An Introduction to Elliptic and Hyperelliptic Curve Cryptography and the NTRU Cryptosystem

Jasper Scholten and Frederik Vercauteren

K.U. Leuven,
Dept. Elektrotechniek-ESAT/COSIC,
Kasteelpark Arenberg 10,
B-3001 Leuven-Heverlee,
Belgium.
`Firstname.Lastname@esat.kuleuven.ac.be`

Abstract. This paper provides a self-contained introduction to elliptic and hyperelliptic curve cryptography and to the NTRU cryptosystem. The goal is to introduce the necessary mathematical background, detail various existing encryption and signature schemes and give an overview of the known security weaknesses.

1 Introduction

In their seminal paper [6], Diffie and Hellman introduced the notion of public key cryptography. They described how two entities can agree on a common secret key by communicating over an insecure channel. This is known as the Diffie-Hellman key agreement protocol. The security of the protocol is related to the apparent difficulty of computing discrete logarithms modulo a large prime number p, i.e. given two numbers $(g \bmod p)$ and $(g^x \bmod p)$, it seems to be infeasible to compute x for general large enough p. A few years later, Rivest, Shamir and Adleman [37] proposed a public key encryption scheme and a digital signature scheme, the security of which is related to factoring a large integer.

The papers [6] and [37] laid the foundations of public key cryptography. Since their appearance, many other schemes have been proposed that are based on the Integer Factorisation Problem and the Discrete Logarithm Problem (DLP), such as the ElGamal encryption and signature scheme [7] and the Digital Signature Algorithm (DSA) [8].

Instead of using the DLP modulo a large prime p as the basis of cryptographic protocols, one can consider the DLP in an arbitrary group that admits an efficient element representation and group law. Let G be such a group, then the DLP in G is defined as follows: given two elements g and $h = g^x \in G$, determine the exponent x. The reason for considering other groups is that the most efficient methods for solving the DLP in a general, i.e. black-box, group take $\mathcal{O}(\sqrt{\mathrm{ord}(g)})$ steps [43]. The DLP for the integers modulo a prime (or more generally, in the multiplicative group of any finite field $\mathbb{F}_q^*$) can be solved in a far more efficient way, requiring only $\mathcal{O}(\exp(\log(q)^{1/2}\log(\log(q))^{1/2}))$ steps. So if one uses a group in which the DLP is as hard as for a general group, one can use much smaller parameters and key sizes than when using the multiplicative group of finite fields and still obtain the same level of security.

Miller [32] and Koblitz [26] proposed to use the group of points on an elliptic curve E defined over a finite field. Later, Koblitz [27] suggested to use the group of points on the Jacobian of a hyperelliptic curve C defined over a finite field. If the curves are chosen carefully then, as far as one knows, the DLP in these groups is as hard as for general groups.

Not only does this imply shorter key sizes, but also smaller footprints and code size. Another cryptosystem with similar properties is the NTRU cryptosystem proposed by Hoffstein, Pipher and Silverman at the rump session of Crypto '96 [22]. The security of the NTRU cryptosystem is based on polynomial arithmetic in the ring $\mathbb{Z}[X]/(X^N - 1)$ modulo two unrelated moduli.

2 Elliptic Curves

Let $\mathbb{F}_q$ denote a finite field of characteristic p, i.e. $q = p^\ell$ with p prime. Although it is possible to define the notion of an elliptic curve over any field by giving a general equation, we will make a distinction between the cases $p = 2$ and $p > 2$. Our treatment will also fail to deal with some specific elliptic curves. We do this in order to keep the exposition close to cryptographic practise, where one often deals with either the case $p = 2$ or the case $q = p$.

If $p = 2$ then an elliptic curve E defined over $\mathbb{F}_q$ is given by an equation

$$y^2 + xy = x^3 + ax^2 + b,$$

where $a, b \in \mathbb{F}_q$ and $b \neq 0$. For every field K containing $\mathbb{F}_q$ (so in particular for $K = \mathbb{F}_q$) one considers the set

$$E(K) := \{(x, y) \in K \times K \mid y^2 + xy = x^3 + ax^2 + b\} \cup \{\infty\}.$$

With this definition, we left out some special elliptic curves, the so-called *supersingular curves*, but we will not need these.

If $p > 2$ then an elliptic curve defined over $\mathbb{F}_q$ is given by an equation

$$y^2 = x^3 + ax + b,$$

where $a, b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$. For every field K containing $\mathbb{F}_q$ one now considers the set

$$E(K) := \{(x, y) \in K \times K \mid y^2 = x^3 + ax + b\} \cup \{\infty\}.$$

With this definition we left out a few elliptic curves in characteristic 3, but again we will not need these.

The set $E(K)$ is called the set of K-rational points on E. The symbol ∞ is called *the point at infinity.*

As is well known, one can endow $E(K)$ with the structure of an Abelian group. It is common practise to denote the group operations in an additive way (i.e. using $+$ and $-$ symbols), as opposed to the multiplicative notation when dealing with groups like $\mathbb{F}_q^*$. The group law is defined by the following general rules: ∞ is the zero element, and any three points that lie on a line (i.e. that are solutions of a linear equation in x and y) add up to zero. Working this out yields the following explicit rules for adding two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$.

The case $p = 2$. The opposite of the point P_1 is $-P_1 = (x_1, y_1 + x_1)$. The addition law is given by the following formulae with $P_3 = P_1 + P_2$:

1. $P_3 = \infty$ if $x_1 = x_2$ and $y_1 \neq y_2$,
2. Define $\lambda = (x_1^2 + y_1)/x_1$ and $\nu = x_1^2$ if $P_1 = P_2$,
3. Define $\lambda = (y_2 + y_1)/(x_2 + x_1)$ and $\nu = (y_1 x_2 + y_2 x_1)/(x_2 + x_1)$ if $x_1 \neq x_2$.
4. If not in case 1, then $x_3 = \lambda^2 + \lambda + x_1 + x_2$ and $y_3 = (\lambda + 1)x_3 + \nu$.

The case $p > 2$. The opposite of the point P_1 is $-P_1 = (x_1, -y_1)$. The addition law is given by the following formulae with $P_3 = P_1 + P_2$:

1. $P_3 = \infty$ if $x_1 = x_2$ and $y_1 \neq y_2$,
2. Define $\lambda = (3x_1^2 + a)/(2y_1)$ and $\nu = (-x_1^3 + ax_1 + 2b)/(2y_1)$ if $P_1 = P_2$,
3. Define $\lambda = (y_2 - y_1)/(x_2 - x_1)$ and $\nu = (y_1 x_2 - y_2 x_1)/(x_2 - x_1)$ if $x_1 \neq x_2$.
4. If not in case 1, then $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = -\lambda x_3 - \nu$.

It is important to know the size of the group $E(\mathbb{F}_q)$. This size determines the security level of the cryptosystems based on it. The following theorem shows that $\#E(\mathbb{F}_q)$ is roughly equal to q.

Theorem 1 (Helmut Hasse). *If E is an elliptic curve over a finite field $\mathbb{F}_q$, then*

$$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q}.$$

3 Hyperelliptic Curves

3.1 Definitions

A hyperelliptic curve C of genus g defined over a field $\mathbb{F}_q$ of characteristic p is given by an equation of the form

$$y^2 + h(x)y = f(x) \qquad (1)$$

where $h(x)$ and $f(x)$ are polynomials with coefficients in $\mathbb{F}_q$, with $\deg h(x) \leq g$ and $\deg f(x) = 2g + 1$. An additional requirement is that C is not a *singular* curve. If $h(x) = 0$ and $p > 2$ this amounts to the requirement that $f(x)$ is a squarefree polynomial. In general, the condition is that there are no x and y in the algebraic closure of $\mathbb{F}_q$ that satisfy the equation (1) and the two partial derivatives $2y + h(x) = 0$ and $h'(x)y - f'(x) = 0$.

For any extension K of $\mathbb{F}_q$ consider the set

$$C(K) := \{(x,y) \in K \times K \mid y^2 + h(x)y = f(x)\} \cup \{\infty\}.$$

It is called the set of K-rational points on C. The point ∞ is called the *point at infinity*; the other points are called *finite points*. As opposed to the case of elliptic curves, there is no natural way to provide $C(K)$ with a group structure. Instead, one can introduce a different object related to C, which to each field extension K of $\mathbb{F}_q$ associates a group. This object is called the *Jacobian* of C.

First we introduce several other groups, consisting of so-called *divisors*. In the remainder of this section, we denote the algebraic closure of $\mathbb{F}_q$ by L. A divisor D of the curve C is a formal sum

$$\sum_{P \in C(L)} c_P[P]$$

with $c_P \in \mathbb{Z}$ such that only finitely many c_P are nonzero. The set of all divisors is denoted by $\mathrm{Div}_C(L)$. Given two divisors $D = \sum_P c_P[P]$ and $D' = \sum_P c'_P[P]$, we define the sum $D + D'$ as $\sum_P (c_P + c'_P)[P]$. This gives $\mathrm{Div}_C(L)$ a group structure.

Next, we define several subgroups of $\mathrm{Div}_C(L)$. The *degree* of a divisor $D = \sum_P c_P[P]$ is $\deg(D) = \sum_P c_P$. The group of degree zero divisors is

$$\mathrm{Div}^0_C(L) := \{D \in \mathrm{Div}_C(L) \mid \deg(D) = 0\}.$$

To a point $P \in C(L)$ one can associate a new point $P^\sigma \in C(L)$. If $P = \infty$, then $P^\sigma := \infty$ by definition. If $P = (x,y) \in C(L)$ is a finite point then $P^\sigma := (x^q, y^q)$. From the assumption that the defining equation of C has coefficients in $\mathbb{F}_q$, it easily follows that $P^\sigma \in C(L)$. To a divisor $D = \sum_P c_P[P]$ one associates the divisor $D^\sigma = \sum_P c_P[P^\sigma]$. This allows us to define the following two subgroups of $\mathrm{Div}_C(L)$. If $\#K = q^r$ then

$$\begin{aligned} \mathrm{Div}_C(K) &:= \{D \in \mathrm{Div}_C(L) \mid D^{\sigma^r} = D\}, \\ \mathrm{Div}^0_C(K) &:= \{D \in \mathrm{Div}^0_C(L) \mid D^{\sigma^r} = D\}. \end{aligned}$$

Before introducing some more subgroups of $\mathrm{Div}_C(L)$ we have to say a few words about *rational functions* on the curve C. Let K be a field, $\mathbb{F}_q \subset K \subset L$, and let $F(x,y) \in K[x,y]$ be a polynomial with coefficients in K. At each finite point $P \in C(L)$ one can evaluate F. This yields a function $F : C(L) \setminus \{\infty\} \to L$. The *order of vanishing* of F at P, denoted by $\mathrm{ord}_P(F)$, is an integer that is zero if and only $F(P) \neq 0$, otherwise it is positive. A precise definition can be given as follows: Let $P = (x_0, y_0)$ and consider the ring $L[[x - x_0, y - y_0]]$ of formal power series in $x - x_0$ and $y - y_0$. The polynomial $F(x,y)$ and the defining polynomial $y^2 + h(x)y - f(x)$ of C can be considered as elements of this ring. These elements generate an ideal. The quotient ring

$$L[[x - x_0, y - y_0]]/(F(x,y), y^2 + h(x)y - f(x))$$

is a vector space over L that can be shown to be finite dimensional if F does not represent the zero function on $C(L)$. By definition, this dimension is $\mathrm{ord}_P(F)$. One should think of this as the smallest degree of the terms in a Taylor series expansion of F.

More generally, let F and G be two polynomials in $K[x,y]$ that do not represent the zero function on $C(L)$, and consider the rational function F/G. For finite P, define $\mathrm{ord}_P(F/G) := \mathrm{ord}_P(F) - \mathrm{ord}_P(G)$. For $P = \infty$, define

$$\mathrm{ord}_P(F/G) = - \sum_{P \in C(L)\setminus\{\infty\}} \mathrm{ord}_P(F/G).$$

This allows one to associate a divisor of degree zero to each rational function that is not zero on $C(L)$:

$$\mathrm{div}(F/G) := \sum_{P \in C(L)} \mathrm{ord}_P(F/G)[P].$$

The divisors that arise as divisors of rational functions with coefficients in L are called *principal divisors*. They form a subgroup of $\mathrm{Div}_C^0(L)$, denoted by $P_C(L)$. For any field K with $\mathbb{F}_q \subset K \subset L$ one defines $P_C(K) := P_C(L) \cap \mathrm{Div}_C^0(K)$.

Finally we are able to introduce the group that we need for the cryptographic application.

Definition 1. *For a field K, with $\mathbb{F}_q \subset K \subset L$, the* group of K-rational points of the Jacobian of C *is $\mathrm{Div}_C^0(K)/P_C(K)$. It is denoted by $J_C(K)$.*

The definition of the groups $J_C(K)$ is quite involved. However, it can be considered as a natural generalisation of the group of points on elliptic curves as explained in Section 2. Let E be an elliptic curve. Consider the map $\phi : E(K) \to J_E(K)$ that maps a point P to the class of the degree-zero-divisor $[P] - [\infty]$. This map can be shown to be a group isomorphism. The fact that it is a homomorphism can be shown as follows: Let P, Q and R be three points of $E(K)$ that add up to zero. So they lie on a line, say with equation $F(x,y) = 0$ for some linear $F \in K[x,y]$. Then

$$\phi(P) + \phi(Q) + \phi(R) = [P] + [Q] + [R] - 3[\infty] \in \mathrm{Div}_C^0(K)$$

represents the zero element of $J_C(K)$ since it is the divisor of the rational function F.

Just as in the elliptic case, it is important to know how big the group $J_C(\mathbb{F}_q)$ is. It has roughly q^g elements, as follows from the following theorem.

Theorem 2 (André Weil). *If C is a hyperelliptic curve over $\mathbb{F}_q$ of genus g, then*

$$(\sqrt{q}-1)^{2g} \leq \#J_C(K) \leq (\sqrt{q}+1)^{2g}.$$

3.2 Explicit group law

In order to be able to explicitly compute with elements of the Jacobian $J_C(K)$ we have to choose a way to represent these elements. The standard way of doing this is by using the so called *Mumford representation*, which we will explain in this section.

Definition 2. *A divisor D on a hyperelliptic curve C of genus g is called* semi-reduced *if it has the form*

$$D = \sum_{P \in C(L) \setminus \{\infty\}} c_P([P] - [\infty])$$

such that

(a) For $P = (x, y)$ with $2y + h(x) = 0$ one has $c_P \in \{0, 1\}$, and
(b) for $P = (x, y)$ and $P' = (x, -y - h(x))$ with $P \neq P'$ one has that either $c_P = 0$ or $c_{P'} = 0$ (or both).

If moreover $\sum c_P \leq g$ then D is called reduced.

The interest of reduced divisors lies in the following proposition.

Proposition 1. *Every element of $J_C(K)$ is represented by exactly one reduced divisor in $\mathrm{Div}^0_C(K)$.*

A convenient way of storing reduced divisors is via their Mumford representation.

Definition 3. *A divisor D in Mumford representation is a pair $[u(x), v(x)]$ of polynomials in $K[x]$ such that*

(a) $u(x)$ is monic,
(b) $u(x)$ divides $f(x) - h(x)v(x) - v(x)^2$,
(c) $\deg(v(x)) < \deg(u(x)) \leq g$.

The relation between reduced divisors and divisors in Mumford representation is the following. Consider the factorisation $u(x) = \prod_{i=1}^{d}(x - x_i)$ over the algebraic closure L of K. Then condition (b) above ensures that the points $(x_i, v(x_i))$ lie on the curve. The divisor $\sum_{i=1}^{d}([(x_i, v(x_i))] - [\infty])$ is a reduced divisor in $\mathrm{Div}^0_C(K)$. This yields a 1–1 correspondence between reduced divisors and divisors in Mumford representation.

If in Definition 3 one would not require that $\deg(u(x)) \leq g$ then a divisor in Mumford representation would correspond to a semi-reduced divisor.

Note that the zero element of $J_C(K)$ is represented by a reduced divisor consisting of the empty sum of points, i.e. $\sum_{P \in C(L) \setminus \{\infty\}} c_P[P]$ with all c_P equal to 0. The corresponding Mumford representation is $[1, 0]$.

3.3 Explicit group law

In this subsection we describe the group law on $J_C(K)$ explicitly in terms of the Mumford representation.

Negation.
Input: A divisor $D = [u, v]$.
Output: A divisor D' representing minus the class of D in $J_C(K)$.

1. Set $v' := -v - h$.
2. Output $D' := [u, v']$.

The most general algorithm for adding divisor classes is Cantor's algorithm. This algorithm was developed by Cantor [3] for the case that $h = 0$ and $2 \nmid q$. The general case was worked out by Koblitz in [27].

Addition (Cantor).
Input: Divisors $D_1 = [u_1, v_1]$ and $D_2 = [u_2, v_2]$.
Output: A divisor D representing the sum $D_1 + D_2$ in $J_C(K)$.

```
1. $d_1 = \gcd(u_1, u_2) = e_1u_1 + e_2u_2$
2. $d = \gcd(d_1, v_1 + v_2 + h) = c_1d_1 + c_2(v_1 + v_2 + h)$
3. $s_1 = c_1e_1$, $s_2 = c_1e_2$, $s_3 = c_2$
4. $u = (u_1u_2)/d^2$, $v = (s_1u_1v_2 + s_2u_2v_1 + s_3(v_1v_2 + f)/d \bmod u$
5. $u' = (f - vh - v^2)/u$, $v' = (-h - v) \bmod u'$
6. if $\deg(u') > g$ then $u = u'$, $v = v'$, goto 5
7. make $u'$ monic by dividing it by its leading coefficient.
8. Output $D = [u', v']$.
```

As presented here, the addition algorithm is not very efficient, and requires an extended Euclidean algorithm in steps 1 and 2. However if one fixes the genus g, one can work out specific algorithms dedicated to the various possible values of $\deg(u_1)$ and $\deg(u_2)$. This way one can formulate algorithms that are much more efficient, and that avoid high-level operations like Euclidean algorithms. Efficient formulae for curves of genus 2 are worked out in [28].

4 Security requirements

Suppose G is a Abelian group. Using additive notation, the Discrete Logarithm Problem (DLP) on G is the following problem: Given two elements R and aR of G, determine the integer a. Groups for which the DLP is a difficult mathematical problem and for which the group law can be performed efficiently are possible candidates for cryptographic applications.

The first type of group used for this purpose was the multiplicative group of the integers modulo a prime number, or more generally, the multiplicative group of a finite field. In this case, the DLP appears to be a difficult problem if the size of the finite field is big enough. However, there is an attack, known as *index* *calculus* that is fairly efficient. It has a running time that is *sub-exponential* in $\log q$. In practise, this means that one should take q of at least 1024 bits in order to have the DLP unfeasible to solve.

In the previous sections we have introduced two types of groups: the group of points on elliptic curves, and the group of points on the Jacobian of hyperelliptic curves. The main advantage of these groups, compared to the multiplicative groups of finite fields, is that under certain conditions, there is no method like index calculus known to solve the DLP. If the groups are chosen with care, then the most efficient way to solve the DLP is by means of Pollard's rho method [36]. For this method, one has to perform roughly $\sqrt{\#G}$ group operations. This means that its running time is exponential in $\log \#G$, and one can use smaller groups for achieving the same level of security. With today's computers it is reasonable to assume that it is unfeasible to perform 2^{80} operations in a reasonable amount of time. Under this assumption, it follows that cryptosystems based on elliptic or hyperelliptic curves are secure if $\#G \approx 2^{160}$. As a consequence, these systems are more efficient, and allow shorter key sizes than their multiplicative group-counterparts.

However, certain elliptic curves or hyperelliptic curves should be avoided since the DLP on them is relatively easy to solve. Here we list the curves that admit an attack that is faster than the Pollard rho method. The impact of these attacks varies from a slight security decrease to a completely insecure cryptosystem.

Index calculus for higher genus curves. There is an index calculus attack on $J_C(\mathbb{F}_q)$ that is more efficient than Pollard's rho method if the genus g of C is not small enough. Initially, this attack was developed for high genus curves by Adleman, Demarrais and Huang ([1]). In [11], Gaudry developed a version of index calculus that could beat Pollard's rho method if $g > 3$. More recent developments even indicate that for $g = 3$ index calculus attacks lead to a security decrease, see [46], [13] and [33]. So it is recommended to only use elliptic curves and hyperelliptic curves of genus 2.

Pohlig-Hellman . If the group order $\#G$ of G factors as $\prod r_i^{e_i}$ then it is possible to reduce the DLP in G to a DLP in subgroups of order r_i, see [35]. So if r is the largest prime divisor of $\#G$, then the DLP in G is as hard as the DLP in the subgroup of order r. For this reason it is important to choose G such that its order is almost prime, i.e. such that $\#G/r$ is small. A typical choice is to require that this quotient is ≤ 4. Ideally, one would like $\#G$ to be prime, but it is not always possible to achieve that. For example, an elliptic curve, or hyperelliptic curve of genus 2 over a field $\mathbb{F}_{2^\ell}$ that is not supersingular always yields a group of even order.

MOV and Frey-Rück Attacks. Let r be the largest prime divisor of the group G. Let k be the smallest positive integer such that $r|q^k - 1$. Then there is a computable injective group homomorphism from the order-r subgroup of G

to $\mathbb{F}_{q^k}^*$, see [9] and [31]. If k is too small, then one can solve the DLP in G by first mapping it to $\mathbb{F}_{q^k}^*$, and use index calculus there. So one should avoid groups for which k is small. Typically, $k > 20$ is a safe choice. A random curve is very unlikely to yield a group for which k is small. There is however a special class of curves, *supersingular* curves, for which k is always small.

Anomalous curves. If the largest prime divisor r of $\#G$ is equal to the characteristic of $\mathbb{F}_q$, then one can transform the DLP to a DLP in the additive group of $\mathbb{F}_q$, where it is trivial to solve. See [42] and [38]. So this case should be avoided.

Weil restriction and cover attacks. Let C be an elliptic curve or hyperelliptic curve of genus g defined over an extension field $\mathbb{F}_{q^e}$. Let G be the group $C(\mathbb{F}_q)$ if C is elliptic, or $J_C(\mathbb{F}_q)$ if C is hyperelliptic. Then sometimes it is possible to find a curve X defined over $\mathbb{F}_q$ such that there is a homomorphism from G to $J_X(\mathbb{F}_q)$ that transfers the DLP from G to $J_X(\mathbb{F}_q)$. If the genus of such an X is not much bigger than eg, then index calculus methods on X enables one to compute the DLP faster than with Pollard's rho method. The original idea behind this construction goes back to Frey, and it has been applied successfully to attack several curves, for the first time in [10]. Although one does not have a precise criterion to determine which curves are subject to this attack, the number of curves that are weakened is still growing. At the moment it seems wise to avoid all curves over small extension degrees, and to only use curves defined over $\mathbb{F}_{2^\ell}$ with ℓ prime, or over prime fields $\mathbb{F}_p$.

Good Curves. To summarise which curves are safe use, we list the required properties. Let q be either a prime, or $q = 2^\ell$, with ℓ prime. Let C be an elliptic curve or a hyperelliptic curve of genus 2 over $\mathbb{F}_q$. Let G be the associated group, i.e. $G = C(\mathbb{F}_q)$ or $G = J_C(\mathbb{F}_q)$. Let r be the largest prime divisor of $\#G$. Then we require

- r does not divide q,
- If $k > 0$ id the smallest integer such that $r | q^k - 1$ then $k > 20$.
- $r > 2^{160}$.
- $\#G/r \leq 4$.

In order to check these requirements, one needs to be able to determine $\#G$. For the case that C is elliptic this can always be done. The first point counting algorithm that runs in polynomial time in $\log(q)$ was found by Schoof [40] in 1985. It was later improved by Elkies and Atkins [41]. The resulting algorithm is called the SEA algorithm. It has a running time of $\mathcal{O}((\log q)^{4+\epsilon})$. In 1999, Satoh [39] found a faster method that only works for fields of small characteristic. Improvements by various people finally led to an algorithm of Harley [16] that runs in time $\mathcal{O}(\ell^{2+\epsilon})$ for $q = p^\ell$ and fixed p.

For groups $G = J_C(\mathbb{F}_q)$ associated to genus 2 curves there exists a generalisation of the SEA algorithm to count the number of elements, optimised by

Gaudry and Schost [12]. Although this algorithm is able to compute the number of points for cryptographically relevant field sizes, it is still quite slow. In order to quickly obtain a safe genus 2 curve over $\mathbb{F}_q$ in the case that q is prime, one should use the *Complex Multiplication method.* This method cannot determine the group size for any given curve, but it constructs special types of curves for which the group size is known in advance.

When $\mathbb{F}_q$ has small characteristic, there are fast p-adic methods to determine $\#G$ for arbitrary hyperelliptic curves. The first such algorithm was found by Kedlaya [24] in 2000. Since then, various improvements and other algorithms were found by various people, see [25] for an overview.

For the employment of a cryptographic system, one only has to find one good curve. So the point counting only needs to be done in the initial set-up of the system. Once one has a safe curve, it can be used as long as no attack is known on that specific curve. The curve, and the size of the group are not part of the keys of the cryptosystem, so even if a number of secret keys are revealed, it does not jeopardise other secret keys. Therefore, one can use standard curves, for which the cardinality of G has already been determined.

5 Schemes

We will describe three types of schemes can are based on elliptic and hyperelliptic curves. These are signature schemes, encryption schemes and key agreement schemes.

5.1 Key Agreement

The Diffie-Hellman key agreement scheme was the first example of public key cryptography. Originally, it was formulated for the multiplicative group of numbers modulo a prime, but it can easily be adjusted to general groups.

Let G be a group whose elements can be represented in an efficient way, and in which the group operations can be evaluated efficiently as well. Suppose that the discrete logarithm problem is a hard problem for the group G. The groups we have in mind for this paper are of course the group of points on elliptic curves and Jacobians of hyperelliptic curves that satisfy the security requirements that were discussed in the previous section.

Diffie-Hellman Key Agreement Scheme. Two parties Alice and Bob wish to agree on a common secret by communicating over a public channel. An eavesdropper Eve, who can listen to all communication between Alice and Bob, should not be able to derive this common secret.

First, we assume that there are the following publicly known system parameters:

- The group G.
- An element $R \in G$ of large prime order r.

The steps that Alice performs are the following:

1. Choose a random integer $a \in [1, r-1]$.
2. Compute $P = aR$ in the group G, and send it to Bob.
3. Receive the element $Q \in G$ from Bob.
4. Compute $S = aQ$ as common secret.

The steps that Bob performs are:

1. Choose a random integer $b \in [1, r-1]$.
2. Compute $Q = bR$ in the group G, and send it to Alice.
3. Receive the element $P \in G$ from Alice.
4. Compute $S = bP$ as common secret.

Note that both Alice and Bob have computed the same values S, as

$$S = a(bR) = (ab)R = b(aR).$$

It is not known how Eve, knowing only P, Q and R, can compute S within reasonable time. If she could solve the discrete logarithm problem in G, then she could compute a from P and R, and then compute $S = aQ$. The problem of computing S from P, Q and R is known as the *Diffie-Hellman problem*.

The pair (a, P) is called Alice's *key pair* consisting of her *private key* a and *public key* P. Likewise, Bob's key pair is (b, Q), with private key b and public key Q.

It is important to realise that the scheme that is described here should be used with additional forms of authentication of the public keys. Otherwise an eavesdropper Eve who is able to intercept and change information that is sent is able to agree on keys separately with Alice and Bob. This is known as a *man in the middle attack*. Although we will not go into details, we just mention one method of adding this additional authentication. In the MQV protocol, by Law, Menezes, Qu, Vanstone and Solinas, [29], Alice and Bob both have long term key pairs, of which the public keys are assumed to be authenticated. During the key agreement, they create ephemeral key pairs, and use both the long term and ephemeral keys to deduce the common secret. A 'man' in the middle can only intercept and change the ephemeral keys. Although this will hinder communication between Alice and Bob, she will not be able to let Alice and Bob believe that they share a common secret if they only both share a secret with Eve.

5.2 Encryption

The ElGamal Encryption Scheme. The first encryption scheme that was based on the discrete logarithm problem was the ElGamal encryption scheme. We will describe it here, again for a general group G.

Suppose that one has the following publicly known system parameters:

- The group G.

- An element $R \in G$ of large prime order.

Suppose Bob has a private key $b \in [1, r-1]$ and public key $Q = bR$. Alice wants to send Bob a message M, which we assume to be encoded as an element of the group G. She wants to encrypt M using Bob's public key Q, such that only Bob can decrypt the message again, using his secret key b.

To encrypt M, Alice does the following:

1. Obtain Bob's public key Q.
2. Choose a secret number $a \in [1, r-1]$.
3. Compute $C_1 = aR$.
4. Compute $C_2 = M + aQ$.
5. Send (C_1, C_2) to Bob.

Bob can decrypt the encrypted message by doing the following:

1. Obtain the encrypted message (C_1, C_2) from Alice.
2. Compute $M = C_2 - bC_1$.

Note that the first part of ElGamal can be considered as a Diffie-Hellman key agreement scheme using Bob's key and an ephemeral key created by Alice. In step 4, the message M is encrypted by adding the common secret aQ derived from this Diffie-Hellman scheme. Instead of encrypting M by adding aQ, one could also use use a symmetric encryption scheme, using a key derived from aQ. This idea lies at the basis of the next encryption scheme we describe.

The (Hyper-)Elliptic Curve Integrated Encryption Scheme. This encryption scheme uses the Diffie-Hellman scheme to derive a secret key, and combines it with tools from symmetric key cryptography to provide better provable security. It can be proved to be secure against adaptive chosen ciphertext attacks.

We again formulate the scheme for any group G and $R \in G$ with large prime order r. The symmetric tools that are used in the scheme are:

- A key derivation function. This is a function $\mathrm{KD}(P)$ that takes as input a key P, in our case this is an element of G, and outputs keying data of any required length.
- A symmetric encryption scheme consisting of a function Enc_k that encrypts the message M to a ciphertext $C = \mathrm{Enc}_k(M)$ using a key k, and a function Dec_k that decrypts C to the message $M = \mathrm{Dec}_k(C)$.
- A Message Authentication Code MAC_k. One can think of this as a keyed hash function. It is a function that takes as input a ciphertext C and a key k. It computes a string $\mathrm{MAC}_k(C)$ that satisfies the following property: Given a number of pairs $(C_i, \mathrm{MAC}_k(C_i))$, it is computationally infeasible to determine a pair $(C, \mathrm{MAC}_k(C))$, with C different from the C_i if one does not know k.

1. Obtain Bob's public key Q.
2. Choose a secret number $a \in [1, r-1]$.

3. Compute $C_1 = aR$.
4. Compute $C_2 = aQ$.
5. Compute two keys k_1 and k_2 from $\mathrm{KD}(C_2)$, i.e. $(k_1||k_2) = \mathrm{KD}(C_2)$.
6. Encrypt the message, $C = \mathrm{Enc}_{k_1}(M)$.
7. Compute $mac = \mathrm{MAC}_{K_2}(C)$.
8. Send (C_1, C, mac).

To decrypt, Bob does the following:

1. Obtain the encrypted message (C_1, C, mac) from Alice.
2. Compute $C_2 = bC_1$.
3. Compute the keys k_1 and k_2 from $\mathrm{KD}(C_2)$.
4. Check whether mac equals $\mathrm{MAC}_{k_2}(C)$. If not, reject the message and stop.
5. Decrypt the message $M = \mathrm{Dec}_{k_1}(C)$.

5.3 Signatures

The Digital Signature Algorithm (DSA) is the basis of the digital NIST signature standard, described in [8]. This algorithm can be adapted for elliptic and hyperelliptic curves. More generally, one can use it for any group G where the DLP is difficult, provided that one has a computable map $G \to \mathbb{Z}$ with large enough image, and few inverses for each element in the image. The elliptic curve version, known as ECDSA, can be found in various standards. The hyperelliptic curve version seems not to have appeared a lot in existing literature.

We describe the scheme for a general group with a map $\phi : G \to \mathbb{Z}$ as described above. In the elliptic curve case, one takes for ϕ the map that associates to a point (x, y) the integer whose binary expansion is the bit string representing the x-coordinate x. If we are working with an elliptic curve E over a prime field $\mathbb{F}_p$, this is just the integer $\phi(x) \in [0, p-1]$ that reduces to x modulo p. If E is defined over $\mathbb{F}_{2^\ell}$, and x is represented by a polynomial $\sum_{i=0}^{\ell-1} c_i X^i \in \mathbb{F}_2[X]$, then $\phi(x) = \sum_{i=0}^{\ell-1} \tilde{c}_i 2^i$, where $\tilde{c}_i \in \{0,1\} \subset \mathbb{Z}$ such that $\tilde{c}_i \equiv c_i \bmod 2$.

In the hyperelliptic curve case, one can take for ϕ the following map. Let $D = [u(x), v(x)]$ be a divisor in Mumford representation. Let $u(x) = \sum_{i=0}^{\deg(u(x))} u_i x^i$ with $u_i \in \mathbb{F}_q$. Define $\phi(D)$ to be the integer whose binary expansion is the concatenation of the bit strings representing the u_i, $i \in [0, \deg(u(x)) - 1]$, as explained above.

Assume the following system parameters are publicly known:

- A group G and a map $\phi \to \mathbb{Z}$ as above,
- an element $R \in G$ with large prime order r,
- a hash function H that maps messages m to 160-bit integers.

To create a key pair, Alice chooses a secret integer $a \in \mathbb{Z}$, and computes $P = aR$. The number a is Alice's secret key, and P is her public key.

If Alice wants to sign a message m, she has to do the following:

1. Choose a random integer $k \in [1, r-1]$, and compute $Q = kR$.

2. Compute $s \equiv k^{-1}(H(m) + a\phi(Q)) \bmod r$.

The signature is

$$(m, Q, s).$$

To verify this signature, a verifier Bob has to do the following:

1. Compute $v_1 \equiv s^{-1}H(m) \bmod r$ and $v_2 \equiv s^{-1}\phi(Q) \bmod r$.
2. Compute $V = v_1 R + v_2 P$.
3. Accept the signature if $V = Q$. Otherwise, reject it.

6 Description of the NTRU Encryption Scheme

The NTRU cryptosystem was introduced at the rump session of Crypto'96 [22] and was later published in the proceedings of the ANTS-III conference [17]. NTRU is a ring based public key cryptosystem and is therefore quite different from the group based cryptosystems whose security relies on the integer factorisation problem or the discrete logarithm problem. This extra structure can be exploited to obtain a very fast cryptosystem: to encrypt/decrypt a message block of length N, NTRU only requires $O(N^2)$ time, whereas the group based schemes require $O(N^3)$ time. Furthermore, NTRU also has a very short key size of $O(N)$ and very low memory requirements, which makes it ideal for constrained devices such as smart cards.

6.1 Definitions and Notation

Denote by $\mathbb{Z}$ the ring of integers and by $\mathcal{P}$ the quotient ring of polynomials $\mathbb{Z}[X]/(X^N - 1)$. The ring $\mathcal{P}$ can be identified in a natural way with the set $\mathbb{Z}^N$ by

$$e = (e_0, e_1, \ldots, e_{N-1}) = \sum_{i=0}^{N-1} e_i X^i \,. \tag{2}$$

Addition of two elements $f, g \in \mathcal{P}$ is defined as pairwise addition of coefficients of the same degree and multiplication is defined by the cyclic convolution product, denoted by $\star$. Let $h = f \star g$, then the k-th coefficient h_k of h is given by

$$h_k = \sum_{i+j \equiv k \bmod N} f_i \cdot g_j \quad (0 \leq k < N) \,. \tag{3}$$

The NTRU cryptosystem works with two relatively prime moduli $p, q \in \mathcal{P}$. The modulus q is typically chosen to be the 2^l with $l = \lfloor \log_2(N) \rfloor$ and p is either 3 or the polynomial $2 + X$. In the remainder of this section, we will assume that $p = 2 + X$, since $p = 3$ is no longer recommended by the NTRU standard [4].

Denote by $\mathbb{Z}_q$ the ring of integers modulo q, which will be represented by the symmetric interval $[-q/2, q/2)$. Let $\mathcal{P}_q = \mathbb{Z}_q[X]/(X^N - 1)$ be the ring of polynomials obtained from $\mathcal{P}$ by reduction modulo q and $\pi_q : \mathcal{P} \to \mathcal{P}_q$ the corresponding homomorphism. Note that if $f, g \in \mathcal{P}_q$ have small coefficients in absolute value, the product $f \star g$ will also have fairly small coefficients. An element $f \in \mathcal{P}$ is called invertible modulo q, if $f_q = \pi_q(f)$ is invertible in $\mathcal{P}_q$, i.e. there exists a polynomial $g_q \in \mathcal{P}_q$ such that $f_q \star g_q = 1$ in $\mathcal{P}_q$. The inverse polynomial g_q will be denoted by f_q^{-1}.

In a similar way, we can define the ring $\mathcal{P}_p = \mathbb{Z}[X]/(X^N - 1, p)$, obtained from $\mathcal{P}$ by reduction modulo p and denote with $\pi_p : \mathcal{P} \to \mathcal{P}_p$ the corresponding homomorphism. For $p = 2 + X$, this ring can be identified with $\mathbb{Z}_2[X]/(X^N - 1)$, by replacing every multiple of 2 by $-X$. To illustrate this procedure, consider the following example:

$$\begin{aligned} X^4 + 3X + 4 &\equiv X^4 + (-X+1)X + 2(-X) \pmod{2+X} \\ &\equiv X^4 - X^2 - X \pmod{2+X} \\ &\equiv X^4 + (1-2)X^2 + (1-2)X \pmod{2+X} \\ &\equiv X^4 + X^3 + 2X^2 + X \pmod{2+X} \\ &\equiv X^4 + X^3 + (-X)X^2 + X \pmod{2+X} \\ &\equiv X^4 + X \pmod{2+X} \end{aligned}$$

6.2 The NTRU Primitive

The NTRU encryption primitive uses the following parameter set $\mathcal{S}$:

- a prime $N \in \mathbb{N}$.
- a modulus $q \in \mathbb{N}$; typically $q = 2^l$ with $l = \lfloor \log_2(N) \rfloor$.
- an element $p \in \mathbb{Z}[X]$ of degree at most one, with small coefficients and invertible modulo q; typically $p = 2 + X$.
- three integers $d_f, d_g, d_r \in \mathbb{N}$. The three integers d_f, d_g, d_r determine three sets of polynomials $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r$. Let $\mathcal{B}_k$ denote the set of binary polynomials in $\mathcal{P}$ with exactly k ones, then the most popular construction is as follows:

$$\mathcal{L}_f = \{1 + p \star f_1 : f_1 \in \mathcal{B}_{d_f}\}, \quad \mathcal{L}_g = \mathcal{B}_{d_g}, \quad \mathcal{L}_r = \mathcal{B}_{d_r}\,. \tag{4}$$

Recommended parameter sets can be found in the NTRU standard [4].

As all public key encryption schemes, the basic NTRU encryption primitive consists of three algorithms: key generation, encryption and decryption.

Key Generation The generation of a key pair proceeds as follows:

1. Generate random polynomials $f \in \mathcal{L}_f$ and $g \in \mathcal{L}_g$, such that f is invertible modulo q and modulo p.
2. Compute $f_q^{-1} \in \mathcal{P}_q$ and $f_p^{-1} \in \mathcal{P}_p$.
3. Compute the polynomial $h \in \mathcal{P}_q$ as

$$h \equiv p \star g \star f_q^{-1} \bmod q\,. \tag{5}$$

4. The *public key* consists of the set $\mathcal{S}$ and the polynomial h.
5. The *private key* consists of the set $\mathcal{S}$ and the polynomials h, f and f_p^{-1}.

Note that for the choice of f given in Equation (4), we have $f_p^{-1} = 1$, so we do not need to compute this value.

Encryption The encryption function of the basic NTRU primitive is probabilistic in that encrypting the same message twice will result in different ciphertexts. The message space of NTRU is the ring $\mathcal{P}_p$, so we assume that the plaintext m is given as an element of $\mathcal{P}_p$. Encryption then consists of the following steps:

1. Generate random polynomial $r \in \mathcal{L}_r$.
2. Compute the ciphertext $e = h \star r + m \bmod q$ as an element of $\mathcal{P}_q$.

Decryption Given a ciphertext $e \in \mathcal{P}_q$, the private key f and its inverse f_p^{-1} modulo p, decryption proceeds as follows:

1. In the first step we need to recover $p \star r \star g + m \star f$ as an element of $\mathcal{P}$ and not just as an element in $\mathcal{P}_q$. To this end we compute
$$\begin{aligned} a &\equiv e \star f \pmod q \\ &\equiv r \star p \star f_q^{-1} \star g \star f + m \star f \pmod q \\ &\equiv p \star r \star g + m \star f \pmod q\,. \end{aligned} \tag{6}$$
2. Assuming that $a = p \star r \star g + m \star f$ in the ring $\mathcal{P}$, we can recover m by working modulo p as
$$a \star f_p^{-1} \equiv m \star f \star f_p^{-1} \equiv m \pmod p\,. \tag{7}$$

For the chosen parameter set, we have $f_p^{-1} = 1$ and thus this simplifies to $a \equiv m \pmod p$.

Note that the plaintext space is $\mathcal{P}_p$, whereas the ciphertext space is $\mathcal{P}_q$; encrypting a message block of $l \log_2 p$ bits, thus results in a ciphertext of $l \log_2 q$ bits. This phenomenon is called message expansion and for the NTRU primitive the expansion factor simply is $\log_p q$, which is around 7 or 8 for typical NTRU parameters.

The above description is the textbook version of the NTRU primitive and as it stands should not be used in practise, since like textbook RSA [37] or ElGamal [7] it is insecure. To turn this primitive into a provably secure scheme, a padding scheme like `NAEP` [20] should be used.

6.3 Analysis of the Decryption Step

The correctness of the NTRU decryption relies on the fact that a is equal to $p \star r \star g + m \star f$ as an element of $\mathcal{P}$ and not just as an element of $\mathcal{P}_q$. Suppose this is not the case, then there exists a non-zero $\epsilon \in \mathcal{P}$ such that

$$a = (p \star r \star g + m \star f) + q \cdot \epsilon\,. \tag{8}$$

Since p and q are relatively prime, the error term $q \cdot \epsilon$ will be non-zero modulo p with very high probability. So instead of recovering m, we would recover

$$m + q \cdot \epsilon \star f_p^{-1} \pmod p\,. \tag{9}$$

Furthermore, note that the sender cannot test if the decryption will fail or not, since the private key f is required.

To devise a criterion for correct decryption we introduce the following norms on $\mathcal{P}$. For $f \in \mathcal{P}$, define the width $|f|_\infty$ of f as

$$|f|_\infty = \max_{0 \leq i < N} f_i - \min_{0 \leq i < N} f_i \,.$$

Note that this can be interpreted as some sort of L^∞ norm on $\mathcal{P}$. Similarly, we introduce the centred L^2 norm on $\mathcal{P}$ by

$$|f|_2 = \left(\sum_{i=0}^{N-1} (f_i - \bar{f})^2 \right)^{1/2} \,, \quad \text{with } \bar{f} = \frac{1}{N} \sum_{i=0}^{N-1} f_i \,.$$

Note that the centred L^2 norm $| \cdot |_2$ is related to the standard L^2 norm $|| \cdot ||$ by the following relation: $|f|_2^2 = ||f||^2 - N\bar{f}^2$.

The following proposition due to Don Coppersmith, indicates that decryption will succeed with very high probability.

Proposition 1 *For any $\epsilon > 0$ the are constants $\gamma_1, \gamma_2 > 0$ depending on ϵ and N, such that for randomly chosen polynomials $f, g \in \mathcal{P}$, the probability is greater than $1 - \epsilon$ that they satisfy*

$$\gamma_1 \, |f|_2 \, |g|_2 \leq |f \star g|_\infty \leq \gamma_2 \, |f|_2 \, |g|_2 .$$

In order to recover the correct message m after decryption, it is necessary that $|p \star r \star g + m \star f|_\infty \leq q$. This turns out to be almost always true if one takes

$$|p \star r \star g|_\infty \leq q/4 \quad \text{and} \quad |f \star m|_\infty \leq q/4 \,. \tag{10}$$

Following Proposition 1, the authors of NTRU suggest to take

$$|f|_2 |m|_2 \simeq \frac{q}{4\gamma_2} \quad \text{and} \quad |r|_2 |g|_2 \simeq \frac{q}{4\gamma_2 |p|_2} \,,$$

for a γ_2 corresponding to a small enough value for ϵ.

Let $b = p \star r \star g + m \star f$ be an element of $\mathcal{P}$, i.e. the coefficients are not reduced modulo q, then we know that decryption fails exactly if and only if $b \neq a$ with $a \equiv p \star r \star g + m \star f \pmod{q}$. Silverman [45] defines two types of decryption failures:

- Gap failure occurs if $|b|_\infty \geq q$.
- Wrap failure occurs if $\min_{0 \leq i < N} b_i \leq -q/2$ or $\max_{0 \leq i < N} b_i > q/2$.

Clearly, if a gap failure occurs, we can never recover the correct value of m. However, if only a wrap failure occurs, the correct value of b can be determined by changing the range into which the coefficients are reduced to $[A, A + q)$, for some value $A \neq -q/2$. To calculate A, note that $(f \star g)(1) = f(1) \cdot g(1)$, with $f(1)$ and $g(1)$ the sum of the coefficients of f and g respectively. Note that the average value of a coefficient of $b = p \star r \star g + m \star f$ is given by $(p(1) \cdot r(1) \cdot g(1) + m(1) \cdot f(1))/N$. Since the decryptor knows $h(1)$ from the public key and $r(1)$ from the definition of $\mathcal{L}_r$, he can compute $I \equiv m(1) \equiv e(1) - r(1) \cdot h(1) \bmod q$. Assuming $m(1)$ lies in the range $[N/2 - q/2, N/2 + q/2]$, we can calculate the average value of a coefficient of b and take

$$A = \left\lfloor \frac{p(1) \cdot r(1) \cdot g(1) + f(1) \cdot I}{N} \right\rceil - \frac{q}{2}\,. \tag{11}$$

Having computed A, the coefficients of a are reduced in the interval $[A, A+q)$. If decryption still fails, then one can try to decrypt using the values $A \pm 1, A \pm 2, \ldots$ until successful. Since wrap failures are more common than gap failures, using the range $[A, A+q)$ is a partial solution to decryption failures. However, if the value of A given in Equation (11) leads to a decryption failure, the wrap failure could still be detected using a timing analysis. The existence of validly created ciphertexts, which cause a decryption failure is a feature unique to NTRU since in a classical public key cryptosystem, the decryption of a validly created ciphertext never fails. This property will turn out to be crucial in the security analysis of NTRU.

7 Security Analysis

In this section, we will highlight two types of attacks on the NTRU cryptosystem: lattice attacks and chosen ciphertext attacks. An exhaustive list of papers analysing the security of NTRU can be found in [34].

7.1 Lattice attacks

Shortly after NTRU was introduced to the cryptographic community, Coppersmith and Shamir [5] devised a lattice attack that recovers the private key for small N. Recall that the private keys f, g and the public key are related by

$$f \star h \equiv p \star g \pmod{q} \tag{12}$$

and that f and g have very small L^2 norm.

Consider the following subset of pairs of polynomials

$$L = \{(u, v) \in \mathcal{P} \times \mathcal{P} \mid u \star H \equiv v \pmod{q}\}\,,$$

with $H = p^{-1} \star h \pmod{q}$, then it is clear that $(f, g) \in L$. After identification of $\mathcal{P}$ with $\mathbb{Z}^N$, we obtain a subset of pairs of vectors in $\mathbb{Z}^N \times \mathbb{Z}^N$. Clearly, if $(u_1, v_1) \in L$ and $(u_2, v_2) \in L$, then any $\mathbb{Z}$-linear combination of these pairs of vectors will also be in L. This shows that L is in fact a subgroup of $\mathbb{Z}^{2N}$ and thus, by definition, a lattice in $\mathbb{R}^{2N}$.

Any lattice admits a basis, i.e. a set B of d linearly independent vectors such that every element of the lattice L can be expressed as a $\mathbb{Z}$-linear combination of the elements in B. The integer d is called the dimension of the lattice and does not depend on the chosen basis. Let U be an integral $d \times d$ matrix with determinant ± 1, then UB also is a basis for L and all bases can be obtained in this way. The volume $v(L)$ of a lattice L is the absolute value of the determinant of any lattice basis. Since all bases are related by a unimodular transformation, the volume of a lattice does not depend on the basis chosen.

Let I_N denote the $N \times N$ identity matrix and let M_H denote the circulant matrix whose columns are circularly shifted versions of H, then it is not difficult to see that the columns of the matrix

$$M = \begin{pmatrix} I_N & 0 \\ M_H & qI_N \end{pmatrix} \tag{13}$$

are a basis of the lattice L. To see this, let $(u, v) \in L$, then there exists a vector $w \in \mathbb{Z}^N$ such that $u \star H = v + qw$. Multiplying the matrix M by the transpose of $(u, -w)$ finally results in (u, v) and also proves the claim.

Since the vector (f, g) is in the lattice L, we conclude that (f, g) is an integer linear combination of the columns of M. Furthermore, since the polynomials f and g have very small L^2 norm, the vector (f, g) is much shorter than a random vector in the lattice L and is in fact likely to be the shortest vector in L.

If an attacker could solve the shortest vector problem (SVP), i.e. given a basis of a lattice L, find a non-zero vector z such that $||z||$ is as small as possible, then the attacker would be able to recover the secret key (f, g). However, Ajtai [2] proved that the SVP is NP-hard under randomised reductions, which implies that an efficient algorithm to solve the SVP is very unlikely.

On the other hand, the LLL-algorithm [30] runs in polynomial time in the dimension d of the lattice, but is only guaranteed to return a vector b with $||b|| \leq 2^{(d-1)/4} v(L)^{1/d}$, whereas the shortest vector z satisfies $||z|| \leq \sqrt{d} v(L)^{1/d}$, i.e. LLL can approximate the shortest vector up to some exponential factor in d. However, it should be stressed that in practise the LLL-algorithm performs much better than this theoretical bound. Experiments [18] show that $N \simeq 120$ can be broken using lattice based techniques and Table 1 contains the extrapolation of these experiments:

Table 1. Bit security of NTRU for various N.

N	Bit security
167	57
251	88
500	178
1000	360

The attack as described above is in fact a simplified version: in practise one would apply the LLL-algorithm to the lattice L_λ with basis

$$\begin{pmatrix} \lambda I_N & 0 \\ M_H & qI_N \end{pmatrix} .$$

Note that (f, g) is no longer in L_λ, but $(\lambda f, g)$ is. The parameter λ is a balancing constant used to optimise the performance of the LLL algorithm and is normally chosen to make λf and g have the same length, i.e. $\lambda = |g|_2/|f|_2$.

7.2 Chosen ciphertext attacks

In a chosen ciphertext attack, the adversary has access to a decryption oracle and can choose many ciphertexts to be decrypted with an unknown key. A chosen ciphertext attack can be either non-adaptive (CCA1) or adaptive (CCA2); during the latter attack, the adversary uses the previous results to select subsequent ciphertexts. Most textbook descriptions of cryptosystems are not secure against chosen ciphertext attacks and the same holds for NTRU. The standard defence against CCA2 is the use of an appropriate padding scheme which prevents the attacker from constructing valid ciphertexts without knowing the corresponding plaintext.

However, the existence of decryption failures of validly created ciphertexts distinguishes NTRU from a classical public key system, since this assumes that decryption of a validly created ciphertext never fails. An immediate consequence of this feature is that there exists two types of chosen ciphertexts on NTRU: the first type uses decryptions of invalid ciphertexts, whereas the second relies on valid ciphertexts that cause a decryption failure. Furthermore, in the latter attack it suffices to know whether the ciphertext caused a decryption failure or not; the attacker does not require to see the decrypted plaintext. This type of attack is also known as a decipherable ciphertext attack. In this section, we will give an example of both types of attack.

CCA based on invalid ciphertexts In this section, we present a very simple attack on a non-padded version of NTRU using the parameter set described in Section 6.2.

Given a decryption oracle, simply feed in the ciphertext $e \equiv p_q^{-1} \star h \pmod{q}$. Since $h \equiv p \star g \star f_q^{-1} \pmod{q}$, the decryption oracle first computes

$$a \equiv e \star f \equiv (p_q^{-1} \star h) \star f \equiv g \pmod{q} .$$

Since for the chosen parameter set, g is a binary polynomial a will simply be g. Furthermore, the reduction modulo p is the identity since the polynomial g is already a binary polynomial. As a result, the decryption oracle simply returns g.

A somewhat more elaborate version of this attack was pointed out to the authors by J. Silverman. In this attack the oracle cannot recognise the decrypted text as being special.

Choose a completely random m and ask the oracle to decrypt $e \equiv p_q^{-1} \star h + m \pmod{q}$, which returns $g + f \star m \pmod{p}$. Then repeat this for $e \equiv p_q^{-1} \star h - m$ to obtain $g - f \star m \pmod{p}$. Adding both results and dividing by 2 modulo p also gives g.

More complicated chosen ciphertext attacks are given by Silverman [44] and Jaulmes and Joux [23].

CCA based on decryption failures The attack described in this section is based on the work of Howgrave-Graham, Nguyen, Pointcheval, Proos, Silverman, Singer and Whyte [19].

Recall that a non-recoverable decryption failure occurs when the width of the polynomial

$$b = p \star r \star g + m \star f \in \mathcal{P}$$

is greater than q, i.e. $|b|_\infty \geq q$. Note however that the polynomials r, g, m, f are polynomials of very small width and by Proposition 1 it is very unlikely that either $r \star g$ or $m \star f$ has width greater than $q/2$. It is therefore quite natural to ask the question if r is somehow related to g and m to f.

Given a polynomial $c(X) \in \mathcal{P}$, we can consider the reversal

$$\bar{c}(X) = c(X^{-1}) \in \mathcal{P}\,,$$

i.e. if $c(X) = \sum_{i=0}^{N} c_i X^i$, then $\bar{c}(X) = \sum_{i=0}^{N} c_{N-i} X^i$, where we define $c_N = c_0$. The autocorrelation polynomial $\hat{c}$ is then defined as $\hat{c}(X) = c(X) \star \bar{c}(X)$ and by definition we have

$$\hat{c}_k = \sum_{i=0}^{N-1} c_i \, c_{(i+k) \bmod N}\,.$$

One of the most important properties of the autocorrelation polynomial is that its constant coefficient $\hat{c}_0$ is maximal. Indeed, by definition we have

$$\hat{c}_0 = \sum_{i=0}^{N-1} c_i^2 = ||c||^2$$

whereas the other coefficients are only about $||c||$ in size. As a consequence, the autocorrelation polynomial $\hat{c}$ if of exceptionally great width compared to the polynomial c itself.

Applying this to the polynomial b, we can assume that r is correlated significantly to $\bar{g}$ and m is correlated significantly to $\bar{f}$, which means that r is almost equal to $X^i \star \bar{g}$ for some i and m is almost equal to $X^j \star \bar{f}$ for some j.

The chosen ciphertext attack then proceeds as follows: the attacker encrypts random messages m with random nonce r until (m, r) causes a decryption failure. Note that the attacker only needs to know whether or not the decryption failed. For each such tuple (m, r), the attacker can assume that r is almost equal to $X^i \star \bar{g}$ for some i and m is almost equal to $X^j \star \bar{f}$ for some j. Unfortunately, the attacker does not know the integer i nor j, but this does not pose a real problem since

$$\widehat{X^i \star \overline{g}} = \hat{g} \qquad \text{and} \qquad \widehat{X^i \star \overline{f}} = \hat{f}\,.$$

In conclusion, the attacker computes the average of many tuples $(\hat{m}, \hat{r})$ to obtain $(\hat{g}, \hat{f})$. Given $\hat{f}$ and $\hat{g}$, the polynomials f and g can be easily recovered using an algorithm due to Gentry and Szydlo [15].

Clearly, the above attack only works when decryption failures are sufficiently frequent. Therefore, the parameter set $\mathcal{S}$ should be chosen such that the probability of decryption failures is (preferably) smaller than 2^{-80}. Unfortunately, since this probability is so small, it is also very hard to determine for a given parameter set.

Remark Two other attacks worth mentioning are: a meet-in-the-middle attack on the NTRU private key [21], originally due to Odlyzko, which has a complexity roughly $\frac{1}{\sqrt{N}}\binom{N/2}{d_f/2}$ and an attack by Gentry [14], which works for composite N by reducing low-dimensional lattices to recover a folded version of the private key.

References

1. L. M. Adleman, J. DeMarrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *Algorithmic number theory — ANTS I*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 28–40. Springer, Berlin, 1994.
2. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *STOC*, pages 10–19, 1998.
3. D. G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177):95–101, 1987.
4. Consortium for Efficient Embedded Security. *Efficient Embedded Security Standards #1: Implementation aspects of NTRUEncrypt and NTRUSign*, version 2.0 edition, 2003.
5. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Advances in cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 52–61. Springer, Berlin, 1997.
6. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, IT-22(6):644–654, 1976.
7. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31(4):469–472, 1985.
8. FIPS 186-2. Digital Signature Standard. Federal Information Processing Standards Publication 186-2, February 2000.
9. 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.
10. P. Gaudry, F. Hess, and N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1):19–46, 2002.

11. P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In *Advances in cryptology—EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Comput. Sci.*, pages 19–34. Springer, Berlin, 2000.
12. P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 239–256. Springer, 2004.
13. P. Gaudry and E. Thomé. A double large prime variation for small genus hyperelliptic index calculus. Cryptology ePrint Archive, Report 2004/153, 2004. Available at http://eprint.iacr.org/.
14. C. Gentry. Key recovery and message attacks on NTRU-composite. In *Advances in cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Comput. Sci.*, pages 182–194. Springer, Berlin, 2001.
15. C. Gentry and M. Szydlo. Cryptanalysis of the revised NTRU signature scheme. In *Advances in cryptology—EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Comput. Sci.*, pages 299–320. Springer, Berlin, 2002.
16. R. Harley. Asymptotically optimal p-adic point-counting. e-mail to NMBRTHRY list, December 2002.
17. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring-based public key cryptosystem. In *Algorithmic number theory – ANTS III*, volume 1423 of *Lecture Notes in Comput. Sci.*, pages 267–288. Springer, Berlin, 1998.
18. J. Hoffstein, J. H. Silverman, and W. Whyte. Estimating breaking times for ntru lattices. Technical Report #012, Version 2, 2003.
19. N. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, and W. Whyte. The impact of decryption failures on the security of NTRU encryption. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 226–246. Springer, 2003.
20. N. Howgrave-Graham, J. H. Silverman, A. Singer, and W. Whyte. NAEP: provable security in the presence of decryption failures. Cryptology ePrint Archive, Report 2003/172, 2003. http://eprint.iacr.org/.
21. N. Howgrave-Graham, J. H. Silverman, and W. Whyte. A Meet-In-The-Middle Attack on an NTRU Private Key. Technical Report #004, Version 2, 2003.
22. J. Pipher J. Hoffstein and J.H. Silverman. NTRU: a new high speed public key cryptosystem. Manuscript, Rump Session Crypto'96, 1996.
23. É. Jaulmes and A. Joux. A chosen-ciphertext attack against NTRU. In *Advances in cryptology—CRYPTO 2000*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 20–35. Springer, Berlin, 2000.
24. K. S. Kedlaya. Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. *J. Ramanujan Math. Soc.*, 16(4):323–338, 2001.
25. K. S. Kedlaya. Computing zeta functions via p-adic cohomology. In Duncan A. Buell, editor, *Algorithmic Number Theory — ANTS-VI*, volume 3076 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2004.
26. N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(177):203–209, 1987.
27. N. Koblitz. Hyperelliptic cryptosystems. *J. Cryptology*, 1(3):139–150, 1989.
28. T. Lange. Formulae for arithmetic on genus 2 hyperelliptic curves. *Preprint*, 2003. Available at http://www.ruhr-uni-bochum.de/itsc/tanja/preprints/expl_sub.pdf.
29. L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An efficient protocol for authenticated key agreement. *Des. Codes Cryptogr.*, 28(2):119–134, 2003.
30. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

31. A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, 39(5):1639–1646, 1993.
32. V. S. Miller. Use of elliptic curves in cryptography. In *Advances in cryptology—CRYPTO '85*, volume 218 of *Lecture Notes in Comput. Sci.*, pages 417–426. Springer, Berlin, 1986.
33. K. Nagao. Improvement of thleriault algorithm of index calculus for jacobian of hyperelliptic curves of small genus. Cryptology ePrint Archive, Report 2004/161, 2004. Available at http://eprint.iacr.org/.
34. NTRU Cryptosystems. Peer Review and Independent Scrutiny of the NTRUEncrypt Public Key Cryptosystem, 2004. Available at http://www.ntru.com/cryptolab/pdf/review.pdf.
35. S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Trans. Information Theory*, IT-24(1):106–110, 1978.
36. J. M. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, 32(143):918–924, 1978.
37. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21(2):120–126, 1978.
38. H.-G. Rück. On the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 68(226):805–806, 1999.
39. T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, 15(4):247–270, 2000.
40. R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170):483–494, 1985.
41. R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995. Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993).
42. I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Math. Comp.*, 67(221):353–356, 1998.
43. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 256–266. Springer, Berlin, 1997.
44. J. H. Silverman. Plaintext awareness and the NTRU PKCS, version 2. Technical Report #007, 2000.
45. J. H. Silverman. Wraps, gaps and lattice constants. Technical Report #011, Version 2, 2001.
46. N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Comput. Sci.*, pages 75–92. Springer, Berlin, 2003.