps fini. Si l'on souhaite calculer le logarithme discret d'un élément $g_a \in \mathbb{G}_1$ en base g, alors on peut se ramener à calculer le logarithme discret de $e(g_a, h) \in \mathbb{G}_T$ en base $e(g, h)$ avec g un générateur de $\mathbb{G}_1$ et h un générateur de $\mathbb{G}_2$. Ces deux calculs de logarithme discrets doivent alors être de même difficulté dans $\mathbb{G}_1$ et $\mathbb{G}_T$. Un couplage n'est calculable efficacement que lorsque $\mathbb{G}_T$ est de taille raisonnable (par exemple, lorsque $\mathbb{G}_T$ est une extension de corps de degré compris entre 2 et 60 par rapport au corps de définition de la courbe elliptique).

En 1986, Victor Miller s'intéressa à l'accouplement de Weil et proposa une méthode pour le calculer en pratique [Mil86a]. Ces travaux furent publiés par la suite [Mil04], après avoir pris une importance considérable en cryptographie. La première utilisation avérée des couplages en cryptographie se trouve dans les travaux de thèse de Burton S. Jr Kaliski [Kal88] datant de 1988. Il programma en Macsyma un couplage de Weil. Le code source est disponible en annexe A de son mémoire de thèse. Macsyma était une bibliothèque de calculs développée en Lisp à partir des années 60 au Massachuset Institute of Technology. Avec ce code source se trouve un exemple de calcul d'accouplement de Weil sur la courbe supersingulière $E : y^2 = x^3 - x$ définie sur le corps $\mathbb{F}_{11}$. Suite aux travaux de Miller et Kaliski, Menezes, Okamoto et Vanstone présentèrent en 1993 [MOV93] une attaque contre le problème du logarithme discret sur des

courbes supersingulières. Deux années plus tard, Frey et Rück proposèrent la même attaque mais avec un calcul de couplage de Tate, plus rapide. Ces deux attaques exploitent la propriété du couplage de transférer le calcul de logarithme discret du groupe de points $E(\mathbb{F}_q)$ de la courbe elliptique E vers un sous-groupe multiplicatif d'un corps fini $\mathbb{F}_{q^k}$. Dans le cas des courbes supersingulières, l'article [MOV93] liste les corps d'immersion (ou de plongement) du couplage, qui sont une extension de degré 1, 2, 3, 4 ou 6 du corps de définition $\mathbb{F}_q$ de la courbe elliptique. Des méthodes spécifiques aux calculs de logarithme discret dans des corps finis existent alors et permettent un calcul bien plus efficace que les méthodes génériques applicables au sous-groupe de la courbe elliptique. Par exemple les courbes elliptiques en caractéristique 2, de la forme $y^2 + y = x^3$, définies sur $\mathbb{F}_{2^{61}}$ et $\mathbb{F}_{2^{127}}$ étaient alors proposées. L'attaque de Menezes, Okamoto et Vanstone eut pour conséquence de proscrire l'utilisation de telles courbes.

Implémentation des couplages en cryptographie

En 1993, lorsque Menezes, Okamoto et Vanstone proposèrent leur attaque, un calcul de couplage était bien loin de s'effectuer en quelques milisecondes. En 1999, Harasawa, Shikata, Suzuki et Imai [HSSI99] annoncèrent un calcul de couplage de Tate en 40000 secondes ($\sim$ 11 heures) sur une courbe supersingulière définie sur un corps premier de 50 chiffres décimaux (soit $\sim$ 170 bits). Leurs calculs nécessitaient aussi une mémoire très importante.

En 2000, Joux [Jou00] introduisit l'idée d'évaluer, à chaque étape de la boucle de calcul du couplage (la *boucle de Miller*), la fonction de Miller en le deuxième point du couplage, afin de ne plus avoir à stocker tous les coefficients de cette fonction en vue d'une évaluation finale en ce deuxième point. Cette façon de procéder lui permis de calculer un couplage en une seconde sur une courbe supersingulière définie sur un corps premier de 150 chiffres décimaux, soit $\sim$ 500 bits. Le corps de plongement était de taille double, soit près de 1024 bits, taille commune des modules RSA dans ces années. Un calcul de logarithme discret n'était pas plus aisé dans le corps de plongement. Avec ce temps de calcul d'une seconde, les couplages étaient alors tout à fait envisageables pour une utilisation dans de nouveaux protocoles. Il ne restait plus qu'à tendre vers un calcul en moins d'une miliseconde, temps alors comparable à un déchiffrement RSA. La contribution de Joux dans cet article n'était pas tant le protocole d'échange à la Diffie-Hellman à trois en un tour (Fig. 2) que le calcul d'un couplage en un temps record, d'ailleurs cet article fut accepté à une conférence de théorie algorithmique des nombres (ANTS). Le paragraphe suivant présente quelques étapes qui ont permis, dans les années 2000, de réduire considérablement les calculs de couplages, au point, de nos jours, de pouvoir les calculer sur des smartphones.

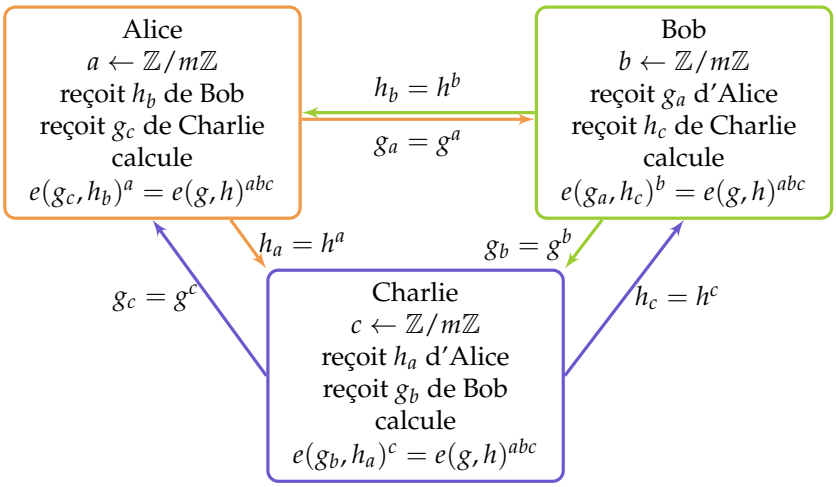


FIGURE 2 – Échange de clé de Joux (a.k.a. Triffie-Hellman). Alice, Bob et Charlie connaissent l'élément $e(g,g)^{abc}$. La sécurité repose sur la difficulté de calculer l'élément $e(g,h)^{abc}$.

Les améliorations apportées aux calculs de couplages deviennent tout de suite très techniques. Leur compréhension nécessite de larges prérequis en géométrie algébrique. Quelques avancées significatives sont toutefois rappelées ici. On considère un couplage $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ avec les trois groupes d'ordre

m. $\mathbb{G}_1$ et $\mathbb{G}_2$ sont des sous-groupes d'une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$. On a $\#E(\mathbb{F}_q) = q + 1 - t$ avec t appelée la trace de la courbe elliptique sur $\mathbb{F}_q$, et donc m divise $q + 1 - t$. La trace est de petite valeur, plus précisément $-2\sqrt{q} \leqslant t \leqslant 2\sqrt{q}$. Le troisième groupe $\mathbb{G}_T$ est un sous-groupe d'ordre m de l'extension $\mathbb{F}_{q^k}$. Le paramètre k revient constamment pour les couplages. Il est appelé degré de plongement ou degré d'immersion.

Les travaux de thèse de Benjamin Lynn, doctorant à Stanford University sous la direction de Daniel Boneh, ont contribué sensiblement à la compréhension des calculs de couplages. En 2002 [BKLS02], Barreto, Kim, Lynn and Scott proposent plusieurs optimisations importantes. Grâce à une représentation compacte du deuxième point Q du couplage $e(P, Q)$, certains facteurs apparaissant dans les calculs deviennent inutiles, ils ne contribuent plus à la valeur finale du couplage. Leur calcul peut être évité. Cette idée se généralise aux couplages avec un point Q en représentation compacte grâce à l'utilisation d'une *tordue* de la courbe initiale, de degré d, avec $d \mid k$ et $d \in \{1, 2, 3, 4, 6\}$ (en grande caractéristique). Ceci est expliqué en détails aux sections 1.4.4.3 et 1.4.4.4.

Une autre voie d'optimisation fut la réduction de la longueur de la *boucle de Miller*, partie importante du calcul de couplage. Pour un couplage de Tate, la boucle de Miller itère sur le paramètre m qui est l'ordre des sous-groupes auxquels appartiennent les deux points P et Q. Par analogie, l'exponentiation g^a se calcule avec une boucle itérant sur a. Deux courbes supersingulières couramment utilisées en petite caractéristique furent $E : y^2 + y = x^3 + x + b$, définie sur $\mathbb{F}_{2^{2m+1}}$ et avec $b \in \{0, 1\}$, de trace $\pm t = 2^{m+1}$ et de degré de plongement $k = 4$; et $E : y^2 = x^3 - x \pm 1$ définie sur $\mathbb{F}_{3^{2m+1}}$, de trace $\pm t = 3^{m+1}$ et de degré de plongement $k = 6$. En 2004, Barreto, Galbraith, Ó hÉigeartaigh et Scott introduisent le couplage *eta*, ou η_T sur des courbes supersingulières en petite caractéristique [BGOS07]. Dursmaa et Lee en 2004 ont initié ces travaux en caractéristique 3. L'idée de Barreto *et al.* est d'itérer la boucle de Miller sur $t - 1$ au lieu de m. La trace étant plus courte de moitié que l'ordre du sous-groupe considéré, la boucle en est réduite d'autant. Barreto *et al.* montrent que le couplage est toujours bilinéaire et non-dégénéré. Cette méthode ne peut pas s'appliquer en grande caractéristique, ou bien si elle s'applique, elle ne permet pas d'améliorer les calculs.

Peu après, Hess, Smart et Vercauteren [HSV06] proposent une nouvelle version, le couplage *ate*, qui cette fois-ci s'applique aux courbes ordinaires. Le degré de plongement k peut être plus grand que 2 ou 3 pour une courbe ordinaire. En pratique il est de 6 à 12. La méthode devient alors intéressante. Leur méthode est expliquée à la section 1.4.4.5.

Pour finir en 2009, Vercauteren [Ver10] introduit les couplages *optimal ate*. Il s'agit d'exprimer plus finement un couplage *ate* en fonction du couplage de Tate correspondant. Les termes correcteurs qui apparaissent entre les deux, s'il y en a, sont alors eux-mêmes susceptibles de définir un couplage bilinéaire et non-dégénéré, grâce à l'égalité des deux couplages ate et Tate, et de ces termes correctifs. Le couplage *ate optimal* est bien approprié aux constructions de courbes avec la méthode de Brezing-Weng et ses variantes. L'exemple le plus répandu en ce moment est un couplage *ate optimal* sur une courbe de Barreto-Naehrig. La longueur de la boucle de Miller y est divisée par quatre. Les détails se trouvent à la section 1.4.4.6.

Construction de courbes appropriées aux couplages

En parallèle, de nouvelles courbes propres aux couplages furent découvertes. Il s'agit de construire des courbes avec un petit degré de plongement k. Les courbes supersingulières furent bien identifiées dès 1993 et l'article [MOV93]. C'est pour ces raisons historiques que les courbes supersingulières furent très utilisées pour instancier des couplages. Depuis peu, avec les protocoles basés sur des groupes d'ordre composé, ces courbes supersingulières connaissent un regain d'intérêt.

En 2001, Miyaji, Nakabayashi et Takano [MNT00] caractérisent des courbes elliptiques de degré de plongement égal à 3, 4 et 6. La première motivation de leurs recherches était de présenter de nouvelles courbes elliptiques vulnérables à l'attaque de Frey et Rück, autrement dit sur lesquelles un couplage de Tate était calculable. Ces courbes sont ordinaires, contrairement aux précédentes. Leurs constructions de courbes ordinaires sur des corps premiers, de degré de plongement 6, se révélèrent bien appropriées aux instanciations de protocoles utilisant des couplages, à un niveau de sécurité de 80 bits.

D'autres méthodes de génération de courbes de petit degré de plongement furent proposées. On peut retenir les méthodes de Cocks–Pinch [CP01], de Brezing-Weng [BW05], de Dupont, Enge et Morain [DEM05] et la classification exhaustive (ou presque) de Freeman, Scott et Teske [FST10]. Un des critères de classification est la valeur du *discriminant* de la courbe elliptique. Pour une courbe définie sur un corps $\mathbb{F}_q$, on écrit la factorisation en facteur non carré $t^2 - 4q = -D\gamma^2$, avec q qui détermine le corps fini et t la trace de la courbe elliptique sur $\mathbb{F}_q$. Le discriminant est le nombre D.

Des recherches minutieuses de cas particuliers pour de valeurs précises de petits discriminants, par exemple $D = 1, 2, 3, 5$, aboutirent à d'intéressants mais rares cas particuliers, parfois proches a posteriori de résultats que pourraient donner des variantes de la méthode de Brezing-Weng. Galbraith, McKee et Valença [GMV07] amorcent cette méthode et décrivent d'autres courbes, généralisant les constructions de [MNT00]. En 2007, Freeman [Fre06] exhibe une famille de courbes de discriminant $D = 5$ et de degré de plongement $k = 10$, avec la possibilité de trouver des exemples de courbes d'ordre premier, ce qui est très recherché. Et bien sûr, il faut mentionner la construction devenue incontournable de courbes avec un discriminant D égal à 3 et un degré de plongement $k = 12$ de Barreto et Naehrig [BN05]. Ce degré de plongement 12 combiné avec la possibilité de trouver facilement, en quelques secondes, une courbe d'ordre premier, font de cette famille de courbes la plus populaire actuellement pour instancier un protocole utilisant un couplage.

Jusqu'en 2012, les courbes supersingulières en petite caractéristique, de degré de plongement 4 sur $\mathbb{F}_{2^n}$ et 6 sur $\mathbb{F}_{3^m}$, étaient aussi très étudiées, notamment pour des implémentations matérielles (assembleur, FPGA...). Depuis les récents records de calculs de logarithme discret dans des corps finis en petite caractéristique, dont les corps de plongement des couplages de la forme $\mathbb{F}_{2^{4\cdot n}}$ et $\mathbb{F}_{3^{6\cdot n}}$ sont des applications directes, ces courbes sont à proscrire en cryptographie utilisant des couplages.

Bibliothèques de calculs de couplages

Ce paragraphe liste quelques bibliothèques de calculs implantant des couplages. Tout d'abord, Magma [BCP97] depuis plusieurs années contient un calcul de couplage de Weil. Depuis 2011, un couplage de Weil et de Tate sur des courbes elliptiques sur des corps finis est disponible. À ce jour, dans la version de 2013, des couplages η_T sur des courbes supersingulières en petite caractéristique, et des couplages *ate* en grande caractéristique, sont disponibles. Grâce aux correspondances entre couplage de Tate, ate et optimal ate, il est possible de tout calculer avec Magma. C'est très pratique pour générer des vecteurs d'entrée-sortie pour tester du code en développement. La bibliothèque Pari [BC55] écrite en C et développée à Bordeaux en France propose aussi, depuis 2011, des calculs de couplages possibles pour des tailles cryptographiques.

La première bibliothèque de calculs optimisés de couplages, PBC, fut développée par Benjamin Lynn en C et est toujours disponible [Lyn14]. Néanmoins ses performances ne sont pas optimales pour tous les couplages.

Une deuxième librairie performante, Miracl, fut développée en Irlande par Michael Scott et ses collaborateurs [Sco11]. Cette bibliothèque, écrite en C++, était très utilisée à des fins de recherche et très performante. Elle permettait également la génération de courbes appropriées aux couplages avec la méthode de Cocks-Pinch et plus généralement, le calcul de polynômes de classes de Weber, pour des discriminants allant jusqu'à 10^9, ce qui était une belle performance. En 2011, cette bibliothèque est devenue payante, son contributeur historique, Michael Scott, ayant fondé une start-up, Certivox, promouvant l'utilisation des couplages dans la vie quotidienne.

Une nouvelle librairie également écrite en C++ a pris le relais de Miracl ces dernières années. Il s'agit de Relic [AG35], développée par Diego Aranha et son équipe. Cette librairie détient certains des derniers records de calculs de couplages et présente l'avantage d'être, pour l'instant, sous licence permettant son utilisation gratuite à des fins de recherche.

Dernièrement, une équipe de l'Université de Tsukuba au Japon a lancé la bibliothèque Tepla [Lab10]. Cette dernière-née propose l'implémentation optimisée en C de couplages sur des courbes de Barreto-Naehrig.

En ce qui concerne les librairies propriétaires (industrielles), les équipes de Microsoft Research de Seattle disposent d'une excellente bibliothèque de calculs sur courbes elliptiques, et notamment d'opti-

misations spécifiques à l'assembleur ARM, très populaire depuis l'émergence des smartphones.

Travaux réalisés : contexte et survol

Les travaux réalisés dans cette thèse s'inscrivent dans la continuité d'un stage de Master 2 effectué en 2010 au laboratoire Chiffre. Ce stage consistait à développer une bibliothèque de calculs de couplages en vue d'une utilisation pour de la diffusion chiffrée (*broadcast* en anglais). Ce besoin s'inscrivait dans le cadre d'un projet ANR de diffusion chiffrée [ENSC+09]. Par la suite, il s'agissait d'améliorer les performances de cette bibliothèque, pour atteindre celles de l'état de l'art. Pour cela, les formes les plus récentes de couplages (optimal ate) furent étudiées. Ensuite, un couplage s'inscrit toujours dans le cadre d'un protocole. Depuis 2005, de nouveaux protocoles font appel à des couplages bilinéaires sur des groupes d'ordre composé, typiquement un module RSA, et non plus simplement sur des groupes d'ordre premier. Il s'agit de choisir soigneusement les courbes elliptiques et les types de couplages qui correspondent à ces nouveaux protocoles.

Réciproquement, les couplages les plus rapides, sur des groupes d'ordre premier, sont des couplages asymétriques. Autrement dit, la représentation des deux groupes de départ, $\mathbb{G}_1$ et $\mathbb{G}_2$, est différente. En particulier, un élément du groupe $\mathbb{G}_2$ bien souvent prend au moins deux fois plus de place qu'un élément du groupe $\mathbb{G}_1$. Or bien souvent les protocoles sont écrits dans le cadre spécifique de couplages symétriques, où $\mathbb{G}_1$ et $\mathbb{G}_2$ sont explicitement isomorphes. Il s'agit alors de choisir quels éléments du protocole seront en fait tirés du premier groupe, du deuxième groupe, et lesquels ont besoin d'une double représentation. Le protocole sera alors réécrit en conséquence. La traduction de protocoles peut aussi se faire de manière bien plus spécifique, en exploitant de nouvelles propriétés et hypothèses de sécurité, disponibles uniquement dans le cadre de couplages asymétriques, comme l'hypothèse SXDH mais cela sort du cadre de cette thèse.

Une autre partie de cette thèse s'intéresse à la construction de courbes appropriées aux couplages. Les courbes elliptiques furent proposées en 1985 indépendamment par Neal Koblitz et Victor Miller. Il est possible de construire un groupe d'ordre premier dans lequel le problème du logarithme discret est difficile. On peut alors baser un cryptosystème à base de DLP sur des courbes elliptiques. L'avantage est la robustesse des courbes elliptiques face au problème du logarithme discret. En effet, hormis pour quelques cas particuliers bien identifiables, il n'existe que des attaques génériques. Ainsi, les paramètres restent petits comparés aux paramètres des corps premiers seuls, bien plus élevés pour un même niveau de sécurité.

En 1989, Koblitz propose d'utiliser comme groupe la jacobienne d'une courbe hyperelliptique. C'est une généralisation des courbes elliptiques. Cette fois-ci, les points de la courbe ne forment pas directement un groupe, c'est pourquoi la structure intermédiaire de la jacobienne intervient. Néanmoins, cette généralisation a ses limites. En effet, pour des courbes de genre plus grand que 3, les attaques génériques contre le problème du logarithme discret connaissent des améliorations.

De même, il est possible de généraliser les courbes elliptiques sur des corps premiers aux courbes elliptiques sur des extensions de corps. Encore une fois, cette généralisation a ses limites. Il est possible, via la méthode de la restriction de Weil, d'obtenir une correspondance entre le groupe d'une courbe elliptique (donc de genre 1) définie sur une extension de degré n d'un corps fini, et un sous-groupe de la jacobienne d'une courbe de genre n. Or le paragraphe précédent exposait les vulnérabilités des courbes de genre supérieur à 3. Ainsi, il est préférable de s'en tenir aux courbes elliptiques définies sur des corps premiers ou des extensions quadratiques de corps premiers. Il est également possible de manipuler des courbes en petite caractéristique, autrement dit, définies sur $\mathbb{F}_{2^\ell}$ ou $\mathbb{F}_{3^\ell}$. Afin d'éviter une attaque par restriction aux scalaires de Weil comme expliquée plus haut, le degré de l'extension ℓ est choisi premier. Ces courbes elliptiques ou de genre 2 en petite caractéristique sont bien appropriées pour des implémentations matérielles et présentent de très bonnes performances. Jusqu'à maintenant, seules les attaques génériques

s'appliquent.

En ce qui concerne les performances des courbes elliptiques et de genre 2, on recherche des améliorations de l'arithmétique des corps finis sur lesquels sont définis les courbes, des améliorations de la loi de groupe (addition et doublement), des améliorations de la multiplication scalaire, notée $[m]P$ sur une courbe elliptique et $[m]\mathcal{D}$ sur une jacobienne. Tout ceci est utilisé dans des protocoles reposant sur le logarithme discret.

De plus face à la technicité grandissante des attaques *par canaux auxiliaires*, ou *side-channel attacks*, on s'intéresse aux courbes sur lesquelles les multiplications scalaires peuvent s'effectuer de manière régulière, tout en présentant de bonnes performances. L'une des pistes très développée est la recherche de lois d'additions unifiées sur les courbes, autrement dit, addition et doublement s'effectuent avec une seule formule, ou du moins avec des formules ayant le même nombre d'opérations.

En cryptographie bilinéaire, on recherche des améliorations de calculs de couplages et également des courbes appropriées aux couplages (*pairing-friendly curves*). De telles constructions sont loin d'être triviales et on manque de diversité de choix de courbes présentant des paramètres de taille optimale.

Implémentation de couplages

Une partie de cette thèse fut consacrée à l'implémentation en langage C dans la bibliothèque du laboratoire Chiffre de fonctions de couplages. L'arithmétique des corps finis premiers était déjà disponible, de même que l'arithmétique de courbes elliptiques définies sur ces corps premiers. Des optimisations en assembleur pour les processeurs intel x86-64 furent apportées en 2011 par Frédéric De Portzamparc. Les corps binaires furent développés par Thomas Prest en 2012. A l'issue de mon stage en 2010, des fonctions de couplages sur des courbes supersingulières en grande caractéristique étaient disponibles, ainsi qu'une première version, assez peu optimisée, de couplage de Tate sur une courbe de Barreto-Naehrig. Par la suite, j'ai développé des fonctions de couplage de type ate et optimal ate, toujours sur des courbes de Barreto-Naehrig. Ces courbes sont en effet parmi les plus efficaces. Ces versions *ate* et *optimal ate* ne sont pas applicables aux courbes supersingulières utilisées.

Jusqu'en 2012, les courbes appropriées aux couplages en petite caractéristique ($p = 2, 3$) étaient aussi assez populaires. Seules les constructions de courbes supersingulières étaient alors disponibles. Une construction de courbe ordinaire de petit degré de plongement était inconnue. Les méthodes de Cocks-Pinch, Brezing-Weng ou encore de Dupont, Enge et Morain ne s'appliquant pas. Des méthodes très efficaces de doublement en caractéristique 2 et de triplement en caractéristique 3 furent développées. Un projet de bibliothèque complète avait même commencé à l'université de Tsukuba au Japon, dédiée à la caractéristique 3. Il se trouve qu'entre l'hiver 2012 et l'été 2013, de nombreux records de calculs de logarithmes discrets furent annoncés successivement. Les cryptosystèmes basés sur des extensions de corps de caractéristique 2 et 3, typiquement $\mathbb{F}_{2^n}$ et $\mathbb{F}_{3^\ell}$ sont définitivement à éviter. Seules les courbes elliptiques *ordinaires*, de degré de plongement bien trop grand (de l'ordre de 2^ℓ ou 3^ℓ) pour qu'un calcul de couplage soit envisageable, ne sont pas concernées par ces attaques. Toutes les fonctions de couplages développées dans la bibliothèque du laboratoire Chiffre sont sur des corps de grande caractéristique uniquement.

Contexte industriel

La laboratoire Chiffre fait partie du service SCC (Service Cryptologie et Composants) lui-même dépendant du service SSI (Sécurité des Systèmes d'Information) de Thales Communications & Security. Il est composé de spécialistes en mathématiques et algorithmie appliquée à la cryptographie. Ses principales missions sont la réalisation d'études en amont en cryptographie fondamentale, l'intégration d'algorithmes et de mécanismes cryptographiques définis par la DGA-MI dans les composants gouvernementaux, la réalisation de dossiers cryptographiques sur des équipements ou de systèmes et la participation à des projets de recherches collaboratifs.

La réalisation d'un produit ou équipement de sécurité pour la DGA suit un cycle de développement qu'on peut brièvement schématiser ainsi : la DGA définit ses besoins et établit un cahier des charges. Lorsque Thales remporte l'appel d'offres, le laboratoire Chiffre intervient au niveau des composants cryptographiques du produit. Dans la LibCryptoLCH, une branche est développée pour les besoins spécifiques de chaque affaire.

Les paramètres cryptographiques utilisés, par exemple les nombres premiers et les paramètres de courbes elliptiques, sont définis par la DGA. C'est pourquoi il est nécessaire que tout code développé puisse prendre en compte tous les paramètres possibles définissables par la DGA. Ainsi, un haut niveau de généricité est recherché à toutes les étapes du développement de la LibCryptoLCH.

Les équipements de sécurité commandés la DGA nécessitent un contrôle et une validation par un troisième intervenant extérieur : l'agence nationale de la sécurité de systèmes d'information (ANSSI). Un service très actif de spécialistes en cryptographie fait partie de l'ANSSI. Ce service est également partagé en un laboratoire de cryptographie et un laboratoire de composants. L'ANSSI délivre quatre labels aux différents produits qui lui sont soumis :

- la certification Critères Communs (CC) ;
- la certification de sécurité de premier niveau (CSPN) ;
- la qualification d'un produit ;
- l'agrément (label réservé aux produits destinés à protéger les informations relevant de la défense et de la sécurité nationale).

Dans le cadre précis de cette thèse, il est probable que le code développé pour les calculs de couplages soit prochainement utilisé dans un produit commercial Thales, par exemple dans une version ultérieure de Teopad (environnement sécurisé sur smartphones et tablettes sous Android). Ce produit fera alors l'objet d'une demande de CSPN. Il est envisageable que le code serve aussi un jour dans un équipement qui nécessitera un agrément de l'ANSSI et utilisera des paramètres définis par la DGA. En ce qui nous concerne, à postériori les corps de petite caractéristique se sont révélés vulnérables à de fulgurantes attaques, et ce depuis début 2013. Or il y a à peine cinq ans, les courbes supersingulières en caractéristique 2 et 3 étaient très populaires, très étudiées et plusieurs équipes développaient des calculs de couplages optimisés et très performants sur ces courbes. Les ingénieurs de la DGA ayant déjà depuis longtemps un avis mitigé sur les corps de petite caractéristique, dès le début de cette thèse les courbes supersingulières en petite caractéristique ont été écartées.

Implémentation pour un protocole de broadcast dans le cadre d'un projet ANR [DGSLB12]

Dans le cadre d'un projet ANR en commun avec Thales, Nagra, CryptoExperts, Paris 8 et l'ENS, une implémentation complète d'un protocole de broadcast et de ses améliorations a été réalisée sur trois ans au laboratoire Chiffre. Renaud Dubois, Marine Sengelin, Romain Perez et Margaux Dugardin ont pris part à ce projet. Une première étape fut d'identifier les protocoles apportant des réponses satisfaisantes. Il est apparu dès le début du projet que les protocoles à base de couplages procuraient des solutions nouvelles et très intéressantes en termes d'efficacité, de capacité de révocation d'utilisateurs compromis, et de bande-passante, contrairement aux solutions à base d'arbres de clés symétriques. Un premier stage lié à ce projet à Thales consita à développer un module de calculs de couplages performant. Puis, les stages suivants ont consisté à développer les protocoles retenus [BGW05] et [PPSS13], améliorer leurs performances, les insérer dans un dispositif général de broadcast, et réaliser un prototype où un centre émetteur (un PC) envoie du contenu via une antenne wifi à des récepteurs, en l'occurence des smartphones. Le centre émetteur peut révoquer à tout moment n'importe quel récepteur de façon individuelle.

Les deux variantes du protocole décrites en [BGW05] utilisent un couplage symétrique. Les travaux présentés à Pairing 2012 à Cologne présentent une application du protocole [BGW05] utilisé avec un couplage asymétrique sur des courbes de Barreto-Naehrig. Sur ces courbes, les éléments du groupe $\mathbb{G}_2$ prennent deux fois plus de place que ceux du groupe $\mathbb{G}_1$. Néanmoins, avec ce couplage asymétrique, la représentation d'éléments du groupe $\mathbb{G}_1$ est six fois plus petite qu'avec un couplage symétrique (sur des courbes supersingulières en grande caractéristique, de degré d'immersion $k = 2$). Pour adapter le protocole à ce couplage asymétrique, on identifie quels éléments sont affectés à $\mathbb{G}_1$, respectivement $\mathbb{G}_2$ et lesquels nécessitent d'être dupliqués dans les deux groupes. Finalement, même lorsqu'il faut dupliquer certains éléments (des paramètres publics par exemple), tout est plus intéressant et plus compact avec ce couplage asymétrique puisque les éléments de $\mathbb{G}_1$ et $\mathbb{G}_2$ sont trois ou six fois plus économiques en espace mémoire qu'avec un couplage symétrique.

Dans cet article une stratégie de pré-calculs est développée afin de réduire le coût du déchiffrement au niveau de chaque récepteur. En effet, dans la deuxième version du protocole original, ce coût est linéaire en le nombre d'utilisateurs autorisés à accéder au contenu diffusé. Un arbre de précalculs permet de faire

baisser cette complexité. Enfin, des temps de calculs sont donnés. Le temps nécessaire au déchiffrement pour 50 000 utilisateurs est de 1.44 s sur un smartphone Samsung Galaxy équipé d'un processeur ARM Cortex A8 (architecture 32 bits). Pour 200 000 utilisateurs, le temps de déchiffrement passe la barre des 2 secondes avec un temps de 2.08 s. Pour 5 millions d'utilisateurs, le temps de déchiffrement est de 6 secondes. Les utilisateurs actuels de smartphones ne peuvent pas accepter un tel temps de latence. De meilleurs temps sont à espérer avec l'introduction de parties critiques du code en assembleur. Par ailleurs, les constructeurs de smartphones et de processeurs à faible consommation d'énergie spécifiques aux systèmes embarqués sont très actifs et proposent tous les six mois de nouveaux produits toujours plus performants. On pourra noter la sortie prévue en 2014 d'une nouvelle série de processeurs ARM bénéficiant d'une architecture 64 bits ce qui permettra de gagner sensiblement en performance.

Ces travaux furent présentés en 2012 à la conférence Pairing [DGSLB12]. La conférence s'est tenue du 16 au 18 mai 2012 à Cologne en Allemagne.

Implémentation de couplages sur des courbes d'ordre composé et comparaison avec les courbes d'ordre premier [Gui13]

En 2005, Boneh, Goh et Nissim proposent un cryptosystème partiellement homomorphe. Le chiffrement homomorphe, brièvement, est la propriété de pouvoir faire des opérations sur les chiffrés, sans avoir à déchiffrer. Un objectif majeur est de pouvoir à la fois additionner et multiplier des chiffrés, ou encore de pouvoir effectuer des opérations binaires comme le Xor sur les chiffrés, sans avoir à déchiffrer. Actuellement, les pistes les plus prometteuses dans ce domaine sont basées sur les réseaux euclidiens. Dans cet article de 2005, les auteurs proposent un moyen d'additionner des chiffrés et de les multiplier une fois. Il est possible de continuer à additionner (avec retenue, ce n'est pas un Xor) les chiffrés après une multiplication. L'addition homomorphe est obtenue par la propriété de la multiplication qui devient une addition dans les exposants, autrement dit, pour un générateur g et deux messages m_1, m_2, on a $g^{m_1} \cdot g^{m_2} = g^{m_1+m_2}$. La multiplication est obtenue avec un couplage. Afin d'avoir de bonnes propriétés, il n'est pas possible d'instancier tel quel ce protocole. Les auteurs utilisent alors non pas un couplage sur un groupe d'ordre premier (cas classique) mais un couplage sur un groupe dont l'ordre est un module RSA dont la connaissance de la factorisation est une trappe. Les choix d'instanciation dans ce cas précis sont alors bien différents. Tout d'abord, les tailles des paramètres sont directement dictées par ce module RSA. Ce module fait par exemple de 1024 à 3072 bits (pour un niveau de sécurité de très faible à standard). Ainsi, il n'est pas nécessaire d'avoir un degré de plongement élevé car le corps de plongement fait déjà le double du module RSA, avec un degré de plongement égal à 1 ou 2. De simples constructions de courbes supersingulières, finalement les plus utilisées au début des années 2000, sont intéressantes dans ce cas de figure, notamment pour leur simplicité. Après implémentation, de tels couplages sur des courbes supersingulières de degré de plongement égal à 2 et présentant un sous-groupe dont l'ordre est un module RSA donné, il se trouve que les couplages s'effectuent en près de 400 ms pour un module RSA de 2048 bits et 1300 ms pour un module de 3072 bits. C'est très lent. Pour comparaison, un couplage optimal ate sur une courbe de Barreto-Naehrig, à comparer avec un module RSA de 3072 bits, se fait en 5 ms sur le même processeur. Pour finir, le protocole de Boneh, Goh et Nissim était intéressant et lança l'utilisation de groupes d'ordre composé, mais en pratique, l'étape de déchiffrement n'était possible en temps raisonnable que pour quelques bits de données, et non pas quelques octets.

En 2009, Freeman proposa une conversion de ces protocoles pour n'utiliser que des groupes bilinéaires d'ordre premier et ainsi pouvoir se ramener aux implémentations records de couplages (en milisecondes). L'implémentation montre que la conversion de Freeman est jusqu'à 250 fois plus rapide que la version initiale. De plus, les paramètres ont des tailles plus raisonnables. Ces premiers travaux de traduction de protocoles furent continués par Lewko en 2012. Là encore, il y a une différence de temps d'exécution entre les versions initiales et les versions converties assez importante, les choix d'instanciation sont alors évidents : les couplages sur des groupes bilinéaires d'ordre composé sont à éviter.

Ces travaux furent présentés à la conférence ACNS en 2013 qui s'est tenue à Banff en Alberta au Canada, du 25 au 28 juin 2013. Les résultats sont parus dans les actes de la conférence [Gui13] et sont disponibles en ligne : `http://eprint.iacr.org/2013/218`.

Recherches de nouvelles propriétés sur des courbes de genre 1 et 2

Comptage de point sur deux familles de courbes de genre 2 et constructions pour les couplages [GV12]

Afin de diversifier un peu les choix possibles pour implémenter un cryptosystème basé sur des courbes de genre 2, on s'est intéressé à deux familles de courbes, déjà étudiées en mathématiques. En 2009, Satoh [Sat09] introduit la famille $\mathcal{C}_1 : y^2 = x^5 + ax^3 + bx$ en cryptographie. Puis en 2011 avec Freeman [FS11], ils étudient également la famille $\mathcal{C}_2 : y^2 = x^6 + ax^3 + b$. Ces deux familles de courbes présentent une propriété particulière qui permet d'avoir une méthode très efficace de comptage de points. En effet, ces deux familles de courbes présentent une jacobienne qui devient isogène au produit de deux courbes elliptiques, elles-mêmes isogènes, sur une extension de petit degré (divisant 8 pour la première famille et divisant 6 pour la deuxième).

Ainsi via cette isogénie, il est possible de déterminer l'ordre de la jacobienne des courbes $\mathcal{C}_1$ et $\mathcal{C}_2$ sur une extension de corps, en faisant appel simplement aux algorithmes de comptage de points sur les courbes elliptiques correspondantes. La difficulté du genre 2 est évitée. Ensuite il s'agit de déduire l'ordre de la jacobienne sur le corps de base en fonction de l'ordre obtenu sur une extension. Satoh donna une première méthode [Sat09]. L'article [GV12] affine les formules de Satoh et présente une méthode similaire pour la deuxième famille de courbes de genre 2. De plus les formules explicites proposées permettent de mettre en lumière d'autres propriétés de ces courbes de genre 2. Dans un premier temps, ces formules sont utilisées pour obtenir des constructions appropriées aux couplages. Quelques nouvelles familles de courbes sont proposées. Néanmoins pour l'instant, les propositions en genre 2 ne sont pas compétitives aux possibilités existantes en genre 1 comme les courbes de Barreto-Naehrig.

La qualité des courbes utilisées pour implémenter un couplage se mesure au ratio ρ entre la taille du sous-groupe premier de la courbe (ou la jacobienne en genre supérieur à 1) qui présente un degré de plongement déterminé k, et la taille totale sur $\mathbb{F}_q$ du groupe de la courbe correspondante (ou de la jacobienne). En genre 1, une courbe elliptique définie sur un corps $\mathbb{F}_q$ a pour ordre $q + 1 - t$, avec la trace t qui vérifie la borne $t^2 \leqslant 4q$. Si la courbe est d'ordre premier m, ce nombre premier aura la même taille que q, i.e. $\log m = \log q$ (à un bit près) et on aura $\rho = 1$ ce qui est optimal. C'est la cas pour les courbes de Barreto-Naehrig. Pour l'instant, il n'existe pas de constructions de courbes de genre 2 ordinaires avec $\rho < 2$ autrement dit, pour l'instant les courbes de genre 2 ordinaires ne sont pas compétitives dans le contexte des couplages.

Ces travaux furent présentés en 2012 à la conférence Pairing [GV12]. La conférence s'est tenue du 16 au 18 mai 2012 à Cologne en Allemagne.

Deux nouvelles familles de courbes de genre 1 et 2 présentant des endomorphismes intéressants [SS13]

Cet article reprend les deux familles de courbes de genre 2 de l'article précedent [GV12]. L'ordre des jacobiennes correspondantes se calcule facilement via un calcul de trace de courbe elliptique. De plus, les jacobiennes sont munies naturellement d'un endomorphisme facilement calculable. Cet article s'intéresse à construire un second endomorphisme afin d'effectuer plus rapidement des multiplications scalaires avec la méthode de Gallant, Lambert et Vanstone [GLV01]. Il est possible de construire un endomorphisme approprié pour cette méthode sur des courbes elliptiques. Puisque les jacobiennes sont isogènes au produit de deux courbes elliptiques, l'idée est de construire un endomorphisme sur ces courbes elliptiques puis de le ramener sur les jacobiennes par l'isogénie. Les calculs sont un peu techniques mais les résultats concluants. En fixant un petit discriminant D pour les courbes elliptiques, on construit un second endomorphisme sur les jacobiennes qui correspond à la multiplication complexe des courbes elliptiques. Les deux endomorphimes des jacobiennes ont des valeurs propres suffisamment différentes pour qu'une méthode GLV en dimension quatre s'applique.

De plus un second endomorphisme est aussi disponible sur les familles de courbes elliptiques correspondantes. Là aussi une multiplication scalaire avec la méthode GLV en dimension quatre est possible. On peut voir ces deux familles de courbes elliptiques comme une généralisation des travaux de Longa et Sica [LS12].

Ces travaux furent présentés au workshop ECC 2013 à Leuven en Belgique et publiés à la conférence Asiacrypt 2013 qui aura lieu à Bengalore en Inde du 1er au 5 décembre 2013.

Autres travaux : arithmétique d'extensions de corps de degré 5 [MGI11]

En tout début de thèse, j'ai eu la chance de participer à un article déjà bien avancé. J'ai travaillé avec Nadia El Mrabet, maître de conférences en informatique à l'université de Paris 8, et Sorina Ionica, post-doc à l'école polytechnique. Leurs travaux proposent des formules de multiplications efficaces pour les extensions de corps (en grande caractéristique) de degré 5. Étant donné un corps fini $\mathbb{F}_q$ et une extension de degré 5 de celui-ci, on représente les éléments de $\mathbb{F}_{q^5}$ par des polynômes de degré 4. Lorsque $p \equiv 1 \bmod 5$ on peut représenter l'extension à l'aide d'un binôme irréductible sur $\mathbb{F}_q$ de la forme $X^5 - \alpha$, avec α aussi petit que possible. Il existe plusieurs choix pour les formules de multiplications. Si les multiplications et les additions dans $\mathbb{F}_q$ ont sensiblement le même coût (ce qui peut arriver si q est de la taille d'un mot machine et que le programmeur a accès aux instructions assembleur), alors la méthode dite *schoolbook*, ou élémentaire, est la plus simple et la plus appropriée. Elle coûte 25 multiplications dans $\mathbb{F}_q$.

Si le coût d'une multiplication devient prépondérant devant une addition, il devient intéressant de regrouper et factoriser les multiplications dans $\mathbb{F}_q$. Cette méthode est bien connue pour les extensions quadratiques (méthode de Karatsuba). Peter Montgomery développa une telle méthode pour les extensions de degré 5, 6 et 7. Sa proposition pour les extensions de degré 5 requiert 13 multiplications dans $\mathbb{F}_q$, pour un sur-coût de additions (62 additions au total).

Une troisième méthode de regroupement des multiplications consiste à utiliser l'interpolation polynomiale. Cette méthode porte le nom de Toom-Cook. Les éléments de l'extension $\mathbb{F}_{q^5}$ sont, en tant que polynômes, évalués en des points bien choisis : $\{0, \infty, -1, 1, 2\}$. Cela coûte seulement un nombre limité d'additions. Puis, les coefficients du résultat sont reconstruits par interpolation. Cette méthode nécessite moins de multiplications que la précédente. par contre, des divisions par de petites constantes, ici 2 et 3, apparaissent dans les formules. Si ces divisions sont suffisamment efficaces, c'est le cas si elles ne coûtent que 2 additions par exemple, alors cette méthode de Toom-Cook est plus efficace que la précédente.

Le troisième co-auteur d'une pré-publication de ces travaux était Nicolas Guillermin, ingénieur de l'armement au Celar. Mon travail fut de reprendre ses travaux d'implémentation et de les poursuivre, en C, afin d'avoir de bonnes mesures de performances des méthodes proposées dans l'article. Les résultats montrent que pour des extension de degré 5 sur un corps premier de 768 bits, le gain avec la dernière méthode est de 8 % sur un processeur Intel 32 bits. Sur un corps de 1024 bits pour un processeur de mots de 64 bits, la dernière méthode devient plus efficace au delà de 1024 bits. Le gain est de 9 % sur un corps premier de 1536 bits. Ces travaux furent publiés en 2011 à Africacrypt [MGI11]. La conférence a eu lieu du 4 au 10 juillet 2011 à Dakar au Sénégal.

Publications

Actes de conférences

[1] Guillevic, A., Ionica, S. : Four Dimensional GLV via the Weil Restriction. In : Sako, K., Sarkar, P. (eds.) *ASIACRYPT 2013 PART I*. LNCS, vol. 8269, to appear. `http://eprint.iacr.org/2013/311`

[2] Guillevic, A. : Comparing the Pairing Efficiency over Composite-Order and Prime-Order Elliptic Curves. In : Jacobson, M. et. al (eds.) *ACNS 2013*. LNCS, vol. 7954, pp. 357-372. `http://eprint.iacr.org/2013/218`

[3] Guillevic, A., Vergnaud, D. : Genus 2 Hyperelliptic Curve Families with Explicit Jacobian Order Evaluation and Pairing-Friendly Constructions In : Abdalla, M., Lange, T. (eds.) *Pairing 2012*. LNCS, vol. 7708, pp. 234-253. `http://eprint.iacr.org/2011/604`

[4] Dubois, R., Guillevic, A., Sengelin Le Breton, M. : Improved Broadcast Encryption Scheme with Constant-Size Ciphertext. In : *Pairing 2012*. LNCS, vol. 7708, pp. 196-202. `http://eprint.iacr.org/2012/370`

[5] El Mrabet, N., Guillevic, A., Ionica, S. : Efficient Multiplication in Finite Field Extensions of Degree 5. In : Nitaj, A., Pointcheval, D. (eds.) *AFRICACRYPT 2011*. LNCS, vol. 6737, pp. 188-205.

Brevets

[2012] Inventeurs : A. Guillevic, R. Dubois et D. Vergnaud. **Intitulé : Procédé de génération d'une clé de session à partir d'une clé secrète**, 2 brevets déposés. Ces deux brevets concernent la délégation partielle de calculs de couplages dans un système embarqué, par exemple entre processeur et carte micro-SD de smartphone.

Pré-publications

[2013] Dubois, R., Dugardin, M., Guillevic, A. : Golden Sequence for the PPSS Broadcast Encryption Scheme with an Asymmetric Pairing. Cryptology ePrint Archive, Report 2013/477, `http://eprint.iacr.org/`.

Présentations

Exposés invités

[Sept. 2013] Four Dimensional GLV via the Weil Restriction. ECC 2013 workshop, invited talk. Leuven, Belgium

Présentations en conférences

[Juin 2013] Comparing the pairing efficiency over composite-order and prime-order elliptic curves ACNS 2013 Conference Banff, Alberta, Canada

[Oct. 2012] Pairing efficiency over composite and prime-order elliptic curves Journées Codage et Cryptographie 2012 Dinard, France

[Sept. 2012] Pairing efficiency over composite and prime-order elliptic curves YACC 2012 Conference Porquerolles, France

[Mai 2012] Improved broadcast encryption scheme with constant-size ciphertext Industrial track, Pairing 2012 Conference Cologne, Germany

[Mai 2012] Genus 2 hyperelliptic curve families with explicit Jacobian order evaluation and pairing-friendly constructions Pairing 2012 Conference Cologne, Germany

Contents

Introduction

Cryptography was until the middle of the 20th century the art of encrypting secret data for secure storage or secure communications. Nowadays cryptography consists in ensuring confidentiality of the communication, integrity of the encrypted data and authentication of the involved parties (e.g. sender, receiver). These functionalities are used everywhere, everyday, to connect securely to our mailbox, to access restrained services on the Internet, for online banking, etc.

For a secure telecommunication, two participants (also known as Alice and Bob) first need to share some secret information indicating how to encrypt the message. In cryptography this is formalized as *sharing a secret key*. This key will parameterize as input the encryption algorithm Alice and Bob have chosen. Alice wants to send securely her message to Bob. She encrypts her message with their secret key. Then Alice sends the encrypted data to Bob through an insecure channel. Bob can decipher with the same shared secret key at the other side of the channel. To perform the encryption operation, cryptographers design encryption algorithms satisfying some precise properties. This is known as symmetric cryptography.

Before sending her message as described above, Alice and Bob need to share some secret information or secret key. This means either they meet physically somewhere to exchange this secret key, or they can use a protocol using *asymmetric cryptography* to agree remotely on some secret data through an insecure channel. An aspect of this notion is commonly sketched as follows. Bob sends to Alice an open lock. He is the only one to have the corresponding key. Alice uses Bob's lock to secure the sensitive data then sends it (closed) back to Bob. Bob then uses his secret key to unlock Alice's data. This is an analog on the real life of a cryptographic scheme known as *public key encryption*. In 1978, Rivest, Shamir and Adelman proposed the well-known RSA scheme providing public-key encryption. Its security relies on the factorization problem: given a large modulus $N = pq$ of two prime numbers, it is very difficult to recover the two prime factors. This is still one of the most widely used cryptosystems in the world.

Another way for Alice and Bob to exchange remotely some secret key is to use a *key agreement protocol*. In 1976, Diffie and Hellman proposed such a scheme (DH-scheme in the following). Their construction handles the keys as elements in a finite field where the exponentiation (computing g^x from g an element in the finite field and x an integer) is easy to compute (on a PC, laptop, smartphone) but impossible to invert in reasonable time (a month, a couple of years, ten years...) which means, given g and g^x, this is infeasible to compute x in reasonable time. This is known as the Discrete Logarithm Problem (DLP).

These two examples, RSA and DH schemes, are now very common in asymmetric cryptography. Their underlying mathematical candidates for one-way functions are widely studied and attacked, but not yet broken. Their weaknesses are well-known, rare and limited. Furthermore there exist simple and easy-to-implement countermeasures. A common countermeasure is to enlarge the key and parameter sizes. The time needed to solve e.g. an instance of the DLP will grow accordingly. However this reduces the efficiency of both encryption and decryption steps (they are slower and Alice waits more time for checking her mailbox). This also augments the bandwidth consumption or the place required for a secure storage of encrypted data. That is why cryptographers are looking for other instantiation of these cryptosystems. For example in DLP-based cryptography, we are always looking for other candidates of groups where the DLP is intractable, i.e. where the function $(g, x) \mapsto g^x$ is very difficult to invert. Such a function is also called a one-way function.

In 1985, Koblitz and Miller introduced from algebraic geometry the use of *elliptic curves* instead of finite fields for DLP-based cryptosystems, then hyperelliptic curves (a generalization of elliptic curves) in 1989. This thesis is mostly about elliptic and hyperelliptic curves. Moreover a second good candidate of one-way function is available on curves. We can combine this second function with the exponentiation

function (used e.g. in the DH scheme) to achieve interesting new properties in cryptosystems. Roughly speaking, a pairing is a map $e : (g, h) \mapsto e(g, h)$ which is bilinear in the sense that $e(g_1 \cdot g_2, h) = e(g_1, h) \cdot e(g_2, h)$ and the same property holds with respect to the right-hand side inputs. This map is not invertible in reasonable time and computer resources. It is computationally infeasible in reasonable time, given a pairing output f and a pairing input g, to compute an input value h such that $e(g, h) = f$. The pairing definition and properties are introduced in Chapter 1 and an efficient implementation is provided in Chapter 3.

We give a few examples of interesting new cryptographic schemes based on pairings. We can cite the Identity-Based encryption scheme (IBE) of Boneh and Franklin [BF01]. With this, Alice can use as Bob's public key simply Bob's email address. She only needs to register to the service an receive at the beginning a secret key (stored e.g. on a smartcard). This simplifies considerably the secured telecommunications. We can also highlight the tri-partite key agreement protocol of Joux [Jou00, Jou04] as one of the first applications of pairings in cryptosystem design. This is a generalization of the Diffie-Hellman key agreement scheme. In the last decade, various encryption schemes, broadcast encryption schemes and signature schemes where proposed, based on bilinear maps. We have chosen to study and implement a broadcast encryption scheme and a hierarchical identity-based encryption scheme using pairings. This work is presented in the second part of Chapter 3. The design of new pairing-based protocols and their implementation is a very active area of research in cryptography, as shown by the programs of the main cryptology conferences.

We now give the outline of this thesis. The preliminaries on elliptic and hyperelliptic curves are introduced in Chapter 1, followed by the pairing definition and properties. Chapter 2 focuses on efficient arithmetic of two families of elliptic curves and hyperelliptic curves. We also investigate pairing-friendly constructions of curves from these families. Finally in Chapter 3 we present an efficient implementation of pairings, of the broadcast encryption scheme of Boneh, Gentry and Waters [BGW05] and its improvement thanks to Phan, Pointcheval, Strefler and Shahandashti [PPSS12, PPSS13] and also a compared implementation of different variants of the hierarchical idendity-based encryption scheme of Lewko and Waters [LW11, Lew12].

Chapter 1

Background on elliptic and hyperelliptic curves in cryptography

This chapter presents briefly the algebraic geometry background needed in this thesis. We start by introducing elliptic curves over finite fields, addition law, scalar multiplication and properties of endomorphisms. Next we present the Tate pairing on an elliptic curve. In the second part of this chapter we give the definition of a genus 2 hyperelliptic curve, its Jacobian together with the addition law. Finally, we introduce the zeta function and the Weil numbers of a Jacobian.

1.1 Motivation

In the 70's, the cryptographic community experienced a revolution with the introduction of asymmetric cryptography. History remembers the Diffie-Hellman key agreement [DH76] and the RSA public-key encryption scheme [RSA78] as the starting point of modern cryptography. The security of the Diffie-Hellman key agreement relies on the intractability of the so-called Diffie-Hellman Problem (DHP). The reader is refereed to e.g. [MvV97, §3.6 and 12.6] for an introduction on this subject. This problem and its variants are beneath a large proportion of protocols used in cryptography nowadays. In this thesis, we are interested in the instantiation of some protocols using variants of the DHP. We briefly sketch the DHP and the related Discrete Logarithm Problem (DLP for short in the following). The DLP in a multiplicative cyclic group $\mathbb{G}$ of order m generated by g is defined as follows: given as inputs a generator (or base point) g and an element $a \in \mathbb{G}$, compute the integer $x \in [0, \ldots, m-1]$ such that $a = g^x$. The Diffie-Hellman key agreement protocol is based on the intractability of the DHP. This key exchange is sketched in Fig. 1.1.

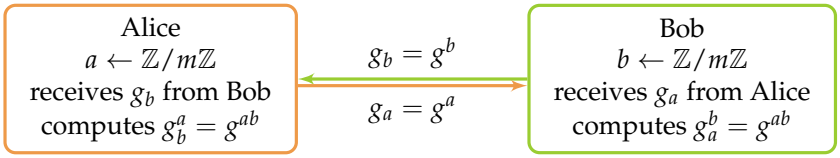


Figure 1.1: Diffie-Hellman key exchange. Alice and Bob share the element g^{ab}.

The Diffie-Hellman Problem on a group $\mathbb{G}$ of order m is defined as follows: given the elements g, g_a, g_b, compute the value g^{ab} with a, b such that $g_a = g^a$ and $g_b = g^b$. If we can solve easily the DLP then we can solve also the DHP. Indeed, we simply compute the discrete logarithm a of g_a then compute $(g_b)^a = g^{a \cdot b}$. The DLP is assumed to be computationally hard in certain well-chosen groups $\mathbb{G}$. Selecting a suitable group for the use of DLP is an active area in cryptography. To ensure a given level of security to a protocol based on the DLP in a group $\mathbb{G}$, we study the complexity of the available attacks on the DLP in the given group $\mathbb{G}$. We then set the group order m accordingly since the attack complexity is directly related to m. We enumerate the main attacks and their complexity in the most used groups. The complexity is given in bits. A security level of ℓ bits in a group $\mathbb{G}$ means that the most efficient attack against the DLP needs (at least) 2^ℓ group operations to compute a discrete logarithm.

1. The Baby-step Giant-step and Pollard-ρ attacks compute a discrete logarithm in a group $\mathbb{G}$ of order m in time complexity $O(\sqrt{m})$ [MvV97, §3.6.2 and 3.6.3]. They are generic attacks available in any group $\mathbb{G}$. To obtain an equivalence of ℓ bits of security in $\mathbb{G}$, we choose a group $\mathbb{G}$ of order $m \sim 2^{2\ell}$ (i.e. $\log m = 2\ell$).
2. The Pohlig-Hellman attack decomposes the DLP in the prime order subgroups of $\mathbb{G}$. Let $m = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$ be the factorization of m. The complexity of this attack is $O(\sum_{i=1}^{k} e_i(\log m + \sqrt{p_i}))$ [MvV97, §3.6.4]. The leading term is $\sqrt{p_j}$ with p_j the largest prime dividing m. That is why we commonly choose a *prime-order* group to instantiate the DLP.
3. In finite fields, another attack better-than-generic named *index calculus* exists. Three optimized variants are used for three different kind of finite fields. We state some complexity results from the survey [JL07].
   a) In the multiplicative group of a finite field of large characteristic and small extension degree $\mathbb{F}_q^\times = \mathbb{F}_{p^e}^\times$ with $e \ll p$, the Number Field Sieve (NFS) method computes a discrete logarithm in sub-exponential time $\exp\left((\sqrt[3]{64/9} + o(1))\ln^{1/3} q \ln^{2/3} \ln q\right)$ [JL07]. This means that the running time in practice of the NFS method is faster than the running time of a generic method such as Pollard-ρ. To achieve a security level of ℓ bits, the size of the finite field is actually larger than 2ℓ. For example, for a 128-security level, a prime finite field $\mathbb{F}_p$ of size $3072 \leqslant \log(p) \leqslant 3248$ is recommended.
   b) The Function Field Sieve (FFS) method computes a discrete logarithm in a finite field of small characteristic and prime degree extension (e.g. $\mathbb{F}_{2^n}$ or $\mathbb{F}_{3^n}$ with n prime) in a complexity $\exp\left((\sqrt[3]{32/9} + o(1))\, n^{1/3} \ln^{2/3} n\right)$ [JL07]. The recommended sizes in this case are even larger than in the previous one. However the arithmetic in characteristic 2 is very efficient in hardware (e.g. FPGA).
   c) The Function Field Sieve (FFS) method computes a discrete logarithm in a finite field of medium sized characteristic and medium prime degree extension (e.g. $\mathbb{F}_{p^k}$ with p, k prime) in a complexity $\exp\left((\sqrt[3]{128/9} + o(1))\ln^{1/3} q \ln^{2/3} \ln q\right)$ [JL07].
   d) When the extension degree n of the finite field is smooth (n is divisible by many small prime numbers), there are prodigious new algorithms solving the DLP (we can cite for example [Jou13a, BGJT13]). Since 2013 these finite fields have been considered weak and should be avoided.

To instantiate a protocol based on the DLP in a group with the smallest possible order for a given level of security (hence optimal parameter sizes) we need a group where the attacks with a better complexity than the generic one are not available. In the 70's these specific index calculus attacks where not yet developed, that is why the multiplicative group of finite fields is widely used. Moreover it has a very efficient group law. However it is not optimal. That's why Koblitz and independently Miller suggested to use the the group of points of an elliptic curve defined over a finite field [Kob90, Kob89, Mil86b]. If we select carefully the curve, only the generic attacks such as the Pohlig-Hellman one apply. We can instantiate the DLP in an elliptic curve group of prime order m with $\log m = 2\ell$ for an equivalent of ℓ-bit security level, in other words the group order size is optimal. However the group law is more complicated (see Sec. 1.2.2) but various improvements have been made and nowadays, Elliptic Curve Cryptography (ECC) is even commonly embedded in smartcards.

In order to be able to use the prime-order groups of elliptic curves in cryptography, we need the following properties.

– We need an efficient method to compute the order of the elliptic curve. For a prime finite field $\mathbb{F}_p$, the order of the multiplicative group is simply $p - 1$. Roughly speaking, to construct a suitable finite field for cryptography, we choose a prime r of 2ℓ bits (to achieve an ℓ-bit security level) and search for a prime $p = h \cdot r + 1$ of size given by tables (based on the NFS complexity), e.g. if $\ell = 128$, we take $3072 \leqslant p \leqslant 3248$. On elliptic curves, this is completely different. It is even worse (much more complicated) on Jacobians (a generalization of elliptic curves proposed to the cryptographic

community in [Kob89]). An algorithm computing a curve order is also named a *point counting* algorithm.
– We need an efficient group law and moreover an efficient exponentiation to compute $g^a \in \mathbb{G}$. On an elliptic curve, the additive notation is commonly used and the cryptographic operation is called a *scalar multiplication*, denoted by $[a]P$ with P a generator (or base point) of the group $\mathbb{G}$. This is explained in Sec. 1.2.4.

In this thesis we describe an improvement of a method to compute efficiently the order of two families of Jacobians, this is explained in Sec. 2.3. We also introduce two new families of elliptic curves (Sec. 2.4) on which we present a method to compute very efficiently a scalar multiplication. We also propose an equivalent method to compute efficiently a scalar multiplication on two families of genus 2 curves in Sec. 2.5.

On certain suitable elliptic curves, a *bilinear map* is available. The properties of this map are explained in Sec. 1.4.1. This bilinear map is also named a *pairing*. In 1999, Harasawa, Shikata, Suzuki and Imai [HSSI99] implemented such bilinear maps with Miller algorithm. They computed a Tate pairing and a Weil pairing on an embedding-degree 2 supersingular curve $E : y^2 = x^3 + x$ defined over a prime finite field $\mathbb{F}_p$ of 163 bits, of order $p + 1$ with a 143-bit prime factor. They computed a Miller function in about 40 000 seconds ($\sim$ 11 hours) on a Pentium SONY QL-50NX at 75MHz. In 2000 Joux showed [Jou00] a method to improve this implementation. Joux was able to compute a pairing on similar curves in less than one second. Joux then proposed a key agreement protocol to show that pairings can be used to design new protocols in cryptography. We sketch Joux's protocol in Fig. 1.2. Pairings are now quite efficient and maybe they will become widespread on smartphones in the forthcoming years.

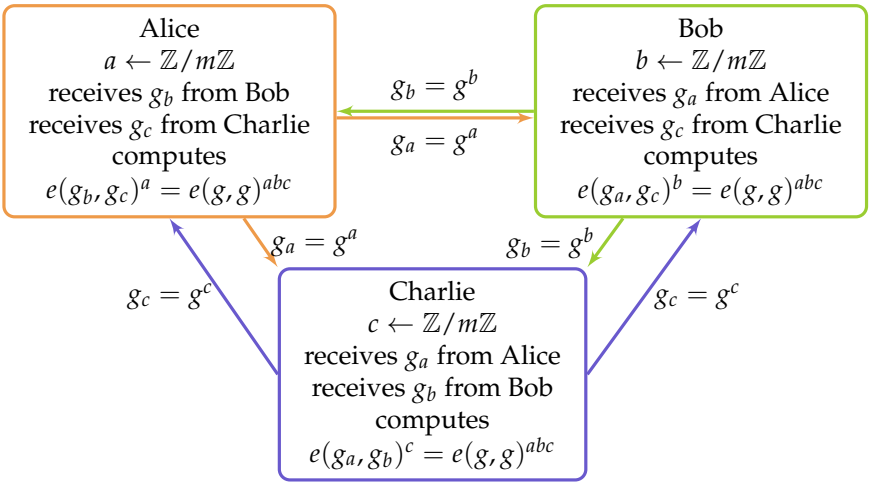


Figure 1.2: Joux key exchange (a.k.a. Tripartite Diffie-Hellman). Alice, Bob and Charlie share the element g^{abc}.

1.2 Elliptic curves

In 1985, Koblitz and Miller independently proposed [Kob90, Kob89, Mil86b] to use in cryptography the group of points of an elliptic curve defined over a finite field. At that time, the multiplicative group of a finite field was commonly used. Nowadays the group of points of an elliptic curve is widely used and recommended as first choice for governmental use [NIS11, FNI10]. The discrete logarithm computation seems indeed less vulnerable in this new group.

1.2.1 Definitions

An elliptic curve is a mathematical object from algebraic geometry. In practice it is usually studied when its coefficients are defined in the field of rational numbers $\mathbb{Q}$ or complex numbers $\mathbb{C}$. In cryptography we consider an elliptic curve defined over a finite field. Le p be a prime number and q a power of p.

We denote by $\mathbb{F}_q$ the finite field of q elements. In all these cases (over $\mathbb{C}$, $\mathbb{Q}$, $\mathbb{F}_q$) we can define an addition law on the set of points on the curve (see Sec. 1.2.2).

An elliptic curve over $\mathbb{C}$ is a projective smooth curve given by an equation of the form

$$E : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \,. \tag{1.1}$$

This is the homogenous Weierstrass form of the curve. We define the values

$$d_2 = a_1^2 + 4a_2,\ d_4 = 2a_4 + a_1a_3,\ d_6 = a_3^2 + 4a_6,\ d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2, \tag{1.2}$$

then we define $\Delta(E)$ to be

$$\Delta(E) = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \tag{1.3}$$

The property $\Delta(E) \neq 0$ is required for the curve to be non-singular. We will assume in all the following that this is the case. The j-invariant of the curve is defined as

$$j(E) = \frac{(d_2^2 - 24d_4)^3}{\Delta(E)} \,. \tag{1.4}$$

The set of points of an elliptic curve is the set of points $(X : Y : Z)$, $Z \neq 0$ satisfying eq. (1.1) plus the point at infinity $P_\infty = (0 : 1 : 0)$. The set of points with coordinates in a given field such as the finite field $\mathbb{F}_q$ is commonly denoted $E(\mathbb{F}_q)$.

The book of Tate and Silverman [ST94] (designed for Master's students) is a good introduction on elliptic curves over $\mathbb{C}$. The more advanced course of Silverman [Sil09] explains important results about the properties of elliptic curves. In the following we will present the background on elliptic curves over finite fields. The reader can refers to [Sil09] and the second volume [Sil94] for the theory over $\mathbb{C}$ and a discussion on the differences that arise when the curve is defined over $\mathbb{F}_q$.

In the next section (Sec. 1.2.2) we will explain the construction of the addition law on the set of points of an elliptic curve defined over a finite field. To start we give the generic expression of an elliptic curve when it's defined over a finite field, and the simplifications (the reduced forms) specific to fields of characteristic 2, 3 and larger than 2 and 3. This will help to compute simpler formulas for the addition law.

An elliptic curve over $\mathbb{F}_q$ can be defined by a generic affine equation named *Weierstrass equation*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \text{ (with } \Delta_E \neq 0) \,. \tag{1.5}$$

The *point at infinity* does not have an expression in affine coordinates. In projective coordinates (1.1) we can write $P_\infty = (0 : 1 : 0)$ as over $\mathbb{C}$. Since this point will be the neutral element of the addition law, it is also denoted by $\mathcal{O}$. We can simplify this equation, depending on the value of $p = \text{char}(\mathbb{F}_q)$. We state the results from [Sil09, A.1.1].

1. If $p \geqslant 5$ we can obtain a short Weierstrass equation of the form

$$E : y^2 = x^3 + a_4x + a_6, \text{ with } \Delta = -16(4a_4^3 + 27a_6^2) \text{ and } j(E) = 1728\frac{4a_4^3}{4a_4^3 + 27a_6^2} \,. \tag{1.6}$$

   This is one of the most used forms in cryptography.

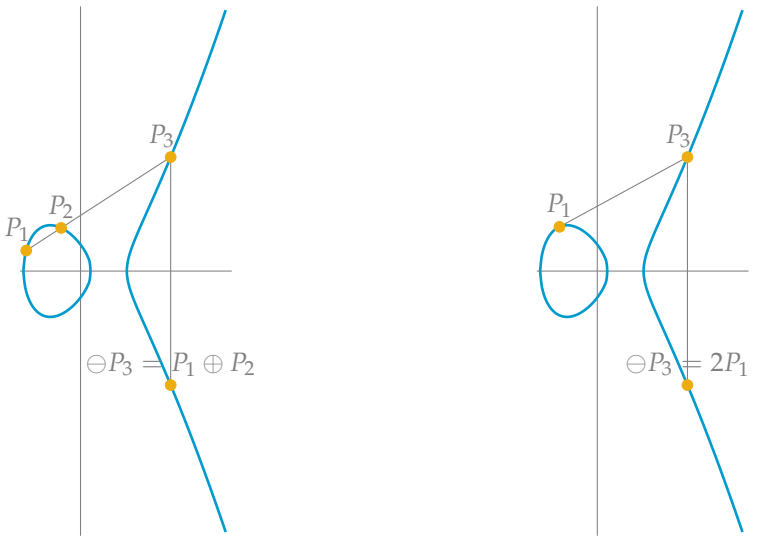
2. If $p = 2$ (i.e. in characteristic 2) we have the following reduced forms:
   – $E : y^2 + xy = x^3 + a_2x^2 + a_6$ if $j(E) \neq 0$, $\Delta = a_6$, $j(E) = 1/a_6$ and
   – $E : y^2 + a_3y = x^3 + a_4x + a_6$ if $j(E) = 0$, in this case $\Delta = a_4^3$.
   The elliptic curves defined over a field of characteristic 2 are wery well used because they are very efficient with a hardware implementation.

3. If $p = 3$ we also have two reduced forms:
   – $E : y^2 = x^3 + a_2x^2 + a_6$ if $j(E) \neq 0$, $\Delta = -a_2^3a_6$, $j(E) = -a_2^3/a_6$ and
   – $E : y^2 = x^3 + a_4x + a_6$ if $j(E) = 0$, in this case $\Delta = -a_4^3$.
   These curves in characteristic 3 have also efficient hardware implementations.

The reduced forms are useful to speed-up the addition law, since some coefficients are equal to zero. Any elliptic curve in a general Weierstrass representation can be turned into one of the above reduced forms with a birational change of variables. There exist other representations, for example the Edwards representation of a curve [Edw07, BL07] is $E : x^2 + y^2 = c^2(1 + dx^2y^2)$ over a field of characteristic strictly greater than 3. The Huff representation of an elliptic curve in characteristic 2 [JTV10, DJ11] is $E : ax(y^2 + y + 1) = by(x^2 + x + 1)$.

1.2.2 Addition law

The set of points of an elliptic curve over a finite field has a group structure with an addition law. The *point at infinity* P_∞ is the neutral element by construction. That's why it is also noted $\mathcal{O}$ in cryptography. We first present a graphical addition law on Fig. 1.3. The addition law was historically defined firstly over $\mathbb{Q}$ and $\mathbb{C}$. The resulting formulas stand for elliptic curves defined over finite fields of characteristic different than 2 and 3. Dedicated addition formulas over $\mathbb{F}_{2^n}$ and $\mathbb{F}_{3^n}$ exist and can be found e.g. online



(a) Addition: draw the line through P_1 and P_2 and reflect the third intersection point

(b) Doubling: draw the tangent at P_1 and reflect the intersection point

Figure 1.3: The chord-and-tangent addition law on an elliptic curve.

[LB, `http://hyperelliptic.org/EFD`]. The difference is that when reducing the general formulas from $\mathbb{C}$ to the finite field, we must avoid the divisions by 2 or 3. Moreover the reduced equation of the curve is not the same (see the previous section 1.2.1).

Let $E : y^2 = x^3 + a_4x + a_6$ be an elliptic curve defined over a field of characteristic different from 2 and 3. Let $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E$, $P_1 \neq \pm P_2$. The negation is straightforward. We then have two different formulas, one for addition and one for doubling.

- Negation. The opposite point of P_1 is $-P_1 = (x_1, -y_1)$.
- Addition. Let $\lambda = \frac{y_1 - y_2}{x_1 - x_2}$. The sum $P_3 = (x_3, y_3)$ of the two points is given by $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$.
- Doubling. Let $\lambda = \frac{3x_1^2 + a_4}{2y_1}$. The doubling $P_3 = (x_3, y_3)$ of the point is given by $x_3 = \lambda^2 - 2x_1$ and $y_3 = \lambda(x_1 - x_3) - y_1$.

This law is commutative and associative, the proof can be found e.g. in [ST94].

1.2.3 Points of order 2 and 3

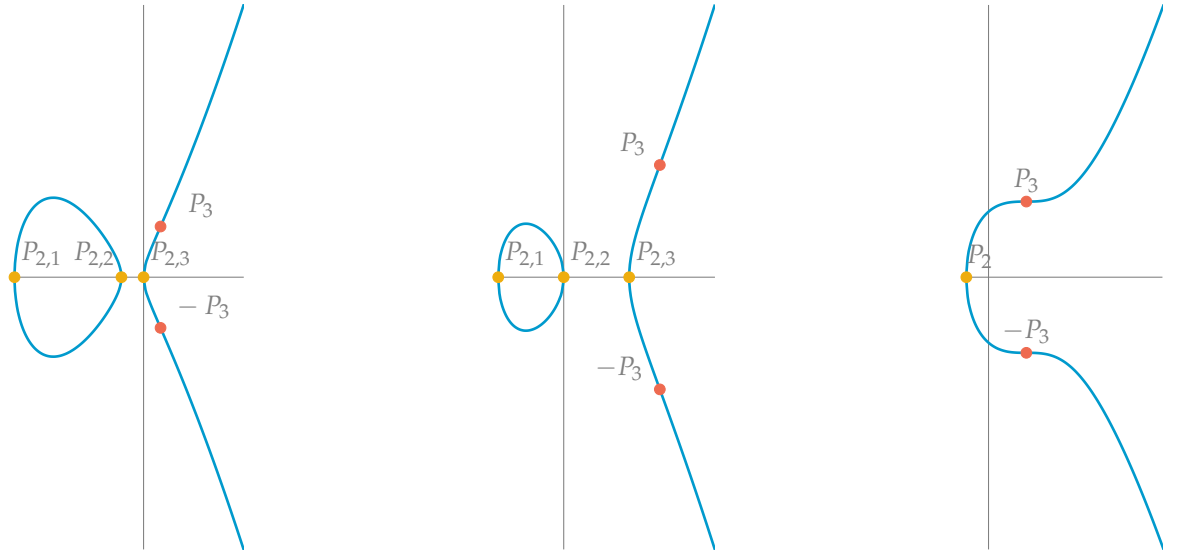
We can characterize graphically the points of order 2 and 3. A point of order two on the curve is such that the tangent at this point is vertical. Since the elliptic curve is symmetric with respect to the abcissa, the y coordinate of a 2-torsion point is equal to 0. There are then three 2-torsion points (different

than $\mathcal{O}$), the points $(x_i, 0)$ where x_i is a root of the polynomial in x on the right side of the equation of E. Graphically (like over $\mathbb{R}$), we can draw one or three such points. Over $\mathbb{C}$, the three points always exist. Over a finite field, it depends if the polynomial in x of the curve equation has roots in the given finite field.

Graphically, the points of order 3 are the inflexion points of the curve. Writing $y = \pm\sqrt{f(x)}$ with $f(x) = x^3 + a_2x^2 + a_4x + a_6$ (this time we keep a_2 and we will cancel a_6 in the following), the inflexion points are the roots of the polynomial $[f''f - \frac{1}{2}f'^2](x)$:

$$3x^4 + 4a_2x^3 + 6a_4x^2 + 12a_6x + (4a_2a_6 - a_4^2)\ . \tag{1.7}$$

Over $\mathbb{C}$ there are four solutions x_j of (1.7) that form eight points on the curve, namely the four (x_j, y_j) plus their opposite $(x_j, -y_j)$. There are eight 3-torsion points different than $\mathcal{O}$ on a curve over $\mathbb{C}$. To find all the 2- and 3-torsion points of an elliptic curve defined over a finite field, we need to consider the points defined over an appropriate extension field.



(a) $y^2 = x^3 + 4x^2 + 2x$, three 2-torsion points, two 3-torsion points

(b) $y^2 = x^3 - 3x$, three 2-torsion and two 3-torsion points

(c) $y^2 = x^3 - 3x^2 + 3x + 3$, one 2-torsion and two 3-torsion points

Figure 1.4: Points of order 2 and 3 on an elliptic curve, representation on $\mathbb{R}$.

1.2.4 Scalar multiplication

Using repeated additions, we may perform a *scalar multiplication* $[m]P = P + P + \ldots + P$, m times, with P a point of the curve and $m \in \mathbb{Z}$. If m is negative, we perform $[-m](-P)$ with $-m > 0$. A well-known efficient implementation of the multiplication $[m]P$ is to write the scalar m in binary representation as explained in Alg. 1.

There exist further improvements. We can cite the binary-signed representation. The negation of a point is almost for free: if $P = (x, y)$ then $-P = (x, -y)$. We write m in binary representation. Then we transform (on the fly) $01\ldots1 \rightarrow -10\ldots0$. In Alg. 1, l. 8 is changed into **if** $m_i = 1$ **then** $S \leftarrow S + P$ else **if** $m_i = -1$ **then** $S \leftarrow S - P$. This technique reduces in average by a factor 2 the number of additions in Alg. 1.

The formulas given in Sec. 1.2.2 require two inversions in $\mathbb{F}_q$ at each step that are expensive, reducing considerably the scalar multiplication efficiency. Different systems of extended coordinates were proposed to avoid inversion. The website [LB] enumerates these different systems. The main idea is to accumulate in a third coordinate (commonly denoted by Z) the denominators and perform a single inversion at the end of the scalar multiplication in order to output the point in affine coordinates. We present in Tab. 1.1 p. 10 three well-known systems in large characteristic: the projective, Jacobian and Edwards

Algorithm 1: Double-and-add scalar multiplication on an elliptic curve.

Input: An elliptic curve E, a point P on the curve, a scalar $m > 0$
Output: The point $S = [m]P$.

```
if m = 0 then
    Return O
else
    Write m in binary representation, m = Σ_{i=0}^{I} m_i 2^i with m_i ∈ {0,1}
    S ← P
    for i from I − 1 to 0 do                 from most significant bit to less significant bit (or left to right)
        S ← 2S                               computed with the doubling formula
        if m_i = 1 then
            S ← S + P                        computed with the addition formula
return S
```

coordinates. The notation M stands for a multiplication, S for a square, $M_a$and M_c for a multiplication by the curve parameter a, resp. c. If the parameter is small, e.g. $a = 1$ then this can be perfomed with an addition instead of a multiplication.

1.2.5 Group of m-torsion points

A *point of order m* is such that $[m]P = \mathcal{O}$ and m is minimal in the sense that for all divisor d of m different than m, $[d]P \neq \mathcal{O}$. An *m-torsion point* is such that $[m]P = \mathcal{O}$. The group of m-torsion points with coordinates in a finite field $\mathbb{F}_q$ is the group of $\mathbb{F}_q$-rational m-torsion points and is denoted $E(\mathbb{F}_q)[m]$.

$$E(\mathbb{F}_q)[m] = \{P \in E(\mathbb{F}_q), [m]P = \mathcal{O}\} .$$

The group of points of m-torsion with coordinates in the algebraic closure of $\mathbb{F}_q$ is denoted $E(\overline{\mathbb{F}_q})[m]$ or $E[m]$. We are interested in the structure of $E[m]$. Let p denotes the characteristic of $\mathbb{F}_q$. If p does not divide m then

$$E[m] \simeq \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z} .$$

For $m = p$ then either $E[p^\ell] = \mathcal{O}$ for all $\ell > 0$ or $E[p^\ell] = \mathbb{Z}/p^\ell\mathbb{Z}$ with the following definition that distinguish these two cases.

Definition 1. *Let E be an elliptic curve defined over $\mathbb{F}_q$ of characteristic p. The curve E is* supersingular *if it has no point of order p over $\overline{\mathbb{F}_q}$, i.e. if $E[p^\ell] = \{\mathcal{O}\}$ for all $\ell > 0$. Otherwise $E[p^\ell] = \mathbb{Z}/p^\ell\mathbb{Z}$ and the curve is* ordinary.

The supersingular curves are interesting in cryptography. They were used at the beginning of the elliptic curve based cryptography, when it was highly difficult to count the number of points of a given curve over a finite field because the order of a supersingular elliptic curve is already known. These curves are nowadays quite used in pairing-based cryptography. In the following section we give somes properties on the elliptic curve order over a finite field. Then it will be possible to express the order of a supersingular elliptic curve.

1.2.6 Elliptic curve order and characteristic polynomial of the Frobenius endomorphism

In cryptography we need to know the order of the elliptic curve we are considering. For a curve defined over a field $\mathbb{F}_q$, $E(\mathbb{F}_q)$ denotes the group order. This is the number of points with coordinates in $\mathbb{F}_q$ (plus the point at infinity). Moreover to understand the pairing-friendly curve constructions presented in Sec. 1.4.3, we will need a relation, for an elliptic curve defined over a finite field $\mathbb{F}_q$, between its number of points with coordinates in an extension field, denoted $\#E(\mathbb{F}_{q^k})$, in terms of the number of points of the curve with coordinates in the basefield, namely $\#E(\mathbb{F}_q)$. For doing that, we will use some properties of the Frobenius endomorphism and of its characteristic polynomial. This characteristic polynomial indeed

Table 1.1: Addition and doubling in projective, Jacobian and Edwards coordinates for points with coordinates in a field of characteristic different than 2 and 3.

(a) Doubling in projective, Jacobian and Edwards coordinates.

Projective	**Jacobian**	**Edwards**
$E : y^2 = x^3 + a_4x + a_6$	$E : y^2 = x^3 + a_4x + a_6$	$E : x^2 + y^2 = c^2(1 + dx^2y^2)$
$(x, y) = (X/Z, Y/Z)$	$(x, y) = (X/Z^2, Y/Z^3)$	$(x, y) = (X/Z, Y/Z)$
$P_1 = (X_1 : Y_1 : Z_1)$, doubling: $P_3 = 2P_1 = (X_3 : Y_3 : Z_3)$		
$X_2 = X_1^2$ $Z_2 = Z_1^2$ $W = a \cdot Z_2 + 3X_2$ $S_1 = 2Y_1 \cdot Z_1$ $S_2 = S_1^2$ $S_3 = S_1 \cdot S_2$ $R = Y_1 \cdot S_1$ $R_2 = R^2$ $B = (X_1 + R)^2 - X_2 - R_2$ $H = W^2 - 2B$ $X_3 = H \cdot S_1$ $Y_3 = W \cdot (B - H) - 2R_2$ $Z_3 = S_3$	$X_2 = X_1^2$ $Y_2 = Y_1^2$ $Y_4 = Y_2^2$ $Z_2 = Z_1^2$ $S = 2((X_1 + Y_2)^2 - X_2 - Y_4)$ $M = 3X_2 + aZ_2^2$ $T = M^2 - 2S$ $X_3 = T$ $Y_3 = M \cdot (S - T) - 8Y_4$ $Z_3 = (Y_1 + Z_1)^2 - Y_2 - Z_2$	$B = (X_1 + Y_1)^2$ $C = X_1^2$ $D = Y_1^2$ $E = C + D$ $H = (c \cdot Z_1)^2$ $J = E - 2H$ $X_3 = c \cdot (B - E) \cdot J$ $Y_3 = c \cdot E \cdot (C - D)$ $Z_3 = E \cdot J$
$5M + 6S + M_a$	$1M + 8S + 1M_a$	$3M + 4S + 3M_c$

(b) Addition in projective, Jacobian and Edwards coordinates.

Projective	**Jacobian**	**Edwards**
$P_1 = (X_1 : Y_1 : Z_1), P_2 = (X_2 : Y_2 : Z_2)$, addition: $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3)$		
$S = Y_1 \cdot Z_2$ $T = X_1 \cdot Z_2$ $Z = Z_1 \cdot Z_2$ $U = Y_2 \cdot Z_1 - S$ $U_2 = U^2$ $V = X_2 \cdot Z_1 - T$ $V_2 = V^2$ $V_3 = V \cdot V_2$ $R = V_2 \cdot T$ $A = U_2 \cdot Z - V_3 - 2R$ $X_3 = V \cdot A$ $Y_3 = U \cdot (R - A) - V_3 \cdot S$ $Z_3 = V_3 \cdot Z$	$Z = Z_1^2$ $U_2 = X_2 \cdot Z$ $S_2 = Y_2 \cdot Z_1 \cdot Z$ $H = U_2 - X_1$ $H_2 = H^2$ $I = 4H_2$ $J = H \cdot I$ $R = 2 \cdot (S_2 - Y_1)$ $V = X_1 \cdot I$ $X_3 = R^2 - J - 2 \cdot V$ $Y_3 = R \cdot (V - X_3) - 2Y_1 \cdot J$ $Z_3 = (Z_1 + H)^2 - Z - H_2$	$A = Z_1 \cdot Z_2$ $B = A_2$ $C = X_1 \cdot X_2$ $D = Y_1 \cdot Y_2$ $E = d \cdot C \cdot D$ $F = B - E$ $G = B + E$ $X_3 = A \cdot F \cdot ((X_1 + Y_1)$ $\cdot (X_2 + Y_2) - C - D)$ $Y_3 = A \cdot G \cdot (D - C)$ $Z_3 = c \cdot F \cdot G$
$12M + 2S + M_a$	$7M + 4S$	$10M + S + M_c$

provides an expression of $\#E(\mathbb{F}_{q^k})$ with respect to $\#E(\mathbb{F}_q)$. Background and definitions are presented in e.g. [Sil09, V.2] and [LV05, §8.1.1].

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let

$$\begin{array}{rcl} \pi_q : E & \to & E \\ (x, y) & \mapsto & (x^q, y^q) \end{array}$$

be the q^{th} power Frobenius endomorphism. The characteristic polynomial of the Frobenius π_q is

$$\chi_{E,\pi_q}(T) = T^2 - tT + q$$

with the trace t such that $-2\sqrt{q} \leqslant t \leqslant 2\sqrt{q}$ by the Hasse bound. A point P is in $E(\mathbb{F}_q)$ if and only if $\pi_q(P) = P$ hence $\#E(\mathbb{F}_q) = \#\ker(\pi_q - Id) = \chi_{\pi_q}(1) = q - t + 1$. Similarly, a point P is in $E(\mathbb{F}_{q^k})$ iff $\pi_{q^k}(P) = P$ so $\#E(\mathbb{F}_{q^k}) = \chi_{\pi_{q^k}}(1)$ with $\chi_{\pi_{q^k}}$ the characteristic polynomial of $\pi_q^k = \pi_{q^k}$. To compute the order of the curve over an extension field $\mathbb{F}_{q^k}$ we only need to know the coefficients of $\chi_{\pi_{q^k}}$. These

coefficients are given by Newton's recurrence formulas. The characteristic polynomial of π_{q^k} is of the form $\chi_{\pi_{q^k}} = T^2 - t_k T + q^k$ with

$$\begin{aligned} t_1 &= t \\ t_2 &= t^2 - 2q \\ t_k &= t \cdot t_{k-1} - q \cdot t_{k-2} \text{ for } k > 2. \end{aligned} \tag{1.8}$$

As an example we can compute the first traces t_k for $k \in \{2, 3, 4, 6\}$.

$$\begin{aligned} \#E(\mathbb{F}_q) &= q + 1 - t \\ \#E(\mathbb{F}_{q^2}) &= q^2 + 1 - (t^2 - 2q) \\ \#E(\mathbb{F}_{q^3}) &= q^3 + 1 - (t^3 - 3tq) \\ \#E(\mathbb{F}_{q^4}) &= q^4 + 1 - (t^4 - 4qt^2 + 2q^2) \\ \#E(\mathbb{F}_{q^6}) &= q^6 + 1 - (t^6 - 6qt^4 + 9q^2t^2 - 2q^3) \end{aligned}$$

So the main point to compute the curve order over $\mathbb{F}_q$ is to compute its trace t. This question was deeply investigated in the last thirty years. This computation is related to the computations of isogenies. At the beginning of ECC, computing a curve order was not feasible so supersingular curves were proposed. We explain now why the order of these curves is easy to compute.

Proposition 1. *Let p denotes the characteristic of $\mathbb{F}_q$. An elliptic curve defined ove a finite field $\mathbb{F}_q$ is supersingular if one of the equivalent conditions holds:*

1. *$E[p^\ell] = \{\mathcal{O}\}$ for all $\ell > 0$;*
2. *the trace of the curve t satisfies $t \equiv 0 \bmod p$;*
3. *the endomorphism ring of E is an order in a quaternion algebra.*

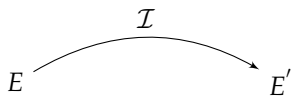
The first condition says that the curve has no point of p-torsion. The second condition gives a quite restrictive condition on the trace. For example if $\mathbb{F}_q$ is a prime field, then $t \equiv 0 \bmod q$. Thanks to the Hasse bound: $|t| \leqslant 2\sqrt{q}$, the only possibility is then $t = 0$ hence $\#E(\mathbb{F}_q) = q + 1$. If $q = p^2$ with p prime then the trace t can be $-2p, -p, 0, p, 2p$ and there are five possibilities for $\#E(\mathbb{F}_q)$. That's why the order of a supersingular curve is easy to compute.

The third condition says that in particular, the endomorphism ring of E is non-commutative. We will present the structure of the endomorphism ring of an elliptic curve in Sec. 1.2.7.

1.2.7 Isogenies and endomorphisms

In this section we will define an isogeny betweeen two elliptic curves. We will present how to compute it with Velu's formulas in Sec. 1.2.8. Then we will present endomorphisms and the structure of the endomorphism ring of an elliptic curve and the difference between ordinary and supersingular curves in this case.

Definition 2. *Let E and E' be two elliptic curves defined over $\mathbb{F}_q$. An* isogeny $\mathcal{I} : E \to E'$ *is a morphism of curves that preserves the point at infinity. The curves E and E' are said* isogenous.



As a consequence, an isogeny is surjective and has finite kernel. The degree of the isogeny is $\deg \mathcal{I} = \#\ker \mathcal{I}$.

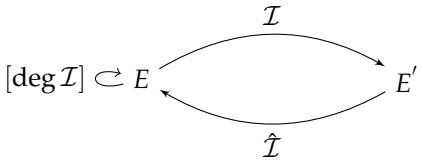
Example 1. *In Sec. 1.2.10.1 we compute an example of a degree-2 isogeny. Let $E : y^2 = x^3 + a_2x^2 + a_4x$ be an elliptic curve defined over $\mathbb{F}_q$. We don't use the reduced Weierstrass equation here because in this way the isogeny has a nicer expression. A point of order 2 on this curve is $P_2 = (0, 0)$. The degree-2 isogeny has kernel $\ker \mathcal{I}_2 = \{P_2, \mathcal{O}\}$. The isogenous curve is $E' : y'^2 = x'^3 - 2a_2x'^2 + (a_2^2 - 4a_4)x'$. The isogeny is given by*

$$\begin{aligned} \mathcal{I}_2 : E &\to E' \\ (x, y) &\mapsto \begin{cases} \mathcal{O} & \text{if } P = (0, 0), \\ \left(x + a_2 + \frac{a_4}{x}, y\left(1 - \frac{a_4}{x^2}\right)\right) & \text{otherwise.} \end{cases} \end{aligned}$$

We explain this computation in Sec. 1.2.10.1. This example is to show that when $\deg(\mathcal{I})$ *is small the isogeny has a simple expression.*

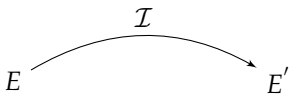
An isogeny has the important property to factor the multiplication-by-m map.

Proposition 2. *Let* E *and* E' *be two isogenous elliptic curves defined over* $\mathbb{F}_q$ *and let* $\mathcal{I}$ *denote the isogeny. There exists a* dual isogeny $\hat{\mathcal{I}} : E' \to E$ *such that* $\hat{\mathcal{I}} \circ \mathcal{I} = [\deg \mathcal{I}]$. *The composition of* $\mathcal{I}$ *and its dual* $\hat{\mathcal{I}}$ *is the multiplication by* $\deg \mathcal{I}$ *on* E.



There is another important result about isogenous elliptic curves.

Theorem 1. *Honda-Tate theorem for elliptic curves. Let* E *and* E' *be two elliptic curves definied over a finite field* $\mathbb{F}_q$. *The two curves are isogeous over* $\mathbb{F}_q$ *iff their respective Frobenius endomorphisms* π_q *have the same characteristic polynomial.*



$$\chi_{E,\pi_q} = \chi_{E',\pi_q}$$

This result is a consequence for genus one curves of the Honda-Tate theorem. This theorem arises in the more general theory of genus g curves. We will use this result in Ch. 2. Note that the curves do not need to be isomorphic but only isogenous. An isomorphism of curves is a stronger notion that we define just after.

Proposition 3. *Let* E *and* E' *be two elliptic curves defined over* $\mathbb{F}_q$. *The curves are* isomorphic *iff they have the same j-invariant.*

$$E \xrightarrow[\text{isomorphism}]{i} E'$$

$$j(E) = j(E')$$

An isogeny $E \to E$ is a curve endomorphism. We will also use ϕ to denote an endomorphism.

$$E \xrightarrow{\phi} E$$

We now state results on endomorphisms on an elliptic curve and its endomorphism ring. We are also interested in the group of the elliptic curve. The following theorem states that a curve isogeny induces a morphism of groups hence a curve endomorphism is also a group endomorphism.

Theorem 2. *[Sil09, Th. III.4.8] Let*

$$\mathcal{I} : E \to E'$$

be an isogeny. Then

$$\mathcal{I}(P + Q) = \mathcal{I}(P) + \mathcal{I}(Q) \text{ for all } P, Q \in E\,.$$

All the multiplication-by-m maps on the curve are endomorphisms. Hence the endomorphism ring of E contains $\mathbb{Z}$. Moreover, we saw in Sec. 1.2.6 that there exists the Frobenius endomorphism π_q. We have this result on $\mathrm{End}(E)$.

Proposition 4. *Let* E *be an elliptic curve defined over* $\mathbb{F}_q$.

1. *If* E *is supersingular then* $\mathrm{End}(E)$ *is an order in a quaternion algebra.*

2. *If E is ordinary then* $\mathrm{End}(E)$ *is an order in a quadratic imaginary field.*

Let E be an ordinary elliptic curve and t the trace of the Frobenius endomorphism. Define the *discriminant* of the curve to be the number D such that $t^2 - 4q = -D\gamma^2$ with D square-free. Moreover if $-D \equiv 2, 3 \bmod 4$ then set $-D$ to be $-4D$, so we have $-D \equiv 0, 1 \bmod 4$ now.

- If $-D \equiv 1 \bmod 4$ then $\mathrm{End}(E) = \mathbb{Z}\left[\frac{1+\sqrt{-D}}{2}\right]$ and there exists an endomorphism ϕ on the curve satisfying $\phi^2 - \phi + \frac{D+1}{4} = 0$.
- If $D \equiv 0 \bmod 4$ then $\mathrm{End}(E) = \mathbb{Z}\left[\sqrt{-D}\right]$ and there exists an endomorphism ϕ on the curve satisfying $\phi^2 + D = 0$.

How to compute this endomorphism for a given curve E over $\mathbb{F}_q$? In the case $-D \equiv 1 \bmod 4$ the degree of ϕ is $\frac{D+1}{4}$. The first step is to compute an isogeny of degree $\frac{D+1}{4}$. In the case $-D \equiv 0 \bmod 4$ we start with an isogeny of degree $D/4$. We obtain a second elliptic curve E' (with Vélu's formulas explained in Sec. 1.2.8). Then there will be an isomorphism from E' to E to turn the isogeny into an endomorphism.

1.2.8 Isogenies with Vélu's formulas

In this section we recall Vélu's formulas for computing isogenies and further improvements on these formulas found independently by Dewaghe [Dew95] and Kohel [Koh96]. A precise description and improvements were given in Lercier's thesis. [Ler97, §4.1]. A more recent description and implementation can be found in De Feo's thesis [DF10]. Let E_a be an elliptic curve defined over an algebraic closed field $\mathbb{K}$, and F a subgroup of the group of points of E_a. There exists an elliptic curve E_b defined over the field $\mathbb{K}$ and an isogeny of kernel F from E_a to E_b with coefficients in $\mathbb{K}$.

The isogeny from E_a into E_b of kernel F is given by

$$P \mapsto \begin{cases} \mathcal{O}_{E_b} & \text{if } P = \mathcal{O}_{E_a}, \\ \left(x + \sum_{Q \in F \setminus \mathcal{O}_{E_a}} x_{P+Q} - x_Q,\ y + \sum_{Q \in F \setminus \mathcal{O}_{E_a}} y_{P+Q} - y_Q\right) & \text{if } P = (x, y) \end{cases} \tag{1.9}$$

and the coefficients of E_b are also given by explicit formulas. To simplify, assume that

$$E_a : y^2 = x^3 + a_2 x^2 + a_4 x + a_6 = f(x)\,. \tag{1.10}$$

There are more general formulas for elliptic curves that are not in reduced Weierstrass form given in []. We write here the simplified version. Let R be the subset of F defined by $F \setminus E_a[2] = R \cup (-R)$, $R \cap (-R) = \emptyset$ and $S = F \cap E_a[2] - \{\mathcal{O}_{E_a}\}$. Now let for all points $Q = (x_Q, x_Q) \in F \setminus \{\mathcal{O}_{E_a}\}$,

$$\begin{array}{rclcrcl} g_Q^x &=& 3x_Q^2 + 2a_2 x_Q + a_4 = f'(x_Q), & & t_Q &=& \begin{cases} g_Q^x & \text{if } Q \in S, \\ 2g_Q^x = 6x_Q^2 + 4a_2 x_Q + 2a_4 & \text{otherwise,} \end{cases} \\ g_Q^y &=& -2y_Q, & & u_Q &=& (g_Q^y)^2 = 4y_Q^2 = 4x_Q^3 + 4a_2 x_Q^2 + 4a_4 x_Q + 4a_6, \\ t &=& \sum_{Q \in R \cup S} t_Q, & & & & \\ w &=& \sum_{Q \in R \cup S} u_Q + x_Q t_Q\,. & & & & \end{array} \tag{1.11}$$

Then E_b is given by

$$E_b : y^2 = x^3 + b_2 x^2 + b_4 x + b_6 \text{ with } b_2 = a_2,\ b_4 = a_4 - 5t \text{ and } b_6 = a_6 - 4a_2 t - 7w \tag{1.12}$$

and the isogeny has degree $\#F$ and is given by

$$\begin{array}{rcl} \mathcal{I} : E_a & \to & E_b \\ P & \mapsto & \begin{cases} \mathcal{O}_{E_b} & \text{if } P = \mathcal{O}_{E_a}, \\ (x_{\mathcal{I}(P)}, y_{\mathcal{I}(P)}) & \text{if } P = (x, y) \end{cases} \end{array} \tag{1.13}$$

with

$$\begin{cases} x_{\mathcal{I}(P)} &=& x + \sum_{Q \in R \cup S} \left(\frac{t_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2}\right), \\ y_{\mathcal{I}(P)} &=& y + \sum_{Q \in R \cup S} \left(\frac{2u_Q y}{(x - x_Q)^3} + \frac{t_Q(y - y_Q) - g_Q^x g_Q^y}{(x - x_Q)^2}\right)\,. \end{cases} \tag{1.14}$$

Assuming that the 2-torsion points are of the form $(x_S, 0)$ we simplify the formulas. We have $g_S^y = 0$, $u_S = 0$ and the formulas are

$$\left\{ \begin{array}{rcl} x_{\mathcal{I}(P)} & = & x + \sum_{Q \in R \cup S} \frac{t_Q}{x - x_Q} + \sum_R \frac{u_R}{(x-x_R)^2}, \\ y_{\mathcal{I}(P)} & = & y \left(1 + \sum_S \frac{t_S}{(x-x_S)^2} + \sum_R \left(\frac{2u_R}{(x-x_R)^3} + \frac{t_R}{(x-x_R)^2} \right) \right) . \end{array} \right. \tag{1.15}$$

1.2.9 Gallant-Lambert-Vanstone method for scalar multiplication

In 2001, Gallant, Lambert and Vanstone [GLV01] introduced a new idea to speed-up scalar multiplication on elliptic curves. This improvement was not available on generic groups or on prime fields. They exploit the existence of some shortcut on the group of points. We denote by E an elliptic curve defined over a finite field $\mathbb{F}_q$ and by r the order of $E(\mathbb{F}_q)$. Given an efficient shortcut ϕ to compute the scalar multiplication $[\lambda]$ on the curve with a given fixed λ, they decompose a random scalar m into $m_0 + m_1\lambda$ mod r with m_0, m_1 of half size compared to m. Since for any $P \in E(\mathbb{F}_q)$, we have $[r]P = \mathcal{O}$, the scalar multiplication can be $[m]P = [m_0]P + [m_1]\phi(P)$. This method requires an elliptic curve with an endomorphism ϕ (the shortcut) efficiently computable and a point P which is an eigenvector for ϕ. Some families of elliptic curves have this property. We give examples in the following.

Why is computing $[m_0]P + [m_1]\phi(P)$ more efficient than computing $[m]P$? Computing $[m]P$ costs $\log_2 m$ doublings and $\log_2 m/2$ additions in average with Alg.1. Computing $[m_0]P + [m_1]\phi(P)$ sequentially costs $\log_2 m_0 + \log_2 m_1$ doublings, half additions (in average) and one evaluation of ϕ. There exists a method to parallelize the computation of $[m_0]P + [m_1]Q$ for a total cost of $\max(\log_2 m_0, \log_2 m_1)$ doublings instead of $\log_2 m_0 + \log_2 m_1$. This saves half the doublings if m_0 and m_1 are balanced. More generally, the method computes $[m_1]P_1 + [m_2]P_2 + \ldots + [m_i]P_i$ in $\max_i \log m_i$ doublings and additions (instead of $\sum_i \log m_i$), plus 2^{i-1} precomputations (and their storage in memory). We present this method applied for two points in Alg. 2.

Algorithm 2: Double scalar-multiplication on an elliptic curve

Input: An elliptic curve E, two points P, Q and two scalars a, b
Output: The point $S = [a]P + [b]Q$.

```
Precompute R = P + Q
Write a = Σ_{i=0}^{I_a} a_i 2^i, b = Σ_{i=0}^{I_b} b_i 2^i with a_i, b_i ∈ {0,1}
if I_a > I_b then S ← P else if I_b > I_a then S ← Q else S ← R
for i from max(I_a, I_b) − 1 to 0 do                                  left to right
    S ← 2S
    if a_i = 1, b_i = 1 then S ← S + R else if a_i = 1, b_i = 0 then S ← S + P else if a_i = 0, b_i = 1 then
    S ← S + Q
return S
```

On average, this technique costs $\max(\log a, \log b)$ doublings and $3/4 \max(\log a, \log b)$ additions on the curve. The naive method computes sequentially $[a]P$ then $[b]Q$ and adds both points. This costs on average $\log a + \log b$ doublings and $1/2(\log a + \log b)$ additions. The technique presented in Alg. 2 is faster if $\log a \approx \log b$. More accurate estimates are described in [GLV01].

This method of Gallant, Lambert and Vanstone is efficient also if the cost for evaluating ϕ is negligeable, for instance if ϕ costs a doubling. Secondly the eigenvalue λ needs to be large enough so that in the decomposition $m = m_0 + m_1\lambda$ mod r, the two m_0, m_1 have (almost) half size of m. So elliptic curves with such a very efficient endomorphism and large eigenvalue are required to apply this method. Finally wa also want a decomposition into m_0 and m_1 of negligeable cost (compared to the computation of $[m_0]P + [m_1]Q$). An elliptic curve may have an *endomorphism* different from the scalar multiplication. We give two examples in the following (Ex. 2 and 3).

Elliptic curves with such an endomorphism are very rare. Nevertheless, they are well-known in cryptography. In characteristic different than 2 and 3, we can mention the two families of curves $E_a : y^2 = x^3 + ax$ of j-invariant 1728 (used in practice over $\mathbb{F}_q$ with $q \equiv 1 \bmod 4$) and the curves $E_b : y^2 = x^3 + b$ of j-invariant 0 (in practice, over $\mathbb{F}_q$ with $q \equiv 1 \bmod 3$).

Example 2. *Let $E_a : y^2 = x^3 + ax$. The curve has* Complex Multiplication *by $\sqrt{-1}$: $\phi : P = (x, y) \mapsto [\sqrt{-1}]P = (-x, iy)$ with i such that $i^2 = -1$. Intuitively, note that $\phi^2(P) = (-(-x), i^2 y) = (x, -y) = -P \rightarrow \phi^2 = [-1]$. If we consider points with coordinates in a finite field $\mathbb{F}_q$ with $q \equiv 1 \bmod 4$ then there exists $i \in \mathbb{F}_q$ s.t. $i^2 = -1$. If P is of prime-order r then the eigenvalue of this endomorphism is $\lambda = \sqrt{-1} \mod r$ and $P \mapsto (-x, iy) = [\lambda]P$. If the curve is defined over $\mathbb{F}_q$ with $q = 3 \mod 4$ then i is not in $\mathbb{F}_q$ but in $\mathbb{F}_{q^2}$ and moreover the curve is supersingular (see Sec. 1.4.3.1).*

Example 3. *Consider the elliptic curve $E_b : y^2 = x^3 + b$. Let ζ_3 be a primitive third root of unity, i.e. such that $\zeta_3^2 + \zeta_3 + 1 = 0$. The curve has* Complex Multiplication *by $\frac{-1+\sqrt{-3}}{2}$. The endomorphism is $\phi : P(x, y) \mapsto (\zeta_3 x, y)$. Note that $\phi^3(P) = P$. Now consider the points with coordinates in $\mathbb{F}_q$, with $q = 1 \mod 3$. In this case there exists a primitive third root of unity $\zeta_3 \in \mathbb{F}_q$. The eigenvalue satisfies $\lambda = \frac{-1+\sqrt{-3}}{2} \mod r$ with r the order of P. Note also that ζ_3 and λ both correspond to a primitive third root of unity but $\zeta_3 \in \mathbb{F}_q$ whereas λ is taken mod r. Note that we need $q = 1 \mod 3$ otherwise $\zeta_3 \notin \mathbb{F}_q$ and the curve is supersingular (see Sec. 1.4.3.1).*

We will explain in the next section how to construct an elliptic curve with an endomorphism of given kernel and how to compute this endomorphism.

1.2.10 Endomorphisms on elliptic curves: two examples

We will explain two examples of endomorphisms on elliptic curves defined over $\mathbb{F}_q$. We will start by computing for our first example an isogeny of degree 2, i.e. an isogeny whose kernel is of the form $\{P_2, \mathcal{O}\}$ with P_2 a 2-torsion point on the curve. For our second example, we will start by computing a degree 3 isogeny whose kernel is of the form $\{P_3, -P_3, \mathcal{O}\}$ with P_3 a 3-torsion point of the curve.

1.2.10.1 Endomorphisms constructed from a degree-2 isogeny

We aim to find an elliptic curve E defined over $\mathbb{F}_q$ with Complex Multiplication by $\sqrt{-2}$, i.e. with an endomorphism ϕ such that on a prime subgroup of $E(\mathbb{F}_q)$, $\phi^2 = [-2]$. We start by finding with Vélu's formulas an isogeny of degree 2, i.e. whose kernel is $\{\mathcal{O}, (x_0, 0)\}$ with $(x_0, 0)$ a 2-torsion point. The general approach can be found in [Sil94, II, Prop. 2.3.1]. Let $E : y^2 = x^3 + a_2x^2 + a_4x + a_6$ be an elliptic curve defined over $\mathbb{F}_q$ with a 2-torsion point $(x_0, 0)$. If $x_0 = 0$ then $a_6 = 0$ and the curve equation is of the form $y^2 = x(x^2 + a_2x + a_4)$. Otherwise $x_0 \neq 0$ but satisfies $x_0^3 + a_2x_0^2 + a_4x_0 + a_6 = 0$ hence $a_6 = -(x_0^3 + a_2x_0^2 + a_4x_0)$ and we can write $y^2 = (x - x_0)((x - x_0)^2 + (a_2 + 3x_0)(x - x_0) + 3x_0^2 + 2a_2x_0 + a_4) = x'(x'^2 + a_2'x' + a_4')$ with the change of variables $x' = x - x_0$, $a_2' = a_2 + 3x_0$, $a_4' = 3x_0^2 + 2a_2x_0 + a_4$ and $a_6' = 0$. We will assume in the following that[1] $x_0 = 0$ and $a_6 = 0$.

Using Vélu's formulas we find $t = a_4$ and $w = 0$. The 2-isogenous elliptic curve of E is $E' : y^2 = x^3 + a_2x^2 + (a_4 - 5t)x + (a_6 - 4a_2t - 7w) = x^3 + a_2x^2 - 4a_4x - 4a_2a_4$. The isogeny is given by

$$\begin{array}{rcl} \mathcal{I} : E & \rightarrow & E' \\ P = (x, y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } P = (0, 0), \\ \left(x + \frac{a_4}{x}, y\left(1 - \frac{a_4}{x^2}\right)\right) & \text{otherwise.} \end{cases} \end{array} \tag{1.16}$$

We note that the equation of E' can be expressed in $E'_{x_0=0,a_6=0} : y^2 = x^3 + a_2x^2 - 4a_4x - 4a_2a_4 = (x + a_2)((x + a_2)^2 - 2a_2(x + a_2) + a_2^2 - 4a_4)$. We remark that $(-a_2, 0)$ is a 2-torsion point of E'. This will

1. If $x_0 \neq 0, a_6 \neq 0$ we obtain $t = 3x_0^2 + 2a_2x_0 + a_4$ and $w = tx_0$. The image of the 2-isogeny is the elliptic curve $E' : y^2 = x^3 + a_2x^2 + (a_4 - 5t)x + (a_6 - 4a_2t - 7w)$. In terms of a_2, a_4, a_6, x_0, we find that $a_2' = a_2$, $a_4' = -15x_0^2 - 10a_2x_0 - 4a_4$ and $a_6' = -5a_2x_0^2 + (-8a_2^2 + 14a_4)x_0 + 22a_6 - 4a_2a_4$. The isogeny is given by

$$\begin{array}{rcl} \mathcal{I} : E & \rightarrow & E' \\ P = (x, y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } P = (x_0, 0), \\ (x + t/(x - x_0), y\,(1 - t/(x - x_0)^2)) & \text{otherwise, with } t = 3x_0^2 + 2a_2x_0 + a_4\,. \end{cases} \end{array}$$

The j-invariants of the two curves are

$$j(E) = \frac{2^8(-a_2^2 + 3a_4)^3}{4a_4^3 + 27a_6^2 + a_2((4a_2^2 - 18a_4)a_6 - a_2a_4^2)} \text{ and } j(E') = 2^4\,\frac{(a_2^2 + 12a_4 + 15x_0(3x_0 + 2a_2))^3}{u_2x_0^2 + u_1x_0 + u_0}$$

with $u_2 = 2(a_2^2 - 3a_4)^2$; $u_1 = (a_2^2 - 3a_4)(2a_2^3 - 7a_2a_4 + 9a_6)$; $u_0 = -8a_2^2a_4^2 + a_2^4a_4 + 16a_4^3 + 27a_6^2 + (7a_2^3 - 27a_2a_4)a_6$.

be a useful indication in the next step to find the complete change of variables from E' to E to turn the isogeny into an endomorphism. The change of variables will start with $(x', y') \mapsto (x' + a_2, y')$.

The j-invariants of the two curves are

$$j(E) = 2^8 \frac{(3a_4 - a_2^2)^3}{(4a_4 - a_2^2)a_4^2} \text{ and } j(E') = \frac{16(a_2^2 + 12a_4)^3}{-8a_2^2a_4^2 + a_2^4a_4 + 16a_4^3} = 2^4 \frac{(a_2^2 + 12a_4)^3}{(4a_4 - a_2^2)^2 a_4} .$$

Now we set

$$j(E) = j(E') \tag{1.17}$$

in order to obtain an endomorphism on E. We will adopt another approach in Sec. 2.4.1. We assume that $a_6 = 0, x_0 = 0, a_4 \neq 0, 4a_4 - a_2^2 \neq 0$. The equation (1.17) turns into

$$a_2^2(-8a_4 + a_2^2)(16a_2^4 - 81a_2^2a_4 + 324a_4)^2 = 0 . \tag{1.18}$$

1. If $a_2 = 0$ then the curve E has equation $y^2 = x^3 + a_4x$, $E' : y^2 = x^3 - 4a_4x$ and $j(E) = 1728$. This is the curve of Example 2. The map from E' to E is $(x, y) \mapsto \left(\frac{ix}{-2}, \frac{1+i}{-4}y\right)$ defined over $\mathbb{F}_q[\sqrt{-1}]$ and the endomorphism is

$$\begin{array}{rcl} \phi : E & \to & E \\ P = (x, y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } P = (0, 0), \\ \left(\frac{-i}{2}\left(x + \frac{a_4}{x}\right), y\frac{1+i}{-4}\left(1 - \frac{a_4}{x^2}\right)\right) & \text{otherwise.} \end{cases} \end{array}$$

   This endomorphism actually computes $P = (x, y) \mapsto (x, y) + (-x, iy)$ which is $[1 + \sqrt{-1}]$. Note that $(1 + \sqrt{-1})^2 = 2\sqrt{-1}$. Applying two times this endomorphism send the 2-torsion points to $\mathcal{O}$ but this endomorphism is not $[\sqrt{-2}]$. Its characteristic polynomial is $\chi^2 - 2\chi + 2$.

2. If $16a_2^4 - 81a_2^2a_4 + 324a_4^2 = 0 \Leftrightarrow a_4 = \frac{9 \pm 5\sqrt{-7}}{72}a_2^2$ the j-invariant is $j(E) = j(E') = -3375$. The endomorphism computes $\left[\frac{1+\sqrt{-7}}{2}\right]$ (see [Sil94, Ch. II, Prop. 2.3.1]). This is the same curve as in [LS12, LS13, Ex. A.3]. The characteristic polynomial of the endomorphism is $\chi^2 - \chi + 2$. We will meet this particular curve a second time in 2.3.1.3.

3. If $(-8a_4 + a_2^2) = 0 \Leftrightarrow a_4 = a_2^2/8$, $E : y^2 = x^3 + a_2x^2 + \frac{a_2^2}{8}x$, $E' : y^2 = x^3 + a_2x^2 - \frac{a_2^2}{2}x - \frac{a_2^3}{2}$ and $j(E) = j(E') = 8000$. We remark that the 2-torsion point of E' is $(-a_2, 0)$. We write $E' : y^2 = (x + a_2)((x + a_2)^2 - 2a_2(x + a_2) + \frac{a_2^2}{2}) = x'(x'^2 - 2a_2x' + \frac{a_2^2}{2})$ and see that the change of variables from E' back to E is $(x, y) \mapsto \left((x + a_2)/(-2), y/(-2\sqrt{-2})\right)$. Finally the endomorphism is

$$\begin{array}{rcl} \phi_2 : E & \to & E : y^2 = x^3 + a_2x^2 + \frac{a_2^2}{8}x \\ P = (x, y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } P = (0, 0), \\ \left(\frac{-1}{2}\left(x + a_2 + \frac{a_2^2}{8x}\right), \frac{y}{-2\sqrt{-2}}\left(1 + \frac{a_2^2}{8x^2}\right)\right) & \text{otherwise.} \end{cases} \end{array} \tag{1.19}$$

   and satisfies $\phi^2 = [-2]$. This time the characteristic polynomial is $\chi^2 + 2$. Note that this is the curve presented in [LS12, LS13, Ex. A.4]. We can compute explicitly its eigenvalue $\lambda = \sqrt{-2}$. The discriminant of the curve is $D = 2$ and q is of the form $q = \frac{t^2+2y^2}{4}$ with t the trace of the curve over $\mathbb{F}_q$. We have also $\#E(\mathbb{F}_q) = q + 1 - t = \frac{(t-2)^2+2y^2}{4}$ hence $\lambda = \sqrt{-2} \equiv \frac{t-2}{y} \bmod \#E(\mathbb{F}_q)$ (if y is invertible mod $\#E(\mathbb{F}_q)$). For a prime-order r point P, there is no ambiguity on $1/y \bmod r$.

1.2.10.2 Endomorphisms constructed from a degree-3 isogeny

For our second example, we aim to find an elliptic curve with an endomorphism ϕ such that $\phi^2 = [-3]$. The 3-torsion points on the curve are given by the solutions of Eq. (1.7): $3x^4 + 4a_2x^3 + 6a_4x^2 + 12a_6x + (4a_2a_6 - a_4^2) = 0$. To simplify the computations, we assume that $4a_2a_6 - a_4^2 = 0$ in order to have $P_3(0, \sqrt{a_6})$ a 3-torsion point of the curve. We assume that $a_6 \neq 0$ (this 3-torsion point cannot be a 2-torsion point).

1. If $a_2 = a_4 = 0$ then the curve has j-invariant 0 and Complex Multiplication by $\zeta_3 = \frac{-1+\sqrt{-3}}{2}$, this is the curve of Example 3.

If $a_2 \neq 0, a_4 \neq 0$ the point $P_3(0, \sqrt{a_6})$ is a 3-torsion point on the curve (with coordinates in $\mathbb{F}_q$ or $\mathbb{F}_{q^2}$). We set $R = \{P_3\}$ in Vélu's formulas notations. We compute $t = 2a_4$, $w = 4a_6$. We obtain an isogeny of degree 3 into the curve $E' : y^2 = x^3 + a_2x^2 - 9a_4x - (8a_2a_4 + 27a_6)$. The j-invariants are

$$j(E) = 2^{12}\frac{(a_2^2 - 3a_4)^3a_2^2}{(8a_2^2 - 27a_4)a_4^3} \text{ and } j(E') = 2^{12}\frac{(a_2^2 + 27a_4)^3a_2^2}{(8a_2^2 - 27a_4)^3a_4}.$$

The isogeny is given by

$$\begin{array}{rcl} E & \to & E' : y^2 = x^3 + a_2x^2 - 9a_4x - (8a_2a_4 + 27a_6) \\ P = (x,y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } P = \pm P_3 = (0, \pm\sqrt{a_6}), \\ \left(x + \frac{2a_4}{x} + \frac{4a_6}{x^2}, y\left(1 + \frac{2a_4}{x^2} + \frac{8a_6}{x^3}\right)\right) & \text{otherwise.} \end{cases} \end{array} \tag{1.20}$$

Now we set $j(E) = j(E')$. Assuming that $a_2 \neq 0, a_4 \neq 0, 8a_2^2 - 27a_4 \neq 0$ (otherwise the curve would be singular), we obtain the equation $-2(-27a_4 + 4a_2^2)(27a_4^2 - 8a_2^2a_4 + a_2^4)(27a_4^2 - 8a_2^2a_4 + 8a_2^4) = 0$. We observe that on the curve E', the x-coordinate of the obvious 3-torsion point is $-4a_2/3$. With the change of variable $x' \mapsto x' + 4a_2/3 = x''$ we obtain $E'' : y^2 = x''^3 - 3a_2x''^2 + \left(\frac{8}{3}a_2^2 - 9a_4\right)x'' + 4a_2a_4 - \frac{16}{27}a_2^3 - \frac{27}{4}\frac{a_4^2}{a_2}$. This expression will be useful to recover the change of variables from E' to E when they will have the same j-invariant.

We obtain these possibilities.

2. $a_4 = \frac{4a_2^2}{27}$, $j = 54000$. This is the same curve as in [LS12, LS13, Ex. A.6]. Here we have $E : y^2 = x^3 + a_2x^2 + 4a_2^2/3^3x + 4a_2^3/3^6$. The isogenous curve obtained with Vélu's formulas is $E' : y^2 = x^3 + a_2x^2 - 4a_2^2/3x - 4a_2^3/3$. The obvious 3-torsion point on E' is $-4a_2/3$ so we apply first $x \mapsto x + 4a_2/3$. We obtain $E'' : y^2 = x^3 - 3a_2x^2 + \frac{4}{3}a_2^2x - \frac{4}{27}a_2^3$. The map to E is now obvious. We apply $(x,y) \mapsto (x/(-3), y/(-3\sqrt{-3}))$. The complete endomorphism is given by

$$\begin{array}{rcl} E & \to & E : y^2 = x^3 + a_2x^2 + \frac{4a_2^2}{3^3}x + \frac{4a_2^3}{3^6} \\ P = (x,y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } P = \pm P_3, \\ \left(\frac{1}{-3}\left(\frac{4a_2}{3} + x + \frac{2^2a_2^2}{3^3x} + \frac{2^4a_2^3}{3^6x^2}\right), \frac{y}{-3\sqrt{-3}}\left(1 + \frac{8a_2^2}{3^3x^2} + \frac{2^5a_2^3}{3^6x^3}\right)\right) & \text{otherwise.} \end{cases} \end{array} \tag{1.21}$$

We can apply the change of variables $(x,y) \mapsto \left(\frac{3^2}{a_2}x + 3, \frac{3^3}{\sqrt{a_2}^3}y\right)$ to obtain a reduced form $E' : y'^2 = x'^3 - 15x' + 22$. The 3-torsion point we consider is $P_3(3,2)$. The endomorphism is then

$$\begin{array}{rcl} E & \to & E : y^2 = x^3 - 15x + 22 \\ P = (x,y) & \mapsto & \begin{cases} \mathcal{O} \text{ if } P = \pm P_3 = (3, \pm 2), \\ \left(\frac{1}{-3}\left(x + \frac{24}{x-3} + \frac{16}{(x-3)^2}\right), \frac{y}{-3\sqrt{-3}}\left(1 + \frac{24}{(x-3)^2} + \frac{32}{(x-3)^3}\right)\right) \text{ otherwise.} \end{cases} \end{array} \tag{1.22}$$

The characteristic polynomial of ϕ is $\chi^2 + 3$. This is the curve we were looking for, the endomorphism corresponds to $[\sqrt{-3}]$.

3. $a_4 = \frac{4 \pm \sqrt{-11}}{27}a_2^2$, $j = -32768$. This is the curve in [LS12, LS13, Ex. A.5]. The characteristic polynomial of ϕ is $\chi^2 - \chi + 3$. The endomorphism corresponds on the curve to $\left[\frac{1+\sqrt{-11}}{2}\right]$. This curve will be useful in Sec. 2.3.2.2.
4. $a_4 = 2\frac{2 \pm 5\sqrt{-2}}{27}a_2^2$, $j = 8000$, this is the curve constructed in the previous paragraph. It can be found in another form in [LS12, LS13, Ex. A.4].

These special cases of curves with supplementary endomorphisms will be useful to identify some special cases in Sec. 2.3.

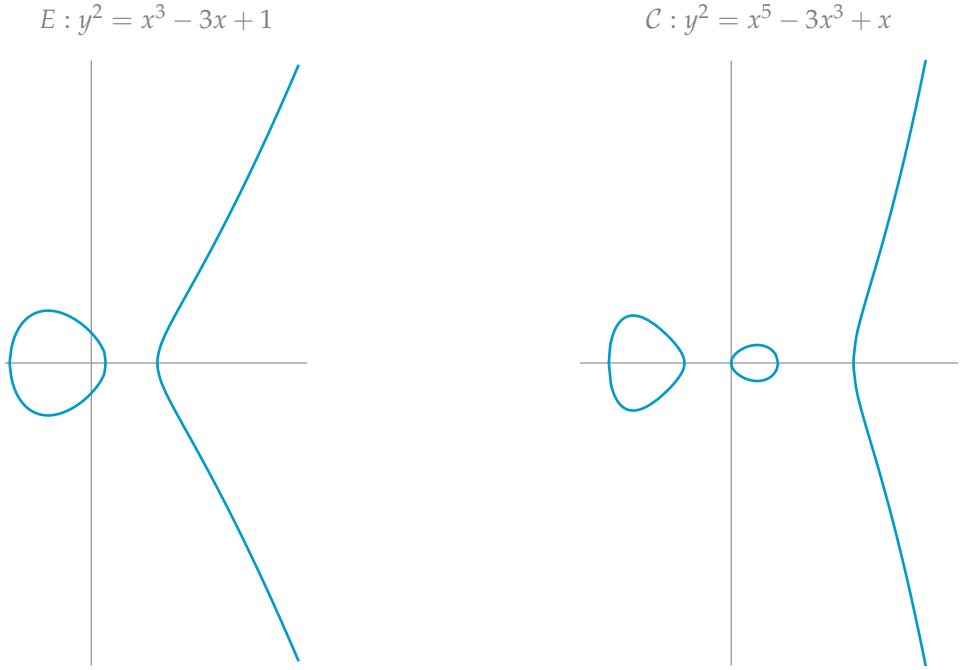
1.3 Genus 2 hyperelliptic curves

Definition 3. *A hyperelliptic genus 2 curve defined over a finite field $\mathbb{F}_q$ of characteristic greater than 2 is a curve defined by an affine equation of the form*

$$\mathcal{C} : y^2 = f(x),$$

with the polynomial f such that $\deg(f) = 5$ *or* 6 *and f has only simple roots over the algebraic closure of* $\mathbb{F}_q$.

For example we draw in Fig. 1.5 a representation of $\mathcal{C} : y^2 = x^5 - 3x^3 + x$.



(a) Elliptic (genus one) curve

(b) Genus two curve

Figure 1.5: An elliptic curve and a genus 2 hyperelliptic curve

Unlike elliptic curves, the points of a genus 2 curve like $\mathcal{C}$ never form a group. But there is a geometric group associated with any genus 2 curve $\mathcal{C}$: it is a two-dimensional object called the Jacobian $J_{\mathcal{C}}$. The abstract geometric definition of $J_{\mathcal{C}}$ is not very convenient for computing with, but we can identify its points with elements of a much more concrete group:

$$J_{\mathcal{C}}(\mathbb{F}_q) = \mathrm{Pic}^0(\mathcal{C})(\mathbb{F}_q).$$

In the rest of this section, we construct the group $\mathrm{Pic}^0(\mathcal{C})(\mathbb{F}_q)$, compute the group law, explain the Mumford representation for the elements, and give expressions for the number of elements in the group.

1.3.1 Divisors and Jacobian of a genus 2 curve

In this section we present the divisors on a genus 2 curve to be able to define a group with an addition law. On an elliptic curve, a divisor is directly identified to a point on the curve. On a genus 2 curve, a divisor is related to a tuple of points on the curve. The divisors are also involved in the definition of a pairing or bilinear map that we will introduce in Sec. 1.4. After the divisors we construct the degree-0 Picard group of the curve, this will be the group of the genus two curve. We will use this group in cryptography. We only are interested on genus two curves over finite fields. We refer to e.g. [BSS05, Ch. VII] for an introduction on this subject and to [ACD+05] for a complete description. The reader can refer to [HS00, Part A] for the theory over perfect fields.

First we need some facts about the function field of the curve. The following is taken from [Sil09, §II.1]. Let $\mathcal{C} : y^2 = F(x)$ be a genus 2 curve as defined above. For each point $P \in \mathcal{C}$, an ideal M_P of $\overline{\mathbb{F}_q}[\mathcal{C}] = \overline{\mathbb{F}_q}[x, y]/(y^2 - F(x))$ is defined by

$$M_P = \left\{ f \in \overline{\mathbb{F}_q}[\mathcal{C}] : f(P) = O \right\} .$$

M_P is a maximal ideal, since there is an isomorphism

$$\begin{array}{rcl} \overline{\mathbb{F}_q}[\mathcal{C}]/M_P & \to & \overline{\mathbb{F}_q} \\ f & \mapsto & f(P) \end{array}$$

Definition 4. *[Sil09, §II.1]. Let $\mathcal{C}$ be a smooth genus one or two curve defined over $\mathbb{F}_q$ and $P \in \mathcal{C}$. The (normalized) valuation on $\overline{\mathbb{F}_q}[\mathcal{C}]_P$ is given by*

$$\begin{aligned} \mathrm{ord}_P : \overline{\mathbb{F}_q}[\mathcal{C}]_P &\to \{0,1,2,\ldots\} \cup \{\infty\} \\ f &\mapsto \mathrm{ord}_P(f) = \sup\{d \in \mathbb{N} : f \in M_P^d\} \,. \end{aligned}$$

So we are interested on how f vanishes at P. Using $\mathrm{ord}_P(f/g) = \mathrm{ord}_P(f) - \mathrm{ord}_P(g)$, we extend ord_P to $\overline{\mathbb{F}_q}(\mathcal{C})$ (the function field of $\mathcal{C}$),

$$\mathrm{ord}_P : \overline{\mathbb{F}_q}(\mathcal{C}) \to \mathbb{Z} \cup \{\infty\} \,.$$

Definition 5. *[Sil09, §II.1]. Let $\mathcal{C}$ be a smooth genus one or two curve defined over $\mathbb{F}_q$, $P \in \mathcal{C}$ and let $f \in \overline{\mathbb{F}_q}(\mathcal{C})$ an element of the function field of the curve. The* order *of f at P is $\mathrm{ord}_P(f)$. If $\mathrm{ord}_P(f) > 0$ then f has a* zero *at P, and if $\mathrm{ord}_P(f) < 0$, then f has a* pole *at P. If $\mathrm{ord}_P(f) \geqslant 0$ then f is* regular *or* defined *at P and we can evaluate $f(P)$. Otherwise f has a pole at P and we can write $f(P) = \infty$.*

Proposition 5. *[Sil09, Prop. II.1.2]. Let $\mathcal{C}$ be a smooth genus one or two curve defined over $\mathbb{F}_q$ and $f \in \overline{\mathbb{F}_q}(\mathcal{C})$ with $f \neq 0$. Then there are only finitely many points of $\mathcal{C}$ at which f has a pole or zero. Further, if f has no pole, then $f \in \overline{\mathbb{F}_q}$.*

This proposition will be useful in the following. Next we define the group of divisors of the curve. There will be a correspondence between the elements $f \in \overline{\mathbb{F}_q}(\mathcal{C})$ and a subgroup of the divisor group of the curve.

Definition 6 (Divisor [Sil09, §II.3])**.** *Let $\mathcal{C}$ be a smooth genus one or two curve defined over $\mathbb{F}_q$. The* divisor group of $\mathcal{C}$*, denoted by* $\mathrm{Div}(\mathcal{C})$ *is the free abelian group generated by the points of $\mathcal{C}$. A* divisor $\mathcal{D} \in \mathrm{Div}(\mathcal{C})$ *is a finite formal sum of points*

$$\mathcal{D} = \sum_{P \in \mathcal{C}} n_P(P) \text{ with } n_P \in \mathbb{Z},\ n_P = 0 \text{ for all but finitely many } P \in \mathcal{C} \,.$$

The degree *of a divisor $\mathcal{D}$ is*

$$\deg(\mathcal{D}) = \sum_{P \in \mathcal{C}} n_P \,.$$

The divisors of degree 0 *form a subgroup of* $\mathrm{Div}(\mathcal{C})$*, denoted by*

$$\mathrm{Div}^0(\mathcal{C}) = \{\mathcal{D} \in \mathrm{Div}(\mathcal{C}),\ \deg(\mathcal{D}) = 0\} \,.$$

Let π_q be the Frobenius map $\mathcal{C} \to \mathcal{C}$, $P = (x, y) \mapsto \pi_q(P) = (x^q, y^q)$. Let π_q act on $\mathrm{Div}(\mathcal{C})$ in the following way: $\pi_q(\mathcal{D}) = \sum_{P \in \mathcal{C}} n_P(\pi_q(P))$. Then $\mathcal{D}$ *is defined over* $\mathbb{F}_q$ if $\pi_q(\mathcal{D}) = \mathcal{D}$.

We note that this does not mean that $P_i \in \mathcal{C}(\mathbb{F}_q)$ for all P_i of $\mathcal{D}$. It suffices for π_q to permute the P_i in an appropriate way. For example, if $\mathcal{C}$ is defined over $\mathbb{F}_q$, let $P \in \mathcal{C}(\mathbb{F}_{q^2})$ and let $\mathcal{D} = (P) + \pi_q(P)$. Then $\pi_q(\mathcal{D}) = (\pi_q(P)) + (\pi_{q^2}(P)) = (\pi_q(P)) + (P) = \mathcal{D}$ and $\mathcal{D}$ is defined over $\mathbb{F}_q$ while P and $\pi_q(P)$ are not.

We denote the group of divisors defined over $\mathbb{F}_q$ by $\mathrm{Div}_{\mathbb{F}_q}(\mathcal{C})$ and similarly for $\mathrm{Div}^0_{\mathbb{F}_q}(\mathcal{C})$. The following explains the correspondence between divisors on the curve $\mathcal{C}$ and elements f in the function field of $\mathcal{C}$.

Let $f \in \overline{\mathbb{F}_q}(\mathcal{C})^*$ an element in the function field of $\mathcal{C}$. Then we associate to f the divisor $\mathrm{div}(f)$ given by

$$\mathrm{div}(f) = \sum_{P \in \mathcal{C}} \mathrm{ord}_P(f)(P) \,. \tag{1.23}$$

This is a divisor (in particular the sum is finite) thanks to Prop. 5. We can see that $\mathrm{div}(\pi_q(f)) = \pi_q(\mathrm{div}(f))$. In particular, if $f \in \mathbb{F}_q(\mathcal{C})$, then $\mathrm{div}(f) \in \mathrm{Div}_{\mathbb{F}_q}(\mathcal{C})$. This formula (1.23) means that the divisor of $f = f_{num}/f_{den}$ is made of the intersection points of f_{num} and $\mathcal{C}$ for the zeros and the intersection points of f_{den} and $\mathcal{C}$ for the poles, counted with their multiplicities.

Since each ord_P is a valuation, the map

$$\mathrm{div} : \overline{\mathbb{F}_q}(\mathcal{C})^* \to \mathrm{Div}(\mathcal{C})$$

is a homomorphism of abelian groups.

We continue with a few terminology. The *support* of a divisor $\mathcal{D} = \sum_i n_i(P_i)$ is the finite set of points $\{P_i\}_{i\in I}$, $P_i \in \mathcal{C}$ such that $n_i \neq 0$, i.e. all the points arising in the effective expression of $\mathcal{D}$.

We then obtain this important definition.

Definition 7. *Principal divisors and related definitions[Sil09, Sec. II.3]*

- *A divisor* $\mathcal{D} \in \mathrm{Div}(\mathcal{C})$ *is* principal *if it has the form* $\mathcal{D} = \mathrm{div}(f)$ *for some* $f \in \overline{\mathbb{F}_q}(\mathcal{C})$. *The principal divisors form a subgroup of* $\mathrm{Div}(\mathcal{C})$.
- *Two divisors* $\mathcal{D}_1, \mathcal{D}_2$ *are* linearly equivalent, *written* $\mathcal{D}_1 \sim \mathcal{D}_2$, *if* $\mathcal{D}_1 - \mathcal{D}_2$ *is principal.*
- *The* divisor class group *or* Picard group *of* $\mathcal{C}$, *denoted by* $\mathrm{Pic}(\mathcal{C})$, *is the quotient of* $\mathrm{Div}(\mathcal{C})$ *by its subgroup of principal divisors.*
- *We let* $\mathrm{Pic}_{\mathbb{F}_q}(\mathcal{C})$ *be the subgroup of* $\mathrm{Pic}(\mathcal{C})$ *fixed by* π_q.

Example 4. *Degree zero divisors. Given a function* $f \in \overline{\mathbb{F}_q}(\mathcal{C})^*$, *as already said in* (1.23) *the associated principal divisor is* $\mathcal{D} = \mathrm{div}(f) = \sum_{P_i\in\mathcal{C}} n_i(P_i)$. *The* P_i *with* $n_i > 0$ *are the zeros of the function* f, *of order* n_i *and the* P_j *with* $n_j < 0$ *are the poles of* f, *of order* $-n_j$.

1. *Let* $E : y^2 = x^3 - 3x - 8$ *be an elliptic curve defined over* $\mathbb{F}_q = \mathbb{F}_{127}$. *We have* $\#E(\mathbb{F}_{127}) = 109$. *Let* $f = \frac{7x+5y+3}{8x+6y+4}$ *in the function field of the curve. We compute the divisor of* f. *We solve the system*

$$\begin{cases} 7x + 5y + 3 = 0 \\ y^2 - x^3 + 3x + 8 = 0 \end{cases}$$

to get the zeros of $\mathrm{div}(f)$. *We obtain three points* $P_1 = (92, 23), P_2 = (70, 3), P_3 = (33, 4)$ *(a line intersects an elliptic curve in three points). The numerator* $f_{num} = 7x + 5y + 3$ *has three zeros at* P_1, P_2, P_3 *and three poles at infinity:* $\mathrm{div}(f_{num}) = (P_1) + (P_2) + (P_3) - 3P_\infty$. *We do the same with the denominator, we solve*

$$\begin{cases} 8x + 6y + 4 = 0 \\ y^2 - x^3 + 3x + 8 = 0 \end{cases}$$

and find the three points $Q_1 = (75, 111), Q_2 = (66, 123), Q_3 = (16, 105)$ *so* $\mathrm{div}(f_{den}) = (Q_1) + (Q_2) + (Q_3) - 3P_\infty$. *Then* $\mathrm{div}(f) = \mathrm{div}(f_{num}) - \mathrm{div}(f_{den}) = (P_1) + (P_2) + (P_3) - (Q_1) - (Q_2) - (Q_3)$ *(the* P_∞ *cancel out).*

2. *Let* E *be an elliptic curve defined over* $\mathbb{F}_q$, $P_1, P_2 \in E$ *with* $P_1 \neq P_2, P_1 \neq -P_2$ *and* $\mathcal{D} = (P_1) - (P_2)$. *Then* $\mathcal{D}$ *is a non-principal degree 0 divisor in* $\mathrm{Div}(E)$. *The two points* $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ *define a line. This line can be expressed by a linear polynomial* $f = (y_2 - y_1)x - (x_2 - x_1)y + y_1x_2 - x_1y_2$ *in* $\mathbb{F}_q(E)$. *Bezout's theorem tells that a line intersects the elliptic curve in three points counted with multiplicity. If we denote by* P_3 *the third intersection point of* E *and the line through* P_1 *and* P_2, *then* $(P_1) + (P_2) + (P_3) - 3P_\infty$ *is a principal divisor.*
3. *Let* $\mathcal{C}$ *a genus 2 curve of the form* $y^2 = F(x)$ *defined over* $\mathbb{F}_q$ *and let* $P \in \mathcal{C}(\mathbb{F}_q)$. *The involution* $i(P)$ *sends* $P = (x_P, y_P)$ *to* $(x_P, -y_P)$. *Define the divisor* $\mathcal{D} = (P) + (i(P)) - \mathcal{D}_\infty$ *with* $D_\infty = 2(P_\infty)$ *or* $\mathcal{D}_\infty = (P_\infty^+) + (P_\infty^-)$. *Then* $\mathcal{D}$ *is a principal divisor. The corresponding function in* $\mathbb{F}_q(\mathcal{C})$ *is* $f = x - x_P$ *which can be written in projective coordinates* $\frac{xz_P - x_Pz}{z_Pz}$.
4. *Let* $\mathcal{C}$ *as in the previous example and let* $P_1, P_2 \in \mathcal{C}$ *be two points not at infinity. Then* $\mathcal{D} = (P_1) + (P_2) - \mathcal{D}_\infty$ *is a non-principal divisor unless* $P_1 = P_2$ *or* $P_1 = i(P_2)$.

Proposition 6 ([Sil09, Prop. II.3.1])**.** *Let* $\mathcal{C}$ *be a smooth genus one or two curve defined over* $\mathbb{F}_q$ *and let* $f \in \overline{\mathbb{F}_q}(\mathcal{C})^*$.

1. $\mathrm{div}(f) = 0$ *if and only if* $f \in \overline{\mathbb{F}_q}^*$, *i.e.* f *is constant.*
2. $\deg(\mathrm{div}(f)) = 0$.

We will use these two properties to define the addition law.

1. Let $f_1, f_2 \in \overline{\mathbb{F}_q}(\mathcal{C})^*$ be two functions of principal divisors denoted by $\mathcal{D}_1, \mathcal{D}_2$. Then the divisor of $f_1 \cdot f_2$ is $\mathcal{D}_1 + D_2$ and the divisor of f_1/f_2 is $\mathcal{D}_1 - D_2$.
2. Let $f \in \overline{\mathbb{F}_q}(\mathcal{C})^*$ whose principal divisor is denoted $\mathcal{D}_f$. Then $\mathrm{div}(f^m) = m\mathcal{D}_f$.

To conclude, we have the following definition.

Definition 8 ([Sil09, §II.3])**.** *We define the* degree-0 part of the divisor class group of $\mathcal{C}$ *to be the quotient of* $\mathrm{Div}^0(\mathcal{C})$ *(the degree-0 divisors of* $\mathcal{C}$*) by the subgroup of principal divisors. We denote this group by* $\mathrm{Pic}^0(\mathcal{C})$*. Similarly, we write* $\mathrm{Pic}^0_{\mathbb{F}_q}(\mathcal{C})$ *for the subgroup of* $\mathrm{Pic}^0(\mathcal{C})$ *fixed by* π_q*.*

Finally, the group of the curve called the Jacobian is identified with the degree-0 Picard group.

$$J_{\mathcal{C}}(\mathbb{F}_q) = \mathrm{Pic}^0_{\mathbb{F}_q}(\mathcal{C})\ . \tag{1.24}$$

We present in the next section (Sec. 1.3.2) the Mumford representation for elements in $\mathrm{Pic}^0_{\mathbb{F}_q}(\mathcal{C})$ to handle a divisor in practice and be able to compute easily the addition law on $J_{\mathcal{C}}(\mathbb{F}_q)$.

1.3.2 Mumford representation of divisors

Mumford introduced [Mum83] a representation of divisors on hyperelliptic curves which we present here. This representation gives another interpretation of the group law and better algorithms to compute it. For simplifications, we assume that the curve is of the form $\mathcal{C} : y^2 = F(x)$ with F of degree 5. This means that the curve has one point at infinity. One can refer to the work of Galbraith, Harrison and Mireles-Morales [GHMM08] for the general case (a curve with two points at infinity).

Proposition 7 ([BSS05, Prop. VII.1] and [ACD+05, Th. 4.145])**.** *Let* $\mathcal{C}$ *be a hyperelliptic curve of genus 2 defined over a field* $\mathbb{F}_q$ *with equation* $y^2 = F(x)$ *and* F *of degree 5. Then the elements of the Jacobian of* $\mathcal{C}$ *that are defined over* $\mathbb{F}_q$ *are in one-to-one correspondence with the pairs of polynomials* $(u(X), v(X))$ *with coefficients in* $\mathbb{F}_q$*, such that* $\deg(v) < \deg(u) \leqslant 2$*, the polynomial* u *is monic, and* u *divides* $v^2 - F$*. If* u *and* v *are two polynomials that satisfy these conditions, the corresponding element of* $J_{\mathcal{C}}$ *is denoted by* $\mathcal{D} = \mathrm{div}(u, v)$*.*

By the Riemann–Roch theorem [Sil09, §II.5], every divisor class has a unique *reduced representative*: that is, a representative in the form $\mathcal{D} = \sum_{i=1}^{\ell} P_i - \ell P_\infty$, with $P_i \in \mathcal{C}$, where $P_i \neq P_\infty$, $P_i \neq -P_j$ for $i \neq j$ and $\ell \leqslant 2$.

Definition 9 (Mumford representation from [ACD+05, §4.4.7])**.** *Let* $\mathcal{C}$ *be a hyperelliptic curve of genus 2 defined over a field* $\mathbb{F}_q$ *with equation* $y^2 = F(x)$ *and* F *of degree 5. Let* $\mathcal{D}$ *be a divisor on the Jacobian* $J_{\mathcal{C}}$*, uniquely represented by* $\mathcal{D} = \sum_{i=1}^{\ell} P_i - \ell P_\infty$*. Put* $P_i = (x_i, y_i)$*. Then the corresponding polynomials* u *and* v *of Th. 7 are defined by*

$$u(X) = \prod_{i=1}^{\ell} (X - x_i)$$

and the property that if P_i *occurs* n_i *times then*

$$\left(\frac{\mathrm{d}}{\mathrm{d}X}\right)^j \Big[v(X)^2 - F(X)\Big]_{X=x_i} = 0, \ \textit{for } 0 \leqslant j \leqslant n_i - 1\ .$$

In practice, for a genus 2 hyperelliptic curve of the form $y^2 = F(x)$ and $\deg(F) = 5$, we obtain this Mumford representation, with $\ell = g = 2$, $\mathcal{D} = (P_1) + (P_2) - 2P_\infty$, $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$:

$$u(X) = (X - x_1)(X - x_2) = X^2 - (x_1 + x_2)X + x_1 x_2\ ,$$

and we can also write only the two coefficients and recall that

$$u = (u_1, u_0) = (-(x_1 + x_2), x_1 x_2)\ .$$

We can see in practice here that $\mathcal{D}$ can be made of points with coordinates in an extension of $\mathbb{F}_q$ but still $\mathcal{D} \in J_{\mathcal{C}}(\mathbb{F}_q)$. We take again the example $P = (x, y) \in \mathcal{C}(\mathbb{F}_{q^2})$ and $\mathcal{D} = (P) + (\pi_q(P)) - 2P_\infty$. Then $u(X) = X^2 - (x + \pi_q(x))X + x\pi_q(x) = X^2 - \mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(x)X + \mathrm{Norm}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(x)$ has coefficients in $\mathbb{F}_q$.

We can compute the second polynomial v of the Mumford representation with Lagrange interpolation [ACD+05, §14.1.2]:

$$v(X) = \sum_{i=1}^{2} \frac{\prod_{j \neq i}(X - x_j)}{\prod_{j \neq i}(x_i - x_j)} y_i$$

which turns into

$$v(X) = \frac{X - x_2}{x_1 - x_2} y_1 + \frac{X - x_1}{x_2 - x_1} y_2 = \frac{y_1 - y_2}{x_1 - x_2} X + \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} .$$

As for u we can denote the two coefficients of v:

$$v = (v_1, v_0) = \left(\frac{y_1 - y_2}{x_1 - x_2}, \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \right) .$$

In Chapter 2 we will use this notation:

$$\begin{aligned} \mathcal{D} = (P_1, P_2) &= (u_1, u_0, v_1, v_0) \\ &= \left(-(x_1 + x_2),\ x_1 x_2,\ \frac{y_1 - y_2}{x_1 - x_2},\ \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \right) . \end{aligned} \tag{1.25}$$

1.3.3 Characteristic polynomial of the Frobenius endomorphism

This section is about the properties of the Frobenius endomorphism on the Jacobian of a genus 2 curve, in the same way as in Sec. 1.2.6. Let $\mathcal{C}$ a genus 2 curve defined over a finite field $\mathbb{F}_q$ and let $J_\mathcal{C}$ its Jacobian. Knowing the coefficients of this characteristic polynomial, we can compute $\#J_\mathcal{C}(\mathbb{F}_q)$ and $\#J_\mathcal{C}(\mathbb{F}_{q^k})$ for any $k > 1$. This will be useful in Sec. 2.3 where we are interested in computing $\#J_{\mathcal{C}_1}(\mathbb{F}_q)$ knowing $\#J_{\mathcal{C}_1}(\mathbb{F}_{q^8})$ and $\#J_{\mathcal{C}_2}(\mathbb{F}_q)$ knowing $\#J_{\mathcal{C}_2}(\mathbb{F}_{q^6})$ for two genus 2 curves $\mathcal{C}_1, \mathcal{C}_2$ defined over a field $\mathbb{F}_q$.

The Frobenius endomorphism is defined as

$$\begin{aligned} \pi_q : J_\mathcal{C} &\to J_\mathcal{C} \\ \mathcal{D} = (u_1, u_0, v_1, v_0) &\mapsto (u_1^q, u_0^q, v_1^q, v_0^q) . \end{aligned}$$

The characteristic polynomial $\chi_{\mathcal{C},\pi_q}$ of the Frobenius endomorphism π_q is of the form

$$\chi_{\mathcal{C},\pi_q}(T) = T^4 - a_q T^3 + b_q T^2 - q a_q T + q^2 \tag{1.26}$$

with a_q, b_q integers satisfying the Weil bounds: $|a_q| \leqslant 4\sqrt{q}$ and $|b_q - 2q| \leqslant 4q$. Compared to the characteristic polynomial of the Frobenius endomorphism over elliptic curves, two coefficients a_q, b_q are involved here, instead of one (the trace t). This polynomial $\chi_{\mathcal{C},\pi_q}$ is a Weil polynomial: its four roots $z_{i,q}$ have norm $|z_{i,q}| = \sqrt{q}$. For simplicity in the following, we order the roots pairwise such that:

$$z_{1,q} z_{2,q} = q,\ z_{3,q} z_{4,q} = q .$$

A divisor $\mathcal{D}$ is on $J_\mathcal{C}(\mathbb{F}_q)$ if and only if $\pi_q(\mathcal{D}) = \mathcal{D}$ so

$$\#J_\mathcal{C}(\mathbb{F}_q) = \chi_{\mathcal{C},\pi_q}(1) = q^2 + 1 - (q+1) a_q + b_q .$$

Similarly, a divisor $\mathcal{D}$ is in $J_\mathcal{C}(\mathbb{F}_{q^k})$ iff $\pi_{q^k}(\mathcal{D}) = \mathcal{D}$ so $\#J_\mathcal{C}(\mathbb{F}_{q^k}) = \chi_{\mathcal{C},\pi_{q^k}}(1)$.

We can compute the coefficient a_{q^k} of $\chi_{\mathcal{C},\pi_{q^k}}$ knowing a_q, b_q from $\chi_{\mathcal{C},\pi_q}$ exactly as we did in Sec. 1.2.6 for elliptic curves. The difference is that the Newton's recurrence formulas are four-step instead of two-step.

$$\begin{aligned} a_{q^2} &= (a_q)^2 - 2b_q \\ a_{q^3} &= a_q a_{q^2} - b_q a_q + 3q a_q \\ a_{q^4} &= a_q a_{q^3} - b_q a_{q^2} + q a_q a_q - 4q^2 \\ a_{q^k} &= a_q a_{q^{k-1}} - b_q a_{q^{k-2}} + q a_q a_{q^{k-3}} - q^2 a_{q^{k-4}} . \end{aligned}$$

We can also compute b_{q^k} with $b_{q^k} = \frac{1}{2}((a_{q^k})^2 - a_{q^{2k}})$. The Frobenius π_{q^k} has characteristic polynomial

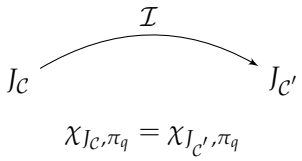
$$\chi_{\mathcal{C},\pi_{q^k}}(T) = T^4 - a_{q^k} T^3 + b_{q^k} T^2 - q^k a_{q^k} T + q^{2k} .$$

We will also use formally the other representation of $\chi_{\mathcal{C},\pi_{q^k}}$:

$$\begin{aligned} \chi_{\mathcal{C},\pi_q}(T) &= (T - z_{1,q})(T - z_{2,q})(T - z_{3,q})(T - z_{4,q}); \\ \chi_{\mathcal{C},\pi_{q^k}}(T) &= (T - z_{1,q}^k)(T - z_{2,q}^k)(T - z_{3,q}^k)(T - z_{4,q}^k) . \end{aligned} \tag{1.27}$$

We saw in the introduction on elliptic curves that two isogenous elliptic curves have the same characteristic polynomial of Frobenius endomorphism. This holds for isogenous Jacobians. This comes from results on Honda-Tate theory. This theory is more general (not only about elliptic curves and Jacobians) and was developed in 1966–1968 in [Tat66, Hon68, Tat68]. A short summary can be found in [Bis11, §II.4]. Here is the important theorem we will need in Sec. 2.3.

Theorem 3. *[Tat66], from [Bis11, Th. II.4.3] Two Jacobians are isogenous if and only if their respective Frobenius endomorphisms have the same characteristic polynomial.*



1.4 Pairings

In this introduction we point out a few historical facts on pairings in cryptography. We suggest also interesting bibliographical references. Then in Sec. 1.4.1 we present the black-box properties of pairings widely used in cryptography. The mathematical prerequisites are presented in Sec. 1.3.1. The Weil and Tate pairings are defined in Sec. 1.4.2. All the pairing variants used in cryptography are derived from these two definitions. The construction of curves suitable for pairing computation is not trivial. An overview of the main methods is provided in Sec.1.4.3. Finally in Sec. 1.4.4 the algorithm to compute a Tate pairing is explained, with its various improvements for practical use in cryptography.

For independent interest, we recall here some historical facts. Bilinear pairings were defined the first time in algebraic geometry, in particular over elliptic curves. The first pairing was introduced in 1948 by the French mathematician André Weil. He gave the name *accouplement* to this map. In 1986, Victor S. Miller worked on the Weil pairing and found a practical algorithm to compute it. His work was recently published in [Mil04]. In 1988 Kaliski was the first to implement the Weil pairing in Macsyma. The source code is available in his PhD thesis [Kal88]. He used it for example to decide whether an elliptic curve has a cyclic group of points. Building on Miller' and Kaliski's work, Menezes, Okamoto and Vanstone presented in 1993 in [MOV93] an attack on supersingular elliptic curves to compute very efficiently discrete logarithms. Two years later, Frey and Rück [FR94] proposed to compute a Tate pairing to speed-up this attack. The main property used here is that the pairing embeds the discrete logarithm problem in the group $E(\mathbb{F}_q)$ to a quite small finite field, namely $\mathbb{F}_{q^2}$. In this field the discrete logarithm problem is vulnerable to more efficient attacks than in the elliptic curve.

A mathematical presentation of the Weil pairing can be found in [Sil09, §III.8]. We recommend to look first at Galbraith's chapter [BSS05, Ch. IX]. For less-theoretical (but more technical) proofs than in Silverman's book, we suggest to read Washington's book [Was03]. The mathematical definition of a pairing contains many new theoretical notions. We will present here the properties of pairings that are used in cryptography and hope this will encourage the reader to look at the mathematics after that.

1.4.1 Black-box properties

Definition 10 (Pairing [BSS05, IX.1])**.** *Let $(\mathbb{G}_1, +), (\mathbb{G}_2, +)$ and $(\mathbb{G}_T, \cdot)$ be three cyclic groups of same order. A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ which is*

1. *bilinear: $e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q)$ and $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$.*
2. *non-degenerate: For all $P \neq \mathcal{O} \in \mathbb{G}_1$, there is some $Q \in \mathbb{G}_2$ such that $e(P, Q) \neq 1$. For all $Q \neq \mathcal{O} \in \mathbb{G}_2$, there is some $P \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$.*
3. *efficiently computable (in polynomial time in the input size).*

We note that if the three groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are of prime order, then $e(P, Q) = 1$ implies that $P = \mathcal{O}$ or $Q = \mathcal{O}$. This is *not* true if the group order is not prime (e.g. is an RSA modulus). A pairing satisfies the following properties (this is a straightforward consequence of the definition):

- $e(P,\mathcal{O}) = e(\mathcal{O},Q) = 1$,
- $e(-P,Q) = e(P,Q)^{-1} = e(P,-Q)$,
- $e([a]P,[b]Q) = e(P,Q)^{ab} = e([b]P,[a]Q)$ for all $a,b \in \mathbb{Z}$. This property is widely used in protocol design.

We now develop the main idea of the MOV and FR attacks. The ECDLP (Sec. 1.1) in $E(\mathbb{F}_q)$ takes in two points P,S to compute the scalar s such that $S = [s]P$. If the curve is supersingular, a pairing is available on the curve. Moreover there exists an explicit isomorphism from $\mathbb{G}_1$ into $\mathbb{G}_2$ provided by the distortion map. In cryptography, authors say that the pairing is symmetric with $\mathbb{G}_1 = \mathbb{G}_2$. We have $\mathbb{G}_1 = E(\mathbb{F}_q)$, $\mathbb{G}_2$ is isomorphic to $\mathbb{G}_1$ through the distorsion map ϕ which sends $P \in \mathbb{G}_1$ to $\phi(P) \in \mathbb{G}_2$ and $\mathbb{G}_T = \mathbb{F}_{q^k}$ with k small ($k \in \{2,3,4,6\}$). Then s satisfies $e(P,S) = e(P,\phi(P))^s$. The ECDLP of S in base P can be transformed into computing the discrete logarithm of $y = e(P,S) \in \mathbb{G}_T$ in base $g = e(P,\phi(P))$. If E is a supersingular curve defined over a prime field $\mathbb{F}_p$ with $\log p = 160$, then $k = 2$ and $\mathbb{G}_T = \mathbb{F}_{p^2}$ is a finite field of 320 bits. The discrete logarithm in a finite field of such size was already computable in reasonable time (weeks or months on a PC) in the 90's [JL07, Tab. 6].

In 2000 at the *SCIS* conference in Japan, Sakai, Ohgishi and Kasahara presented an ID-based cryptosystem using the Weil pairing [RSK00]. However the history recalls mostly the 3-partite Diffie-Hellman key exchange (Triffie-Hellman) introduced in 2000 by Joux [Jou00, Jou04] and the identity-based encryption of Boneh and Franklin [BF01] as the first use of pairing as a new tool in cryptography. This was the beginning of a prolific area in cryptography.

1.4.2 Weil and Tate pairings

The Weil and Tate pairings are bilinear maps on curves defined over a field $\mathbb{K}$. We rewrite here the presentation in [Sil09, III.8]. We will also need the definition of *divisors* on a curve presented in Sec. 1.3.1.

Let E be an elliptic curve defined over a field $\mathbb{K}$. Let $m \geqslant 2$ be an integer coprime to $p = \mathrm{char}(\mathbb{K})$ if $p > 0$. Define the group of $\mathbb{K}$-rational m-torsion points of the curve to be

$$E(\mathbb{K})[m] = \{P \in E(\mathbb{K}),\ [m]P = \mathcal{O}\}\ .$$

We need to characterize the structure of $E[m]$, the m-torsion points over an algebraic closure of $\mathbb{K}$. We have this very useful result.

Proposition 8 ([Sil09, Corollary 6.4, III.6])**.** *Let E be an elliptic curve defined over a field $\mathbb{K}$ and let $m \in \mathbb{Z}$ with $m \neq 0$.*

1. $\deg([m]) = m^2$*, i.e. the multiplication-by-m map has degree* m^2*.*
2. *If $m \neq 0$ in $\mathbb{K}$, i.e. if either* $\mathrm{char}(\mathbb{K}) = 0$ *or* $p = \mathrm{char}(\mathbb{K}) > 0$ *and* $p \nmid m$*, then the m-torsion points of E over an algebraic closure of $\mathbb{K}$ are*
$$E[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}\ .$$

Let $T \in E[m]$. There exists a function $f \in \overline{\mathbb{K}}(E)$ such that

$$\mathrm{div}(f) = m(T) - m(\mathcal{O})\ .$$

We will present a method to compute it in Sec. 1.4.4.

This function has a zero of order m at T and a pole of order m at $\mathcal{O}$. Letting $T' \in E$ with $[m]T' = T$, there is similarly a function $g \in \overline{\mathbb{K}}(E)$ satisfying

$$\mathrm{div}(g) = [m]^*(T) - [m]^*(\mathcal{O}) = \sum_{R \in E[m]} (T' + R) - (R)\ .$$

The notation $[m]^*(T)$ means that we consider the pre-image of T under the map $[m]$. The point $T' \in E$ is chosen such that $[m]T' = T$. Observe that $[m^2]T' = \mathcal{O}$ which means that we can choose T' as an arbitrary m^2-torsion point. To enumerate the pre-images of T under $[m]$ we simply enumerate all the points $T' + R$ with R an m-torsion point. We have $[m](T' + R) = [m]T' + [m]R = T + \mathcal{O} = T$. We can also write

$$\begin{aligned}\mathrm{div}(g) &= [m]^*(T) - [m]^*(\mathcal{O}) \\ &= (T' + R_1) + (T' + R_2) + (T' + R_3) + \ldots + (T' + R_{m^2}) \\ &\quad -(R_1) - (R_2) - (R_3) - \ldots - (R_{m^2})\ .\end{aligned}$$

The function g has m^2 distinct zeros at $T' + R_i$ and m^2 distinct poles at R_i with R_i enumerating the m-torsion points on E. There are m^2 such m-torsion points, i.e. $\#E[m] = m^2$. Now we consider the function $f \circ [m]$. The zeros of this function are the points S such that $f([m]S) = 0$, i.e. such that $[m]S = T$. These points are exactly the points $T' + R$ which are zeros of g. These zeros are of order m. The poles of $f \circ [m]$ are the points S such that $[m]S$ is a pole of f. The function f has a pole of order m at $\mathcal{O}$ hence the poles of $f \circ [m]$ are the m^2 points of order m and they have order m. We deduce that the function $f \circ [m]$ has m^2 zeros of order m at the points $T' + R$ with T' such that $[m]T' = T$ and $R \in E[m]$. The function $f \circ [m]$ has m^2 poles of order m at the points R with R an m-torsion point. Hence

$$\mathrm{div}(f \circ [m]) = m \, \mathrm{div}(g) \, .$$

The functions $f \circ [m]$ and g^m have the same divisor, so up to a multiplication by an element of $\overline{\mathbb{K}}^*$ (by Prop. 6), we may assume that

$$f \circ [m] = g^m \, .$$

Now suppose that $S \in E[m]$ is another m-torsion point ($S = T$ is allowed). Then for any point $X \in E$, $g(X+S)^m = f([m]X + [m]S) = f([m]X) = g(X)^m$. We deduce that $g(X+S)/g(X)$ is an m-th root of unity.

Definition 11 (Weil pairing (accouplement de Weil) [Sil09, III.8])**.** *We define a pairing*

$$e_{\mathrm{Weil},m} : E[m] \times E[m] \to \boldsymbol{\mu}_m$$

with $\boldsymbol{\mu}_m$ the group of m^{th} roots of unity by setting

$$e_{\mathrm{Weil},m}(S,T) = g(X+S)/g(X),$$

where $X \in E$ is any point such that $g(X+S)$ and $g(X)$ are both defined and non-zero. Note that although g is only defined up to multiplication by an element of $\overline{\mathbb{K}}^*$, $e_{\mathrm{Weil},m}(S,T)$ does not depend on this choice. This pairing is called the Weil pairing.

There is a second definition [Sil09, §III.8 Remark 8.5] which can be proven equivalent to the first one. Choose arbitrary points $X, Y \in E$ and functions $f_S, f_T \in \overline{\mathbb{K}}(E)$ satisfying

$$div(f_S) = m(X+S) - m(X) \text{ and } div(f_T) = m(Y+T) - m(Y) \, .$$

Then

$$e_{\mathrm{Weil},m}(S,T) = \frac{\dfrac{f_S(Y+T)}{f_S(Y)}}{\dfrac{f_T(X+S)}{f_T(X)}} \, .$$

The value $e_m(S,T)$ is well-defined which means that it does not depend on the choice of X and Y but only on the two points S and T.

This second definition of Weil pairing is close to the Tate pairing definition that we will present in the sequel. We state here the presentation given in [BSS05, IX.3]. Let E be an elliptic curve over a field $\mathbb{K}_0$. Let m be a positive integer which is coprime to the characteristic of the field $\mathbb{K}_0$. The set of m-th roots of unity is defined to be $\boldsymbol{\mu}_m = \left\{ u \in \overline{\mathbb{K}_0}^*, \; u^m = 1 \right\}$. Define the field $\mathbb{K} = \mathbb{K}_0(\boldsymbol{\mu}_m)$ to be the extension of $\mathbb{K}_0$ generated by the m-roots of unity. We define the group

$$mE(\mathbb{K}) = \{[m]P, \; P \in E(\mathbb{K})\} \, .$$

We need to consider the quotient group $E(\mathbb{K})/mE(\mathbb{K})$. We can see it as the set of points on $E(\mathbb{K})$ up to a point in $mE(\mathbb{K})$. In other words, two points P_1, P_2 on the curve $E(\mathbb{K})$ represent the same equivalence class of $E(\mathbb{K})/mE(\mathbb{K})$ if $P_1 - P_2 \in mE(\mathbb{K})$, i.e. there exists a point P' in $E(\mathbb{K})$ such that $P_1 - P_2 = [m]P'$.

Let $P \in E(\mathbb{K})[m]$ and $Q \in E(\mathbb{K})$, in a way Q is a representative of a class in $E(\mathbb{K})/mE(\mathbb{K})$. Since $[m]P = \mathcal{O}$, there exists a function f such that its divisor is $(f) = m(P) - m(\mathcal{O})$. This is a degree 0 divisor. Let $\mathcal{D}$ be any degree zero divisor equivalent to $(Q) - (\mathcal{O})$ such that $\mathcal{D}$ is defined over $\mathbb{K}$ and the support of $\mathcal{D}$ is disjoint from the support of (f) (i.e. there is no common point between the points describing $\mathcal{D}$ and the zeros and poles of f).

Definition 12 (Tate pairing, [BSS05, IX.3]). *The Tate pairing is the map*

$$\begin{array}{rcl} \langle \cdot, \cdot \rangle_m : E(\mathbb{K})[m] \times E(\mathbb{K})/mE(\mathbb{K}) & \to & \mathbb{K}^*/(\mathbb{K}^*)^m \\ (P, Q) & \mapsto & \langle P, Q \rangle_m = f(\mathcal{D}) \end{array}$$

The pairing value is a representative of an equivalence class. In cryptography, e.g. for any key agreement protocol, we need a unique output value. The *reduced Tate pairing* (Def. 14) over finite fields is introduced for this purpose. We need before that to explicit the groups $\mathbb{K}^*$ and $E(\mathbb{K})[m]$. We first give an important definition.

Definition 13 (Embedding degree [BSS05, IX.5])**.** *Let E be an elliptic curve defined over a finite field $\mathbb{K}_0 = \mathbb{F}_q$. Let m be an integer coprime to q which divides $\#E(\mathbb{F}_q)$. Let $\mathbb{K} = \mathbb{F}_q(\boldsymbol{\mu}_m)$ be the finite field extension of $\mathbb{F}_q$ generated by the m-roots of unity. We define the* embedding degree *k to be the integer such that $\mathbb{K} = \mathbb{F}_{q^k}$.*

Proposition 9. *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, m be an integer coprime to q s.t. $m \mid \#E(\mathbb{F}_q)$ and k be the embedding degree of E with respect to q and m. Then k is also the smallest positive integer such that m divides $q^k - 1$.*

Thanks to the properties of m-torsion points (Prop. 8), we state now this important result.

Theorem 4 (Balasubramanian and Koblitz, [BSS05, IX.12])**.** *Let E be an elliptic curve over a finite field $\mathbb{F}_q$ and let m be a prime dividing $\#E(\mathbb{F}_q)$. Suppose that m does not divide $(q-1)$ (i.e. $k > 1$) and that $\gcd(m, q) = 1$. Then $E[m] \subset E(\mathbb{F}_{q^k})$ if and only if m divides $(q^k - 1)$.*

With this theorem, when the embedding degree is strictly greater than one, we know that the full m-torsion of E will be on $\mathbb{F}_{q^k}$. This is useful to define $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$.

We combine these two results. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, let $m \mid \#E(\mathbb{F}_q)$ and let $k > 1$ be the embedding degree of E with respect to q and m. Then $E(\mathbb{F}_{q^k})[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ and more precisely, by definition of the embedding degree, the full m-torsion is not defined over any proper subfield of $\mathbb{F}_{q^k}$, in other words, $E(\mathbb{F}_{q^i})[m] = \mathbb{Z}/m\mathbb{Z}$ for all $1 \leqslant i < k$. Finally, in most applications, we will set $\mathbb{G}_1 = E(\mathbb{F}_q)[m] \simeq \mathbb{Z}/m\mathbb{Z}$ and $\mathbb{G}_2 \subset E(\mathbb{F}_{q^k})[m]$ such that $\mathbb{G}_1 \cap \mathbb{G}_2 = \{\mathcal{O}\}$. Thanks to Balasubramanian and Koblitz theorem (Th. 4), we know that $\mathbb{G}_T \subset \mathbb{F}_{q^k}$.

Definition 14 (Reduced Tate pairing, [BSS05, IX.5])**.** *The reduced Tate pairing is the map*

$$\begin{array}{rcl} e_{\text{Tate},m} : E(\mathbb{F}_{q^k})[m] \times E(\mathbb{F}_{q^k})/mE(\mathbb{F}_{q^k}) & \to & \boldsymbol{\mu}_m \subset \mathbb{F}_{q^k}^* \\ (P, Q) & \mapsto & \langle P, Q \rangle_m^{\frac{q^k-1}{m}} = f(\mathcal{D})^{\frac{q^k-1}{m}} \end{array}$$

The practical computation of the function f will be explained in Sec. 1.4.4. The powering to $(q^k - 1)/m$ cancels all the terms which are not m-th roots of unity. Since the pairing takes its values in the multiplicative group $\mathbb{F}_{q^k}^*$, after this powering any output value will be in the subgroup $\boldsymbol{\mu}_m$ rather than in an equivalence class.

1.4.3 Pairing-friendly curves

Freeman, Scott and Teske propose this definition for pairing-friendly curves in [FST10].

Definition 15 ([FST10, Def. 2.3])**.** *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. We say that E is* pairing-friendly *if the following two conditions hold:*

1. *there is a prime $r \geqslant \sqrt{q}$ dividing $\#E(\mathbb{F}_q)$, and*
2. *the embedding degree of E with respect to r is less than $\log_2(r)/8$.*

The first condition says that the ρ-value is less than 2, where $\rho = \log q / \log r$ is used to measure how far the curve parameters are from the optimal case where the curve is of prime order ($\rho = 1$ in this case). Finding pairing-friendly elliptic curves has been a quite active field of research, especially in the last decade, until the survey paper of Freeman, Scott ant Teske [FST10]. We recall the main idea of the available constructions and the usual notations. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$.

We denote by t its trace over $\mathbb{F}_q$ and by m its order over $\mathbb{F}_q$, $\#E(\mathbb{F}_q) = q + 1 - t = m$. Moreover we consider a prime divisor r of $\#E(\mathbb{F}_q)$. We denote by k the embedding degree with respect to r and q, i.e. the smallest integer such that $E[r] \subset E(\mathbb{F}_{q^k})$. The high-level structure of the constructions in the literature follow essentially these two steps [FST10, §2, p. 9].

1. Fix k, and compute integers t, r, q such that there is an elliptic curve $E(\mathbb{F}_q)$ that has trace t, a subgroup of prime-order r, and embedding degree k.
2. Use the complex multiplication method to find the equation of the curve E over $\mathbb{F}_q$.

An ordinary elliptic curve with these properties can be constructed if an only if the following conditions hold [FST10, §2, p. 9]:

1. q is prime or a prime power. At the moment, there is not any construction for interesting (i.e. with $\rho < 2$) ordinary pairing-friendly elliptic curves over extension fields.
2. r is prime.
3. t is relatively prime to q to ensure that the curve is not supersingular. Note that t must satisfies the Hasse bound $|t| \leqslant 2\sqrt{q}$, this is induced through condition 6.
4. r divides $q + 1 - t$.
5. r divides $q^k - 1$, and $r \nmid q^i - 1$ for $1 \leqslant i < k$.
6. $4q - t^2 = Dy^2$ for some sufficiently small positive integer D and some integer y.

1.4.3.1 Supersingular curves

The first pairing-friendly elliptic curves to be proposed were supersingular. A supersingular curve over a finite field $\mathbb{F}_q$ is such that $\#E(\mathbb{F}_q) \equiv 1 \mod p$ with $p = \mathrm{char}(\mathbb{F}_q)$ or equivalently, $\#E(\mathbb{F}_q) = q + 1 - t$ with $p \mid t$. Actually, supersingular curves were used in ECC because their order is well-known, running a point-counting algorithm is not needed (this was quite costly in the 80's). These supersingular curves were attacked with the MOV and FR methods [MOV93, FR94] to embed the discrete logarithm computation from the elliptic curve subgroup $E(\mathbb{F}_p)$ into the finite field $\mathbb{F}_{p^2}$. (For curves defined over $\mathbb{F}_{2^n}$ or $\mathbb{F}_{3^n}$, the embedding degree can be higher, up to 4, resp. 6). They were proposed again in cryptography in [BF01, Jou00] with larger parameter size for use in the first pairing-based cryptography applications.

Example 5. *Let p be a large prime ($p \geqslant 5$), $p \equiv 3 \mod 4$ and $E : y^2 = x^3 + ax$ with $a \in \mathbb{F}_p$ that is not a square. This curve has j-invariant $j = 1728$ and $p + 1$ points (hence trace $t = 0$). For any $m > 2$ such that $m \nmid p - 1$, $m \mid p + 1$, we have $m | p^2 - 1 = (p + 1)(p - 1)$ hence the embedding degree is $k = 2$. We have $\#E(\mathbb{F}_{p^2}) = p^2 + 1 - t_{p^2}$ with $t_{p^2} = t^2 - 2p = -2p$ hence $\#E(\mathbb{F}_{p^2}) = p^2 + 1 + 2p = (p + 1)^2$. There exists a distortion map $(x, y) \mapsto (-x, iy)$ with $i = \sqrt{-1} \in \mathbb{F}_{p^2}$.*

Example 6. *Let p be a large prime ($p \geqslant 5$), $p \equiv 2 \mod 3$ and $E : y^2 = x^3 + b$ with $b \in \mathbb{F}_p$. This curve has $p + 1$ points (and trace $t = 0$). There exists a distortion map $(x, y) \mapsto (\zeta_3 x, y)$ with ζ_3 a primitive third root of unity, i.e. $\zeta_3^2 + \zeta_3 + 1 = 0 \in \mathbb{F}_{p^2}$.*

The two following methods, the Cocks-Pinch and the Brezing-Weng algorithms, search for parameters satisfying the constraints presented in Sec. 1.4.3. Let E be an elliptic curve and let $\#E(\mathbb{F}_p) = p + 1 - t = hr$ with r a large prime and h the related cofactor. Hence $p \equiv t - 1 \mod r$. Let $\Delta = t^2 - 4p$ with a square-free factorization into $\Delta = -Dy^2$. The second useful formula is $Dy^2 = 4p - t^2 = 4hr - (t - 2)^2$, hence $-Dy^2 \equiv (t - 2)^2 \mod r$.

1.4.3.2 Cocks-Pinch Method

We recall in Alg. 3 p. 28 the method proposed by Cocks and Pinch in 2001 to construct pairing-friendly elliptic curves [CP01] (see also [BSS05, Algorithm IX.4]). The obtained elliptic curves have ρ-value around 2. Any prime can be chosen as input value. As r divides $\Phi_k(p)$, we can rewrite it as $\Phi_k(p) \equiv 0 \mod r$. With properties of cyclotomic polynomials, we obtain $p \equiv \zeta_k \mod r$ with ζ_k a primitive k-th root of

Algorithm 3: Cocks-Pinch method to find a pairing-friendly elliptic curve.

Input: Square-free integer D, size of r and embedding degree k to match the security level in bits, knowing that $\rho \approx 2$.
Output: Prime order r, prime number p

```
repeat
    Pick at random a prime r of prescribed size until −D is a square in the finite field F_r and F_r
    contains a primitive k-th root of unity ζ_k, that is r ≡ 1 mod k.
    Lift t and y from F_r to Z and set p = (1/4)(t^2 + Dy^2).
until p is prime.
return r, p
```

unity. Furthermore, $t \equiv 1 + p \mod r$ so this method chooses $t = 1 + \zeta_k$ in $\mathbb{F}_r$. Then $y = (t-2)/\sqrt{-D}$ in $\mathbb{F}_r$. To obtain the curve parameters a and b, we need to compute a j-invariant for the curve of given trace t over $\mathbb{F}_p$. The first method is to compute the Hilbert class polynomial $\mathcal{H}_D$ associated to D, then to compute a root of this polynomial modulo p, the root will be a candidate for the j-invariant. This polynomial has very large coefficients and is not computable in reasonable time and memory for large D, e.g. $D > 10^9$. There exists some variants such as the computation of the Weber polynomial associated to D. This polynomial has smaller coefficients. A root of the Weber polynomial modulo p can give a root of the Hilbert class polynomial of D. Correspondences between roots of $\mathcal{H}_D$ and roots of Weber polynomials for various D are given in e.g. [KKSZ10]. Computing class polynomials (Hilbert, Weber) can be performed with the Miracl library [Sco11], and more recently with the work of Enge [Eng12] and Sutherland [Sut12].

1.4.3.3 Brezing-Weng and Scott-Barreto methods

The method proposed by Brezing and Weng and the other version proposed by Barreto and Scott compute the parameters in a number field $K \simeq \mathbb{Q}[x]/(r(x))$ instead of a finite prime field $\mathbb{F}_r$. The parameters will be polynomials modulo an irreducible polynomial (a cyclotomic polynomial in a first version) instead of integers modulo a prime. The choice of D is limited to few tiny values such as 1, 2, 3. Otherwise the polynomials $p(x), r(x)$ defining the primes p and r will have a too high degree. In this case there will be no choice on r and p. There is a heuristic on the form of polynomials $p(x), r(x)$ taking many prime values when iterating over x.

Definition 16 ([FST10, Def. 2.5])**.** *Let $f(x)$ be a polynomial with rational coefficients. We say f* represents primes *if the following conditions are satisfied:*

1. *$f(s)$ is non-constant;*
2. *$f(x)$ has positive leading coefficient;*
3. *$f(x)$ is irreducible;*
4. *$f(x) \in \mathbb{Z}$ for some $x \in \mathbb{Z}$ (equivalently, for an infinite number of $x \in \mathbb{Z}$);*
5. $\gcd(\{f(x) : x, f(x) \in \mathbb{Z}\}) = 1$.

We adopt the approach in [FST10]. The polynomial method uses the formula

$$Dy^2 = 4p(x) - t(x)^2 = 4h(x)r(x) - (t(x)-2)^2 \tag{1.28}$$

from $h(x)r(x) = p(x) + 1 - t(x)$ with $h(x)$ a cofactor as small as possible and $r(x), p(x)$ are prime for a given x.

We give an example for $k = 16$. The 16-th cyclotomic polynomial is $\Phi_{16}(x) = x^8 + 1$. We can start with $r(x) = x^8 + 1$. We build $\mathbb{K} = \mathbb{Q}[x]/(\Phi_{16}(x))$. We know that $\mathbb{K}$ contains $\zeta_{16} = x$, $\zeta_8 = x^2 = \frac{1+i}{\sqrt{2}}$, $\zeta_4 = i = x^4 = \sqrt{-1}$ and also $\sqrt{-2} = x^6 + x^2 \in \mathbb{K}$. So we can try with $D = 1$ or $D = 2$. We obtain $t(x) = x^e + 1$, e odd, $1 \leqslant e \leqslant 15$. Unfortunately in any case, the polynomial $p(x)$ is not irreducible.

1. $r(x) = x^8 + 1$

Algorithm 4: Polynomial method to find a pairing-friendly elliptic curve.

Input: an embedding degree k, a square-free discriminant D
Output: irreducible polynomials p and r, polynomials t, y, h such that (1.28) is satisfied

1 Construct a number field $\mathbb{K} \simeq \mathbb{Q}[x]/(r(x)) \supset \mathbb{Q}[\zeta_k]$, the number field $\mathbb{K}$ contains the primitive k-th roots of unity ζ_k. For example, simply choose $r(x) = \Phi_k(x)$ the k-th cyclotomic polynomial.
2 Choose $t(x)$ to be a polynomial corresponding to $1 + \zeta_k \in \mathbb{K}$. For example if $\mathbb{K} = \mathbb{Q}[x]/(\Phi_k(x))$ then $t(x) = 1 + x^e$ with $1 \leqslant e < k - 1$, e coprime to k.

3 **if** $\sqrt{-D} \in \mathbb{K}$ **then**
  Brezing-Weng method:
4 The equation (1.28) factors into $\left(t(x) - 2 + y\sqrt{-D}\right)\left(t(x) - 2 - y\sqrt{-D}\right) \equiv 0 \mod r(x)$
5 Set $y(x) = \pm(t(x) - 2)/\sqrt{-D} \in \mathbb{K}$

6 **else** (in that case $\sqrt{-D} \notin \mathbb{K}$)
  Scott-Barreto method:
7 Search for a suitable polynomial $h(x)$ of degree 0 or 1 such that (1.28) is satisfied.

8 Set $p(x) = \frac{1}{4}\left(t^2(x) + Dy^2(x)\right)$.
9 **if** *$p(x)$ represents primes and $r(x)$ has positive leading coefficient* **then**
10 **return** $p(x), r(x), t(x), y(x), h(x)$
11 **else**
12 Return to step 1 and choose a different $r(x)$.

2. $t(x) = x^e + 1$
3. If $D = 1$ then $\sqrt{-D} = x^4, 1/\sqrt{-D} \mod r(x) = -x^4$
   - $y(x) = \pm(t(x) - 2)/\sqrt{-D} = (x^e - 1)(-x^4) = -x^{4+e} + x^4$
   - We choose $e = 1, 5, 9, 13$ to minimize both the degrees of t and y.

e	$t(x)$	$y(x)$	$p(x)$
1	$x+1$	$-x^5+x^4$	$(x^2+1)(x^8-2x^7+2x^5-2x^3+2x+1)/4$
5	x^5+1	$x+x^4$	$(x+1)^2(x^8-2x^7+4x^6-6x^5+8x^4-6x^3+4x^2-2x+1)/4$
9	$-x+1$	x^5+x^4	$(x^2+1)(x^8+2x^7-2x^5+2x^3-2x+1)/4$
13	$-x^5+1$	$-x+x^4$	$(x-1)^2(x^8+2x^7+4x^6+6x^5+8x^4+6x^3+4x^2+2x+1)/4$

   We see here that the method fails with r a cyclotomic polynomial. We need to choose another r.
4. If $D = 2$ then $\sqrt{-D} = x^6 + x^2, 1/\sqrt{-D} \mod r(x) = \frac{-1}{2}(x^6 + x^2)$
   - $y(x) = \pm(t(x) - 2)/\sqrt{-D} = -(x^e - 1)(x^6 + x^2)/2 = (-x^{e+6} - x^{e+2} + x^6 + x^2)/2$. We will have $\rho \geqslant 12/8 = 1.5$ in any case.

Constructions from [KSS08] are obtained with a systematic search (with computer). As in [FST10] we can cite some examples.

Example 7 ([KSS08, Example 4.2])**.**

$$\begin{aligned} k &= 16 \\ D &= 1 \\ t(x) &= (2x^5 + 41x + 35)/35 \\ p(x) &= (x^{10} + 2x^9 + 5x^8 + 48x^6 + 152x^5 + 240x^4 + 625x^2 + 2398x + 3125)/980 \\ r(x) &= x^8 + 48x^4 + 625 \\ x &\equiv \pm 25 \mod 70 \end{aligned}$$

1.4.3.4 Barreto-Naehrig Construction of Pairing-Friendly Elliptic Curves

In 2005 at the SAC conference, Barreto and Naehrig [BN05] proposed a particular case of pairing-friendly curves with $D = 3$. These so-called BN curves are now very popular. The embedding degree

$k = 12$ is optimal for curves with $\rho = 1$ and security level equivalent to an AES 128. The 12-th cyclotomic polynomial is $\Phi_{12}(x) = x^4 - x^2 + 1$. We want to find two polynomials $r(x)$ and $p(x)$ irreducible, of small degree, such that $r(x)$ defines the elliptic curve order and $r(x) \mid \Phi_{12}(x)$. Barreto and Naehrig observed that

$$\begin{aligned}\Phi_{12}(6x^2) &= (36x^4 + 36x^3 + 18x^2 + 6x + 1)(36x^4 - 36x^3 + 18x^2 - 6x + 1) \text{ and} \\ \Phi_{12}(2x^2) &= (4x^4 - 4x^3 + 2x^2 - 2x + 1)(4x^4 + 4x^3 + 2x^2 + 2x + 1)\,.\end{aligned}$$

We may set $t(x) = 6x^2 + 1$ in the first decomposition and $t(x) = 2x^2 + 1$ in the second one. Letting $r(x) = (36x^4 + 36x^3 + 18x^2 + 6x + 1)$, one may write $\Phi_{12}(6x^2) = r(x)r(-x)$. Let $\#E(\mathbb{F}_p) = r(x)$. Then $p(x) = r(x) + t(x) - 1 = r(x) + 6x^2$ and moreover, $t^2 - 4p$ factors into $-3(6x^2 + 4x + 1)^2$, thus $D = 3$ and $y = 6x^2 + 4x + 1$. To sum up, the coefficients of the curve are given by

$$\begin{aligned}k &= 12 \\ t(x) &= 6x^2 + 1 \\ r(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1 \\ p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \\ Dy^2(x) &= 108x^4 + 144x^3 + 84x^2 + 24x + 3 = 3(6x^2 + 4x + 1)^2\end{aligned}$$

with x taking positive or negative values. The curve equation is of the form $E : y^2 = x^3 + b$ with $b \in \mathbb{F}_p$ and E is not supersingular contrary to the example 6 because here $p \equiv 1 \mod 3$. The same method applied to $\Phi_{12}(2x^2)$ fails because $t^2 - 4p = -(6x^2 + 4x + 3)(2x^2 - 4x + 1)$ with no square in this case.

It is quite easy to find values for x such that both p and r are prime numbers of a given size. To achieve $\log p = \log r = 256$, we need to search for good values of x in the range

$$2^{62} < x_{\min} = \texttt{0x57e2266168ce663b} \leqslant x \leqslant x_{\max} = \texttt{0x6882f5c030b0f7ef} < 2^{63}\,. \tag{1.29}$$

In practice we start at $x = \texttt{0x6000000000000001} = 2^{62} + 2^{61} + 1$ to obtain sparce values for p and r.

1.4.4 Tate pairing: Miller algorithm and improvements

As mentioned above, in 1986 Miller provided an efficient algorithm to compute the Weil pairing. His work was widely used and was finally published in the Journal of Cryptology in 2004 [Mil04]. His algorithm is mostly used to compute the Tate pairing since this pairing turns out to be more efficient in practice on various elliptic curves. The original manuscript is available online [Mil86a]. Let P, Q be two points of order m on an elliptic curve E, with coordinates in $\mathbb{F}_q$. The aim is to compute a function f such that $\mathrm{div}(f) = m(P) - m(\mathcal{O})$. Miller's algorithm uses a double-and-add method with intermediate functions f_i. Let f_i be a function whose divisor is

$$\mathrm{div}(f_i) = i(P) - ([i]P) - (i-1)(\mathcal{O})\,. \tag{1.30}$$

Then f_m is such that

$$\mathrm{div}(f_m) = m(P) - ([m]P) - (m-1)(\mathcal{O}) = m(P) - m(\mathcal{O}) = \mathrm{div}(f)$$

since $[m]P = \mathcal{O}$. The recursive formula is the following. Let f_i, f_j be two functions as in (1.30).

$$\begin{aligned}\mathrm{div}(f_{i+j}) &= (i+j)(P) - ([i+j]P) - (i+j-1)(\mathcal{O}) \\ &= i(P) - (i-1)(\mathcal{O}) + j(P) - (j-1)(\mathcal{O}) - ([i+j]P) - (\mathcal{O}) \\ &= i(P) - ([i]P) - (i-1)(\mathcal{O}) \\ &\quad + j(P) - ([j]P) - (j-1)(\mathcal{O}) \\ &\quad + ([i]P) + ([j]P) - ([i+j]P) - (\mathcal{O})\,.\end{aligned}$$

To express $\mathrm{div}(f_{i+j})$ in terms of $\mathrm{div}(f_i), \mathrm{div}(f_j)$ and few additional divisors, we observe that a line through the points $[i]P$ and $[j]P$ has divisor $\mathrm{div}(\ell_{[i]P,[j]P}) = ([i]P) + ([j]P) + (-[i+j]P) - 3(\mathcal{O})$. We implicitly compute the coefficients of this line when computing the sum of the two points (see the graphical representation of the addition law, Sec. 1.2.2 and especially Fig. 1.3a). From the computation above we have then $\mathrm{div}(f_{i+j}) = \mathrm{div}(f_i) + \mathrm{div}(f_j) + \mathrm{div}(\ell_{[i]P,[j]P}) - (([i+j]P) + (-[i+j]P) - 2\mathcal{O})$. The last term

$([i+j]P) + (-[i+j]P) - 2\mathcal{O}$ is the divisor of the vertical line through $[i+j]P$. We denote by $\ell_{i,j}$ the line through $[i]P$ and $[j]P$ and by v_{i+j} the vertical line at $[i+j]P$. More generally in the following we will denote by $\ell_{P,Q}$ the line through two points P, Q and by v_R the vertical line at a point R. Finally,

$$\mathrm{div}(f_{i+j}) = \mathrm{div}(f_i) + \mathrm{div}(f_j) + \mathrm{div}(\ell_{i,j}) - \mathrm{div}(v_{i+j}) \,. \tag{1.31}$$

Then we have

$$f_{i+j} = f_i f_j \frac{\ell_{i,j}}{v_{i+j}} \tag{1.32}$$

up to a constant term.

1.4.4.1 Miller's algorithm

We are now able to present Miller's algorithm. The two progression formulas are

$$\begin{aligned} f_{2i} &= f_{i+i} = f_i^2 \frac{\ell_{i,i}}{v_{2i}} \text{ with } \ell_{i,i} \text{ the tangent at } [i]P \text{ and } v_{2i} \text{ the vertical line at } [2i]P \\ f_{i+1} &= f_i f_1 \frac{\ell_{i,1}}{v_{i+1}} \text{ with } \ell_{i,1} \text{ the line through } P \text{ and } [i]P \text{ and } v_{i+1} \text{ the vertical line at } [i+1]P \,. \end{aligned}$$

We develop Miller's method applied for computing a Tate pairing in Alg. 5.

Algorithm 5: Miller's algorithm, reduced Tate pairing $e_{\mathrm{Tate},m}^{(q^k-1)/m}$ [BSS05]

Input: $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_q$, $P \in E(\mathbb{F}_{q^k})[m]$, $Q \in E(\mathbb{F}_{q^k})$, m
Output: $e_{\mathrm{Tate},m}(P,Q)^{(q^k-1)/m} \in \mathbb{F}_{q^k}^*$

Choose $S \in E(\mathbb{F}_{q^k})$ such that P and $Q+S$ are linearly independent
$Q' \leftarrow Q + S$
$P_j \leftarrow P$
$f \leftarrow 1$
Miller loop
for $j \leftarrow \lfloor \log_2(m) \rfloor - 1, \ldots, 0$ **do**
  $\ell \leftarrow$ tangent at P_j
  $v \leftarrow$ vertical line at $2P_j$
  $P_j \leftarrow 2P_j$
  $f \leftarrow f^2 \cdot \dfrac{\ell(Q')v(S)}{v(Q')\ell(S)}$ — step $f_{2i} \leftarrow f_i^2 \ell_{P_i,P_i} / v_{P_{2i}}$
  if $m_j = 1$ **then**
    $\ell \leftarrow$ line through P_j and P
    $v \leftarrow$ vertical line at $(P_j + P)$
    $P_j \leftarrow P_j + P$
    $f \leftarrow f \cdot \dfrac{\ell(Q')v(S)}{v(Q')\ell(S)}$ — step $f_{i+1} \leftarrow f_i f_1 \ell_{P_i,P_1} / v_{P_{i+1}}$
Final exponentiation
$f \leftarrow f^{(q^k-1)/m}$
return f

Miller's algorithm (Alg. 5) is practical. This pairing is a good candidate for a cryptographic pairing. In particular, the third condition on efficiency is met. Since 2002 there has been various improvements to this algorithm. We present the main contributions of Barreto, Kim, Lynn and Scott in [BKLS02].

First, we clarify the definition of the two subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$. We will use Th. 4 and Prop. 8. We first set $\mathbb{G}_1 = E(\mathbb{F}_q)[m]$. This means that $P \in \mathbb{G}_1$ (in the left-hand side of the pairing) is of order m and has coefficients in $\mathbb{F}_q$. Secondly we use the fact that in this setting, $E(\mathbb{F}_q)[m] \cong \mathbb{Z}/m\mathbb{Z}$, $E(\mathbb{F}_{q^k})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ and for any subfield $\mathbb{F}_{q^i}$ with $1 \leqslant i < k$ we have $E(\mathbb{F}_{q^i})[m] \cong \mathbb{Z}/m\mathbb{Z}$ by definition of the embedding degree (see Def. 13 and Prop. 9). So we find an m-torsion point $G_2 \in E(\mathbb{F}_{q^k})$ such that

$G_2 \notin E(\mathbb{F}_q)$. We set $\mathbb{G}_2 = \langle G_2 \rangle$ to be the subgroup of order m of $E(\mathbb{F}_{q^k})$ generated by G_2. In this way we know that $\mathbb{G}_1 \cap \mathbb{G}_2 = \{\mathcal{O}\}$.

With this setting for $\mathbb{G}_1$ and $\mathbb{G}_2$, for all points $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ different from $\mathcal{O}$, the two points are linearly independent. We can set $S = \mathcal{O}$ in Alg. 5 and remove all the terms $\ell(S), v(S)$. Indeed, these terms are in $\mathbb{F}_q$ with $S = \mathcal{O}$ and they are sent to 1 after the final exponentiation. We obtain the simplified algorithm presented in Alg. 6.

Algorithm 6: Miller's algorithm, reduced Tate pairing $e_{\text{Tate},m}^{(p^k-1)/m}$ [BKLS02]

Input: $E : y^2 = x^3 + ax + b$, $P \in E(\mathbb{F}_q)[m]$, $Q \in E(\mathbb{F}_{q^k})[m] \setminus E(\mathbb{F}_q)[m]$, m

Output: $e_{\text{Tate},m}(P,Q)^{(q^k-1)/m} \in \mathbb{F}_{q^k}^*$

1 **if** $P = \mathcal{O}$ *or* $Q = \mathcal{O}$ **then** Return 1 **else**
2 $\quad P_j \leftarrow P$
3 $\quad f \leftarrow 1$
$\quad$ **Miller loop**
4 $\quad$ **for** $j \leftarrow \lfloor \log_2(m) \rfloor - 1, \ldots, 0$ **do**
5 $\quad\quad \ell \leftarrow$ tangent at P_j
6 $\quad\quad v \leftarrow$ vertical line at $2P_j$
7 $\quad\quad P_j \leftarrow 2P_j$
8 $\quad\quad f \leftarrow f^2 \cdot \frac{\ell(Q)}{v(Q)}$ step $f_{2i} \leftarrow f_i^2 \ell_{P_i,P_i} / v_{P_{2i}}$
9 $\quad\quad$ **if** $m_j = 1$ **then**
10 $\quad\quad\quad \ell \leftarrow$ line through P_j and P
11 $\quad\quad\quad v \leftarrow$ vertical line at $(P_j + P)$
12 $\quad\quad\quad P_j \leftarrow P_j + P$
13 $\quad\quad\quad f \leftarrow f \cdot \frac{\ell(Q)}{v(Q)}$ step $f_{i+1} \leftarrow f_i f_1 \ell_{P_i,P_1} / v_{P_{i+1}}$
$\quad$ **Final exponentiation**
14 $\quad f \leftarrow f^{(p^k-1)/m}$
15 $\quad$ **return** f

1.4.4.2 Example: Tate pairing on a supersingular curve

We state in Alg. 7 a Tate pairing computation. The intermediate values g and h are computed in Alg. 8 and Alg. 9, with the normal-font numbers in $\mathbb{F}_q$ and the bold ones ($\mathbf{X}$) in $\mathbb{F}_{q^2}$. Algorithm 7 uses an optimization presented first in [BKLS02]. A degree-2 twisted elliptic curve is used to remove the denominators, namely the vertical lines $v_{2T}(Q)$ and $v_{T+P}(Q)$. This trick is explained in Sec. 1.4.4.3. Moreover on a supersingular curve of the form $y^2 = x^3 + ax$, we can always set $a = 1$ in order to save a multiplication in the tangent computation, line 10 of Alg. 8. Moreover if $p \equiv 1 \mod 3$ then -3 is a square in $\mathbb{F}_p$ and we can set $a = -3$ in order to compute $t_5 = 3(X_T^2 - t_4^2) = 3(X_T + t_4) \cdot (X_T - t_4)$ in one multiplication, we save one more square.

1.4.4.3 Twists of curves

Twisted elliptic curves were introduced in [BKLS02] to speed-up pairing computations. A general overview of use in pairings is explained in [HSV06, §4]. We recall here these properties.

Definition 17 (Twist of elliptic curve [HSV06, Def. 1])**.** *Let E and E' be two elliptic curves defined over a finite field $\mathbb{F}_q$, then E' is called a* twist of degree *d of E if there exists an isomorphism $\phi_d : E' \to E$ defined over $\mathbb{F}_{q^d}$ and d is minimal.*

We note that the two elliptic curves are defined over a finite field $\mathbb{F}_q$ and the isomorphism is defined over an extension of degree d of $\mathbb{F}_q$. In other words the expression of the isomorphism contains coefficients in $\mathbb{F}_{q^d}$. We now give a useful classification.

Algorithm 7: Tate pairing $e_{\text{Tate},m}(P,\phi(Q))^{\frac{p^2-1}{m}}$ on a supersingular curve of embedding degree 2

Input: $E : y^2 = x^3 + ax$ defined over $\mathbb{F}_p$, $P = (x_P, y_P), Q = (x_Q, y_Q) \in E(\mathbb{F}_p)[m]$, m
Output: $e_{\text{Tate},m}(P,\phi(Q)) \in \boldsymbol{\mu}_m \subset \mathbb{F}^*_{p^2}$

$R = (X_R : Y_R : Z_R) \leftarrow (x_P : y_P : 1)$
$f \leftarrow 1$
for $i \leftarrow \lfloor \log_2(m) \rfloor - 1, \ldots, 0$ **do**
  $(R, \ell) \leftarrow g(R, Q)$ (see Alg 8 for computing g) — $8M_p + 6S_p$
  $f \leftarrow f^2 \cdot \ell$ — $S_{p^2} + M_{p^2} = 5M_p$
  if $m_i = 1$ **then**
    $(R, \ell) \leftarrow h(R, P, Q)$ (see Alg 9 for computing h) — $11M_p + 3S_p$
    $f \leftarrow f \cdot \ell$ — $M_{p^2} = 3M_p$

Miller loop: $\log_2 m \cdot (13M_p + 6S_p) + \mathrm{HW}(m) \cdot (14M_p + 3S_p)$
$f \leftarrow f^{p-1}$ — $2M_p + I_p$
$f \leftarrow f^{(p+1)/m} = f^h$ — $\log_2 h\, S_{p^2} + \mathrm{HW}(h) M_{p^2}$
return f — **Final exp.:** $\log_2 h\, S_{p^2} + \mathrm{HW}(h) M_{p^2} + 2M_p + I_p$

Algorithm 8: Function $g(T, Q)$ [CSB04]

Input: E, $T = (X_T : Y_T : Z_T)$, $Q = (x_Q, y_Q) \in E(\mathbb{F}_p)$
Output: $2T \in E(\mathbb{F}_p)$, $\ell_{T,T}(\phi(Q)) \in \mathbb{F}^*_{p^2}$ with $\phi(x_Q, y_Q) = (-x_Q, y_Q X)$

$t_1 \leftarrow 2Y_T^2$ — S_p
$t_2 \leftarrow 2X_T\, t_1$ — M_p
$t_3 \leftarrow 2t_1^2$ — S_p
$t_4 \leftarrow Z_T^2$ — S_p
if $a = -3$ **then** — when $p \equiv 1 \mod 3$
  $t_5 \leftarrow 3(X_T + t_4)(X_T - t_4)$ — M_p
else if $a = 1$ **then** — in any case with a supersingular curve with $j = 1728$
  $t_5 \leftarrow 3X_T^2 + t_4^2$ — $2S_p$
else — otherwise
  $t_5 \leftarrow 3X_T^2 + at_4^2$ — $2S_p$+M_p
$X_{2T} \leftarrow t_5^2 - 2t_2$ — S_p
$Y_{2T} \leftarrow t_5(t_2 - X_{2T}) - t_3$ — M_p
$Z_{2T} \leftarrow 2Y_T\, Z_T$ — M_p
$\ell \leftarrow \left[t_5(X_T + t_4\, x_Q) - t_1\right] + \left[Z_{2T}\, t_4\, y_Q\right] X$ — $4M_p$
return $((X_{2T} : Y_{2T} : Z_{2T}),\ \ell)$ — $6S_p$+$8M_p$

Proposition 10 ([HSV06, Prop. 1]). *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ with $q = p^n$. Assume that $p \geqslant 5$, then the set of twists of E is canonically isomorphic with $\mathbb{F}_q^*/(\mathbb{F}_q^*)^d$ with $d = 2$ if $j(E) \neq 0, 1728$, $d = 4$ if $j(E) = 1728$ and $d = 6$ if $j(E) = 0$.*

We give an example with $d = 2$. Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let E' be its quadratic twist, $d = 2$. The twist is given by the equation $E' : \alpha y^2 = x^3 + ax + b$, with $\alpha \in \mathbb{F}_q$ a non-square. We can schematize their groups of points over $\mathbb{F}_q$ and $\mathbb{F}_{q^2}$ in this way.

$$\begin{array}{ccccc} & & E(\mathbb{F}_{q^2}) & \overset{\text{isomorphism}}{\simeq} & E'(\mathbb{F}_{q^2}) \\ & & \cup & & \cup \\ \#E(\mathbb{F}_q) = q + 1 - t_q & & E(\mathbb{F}_q) & & E'(\mathbb{F}_q) \quad \#E'(\mathbb{F}_q) = q + 1 + t_q \end{array}$$

The following map ϕ_2 sends a point in $E'(\mathbb{F}_q)$ to a point in $E(\mathbb{F}_{q^2})$, with $\sqrt{\alpha} \in \mathbb{F}_{q^2}$.

$$\begin{array}{rcl} \phi_2 : E' & \to & E \\ (x', y') & \mapsto & (x', y'\sqrt{\alpha})\,. \end{array}$$

Algorithm 9: function $h(P, T, Q)$ [CSB04]

Input: E, $P = (x_P, y_P)$, $T = (X_T : Y_T : Z_T)$, $Q = (x_Q, y_Q) \in E(\mathbb{F}_p)$
Output: $T + P \in E(\mathbb{F}_p)$, $\ell_{T,P}(\phi(Q)) \in \mathbb{F}_{p^2}^*$

1	$t_1 \leftarrow Z_T^2$	S_p
2	$t_2 \leftarrow Z_T\, t_1$	M_p
3	$t_3 \leftarrow x_P\, t_1$	M_p
4	$t_4 \leftarrow y_P\, t_2$	M_p
5	$t_5 \leftarrow t_3 - X_T$	
6	$t_6 \leftarrow t_4 - Y_T$	
7	$t_7 \leftarrow t_5^2$	S_p
8	$t_8 \leftarrow t_5\, t_7$	M_p
9	$t_9 \leftarrow X_T\, t_7$	M_p
10	$X_{T+P} \leftarrow t_6^2 - (t_8 + 2t_9)$	S_p
11	$Y_{T+P} \leftarrow t_6(t_9 - X_{T+P}) - Y_T\, t_8$	$2M_p$
12	$Z_{T+P} \leftarrow Z_T\, t_5$	M_p
13	$\ell \leftarrow \left[-Z_{T+P}\, y_P + t_6(x_Q + x_P)\right] + \left[Z_{T+P}\, y_Q\right]X$	$3M_p$
14	**return** $((X_{T+P} : Y_{T+P} : Z_{T+P}),\ \ell)$	$3S_p$+$11M_p$

Note that the orders satisfy $\#E(\mathbb{F}_q) = q + 1 - t_q$ and $\#E'(\mathbb{F}_q) = q + 1 + t_q$, the traces are opposite. Since the isomorphism ϕ_2 contains a coefficient $\sqrt{\alpha}$ in $\mathbb{F}_{q^2}$ the two groups $\#E(\mathbb{F}_{q^2})$ and $\#E'(\mathbb{F}_{q^2})$ have the same order, which is $(q + 1 - t_q)(q + 1 + t_q)$. The idea behind is to compress the representation of the points in $E(\mathbb{F}_{q^2})$. We manipulate points of the form (x, y) with $x, y \in \mathbb{F}_q$, these points belong to $E(\mathbb{F}_q)$, and secondly we have points of the form $(x, \sqrt{\alpha}y)$ with $x, y \in \mathbb{F}_q$. The group $E(\mathbb{F}_{q^2})$ is isomorphic to the sum $E(\mathbb{F}_q) \oplus \phi_2(E'(\mathbb{F}_q))$.

For a pairing-friendly curve, we consider the twist from on top of the elliptic curve, over $\mathbb{F}_{q^k}$ with k even.

$$\begin{array}{ccc} E(\mathbb{F}_{q^k}) & \overset{\text{isomorphism}}{\simeq} & E'(\mathbb{F}_{q^k}) \\ \cup & & \cup \\ E(\mathbb{F}_{q^{k/2}}) & & E'(\mathbb{F}_{q^{k/2}}) \\ \cup & & \\ E(\mathbb{F}_q) & & \end{array}$$

A twist is used to obtain a compressed form of the second point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$. We recall from Sec. 1.4.2 (and a consequence of Prop. 8 and Th. 4) that $E(\mathbb{F}_{q^i})[m]$ has the structure of $\mathbb{Z}/m\mathbb{Z}$ for all $1 \leqslant i < k$ with k the embedding degree of E with respect to q and m.

We can decompose $E(\mathbb{F}_{q^k})$ in two subgroups and write

$$\begin{aligned} E(\mathbb{F}_{q^k}) &\simeq E(\mathbb{F}_{q^{k/2}}) \oplus \phi_2(E'(\mathbb{F}_{q^{k/2}})) \\ \#E(\mathbb{F}_{q^k}) &= \#E(\mathbb{F}_{q^{k/2}}) \cdot \#E'(\mathbb{F}_{q^{k/2}}) \\ &= (q^{k/2} + 1 - t_{q^{k/2}})(q^{k/2} + 1 + t_{q^{k/2}})\,. \end{aligned}$$

This means that any point Q in the subgroup of $E(\mathbb{F}_{q^k})$ of order $(q^{k/2} + 1 + t_{q^{k/2}})$ corresponds to a point Q' of same order on $E'(\mathbb{F}_{q^{k/2}})$ via the map $Q' = (x', y') \mapsto (x', y'\sqrt{\alpha})$. Moreover since we know that $\mathbb{G}_2 \not\subset E(\mathbb{F}_{q^{k/2}})$, we obtain that

$$\mathbb{G}_2 \subset \phi_2(E'(\mathbb{F}_{q^{k/2}}))\,.$$

More precisely, we know that

$$\begin{aligned} r^2 \mid \#E(\mathbb{F}_{q^k}) &= (q^{k/2} + 1 - t_{q^{k/2}})(q^{k/2} + 1 + t_{q^{k/2}}) \\ r \mid \#E(\mathbb{F}_{q^{k/2}}) &= (q^{k/2} + 1 - t_{q^{k/2}}) \\ r \mid \#E(\mathbb{F}_q) &= (q + 1 - t_q) \end{aligned}$$

By definition of k (and with some restrictions on q, see [BCF09]), we deduce that $r^2 \nmid \#E(\mathbb{F}_{q^{k/2}})$ and we conclude that $r \mid (q^{k/2} + 1 + t_{q^{k/2}}) = \#E^{'}(\mathbb{F}_{q^{k/2}})$.

A point $Q \in \mathbb{G}_2$ of order m can be compressed in the form $\phi_2(Q^{'})$ with $Q^{'}$ a point of order m in the quadratic twist $E^{'}$ defined over $\mathbb{F}_{q^{k/2}}$. The point Q has the form $Q = (x_0, y_1\sqrt{\alpha})$ with $Q^{'}(x_0, y_1)$ a point on $E^{'}(\mathbb{F}_{q^{k/2}})$.

In [BKLS02], the authors remark that in a pairing computation, the vertical lines evaluated at $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$ have the form $v_T(Q) = x_T - x_Q$ with $x_T \in \mathbb{F}_q$ and $x_Q \in \mathbb{F}_{q^{k/2}}$ with the above compression. Hence $v_T(Q) \in \mathbb{F}_{q^{k/2}}$ simplifies after the final exponentiation: $v_T(Q)^{\frac{q^k-1}{m}} = v_T(Q)^{(q^{k/2}-1)\frac{q^{k/2}+1}{m}}$. By definition of the embedding degree, k is the smallest integer such that $m \mid q^k - 1$. Thus $m \nmid q^{k/2} - 1$ and $m \mid q^{k/2} + 1$. We can write $v_T(Q)^{\frac{q^k-1}{m}} = \left(v_T(Q)^{q^{k/2}-1}\right)^{\frac{q^{k/2}+1}{m}} = 1$ since $v_T(Q) \in \mathbb{F}_{q^{k/2}}$. This is an elegant and very efficient simplification. This can be generalized to higher degree twists.

The general idea is to compress a point in $\mathbb{G}_2 \subset E(F_{q^k})$ into a simpler form, then see that some computations simplify after the final exponentiation. We give in the following table (Tab. 1.2 the different forms of a twist, with respect to its degree d from [HSV06, §4], then we compress the second point Q thanks to this degree d twist.

Table 1.2: Twists of elliptic curves of degree 2, 3, 4,and 6 in large characteristic

d	E defined over $\mathbb{F}_q$	$\mathbb{F}_{q^k}$	twist $E^{'}$ defined over $\mathbb{F}_q$	$\phi_d(x,y)$
2	$y^2 = x^3 + ax + b$	$\mathbb{F}_{q^{k/2}}[Z]/(Z^2 - \alpha)$	$y^2 = x^3 + \frac{a}{\alpha^2}x + \frac{b}{\alpha^3}$	$(xZ, \alpha yZ)$
2	$y^2 = x^3 + ax + b$	$\mathbb{F}_{q^{k/2}}[Z]/(Z^2 - \alpha)$	$\alpha y^2 = x^3 + ax + b$	(x, yZ)
4	$y^2 = x^3 + ax$	$\mathbb{F}_{q^{k/4}}[Z]/(Z^4 - \alpha)$	$y^2 = x^3 + \frac{a}{\alpha}x$	(xZ^2, yZ^3)
3	$y^2 = x^3 + b$	$\mathbb{F}_{q^{k/3}}[Z]/(Z^3 - \alpha^2)$	$y^2 = x^3 + \frac{b}{\alpha^2}$	$(xZ, y\alpha)$
6	$y^2 = x^3 + b$	$\mathbb{F}_{q^{k/6}}[Z]/(Z^6 - \alpha)$	$y^2 = x^3 + \frac{b}{\alpha}$	(xZ^2, yZ^3)

Now, there is a refinement for degree 3, 4 and 6 twists. Any two degree-2 twists $E^{'}$ and $E^{''}$ defined over $\mathbb{F}_q$ of a same elliptic curve E also defined over $\mathbb{F}_q$ are isomorphic over $\mathbb{F}_q$ and isomorphic to E over $\mathbb{F}_{q^2}$. Indeed, we have

$$\begin{aligned} E : y^2 &= x^3 + ax + b \\ E^{'} : \alpha y^{'2} &= x^{'3} + ax^{'} + b \\ E^{''} : \beta y^{''2} &= x^{''3} + ax^{''} + b \end{aligned}$$

then we have this isomorphism from $E^{'}$ into $E^{''}$:

$$\begin{aligned} E^{'} &\to E^{''} \\ (x^{'}, y^{'}) &\mapsto (x^{'}, y^{'}\sqrt{\alpha/\beta}) \text{ with } \sqrt{\alpha/\beta} \in \mathbb{F}_q \end{aligned}$$

Since both α and β are non-square in $\mathbb{F}_q$, the quantity α/β is a square, then $\sqrt{\alpha/\beta} \in \mathbb{F}_q$ and these two curves are isomorphic over $\mathbb{F}_q$. There is only one choice, up to isomorphism over $\mathbb{F}_q$, for a quadratic twist of a given curve E defined over $\mathbb{F}_q$. This is not the same for degree 3, 4 and 6 twists because we have two different choices for the element α defining the twist. We list here the different cases and the corresponding twist orders.

For a number theoretical explanation, this comes from the different choices we have for primitive d-th roots of unity. For degree 4 twists, there is ζ_4, $-\zeta_4$ and for degree 3 and 6 twists, there is ζ_3, ζ_3^2 and ζ_6, ζ_6^5.

The next step is to compress the representation of $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$ thanks to this degree-d twist. If $d \mid k$ and a degree-d twist of $E(\mathbb{F}_{q^{k/d}})$ is available then the points in $E(\mathbb{F}_{q^k})$ have a factor-d compression. First, we have to choose the right twist. To do that, we compute the trace t_q of $E(\mathbb{F}_q)$ then compute the two orders of the two twists and choose the twist whose order is a multiple of m, with m related to the pairing

Table 1.3: Degree 3, 4 and 6 twist of elliptic curves.

Twist degree	curve eq.	curve order
4	$E : y^2 = x^3 + ax$	$q+1-t_q$, with t_q even, $t_q^2 - 4q = -4y^2$
$q \equiv 1 \mod 4$	$E' : y'^2 = x'^3 + a/\alpha x'$	$q+1-2y$
α is not a square	$E'' : y''^2 = x''^3 - a/\alpha x''$	$q+1+2y$
3	$E : y^2 = x^3 + b$	$q+1-t_q$, with $t_q^2 - 4q = -3y^2$
$q \equiv 1 \mod 3$	$E' : y'^2 = x'^3 + b/\alpha^2$	$q+1-(3y-t_q)/2$
α is not a cube	$E'' : y''^2 = x''^3 + b/\alpha^4$	$q+1-(-3y-t_q)/2$
6	$E : y^2 = x^3 + b$	$q+1-t_q$, with $t_q^2 - 4q = -3y^2$
$q \equiv 1 \mod 6$	$E' : y'^2 = x'^3 + b/\alpha$	$q+1-(-3y+t_q)/2$
α is neither a square nor a cube	$E'' : y''^2 = x''^3 + b/\alpha^5$	$q+1-(3y+t_q)/2$

($e_{\text{Tate},m}, e_{\text{Weil},m}$. Then, Q is compressed in the form $\phi_d(Q')$ with Q' a point on the right twist, defined over $\mathbb{F}_{q^{k/d}}$. We obtain a factor d compression. If this compression gives a point Q whose x-coordinate is in a proper subgroup of $\mathbb{F}_{q^k}$ then the vertical lines in the pairing computation are also in this proper subfield and we can remove them from the pairing computation, since they are neutralized through the final exponentiation. This simplification is compatible with degree 2, 4 and 6 twists but not with degree 3 twists. We now give an example with degree 6 twists.

Example 8 (Factor-6 compression of $\mathbb{G}_2$ with a degree-6 twist and $6 \mid k$)**.** *Let $E : y^2 = x^3 + b$ be a pairing-friendly elliptic curve defined over $\mathbb{F}_q$, of embedding degree k such that $6 \mid k$. Let $E' : y^2 = x^3 + b/\beta$ the right degree 6 twist defined over $\mathbb{F}_q$, with $\beta \in \mathbb{F}_q$ neither a square nor a cube and let $\mathbb{F}_{q^k}$ defined by $\mathbb{F}_{q^{k/6}}[Z]/(Z^6 - \beta)$. We can have a factor 6 compression for $\mathbb{G}_2$ on this curve E.*

$$
\begin{array}{lcc}
 & E(\mathbb{F}_{q^k}) \ni Q = (x'Z^2, y'Z^3) & E'(\mathbb{F}_{q^k}) \\
 & \cup & \cup \\
\#E(\mathbb{F}_{q^{k/6}}) = q^{k/6} + 1 - t_{q^{k/6}} & E(\mathbb{F}_{q^{k/6}}) & Q' = (x', y') \in E'(\mathbb{F}_{q^{k/6}}) \\
 & \cup & \#E'(\mathbb{F}_{q^{k/6}}) = q + 1 - (\pm 3y + t_{q^{k/6}})/2 \\
\#E(\mathbb{F}_q) = q + 1 - t_q & E(\mathbb{F}_q) &
\end{array}
\qquad \phi_6
$$

A point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$ is compressed in the form $Q = (x'Z^2, y'Z^3)$ with $(x', y') \in E'(\mathbb{F}_{q^{k/6}})[m]$. Hence a vertical line has the form $v_T(Q) = x_T - x'Z^2 \in \mathbb{F}_{q^{k/3}}$. This vertical line is in the subgroup $\mathbb{F}^*_{q^{k/3}}$ of $\mathbb{F}^*_{q^k}$ hence $v^{\frac{p^k-1}{m}} = v^{\frac{(p^{k/3}-1)(1+p^{k/3}+p^{2k/3})}{m}}$ and $v^{p^{k/3}-1} = 1$ so we can remove v from the computations since its contribution is neutralized by the final exponentiation.

To conclude this paragraph, using a degree-d twist to compress the second point Q is useful to remove the vertical line computations in the algorithm when d is even and in general, we can optimize the line and tangent computations thank to the compression of Q.

1.4.4.4 Implementation of a Tate pairing on a BN curve

The implementation uses Alg. 6 without the verticals. We will explicit the line and tangent computations with a degree 6 twist. We re-use the functions g and h explained in Alg. 8 and Alg. 9 and adopt the same notations. This time the degree 6 twist is $\phi_6 : (x'_Q, y'_Q) \mapsto (x'_Q u^2, y'_Q u^3) \in E(\mathbb{F}_{p^{12}})$ for a D-twist, i.e. $Q \in E'(\mathbb{F}_{p^2}) : y'^2 = x'^3 + b/\beta$ with β a non-square and non-cube in $\mathbb{F}_{p^2}$. The element β is also used to define the extension field $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}[u]/(u^6 - \beta)$. We obtain the following for tangent and line computations, with the black numbers in $\mathbb{F}_p$, the light gray bold ones ($\mathbf{X}$) in $\mathbb{F}_{p^2}$ and the gray bold ones ($\mathbf{u}$) in $\mathbb{F}_{p^{12}}$.

$$
\begin{array}{rcl}
\ell_{T,T}(x'_Q, y'_Q) & = & 2Y_T\, Z_T^3\, y'_Q - 2Y_T^2 - (3X_T^2 + aZ_T^4)(Z_T^2\, x'_Q - X_T) \\
\ell_{T,T}(x'_Q u^2, y'_Q u^3) & = & Z_{2T}\, t_4 y'_Q u^3 - t_1 - t_5(t_4 x'_Q u^2 - X_T) \\
 & = & t_5\, X_T - t_1 - t_5\, t_4\, x'_Q u^2 + Z_{2T}\, t_4\, y'_Q u^3
\end{array}
\tag{1.33}
$$

$$\begin{aligned} \ell_{T,P}(x'_Q, y'_Q) &= Z_{T+P}(y'_Q - Y_P) - (Y_P\, Z_T^3 - Y_T)(x'_Q - X_P) \\ \ell_{T,P}(x'_Q u^2, y'_Q u^3) &= Z_{T+P}(y'_Q u^3 - Y_P) - t_6(x'_Q u^2 - X_P) \\ &= t_6 X_P - Z_{T+P} Y_P - t_6 x'_Q u^2 + Z_{T+P} y'_Q u^3 \end{aligned} \tag{1.34}$$

In both cases the line an tangent are sparse elements of $\mathbb{F}_{p^{12}}$ of the form $\ell = \ell_{00} + \ell_2 U^2 + \ell_3 U^3$ with $\ell_{00} \in \mathbb{F}_p$ and $\ell_2, \ell_3 \in \mathbb{F}_{p^2}$. A multiplication in $\mathbb{F}_{p^{12}}$ costs $18M_{p^2} \sim 54M_p$ in our implementation. A dedicated line-multiplication for the steps $f \leftarrow f^2 \cdot \ell_{T,T}(\phi_6(Q))$ (Alg. 6 line 8 and Alg. 7 line 5) and $f \leftarrow f \cdot \ell_{T,P}(\phi_6(Q))$ (Alg. 6 line 8 and Alg. 7 line 8) permits to save up to $5M_{p^2}$. In this first version of Tate pairing, we implemented a quite naive line multiplication in $12M_p + 3 \times 3M_{p^2} \sim 39M_p$. A more optimized version is presented in Sec. 3.2.2, see Alg. 15 and 14 with a line multiplication in $10M_{p^2} + 6M_p \sim 36M_p$, saving $3M_p$ more.

The final exponentiation is decomposed in two steps: $\frac{p^k-1}{m} = \frac{p^k-1}{\Phi_k(p)} \frac{\Phi_k(p)}{m}$ with Φ_k the k-th cyclotomic polynomial. The first part can be computed with one inversion and some Frobenius maps. The second part is an exponentiation in $\mathbb{F}_{p^{12}}$ but this time, with a smaller exponent (compared to the size of $(p^k - 1)/m$). In our context,

$$\frac{p^{12}-1}{m} = (p^6-1)\frac{p^6+1}{\Phi_{12}(p)}\frac{\Phi_{12}(p)}{m} = (p^6-1)(p^2+1)\frac{p^4-p^2+1}{m}\,.$$

In practice we compute $f^{p^6-1} = f^{p^6} \cdot f^{-1}$ with f^{p^6} almost free (it costs only 6 subtractions in $\mathbb{F}_p$) and f^{-1} as optimized as possible with a recursive norm computation and one final inversion in $\mathbb{F}_p$. The computation of f^{p^2+1} costs one Frobenius map f^{p^2} in $5M_{p^2}$ and one $M_{p^{12}}$. The last part is an exponentiation in $\mathbb{F}_{p^{12}}$ with an exponent of roughly $3\log p$ bits. This exponentiation can be optimized very-well with the formulas in [GS10, DSD07]. The details are presented in Sec. 3.2.3 and Alg. 17.

1.4.4.5 The ate pairing

After the introduction of Tate pairing and the improvements for supersingular curves (eta pairings or η), Hess, Smart and Vercauteren presented in the paper [HSV06] a similar optimization for ordinary curves. They named their algorithm the *ate pairing*.

Definition 18. *Let E be an ordinary pairing-friendly elliptic curve dedined over $\mathbb{F}_q$, of embedding degree $k > 1$ with respect to q and $m \mid \#E(\mathbb{F}_q)$. Let π_q be the q-power Frobenius, $\pi_q : (x, y) \mapsto (x^q, y^q)$. Define the two groups*
$\mathbb{G}_1 = E[m] \cap \ker(\pi_q - \mathrm{Id})$,
$\mathbb{G}_2 = E[m] \cap \ker(\pi_q - [q])$.
The ate pairing is defined as

$$\begin{aligned} e_{\mathrm{ate},m} : \mathbb{G}_2 \times \mathbb{G}_1 &\to \mathbb{G}_T \\ (Q, P) &\mapsto f_{t-1,Q}(P)^{\frac{q^k-1}{m}}\,. \end{aligned}$$

The two differences with the Tate pairing are firstly the swap of the two input groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and secondly the loop is over $t-1$ instead of m (hence of length divided by two). These two pairings are related through this formula (1.35) we will prove in the following.

Theorem 5 (variant of [HSV06, Th. 1])**.** *Let $E(\mathbb{F}_q)$ be an ordinary pairing-friendly elliptic curve of embedding degree $k > 1$ with respect to q and $m \mid \#E(\mathbb{F}_q)$. Let π_q be the q-power Frobenius, $\pi_q : (x, y) \mapsto (x^q, y^q)$. Let $\mathbb{G}_1 = E[m] \cap \ker(\pi_q - \mathrm{Id})$, $\mathbb{G}_2 = E[m] \cap \ker(\pi_q - [q])$ and let $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$.*

$$e_{\mathrm{ReducedTate},m}(Q, P)^{\frac{(t-1)^k-1}{m}q} = e_{\mathrm{ate},m}(Q, P)^k\,. \tag{1.35}$$

For the Tate pairing, it is more efficient to compute the Miller function $f_{m,P}(Q)$, whose divisor is $m(P) - (m)\mathcal{O}$, evaluated at Q instead of $f_{m,Q}(P)$. Here, Hess, Smart and Vercauteren proposed to compute a Miller function whose divisor depends on the point $Q \in \mathbb{G}_2$, evaluated at $P \in \mathbb{G}_1$, but of reduced length. We explain in the following where this idea comes from.

Hess, Smart and Vercauteren remarked that $t - 1 \equiv \zeta_k \mod m$ since $m \mid \Phi_k(t-1)$ by construction. Secondly, they used the following property.

Proposition 11 ([HSV06, §2] from [GHS02, §6 p.330])**.** *Let N be an integer such that $m \mid N \mid q^k - 1$. Then*

$$e_{\text{ReducedTate},m}(P,Q) = f_{m,P}(Q)^{\frac{q^k-1}{m}} = f_{N,P}(Q)^{\frac{q^k-1}{N}} . \tag{1.36}$$

Proof. We prove that for an m-torsion point P and any non-zero integer n coprime to m, $f_{m,P}^{\frac{q^k-1}{m}} = f_{mn,P}^{\frac{q^k-1}{m\cdot n}}$. We start with

$$\begin{array}{rcl} \operatorname{div}(f_{m,P}) & = & m(P) - m(\mathcal{O}) \\ \operatorname{div}(f_{m\cdot n,P}) & = & mn(P) - mn(\mathcal{O}) \\ & = & n\,(m(P) - m(\mathcal{O})) \end{array} \tag{1.37}$$

In terms of functions, we get $f_{m\cdot n,P} = f_{m,P}^n$. Then we write $N = m \cdot n$ since $N \mid m$. Then $f_{N,P} = f_{m,P}^n$. The reduced Tate pairing is $e_{ReducedTate}(P,Q) = f_{m,P}(Q)^{\frac{q^k-1}{m}}$. We then write

$$\begin{array}{rcl} f_{N,P}(Q)^{\frac{q^k-1}{N}} & = & f_{m\cdot n,P}(Q)^{\frac{q^k-1}{m\cdot n}} \\ & = & \left(f_{m,P}^n(Q)\right)^{\frac{q^k-1}{m\cdot n}} \\ & = & (f_{m,P}(Q))^{n\frac{q^k-1}{m\cdot n}} \\ & = & (f_{m,P}(Q))^{\frac{q^k-1}{m}} \\ & = & e_{\text{ReducedTate},m}(P,Q) . \end{array}$$

□

We can replace the integer m by any N such that $m \mid N \mid q^k - 1$. This enlarges the Miller function computation and reduces the final exponentiation. The next step is to choose an appropriate N which can be decomposed efficiently. Hess, Smart and Vercauteren choose in a first step

$$N = \gcd((t-1)^k - 1, q^k - 1)$$

by definition, $N \mid q^k - 1$. Moreover, $m \mid \Phi_k(t-1)$ since $t - 1 \equiv q \mod m$; $\Phi_k(t-1) \mid (t-1)^k - 1$ and $m \mid q^k - 1$ hence $m \mid N$. We have

$$e_{\text{ReducedTate}}(P,Q) = f_{m,P}(Q)^{\frac{q^k-1}{m}} = f_{N,P}(Q)^{\frac{q^k-1}{N}} .$$

The second step is to write, with L such that $L \times N = (t-1)^k - 1$:

$$\begin{array}{rcl} e_{\text{ReducedTate}}(P,Q)^L & = & f_{N,P}(Q)^{\frac{q^k-1}{N}L} \\ & = & f_{N,P}^L(Q)^{\frac{q^k-1}{N}} \\ & = & f_{NL,P}(Q)^{\frac{q^k-1}{N}} \\ & = & f_{(t-1)^k-1,P}(Q)^{\frac{q^k-1}{N}} . \end{array}$$

This is also true if we swap the two points P and Q:

$$e_{\text{ReducedTate}}(Q,P)^L = f_{(t-1)^k-1,Q}(P)^{\frac{q^k-1}{N}} .$$

The next step is to decompose $(t-1)^k - 1$. Now, the main idea is to remark that over $\mathbb{G}_1 \subset E(\mathbb{F}_q)$, computing $[t-1]P$ for a given $P \in \mathbb{G}_1$ costs a scalar multiplication of length $\log(t-1) \approx \log q/2$. On the other hand, over $\mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$ this computation is almost free since $t-1$ is the eigenvalue of an endomorphism. Let $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$. Then $\pi_q(Q) = (x^q, y^q) = [\zeta_k]Q = [t-1]Q$. We computed at a cost of two Frobenius in $\mathbb{F}_{q^k}$ the point $[t-1]Q$. This can be explained in two ways. First $\mathbb{G}_2$ is constructed as $\mathbb{G}_2 = E[m] \cap \ker(\pi_q - [q])$. Hence for all $Q \in \mathbb{G}_2$, $\pi_q(Q) = [q]Q$. Moreover, since $q \equiv t-1 \mod m$ and Q is an m-torsion point, we conclude that $\pi_q(Q) = [q]Q = [t-1]Q$. The second explanation is the following. Let Q be an m-torsion point in $E(\mathbb{F}_{q^k})[m]$ such that $Q \notin E(\mathbb{F}_q)$. By definition of the embedding degree, we know that Q is not in any subgroup defined over a proper subfield of $\mathbb{F}_{q^k}$. Since E is actually

defined over $\mathbb{F}_q$, the q-power Frobenius acts over $E(\mathbb{F}_{q^k})$ as $[\zeta_k]$ with ζ_k a k-th primitive root of unity. Hence $\pi_q(Q) = [\zeta_k]Q$ (and $\pi_{q^k}(Q) = \pi_q^k(Q) = Q$). Moreover, Q is an m-torsion point and $m \mid \Phi_k(t-1)$ which means that $t - 1 \equiv \zeta_k \mod m$. The eigenvalue of $[\zeta_k]$ in the subgroup of order m is then $t-1$ and to conclude, $\pi_q(Q) = [\zeta_k]Q = [t-1]Q$.

To simplify the computation of $f_{(t-1)^k,Q}(P)$, we need this lemma.

Lemma 1. *The Miller function $f_{s,Q} : \mathrm{div}(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)(\mathcal{O})$ satisfies the property*

$$f_{s^2,Q} = f_{s,Q}^s \cdot f_{s,[s]Q} \cdot \tag{1.38}$$

More generally,

$$f_{s\cdot t,Q} = f_{s,Q}^t \cdot f_{t,[s]Q} \cdot \tag{1.39}$$

We prove this lemma with explicit divisor computations:

$$\begin{aligned}
\mathrm{div}(f_{s,Q}) &= s(Q) - ([s]Q) - (s-1)(\mathcal{O}) \\
\mathrm{div}(f_{s,Q}^s) &= s\mathrm{div}(f_{s,Q}) \\
&= s^2(Q) - s([s]Q) - (s^2 - s)(\mathcal{O}) \\
&= s^2(Q) - ([s^2]Q) - (s^2-1)(\mathcal{O}) \\
&\quad - \big(s([s]Q) - ([s^2]Q) - (s-1)(\mathcal{O})\big) \\
&= \mathrm{div}(f_{s^2,Q}) - \mathrm{div}(f_{s,[s]Q})
\end{aligned}$$

$$\begin{aligned}
\mathrm{div}(f_{s,Q}^t) &= t\mathrm{div}(f_{s,Q}) \\
&= t\cdot s(Q) - t([s]Q) - (t\cdot s - t)(\mathcal{O}) \\
&= ts(Q) - ([ts]Q) - (ts-1)(\mathcal{O}) \\
&\quad - (t([s]Q) - ([t][s]Q) - (t-1)(\mathcal{O})) \\
&= \mathrm{div}(f_{ts,Q}) - \mathrm{div}(f_{t,[s]Q})
\end{aligned}$$

Lemma 2. *The Miller function satisfies the property*

$$f_{(t-1)^k,Q}(P) = f_{t-1,Q}^{(t-1)^{k-1}} \cdot f_{t-1,[t-1]Q}^{(t-1)^{k-2}} \cdot f_{t-1,[(t-1)^2]Q}^{(t-1)^{k-3}} \cdot f_{t-1,[(t-1)^3]Q}^{(t-1)^{k-4}} \cdots f_{t-1,[(t-1)^{k-3}]Q}^{(t-1)^2} \cdot f_{t-1,[(t-1)^{k-2}]Q}^{t-1} \cdot f_{t-1,[(t-1)^{k-1}]Q} \cdot \tag{1.40}$$

We continue the divisor computations to obtain this lemma, with $s = t-1$ to simplify the notations.

$$\begin{aligned}
f_{s^k,Q}(P) &= f_{s^{k-1},Q}^s \cdot f_{s,[s^{k-1}]Q} \\
&= f_{s^{k-2},Q}^{s^2} \cdot f_{s,[s^{k-2}]Q}^s \cdot f_{s,[s^{k-1}]Q} \\
&= f_{s^{k-3},Q}^{s^3} \cdot f_{s,[s^{k-3}]Q}^{s^2} \cdot f_{s,[s^{k-2}]Q}^s \cdot f_{s,[s^{k-1}]Q} \\
&= \cdots \\
&= f_{s,Q}^{s^{k-1}} \cdot f_{s,[s]Q}^{s^{k-2}} \cdot f_{s,[s^2]Q}^{s^{k-3}} \cdot f_{s,[s^3]Q}^{s^{k-4}} \cdots f_{s,[s^{k-3}]Q}^{s^2} \cdot f_{s,[s^{k-2}]Q}^s \cdot f_{s,[s^{k-1}]Q} \cdot
\end{aligned}$$

The next observation of Hess, Smart and Vercauteren is to note that the iterated computations of $[s^j]Q = [(t-1)^j]Q$ can be performed very efficiently with the Frobenius endomorphism: $[(t-1)^j]Q = \pi_q^j(Q)$ and moreover, since we evaluate this function at $P \in E(\mathbb{F}_p)$ with $\pi_q(P) = P$,

$$f_{t-1,[(t-1)^j]Q}(P) = f_{t-1,\pi_q^j(Q)}(P) = (f_{t-1,Q}(P))^{\sigma_q^j}$$

with σ_q the q-th power Frobenius in $\mathbb{F}_{q^k}$. We obtain this third lemma.

Lemma 3.

$$f_{(t-1)^k,Q}(P) = f_{t-1,Q}^{(t-1)^{k-1}} \cdot f_{t-1,Q}^{(t-1)^{k-2}\sigma_q} \cdot f_{t-1,Q}^{(t-1)^{k-3}\sigma_q^2} \cdot f_{t-1,Q}^{(t-1)^{k-4}\sigma_q^3} \cdots f_{t-1,Q}^{(t-1)^2\sigma_q^{k-3}} \cdot f_{t-1,Q}^{(t-1)\sigma_q^{k-2}} \cdot f_{t-1,Q}^{\sigma_q^{k-1}} \cdot \tag{1.41}$$

We can also simplify the terms $f_{t-1,Q}^{(t-1)^{k-1-j}\sigma_q^j}$, $0 \leqslant j \leqslant k-1$. The pairing output is of order m, i.e. $f_{m,Q}(P)^m = 1 \in \mathbb{F}_{q^k}$. Let $f \in \mathbb{F}_{q^k}$ of order m. Then $f^{\sigma_q} = f^q \equiv f^{q \mod m} \equiv f^{t-1}$ up to m-th powers and

more generally, $f^{\sigma_q^j} \equiv f^{(t-1)^j}$. We deduce that $(f_{t-1,Q}(P))^{(t-1)^{k-1-j}} \equiv (f_{t-1,Q}(P))^{\sigma_q^{k-1-j}}$ up to m powers $\in \mathbb{F}_{q^k}$ and

$$f_{t-1,Q}^{(t-1)^{k-1-j}\sigma_q^j} \equiv f_{t-1,Q}^{\sigma_q^{k-1-j}\sigma_q^j} \equiv f_{t-1,Q}^{\sigma_q^{k-1}} .$$

We can conclude that $(f_{(t-1)^k,Q}(P)) \equiv (f_{t-1,Q}(P))^{k\ \sigma_q^{k-1}}$ and since $\sigma_q^k = \mathrm{Id}$ in $\mathbb{F}_{q^k}$,

$$(f_{(t-1)^k,Q}(P))^{\sigma_q} \equiv (f_{t-1,Q}(P))^k . \tag{1.42}$$

We can now conclude about the ate pairing. We recall that $N = \gcd((t-1)^k - 1, q^k - 1)$ and $(t-1)^k - 1 = L \cdot N$. Then

$$\begin{aligned}
e_{\text{ReducedTate},m}(Q,P) &= f_{m,Q}(P)^{\frac{q^k-1}{m}} = f_{N,Q}(P)^{\frac{q^k-1}{N}} \\
e_{\text{ReducedTate},m}(Q,P)^L &= f_{N,Q}(P)^{\frac{q^k-1}{N}L} = f_{NL,Q}(P)^{\frac{q^k-1}{N}} \\
&= f_{(t-1)^k-1,Q}(P)^{\frac{q^k-1}{N}} \\
&= \left(f_{(t-1)^k,Q}(P) \cdot f_{-1,Q}(P)\right)^{\frac{q^k-1}{N}} = f_{(t-1)^k,Q}(P)^{\frac{q^k-1}{N}} \\
e_{\text{ReducedTate},m}(Q,P)^{L\ \sigma_q} &= f_{(t-1)^k,Q}(P)^{\sigma_q\ \frac{q^k-1}{N}} \\
&= f_{t-1,Q}(P)^{k\frac{q^k-1}{N}} = f_{t-1,Q}(P)^{k\frac{q^k-1}{m\cdot n}}
\end{aligned}$$

We multiply by n both sides to obtain

$$e_{\text{ReducedTate},m}(Q,P)^{L\cdot n\ \sigma_q} = e_{\text{ate},m}(Q,P)^k$$

with $N = m \cdot n$. Since $L \cdot n \cdot m = (t-1)^k - 1$, we rewrite $L \cdot n = \frac{(t-1)^k-1}{m}$ and obtain Th. 5, with σ_q the q-power Frobenius:

$$e_{\text{ReducedTate},m}(Q,P)^{\frac{(t-1)^k-1}{m}\sigma_q} = e_{\text{ate},m}(Q,P)^k .$$

1.4.4.6 The optimal ate pairing

Vercauteren introduced the optimal ate pairing in [Ver10]. Vercauteren summed-up the ate pairing concept in this way, for an m-torsion point Q and any $\ell \nmid m$:

$$\begin{aligned}
e_{\text{ReducedTate},m}(Q,P)^\ell &= f_{m,Q}(P)^{\frac{q^k-1}{m}\ell} \\
&= f_{\ell\cdot m,Q}(P)^{\frac{q^k-1}{m}} \text{ (see (1.37)).}
\end{aligned} \tag{1.43}$$

Hence the aim is to find ℓ such that ℓm simplifies into a power of a small λ with λ such that $[\lambda](Q)$ is almost free. This means that λ is the eigenvalue of an endomorphism on $\mathbb{G}_2$. With the observation that multiplication by q on $\mathbb{G}_2$ is a Frobenius map (hence almost free) and is the identity on $\mathbb{G}_1$, the ate pairing takes $\lambda \equiv q \mod m$. In his paper on optimal ate pairings, Vercauteren introduced an efficient way to compute $f_{\ell m,Q}(P)$ instead of $f_{\lambda,Q}(P)$. The idea here is to express ℓm in term of powers of q with small coefficients.

Theorem 6 ([Ver10, Th. 1])**.** *Let $\lambda = \ell \cdot m$ with $\ell \nmid m$ and write $\lambda = \sum_{i=0}^{e} c_i q^i$ then*

$$\begin{aligned}
e_{[c_0,c_1,\ldots,c_e]} : \mathbb{G}_2 \times \mathbb{G}_1 &\to \mathbb{G}_T \simeq \mu_m \\
(Q,P) &\mapsto \left(\prod_{i=0}^{e} f_{c_i,Q}^{q^i}(P) \cdot \prod_{i=0}^{e-1} \frac{\ell_{[s_{i+1}]Q,[c_i q^i]Q}(P)}{v_{[s_i]Q}(P)}\right)^{\frac{q^k-1}{r}} \text{ with } s_i = \sum_{j=i}^{e} c_j q^j
\end{aligned} \tag{1.44}$$

defines a bilinear pairing. Furthermore, if

$$mkq^{k-1} \not\equiv \frac{q^k-1}{m} \sum_{i=0}^{e} i c_i q^{i-1} \mod m, \tag{1.45}$$

then the pairing is non-degenerate.

This definition is well-suited for pairing computations on Barreto-Naehrig curves.

Example 9 ([Ver10, §4]). *Observe that*

$$\begin{array}{rcccl} \zeta_{12} & \equiv & q & \equiv & 6x^2 \mod m, \\ \zeta_6 & \equiv & q^2 & \equiv & -(36x^3+18x^2+6x+1) \mod m, \\ \zeta_4 & \equiv & q^3 & \equiv & -(36x^3+24x^2+12x+3) \mod m \end{array} \tag{1.46}$$

and that

$$\zeta_{12} - \zeta_6 + \zeta_4 + 6x + 2 \equiv 0 \mod m\,.$$

A possibility for an optimal ate pairing on a BN curve is then

$$e_{\text{opt.ate},m}(Q,P) = \left(f_{6x+2,Q}(P)\cdot \ell_{Q_3,-Q_2}(P)\cdot \ell_{-Q_2+Q_3,Q_1}(P)\cdot \ell_{Q_1-Q_2+Q_3,[6x+2]Q}(P)\right)^{\frac{q^k-1}{m}} \tag{1.47}$$

with $Q_i = [q^i]Q = \pi_{q^i}(Q)$. The Miller function $f_{6x+2,Q}(P)$ has length $\log p/4$, instead of $\log p/2$ for an ate pairing and $\log p$ for a Tate pairing. This pairing is implemented in Sec. 3.2, see. Alg. 18.

We explain this optimal ate pairing computation. The endomorphisms of eigenvalues $\zeta_{12}, \zeta_6, \zeta_4$ are efficiently computable on $E'(\mathbb{F}_{q^2})$ and $E(\mathbb{F}_{q^k})$. They cost less than a doubling and we know explicitly their eigenvalue modulo m, see Ex. 9. We set $\lambda = t-1 = 6x^2$, λ is the eigenvalue of $\zeta_{12} \mod m$. We have

$$\lambda - \lambda^2 + \lambda^3 + 6x + 2 = 6x^2 - 36x^4 + 216x^6 + 6x + 2 = m\cdot(6x^2 - 6x + 2)\,. \tag{1.48}$$

We set $N = (6x^2-6x+2)\cdot m$ and $n = 6x^2-6x+2$. We have the equalities

$$\begin{array}{lcl} e_{\text{Tate},m}(Q,P)^{\frac{p^k-1}{m}} & = & f_{m,Q}(P)^{\frac{p^k-1}{m}} = f_{N,Q}(P)^{\frac{p^k-1}{N}} \\ e_{\text{Tate},m}(Q,P)^{\frac{p^k-1}{m}(6x^2-6x+2)} & = & f_{N,Q}(P)^{\frac{p^k-1}{m}} \end{array} \tag{1.49}$$

We now decompose the N in $f_{N,Q}(P)$ in terms of eigenvalues of endomorphisms.

$$\begin{array}{rcl} f_{N,Q}(P) & = & f_{\lambda-\lambda^2+\lambda^3+6x+2,Q}(P) \\ & = & f_{\lambda-\lambda^2+\lambda^3,Q}(P) f_{6x+2,Q}(P) \frac{\ell_{[\lambda-\lambda^2+\lambda^3]Q,[6x+2]Q}}{v_{[\lambda-\lambda^2+\lambda^3+6x+2]Q}}(P)\,. \end{array} \tag{1.50}$$

Since Q is an m-torsion point, $[\lambda-\lambda^2+\lambda^3+6x+2]Q = \mathcal{O}$ and the line and vertical can be removed from computations. Then we decompose $f_{\lambda-\lambda^2+\lambda^3,Q}(P)$ first with the additive property from (1.32) and secondly in the same way as in Lem. 2 and 3

$$\begin{array}{rcl} f_{\lambda-\lambda^2+\lambda^3,Q} & = & f_{\lambda,Q} f_{-\lambda^2+\lambda^3,Q} \frac{\ell_{[\lambda]Q,[-\lambda^2+\lambda^3]Q}}{v_{[\lambda-\lambda^2+\lambda^3]Q}} \\ f_{-\lambda^2+\lambda^3,Q} & = & f_{-\lambda^2,Q} f_{\lambda^3,Q} \frac{\ell_{[-\lambda^2]Q,[\lambda^3]Q}}{v_{[-\lambda^2+\lambda^3]Q}} \\ f_{\lambda-\lambda^2+\lambda^3,Q} & = & f_{\lambda,Q} f_{-\lambda^2,Q} f_{\lambda^3,Q} \frac{\ell_{[-\lambda^2]Q,[\lambda^3]Q}}{v_{[-\lambda^2+\lambda^3]Q}} \frac{\ell_{[\lambda]Q,[-\lambda^2+\lambda^3]Q}}{v_{[\lambda-\lambda^2+\lambda^3]Q}}\,. \end{array} \tag{1.51}$$

We can remove the vertical lines since they disappear after the final exponentiation. We decompose each term $f_{\lambda^j,Q}$ with the property

$$f_{\lambda^j,Q}(P) = f_{\lambda,Q}(P)^{j\lambda^{j-1}} \tag{1.52}$$

with λ the eigenvalue of $q \mod m$. This equality 1.52 is obtained directly from

$$f_{(t-1)^k,Q}(P) = f_{t-1,Q}(P)^{k\sigma_q^{k-1}} = f_{t-1,Q}(P)^{k(t-1)^{k-1}} \text{ since we are in } \mu_m \subset \mathbb{F}_{q^k}^*\,.$$

We obtain

$$\left(f_{\lambda,Q} f_{-\lambda^2,Q} f_{\lambda^3,Q}\right)(P) = f_{\lambda,Q}(P)^{1-2\lambda+3\lambda^2} \tag{1.53}$$

and we conclude that

$$\begin{aligned} f_{N,Q}(P) &= f_{\lambda-\lambda^2+\lambda^3+6x+2,Q}(P) \\ &= f_{6x+2,Q}(P) f_{\lambda,Q}(P)^{1-2\lambda+3\lambda^2} \ell_{[-\lambda^2]Q,[\lambda^3]Q}(P) \ell_{[\lambda]Q,[-\lambda^2+\lambda^3]Q}(P) \,. \end{aligned} \tag{1.54}$$

As previously noted by Naehrig, Niederhagen and Schwabe in [NNS10, §2], we can remove the line computation $\ell_{Q_1-Q_2+Q_3,[6x+2]Q}(P)$ since this is a vertical line. Indeed, $\lambda - \lambda^2 + \lambda^3 + 6x + 2 \equiv 0 \mod m$ as stated in (1.48), Q is an m-torsion point and $\pi_q(Q) - \pi_q^2(Q) + \pi_q^3(Q) = Q_1 - Q_2 + Q_3 = -[6x+2]Q$.

We finish, with $n = 6x^2 - 6x + 2$ and $N = m \cdot n$:

$$\begin{aligned} e_{\text{ReducedTate},m}(Q,P)^n &= f_{N,Q}(P)^{\frac{q^k-1}{m}} \\ &= f_{6x+2,Q}(P)^{\frac{q^k-1}{m}} f_{\lambda,Q}(P)^{\frac{q^k-1}{m}(1-2\lambda+3\lambda^2)} \ell_{[-\lambda^2]Q,[\lambda^3]Q}(P) \ell_{[\lambda]Q,[-\lambda^2+\lambda^3]Q}(P) \\ &= e_{\text{opt ate}}(Q,P) e_{\text{ate}}(Q,P)^{1-2\lambda+3\lambda^2} \,. \end{aligned} \tag{1.55}$$

We can also deduce that

$$e_{\text{opt ate},m}(Q,P) = e_{\text{ate},m}(Q,P)^{-3(6x^3+6x^2+3x+1)} \tag{1.56}$$

Chapter 2

Genus 2 Jacobians: isogenies, point counting and endomorphisms

This chapter studies the properties of two families of splitting genus two curves. We will introduce $\mathcal{C}_1 : Y^2 = X^5 + aX^3 + bX$ and $\mathcal{C}_2 : Y^2 = X^6 + aX^3 + b$ defined over a finite field $\mathbb{F}_q$. Both are genus two hyperelliptic curves. They are moreover isogenous over a small degree extension field to the product of two elliptic (i.e. genus one) curves. We explicit the isogeny in terms of divisors of the Jacobian in Sec. 2.2. Satoh and Freeman [Sat09, FS11] studied these curves and proposed an efficient point-counting algorithm thanks to the isogenies. We present a refinement of their method in Sec. 2.3. This work was published at the *PAIRING'2012* conference [GV12]. In Sec. 2.6 we propose pairing-friendly constructions for genus 2 curves of the form $\mathcal{C}_1$ and $\mathcal{C}_2$. This was also published in he same paper at *PAIRING'2012* [GV12].

This is just the beginning of the interesting properties of these curves. We explain in Sec.2.4 that the two isogenous elliptic curves have a very interesting property: in certain conditions easily met, we can construct two different endomorphisms on these curves. Their eigenvalues are far enough to use them as if they where independent. These two endomorphisms can be used to speed-up a scalar multiplication. This property was independently discovered by Smith [Smi13] in a completely different way, and used for different applications. In our work we also sketch in Sec. 2.5 the computations to obtain two corresponding endomorphisms on the Jacobians. This work was presented as an invited talk at the ECC 2013 workshop in Leuven, Belgium and was accepted to be presented at the Asiacrypt 2013 conference in Bengalore, India.

2.1 Preliminaries

In 1985, the idea of using the group of rational points on an elliptic curve over a finite field in public-key cryptography was introduced independently by Miller [Mil86b] and Koblitz [Kob87]. The main advantage of using elliptic curves is efficiency since no sub-exponential algorithms are known for solving the discrete logarithm problem in these groups (and thus key sizes can remain small). In 1989, Koblitz [Kob89] suggested using Jacobians of hyperelliptic curves in cryptography. Genus 1 hyperelliptic curves are elliptic curves; genus 2 and 3 hyperelliptic curves are more complicated but are an attractive replacement for elliptic curves in cryptography. They are as efficient as genus one curves for bandwidth but still have a slower group law.

As for any group used for the discrete logarithm problem, one needs the order of the group to contain a large prime factor. This raised the problem of finding hyperelliptic curves over a finite field whose Jacobian order is (almost) a prime. For elliptic curves over finite fields, the Schoof-Elkies-Atkin (SEA) algorithm [Sch98, LLV05] runs in polynomial time in any characteristic and in small characteristic, there are even faster algorithms based on the so-called p-adic method [Sat02, LLV05]. For genus 2 hyperelliptic curves, the p-adic method gives efficient point counting algorithms in small characteristic, but up to now, no algorithms as efficient as SEA are known when the characteristic of the underlying finite field is large (though substantial progress has recently been made in [GKS11] and [GS12]). The strategy is then to select a particular case which reduces to already known point-counting methods. Using basic properties on character sums, Furukawa, Kawazoe and Takahashi [FKT04] gave an explicit closed formula for the

order of Jacobians of very special curves of type $Y^2 = X^5 + bX$ where $b \in \mathbb{F}_q$. Satoh [Sat09] considered an intermediate approach and showed that point counting on specific Jacobians of certain genus 2 curves can be performed much faster than point counting on Jacobians of generic curves. He gave an algorithm to test whether the order of the Jacobian of a given hyperelliptic curve in the form $Y^2 = X^5 + aX^3 + bX$ has a large prime factor. His method relies on the fact that the Jacobian of the curve is $\mathbb{F}_{q^4}$-isogenous to a square of an elliptic curve defined over $\mathbb{F}_{q^4}$, hence their respective zeta functions are the same over $\mathbb{F}_{q^4}$ and can be computed by the SEA algorithm. Satoh's method obtains candidates for the zeta function of the Jacobian over $\mathbb{F}_q$ from the zeta function over $\mathbb{F}_{q^4}$. The methodology can be formalized as an efficient probabilistic polynomial algorithm but is not explicit and gives 26 possible orders to test for the Jacobian.

The second requirement on a group used for the discrete logarithm problem is an efficient *exponentiation* (denoted g^x with a multiplicative notation such as on $\mathbb{F}_q^*$) or *scalar multiplication* (denoted $[x]P$ with the additive notation of elliptic curves). Various techniques were introduced to speed-up the scalar multiplication. Firstly there exist exponent-recoding techniques such as sliding window and Non-Adjacent-Form representation. These techniques are valid for generic groups and improved for elliptic curves as the inversion (or negation in additive notation) is free.

Secondly, in 2001, Gallant, Lambert and Vanstone [GLV01] introduced a method which uses endomorphisms on the elliptic curve to decompose the scalar multiplication in a 2-dimensional multi-multiplication. Given an elliptic curve E defined over a prime finite field $\mathbb{F}_p$ with a fast endomorphism ϕ and a point P of large prime order m such that $\phi(P) = [\lambda]P$, the computation of $[k]P$ is decomposed as

$$[k]P = [k_1]P + [k_2]\phi(P),$$

with $k = k_1 + \lambda k_2 \pmod m$ such that $|k_1|, |k_2| \simeq \sqrt{m}$. Gallant et al. provided examples of curves whose endomorphism ϕ is given by: complex-multiplication by $\sqrt{-1}$ (j-invariant $j = 1728$), $\frac{1+\sqrt{-3}}{2}$ ($j = 0$), $\sqrt{-2}$ ($j = 8000$) and $\frac{1+\sqrt{-7}}{2}$ ($j = -3375$). These examples were well-known in algebraic geometry, e.g. they are presented as toy examples in [Sil94, II, Prop. 2.3.1]. We explained where these examples come from in Sec. 1.2.10.1.

In 2009 Galbraith, Lin and Scott [GLS09] presented a very efficient method to construct an efficient endomorphism on elliptic curves E defined over $\mathbb{F}_{p^2}$ which are quadratic twists of elliptic curves defined over $\mathbb{F}_p$. In this case, a fast endomorphism ψ is obtained by carefully exploiting the Frobenius endomorphism. This endomorphism verifies the equation $\psi^2 + 1 = 0$ on $E(\mathbb{F}_{p^2})$. In 2012, Longa and Sica improved the GLS construction, by showing that a 4-dimensional decomposition of scalar multiplication is possible, on GLS curves allowing efficient complex multiplication ϕ. Let λ, μ denote the eigenvalues of the two endomorphisms ϕ, ψ. Then we can decompose the scalar k into $k = k_0 + k_1\lambda + k_2\mu + k_3\lambda\mu$ and compute

$$[k]P = [k_0]P + [k_1]\phi(P) + [k_2]\psi(P) + [k_3]\phi \circ \psi(P)\ .$$

Note that most curves presented in the literature have particular j-invariants. GLV curves have j-invariant 0, 1728, 8000, or -3375, while GLS curves have j-invariant in $\mathbb{F}_p$, even though they are defined over $\mathbb{F}_{p^2}$.

In 2013, Bos, Costello, Hisil and Lauter proposed in [BCHL13b] a 4-dimensional GLV technique to speed-up scalar multiplication in genus 2. They considered the Buhler-Koblitz genus 2 curves $y^2 = x^5 + b$ and the Furukawa-Kawazoe-Takahashi curves $y^2 = x^5 + ax$. These two curves have a very efficient dimension-4 GLV technique available. On BK curves, they proposed 2-dimensional GLV on the corresponding Kummer surface. Recently at *CHES'2013* the same authors [BCHL13a] proposed a 8-GLV scalar decomposition on genus-2 Buhler-Koblitz curves defined over a quadratic extension field. They choose the primes $p = 2^{61} - 1$, $p = 2^{64} - 189$, $p = (2^{31} - 307656) \cdot 2^{32} - 1$ and target a 112-bit security level. The parameter sizes are not optimal because of Weil descent attack nevertheless their implementation is well-suited for 32-bit and 64-bit architectures.

In Sec. 2.4 and 2.5 we provide two new families of genus-2 curves defined over a prime field, and elliptic curves defined over a quadratic extension field whose j-invariant is in $\mathbb{F}_{p^2}$ (contrary to the previous constructions where $j \in \mathbb{F}_p$). A four dimensional GLV decomposition technique is available on this curves.

In recent years, many useful cryptographic protocols have been proposed that make use of a bilinear map, or *pairing*, between two groups in which the discrete logarithm problem is hard (*e.g.* [BF01, BF03,

BLS01, BLS04]). Pairing-based cryptosystems can be constructed by using the Weil or Tate pairing on abelian varieties over finite fields. These pairings take as input points on an abelian variety defined over the field $\mathbb{F}_q$ and produce as output elements of an extension field $\mathbb{F}_{q^k}$. The degree of this extension is known as the *embedding degree*. In cryptography, abelian varieties obtained as Jacobians of hyperelliptic curves are often used. Suitable hyperelliptic curves for pairing-based cryptography are called *pairing-friendly*. Such pairing-friendly curves are rare and thus require specific constructions.

For a pairing-based cryptosystem to be secure and practical, the group of rational points on the Jacobian should have a subgroup of large prime order r, and the embedding degree k should be large enough so that the discrete logarithm problem in $\mathbb{F}_{q^k}$ is difficult but small enough to make the pairing efficiently computable. The efficiency parameter in pairing-friendly constructions is the so-called *ρ-value*: for a Jacobian of hyperelliptic curve of genus g it is defined as $\rho = g \log q / \log r$. It measures the ratio of the bit-sizes of the order of the Jacobian and the subgroup order r. The problem of constructing pairing-friendly elliptic curves with small ρ-values has been studied extensively [FST10]. Unfortunately, there are very few results for constructing pairing-friendly hyperelliptic curves of genus $g \geq 2$ with small ρ-values [GHV07, BBC$^+$11a]. Galbraith, Pujolas, Ritzenthaler and Smith [GPRS09] gave (supersingular) genus 2 pairing-friendly hyperelliptic curves with ρ-values close to 1 but only for embedding degrees $k \in \{4, 5, 6, 12\}$. Freeman, Stevenhagen and Streng presented in [FSS08] a general method that produced pairing-friendly (ordinary) genus 2 pairing-friendly hyperelliptic curves with $\rho \simeq 8$ for all embedding degrees k. Kawazoe and Takahashi [KT08] (see also [Kac10]) presented an algorithm which constructed hyperelliptic curves of the form $Y^2 = X^5 + bX$ (thanks to the closed formula for its Jacobian order). Following Satoh's approach, Freeman and Satoh [FS11] constructed pairing-friendly genus 2 hyperelliptic curves of the form $Y^2 = X^5 + aX^3 + bX$ and $Y^2 = X^6 + aX^3 + b$ (with $a, b \in \mathbb{F}_q^*$) by means of elliptic curves that become pairing-friendly over a finite extension of the underlying finite field. Constructions from [KT08, Kac10, FS11] produce pairing-friendly Jacobians with $2.22 \leqslant \rho \leqslant 4$ only for embedding degrees divisible by 3 or 4.

Our contributions.

Satoh's approach to compute the Jacobian order of a hyperelliptic curve $Y^2 = X^5 + aX^3 + bX$ is not explicit. For each candidate, he has to check that the order is not weak for cryptographic use. In [GS01, §4], Gaudry and Schost showed that the Jacobians of hyperelliptic curves of the form $Y^2 = X^6 + aX^3 + b$ are also isogenous to a product of two elliptic curves over an extension field. Satoh claimed that his method applies as well to this family but did not derive an algorithm for it.

Our first contribution is to extend and generalize Satoh's idea to provide *explicit* formulas for the zeta function of the Jacobian of genus 2 hyperelliptic curves of the form $Y^2 = X^5 + aX^3 + bX$ and $Y^2 = X^6 + aX^3 + b$ (with $a, b \in \mathbb{F}_q^*$). Our results are proved by elementary polynomial root-finding techniques. This permits to generate efficiently a random hyperelliptic curve, in one of these two forms, suitable for cryptographic use. These curves enable various improvements to make scalar multiplication in the Jacobian efficient (*e.g.* the Gallant-Lambert-Vanstone algorithm [GLV01], Takashima's algorithm [Tak06] or Gaudry's algorithm [Gau07]). These large families of curves are still very specific but there is no evidence that they should be more vulnerable to discrete logarithm attacks than the absolutely simple Jacobians.

Two algorithms proposed in [FS11] to produce pairing-friendly genus 2 hyperelliptic curves are very general as they are still valid for arbitrary abelian varieties over any finite field. Assuming that the finite field is a prime field and the abelian variety is of the above form, we can consider any embedding degree. The security restrictions concerning the embedding degree (which must be a multiple of 3 or 4) made in [FS11] are unnecessary in this particular case. Satoh and Freeman exclude constructions which need an elliptic curve defined over a quadratic extension of a prime field (with j-invariant in $\mathbb{F}_{p^2}$), resulting in restricted sets of parameters $a, b \in \mathbb{F}_p$. Using our closed formulas for the Jacobian order, we use two approaches that construct pairing-friendly elliptic curves and adapt them to produce pairing-friendly genus 2 curves. The first one is based on the Cocks-Pinch method [CP01] (see also [BSS05, Algorithm IX.4]) of constructing individual ordinary pairing-friendly elliptic curves. The other is based on cyclotomic polynomials as originally proposed by Brezing and Weng [BW05] which generates families of curves while

achieving better ρ-values. We adapt both constructions using the elliptic curve complex multiplication method (CM) [AM93, BSS05] to compute one of the two elliptic curves to which the Jacobian is isogenous to (even if the curve j-invariant is in $\mathbb{F}_{p^2}$ rather than in a prime field $\mathbb{F}_p$). In particular, this method can construct pairing-friendly elliptic curves over $\mathbb{F}_{p^2}$ but unfortunately with $\rho \simeq 4$.

Our approach contains the previous constructions by Kawazoe and Takahashi [KT08] and is in a sense a specialization of Freeman and Satoh [FS11]. It also produces new families for ordinary genus 2 hyperelliptic curves. Explicit examples of cryptographically interesting curves are given.

2.2 Two splitting Jacobians

In the following, $p \geq 5$ denotes a prime number and q a power of p. In this section, we consider the genus 2 hyperelliptic curves defined over a finite field $\mathbb{F}_q$:

$$\mathcal{C}_1 : Y^2 = X^5 + aX^3 + bX, \tag{2.1}$$

with $a, b \neq 0 \in \mathbb{F}_q$. We denote by $J_{\mathcal{C}_1}$ the Jacobian of this curve. The Jacobian splits into the product of two isogenous elliptic curves in an extension of $\mathbb{F}_q$ of degree 1, 2, 4 or 8 [Sat09].

We will also consider the genus 2 curves defined over $\mathbb{F}_q$

$$\mathcal{C}_2 : Y^2 = X^6 + aX^3 + b, \tag{2.2}$$

with $a, b \neq 0 \in \mathbb{F}_q$. In the same way, we denote by $J_{\mathcal{C}_2}$ the Jacobian of the curve. The Jacobian splits into the product of two isogenous elliptic curves in an extension of $\mathbb{F}_q$ of degree 1, 2, 3 or 6.

A Jacobian which never splits into lower genus Jacobians is an *absolutely simple* Jacobian. A splitting Jacobian is a *non-simple* Jacobian. Here the Jacobian is *non absolutely simple*. We aim to investigate the isogeny in order to count the number of points of the Jacobian and transport the endomorphisms available on the genus 1 curve to the Jacobian.

We will not consider Satoh's isogeny [Sat09, §3] as in [GV12, §2.1] rather consider Freeman and Satoh's isogeny given in [FS11, Proof of Prop. 4.1].

We will explicit the isogeny with respect to divisors of the Jacobian (and not simply maps between points on the genus one curve and the genus two curve). A divisor $\mathcal{D} \in J_{\mathcal{C}_1}(\mathbb{F}_q)$ is given by two points $P_1 = (X_1, Y_1), P_2 = (X_2, Y_2)$ on $\mathcal{C}_1$ and the Mumford representation is $\mathcal{D} = (u_1, u_0, v_1, v_0)$ with

$$u_1 = -(X_1 + X_2),\ u_0 = X_1 + X_2,\ v_1 = \frac{Y_1 - Y_2}{X_1 - X_2},\ v_0 = \frac{X_1 Y_2 - X_2 Y_1}{X_1 - X_2} \tag{2.3}$$

and the u_i, v_i are in $\mathbb{F}_q$ (as explained in Sec. 1.3.2). In particular we have $v_1 X_1 + v_0 = Y_1$ hence $-v_1 u_1 + 2v_0 = Y_1 + Y_2$.

2.2.1 Isogeny from $J_{\mathcal{C}_1}$ into two elliptic curves $E_{1,c} \times E_{1,c}$

It was shown in [LM97, Sat09, FS11, §2, §3, §4.1] that the Jacobian of $\mathcal{C}_1$ is isogenous to the product of the two elliptic curves $E_{1,c} \times E_{1,c}$. The curve $E_{1,c}$ is defined over $\mathbb{F}_q[\sqrt{b}]$ by

$$E_{1,c} : y^2 = (c+2)x^3 - (3c-10)x^2 + (3c-10)x - (c+2) \tag{2.4}$$

with $c = a/\sqrt{b}$ which is in $\mathbb{F}_q$ or $\mathbb{F}_{q^2}$. The j-invariant of this curve is

$$j(E_{1,c}) = 2^6 \frac{(3c-10)^3}{(c+2)^2(c-2)}\ . \tag{2.5}$$

Freeman and Satoh [FS11] gave two maps φ_1, φ_2 from points on the genus 2 curve $\mathcal{C}_1$ to points on the curve $E_{1,c}$. From these two maps the $(2,2)$-isogeny $\mathcal{I}_{(2,2)}$ between $J_{\mathcal{C}_1}$ and $E_{1,c} \times E_{1,c}$ is given by [Sil09, Remark II.3.4]

$$\begin{array}{rcl} \mathcal{I}_{(2,2)} : J_{\mathcal{C}_1} & \to & E_{1,c} \times E_{1,c} \\ P + Q - 2P_\infty & \mapsto & (\varphi_{1*}(P) + \varphi_{1*}(Q), \varphi_{2*}(P) + \varphi_{2*}(Q)) \end{array} \tag{2.6}$$

and its dual is

$$\begin{array}{rcl} \hat{\mathcal{I}}_{(2,2)} : E_{1,c} \times E_{1,c} & \to & J_{\mathcal{C}_1} \\ (S_1, S_2) & \mapsto & \varphi_1^*(S_1) + \varphi_2^*(S_2) - 4P_\infty \end{array} \tag{2.7}$$

with $\varphi_j^*(S_j) = \sum_{P \in \mathcal{C}_1, \varphi_{j*}(P) = S_j} P$. In other words we add the points in the pre-image of S_j with respect to φ_j. We explicit the isogeny $\mathcal{I}_{(2,2)}$ and the maps φ_1 and φ_2.

2.2.1.1 Maps between genus 2 curves

We follow some hints in [CF96]. We need to find an expression of $\mathcal{C}_1$ of the form $Y'^2 = X'^6 + a_4 X'^4 + a_2 X'^2 + a_0$. The isogeny to the elliptic curve will be $(X', Y') \mapsto (X'^2, Y')$. We introduce the genus 2 hyperelliptic curve

$$\mathcal{C}_1' : Y'^2 = X'^5 + cX'^3 + X' \text{ with } c = a/\sqrt{b} \neq 0\,. \tag{2.8}$$

The map from $\mathcal{C}_1$ to $\mathcal{C}_1'$ and the induced isogeny are defined over $\mathbb{F}_q[\sqrt[8]{b}]$ and given by

$$\begin{array}{rcl@{\qquad}rcl} \mathcal{C}_1 & \to & \mathcal{C}_1' & \mathcal{C}_1' & \to & \mathcal{C}_1 \\ (X, Y) & \mapsto & \left(\frac{X}{\sqrt[4]{b}}, \frac{Y}{\sqrt[8]{b}^5}\right) & (X', Y') & \mapsto & \left(X' \sqrt[4]{b}, Y' \sqrt[8]{b}^5\right) \\ J_{\mathcal{C}_1} & \to & J_{\mathcal{C}_1'} & J_{\mathcal{C}_1'} & \to & J_{\mathcal{C}_1} \\ (u_1, u_0, v_1, v_0) & \mapsto & \left[\frac{u_1}{\sqrt[4]{b}}, \frac{u_0}{\sqrt{b}}, \frac{v_1}{\sqrt[8]{b}^3}, \frac{v_0}{\sqrt[8]{b}^5}\right] & (u_1', u_0', v_1', v_0') & \mapsto & [u_1' \sqrt[4]{b}, u_0' \sqrt{b}, v_1' \sqrt[8]{b}^3, v_0' \sqrt[8]{b}^5] \end{array} \tag{2.9}$$

The next step consists in writing the curve in a way we can see the maps to the two elliptic curves. Freeman and Satoh proposed to write $X' = \frac{X''+1}{X''-1}$. We obtain the curve

$$\mathcal{C}_1'' : Y''^2 = (c+2)X''^6 - (3c-10)X''^4 + (3c-10)X''^2 - (c+2) \text{ with } c \neq \pm 2\,. \tag{2.10}$$

The change of variables is

$$\begin{array}{rcl@{\qquad}rcl} \mathcal{C}_1' & \to & \mathcal{C}_1'' & \mathcal{C}_1'' & \to & \mathcal{C}_1' \\ (X', Y') & \mapsto & \left(\frac{X'+1}{X'-1}, \frac{8Y'}{(X'-1)^3}\right) & (X'', Y'') & \mapsto & \left(\frac{X''+1}{X''-1}, \frac{Y''}{(X''-1)^3}\right) \end{array}$$

The map to the elliptic curve is then obvious: we set $(x, y) = (X''^2, Y'')$. This point is on the elliptic curve $E_{1,c}$ defined over $\mathbb{F}_q[c]$ by the equation

$$E_{1,c} : y^2 = (c+2)x^3 - (3c-10)x^2 + (3c-10)x - (c+2) \text{ with } c = a/\sqrt{b}. \tag{2.11}$$

The other map to the same curve uses the following equivalent equalities:

$$\begin{array}{rcl} \mathcal{C}_1'' : Y''^2 & = & (c+2)X''^6 - (3c-10)X''^4 + (3c-10)X''^2 - (c+2) \\ \Leftrightarrow \frac{Y''^2}{X''^6} & = & (c+2) - (3c-10)\frac{1}{X''^2} + (3c-10)\frac{1}{X''^4} - (c+2)\frac{1}{X''^6} \\ \Leftrightarrow \frac{-Y''^2}{X''^6} & = & (c+2)\frac{1}{X''^6} - (3c-10)\frac{1}{X''^4} + (3c-10)\frac{1}{X''^2} - (c+2) \\ \Leftrightarrow \left(\frac{iY''}{X''^3}\right)^2 & = & (c+2)\frac{1}{X''^6} - (3c-10)\frac{1}{X''^4} + (3c-10)\frac{1}{X''^2} - (c+2) \end{array}$$

We recognize here the other map: we set $(x, y) = \left(\frac{1}{X''^2}, \frac{iY''}{X''^3}\right)$. This point is on the same elliptic curve $E_{1,c}$.

The direct formulas of the maps from $\mathcal{C}_1$ to $E_{1,c} \times E_{1,c}$ are the following, with $i \in \mathbb{F}_q$ or $\mathbb{F}_{q^2}$ such that $i^2 = -1$:

$$\begin{array}{rcl@{\qquad}rcl} \varphi_1 : \mathcal{C}_1 & \to & E_{1,c} & \varphi_2 : \mathcal{C}_1 & \to & E_{1,c} \\ (x, y) & \mapsto & \left(\left(\frac{x + \sqrt[4]{b}}{x - \sqrt[4]{b}}\right)^2, \frac{8y\sqrt[8]{b}}{(x - \sqrt[4]{b})^3}\right) & (x, y) & \mapsto & \left(\left(\frac{x - \sqrt[4]{b}}{x + \sqrt[4]{b}}\right)^2, \frac{8iy\sqrt[8]{b}}{(x + \sqrt[4]{b})^3}\right) \end{array} \tag{2.12}$$

The maps are defined over $\mathbb{F}_q[i, \sqrt[8]{b}]$.

The isogeny in terms of divisors requires a quite tedious computation. The critical point is the computation of v_1'' and v_0'', especially the quantity $\frac{Y_2'(X_1'-1)^3-Y_1'(X_2'-1)^3}{X_2'-X_1'}$. With some help from Maple we obtained the following equalities:

$$\begin{aligned}
u_1'' &= -X_1'' - X_2'' = \frac{-2(u_0-\sqrt{b})}{u_0+u_1\sqrt[4]{b}+\sqrt{b}} \\
u_0'' &= X_1''X_2'' = \frac{u_0-u_1\sqrt[4]{b}+\sqrt{b}}{u_0+u_1\sqrt[4]{b}+\sqrt{b}} \\
v_1'' &= \frac{Y_1''-Y_2''}{X_1''-X_2''} = \frac{4}{\sqrt[8]{b}}\frac{(u_1v_0-u_0v_1)u_1-u_0v_0+3(u_1v_0-u_0v_1)\sqrt[4]{b}+3v_0\sqrt{b}+v_1\sqrt[4]{b}^3}{(u_0+u_1\sqrt[4]{b}+\sqrt{b})^2} \\
v_0'' &= \frac{X_1''Y_2''-X_2''Y_1''}{X_1''-X_2''} = \frac{-4}{\sqrt[8]{b}}\frac{(u_1v_0-u_0v_1)u_1-u_0v_0+(u_1v_0-u_0v_1)\sqrt[4]{b}-v_0\sqrt{b}-v_1\sqrt[4]{b}^3}{(u_0+u_1\sqrt[4]{b}+\sqrt{b})^2}
\end{aligned} \tag{2.13}$$

We explain the operation count. Firstly we will store the denominator $z = u_0+u_1\sqrt[4]{b}+\sqrt{b}$ separately. The terms u_1'' and u_0'' have denominator z. Computing their numerator is free. Then u_1'' and u_0'' are free if we consider the two operandes of the fraction independantly. Computing v_1'' and v_0'' can be done with a common precomputation which costs $4M$:

$$\begin{aligned}
s &= u_1v_0 \\
t &= s-u_0v_1 \\
s &= tu_1 \\
r &= s-u_0v_0
\end{aligned}$$

and

$$\begin{aligned}
v_1'' &= \frac{4}{\sqrt[8]{b}}\frac{r+3t\sqrt[4]{b}+3v_0\sqrt{b}+v_1\sqrt[4]{b}^3}{(u_0+u_1\sqrt[4]{b}+\sqrt{b})^2}, \\
v_0'' &= \frac{-4}{\sqrt[8]{b}}\frac{r+t\sqrt[4]{b}-v_0\sqrt{b}-v_1\sqrt[4]{b}^3}{(u_0+u_1\sqrt[4]{b}+\sqrt{b})^2}.
\end{aligned}$$

The inverse map is the following.

$$\begin{aligned}
u_1 = \sqrt[4]{b}u_1' &= 2\sqrt[4]{b}\frac{1-u_0''}{1+u_0''+u_1''} \\
u_0 = \sqrt{b}u_0' &= \sqrt{b}\frac{u_0''-u_1''+1}{u_0''+u_1''+1} \\
v_1 = \sqrt[8]{b}^3v_1' &= \sqrt[8]{b}^3\frac{1}{2}\frac{(u_1''v_0''-u_0''v_1'')u_1''-u_0''v_0''+3(u_1''v_0''-u_0''v_1'')+3v_0''+v_1''}{(u_0''+u_1''+1)^2} \\
v_0 = \sqrt[8]{b}^5v_0' &= \sqrt[8]{b}^5\frac{1}{2}\frac{(-u_1''v_0''+u_0''v_1'')u_1''+u_0''v_0''-u_1''v_0''+u_0''v_1''+v_0''+v_1''}{(u_0''+u_1''+1)^2}
\end{aligned}$$

2.2.1.2 Computing $\mathcal{I}_{(2,2)}$ on $J_{\mathcal{C}_1}(\mathbb{F}_q)$.

We show first how to compute explicitly the $(2,2)$-isogeny on $J_{\mathcal{C}_1}(\mathbb{F}_q)$ with only a small number of operations over the extension fields of $\mathbb{F}_q$. Let $\mathcal{D}$ be a divisor in $J_{\mathcal{C}_1}(\mathbb{F}_q)$ given by its Mumford coordinates

$$\mathcal{D} = (u_1, u_0, v_1, v_0),\ u_0, u_1, v_0, v_1 \in \mathbb{F}_q\,.$$

It corresponds to two points $P_1 = (X_1, Y_1), P_2 = (X_2, Y_2) \in \mathcal{C}_1(\mathbb{F}_q)$ or $\mathcal{C}_1(\mathbb{F}_{q^2})$. The correspondance between $\mathcal{D}$ and the two points is given in eq. (2.3). The generic formula for the isogeny is given in eq. 2.6. We will now explain this isogeny. We need to express φ_{1*} with respect to $\mathcal{D} = (u_1, u_0, v_1, v_0)$. We will

proceed in two steps. We already know $\mathcal{D}'' \in J_{\mathcal{C}_1''}$ with respect to $\mathcal{D} \in J_{\mathcal{C}_1}$. We will compute $\varphi_{1*}(\mathcal{D}'')$, then $\varphi_{1*}(\mathcal{D})$. We express the map which sends a divisor $\mathcal{D}'' \in J_{\mathcal{C}_1''}$ to two points on $E_{1,c}$, then we add the two points to obtain one point on $E_{1,c}$, then go back to $\mathcal{C}_1''$ where we get two points on the curve, then add the two points to get one divisor. Let $\mathcal{D}''$ a divisor in $J_{\mathcal{C}_1''}$. We send with φ_{i*} the two points $P_1'' = (X_1'', Y_1''), P_2'' = (X_2'', Y_2'')$ corresponding to $\mathcal{D}''$ in $E_{1,c}$ and add them with the addition law on $E_{1,c}$. We recall that $P_j'' \mapsto (X_j''^2, Y_j'') = (x_j, y_j) \in E_{1,c}$ which is defined over $\mathbb{F}_q[\sqrt{b}]$ by

$$E_{1,c}: \; y^2 = (c+2)x^3 - (3c-10)x^2 + (3c-10)x - (c+2) \; .$$

We denote

$$S = (x_3, y_3) = \varphi_{1,*}(P_1) + \varphi_{1,*}(P_2) \in E_{1,c} \; .$$

We will also use the representation with the two-torsion point, namely

$$E_{1,c}: \; y^2 = (c+2)(x-1)^3 + 2(3c-2)(x-1)^2 + 2(3c-2)(x-1) \; .$$

We observed that the expression of x_3 is simpler with this representation. The addition law is then

$$\begin{array}{rcl} \lambda & = & \frac{y_2-y_1}{(x_2-1)-(x_1-1)} = \frac{y_2-y_1}{x_2-x_1}, \\ x_3 - 1 & = & \frac{\lambda^2}{c+2} - (x_1+x_2-2) - 2 - \frac{-3c+10}{c+2} - 1 = \frac{\lambda^2}{c+2} - (x_1+x_2-2) - \frac{16}{c+2}, \\ y_3 & = & \lambda(x_1 - x_3) - y_1 \; . \end{array}$$

We add the two points with this addition law.

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{Y_2'' - Y_1''}{X_2''^2 - X_1''^2} = \frac{Y_2'' - Y_1''}{(X_2'' - X_1'')(X_2'' + X_1'')} = \frac{v_1''}{-u_1''} \; .$$

We continue with x_3. We note that $x_1 + x_2 - 2 = u_1''^2 - 2u_0'' - 2$. Then

$$x_3 - 1 = \frac{v_1''^2}{(c+2)u_1''^2} - (u_1''^2 - 2u_0'' - 2) - \frac{16}{c+2} \; .$$

Concerning y_3, we need to find an expression with respect to v_1'', v_0'' so we introduce both y_1 and y_2 in this way.

$$\begin{array}{rcl} y_3 & = & \lambda(x_1 - x_3) - y_1 \\ y_3 & = & \lambda(x_2 - x_3) - y_2 \\ 2y_3 & = & \lambda(x_1 + x_2 - 2x_3) - (y_1 + y_2) = \lambda(x_1 + x_2 - 2 - 2(x_3 - 1)) - (y_1 + y_2) \end{array} \tag{2.14}$$

with $x_1 + x_2 - 2 = u_1''^2 - 2u_0'' - 2$ and $y_1 + y_2 = -v_1''u_1'' + 2v_0''$ since $y_i = Y_i'' = v_1''X_i'' + v_0''$. We obtain $y_3 = \frac{1}{2}\left(\lambda\left(u_1''^2 - 2u_0'' - 2 - 2(x_3-1)\right) + v_1''u_1'' - 2v_0''\right)$. Since $\lambda = -v_1''/u_1''$ two terms simplify. To sum up,

$$\begin{array}{rcl} x_3 - 1 & = & \frac{v_1''^2}{(c+2)u_1''^2} - (u_1''^2 - 2u_0'' - 2) - \frac{16}{c+2} \\ y_3 & = & \frac{1}{2}\left(\frac{-v_1''}{u_1''}\left(u_1''^2 - 2u_0'' - 2 - 2(x_3-1)\right) + v_1''u_1'' - 2v_0''\right) \\ & = & \frac{v_1''}{u_1''}\left(u_0'' + 1 + (x_3 - 1)\right) - v_0'' \; . \end{array} \tag{2.15}$$

We will now write down x_3, y_3 with respect to $\mathcal{D}(u_1, u_0, v_1, v_0)$. We start by computing the intermediate values λ and $x_1 + x_2 - 2$.

$$\begin{array}{rcl} \lambda & = & \frac{-v_1''}{u_1''} = \frac{2}{\sqrt[8]{b}} \frac{(u_1v_0 - u_0v_1)u_1 - u_0v_0 + 3(u_1v_0 - u_0v_1)\sqrt[4]{b} + 3v_0\sqrt{b} + v_1\sqrt[4]{b^3}}{(u_0 + u_1\sqrt[4]{b} + \sqrt{b})(u_0 - \sqrt{b})} \\ x_1 + x_2 - 2 & = & \frac{\left[32u_0^2 b\right] + \left[-16u_0(u_0^2 + b)\sqrt{b}\right] + \left[-4u_0u_1(u_0^2 - b)\right]\sqrt[4]{b} + \left[4u_1(u_0^2 - b)\right]\sqrt[4]{b^3}}{(u_0 + u_1\sqrt[4]{b} + \sqrt{b})^2(u_0 - \sqrt{b})^2} \end{array} \tag{2.16}$$

$$\frac{\lambda^2}{c+2} = \frac{\lambda^2\sqrt{b}}{a+2\sqrt{b}} = \frac{4\sqrt[4]{b}}{a+2\sqrt{b}}\left(\frac{(u_1v_0-u_0v_1)u_1-u_0v_0+3(u_1v_0-u_0v_1)\sqrt[4]{b}+3v_0\sqrt{b}+v_1\sqrt[4]{b}^3}{(u_0+u_1\sqrt[4]{b}+\sqrt{b})(u_0-\sqrt{b})}\right)^2 \tag{2.17}$$

We now express x_3 in terms of (u_1, u_0, v_1, v_0) and $a, b, \sqrt{b}, \sqrt[4]{b}$.

$$x_3 - 1 = \frac{\lambda^2}{c+2} - (x_1+x_2-2) - \frac{16}{c+2} = \frac{\lambda^2\sqrt{b}}{a+2\sqrt{b}} - (x_1+x_2-2) - \frac{16\sqrt{b}}{a+2\sqrt{b}}\,. \tag{2.18}$$

The denominator of $x_3 - 1$ is

$$z_{x_3-1} = (a+2\sqrt{b})z^2 \tag{2.19}$$

with

$$z = (u_0+u_1\sqrt[4]{b}+\sqrt{b})(u_0-\sqrt{b}) = u_0^2 - b + u_0u_1\sqrt[4]{b} - u_1\sqrt[4]{b}^3$$

and

$$z^2 = \left[(u_0^2-b)^2 - 2u_0u_1^2b\right] + 2u_0u_1(u_0^2-b)\sqrt[4]{b} + u_1^2(u_0^2+b)\sqrt{b} - 2(u_0^2-b)u_1\sqrt[4]{b}^3\,.$$

Computing y_3 is quite complicated because we deal with divisors so we do not have directly the coefficients of the two points. We use this trick:

$$2y_3 = \lambda(x_1+x_2-2x_3) - (y_1+y_2)$$

Since x_1+x_2 was already computed for x_3, getting $(x_1+x_2-2x_3)$ costs only additions. We multiply the numerators of λ and $(x_1+x_2-2x_3)$ which costs $1M_{p^4}$. The denominator is z^3 and since z^2 is already computed, this costs $1M_{p^4}$. We have $y_1+y_2 = -v_1''u_1'' + 2v_0''$. In details,

$$\begin{aligned} y_1+y_2 &= -v_1''u_1'' + 2v_0'' \\ 2y_3 &= \frac{-v_1''}{u_1''}\left(u_1''^2 - 2u_0'' - 2 - 2(x_3-1)\right) + v_1''u_1'' - 2v_0'' \\ y_3 &= \frac{v_1''}{u_1''}\left(u_0'' + 1 + (x_3-1)\right) - v_0''\,. \end{aligned}$$

The numerator of (y_1+y_2) contains products of u_0, u_1, v_0, v_1 previously computed and its denominator is simply z^3. The total cost of y_3 is then $2M_{p^4}$. Finally, computing (x_3, y_3) costs

$$6M_p + 2S_p + 5M_{p^2} + S_{p^4} + 2M_{p^4}\,.$$

Now we show that computing $S_2 = (x_{3,2}, y_{S_2}) = \varphi_{2*}(P_1) + \varphi_{2*}(P_2)$ is free. We notice that

$$\varphi_1(X_j, Y_j) = \varphi_2(-X_j, iY_j).$$

Rewriting this equation in terms of divisors, we deduce that

$$S_2 = \varphi_{2*}(u_1, u_0, v_1, v_0) = \varphi_{1*}(-u_1, u_0, -iv_1, iv_0)\,.$$

We can simply compute S_2 with φ_{1*}:

$$\begin{aligned} x_{S_2} &= x_3((-u_1, u_0, -iv_1, iv_0)) \text{ with} \\ \lambda_{S_2} &= \lambda((-u_1, u_0, -iv_1, iv_0)) = \frac{2i}{\sqrt[8]{b}}\frac{(v_0u_1-v_1u_0)(u_1-3\sqrt[4]{b}) - v_0u_0 + 3\sqrt{b}v_0 - \sqrt[4]{b}^3v_1}{(u_0-\sqrt{b})(u_0-\sqrt[4]{b}u_1+\sqrt{b})} = \pi_{p^2}(\lambda) \end{aligned}$$

and

$$(x_1+x_2)((-u_1, u_0, -iv_1, iv_0)) = 2\frac{u_0^2 + \sqrt{b}u_1^2 - 6\sqrt{b}u_0 + b}{(u_0-\sqrt[4]{b}u_1+\sqrt{b})^2} = \pi_{p^2}(x_1+x_2)\,.$$

We deduce that $x_{S_2} = \pi_{p^2}(x_3)$, $y_{S_2} = \pi_{p^2}(y_3)$ and

$$\varphi_{2*}(P_1) + \varphi_{2*}(P_2) = \pi_{p^2}(\varphi_{1*}(P_1) + \varphi_{1*}(P_2))$$

thus

$$\varphi_{2*}(\mathcal{D}) = \pi_{p^2}(\varphi_{1*}(\mathcal{D}))\,. \tag{2.20}$$

Computing $S_2(x_{S_2}, y_{S_2})$ costs two Frobenius π_{p^2} which are performed with four negations in $\mathbb{F}_{q^2}$.

2.2.1.3 Computing $\hat{\mathcal{I}}_{(2,2)}$ from $E_{1,c} \times E_{1,c}$ to $J_{\mathcal{C}_1}$.

Now, to go back from $S = (x_3, y_3) = \varphi_{1*}(\mathcal{D}) \in E_{1,c}$ to $J_{\mathcal{C}_1''}$. We have two possibilities for the square root of x_3. The generic formula is

$$\varphi_1^*(S) = \sum_{T \in \mathcal{C}_1, \varphi_{1*}(T)=S} T\,.$$

The two points in the pre-image of S under φ_{1*} are $(\sqrt{x_3}, y_3)$, $(-\sqrt{x_3}, y_3) \in \mathcal{C}_1''$. We add these two points to get $\varphi_1^*(S)$ but in $\mathcal{C}_1''$ for the moment. This means that we compute the divisor in $J_{\mathcal{C}_1''}$ of the two points $(\sqrt{x_3}, y_3)$ and $(-\sqrt{x_3}, y_3)$. We obtain

$$\mathcal{D}_3'' = \varphi_1^*(\varphi_{1*}(\mathcal{D})) = \varphi_1^*(S) = (0, -x_3, 0, -y_3) \in J_{\mathcal{C}_1''}$$

with the square roots which simplify and two coefficients equal to zero. With these two coefficients equal to zero, it is quite easy to go back to $J_{\mathcal{C}_1'}$ then $J_{\mathcal{C}_1}$. We obtain

$$\mathcal{D}_3' = \left(2\frac{1-x_3}{1+x_3},\ 1,\ \frac{y_3(-3+x_3)}{2(1+x_3)^2},\ \frac{-y_3}{2(1+x_3)}\right)$$

and finally

$$\mathcal{D}_3 = \left(-2\sqrt[4]{b}\frac{1-x_3}{1+x_3},\ \sqrt{b},\ \frac{\sqrt[8]{b}^3}{2}\frac{y_3(-3+x_3)}{(1+x_3)^2},\ \frac{\sqrt[8]{b}^5}{2}\frac{y_3}{(1+x_3)}\right)\,.$$

To obtain the final result in this isogeny computation, we need to add two divisors on the Jacobian, namely $\mathcal{D}_3 = \varphi_1^*(\varphi_{1*}(\mathcal{D}))$ and $\varphi_2^*(\varphi_{2*}(\mathcal{D}))$. We note that the four coefficients of $\mathcal{D}_3$ are in $\mathbb{F}_{q^4}$ and not $\mathbb{F}_{q^8}$. Indeed it's quite obvious that y_3 is of the form $\sqrt[8]{b}y_3'$ with $y_3' \in \mathbb{F}_{q^4}$. Hence the $\sqrt[8]{b}$ term simplifies with $\sqrt[8]{b}^3$ for the third coefficient and with $\sqrt[8]{b}^5$ for the fourth coefficient of $\mathcal{D}_3$.

First, we show that $\varphi_2^*(\varphi_{2*}(\mathcal{D})) = \pi_{p^2}(\mathcal{D}_3)$. This will help to simplify our computations. A similar computation for $\varphi_2^*(S_2)$ as above with $\varphi_1^*(S)$ gives

$$\begin{aligned}\varphi_2^*(x_{S_2}, y_{S_2}) &= (\sqrt{x_{S_2}}, y_{S_2}) + (-\sqrt{x_{S_2}}, y_{S_2}) \\ &= \left(+2\sqrt[4]{b}\frac{1-x_{S_2}}{1+x_{S_2}},\ \sqrt{b},\ \frac{i\sqrt[8]{b}^3 y_{S_2}(-3+x_{S_2})}{2(1+x_{S_2})^2},\ +\frac{i\sqrt[8]{b}^5 y_{S_2}}{2(1+x_{S_2})}\right)\,.\end{aligned}$$

Since $x_{S_2} = \pi_{p^2}(x_3)$ and $y_{S_2} = \pi_{p^2}(y_3)$, we have

$$\varphi_2^*(x_{S_2}, y_{S_2}) = \left(+2\sqrt[4]{b}\frac{1-\pi_{p^2}(x_3)}{1+\pi_{p^2}(x_3)},\ \sqrt{b},\ \frac{i\sqrt[8]{b}^3\pi_{p^2}(y_3)(-3+\pi_{p^2}(x_3))}{2(1+\pi_{p^2}(x_3))^2},\ +\frac{i\sqrt[8]{b}^5\pi_{p^2}(y_3)}{2(1+\pi_{p^2}(x_3))}\right)\,.$$

We remark that

$$\varphi_2^*(x_{S_2}, y_{S_2}) = \pi_{p^2}(\varphi_1^*(x_3, y_3))\,.$$

Finally,

$$\varphi_2^*\big(\varphi_{2*}(P_1) + \varphi_{2*}(P_2)\big) = \pi_{p^2}\Big(\varphi_1^*\Big(\varphi_{1*}(P_1) + \varphi_{1*}(P_2)\Big)\Big)\,,$$

in other words,

$$\varphi_2^*(\varphi_{2*}(\mathcal{D})) = \pi_{p^2}\big(\varphi_1^*(\varphi_{1*}(\mathcal{D}))\big)\,. \tag{2.21}$$

With our previous notations, we finally have to compute $\mathcal{D}_3 + \pi_{p^2}(\mathcal{D}_3)$ on the Jacobian $J_{\mathcal{C}_1}$. We can use the addition formulas from [CL11]. This ends our isogeny computation.

2.2.2 Isogeny from $J_{\mathcal{C}_2}$ into two elliptic curves $E_{2,c} \times E_{2,-c}$

We consider an analogous family of degree 6 curves. These curves were studied by Duursma and Kiyavash [DK05] and by Gaudry and Schost [GS01]. Their equation is

$$\mathcal{C}_2 : Y^2 = X^6 + aX^3 + b \text{ with } a, b \neq 0 \in \mathbb{F}_q \quad (2.2)\,.$$

The Jacobian of the curve denoted $J_{\mathcal{C}_2}$ is isogenous to the product of the two elliptic curves $E_{2,c} \times E_{2,-c}$ defined over $\mathbb{F}_q[c]$, where

$$\begin{array}{rl} E_{2,c}: & y^2 = (c+2)x^3 + (-3c+30)x^2 + (3c+30)x + (-c+2) \text{ and} \\ E_{2,-c}: & y^2 = (-c+2)x^3 + (3c+30)x^2 + (-3c+30)x + (c+2), \end{array} \tag{2.22}$$

with $c = a/\sqrt{b}$. The construction of the isogeny is similar to the one for $\mathcal{I}_{(2,2)}$ and $J_{\mathcal{C}_1}$. We recall the formulas for maps from $\mathcal{C}_2$ to $E_{2,c}$ and to $E_{2,-c}$. For explicit computations, the reader is referred to Freeman and Satoh [FS11, Prop. 4].

$$\begin{array}{rcl} \varphi_c : \mathcal{C}_2 & \to & E_{2,c} \\ (X,Y) & \mapsto & \left(\left(\frac{X+\sqrt[6]{b}}{X-\sqrt[6]{b}} \right)^2, \frac{8Y}{(X-\sqrt[6]{b})^3} \right) \end{array} \quad \begin{array}{rcl} \varphi_{-c} : \mathcal{C}_2 & \to & E_{2,-c} \\ (X,Y) & \mapsto & \left(\left(\frac{X-\sqrt[6]{b}}{X+\sqrt[6]{b}} \right)^2, \frac{8Y}{(X+\sqrt[6]{b})^3} \right) \end{array} \tag{2.23}$$

This maps induce an isogeny

$$\begin{array}{rcl} \mathcal{I} : J_{\mathcal{C}_2} & \to & E_c \times E_{-c} \\ \mathcal{D} = (P_1, P_2) & \mapsto & \{\varphi_{c*}(P_1) + \varphi_{c*}(P_2),\ \varphi_{-c*}(P_1) + \varphi_{-c*}(P_2)\} \end{array} \tag{2.24}$$

Note that the isogeny constructed using these maps is defined over an extension field of degree 1, 2, 3 or 6. We compute the isogeny from $J_{\mathcal{C}_2}$ to $E_c \times E_{-c}$ as in Sec. 2.2.1 for $J_{\mathcal{C}_1}$.

Let $\mathcal{D} = ((X_1, Y_1), (X_2, Y_2))$ a divisor in the Jacobian $J_{\mathcal{C}_2}$. We denote

$$u_1 = -(X_1 + X_2), u_0 = X_1 X_2,\ \ v_1 = \frac{Y_1 - Y_2}{X_1 - X_2},\ v_0 = \frac{X_1 Y_2 - X_2 Y_1}{X_1 - X_2}\,.$$

We obtain these formulas.

$$\begin{aligned} u_1'' &= -X_1'' - X_2'' = \frac{-2(u_0 - \sqrt[3]{b})}{u_0 + u_1\sqrt[6]{b} + \sqrt[3]{b}} \\ u_0'' &= X_1'' X_2'' = \frac{u_0 - u_1\sqrt[6]{b} + \sqrt[3]{b}}{u_0 + u_1\sqrt[6]{b} + \sqrt[3]{b}} \\ v_1'' &= \frac{Y_1'' - Y_2''}{X_1'' - X_2''} \\ v_0'' &= \frac{X_1'' Y_2'' - X_2'' Y_1''}{X_1'' - X_2''} \end{aligned}$$

With analogy from (2.13), we obtain

$$\begin{aligned} v_1'' &= \frac{4}{\sqrt[6]{b}} \frac{(u_1 v_0 - u_0 v_1)u_1 - u_0 v_0 + 3(u_1 v_0 - u_0 v_1)\sqrt[6]{b} + 3v_0\sqrt[3]{b} + v_1\sqrt{b}}{(u_0 + u_1\sqrt[6]{b} + \sqrt[3]{b})^2}\,, \\ v_0'' &= \frac{-4}{\sqrt[6]{b}} \frac{(u_1 v_0 - u_0 v_1)u_1 - u_0 v_0 + (u_1 v_0 - u_0 v_1)\sqrt[6]{b} - v_0\sqrt[3]{b} - v_1\sqrt{b}}{(u_0 + u_1\sqrt[6]{b} + \sqrt[3]{b})^2}\,. \end{aligned}$$

We deduce the coefficients of the addition law on the curve E_c. We denote by $S = (x_3, y_3)$ the result of $\varphi_{c*}(\mathcal{D}) = \varphi_{c*}(P_1) + \varphi_{c*}(P_2)$. We have $(c+2)x_3 = \left(\lambda^2 - (c+2)(x_1 + x_2) - (-3c+30)\right)$ hence

$$\begin{array}{rcl} x_3 & = & \frac{\lambda^2}{c+2} - (x_1 + x_2) + \frac{3c-30}{c+2} \\ x_3 - 1 & = & \frac{\lambda^2}{c+2} - (x_1 + x_2 - 2) - \frac{36}{c+2}\,. \end{array} \tag{2.25}$$

We start with the coefficient λ.

$$\begin{aligned}\lambda &= \frac{y_1-y_2}{x_1-x_2}=\frac{Y_1''-Y_2''}{X_1''^2-X_2''^2}=\frac{Y_1''-Y_2''}{X_1''-X_2''}\frac{1}{X_1''+X_2''}=\frac{-v_1''}{u_1''}\\ &= \\ &= \frac{2}{\sqrt[6]{b}}\frac{(u_1v_0-u_0v_1)u_1-u_0v_0+3(u_1v_0-u_0v_1)\sqrt[6]{b}+3v_0\sqrt[3]{b}+v_1\sqrt{b}}{(u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})(u_0-\sqrt[3]{b})}\end{aligned}$$

The we compute x_1+x_2 and x_1+x_2-2 in the next step.

$$x_1+x_2 = X_1''^2+X_2''^2=u_1''^2-2u_0''=\frac{2(u_0^2+(u_1^2-6u_0)\sqrt[3]{b}+\sqrt[3]{b}^2)}{(u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})^2}$$

With the expression

$$(u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})^2=u_0^2+2u_0u_1\sqrt[6]{b}+(2u_0+u_1^2)\sqrt[3]{b}+2u_1\sqrt{b}+\sqrt[3]{b}^2$$

we get

$$x_1+x_2-2 = -2\sqrt[6]{b}\frac{u_0u_1+4u_0\sqrt[6]{b}+u_1\sqrt[3]{b}}{(u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})^2}\,.$$

We will need later λ^2.

$$\lambda^2 = \frac{4}{\sqrt[3]{b}}\frac{\left(\left[(u_1v_0-u_0v_1)u_1-u_0v_0+3v_0\sqrt[3]{b}\right]+\left[3(u_1v_0-u_0v_1)+v_1\sqrt[3]{b}\right]\sqrt[6]{b}\right)^2}{(u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})^2(u_0-\sqrt[3]{b})^2} \tag{2.26}$$

The curve equation is

$$E_c: y^2=(c+2)x^3+3(-c+10)x^2+3(c+10)x+(-c+2)\,.$$

This is not a usual reduced Weierstrass equation. In this setting, the addition law on the curve $E_{2,c}$ is

$$x_3-1 = \tfrac{1}{a+2\sqrt{b}}\left[\lambda^2\sqrt{b}-(x_1+x_2-2)(a+2\sqrt{b})-36\sqrt{b}\right]\,. \tag{2.27}$$

We set the common divisor of all terms in x_3-1 to be

$$\begin{aligned}z_3 &= (a+2\sqrt{b}) \quad (u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})^2(u_0-\sqrt[3]{b})^2\\ &= (a+2\sqrt{b}) \quad \Big(\left[u_0^4+u_1^2b\right]+\left[2u_1(u_0^3+b)\right]\sqrt[6]{b}+\left[u_0^2u_1^2+b\right]\sqrt[3]{b}\\ &\qquad +\left[-2u_0^2u_1\right]\sqrt{b}+\left[-2u_0(u_0+u_1^2)\right]\sqrt[3]{b}^2+\left[-2u_0u_1\right]\sqrt[6]{b}^5\Big)\,.\end{aligned} \tag{2.28}$$

We compute $-(x_1+x_2-2)(a+2\sqrt{b})$ with denominator z_3:

$$\begin{aligned}&-(x_1+x_2-2)(a+2\sqrt{b})\\ &= 2\sqrt[6]{b}(a+2\sqrt{b})\frac{u_1(u_0^3+b)+4u_0^3\sqrt[6]{b}-u_0^2u_1\sqrt[3]{b}-8u_0^2\sqrt{b}-u_0u_1\sqrt[3]{b}^2+4u_0\sqrt[6]{b}^5}{(a+2\sqrt{b})(u_0+u_1\sqrt[6]{b}+\sqrt[3]{b})^2(u_0-\sqrt[3]{b})^2}\,.\end{aligned} \tag{2.29}$$

The computation of y_3 is similar to the computation in the preceding section (Sec. 2.2.1, eq. (2.14)). We obtain in the same way that

$$\begin{aligned}x_3-1 &= \frac{v_1''^2}{(c+2)u_1''^2}-(u_1''^2-2u_0''-2)-\frac{36}{c+2}\,,\\ y_3 &= \frac{v_1''}{u_1''}\left(u_0''+1+(x_3-1)\right)-v_0''\,.\end{aligned} \tag{2.30}$$

2.3 Point counting on two families of genus 2 splitting Jacobians

In this section we are interested in computing the Jacobian order over $\mathbb{F}_q$ of the two genus 2 curves $\mathcal{C}_1 : Y^2 = X^5 + aX^3 + bX$, and $\mathcal{C}_2 : Y^2 = X^6 + aX^3 + b$ with $a, b \in \mathbb{F}_q$. We saw in Sec. 2.2 that the Jacobians of the two curves $\mathcal{C}_1$ and $\mathcal{C}_2$ are isogenous to the product of two elliptic curves. We can say from Honda-Tate theorem (Th. 3) that their respective characteristic polynomial of Frobenius endomorphism are equal. In practice, the involved isogenies are defined over an extension $\mathbb{F}_{q^n}$ so the equalities hold for the characteristic polynomials of π_{q^n}. From these equalities we aim to compute the characteristic polynomial of π_q over $\mathcal{C}_1$ and $\mathcal{C}_2$.

2.3.1 Point Counting on $J_{\mathcal{C}_1}(\mathbb{F}_q)$

In this section, we assume that $a \neq 0, b \neq 0$. The case $a = 0$ corresponds to the curves studied by Furukawa, Kawazoe and Takahashi in [FKT04]. The isogeny computed in Sec. 2.2 between the Jacobian $J_{\mathcal{C}_1}$ and the product of the two curves $E_{1,c} \times E_{1,c}$ is defined over $\mathbb{F}_{q^n}$ with $n \mid 8$. We deduce that $\chi_{\mathcal{C}_1,\pi_{q^n}} = \chi_{E_{1,c},\pi_{q^n}}\chi_{E_{1,c},\pi_{q^n}}$ thanks to Honda-Tate theorem (Th. 3). Thus the Jacobian order $\#J_{\mathcal{C}_1}(\mathbb{F}_{q^n}) = \chi_{\mathcal{C}_1,\pi_{q^n}}(1)$ is the product of the two elliptic curve orders. This was already stated for $J_{\mathcal{C}_1}$ by Satoh in [Sat09]. In 1997 Leprévost and Morain also computed this isogeny and obtained results on the Jacobian order in [LM97] in a more general context of character sum computation. They did not investigate the Jacobian order computation in the way we are interested here. We aim to deduce the explicit Jacobian order over $\mathbb{F}_q$ from its order over $\mathbb{F}_{q^8}$. We will present a refinement of Satoh's method. This provides elegant formulas and will permit us to obtain more interesting results on this Jacobian in Sec. 2.5.1.

Satoh used the notation $Z_{J_{\mathcal{C}_1}}(T, \mathbb{F}_q)$ from the notation of the zeta function. We will use the notation of the characteristic polynomial of the Frobenius endomorphism $\chi_{\mathcal{C}_1,\pi_q}$. Let us denote

$$\chi_{\mathcal{C}_1,\pi_q}(T) = T^4 - a_q T^3 + b_q T^2 - q a_q T + q^2 = (T - z_{1,q})(T - z_{2,q})(T - z_{3,q})(T - z_{4,q})\,. \tag{2.31}$$

We assume the same root ordering as in Sec. 1.3.3: $z_{1,q}z_{2,q} = q$ and $z_{3,q}z_{4,q} = q$, then

$$\begin{array}{rcl} a_q & = & \sum_{i=1}^{4} z_{i,q} = z_{1,q} + z_{2,q} + z_{3,q} + z_{4,q} \\ b_q & = & \prod_{1\leqslant i<j\leqslant 4} z_{i,q}z_{j,q} = z_{1,q}z_{2,q} + z_{1,q}z_{3,q} + z_{1,q}z_{4,q} + z_{2,q}z_{3,q} + z_{2,q}z_{4,q} + z_{3,q}z_{4,q} \\ & = & 2q + (z_{1,q} + z_{2,q})(z_{3,q} + z_{4,q})\,. \end{array} \tag{2.32}$$

We know that the polynomial $\chi_{\mathcal{C}_1,\pi_q}$ over an extension of $\mathbb{F}_q$ is given by (1.27):

$$\begin{array}{rcl} \chi_{\mathcal{C}_1,\pi_{q^i}}(T) & = & T^4 - a_{q^i}T^3 + b_{q^i}T^2 - q^i a_{q^i} T + q^{2i} \\ & = & (T - z_{1,q}^i)(T - z_{2,q}^i)(T - z_{3,q}^i)(T - z_{4,q}^i) \end{array}$$

with $z_{j,q}$ the four roots of $\chi_{\mathcal{C}_1,\pi_q}$. Our goal is to find two simple formulas for computing (a_q, b_q) in terms of (a_{q^2}, b_{q^2}) without computing the roots in $\mathbb{C}$, and apply the two formulas recursively. The Newton-Girard formulas Satoh used give $a_{q^2} = (a_q)^2 - 2b_q$ and $b_{q^2} = -(a_{q^4} - (a_{q^2})^2)/2$ but the expression for b_q can be improved. Our computation gives

$$a_{q^2} = (a_q)^2 - 2b_q \tag{2.33}$$

$$b_{q^2} = (b_q)^2 - 4qb_q + 2q^2 - 2qa_{q^2} \tag{2.34}$$

Knowing a_{q^2} and b_{q^2}, we can solve first the second equation (2.34) for b_q then recover a_q using (2.33). We need to know the extension degree of $\mathbb{F}_q$ where the isogeny is defined in order to solve the corresponding system. In each case, two solutions are possible for b_q. This method was developed in [GV12]. We give here a more precise result. In order to reduce the number of possibilities, we will consider the two halves of the isogeny, namely φ_1 and φ_2. We write

$$\chi_{\mathcal{C}_1,\pi_{q^i}}(T) = (T^2 - (z_{1,q}^i + z_{2,q}^i)T + q^i)(T^2 - (z_{3,q}^i + z_{4,q}^i)T + q^i)$$

The two half isogenies φ_1 and φ_2 are defined over an extension field $\mathbb{F}_{q^j}$ with $j \mid 8$. If we denote by t_{q^j} the trace of the Frobenius endomorphism π_{q^j} on $E_{1,c}$ with j such that φ_1, φ_2 are defined over $\mathbb{F}_{q^j}$ then we

have

$$\begin{aligned}\chi_{\mathcal{C}_1,\pi_{q^j}}(T) &= (T^2-(z_{1,q}^j+z_{2,q}^j)T+q^j)(T^2-(z_{3,q}^j+z_{4,q}^j)T+q^j)\\ &= \chi_{E_1,\pi_{q^j}}(T)\\ &= (T^2-t_{q^j}T+q^2)^2\end{aligned}$$

so by identification, we obtain the system (over $\mathbb{Z}$)

$$\begin{cases} z_{1,q}^j+z_{2,q}^j = t_{q^j} \\ z_{2,q}^j+z_{3,q}^j = t_{q^j} \end{cases} \tag{2.35}$$

Since $j \in \{1,2,4,8\}$ we can solve the system (2.35) step by step with

$$(z_{1,q^j}+z_{2,q^j})^2 = z_{1,q^{2j}}+z_{2,q^{2j}}+2q^j$$

which gives

$$z_{1,q^j}+z_{2,q^j} = \pm\sqrt{z_{1,q^{2j}}+z_{2,q^{2j}}+2q^j} \tag{2.36}$$

knowing that the two coefficients a_{q^j}, b_{q^j} are in $\mathbb{Z}$.

The maps φ_1, φ_2 contain the coefficients $\sqrt{b}$, $\sqrt[4]{b}$, $\sqrt[8]{b}$ and φ_2 contains moreover $\sqrt{-1}$. We deduce easily these possibilities:

1. φ_1 and φ_2 are defined over $\mathbb{F}_q$ (2.3.1.1) ;
2. φ_1 is defined over $\mathbb{F}_q$ and φ_2 over $\mathbb{F}_{q^2}$ (2.3.1.2) ;
3. φ_1 and φ_2 are defined over $\mathbb{F}_{q^2}$ (2.3.1.3) ;
4. φ_1 and φ_2 are defined over $\mathbb{F}_{q^4}$ (2.3.1.4) ;
5. φ_1 and φ_2 are defined over $\mathbb{F}_{q^8}$ (2.3.1.5).

We assume that φ_1 gives us informations on z_{1,q^i}, z_{2,q^i} and φ_2 concerns z_{3,q^i}, z_{4,q^i}.

We will need the following isogeny. Let

$$E_1' : y^2/\sqrt[4]{b} = (c+2)x^3-(3c-10)x^2+(3c-10)x-(c+2) \tag{2.37}$$

defined over $\mathbb{F}_q[\sqrt[4]{b}]$ a quadratic twist of E_1 (which is defined over $\mathbb{F}_q[\sqrt[4]{b}]$). The map from $\mathcal{C}_1$ to E_1' is defined over $\mathbb{F}_q[\sqrt[4]{b}]$ and is given by

$$\begin{aligned}\varphi_1' : \mathcal{C}_1 &\to E_1'\\ (X,Y) &\mapsto \left(\left(\tfrac{X+\sqrt[4]{b}}{X-\sqrt[4]{b}}\right)^2, \tfrac{8Y}{(X-\sqrt[4]{b})^3}\right)\end{aligned} \tag{2.38}$$

We removed the term in $\sqrt[8]{b}$ in the Y coordinate.

It is important to determine the extension degree where we can find a square, fourth and eighth root of b and a square root of -1. It is well-known that -1 is a square in $\mathbb{F}_q$ if and only if $q \equiv 1 \bmod 4$ (and is not a square when $q \equiv 3 \bmod 4$). We denote by i a square root of -1, by ζ_8 a square root of i (we have $\zeta_8^4 = -1$) and by ζ_{16} a square root of ζ_8 (we have $\zeta_{16}^8 = -1$). We denote by $\beta_2, \beta_4, \beta_8$ elements in extension fields of $\mathbb{F}_q$ such that $\beta_2^2 = b$, $\beta_4^4 = b$ and $\beta_8^8 = b$. We obtained the following observations.

1. If $q \equiv 3 \bmod 4$ then -1 has no square root in $\mathbb{F}_q$ ($i \notin \mathbb{F}_q$) but has in $\mathbb{F}_{q^2}$ ($i \in \mathbb{F}_{q^2}$) and there exists an element $\zeta_8 \in \mathbb{F}_{q^2}$ such that $\zeta_8^4 = -1$ (because $q^2 \equiv 1 \bmod 8$).
   a) If b is a square then there exists $\beta_2 \in \mathbb{F}_q$ such that $\beta_2^2 = b$ and moreover, one of $\beta_2, -\beta_2$ is also a square in $\mathbb{F}_q$. Then these exists an element $\beta_4 \in \mathbb{F}_q$ such that $\beta_4^4 = b$ (with β_4^2 equals to one of $\beta_2, -\beta_2$). With the same argument, there exists an element β_8 in $\mathbb{F}_q$ such that $\beta_8^8 = b$, in other words, b has an eighth root in $\mathbb{F}_q$. The isogeny is defined over $\mathbb{F}_{q^2}$ with φ_1 defined over $\mathbb{F}_q$ and φ_2 defined over $\mathbb{F}_{q^2}$ (because the square root of -1 is in $\mathbb{F}_{q^2}$ and not in $\mathbb{F}_q$). This case is treated in Sec. 2.3.1.2. The conclusion is that $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1+t_q)(q+1-t_q)$.

b) If b is not a square then $-b$ has a square root in $\mathbb{F}_q$ which we denote by $\bar{\beta}_2$. A square root of b can be writen $\beta_2 = \pm i\bar{\beta}_2$ with $\bar{\beta}_2 \in \mathbb{F}_q$ and $i \in \mathbb{F}_{q^2}$ such that $i^2 = -1$ ($i \notin \mathbb{F}_q$). We see with this notation that a fourth root of b is $\beta_4 = \zeta_8\bar{\beta}_4$ with $\bar{\beta}_4 \in \mathbb{F}_q$ such that $\bar{\beta}_4^4 = -b$. Since $\mathbb{F}_{q^2}$ contains an eighth root of unity ($\zeta_8 \in \mathbb{F}_{q^2}$), we deduce that b has a square and a fourth root in $\mathbb{F}_{q^2}$. We write $\beta_8 = \zeta_{16}\bar{\beta}_8$ with $\bar{\beta}_8 \in \mathbb{F}_q$ such that $\bar{\beta}_8^8 = -b$ and $\zeta_{16} \in \mathbb{F}_{q^2}$ or $\mathbb{F}_{q^4}$ such that $\zeta_{16}^8 = -1$. We only need to know whether ζ_8 is a square or not in $\mathbb{F}_{q^2}$, i.e. whether ζ_{16} is in $\mathbb{F}_{q^2}$ or in $\mathbb{F}_{q^4}$. Since $q \equiv 3 \bmod 4$ we have $q^2 \equiv 9 \bmod 16$ and $\zeta_{16} \notin \mathbb{F}_{q^2}$. We conclude that if $q \equiv 3 \bmod 4$ and b is not a square in $\mathbb{F}_q$ then the square and fourth roots of b are in $\mathbb{F}_{q^2}$ (we can write $\beta_2, \beta_4 \in \mathbb{F}_{q^2}$) and the eighth roots of b are in $\mathbb{F}_{q^4}$ but not in $\mathbb{F}_{q^2}$. The isogeny is defined over $\mathbb{F}_{q^4}$. The Jacobian is isogenous to two quadratic twists over $\mathbb{F}_{q^2}$ thanks to the map (2.38). We can say that $\#J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) = (q^2 + 1 + t_{q^2})^2$ with t_{q^2} the trace of E_1 over $\mathbb{F}_{q^2}$ (and $-t_{q^2}$ is the trace of the quadratic twist E_1' over $\mathbb{F}_{q^2}$). This case is considered in 2.3.1.4.

2. If $q \equiv 1 \bmod 4$ then $i \in \mathbb{F}_q$.

a) If b is an eighth power then the isogeny is defined over $\mathbb{F}_q$ (2.3.1.1). We have $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q + 1 - t_q)^2$.

b) If b is a square and a fourth power but not an eighth power, the isogeny is defined over $\mathbb{F}_{q^2}$ but the Jacobian is isogenous to two quadratic twists of E_1 over $\mathbb{F}_q$ (with the map (2.38)). We have $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q + 1 + t_q)^2$ with t_q the trace of E_1 over $\mathbb{F}_q$.

c) If b is a square but not a fourth power, $\beta_2 \in \mathbb{F}_q$ and $\beta_4 \in \mathbb{F}_{q^2}$. Since $q^2 \equiv 1 \bmod 8$, $\beta_8 \in \mathbb{F}_{q^4}$ but not in $\mathbb{F}_{q^2}$. The isogeny is defined over $\mathbb{F}_{q^4}$. The Jacobian is isogenous over $\mathbb{F}_{q^2}$ to two quadratic twists (see (2.38)) and $\#J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) = (q^2 + 1 + t_{q^2})^2$ (Sec. 2.3.1.4).

d) If b is not a square, $\beta_2 \in \mathbb{F}_{q^2}$, $\beta_4 \notin \mathbb{F}_{q^2}$, $\beta_4 \in \mathbb{F}_{q^4}$ and $\beta_8 \in \mathbb{F}_{q^8}$. This case is solved in Sec. 2.3.1.5. We can start from $\#J_{\mathcal{C}_1}(\mathbb{F}_{q^4}) = (q^4 + 1 + t_{q^4})^2 = (q^4 + 1 - 2q^2 + (t_{q^2})^2)^2$.

2.3.1.1 φ_1 and φ_2 are defined over $\mathbb{F}_q$.

$$J_{\mathcal{C}_1}(\mathbb{F}_q) \quad \overset{\text{of same order}}{\longleftrightarrow} \quad E_{1,c}(\mathbb{F}_q) \times E_{1,c}(\mathbb{F}_q)$$

This happens when both an eighth root of b and a square root of -1 are in $\mathbb{F}_q$. We need in particular $q \equiv 1 \bmod 4$ to have $i \in \mathbb{F}_q$. We denote by t_q the trace of E_1 over $\mathbb{F}_q$. This case is solved directly by Honda-Tate theorem (Th. 3). We have $\chi_{\mathcal{C}_1,\pi_q}(T) = \chi_{E_1,\pi_q}(T) \cdot \chi_{E_1,\pi_q}(T) = (T^2 - t_qT + q)^2$. We conclude that $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q + 1 - t_q)^2$.

2.3.1.2 φ_1 is defined over $\mathbb{F}_q$ and φ_2 over $\mathbb{F}_{q^2}$.

$$\begin{array}{ccc} J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) & \overset{\text{of same order}}{\longleftrightarrow} & E_{1,c}(\mathbb{F}_{q^2}) \times E_{1,c}(\mathbb{F}_{q^2}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_q) & & E_{1,c}(\mathbb{F}_q) \times E_{1,c}(\mathbb{F}_q) \end{array}$$

This happens when b is a square ($\beta_2 \in \mathbb{F}_q$) and $q \equiv 3 \bmod 4$. In this case -1 is not a square in $\mathbb{F}_q$. If β_2 is not a square in $\mathbb{F}_q$ then $-\beta_2$ is a square and there exists $\beta_4 \in \mathbb{F}_q$ such that $\beta_4^2 = -\beta_2$, hence $\beta_4^4 = b$. With the same argument, we can find an eighth root β_8 of b in $\mathbb{F}_q$.

We can see that φ_2 corresponds to φ_1 composed with the quadratic twist map $(x, y) \mapsto (x, iy)$. We see that $T^2 - (z_{1,q} + z_{2,q})T + q = T^2 - t_qT + q$ (because φ_1 is defined over $\mathbb{F}_q$) and we find that $T^2 - (z_{3,q} + z_{4,q})T + q = T^2 + t_qT + q$. We have $\chi_{\mathcal{C}_1,\pi_q}(T) = (T^2 - t_qT + q)(T^2 + t_qT + q)$. We conclude that $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q + 1 - t_q)(q + 1 + t_q)$. If $a = 0$ this is Th. 7 in [FKT04].

2.3.1.3 φ_1 and φ_2 are defined over $\mathbb{F}_{q^2}$.

This happens when $q \equiv 1 \bmod 4$ and b is a square and a fourth power but not an eight power in $\mathbb{F}_q$. In particular the curve E_1 is defined over $\mathbb{F}_q$.

$$\begin{array}{ccc} J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) & \overset{\text{of same}}{\underset{\text{order}}{\longleftrightarrow}} & E_{1,c}(\mathbb{F}_{q^2}) \times E_{1,c}(\mathbb{F}_{q^2}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_q) & & E_{1,c}(\mathbb{F}_q) \times E_{1,c}(\mathbb{F}_q) \end{array}$$

We observe that the quadratic twist $E_1^{'}$ (2.37) is defined over $\mathbb{F}_q$ and the product $E_1^{'} \times E_1^{'}$ is isogenous to $J_{\mathcal{C}_1}$ over $\mathbb{F}_q$ since $\beta_2, \beta_4 \in \mathbb{F}_q$. The trace of the curve $E_1^{'}$ over $\mathbb{F}_q$ is $-t_q$ with t_q the trace of E_1 over $\mathbb{F}_q$. We conclude that $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1+t_q)^2$. If $a = 0$ this is Th. 8 in [FKT04].

2.3.1.4 φ_1 and φ_2 are defined over $\mathbb{F}_{q^4}$.

This happens when

1. $q \equiv 3 \bmod 4$ and b is not a square in $\mathbb{F}_q$;

$$\begin{array}{ccc} J_{\mathcal{C}_1}(\mathbb{F}_{q^4}) & \overset{\text{of same}}{\underset{\text{order}}{\longleftrightarrow}} & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^4}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) & & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^2}) \\ \cup & & \\ J_{\mathcal{C}_1}(\mathbb{F}_q) & & \end{array}$$

2. $q \equiv 1 \bmod 4$ and b is a square but not a fourth power in $\mathbb{F}_q$.

$$\begin{array}{ccc} J_{\mathcal{C}_1}(\mathbb{F}_{q^4}) & \overset{\text{of same}}{\underset{\text{order}}{\longleftrightarrow}} & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^4}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) & & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^2}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_q) & & E_{1,c} \times E_{1,c}(\mathbb{F}_q) \end{array}$$

We proceed in two steps. Firstly we compute the Jacobian order over $\mathbb{F}_{q^2}$ and secondly over $\mathbb{F}_q$. Thanks to the isogeny (2.37) defined over $\mathbb{F}_{q^2}$ with the product of the two quadratic twists, We start with

$$\chi_{\mathcal{C}_1,\pi_{q^2}}(T) = \chi_{E_1',\pi_{q^2}}(T) \cdot \chi_{E_1',\pi_{q^2}}(T) = (T^2 + t_{q^2}T + q^2)^2 \,.$$

We obtain directly the system

$$\begin{cases} z_{1,q^2} + z_{2,q^2} = -t_{q^2} \\ z_{3,q^2} + z_{4,q^2} = -t_{q^2} \end{cases} \tag{2.39}$$

Secondly we apply the formula (2.36), in our case this is $(z_{1,q} + z_{2,q})^2 = z_{1,q^2} + z_{2,q^2} + 2q$. To simplify the computations, we introduce an additional notation. If $E_{1,c}$ is defined over $\mathbb{F}_q$, i.e. if $c \in \mathbb{F}_q$ then from the expression $t_q^2 - 4q = -D\gamma^2$ we can write the two roots $\alpha_q, \overline{\alpha}_q$ of the characteristic polynomial of π_q: $\alpha_q = \frac{t_q + \sqrt{-D}\gamma}{2}$, $\overline{\alpha_q} = \frac{t_q - \sqrt{-D}\gamma}{2}$. Otherwise, c is in $\mathbb{F}_{q^2}$ but not in $\mathbb{F}_q$. We denote

$$(t_{q^2})^2 - 2q^2 = -D\gamma^2 = (t_{q^2} - 2q)(t_{q^2} + 2q)$$

and we decompose it into

$$\begin{array}{rcl} t_{q^2} - 2q & = & -D_1\gamma_1^2 \\ t_{q^2} + 2q & = & D_2\gamma_2^2 \end{array} \tag{2.40}$$

with $D_1, D_2 > 0$ and square-free. We can write $q = \frac{D_1\gamma_1^2 + D_2\gamma_2^2}{4} = \frac{\sqrt{-D_1}\gamma_1 + \sqrt{D_2}\gamma_2}{2} \cdot \frac{-\sqrt{-D_1}\gamma_1 + \sqrt{D_2}\gamma_2}{2} = \alpha_q\overline{\alpha_q}$ with $\alpha_q = \frac{\sqrt{-D_1}\gamma_1 + \sqrt{D_2}\gamma_2}{2}$. We note that $\alpha_q + \overline{\alpha_q} = \sqrt{D_2}\gamma_2$ is not necessarily in $\mathbb{Z}$. $\alpha_q + \overline{\alpha_q} \in \mathbb{Z} \Leftrightarrow D_2 = 1$ (by definition of D_2 which is square-free).

We note that if $E_{1,c}$ is defined over $\mathbb{F}_q$ then its trace is $t_q = \alpha_q + \overline{\alpha_q} \in \mathbb{Z}$ hence $D_2 = 1, \gamma_2 = t_q$. We have also $t_{q^2} + 2q = (t_q)^2$ in this case.

Since

$$\chi_{E_{1,c},\pi_{q^2}}(T) = (T - \alpha_q^2)(T - \overline{\alpha_q}^2) \tag{2.41}$$

We find again that $t_{q^2} = \alpha_q^2 + \overline{\alpha_q^2} = -D_1\gamma_1^2 + D_2\gamma_2^2$ (see eq. (2.40)).

We obtain

$$\left\{ \begin{array}{l} (z_{1,q} + z_{2,q})^2 = z_{1,q^2} + z_{2,q^2} + 2q = -t_{q^2} + 2q = D_1\gamma_1^2 \\ (z_{3,q} + z_{4,q})^2 = z_{3,q^2} + z_{4,q^2} + 2q = -t_{q^2} + 2q = D_1\gamma_1^2 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} z_{1,q} + z_{2,q} = \pm\sqrt{D_1}\gamma_1 \\ z_{3,q} + z_{4,q} = \pm\sqrt{D_1}\gamma_1 \end{array} \right. \tag{2.42}$$

We obtain these three possibilities:

$$\left\{ \begin{array}{l} a_q = 2\sqrt{D_1}\gamma_1 \\ b_q = D_1\gamma_1^2 + 2q = -t_{q^2} + 4q \end{array} \right. , \left\{ \begin{array}{l} a_q = -2\sqrt{D_1}\gamma_1 \\ b_q = D_1\gamma_1^2 + 2q = -t_{q^2} + 4q \end{array} \right. , \left\{ \begin{array}{l} a_q = 0 \\ b_q = -D_1\gamma_1^2 + 2q = t_{q^2} \end{array} \right. \tag{2.43}$$

In the third case, $a_q = 0 \in \mathbb{Z}$ and in each case, b_q is in $\mathbb{Z}$. However, we do not have necessarily $a_q \in \mathbb{Z}$ when $a_q = \pm 2\sqrt{D_1}\gamma_1$. To ensure that we need to have $D_1 = 1$ or $D_1\gamma_1^2 = 0$. If $D_1 = 1$ the Jacobian order will factors in either $(q + 1 \pm \gamma_1)^2$ ($a_q = \pm 2\gamma_1$) or $(q + 1 + \gamma_1)(q + 1 - \gamma_1)$ ($a_q = 0$). If $D_1\gamma_1^2 = 0$ then the curve is supersingular and $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q + 1)^2$.

1. If $q \equiv 3 \bmod 4$ and b is not a square in $\mathbb{F}_q$ then the curves E_1 and E_1' are not defined over $\mathbb{F}_q$ but over $\mathbb{F}_{q^2}$. We need to identify when the Jacobian splits over $\mathbb{F}_q$ whereas the curve $E_{1,c}$ is defined over $\mathbb{F}_{q^2}$.
2. If $q \equiv 1 \bmod 4$ and b is a square but not a fourth power in $\mathbb{F}_q$ then the curve E_1 is defined over $\mathbb{F}_q$ (hence $D_2\gamma_2^2 = (t_q)^2$) and the curve E_1' is defined over $\mathbb{F}_{q^2}$. We have $t_{q^2} - 2q = (t_q)^2 - 4q = -D_1\gamma_1^2$. If $D_1 = 1$ then the curve has j-invariant 0 and $c = 14/9$.

There is no reason to have $\sqrt{D_1}\gamma_1 \in \mathbb{Z}$ for a random curve. However this may happen for example if the curve is supersingular, in which case $D_1\gamma_1 = 0$. We state here a result from [Has97] pointed out to us in another context by B. Smith. Our curve E_1 defined over $\mathbb{F}_{q^2}$ is related to the curve

$$\mathcal{E}_{d,u}^{(2)}/\mathbb{Q}(\sqrt{d}) : y^2 = x^3 + 6(3\sqrt{d}u - 5)x - 8(9\sqrt{d}u - 7),\ j = 2^6\frac{(3\sqrt{d}u - 5)^3}{(\sqrt{d}u - 1)(\sqrt{d}u + 1)^2} \tag{2.44}$$

presented in [Has97, Th. 2.2] (and used in cryptography in [Smi13]), with d a square-free integer different from 1 and u a rational number, through

$$c = \frac{a}{\sqrt{b}} = 2\sqrt{d}u\ . \tag{2.45}$$

We see with this simple change of notations that E_1 is the reduction over $\mathbb{F}_{q^2}$ of $\mathcal{E}_{d,u}^{(2)}(\mathbb{Q}(\sqrt{d}))$. Hasegawa listed in [Has97, Rem. 4.7 (ii)] the degenerate cases, i.e. when the Weil restriction of $\mathcal{E}_{d,u}^{(2)}$ from $\mathbb{Q}(\sqrt{d})$ to $\mathbb{Q}$ (denoted $\mathrm{Res}_{\mathbb{Q}(\sqrt{d})/\mathbb{Q}}\left(\mathcal{E}_{d,u}^{(2)}\right)$ in Hasegawa's paper) is isogenous over $\mathbb{Q}$ to a power of an elliptic curve over $\mathbb{Q}$. This occurs if and only if the curve $\mathcal{E}_{d,u}^{(2)}$ is isogenous over $\mathbb{Q}(\sqrt{d})$ to an elliptic curve defined by an equation with rational coefficients. In this case $\mathcal{E}_{d,u}^{(2)}$ has Complex Multiplication (this is a rare property over $\mathbb{Q}$). The degenerate case we are interested in here is when $(d, u) = (-7, \pm 5/9)$ i.e. with our notations, when $c = \pm 10/9\sqrt{-7}$. We observe that in our context, we are over a quadratic extension $\mathbb{F}_{q^2}$ of a finite field.

1. We assume that b is not a square. This degenerate case corresponds to $c = a/\sqrt{b} = \pm 10/9\sqrt{-7}$ hence $(a, b) = (\pm 10/9v, v^2/(-7))$ with $v \in \mathbb{F}_q^*$ and -7 which is not a square in $\mathbb{F}_q$. In this case which is also treated in [FS11, Prop. 4.6], $j(E_1) = -3375$ and the curve is supersingular. The trace of the curve is $t_{q^2} = -2q$. We already met an elliptic curve of j-invariant $j = -3375$ in Sec. 1.2.10.1. We constructed the curve $E : y^2 = x^3 + a_2x^2 + a_4x$, with a_2, a_4 such that $a_4 = \frac{9\pm 5\sqrt{-7}}{72}a_2^2$. The curve has Complex Multiplication by $\frac{1+\sqrt{-7}}{2}$ (over $\mathbb{C}$). Since -7 is not a square in $\mathbb{F}_q$, the map from the

Complex Multiplication is a distortion map and the curve is supersingular. To conclude, the genus-2 curve

$$\mathcal{C}_1(\mathbb{F}_q) : Y^2 = X^5 + \pm\frac{10}{9}vX^3 + \frac{v}{-7}X$$

with -7 not a square in $\mathbb{F}_q$ and $v \in \mathbb{F}_q^*$ is supersingular and of order $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1)^2$.

2. We assume that b is a square. This time -7 is a square in $\mathbb{F}_q$. The curve E_1 is not supersingular and has Complex Multiplication by $\frac{1+\sqrt{-7}}{2}$. The discriminant of the curve is $D = -7$. We do not have $\sqrt{D_1}\gamma_1$ in $\mathbb{Z}$. We are not in a special case.

In the general case, we have

$$\begin{cases} a_q = 0 \\ b_q = -D_1\gamma_1^2 + 2q = t_{q^2} \end{cases} \tag{2.46}$$

and the Jacobian order is $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = q^2 + 1 + t_{q^2}$.

2.3.1.5 φ_1 and φ_2 are defined over $\mathbb{F}_{q^8}$.

This case corresponds to $q \equiv 1 \bmod 4$ and b is not a square in $\mathbb{F}_q$.

$$\begin{array}{ccc} J_{\mathcal{C}_1}(\mathbb{F}_{q^8}) & \overset{\text{of same order}}{\longleftrightarrow} & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^8}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_{q^4}) & & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^4}) \\ \cup & & \cup \\ J_{\mathcal{C}_1}(\mathbb{F}_{q^2}) & & E_{1,c} \times E_{1,c}(\mathbb{F}_{q^2}) \\ \cup & & \\ J_{\mathcal{C}_1}(\mathbb{F}_q) & & \end{array}$$

First we note that if $c = \pm 10/9\sqrt{-7}$ with -7 not a square in $\mathbb{F}_q$, the curve is supersingular and $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1)^2$. Otherwise we proceed in three steps. We compute the Jacobian order over $\mathbb{F}_{q^4}$, over $\mathbb{F}_{q^2}$ then over $\mathbb{F}_q$. We remark that the Jacobian is isogenous over $\mathbb{F}_{q^4}$ to $E_1' \times E_1'$ through the map (2.38).

We start with

$$\chi_{\mathcal{C}_1,\pi_{q^4}}(T) = \chi_{E_1',\pi_{q^4}}(T) \cdot \chi_{E_1',\pi_{q^4}}(T) = (T^2 + t_{q^4}T + q^4)^2$$

with $t_{q^4} = (t_{q^2})^2 - 2q^2$ the trace of E_1 over $\mathbb{F}_{q^4}$. The trace of E_1' over $\mathbb{F}_{q^4}$ is $-t_{q^4}$. The corresponding system is

$$\begin{cases} z_{1,q^4} + z_{2,q^4} = -t_{q^4} = -(t_{q^2})^2 + 2q^2 \\ z_{3,q^4} + z_{4,q^4} = -t_{q^4} = -(t_{q^2})^2 + 2q^2 \end{cases} \tag{2.47}$$

We continue with

$$\begin{cases} (z_{1,q^2} + z_{2,q^2})^2 = -(t_{q^2})^2 + 4q^2 = D_1D_2\gamma_1^2\gamma_2^2 \\ (z_{3,q^2} + z_{4,q^2})^2 = -(t_{q^2})^2 + 4q^2 = D_1D_2\gamma_1^2\gamma_2^2 \end{cases} \Rightarrow \begin{cases} z_{1,q^2} + z_{2,q^2} = \pm\sqrt{D_1D_2}\gamma_1\gamma_2 \\ z_{3,q^2} + z_{4,q^2} = \pm\sqrt{D_1D_2}\gamma_1\gamma_2 \end{cases} \tag{2.48}$$

We assume that $\sqrt{D_1D_2}\gamma_1\gamma_2$ is not in $\mathbb{Z}$. Unless the curve is supersingular or isogenous to two elliptic curves of j-invariant equals to 1728 (in this case$D_1D_2 = 4$), our assumption holds. Furokawa, Kawwazoe and Takahashi exposed a result on when the curve $Y^2 = X^5 + bX$ is supersingular. When $a = 0$ in our notations, Th. 3 in [FKT04] states that $a_q \equiv 0 \bmod p$ and $b_q \equiv 0 \bmod p$ when $p \not\equiv 1, 3 \bmod 8$.

We assume that the curve is not supersingular (and that $a \neq 0$).

$$\begin{cases} z_{1,q^2} + z_{2,q^2} = \sqrt{D_1D_2}\gamma_1\gamma_2 \\ z_{3,q^2} + z_{4,q^2} = -\sqrt{D_1D_2}\gamma_1\gamma_2 \end{cases} \Rightarrow \begin{cases} z_{1,q} + z_{2,q} = \pm\sqrt{\sqrt{D_1D_2}\gamma_1\gamma_2 + 2q} \\ z_{3,q} + z_{4,q} = \pm\sqrt{-\sqrt{D_1D_2}\gamma_1\gamma_2 + 2q} \end{cases} \tag{2.49}$$

We obtain easily $b_q = 2q \pm \sqrt{\sqrt{D_1D_2}\gamma_1\gamma_2 + 2q}\sqrt{-\sqrt{D_1D_2}\gamma_1\gamma_2 + 2q} = 2q \pm \sqrt{4q^2 - D_1D_2\gamma_1\gamma_2} = 2q \pm t_{q^2}$. We use (2.33) to compute a_q. We obtain $a_q^2 = a_{q^2} + 2b_q = 2b_q$ hence $a_q = \pm\sqrt{2(2q \pm t_{q^2})}$. We

deduce that either $D_2 = 2$ then $a_q = \pm 2\gamma_2$ if $b_q = 2q + t_{q^2} = D_2\gamma_2$ or $D_1 = 2$ then $a_q = \pm 2\gamma_1$ if $b_q = 2q - t_{q^2} = D_1\gamma_1^2$. To conclude,

$$\#J_{\mathcal{C}_1}(\mathbb{F}_q) = q^2 + 1 \pm 2(q+1)\gamma_i + 2\gamma_i^2,\ i \in \{1,2\}\ .$$

We note that in this case, we have either $p = \frac{D_1\gamma_1^2+2\gamma_2^2}{4} = \mathfrak{p}\bar{\mathfrak{p}}$ with $\mathfrak{p} = \frac{\sqrt{D_1}\gamma_1+\sqrt{-2}\gamma_2}{2}$ or $p = \frac{2\gamma_1^2+D_2\gamma_2^2}{4} = \mathfrak{p}\bar{\mathfrak{p}}$ with $\mathfrak{p} = \frac{\sqrt{2}\gamma_1+\sqrt{-D_2}\gamma_2}{2}$. The discriminant $D = D_1D_2$ of the curve does not need to be even. We may have both D_1 and D_2 even, so that D is equal to 4 times an odd integer. In Sec. 2.4.1 we will prove that the curve E_1 defined over $\mathbb{F}_{q^2}$ (with b not a square in $\mathbb{F}_q$) has an endomorphism corresponding to $[p\sqrt{\pm 2}]$. Moreover the curve has another endomorphism coming from the complex multiplication by $\sqrt{D}$. We will see that this complex multiplication decomposes into two endomorphisms, either $[p\sqrt{-2}], [p\sqrt{D_1}]$ or $[p\sqrt{2}], [p\sqrt{-D_2}]$.

Eventually we have the following theorem.

Theorem 7. *Let $\mathcal{C}_1$ be a hyperelliptic curve defined over a finite field $\mathbb{F}_q$ by the equation $\mathcal{C}_1(\mathbb{F}_q) : Y^2 = X^5 + aX^3 + bX$ with $a, b \neq 0 \in \mathbb{F}_q$. Let E_1 be the elliptic curve defined over $\mathbb{F}_q[\sqrt{b}]$ by the equation $y^2 = (c+2)x^3 - (3c-10)x^2 + (3c-10)x - (c+2)$ with $c = a\sqrt{b}$. Let t_q be the trace of $E_1(\mathbb{F}_q)$ if b is a square in $\mathbb{F}_q$ and let t_{q^2} be the trace of $E_1(\mathbb{F}_{q^2})$ if b is not a square in $\mathbb{F}_q$.*

1. *If b is an eighth power in $\mathbb{F}_q$ and moreover $\sqrt{-1} \in \mathbb{F}_q$ then $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1-t_q)^2$ (Sec. 2.3.1.1). Otherwise if $\sqrt{-1} \notin \mathbb{F}_q$ then $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1-t_q)(q+1+t_q)$ (Sec. 2.3.1.2). If b is a fourth power in $\mathbb{F}_q$ but not an eight power then $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = (q+1+t_q)^2$ (Sec. 2.3.1.3). In these three cases the Jacobian splits over $\mathbb{F}_q$.*
2. *If $q \equiv 1 \bmod 4$ and b is a square but not a fourth power in $\mathbb{F}_q$, or if $q \equiv 3 \bmod 4$ and b is not a square in $\mathbb{F}_q$, then (Sec. 2.3.1.4) $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = q^2 + 1 + t_{q^2}$.*
3. *If $q \equiv 1 \bmod 4$ and b is not a square in $\mathbb{F}_q$, then (Sec. 2.3.1.5) $\#J_{\mathcal{C}_1}(\mathbb{F}_q)$ is equal to $q^2 + 1 \pm 2\gamma_i(q+1) + 2\gamma_i^2$ where $\gamma_i \in \mathbb{N}$ is such that either $2q + t_{q^2} = 2\gamma_1^2$ or $2q - t_{q^2} = 2\gamma_2^2$.*

In practice, when Th. 7 presents two order possibilities one can easily discriminate between them by checking whether the scalar multiplication of a random point by the possible orders gives the infinity point.

The first case (Th. 7 (1)) is not interesting for a cryptographic application because the Jacobian order factors trivially over Fq whereas we are interested in almost-prime order jacobians. In the second case (Th. 7 (2)) the Jacobian has the same order as the elliptic curve $E_{1,c}(\mathbb{F}_{q^2})$. We can use either $E_{1,c}$ or $J_{\mathcal{C}_1}$. At the moment, the addition law is more efficient on elliptic curves so it is preferable to use $E_{1,c}(\mathbb{F}_{q^2})$ for a cryptographic application. The last case (Th. 7 (3)) provides an interesting family of genus 2 curves with an efficient point counting method. Moreover in Sec. 2.5 we will explicit two fast endomorphisms on the Jacobian allowing a fast four-dimensional GLV technique for scalar multiplication.

Example 10. *The numerical example in [Sat09] takes $q = p = 509$ and $\mathcal{C}_1(\mathbb{F}_p) : Y^2 = X^5 + 3X^3 + 7X$ ($b = 7$ is not a square). The curve $E_1(\mathbb{F}_{p^4})$ (which corresponds to our quadratic twist $E'_{1,c}(\mathbb{F}_{p^4})$) has a trace $t_{q^4} = 126286$. We deduce that $t_{q^2} = \pm\sqrt{2p^2 - t_{q^4}} = \pm 626$. As $2p + 626 = 2 \cdot 2 \cdot 3 \cdot 137$ is not 2 times a square, we try $2p - 626 = 2 \cdot 14^2 = 392$, so $n = \pm 14$ and $\#J_{\mathcal{C}_1}(\mathbb{F}_p) \in \{245194, 273754\}$. To finish, we have to exclude one of the two possibilities as in [Sat09] by taking a random point P and test whether $[245194]P = \mathcal{O}$ or $[273754]P = \mathcal{O}$. We conclude that $\#J_{\mathcal{C}_1}(\mathbb{F}_p) = 245194$.*

In the two following examples, we take at random a prime $p \equiv 1 \bmod 4$ of 128 bits and start with $a = -3$ and $b = -2$ until b is not a square mod p. Then let $c = a/\sqrt{b}$, $E_{1,c}(\mathbb{F}_{p^2})$ be as in eq. (2.4) and t_{p^2} be its trace. We deduce the Jacobian order and factor it. We repeat this process with subsequent b-values until the Jacobian order is almost prime.

Example 11. $p = \texttt{0x84c4f7a6b9aee8c6b46b34fa2a2bae69} = 1 \bmod 8$. *The 17th test provided $b = -38$, $t_{p^2} = \texttt{0x702461acf6a929e295786868f846ab40} = 0 \bmod 2$, $b_p = 2p - t_{p^2} = 2\gamma_2^2$ as expected with $\gamma_2 = -\texttt{0x8c1fc81b9542ce23}$. We find $\#J_{\mathcal{C}_1}(\mathbb{F}_p) = 2^5 r$ with r a 250-bit prime of cryptographic size close to the 128-bit security level. $r = \texttt{0x226ddb780b2ded62d1d70138d9c7361794679a609fbe5ae85918c88f5b6ea7d}$.*

Example 12. $p = \texttt{0xb081d45d7d08109c2905dd6187f7cbbd} = 5 \mod 8$. *The 17th test provided* $b = -41$, $t_{p^2} = -\texttt{0x11753eaa61f725ff118f63bb131c8b8f2} = 0 \mod 2$, $b_p = 2p + t_{p^2} = 2\gamma_1^2$ *as expected with* $\gamma_1 = -\texttt{0x611e298cc019b06e}$. *We find* $\#J_{\mathcal{C}_1}(\mathbb{F}_p) = 2 \cdot 5 \cdot r$ *with r a 252-bit prime of cryptographic size close to the 128-bit security level:*
$r = \texttt{0xc2b7a2f39d49b6b579d4c15a8440315cd1ccc424df912e6748c949008ebd989}$.

2.3.2 Point Counting on $J_{\mathcal{C}_2}(\mathbb{F}_q)$

We use the same method as for computing $\#J_{\mathcal{C}_1}$. We consider the two isogenies φ_c, φ_{-c} given in Sec. 2.2.2 by (2.23). The two isogenies contain coefficients with $\sqrt{b}$, $\sqrt[3]{b}$, $\sqrt[6]{b}$. If the two isogenies are defined over $\mathbb{F}_{q^j}$, thanks to Honda-Tate theorem (Th. 3) we write

$$\begin{array}{rcl} \chi_{\mathcal{C}_2,\pi_{q^j}}(T) & = & \chi_{E_c,\pi_{q^j}}(T)\chi_{E_{-c},\pi_{q^j}}(T) \\ (T^2 - (z_{1,q^j} + z_{2,q^j})T + q^j)(T^2 - (z_{3,q^j} + z_{4,q^j})T + q^j) & = & (T^2 - t_{q^j,c}T + q^j)(T^2 - t_{q^j,-c}T + q^j) \end{array} \quad (2.50)$$

with $t_{q^j,c}$ the trace of $E_c(\mathbb{F}_{q^j})$ and $t_{q^j,-c}$ the trace of $E_{-c}(\mathbb{F}_{q^j})$. There are four possibilities:

1. φ_c and φ_{-c} are defined over $\mathbb{F}_q$ (2.3.2.1) ;
2. φ_c and φ_{-c} are defined over $\mathbb{F}_{q^2}$ (2.3.2.3) ;
3. φ_c and φ_{-c} are defined over $\mathbb{F}_{q^3}$ (2.3.2.2) ;
4. φ_c and φ_{-c} are defined over $\mathbb{F}_{q^6}$ (2.3.2.4).

We assume that φ_c gives us informations on $z_{1,q^i} + z_{2,q^i}$ and φ_{-c} concerns $z_{3,q^i} + z_{4,q^i}$. The two curves are isogenous over $\mathbb{F}_q[\sqrt{-3}]$. This is stated in [FS11, Proof of Prop. 4.2]. A detailed computation is given in Sec. 2.4.2.1. There exists an isogeny from E_c into E_{-c} of kernel $\{P_3, -P_3, \mathcal{O}\} \subset E_c[3]$ with $P_3 = (3,\ c+2)$ a 3-torsion point on E_c. The isogeny has coefficients with $\sqrt{b}$ and $\sqrt{-3}$. The two curves have the same order (by Honda-Tate theorem) over $\mathbb{F}_q[\sqrt{b}, \sqrt{-3}]$. We deduce that if both b and -3 are squares in $\mathbb{F}_q$ then the curves have the same trace over $\mathbb{F}_q$ and we will be able to simplify our computations with $t_{q,c} = t_{q,-c}$. In any case the curves have the same trace over $\mathbb{F}_{q^2}$ and we have $t_{q^2,c} = t_{q^2,-c}$.

2.3.2.1 φ_c and φ_{-c} are defined over $\mathbb{F}_q$.

$$J_{\mathcal{C}_2}(\mathbb{F}_q) \quad \overset{\text{of same order}}{\longleftrightarrow} \quad E_c(\mathbb{F}_q) \times E_{-c}(\mathbb{F}_q)$$

This case is easy. We use Honda-Tate theorem and obtain $\chi_{\mathcal{C}_2,\pi_q}(T) = (T^2 - t_{q,c}T + q^2)(T^2 - t_{q,-c}T + q^2)$. Moreover if $q \equiv 1 \mod 3$ then $\sqrt{-3} \in \mathbb{F}_q$, $t_{q,c} = t_{q,-c}$ and $\chi_{\mathcal{C}_2,\pi_q}(T) = (T^2 - t_{q,c}T + q^2)^2$. Otherwise ($q \equiv 2 \mod 3$, $\sqrt{-3} \notin \mathbb{F}_q$) $\chi_{\mathcal{C}_2,\pi_q}(T) = (T^2 - t_{q,c}T + q^2)(T^2 + t_{q,c}T + q^2)$. One (single) trace computation is required. To sum-up,

- if $q \equiv 1 \mod 3$ then $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = (q^2 + 1 - t_{q,c})^2$,
- else $q \equiv 2 \mod 3$ and $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = (q^2 + 1 - t_{q,c})(q^2 + 1 + t_{q,c})$.

2.3.2.2 φ_c and φ_{-c} are defined over $\mathbb{F}_{q^3}$.

$$\begin{array}{ccc} J_{\mathcal{C}_2}(\mathbb{F}_{q^3}) & \overset{\text{of same order}}{\longleftrightarrow} & E_c \times E_{-c}(\mathbb{F}_{q^3}) \\ \cup & & \cup \\ J_{\mathcal{C}_2}(\mathbb{F}_q) & & E_c \times E_{-c}(\mathbb{F}_q) \end{array}$$

This case is also quite simple because there are simplifications. If the isogenies are defined over $\mathbb{F}_{q^3}$ then $\sqrt{b} \in \mathbb{F}_q$ and the two curves are defined over $\mathbb{F}_q$. Secondly, $\sqrt[3]{b}$, $\sqrt[6]{b} \in \mathbb{F}_{q^3}$. We can deduce that $q \equiv 1 \mod 3$ since there exist elements in $\mathbb{F}_q$ (e.g. b) that do not have a cube root in $\mathbb{F}_q$. We can also deduce from $q \equiv 1 \mod 3$ that $\sqrt{-3} \in \mathbb{F}_q$ and the curves E_c and E_{-c} are isogenous over $\mathbb{F}_q$. Finally, $t_{c,q} = t_{-c,q}$. The order of the two curves over $\mathbb{F}_{q^3}$ is $q^3 + 1 - t_{c,q^3}$ with $t_{c,q^3} = (t_{c,q})^3 - 3qt_{q,c}$ (see Ex. 1.8). We start with

$$\begin{array}{rcl} \chi_{\mathcal{C}_2,\pi_{q^3}}(T) & = & \chi_{E_c,\pi_{q^3}}(T) \cdot \chi_{E_{-c},\pi_{q^3}}(T) = (T^2 - t_{c,q^3}T + q^3)^2 \\ & = & (T^2 - (z_{1,q}^3 + z_{2,q}^3)T + q^3)(T^2 - (z_{3,q}^3 + z_{4,q}^3)T + q^3) \end{array}$$

and obtain the system

$$\begin{cases} z_{1,q}^3 + z_{2,q}^3 = t_{c,q^3} = (t_{c,q})^3 - 3qt_{c,q} \\ z_{3,q}^3 + z_{4,q}^3 = t_{c,q^3} = t_{c,q}((t_{c,q})^2 - 3q) \end{cases} \Rightarrow \begin{cases} a_{q^3} = 2t_{c,q^3} \\ b_{q^3} = (t_{c,q^3})^2 + 2q^3 \end{cases} \tag{2.51}$$

with a_{q^3} and b_{q^3} the zeta function coefficients of $J_{\mathcal{C}_2}$ over $\mathbb{F}_{q^3}$. We note that $z_{1,q}^3 + z_{2,q}^3 = (z_{1,q} + z_{2,q})^3 - 3q(z_{1,q} + z_{2,q})$ and $z_{3,q}^3 + z_{4,q}^3 = (z_{3,q} + z_{4,q})^3 - 3q(z_{3,q} + z_{4,q})$. After some computations we can obtain this system to solve

$$\begin{cases} a_{q^3} &= (a_q)^3 - 3a_q(b_q - q) \\ b_{q^3} &= (b_q)^3 - 3q^2(b_q) - 3q(a_q)^2(b_q) + 6q^2(a_q)^2 \end{cases}$$

This system is not linear and the two equations are not independent. We will instead consider the intermediate values $z_{1,q} + z_{2,q}$ and $z_{3,q} + z_{4,q}$. From 2.51 we obtain the system

$$\begin{cases} z_{1,q}^3 + z_{2,q}^3 = (z_{1,q} + z_{2,q})^3 - 3q(z_{1,q} + z_{2,q}) &= (t_{c,q})^3 - 3qt_{c,q} \\ z_{3,q}^3 + z_{4,q}^3 = (z_{3,q} + z_{4,q})^3 - 3q(z_{3,q} + z_{4,q}) &= (t_{c,q})^3 - 3qt_{c,q} \end{cases} \tag{2.52}$$

An obvious solution is $z_{1,q} + z_{2,q} = z_{3,q} + z_{4,q} = t_{c,q}$. This happens when the isogenies φ_c, φ_{-c} are defined over $\mathbb{F}_q$, i.e. b is a square and a cube. We assumed that this is not the case. The two other solutions are

$$\begin{cases} z_{1,q} + z_{2,q} = \left(-t_{c,q} \pm \sqrt{3(4q - (t_{c,q})^2)}\right)/2 \\ z_{3,q} + z_{4,q} = \left(-t_{c,q} \pm \sqrt{3(4q - (t_{c,q})^2)}\right)/2 \end{cases} \tag{2.53}$$

We obtain these three solutions.

$$\begin{cases} a_q = -t_{c,q} \\ b_q = -q + (t_{c,q})^2 \end{cases} \tag{2.54}$$

$$\begin{cases} a_q = -t_{c,q} + \sqrt{3(4q - (t_{c,q})^2)} \\ b_q = 2q + \frac{1}{4}\left((t_{c,q})^2 + 3(4q - (t_{c,q})^2) + t_{c,q}\sqrt{3(4q - (t_{c,q})^2)}\right) \end{cases} \tag{2.55}$$

$$\begin{cases} a_q = -t_{c,q} - \sqrt{3(4q - (t_{c,q})^2)} \\ b_q = 2q + \frac{1}{4}\left((t_{c,q})^2 + 3(4q - (t_{c,q})^2) - t_{c,q}\sqrt{3(4q - (t_{c,q})^2)}\right) \end{cases} \tag{2.56}$$

With the first solution we obtain $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 - q + 1 + (1 + q + t_{c,q})t_{c,q}$. Note that $\#E_c(\mathbb{F}_{q^3}) = \#E_{-c}(\mathbb{F}_{q^3}) = q^3 + 1 - t_{c,q^3} = (q + 1 - t_{c,q})(q^2 - q + 1 + (1 + q + t_{c,q})t_{c,q}) = \#E_c(\mathbb{F}_q)\#J_{\mathcal{C}_2}(\mathbb{F}_q)$.

The first solution has its coefficients in $\mathbb{Z}$. The two other solutions are special cases requiring that $4q - (t_{c,q})^2$ is of the form $3\gamma^2$ in order to have $a_q, b_q \in \mathbb{Z}$. We then obtain

$$\begin{cases} a_q = -t_{c,q} + 3\gamma \\ b_q = 2q + (-t_{c,q} + 3\gamma)^2/4 \end{cases} \Rightarrow \#J_{\mathcal{C}_2}(\mathbb{F}_q) = (q + 1 - (-t_{c,q} + 3\gamma)/2)^2 \tag{2.57}$$

$$\begin{cases} a_q = -t_{c,q} - 3\gamma \\ b_q = 2q + (-t_{c,q} - 3\gamma)^2/4 \end{cases} \Rightarrow \#J_{\mathcal{C}_2}(\mathbb{F}_q) = (q + 1 - (-t_{c,q} - 3\gamma)/2)^2 \tag{2.58}$$

We will identify exactly when this happens. Let E_c' and E_{-c}' be two isogenous elliptic curves defined over $\mathbb{F}_q$ of trace $(-t_{c,q} + 3\gamma)/2$. These curves are isogenous to $J_{\mathcal{C}_2}(\mathbb{F}_q)$. They are also isogenous over $\mathbb{F}_{q^3}$ to E_c and E_{-c}.

$$\begin{array}{ccccc} \overset{\text{of same order}}{\longleftrightarrow} J_{\mathcal{C}_2}(\mathbb{F}_{q^3}) & \overset{\text{of same order}}{\longleftrightarrow} & E_c'(\mathbb{F}_{q^3}) \times E_{-c}'(\mathbb{F}_{q^3}) & \overset{\text{of same order}}{\longleftrightarrow} & E_c(\mathbb{F}_{q^3}) \times E_{-c}(\mathbb{F}_{q^3}) \\ \cup & & \cup & & \cup \\ J_{\mathcal{C}_2}(\mathbb{F}_q) & \xrightarrow[\phi_c \circ \varphi_c, \phi_{-c} \circ \varphi_{-c}]{\text{isogeny}} & E_c' \times E_{-c}'(\mathbb{F}_q) & & E_c \times E_{-c}(\mathbb{F}_q) \end{array}$$

For a second time we will use the results of Hasegawa stated in [Has97]. We consider the elliptic curve

$$\mathcal{E}_{d,u}^{(3)}(\mathbb{Q}(\sqrt{d})) : y^2 = x^3 - 3(4\sqrt{d}u + 5)x + 2(2du^2 + 14\sqrt{d}u + 11),\ j = -2^4 3^3 \frac{(4\sqrt{d}u + 5)^3}{(\sqrt{d}u - 1)^3(\sqrt{d}u + 1)} \tag{2.59}$$

from [Has97, §2. p. 349]. We see that with the change of notations $c = 2\sqrt{d}u$ (or $a = 2u, b = 1/d$) we obtain exactly the reduced form of E_c. Then Remark 4.7 states the result we are interested in. The curve $\mathcal{E}^{(3)}_{d,u}$ is isogenous over $\mathbb{Q}(\sqrt{d})$ to an elliptic curve defined by an equation with rational coefficients when $(d,u) = (-3,0)$ and $(d,u) = (-11, \pm 1/4)$. This corresponds to $c = 0$ and more precisely $(a,b) = (0,-1/3)$. The second possibility is $c = \pm\sqrt{-11}/2$, $(a,b) = (\pm 1/2v, v^2/(-11))$ (also stated in [FS11, Prop. 4.8]). The j-invariant of the curve is $j(E_1) = -32768$. We already met such a curve when computing endomorphisms obtained from a degree 3 isogeny in Sec. 1.2.10.2, item 3. This elliptic curve has Complex Multiplication by $\left[\frac{1+\sqrt{-11}}{2}\right]$ hence $D = 11$ and we are not in a special case. In the next section the curve will be supersingular in this case.

2.3.2.3 φ_c and φ_{-c} are defined over $\mathbb{F}_{q^2}$.

$$\begin{array}{ccc} J_{\mathcal{C}_2}(\mathbb{F}_{q^2}) & \xleftrightarrow[\text{order}]{\text{of same}} & E_c \times E_{-c}(\mathbb{F}_{q^2}) \\ \cup & & \\ J_{\mathcal{C}_2}(\mathbb{F}_q) & & \end{array}$$

In this case, the two elliptic curves are isogenous and have the same trace over $\mathbb{F}_{q^2}$ and b is not a square. We start with

$$\chi_{\mathcal{C}_2,\pi_{q^2}}(T) = \chi_{E_c,\pi_{q^2}}(T) \cdot \chi_{E_{-c},\pi_{q^2}}(T) = (T^2 - t_{c,q^2}T + q^2)^2$$

and

$$\left\{ \begin{array}{l} z_{1,q^2} + z_{2,q^2} = t_{c,q^2} \\ z_{3,q^2} + z_{4,q^2} = t_{c,q^2} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} a_{q^2} = 2t_{c,q^2} \\ b_{q^2} = t^2_{c,q^2} + 2q^2 \end{array} \right. \tag{2.60}$$

We solve

$$\left\{ \begin{array}{l} (z_{1,q} + z_{2,q})^2 = t_{c,q^2} + 2q \\ (z_{3,q} + z_{4,q})^2 = t_{c,q^2} + 2q \end{array} \right. \tag{2.61}$$

We write $(t_{c,q^2})^2 - 4q^2 = (t_{c,q^2} - 2q)(t_{c,q^2} + 2q) = -D_3\gamma_3^2 D_1 \gamma_1^2$. We obtain

$$\left\{ \begin{array}{l} z_{1,q} + z_{2,q} = \pm\sqrt{D_1}\gamma_1 \\ z_{3,q} + z_{4,q} = \pm\sqrt{D_1}\gamma_1 \end{array} \right. \tag{2.62}$$

Either we face a special case with $D_1 = 1$ (we recall that if the curve is actually defined over $\mathbb{F}_q$ then $D_1 = 1$ and $t_{c,q^2} + 2q = (t_{c,q})^2$), or this is a normal case ($D_1 \neq 1$) and we get

$$\left\{ \begin{array}{l} a_q = 0, \\ b_q = -D_1\gamma_1^2 + 2q = -t_{c,q^2} \end{array} \right. \tag{2.63}$$

and

$$\chi_{\mathcal{C}_2,\pi_q}(T) = T^4 - t_{c,q^2}T^2 + q^2\,. \tag{2.64}$$

The Jacobian $J_{\mathcal{C}_2}(\mathbb{F}_q)$ has the same order as the elliptic curve $E_c(\mathbb{F}_{q^2})$.

2.3.2.4 φ_c and φ_{-c} are defined over $\mathbb{F}_{q^6}$.

$$\begin{array}{ccc} JCs(\mathbb{F}_{q^6}) & \xrightarrow[\varphi_c,\varphi_{-c}]{\text{isogeny}} & E_c \times E_{-c}(\mathbb{F}_{q^6}) \\ \cup & & \cup \\ J_{\mathcal{C}_2}(\mathbb{F}_{q^2}) & & E_c \times E_{-c}(\mathbb{F}_{q^2}) \\ \cup & & \\ J_{\mathcal{C}_2}(\mathbb{F}_q) & & \end{array}$$

We proceed in two steps. First we apply the formulas obtained when the isogeny is defined over $\mathbb{F}_{q^3}$. We use the notation $(t_{c,q^2})^2 - 4q^2 = (t_{c,q^2} - 2q)(t_{c,q^2} + 2q) = -D_3\gamma_3^2 D_1\gamma_1^2$. We obtain

$$\left\{ \begin{array}{l} z_{1,q^2} + z_{2,q^2} = (-t_{c,q^2} + \sqrt{3D_1D_3}\gamma_1\gamma_3)/2 \\ z_{3,q^2} + z_{4,q^2} = (-t_{c,q^2} - \sqrt{3D_1D_3}\gamma_1\gamma_3)/2 \end{array} \right. \tag{2.65}$$

and deduce that

$$\begin{cases} a_{q^2} = -t_{c,q^2} \\ b_{q^2} = -q^2 + (t_{c,q^2})^2 \end{cases} \tag{2.66}$$

The last step starts from

$$\begin{cases} (z_{1,q}+z_{2,q})^2 &= (-t_{c,q^2} + \sqrt{3D_1D_3}\gamma_1\gamma_3)/2 + 2q = (3D_3\gamma_3^2 + 2\sqrt{3D_1D_3}\gamma_1\gamma_3 + D_1\gamma_1^2)/4 \\ &= (\sqrt{3D_3}\gamma_3 + \sqrt{D_1}\gamma_1)^2/4 \\ (z_{3,q}+z_{4,q})^2 &= (-t_{c,q^2} - \sqrt{3D_1D_3}\gamma_1\gamma_3)/2 + 2q = (3D_3\gamma_3^2 - 2\sqrt{3D_1D_3}\gamma_1\gamma_3 + D_1\gamma_1^2)/4 \\ &= (-\sqrt{3D_3}\gamma_3 + \sqrt{D_1}\gamma_1)^2/4 \end{cases} \tag{2.67}$$

We deduce that

$$\begin{cases} z_{1,q}+z_{2,q} = \pm(\sqrt{3D_3}\gamma_3 + \sqrt{D_1}\gamma_1)/2 \\ z_{3,q}+z_{4,q} = \pm(\sqrt{3D_3}\gamma_3 - \sqrt{D_1}\gamma_1)/2 \end{cases} \tag{2.68}$$

To obtain integer values for a_q, b_q we have two choices. We can force D_1 to be equal to 1 then get

$$\begin{cases} a_q = \pm\sqrt{D_1}\gamma_1 = \pm\gamma_1 \\ b_q = t_{c,q^2} + q = \gamma_1^2 - q \end{cases} \tag{2.69}$$

$$\Rightarrow \begin{array}{l} \chi_{\mathcal{C}_2,\pi_q}(T) = T^4 \mp \gamma_1 T^3 + (t_{c,q^2}+q)T^2 \mp q\gamma_1 T + q^2, \\ \#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 - q + 1 \mp (1+q)\gamma_1 + \gamma_1^2 \end{array} \tag{2.70}$$

or we can set $D_3 = 3$ to get

$$\begin{cases} a_q = \pm 3\gamma_3 \\ b_q = 3q - t_{c,q^2} = q + 3\gamma_3^2 \end{cases} \tag{2.71}$$

$$\Rightarrow \begin{array}{l} \chi_{\mathcal{C}_2,\pi_q}(T) = T^4 \mp 3\gamma_3 T^3 + (3q - t_{c,q^2})T^2 \mp 3q\gamma_3 T + q^2, \\ \#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 + q + 1 \mp 3\gamma_3(q+1) + 3\gamma_3^2. \end{array} \tag{2.72}$$

We note that in this case, we have $p = \mathfrak{p}\overline{\mathfrak{p}} = \frac{D_1\gamma_1^2 + 3\gamma_3^2}{4}$ with $\mathfrak{p} = \frac{\sqrt{D_1}\gamma_1 + \sqrt{-3}\gamma_3}{2}$ and the discriminant of the curve is a multiple of 3, $D = 3D_1$. In Sec. 2.4.2.1 we will prove that the curve E_c defined over $\mathbb{F}_{q^2}$ (with b not a square in $\mathbb{F}_q$) has an endomorphism corresponding to $[p\sqrt{-3}]$.

We obtain the following theorem:

Theorem 8. *Let $\mathcal{C}_2$ be a hyperelliptic curve defined over a finite field $\mathbb{F}_q$ by the equation $\mathcal{C}_2(\mathbb{F}_q) : Y^2 = X^6 + aX^3 + b$ with $a, b \neq 0 \in \mathbb{F}_q$. Let E_c and E_{-c} be the elliptic curves defined over $\mathbb{F}_q[\sqrt{b}]$ by the equation $y^2 = (c+2)x^3 - (3c-30)x^2 + (3c+30)x + (-c+2)$ and assume that the curves are not supersingular. Let t_{q^2} be the trace of $E_c(\mathbb{F}_{q^2})$ and if b is a square then let t_q be the trace of $E_c(\mathbb{F}_q)$.*

1. *If b is a sixth power then $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = (q+1-t_q)^2$ if $\sqrt{-3} \in \mathbb{F}_q$ and $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = (q+1-t_q)(q+1+t_q)$ if $\sqrt{-3} \notin \mathbb{F}_q$.*
2. *If b is a square but not a third power then $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 - q + 1 + (1+q+t_q)t_q$.*
3. *If b is a third power but not a square then $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 + 1 - t_{q^2}$.*
4. *If b is neither a cube nor a square then there exists $n \in \mathbb{N}$ such that $2q - t_{q^2} = 3n^2$ and $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 + q + 1 + (q+1+n)3n$ or $\#J_{\mathcal{C}_2}(\mathbb{F}_q) = q^2 + q + 1 - (q+1-n)3n$.*

This explicit point counting is used in Sec. 2.6 to construct pairing-friendly genus 2 curves of the form $\mathcal{C}_2$ over a prime field $\mathbb{F}_q$.

Example 13. *We consider the 127-bit Mersenne prime $p = 2^{127} - 1$ which allows efficient implementation of the modular arithmetic operations required in cryptography. Looking for a curve $\mathcal{C}_2$ over $\mathbb{F}_p$ with small parameters a and b and suitable for a cryptographic use, we find easily $\mathcal{C}_2(\mathbb{F}_p) : Y^2 = X^6 - 3X^3 - 92$ with $b = -92$ which is neither a square nor a cube. Let $\mathbb{F}_{p^2} = \mathbb{F}_p[X]/(X^2+1) = \mathbb{F}_p[i]$, $c = a/\sqrt{b} \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$ and $E_c(\mathbb{F}_{p^2}) : Y^2 + X^3 + 3(2c-5)X + c^2 - 14c + 22$. A few second computation gives us*
$t_{p^2} =$ `0x6089c0341e5414a24bef1a1a93c54fd2`
and $2p - t_{p^2} = 3\gamma_3^2$ as expected with $\gamma_3 = \pm$`0x74a69cde5282dbb6`. Hence $\#J_{\mathcal{C}_2}(\mathbb{F}_p) = p^2 + p + 1 + 3\gamma_3(p+1) + 3\gamma_3^2$. Using few random points on the Jacobian, we find $\gamma_3 < 0$ and that $\#J_{\mathcal{C}_2}(\mathbb{F}_p)$ has a 250-bit prime factor: $r =$* `0x25ed097b425ed0974c75619931ea7f1271757b237c3ff3c5c00a037e7906557` *and provides a security level close to 128-bits.*

2.4 Endomorphisms on the two families of elliptic curves and application to scalar multiplication

2.4.1 Endomorphisms on $E_{1,c}$

In this section we compute explicitly a fast endomorphism on the curve $E_{1,c}$ defined over $\mathbb{F}_{q^2}$, presented in Sec. 2.2.1. This endomorphism is different than the Complex Multiplication. We then construct a curve $E_{1,c}$ over $\mathbb{F}_{q^2}$ with an efficient Complex Multiplication (we choose a small discriminant). These two distinct, fast endomorphisms can be used for a four-dimensional GLV scalar multiplication. These properties on such curves $E_{1,c}$ where independently developed in [Smi13] from a different point of view and for a different application.

We introduce the elliptic curve in reduced form

$$E_{1,c} : y^2 = x^3 + 27(3c-10)x + 108(14-9c) \tag{2.73}$$

defined over $\mathbb{F}_{p^2}$, whose j-invariant is

$$j(E_{1,c}) = 2^6 \frac{(3c-10)^3}{(c-2)(c+2)^2} .$$

We assume that $c \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$ and $c^2 \in \mathbb{F}_p$. We denote $a_{4,c} = -27(3c-10)$ and $a_{6,c} = 108(14-9c)$. We will explain how to compute an endomorphism ϕ_2 such that $\phi_2^2 \pm 2 = 0$ on $E_{1,c}(\mathbb{F}_{p^2})$ in Sec. 2.4.1.1. If the discriminant D of the curve is small enough, we will explain in Sec. 2.4.1.2 how to compute a second endomorphism.

This curve is exactly the curve $\tilde{\mathcal{E}}_{2,\Delta,s}/\mathbb{Q}(\sqrt{\Delta}) : y^2 = x^3 - 6(5-3s\sqrt{\Delta})x + 8(7-9s\sqrt{\Delta})$ in [Smi13, §5] with a change of variables of the form $c = 2s\sqrt{D}$. The author in [Smi13] proposes this curve for fast 2-dimensional GLV. Since a Complex Multiplication by a small discriminant is not imposed, a prime number p providing fast arithmetic in $\mathbb{F}_p$ (with fast modular reduction) can be used, such as $p = 2^{127}-1$ or $p = 2^{255}-19$. In this thesis, we do not choose p a priori, we choose a small discriminant to get a second endomorphism. The two methods may provide similar efficiency. More work is needed to benchmark the two methods.

2.4.1.1 First Endomorphism from Vélu's formulas

We aim to compute a 2-isogeny on $E_{1,c}$. Note that we can write

$$E_{1,c} : y^2 = (x-12)(x^2+12x+81c-126). \tag{2.74}$$

Hence there always exists a 2-torsion point $P_2 = (12, 0)$ on $E_{1,c}(\mathbb{F}_{p^2})$. We apply Velu's formulas to compute the isogeny whose kernel is generated by P_2. We obtain an isogeny from $E_{1,c}$ into $E_b : y^2 = x^3 + b_4x + b_6$ with $b_4 = -2^2 \cdot 27(3c+10)$, $b_6 = -2^2 \cdot 108(14+9c)$. We observe that E_b has j-invariant

$$j(E_b) = 2^6 \frac{(3c+10)^3}{(c+2)(c-2)^2}$$

and is isomorphic over $\mathbb{F}_{p^2}$ to the curve whose equation is

$$E_{1,-c} : y^2 = x^3 + 27(-3c-10)x + 108(14+9c) \tag{2.75}$$

through $(x_b, y_b) \mapsto (x_b/(-2), y_b/(-2\sqrt{-2}))$. Note that $\sqrt{-2} \in \mathbb{F}_{p^2}$ and thus this isomorphism is defined over $\mathbb{F}_{p^2}$. We define the isogeny

$$\begin{array}{rcl} \mathcal{I}_2 : E_{1,c} & \to & E_{1,-c} \\ (x,y) & \mapsto & \left(\frac{-x}{2} + \frac{162+81c}{-2(x-12)}, \frac{-y}{2\sqrt{-2}}\left(1 - \frac{162+81c}{(x-12)^2}\right)\right) \\ & & = \left(\frac{x^2-12x+162+81c}{-2(x-12)}, y\frac{x^2-24x-18-81c}{-2\sqrt{-2}(x-12)^2}\right) \end{array} \tag{2.76}$$

We show that we can use this isogeny to get an efficiently computable endomorphism on $E_{1,c}$. Observe that since $c \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$ and $c^2 \in \mathbb{F}_p$, we have that

$$\pi_p(c) = c^p = -c,\ \pi_p(j(E_{1,c})) = j(E_{1,-c}) \tag{2.77}$$

hence the curves $E_{1,c}$ and $E_{1,-c}$ (2.75) are *isogenous* over $\mathbb{F}_{p^2}$ via the Frobenius map π_p. They are not isomorphic, because they do not have the same j-invariant.

To sum up, we obtain an efficiently computable endomorphism ϕ_2 by composing $\pi_p \circ \mathcal{I}_2$ in this way:

$$\begin{array}{rcl} \phi_2 : E_{1,c} & \to & E_{1,c} \\ (x,y) & \mapsto & \left(\dfrac{-x^p}{2} - \dfrac{162-81c}{2(x^p-12)}, \dfrac{-y^p}{2\sqrt{-2}^p}\left(1 - \dfrac{162-81c}{(x^p-12)^2}\right)\right) \\ & = & \left(\dfrac{x^{2p}-12x^p+162-81c}{-2(x^p-12)}, y^p \dfrac{x^{2p}-24x^p-18+81c}{-2\sqrt{-2}^p(x^p-12)^2}\right) \end{array} \tag{2.78}$$

If we compute formally ϕ_2^2 for points defined over $\mathbb{F}_{p^2}$ then we obtain exactly the formulas to compute $\pi_{p^2} \circ [-2]$ on $E_{1,c}$ if $\sqrt{-2} \in \mathbb{F}_p$, $\pi_{p^2} \circ [2]$ if $\sqrt{-2} \notin \mathbb{F}_p$. This difference occurs because a term $\sqrt{-2}\sqrt{-2}^p$ appears in the formula. If $p \equiv 1,3 \bmod 8$, $\sqrt{-2}^p = \sqrt{-2}$ and if $p \equiv 5,7 \bmod 8$, $\sqrt{-2}^p = -\sqrt{-2}$. Hence ϕ_2 restricted to points defined over $\mathbb{F}_{p^2}$ verifies the equation

$$\begin{array}{l} \phi_2^2 + 2 = 0 \text{ over } \mathbb{F}_{p^2} \text{ if } p \equiv 1,3 \mod 8, \\ \phi_2^2 - 2 = 0 \text{ over } \mathbb{F}_{p^2} \text{ if } p \equiv 5,7 \mod 8\,. \end{array} \tag{2.79}$$

We note that the above construction does not come as a surprise. Since $2\mathrm{End}(J_{\mathcal{C}_1}) \subseteq \mathrm{End}(E_{1,c} \times E_{1,c})$ and since the Jacobian $J_{\mathcal{C}_1}$ is equipped with a p-power Frobenius endomorphism, we deduce that there are endomorphisms with inseparability degree p on the elliptic curve $E_{1,c}$. Our construction is simply an efficient method to compute such an endomorphism.

2.4.1.2 Second endomorphism from complex multiplication

In the following, we suppose that the complex multiplication discriminant D of the curve $E_{1,c}$ is small. A natural way to obtain an efficiently computable endomorphism is to take ϕ_D the generator for the endomorphism ring (i.e. $\sqrt{-D}$). It was shown in [GV12, proof of Th. 1 (4.) §2.2] showed that $D = 2D'$, for some integer D'. Let t_{p^2} be the trace of $E_{1,c}(\mathbb{F}_{p^2})$. The equation of the complex multiplication is then

$$(t_{p^2})^2 - 4p^2 = -2D'\gamma^2, \tag{2.80}$$

for some $\gamma \in \mathbb{Z}$. We prove that there is an endomorphism on $E_{1,c}$ whose degree of separability is D'. In order to do that, we will need to compute first the general equation of ϕ_2 (given by (2.78)).

Lemma 4. *There are integers m and n such that if $p \equiv 1,3 \pmod 8$, then*

$$t_{p^2} + 2p = D'm^2 \text{ and } t_{p^2} - 2p = -2n^2 \tag{2.81}$$

and if $p \equiv 5,7 \pmod 8$, then

$$t_{p^2} + 2p = 2n^2 \text{ and } t_{p^2} - 2p = -D'm^2. \tag{2.82}$$

Moreover, the characteristic equation of ϕ_2 is

$$\phi_2^2 - 2n\,\phi_2 + 2p\,\mathrm{Id} = 0\,. \tag{2.83}$$

The endomorphism ϕ_2 corresponds to the root $\frac{2n-m\sqrt{-D}}{2}$ if $p \equiv 1,3 \bmod 8$ and to the root $\frac{2n+m\sqrt{-D}}{2}$ if $p \equiv 5,7 \bmod 8$.

Proof. We have that $\mathrm{Tr}(\phi_2^2) - \mathrm{Tr}^2(\phi_2) + 2\deg(\phi_2) = 0$. We know that $\deg(\phi_2) = 2p$ because $\phi_2 = \pi_p \circ \mathcal{I}_2$ and $\deg(\pi_p) = p, \deg(\mathcal{I}_2) = 2$, so $\mathrm{Tr}^2(\phi_2) = \mathrm{Tr}(\phi_2^2) + 4p$. Now, if $p \equiv 1,3 \bmod 8$, $\mathrm{Tr}(\phi_2^2) = \mathrm{Tr}(\pi_{p^2} \circ [-2]) = -2t_{p^2}$ and we get $\mathrm{Tr}^2(\phi_2) = -2t_{p^2} + 4p = -2(t_{p^2} - 2p)$. We may thus write $t_{p^2} - 2p = -2n^2$,

for some integer n. If $p \equiv 5,7 \bmod 8$, $\mathrm{Tr}(\phi_2^2) = \mathrm{Tr}(\pi_{p^2} \circ [2]) = 2t_{p^2}$ and we get $\mathrm{Tr}^2(\phi_2) = 2t_{p^2} + 4p = 2(t_{p^2} + 2p)$. Hence $t_{p^2} + 2p = 2n^2$ again. Using the complex multiplication equation (2.80), we have that there is an integer m such that $t_{p^2} + 2p = D'm^2$, if $p \equiv 1,3 \pmod 8$ and $t_{p^2} - 2p = -D'm^2$, if $p \equiv 5,7 \pmod 8$. As a consequence, $p = \frac{2n^2 + D'm^2}{4}$; $t_{p^2} = \frac{-2n^2 + D'm^2}{2}$ if $p \equiv 1,3 \bmod 8$ and $t_{p^2} = \frac{2n^2 - D'm^2}{2}$ if $p \equiv 5,7 \bmod 8$. Using these notations, the characteristic equation of ϕ_2 is

$$\phi_2^2 - 2n\,\phi_2 + 2p\,\mathrm{Id} = 0\,.$$

We compute the two roots of the polynomial $\chi^2 - 2n\chi + 2p = 0$. We start with $\Delta = 4n^2 - 8p = 2(2n^2 - 4p)$ and inject $4p = D'm^2 + 2n^2$ in the expression to cancel the terms in n^2. Then $\Delta = -2D'm^2$ and the two roots are $\frac{2n \pm \sqrt{-2D'}m}{2}$. We know that $\phi_2^2 = [-2] \circ \pi_{p^2}$ if $p \equiv 1,3 \bmod 8$ and $\phi_2^2 = [2] \circ \pi_{p^2}$ if $p \equiv 5,7 \bmod 8$, with $\pi_{p^2} = \frac{t_{p^2} + n \cdot m\sqrt{-D}}{2}$. We compute

$$\phi_2^2 \leftrightarrow \left(\frac{2n \pm \sqrt{-2D'}m}{2}\right)^2 = \frac{2n^2 - D'm^2}{2} \pm n \cdot m\sqrt{-2D'}\,.$$

With the expression of t_{p^2}, we conclude that

$$\begin{cases} \phi_2 \text{ corresponds to } \frac{2n - m\sqrt{-2D'}}{2} \text{ if } p \equiv 1,3 \bmod 8, \\ \phi_2 \text{ corresponds to } \frac{2n + m\sqrt{-2D'}}{2} \text{ if } p \equiv 5,7 \bmod 8. \end{cases} \tag{2.84}$$

□

Theorem 9. *[GI13, Th. 1] Let $E_{1,c}$ be an elliptic curve given by equation (2.73), defined over $\mathbb{F}_{p^2}$. Let $-D$ be the complex multiplication discriminant and consider D' such that $D = 2D'$. There is an endomorphism $\phi_{D'}$ of $E_{1,c}$ with degree of separability D'. The characteristic equation of this endomorphism is*

$$\phi_{D'}^2 + D'm\,\phi_{D'} + D'p\,\mathrm{Id} = 0\,. \tag{2.85}$$

Proof. Since $D = 2D'$, we have that ϕ_D is the composition of a horizontal isogeny of degree 2 with a horizontal[1] isogeny of degree D'. We denote by $\mathcal{I}_2 : E_{1,c} \to E_{1,-c}$ the isogeny given by equation (2.76). Note that $\mathcal{I}_2$ is a horizontal isogeny of degree 2. Indeed, since $\pi_p : E_{1,-c} \to E_{1,c}$, it follows that $(\mathrm{End}(E_{1,c}))_2 \simeq (\mathrm{End}(E_{1,-c}))_2$. Since $2|D$, there is a unique horizontal isogeny of degree 2 starting from $E_{1,c}$. Hence the complex multiplication endomorphism on $E_{1,c}$ is $\phi_D = \mathcal{I}_{D'} \circ \mathcal{I}_2$, with $\mathcal{I}_{D'} : E_{1,-c} \to E_{1,c}$ a horizontal isogeny of degree D'. We define $\phi_{D'} = \mathcal{I}_{D'} \circ \pi'_p$, with $\pi'_p : E_{1,c} \to E_{1,-c}$. To compute the characteristic polynomial of $\phi_{D'}$, we observe that

$$\phi_{D'} \circ \phi_2 = \phi_D \circ \pi_{p^2}\,. \tag{2.86}$$

By using equation (2.83), we obtained in Lem. 4 that ϕ_2 seen as an algebraic integer in $\mathbb{Z}[\sqrt{-D}]$ is $\frac{2n - m\sqrt{-2D'}}{2}$ if $p \equiv 1,3 \bmod 8$ and $\frac{2n + m\sqrt{-2D'}}{2}$ if $p \equiv 5,7 \bmod 8$. Secondly ϕ_D corresponds to $\sqrt{-D}$ and π_{p^2} to $\frac{t_{p^2} + n \cdot m\sqrt{-D}}{2}$. We then solve the equality (2.86) and conclude that $\phi_{D'}$ seen as algebraic integer in $\mathbb{Z}[\sqrt{-D}]$ is $\frac{-D'm - n\sqrt{-2D'}}{2}$ if $p \equiv 1,3 \bmod 8$ and $\frac{-D'm + n\sqrt{-2D'}}{2}$ if $p \equiv 5,7 \bmod 8$. Hence we have $\phi_{D'}^2 + D'm\,\phi_{D'} + D'p\,\mathrm{Id} = 0$. □

We remark that if $p \equiv 1,3 \bmod 8$ then $\phi_{D'}^2 = [D'] \circ \pi_{p^2}$ and if $p \equiv 5,7 \bmod 8$ then $\phi_{D'}^2 = [-D'] \circ \pi_{p^2}$ as expected.

The endomorphism $\phi_{D'}$ constructed in Theorem 9 is computed as the composition of a horizontal isogeny with the p-power of the Frobenius. Since computing the p-power Frobenius for extension fields of degree 2 costs one negation, we conclude that $\phi_{D'}$ may be computed with Vélu's formulæ with half the operations needed to compute ϕ_D over $\mathbb{F}_{p^2}$.

1. An isogeny $I : E \to E'$ of degree ℓ is called horizontal if $(\mathrm{End}(E))_\ell \simeq (\mathrm{End}(E'))_\ell$.

2.4.1.3 Four dimensional Gallant-Lambert Vanstone method

Assume that $E_{1,c}$ is such that $\#E_{1,c}(\mathbb{F}_{p^2})$ is divisible by a large number of cryptographic size. Let $\Psi = \phi_{D'}$ and $\Phi = \phi_2$. We observe that Φ and Ψ viewed as algebraic integers are represented by $\frac{2n \pm m\sqrt{-D}}{2}$ and $\frac{-D'm \pm n\sqrt{-D}}{2}$. These two numbers are linear combinations of $\sqrt{-D}$ (the Complex Multiplication). However the dependancy contains large coefficients: n, m with $\log n \sim \log m \sim \frac{1}{2}\log p \sim \frac{1}{4}\log r$ hence they are large enough. Consequently, one may use $1, \Phi, \Psi, \Phi\Psi$ to compute the scalar multiple kP of a point $P \in E_{1,c}(\mathbb{F}_{p^2})$ using a four dimensional GLV algorithm. We do not give here the details of the algorithm which computes decompositions

$$k = k_1 + k_2\lambda + k_3\mu + k_4\lambda\mu,$$

with λ and μ the eigenvalues of Φ and Ψ and $|k_i| < Cr^{1/4}$. Such an algorithm is obtained by working over $\mathbb{Z}[\Phi, \Psi]$, using a similar analysis to the one proposed by Longa and Sica [LS12].

2.4.1.4 Eigenvalues

We deduce that the eigenvalue of ϕ_2 is $p\sqrt{-2}$ if $p \equiv 1 \mod 8$ and $p\sqrt{2}$ if $p \equiv 5 \mod 8$. We can explicitly compute this eigenvalue mod $\#E_{1,c}(\mathbb{F}_{p^2})$. We will use the formulas (2.81) and (2.82).

If $p \equiv 1, 3 \mod 8$, we obtain

$$\begin{array}{rcllcl} \#E_{1,c}(\mathbb{F}_{p^2}) & = & (p+1)^2 - D'm^2 & \rightarrow & \sqrt{D'} & \equiv & (p+1)/m \\ & = & (p-1)^2 + 2n^2 & \rightarrow & \sqrt{-2} & \equiv & (p-1)/n, \\ & = & (1 - t_{p^2}/2)^2 + 2D'(nm/2)^2 & \rightarrow & \sqrt{-2D'} & \equiv & (2 - t_{p^2})/(nm)\,. \end{array} \tag{2.87}$$

If $p \equiv 5, 7 \bmod 8$, we obtain

$$\begin{array}{rcllcl} \#E_{1,c}(\mathbb{F}_{p^2}) & = & (p-1)^2 + D'm^2 & \rightarrow & \sqrt{-D'} & \equiv & (p-1)/m \\ & = & (p+1)^2 - 2n^2 & \rightarrow & \sqrt{2} & \equiv & (p+1)/n, \\ & = & (1 - t_{p^2}/2)^2 + 2D'(nm/2)^2 & \rightarrow & \sqrt{-2D'} & \equiv & (2 - t_{p^2})/(nm)\,. \end{array} \tag{2.88}$$

The eigenvalue of ϕ_2 on $E_{1,c}(\mathbb{F}_{p^2})$ is $\sqrt{-2} \equiv (p-1)/n \bmod \#E_{1,c}(\mathbb{F}_{p^2})$ if $p \equiv 1, 3 \bmod 8$ or $\sqrt{2} \equiv (p+1)/n \bmod \#E_{1,c}(\mathbb{F}_{p^2})$ if $p \equiv 5, 7 \mod 8$.

The eigenvalue of $\phi_{D'}$ on $E_{1,c}(\mathbb{F}_{p^2})$ is $\sqrt{D'} \equiv (p+1)/m \mod \#E_{1,c}(\mathbb{F}_{p^2})$ if $p \equiv 1, 3 \mod 8$ or $\sqrt{-D'} \equiv (p-1)/m \mod \#E_{1,c}(\mathbb{F}_{p^2})$ if $p \equiv 5, 7 \mod 8$.

Remark 1. *There is no ambiguity on the endomorphism ring of $E_{1,c}$. Note that the curve is ordinary. Its endomorphism ring is $\mathrm{End}(E_{1,c}) = \mathbb{Z}[\sqrt{-D}]$ with the complex multiplication corresponding to the endomorphism ϕ_D of eigenvalue $\sqrt{-D}$. We obtained two other endomorphisms $\phi_2, \phi_{D'}$ with eigenvalue $\sqrt{2}$ and $\sqrt{-D'}$ if $p \equiv 1, 3 \bmod 8$, resp. $\sqrt{-2}$ and $\sqrt{D'}$ if $p \equiv 5, 7 \bmod 8$ (with $-D = -2D'$) but these eigenvalues are expressions modulo $\#E_{1,c}(\mathbb{F}_{p^2})$. Proof of Th. 9 tells that ϕ_2 corresponds to $(2n \pm m\sqrt{-2D'})/2$ and $\phi_{D'}$ corresponds to $(-mD' \pm n\sqrt{-2D'})/2$. For clarity, we explicit the relation between these generic eigenvalues and $\sqrt{\pm 2}, \sqrt{\pm D'}$ obtained in another way in eqs.* (2.87) *and* (2.88).

If $p \equiv 1, 3 \bmod 8$ then $t_{p^2} = (-2n^2 + D'm^2)/2$ according to eq. (2.81) *of Lemma 4. Moreover, $\sqrt{-D} = \sqrt{-2D'} \equiv (2 - t_{p^2})/(nm) \bmod \#E_{1,c}(\mathbb{F}_{p^2})$ from eq.* (2.87). *We obtain that ϕ_2 has eigenvalue*

$$\begin{array}{rcl} (2n - m\sqrt{-2D'})/2 & \equiv & \frac{1}{2}\left(2n - m\frac{2 - t_{p^2}}{nm}\right) \\ & \equiv & (2n^2 - 4 + D'm^2)/(4n) \\ & \equiv & (p-1)/n \equiv \sqrt{-2} \bmod \#E_{1,c}(\mathbb{F}_{p^2}) \text{ from } (2.87). \end{array} \tag{2.89}$$

Secondly if $p \equiv 5, 7 \bmod 8$ then the trace is $t_{p^2} = (2n^2 - D'm^2)/2$ (eq. (2.82) *Lem. 4) and we obtain this time*

$$\begin{array}{rcl} (2n + m\sqrt{-2D'})/2 & \equiv & \frac{1}{2}\left(2n + m\frac{2 - t_{p^2}}{nm}\right) \\ & \equiv & (2n^2 + 4 + D'm^2)/(4n) \\ & \equiv & (p+1)/n \equiv \sqrt{2} \bmod \#E_{1,c}(\mathbb{F}_{p^2}) \text{ from } (2.88). \end{array} \tag{2.90}$$

We conclude that ϕ_2 has eigenvalue

$$\phi_2 : \lambda_{\phi_2} = \begin{cases} \frac{2n-m\sqrt{-D}}{2} \equiv \sqrt{-2} \bmod \#E_{1,c}(\mathbb{F}_{p^2}) \text{ if } p \equiv 1,3 \bmod 8, \\ \frac{2n+m\sqrt{-D}}{2} \equiv \sqrt{2} \bmod \#E_{1,c}(\mathbb{F}_{p^2}) \text{ if } p \equiv 5,7 \bmod 8\,. \end{cases} \tag{2.91}$$

We can do the same for the second endomorphism $\phi_{D'}$. We obtain that $\phi_{D'}$ has eigenvalue

$$\phi_{D'} : \lambda_{\phi_{D'}} = \begin{cases} \frac{-D'm-n\sqrt{-D}}{2} \equiv -\sqrt{D'} \bmod \#E_{1,c}(\mathbb{F}_{p^2}) \text{ if } p \equiv 1,3 \bmod 8, \\ \frac{-D'm+n\sqrt{-D}}{2} \equiv -\sqrt{-D'} \bmod \#E_{1,c}(\mathbb{F}_{p^2}) \text{ if } p \equiv 5,7 \bmod 8\,. \end{cases} \tag{2.92}$$

2.4.1.5 Example with $-D = -40$

By equations (2.81) and (2.82), we have that

$$4p = 2n^2 + D'y^2.$$

Using Magma, we computed an example with $p \equiv 5 \mod 8$, $D' = 20$.

Example 14. *We first search 63-bit numbers n, y such that $4 \mid n$, $y \equiv 1 \mod 4$, $p = (2n^2 + 20y^2)/4$ is prime and $\#E_{1,c}(\mathbb{F}_{p^2})$ is almost prime. We can expect an order of the form $4r$, with r prime. In few seconds, we find the following parameters.*

$$\begin{aligned} n &= \texttt{0x55d23edfa6a1f7e4} \\ y &= \texttt{0x549906b3eca27851} \\ t_{p^2} &= -\texttt{0xfaca844b264dfaa353355300f9ce9d3a} \\ p &= \texttt{0x9a2a8c914e2d05c3f2616cade9b911ad} \\ r &= \texttt{0x1735ce0c4fbac46c2245c3ce9d8da0244f9059ae9ae4784d6b2f65b29c444309} \\ c^2 &= \texttt{0x40b634aec52905949ea0fe36099cb21a} \end{aligned}$$

with r, p prime and $\#E_{1,c}(\mathbb{F}_{p^2}) = 4r$.

We use Vélu's formulas to compute a degree-5 isogeny from $E_{1,c}$ into $E_{b,5}$. We find a 5-torsion point $P_5(X_5, Y_5)$ in $E_{1,c}(\mathbb{F}_{p^8})$. The function `IsogenyFromKernel` in Magma evaluated at $(E_{1,c}(\mathbb{F}_{p^8}), (X - X_{P_5})(X - X_{2P_5}))$ outputs a curve $E_{b,5}$ with $b_{5,4} = -25 \cdot 27(3c + 10) = 5^2 a_{4,-c}$ and $b_{5,6} = 125 \cdot 108(9c + 14) = 5^3 a_{6,-c}$. Hence $E_{b,5}$ and $E_{1,-c}$ are *isomorphic* over $\mathbb{F}_{p^2}$ through $i_{\sqrt{5}} : (x_{b,5}, y_{b,5}) \mapsto (x_{b,5}/5, y_{b,5}/(5\sqrt{5}))$. The above function outputs also the desired isogeny with coefficients in $\mathbb{F}_{p^2}$:

$$\begin{aligned} \mathcal{I}_5 : E_{1,c} &\to E_{b,5} \\ (x, y) &\mapsto \left(x + \frac{2 \cdot 3^3 \left(\frac{3}{5}(13c + 40)x + 4(27c + 28)\right)}{x^2 + \frac{27}{2}cx - \frac{81}{10}c + 162} \right. \\ &\quad + \frac{-2^3 \cdot 3^4((9c + 16)x^2 + \frac{2}{5}11(27c + 64)x + \frac{2}{5}3^3(53c + 80)}{(x^2 + \frac{27}{2}cx - \frac{81}{10}c + 162)^2}, \\ &\quad y \left(1 + \frac{-2^4 \cdot 3^4((9c + 16)x^3 + \frac{3}{5}11(27c + 64)x^2 + \frac{2}{5}3^4(53c + 80)x + \frac{2}{5^2}3^2(4419c + 13360))}{(x^2 + \frac{27}{2}cx - \frac{81}{10}c + 162)^3} \right. \\ &\quad \left. \left. + \frac{2 \cdot 3^3 \left(\frac{3}{5}(13c + 40)x^2 + 2^3(27c + 28)x + 2\frac{3}{5}(369c + 1768)\right)}{(x^2 + \frac{27}{2}cx - \frac{81}{10}c + 162)^2} \right) \right) \end{aligned} \tag{2.93}$$

We finally obtain a second computable endomorphism ϕ_5 on $E_{1,c}$ in this example by composing $\pi_p \circ i_{\sqrt{5}} \circ \mathcal{I}_5$.

2.4.1.6 Example with $-D = -4$

Assume that curve is defined over $\mathbb{F}_{p^2}$, with $p \equiv 1 \mod 8$. Our construction gives two endomorphisms ϕ_2, ϕ_{-2} such that $\phi_2^2 - 2 = 0$, $\phi_{-2}^2 + 2 = 0$. The discriminant of the curve is $-D = -4$. The curve

is of the form $E_\alpha : y^2 = x^3 + \alpha x$ with $\alpha \in \mathbb{F}_{p^2}$. A 2-torsion point is $P_2(0,0)$. Vélu's formulas applied to this point give us an isogeny $(x,y) \mapsto (x + \frac{\alpha}{x}, y - y\frac{\alpha}{x^2})$ into $E_b : y^2 = x^3 - 4\alpha x$. The j-invariant of this curve is 1728 hence the curves are *isomorphic*. Applying $(x_b, y_b) \mapsto (x_b/(2i), y_b/(2i)(1+i))$ (as $(1+i)^4 = -4$) to go back in E_α does not give us the endomorphism we are looking for, this gives us $[1+\sqrt{-1}]$ actually. We use the same trick as previously. If $\alpha \in \mathbb{F}_{p^2}$ is such that $\pi_p(\alpha) = \alpha^p = -\alpha$ (this is the case for example if $\alpha = \sqrt{a}$ with $a \in \mathbb{F}_p$ a non-square) then $(x_b, y_b) \mapsto (x_b^p/(-2), y_b^p/(-2\sqrt{-2}))$ gives us the endomorphism ϕ_2 and $(x_b, y_b) \mapsto (x_b^p/2, y_b^p/2\sqrt{2})$ gives us ϕ_{-2}. Note that $\sqrt{-1}, \sqrt{2}, \sqrt{-2} \in \mathbb{F}_p$ since $p \equiv 1 \mod 8$. We obtain

$$\begin{array}{rcl} \phi_2 : E_\alpha & \to & E_\alpha \\ (x,y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } (x,y) = (0,0), \\ \left(\frac{(x^p)^2+\alpha}{2x^p}, \frac{y^p}{2\sqrt{2}}\left(1 - \frac{\alpha}{(x^p)^2}\right)\right) & \text{otherwise,} \end{cases} \\ \phi_{-2} : E_\alpha & \to & E_\alpha \\ (x,y) & \mapsto & \begin{cases} \mathcal{O} & \text{if } (x,y) = (0,0), \\ \left(\frac{(x^p)^2+\alpha}{-2x^p}, \frac{y^p}{-2\sqrt{-2}}\left(1 - \frac{\alpha}{(x^p)^2}\right)\right) & \text{otherwise.} \end{cases} \end{array} \tag{2.94}$$

Since the j-invariant $j = 1728 \in \mathbb{F}_p$, we observe that the curve E_α is a GLS curve and is treated in [LS12, App. B]. The 4 dimensional GLV algorithm of Longa and Sica on this curve uses an endomorphism Ψ such that $\Psi^4 + 1 = 0$. With our method we obtain two distinct endomorphisms but these three ones Ψ, ϕ_2, ϕ_{-2} are linearly dependent on the subgroup $E(\mathbb{F}_{p^2}) \setminus E[2]$.

In this case the corresponding Jacobian splits into two isogenous elliptic curves over $\mathbb{F}_p$, namely the two quartic twists defined over $\mathbb{F}_p$ of $E_{1,c}$.

2.4.2 Endomorphisms on $E_{2,c}$

The construction of two efficiently computable endomorphisms on $E_{2,c}$, with degree of inseparability p, is similar to the one we gave for $E_{1,c}$.

2.4.2.1 First endomorphism from Velu's formulas

We consider the elliptic curve over $\mathbb{F}_{p^2}$ given by eq. (2.22) in the reduced form:

$$E_{2,c} : y^2 = x^3 + 3(2c-5)x + c^2 - 14c + 22\,. \tag{2.95}$$

We assume that $c \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$, $c^2 \in \mathbb{F}_p$, c is not a cube in $\mathbb{F}_{p^2}$. In this case the isogeny (2.23) between $J_{\mathcal{C}_2}$ and $E_{2,c} \times E_{2,-c}$ is defined over $\mathbb{F}_{p^6}$. The 3-torsion subgroup $E_{2,c}(\mathbb{F}_{p^2})[3]$ contains the order 3 subgroup $\{\mathcal{O}, (3, c+2), (3, -c-2)\}$. We compute an isogeny whose kernel is this 3-torsion subgroup. With Vélu's formulas we obtain the curve $E_b : y^2 = x^3 - 27(2c+5)x - 27(c^2+14c+22)$. The curve E_b is isomorphic over $\mathbb{F}_{p^2}$ to $E_{2,-c} :: y^2 = x^3 - 3(2c+5)x + c^2 + 14c + 22$, via the isomorphism $(x,y) \mapsto \left(x/(-3), y/(-3\sqrt{-3})\right)$. We define the isogeny

$$\begin{array}{rcl} \mathcal{I}_3 : E_{2,c} & \to & E_{2,-c} \\ (x,y) & \mapsto & \left(\frac{-1}{3}\left(x + \frac{12(c+2)}{x-3} + \frac{4(c+2)^2}{(x-3)^2}\right), \frac{-y}{3\sqrt{-3}}\left(1 - \frac{12(c+2)}{(x-3)^2} - \frac{8(c+2)^2}{(x-3)^3}\right)\right)\,. \end{array} \tag{2.96}$$

Finally, we observe that $\pi_p(c) = -c$ and $\pi_p(j(E_{2,c})) = j(E_{2,-c})$. This implies that $E_{2,c}$ and $E_{2,-c}$ are isogenous through the Frobenius map π_p. We obtain the endomorphism $\phi_3 = \mathcal{I}_3 \circ \pi_p$ over $\mathbb{F}_{p^2}$ which is given by the following formula

$$\begin{array}{rcl} \phi_3 : E_{2,c} & \to & E_{2,c} \\ (x,y) & \mapsto & \left(\frac{-1}{3}\left(x^p + \frac{12(2-c)}{x^p-3} + \frac{4(2-c)^2}{(x^p-3)^2}\right), \frac{y^p}{-3\sqrt{-3}^p}\left(1 - \frac{12(2-c)}{(x^p-3)^2} - \frac{8(2-c)^2}{(x^p-3)^3}\right)\right) \end{array} \tag{2.97}$$

We compute formally ϕ_3^2 and obtain $\phi_3^2 = \pi_{p^2} \circ [\pm 3]$. There is a term $\sqrt{-3}\sqrt{-3}^p$ in the y side of ϕ_3^2. We observe that if $p \equiv 1 \mod 3$ then $\left(\frac{-3}{p}\right) = 1$ and $\sqrt{-3}\sqrt{-3}^p = -3$ so $\phi_3^2 = \pi_{p^2} \circ [-3]$. If $p \equiv 2 \mod 3$ then $\phi_3^2 = \pi_{p^2} \circ [3]$. We conclude that for points in $E_{2,c}(\mathbb{F}_{p^2})$, we have

$$\begin{array}{l} \phi_3^2 + 3 = 0 \text{ over } \mathbb{F}_{p^2} \text{ if } p \equiv 1 \mod 3, \\ \phi_3^2 - 3 = 0 \text{ over } \mathbb{F}_{p^2} \text{ if } p \equiv 2 \mod 3\,. \end{array} \tag{2.98}$$

2.4.2.2 Second endomorphism from Complex Multiplication

With the same arguments as for $E_{1,c}$, we deduce this lemma.

Lemma 5. *There are integers m and n such that if* $p \equiv 1 \pmod 3$, *then*

$$t_{p^2} + 2p = D' m^2 \text{ and } t_{p^2} - 2p = -3n^2$$

and if $p \equiv 2 \pmod 3$, *then*

$$t_{p^2} + 2p = 3n^2 \text{ and } t_{p^2} - 2p = -D' m^2 .$$

The endomorphism ϕ_3 *has characteristic equation*

$$\phi_3^2 - 3n\,\phi_3 + 3p\,\mathrm{Id} = 0 \tag{2.99}$$

and corresponds to the number $\frac{3n - m\sqrt{-D}}{2}$ *if* $p \equiv 1 \bmod 3$ *and* $\frac{3n + m\sqrt{-D}}{2}$ *if* $p \equiv 2 \bmod 3$.

Proof. We start again from $\phi_3^2 - \mathrm{Tr}(\phi_3)\phi_3 + \deg(\phi_3)\mathrm{Id} = 0$. We have that $\mathrm{Tr}(\phi_3^2) - \mathrm{Tr}^2(\phi_3) + 2\deg(\phi_3) = 0$. We know that $\deg(\phi_3) = 3p$ since $\phi_3 = \pi_p \circ \mathcal{I}_3$ with $\deg(\pi_p) = p$ and $\deg(\mathcal{I}_3) = 3$. Then the equation is $\mathrm{Tr}^2(\phi_3) = \mathrm{Tr}(\phi_3^2) + 6p$. Now if $p \equiv 1 \bmod 3$ then $\mathrm{Tr}(\phi_3^2) = \mathrm{Tr}(\pi_{p^2} \circ [-3]) = -3t_{p^2}$ and we get $\mathrm{Tr}^2(\phi_3) = -3t_{p^2} + 6p = -3(t_{p^2} - 2p)$. We may thus write $t_{p^2} - 2p = -3n^2$, for some integer n. Secondly if $p \equiv 2 \bmod 3$ then $\mathrm{Tr}(\phi_3^2) = \mathrm{Tr}(\pi_{p^2} \circ [3]) = 3t_{p^2}$ and we get $\mathrm{Tr}^2(\phi_3) = 3t_{p^2} + 6p = 3(t_{p^2} + 2p)$. We obtain $t_{p^2} + 2p = 3n^2$, for some integer n. Using the complex multiplication equation $(t_{p^2})^2 - 4p^2 = -3D'\gamma^2$, there is an integer m such that $t_{p^2} + 2p = D' m^2$ if $p \equiv 1 \bmod 3$ or $t_{p^2} - 2p = -D' m^2$ if $p \equiv 2 \bmod 3$. As a consequence, we can write $4p = 3n^2 + D' m^2$ and $2t_{p^2} = -3n^2 + D' m^2$ if $p \equiv 1 \bmod 3$, $2t_{p^2} = 3n^2 - D' m^2$ if $p \equiv 2 \bmod 3$.

The characteristic equation of ϕ_3 is

$$\phi_3^2 - 3n\,\phi_3 + 3p\,\mathrm{Id} = 0 .$$

We also compute formally the two roots of the characteristic equation of ϕ_3. We start with $\Delta = 9n^2 - 12p = 3(3n^2 - 4p)$ and inject $4p = D' m^2 + 3n^2$ in the expression to cancel the terms in n^2. Then $\Delta = -3D' m^2$ and the two roots of $\chi^2 - 3n\chi + 3p$ are $\frac{3n \pm m\sqrt{-3D'}}{2} = \frac{3n \pm m\sqrt{-D}}{2}$. We know that $\phi_3^2 = [-3] \circ \pi_{p^2}$ if $p \equiv 1 \bmod 3$ and $\phi_3^2 = [3] \circ \pi_{p^2}$ if $p \equiv 2 \bmod 3$, with $\pi_{p^2} = (t_{p^2} + n \cdot m\sqrt{-D})/2$. We compute

$$\left(\frac{3n \pm \sqrt{-3D'}m}{2}\right)^2 = \frac{3}{2}\left(\frac{3n^2 - D' m^2}{2} \pm n \cdot m\sqrt{-3D'}\right) .$$

With the expression of t_{p^2}, we conclude that

$$\begin{cases} \phi_3 \text{ corresponds to } \frac{3n - m\sqrt{-3D'}}{2} \text{ if } p \equiv 1 \bmod 3, \\ \phi_3 \text{ corresponds to } \frac{3n + m\sqrt{-3D'}}{2} \text{ if } p \equiv 2 \bmod 3. \end{cases} \tag{2.100}$$

□

As a consequence, we have the following theorem, whose proof is similar to the proof of Th. 9.

Theorem 10. *Let* $E_{2,c}$ *be an elliptic curve given by equation* (2.95), *defined over* $\mathbb{F}_{p^2}$. *Let* $-D$ *be the complex multiplication discriminant and consider* D' *such that* $-D = -3D'$. *There is an endomorphism* $\phi_{D'}$ *of* $E_{2,c}$ *with degree of separability* D'. *The characteristic equation of this endomorphism is*

$$\phi_{D'}^2 - D' m\,\phi_{D'} + D' p\,\mathrm{Id} = 0 . \tag{2.101}$$

Remark 2. *The eigenvalue of* ϕ_3 *is* $\sqrt{-3}$ *and the eigenvalue of* $\phi_{D'}$ *is* $\sqrt{D'}$ *when* $p \equiv 1 \bmod 3$, *resp.* $\sqrt{3}, \sqrt{-D'}$ *when* $p \equiv 2 \bmod 3$. *However these values are expressed modulo the elliptic curve order* $\#E(\mathbb{F}_{p^2})$. *To obtain the general expression, we compute the algebraic integer in* $\mathrm{End}(E_{2,c}) = \mathbb{Z}[\sqrt{-D}]$ *to which* ϕ_3 *and* $\phi_{D'}$ *correspond,*

from their characteristic equation. We obtain that ϕ_3 *corresponds to* $\frac{3n-m\sqrt{-3D'}}{2} \equiv \sqrt{-3} \pmod{\#E_{2,c}(\mathbb{F}_{p^2})}$ *if* $p \equiv 1 \bmod 3$ *and* $\frac{3n+m\sqrt{-3D'}}{2} \equiv \sqrt{3}$ *if* $p \equiv 2 \bmod 3$. *In the same way,* $\phi_{D'}$ *corresponds to* $\frac{-mD'-n\sqrt{-3D'}}{2} \equiv -\sqrt{D'}$ *if* $p \equiv 1 \bmod 3$ *and* $\frac{-mD'+n\sqrt{-3D'}}{2} \equiv -\sqrt{-D'}$ *if* $p \equiv 2 \bmod 3$.

The two endomorphisms seen as algebraic integers do not generate an additional dimension of the endomorphism ring. However the coefficients m, n *involved in their expression in term of* ϕ_D *are large enough so that the lattice reduction algorithm will succeed in the GLV-decomposition step. We obtain a four-dimensional GLV algorithm on* $E_{2,c}$.

2.4.2.3 Eigenvalues

To compute the eigenvalues of $\phi_{D'}$ and ϕ_3, we write $p = \frac{3n^2+D'm^2}{4}$, $t_{p^2} = \frac{D'm^2-3n^2}{2}$. We obtain

$$\begin{array}{rcccccl} \#E_{2,c}(\mathbb{F}_{p^2}) & = & (p-1)^2 - D'm^2 & \rightarrow & \sqrt{D'} & \equiv & (p-1)/m \mod \#E_{2,c}(\mathbb{F}_{p^2}), \\ & = & (p+1)^2 + 3n^2 & \rightarrow & \sqrt{-3} & \equiv & (p+1)/n, \\ & = & (t_{p^2}/2-1)^2 + 3D'(nm/2)^2 & \rightarrow & \sqrt{-3D'} & \equiv & (t_{p^2}-2)/nm\,. \end{array}$$

The eigenvalue of ϕ_3, mod $\#E_{2,c}(\mathbb{F}_{p^2})$ is $p(p+1)/n$ and the eigenvalue of $\phi_{D'}$, mod $\#E_{2,c}(\mathbb{F}_{p^2})$ is $p(p-1)/m$.

2.4.2.4 Example with $D = -3$.

This case is kind of a *degenerate* case. The curve $E_{2,c}$ is a GLS curve E_β whose Weierstrass equation is

$$E_\beta(\mathbb{F}_{p^2}) : y^2 = x^3 + \beta$$

where $\beta \notin \mathbb{F}_p$, $\beta^2 \in \mathbb{F}_p$. Longa and Sica obtained two endomorphisms Φ, Ψ such that the characteristic polynomial of Φ satisfies $\chi^2 + \chi + 1 = 0$ and the characteristic polynomial of Ψ is such that $\chi^2 + 1 = 0$. Our construction yields the following efficiently computable endomorphism

$$\phi_3(x,y) = \left(\frac{1}{3}\left(x^p + \frac{4\beta^p}{x^{2p}}\right), \frac{y^p}{\sqrt{3}}\left(1 + \frac{8\beta^p}{x^{3p}}\right)\right).$$

When restricted to points defined over $\mathbb{F}_{p^2}$, this endomorphism verifies the equation $\phi_3^2 - 3 = 0$, while the complex multiplication endomorphism Φ has characteristic equation $\chi^2 + \chi + 1 = 0$. Longa and Sica's algorithm uses the complex multiplication Φ and an endomorphism Ψ verifying $\Psi^2 + \mathrm{Id} = 0$ for points defined over $\mathbb{F}_{p^2}$. We observe that $2\phi_3 \circ \Psi - 1 = 2\Phi$.

The costs of all these endomorphisms are comparable. The main difference is their characteristic polynomial, thus their eigenvalue. It would be interesting to compare which choice of endomorphisms give the best lattice reduction on average at the beginning of a scalar multiplication.

2.5 Two independent endomorphisms on the Jacobians $J_{\mathcal{C}_1}$ and $J_{\mathcal{C}_2}$ from the two endomorphisms available on the isogenous elliptic curves

2.5.1 Endomorphisms on $J_{\mathcal{C}_1}$

The first endomorphism ψ on $J_{\mathcal{C}_1}$ is induced by the curve automorphism $(x,y) \rightarrow (-x, iy)$, with i a square root of -1. The characteristic polynomial is $\chi^2 + 1 = 1$. The second endomorphism is constructed as $\phi = \hat{\mathcal{I}} \circ (\phi_{D'}, \phi_{D'}) \circ \mathcal{I}$, where $\phi_{D'}$ is the elliptic curve endomorphism constructed in Theorem 9. In order to compute the characteristic equation for ϕ, we follow the lines of the proof of Theorem 1 in [GLS09]. We reproduce the computation for the Jacobian of $\mathcal{C}_1$.

Theorem 11. *Let* $\mathcal{C}_1 : Y^2 = X^5 + aX^3 + bX$ *a hyperelliptic curve defined over* $\mathbb{F}_p$ *with ordinary Jacobian and let* r *a prime number such that* $r || J_{\mathcal{C}_1}(\mathbb{F}_p)$. *Let* $\mathcal{I} : J_{\mathcal{C}_1} \rightarrow E_{1,c} \times E_{1,c}$ *the* $(2,2)$*-isogeny defined by equation* (2.6) *and assume* $\mathcal{I}$ *is defined over an extension field of degree* $k > 1$. *We define* $\phi = \hat{\mathcal{I}} \circ (\phi_{D'} \times \phi_{D'}) \circ \mathcal{I}$ *where* $\phi_{D'}$ *is the endomorphism defined in Theorem 9. Then*

1. *For $\mathcal{D} \in J_{\mathcal{C}_1}[r](\mathbb{F}_p)$, we have $\phi(\mathcal{D}) = \lambda\mathcal{D}$, with $\lambda \in \mathbb{Z}$.*
2. *The characteristic equation of ϕ is $\phi^2 + 2D'm\,\phi + 4D'p\,\mathrm{Id} = 0$.*

Proof. 1. Note that $\mathrm{End}(J_{\mathcal{C}_1})$ is commutative, and ϕ is defined over $\mathbb{F}_p$ (see [Bis11, Prop. III.1.3]). Hence, for $\mathcal{D} \in J_{\mathcal{C}_1}(\mathbb{F}_p)$, we have that $\pi(\phi(\mathcal{D})) = \phi(\pi(\mathcal{D})) = \phi(\mathcal{D})$. Since there is only one subgroup of order r in $J_{\mathcal{C}_1}(\mathbb{F}_p)$, we obtain that $\phi(\mathcal{D}) = \lambda\mathcal{D}$.

2. Since $\hat{\mathcal{I}} \circ \mathcal{I} = [2]$ then

$$\phi^2 = \hat{\mathcal{I}} \circ (\phi_{D'} \times \phi_{D'}) \circ \mathcal{I} \circ \hat{\mathcal{I}} \circ (\phi_{D'} \times \phi_{D'}) \circ \mathcal{I} = [2]\hat{\mathcal{I}} \circ (\phi_{D'}^2, \phi_{D'}^2) \circ \mathcal{I}\,. \tag{2.102}$$

Since $\phi_{D'}$ verifies the equation

$$\phi_{D'}^2 + D'm\,\phi_{D'} + D'p\,\mathrm{Id} = 0\,, \tag{2.103}$$

we have

$$[2]\hat{\mathcal{I}} \circ ((\phi_{D'}^2, \phi_{D'}^2) + D'm\,(\phi_{D'}, \phi_{D'}) + D'p\,(\mathrm{Id}, \mathrm{Id})) \circ \mathcal{I} = \mathcal{O}_{J_{\mathcal{C}_1}}\,. \tag{2.104}$$

Using equation (2.102), we conclude that $\phi^2 + 2D'm\,\phi + 4D'p\,\mathrm{Id} = 0$.

□

Remark 3. *We compute the eigenvalue of this endomorphism $\phi = \hat{\mathcal{I}} \circ (\phi_{D'}, \phi_{D'}) \circ \mathcal{I}$. The two roots of the polynomial $\chi^2 + 2D'm\chi + 4D'p$ (Th. 11 (2)) are $(-D'm \pm n\sqrt{-D})$. Note that the endomorphism $\phi_{D'}$ on $E_{1,c}(\mathbb{F}_{p^2})$ has eigenvalue $(-D'm \pm n\sqrt{-D})/2$ (see* (2.92)*). The eigenvalue of ϕ is then twice the eigenvalue of $\phi_{D'}$.*

We can also compute these values modulo the Jacobian order. It was shown in [GV12] that when $p \equiv 1 \bmod 4$ and b (in the curve equation $\mathcal{C}_1$) is not a square then the Jacobian order is equal to $p^2 + 1 \pm 2n(p+1) + 2n^2$ [GV12, Th. 1 (4.) §2.2] with n such that $t_{p^2} + 2p = 2n^2$ (this happens if $p \equiv 5 \bmod 8$) or $t_{p^2} - 2p = -2n^2$ (if $p \equiv 1 \bmod 8$). To simplify, we put the sign $\pm$ in $n \in \mathbb{Z}$, then $\#J_{\mathcal{C}_1}(\mathbb{F}_p) = p^2 + 1 + 2n(p+1) + 2n^2$. We will compute the eigenvalue of the endomorphisms whose characteristic equations are $\chi^2 + 1 = 0$ and $\chi^2 + 2D'm\chi + 4D'p = 0$. We know that $4p = 2n^2 + D'm^2$.

$$\begin{array}{rcllcl}
\#J_{\mathcal{C}_1}(\mathbb{F}_p) & = & (p+n)^2 + (n+1)^2 & \to & \sqrt{-1} & \equiv \dfrac{p+n}{n+1} \bmod \#J_{\mathcal{C}_1}(\mathbb{F}_p), \\
& = & (p+n+1)^2 - 2D'm^2/4 & \to & \sqrt{2D'} & \equiv 2\dfrac{p+n+1}{m}, \\
& = & (-p+n^2+n+1)^2 + 2D'(m(n+1)/2)^2 & \to & \sqrt{-2D'} & \equiv 2\dfrac{-p+n^2+n+1}{m(n+1)}\,.
\end{array} \tag{2.105}$$

Hence the eigenvalue of ϕ_{-1} is $\sqrt{-1} = \frac{p+n}{n+1}$. We can also compute the eigenvalue of ϕ, modulo $\#J_{\mathcal{C}_1}(\mathbb{F}_q)$ with the values above in (2.105)*.*

We detailed in Sec. 2.2.1.2 how to compute efficiently the $(2,2)$-isogeny from $J_{\mathcal{C}_1}$ into $E_{1,c} \times E_{2,c}$. We briefly say that the composition of $\mathcal{I}$, $\phi_{D'}$ and $\hat{\mathcal{I}}$ is practical. Let $\mathcal{D}$ be a divisor on the Jacobian $J_{\mathcal{C}_1}(\mathbb{F}_p)$. We first compute $\varphi_{1*}(\mathcal{D}) = S_1(x_3, y_3)$, with the notations of Sec. 2.2.1.2. We also denote $S_2 = \varphi_{2*}(\mathcal{D})$. We then apply the endomorphism $\phi_{D'}$ on S_1. As $\phi_{D'}$ is defined over $\mathbb{F}_{p^2}$, it commutes with π_{p^2} hence $\phi_{D'}(S_2) = \phi_{D'}(\pi_{p^2}(S_1)) = \pi_{p^2}(\phi_{D'}(S_1))$ is free. Unfortunately S_1 has coefficients in $\mathbb{F}_{p^4}$ hence we need to perform some multiplications in $\mathbb{F}_{p^4}$ to compute $\phi_{D'}(S_1)$. More precisely, y_3 is of the form $\sqrt[8]{b}\gamma_3$ with $\gamma_3 \in \mathbb{F}_{p^4}$. As the endomorphism is of the form $\phi_{D'}(x,y) = (\phi_{D',x}(x), y\phi_{D',y}(x))$ the $\sqrt[8]{b}\gamma_3$ term is not involved in the endomorphism computation.

We detailed in Sec. 2.2.1.3 how to compute efficiently the dual isogeny $\hat{\mathcal{I}}$ from $E_{1,c} \times E_{2,c}$ into $J_{\mathcal{C}_1}$. We concluded that applying $\varphi_{1*}(P_1) + \varphi_{1*}(P_2)$ costs roughly as much as an addition on $J_{\mathcal{C}_1}$ over $\mathbb{F}_p$, $\varphi_{2*}(P_1) + \varphi_{2*}(P_2)$ is cost free. Then computing $\phi_{D'}$ depends on the size of D' and costs few multiplications over $\mathbb{F}_{p^4}$, for example if $D' = 2, 3, 5$. Finally adding $\varphi_1^*(\phi_{D'}(S_1)) + \varphi_2^*(\phi_{D'}(S_2))$ is simplified thanks to the equality $\varphi_2^*(\phi_{D'}(S_2)) = \pi_{p^2}\left(\varphi_1^*(\phi_{D'}(S_1))\right)$ and costs roughly an addition of divisors over $\mathbb{F}_{p^2}$.

2.5.1.1 Eigenvalues

As for the familly of elliptic curves studied in Sec. 2.4.1, we can compute explicitly the eigenvalues (as in Sec. 2.4.1.4) of the two endomorphisms on $J_{\mathcal{C}_1}$. Since we want the first endomorphism to be defined over $\mathbb{F}_q$, we set $q \equiv 1 \bmod 4$. We know from Th. 7 that when b is not a square in $\mathbb{F}_q$ the Jacobian order is equal to $q^2+1 \pm 2\gamma_i(q+1)+2\gamma_i^2$ where $\gamma_i \in \mathbb{N}$ such that either $2q + t_{q^2} = 2\gamma_1^2$ or $2q - t_{q^2} = -2\gamma_2^2$. To simplify, we put the sign in $\gamma_i \in \mathbb{Z}$, then $\#J_{\mathcal{C}_1}(\mathbb{F}_q) = q^2+1+2\gamma_i(q+1)+2\gamma_i^2$. We will compute the eigenvalue of the endomorphisms whose characteristic equations are $\chi^2+1=0$ and $\chi^2+2D'p\chi+1=0$. We separate in two cases.

1. If $t_{q^2}+2q = D'\gamma_1^2$ and $t_{q^2}-2q = -2\gamma_2^2$ then we can write $q = (D'\gamma_1^2+2\gamma_2^2)/4$ and moreover,

$$\begin{aligned}
\#J_{\mathcal{C}_1}(\mathbb{F}_q) &= (q+\gamma_2)^2+(\gamma_2+1)^2 &\rightarrow \quad \sqrt{-1} &\equiv \frac{q+\gamma_2}{\gamma_2+1} \bmod \#J_{\mathcal{C}_1}(\mathbb{F}_q),\\
&= (q+\gamma_2+1)^2+2D'\gamma_1^2/4 &\rightarrow \quad \sqrt{-2D'} &\equiv 2\frac{q+\gamma_2+1}{\gamma_1},\\
&= (-q+\gamma_2^2+\gamma_2+1)^2-2D'(\gamma_1(\gamma_2+1)/2)^2 &\rightarrow \quad \sqrt{2D'} &\equiv 2\frac{-q+\gamma_2^2+\gamma_2+1}{\gamma_1(\gamma_2+1)}.
\end{aligned}$$

   we simplified with $t = (D'\gamma_1^2-2\gamma_2^2)/2$ and obtained $\sqrt{2D'} \equiv \dfrac{-t_{q^2}+2(\gamma_2+1)}{\gamma_1(\gamma_2+1)} \bmod \#J_{\mathcal{C}_1}(\mathbb{F}_q)$. We recall from Sec. 2.4.1.4 that the eigenvalue of $\phi_{-2}\circ\phi_{D'}$ on $E_{1,c}(\mathbb{F}_{q^2})$ is $\sqrt{-2D'} \equiv (2-t_{q^2})/(\gamma_1\gamma_2)$.
2. If $2q+t_{q^2} = 2\gamma_1^2$ and $2q-t_{q^2} = -D'\gamma_2^2$ then γ_1 and γ_2 are simply swapped. We can compute the eigenvalues in the same way.

2.5.2 Endomorphisms on $J_{\mathcal{C}_2}$

The first endomorphism ψ on $J_{\mathcal{C}_2} : Y^2 = X^6+aX^3+b$ is induced by the curve automorphism $(x,y) \rightarrow (\zeta_3 x, y)$. Its characteristic equation is $\chi^2+\chi+1=0$. The second endomorphism is computed from a Complex Multiplication available on $E_{2,c}$. The construction is very similar to the one in the previous section (Sec. 2.5.1) for the other family of Jacobians. We briefly give some results. We assume that b is not a square neither a cube in $\mathbb{F}_q$. The second endomorphism ϕ on $J_{\mathcal{C}_2}$ is constructed as $\hat{\mathcal{I}}\circ(\phi_{D'},\phi_{D'})\circ\mathcal{I}$ with $\phi_{D'}$ the endomorphism constructed in Sec. 2.4.2.2 on $E_{2,c}$, whose characteristic polynomial is $\chi^2+D'm\chi+D'p$ (and reduced to $\mathbb{F}_{p^2}$, we have $\phi_{D'}^2 \pm D' = 0$). We can compute accordingly the eigenvalue of ϕ modulo $\#J_{\mathcal{C}_2}(\mathbb{F}_q)$ as previously in Sec. 2.5.1.1, this time from the expression of the Jacobian order given in Th. 8.

2.6 Pairing-Friendly constructions for $J_{\mathcal{C}_1}$ and $J_{\mathcal{C}_2}$

In this section we construct pairing-friendly genus-2 curves of the form $\mathcal{C}_1$ and $\mathcal{C}_2$ over a prime field. After the recent work on endomorphisms on $J_{\mathcal{C}_1}$ and $J_{\mathcal{C}_2}$ we realized that the pairing computation on these Jacobians can be speed-up. I would be also possible to construct pairing-friendly elliptic curves of the form $E_{1,c}$ and $E_{2,c}$, defined over a quadratic extension $\mathbb{F}_{p^2}$. The two endomorphisms would provide an efficient decomposition of the Miller loop. However a construction of pairing-friendly elliptic curves over $\mathbb{F}_{p^2}$ with a large prime-order r subgroup such that $\rho = 2\log p/\log r < 2$ is not known at the moment. The speed-up from the two endomorphisms will be completely offset because of the large parameter size.

We recall some basic facts on pairing-friendly constructions. We have several constraints for suitable pairing-friendly constructions inherent to elliptic curves:

1. The embedding degree k must be small, in order to achieve the same security level in bits in the elliptic curve r-torsion subgroup $E(\mathbb{F}_p)[r]$ and in the finite field extension $\mathbb{F}_{p^k}$. In practice, this means $6 \leqslant k \leqslant 60$. More precise recommendations are given in [FST10, Tab. 1]. For a random elliptic curve, we have usually $k \simeq r$ so this is a huge constraint.
2. The trace t of the curve must satisfy $|t| \leqslant 2\sqrt{p}$.

3. The determinant of the curve $\Delta = t^2 - 4p = -Dy^2$ must have a very small square-free part $D < 10^9$ in order to run the CM-method in reasonable time.
4. The size $\log r$ of the subgroup must be close to the optimal case, that is $\rho = g \log p / \log r \sim 1$ with g the genus of the curve. Quite generic methods for elliptic curves achieve $1 \leqslant \rho \leqslant 2$. We will try to find constructions for genus 2 curves with $2 \leqslant \rho \leqslant 4$.

The two methods use the same shortcuts in formulas. Let E an elliptic curve and let $\#E(\mathbb{F}_p) = p + 1 - t = hr$ with r a large prime and h the cofactor. Hence $p \equiv t - 1 \mod r$. Let $\Delta = t^2 - 4p = -Dy^2$. The second useful formula is $Dy^2 = 4p - t^2 = 4hr - (t-2)^2$, hence $-Dy^2 \equiv (t-2)^2 \mod r$.

2.6.1 Cocks-Pinch Method

We first recall the method proposed by Cocks and Pinch in 2001 to construct pairing-friendly elliptic curves [CP01] (see also [BSS05, Algorithm IX.4]):

Algorithm 10: Cocks-Pinch method to find a pairing-friendly elliptic curve.

Input: Square-free integer D, size of r and embedding degree k to match the security level in bits, knowing that $\rho \approx 2$.
Output: Prime order r, prime number p, elliptic curve parameters $a, b \in \mathbb{F}_p$ such that $E(\mathbb{F}_p) : Y^2 = X^3 + aX + b$ has a subgroup of order r and embedding degree k with respect to r.

1 **repeat**
2 Pick at random a prime r of prescribed size until $-D$ is a square in the finite field $\mathbb{F}_r$ and $\mathbb{F}_r$ contains a primitive k-th root of unity ζ_k, that is $r \equiv 1 \mod k$.
3 As r divides $\Phi_k(p)$, we can rewrite it as $\Phi_k(p) \equiv 0 \mod r$. With properties of cyclotomic polynomials, we obtain $p \equiv \zeta_k \mod r$ with ζ_k a primitive k-th root of unity. Furthermore, $t \equiv 1 + p \mod r$ so this method chooses $t = 1 + \zeta_k$ in $\mathbb{F}_r$. Then $y = (t-2)/\sqrt{-D}$ in $\mathbb{F}_r$.
4 Lift t and y from $\mathbb{F}_r$ to $\mathbb{Z}$ and set $p = \frac{1}{4}(t^2 + Dy^2)$.
5 **until** *p is prime.*
6 **return** r, p, $a, b \in \mathbb{F}_p$

We propose to adapt this method to the Jacobian families of cryptographic interest presented above. See the size recommendations in [BBC+11b, Tab. 3.1] depending on the security level in bits to choose accordingly the embedding degree. First, we know explicitly the Jacobian order. Just as in the case of elliptic curves, the definition of the embedding degree is equivalent to ask for $r \mid \#J_{\mathcal{C}}(\mathbb{F}_p)$ and $r \mid \Phi_k(p)$. We will use the property $p \equiv \zeta_k \mod r$ as well. The aim is to express the other parameters, namely the square part y and the trace of the elliptic curve isogenous to the Jacobian over some extension field, in terms of $\zeta_k \mod r$. We will use the same notations as previously, see Th.7 and Th.8. Let i be a primitive fourth root of unity an ω be a primitive third root of unity in $\mathbb{F}_r$.

2.6.1.1 Pairing-friendly Hyperelliptic curve $\mathcal{C}_1$

If b is not a square in $\mathbb{F}_p$ but $\sqrt{b}, \sqrt[4]{b} \in \mathbb{F}_{p^2}$ ($p \equiv 3 \mod 4$), then $\#J_{\mathcal{C}_1}(\mathbb{F}_p) = \#E_1(\mathbb{F}_{p^2}) = p^2 + 1 - t_{p^2}$ (Th.7(2.)). A pairing-friendly Jacobian of this type has exactly the same order as the corresponding elliptic curve $E_1(\mathbb{F}_{p^2})$. Hence any pairing-friendly elliptic curve defined over a quadratic extension $\mathbb{F}_{p^2}$ (and of even order) will provide a pairing-friendly Jacobian of this type over the prime field $\mathbb{F}_p$, with the same order and the same ρ-value. Choosing the Jacobian instead of the elliptic curve will be appropriate only if the group law on the Jacobian over $\mathbb{F}_p$ is faster than the group law on the elliptic curve over $\mathbb{F}_{p^2}$. Note that the methods described in [FST10] are suitable for generating pairing-friendly elliptic curves over prime fields (in large characteristic), not over field extensions.

$\mathcal{C}_1$ with b a square but not a fourth power. This case is already almost solved in [FS11]. The Cocks-Pinch method adapted with $r \mid \#J_{\mathcal{C}_1}(\mathbb{F}_p) = (p-1)^2 + (t'_p)^2$ instead of $r \mid p + 1 - t'_p$ produces indeed the same algorithm as [FS11, Alg. 5.5] followed by [FS11, Alg. 5.11] with $\pi = (t'_p - y\sqrt{-D})/2$, $d = 4$. We

show that $d \mid k$ is unnecessary. It is completely hopeless to expect a prime power $q = \pi\overline{\pi} = p^n$ hence we assume that $q = p$ is prime.

Definition 19. *Embedding degree and embedding field[BCF09, Def. 2.1 and 2.2] Let A be an abelian variety defined over $\mathbb{F}_q$, where $q = p^m$ for some prime p and integer m. Let $r \neq p$ be a prime dividing $\#A(\mathbb{F}_q)$. The* embedding degree *of A with respect to r is the smallest integer k such that r divides $q^k - 1$.*

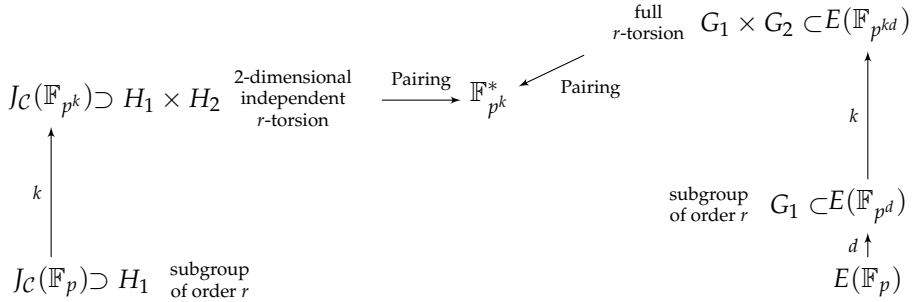
The minimal embedding field *of A with respect to r is the smallest extension of $\mathbb{F}_p$ containing the rth roots of unity $\mu_r \subset \mathbb{F}_p$.*

Let k be the embedding degree of the Jacobian $J_{\mathcal{C}_1}(\mathbb{F}_p)$: $r \mid \#J_{\mathcal{C}_1}(\mathbb{F}_p)$, $r \mid \Phi_k(p)$. From the Jacobian point of view, there is no security problem induced by a difference between embedding degree and embedding field because $\mathbb{F}_p$ is a prime field. From elliptic curve side, the one-dimensional part of the r-torsion arises in $E_1'(\mathbb{F}_{p^4})$, not below. An elementary observation about elliptic curve orders shows that

$$\begin{aligned} \#E_1'(\mathbb{F}_p) &= p + 1 - t_p' \\ \#E_1'(\mathbb{F}_{p^2}) &= (p + 1 - t_p')(p + 1 + t_p') \\ \#E_1'(\mathbb{F}_{p^4}) &= (p + 1 - t_p')(p + 1 + t_p')((p+1)^2 + (t_p')^2) \end{aligned}$$

and the last factor of $\#E_1'(\mathbb{F}_{p^4})$ is the Jacobian order. Hence $r \mid \#E_1'(\mathbb{F}_{p^4})$ but not underneath. The full r-torsion arises in $E_1'(\mathbb{F}_{p^{4k/\gcd(4,k)}})$ but the embedding field is $\mathbb{F}_{p^k}$. So the elliptic curve $E_1'(\mathbb{F}_{p^4})$ will not be suitable for a pairing implementation when $\gcd(k,4) \in \{1,2\}$ which does not matter because we are interested in Jacobians suitable for pairing, not elliptic curves. See Fig. 2.1.

Figure 2.1: Difference between Jacobian and elliptic curve embedding degree



Moreover we note that taking an even trace t_p' and a prime $p \equiv 1 \mod 4$ permits always to find valid parameters, namely a $c \in \mathbb{F}_p$ satisfying the j-invariant equation, hence coefficients $a, b \in \mathbb{F}_p$ of $\mathcal{C}_1$.

$\mathcal{C}_1$ with b not a square and $p \equiv 1 \mod 4$. In this case we have $\sqrt[4]{b} \notin \mathbb{F}_{p^2}$, $\sqrt[4]{b} \in \mathbb{F}_{p^4}$ and $\#J_{\mathcal{C}_1}(\mathbb{F}_p) = p^2 + 1 + 2n^2 - 2n(1+p) = (p-n)^2 + (n-1)^2$ with $2p \pm t_{p^2}' = 2n^2$. The isogenous elliptic curve is defined over $\mathbb{F}_{p^2}$. We have $\Delta = (t_{p^2}')^2 - 4p^2 = (t_{p^2}' + 2p)(t_{p^2}' - 2p)$. With $2p - t_{p^2}' = 2n^2$ we obtain $2p + t_{p^2}' = 4p - 2n^2$ and find $\Delta = -4n^2(2p - n^2)$. With $2p + t_{p^2}' = 2n^2$ we obtain $2p - t_{p^2}' = 4p - 2n^2$ and find also $\Delta = -4n^2(2p - n^2)$. In both cases let $Dy^2 = 2p - n^2$ thus $\Delta = -D(2ny)^2$ and $p = (Dy^2 + n^2)/2$. The Jacobian order is a sum of two squares in p and n hence $n = (p+i)/(1+i) = (p+i)(1-i)/2 \mod r$. Furthermore $y^2 \equiv (2p - n^2)/D \mod r$ with $p \equiv \zeta_k \mod r$ and we find that

$n \equiv (\zeta_k + i)(1-i)/2 \mod r$ and $y \equiv \pm(\zeta_k - i)(1+i)/(2\sqrt{D}) \mod r$.

The trace will be even by construction as $t_{p^2}' = \pm(2p - 2n^2)$ and to find valid parameters, $p \equiv 1 \mod 4$ is required. To find the coefficients of the curve $\mathcal{C}_1(\mathbb{F}_p)$, do the following (Alg. 11).

We adapt the program `cm.cpp` of Miracl[2] [Sco11] to compute the j-invariant of an elliptic curve defined over $\mathbb{F}_{p^2}$ (instead of $\mathbb{F}_p$). Indeed, it is not convenient for step 5 as it searches for an elliptic curve

2. We learned very recently that the MIRACL library status has changed. This library is now a commercial product of Certivox [Cer12]. The CM software [Eng12] can be an even more efficient alternative to compute class polynomials.

Algorithm 11: Pairing-friendly Jacobian of type $J_{\mathcal{C}_1}$, Th.7(3.)

Input: Square-free integer D, size of r and embedding degree k to match the security level in bits, knowing that $\rho \approx 4$.

Output: Prime order r, prime number p, Jacobian parameters $a, b \in \mathbb{F}_p$ such that the Jacobian of the curve $\mathcal{C}_1(\mathbb{F}_p) : Y^2 = X^5 + aX^3 + bX$ has a subgroup of order r and embedding degree k with respect to r.

1 **repeat**
2 Choose a prime r of prescribed size with $i, \sqrt{D}, \zeta_k \in \mathbb{F}_r$.
3 Let $n = (\zeta_k + i)(1 - i)/2$ and $y = \pm(\zeta_k - i)(1 + i)/(2\sqrt{D}) \in \mathbb{F}_r$.
4 Lift n and y from $\mathbb{F}_r$ to $\mathbb{Z}$ and set $p = (n^2 + Dy^2)/2$.
5 **until** $p \equiv 1 \mod 4$ *and* p *is prime.*
6 Run the CM method to find the j-invariant of an elliptic curve $E_1'(\mathbb{F}_{p^2})$ of trace $\pm t'_{p^2}$ and $\Delta = -4D(ny)^2$.
7 Solve $j(E_1') = 2^6 \frac{(3c-10)^3}{(c-2)(c+2)^2}$ in $\mathbb{F}_{p^2}$ and choose the solution satisfying $c^2 \in \mathbb{F}_p$.
8 Choose $a, b \in \mathbb{F}_p$ such that $a \neq 0$ and $b = (a/c)^2$ (b is a square in $\mathbb{F}_{p^2}$ but not in $\mathbb{F}_p$).
9 **return** r, p, $a, b \in \mathbb{F}_p$

defined over a prime field. We isolate parts of the program which compute the Weber polynomial of a number field of discriminant D. Then we call the `factor` function but to find a factor $\mod p$ of degree 2 (instead of degree 1) of the Weber polynomial when $D \not\equiv 3 \mod 8$ and a factor of degree 6 (instead of degree 3) when $D \equiv 3 \mod 8$. The papers [KSZ07, KKSZ10] contain efficient formulas to recover Hilbert polynomial roots in $\mathbb{F}_p$ from Weber polynomial roots in $\mathbb{F}_p$ or $\mathbb{F}_{p^3}$. We find in $\mathbb{F}_{p^2}$ or $\mathbb{F}_{p^6}$ a root of the factor of degree 2 or 6 of Weber polynomial and apply the corresponding transformation to get an element in $\mathbb{F}_{p^2}$. We obtain the j-invariant of (an isogenous curve to) the curve $E_1'(\mathbb{F}_{p^2})$. We solve $j(E_1') = 2^6 \frac{(3c-10)^3}{(c-2)(c+2)^2}$ and find for various examples a solution $c \in \mathbb{F}_{p^2}$ satisfying $c^2 \in \mathbb{F}_p$. It comes from the appropriate restrictions $2p \pm t'_{p^2} = 2n^2$, $p \equiv 1 \mod 4$, n odd. Sometimes we have to choose a quadratic twist of $\mathcal{C}_1$, of the form $Y^2 = \nu(X^5 + aX^3 + bX)$ with $\nu \in \mathbb{F}_p$ non-square.

Example 15. $k = 6$, $D = 516505$, $\rho = 4.1$

p = `0x9d3e97371e27d006f11762f0d56b4fbf2caca7d606e92e8b6f35189723f46f57ed46e9650ce1cca1bd90dc393db35cc38970cb0abbe236bf2c4ac2f65f1b50afb135` *(528 bits)*,
r = `0x679d8c817e0401203364615b9d34bdb3a0b89e70fa8d6807fa646e25140f25ad` *(255 bits)*,
n = `0x28f34a88ab9271c2ea6d70f4a3dc758a025ad6e4ee51c16867763e8d940022de5`,
y = $-$`0x65110defe8f4669a158149675afaa23dba326d49ce841d7ef9855c7d8a65df95`,
a = 1,
b = `0x85eb6f5b5594c1bca596a53066216ad79588cf39984314609bbd7a3a3022 41fc786703a19bc1ccb44fc9e09b9c17ac62fc38d6bf82851d3d8b753c79da7338ca56b0`,
$\mathcal{C}_1(\mathbb{F}_p) : Y^2 = 2(X^5 + aX^3 + bX)$.

2.6.1.2 Pairing-friendly Hyperelliptic curve $\mathcal{C}_2$

If b is a cube but not a square then $\#J_{\mathcal{C}_2}(\mathbb{F}_p) = p^2 + 1 - t_{p^2}$ (Th.8(3.)). This case is close to the elliptic curve case. Actually, this is the same construction as finding a pairing-friendly elliptic curve over a field $\mathbb{F}_{p^2}$. But in practice the methods to find such pairing-friendly elliptic curves over $\mathbb{F}_p$ fail over $\mathbb{F}_{p^2}$. Indeed, the expression for p is $p^2 = \frac{1}{4}((t'_{p^2})^2 + Dy^2)$ but this is hopeless to find a prime square. We did not find in the literature any such construction.

$\mathcal{C}_2$ with b a square but not a cube. This case is treated in [FS11, Alg. 5.5, Alg. 5.11] and corresponds to $d = 3$ and $\pi = (t_p - y\sqrt{-D})/2$. This is also a Cocks-Pinch-like method with $r \mid p^2 - p + 1 + (1 + p)t_p + (t_p)^2$ and $r \mid \Phi_k(p)$. As above for $\mathcal{C}_1$, the condition "$3 \mid k$" is not necessary since we consider the embedding degree of the Jacobian, not the elliptic curve.

We found that $p \equiv 1 \mod 3$ and $p+1 \pm t_p \equiv 0 \mod 3$ are enough to find always valid parameters. Freeman and Satoh pointed out that the equation $j(E_c) = 2^8 3^3 (2c-5)^3/((c-2)(c+2)^3)$ has a solution in $\mathbb{F}_p$ in only one third of the cases [FS11, § 6]. One can explain this phenomenon by simple arithmetic considerations.

The elliptic curve E_c has a 3-torsion point which means $p+1-t_p \equiv 0 \mod 3$, which happens one third of the cases when $p \equiv 1 \bmod 3$. Assuming that $p \equiv 1 \mod 3$, if $p+1+t_p \equiv 0 \mod 3$ then $E_c(\mathbb{F}_p)$ has not 3-torsion point but its quadratic twist has. These two elliptic curves have the same j-invariant and admit a 3-torsion subgroup over $\mathbb{F}_{p^2}$. In practice we verify that the equation has a solution when $p+1 \pm t_p \equiv 0 \mod 3$. Combining the two conditions $p \equiv 1 \mod 3$ and $p+1 \pm t_p \equiv 0 \mod 3$, the equation from $j(E_c)$ has indeed a solution one third of the time ($\frac{1}{2} \cdot \frac{2}{3}$). When $p \equiv 1 \mod 3$ and $t_p \equiv 2 \bmod 3$, we can always find a solution in step 2 of [FS11, Alg. 5.11] and finish to run this algorithm. When $p \equiv 1 \mod 3$ and $t_p \equiv 1 \mod 3$, we can still find a solution in step 2 and construct the coefficients of $\mathcal{C}_2(\mathbb{F}_p)$ in step 3 of [FS11, Alg. 5.11]. But in step 6, we have to choose not $\mathcal{C}_2$ itself but its quadratic twist.

$\mathcal{C}_2$ with b neither a square nor a cube. $\#J_{\mathcal{C}_2}(\mathbb{F}_p) = p^2+p+1-(p+1)3n+3n^2$. Here the parameters satisfy $2p - t_{p^2} = 3n^2$. Let $2p + t_{p^2} (= 4p - 3n^2) = Dy^2$. Hence

$$p = \frac{1}{4}\left(3n^2 + Dy^2\right) .$$

Note that $3 \nmid D$ otherwise p would not be prime. Solving $p^2+p+1-(p+1)3n+3n^2 \equiv 0 \mod r$ gives $p = (1-\omega^2)n + \omega^2$ or $p = (1-\omega)n + \omega$ with ω a primitive third root of unity. As $y^2 = (4p-3n^2)/D \bmod r$ and with $p \equiv \zeta_k \mod r$ we find

$$n \equiv (\zeta_k - \omega)/(1-\omega) \mod r \text{ and } y \equiv \pm(\omega\zeta_k + \omega^2)/\sqrt{D} \mod r .$$

The last version of the Cocks-Pinch method is presented in Alg. 12.

Algorithm 12: Pairing-friendly Jacobian of type $J_{\mathcal{C}_2}$, Th.8(4.)

Input: Square-free integer D, $3 \nmid D$, size of r and embedding degree k to match the security level in bits, knowing that $\rho \approx 4$.

Output: Prime order r, prime number p, Jacobian parameters $a, b \in \mathbb{F}_p$ such that the Jacobian of the curve $\mathcal{C}_2(\mathbb{F}_p) : Y^2 = X^6 + aX^3 + b$ has a subgroup of order r and embedding degree k with respect to r.

1 **repeat**
2 Choose a prime r of prescribed size such that a third root of unity ω, $\sqrt{D}$ and $\zeta_k \in \mathbb{F}_r$.
3 Let $n = (\zeta_k - \omega)/(1-\omega)$ and $y = \pm(\omega\zeta_k + \omega^2)/\sqrt{D} \in \mathbb{F}_r$.
4 Lift n and y from $\mathbb{F}_r$ to $\mathbb{Z}$ and set $p = (3n^2 + Dy^2)/4$.
5 **until** $p \equiv 1 \mod 3$ *and p is prime.*
6 Run the CM method to find the j-invariant of an elliptic curve $E_c(\mathbb{F}_{p^2})$ of trace t_{p^2} and $\Delta = -3D(ny)^2$. More precisely, run the CM method with $3D$. Find a degree 2 or 6 factor of the Weber polynomial $\bmod p$, then apply the right transformation from [KSZ07, KKSZ10] to obtain a root in $\mathbb{F}_{p^2}$ of the corresponding Hilbert polynomial.
7 Solve $j(E_c) = 2^8 3^3 \frac{(2c-5)^3}{(c-2)(c+2)^3}$ in $\mathbb{F}_{p^2}$ and choose a solution $c \in \mathbb{F}_{p^2}$ such that $c^2 \in \mathbb{F}_p$. Choose $a, b \in \mathbb{F}_p$ such that $(a/c)^2$ is not a cube and $b = (a/c)^2$. Hence b is neither a square nor a cube.
8 **return** r, p, $a, b \in \mathbb{F}_p$

2.6.2 Brezing-Weng Method

The method proposed by Brezing-Weng is to use a polynomial ring built with a cyclotomic polynomial instead of a finite prime field $\mathbb{F}_r$. The parameters will be polynomials modulo a cyclotomic polynomial instead of integers modulo a prime. But the choice of D is limited to few values. We tried with D square-free in the range 1 - 35 according to the embedding degree $5 \leqslant k \leqslant 36$. We ran a search (with Magma [BCP97]) over different cyclotomic fields and with a change of basis as in [KSS08] and [Kac10].

We obtained complete families with $\rho \simeq 3$ and recover constructions already mentioned in previous papers [KT08, FS11] and new complete families for other embedding degrees:

Example 16. $k = 22, D = 2, \rho = 2.8$

$$\begin{aligned} r &= \Phi_{88}(x) = x^{40} - x^{36} + x^{32} - x^{28} + x^{24} - x^{20} + x^{16} - x^{12} + x^8 - x^4 + 1 \\ n &= \tfrac{1}{2}\left(x^{28} - x^{22} - x^6 + 1\right) \\ y &= \tfrac{1}{2}\left(x^{17} + x^{11}\right) \\ t'_{p^2} &= \tfrac{1}{4}\left(-x^{56} + 2x^{50} - x^{44} + 4x^{34} + 4x^{22} - x^{12} + 2x^6 - 1\right) \\ p &= \tfrac{1}{8}\left(x^{56} - 2x^{50} + x^{44} + 8x^{28} + x^{12} - 2x^6 + 1\right) \\ x &\equiv 1 \mod 2 \end{aligned}$$

Example 17. $k = 26, D = 2, \rho = 2.33$

$$\begin{aligned} r &= \Phi_{104}(x) = x^{48} - x^{44} + x^{40} - x^{36} + x^{32} - x^{28} + x^{24} - x^{20} + x^{16} - x^{12} + x^8 - x^4 + 1 \\ n &= \tfrac{1}{2}\left(x^{28} - x^{26} - x^2 + 1\right) \\ y &= \tfrac{1}{2}\left(x^{15} + x^{13}\right) \\ t'_{p^2} &= \tfrac{1}{4}\left(-x^{56} + 2x^{54} - x^{52} + 4x^{30} + 4x^{26} - x^4 + 2x^2 - 1\right) \\ p &= \tfrac{1}{8}\left(x^{56} - 2x^{54} + x^{52} + 8x^{28} + x^4 - 2x^2 + 1\right) \\ x &\equiv 1 \mod 2 \end{aligned}$$

Some constructions ($k \in \{7, 17, 19, 23, 29, 31\}$) have a cyclotomic polynomial of too high degree for r. Hence there are very few possibilities for choosing a suitable integer x such that $p(x)$ and $r(x)$ are prime and of the desired size. Moreover the ρ-value is close to 4. It would be preferable to use the Cocks-Pinch-like method.

2.6.3 More Pairing-Friendly constructions with $D = 1, 2, 3$

We observed that when $D = 1$, the obtained genus 2 hyperelliptic curve of the form $\mathcal{C}_1(\mathbb{F}_p)$ with b a square splits actually into two non-isogenous elliptic curves over $\mathbb{F}_p$. We observed the same decomposition for genus 2 hyperelliptic curve of the form $\mathcal{C}_2$ obtained with $D = 3$ and b a square but not a cube. A theoretical explanation can be found in [FS11, Proposition 3.10]. From Th. 7 2 we get the explicit decomposition. We give here a practical point of view from explicit zeta function computation. Let $E_1(\mathbb{F}_q)$ be an elliptic curve defined over a finite field $\mathbb{F}_q$ of trace t_q an satisfying $(t_q)^2 - 4q = -y^2$, *i.e.* $D = 1$. The zeta function of E_1 is $Z_{E_1}(T, \mathbb{F}_q) = T^2 - t_q T + q = (T - \frac{t_q + iy}{2})(T - \frac{t_q - iy}{2})$ with $i \in \mathbb{C}$ such that $i^2 = -1$. We will use the notation $\alpha = \frac{t_q + iy}{2}$. With the formula given in [FS11, Proposition 3.4] we find that the zeta function of the order 4 Weil restriction of $E_1(\mathbb{F}_q)$ is

$$Z_{J_{\mathcal{C}_1}}(T, \mathbb{F}_q) = (T - i\alpha)(T + i\alpha)(T - i\overline{\alpha})(T + i\overline{\alpha}) = (T^2 - yT + q)(T^2 + yT + q)\,.$$

Note that $q + 1 - y$ and $q + 1 + y$ are the orders of the two quartic twists of $E_1(\mathbb{F}_q)$. Hence the obtained Jacobian always splits into the two quartic twists of $E_1(\mathbb{F}_q)$.

For $J_{\mathcal{C}_2}(\mathbb{F}_q)$ and $D = 3$ when b is a square but not a cube, a similar computation explains the matter. Here E_c is an elliptic curve defined over $\mathbb{F}_q$ of trace t_q and such that $(t_q)^2 - 4q = -3y^2$. Let us denote $\alpha = \frac{t_q + i\sqrt{3}y}{2}$ one of the two roots of its zeta function. The zeta function of the order 3 Weil restriction of $E_c(\mathbb{F}_q)$ is

$$Z_{J_{\mathcal{C}_2}}(T, \mathbb{F}_q) = \left(T^2 + \tfrac{t+3y}{2}T + q\right)\left(T^2 + \tfrac{t-3y}{2}T + q\right)\,.$$

We recognize the two cubic twists of $E_c(\mathbb{F}_q)$. Trying with an order 6 Weil restriction, we find

$$Z_{J_{\mathcal{C}_2}}(T, \mathbb{F}_q) = \left(T^2 - \tfrac{t-3y}{2}T + q\right)\left(T^2 - \tfrac{t+3y}{2}T + q\right)\,.$$

Hence the Jacobian splits into the two sextic twists of $E_c(\mathbb{F}_q)$. Freeman and Satoh suggested to construct an order 8 Weil restriction when $D = 1, 2$ and an order 12 Weil restriction when $D = 3$. For $k = 32, 64, 88$ and $D = 2$ this order 8 Weil restriction corresponds to families previously found by Kawazoe and Takahashi.

2.6.3.1 Order-8 Weil restriction when D = 1

Let $E(\mathbb{F}_p)$ an elliptic curve defined over a prime field $\mathbb{F}_p$, of trace t_p and satisfying $(t_p)^2 - 4p = -y^2$ (that is, $D = 1$). The two roots of its zeta function over $\mathbb{C}$ are $\alpha = (t_p + iy)/2$ and $\overline{\alpha}$. Let ζ_8 denotes an eighth root of unity. The zeta function of the order 8 Weil restriction of $E(\mathbb{F}_p)$ is

$$\begin{aligned} Z(T,\mathbb{F}_p) &= \left((T-\zeta_8\alpha)(T-\zeta_8^7\overline{\alpha})(T-\zeta_8^5\alpha)(T-\zeta_8^3\overline{\alpha})\right)\left((T-\zeta_8^3\alpha)(T-\zeta_8^5\overline{\alpha})(T-\zeta_8^7\alpha)(T-\zeta_8\overline{\alpha})\right) \\ &= (T^4+tyT^2+p^2)(T^4-tyT^2+p^2) \end{aligned}$$

We see this zeta function factors as two degree 4 zeta functions, that is into two genus 2 hyperelliptic curve zeta functions. So we start from an elliptic curve $E(\mathbb{F}_p)$ as above, with $(t_p)^2 - 4p = -y^2$ and search for suitable p, t, y such that there exists a genus 2 hyperelliptic curve of order $\#J_{\mathcal{C}}(\mathbb{F}_p) = p^2 + 1 \pm ty$ suitable for pairing-based cryptography.

To apply one of the two previous methods (Cocks-Pinch or Brezing-Weng), we have to find an expression of t and y in terms of p modulo r.

$$t = \zeta_8 + \zeta_8^7\zeta_k \text{ and } y = -\zeta_8^7 - \zeta_8\zeta_k \mod r\,.$$

To finish, $p = (t^2 + y^2)/4$.

Example 18. $k = 8, D = 1, \rho = 3.0$

$$\begin{aligned} r &= x^4 + 2x^2 + 4x + 2 \\ t &= x \\ y &= \tfrac{1}{3}(-x^3 + 2x^2 - 3x + 2) \\ p &= \tfrac{1}{36}(x^6 - 4x^5 + 10x^4 - 16x^3 + 26x^2 - 12x + 4) \\ x &\equiv 4 \mod 6 \end{aligned}$$

2.6.3.2 Order-8 Weil restriction when D = 2

Let $E(\mathbb{F}_p)$ an elliptic curve defined over a prime field $\mathbb{F}_p$, of trace t_p and satisfying $(t_p)^2 - 4p = -2y^2$ (that is, $D = 2$). The two roots of its zeta function over $\mathbb{C}$ are $\alpha = (t_p + i\sqrt{2}y)/2$ and $\overline{\alpha}$. Let ζ_8 denotes an eighth root of unity. The zeta function of the order 8 Weil restriction of $E(\mathbb{F}_p)$ is

$$\begin{aligned} Z(T,\mathbb{F}_p) &= \left((T-\zeta_8\alpha)(T-\zeta_8^7\overline{\alpha})(T-\zeta_8^3\alpha)(T-\zeta_8^5\overline{\alpha})\right)\left((T-\zeta_8^5\alpha)(T-\zeta_8^3\overline{\alpha})(T-\zeta_8^7\alpha)(T-\zeta_8\overline{\alpha})\right) \\ &= (T^4-2yT^3+2y^2T^2-2ypT+p^2)(T^4+2yT^3+2y^2T^2+2ypT+p^2) \end{aligned}$$

and $\#J_{\mathcal{C}}(\mathbb{F}_p) = p^2 + 1 - 2yp + 2y^2 - 2y = (p-y)^2 + (y-1)^2$. We recognize the order of $J_{\mathcal{C}_1}(\mathbb{F}_p)$ when the considered isogeny is defined over $\mathbb{F}_{p^4}$ (and with n and y swapped). Hence it is the construction detailed above in Alg. 11 with $D = 2$.

2.6.3.3 Order-12 Weil restriction when D = 3

Let $E(\mathbb{F}_p)$ an elliptic curve defined over a prime field $\mathbb{F}_p$, of trace t_p and satisfying $(t_p)^2 - 4p = -3y^2$ (i.e. $D = 3$). The two roots of its zeta function over $\mathbb{C}$ are $\alpha = (t_p + i\sqrt{3}y)/2$ and $\overline{\alpha}$. Let ζ_{12} denotes a twelfth root of unity. The zeta function of the order 12 Weil restriction of $E(\mathbb{F}_p)$ is

$$\begin{aligned} Z(T,\mathbb{F}_p) &= \left((T-\zeta_{12}\alpha)(T-\zeta_{12}^{11}\overline{\alpha})(T-\zeta_{12}^7\alpha)(T-\zeta_{12}^5\overline{\alpha})\right)\left((T-\zeta_{12}^5\alpha)(T-\zeta_{12}^7\overline{\alpha})(T-\zeta_{12}^{11}\alpha)(T-\zeta_{12}\overline{\alpha})\right) \\ &= \left(T^4 - \left(-p + t_p\tfrac{t_p+3y}{2}\right)T^2 + p^2\right)\left(T^4 - \left(-p + t_p\tfrac{t_p-3y}{2}\right)T^2 + p^2\right) \end{aligned}$$

which can be interpreted as the zeta functions of two Jacobians of hyperelliptic curves defined over $\mathbb{F}_p$ of order $p^2 + p + 1 - t_p(t_p \pm 3y)/2$. For further simplifications, we can also write $\#J_{\mathcal{C}}(\mathbb{F}_p) = (p-1)^2 + ((t_p - 3y)/2)^2 = (p+1)^2 - 3((t_p + y)/2)^2$.

To apply the Cocks-Pinch or Brezing-Weng method, we use

$$t_p \equiv -\omega(\omega p - 1)/i \mod r,\ y \equiv -\omega(\omega p + 1)/\sqrt{3} \mod r$$

with ω a third root of unity and i a fourth root of unity. We found new families with $\rho = 3$ (with Brezing-Weng method). It would be interesting to know if these quite special curves provide more features such as compression due to twists of higher degree.

2.7 Conclusion

In this Chapter we studied widely two families of genus 2 hyperelliptic curves of the form $Y^2 = X^5 + aX^3 + bX$ and $Y^2 = X^6 + aX^3 + b$ (with $a, b \in \mathbb{F}_q^*$). These curves are isogenous over a small degree extension field to the product of two isogenous elliptic curves. We first computed these isogenies between Jacobian and product of two elliptic curves. We then provided explicit formulas for the zeta function of the Jacobians. We derived our formulas from careful decomposition of the zeta function from the extension field where the isogeny is defined to the base field where the Jacobian is defined.

We also presented several algorithms to obtain pairing-friendly hyperelliptic curves families. The constructions require to run the CM method to find a j-invariant in $\mathbb{F}_{p^2}$. We explained the differences with a j-invariant in $\mathbb{F}_p$ and gave references to fill the gap. It is worth noting that it is also possible to adapt the Dupont-Enge-Morain technique [DEM05] to our setting but unfortunately it provides curves with $\rho \simeq 4$. It remains open to construct pairing-friendly hyperelliptic curves with $1 \leqslant \rho < 2$.

Our work is also about efficient scalar multiplication on these genus 1 and 2 curves with a 4-dimensional GLV technique. We proposed for this purpose the construction of two independent endomorphisms both on the Jacobians (defined over a field $\mathbb{F}_q$) and on the isogenous elliptic curves when they are defined over a quadratic extension of the field $\mathbb{F}_q$. Surprisingly, Smith [Smi13] studied at the same time from a different point of view the same two families of elliptic curves defined over quadratic extension of finite fields. Smith observed that one can choose a prime p relevant for fast modular reduction, then build such a curve over the field $\mathbb{F}_{p^2}$ while still having an endomorphism on the curve, together with the fastest possible finite field arithmetic. Smith proposed these curves for 2-dimensional GLV technique combined with optimal finite field arithmetic. These two different applications of these families of curves seems to be roughly equivalent in terms of performances. It would be interesting to investigate the running time of these two methods and compare them with other popular elliptic curves such as Edwards or Huff curves. Concerning genus 2 curves with 4-dimensional GLV scalar multiplication, it would be interesting to apply the methods in [FHLS14] for protected scalar multiplications.

Chapter 3

Pairing implementation on elliptic curves and application to protocols

In this chapter we present the different state-of-the-art implementations of pairings developed for the cryptographic library of Thales Communications & Security and their use in protocols. We explain in Sec. 3.1 the the library structure and the finite-field extension arithmetic. We explain in Sec. 3.2 our optimized implementation of an ate and an optimal ate pairing on a Barreto-Naehrig curve. In Sec. 3.3 we investigate pairings on composite-order elliptic curves. These pairings are used in protocols since 2005. They provide useful additional properties but they are much slower. These pairings need special curves and dedicated pairing computation. We present the first implementation and benchmarks of such pairings. We then chose two protocols based on such pairings and present timings. These results were presented at the *ACNS'2013* conference ([Gui13]). Our efficient pairing of Sec. 3.2 is used in Sec. 3.4 to develop a prototype of a broadcasting scheme. Indeed the pairing development for Thales is part of an ANR project on efficient broadcast protocols. We present the performances we obtained and show that the chosen broadcast scheme is practical and almost ready for industrial use. The results were presented at the *Pairing'2012* conference [DGB12].

3.1 The LIBCRYPTOLCH

The library of the *Laboratoire Chiffre (LCH)* is called LibCryptoLCH. It is written in C. It contains a civil part which contains the contributions of this PhD thesis. The organization of the library is sketched in Sec. 3.1.1. Then in Sec. 3.1.2 and 3.1.3 we explain how we designed finite-field arithmetic. This will be needed for the pairing computation described in Sec. 3.2.

3.1.1 Organization of the LIBCRYPTOLCH

The library is organized in modules, as shown in Fig. 3.1 and 3.2. We present in Fig. 3.1 the main modules on top of which the pairing module was developed. The modules are continually improved. In 2011 the `Modular` package was highly improved thanks to the work of F. de Portzamparc. At the moment, the modular multiplication is written in assembly language for Sparc, ARM (work of Dubois) and Intel x86-64 processors (work of F. de Portzamparc). The multiplication is almost 3 times faster in assembly language compared to pure C language function. The x86-64 code is relevant for common PC and the ARM implementation becomes very interesting at the moment for smartphones such as Samsung with `armeabi` architecture. A work in progress is to adapt the library to such smartphones and activate the ARM parts of the code to speed-up the pairings on such platforms. This is possible since December 2012 and the release of Android rd8 version of the development toolkit. The package `Modular` is generic and valid for any modulus. In particular, this package is not optimized for a sparse prime number with fast modular reduction such as $p = 2^{127} - 1$ or $p = 2^{255} - 19$. Though, we obtain acceptable performances thanks to the assembly code.

The module `EllipticQuad` is a duplicate of the module `Elliptic` in order to provide arithmetic of elliptic curves defined over quadratic extension fields. It uses Jacobian coordinates. This arithmetic is used

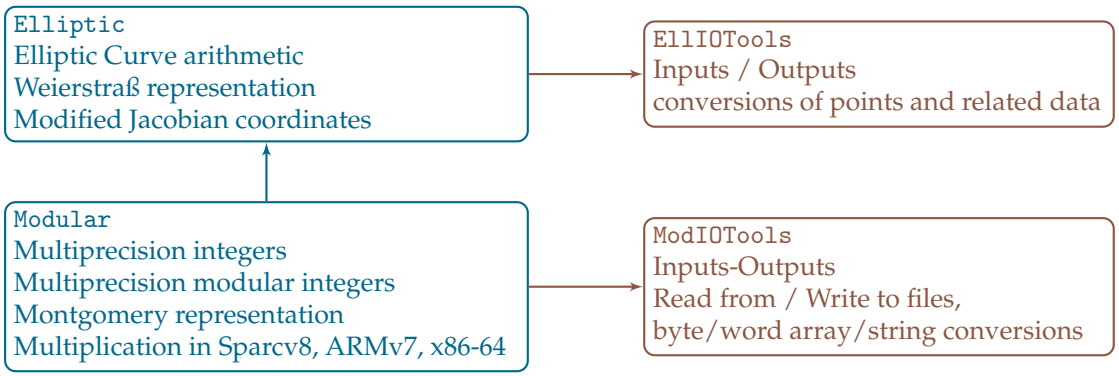


Figure 3.1: Important modules of the LibCryptoLCH, used for the pairing implementations

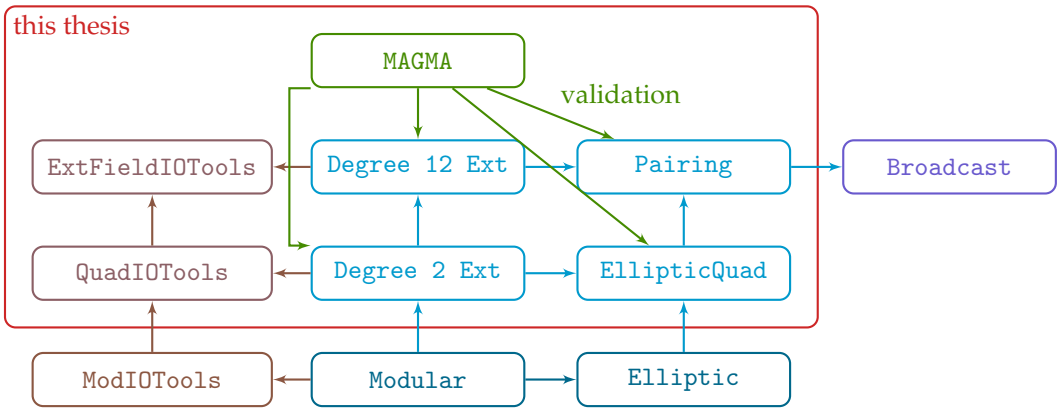


Figure 3.2: Organization of the packages developed during this PhD (circled in red)

to perform the operations in $\mathbb{G}_2$ for a pairing on a Barreto-Naehrig curve. This package is not directly used for pairing computation but is needed for any protocol using BN curves. The modules of extension fields $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^{12}}$ are the two essential building blocks for the pairing package. The module `Quadratic` was developed in internship in 2010. The module `ExtField` was started at the end of internship and continually improved along this PhD. The arithmetic for extension fields is based on the work of Devegili, Ó hÉigeartaigh, Scott and Dahab [DhSD06a]. They studied and compared the efficiency of various formulas for multiplication and squaring in finite-field extensions of degree a multiple of 2 and 3. Based on their results and recommendations, we designed very efficient arithmetic for $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^{12}}$ extension fields. We explain our arithmetic for degree-2 extensions in Sec. 3.1.2 and for degree-3 and 6 extensions in Sec. 3.1.3.

In addition, we need an efficient inversion function. We expose a well-known formula for efficient inversion in finite-field extensions based on a norm computation. We first recall the definition of the norm.

Definition 20. *[LN97, Def. 2.27 §2.3] Let $\mathbb{F}_q$ be a finite field and $\mathbb{F}_{q^m}$ an extension of $\mathbb{F}_q$. For $a \in \mathbb{F}_{q^m}$, the norm $\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a)$ of a over $\mathbb{F}_q$ is defined by*

$$\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a) = a \cdot a^q \cdots a^{q^{m-1}} = a^{(q^m-1)/(q-1)}\,. \tag{3.1}$$

Moreover we have this useful theorem.

Theorem 12. *[LN97, Th. 2.28 §2.3] The norm function $\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ satisfies the following properties:*

1. $\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(ab) = \mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a) \cdot \mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(b)$ *for all* $a, b \in \mathbb{F}_{q^m}$;
2. $\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ *maps* $\mathbb{F}_{q^m}$ *onto* $\mathbb{F}_q$;
3. $\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a) = a^m$ *for all* $a \in \mathbb{F}_q$;
4. $\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a^q) = \mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a)$ *for all* $a \in \mathbb{F}_{q^m}$.

Finally the norm is transitive:

Theorem 13. *[LN97, Th. 2.29 §2.3] Let $\mathbb{F}_q$ be a finite field, let $\mathbb{F}_{q^m}$ be an extension of $\mathbb{F}_q$ and let $\mathbb{F}_{q^{m\cdot n}}$ be an extension of $\mathbb{F}_{q^m}$. Then*

$$\mathrm{Norm}_{\mathbb{F}_{q^{m\cdot n}}/\mathbb{F}_q}(a) = \mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}\left(\mathrm{Norm}_{\mathbb{F}_{q^{m\cdot n}}/\mathbb{F}_{q^m}}(a)\right) \tag{3.2}$$

for all $a \in \mathbb{F}_{q^{m\cdot n}}$.

To invert an element $a \in \mathbb{F}_{q^m}$ we need an efficient method. Computing naively $a^{-1} = a^{q^m-2}$ is very costly since $q^m - 2$ is a large exponent. We use the following formula:

$$a^{-1} = \frac{a^{q+q^2+\ldots+q^{m-1}}}{a^{1+q+q^2+\ldots+q^{m-1}}} = \frac{1}{\mathrm{Norm}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a)}\left(a^q a^{q^2}\cdots a^{q^{m-1}}\right) \tag{3.3}$$

performed with a norm computation, one inversion in $\mathbb{F}_q$, several Frobenius and $m-2$ multiplications. The formula can be specifically optimized for any given degree m extension.

3.1.2 Quadratic extension field

The prime finite field we will denote by $\mathbb{F}_p$ is simply implemented with the `Modular` package. The modulus is set to p. The quadratic extension is built as $\mathbb{F}_{p^2}[X]/(X^2-\alpha)$ with α a *tiny* non-residue in $\mathbb{F}_p$. If $p \equiv 3 \bmod 4$ we set $\alpha = -1$, otherwise we choose a small non-residue such as $2, 3, \ldots$ allowing fast reduction modulo the irreducible polynomial $X^2 - \alpha$. An element in $\mathbb{F}_{p^2}$ is represented as a vector of two coefficients in $\mathbb{F}_p$: $a = a_0 + a_1 X$ with $a_0, a_1 \in \mathbb{F}_p$. The reduction is

$$a_0 + a_1 X + a_2 X^2 \equiv (a_0 + \alpha a_2) + a_1 X\,.$$

If $\alpha = 2$ for example, the reduction costs two additions: $a_0 + \alpha a_2 = a_0 + a_2 + a_2$. The addition and subtraction are performed coefficient-wise. The well-known formula of Karatsuba is used for multiplication. The squaring is performed with the Complex method advised in [DhSD06a, Tab. 2 and Tab. 16].

$$a = a_0 + a_1 X,\ b = b_0 + b_1 X,\ r = r_0 + r_1 X$$

Multiplication: Karatsuba – 2		
r	$=$	$a \cdot b$
v_0	$=$	$a_0 b_0$
v_1	$=$	$a_1 b_1$
r_0	$=$	$v_0 + v_1\alpha$
r_1	$=$	$(a_0+a_1)(b_0+b_1) - v_0 - v_1$
$3M_p + 5Add_p + 1M_\alpha$		

(3.4)

Squaring: Complex, $\alpha = -1$		
r	$=$	a^2
v_0	$=$	$a_0 a_1$
r_0	$=$	$(a_0+a_1)(a_0-a_1)$
r_1	$=$	$2v_0$
$2M_p + 3Add_p$		

Squaring: Complex-like, $\alpha \neq -1$		
r	$=$	a^2
v_0	$=$	$a_0 a_1$
r_0	$=$	$(a_0+a_1)(a_0+a_1\alpha) - (v_0 + v_0\alpha)$
r_1	$=$	$2v_0$
$2M_p + 5Add_p + 2M_\alpha$		

We now present the formulas to compute Frobenius, norm and inversion in $\mathbb{F}_{p^2}$. The Frobenius map is almost free in a quadratic extension. It is computed as $a^p = a_0 - a_1 X$ from $a = a_0 + a_1 X$. This costs only one subtraction. The norm is a map from $\mathbb{F}_{p^2}$ to $\mathbb{F}_p$. The inversion needs these two previous operations.

The inversion is computed as $a^{-1} = \frac{a^p}{a^{p+1}} = a^p / \text{Norm}_{\mathbb{F}_{p^2}/\mathbb{F}_p}(a)$.

$$a = a_0 + a_1 X, r = r_0 + r_1 X$$

Frobenius			$\text{Norm}_{\mathbb{F}_{p^2}/\mathbb{F}_p}$			Inversion		
r	$=$	a^p	r_0	$=$	$a \cdot a^p \in \mathbb{F}_p$	r	$=$	a^{-1}
r_0	$=$	a_0	r_0	$=$	$a_0^2 - a_1^2\alpha$	v_0	$=$	$\text{Norm}_{\mathbb{F}_{p^2}/\mathbb{F}_p}(a)$
r_1	$=$	$-a_1 = p - a_1$				v_1	$=$	v_0^{-1}
						r_0	$=$	$v_0 \cdot a_0$
						r_1	$=$	$-v_0 \cdot a_1 = p - a_1$
$1Add_p$			$2S_p + Add_p + 1M_\alpha$			$I_p + 2M_p + 2S_p + 2Add_p + 1M_\alpha$		

(3.5)

3.1.3 Degree 6 extension field

A Barreto-Naehrig curve has embedding degree $k = 12$. The pairing value is an element in a subgroup of a finite field extension of degree 12. The degree 6 twist of $E(\mathbb{F}_{p^{12}})$ allow elements in $\mathbb{G}_2$ to have their coefficients in $\mathbb{F}_{p^2}$. In this pairing context, we have chosen to build $\mathbb{F}_{p^{12}}$ as a degree 6 extension on top of a degree 2 extension. The package `ExtField` implements this degree 6 extension in top of the package `Quadratic`. In a later version of the LibCryptoLCH, a generic structure and package will be used to construct any degree 2 or 3 extension in top of any similar extension. This is a work in progress. This is planned to be finished for December 2013.

The degree 6 extension is a combination of a degree 2 extension on top of a degree 3 extension. We will use these notations.

$$\begin{array}{rcl} \mathbb{F}_{p^{12}} & \simeq & \mathbb{F}_{p^2}[U]/(U^6 - \beta) \\ {\scriptstyle 6}\,| & & \\ \mathbb{F}_{p^2} & \simeq & \mathbb{F}_p[X]/(X^2 - \alpha) \\ {\scriptstyle 2}\,| & & \\ \mathbb{F}_p & & \end{array} \quad \text{or} \quad \begin{array}{rcl} \mathbb{F}_{p^{12}} & \simeq & \mathbb{F}_{p^6}[Z]/(Z^2 - Y) \\ {\scriptstyle 2}\,| & & \\ \mathbb{F}_{p^6} & \simeq & \mathbb{F}_{p^2}[Y]/(Y^3 - \beta) \\ {\scriptstyle 3}\,| & & \\ \mathbb{F}_{p^2} & \simeq & \mathbb{F}_p[X]/(X^2 - \alpha) \\ {\scriptstyle 2}\,| & & \\ \mathbb{F}_p & & \end{array} \tag{3.6}$$

The polynomial $X^2 - \alpha$ is irreducible over $\mathbb{F}_p$, the polynomials $Y^3 - \beta$ and $U^6 - \beta$ are irreducible over $\mathbb{F}_{p^2}$ (this is the same $\beta \in \mathbb{F}_{p^2}$) and $Z^2 - Y$ is irreducible in $\mathbb{F}_{p^6}$. The correspondence from a representation to another one is the following. We represent an element in $\mathbb{F}_{p^{12}}$ as

$$u = u_0 + u_1 U + u_2 U^2 + u_3 U^3 + u_4 U^4 + u_5 U^5 \in \mathbb{F}_{p^2}[U]/(U^6 - \beta), u_i \in \mathbb{F}_{p^2}\ .$$

With $Y = U^2$ and $Z = U$, $u \in \mathbb{F}_{p^6}[Z]/(Z^2 - Y)$ is also

$$u = v_0 + v_1 Z \text{ with } v_0 = u_0 + u_2 Y + u_4 Y^2 \in \mathbb{F}_{p^6}, u_i \in \mathbb{F}_{p^2} \text{ and } v_1 = u_1 + u_3 Y + u_5 Y^2 \in \mathbb{F}_{p^6}, u_i \in \mathbb{F}_{p^2}\ .$$

If $q \equiv 1 \bmod 3$ then we can build a degree 3 extension with a binomial of the form $Y^3 - \beta$. Since $q = p^2$ we actually have $q \equiv 1 \bmod 3$ and we build $\mathbb{F}_{q^3} \simeq \mathbb{F}_q[Y]/(Y^3 - \beta)$. An element $a \in \mathbb{F}_{q^3}$ is of the form $a = a_0 + a_1 Y + a_2 Y^2 \mod Y^3 - \beta$ with $a_i \in \mathbb{F}_{p^2}$. We use this theorem to find a tiny non-residue $\beta \in \mathbb{F}_{p^2}$ in order to build $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}[U]/(U^6 - \beta)$. Finding β is completely straightforward with Magma but we need to be able to find it with the functions available in the LibCryptoLCH.

Theorem 14 ([BS10, Th. 4]). *Let $m > 1, n > 0$ be integers, p an odd prime and $\alpha \in \mathbb{F}_{p^n}^\times$. The binomial $X^m - \alpha$ is irreducible in $\mathbb{F}_{p^n}[X]$ if the following two conditions are satisfied:*

1. *Each prime factor d of m divides $p - 1$ and $\text{Norm}_{\mathbb{F}_{p^n}/\mathbb{F}_p}(\alpha) \in \mathbb{F}_p$ is not a d-th residue in $\mathbb{F}_p$;*
2. *If $m \equiv 0 \bmod 4$ then $p^n \equiv 1 \bmod 4$.*

Thanks to this theorem (with $n = 2, m = 6$), to test if for a given $\beta \in \mathbb{F}_{p^2}$, the polynomial $U^6 - \beta$ is irreducible, we need to

– check that $p \equiv 1 \bmod 6$;
– compute $N_\beta = \mathrm{Norm}_{\mathbb{F}_{p^2}/\mathbb{F}_p}(\beta)$;
– check that $N_\beta^{\frac{p-1}{2}} \neq 1$ and $N_\beta^{\frac{p-1}{3}} \neq 1$.

This is easily achieved with the LibCryptoLCH functions available in the `Modular` and `Quadratic` packages.

We now give our arithmetic for $\mathbb{F}_{q^3} \simeq \mathbb{F}_{p^{2\cdot 3}}$.

$$a = a_0 + a_1 Y + a_2 Y^2,\ b = b_0 + b_1 Y + b_2 Y^2,\ r = r_0 + r_1 Y + r_2 Y^2$$

Multiplication: Karatsuba–3			Squaring: Chung-Hasan–2		
r	$=$	$a \cdot b$	r	$=$	a^2
v_0	$=$	$a_0 b_0$	s_0	$=$	a_0^2
v_1	$=$	$a_1 b_1$	s_1	$=$	$2a_0 a_1$
v_2	$=$	$a_2 b_2$	s_2	$=$	$(a_0 - a_1 + a_2)^2$
r_0	$=$	$v_0 + \beta((a_1 + a_2)(b_1 + b_2) - v_1 - v_2)$	s_3	$=$	$2a_1 a_2$
r_1	$=$	$(a_0 + a_1)(b_0 + b_1) - v_0 - v_1 + \beta v_2$	s_4	$=$	a_2^2
r_2	$=$	$(a_0 + a_2)(b_0 + b_2) - v_0 + v_1 - v_2$	r_0	$=$	$s_0 + \beta s_3$
			r_1	$=$	$s_1 + \beta s_4$
			r_2	$=$	$s_1 + s_2 + s_3 - s_0 - s_4$
$6M_{p^2} + 15Add_{p^2} + 2M_\alpha$			$2M_{p^2} + 3S_{p^2} + 12Add_{p^2} + 2M_\beta$		

(3.7)

The multiplication and squaring in $\mathbb{F}_{q^6}$ are composed with the formulas for quadratic and cubic extensions. The costs of these operations are explained in Tab. 3.1 and Tab. 3.2. Other implementations suggest to use the Toom-Cook-3 method. This method is based on evaluation then interpolation. The drawback of this method is the need of division by small constant numbers such as $2, 3$. We have chosen to use the formulas which do not need divisions by small constants.

$\mathbb{F}_{p^{12}}$	$M_{p^{12}}$			$M_{p^{12}}$	
2 |	Karatsuba–2				
$\mathbb{F}_{p^6}$	$3M_{p^6} + 5A_{p^6} + 1M_Y$	M_{p^6}			M_{p^6}
3 |		Karatsuba–3			
$\mathbb{F}_{p^2}$		$6M_{p^2} + 13A_{p^2} + 2M_\beta$	M_{p^2}		
2 |			Karatsuba–2		
$\mathbb{F}_p$			$3M_p + 5A_p + 1M_\alpha$	$54M_p$	$18M_p$

Table 3.1: Multiplication in $\mathbb{F}_{p^{12}}$ and $\mathbb{F}_{p^6}$

$\mathbb{F}_{p^{12}}$	$S_{p^{12}}$			$S_{p^{12}}$	
2 |	Complex–2				
$\mathbb{F}_{p^6}$	$2M_{p^6} + 4A_{p^6} + 2M_Y$	S_{p^6}			S_{p^6}
3 |		Chung-Hasan–2			
$\mathbb{F}_{p^2}$		$2M_{p^2} + 3S_{p^2} + 10A_{p^2} + 2M_\beta$	S_{p^2}		
2 |			Complex–2		
$\mathbb{F}_p$			$2M_p + 4A_p + 2M_\alpha$	$36M_p$	$12M_p$

Table 3.2: Squaring in $\mathbb{F}_{p^{12}}$ and $\mathbb{F}_{p^6}$

With this efficient arithmetic on extension fields we can now implement a pairing on a Barreto-Naehrig curve. The pairing operations take place in $\mathbb{F}_p$, $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^{12}}$.

3.2 Implementation of ate and optimal ate pairing on a BN curve

In this section we explain step by step the implementation of a state-of-the-art pairing on a Barreto-Naehrig curve. We recall in Sec. 3.2.1 the steps of the advances of the library development. Then we state the general algorithm to compute a pairing. The two main parts of a pairing computation are the Miller loop (Sec. 3.2.2) and the final exponentiation (Sec. 3.2.3). We finish with our timings in Sec. 3.2.4.

3.2.1 Starting point

The work presented in this section started in Master's internship in 2010. A Tate pairing on a supersingular elliptic curve and on a BN curve was implemented in the cryptographic library. The implementation is explained in the introduction (see 1.4). The arithmetic efficiency of the extension field was improved, as well as the Tate pairing computation and the final exponentiation. Finally an ate and an optimal ate pairings were added to the LibCryptoLCH. Now the optimal ate pairing computation is four times faster than the Tate pairing computation of 2010 (both on the same BN curve). The ate and optimal ate pairing implementations are explained in the following.

Algorithm 13: Tate pairing $e_{\text{Tate}}(P,\phi_6(Q))^{\frac{p^{12}-1}{m}}$ on a BN curve

Input: $E(\mathbb{F}_p): y^2 = x^3 + b$, $P(x_P, y_P) \in E(\mathbb{F}_p)[m]$, $Q(x_Q, y_Q) \in E'(\mathbb{F}_{p^2})[m]$, m, t, x
Output: $e_{\text{Tate}}(P,\phi_6(Q)) \in \boldsymbol{\mu}_m \subset \mathbb{F}^*_{p^{12}}$

1 $S(X_S : Y_S : Z_S) \leftarrow (x_P : y_P : 1)$
2 $f \leftarrow 1$
3 **for** $i \leftarrow \lfloor \log_2(m) \rfloor - 1, \ldots, 0$ **do**
4 $\quad (S, \ell) \leftarrow g(S, Q)$ (see (1.33), (3.8)) $\quad 10M_p + 5S_p$
5 $\quad f \leftarrow f^2 \cdot \ell$ (see (3.15) and Alg. 15) $\quad S_{p^{12}} + 10M_{p^2} + 6M_p = 72M_p$
6 $\quad$ **if** $m_i = 1$ **then**
7 $\quad\quad (S, \ell) \leftarrow h(S, P, Q)$ (see (1.34), (3.9)) $\quad 11M_p + 3S_p$
8 $\quad\quad f \leftarrow f \cdot \ell$ (see (3.15) and Alg. 15) $\quad 10M_{p^2} + 6M_p = 36M_p$

Miller loop: $\log_2 m \cdot (82M_p + 5S_p) + \text{HW}(m) \cdot (47M_p + 3S_p)$

9 $f \leftarrow f^{p^6-1}$ $\quad 3M_{p^6} + 2S_{p^6} + 10M_{p^2} + 3S_{p^2} + 2M_p + 2S_p + I_p = 116M_p + 2S_p + I_p$
10 $f \leftarrow f^{p^2+1}$ $\quad 8M_p + M_{p^{12}} = 64M_p$
11 $f \leftarrow f^{\frac{p^4-p^2+1}{m}}$ (see Alg. 16)
$\quad \big(54(\text{HW}(t) + \text{HW}(|6x+5|)) + 18(\log(t) + \log(|6x+5|)) + 763\big)M_p$
12 **return** f

The difference between Tate pairing and ate pairing is firstly the swap of the two inputs points. Instead of computing a Miller function $f_{P,m}(Q)$ over $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)[m]$ evaluated at $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^k})[m]$, an ate pairing consists of a Miller function $f_{Q,t-1}(P)$ over the point Q, evaluated at P. We compute in particular $[t-1]Q$ with Q of coefficients in $\mathbb{F}_{p^2}$, instead of $[m]P$ with coefficients in $\mathbb{F}_p$. This function is nevertheless more efficient because of the reduced length from m to $t-1$. On a BN curve, the trace t has half the size of m. In this section we will explain the line an tangent computations for ate pairing (variant of Alg. 13 line 4 and 4) and the line multiplication $\ell \cdot f$ optimized for sparse elements ℓ of $\mathbb{F}_{p^{12}}$ (variant of Alg. 13 line 5 and 8). The final exponentiation uses exactly the same exponent for Tate, ate and optimal ate pairings. A practical improvement of this exponentiation was proposed in [GS10, DSD07]. We will recall this faster exponentiation and explain our implementation.

3.2.2 Line and Tangent Computation

In this section we explain the computations of lines and tangents. The intermediate functions denoted g and h in Alg. 18 contain the doubling (g) and addition (h) of points and the coefficients of the line through the considered points. We re-use the functions g and h of Tate pairing computation from Alg. 8 and Alg. 9 in Sec. 1.4.4.2. We denote by $\ell_{T,Q}$ the line through $T \in \mathbb{G}_2$ and $Q \in \mathbb{G}_2$ and by $\ell_{T,T}$ the

tangent line at $T \in \mathbb{G}_2$. The line is evaluated at $P \in \mathbb{G}_1$ with coefficients in $\mathbb{F}_p$. We will use the twist map to obtain for ℓ a sparse element in $\mathbb{F}_{p^{12}}$ and to save multiplications in the step following the line computation. We now give the formulas for doubling T, with T in *compressed* form thanks to a degree-6 twist. We recall that two twists are possible for a BN curve $E : y^2 = x^3 + b$. Its degree-6 D-twist over $\mathbb{F}_{p^2}$ is $E' : y^2 = x^3 + b/\beta$ (D stands for *division* by β). Its degree-6 M-twist is $E'' : y^2 = x^3 + b\beta$ (M stands for *multiplication* by β) with β a non-square and non-cube in $\mathbb{F}_{p^2}$. With a D-twist and a twist map denoted ϕ_6, we have $\phi_6(T') = (X'_T U^2, Y'_T U^3, Z'_T) \sim (X'_T, Y'_T, Z'_T/U)$ in Jacobian coordinates.

Doubling on the twist E' with $a' = a/U^4 \neq 0$ and $T(X'_T, Y'_T, Z'_T) \xrightarrow[\phi_6]{\text{twist}} (X'_T, Y'_T, Z'_T/U)$

$$
\begin{array}{rcl}
t'_1 & = & 2Y'^2_T \\
t'_2 & = & 2X'_T t'_1 \\
t'_3 & = & 2t'^2_1 \\
t'_4 & = & Z'^2_T \longrightarrow Z'^2_T/U^2 \\
t'_5 & = & 3X'^2_T + a' t'^2_4 \longrightarrow 3X'^2_T + a/U^4 t'^2_4 U^4 = 3X'^2_T + a t'^2_4 \\
X'_{2T} & = & t'^2_5 - 2t'_2 \\
Y'_{2T} & = & t'_5(t'_2 - X'_{2T}) - t'_3 \\
Z'_{2T} & = & 2Y'_T Z'_T \longrightarrow 2Y'_T Z'_T/U
\end{array}
$$

cost: $4M_{p^2} + 6S_{p^2} + 11Add_{p^2}$

Doubling with $a = a' = 0$ and $T'(X'_T, Y'_T, Z'_T) \xrightarrow[\phi_6]{\text{twist}} (X'_T, Y'_T, Z'_T/U)$

$$
\begin{array}{rcl}
t'_1 & = & 2Y'^2_T \\
t'_2 & = & 2X'_T t'_1 \\
t'_3 & = & 2t'^2_1 \\
t'_4 & = & Z'^2_T \longrightarrow Z'^2_T/U^2 \\
t'_5 & = & 3X'^2_T \\
X'_{2T} & = & t'^2_5 - 2t'_2 \\
Y'_{2T} & = & t'_5(t'_2 - X'_{2T}) - t'_3 \\
Z'_{2T} & = & 2Y'_T Z'_T \longrightarrow 2Y'_T Z'_T/U
\end{array}
$$

cost: $3M_{p^2} + 5S_{p^2} + 10Add_{p^2}$

(3.8)

Addition $T'(X'_T, Y'_T, Z'_T), Q'(x'_Q, y'_Q) \xrightarrow[\phi_6]{\text{twist}} (X'_T, Y'_T, Z'_T/U), Q'(x'_Q U^2, y'_Q U^3)$

$$
\begin{array}{rcl}
t'_1 & = & Z'^2_T \longrightarrow Z'^2_T/U^2 \\
t'_2 & = & Z'_T t'_1 \longrightarrow Z'_T t'_1/U^3 \\
t'_3 & = & x'_Q t'_1 \longrightarrow x'_Q U^2 t'_1/U^2 = x'_Q t'_1 \\
t'_4 & = & y'_Q t'_2 \longrightarrow y'_Q U^3 t'_2/U^3 = y'_Q t'_2 \\
t'_5 & = & t'_3 - X'_T \\
t'_6 & = & t'_4 - Y'_T \\
t'_7 & = & t'^2_5 \\
t'_8 & = & t'_5 t'_7 \\
t'_9 & = & X'_T t'_7 \\
X'_{T+P} & = & t'^2_6 - (t'_8 + 2t'_9) \\
Y'_{T+P} & = & t'_6(t'_9 - X'_{T+P}) - Y'_T t'_8 \\
Z'_{T+P} & = & Z'_T t'_5 \longrightarrow Z'_T t'_5/U
\end{array}
$$

cost: $8M_{p^2} + 3S_{p^2} + 7Add_{p^2}$

(3.9)

We now explain the line and tangent computation. We start from the same doubling and addition formulas ((3.8) and (3.9)) and the line and tangent computations from (1.33) and (1.34). This time, the line is through points in $\mathbb{G}_2$ and evaluated at a point in $\mathbb{G}_1$. In the doubling and addition formulas we represented the coefficients with $U \in \mathbb{F}_{p^{12}}$ but in practice these computations are entirely performed in $\mathbb{F}_{p^2}$. The computations in $\mathbb{F}_{p^{12}}$ are for the line multiplication. The same font and color code is used. Elements in the finite field $\mathbb{F}_q$ are in black, those in $\mathbb{F}_{q^2}$ are in gray and bold font and the elements in $\mathbb{F}_{q^{12}}$ are in dark gray and bold font. We start with

$$\ell_{T',T'}(x, y) = 2Y'_T Z'^3_T y - 2Y'^2_T - (3X'^2_T + a' Z'^4_T)(Z'^2_T x - X'_T) .$$

The twist map is

$$\phi_6(T') = \phi_6(X'_T, Y'_T, Z'_T) = (X'_T U^2, Y'_T U^3, Z'_T) = (X'_T, Y'_T, Z'_T/U) .$$

Moreover $a^{'} = 0$ so we obtain

$$\begin{aligned}\ell_{\phi_6(T^{'}),\phi_6(T^{'})}(x,y) &= 2Y_T^{'}\frac{Z_T^{'3}}{U^3}y - 2Y_T^{'2} + 3X_T^{'3} - 3xX_T^{'2}\frac{Z_T^{'2}}{U^2} \\ &= \frac{1}{U^3}\left[2Y_T^{'}Z_T^{'3}y + (-2Y_T^{'2} + 3X_T^{'3})U^3 - 3xX_T^{'2}Z_T^{'2}U\right] \\ &= \frac{1}{U^3}\left[yt_4^{'}Z_{2T}^{'} + (-t_1^{'} + t_5^{'}X_T^{'})U^3 - t_5^{'}t_4^{'}xU\right] .\end{aligned}$$

And we finally get after the final exponentiation

$$\ell \equiv (t_5^{'}X_T^{'} - t_1^{'})U^3 - t_5^{'}t_4^{'}xU + yt_4^{'}Z_{2T}^{'} . \tag{3.10}$$

Note that if we use the second form of the twist, namely $E^{''}(\mathbb{F}_{q^2}) : y^2 = x^3 + b\beta$ instead of $E^{'}(\mathbb{F}_{q^2}) : y^2 = x^3 + b/\beta$, the computation of ϕ_6 becomes $\phi_6(T^{''}) = (X_T^{''}, Y_T^{''}, Z_T^{''}U)$ (we have $Z_T^{''}U$ instead of $Z_T^{'}/U$) and

$$\ell_{\phi_6(T^{''}),\phi_6(T^{''})}(x,y) = yt_4^{''}Z_{2T}^{''}U^3 - t_5^{''}t_4^{''}xU^2 + (t_5^{''}X_T^{''} - t_1^{''}) . \tag{3.11}$$

The line computation for an ate pairing is the following.

$$\begin{aligned}\ell_{T^{'},Q^{'}}(x,y) &= Z_{T+Q}^{'}/U(y_P - Y_Q^{'}U^3) - (Y_Q^{'}Z_T^{3'} - Y_T^{'})(x_P - X_Q^{'}U^2) \\ \ell_{\phi_6(T^{'}),\phi_6(Q^{'})}(x,y) &= Z_{T+Q}^{'}/U(y - Y_Q^{'}U^3) - t_6^{'}(x - X_Q^{'}U^2) \\ &= t_6^{'}X_Q^{'}U^2 - Z_{T+Q}^{'}Y_Q^{'}U^2 - t_6^{'}x + Z_{T+Q}^{'}/Uy \\ &= \tfrac{1}{U}\left[(t_6^{'}X_Q^{'} - Z_{T+Q}^{'}Y_Q^{'})U^3 - t_6^{'}xU + Z_{T+Q}^{'}y\right]\end{aligned}$$

Then after the final exponentiation we get

$$\ell_{\phi_6(T^{'}),\phi_6(Q^{'})}(x,y) \equiv (t_6^{'}X_Q^{'} - Z_{T+Q}^{'}Y_Q^{'})U^3 - t_6^{'}xU + Z_{T+Q}^{'}y . \tag{3.12}$$

If the second twist is used, we have $\phi_6(T^{''}) = (X_T^{''}, Y_T^{''}, Z_T^{''}U)$ and the line computation is

$$\ell_{\phi_6(T^{'}),\phi_6(Q^{'})}(x,y) = \frac{1}{U^2}\left[(t_6^{'}X_Q^{'} - Z_{T+Q}^{'}Y_Q^{'}) - t_6^{'}xU^2 + Z_{T+Q}^{'}yU^3\right] .$$

Then after the final exponentiation,

$$\ell_{\phi_6(T^{'}),\phi_6(Q^{'})}(x,y) \equiv (t_6^{'}X_Q^{'} - Z_{T+Q}^{'}Y_Q^{'}) - t_6^{'}xU^2 + Z_{T+Q}^{'}yU^3 . \tag{3.13}$$

In both cases (addition and doubling), the line ℓ is a sparse number of $\mathbb{F}_{p^{12}}$ of the same form: $\ell = \ell_0 + \ell_1 U + \ell_3 U^3$ and $\ell_2 = \ell_4 = \ell_5 = 0$ for a D-type twist. We implement a dedicated multiplication in $\mathbb{F}_{p^{12}}$ of a line ℓ of this form and another element $f \in \mathbb{F}_{p^{12}}$ (not sparse). Instead of $18M_{p^2}$ this multiplication costs $13M_{p^2}$. We save $5M_{p^2} = 15M_p$ at each line multiplication. We note that the line for the ate pairing on a BN curve curve with a D-twist (i.e. $E^{'} : y^2 = x^3 + b/\beta$) has the same sparse form $\ell_0 + \ell_1 U + \ell_3 U^3$ as the line for a Tate pairing on a BN curve but with an M-twist, i.e. $E^{''} : y^2 = x^3 + b\beta$. In our implementation we developed two specific line multiplication functions. We give the pseudo-code in Alg. 14 for a D-type twist and Alg. 15 for an M-type twist. The only improvement compared to e.g. [GAL+12, Alg. 5] is the number of additions. In our algorithm for a D-twist we perform 25 Add_{p^2} and 3 multiplications by β followed by and addition, so 28 additions. In the above cited paper their Alg. 5 needs 44 additions in $\mathbb{F}_{p^2}$. Both algorithms need 13 multiplications in $\mathbb{F}_{p^2}$.

The line multiplication for a Tate pairing computation with an M-type twist uses the same algorithm (Alg. 14) but the coefficient ℓ_0 is in $\mathbb{F}_p$ instead of $\mathbb{F}_{p^2}$, we need 2 M_p to multiply ℓ_0 by any f_i (instead of M_{p^2}). The final cost is $10M_{p^2} + 6M_p$ instead of $13M_{p^2}$. We save $3M_p$ assuming that $M_{p^2} \sim 3M_p$. We give in eq. (3.14) the schedule of the function.

$$
\begin{array}{rcl}
\ell & = & \ell_0 + \ell_1 U + \ell_3 U^3 \\
f & = & f_0 + f_1 U + f_2 U^2 + f_3 U^3 + f_4 U^4 + f_5 U^5 \\
h & = & f \cdot \ell = h_0 + h_1 U + h_2 U^2 + h_3 U^3 + h_4 U^4 + h_5 U^5 \\
 & & \ell_1 f_5 \\
 & & \ell_0 f_0 \\
 & & \ell_3 f_3 \\
h_0 & = & \ell_0 f_0 + \beta(\ell_1 f_5 + \ell_3 f_3) \\
 & & \ell_1 f_1 \\
 & & \ell_3 f_4 \\
h_1 & = & (\ell_0 + \ell_1)(f_0 + f_1) - \ell_0 f_0 - \ell_1 f_1 + \beta \ell_3 f_4 \\
 & & \ell_0 f_2 \\
 & & \ell_3 f_5 \\
h_2 & = & \ell_0 f_2 + \ell_1 f_1 + \beta \ell_3 f_5 \\
 & & \ell_1 f_2 \\
h_3 & = & \ell_1 f_2 + (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0 - \ell_3 f_3 \\
 & & \ell_0 f_4 \\
h_4 & = & \ell_0 f_4 + (\ell_1 + \ell_3)(f_1 + f_3) - \ell_1 f_1 - \ell_3 f_3 \\
 & & \ell_0 f_4 + \ell_1 f_2 + \ell_1 f_5 + \ell_3 f_4 + \ell_0 f_2 + \ell_3 f_5 \\
h_5 & = & (\ell_0 + \ell_1 + \ell_3)(f_2 + f_4 + f_5) - (\ell_0 f_4 + \ell_1 f_2 + \ell_1 f_5 + \ell_3 f_4 + \ell_0 f_2 + \ell_3 f_5)
\end{array}
\tag{3.14}
$$

We found another optimized line multiplication for the second form of twist (denoted M-twist), in $13M_{p^2} + 21Add + 4(M_\beta + Add)$. This function can be used to compute a line multiplication for a Tate pairing with a D-twist, in $10M_{p^2} + 6M_p$ instead of $13M_{p^2}$ because in this case, ℓ_0 is in $\mathbb{F}_p$ instead of $\mathbb{F}_{p^2}$.

$$
\begin{array}{rcl}
\ell & = & \ell_0 + \ell_2 U^2 + \ell_3 U^3 \\
f & = & f_0 + f_1 U + f_2 U^2 + f_3 U^3 + f_4 U^4 + f_5 U^5 \\
h & = & f \cdot \ell = h_0 + h_1 U + h_2 U^2 + h_3 U^3 + h_4 U^4 + h_5 U^5 \\
 & & \ell_0 f_0 \\
 & & \ell_2 f_2 \\
 & & \ell_3 f_5 \\
h_2 & = & \beta \ell_3 f_5 + (\ell_0 + \ell_2)(f_0 + f_2) - \ell_0 f_0 - \ell_2 f_2 \\
 & & \ell_2 f_4 \\
 & & \ell_3 f_3 \\
h_0 & = & \ell_0 f_0 + \beta(\ell_2 f_4 + \ell_3 f_3) \\
 & & \ell_2 f_1 \\
h_3 & = & \ell_2 f_1 + (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0 - \ell_3 f_3 \\
 & & \ell_0 f_4 \\
 & & \ell_3 f_1 \\
h_4 & = & \ell_2 f_2 + \ell_0 f_4 + \ell_3 f_1 \\
 & & \ell_0 f_5 \\
h_5 & = & \ell_0 f_5 + (\ell_2 + \ell_3)(f_2 + f_3) - \ell_2 f_2 - \ell_3 f_3 \\
 & & (\ell_0 + \ell_2 + \ell_3)(f_1 + \beta(f_4 + f_5)) \\
h_1 & = & (\ell_0 + \ell_2 + \ell_3)(f_1 + \beta(f_4 + f_5)) - \beta(\ell_0 f_4 + \ell_0 f_5 + \ell_2 f_4 + \ell_3 f_5) - \ell_2 f_1 - \ell_3 f_1
\end{array}
\tag{3.15}
$$

3.2.3 Final Exponentiation

We present the final exponentiation in Alg. 17. A well-known trick is to decompose the exponentiation into

$$\frac{p^{12}-1}{m} = (p^6-1)\frac{p^6+1}{\Phi_{12}(p)}\frac{\Phi_{12}(p)}{m} = (p^6-1)(p^2+1)\frac{p^4-p^2+1}{m}$$

with $\phi_{12}(p) = p^4 - p^2 + 1$ the 12-th cyclotomic polynomial. The computation of $f^{(p^6-1)(p^2+1)}$ is decomposed with the Frobenius map. The computation of $f^{\frac{p^4-p^2+1}{m}}$ is the most expensive part. Firstly we can use the optimized squaring formulas of Granger and Scott [GS10] after performing $f \leftarrow f^{(p^6-1)(p^2+1)}$.

Algorithm 14: Line multiplication in ate pairing for a D-type twist $E' : y^2 = x^3 + b/\beta$

Input: line $\ell = \ell_0 + \ell_1 U + \ell_3 U^3 \in \mathbb{F}_{p^{12}}$, with $\ell_i \in \mathbb{F}_{p^2}$, element
$f = f_0 + f_1 U + f_2 U^2 + f_3 U^3 + f_4 U^4 + f_5 U^5 \in \mathbb{F}_{p^{12}}$ with $f_i \in \mathbb{F}_{p^2}$.
Output: $h = \ell \cdot f \in \mathbb{F}_{p^{12}}$.

$s_0 \leftarrow \ell_0 \cdot f_0$	
$s_1 \leftarrow \ell_1 \cdot f_1$	
$s_3 \leftarrow \ell_3 \cdot f_3$	
$u_0 \leftarrow \ell_1 \cdot f_5$	
$u_1 \leftarrow u_0 + s_3$	$u_1 = \ell_1 f_5 + \ell_3 f_3$
$h_0 \leftarrow s_0 + \beta u_1$	$h_0 = \ell_0 f_0 + \beta(\ell_1 f_5 + \ell_3 f_3)$
$u_1 \leftarrow \ell_0 + \ell_1$	
$u_2 \leftarrow f_0 + f_1$	
$u_3 \leftarrow u_1 \cdot u_2$	$u_3 = (\ell_0 + \ell_1)(f_0 + f_1)$
$u_1 \leftarrow u_3 - s_0$	$u_1 = (\ell_0 + \ell_1)(f_0 + f_1) - \ell_0 f_0$
$u_2 \leftarrow u_1 - s_1$	$u_2 = (\ell_0 + \ell_1)(f_0 + f_1) - \ell_0 f_0 - \ell_1 f_1 = \ell_0 f_1 + \ell_1 f_0$
$u_1 \leftarrow \ell_3 \cdot f_4$	
$h_1 \leftarrow u_2 + \beta u_1$	$h_1 = \ell_0 f_1 + \ell_1 f_0 + \beta \ell_3 f_4$
$u_2 \leftarrow u_0 + u_1$	$u_2 = \ell_1 f_5 + \ell_3 f_4$
$u_0 \leftarrow \ell_0 \cdot f_2$	
$u_1 \leftarrow u_0 + s_1$	$u_1 = \ell_0 f_2 + \ell_1 f_1$
$u_3 \leftarrow u_0 + u_2$	$u_3 = \ell_0 f_2 + \ell_1 f_5 + \ell_3 f_4$
$u_0 \leftarrow \ell_3 \cdot f_5$	
$h_2 \leftarrow u_1 + \beta u_0$	$h_2 = \ell_0 f_2 + \ell_1 f_1 + \beta \ell_3 f_5$
$u_1 \leftarrow u_0 + u_3$	$u_1 = \ell_3 f_5 + \ell_0 f_2 + \ell_1 f_5 + \ell_3 f_4$
$u_0 \leftarrow \ell_0 + \ell_3$	
$u_2 \leftarrow f_0 + f_3$	
$u_3 \leftarrow u_0 \cdot u_2$	$u_3 = (\ell_0 + \ell_3)(f_0 + f_3)$
$u_0 \leftarrow u_3 - s_0$	$u_0 = (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0$
$u_2 \leftarrow u_0 - s_3$	$u_2 = (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0 - \ell_3 f_3 = \ell_0 f_3 + \ell_3 f_0$
$u_3 \leftarrow \ell_1 \cdot f_2$	
$h_3 \leftarrow u_2 + u_3$	$h_3 = \ell_0 f_3 + \ell_3 f_0 + \ell_1 f_2$
$u_0 \leftarrow u_3 + u_1$	$u_0 = \ell_1 f_2 + \ell_3 f_5 + \ell_0 f_2 + \ell_1 f_5 + \ell_3 f_4$
$u_1 \leftarrow \ell_1 + \ell_3$	
$u_2 \leftarrow f_1 + f_3$	
$u_3 \leftarrow u_1 \cdot u_2$	$u_3 = (\ell_1 + \ell_3)(f_1 + f_3)$
$u_2 \leftarrow u_3 - s_1$	$u_2 = (\ell_1 + \ell_3)(f_1 + f_3) - \ell_1 f_1$
$u_3 \leftarrow u_2 - s_3$	$u_3 = (\ell_1 + \ell_3)(f_1 + f_3) - \ell_1 f_1 - \ell_3 f_3 = \ell_1 f_3 + \ell_3 f_1$
$u_2 \leftarrow \ell_0 \cdot f_4$	
$h_4 \leftarrow u_3 + u_2$	$h_4 = \ell_0 f_4 + \ell_1 f_3 + \ell_3 f_1$
$u_3 \leftarrow u_2 + u_0$	$u_3 = \ell_0 f_4 + \ell_1 f_2 + \ell_3 f_5 + \ell_0 f_2 + \ell_1 f_5 + \ell_3 f_4$
$u_0 \leftarrow u_1 + \ell_0$	$u_0 = \ell_0 + \ell_1 + \ell_3$
$u_2 \leftarrow f_2 + f_4$	
$u_1 \leftarrow u_2 + f_5$	$u_1 = f_2 + f_4 + f_5$
$u_2 \leftarrow u_0 \cdot u_1$	$u_2 = (\ell_0 + \ell_1 + \ell_3)(f_2 + f_4 + f_5)$
$h_5 \leftarrow u_2 - u_3$	
return h	$13M_{p^2} + 3M_\beta + 18Add_{p^2} + 7Sub_{p^2}$

Algorithm 15: Line multiplication for an M-type twist $E'' : y^2 = x^3 + b \cdot \beta$

Input: line $\ell = \ell_0 + \ell_2 U^2 + \ell_3 U^3 \in \mathbb{F}_{p^{12}}$, with $\ell_i \in \mathbb{F}_{p^2}$, element
$f = f_0 + f_1 U + f_2 U^2 + f_3 U^3 + f_4 U^4 + f_5 U^5 \in \mathbb{F}_{p^{12}}$ with $f_i \in \mathbb{F}_{p^2}$.
Output: $h = \ell \cdot f \in \mathbb{F}_{p^{12}}$.

1	$u_0 \leftarrow \ell_0 + \ell_2$	
2	$u_2 \leftarrow f_0 + f_2$	
3	$u_1 \leftarrow u_0 \cdot u_2$	$u_1 = (\ell_0 + \ell_2)(f_0 + f_2)$
4	$u_0 \leftarrow \ell_0 \cdot f_0$	
5	$u_2 \leftarrow \ell_2 \cdot f_2$	
6	$u_4 \leftarrow u_1 - u_0$	$u_4 = (\ell_0 + \ell_2)(f_0 + f_2) - \ell_0 f_0$
7	$u_1 \leftarrow u_4 - u_2$	$u_1 = (\ell_0 + \ell_2)(f_0 + f_2) - \ell_0 f_0 - \ell_2 f_2$
8	$u_4 \leftarrow \ell_3 \cdot f_5$	
9	$h_2 \leftarrow u_1 + \beta u_4$	$h_2 = (\ell_0 + \ell_2)(f_0 + f_2) - \ell_0 f_0 - \ell_2 f_2 + \beta \ell_3 f_5$
10	$u_1 \leftarrow \ell_2 \cdot f_4$	
11	$u_3 \leftarrow \ell_3 \cdot f_3$	
12	$u_5 \leftarrow u_1 + u_3$	$u_5 = \ell_2 f_4 + \ell_3 f_3$
13	$h_0 \leftarrow u_0 + \beta u_5$	$h_0 = \ell_0 f_0 + \beta(\ell_2 f_4 + \ell_3 f_3)$
14	$u_5 \leftarrow u_1 + u_4$	$= \ell_2 f_4 + \ell_3 f_5$
15	$u_1 \leftarrow \ell_0 + \ell_3$	
16	$u_4 \leftarrow f_0 + f_3$	
17	$u_6 \leftarrow u_1 \cdot u_4$	$u_6 = (\ell_0 + \ell_3) \cdot (f_0 + f_3)$
18	$u_1 \leftarrow u_6 - u_0$	$u_1 = (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0$
19	$u_6 \leftarrow u_1 - u_3$	$u_6 = (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0 - \ell_3 f_3$
20	$u_1 \leftarrow \ell_2 \cdot f_1$	
21	$h_3 \leftarrow u_1 + u_6$	$h_3 = \ell_2 f_1 + (\ell_0 + \ell_3)(f_0 + f_3) - \ell_0 f_0 - \ell_3 f_3$
22	$u_4 \leftarrow \ell_0 \cdot f_4$	
23	$u_6 \leftarrow u_5 + u_4$	$u_6 = \ell_0 f_4 + \ell_2 f_4 + \ell_3 f_5$
24	$u_5 \leftarrow u_2 + u_4$	$u_5 = \ell_2 f_2 + \ell_0 f_4$
25	$u_4 \leftarrow \ell_3 \cdot f_1$	
26	$h_4 \leftarrow u_4 + u_5$	$h_4 = \ell_3 f_1 + \ell_2 f_2 + \ell_0 f_4$
27	$u_5 \leftarrow u_4 + u_1$	$u_5 = \ell_3 f_1 + \ell_2 f_1$
28	$u_1 \leftarrow \ell_0 \cdot f_5$	
29	$u_4 \leftarrow u_1 - u_2$	$u_4 = \ell_0 f_5 - \ell_2 f_2$
30	$u_2 \leftarrow u_4 - u_3$	$u_2 = \ell_0 f_5 - \ell_2 f_2 - \ell_3 f_3$
31	$u_3 \leftarrow f_2 + f_3$	
32	$u_4 \leftarrow \ell_2 + \ell_3$	
33	$u_0 \leftarrow u_3 \cdot u_4$	$u_0 = (f_2 + f_3)(\ell_2 + \ell_3)$
34	$h_5 \leftarrow u_0 + u_2$	$h_5 = \ell_0 f_5 - \ell_2 f_2 - \ell_3 f_3 + (f_2 + f_3)(\ell_2 + \ell_3)$
35	$u_2 \leftarrow u_1 + u_6$	$u_2 = \ell_0 f_5 + \ell_0 f_4 + \ell_2 f_4 + \ell_3 f_5$
36	$u_6 \leftarrow \ell_0 + u_4$	$u_6 = (\ell_0 + \ell_2 + \ell_3)$
37	$u_4 \leftarrow f_4 + f_5$	
38	$u_0 \leftarrow f_1 + \beta u_4$	$u_0 = f_1 + \beta(f_4 + f_5)$
39	$u_4 \leftarrow u_6 \cdot u_0$	$u_4 = (\ell_0 + \ell_2 + \ell_3)(f_1 + \beta(f_4 + f_5))$
40	$u_6 \leftarrow u_5 + \beta u_2$	$u_6 = (\ell_2 f_1 + \ell_3 f_1) + \beta(\ell_0 f_5 + \ell_0 f_4 + \ell_2 f_4 + \ell_3 f_5)$
41	$h_1 \leftarrow u_4 - u_6$	$h_1 = (\ell_0 + \ell_2 + \ell_3)(f_1 + \beta(f_4 + f_5)) - \beta(\ell_0 f_5 + \ell_0 f_4 + \ell_2 f_4 + \ell_3 f_5) - (\ell_2 f_1 + \ell_3 f_1)$
42	**return** h	

Secondly we use the decomposition of the exponent $(p^4 - p^2 + 1)/m$ in terms of $p(x), x$ developed in [DSD07]. We recall that for a BN curve, the parameters have the form

$$\begin{aligned} m(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1 \\ p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \end{aligned}$$

with x taking positive or negative values. Then

$$\begin{aligned} (p^4 - p^2 + 1)/m &= p^3 + (6x^2+1)p^2 + (-36x^3 - 18x^2 - 12x + 1)p - 36x^3 - 30x^2 - 18x - 2 \\ f^{(p^4-p^2+1)/m} &= (f^{p^3}) \cdot (f^{p^2})^{6x^2+1} \cdot (f^p)^{-36x^3-18x^2-12x+1} \cdot f^{-36x^3-30x^2-18x-2} \end{aligned} \tag{3.16}$$

and finally the two exponentiations with large exponent in the right-hand side are optimized as shown in Alg. 16. A step-by-step description is given in Alg. 17 with the cost of each operation. The cost of this final exponentiation is $I_p + 2S_p + \big(872 + 54(\mathrm{HW}(s) + \mathrm{HW}(t)) + 18(\log(s) + \log(t))\big)M_p$, with I_p an inversion in $\mathbb{F}_p$ and $s = |6x + 5|$. For example at a 128-bit security level, the parameter x is 63-bit long. We can approximate $\log(s) = 66$ bits and $\log(t) = \log(6x^2 + 1) = 128$ bits. We can assume that the Hamming weight is approximately half the size of the numbers s and t. Then the cost of the final exponentiation is in average $I_p + 2S_p + 9602M_p$.

Algorithm 16: Final Exponentiation on a BN curve, last part, [DSD07]

Input: $f \in \mathbb{F}_{p^{12}}$, x and p

Output: $f^{\frac{p^4-p^2+1}{m}} \in \mathbb{F}_{p^{12}}$

1 **if** $x < 0$ **then**
2 $\quad a \leftarrow f^{6|x|-5}$
3 **else**
4 $\quad a \leftarrow f^{6x+5}$
5 $\quad a \leftarrow a^{p^6}$ (Frobenius, free)

$\log(|6x+5|)M_{p^{12}} + \mathrm{HW}(|6x+5|)S_{p^{12}}$

6 $b \leftarrow a^p$ (Frobenius) — $5M_{p^2}$
7 $b \leftarrow ab$ — $M_{p^{12}}$
8 Compute f^p, f^{p^2} and f^{p^3} (Frobenius) — $5M_{p^2} + 8M_p + 8M_p$
9 $f \leftarrow f^{p^3} \cdot \left[b \cdot (f^p)^2 \cdot f^{p^2}\right]^{6x^2+1} \cdot b \cdot (f^p \cdot f)^9 \cdot a \cdot f^4$ — $\big(54\mathrm{HW}(t) + 18\log(t) + 663\big)M_p$
10 **return** f

$\Big(54\big(\mathrm{HW}(t) + \mathrm{HW}(|6x+5|)\big) + 18\big(\log(t) + \log(|6x+5|)\big) + 763\Big)M_p$

3.2.4 Performances for Tate, ate and optimal ate pairings on BN curves

We can now present the complete optimal ate pairing algorithm in Alg. 18. The Miller loop needs the functions f and g of line an tangent computation. The accumulation of lines is described step by step in Alg. 14 and Alg. 15. The first algorithm is an optimization from $54M_p$ (generic multiplication in $\mathbb{F}_{p^{12}}$) to $39M_p$. It is valid for a pairing with a compression of the second input point with a degree 6 twisted curve of the form $E' : y^2 = x^3 + b/\beta$ with β a non-square and non-cube in $\mathbb{F}_{p^2}$. This twist is named D-twist (for `division` by β). The second algorithm is designed for a twist of the other type, i.e. $E'' : y^2 = x^3 + b\beta$. This twist is named M-twist, for `multiplication` by β.

We present in Tab. 3.3 our running times for a Tate, an ate and an optimal ate pairing on the same BN curve. The code was run on a Xeon E5530 PC with x86-64 Intel processor.

Algorithm 17: Final exponentiation on a BN curve

Input: x defining the curve parameters, sign(x), t trace of the curve, $f \in \mathbb{F}_{p^{12}}$

Output: $h = f^{\frac{p^{12}-1}{m}} \in \mathbb{G}_T$

1 $f_2 \leftarrow f^{p^6-1}$ — $I_p + 116M_p + 2S_p$

2 $f \leftarrow f_2^{p^2+1}$ — $f = f^{(p^6-1)(p^2+1)}$: $8M_p + M_{p^{12}}$

Now we can use the optimized formula $S_{\Phi_6(p^2)} \simeq 18M_p$ instead of $S_{p^{12}} \simeq 36M_p$

3 **if** $x > 0$ **then** Compute $(f^{6x+5})^{-1}$

4 $\quad s \leftarrow 6x + 5$

5 $\quad f_3 \leftarrow f^s$ — $f_3 = f^{6x+5}$: $\log_2(6x+5)S_{\Phi_6(p^2)} + \mathrm{HW}(6x+5)M_{p^{12}}$

6 $\quad a \leftarrow f_3^{p^6}$ — $\mathrm{Norm}(f_3) = 1$ then $f_3^{-1} = f_3^{p^6}$

7 **else** (i.e. $x < 0$) Compute $f^{6|x|-5}$

8 $\quad s \leftarrow 6|x| - 5$

9 $\quad a \leftarrow f^s$ — $a = f^{-6x-5}$: $\log_2(6|x|-5)S_{\Phi_6(p^2)} + \mathrm{HW}(6|x|-5)M_{p^{12}}$

10 $f_3 \leftarrow a^p$ — $f_3 = f^{(6x-5)p}$: $5M_{p^2}$

11 $b \leftarrow a \cdot f_3$ — $b = f^{6x-5} \cdot f^{(6x-5)p} = f^{(6x-5)(p+1)}$: $M_{p^{12}}$

12 $f_1 \leftarrow f^p$ — $5M_{p^2}$

13 $f_2 \leftarrow f^{p^2}$ — $8M_p$

14 $f_3 \leftarrow f_1^{p^2}$ — $f_3 = f^{p^3}$: $8M_p$

15 $f_4 \leftarrow a \cdot b$ — $M_{p^{12}}$

16 $a \leftarrow b \cdot f_2$ — $a = b \cdot f^{p^2}$: $M_{p^{12}}$

17 $f_2 \leftarrow f_1^2$ — $f_2 = (f^p)^2$: $S_{\Phi_6(p^2)}$

18 $b \leftarrow f_1 \cdot f$ — $b = f^p \cdot f$: $M_{p^{12}}$

19 $f_1 \leftarrow f_2 \cdot a$ — $f_1 = (f^p)^2 \cdot (b \cdot f^{p^2})$: $M_{p^{12}}$

20 $a \leftarrow f_1^t$ — $a = [(f^p)^2 \cdot (b \cdot f^{p^2})]^{6x^2+1}$: $\log_2(t)S_{\Phi_6(p^2)} + \mathrm{HW}(t)M_{p^{12}}$

21 $f_2 \leftarrow f_3 \cdot a$ — $f_2 = f^{p^3} \cdot [(f^p)^2 \cdot (b \cdot f^{p^2})]^{6x^2+1}$: $M_{p^{12}}$

22 $f_1 \leftarrow f_2 \cdot f_4$ — $f_1 = f^{p^3} \cdot [(f^p)^2 \cdot (b \cdot f^{p^2})]^{6x^2+1} \cdot a \cdot b$: $M_{p^{12}}$

23 $f_2 \leftarrow b^2$ — $f_2 = (f^p \cdot f)^2$: $S_{\Phi_6(p^2)}$

24 $f_4 \leftarrow f^2$ — $S_{\Phi_6(p^2)}$

25 $f_3 \leftarrow f_2^2$ — $f_3 = (f^p \cdot f)^4$: $S_{\Phi_6(p^2)}$

26 $a \leftarrow f_4^2$ — $a = f^4$: $S_{\Phi_6(p^2)}$

27 $f_2 \leftarrow f_3^2$ — $f_2 = (f^p \cdot f)^8$: $S_{\Phi_6(p^2)}$

28 $f_4 \leftarrow f_1 \cdot a$ — $f_4 = f^4 \cdot f^{p^3} \cdot [(f^p)^2 \cdot (b \cdot f^{p^2})]^{6x^2+1} \cdot a \cdot b$: $M_{p^{12}}$

29 $f_3 \leftarrow b \cdot f_2$ — $f_3 = (f^p \cdot f) \cdot (f^p \cdot f)^8 = (f^p \cdot f)^9$: $M_{p^{12}}$

30 $h \leftarrow f_3 \cdot f_4$ — $h = (f^p \cdot f)^9 \cdot f^4 \cdot f^{p^3} \cdot [(f^p)^2 \cdot (b \cdot f^{p^2})]^{6x^2+1} \cdot a \cdot b$: $M_{p^{12}}$

$(\mathrm{HW}(t)+10)M_{p^{12}} + (\log t + 6)S_{\Phi_6(p^2)} + 10M_{p^2} + 16M_p = \big(54\mathrm{HW}(t) + 18\log(t) + 694\big)M_p$ **return** h

with $s = |6x+5|$, $I_p + 2S_p + \big(872 + 54(\mathrm{HW}(s) + \mathrm{HW}(t)) + 18(\log(s) + \log(t))\big)M_p$

Algorithm 18: Optimal ate pairing $e_{\text{opt ate}}(P, \phi_6(Q))^{\frac{p^{12}-1}{n}}$ on a BN curve

Input: $E(\mathbb{F}_p)$, $P(x_P, y_P) \in E(\mathbb{F}_p)[n]$, $Q(x_Q, y_Q) \in E'(\mathbb{F}_{p^2})[n]$, t, x
Output: $e_{\text{opt ate}}(P, \phi_6(Q)) \in \mu_n \subset \mathbb{F}^*_{p^{12}}$

1 $R(X_R : Y_R : Z_R) \leftarrow (x_Q : y_Q : 1)$
2 $f \leftarrow 1$
3 $s \leftarrow 6x + 2$
4 **for** $m \leftarrow \lfloor \log_2(s) \rfloor - 1, \ldots, 0$ **do**
5 $\quad (R, \ell) \leftarrow g(R, P)$ — $6M_{p^2} + 5S_{p^2} + 4M_p = 32M_p$
6 $\quad f \leftarrow f^2 \cdot \ell$ — $S_{p^{12}} + 13M_{p^2} = 36 + 39 = 75M_p$
7 $\quad$ **if** $s_m = 1$ **then**
8 $\quad\quad (R, \ell) \leftarrow h(R, Q, P)$ — $10M_{p^2} + 3S_{p^2} + 4M_p = 40M_p$
9 $\quad\quad f \leftarrow f \cdot \ell$ — $13M_{p^2} = 39M_p$
10 $Q_1 \leftarrow \pi_p(Q)$ — $M_{p^2} = 3M_p$
11 $Q_2 \leftarrow \pi_{p^2}(Q)$ — $2M_p$
12 $(R, \ell) \leftarrow h(R, Q_1, P)$ — $6M_{p^2} + 5S_{p^2} + 4M_p = 32M_p$
13 $f \leftarrow f \cdot \ell$ — $13M_{p^2} = 39M_p$
14 $(R, \ell) \leftarrow h(R, Q_2, P)$ — $6M_{p^2} + 5S_{p^2} + 4M_p = 32M_p$
15 $f \leftarrow f \cdot \ell$ — $13M_{p^2} = 39M_p$

line 10 to line 15: $147M_p$

Miller Loop: $147M_p + \log_2(6x+2) \cdot 107M_p + \text{HW}(6x+2) \cdot 79M_p$

16 $f \leftarrow f^{p^6-1}$ — $3M_{p^6} + 2S_{p^6} + 10M_{p^2} + 3S_{p^2} + 2M_p + 2S_p + I_p = 116M_p + 2S_p + I_p$
17 $f \leftarrow f^{p^2+1}$ — $8M_p + M_{p^{12}} = 62M_p$
18 **if** $x < 0$ **then**
19 $\quad a \leftarrow f^{6|x|-5}$ — $\log_2(6x+5)S_{\Phi_6(p^2)} + \text{HW}(6x+5)M_{p^{12}}$
20 **else** $(f^{p^6} = f^{-1})$
21 $\quad a \leftarrow (f^{p^6})^{6x+5}$
22 $b \leftarrow a^p$ — $5M_{p^2} = 15M_p$
23 $b \leftarrow ab$ — $M_{p^{12}} = 54M_p$
24 Compute f^p, f^{p^2} and f^{p^3} — $5M_{p^2} + 8M_p + 8M_p = 31M_p$
25 $c \leftarrow b \cdot (f^p)^2 \cdot f^{p^2}$ — $S_{\Phi_6(p^2)} + 2M_{p^{12}} = 126M_p$
26 $c \leftarrow c^{6x^2+1}$ — $\log_2(6x^2+1)S_{\Phi_6(p^2)} + \text{HW}(6x^2+1)M_{p^{12}}$
27 $f \leftarrow f^{p^3} \cdot c \cdot b \cdot (f^p \cdot f)^9 \cdot a \cdot f^4$ — $7M_{p^{12}} + 5S_{\Phi_6(p^2)} = 468M_p$

Exponentiation $f \leftarrow f^{(p^6-1)(p^2+1)(p^4-p^2+1)/n}$:
$(872 + 18\log_2(6x+5) + 54\text{HW}(6x+5) + 18\log_2(6x^2+1) + 54\text{HW}(6x^2+1))M_p + 2S_p + I_p$

28 **return** f

Table 3.3: Benchmarks for Tate, ate and optimal ate pairing on a BN curve, with $\mathbb{F}_{p^2} \simeq \mathbb{F}_p[X]/(X^2+1)$, $\mathbb{F}_{p^{12}} \simeq \mathbb{F}_{p^2}[U]/(U^6 - (X+2))$.

$\log p$, $k \log p$, equiv. AES	256, 3072, AES-128	640, 7680, AES-192	1280, 15360, AES-256
Miller Loop	2.35 ms	18.4 ms	109.2 ms
Final Exp.	2.70 ms	15.8 ms	75.5 ms
Optimal ate pairing	5.05 ms	34.2 ms	184.7 ms

3.3 Pairings on Composite-order Elliptic Curves

We presented our efficient implementation of pairings in Sec. 3.1. In this section we will study and implement (based on the work of the preceding section) a new tool on pairing-friendly groups. This tool uses *composite-order pairing-friendly groups*. We will outline the key ingredients of this tool. We then briefly introduce three major protocols based on this tool we will more deeply study in this section. Finally we will discuss about the parameter size issues in this setting.

We start by an analogy with Joux's key agreement from Diffie-Hellman key exchange. These two key exchanges are presented in the introduction in Sec. 1.1. The principle in Joux's key agreement is to send over an insecure channel only partial pieces of information, namely the g_a, g_b, g_c and compose the secret $e(g,g)^{abc}$ thanks to the bilinear map. We denote the bilinear map by $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ (as previously). The new idea introduced in composite-order bilinear groups is that the three bilinear groups $\mathbb{G}_i$ have a composite-order N, however the factorization of N into $p_1 \cdot p_2$ is a secret information. This permits to hide an information into a prime-order subgroup $\mathbb{G}_{(p_1)}$ of $\mathbb{G}_i$. Because the factorization of N is not publicly available, the global information in $\mathbb{G}_i$ cannot be decomposed into the private information in $\mathbb{G}_{(p_1)}$ and the hiding term in $\mathbb{G}_{(p_2)}$.

We now present the three papers we will study and implement in the remainder of this section. In 2005, Boneh, Goh and Nissim [BGN05] introduced the first public-key homomorphic encryption scheme using composite-order groups equipped with a pairing. The scheme enables several homomorphic additions and one multiplication on few bits. The security relies on the subgroup decision assumption. Decryption time grows exponentially with respect to the input size so this approach for homomorphic encryption is not yet very practical for large data. However the idea was developed for other interests. We refer to Sec. 3.3.4.1 for more informations on BGN. In 2005, a Hierarchical Identity Based Encryption (HIBE) was proposed by Boneh, Boyen and Goh [BBG05]. It relies on the ℓ-bilinear Diffie-Hellman exponent assumption. In 2009, Waters introduced the Dual System Encryption method [Wat09], resulting in very interesting properties for security proofs. In 2011, Lewko and Waters published [LW11] a HIBE relying on the subgroup decision assumption. HIBE has become very practical in the sense that the maximal hierarchy depth is not static i.e. can be augmented without resetting all the system parameters. We refer to Sec. 3.3.4.2 for more details.

The subgroup decision assumption is that given a group $\mathbb{G}$ of composite order $p_1 p_2 = N$ (e.g. an RSA modulus), it is hard do decide whether a given element $g \in \mathbb{G}$ is in the subgroup of order p_1 without knowing p_1 and p_2. N must be infeasible to factor to achieve this hardness. This results in very large parameter sizes, e.g. $\log_2 N = 3072$ or 3248 for a 128-bit security level, according to NIST or ECRYPT II recommendations. Moreover, the pairing computation is much slower in this setting but exact performances were not given yet. To reduce the parameter sizes, Freeman [Fre10] proposed to use a copy of the (e.g. 256-bit) same prime-order group instead of a group whose order (of e.g. 3072 bits) has two or more distinct primes. His paper provides conversions of protocols and in particular of the BGN scheme, from the composite-order to the prime-order setting. Then Lewko at *EUROCRYPT'2012* [Lew12] provided a generic conversion. These conversions achieve much smaller parameter sizes but have a drawback: they no longer require only one but several pairings. More precisely, Lewko's conversion for the HIBE scheme needs at least $2n$ pairings over a prime order group (of e.g. 256-bit) instead of one pairing over a n-prime composite order group (of e.g. 3072-bit).

The translated protocols remain interesting because it is commonly assumed that a pairing is much slower over a composite-order than over a prime-order elliptic curve. An overhead factor around 50 (at an estimate attributed to Scott) was given in [Fre10, §1] for a 80-bit security level. A detailed and precise comparison would be interesting and useful to protocol designers and application developers.

Composite-order pairing-friendly groups require larger parameter sizes because they rely on the difficulty of the factorization problem and there are specific methods to attack it. The Number Field Sieve (NFS) algorithm is the fastest method to factor a two-prime modulus. Lenstra studied carefully its complexity and made recommendations. Lenstra stated that at a 128-bit security level, an RSA modulus can have no more than 3 prime factors of the same size, 4 factors at a 192-bit level and 5 at a 256-bit level [Len01, §4]. We complete his work to obtain the modulus sizes with more than two prime factors, at these three security levels. We then find supersingular elliptic curves of such orders and benchmark a Tate pairing over these curves. We also implemented an optimal ate pairing over a prime-order Barreto-Naehrig

curve, considered as the fastest pairing (at least in software). With these timings, we are able to estimate the total cost of the protocols in composite-order and prime-order settings. We then compare the BGN protocol [BGN05] in the two settings and do the same for the unbounded HIBE protocol of Lewko and Waters [LW11] and its translation [Lew12, §B].

Sec. 3.3.1 presents our results on the modulus sizes with more than two prime factors, at the 128, 192 and 256-bit security level. In Sec. 3.3.2, we present the possibilities to construct pairing-friendly elliptic curves of composite order and our choice for the implementation. We develop a theoretical estimation of each pairing in Sec. 3.3.3. Our implementation results are presented in Sec. 3.3.4. This work was presented at the *ACNS'2013* conference [Gui13]. Updated key size and benchmarks are reported here.

3.3.1 Parameter sizes

In this section, we extend Lenstra's estimates [Len01] to RSA modulus sizes with up to nine prime factors. We present in Tab. 3.4 the usual key length recommendations from `http://www.keylength.com`. The NIST recommendations are the less conservative ones. A modulus of length 3072 is recommended to achieve a security level equivalent to a 128 bit symmetric key. The ECRYPT II recommendations are slightly larger: 3248 bit modulus are suggested.

Table 3.4: Cryptographic key length recommendations, January 2013. All key sizes are provided in bits. These are the minimal sizes for security.

Method	Date	Sym-metric	Asymmetric	Discrete Log		Elliptic curve	Hash function
				Key	Group		
Lenstra / Verheul [LV01]	2076	129	6790–5888	230	6790	245	257
Lenstra Updated [Len04]	2090	128	4440–6974	256	4440	256	256
ECRYPT II (EU) [oEiCI11]	2031–2040	128	3248	256	3248	256	256
NIST (US) [NIS11]	> 2030	128	3072	256	3072	256	256
FNISA (France) [FNI10]	> 2020	128	4096	200	4096	256	256
NSA (US) [NSA10]	–	128	–	–	–	256	256
RFC3766 [OH04]	–	128	3253	256	3253	242	–

We explain here where these key sizes come from. The running-time complexity of the most efficient attacks on discrete logarithm computation and factorization are considered and balanced to fit the last records. We consider the Number Field Sieve attack (NFS, see e.g. [LL93] for an overview) whose complexity is given by the L-function [Len01, §3.1]:

$$L[\alpha = \tfrac{1}{3},\ c = (\tfrac{64}{9})^{1/3}](N) = \exp\left(\Big((64/9)^{1/3} + o(1)\Big)(\log N)^{1/3}(\log\log N)^{2/3}\right) \text{ (NFS)} \tag{3.17}$$

and we consider its logarithm in base 2:

$$\log_2 L[\alpha, c](n) = \big(c + o(1)\big)\ n^{\alpha} \log_2^{1-\alpha}(n \ln 2) \tag{3.18}$$

with $n = \log_2 N$. We also consider the Elliptic Curve Method (ECM) that depends on the modulus size and on the size of the smallest prime p_i in the modulus. This attack is less efficient for a modulus of only two prime factors but become competitive for more prime factors. We consider that all the prime factors p_i have the same size. The ECM complexity is [Len01, §4]

$$E[\alpha = \tfrac{1}{2}, c = \sqrt{2}](N, p_i) = (\log_2 N)^2 \exp\left(\Big(\sqrt{2} + o(1)\Big)(\log p_i)^{1/2}(\log\log p_i)^{1/2}\right) \text{ (ECM)}. \tag{3.19}$$

We have also

$$\log_2 E[\alpha, c](n, \ell) = 2\log_2 n + (c + o(1))\ \ell^{\alpha} \log_2^{1-\alpha}(\ell \ln 2) \tag{3.20}$$

with $n = \log_2 N$ and $\ell = \log_2 p_i$. To estimate the required modulus size, we compute the logarithm in base 2 of the L-function (3.18) and translate it such that $\log_2 L[c, \alpha](512) = 56$ (estimations in [Len01, §3]) or $\log_2 L[c, \alpha](512) = 50$ (Ecrypt recommendations [oEiCI12, §6.2.1]). We obtain $\delta = -14$ for the first and $\delta = -8$ for the second. Fig. 3.3.

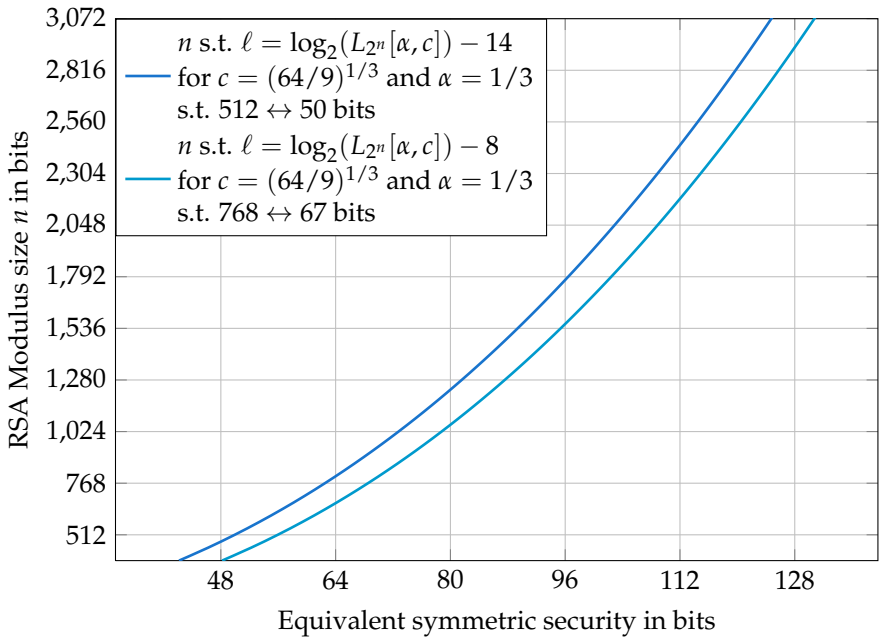


Figure 3.3: Estimated complexity of RSA modulus factorization with NFS method

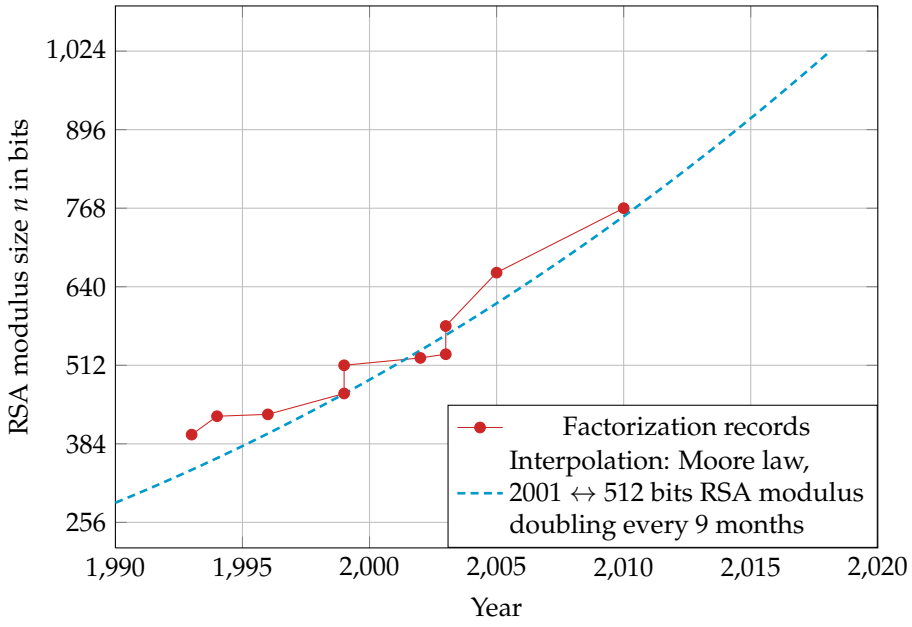


Figure 3.4: Records of RSA modulus factorization

Figure 3.4 presents the records of RSA modulus factorization and an interpolation according to [Len01, §3] by a Moore Law doubling every nine months.

We translate slightly the results on ECM complexity with recent records. We take the record of R. Propper of September 2013 (`http://www.loria.fr/~zimmerma/records/top50.html`). A 274-bit (83 digits) was factored from the 946-bit (285 digits) composite number $7^{337}+1$. We assume that a effort of order 2^{70} was provided so we adjust a constant δ such that $\log_2 E[1/2,\sqrt{2}](946,274)-\delta=70$. We obtain $\delta=36$. As before we denote by n the size in bits of the modulus to be factored and we denote by ℓ the size of the considered prime factor.

$$\text{ECM complexity in bits} = 2\log_2 n + \sqrt{2}\,\ell^{1/2}\,\log_2^{1/2}(\ell\ln 2) - 36\,.$$

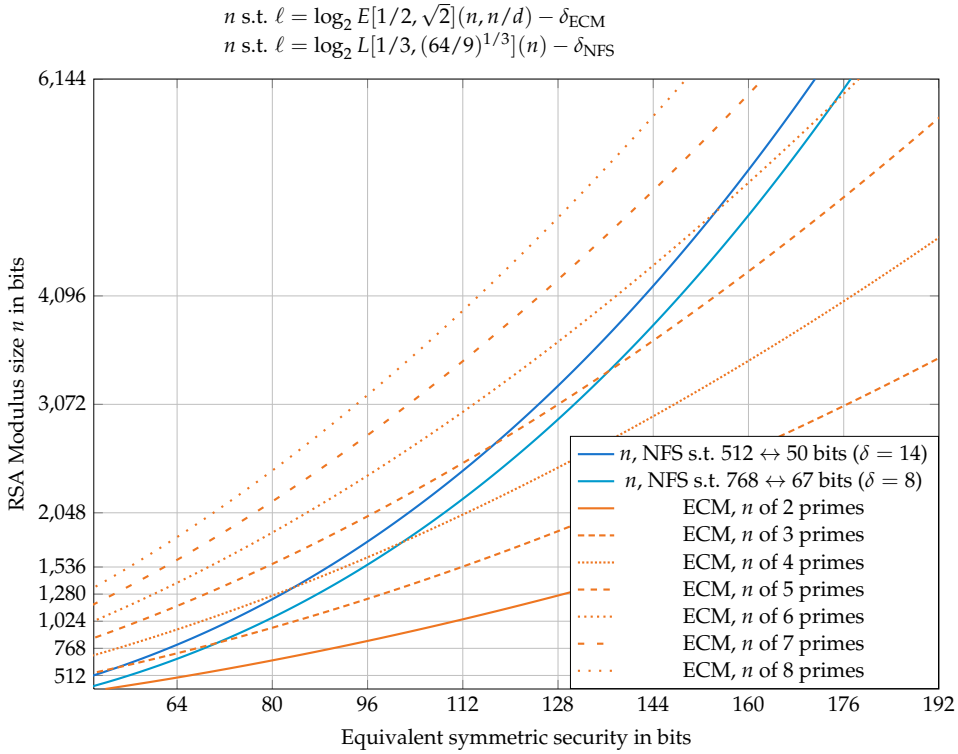


Figure 3.5: Estimated complexity of RSA modulus factorization with ECM method

To sum up, we obtain the two following formulas.

1. A two-prime RSA modulus N of n bits has a security equivalent to an s-bit symmetric key, with $s = \log_2 L[\frac{1}{3},\left(\frac{64}{9}\right)^{1/3}](n) - 14 = \left(\frac{64}{9}\right)^{1/3} n^{1/3}\log_2^{2/3}(n\ln 2) - 14$ according to Ecrypt recommendations [oEiCI12, §6.2.1], assuming that a 512-bit RSA modulus is equivalent to a 50-bit symmetric key.
2. A k-prime RSA modulus N of n bits has a security equivalent to an s-bit symmetric key, with $s = \log_2 E[\frac{1}{2},\sqrt{2}](n,\ell) - 36$ assuming that a 274-bit prime was factored from a 946-bit number in time complexity 2^{70} (`http://www.loria.fr/~zimmerma/records/ecmnet.html`).

The first line in Tab. 3.5 corresponds to ECRYPT recommendations. The threshold between NFS and ECM is represented through bold font. We do not consider security levels under 128 bits. For a 128-bit security level, a modulus of 3248 bits with two prime factors (of 1624 bits) is enough to prevent the NFS attack and the attack with ECM is much slower. This attack becomes slightly more efficient than the NFS one against a modulus with 6 prime factors (each of the same size). A modulus of 3664 bits instead of 3248 bits can be considered. For 8 primes in the modulus, the size is enlarged by 50%: 4840 bits instead of 3248 bits and each prime factor is 605-bit long. Table 3.5 could be used by protocol designers to set the size of the security parameter λ. Our Tab. 3.5 can also be used when setting the parameter sizes for protocols (or security proofs) relying on the Φ-hiding assumption. In 2010 at *CRYPTO*, Kiltz, O'Neill and Smith

[KOS10] used this assumption to obtain a nice result about RSA-OAEP. Then at *AFRICACRYPT'2011* Herrmann [Her11] explained new results about the security of this assumption. We emphasize that setting the security parameter λ in protocols is not completely straightforward if the modulus contains more than 5 prime factors. The NIST recommendations are also well known and the most widely used. The

Table 3.5: RSA-Multi-Prime modulus size from two up to nine prime factors, according to ECRYPT recommendations for the two prime factor case

Security Equivalence	AES-128		AES-192		AES-256	
Nb of primes in the modulus	$\log p_i$	$\log N$	$\log p_i$	$\log N$	$\log p_i$	$\log N$
2	1624	(ECRYPT) 3248	3968	7936	7724	15448
3	1083	3248	2646	7936	5150	15448
4	812	3248	1984	7936	3862	15448
5	650	3248	1587	7936	3090	15448
6	611	**3664**	1323	7936	2575	15448
7	608	**4256**	1147	**8024**	2207	15448
8	605	**4840**	1143	**9144**	1931	15448
9	603	**5424**	1140	**10256**	1829	**16456**

three main RSA modulus length are 3072, 7680 and 15360 to match respectively an AES-128, AES-192 and AES-256. We observe that to obtain an equivalence between a 3072 bit RSA modulus and an AES-128, the same equation 1 is used with $\delta = 10.7$ this time (instead of $\delta = 14$). So we translate also by -3.3 our computations with the ECM complexity and we obtain Tab. 3.6.

Table 3.6: RSA-Multi-Prime modulus size from two up to nine prime factors, according to NIST recommendations for the two-prime factor case

Security Equivalence	AES-128		AES-192		AES-256	
Nb of primes in the modulus	$\log p_i$	$\log N$	$\log p_i$	$\log N$	$\log p_i$	$\log N$
2	1536	(NIST) 3072	3840	7680	7680	15360
3	1024	3072	2560	7680	5120	15360
4	768	3072	1920	7680	3840	15360
5	615	3072	1280	7680	2560	15360
6	588	**3528**	1536	7680	3072	15360
7	584	**4088**	1115	**7808**	2194	15360
8	581	**4648**	1111	**8888**	1920	15360
9	579	**5208**	1108	**9976**	1789	**16104**

The conclusion is exaclty the same: up to 5 prime factors of same size un the modulus, at a 128-bit security level, the ECM method does not induces any consequence on the modulus size. Beyond that, the modulus size must be enlarged.

3.3.2 Composite-order elliptic curves

We introduced the pairings in the chapter 1, in Sec. 1.4. Let E be an elliptic curve defined over a prime field $\mathbb{F}_p$. A pairing is a bilinear, non-degenerate and efficient map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. From an algebraic point of view, $\mathbb{G}_1$ and $\mathbb{G}_2$ are two distinct subgroups of $E(\overline{\mathbb{F}_p})$, of same order n. If $n \mid \#E(\mathbb{F}_p)$ then $\mathbb{G}_1 \subset E(\mathbb{F}_p)$, this is the common setup. Let k be the smallest integer such that $n \mid p^k - 1$, k is the *embedding degree*. Then $\mathbb{G}_2 \subset E(\mathbb{F}_{p^k})$ and $\mathbb{G}_T \subset \mathbb{F}^*_{p^k}$. For supersingular or some of the $k = 1$ curves, an efficient isomorphism is available from $\mathbb{G}_1$ into $\mathbb{G}_2$. This gives a symmetric pairing and we can use the notation $\mathbb{G}_1 = \mathbb{G}_2$ to implicitly denote the use of the isomorphism in the pairing computation. In the remaining of this section, we will consider $\mathbb{G}_1$ and $\mathbb{G}_2$ as two distinct subgroups of E, of same order n. The target group $\mathbb{G}_T$ is the order-n (multiplicative) subgroup of $\mathbb{F}^*_{p^k}$. $\mathbb{G}_1$ and $\mathbb{G}_2$ have to be strong enough against a generic attack to a discrete logarithm problem. The third group $\mathbb{G}_T$ is more vulnerable because computing a discrete logarithm in a finite field is easier with the index calculus attack. Its size has to be enlarged.

Finding optimal pairing-friendly elliptic curves is an active field of research (see the survey [FST10]). At a 128-bit security level, the optimal choice would be to construct an elliptic curve whose order is a prime of 256 bits and over a prime finite field of the same size. For an embedding degree $k = 12$, an element in the third group is 3072 bit long in order to match the NIST recommendations. Such optimal pairing-friendly curves exist [BN05] (Barreto-Naehrig (BN) curves), but have a special form: the parameters p (defining the finite field), n (elliptic curve order) and t (trace) are given by degree 4 polynomials. We have $p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$, $n(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$ and $t(x) = 6x^2 + 1$. These BN curves are presented in Sec. 1.4.3.4 and their related pairing computation is explained in Sec. 3.2.

3.3.2.1 Issues in composite-order elliptic curve generation

For our particular purpose, the pairing-friendly elliptic curve order needs to contain a composite-order modulus N. Hence the order is chosen *before* the other curve parameters and no special form can be imposed to N. For example, finding such an elliptic curve over a non-prime field (e.g. in characteristic 2 or 3) is completely infeasible at the moment. As for BN curves, all the complete pairing-friendly elliptic curve families in the survey [FST10], defined by polynomials, are not convenient.

Secondly, the parameter sizes of composite-order elliptic curves are not optimal. The curve order is preferably chosen of the form hN with h a cofactor as small as possible. Due to the Hasse bound, the size of p (defining $\mathbb{F}_p$) is the same as the size of hN. This means that the prime field $\mathbb{F}_p$ already achieves the recommended size (say, 3072) to avoid an index calculus attack. Consequently, an embedding degree $k = 1$ is enough. As $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct, an embedding degree of 1 means that both $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of $E(\mathbb{F}_p)$, then $N^2 \mid E(\mathbb{F}_p)$ and $\log_2 p \geqslant 2 \log_2 N$. This mean that for a 3072 bit modulus N, p will have more than 6144 bits. Such curves exist, for example see [KM05, §6] or more recently [BRS11]. The elliptic curve point coordinates are more than 6144 bit long.

Tate pairing computation is described in Alg. 7. It consists in a Miller loop over the considered elliptic curve group order. A final exponentiation in $\mathbb{F}_{p^k}^*$ at the end is performed to obtain a unique pairing value. Optimal ate pairing computation on a BN curve is detailed in Alg. 18. Convenient supersingular curves do not benefit from pairing optimization such as η_T pairing, as the trace is zero (in large characteristic), or decomposition of the Miller loop length, as there is no efficiently computable endomorphism over $\mathbb{F}_p$ on such curves, except the scalar multiplication. For ordinary curves with $6 \mid k$ and $D = 3$ (BN curves) or $4 \mid k$ and $D = 1$, the complex multiplication induces an easy computable endomorphism thus permits to reduce the Miller loop length up to a factor 4.

Pairing computation over curves of embedding degree 2 needs multiplications over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ with $\log_2 p = 1536$. Pairing computation over curves of embedding degree 1 needs multiplications over $\mathbb{F}_p$ with $\log_2 p = 3072$. Recently in [ZZX12] it was shown that self-pairings on these particular curves may be speed-up thanks to the distortion map. Zhao et. al. gave efficient formulas of Weil pairing with denominator elimination thanks to the distortion map, although $k = 1$ instead of $k = 2$. Such ordinary $k = 1$ curves with efficient endomorphisms are rare. Few constructions are proposed in [BRS11]. More work is needed to determine in which cases pairings on these curves are competitive with $k = 2$ curves.

As mentioned in recent works, some properties (canceling, projecting) are achieved with only composite-order elliptic curves or only asymmetric pairings. More precisely, at *ASIACRYPT'2012*, Seo [Seo12] presented results on the impossibility of projecting pairings in certain cases. An ordinary composite-order elliptic curve is the only choice in this case. Such constructions are possible, see e.g. Boneh, Rubin and Silverberg paper [BRS11] but this seems to be the worst case in terms of parameter sizes and efficiency.

3.3.2.2 Our choices

If we want to reduce the size of p (hence of $\mathbb{G}_1$), we can choose a supersingular elliptic curve of embedding degree $k = 2$. This means that $\mathbb{G}_1 \subset E(\mathbb{F}_p)$, $\mathbb{G}_2 \nsubseteq E(\mathbb{F}_p)$ and both $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of $E(\mathbb{F}_{p^2})$.

$$\begin{array}{rcc|c|l} \mathbb{G}_1 \text{ and } \mathbb{G}_2 & \subset & E(\mathbb{F}_{p^2}) & N^2 & \mid \#E(\mathbb{F}_{p^2}) \\ & & \mid & & \\ \mathbb{G}_1 & \subset & E(\mathbb{F}_p) & N & \mid \#E(\mathbb{F}_p),\ N^2 \nmid \#E(\mathbb{F}_p) \end{array}$$

A supersingular elliptic curve of given subgroup order and embedding degree 2 is easy to construct:

1. Let N be a composite-order modulus.
2. Find the smallest integer h, $4 \mid h$, such that $hN - 1$ is prime.
3. Let $p = hN - 1$. The elliptic curve $E(\mathbb{F}_p) : y^2 = x^3 - x$ is supersingular, of order $hN = p + 1$ and embedding degree 2.

As $p = 3 \bmod 4$, -1 is not a square in $\mathbb{F}_p$. If $\mathbb{F}_{p^2} = \mathbb{F}_p[Z]/(Z^2 + 1)$, a distortion map is available: $\phi : E(\mathbb{F}_{p^2}) \to E(\mathbb{F}_{p^2})$, $(x, y) \mapsto (-x, Zy)$. In particular, $\phi(\mathbb{G}_1) = \mathbb{G}_2$ and the pairing is symmetric. As mentioned above, the improved pairing variant denoted η_T is not possible as this supersingular curve has trace 0 ($\#E(\mathbb{F}_p) = p + 1$). We implemented a Tate pairing on this curve. The parameter sizes for a security level equivalent to AES-128 are summarized in Tab. 3.7. We assume that the points on the elliptic curves are in compressed representation.

Table 3.7: Parameter sizes for prime order and composite order pairing-friendly elliptic curves, minimum and maximum in theory, according to Tab. 3.5 and Tab. 3.6

Elliptic curve, order			size of $\mathbb{G}_1$ order $\log_2 N$ min – max	size of elts in $\mathbb{G}_1$ $\log_2 p$ min – max	emb. deg. k	size of elts in $\mathbb{G}_2$	size of elts in $\mathbb{G}_T$ $k \log_2 p$ min – max
BN, prime order			256	256 – 269	12	512 – 538	3072 – 3248
supersingular curve	Prime order		256	1468 – 1624	2	As for elts in $\mathbb{G}_1$	2936 – 3248
	Composite order	2 primes	3072 – 3248	$\geqslant$ 3074 – $\geqslant$ 3250			$\geqslant$ 6148 – $\geqslant$ 6500
		3 primes	3072 – 3248	$\geqslant$ 3074 – $\geqslant$ 3250			$\geqslant$ 6148 – $\geqslant$ 6500
		4 primes	3072 – 3248	$\geqslant$ 3074 – $\geqslant$ 3250			$\geqslant$ 6148 – $\geqslant$ 6500
		5 primes	3072 – 3248	$\geqslant$ 3074 – $\geqslant$ 3250			$\geqslant$ 6148 – $\geqslant$ 6500
		6 primes	3528 – 3664	$\geqslant$ 3530 – $\geqslant$ 3666			$\geqslant$ 7060 – $\geqslant$ 7332
		7 primes	4088 – 4256	$\geqslant$ 4090 – $\geqslant$ 4258			$\geqslant$ 8180 – $\geqslant$ 8516
		8 primes	4648 – 4840	$\geqslant$ 4650 – $\geqslant$ 4842			$\geqslant$ 9300 – $\geqslant$ 9684
		9 primes	5208 – 5424	$\geqslant$ 5210 – $\geqslant$ 5426			$\geqslant$ 10420 – $\geqslant$ 10852

3.3.3 Theoretical estimation

In this section we will estimate the number of multiplications over the base field for each pairing in Tab. 3.7.

3.3.3.1 Prime order BN curve

We aim to implement a state of the art optimal ate pairing on a BN curve. We use various techniques described e.g. in [NNS10, BGDM+10]. A careful operation count is detailed in Alg. 18 (see Sec. 3.2). We use the finite field arithmetic described in [DhSD06b] and [GS10] for speeding up the pairing final exponentiation and exponentiations in $\mathbb{G}_T$. Operation counts in Tab. 3.8 describe our choices according to recommendations made in [DhSD06b]. The arithmetic operations in $\mathbb{F}_p$ are denoted M_p for a multiplication, S_p for a square, I_p for an inversion and HW denotes the Hamming weight. We build the extensions as $\mathbb{F}_{p^2} = \mathbb{F}_p[X]/(X^2 - \alpha)$, $\mathbb{F}_{p^6} = F_{p^2}[Y]/(Y^3 - \beta)$, $\mathbb{F}_{p^{12}} = F_{p^6}[Z]/(Z^2 - \gamma)$. M_α, M_β and M_γ denote resp. a multiplication by α, β and γ, performed with few additions if α, β and γ are well chosen. For exponentiation in $\mathbb{F}_{p^k}$, $S_{\Phi_6(p^2)}$ denotes the improved squaring formula from [GS10]. Details are provided in Alg. 18 which computes $e_{\text{OptAte}}(P, \psi_6(Q)) = f^{\frac{p^{12}-1}{r}}$ with
$f = f_{6x+2,\psi_6(Q)}(P) \cdot \ell_{[6x+2]\psi_6(Q),\pi_p(\psi_6(Q))}(P) \cdot \ell_{[6x+2]\psi_6(Q)+\pi_p(\psi_6(Q)),-\pi_p^2(\psi_6(Q))}(P)$ with ψ_6 the sextic twist map, π_p the p-power Frobenius and π_{p^2} the p^2-power Frobenius.

Table 3.8: Approximation of arithmetic operations in finite field extensions

$M_{p^{12}}$	=	$3M_{p^6} + 5A_{p^6} + 1M_\gamma$	$\to 54M_p$	$S_{p^{12}}$	=	$2M_{p^6} + 4A_{p^6} + 2M_\gamma$	$\to 36M_p$
M_{p^6}	=	$6M_{p^2} + 13A_{p^2} + 2M_\beta$	$\to 18M_p$	S_{p^6}	=	$2M_{p^2} + 3S_{p^2} + 10A_{p^2} + 2M_\beta$	$\to 12M_p$
M_{p^2}	=	$3M_p + 5A_p + 1M_\alpha$	$\to 3M_p$	S_{p^2}	=	$2M_p + 4A_p + 2M_\alpha$	$\to 2M_p$

3.3.3.2 Supersingular curve

A Tate pairing may not benefit from the previous optimizations. We can still simplify the Miller loop thanks to the even embedding degree ($k = 2$). The denominators cancel in the final exponentiation thus we can remove them in the computations. Details are provided in Alg. 7 (see Sec. 1.4.4.2) with ϕ the distortion map from $\mathbb{G}_1$ into $\mathbb{G}_2$.

The algorithm for a supersingular elliptic curve of composite order is the same as Alg. 7. In addition, we take $m = N$ the modulus, hence $\log_2 m = 3072$ for example. By construction, the cofactor h will be as small as possible, resulting in very cheap final exponentiation, e.g. $\log_2 h = 12$. We detail in Tab. 3.9 the different estimations for a pairing computation.

Table 3.9: Estimations for pairings on prime-order and composite-order elliptic curves, assuming that for a composite-order supersingular curve, $\log_2 N$ is as in Tab. 3.7, $\mathrm{HW}(N) = \log_2 N/2$, $\log_2 h = 12$ and $\mathrm{HW}(h) = 5$ and we use Alg. 7, and for a BN curve, $\log_2 n = \log_2 p = 256$, $\mathrm{HW}(x) = 4, \mathrm{HW}(6x+5) = 10, \mathrm{HW}(6x^2+1) = 33$.

Curve	Pairing	nb primes	Miller loop min – max	Final exp. (+ I_p) min – max
BN	opt. ate	1	$7204\ M_p$	$6669\ M_p$
supersingular (SsC)	Tate	1	$4224M_p + 1728S_p$	$3730M_p - 4745M_p$
		2	$61440M_p + 23040S_p$/ $64960M_p + 24360S_p$	$41M_p + I_p$
		3	$61440M_p + 23040S_p$/ $64960M_p + 24360S_p$	
		4	$61440M_p + 23040S_p$/ $64960M_p + 24360S_p$	
		5	$61440M_p + 23040S_p$/ $64960M_p + 24360S_p$	
		6	$70560M_p + 26460S_p$/ $73280M_p + 27480S_p$	
		7	$81760M_p + 30660S_p$/ $85120M_p + 31920S_p$	
		8	$92960M_p + 34860S_p$/ $96800M_p + 36300S_p$	
		9	$104160M_p + 39060S_p$/$108480M_p + 40680S_p$	

3.3.4 Implementation results

We implemented in C the above pairings (Tab. 3.7), we compiled with gcc 4.4.3 and ran the software implementation on a 2.6 GHz Intel Celeron 64 bits PC with 1 GB RAM and Ubuntu 10.04.4 LTS OS. The developed code is part of a proprietary library, the LibCryptoLCH developed at Thales Communications & Security (France). The finite field arithmetic uses the Montgomery representation and the modular multiplication is written in x86-64 assembly language. Our timings are competitive compared to others proprietary generic libraries such as the one used at Microsoft Research [ALNS12]. The Authors in [ALNS12] develop a C library then add different optimized assembly part of code for x86 or ARMv7 processors. They run their library on a x86-64, Intel Core2 E6600 @ 2.4 GHz, Windows 7 (64-bit) and on a ARM, dual-core Cortex A9 @ 1GHz, Windows device. They obtain a pairing on average at 55.19 ms (ARM) and 6.31 ms (x86-64) in projective coordinates and 51.01 ms (ARM) and 5.92 ms (x86-64) in affine coordinates, over a BN curve of 254 bit prime order group. Our timings are slower than other state-of-the-art ones can be ([NNS10, AKL+11]) because our software is not optimized for a particular sparse prime number which might result in very specific and optimized modular reduction.

Results are presented in Fig. 3.6. We present in Tab. 3.10 our results for a BN curve, a prime-order and a composite two-prime order supersingular curve. The first line shows our results of an implementation of an optimal ate pairing on a Barreto-Naehrig curve. See for example [Ver10, BGDM+10, NNS10] on how to implement it efficiently. We choose a quite sparse but still random parameter $x = \texttt{0x580000000000100d}$ resulting in quite sparse prime order and prime field. Our modular reduction is not optimized for this value. Our extension field is optimized for towers built with binomials with small coefficients. For instance the first extension is built as $\mathbb{F}_{p^2} \simeq \mathbb{F}_p[X]/(X^2+1)$ as $p \equiv 3 \bmod 4$ which allows a fast reduction $\bmod X^2+1$ in the Karatsuba multiplication. The second extension is built as $\mathbb{F}_{p^{12}} \simeq \mathbb{F}_{p^2}[Y]/(Y^6-2)$ resulting in fast polynomial reduction too. Our implementation perform a pairing in 5.05 ms in average which is comparable to the 5.73 ms over an x86-64 Intel Core2 E6600 of the Microsoft Research Team [ALNS12, Tab.2].

Table 3.10: Timings for exponentiation in milliseconds (ms), Ate and Tate pairings on prime order n and composite order $n = n_1 \cdots n_i$ elliptic curves for different security levels.

Pairing	$\log_2 n$	$\log_2 n_i$	$\log_2 p$	$k \cdot \log_2 p$	Miller Loop	F. Exp.	Pairing
BN,o.ate	256	–	256	3072	2.35	2.70	5.05
	269	–	269	3228	3.22	3.80	7.29
(1), Tate	256	–	1536	3072	19.70	20.50	40.20
(2), Tate	1024	512	1036	3072	56.88	0.10	56.98
(2), Tate	2048	1024	2059	4118	392.50	0.40	392.90
(2), Tate	3072	1536	3083	6166	1295.6	0.7	1296.3
(3), Tate	3072	1024	3083	6166	1275.6	0.7	1276.3

Pairing	Exp. $\mathbb{G}_1$	g^{p_i} $\mathbb{G}_1$	Exp. $\mathbb{G}_2$	Exp. $\mathbb{G}_T$	g^{p_i} $\mathbb{G}_T$
BN,o.ate	0.55	–	1.91	5.16	–
	0.77	–	2.56	5.98	–
(1), Tate	8.30	–	–	2.20	–
(2), Tate	24.38	13.12	–	7.81	3.9
(2), Tate	172.5	86.25	–	50.63	25.8
(2), Tate	586.2	301.8	–	166.10	81.9
(3), Tate	556.9	222.5	–	174.88	60.1

In 2012 Zhang et al. in [ZXW+12] published an optimized implementation of composite-order bilinear pairings on GPU. They obtained a very efficient execution time of 17.4 ms, 11.9 ms and 8.7 ms per pairing in average with a 1024 bit modulus on three different GPU [ZXW+12, §8]. With PBC library [Lyn14] on an Intel Core 2 E8300 CPU at 2.83 GHz and 3GB RAM they obtained 171.1 ms. With our library on an Intel Celeron as specified above, we obtain 57 ms for a pairing over a 1024 bit modulus order elliptic curve and 393 ms for a 2048 bit modulus order. This means our library is two times faster than PBC in this setting, mostly because of our x86-64 implementation of the multiplication in $\mathbb{F}_p$. We present in Fig. 3.6 our timings for pairing and scalar multiplication on supersingular composite-order elliptic curves. We also present in Fig. 3.7 our benchmark results, plotted with a logarithmic scale to visualize also the timings for pairings on BN curves.

For this 128-bit security level, a pairing on an elliptic curve of composite order with two primes is 254 times slower than over a prime-order elliptic curve (1.27 s compared to 5.05 ms). The Miller loop is very expensive, indeed it runs over N. The only possible optimizations may use techniques such as sliding-window. The final exponentiation is very cheap because it consists in $f^{(p-1)h} = (f^p \cdot f^{-1})^h$ computed with one inversion, one multiplication, one Frobenius map and one very small exponentiation (h is only a dozen bits) in $\mathbb{F}_{p^2}$.

3.3.4.1 Application to BGN cryptosystem

In 2005, Boneh, Goh and Nissim published in [BGN05] a somewhat homomorphic encryption scheme which can add several times different ciphertexts, perform one multiplication then continue to add ciphertexts. Freeman proposed a conversion to a prime-order setting in [Fre10]. We compare the two settings. Our results show that the protocol is much slower on a composite-order elliptic curve, as presented in Tab. 3.11.

Protocol Setup(τ)

1. Generate two random τ-bit primes p_1, p_2 and set $N = p_1 p_2$.
2. Generate a (symmetric) bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ with $\mathbb{G}_1$ and $\mathbb{G}_T$ of order N.
3. Pick two random generators $g_1, u_1 \leftarrow \mathbb{G}_1$ and set $u_{1(p_1)} = u_1^{p_2} \Rightarrow u_{1(p_1)}$ is a random generator of the subgroup of order p_1 of $\mathbb{G}_1$. We denote by $\mathbb{G}_{1(p_1)}$ this subgroup. Set $g_T = e(g_1, g_1)$ as generator of $\mathbb{G}_T$ and $h_T = e(g_1, u_{1(p_1)}) = g_T^{p_2}$ as generator of the subgroup $\mathbb{G}_{T(p_1)}$ of order p_1 of $\mathbb{G}_T$.

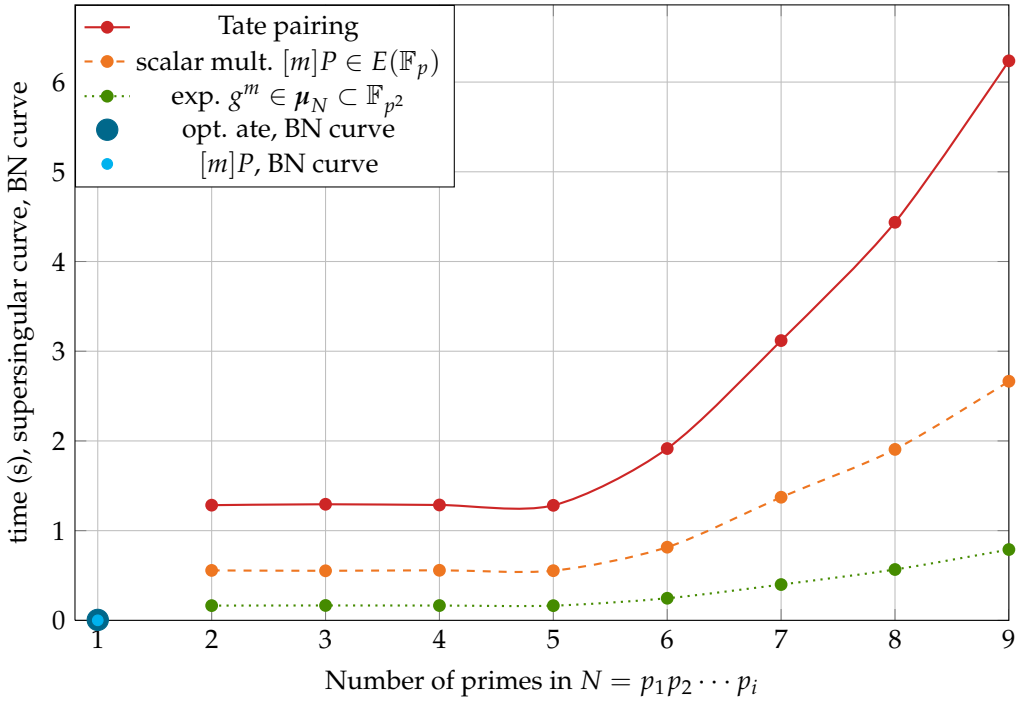


Figure 3.6: Average execution time (s) for a scalar multiplication on $E(\mathbb{F}_p)$, an exponentiation in $\mu_N \subset \mathbb{F}_{p^2}$ and a Tate pairing over a composite-order supersingular curve, with modulus sizes from Tab. 3.6 col. 1.

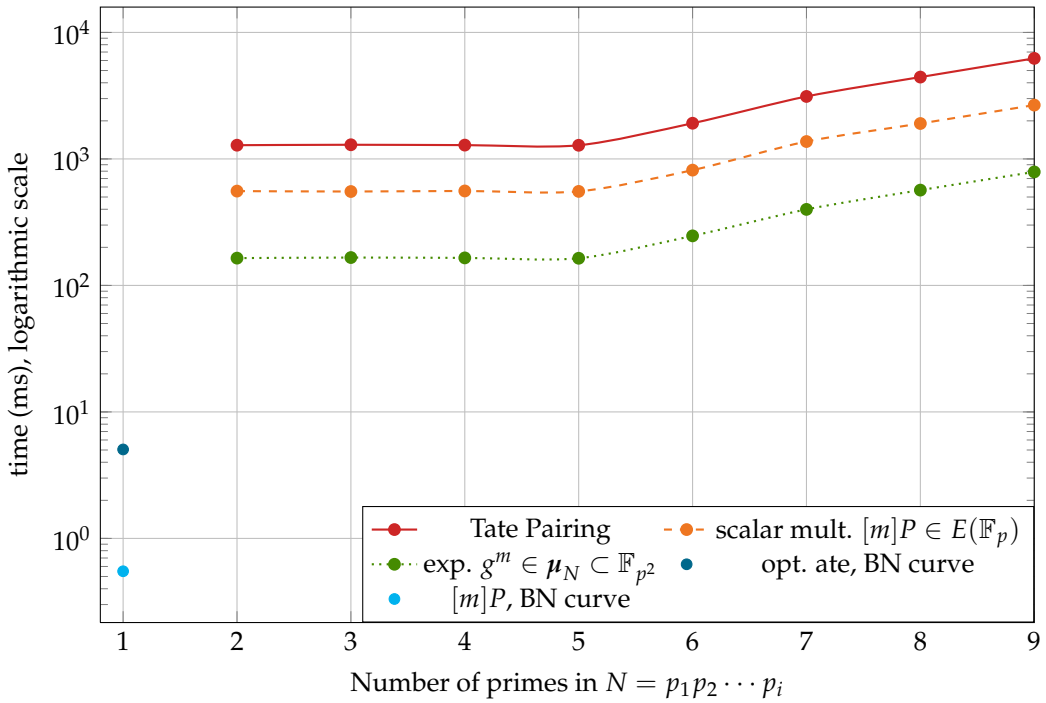


Figure 3.7: Average execution time (ms) for a scalar multiplication on $E(\mathbb{F}_p)$, an exponentiation in $\mu_N \subset \mathbb{F}_{p^2}$, an opt. ate pairing on a BN curve and a Tate pairing over a composite-order supersingular curve. We can see the gap from prime-order to composite-order groups in terms of efficiency.

4. $\mathcal{PK} = (N, \mathbb{G}_1, \mathbb{G}_T, e, g_1, u_{1(p_1)}, g_T, h_T)$. $\mathcal{SK} = p_1$.

Encrypt($\mathcal{PK}, m$): $m \in \mathbb{N}, m < p_2$. Pick a random $r \leftarrow \{0, 1, \ldots, N-1\}$. The ciphertext is

$$c = g_1^m \cdot u_{1(p_1)}^r \in \mathbb{G}_1 .$$

Homomorphic Addition (c_1, c_2) mod N, $\in \mathbb{G}_1$. Pick a random $r \leftarrow \{0, 1, \ldots, N-1\}$.

$$c = c_1 \cdot c_2 \cdot u_{1(p_1)}^r = g_1^{m_1+m_2 \bmod N} \cdot u_{1(p_1)}^{r'} \in \mathbb{G}_1 .$$

Decrypt($\mathcal{SK}, c \in \mathbb{G}_1$): We have $c^{p_1} = (g_1^m \cdot u_{1(p_1)}^r)^{p_1} = (g_1^{p_1})^m$. Compute the discrete log of c^{p_1} in base $g_1^{p_1}$. This is very slow or m must be very small (few bits). Since the discrete logarithm value is in a small interval, one may use the method described in [BL12].
Homomorphic Multiplication (c_3, c_4) mod N (once). Pick a random $r \leftarrow \{0, 1, \ldots, N-1\}$.

$$c = e(c_3, c_4) \cdot h_T^r = g_T^{m_3 \cdot m_4 \bmod N} \cdot h_T^{r'} \in \mathbb{G}_T .$$

Homomorphic Addition (c_5, c_6) mod $N \in \mathbb{G}_T$. Pick a random $r \leftarrow \{0, 1, \ldots, N-1\}$.

$$c = c_5 \cdot c_6 \cdot h_T^r = g_T^{m_5+m_6 \bmod N} \cdot h_T^{r'} \in \mathbb{G}_T .$$

Decrypt($\mathcal{SK}, c \in \mathbb{G}_T$). Compute c^{p_1} then its discrete log in base $g_T^{p_1}$.

Implementation. In the Encrypt step of the BGN protocol, a random r is picked in $\{0, 1, \ldots, N-1\}$ with $N = p_1 p_2$ the RSA modulus. Then $u_{1(p_1)}^r$ is computed. The size of r is up to 3072 bits. We used the same curve as in Tab. 3.10, the line with $\log_2 N = 3072$ and $\log_2 p_i = 1536$. We assumed that to compute several pairings on the same curve, we compute each Miller loop, then multiply the outputs and apply a single final exponentiation. There are four distinct products of two or three pairings in the second protocol.

Table 3.11: Timings for the BGN protocol over a composite order elliptic curve and its equivalent over a prime order elliptic curve for a security level equivalent to AES-128. We don't consider the discrete log computation, see e.g. [BL12] for efficient DL computation in this particular setting.

Operation	Composite-order E.C. [BGN05, §3]		Prime-order E.C. [Fre10, §5]	
Encrypt or Add	1 exp. in $\mathbb{G}_1$	1300 ms	1 exp. in $\mathbb{G}_1$ and $\mathbb{G}_2$	3.8 ms
Decrypt	$C^{p_1} \in \mathbb{G}_1$	645 ms	π_1: 4 exp. in $\mathbb{G}_1$ π_2: 4 exp. in $\mathbb{G}_2$	4.0 ms 11.2 ms
Multiply	1 pairing + 1 exp. in $\mathbb{G}_T$	3364 ms	1 exp. in $\mathbb{G}_1$ and $\mathbb{G}_2$ + 4×(3 pairings)	119.8 ms
Encrypt or Add	1 exp. in $\mathbb{G}_T$	409 ms	1 exp. in $\mathbb{G}_1$ and $\mathbb{G}_2$ + 4×(2 pairings)	87.8 ms
Decrypt (without DL)	$C^{p_1} \in \mathbb{G}_T$	204 ms	$\pi_t(C)$ 16 exp. in $\mathbb{G}_T$	108.8 ms

The arithmetic on the composite-order elliptic curve $E(\mathbb{F}_p)$ is more than 3 times slower than in $\mathbb{G}_T \subset \mathbb{F}_{p^2}$, this means that the encryptions and exponentiations for decryption in $\mathbb{G}_T$ are more efficient. The converse is observed over a prime-order elliptic curve. This protocol over an optimal prime-order elliptic curve is dramatically faster than over a composite-order elliptic curve. More precisely, the exponentiation in the decryption step is 161 times faster in $\mathbb{G}_1$, 57 times faster in $\mathbb{G}_2$ and 2 times faster in $\mathbb{G}_T$ over a prime-order elliptic curve than over a composite-order one.

3.3.4.2 Application to Hierarchical Identity Based Encryption

In this section, we detail and implement the Hierarchical Identity Based Encryption (HIBE in the following) of Lewko and Waters published at *EUROCRYPT'2011* [LW11] and compare it with its translation in the prime-order setting due to Lewko [Lew12]. Any random value is picked uniformly at random from the considered set.

Lewko-Waters HIBE scheme. We only recall the Setup, KeyGen, Encrypt, Delegate and Decrypt steps. The complete description of the scheme with the security proofs are available in [LW11].

Setup($\lambda \rightarrow$ PP, MSK). The setup algorithm takes as input the security parameter λ (e.g. see Tab. 3.5 to select an appropriate λ) and chooses a bilinear group $\mathbb{G}_1$ of order $N = p_1p_2p_3$, where p_1, p_2, p_3 are distinct primes. Let $\mathbb{G}_{1(p_i)}$ denote the subgroup of order p_i in $\mathbb{G}_1$. The algorithm then picks g, u, h, v, w from $\mathbb{G}_{1(p_1)}$, and α from $\mathbb{Z}_N$. It sets the public parameters as:

$$\text{PP} := \{N, \mathbb{G}_1, g, u, h, v, w, e(g,g)^{\alpha}\} .$$

The master secret key is α.

KeyGen($(\mathcal{I}_1, \ldots, \mathcal{I}_j)$, MSK, PP) $\rightarrow$ $\text{SK}_{\vec{\mathcal{I}}}$. The key generation algorithm picks at random values $r_1, \ldots, r_j$, $y_1, \ldots, y_j$ from $\mathbb{Z}_N$. It also picks random values $\lambda_1, \ldots, \lambda_j \in \mathbb{Z}_N$ subject to the constraint that $\alpha = \lambda_1 + \lambda_2 + \ldots + \lambda_j$. The secret key is computed as:

$$K_{i,0} := g^{\lambda_i} w^{y_i},\ K_{i,1} := g^{y_i},\ K_{i,2} := v^{y_i}(u^{\mathcal{I}_i}h)^{r_i},\ K_{i,3} := g^{r_i}\ \forall i \in \{1, \ldots, j\} .$$

Encrypt(M, $(\mathcal{I}_1, \ldots, \mathcal{I}_j)$, PP), $\rightarrow$ CT. The encryption algorithm picks s, t_1, ..., t_j randomly from $\mathbb{Z}_N$. It creates the ciphertext as:

$$C := M \cdot e(g,g)^{\alpha s}, C_0 := g^s,$$

$$C_{i,1} := w^s v^{t_i},\ C_{i,2} := g^{t_i},\ C_{i,3} := (u^{\mathcal{I}_i}h)^{t_i}\ \forall i \in \{1, \ldots, j\} .$$

Delegate(PP, SK, $\mathcal{I}_{j+1}$) $\rightarrow$ $SK^{'}$. $\mathcal{I}_{j+1}$ denotes the identity of a group under $\mathcal{I}_j$ in the hierarchy. The delegation algorithm takes in a secret key $SK = \{K_{i,0}, K_{i,1}, K_{i,2}, K_{i,3}\ \forall i \in \{1, \ldots, j\}\}$ for $(\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_j)$ and a level $j+1$ identity $\mathcal{I}_{j+1}$. It produces a secret key $SK^{'}$ for $(\mathcal{I}_1, \ldots, \mathcal{I}_{j+1})$ as follows. It picks $y_1^{'}, \ldots, y_{j+1}^{'}$ and $r_1^{'}, \ldots, r_{j+1}^{'} \in \mathbb{Z}_N$ at random, $\lambda_1^{'}, \ldots, \lambda_{j+1}^{'} \in \mathbb{Z}_N$ randomly up to the constraint that $\lambda_1^{'} + \ldots + \lambda_{j+1}^{'} = 0$ and computes:

$$K_{i,0}^{'} := K_{i,0} \cdot g^{\lambda_i^{'}} \cdot w^{y_i^{'}}, \qquad K_{i,1}^{'} := K_{i,1} \cdot g^{y_i^{'}},$$
$$K_{i,2}^{'} := K_{i,2} \cdot v^{y_i^{'}}(u^{\mathcal{I}_i}h)^{r_i^{'}}, \quad K_{i,3}^{'} := K_{i,3} \cdot g^{r_i^{'}}, \quad \forall i \in \{1, \ldots, j+1\},$$

where $K_{j+1,1}, K_{j+1,2}, K_{j+1,3}$ are defined to be the identity element in $\mathbb{G}_1$.

Decryption(CT, SK) $\rightarrow$ M. The decryption algorithm takes in a secret key SK = $\{K_{i,0}, K_{i,1}, K_{i,2}, K_{i,3}$ $\forall i \in \{1, \ldots, j\}\}$ for $(\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_j)$ and a ciphertext CT encrypted to $(\mathcal{I}_1, \ldots, \mathcal{I}_\ell)$. Assuming $(\mathcal{I}_1, \ldots, \mathcal{I}_j)$ is a prefix of $(\mathcal{I}_1, \ldots, \mathcal{I}_\ell)$, the message is decrypted as follows. The decryption algorithm computes:

$$B := \prod_{i=1}^{j} \frac{e(C_0, K_{i,0}) \cdot e(C_{i,2}, K_{i,2})}{e(C_{i,1}, K_{i,1}) \cdot e(C_{i,3}, K_{i,3})} .$$

The message is then computed as $M = C/B$.

Lewko HIBE translation in prime order bilinear group. We also studied the Lewko HIBE translation in prime order bilinear group. We only consider in Tab. 3.13 the Setup, Encrypt, KeyGen, Delegate and Decrypt steps writen only from practical point of view, with $m = 6$ the dimension of the group $\mathbb{G}$ used ($\mathbb{G} = \mathbb{G}_1^m$). For a complete description of the scheme with $m = 10$ for the security proof, see [Lew12, §B.3] and [Lew12, §2.2] for notations. Moreover the scheme in [Lew12] is described with a symmetric pairing. We apply the protocol to an asymmetric pairing to improve its practical efficiency. There are two possible approaches. We can set the secret keys in $\mathbb{G}_1$ and the ciphertexts in $\mathbb{G}_2$ to optimize the needs in secured memory which can be quite expensive in constrained devices. Or we can set in $\mathbb{G}_2$ the secrets keys (with

Table 3.12: Lewko and Waters HIBE scheme over a composite order bilinear group.

Operation	Randomness complexity	Computation	Timing $j=3$ Tab. 3.10
Setup	$N=p_1p_2p_3$, 5 elts $\in \mathbb{G}_{1(p_1)}$, 1 elt $\in \mathbb{Z}_N$	1 pairing	1.27 s
KeyGen	$3j-1$ elts in $\mathbb{Z}_N$	$7j$ exp. in $\mathbb{G}_1$	11.55 s
Encrypt	$j+1$ elts $\in \mathbb{Z}_N$	$4+4j$ exp. in $\mathbb{G}_1$, 1 exp. in $\mathbb{G}_T$	8.96 s
Delegate $j \to j+1$	$3j+2$ elts in $\mathbb{Z}_N$	$7(j+1)$ exp. in $\mathbb{G}_1$	15.40 s
Decryption	–	$4j$ pairings	5.08 s

double secured memory) and set in $\mathbb{G}_1$ the ciphertexts to improve the bandwidth. We will choose this second option.

Vectors of group elements are considered and denoted $\vec{v}=(v_1,\ldots,v_m)\in\mathbb{F}_r^m$ (with r the subgroup of prime order of an elliptic curve), and for $g_1\in\mathbb{G}_1$ (we recall that this is an elliptic curve and not a finite field despite the multiplicative notation),

$$g_1^{\vec{v}}=(g_1^{v_1},g_1^{v_2},\ldots,g_1^{v_m})\in\mathbb{G}_1^m\,. \tag{3.21}$$

Moreover, for any $a\in\mathbb{F}_r$ and $\vec{v},\vec{w}\in\mathbb{F}_r^m$, we have:

$$g_1^{a\vec{v}}=(g_1^{av_1},g_1^{av_2},\ldots,g_1^{av_m}),\ g_1^{\vec{v}+\vec{w}}=(g_1^{v_1+w_1},g_1^{v_2+w_2},\ldots,g_1^{v_m+w_m})\,. \tag{3.22}$$

The corresponding pairing is defined as follows, with e a bilinear pairing $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$:

$$e_m(g_1^{\vec{v}},g_2^{\vec{w}})=\prod_{i=1}^{m}e(g_1^{v_i},g_2^{w_i})=e(g_1,g_2)^{\vec{v}\cdot\vec{w}}\in\mathbb{G}_T\subset\mathbb{F}_{p^k}^*\,. \tag{3.23}$$

The pairing e_m costs m pairings e. More precisely, as e_m is a product of m pairings, it costs m Miller loops then one final exponentiation if we set e to be a (variant of a) Tate pairing.

Setup($\lambda\to$ PP, MSK). The setup algorithm takes in the security parameter λ and chooses a bilinear group $\mathbb{G}_1$ of sufficiently large prime order r and a generator g_1; $\mathbb{G}_2$ of same prime order r with a generator g_2 and finally $\mathbb{G}_T$ of same order r. Let $g_T=e(g_1,g_2)$ be a generator of G_T. Let $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ denote the bilinear map. We set $m=6$. Hence

$$\begin{array}{rcl} e_m=e_6:\mathbb{G}_1^6\times\mathbb{G}_2^6 & \to & \mathbb{G}_T \\ (g_1^{\vec{v}},g_2^{\vec{w}}) & \mapsto & \prod_{i=1}^6 e(g_1^{v_i},g_2^{w_i}) \end{array}$$

The algorithm samples random dual orthonormal bases, $(\mathbb{D},\mathbb{D}^*)\leftarrow\mathrm{Dual}(\mathbb{F}_r^m)$. Let $\vec{d}_1,\ldots,\vec{d}_6$ denote the elements of $\mathbb{D}$ and $\vec{d}_1^*,\ldots,\vec{d}_6^*$ denote the elements of $\mathbb{D}^*$. They satisfy the property $\vec{d}_i\cdot\vec{d}_i^*=\psi\in\mathbb{F}_r^*\ \forall i$ and $\vec{d}_i\cdot\vec{d}_j^*=0\ (\mathrm{mod}\, r)$ for $i\neq j$. It also picks random exponents $\alpha_1,\alpha_2,\theta,\sigma,\gamma,\xi\in\mathbb{F}_r$. The public parameters are

$$\mathrm{PP}=\left\{\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,r,e(g_1,g_2)^{\alpha_1\vec{d}_1\cdot\vec{d}_1^*},e(g_1,g_2)^{\alpha_2\vec{d}_2\cdot\vec{d}_2^*},g_1^{\vec{d}_1},\ldots,g_1^{\vec{d}_6}\right\}, \tag{3.24}$$

and the master secret key is

$$\mathrm{MSK}=\left\{\alpha_1,\alpha_2,g_2^{\vec{d}_1^*},g_2^{\vec{d}_2^*},g_2^{\gamma\vec{d}_1^*},g_2^{\xi\vec{d}_2^*},g_2^{\theta\vec{d}_3^*},g_2^{\theta\vec{d}_4^*},g_2^{\sigma\vec{d}_5^*},g_2^{\sigma\vec{d}_6^*}\right\}\,. \tag{3.25}$$

KeyGen($(\mathcal{I}_1,\ldots,\mathcal{I}_j)$, MSK, PP) $\to \mathrm{SK}_{\vec{\mathcal{I}}}$. The key generation algorithm picks at random values $r_1^i,r_2^i\in\mathbb{F}_r$ for $1\leqslant i\leqslant j$. It also picks random values $y_1,\ldots,y_j\in\mathbb{F}_r$ and $w_1,\ldots,w_j\in\mathbb{F}_r$ up to the constraint that $y_1+y_2+\ldots+y_j=\alpha_1$ and $w_1+w_2+\ldots+w_j=\alpha_2$. For each $1\leqslant i\leqslant j$ it computes $K_i:=g_2^{y_i\vec{d}_1^*+w_i\vec{d}_2^*+r_1^i\mathcal{I}_i\theta\vec{d}_3^*-r_1^i\theta\vec{d}_4^*+r_2^i\mathcal{I}_i\sigma\vec{d}_5^*-r_2^i\sigma\vec{d}_6^*}\in\mathbb{G}_2$. The secret key is created as:

$$\mathrm{SK}_{\mathcal{I}}:=\left\{g_2^{\gamma\vec{d}_1^*},g_2^{\xi\vec{d}_2^*},g_2^{\theta\vec{d}_3^*},g_2^{\theta\vec{d}_4^*},g_2^{\sigma\vec{d}_5^*},g_2^{\sigma\vec{d}_6^*},K_1,\ldots,K_j\in\mathbb{G}_2\right\}. \tag{3.26}$$

Encrypt(M, $(\mathcal{I}_1, \ldots, \mathcal{I}_j)$, PP), $\rightarrow$ CT. The encryption algorithm picks s_1, s_2 and t_1^i, t_2^i for $1 \leqslant i \leqslant j$ randomly from $\mathbb{F}_r$. It computes

$$C_0 := M \cdot e(g_1, g_2)^{\alpha_1 s_1 \vec{d}_1 \cdot \vec{d}_1^*} \cdot e(g_1, g_2)^{\alpha_2 s_2 \vec{d}_2 \cdot \vec{d}_2^*} \in \mathbb{G}_T \tag{3.27}$$

(note that $e(g_1, g_2)^{\alpha_1 \vec{d}_1 \cdot \vec{d}_1^*}$ and $e(g_1, g_2)^{\alpha_2 \vec{d}_2 \cdot \vec{d}_2^*}$ are precomputed and stored in PP). It computes also

$$C_i := g_1^{s_1 \vec{d}_1 + s_2 \vec{d}_2 + t_1^i \vec{d}_3 + \mathcal{I}_i t_1^i \vec{d}_4 + t_2^i \vec{d}_5 + \mathcal{I} t_2^i \vec{d}_6} \tag{3.28}$$

for $1 \leqslant i \leqslant j$. The ciphertext is CT := $\left\{C_0 \in \mathbb{G}_T, C_1, \ldots, C_j \in \mathbb{G}_1\right\}$.
Delegate(PP, $\mathrm{SK}_{\vec{\mathcal{I}}}$, $\mathcal{I}_{j+1}$) $\rightarrow \mathrm{SK}_{\vec{\mathcal{I}}|\mathcal{I}_{j+1}}$. The delegation algorithm picks random values $\omega_1^i, \omega_2^i \in \mathbb{F}_r$ for $1 \leqslant i \leqslant j+1$. It also picks random values $y_1', \ldots, y_j' \in \mathbb{F}_r$ and $w_1', \ldots, w_j' \in \mathbb{F}_r$ up to the constraint that $y_1' + y_2' + \ldots + y_{j+1}' = 0$ and $w_1' + w_2' + \ldots + w_{j+1}' = 0$. The delegation algorithm takes in a secret key $SK_{\vec{\mathcal{I}}}$ with elements denoted as above. It computes $K_i' := K_i \cdot g_2^{y_i' \gamma \vec{d}_1^* + w_i' \xi \vec{d}_2^* + \omega_1^i \mathcal{I}_i \theta \vec{d}_3^* - \omega_1^i \theta \vec{d}_4^* + \omega_2^i \mathcal{I}_i \sigma \vec{d}_5^* - \omega_2^i \sigma \vec{d}_6^*} \in \mathbb{G}_2$ for $1 \leqslant i \leqslant j$ and $K_{j+1} := g_2^{y_{j+1}' \gamma \vec{d}_1^* + w_{j+1}' \xi \vec{d}_2^* + \omega_1^{j+1} \mathcal{I}_{j+1} \theta \vec{d}_3^* - \omega_1^{j+1} \theta \vec{d}_4^* + \omega_2^{j+1} \mathcal{I}_{j+1} \sigma \vec{d}_5^* - \omega_2^{j+1} \sigma \vec{d}_6^*} \in \mathbb{G}_2$. $SK_{\vec{\mathcal{I}}|\mathcal{I}_{j+1}}$ is formed as

$$\left\{ g_2^{\gamma \vec{d}_1^*}, g_2^{\xi \vec{d}_2^*}, g_2^{\theta \vec{d}_3^*}, g_2^{\theta \vec{d}_4^*}, g_2^{\sigma \vec{d}_5^*}, g_2^{\sigma \vec{d}_6^*} (\text{ from } SK_{\vec{\mathcal{I}}}),\ K_1', \ldots, K_j', K_{j+1} \in \mathbb{G}_2 \right\} . \tag{3.29}$$

Decryption(CT, $SK_{\vec{\mathcal{I}}}$) $\rightarrow$ M. Assuming $(\mathcal{I}_1, \ldots, \mathcal{I}_j)$ is a prefix of $(\mathcal{I}_1, \ldots, \mathcal{I}_\ell)$, the decryption algorithm computes $B := \prod_{i=1}^{j} e_m(C_0, K_i)$. The message is then computed as $M = C_0 / B$.

Table 3.13: Lewko HIBE scheme translation over prime order bilinear group.

Operation	Randomness complexity	Computation	Timing Tab. 3.10 $j = 3, m = 6$
Setup	r, $2m^2$ elts in $\mathbb{F}_r$ for $(\mathbb{D}, \mathbb{D}^*)$, 6 elts $\in \mathbb{F}_r$	1 pairing e, 2 exp. in $\mathbb{G}_T$, m^2 exp. in G_1, $m(m+2)$ exp. in $\mathbb{G}_2$	127 ms
KeyGen	$2j + 2(j-1)$ elts $\in \mathbb{F}_r$	$j \cdot m^2$ exp. in $\mathbb{G}_2$, some mult. in $\mathbb{F}_p$ and $\mathbb{G}_2$	206 ms
Encrypt	$2 + 2j$ elts in $\mathbb{F}_r$	$j \cdot m^2$ exp. in $\mathbb{G}_1$, 2 exp. in $\mathbb{G}_T$, some mult. in $\mathbb{F}_p$	70 ms
Delegate $j \rightarrow j+1$	$2(j+1) + 2j$ elts in $\mathbb{F}_r$	$(j+1)m^2$ exp. in $\mathbb{G}_2$	80 ms
Decryption	–	$j \cdot m$ pairings e	45.0 ms

Each step is summarized in Tab. 3.13. We choose a hierarchy depth of $j = 3$. This instantiation (Tab. 3.13) is 10 times more efficient than with a composite-order elliptic curve (Tab 3.12) for Setup, 56 times for KeyGen, 128 times for Encrypt, 192 times for Delegate and 112 times for Decryption. In other words, the important operations of delegation, encryption and decryption are more than a hundred times faster over a prime-order bilinear curve with an asymmetric pairing compared to a composite-order supersingular curve with a symmetric pairing.

3.3.5 Conclusion

We studied well-known protocols based on composite-order or prime-order elliptic curves. We justified the sizes of the composite orders when more than two primes are present in the modulus. We analyzed the Number Field Sieve complexity and the Elliptic Curve Method to find the size bounds. We then compared the cost of the homomorphic encryption scheme of Boneh, Goh and Nissim over a composite-order and the corresponding scheme over a prime-order pairing-friendly elliptic curve given by Freeman. In the former case, a pairing took 3 seconds, compared to 13 ms in the latter case. Even with 12 pairings instead of one in the Multiply step of the protocol, the prime-order translation remained 28 times faster. We also compared the unbounded HIBE protocol of Waters and Lewko and its translation given by Lewko. The prime-order setting is between 10 times to 192 times faster than the composite-order setting. Despite useful properties of bilinear composite-order structures to design new protocols, the resulting schemes are not very competitive compared to protocols relying on other assumptions which in

particular, need prime-order bilinear structures with asymmetric pairings. Some special protocols need extra properties such as canceling and projecting pairings. Only composite-order groups or supersingular curves achieve these properties.

We recommend to avoid composite-order groups whenever possible. Moreover, we did not investigate multi-exponentiation techniques to compute simultaneously several pairings on the same elliptic curve, neither did we use the Frobenius map to decompose exponents when performing exponentiation in $\mathbb{F}_{p^{12}}$. Hence some speed-ups are still available for protocols in the prime-order setting.

3.4 The BGW and PPSS broadcast protocols in practice

In this section, we first recall the general principles of a broadcast encryption scheme and the common notations in Sec. 3.4.1. Then we present in Sec. 3.4.2 the BGW protocol and its PPSS improvement. In Sec. 3.4.3 we expose our implementation.

This section is about a practical implementation of two pairing-based broadcast encryption protocols. The first one [BGW05] was published in 2005 at the *Crypto* conference by Boneh, Gentry and Waters. This pairing-based protocol achieves very efficient overhead size. The second one is a security improvement by Phan, Pointcheval, Strefler and Shahandashti. This improvement was designed for the needs of a project on broadcast encryption launched in 2009. This project [ENSC+09] is funded by the french Agence Nationale de la Recherche and lead by École Normale Supérieure, Université Paris 8, Thales, Nagra and Cryptoexperts. The aim is to identify new interesting protocols for the future generations of pay-TV systems on one side, and positioning systems and military telecommunications on the other side.

3.4.1 Preliminaries

We state some basic facts about broadcast encryption. A *broadcast encryption system* is deployed so send securely and efficiently digital content from a service center to a large set of users, over an insecure channel. This is widely used for e.g. Pay-TV systems, wireless networks, military radio communications and positioning systems (GPS, Galileo).

We enumerate the common words used in broadcast encryption.

Set of users: set of people who subscribed to a pay-TV service, or set of radios deployed on the battlefield, etc. Depending on the context, this is the set of all the persons/devices able to (physically) receive the encrypted data. We denote by $\mathcal{U}$ this set and by n the number of users in the system. In any system, the maximal number of users is usually bounded by at most $2^{32} \approx 4.2 \cdot 10^9$ since there are around six billion of people living on the earth.

Session: a time period when the secret key used to encrypt the data is valid.

Session key: the secret key (generated at random) used to encrypt the data broadcasted during the corresponding session.

Authorized user, privileged user, member: a user e.g. who has paid his subscription, who is allowed to decipher the encrypted data. The set of authorized users is denoted by $\mathcal{S}$. The set is fixed for one session and can change at the next session.

Revoked user: a user who is not allowed to decrypt sensitive data at some point, because he has not paid for it, or he has shared his secret keys with unauthorized users. In military context, the device is compromised (stolen by the enemy). We denote by r the number of revoked users in the system. The set of authorized users is denoted by $\mathcal{R}$.

Broadcaster: the center delivering encrypted data.

Receiver: any user device, revoked or not.

Overhead: the header added to the encrypted data. We denote it by Hdr. It contains informations to decrypt the data. In particular it contains a description of the authorized (or revoked) users hence its length is at least $O\log(n)$.

(t,n)-collusion secure: a broadcast protocol is secure under (t,n)-collusion if for all subset $\mathcal{R} \subset \mathcal{U}$ with $r = \#\mathcal{R} \leqslant t$, the revoked users from $\mathcal{R}$ are not able to decipher the data. Fully collusion-secure protocols are mostly appreciated.

Hybrid encryption is commonly used. This is a very basic trick in cryptography. We describe it in Fig. 3.8.

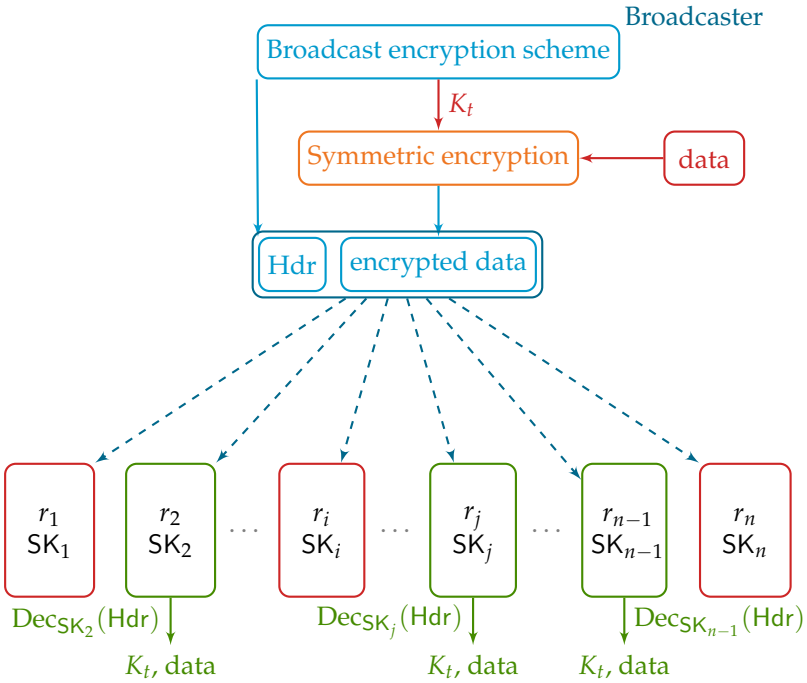


Figure 3.8: Broadcast scheme with hybrid encryption

When the set of users is quite small (e.g. less than a thousand of receivers), a naive method may be the best solution to broadcast encrypted content. We describe it in Fig. 3.9. Each receiver has a personal private key (stored in secured memory such as a smart card). At each session, the broadcaster generates at random a private session key K_t and encrypts the data with it. He encrypts the session key $\mathbb{K}_t$ with the private key $\mathsf{SK}_{j\in\mathcal{S}}$ of each authorized user. He adds in the header Hdr this list and a description of the set of authorized users (or a list of index). This is sketched in Fig. 3.9. In this setting, the bandwidth consumption is linear in the number of authorized users.

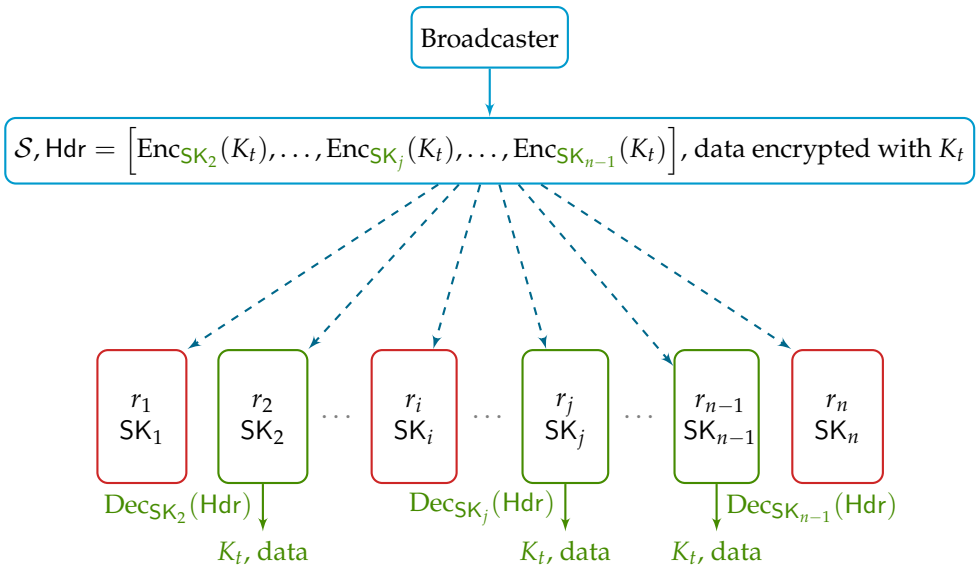


Figure 3.9: Naive broadcast encryption scheme for few users

The system constraints are
– the **bandwidth consumption**, related to the overhead size ω,
– the sender computation time τ_s, public key (resp. secret key) memory PK_s (resp. SK_s),
– the **users** (receivers) **computation time** τ_u, public key (resp. secret key) memory PK_u (resp. SK_u).

There was a lot of *tree-based* improvements, where the users are sorted in different groups and a certain set of secret keys is attributed to each group. More formally, lots of them are combinatorial tree-based schemes using the subset cover framework [NNL01]. However the overhead size is the minimal number of primary blocks used to cover the set. In other words, for the worst case of $r = \frac{n}{2}$, i.e. half the users are revoked ones, the others are members, the overhead size is the same as in the naive solution.

Boneh, Gentry and Waters introduced in [BGW05] two versions (denoted BGW_1 and BGW_2 in the following) of a pairing based protocol. This solve the problem of the bandwidth consumption when half the users are revoked and randomly distributed in the tree of users. The overhead size is in $O(1)$ (plus the description of $\mathcal{S}$) for BGW_1 and in $O(\sqrt{n})$ in BGW_2, for n users in the system. This comes at a time complexity expanse, as given in the table 3.14. Indeed, this protocol uses asymmetric cryptography. Delerablée, Paillier and Pointcheval described another scheme in [DPP07], reducing the time complexity. However the implementation is more complex, as it requires to handle formal sums of points.

Reference	ω	τ_r	PK_r	SK_r
Complete Subtree [NNL01]	$O(r\log(\frac{n}{r}))$	$O(\log\log n)$	–	$O(\log(n))$
Subset difference [NNL01]	$O(r)$	$O(\log(n))$	–	$O(\log^2(n))$
BGW_1 [BGW05]	$O(1)$	$O(n-r)$	$O(n)$	$O(1)$
BGW_2 [BGW05]	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(1)$
DPP_1 [DPP07]	$O(1)$	$O(r^2)$	$O(n)$	$O(1)$
DPP_2 [DPP07]	$O(r)$	$O(r)$	$O(1)$	$O(1)$
Sec. 3.4.2.1	$O(1)$	$\min(O(r), O(n-r))$	$O(n)$	$O(1)$
Sec. 3.4.2.2	$O(\sqrt{n})$	$\min(O(\frac{r}{\sqrt{n}}), O(\frac{n-r}{\sqrt{n}}))$	$O(\sqrt{n})$	$O(1)$

Table 3.14: Complexities of well known broadcast encryption schemes

To our knowledge, there is very few commercial products using pairings (some for IBE, see [Vol]), and none for broadcast. Despite there are several software and hardware pairing implementations with precise benchmarks, to our knowledge, there is not yet an entire broadcast protocol based on pairings implemented and presented with precise timings.

Our contributions. A practical instantiation was not explained in [BGW05]. A straightforward implementation of the protocol uses a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. This results in quite large size for the bandwidth elements. Each element (in $\mathbb{G}$) is of size half an RSA modulus size. For a 128-bit security level, this means 1536 bits per element instead of 128 in a combinatorial tree based protocol. We propose to design BGW with an appropriate asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. In this way, the elements in $\mathbb{G}_1$ have a size close to the optimal case in public key cryptography, i.e. 256 bits for the example above, rather than half a RSA modulus size. We adapt the protocol and set in the right groups $\mathbb{G}_1$ or $\mathbb{G}_2$ the different elements (public and private keys, bandwidth elements), knowing that the elements in $\mathbb{G}_1$ have the smallest size, those of $\mathbb{G}_2$ have quite medium size (at most half an RSA modulus) and those of $\mathbb{G}_T$ are close to an RSA modulus size. The resulting bandwidth consumption is divided by 6 at a 128-bit security level. We adapt accordingly the security proof.

The protocol security relies on the difficulty of a non-standard problem, the ℓ-BDHE (ℓ-Bilinear Diffie-Hellman Exponent problem). About one year after the publication in 2005 of BGW, Cheon proposed attacks in [Che06, Che10] against the family of Diffie-Hellman related problems used in the public key based protocols, including the ℓ-BDHE. More recently at the *PKC'2012* conference, an implementation of such an attack was presented at a security level of 80 bits [SHI+12]. We analyze the impact of Cheon's attacks on the size of the three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$. We propose a resistant elliptic curve.

The BGW scheme relies on public key tools. Hence the computation time is quite slower than in a symmetric key based protocol, especially for decryption. We provide an efficient trade-off between

memory and precomputation. Finally our practical implementation on a smartphone shows that with all our improvements, this BGW broadcast encryption scheme can be efficiently used for commercial applications.

The remaining of this section is organized as follows: in Sec. 3.4.2 we describe how BGW can benefit from the use of an asymmetric pairing and adapt the security proof. In Sec. 3.4.3, we detail our choice of a pairing-friendly elliptic curve and consider modifications due to Cheon's attacks. In Sec. 3.4.4, we describe how to use well chosen precomputation to dramatically reduce the computation cost. Finally, in Sec. 3.4.5 we give our results of a complete implementation of the protocol on a smartphone.

3.4.2 BGW with an asymmetric pairing

Boneh, Gentry and Waters [BGW05] describe a scheme with a minimal overhead. The scheme uses a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We presented the properties of pairings commonly used in cryptography in Sec. 1.4 and their state-of-the-art implementation in Sec. 3.2. We will start by presenting the BGW protocol adapted to an asymmetric pairing. Then we will propose a re-writing of the security justification. We will also investigate Cheon's attack on the underlying ℓ-BDHE problem. We will use the additive notation for both $\mathbb{G}_1$ and $\mathbb{G}_2$ and the multiplicative notation for $\mathbb{G}_T$.

In the original paper, the scheme is described with a symmetric pairing: $\mathbb{G}_1 = \mathbb{G}_2$ that is, we can swap the inputs $e(P, Q) = e(Q, P)$. In practice, the third group $\mathbb{G}_T$ is a finite field extension of the form $(\mathbb{F}_{p^k})^*$, of size $k \log p$ an RSA modulus. To use a symmetric pairing, supersingular or embedding-degree 1 curves shall be used (as shown in Sec. 1.4.3.1), which is inefficient. $\mathbb{G}_1$ and $\mathbb{G}_2$ have the same size, an explicit isomorphism exists between these two groups and their size is half the size of $\mathbb{G}_T$ (in large characteristic). For a justification, see Sec. 3.4.3. We propose to adapt the scheme to an asymmetric pairing in order to have a group $\mathbb{G}_1$ with smaller coefficients. We reorganize the elements and set in $\mathbb{G}_1$ those on which the bandwidth depends. Let n be the total number of users and r the number of revoked users. To remove confusion with the finite field characteristic (used later) commonly denoted p, we will denote by m the groups order.

3.4.2.1 First version of the scheme

We start by re-writing from [BGW05] the special case where the ciphertexts and private keys are of constant size. The n users are considered globally. The number of revoked users is r hence $n - r$ users must be able to decipher. Figure 3.10 presents this first version of BGW.

Setup(n)**.** Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order m with an asymmetric pairing e from $\mathbb{G}_1 \times \mathbb{G}_2$ onto $\mathbb{G}_T$. Let P be a random generator for $\mathbb{G}_1$ and Q for $\mathbb{G}_2$. Let α be a random element in $\mathbb{Z}_m$. The set-up step computes $P_i = \alpha^i P \in \mathbb{G}_1$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$. Note that P_{n+1} is missing. It also computes $Q_i = \alpha^i Q \in \mathbb{G}_2$ for $i = 1, 2, \ldots, n$. Then it picks at random $\gamma \leftarrow \mathbb{Z}_m$ and set $V = \gamma P \in \mathbb{G}_1$. The broadcaster public key is

$$\mathsf{PK}_s = (P, P_1, \ldots, P_n, P_{n+2}, \ldots, P_{2n}, V, Q, Q_1) \in \mathbb{G}_1^{2n+1} \times \mathbb{G}_2^2 \,. \tag{3.30}$$

Each user i receives an additional public key Q_i. The additional public key $(Q_1, \ldots, Q_n) \in \mathbb{G}_2^n$ is dispatched among all users. The complete public key PK is in $\mathbb{G}_1^{2n+1} \times \mathbb{G}_2^{n+1}$. The secret key for user i is $\mathsf{SK}_{u,i} = \gamma P_i \in \mathbb{G}_1$; its public key is $\mathsf{PK}_{u,i} = \big(Q_i,\ (P_i)_{1 \leqslant i \leqslant 2n,\, i \neq n+1}\big)$. Let $\mathcal{S} = \mathcal{U} \setminus \mathcal{R}$ be the subset of authorized users and $\#\mathcal{S} = n - r$.

Encrypt$(\mathcal{S}, \mathsf{PK}_s)$**.** The encryption step picks at random $k_t \leftarrow \mathbb{Z}_m$ and compute the session key $K_t = e(P_{n+1}, Q)^{k_t} = e(P_n, Q_1)^{k_t} \in \mathbb{G}_T$. It sets

$$\mathsf{Hdr} = \left(k_t Q,\ k_t\Big(V + \sum_{j \in \mathcal{S}} P_{n+1-j}\Big)\right) \in \mathbb{G}_2 \times \mathbb{G}_1 \tag{3.31}$$

and outputs (Hdr, K_t).

Decrypt$(i, \mathcal{S}, \mathsf{Hdr}, \mathsf{SK}_{u,i}, \mathsf{PK}_{u,i})$. Let $\mathsf{Hdr} = (C_0, C_1)$. The i-th user computes

$$K_t = \frac{e(C_1,\ Q_i)}{e\Big(\mathsf{SK}_{u,i} + \sum_{\substack{j\in\mathcal{S}\\ j\neq i}} P_{n+1-j+i},\ C_0\Big)}$$

The blue elements are broadcasted, the bandwidth depends on them. The user secret key is in red. The other elements on black are parameters and public keys. The verification uses the relation $e([i]P, [j]Q) = e(P,Q)^{ij} = e([j]P, [i]Q)$. We have chosen to set C_1 in $\mathbb{G}_1$ to save bandwidth, as the elements in $\mathbb{G}_1$ have coefficients a least twice as small as those in $\mathbb{G}_2$. It would be great to set C_0 in $\mathbb{G}_1$ as for C_1. Unfortunately in this case the user would have to compute the sum over all authorized users in $\mathbb{G}_2$ which is more time consuming than in $\mathbb{G}_1$. The storage size needed for a user i would be increased too. Our chosen trade-off will appear more natural through the generalized version of the scheme.

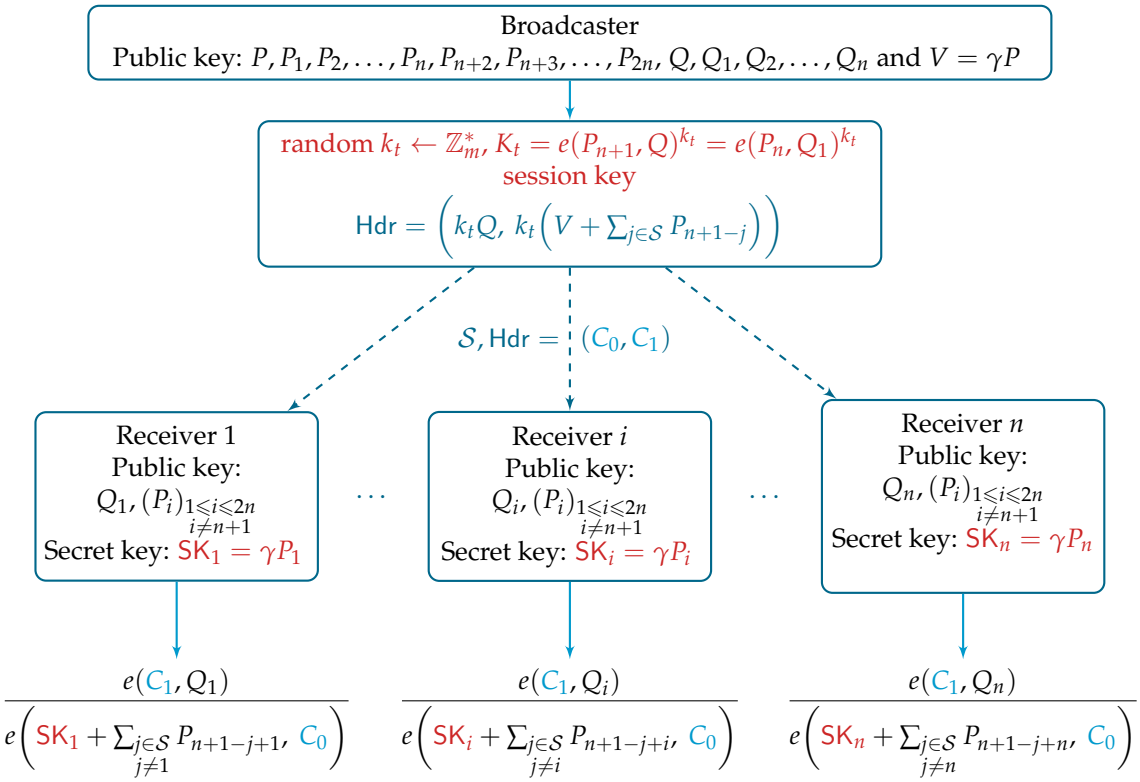


Figure 3.10: BGW protocol, first version, for a medium number of users.

3.4.2.2 General scheme

To reduce the public key size, the n users are organized into A groups of B users with $AB \geqslant n$. In [BGW05] the authors suggest to choose $B = \lfloor\sqrt{n}\rfloor$ and $A = \lceil\frac{n}{B}\rceil$. We can also divide users into groups according to their country, subscription or other criterion due to the system (Pay-TV, OTAR). A user i is referenced by its group number (say a) and its range in that group (say b). Hence $i = \{a, b\}$ with $1 \leqslant a \leqslant A$ and $1 \leqslant b \leqslant B$. The header Hdr will contain A public elements (instead of a unique C_1), each one dedicated to a determined group of users. Here we see relevant to set all these elements in $\mathbb{G}_1$. There is still the C_0 element that we need to set in $\mathbb{G}_2$ in order to keep in $\mathbb{G}_1$ the user public and private keys and a part of the decryption. The scheme is sketched in Fig. 3.11.

Setup$_B(n)$. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P, Q$ be as in the previous section (3.4.2.1). Let α be a random element in $\mathbb{Z}_m$. This step computes $P_i = \alpha^i P \in \mathbb{G}_1$ for $i = 1, 2, \ldots, B, B+2, \ldots, 2B$. These elements belong to the common public key. For each group of users, the user number $i = \{a, b\}$ receives the set of (P_i) and an additional public key $Q_b = \alpha^b Q \in \mathbb{G}_2$. The setup phase then picks uniformly at random the

elements $\gamma_1, \gamma_2, \ldots, \gamma_A \leftarrow \mathbb{Z}_m$ and sets $V_1 = \gamma_1 P, \ldots, V_A = \gamma_A P \in \mathbb{G}_1$. The centralized public key is $\mathsf{PK}_s = (P, P_1, P_2, \ldots, P_B, P_{B+2}, \ldots, P_{2B}, V_1, \ldots, V_A, Q, Q_1) \in \mathbb{G}_1^{2B+A} \times \mathbb{G}_2^2$. The secret key for the user number b in the group a is $\mathsf{SK}_{u,\{a,b\}} = \gamma_a P_b \in \mathbb{G}_1$. Its public key is $\mathsf{PK}_{u,\{a,b\}} = \big(Q_b, (P_i)_{1 \leqslant i \leqslant 2B,\, i \neq B+1}\big)$. The user does not need the others Q_ℓ hence to save memory on his constrained device (e.g. smartphone, set-up box) we don't add them. Note that this scheme is relevant even for unbalanced group sizes. For larger groups, the computation time will increase, but the bandwidth consumption will be the same: one group element (in $\mathbb{G}_1$) per group of users, whatever the size of the group is.

Encrypt$(\mathcal{S}, \mathsf{PK}_s)$. For each group a of users, we denote by $\mathcal{S}_a$ the set of authorized users in this group. The encryption step picks a random k_t in $\mathbb{Z}_m$ and computes the session key as $K_t = e(P_{B+1}, Q)^{k_t} = e(P_B, Q_1)^{k_t} \in \mathbb{G}_T$. The overhead is

$$\mathsf{Hdr} = \Big(k_t Q,\ k_t\big(V_1 + \sum_{j \in \mathcal{S}_1} P_{B+1-j}\big),\ k_t\big(V_2 + \sum_{j \in \mathcal{S}_2} P_{B+1-j}\big), \ldots,\ k_t\big(V_A + \sum_{j \in \mathcal{S}_A} P_{B+1-j}\big)\Big) \in \mathbb{G}_2 \times \mathbb{G}_1^A\,. \quad (3.32)$$

Decrypt$\Big(i = \{a,b\}, \mathcal{S}_a, \mathsf{Hdr}, \mathsf{SK}_{u,\{a,b\}}, \mathsf{PK}_{u,\{a,b\}}\Big)$. Let denote $\mathsf{Hdr} = (C_0, C_1, \ldots, C_A)$. A user i is indexed by a number b in a group a. The user $i = \{a,b\}$ computes the session key as

$$K_t = \frac{e(C_a, Q_b)}{e\big(\mathsf{SK}_{u,\{a,b\}} + \sum_{\substack{j \in \mathcal{S}_a \\ j \neq b}} P_{B+1-j+b},\ C_0\big)}\,.$$

The verification uses the same bilinearity property as previously:

$$e([i]P, [j]Q) = e(P, Q)^{ij} = e([j]P, [i]Q)\,.$$

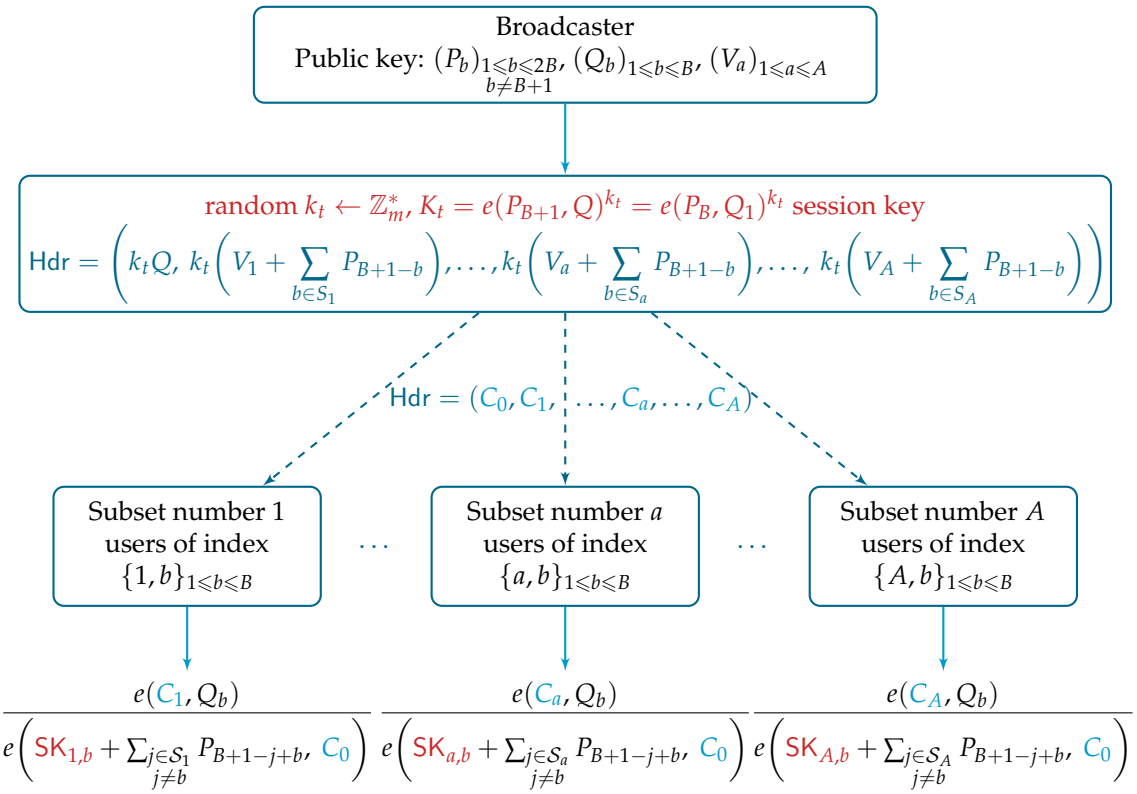


Figure 3.11: BGW protocol, second version, for a large number of users.

Table 3.15 gives the protocol complexity with an asymmetric pairing. BGW_1 denotes the one instance version described in the previous section (Sec. 3.4.2.1), BGW_2 denotes the parallel instance version explained in this section. ω is the bandwidth consumption, PK_s denotes the sender's memory for the public key, τ_s the time computation and respectively PK_u, τ_u denote the receiver's ones. r_a is the number of revoked users in the group a. Note that they are at most B users in a group a.

Scheme	ω	PK_s	τ_s	PK_r	τ_r
BGW_1	$\mathbb{G}_2 \times \mathbb{G}_1$	$\mathbb{G}_1^{2n+1} \times \mathbb{G}_2^{n+1}$	$(n-r)\mathrm{Add}_{\mathbb{G}_1}$	$\mathbb{G}_1^{2n-1} \times \mathbb{G}_2$	$(n-r)\mathrm{Add}_{\mathbb{G}_1}$
BGW_2	$\mathbb{G}_2 \times \mathbb{G}_1^A$	$\mathbb{G}_1^{2B+A} \times \mathbb{G}_2^{B+1}$	$(n-r)\mathrm{Add}_{\mathbb{G}_1}$	$\mathbb{G}_1^{2B-1} \times \mathbb{G}_2$	$(B-r_a)\mathrm{Add}_{\mathbb{G}_1}$

Table 3.15: Theoretical complexity for BGW protocol, asymmetric pairing

3.4.2.3 Security proof

In [BGW05, §3.3], the authors prove the semantic security of the general system. We faced some trouble when adapting the security proof to an asymmetric pairing in the setting above. We need to add a copy in $\mathbb{G}_2$ of the inputs elements in $\mathbb{G}_1$ to the problem. This difficulty rises in the challenge phase. To generate a consistent input for the adversary, the challenger must have a copy in $\mathbb{G}_2$ of the inputs in $\mathbb{G}_1$. This is transparent with a symmetric pairing (in which case an isomorphism from $\mathbb{G}_1$ into $\mathbb{G}_2$ is available). This is also quite easy if an isomorphism from $\mathbb{G}_2$ into $\mathbb{G}_1$ is available.

More precisely, let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ three cyclic groups of prime order together with an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Let P a generator for $\mathbb{G}_1$ and Q for $\mathbb{G}_2$. Let $Q^{'}$ a random element in $\mathbb{G}_2$. In the challenge phase, the challenger must compute a corresponding $P^{'} \in \mathbb{G}_1$ such that $\log_P(P^{'}) = \log_Q(Q^{'})$ without knowing $\log_Q(Q^{'})$. In other words, in this construction there is some $k_t \in \mathbb{Z}_m$ such that $Q^{'} = [k_t]Q$ and we have to find a corresponding $P^{'} \in \mathbb{G}_1$ such that $P^{'} = [k_t]P$ with the same $k_t \in \mathbb{Z}_m$, without knowing k_t. Therefore we need an explicit isomorphism ϕ which maps the generator $Q \in \mathbb{G}_2$ to $P \in \mathbb{G}_1$. With this map we can compute $\phi(Q^{'}) = P^{'}$. In this way we can end the security proof as in the original paper. Such a map usually does not exists for ordinary pairing-friendly elliptic curves. For supersingular (and embedding degree one) curves, there is a distortion map from $\mathbb{G}_1$ to $\mathbb{G}_2$ which provides an explicit isomorphism, thus a symmetric pairing. For ordinary elliptic curves, the trace map [BSS05, IX.7.4] is degenerated, as $\mathbb{G}_2$ is commonly built as the trace-zero subgroup. With the notations from [GPS08], the security proof must be written assuming that the pairing is of Type 3 : $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Hence the adversary needs to receive $P^{'}$, that is why it must appears in the challenger inputs.

Let start with an asymmetric variant of ℓ-BDHE problem:

Definition 21 (ℓ-BDHEasym)**.** *Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be three cyclic groups of prime order together with an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Given $(P, P_1, \ldots, P_\ell, P_{\ell+2}, \ldots, P_{2\ell}) \in \mathbb{G}_1^{2\ell}$, $(Q, Q_1, \ldots, Q_\ell) \in \mathbb{G}_2^{\ell+1}$ such that $P_i = [\alpha^i]P$, $Q_i = [\alpha^i]Q$, and $(P^{'}, Q^{'}) \in \mathbb{G}_1 \times \mathbb{G}_2$ such that $\log_P P^{'} = \log_Q Q^{'}$, compute*

$$e(P_{\ell+1}, Q^{'}) \text{ which is the same as computing } e(P^{'}, Q_{\ell+1}) .$$

Definition 22 (Decisional $\ell -$ BDHEasym)**.** *Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be three cyclic groups of prime order together with an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Let $\mathbf{Y}_{P,Q,\alpha,\ell} = (P_1, P_2, \ldots, P_\ell, P_{\ell+2}, \ldots, P_{2\ell}, Q_1, Q_2, \ldots, Q_\ell)$. An algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ε in solving the decisional $\ell -$ BDHEasym in $\mathbb{G}_T$ if $\left|\Pr[\mathcal{B}(P, Q, P^{'}, Q^{'}, \mathbf{Y}_{P,Q,\alpha,\ell}, e(P_{\ell+1}, Q^{'})) = 0] - \Pr[\mathcal{B}(P, Q, P^{'}, Q^{'}, \mathbf{Y}_{P,Q,\alpha,\ell}, T) = 0]\right| \geqslant \varepsilon$ where the probability is over the random choice of generators $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$, of random point $P^{'} \in \mathbb{G}_1$, the random choice of $\alpha \in \mathbb{Z}_m$, the random choice of $T \in \mathbb{G}_T$ and the random bits consumed by $\mathcal{B}$. The distribution on the left is denoted by $\mathcal{P}_{\mathrm{BDHEasym}}$ and the distribution on the right by $\mathcal{R}_{\mathrm{BDHEasym}}$.*

The decision $(\tau, \varepsilon, \ell) -$ BDHEasym assumption holds in $\mathbb{G}_T$ if no τ-time algorithm has advantage at least ε in solving the decision ℓ-BDHE problem in $\mathbb{G}_T$.

According to the definitions in [PPS11], BGW and the variants presented here are asymmetric broadcast encryption with a static set of users (the joint is made at setup only) and stateless users (the public and private keys do not evolve from a session to another). A selective security for key indistinguishability is proven (the target set is chosen before the setup phase).

Suppose there exists a τ-time adversary, $\mathcal{A}$, who receives an instance of the protocol. The adversary is able to distinguish between a valid and a random session key with advantage $\mathrm{AdvBr}_{\mathcal{A},B} > \varepsilon$ for a system parameterized with a given B. One build an algorithm, $\mathcal{B}$, that has advantage ε in solving the decision $B-$BDHEasym problem in $\mathbb{G}_T$.

Algorithm $\mathcal{B}$ takes as input a random decision $B-$BDHEasym challenge $(P, Q, P', Q', \mathbf{Y}_{P,Q,\alpha,B}, Z)$ where $\mathbf{Y}_{P,Q,\alpha,B} = (P_1, P_2, \ldots, P_B, P_{B+2}, \ldots, P_{2B}, Q_1, Q_2, \ldots, Q_B)$ and Z is either $e(P_{B+1}, Q')$ or a random element in $\mathbb{G}_T$. The aim of $\mathcal{B}$ is to decide if Z is valid or random. For doing that, $\mathcal{B}$ simulates a session of the broadcast protocol and submits it to $\mathcal{A}$. Then $\mathcal{B}$ uses $\mathcal{A}$'s answer to decide if Z is valid or random. Algorithm $\mathcal{B}$ proceeds as follows.

Init. Algorithm $\mathcal{B}$ runs $\mathcal{A}$ and receives the set $\mathcal{S} = \cup_{1 \leqslant a \leqslant A} \mathcal{S}_a$ of users that $\mathcal{A}$ wishes to be challenged on.

Setup. $\mathcal{B}$ needs to generate a public key PK and private keys $\mathsf{SK}_{u,i}$ for users $i \notin \mathcal{S}$. We can use the same idea as in the original proof. Algorithm $\mathcal{B}$ chooses uniformly at random $u_a \in \mathbb{Z}_m$ for $1 \leqslant a \leqslant A$. The users are divided into A groups of at most B users. A user i is number b in a precise group a. For $a = 1, \ldots, A$, algorithm $\mathcal{B}$ sets $V_a = [u_a]P - \sum_{j \in \mathcal{S}_a} P_{B+1-j}$. It gives $\mathcal{A}$ the public key

$$\mathsf{PK} = (P_1, \ldots, P_B, P_{B+2}, \ldots, P_{2B},\ Q_1, \ldots, Q_B,\ V_1, \ldots, V_A)$$

which is in $\mathbb{G}_1^{2B-1} \times \mathbb{G}_2^B \times \mathbb{G}_1^A$.

Boneh, Gentry and Waters note in their paper [BGW05] that since P, α and the u_a values are chosen uniformly at random, the public key

$$\mathsf{PK}_{\text{original}} = (P, P_1, \ldots, P_B, P_{B+2}, \ldots, P_{2B},\ V_1, \ldots, V_A) \in \mathbb{G}_1^{2B+A}$$

has an identical distribution to that in the actual construction. Here it is necessary to give $(Q_1, \ldots, Q_B) \in \mathbb{G}_2^B$ too. If we assume that P is a generator chosen at random in $\mathbb{G}_1$ and Q (which generates $\mathbb{G}_2$) is also chosen at random and independent from P, we can consider that all these elements are uniformly distributed at random.

Next the adversary needs all private keys that are not in the target set $\mathcal{S}$. For each user $i = \{a, b\} \notin \mathcal{S}$, algorithm $\mathcal{B}$ computes the corresponding private key

$$\mathsf{SK}_{u,\{a,b\}} = [u_a]P_b - \sum_{j \in \mathcal{S}_a} P_{B+1-j+b}\,.$$

The same equality holds as in the original proof

$$\mathsf{SK}_{u,\{a,b\}} = [u_a][\alpha^b]P - [\alpha^b] \sum_{j \in \mathcal{S}_a} P_{B+1-j} = [\alpha^b]V_a\,.$$

The authors in [BGW05] note that the unknown value P_{B+1} is not involved in the sum, as i is a revoked user ($i = \{a, b\}$ with $b \notin \mathcal{S}_a$).

Challenge. To generate the challenge, $\mathcal{B}$ computes Hdr as

$$(Q', [u_1]P', \ldots, [u_A]P')\,.$$

$\mathcal{B}$ then randomly chooses a bit $b \in \{0, 1\}$ and sets $K_b = Z$ and picks a random K_{1-b} in $\mathbb{G}_T$. It gives (Hdr, K_0, K_1) as the challenge to $\mathcal{A}$.

We use the same justification as in the above cited paper. The algorithm knows both Q' and P' such that $\log_P(P') = \log_Q(Q')$ hence can compute a valid Hdr. When the input to $\mathcal{B}$ is a B-BDHEasym tuple, $Z = e(P_{B+1}, Q')$ and (Hdr, K_0, K_1) is a valid challenge to $\mathcal{A}$ as in a real attack. Let k_t such that $P' = [k_t]P$. P' and Q' are bound together in the sense that $P' = [k_t]P$ and $Q' = [k_t]Q$ with the same $k_t \in \mathbb{Z}_m$. $[u_a]P' = [k_t][u_a]P = [k_t]\big([u_a]P - \sum_{j \in \mathcal{S}_a} P_{B+1-j} + \sum_{j \in \mathcal{S}_a} P_{B+1-j}\big) = [k_t]\big(V_a + \sum_{j \in \mathcal{S}_a} P_{B+1-j}\big)$. We can see in this form that $(Q', [u_1]P', \ldots, [u_A]P')$ is a valid encryption of the key $e(P_{B+1}, Q)^{k_t}$. Then $e(P_{B+1}, Q)^{k_t} = e(P_{B+1}, Q') = Z = K_b$. Hence (Hdr, K_0, K_1) is a valid challenge to $\mathcal{A}$. On the other hand, when the input to $\mathcal{B}$ is a random tuple, Z is a random element from $\mathbb{G}_T$, and K_0, K_1 are random elements from $\mathbb{G}_T$.

Guess This last step is the same as in the paper [BGW05]. The adversary $\mathcal{A}$ outputs a guess b' of b. If $b = b'$ the algorithm $\mathcal{B}$ outputs 0, i.e. it guesses that $Z = e(P_{B+1}, Q')$. Otherwise, it outputs 1, i.e. Z is a random element in $\mathbb{G}_T$. If $(P, Q, P', Q', \mathbf{Y}_{P,Q,\alpha,B}, Z)$ is sampled from $\mathcal{R}_{\text{BDHEasym}}$ then $\Pr\left[\mathcal{B}(P, Q, P', Q', \mathbf{Y}_{P,Q,\alpha,B}, Z) = 0\right] = 1/2$. If $(P, Q, P', Q', \mathbf{Y}_{P,Q,\alpha,B}, Z)$ is sampled from $\mathcal{P}_{\text{BDHEasym}}$ then

$$\left|\Pr\left[\mathcal{B}(P, Q, P', Q', \mathbf{Y}_{P,Q,\alpha,B}, Z) = 0\right] - 1/2\right| = \text{AvdBr}_{\mathcal{A},B} \geqslant \varepsilon\,.$$

It follows that $\mathcal{B}$ has advantage at least ε in solving the B-BDHEasym problem in $\mathbb{G}_T$. This conclude the security proof.

3.4.2.4 Attacks on Diffie-Hellman problem with auxiliary inputs

The security relies on the ℓ-Bilinear Diffie-Hellman Exponent assumption (defined in Def. 21) which is a weaker problem than the Diffie-Hellman one. The difficulty of this problem was first studied in [Che06]. See also improvements in [KKM07, Che10] and the implementation in [SHI+12]. We state here the results on the complexities of these attacks and explain the possibilities to avoid as much as possible these attacks when choosing a pairing-friendly elliptic curve.

Theorem 15 ([KKM07, Theorem 1'])**.** *Let P be an element of prime order m in an abelian group $\mathbb{G}$. Suppose that d is a positive divisor of $m-1$. If $P, [\alpha]P, [\alpha^d]P$ are given, α can be computed within $O(\sqrt{m/d} + \sqrt{d})$ group operations using space for $O(\max(\sqrt{m/d}, \sqrt{d}))$ groups elements.*

Theorem 16 ([KKM07, Theorem 2'])**.** *Let P be an element of prime order m in an abelian group $\mathbb{G}$. Suppose that d is a positive divisor of $m+1$ and $[\alpha^i]P$ are given for $1 \leqslant i \leqslant 2d$. Then α can be computed within $O(\sqrt{m/d} + d)$ group operations using space for $O(\max(\sqrt{m/d}, \sqrt{d}))$ groups elements.*

The main idea for the first theorem is to find a divisor d of $m-1$ in the range $2 \leqslant d \leqslant B$ or $B+2 \leqslant d \leqslant 2B$ to reduce the complexity from $O(\sqrt{m})$ to $O(\sqrt{m/d} + \sqrt{d})$. A decomposition of the classical Baby Step Giant Step (BSGS) algorithm in two phases reduces the complexity of BSGS from $O(\sqrt{m})$ to two BSGS running, the first in $O(\sqrt{m/d})$ and the second in $O(\sqrt{d})$. We have to take into account this attack to choose properly a convenient elliptic curve when setting the system parameters.

1. We can enlarge the parameters in order to prevent the system from these attacks and match the previously chosen security level. Assuming that $B \ll m$, we consider that the attack is in at most $O(\sqrt{m/2B})$. For a 128-bit security level, instead of a prime order group $\mathbb{G}_1$ of size $\log m = 256$, we have to set $\log m = 256 + \log(2B)$. If the system is designed for 10^6 users and $B \approx 10^3$, enlarging $\log m$ with at least 12 bits is enough and quite cheap if it does not affect considerably the size of $\mathbb{G}_T$.
2. If enlarging m with a few bits will enlarge the size of $\mathbb{G}_T$ of a few hundred bits (because of the gap caused by the embedding degree), we may prefer to choose directly a safe prime order m, such that $m-1$ and $m+1$ are not divisible by factors smaller than $2B$. Of course either $m-1$ or $m+1$ will be a multiple of 4 but we loose only 2 bits.

3.4.3 Choice of the pairing-friendly elliptic curve

The two instantiations are the Weil pairing and the Tate pairing over elliptic curves (defined over finite fields). They can be quite efficiently computed with the algorithm due to Miller and the various improvements described in Sec. 1.4 and 3.2. Let p be a large prime and $E(\mathbb{F}_p)$ an elliptic curve defined by a reduced Weierstraß equation $y^2 = x^3 + ax + b$. Remember that $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of prime order m of the elliptic curve and $\mathbb{G}_T$ is a multiplicative subgroup of order m of an extension field $\mathbb{F}_{p^k}^*$. The main difficulty is to find suitable elliptic curves for pairings. An almost exhaustive study of known pairing-friendly elliptic curves can be found in [FST10]. If the protocol relies exclusively on the Diffie-Hellman problem, to achieve the same complexity in the three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ we must choose carefully the size of the groups as following. If we consider a non-pairing-friendly elliptic curve (i.e. of large embedding degree), ordinary, over a prime field in large characteristic, of trace $\not\equiv 0 \mod p$ and $\neq 1$ then up to now, only generic attacks such as Pohlig-Hellman exists for solving the Diffie-Hellman problem in

such a curve. If the protocol relies exclusively on the Diffie-Hellman problem (without pairings), for a N-bit security level, a prime order group $\mathbb{G}_1$ of size $\log m = 2N$ bits is convenient.

In a pairing-based context, the group $\mathbb{G}_T \subset \mathbb{F}_{p^k}^*$ is exposed to the less difficult index calculus attack. Hence the size $k \log p$ of $\mathbb{G}_T$ is greater than those of $\mathbb{G}_1$ and $\mathbb{G}_2$. An RSA modulus size is commonly considered to be safe. We emphasize that since 2012, crazy improvements were achieved to compute discrete logarithm problems in finite field extensions in small and medium characteristic. We can cite a few: a Japanese team [HSST12, SHI+12] broke records in $\mathbb{F}_{3^{6\cdot 97}}$, we can also cite the recent work in [AMORH13], an Irish team [GGMZ13a, GGMZ13b] improved the FFS algorithm, then French people announced fantastic records, from [Jou12, DGV13, Jou13b, BBD+13] to [BGJT13]. This comes from a powerful improvement of the Function Field Sieve method. These improvements do not apply to the Number Field Sieve method used to compute discrete logarithms in small extensions of large prime fields. In other words, the pairing-friendly curves in small characteristic, over $\mathbb{F}_{2^n}$ or $\mathbb{F}_{3^n}$, shall be avoided. Up to now the pairing-friendly curves in large characteristic such as the BN curves are not concerned with these attacks.

The following key-size (Tab. 3.16) are recommended by the ECRYPT II research group [oEiCI11, Tab. 7.2].

Security (bits)	RSA	Discrete Logarithm		Elliptic curve
		field	subfield	
80	1248	1248	160	160
112	2432	2432	224	224
128	**3248**	**3248**	**256**	**256**
160	5312	5312	320	320
192	7936	7936	384	384
256	15424	15424	512	512

Table 3.16: Ecrypt II key-size recommendations

We have chosen a 128-bit security level. A supersingular curve (over a prime field in large characteristic) has an embedding degree k at most 2 resulting in $\log p = 1624$, $\log m = 256 + \delta$ and $\rho = \log p / \log m \approx 6$. The notation $+\delta$ means that enlarging m by a few bits will not impact on $\log p$, hence on the size of $\mathbb{F}_{p^k}$. The well-known Barreto-Naehrig curves (BN, [BN05]) fit almost exactly the recommended sizes of $\mathbb{G}_1$ and $\mathbb{G}_T$, taken into account Cheon's attack. Indeed, for these curves, $k = 12$ and $\log m = \log p$. Hence with $k \log p = 3264$ and $\log m = \log p = 272$, the parameters are strong enough against the ℓ-BDHE problem for a 128-bit security level and a BGW protocol with at most $2B$ users per group and $\log(2B) = 16$.

If we prefer to follow NIST recommendations, the $k = 12$ embedding degree is exactly what we need : $\log m = 256$ and as $\rho = \log p / \log m = 1.0$, $k \log p = 3072$ as expected. In particular, for a 128 bit security level, using an asymmetric pairing decreases the size of the element in the group $\mathbb{G}_1$ by a factor of 6. To prevent from Cheon's attacks, we can increase the size of m by 12 bits but it results in increasing the size of $\mathbb{F}_{p^k}$ by 144 bits. To avoid this, we must generate a strong BN curve, without any integer d dividing m and less than 2^{12}. We heard about this attack after launching the prototype development. Hence the benchmarks were computed for this curve.

x = - `0x400000000000031C` (which defines p, m and t)
p = `0x24000000000006FE700000000082705C800000043937699E80000D20DA314BD9`
m = `0x24000000000006FE700000000082705C200000043937604A80000D20D9F74979`
t = `0x6000000000000954000000000003A0261`
b = `0x17`

The elliptic curve defined over the prime field $\mathbb{F}_p$ with parameter equation $a = 0$ and b above has prime order m and trace t. The three numbers x, $m - 1$ and $m + 1$ are smooth.

$$
\begin{aligned}
x &= 2^2 \cdot 5^2 \cdot 43 \cdot 139 \cdot 757 \cdot 10192497083,\\
m-1 &= 2^3 \cdot 3 \cdot 5^2 \cdot 23 \cdot 43 \cdot 71 \cdot 139 \cdot 757 \cdot 338172217 \cdot 10192497083\\
&\quad \cdot 1065629744969022147085838680434831409024186859,\\
m+1 &= 2 \cdot 7 \cdot 11 \cdot 31 \cdot 67 \cdot 179 \cdot 1297 \cdot 839731 \cdot 15999517 \cdot 282551569\\
&\quad \cdot 35836294153183 \cdot 251224184937629 \cdot 6415963443272843.
\end{aligned}
$$

Assuming that there are around 2^{10} users per group, we have $\log(2B) \leqslant 12$ and the security for this curve is 116 bits instead of 128 bits. Then we heard about Cheon's attack and tried to find a "strong" curve. Because of the parameter structure, the curve order m is such that 12 divides $m-1$ and 2 divides $m+1$. We ran a search over almost prime x to find an m such that no divisor less than 2^{12} divides either $m-1$ or $m+1$ (except 12 for $m-1$ and ones less than 16 for $m+1$). We found a few appropriate curves, for example

$$
\begin{aligned}
x &= \texttt{0x4000000000087F7F} = 248861 \cdot 18531172093771\\
p &= \texttt{0x2400000000131EDE500003CEEC974A28964D2C8BEE1F7C511355420E690A2713}\\
m &= \texttt{0x2400000000131EDE500003CEEC974A28364D2C8BEE05FDD41355405D1C6EA10D}\\
m-1 &= x \cdot 12 \cdot 757798571 \cdot 43164459611077952667523 7 \cdot 89953974744006091548728 9\\
m+1 &= 2 \cdot 480707 \cdot 420180967 \cdot 107234028019 \cdot 1416027609325038349\\
&\quad \cdot 26545460664267993656900293976638 1\\
t &= \texttt{0x6000000000197E7D000001B14C9B8607}\\
b &= \texttt{0xC}
\end{aligned}
$$

For this curve, $12 \mid m-1$ and the next divisor is 248861; $2 \mid m+1$ and the next divisor is 480707. Because of the 12, we loose 4 bits. Our implementation doesn't depends on a particular p or m hence changing their value will not infer on the timings if their size remains the same.

The possible choice are presented in Tab. 3.17. The notation $+\delta$ means that enlarging m by a few bits will not impact on $\log p$, hence on the size of $\mathbb{F}_{p^k}$.

Recommendations	Curve	k	$k \log p$	$\log p$	$\rho = \log p / \log m$	$\log m$
Ecrypt II	Supersingular	2	3248	1624	not fixed, ≈ 6 here	$256 + \delta$
	Barreto-Naehrig	12	3264	272	1.0	272
NIST	Supersingular	2	3072	1536	not fixed, ≈ 6 here	$256 + \delta$
	Strong BN	12	3072	256	1.0	256

Table 3.17: Parameters size depending on the embedding degree

This work and the benchmarks presented in Sec. 3.4.5 were done in 2011 (this was a joint work with Dubois and Sengelin Le Breton). At that time, the efficient ate and optimal ate pairings presented in Sec. 3.2 were not yet available in the cryptographic library used in the lab. Only a Tate pairing was implemented. We give in Tab. 3.18 the timings of the library in 2011.

Curve	k	$\log m$	$\log p$	Miller's Loop	Exponentiation	Pairing
Supersingular	2	256	1536	29.88 ms	25.99 ms	55.87 ms
Barreto-Naehrig	12	256	256	14.51 ms	5.18 ms	19.69 ms

Table 3.18: Our Implementation of pairing computation on a AMD64 3Ghz (Ubuntu 10.10), LibCryptoLCH, 2011

3.4.4 Reducing Time Complexity

In this section we present a way to reduce the complexity of the decryption step. We recall that the decryption computes the sum

$$\sum_{j \in \mathcal{S},\, j \neq i} P_{n+1-j+i}$$

with i the index of the user, $\mathcal{S}$ the set of authorized users and n the total number of users in the system. This sum is linear in $(n-r)$ with r the number of revoked users. This becomes very time consuming with

a system of a large number of users. We propose a method to precompute a table of values involved in this sum in order to speed-up the decryption step.

The public keys are points on an elliptic curve hence addition is as cheap as subtraction. If the number of revoked users is small ($r \ll n/2$), the initial computation in $O(n-r)$ is quite slow. We can instead consider that the value $\Sigma_i^n = \sum_{1 \leqslant j \neq i \leqslant n} P_{n+1-j+i}$ is precomputed for each user i. Then

$$S = \Sigma_i^n - \sum_{j \in \mathcal{R}} P_{n+1-j+i}$$

with $\mathcal{R}$ the set of revoked users. Now the complexity is $O(\min(r, n-r))$ (where O is the cost of a point addition, EllAdd). We can do better with a precomputed tree.

3.4.4.1 Binary public key tree precomputation

In this section we describe how to decrease the computation time from $O(\min(r, n-r))$ using only twice memory. The tweak consists in two modifications.

1. We expend the public key into a binary public key tree T twice long obtained by
   - sorting all users in a binary tree whose leaves are the users;
   - precomputing for each node in the tree from the leaves to the root the sum over each public key of the nodes below.
2. For each encryption and decryption step, choose the optimal including/excluding tree to compute the sum. For example, for each decryption, we use Alg. 19 if $r < n/2$ or its variant if $r \geqslant n/2$ to compute the value of the sum S.

Let consider a user i in a system of at most n users. This user needs the elements $P_{n+1-j+i}$, $1 \leqslant j \neq i \leqslant n$ of the public key $P_1, \ldots, P_n, P_{n+2}, \ldots, P_{2n}$ that is, $n-1$ elements in $\mathbb{G}_1$. The user needs also $Q_i \in \mathbb{G}_2$ (which does not need to appear in the tree). Each user computes a different (translated by i) tree. We assume that the nodes are labeled in the same way for each user. The difference from a user to another is only the initialization of the leaf values.

Example 19 (Precomputing the tree)**.** *Suppose that $n = 16$. The user $i = 9$ computes the tree represented in Fig. 3.12. The leaves are the $P_{n+1-j+9}$ with $1 \leqslant j \neq i \leqslant n$. We represent two lines : the users and the $n+1-j+9$ index. For each node in the tree, the user computes the sum of the two children. The value $P_{n+1} = P_{17}$ is missing, the user sets $\mathcal{O}$ instead on the corresponding leaf. The user 9 does not need the values $P_1, \ldots, P_9$ and $P_{26}, \ldots, P_{32}$. The value stored at node 31 is the sum of all public keys from P_{10} to P_{25}, except P_{17} (replaced by $\mathcal{O}$). The value stored at node 25 is the sum of the public keys P_{22} to P_{25}.*

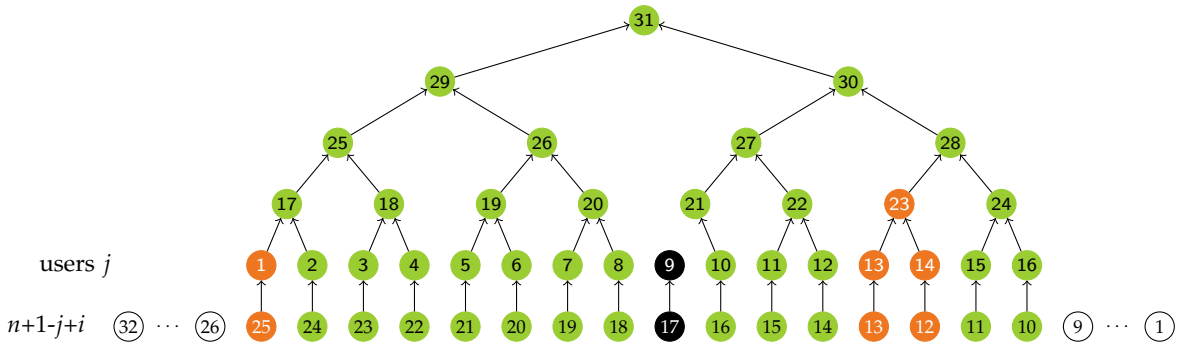


Figure 3.12: Public keys and precomputation with $n = 16$, for user $i = 9$

Example 20 (Computing S quickly with the tree)**.** *We consider the same set of $n = 16$ users, indexed from 1 to 16. We assume that at the current session, the users 1, 13 and 14 are revoked ($r = 3 < n/2 = 8$). They are in red on Fig. 3.12. For user 9, using algorithm 19, the sum S is computed by summing the following elements of T:*

$$S = T_{31} - T_1 - T_{23}$$

Algorithm 19: Improved computation of S when $r < n/2$

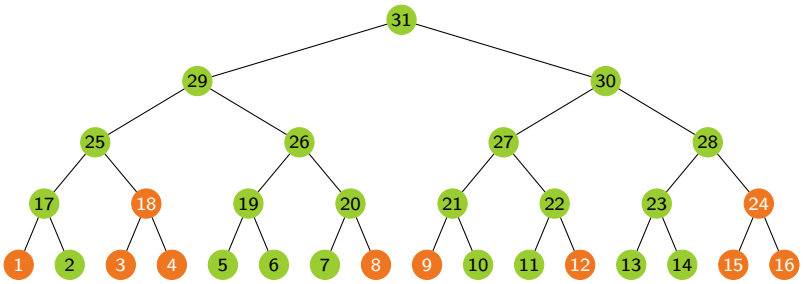
Input: The user ID i, the set of privileged users $\mathcal{S}$, the precomputed public tree T (for user i)
Output: The sum of points on the elliptic curve $S = \sum_{j \in \mathcal{S}, j \neq i} P_{n+1-j+i}$

```
Let T' be the binary tree whose nodes are those of T. for each node of T', from the leafs do the root do
    if it is the leaf of an authorized user or if there exists a green node below then
        color the node in green
    else
        color the node in red
S ← T_root
for each red node with a green parent do
    subtract the related public value from S
return S
```

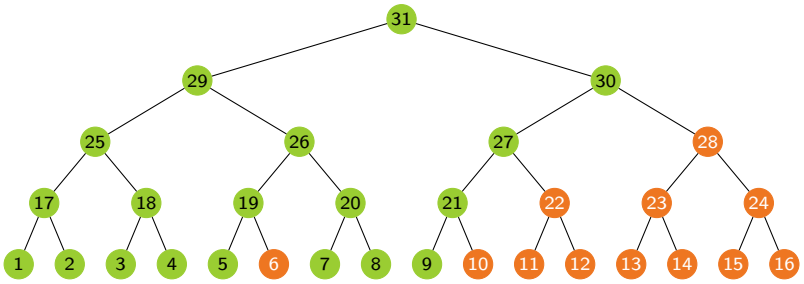
Note that a subtraction is as cheap as an addition on an elliptic curve. The resulting cost is only 2 EllAdd, while it would have been 13 *on the original scheme.*

When $r > n/2$, we can apply the same method but instead of covering the revoked users and subtracting the corresponding public keys from Σ_i^n, we cover the members and add the corresponding public keys, starting from $S = \mathcal{O}$. In [BGW05] the authors propose to store the previous sum S from a session to the next, subtract the new revoked users and add the no-longer revoked ones. This is efficient only if the proportion of newly revoked and re-authorized users is very small.

3.4.4.2 Complexity analysis



(a) Random distribution of users



(b) Sorted distribution of users

Figure 3.13: Examples of a random and an sorted distribution of users

Example 21 (Computing S quickly in a tree)**.** *We consider a set of* $n = 16$ *users, indexed from* 1 *to* 16 *as illustrated in the two figures 3.13a and 3.13b. (revoked users are in black). For the user '2' the session key is computed by subtracting the values in the nodes '1,3,4,8,9,12,24' thus the cost is 6 EllAdd. Note that in Fig. 3.13b (the kind of sorted tree) the cost is only 3 EllAdd (we subtract the node '30' and '6'; and add the node '9'). The cost would have been* 8 *EllAdd in the original scheme.*

Algorithm 19 has something common with the Subset Cover computation. However here there is no need to store extra secrets elements, as the difference of the subset is done by a simple elliptic point subtraction. It is obvious that the number of operations is always lower than the number r of revoked users in Alg. 19 and lower than the number of members $n - r$ in its variant ($r > n/2$). It can be equal in the worst case: in this case S is just a difference (r operations) or just a sum ($n - r$ operations).

The average case is hard to analyze [AK08] as it strongly depends on the distribution of revoked users in the tree. When r or $(n - r)$ is small, with a uniform distribution, the complexity will be close to it. In practice the users are sorted by behavior so that nodes that are close are mostly to be revoked together. In a real world application the behavior is the subscription date or product. However some random revocations (rare events) appear with compromising, expiration, etc.

3.4.5 Implementation on a smartphone

In this last section we present our implementation results of BGW. The broadcaster is hosted on a PC and some users a simulated on smartphones.

For any implementation a trade-off between specificity (using a sparse modulus for quick reduction, using very specific curves) and performances has to be done. We chose to develop a very generic library in C language which can use any modulus and any type of pairing-friendly elliptic curve in Weiertraß representation over a prime finite field (i.e. in large characteristic). The BN curves and supersingular curves have been implemented. The library LIBCRYPTOLCH [Tha13] is a proprietary industrial library using a modular approach as in OpenSSL. It implements arithmetic over $\mathbb{F}_p$ using Montgomery multiplication, elliptic curve computation over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ using the modified Jacobian coordinates. The pairing computation is specific for each $\mathbb{F}_{p^k}$ field. The construction of the extension field $\mathbb{F}_{p^k}$ and its arithmetic is quite automated by using macros in C. The implementation details are presented in Sec. 3.2.

We now present some computational results of our improved implementation for 128-bit security level. Our proof of concept consists in a standard PC to represent the sender, and a smartphone to represent the receiver. The smartphone can be personalized with any secret key of the system. Thus the given results for decryption step on one receiver device are the same as would be in a real system with a million smartphones. The smartphone is a dual core 1.2 Ghz Samsung Galaxy II with Android OS. The PC is a 3Ghz Intel(R) Core(TM)2 Duo CPU with 2.9 Gio RAM. The last improvements described in Sec. 3.4.4 where unfortunately not yet implemented.

The broadcaster runs the system initialization, the key attribution to a new user and the session key encryption. First, we simulate the decryption time for an authorized user on the PC to estimate the growing cost of decryption with respect to the total number of users n, see Tab. 3.19.

Smartphones with Android platform use the Java programming language. Thanks to the Java Native Interface, we can load the library in C language, run the decryption on the smartphone and measure its timing. For doing that, we call the `currentTimeMillis()` function of the `system` class. Results are presented in Tab. 3.19 and Tab. 3.21. We measure the worst case $r = n/2$ of BGW_2 so the improvements described in Sec. 3.4.4 are not visible. The users are divided in A parallel groups of B users with $B = \lceil \sqrt{n} \rceil$.

Number of users n	Setup	User init.	Encryption $r = n/2$	Decryption (simulation)	Decryption (smartphone)
50000	22.15 s	0.03 s	3.58 s	1.10 s	1.44 s
100000	40.45 s	0.03 s	7.03 s	1.13 s	1.79 s
200000	1 m 16 s	0.03 s	14.72 s	1.14 s	2.08 s
500000	3 m 07 s	0.05 s	32.97 s	1.16 s	2.65 s
1000000	6 m 09 s	0.07 s	1 m 04 s	1.18 s	3.33 s
3000000	18 m 24 s	0.12 s	3 m 07 s	1.23 s	4.96 s
5000000	30 m 42 s	0.16 s	5 m 11 s	1.27 s	6.09 s

Table 3.19: Computation time obtained on a 3 Ghz PC (encryption) and a smartphone Samsung Galaxy SII 1.20 Ghz Android (decryption)

The decryption time depends on the total number of users and on the ratio of revoked users. The Tab. 3.20 and Tab. 3.21 show the increasing encryption and decryption times when r decreases from 87.5% to 0%.

Number n of users \ Members	12.5%	25%	50%	100%
50000	2.46 s	2.62 s	3.58 s	7.17 s
100000	3.11 s	4.10 s	7.03 s	13.84 s
200000	3.74 s	7.27 s	14.72 s	26.28 s
500000	9.65 s	16.46 s	32.97 s	1 m 03 s
1000000	16.99 s	33.46 s	1 m 04 s	2 m 06 s
3000000	49.67 s	1 m 36 s	3 m 07 s	6 m 11 s
5000000	1 m 20 s	2 m 37 s	5 m 11 s	10 m 18 s

Table 3.20: Encryption time with respect to the authorized user percentage obtained on the 3Ghz PC

An acceptable decryption time on the smartphone must be less than 2 seconds from our point of view. Here this correspond to less than 200 000 users according to Tab. 3.21. For larger n, we need to reduce this time. The pairing computation is not very time consuming. The sum $\sum_{j\in\mathcal{S}_a, j\neq b} P_{B+1-j+b}$ is the most important part of the computation time. With a first trick: addition over $\mathcal{S}_a$ when $n - r \ll n$ and subtraction over $\mathcal{R}_a$ (the revoked users of group a) when $r \ll n$, the worst case of $r = n/2$ become the upper bound. This means still at most $3.33s$ when $r = n/2$. With a precomputed tree, the average case will have faster encryption and decryption times than those presented in Tab. 3.21.

Number n of users \ Members	12.5%	25%	50%	100%
50000	1.18 s	1.20 s	1.44 s	1.93 s
100000	1.28 s	1.46 s	1.79 s	2.46 s
200000	1.36 s	1.60 s	2.08 s	3.03 s
500000	1.55 s	1.91 s	2.65 s	4.15 s
1000000	1.75 s	2.25 s	3.33 s	5.46 s
3000000	2.23 s	3.15 s	4.96 s	8.63 s
5000000	2.65 s	3.78 s	6.09 s	10.84 s

Table 3.21: Decryption time with respect to the authorized user percentage obtained on the smartphone

We manage to develop a functional prototype based on improved state-of-the-art broadcast protocol with a relative effectiveness. This provides consistent simulation time. In a real system, a dedicated Android implementation of the finite field arithmetic, the elliptic curve arithmetic and the pairing computation will certainly improve by a factor 2 or 3 our results, leading to less than 2 seconds to decipher, even for 5 000 000 users in the worth case of $r = n/2$.

3.4.6 Perspectives

We presented an improved version of BGW suitable for use with a pairing on one of the fastest pairing-friendly elliptic curves. Our presentation can be easily adapted to other well-suited pairing friendly elliptic curves. We considered the attacks on the underlying non-standard problem. We also provided computation time on a prototype, the broadcaster hosted on a standard PC and each receiver hosted on a Samsung Galaxy II smartphone with Android operating system. For large groups of users (more than 200000), the decryption time is up to 2 seconds which can be too slow. Hence we proposed improvements based on a time-memory trade-off. Because of the use of an asymmetric pairing, the public key size remains reasonable, hence doubling this size is feasible in order to reduce under 2 seconds the decryption time.

Since the new release of the Android Development Toolkit `rd8` of December 2012, it is possible to write some parts of (inline) code in ARM assembly language inside our C functions, then thanks to the Java Native Interface, the assembly and C codes are compiled to build an Android class. This is a work in progress. The results are expected to be available before the end of 2013. To finish, the PPSS protocol is a security improvement of BGW. We expect similar performances. The final results will appear in the last version of the ANR VERSO-09 project report.

3.5 Conclusion

In this chapter we presented our state-of-the-art implementation of pairings. The last gap to fill in order to break records is to design a dedicated assembly code for modular reduction for a given p, such as a $p(x)$ of a BN curve with $x = 2^{62} - 2^{54} + 2^{44}$ [BGDM+10]. At the moment, dedicated implementation for ARM architectures is very popular [SRH13]. Retrospectively, it was a good idea to focus on pairing-friendly curves over large characteristic fields. Indeed the new records announced since December 2012 have convinced the community to bannish the use supersingular pairing-friendly curves in small characteristic, because the new improved versions of the Function Field Sieve attack are prodigious.

The elliptic curves in large characteristic are still various enough to fill completely a 3-year PhD. Since 2005 supersingular and embedding-degree one curves are used to construct composite-order pairing-friendly groups. These curves are quite different than the ordinary ones. These is still some work to do to obtain an optimized pairing on these curves. Whatever happens these curves will remain much slower than curves such as Barreto-Naehrig curves.

Until December 2013 and the end of this PhD, some work still remains to do. For example we would like to link the ARM assembly code (for the modular multiplication) to the C code in the Android Development Toolkit in order to obtain a factor 3 speed-up on ARM smartphones for a pairing computation. Then we would have a much faster timing for any step of the BGW and PPSS broadcast encryption scheme we implemented.

Conclusion

In this thesis we discussed efficient arithmetic and pairing computation on elliptic curves for use in cryptographic protocols. We studied how to do efficient arithmetic on two families of elliptic curves and two other families of genus 2 hyperelliptic curves, isogenous to each other over an extension field. We can perform efficiently a scalar multiplication on these genus 2 curves with a 4-GLV decomposition method. The curve is naturally equipped with a first endomorphism. We showed an explicit way of constructing a second endomorphism from complex multiplication. Since the genus 2 curve is isogenous over some small extension field to the product of two elliptic curves, we first construct the elliptic curve with an endomorphism from complex multiplication and transport this endomorphism on the Jacobian of the genus 2 curve. We can do that for the two families we studied. We then know an second endomorphism on the genus 2 curve, an explicit way to compute its expression and its eigenvalue.

We discovered that the isogenous elliptic curve when defined over a quadratic extension of a finite field has anyway an endomorphism, different than the complex multiplication, coming from the composition the isogeny with the genus 2 curve and a Frobenius on this genus 2 curve. The same method of efficient scalar multiplication with a 4-dimensional GLV decomposition is available on this elliptic curve too. Previously known such elliptic curves with two distinct endomorphisms appear as a degenerate case of our construction. These results were discovered together with Ionica and presented at the *ECC'2013* workshop and *ASIACRYPT'2013* conference.

We also proposed an improvement of point-counting method, then pairing-friendly constructions on our two families of genus 2 curves. However we did not manage to construct interesting curves with optimal parameter sizes, in other words with $\rho = 1$. We only found constructions with $2 \leqslant \rho \leqslant 4$, i.e. the prime subgroup order considered for a pairing implementation is between a quarter and a half of the curve order. Finding such optimal genus 2 curves over prime fields or elliptic curves over a quadratic extension of a prime field with almost optimal parameter size seems a very hard task. No known such construction exists at the moment and the known methods for elliptic curves defined over prime fields fail to generalize to extension fields. Our results were presented at the *PAIRING'2012* conference.

We also presented in the second part of this thesis our efficient implementation of pairings in the cryptographic library of Thales and their application in a broadcast protocol. This is a joint work with Dubois and Sengelin Le Breton [DGSLB12] in a more general context of a project on broadcast encryption funded by the french ANR. This work was presented at the *PAIRING'2012* conference and was continued with Perez and Dugardin.

We also used our pairing implementation to compare two different instantiations (on different curves, with a different hard problem) of a HIBE protocol. The first version proposed by Lewko and Waters [LW11] uses composite-order pairing-friendly groups. The second version proposed by Lewko [Lew12] is a translation of the first protocol on a vector space over pairing-friendly prime-order groups. Our results show definitively that pairings on composite-order pairing-friendly groups are much slower, around a hundred time slower than pairings over prime-order groups. In protocols, this results in a slow-down of a factor 10 to 30, depending on the cryptographic operation (e.g. encryption, delegation). These results were presented at the *ACNS'2013* conference.

List of Figures

List of Tables

List of Algorithms

Bibliography

[ACD+05] R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, volume 34 of *Discrete Mathematics and its Applications*. CRC Press, Boca Raton, FL, 2005.

[AG35] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient LIbrary for Cryptography. available at `http://code.google.com/p/relic-toolkit/`, 2013, v-0.3.5. C++ language, LGPL license.

[AK08] Per Austrin and Gunnar Kreitz. Lower bounds for subset cover based broadcast encryption. In Serge Vaudenay, editor, *AFRICACRYPT 08: 1st International Conference on Cryptology in Africa*, volume 5023 of *Lecture Notes in Computer Science*, pages 343–356, Casablanca, Morocco, June 11–14, 2008. Springer, Berlin, Germany.

[AKL+11] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio López. Faster explicit formulas for computing pairings over ordinary curves. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 48–68, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.

[AL13] Michel Abdalla and Tanja Lange, editors. *Pairing-Based Cryptography - Pairing 2012 - 5th International Conference, Cologne, Germany, May 16-18, 2012, Revised Selected Papers*, volume 7708 of *Lecture Notes in Computer Science*. Springer, 2013.

[ALNS12] Tolga Acar, Kristin Lauter, Michael Naehrig, and Daniel Shumow. Affine pairings on ARM. In Abdalla and Lange [AL13], pages 203–209.

[AM93] Arthur O. L. Atkin and François Morain. Elliptic curves and primality proving. *Math. Comput.*, 61:29–68, 1993.

[AMORH13] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Weakness of $f_{3^{6\cdot 509}}$ for discrete logarithm cryptography. Cryptology ePrint Archive, Report 2013/446, 2013. `http://eprint.iacr.org/`.

[BBC+11a] Jennifer Balakrishnan, Juliana Belding, Sarah Chisholm, Kirsten Eisenträger, Katherine Stange, and Edlyn Teske. Pairings on hyperelliptic curves. In *WIN - Women in Numbers: Research Directions in Number Theory*, volume 60 of *Fields Institute Communications*, pages 87–120. Amer. Math. Soc., Providence, RI, 2011.

[BBC+11b] Jennifer Balakrishnan, Juliana Belding, Sarah Chisholm, Kirsten Eisenträger, Katherine Stange, and Edlyn Teske. Pairings on hyperelliptic curves. In *WIN - Women in Numbers: Research Directions in Number Theory*, volume 60 of *Fields Institute Communications*, pages 87–120. Amer. Math. Soc., Providence, RI, 2011.

[BBD+13] Razvan Barbulescu, Cyril Bouvier, Jérémie Detrey, Pierrick Gaudry, Hamza Jeljeli, Emmanuel Thomé, Marion Videau, and Paul Zimmermann. Discrete logarithm in $gf(2^809)$ with FFS. Cryptology ePrint Archive, Report 2013/197, 2013. `http://eprint.iacr.org/`.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.

[BC55] Karim Belabas and Henri Cohen. PARI/GP Library. available at `http://pari.math.u-bordeaux.fr`, 2013, v-2.5.5. C language, GPL license.

[BCF09] Naomi Benger, Manuel Charlemagne, and David Mandell Freeman. On the security of pairing-friendly abelian varieties over non-prime fields. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009: 3rd International Conference on Pairing-based Cryptography*, volume 5671 of *Lecture Notes in Computer Science*, pages 52–65, Palo Alto, CA, USA, August 12–14, 2009. Springer, Berlin, Germany.

[BCHL13a] Joppe W. Bos, Craig Costello, Hüseyin Hisil, and Kristin Lauter. High-performance scalar multiplication using 8-dimensional glv/gls decomposition. In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pages 331–348. Springer, 2013.

[BCHL13b] JoppeW. Bos, Craig Costello, Huseyin Hisil, and Kristin Lauter. Fast cryptography in genus 2. In Thomas Johansson and PhongQ. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 194–210. Springer Berlin Heidelberg, 2013.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[BGDM+10] Jean-Luc Beuchat, Jorge E. González-Díaz, Shigeo Mitsunari, Eiji Okamoto, Francisco Rodríguez-Henríquez, and Tadanori Teruya. High-speed software implementation of the optimal ate pairing over Barreto-Naehrig curves. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *PAIRING 2010: 4th International Conference on Pairing-based Cryptography*, volume 6487 of *Lecture Notes in Computer Science*, pages 21–39, Yamanaka Hot Spring, Japan, December 13–15, 2010. Springer, Berlin, Germany.

[BGJT13] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. Cryptology ePrint Archive, Report 2013/400, 2013. `http://eprint.iacr.org/`.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, Cambridge, MA, USA, February 10–12, 2005. Springer, Berlin, Germany.

[BGOS07] Paulo S. L. M. Barreto, Steven D. Galbraith, Colm O'Eigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptography*, 42(3):239–271, 2007.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany.

[Bis11] Gaetan Bisson. *Endomorphism rings in cryptography*. PhD thesis, Institut National Polytechnique de Lorraine, France and Technische Univertiteit Eindhoven, The Netherlands, 2011.

[BKLS02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany.

[BL07] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 29–50, Kuching, Malaysia, December 2–6, 2007. Springer, Berlin, Germany.

[BL12] Daniel J. Bernstein and Tanja Lange. Computing small discrete logarithms faster. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 317–338. Springer, 2012.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532, Gold Coast, Australia, December 9–13, 2001. Springer, Berlin, Germany.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

[BN05] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford Tavares, editors, *SAC 2005: 12th Annual International Workshop on Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331, Kingston, Ontario, Canada, August 11–12, 2005. Springer, Berlin, Germany.

[BRS11] Dan Boneh, Karl Rubin, and Alice Silverberg. Finding composite order ordinary elliptic curves using the Cocks-Pinch method. *Journal of Number Theory*, 131(5):832 – 841, 2011.

[BS10] Naomi Benger and Michael Scott. Constructing tower extensions of finite fields for implementation of pairing-based cryptography. In M. Anwar Hasan and Tor Helleseth, editors, *WAIFI*, volume 6087 of *Lecture Notes in Computer Science*, pages 180–195. Springer, 2010.

[BSS05] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Advances in Elliptic Curve Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2005.

[BW05] Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37(1):133–141, 2005.

[Cer12] Certivox. MIRACL Crypto SDK, 2012. `http://certivox.com/index.php/solutions/miracl-crypto-sdk/`.

[CF96] J.W.S. Cassels and E.V. Flynn. *Prolegomena to a Middelbrow Arithmetic of Curves of Genus 2*, volume 230 of *London Mathematical Society*. Cambridge University Press, 1996.

[Che06] Jung Hee Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany.

[Che10] Jung Hee Cheon. Discrete logarithm problems with auxiliary inputs. *Journal of Cryptology*, 23(3):457–476, July 2010.

[CL11] Craig Costello and Kristin Lauter. Group law computations on jacobians of hyperelliptic curves. In Ali Miri and Serge Vaudenay, editors, *SAC 2011: 18th Annual International Workshop on Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 92–117, Toronto, Ontario, Canada, August 11–12, 2011. Springer, Berlin, Germany.

[CP01] Clifford Cocks and Richard G.E. Pinch. ID-based cryptosystems based on the Weil pairing, 2001. Unpublished manuscript.

[CSB04] Sanjit Chatterjee, Palash Sarkar, and Rana Barua. Efficient computation of Tate pairing in projective coordinate over general characteristic fields. In Choonsik Park and Seongtaek Chee, editors, *ICISC 04: 7th International Conference on Information Security and Cryptology*, volume 3506 of *Lecture Notes in Computer Science*, pages 168–181, Seoul, Korea, December 2–3, 2004. Springer, Berlin, Germany.

[DEM05] Régis Dupont, Andreas Enge, and François Morain. Building curves with arbitrary small MOV degree over finite prime fields. *Journal of Cryptology*, 18(2):79–89, April 2005.

[Dew95] L. Dewaghe. Un corollaire aux formules de Vélu. Draft, 1995.

[DF10] Luca De Feo. *Fast Algorithms for Towers of Finite Fields and Isogenies*. PhD thesis, Ecole Polytechnique, december 2010. `https://github.com/defeo/PhD-Thesis`, `http://tel.archives-ouvertes.fr/tel-00547034`.

[DGB12] Renaud Dubois, Aurore Guillevic, and Marine Sengelin Le Breton. Improved broadcast encryption scheme with constant-size ciphertext. In Abdalla and Lange [AL13], pages 196–202.

[DGSLB12] Renaud Dubois, Aurore Guillevic, and Marine Sengelin Le Breton. Improved broadcast encryption scheme with constant-size ciphertext. In Michel Abdalla and Tanja Lange, editors, *PAIRING 2012: 5th International Conference on Pairing-based Cryptography*, volume 7708 of *Lecture Notes in Computer Science*, pages 196–202, Cologne, Germany, May 16–18, 2012. Springer, Berlin, Germany.

[DGV13] Jérémie Detrey, Pierrick Gaudry, and Marion Videau. Relation collection for the function field sieve. Cryptology ePrint Archive, Report 2013/071, 2013. `http://eprint.iacr.org/`.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DhSD06a] Augusto Jun Devegili, Colm Ó hÉigeartaigh, Michael Scott, and Ricardo Dahab. Multiplication and squaring on pairing-friendly fields. Cryptology ePrint Archive, Report 2006/471, 2006. `http://eprint.iacr.org/2006/471`.

[DhSD06b] Augusto Jun Devegili, Colm Ó hÉigeartaigh, Michael Scott, and Ricardo Dahab. Multiplication and squaring on pairing-friendly fields. Cryptology ePrint Archive, Report 2006/471, 2006.

[DJ11] Julien Devigne and Marc Joye. Binary huff curves. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 340–355, San Francisco, CA, USA, February 14–18, 2011. Springer, Berlin, Germany.

[DK05] Ivan Duursma and N. Kiyavash. The vector decomposition problem for elliptic and hyperelliptic curves. *Journal of the Ramanujan Mathematical Society*, 20(1):59–76, 2005.

[DPP07] Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *PAIRING 2007: 1st International Conference on Pairing-based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59, Tokyo, Japan, July 2–4, 2007. Springer, Berlin, Germany.

[DSD07] Augusto Jun Devegili, Michael Scott, and Ricardo Dahab. Implementing cryptographic pairings over Barreto-Naehrig curves (invited talk). In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *PAIRING 2007: 1st International Conference on Pairing-based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 197–207, Tokyo, Japan, July 2–4, 2007. Springer, Berlin, Germany.

[Edw07] Harold M. Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44:393–422, 2007. `http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html`.

[Eng12] Andreas Enge. CM Software, February 2012. `http://www.multiprecision.org/index.php?prog=cm`.

[ENSC+09] École Normale Supérieure, Paris 8, Thales Communications, Nagra, and Cryptoexperts. Broadcast encryption for secure telecommunications. Technical Report VERSO–09, Agence Nationale de la Recherche, 2009. `https://crypto.di.ens.fr/projects:best:main`.

[FHLS14] Armando Faz-Hernandez, Patrick Longa, and Ana H. Sanchez. Efficient and secure algorithms for glv-based scalar multiplication and their implementation on glv-gls curves. In *CT-RSA*, LNCS. Springer, 2014. to appear, pre-print available at `http://eprint.iacr.org/2013/158`.

[FKT04] Eisaku Furukawa, Mitsuru Kawazoe, and Tetsuya Takahashi. Counting points for hyperelliptic curves of type $y^2 = x^5 + ax$ over finite prime fields. In Mitsuru Matsui and Robert J. Zuccherato, editors, *SAC 2003: 10th Annual International Workshop on Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 26–41, Ottawa, Ontario, Canada, August 14–15, 2004. Springer, Berlin, Germany.

[FNI10] FNISA. Mécanismes cryptographiques - règles et recommandations. Technical Report Rev. 1.20, FNISA, France, January 2010.

[FR94] G. Frey and H. G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, 1994.

[Fre06] David Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 452–465. Springer, 2006.

[Fre10] David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[FS11] David Mandell Freeman and Takakazu Satoh. Constructing pairing-friendly hyperelliptic curves using Weil restriction. *J. Number Theory*, 131(5):959–983, 2011.

[FSS08] David Freeman, Peter Stevenhagen, and Marco Streng. Abelian varieties with prescribed embedding degree. In Alfred J. van der Poorten and Andreas Stein, editors, *Algorithmic Number Theory - ANTS VIII*, volume 5011 of *Lect Notes Comput. Sci.*, pages 60–73. Springer, 2008.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, April 2010.

[GAL+12] Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. Efficient implementation of bilinear pairings on arm processors. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 2012.

[Gau07] Pierrick Gaudry. Fast genus 2 arithmetic based on theta functions. *J. Math. Crypt.*, 1(3):243–265, 2007.

[GGMZ13a] Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$. Cryptology ePrint Archive, Report 2013/074, 2013. `http://eprint.iacr.org/`.

[GGMZ13b] Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Solving a 6120-bit dlp on a desktop computer. Cryptology ePrint Archive, Report 2013/306, 2013. `http://eprint.iacr.org/`.

[GHMM08] S. D. Galbraith, M. Harrison, and D. Mireles-Morales. Efficient hyperelliptic arithmetic using balanced representation for divisors. In A. J. van der Poorten and A. Stein, editors, *ANTS*, volume 5011 of *LNCS*, pages 342–356. Springer, 2008.

[GHS02] Steven D. Galbraith, Keith Harrison, and David Soldera. Implementing the tate pairing. In Claus Fieker and David R. Kohel, editors, *ANTS*, volume 2369 of *Lecture Notes in Computer Science*, pages 324–337. Springer, 2002.

[GHV07] Steven D. Galbraith, Florian Hess, and Frederik Vercauteren. Hyperelliptic pairings (invited talk). In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *PAIRING 2007: 1st International Conference on Pairing-based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 108–131, Tokyo, Japan, July 2–4, 2007. Springer, Berlin, Germany.

[GI13] Aurore Guillevic and Sorina Ionica. Four dimensional glv via the weil restriction. Cryptology ePrint Archive, Report 2013/311, 2013.

[GKS11] Pierrick Gaudry, David R. Kohel, and Benjamin A. Smith. Counting points on genus 2 curves with real multiplication. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 504–519, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Germany.

[GLS09] Steven D. Galbraith, Xibin Lin, and Michael Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 518–535, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[GLV01] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 190–200, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.

[GMV07] Steven D. Galbraith, James F. McKee, and P. C. Valença. Ordinary abelian varieties having small embedding degree. *Finite Fields and Their Applications*, 13(4):800–814, 2007. `http://eprint.iacr.org/2004/365`.

[GPRS09] Steven D. Galbraith, Jordi Pujolas, Christophe Ritzenthaler, and Benjamin Smith. Distortion maps for supersingular genus two curves. *J. Math. Crypt.*, 3(1):1–18, 2009.

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GS01] Pierrick Gaudry and Éric Schost. On the invariants of the quotients of the jacobian of a curve of genus 2. In Serdar Boztas and Igor Shparlinski, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes 2001*, volume 2227 of *Lect Notes Comput. Sci.*, pages 373–386. Springer, 2001.

[GS10] Robert Granger and Michael Scott. Faster squaring in the cyclotomic subgroup of sixth degree extensions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 209–223, Paris, France, May 26–28, 2010. Springer, Berlin, Germany.

[GS12] Pierrick Gaudry and Éric Schost. Genus 2 point counting over prime fields. *Journal of Symbolic Computation*, 47(4):368–400, 2012.

[Gui13] Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Jacobson et al. [JLMSN13], pages 357–372. `http://eprint.iacr.org/2013/218`.

[GV12] Aurore Guillevic and Damien Vergnaud. Genus 2 hyperelliptic curve families with explicit jacobian order evaluation and pairing-friendly constructions. In Michel Abdalla and Tanja Lange, editors, *PAIRING 2012: 5th International Conference on Pairing-based Cryptography*, volume 7708 of *Lecture Notes in Computer Science*, pages 234–253, Cologne, Germany, May 16–18, 2012. Springer, Berlin, Germany.

[Has97] Yuji Hasegawa. $\mathbb{Q}$-curves over quadratic fields. *manuscripta mathematica*, 94:347–364, 1997.

[Her11] Mathias Herrmann. Improved cryptanalysis of the multi-prime ϕ-hiding assumption. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11: 4th International Conference on Cryptology in Africa*, volume 6737 of *Lecture Notes in Computer Science*, pages 92–99, Dakar, Senegal, July 5–7, 2011. Springer, Berlin, Germany.

[Hon68] Taira Honda. Isogeny classes of abelian varieties over finite fields. *Journal of the Mathematical Society of Japan*, 20.I(2):83–95, 1968.

[HS00] Marc Hindry and Joseph H. Silverman. *Diophantine Geometry, An Introduction*, volume 201 of *GTM*. Springer, 2000.

[HSSI99] Ryuichi Harasawa, Junji Shikata, Joe Suzuki, and Hideki Imai. Comparing the MOV and FR reductions in elliptic curve cryptography. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 190–205, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany.

[HSST12] Takuya Hayashi, Takeshi Shimoyama, Naoyuki Shinohara, and Tsuyoshi Takagi. Breaking pairing-based cryptosystems using η_t pairing over $gf(3^{97})$. Cryptology ePrint Archive, Report 2012/345, 2012. `http://eprint.iacr.org/`.

[HSV06] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

[JL07] A. Joux and R. Lercier. Algorithmes pour résoudre le problème du logarithme discret dans les corps finis. In *Nouvelles Méthodes Mathématiques en Cryptographie*, Fascicules Journées Annuelles, pages 23–53. Société Mathématique de France, June 2007.

[JLMSN13] Michael J. Jr. Jacobson, Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors. *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, volume 7954 of *Lecture Notes in Computer Science*. Springer, 2013.

[Jou00] Antoine Joux. A one round protocol for tripartite diffie-hellman. In Wieb Bosma, editor, *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[Jou04] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, September 2004.

[Jou12] Antoine Joux. Faster index calculus for the medium prime case. application to 1175-bit and 1425-bit finite fields. Cryptology ePrint Archive, Report 2012/720, 2012. `http://eprint.iacr.org/`.

[Jou13a] Antoine Joux. Faster index calculus for the medium prime case application to 1175-bit and 1425-bit finite fields. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 177–193. Springer, 2013.

[Jou13b] Antoine Joux. A new index calculus algorithm with complexity $l(1/4 + o(1))$ in very small characteristic. Cryptology ePrint Archive, Report 2013/095, 2013. `http://eprint.iacr.org/`.

[JTV10] Marc Joye, Mehdi Tibouchi, and Damien Vergnaud. Huff's model for elliptic curves. In Guillaume Hanrot, François Morain, and Emmanuel Thomé, editors, *ANTS*, volume 6197 of *Lecture Notes in Computer Science*, pages 234–250. Springer, 2010.

[Kac10] Ezekiel J. Kachisa. Generating more Kawazoe-Takahashi genus 2 pairing-friendly hyperelliptic curves. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *PAIRING 2010: 4th International Conference on Pairing-based Cryptography*, volume 6487 of *Lecture Notes in Computer Science*, pages 312–326, Yamanaka Hot Spring, Japan, December 13–15, 2010. Springer, Berlin, Germany.

[Kal88] Burton S. Jr Kaliski. *Elliptic Curvaes and Cryptography: A Pseudorandom Bit Generator and Other Tools*. PhD thesis, Massachusetts Institute of Technology, February 1988. available at `http://groups.csail.mit.edu/cis/theses/kaliski-phd.pdf`.

[KKM07] Shunji Kozaki, Taketeru Kutsuma, and Kazuto Matsuo. Remarks on Cheon's algorithms for pairing-related problems. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *PAIRING 2007: 1st International Conference on Pairing-based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 302–316, Tokyo, Japan, July 2–4, 2007. Springer, Berlin, Germany.

[KKSZ10] Elisavet Konstantinou, Aristides Kontogeorgis, Yannis C. Stamatiou, and Christos Zaroliagis. On the efficient generation of prime-order elliptic curves. *Journal of Cryptology*, 23(3):477–503, July 2010.

[KM05] Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 13–36, Cirencester, UK, December 19–21, 2005. Springer, Berlin, Germany.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):pp. 203–209, 1987.

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1(3):139–150, 1989.

[Kob90] Neal Koblitz. A family of Jacobians suitable for discrete log cryptosystems. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 94–99, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany.

[Koh96] David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkeley, 1996.

[KOS10] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 295–313, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.

[KSS08] Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008: 2nd International Conference on Pairing-based Cryptography*, volume 5209 of *Lecture Notes in Computer Science*, pages 126–135, Egham, UK, September 1–3, 2008. Springer, Berlin, Germany.

[KSZ07] Elisavet Konstantinou, Yannis Stamatiou, and Christos Zaroliagis. Efficient generation of secure elliptic curves. *International Journal of Information Security*, 6:47–63, 2007.

[KT08] Mitsuru Kawazoe and Tetsuya Takahashi. Pairing-friendly hyperelliptic curves with ordinary jacobians of type $y^2 = x^5 + ax$. In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008: 2nd International Conference on Pairing-based Cryptography*, volume 5209 of *Lecture Notes in Computer Science*, pages 164–177, Egham, UK, September 1–3, 2008. Springer, Berlin, Germany.

[Lab10] Laboratory of Cryptography and Information Security, University of Tsukuba, Japan. University of tsukuba elliptic curve and pairing library. available at `http://www.cipher.risk.tsukuba.ac.jp/tepla/`, 2013, v-1.0. C language.

[LB] Tanja Lange and Daniel Bernstein. Explicit-formulas database. `http://www.hyperelliptic.org/EFD/`.

[Len01] Arjen K. Lenstra. Unbelievable security. matching AES security using public key systems (invited talk). In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 67–86, Gold Coast, Australia, December 9–13, 2001. Springer, Berlin, Germany.

[Len04] Arjen K. Lenstra. Key lengths, contribution to the handbook of information security. June 2004.

[Ler97] Reynald Lercier. *Algorithmique des courbes elliptiques dans les corps finis*. PhD thesis, École Polytechnique, 1997.

[Lew12] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.

[LL93] Arjen K. Lenstra and Hendrik W. Jr. Lenstra, editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1993.

[LLV05] Reynald Lercier, David Lubicz, and Frederik Vercauteren. *Point Counting on Elliptic and Hyperelliptic Curves*, volume 34 of *Discrete Mathematics and its Applications.*, chapter 17, pages 239–263. CRC Press, Boca Raton, FL, 2005.

[LM97] F. Leprévost and F. Morain. Revêtements de courbes elliptiques à multiplication complexe par des courbes hyperelliptiques et sommes de caractères. *J. Number Theory*, 64:165–182, 1997.

[LN97] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2nd edition, 1997.

[LS12] Patrick Longa and Francesco Sica. Four-dimensional gallant-lambert-vanstone scalar multiplication. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 718–739, Beijing, China, December 2–6, 2012. Springer, Berlin, Germany.

[LS13] Patrick Longa and Francesco Sica. Four dimensional gallant-lambert-vanstone scalar multiplication. *Journal of Cryptology*, pages 1–36, 2013.

[LV01] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[LV05] David Lubicz and Frederik Vercauteren. *Cohomological Background on Point Counting*, volume 34 of *Discrete Mathematics and its Applications.*, chapter 8, pages 133–142. CRC Press, Boca Raton, FL, 2005.

[LW11] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.

[Lyn14] Benjamin Lynn. Pairing-based cryptography library. available at `http://crypto.stanford.edu/pbc/`, 2013, v-0.5.14. C language, LGPL license.

[MGI11] Nadia El Mrabet, Aurore Guillevic, and Sorina Ionica. Efficient multiplication in finite field extensions of degree 5. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11: 4th International Conference on Cryptology in Africa*, volume 6737 of *Lecture Notes in Computer Science*, pages 188–205, Dakar, Senegal, July 5–7, 2011. Springer, Berlin, Germany.

[Mil86a] Victor Miller. Short programs for functions on curves, 1986.

[Mil86b] Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology – CRYPTO'85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Santa Barbara, CA, USA, August 18–22, 1986. Springer, Berlin, Germany.

[Mil04] Victor S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, September 2004.

[MNT00] Atsuko Miyaji, Masaki Nakabayashi, and Shunzo Takano. Characterization of elliptic curve traces under fr-reduction. In Dongho Won, editor, *ICISC*, volume 2015 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2000.

[MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[Mum83] David Mumford. *Tata Lectures on Theta*, volume Part II. Birkhauser-Boston, 1983. based on lectures given at the Tata Institute in 1978-79.

[MvV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. The CRC Press series on discrete mathematics and its applications. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA, 1997. `http://cacr.uwaterloo.ca/hac/`.

[MW99] Ueli M. Maurer and Stefan Wolf. The relationship between breaking the diffie-hellman protocol and computing discrete logarithms. *SIAM J. Comput.*, 28(5):1689–1721, 1999.

[NIS11] NIST. Recommendation for key management, special publication 800-57 part 1 rev. 3. Technical Report 800-57 Part 1 Rev. 3, NIST, USA, May 2011.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.

[NNS10] Michael Naehrig, Ruben Niederhagen, and Peter Schwabe. New software speed records for cryptographic pairings. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *Progress in Cryptology - LATINCRYPT 2010: 1st International Conference on Cryptology and Information Security in Latin America*, volume 6212 of *Lecture Notes in Computer Science*, pages 109–123, Puebla, Mexico, August 8–11, 2010. Springer, Berlin, Germany.

[NSA10] NSA. Fact sheet suite b cryptography. Technical report, NSA, USA, 11 2010.

[oEiCI11] European Network of Excellence in Cryptology II. Ecrypt ii yearly report on algorithms and keysizes. Technical Report D.SPA.17 Rev. 1.0, ICT-2007-216676 ECRYPT II, European Union, June 2011. `http://www.ecrypt.eu.org/documents/D.SPA.17.pdf`.

[oEiCI12] European Network of Excellence in Cryptology II. Ecrypt ii yearly report on algorithms and keysizes. Technical Report D.SPA.20 Rev. 1.0, ICT-2007-216676 ECRYPT II, European Union, Sept 2012. `http://www.ecrypt.eu.org/documents/D.SPA.20.pdf`.

[OH04] H. Orman and P. Hoffman. Determining strengths for public keys used for exchanging symmetric keys, rfc 3766. Technical Report RFC 3766, 04 2004.

[PPS11] Duong Hieu Phan, David Pointcheval, and Mario Strefler. Security notions for broadcast encryption. In Javier Lopez and Gene Tsudik, editors, *ACNS 11: 9th International Conference on Applied Cryptography and Network Security*, volume 6715 of *Lecture Notes in Computer Science*, pages 377–394, Nerja, Spain, June 7–10, 2011. Springer, Berlin, Germany.

[PPSS12] Duong Hieu Phan, David Pointcheval, Siamak Fayyaz Shahandashti, and Mario Strefler. Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP 12: 17th Australasian Conference on Information Security and Privacy*, volume 7372 of *Lecture Notes in Computer Science*, pages 308–321, Wollongong, NSW, Australia, July 9–11, 2012. Springer, Berlin, Germany.

[PPSS13] Duong Hieu Phan, David Pointcheval, Siamak Fayyaz Shahandashti, and Mario Strefler. Adaptive cca broadcast encryption with constant-size secret keys and ciphertexts. *Int. J. Inf. Sec.*, 12(4):251–265, 2013.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[RSK00] Kiyoshi Ohgishi Ryuichi Sakai and Masao Kasahara. Cryptosystems based on pairing. In *Symposium on Cryptography and Information Security (SCIS), Okinawa, Japan, January 26-28*, 2000.

[Sat02] Takakazu Satoh. On p-adic point counting algorithms for elliptic curves over finite fields. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory* - ANTS-*V*, volume 2369 of *Lect Notes Comput. Sci.*, pages 43–66. Springer, 2002.

[Sat09] Takakazu Satoh. Generating genus two hyperelliptic curves over large characteristic finite fields. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 536–553, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[Sch98] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comput.*, 44:483–494, 1998.

[Sco11] Michael Scott. MIRACL library. `www.shamus.ie`, August 2011. V5.5.4.

[Seo12] Jae Hong Seo. On the (im)possibility of projecting property in prime-order setting. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 61–79. Springer, 2012.

[SHI+12] Yumi Sakemi, Goichiro Hanaoka, Tetsuya Izu, Masahiko Takenaka, and Masaya Yasuda. Solving a discrete logarithm problem with auxiliary input on a 160-bit elliptic curve. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Workshop on Theory and Practice in Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 595–608, Darmstadt, Germany, May 21–23, 2012. Springer, Berlin, Germany.

[Sil94] Joseph H. Silverman. *Advanced Topics in the Arithmetic of Elliptic Curves*, volume 151 of *Graduate Texts in Mathematics*. Springer, 1994.

[Sil09] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, 2009. 2nd edition.

[Smi13] Benjamin Smith. Families of fast elliptic curves from $\mathbb{Q}$-curves. In Kazue Sako and Palash Sarkar, editors, *Asiacrypt*, LNCS. Springer, 2013. to appear.

[SRH13] Ana Helena Sánchez and Francisco Rodríguez-Henríquez. NEON implementation of an attribute-based encryption scheme. In Jacobson et al. [JLMSN13], pages 322–338.

[SS13] Kazue Sako and Palash Sarkar, editors. *Four dimensional GLV via the Weil restriction*, LNCS. Springer, 2013.

[ST94] Joseph H. Silverman and John Tate. *Rational Points on Elliptic Curves*. Undergraduate Texts in Mathematics. Springer, 1994.

[Sut12] Andrew V. Sutherland. Accelerating the CM method. *LMS J. Comput. Math.*, 15:172–204, 2012.

[Tak06] Katsuyuki Takashima. A new type of fast endomorphisms on jacobians of hyperelliptic curves and their cryptographic application. *IEICE Transactions*, 89-A(1):124–133, 2006.

[Tat66] John Tate. Endomorphism of abelian varieties over finite fields. *Inventiones mathematicae*, 2(2):134–144, 1966.

[Tat68] John Tate. Classes d'isgénie des variétés abéliennes sur un corps fini (d'aprè t. honda). *Séminaire Bourbaki*, 2 I(352):95–110, 1968.

[Tha13] Thales Communications and Security. LIBCRYPTOLCH librairie cryptographique du laboratoire chiffre, 2013.

[Ver10] Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.

[Ver12] Damien Vergnaud. *Exercices et problèmes de cryptographie*. Sciences Sup. Dunod, march 2012.

[Vol] Voltage security. Voltage identity-based encryption. `http://www.voltage.com/technology/ibe.htm`.

[Was03] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography*. Discrete Mathematics and Its Applications. Taylor & Francis, 2003.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.

[ZXW+12] Ye Zhang, Chun Jason Xue, Duncan S. Wong, Nikos Mamoulis, and Siu Ming Yiu. Acceleration of composite order bilinear pairing on graphics hardware. In Tat Wing Chim and Tsz Hon Yuen, editors, *ICICS*, volume 7618 of *Lecture Notes in Computer Science*, pages 341–348. Springer, 2012.

[ZZX12] Changan Zhao, Fangguo Zhang, and Dongqing Xie. Faster computation of self-pairings. *IEEE Transactions on Information Theory*, 58(5):3266–3272, 2012.

Résumé

Depuis 2000 les couplages sont devenus un très bon outil pour la conception de nouveaux protocoles cryptographiques. Les signatures courtes et le chiffrement basé sur l'identité sont devenus réalisables grâce aux couplages.

Les travaux réalisés dans cette thèse comprennent deux aspects complémentaires. Une partie consiste en l'implémentation optimisée de couplages sur différentes courbes elliptiques, en fonction des protocoles visés. Une implémentation sur des courbes supersingulières en grande caractéristique et sur des courbes de Barreto-Naehrig est détaillée. La bibliothèque développée au Laboratoire Chiffre de Thales est utilisée avec des courbes de Barreto-Naehrig dans un protocole de diffusion chiffrée. La seconde application évalue la différence de temps de calcul pour des protocoles utilisant les couplages sur des courbes d'ordre composé (un large module RSA) et la traduction de ces protocoles qui utilise plusieurs couplages sur des courbes plus habituelles. Les résultats montrent une différence d'un facteur de 30 à 250 en fonction des étapes des protocoles, ce qui est très important.

Une seconde partie porte sur deux familles de courbes de genre deux. Les jacobiennes de ces courbes sont isogènes au produit de deux courbes elliptiques sur une extension de corps de petit degré. Cette isogénie permet de transférer les propriétés des courbes elliptiques vers les jacobiennes. Le comptage de points est aisé et ne requiert qu'un comptage de points sur une des courbes elliptiques isogènes, plus quelques ajustements. On présente aussi la construction de deux endomorphismes à la fois sur les jacobiennes et sur les courbes elliptiques. Ces deux endomorphismes permettent des multiplications scalaires efficaces en suivant la méthode de Gallant, Lambert et Vanstone, ici en dimension quatre.

mots-clés : courbes elliptiques, courbes de genre 2, endomorphismes, couplages, implémentation, groupes d'ordre composé.

Abstract

Since 2000 pairings became a very useful tool to design new protocols in cryptography. Short signatures and identity-based encryption became also practical thanks to these pairings.

This thesis contains two parts. One part is about optimized pairing implementation on different elliptic curves according to the targeted protocol. Pairings are implemented on supersingular elliptic curves in large characteristic and on Barreto-Naehrig curves. The pairing library developed at Thales is used in a broadcast encryption scheme prototype. The prototype implements pairings over Barreto-Naehrig curves. Pairings over supersingular curves are much slower and have larger parameters. However these curves are interesting when implementing protocols which use composite-order elliptic curves (the group order is an RSA modulus). We implement two protocols that use pairings on composite-order groups and compare the benchmarks and the parameter size with their counterpart in a prime-order setting. The composite-order case is 30 up to 250 times much slower according to the considered step in the protocols: the efficiency difference in between the two cases is very important.

A second part in this thesis is about two families of genus 2 curves. Their Jacobians are isogenous to the product of two elliptic curves over a small extension field. The properties of elliptic curves can be translated to the Jacobians thanks to this isogeny. Point counting is as easy as for elliptic curves in this case. We also construct two endomorphisms both on the Jacobians and the elliptic curves. These endomorphisms can be used for scalar multiplication improved with a four-dimensional Gallant-Lambert-Vanstone method.

keywords: elliptic curves, genus 2 curves, endomorphisms, pairings, implementation, composite-order groups.