Journal of Math-for-Industry, Vol.1(2009A-5), pp.33–44

Math-for-industry
Education & Research Hub

An efficient method of generating rational points on elliptic curves

Hisayoshi Sato and Keisuke Hakuta

Received on February 26, 2009

Abstract.
Several digital signature schemes based on the discrete logarithm problem or the computational Diffie-Hellman problem which have tight security reductions are proposed in these years. For these schemes, the groups of rational points on elliptic curves are employed for efficiency. These schemes need cryptographic hash functions with the range in the group of rational points on elliptic curves in their procedures for generating/verifying signatures. However, no efficient algorithm for such hash functions is known except for special type of elliptic curves, consequentially, the signature schemes becomes inefficient even if elliptic curves are employed. In this paper, in order to improve the efficiency of the signature schemes, a new method of generating rational points on elliptic curves is proposed. The proposed method is based on the norm map from a quadratic extension field of the definition field. This method consists of one powering for determination of quadratic residuosity and a square root extraction, and at most 16 times multiplications in the definition field. The security when the proposed algorithm is used as a hash function is also investigated.

Keywords. cryptography, elliptic curve, point generation, signature scheme

1. Introduction

Public key cryptography is one of the most important techniques for information security. The security of most public key cryptography is based on the intractability of computational problem such as integer factoring problem, discrete logarithm problem and so on. Along with development of public key cryptography, various attacking techniques for such cryptographic schemes or computational problems also have been evolved. In particular, study of algorithms for integer factoring problem has drastically progressed ([17], [18]) using sophisticated mathematics, consequently, moduluses (composite numbers) with thousands-bit length are needed to assure the security of schemes based on integer factoring problem. Using huge moduluses makes the efficiency of execution declining.

Against this background, the elliptic curve cryptography ([14], [21]) attracts attention for its efficiency. Although various attacks for elliptic curve cryptography, discrete logarithm problem on the groups of rational points on elliptic curves and related calculating problems have been studied, effective attacks have not found. Therefore, the size of the groups which has relation to the computational efficiency for execution of the schemes can be suppressed relatively small for the same security level.

Signature scheme is a type of public key cryptography and plays central roles in the information security. Signature schemes based on integer factoring problem have often been used so far, but by the reason mentioned above, schemes based on discrete logarithm problem over elliptic curves are promoted to prevalent candidates.

In general, "provably secure" signature schemes are applied in real systems. Here, we say "provably secure" if it can be proven that the computational cost for breaking the scheme is much the same as that for calculation of the underlying calculating problem. Moreover, the security reduction often comes to an issue. Preferably, both calculating costs are desired to be nearly equal, and in this case, the scheme is called to be secure with tight security reduction. However, the signature schemes based on elliptic curves in practical use do not achieve the tight security reduction.

Recently, several signature schemes based on elliptic curves with tight security reduction are proposed ([4], [11]). These schemes employ cryptographic hash functions ([20]) with the range in the group of rational points on elliptic curves in the procedures of signature generation/verification. The hash function must be public, that is, anyone can calculate the outputs of the function for any inputs of his own choice. On the other hand, the outputs of the hash functions should be "random". That is, the distribution of outputs is uniform and random in the range, calculating the discrete logarithm of outputs with respect to a fixed and public base point should be intractable.

Most naive implementation method for such hash function is "trial-and-error method" (Section 3). However, this method has ambiguous execution time, namely, the execution time depends on the input and takes long time in some cases, since this method consists of repeating trial-and-error, and entities with usual calculation ability (*ex.*

polynomial time algorithm) can not predict the repeating times until the last moment. Steps in the repeating procedure have considerably heavy calculation such as modular powerings. This property of the method has an adverse affect on execution in constrained devices such as smart cards.

In this paper, in order to avoid the ambiguity of execution time, we propose a new algorithm which outputs rational points for bit strings as inputs. The proposed algorithm is based on the norm map from the quadratic extension of the definition field of the elliptic curve. The group of points which are rational over the quadratic extension field contains a subset which can be parameterized by a rational curve. This subset gives non-trivial points rational over the definition field by the norm map. Moreover, this algorithm needs only one powering calculation and the execution time is almost stable. We also investigate the security in case that the proposed algorithm is used as a hash function with the range in the group of rational points. Unfortunately, the proposed algorithm is not ideal as it is in terms of "indifferentiability" setting. We explore some countermeasures for ideal security.

This paper is organized as follows: Notations in this paper are introduced in Section 2, In Section 3, conventional methods are briefly reviewed. The new basic algorithm for generating rational points on elliptic curves using the norm map is proposed in Section 4. In Section 5 and Section 6, we give an efficient algorithm using Jacobian coordinates and a referential method for extraction of square roots, and in Section 7, we briefly review the basic results on the parameterization of quadratic curves. In Section 8, we propose an efficient algorithm by integrating the results in previous sections. We also investigate the size of the range of the proposed algorithm in Section 9, and consider the security of the proposed algorithm from the aspect of the notion "indifferentiability" in Section 10 and Section 11. In Section 12, we propose an algorithm using the norm map from the quartic extension for some security aspects. We conclude in Section 13.

2. Notations

In this paper, we use the following notations:

For a finite set S, let us denote the number of its elements by $\#S$. For a subset $S_0 \subset S$, the complement subset of S_0 in S is denoted by $S \setminus S_0 = \{x \in S \mid x \notin S_0\}$. Let p be a prime number and $n \geq 1$ be an integer. The finite field with $q = p^n$ elements is denoted by $\mathbb{F}_q$, and its multiplicative group is written in $\mathbb{F}_q^\times = \mathbb{F}_q \setminus \{0\}$. For an extension $\mathbb{F}_{q^m}/\mathbb{F}_q$ of finite fields, let $G(\mathbb{F}_{q^m}/\mathbb{F}_q)$ be the Galois group. This group is a cyclic group which is generated by the q^{th}-Frobenius map. For a finite cyclic group G, $G = \langle g \rangle$ means that $g \in G$ is one of generators of G. Let $\mathcal{X}_q : \mathbb{F}_q^\times \to \{\pm 1\}$ be the quadratic character. Namely, for any $a \in \mathbb{F}_q^\times$, $\mathcal{X}_q(a) = 1 \Leftrightarrow a \in \left(\mathbb{F}_q^\times\right)^2$. When $q = p$, that is, when the field is the prime field with characteristic p, $\mathcal{X}$ is nothing less than the Legendre symbol (quadratic residue symbol). Let us extend $\mathcal{X}$ all through the field by putting $\mathcal{X}_q(0) = 0$.

Let $E/\mathbb{F}_q$ be an elliptic curve defined over $\mathbb{F}_q$ and $E(\mathbb{F}_q)$ be the group of $\mathbb{F}_q$-rational points of E. For any $P = (x, y) \in E(\mathbb{F}_{q^m})$ and $\sigma \in G(\mathbb{F}_{q^m}/\mathbb{F}_q)$, let us denote $P^\sigma = (\sigma(x), \sigma(y)) \in E(\mathbb{F}_{q^m})$.

3. Naive Method of Generating Rational Points

Let $p \geq 5$ be a prime, $q = p^n$ $(n \geq 1)$ be a power of p, and $E : y^2 = F(x) = x^3 + ax + b/\mathbb{F}_q$ be an elliptic curve. We consider a function which maps an input r $(\in \{0,1\}^*)$ to a rational point P in $E(\mathbb{F}_q)$. As a naive example for this function, we can consider a function which outputs a scalar multiplication rP_0 or $H(r)P_0$, where $P_0 \in E(\mathbb{F}_q)$ is a fixed and public point and $H : \{0,1\}^* \to \mathbb{Z}_\ell$ is a hash function (ℓ is the order of the target subgroup in $E(\mathbb{F}_q)$). However, if we consider this function as a cryptographic hash function, in order to be a public function (that is, anyone can calculate outputs for any inputs), P_0 must be public, thus the discrete logarithm of the outputs are also public information. This means that this function is distinguishable from a random oracle with the range in $E(\mathbb{F}_q)$ ([6], [19]). This property has some problem for certain public-key encryption schemes or digital signature schemes.

For particular kind of elliptic curves such as supersingular elliptic curves, there exists several efficient algorithms which output "random" points ([3], [1], [23]). However, these methods depend on the special form of the defining equation of the elliptic curves and are not applied to general (ordinary) elliptic curves.

On the other hand, there is the most simplest method: for an input r (or the output $H(r)$ of a hash function), a candidate of an x-coordinate of the elliptic curve (defined by the usual Weierstrass equation $y^2 = f(x)$) is generated, and then it is checked that the corresponding y-coordinate is rational over $\mathbb{F}_q$ or not. If it is not rational (in this case, the y-coordinate is in $\mathbb{F}_{q^2}$), then the x-coordinate is updated according to the predetermined procedure (ex. $x \leftarrow x + 1$), and the corresponding y-coordinate is checked to be rational or not again. This procedure is repeated until the y-coordinate is rational. We call this the *trial-and-error method* in this paper. For the simplicity, we assume that the characteristic p is sufficiently large:

Input : $r \in \{0,1\}^*$
Output : $P \in E(\mathbb{F}_q)$

1. Generate $x \in \mathbb{F}_q$ depending on r
2. If the corresponding y-coordinate is in $\mathbb{F}_q$, then output $P = (x, y)$. Else set $x \leftarrow x + 1$ and go to the step 2

For the number of rational points on elliptic curves, it is well-known that the following holds (H. Hasse 1933):

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

Since $2\sqrt{q} \ll q$, this inequality means that $\#E(\mathbb{F}_q)$ is around q. For $x_0 \in \mathbb{F}_q$, if there exists $y_0 \in \mathbb{F}_q$ such that

$y_0^2 = F(x_0)$, then $(x_0, y_0), (x_0, -y_0) \in E(\mathbb{F}_q)$, thus the above inequality indicates that for about half of $x_0 \in \mathbb{F}_q$, it holds that $y_0 \in \mathbb{F}_q$. Hence, the average of repeat count of the step 2 in the above algorithm is 2. Moreover, the distribution of $x \in \mathbb{F}_q$ such that the corresponding $y \in \mathbb{F}_q$ can be assumed to be uniform and random in practice, but the case such that we have to repeat the step 2 twice or more times can be occur with considerable probability.

In particular, when we use a hash function with the range in $E(\mathbb{F}_q)$, the uncertainty of the running time of the algorithm should be avoided especially for constrained devices such as smart cards. A method avoiding the repetition is desired even if the y-coordinate is not in $\mathbb{F}_q$.

This is nothing less than finding a correspondence a random number $r \in \mathbb{F}_q$ and a rational point P on the elliptic curve. But it is well-known that there exists no rational map from the projective line $\mathbb{P}^1(\mathbb{F}_q)$ to an elliptic curve except for a constant map (*e.g.* [22], Corollary 3.8), thus, we have to consider a non-rational map such as a meromorphic map.

It seems hard to consider an algorithm which generates a rational point with x-coordinate directly from a given a random number r except for the "trial-and-error method". Hence we consider a method using a field extension $\mathbb{F}_{q^m}$ of the definition field $\mathbb{F}_q$ of the elliptic curve in this paper. In this case, there exists the norm map from $E(\mathbb{F}_{q^m})$ to $E(\mathbb{F}_q)$, the image of this map dominates in $E(\mathbb{F}_q)$ (the order of the co-kernel is at most m^2). So, we consider a sufficiently large subset of $E(\mathbb{F}_{q^m})$ which has a parameterization with $\mathbb{F}_q$, and investigate a method for generating a point in $E(\mathbb{F}_q)$ by applying the norm map.

4. Basic Algorithm

In this section, we propose a basic algorithm which generates rational points for inputs in $\mathbb{F}_q$.

Let $E/\mathbb{F}_q$ be an elliptic curve defined over a finite field $\mathbb{F}_q$ ($q = p^n$, $p > 3$) and we assume that it is given by the following Weierstrass equation:

$$E : y^2 = F(x) = x^3 + ax + b, \quad a, b \in \mathbb{F}_q. \tag{2}$$

The basic idea for generating rational points is: Find a point in $E(\mathbb{F}_{q^2})$, and then calculate output of the norm map to $E(\mathbb{F}_q)$. Here, if the points in $E(\mathbb{F}_{q^2})$ are included in the kernel of the norm map, then the outputs are always trivial. The points in the kernel is characterized by the points such that the x-coordinates are in $\mathbb{F}_q$ and the y-coordinates are in $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$ (and this means $y^2 \in \mathbb{F}_q$ by the defining equation). Thus we have to consider points with x-coordinates $\in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$.

Let us denote $G(\mathbb{F}_{q^2}/\mathbb{F}_q) = \{1, \sigma\}$ the Galois group(σ : q^{th}-Frobenius map). σ acts on a point $P = (x, y) \in E(\mathbb{F}_{q^2})$ by $P^\sigma = (\sigma(x), \sigma(y))$. Then the norm map $N : E(\mathbb{F}_{q^2}) \to E(\mathbb{F}_q)$ is defined as follows[1]:

$$N : E(\mathbb{F}_{q^2}) \to E(\mathbb{F}_q), \quad P \mapsto P + P^\sigma. \tag{3}$$

Fix any element $\xi \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ such that $\xi^2 \in \mathbb{F}_q$, then

$$\mathbb{F}_{q^2} = \mathbb{F}_q(\xi), \quad c := \xi^2 \in \mathbb{F}_q. \tag{4}$$

Let $x = \alpha + \beta\xi \in \mathbb{F}_{q^2}$ and $\alpha, \beta (\neq 0) \in \mathbb{F}_q$. We consider an efficient algorithm for finding rational points such that the x-coordinates are $\mathbb{F}_{q^2}$-rational and the y-coordinates satisfy $y^2 \in \mathbb{F}_q$. In this case, the points (x, y) are always in $E(\mathbb{F}_{q^2})$. Moreover, these points give non-trivial points in $E(\mathbb{F}_q)$ by the norm map.

$E(\mathbb{F}_{q^2}) \setminus E(\mathbb{F}_q)$	$x \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$	(a) $y^2 \in \mathbb{F}_q$
		(b) $y^2 \notin \mathbb{F}_q$
	(c) $x \in \mathbb{F}_q$, $y \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, $y^2 \in \mathbb{F}_q$	
$E(\mathbb{F}_q)$	(d) $x, y \in \mathbb{F}_q$	

Table 1: Classification of points in $E(\mathbb{F}_{q^2})$

We will concentrate on the part (a) in Table 1:

$$E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q) := \left\{(x, y) \in E(\mathbb{F}_{q^2}) \mid x \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q,\ y^2 \in \mathbb{F}_q\right\}.$$

The part (c) coincides with the kernel of the norm map, thus it is useless in the algorithm generating points in $E(\mathbb{F}_q)$ using the norm map. For the part (b), the probability of $y \in \mathbb{F}_{q^2}$ when the x-coordinates run over $\mathbb{F}_{q^2}$ is almost 1/2 by the Hasse bound (1), thus the situation is same as in part (d) (*i.e.* trial-and-error method).

For these reasons, we will concentrate on the part (a) in this paper, and consider efficient algorithms generating points in $E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q)$. In the following, we will consider how points in $E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q)$ are parameterized.

Let us substitute $x = \alpha + \beta\xi$ in the defining equation (2), and rewrite to the representation of $\mathbb{F}_{q^2}$ with $\mathbb{F}_q$-basis $[\xi, 1]$, then we have:

$$\begin{aligned}&(\alpha + \beta\xi)^3 + a(\alpha + \beta\xi) + b\\ &\quad = (3\alpha^2 + \beta^2 c + a)\beta\xi + (\alpha^3 + 3\alpha\beta^2 c + a\alpha + b).\end{aligned}$$

Then we can see

$$y^2 \in \mathbb{F}_q \ \Leftrightarrow\ \alpha^2 + (c/3)\beta^2 = -a/3. \tag{5}$$

The right hand side of the above condition gives a quadratic curve C over $\mathbb{F}_q$ with α and β as variables:

$$C : \alpha^2 + (c/3)\beta^2 = -a/3. \tag{6}$$

Two rational points $(\alpha + \beta\xi, \pm\sqrt{F(\alpha + \beta\xi)})$ in $E(\mathbb{F}_{q^2})$ corresponds to a point $(\alpha, \beta) \in C(\mathbb{F}_q)$. As is well-known, $\mathbb{F}_q$-rational points (α, β) on quadratic curves over $\mathbb{F}_q$ have parameterizations over $\mathbb{F}_q$:

$$\alpha = \phi(t)/\omega(t), \quad \beta = \psi(t)/\omega(t), \tag{7}$$

[1] This map is also called the trace map since the group operation on elliptic curves are often represented additively. In this paper, we call this map the norm map according to [16].

where $\phi(t)$, $\psi(t)$ and $\omega(t)$ are polynomial in t with degree 1 or 2 in $\mathbb{F}_q[t]$. Note that the denominators of α and β are same. See Section 7 for the concrete form of the parameterization. Namely, this means that there exists a rational map $\mathbb{A}^1 \to C$ over $\mathbb{F}_q$.

For convenience, we use the following notations. At first, we consider the following equivalent relation:

$$P_1 \sim P_2 \overset{\text{def}}{\Leftrightarrow} P_1 = \pm P_2.$$

For the set $E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q)$ of rational points, if P is in $E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q)$, then it holds $-P \in E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q)$. The set divided by this relation is denoted by: $\bar{E}^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q) := E^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q)/\sim$, $\bar{E}(\mathbb{F}_q) := E(\mathbb{F}_q)/\sim$. Using these notations, we can represent the correspondence discussed above as follows:

$$\begin{array}{ccccccc} \mathbb{A}^1(\mathbb{F}_q) & \to & C(\mathbb{F}_q) & \to & \bar{E}^{(1)}(\mathbb{F}_{q^2}/\mathbb{F}_q) & \to & \bar{E}(\mathbb{F}_q) \\ t & \mapsto & (\alpha,\beta) & \mapsto & P & \mapsto & P+P^\sigma, \end{array} \tag{8}$$

where $P = \left(\alpha+\beta\xi, \sqrt{F(\alpha+\beta\xi)}\right)$.

Next, we will construct an algorithm generating rational points on elliptic curves by using the above correspondence (8). By the parameterization (7) of C over $\mathbb{F}_q$, an $\mathbb{F}_q$-rational point(α,β) on C is generated. For this point, a point $(\alpha+\beta\xi, y)$ in $E(\mathbb{F}_{q^2})$ is generated. Note that y^2 is in $\mathbb{F}_q$. Here we divide into two cases:

(a)-1. Case for $y^2 \in (\mathbb{F}_q^\times)^2$ (namely, $y \in \mathbb{F}_q^\times$): Let $P_0 := (\alpha+\beta\xi, y) \in E(\mathbb{F}_{q^2})$. Since $P_0^\sigma = (\alpha-\beta\xi, y)$, we have the following point in $E(\mathbb{F}_q)$ by the norm map:

$$P := P_0 + P_0^\sigma = (-2\alpha, -y). \tag{9}$$

(a)-2. Case for $y^2 \notin (\mathbb{F}_q^\times)^2$: Put $y = z\xi$, $z \in {\mathbb{F}_q}^2$ and $P_0 := (\alpha+\beta\xi, y) \in E(\mathbb{F}_{q^2})$. Then $P_0^\sigma = (\alpha-\beta\xi, -y)$, thus we have the following point in $E(\mathbb{F}_q)$ by the norm map:

$$\begin{aligned} P &:= P_0 + P_0^\sigma \\ &= \left((z/\beta)^2 - 2\alpha,\ -((z/\beta)^2 - 3\alpha)(z/\beta) \right). \end{aligned} \tag{10}$$

In both cases, we can have non-trivial points. These are summarized as follows:

[Algorithm 1] Generating a point in $E(\mathbb{F}_q)$:

$E : y^2 = x^3 + ax + b$, $a, b \in \mathbb{F}_q$: an elliptic curve,
the parameter representation (7) : $\alpha = \phi(t)/\omega(t)$, $\beta = \psi(t)/\omega(t)$ (where $\phi(t), \psi(t), \omega(t) \in \mathbb{F}_q[t]$), $c \in \mathbb{F}_q$ is an element given by (4).

Input : $t \in \{\mathbb{F}_q^\times/\{\pm 1\}\} \cup \{0\}$, $\iota \in \{1, -1\}$
Output : $P \in E(\mathbb{F}_q)$

1. calculate $\alpha = \phi(t)/\omega(t)$.
2. calculate $\tau = -8\alpha^3 - 2a\alpha + b$.
3. if $\chi_q(\tau) = 1$ then calculate $y = \sqrt{\tau}$ and output $P = (-2\alpha, \iota y)$.
4. else calculate $z = \sqrt{\tau/c}$, $w = z\omega(t)/\psi(t)$ and output $P = (w^2 - 2\alpha, \iota(w^2 - 3\alpha)w)$.

Where $t \in \{\mathbb{F}_q^\times/\{\pm 1\}\} \cup \{0\}$ means that the sign $\pm$ is truncated. If $q = p$ $(n = 1)$, then we merely consider elements $0 \le t \le (p-1)/2$. In general, if elements in $\mathbb{F}_q$ are given in polynomial form: $t = \sum_{i=0}^{n-1} t_i x^i$, $t_i \in \mathbb{F}_p$, then we can choose the following representatives: let j be the index such that for any $i > j$, $t_i = 0$ and $t_j \neq 0$. then for t and $-t$, the representative is t if $t_j \le (p-1)/2$, $-t$ otherwise.

The concrete form of the parameter representation (7) are given in Section 7. In the next sections, we investigate efficient methods to make the basic algorithm concrete and efficient.

5. Algorithm with Jacobian Coordinates

The proposed algorithm in Section 4 is described with the affine coordinates. However, in the affine coordinates, the modular inverse is needed in the procedure, it decreases the efficiency in general. On the other hand, using Jacobian coordinates (*ex.* see [2]) avoids the modular inverse and makes the arithmetic on the elliptic curve considerably efficient. In this section, we apply Jacobian coordinates to the proposed algorithm in order to makes the proposed algorithm efficient and practical.

The affine coordinates and Jacobian coordinates are related by $(X/Z^2, Y/Z^3) \leftrightarrow [X, Y, Z]$ $(Z \neq 0)$. Hence, it is enough to pull out the denominators in the affine coordinates to the Z-coordinate. The resulting algorithm becomes as follows:

[Algorithm 2 : Jacobi coordinate] Generating a point in $E(\mathbb{F}_q)$:

$E : y^2 = x^3 + ax + b$, $a, b \in \mathbb{F}_q$: an elliptic curve,
the parameter representation (7) : $\alpha = \phi(t)/\omega(t)$, $\beta = \psi(t)/\omega(t)$ (where $\phi(t), \psi(t), \omega(t) \in \mathbb{F}_q[t]$), $c \in \mathbb{F}_q$ is an element given by (4).

Input : $t \in \{\mathbb{F}_q^\times/\{\pm 1\}\} \cup \{0\}$, $\iota \in \{1, -1\}$
Output : $P \in E(\mathbb{F}_q)$

1. calculate $\alpha = \phi(t)$, $\gamma = \omega(t)$.
2. calculate $\tau = -8\alpha^3\gamma - 2a\alpha\gamma^3 + b\gamma^4$.
3. if $\chi_q(\tau) = 1$ then calculate $y = \sqrt{\tau}$ and output $P = [-2\alpha\gamma,\ \iota y\gamma,\ \gamma]$.
4. else calculate $z = \sqrt{c\tau}$, $w = \psi(t)$ and output $P = [c(\tau - 2c\alpha w^2\gamma), \iota c(\tau - 3c\alpha w^2\gamma)\cdot z,\ cw\gamma]$.

Moreover, this algorithm can be transformed into more efficient form by using the concrete parameterization for $\phi(t)$, $\psi(t)$ and $\omega(t)$ (Section 7). In the next section, we recall the efficient method for square root extraction, finally, in Section 8, we propose a practical efficient algorithm integrating these result.

[2]Since $y^2 \in \mathbb{F}_q$, the conjugate of y over $\mathbb{F}_q$ is $-y$, similarly, the conjugate of ξ is $-\xi$. Thus the conjugate of y/ξ is y/ξ, that is, we have $z := y/\xi \in \mathbb{F}_q$.

6. Square Root Extraction

In the procedure of the proposed basic algorithm (Algorithm 1 and 2), there is a conditional branch depending on the value of the quadratic character, followed by an extraction of a square root. In general, an extraction of a square root is done by the algorithm of Tonelli-Shanks ([5]). In particular, in case of $q = p$ (namely the case that the definition field is the prime field with characteristic p) and $p \equiv 3 \bmod 4$, it is executed simply and efficiently. For convenience to the readers, the precise algorithm is reviewed in the following.

Lemma 1. *Let us assume that $p \equiv 3 \bmod 4$. For any $x \in \mathbb{F}_p^\times$, put $g = x^{\frac{p+1}{4}}$. Then x is a quadratic residue if and only if $g^2 = x$. In this case, $\sqrt{x} = \pm g$.*

Proof. The right-hand-side is transformed as $\left(x^{\frac{p+1}{4}}\right)^2 = x^{\frac{p+1}{2}} = x^{\frac{p-1}{2}}x$. Then by Euler's criterion, we have $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}}$. x is a quadratic residue if and only if $x^{\frac{p-1}{2}} = 1$ by the definition, thus we have the lemma. □

The condition $p \equiv 3 \bmod 4$ is necessary so that the index $(p+1)/4$ is an integer. Using this lemma, the step 3 and the step 4 in the proposed algorithm in the previous section are specified as follows: Let $c \in \mathbb{F}_p$ be a non-quadratic residue (note that in this case, $\xi = \sqrt{c} \in \mathbb{F}_{q^2}$ satisfies the condition (4)) and put $g = c^{\frac{p+1}{4}}$.

1. $\tau' = \tau^{\frac{p+1}{4}}$
2. if $\tau'^2 = \tau$ then output [QR, τ'],
3. else output [NQR, $g\tau'$].

Where "QR" means that τ is a quadratic residue, "NQR" means that it is a non-quadratic residue. $g\tau'$ in the step 3 is an element which satisfies $(g\tau')^2 = c\tau$.

In the case that the modulus p is the NIST-prime $P_{256} = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$, this algorithm is executed in 255.3μ-sec on Pentium® M 1.86Ghz, 1GB RAM.

On the other hand, a 256-bit length scalar multiplication on the NIST curve[3] (on $\mathbb{F}_{P_{256}}$) is executed in 1,200μ-sec for a fixed point with the comb method (single table, width 8), and 4,900μ-sec for a random point with the 4-ary method. We can say that the above algorithm is sufficiently efficient compared with a scalar multiplication on elliptic curves.

In case of $p \equiv 1 \bmod 4$, the number of conditional branches increases according to the number of elements with 2-power orders (namely, if $p-1 = 2^e q$ $(2 \nmid q)$, then there exist 2^e such elements), the efficiency is spoiled. However, in case of $e = 2$, that is, in the case of $p \equiv 1 \bmod 4$, $\not\equiv 1 \bmod 8$, the algorithm is given in the following.

Under this condition, the supplement of the quadratic reciprocity law asserts $(-1/p) = 1$ and $(2/p) = -1$. Let us assume that $p \equiv 1 \bmod 4$, $\not\equiv 1 \bmod 8$ and $(c/p) = -1$. Put $p' = (p-1)/4$ (: odd). Moreover, let us put $g = c^{\frac{p'+1}{2}}$ and $\zeta = c^{p'}$.

1. $\tau' = \tau^{\frac{p'-1}{2}}$.
2. $\omega = \tau \cdot \tau'^2$.
3. $\tau' = \tau \cdot \tau'$.
4. if $\omega = 1$ then output [QR, τ'],
5. else if $\omega = -1$ then output [QR, $\zeta\tau'$],
6. else if $\omega = \zeta$ then output [NQR, $g\zeta\tau'$],
7. else output [NQR, $g\tau'$].

7. Parameterization of Quadratic Curves

In this section, for convenience to the readers, we briefly review the concrete parameterizations for quadratic curves

$$C : \alpha^2 + (c/3)\beta^2 = -a/3,\ \chi_q(c) = -1$$

defined over $\mathbb{F}_q$ with respect to α and β.

Let $(\alpha_0, \beta_0) \in C(\mathbb{F}_q)$ be a rational point on the curve. Let t be the coordinate of the intersection point of the line through (α_0, β_0) and another rational point (α, β) and the α-axis or β-axis then we can see that $t \in \mathbb{F}_q$, and the parameterization is given by representing (α, β) as a rational function of (α_0, β_0) and t. More precisely, the initial rational point (α_0, β_0) and the projecting axis depend on the condition of coefficients.

Case of $\chi_q(-a) = 1$.

In this case, there exists a rational point $(\alpha_0, \beta_0) \in C(\mathbb{F}_q)$, $\alpha_0 \neq 0$, from this point, we have the following parameterization:

$$\alpha = \frac{\alpha_0(ct^2 + a)}{ct^2 - a - 2c\beta_0 t}, \quad \beta = \frac{2at + \beta_0(ct^2 - a)}{ct^2 - a - 2c\beta_0 t}. \tag{11}$$

The concrete form of the rational point (α_0, β_0) is given as follows:

If $\chi_q(3) = 1$, put $e = \sqrt{-a/3} \in \mathbb{F}_q$, then C has a rational point $(e, 0)$. Otherwise $(\chi_q(3) = -1)$, we can put $c = 3$. Let $e = \sqrt{-a} \in \mathbb{F}_q$. Then C has a following rational point in each case:

$\chi_q(2) = 1$ and $f = \sqrt{2} \in \mathbb{F}_q$: $(fe/3, e/3)$.

$\chi_q(-1) = 1$ and $f = \sqrt{-1} \in \mathbb{F}_q$: $(fe/3, 2e/3)$.

$\chi_q(-1) = \chi_q(2) = -1$, $f = \sqrt{-3}$ and $g = \sqrt{6} \in \mathbb{F}_q$: $(fe/3, ge/3)$.

For each case, substituting the rational point (α_0, β_0) to the parameterization (11) and we can have the concrete representation of the parameterization.

Case of $\chi_q(-a) = -1$.

In this case, we can put $c = -a$ and C has a rational point $(0, 1)$. The intersection point with α-axis gives the following parameterization:

$$\alpha = \frac{2at}{3t^2 - a}, \quad \beta = \frac{3t^2 + a}{3t^2 - a}. \tag{12}$$

[3] See [9], [13]. NIST curves are defined over special prime fields whose characteristics are so-called NIST primes. These primes have few Hamming weights in the binary representations, this property leads high efficiency in practical use.

8. Efficient Algorithm

Summarizing results in previous sections, we have an efficient algorithm for the case of $q = p$ (that is, for the prime field) and $p \equiv 3 \bmod 4$ (we omit the precise description for the case of $p \equiv 1 \bmod 4$). In the following description, we divide into steps so that each step has at most one modular multiplication (except for some steps). This description has not been proven to be optimal, but the number of temporal parameters is reduced to minimal as possible. All calculations are done over the prime field $\mathbb{F}_p$. The following algorithm is for the case of $(3/p) = (-a/p) = -1$ (in this case, the parameterization is given by (12)).

[Algorithm 3] Generating a point in $E(\mathbb{F}_p)$:

$E : y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_p$: an elliptic curve, where $p \equiv 3 \bmod 4$, $(3/p) = (-a/p) = -1$.
$e = \sqrt{-a/3},\ g = 3^{\frac{p+1}{4}} \in \mathbb{F}_p$.

Input : $t \in \{0, 1, \ldots (p-1)/2\},\ \iota \in \{1, -1\}$
Output : $P \in E(\mathbb{F}_p)$

1. $T_0 = 1 + t^2$.
2. $T_1 = e(2 - T_0)$.
3. $T_2 = T_0 T_1$.
4. $T_3 = T_0^2$.
5. $T_1 = T_1^2$.
6. $T_1 = T_1 T_2$.
7. $T_4 = T_3^2$.
8. $T_4 = bT_4$.
9. $T_3 = T_2 T_3$.
10. $T_3 = aT_3$.
11. $T_4 = T_1 + T_3 + T_4$.
12. $T_1 = 3T_1 + T_3$.
13. $T_3 = -3T_1 + T_4$.
14. $T_5 = T_3^{\frac{p+1}{4}}$.
15. if $T_5^2 = T_3$,
    15-1. $T_5 = T_0 T_5$.
    15-2. output $P = [-2T_2,\ \iota T_5,\ T_0]$.
16. else
    16-1. $T_5 = gT_5$,
    16-2. $T_5 = T_4 T_5$,
    16-3. $T_0 = tT_0$,
    16-4. $T_0 = eT_0$.
    16-5. output $P = [-3(T_1 - T_4),\ \iota 3T_5,\ 6T_0]$.

The multiplication-by-2 and 3 can be calculated by few modular additions (and modular subtractions), thus we do not treat these as modular multiplications. Then the above algorithm is executed by at most 1 modular powering and 15 modular multiplications for general a and e. For the case of $a = -3$ ($e = 1$), this needs at most 1 modular powering and 12 modular multiplications in order to generate a rational point.

Let as assume that the bit length of p is k: $|p| = k$ (k is typically 192 and 256, etc.). The modular powering needs $(3/2)k$ times modular multiplications on the average (on randomly chosen p) when it is done by the simple binary method (We assume that the calculation cost for modular squarings is comparable to that for general modular multiplications). Hence, the above algorithm can be executed by at most $(3/2)k + 15$ times modular multiplications on the average.

On the other hand, the calculation cost for the addition of points on elliptic curve is 16 times modular multiplications in general (when we use the Jacobian-coordinates), and the cost for doubling needs 10 times modular multiplications ([2], IV.1.1). Moreover, the scalar multiplication (for unknown points) needs k times doubling and $k/2$ times additions on the elliptic curve (on the average) in case of using the binary method. Therefore, in total, it needs $10k + 16(k/2) = 18k$ times modular multiplications. The ratio of the costs of the proposed algorithm and the scalar multiplication on elliptic curves is given by

$$((3/2)k + 15)/18k \approx 1/12 \approx 0.083.$$

The ratio is approximately stable for other methods for modular powering and scalar multiplication such as the sliding window method ([2], IV.2.3). In case of trial-and-error method, it needs calculation of square root in each repeating step (the step 2), thus if the repeating count of the step is 3 or 4, then the ratio becomes 0.25 or 0.33 respectively, the cost of trial-and-error method turns into non-negligible overhead. Moreover, we can not expect that this naive method necessarily terminates in the repeating count 4. In contrast with this naive method, the proposed algorithm outputs a rational point in a fixed time, this is a definite advantage in practical use.

9. Number of Points Generated by the Basic Algorithm

We consider the number of points which are generated by the proposed algorithm. We begin with a easy result on the rational map of degree 3. We focus on the rational map $\phi : x \mapsto (sx^3 + t)/(ux^2 + v)$ $(s, t, u, v \in \mathbb{F}_q)$, and show the following proposition on the size of the image $\phi(\mathbb{F}_q)$ of $\mathbb{F}_q$ by the rational map.

Proposition 1. *Let us assume that $s, t, u, v \in \mathbb{F}_q$, $s, t, u \neq 0$, $s^2v^3 + t^2u^3 \neq 0$. Then the number of the image of $\mathbb{F}_q$ by the map $\phi : x \mapsto (sx^3 + t)/(ux^2 + v)$ is approximated as follows:*

$$\#\{\phi(x) \mid x \in \mathbb{F}_q\} \ \approx\ \left(\frac{2}{3}\right) q. \tag{13}$$

Proof. Let us assume that $x_1, (\neq) x_2 \in \mathbb{F}_q$ have same image: $(sx_1^3 + t)/(ux_1^2 + v) = (sx_2^3 + t)/(ux_2^2 + v)$. Then transforming the equation, we have

$$(x_1 - x_2)\left(sux_1^2x_2^2 + sv(x_1^2 + x_1x_2 + x_2^2) - tu(x_1 + x_2)\right) = 0.$$

Since $x_1 - x_2 \neq 0$, the second term on the left hand side is equal to 0: $(sv + sux_1^2)x_2^2 + (svx_1 - tu)x_2 - tux_1 + svx_1^2 = 0$. We regard this as an equation on x_2 for a given x_1, then the discriminant D of this equation is $D = (-svx_1 + tu)(4usx_1^3 + 3svx_1 + tu)$. D is a quadratic residue, namely, x_1 gives an X-coordinate of an $\mathbb{F}_q$-rational point on the curve

$$A : Y^2 = f(X) = (-svX + tu)(4usX^3 + 3svX + tu)/\mathbb{F}_q,$$

if and only if there exists $x_2 \in \mathbb{F}_q$ and the images by ϕ coincide. A is an elliptic curve under the condition $s, t, u \neq$

0 and $s^2v^3 + t^2u^3 \neq 0$. Hence by the Hasse bound (1), the number of $\mathbb{F}_q$-rational points has bounds

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

There exist at most 4-x_1's such that $D = 0$, when q is sufficiently large, this can be negligible. Moreover, we ignore the point at infinity, then the number of distinct x-coordinates of points in $A(\mathbb{F}_q)$ is about $\#A(\mathbb{F}_q)/2$.

Among these x-coordinates, three distinct x-coordinates correspond to one point by ϕ, hence the image of ϕ contains $(\#A(\mathbb{F}_q)/2)/3 = \#A(\mathbb{F}_q)/6$-distinct points. Moreover, there exist $q - (\#A(\mathbb{F}_q)/2)$-$x_1$'s in $\mathbb{F}_q$ which do not give x-coordinates in $A(\mathbb{F}_q)$, these are mapped to distinct points by ϕ, thus the image of ϕ contains $\#A(\mathbb{F}_q)/6 + q - (\#A(\mathbb{F}_q)/2) = q - (\#A(\mathbb{F}_q)/3)$ rational points. Therefore by the Hasse bound (1) (ignoring the term of $\sqrt{q}$), we have $\#A(\mathbb{F}_q) \approx q$ and the desired result. □

The proposed algorithm is based on the correspondence (8). To see the correspondence in the concrete, we consider Algorithm 1. The correspondence $t \mapsto \alpha = e(1-t^2)/(1+t^2)$ is 2-to-1. We assume that α's are distributed uniformly and randomly in $\mathbb{F}_q$ (namely, we assume that the probability such that an uniformly and randomly chosen element in $\mathbb{F}_q$ is in the image of the above correspondence is equal to 1/2). Moreover, we assume that the probability that τ is a quadratic residue is 1/2. Under these assumptions, the number of points given in the step 3 is about $(q/2) \cdot (1/2) \cdot 2 = q/2$ (including $\pm P$) [4]. On the other hand, the x-coordinates of P generated in the step 4 are represented using α as follows:

$$x(P) = \frac{2\alpha^3 - b}{3\alpha^2 + a}.$$

The number of elements in the image of the correspondence $x \mapsto (2x^3-b)/(3x^2+a)$ is about $(2/3)q$ when x runs over $\mathbb{F}_q$ by Proposition 1 (Since a and b are the coefficients of the definition equation of the elliptic curve, it holds that $4a^3 + 27b^2 \neq 0$, thus note that the assumption in the proposition is satisfied). α's are distributed on the half of $\mathbb{F}_q$ uniformly and randomly, and the half of them go to the step 4 by the assumption, the number of image of the correspondence $\alpha \mapsto x(P)$ in the step 4 is $q/6$.

But for each $x(P)$, if $(1 - x(P))/(1 + x(P))$ is quadratic residue and t' is a square root of it, then the $x(P)$ coincides with the image of t' in the step 3. That is, points which satisfy the condition are doubly generated in the step 3 and the step 4. Thus we have to count the number excluding the overlaps. Hence we assume that for half of $x(P)$'s, $(1 - x(P))/(1 + x(P))$'s are quadratic residue. Then, the number of points newly generated in the step 4 is $(q/6)(1/2)2 = q/6$ taking account of the sign $\pm$ of y-coordinate.

Putting together with points generated in the step 3, we have that the number N of points generated by the proposed algorithm is about

$$N \approx q/2 + q/6 = \frac{2}{3}q = 0.667 \cdot q \tag{14}$$

Examples. N in the Table 2 shows the number of rational points generated by the proposed algorithm, ratio shows the ratio of N for the total number of rational points: ratio= $N/\#E(\mathbb{F}_p)$. This results in the table support the above estimates.

p	a	b	$\#E(\mathbb{F}_p)$	N	ratio
439	63	62	431	300	0.696
463	411	150	465	314	0.675
487	129	355	499	346	0.693
499	143	126	523	342	0.654
523	25	77	537	368	0.685

Table 2: Number of Points Generated by the Proposed Algorithm

If it is enough to just generate a rational point on the curve, then the proposed algorithm is useful as it is. However, if we need a hash function with the range in the group of rational points of an elliptic curve, then the proposed algorithm do not possess enough range, and give a bias to the distribution of the image. We will consider this problem in the next section.

10. Indifferentiability in Ideal Primitive Model

In this section, we will briefly recall the notion of indifferentiability proposed by Maurer et al and consider the indifferentiability of the composition of a cryptographic hash function (maps bit-stings to bit-strings) and the proposed algorithm. The indifferentiability is a notion for a cryptographic function composed by "small" components. Here, we will deal with "ideal primitive model". That is, we will regard the "small" components as "small" ideal primitives, and then we will consider that the composed function can be seen as a "big" ideal primitive or not in the framework of indifferentiability. For details, see [19], [6].

The ideal primitive is an algorithm entity such that returns output immediately when an input is submitted by the other entity. The ideal primitive that will be dealt in this paper is the random oracle. The random oracle is an ideal primitive such that for a new query, it outputs a random value, for same input query, it replies same value. There are several ideal ciphers such as ideal ciphers, random permutations, and so on.

We will consider that a Turing machine M with oracle access to an ideal primitive $\mathcal{G}$ (*i.e.* $M(\mathcal{G})$) behaves just like an ideal primitive $\mathcal{F}$ in the framework of indifferentiability.

Definition 1 (Indifferentiability [6])**.** A Turing machine M with oracle access to the ideal primitive $\mathcal{G}$ is $(t_{\mathcal{D}}, t_{\mathcal{S}}, q, \epsilon)$-indifferentiable from the ideal primitive $\mathcal{F}$, if for any distinguisher $\mathcal{D}$, there exists a system (called a simulator) $\mathcal{S}$

[4] The points generated in the step 3 are characterized so that for the x-coordinates x, $(2 - x)/(2 + x)$ are quadratic residues.

such that satisfies following:

$$| \text{Prob}\,[\mathcal{D}(M,\mathcal{G})=1] - \text{Prob}\,[\mathcal{D}(\mathcal{F},\mathcal{S}(\mathcal{F}))=1] | \ < \ \epsilon,$$

where the simulator $\mathcal{S}$ has an oracle access to $\mathcal{F}$, and executes in at most time $t_{\mathcal{S}}$. The distinguisher executes in at most $t_{\mathcal{D}}$, makes at most q-queries. $M(\mathcal{G})$ is computationally indifferentiable from $\mathcal{F}$, if ϵ is a negligible function in the security parameter k

If $M(\mathcal{G})$ is computationally indifferentiable from $\mathcal{F}$, then we can substitute $M(\mathcal{G})$ in stead of the big ideal primitive. This makes easy to design a cryptographic scheme and assure that there is no structural flaw.

Next we consider a method to construct a cryptographic hash function with the range in the group of rational points on an elliptic curve (inputs are bit-strings).

At first, let us consider a very simple idea: Fix a point $P_0 \in E(\mathbb{F}_q)$ (*ex.* the generator of the target subgroup). Let $H : \{0,1\}^* \to \{0,1\}^n$ (n =bit length of the target subgroup of $E(\mathbb{F}_q)$) be an ordinary hash function (such as SHA [7], [8]). In fact, the composite function $r \in \{0,1\}^* \mapsto H(r)P_0$ (*i.e.* scalar multiplication on the fixed point), is not indifferentiable from a random oracle $\mathcal{H} : \{0,1\}^* \to E(\mathbb{F}_q)$ assuming the intractability of the discrete logarithm problem on the elliptic curve. Indeed, a simulator $\mathcal{P}$ has to reply the discrete logarithm x of $P = \mathcal{H}(r)$ based on P_0 (*i.e.* $P = xP_0$) for an H-query r from the distinguisher $\mathcal{D}$. Hence, if we assume that the intractability of the discrete logarithm problem, the existence of a simulator is denied, and the indifferentiability of this correspondence does not hold.

Next let us consider the composite function of H and the proposed algorithm. Let $\Phi : \{0,1,\ldots,(p-1)/2\} \times \{1,-1\} \to E(\mathbb{F}_q)$ be the function given by Algorithm 3. Then we consider the function $\Phi \circ H : r \in \{0,1\}^* \mapsto \Phi(H(r))$ [5]. In order to investigate this situation, we firstly consider the following slightly general situation:

Let Ψ be a (probabilistic) polynomial time algorithm, and let M be a Turing machine defined in the following:

1. For an input x, M makes a query x to a random oracle $\mathcal{G}$.
2. M inputs the reply x' from $\mathcal{G}$ to the algorithm Ψ, and calculates the output y.
3. M outputs y and ends.

Moreover, let us assume that Ψ is "invertible", that is, there exists an efficient algorithm Ψ^{-1} which for any element in the image of Ψ, outputs one of inputs mapped to the element. Under the assumptions, it can be seen that $M(\mathcal{G})$ is indifferentiable from a random oracle $\mathcal{F}$ by constructing a simulator as follows:

Simulator $\mathcal{P}$: For a query x from a distinguisher $\mathcal{D}$, $\mathcal{P}$ makes a query x to $\mathcal{F}$, inputs the reply y to Ψ^{-1} and calculates the outputs x'. Finally $\mathcal{P}$ returns x' to $\mathcal{D}$ as the simulated reply for a $\mathcal{G}$-query x.

Clearly, if the outputs of Ψ^{-1} is correct, then $\mathcal{P}$ can simulate accurately. That is, $\mathcal{D}$ can not distinguish the reply from $\mathcal{F}$ and the output $M(\mathcal{G})$ for a query x. Hence, the distinguishing advantage of $\mathcal{D}$ is no more than the probability that Ψ^{-1} fails to invert. Moreover, the calculating cost $t_{\mathcal{P}}$ of $\mathcal{P}$ is $t_{\mathcal{P}} = q_D T$ where q_D is the number of queries $\mathcal{D}$, and T is the cost for one operation of Φ^{-1}:

Lemma 2. *Let ϵ' be the success probability of the algorithm Φ^{-1}. Then for any distinguisher $\mathcal{D}$, there exists a simulator $\mathcal{P}$ such that $M(\mathcal{G})$ is $(t_{\mathcal{D}}, t_{\mathcal{P}}, q_D, \epsilon)$ -indifferentiable from $\mathcal{F}$, where $t_{\mathcal{P}} = q_D T$, and $\epsilon = 1 - \epsilon'$.*

Let us come back to our situation. Consider the indifferentiability of the proposed composite function $\Phi \circ H$ from a random oracle $\mathcal{H} : \{0,1\}^* \to E(\mathbb{F}_q)$. By the above discussion, we have to consider the inverse algorithm Φ^{-1}. But this can be constructed as follows: for a given rational point P, determine that P is in the image of Φ or not, if so, Φ^{-1} can be calculated by solving a polynomial over a finite field. More precisely, for the x-coordinate of P, check that the corresponding parameter t is in $\mathbb{F}_q$ or not.

Therefore, we can see that the indifferentiability depends on the success probability of inversion Φ^{-1}. The probability is given by $\epsilon' = N/\#E(\mathbb{F}_q)$ where N is the number of rational points generated by Φ (the number of elements in the image of Φ). By the discussion in Section 9, this ratio is estimated around 0.67. Hence the advantage of the distinguisher in the indifferentiability setting is nearly 0.33 (note that this does not depend on the security parameters). This is far from negligible, thus in order to construct a desired hash function, we have to improve or modify Φ so that the image is dominant in $\#E(\mathbb{F}_q)$. As an example of the modification, we consider the additional composition with scalar multiplications. We investigates this modification in the general setting in the next section.

11. Observation on the Composition with Scalar Multiplications

In this section, we will consider the following general setting:

Let S be a finite set with elements n: $\#S = n$, $S_0 \subset S$ be a subset with elements n_0: $\#S_0 = n_0$, and put $\epsilon := n_0/n$. Moreover, we assume that there exists the following bijection:

$$\Phi : S \to S : \text{a bijection.}$$

Let $\Phi^r := \Phi \circ \Phi \circ \cdots \circ \Phi$ be the r-times compositions of Φ ($r \in \mathbb{Z}_{\geq 0}$), and T be the order of Φ: $\Phi^T = id$, $\Phi^r \neq id$ for any $0 < r < T$. Let us assume the followings:

Assumption 1. $T \gg 0$.

Assumption 2. Φ is "random", *i.e.* for any subset $S_1(\neq \phi), S_2 \subset S$, it holds that

$$\#(S_1 \cap \Phi(S_2))/\#S_1 = \#(S_1 \cap S_2)/\#S_1.$$

[5] More precisely, we have to consider a function from $\{0,1\}^n$ to $\{0,1,\ldots,(p-1)/2\} \times \{1,-1\}$. However, we omit this since it has no big effect on the subsequent discussion.

Fix any $0 < t_0 < T$. Let us consider the uniform distribution on a finite set $\Sigma_0 := S_0 \times \{0, 1, \ldots, t_0 - 1\}$. Namely, $(\sigma_0, t) \in \Sigma_0$ appears with probability $1/(n_0 t_0)$. Moreover, we consider a map $\mathcal{X} : (\sigma_0, t) \mapsto \Phi^t(\sigma_0)$ from Σ_0 to the finite set S. That is, we consider an S-valued random variable. The distribution of $\mathcal{X}$ is defined as follows:

$$p_{\mathcal{X}}(\sigma) := \frac{\#\{(\sigma_0, t) \mid \mathcal{X}(\sigma_0, t) = \sigma\}}{\#\Sigma_0}.$$

Proposition 2. *For $\sigma \in S$, we have*

$$p_{\mathcal{X}}(\sigma) = \begin{cases} \dfrac{1}{n} - \dfrac{1}{t_0 n} & \text{if } \sigma \notin S_0, \\ \dfrac{1}{n} + \dfrac{1}{t_0 n}\left(\dfrac{1}{\epsilon} - 1\right) & \text{if } \sigma \in S_0. \end{cases}$$

Proof.

Let us assume that $\sigma \notin S_0$. If $t = 0$, then there exists no σ_0 such that $\mathcal{X}(\sigma_0, 0) = \sigma$. For each $0 < t < t_0$, the probability for existence of σ_0 such that $\mathcal{X}(\sigma_0, t) = \sigma$ is equal to ϵ by the assumption. Hence, the total probability of appearance is equal to $(t_0 - 1)\epsilon/\#\Sigma_0$, this gives the desired result.

Similarly, in the case of $\sigma \in S_0$, if $t = 0$, then we have $\mathcal{X}(\sigma, 0) = \sigma \in S_0$. It is similar for each $0 < t < t_0$, hence we can see that the probability of appearance is $(1 + (t_0 - 1)\epsilon)/\#\Sigma_0$. □

Put $\mathcal{F} := \bigcup_{0 \le t \le t_0} \Phi^t(S_0)$. The following lemma is clear from the assumptions.

Lemma 3. $P_{t_0} := \#\mathcal{F}/\#S = 1 - (1 - \epsilon)^{t_0}$.

P_{t_0} approaches to 1 rapidly according with the increase of t_0. Figure 1 shows the case of $\epsilon \approx 0.6$.

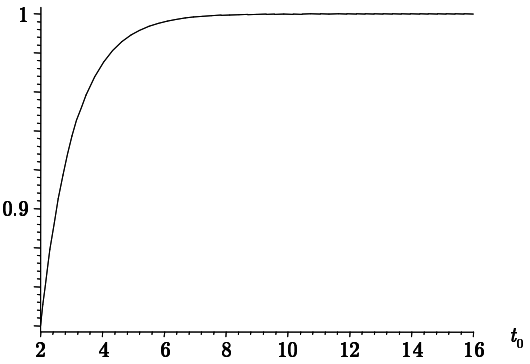

Figure 1: Distribution of the images

In case of $t_0 = 128$, we have $1 - P_{t_0} \approx 1/2^{169}$, thus the probability that a point does not contained in the image is negligible in the sense of 128-bit security.

However, in order that the distribution $\mathcal{X}$ on S is indistinguishable from the uniform distribution on S, $1/t_0$ must be negligible (with respect to the security parameter):

Put $|n| = k$, we call this a security parameter. In this case, the distribution $\mathcal{X}$ is denoted by $\mathcal{X}_k$, and the uniform distribution on S is denoted by $\mathcal{U}_k$. For the security parameter k, we assume that $\epsilon = \#S_0/\#S$ is a constant. Moreover, we assume the following:

Assumption 3. For any k, and a given $\sigma \in S$, it can be checked that $\sigma \in S_0$ or not in polynomial time (in k).

Under this assumption, the following holds:

Proposition 3. *Ensembles*[6] *$\{\mathcal{X}_k\}$ and $\{\mathcal{U}_k\}$ are indistinguishable if and only if $1/t_0$ is negligible with respect to the security parameter k.*

Proof. We construct a distinguisher D as follows: for an input $\sigma \in S$, if $\sigma \in S_0$, then $D(\sigma) = 1$, else $D(\sigma) = 0$. By the assumption, D runs in polynomial time. The advantage of D is calculated as follows:

$$\begin{aligned} &\Pr[D(\mathcal{U}_k) = 1] \\ &= \sum_{\sigma \in S} \Pr[D(\sigma) = 1 \mid \mathcal{U}_k = \sigma] \cdot \Pr[\mathcal{U}_k = \sigma] \\ &= \sum_{\sigma \in S_0} 1 \cdot \frac{1}{n} \;=\; \frac{n_0}{n} \;=\; \epsilon, \end{aligned}$$

$$\begin{aligned} &\Pr[D(\mathcal{X}_k) = 1] \\ &= \sum_{\sigma \in S} \Pr[D(\sigma) = 1 \mid \mathcal{X}_k = \sigma] \cdot \Pr[\mathcal{X}_k = \sigma] \\ &= \sum_{\sigma \in S_0} 1 \cdot \left(\frac{1}{n} + \left(\frac{1}{\epsilon} - 1\right) \cdot \frac{1}{t_0 n}\right) \\ &= \epsilon \left(1 + \left(\frac{1}{\epsilon} - 1\right) \cdot \frac{1}{t_0}\right). \end{aligned}$$

Hence the advantage is

$$|\Pr[D(\mathcal{X}_k) = 1] - \Pr[D(\mathcal{U}_k) = 1]| = \frac{1 - \epsilon}{t_0}.$$

By the assumption, ϵ is a constant, hence if t_0 is a polynomial, then we can see that there exists an efficient (polynomial time) algorithm D which distinguish $\mathcal{X}_k$ and $\mathcal{U}_k$. By the contraposition, we see that if $\mathcal{X}_k$ is indistinguishable from $\mathcal{U}_k$, then $1/t_0$ is negligible.

On the other hand, the statistical distance $\Delta(k)$ of ensembles $\mathcal{X}_k$ and $\mathcal{U}_k$ is

$$\begin{aligned} \Delta(k) \;&:=\; \frac{1}{2} \sum_{\sigma \in S} |\Pr[\mathcal{X}_k = \sigma] - \Pr[\mathcal{U}_k = \sigma]| \\ &=\; \frac{1}{2} \sum_{\sigma \in S} \left| \Pr[\mathcal{X}_k = \sigma] - \frac{1}{n} \right| \\ &=\; \left(\frac{1}{2} \sum_{\sigma \in S_0} \left(\frac{1}{\epsilon} - 1\right) \cdot \frac{1}{t_0 n}\right) + \left(\frac{1}{2} \sum_{\sigma \notin S_0} \frac{1}{t_0 n}\right) \\ &=\; \frac{1 - \epsilon}{t_0}. \end{aligned}$$

[6] a family of random variables

Hence if $1/t_0$ is negligible, then $\mathcal{X}_k$ is statistically indistinguishable from $\mathcal{U}_k$[7]. The statistically indistinguishability implies the (computational) indistinguishability ([12], §3.8.4, Exercise 6), hence we have shown the opposite direction. □

By this proposition, in order to achieve the indifferentiability, we have to set $1/t_0$ negligible. For example, $t_0 \approx 2^{128}$ for the 128-bit security.

In the case of the proposed composite function $\Phi \circ H$, we have to composite a randomly chosen scalar multiplication with 128 bit length scalar (note that several assumptions are still assumed). This has a significant impact on the efficiency. We still have to consider an efficient method for making the function have dominant range in the rational points. In the next section, we propose the algorithm using the norm map from the quartic extension of the definition field whose image is expected to be dominant and uniform.

12. Generating Rational Points using the Norm Map from the Quartic Extension

As mentioned in the previous section, the image of the basic algorithm using the norm map from the quadratic extension has a distribution with a non-negligible bias. In this section, we consider the double application of the basic algorithm which based on the norm map from the quartic extension.

Let $\mathbb{F}_q$ be the definition field. Although the purpose is to generate $\mathbb{F}_q$-rational points on an elliptic curve $E/\mathbb{F}_q$, we start to apply the basic algorithm to E regarding it as an elliptic curve defined over the quadratic extension $\mathbb{F}_{q^2}$.

Let $\xi \in \mathbb{F}_{q^4}$ and $c = \xi^2 \in \mathbb{F}_{q^2}$. The coefficients a, b of the definition equation of E are elements in $\mathbb{F}_q$. Since $\chi_{q^2}(-a) = 1$, $\chi_{q^2}(3) = 1$, the quadratic curve over $\mathbb{F}_{q^2}$ defined by the equation

$$\alpha^2 + (c/3)\beta^2 = -a/3$$

has a parameterization as follows: Let $e = \sqrt{-a/3} \in \mathbb{F}_{q^2}$ and

$$\alpha = \frac{e(ct^2+a)}{ct^2-a}, \quad \beta = \frac{2at}{ct^2-a}. \tag{15}$$

Applying the basic algorithm for $t \in \mathbb{F}_{q^2}$, we have a point P in $E(\mathbb{F}_{q^2})$. Moreover, we apply the norm map to the point and have a point in $E(\mathbb{F}_q)$: $P + P^\sigma$ ($\sigma : q^{\text{th}}$ Frobenius map). For some t, the α in the basic algorithm is in $\mathbb{F}_q$, hence after applying the basic algorithm, it can be seen that the image in $E(\mathbb{F}_{q^2})$ includes $E(\mathbb{F}_q)$. If the order $\#E(\mathbb{F}_q)$ is odd, then by the norm map from $\mathbb{F}_{q^2}$ to $\mathbb{F}_q$, $E(\mathbb{F}_q)$ is mapped to $2E(\mathbb{F}_q) = E(\mathbb{F}_q)$, thus we can see that this algorithm is surjective. In the following, we describe several algorithm needed in the generating method using the norm map from $\mathbb{F}_{q^2}$.

[7]The distinguisher D achieves the statistical distance as the advantage, thus this is optimal.

Basis of $\mathbb{F}_{p^4}$ over $\mathbb{F}_{p^2}$

For efficiency, the basis should be simple as possible. We focus on $\mathbb{F}_{p^4}$ since the main purpose is an elliptic curve over the prime field $\mathbb{F}_p$. Let $(\cdot/p)$ $(= \chi_p(\cdot))$ be the Legendre symbol. The algorithm depends on the value p mod 4:

Case of $p \equiv 3 \bmod 4$: Note that $(-1/p) = -1$ in this case.
1. Let $s \in \mathbb{F}_p$ be an element such that $(s^2 + 1/p) = -1$. Then, $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{-1})$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\sqrt{s+\sqrt{-1}})$.
2. Let $s \in \mathbb{F}_p$ be an element such that $(s/p) = (1 - s/p) = -1$. Then, $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{s})$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\sqrt{1+\sqrt{s}})$.

Example. NIST prime P-256 $(= 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1)$: As examples for the basis in 1, we can take $s = 5, 6, 8, \ldots$, and as examples for the basis in 2, we can take $s = -2, 3, 6, \ldots$.

Case of $p \equiv 1 \bmod 4$: Note that $(-1/p) = 1$ in this case. Let $s \in \mathbb{F}_p$ be an element such that $(s/p) = -1$. Then $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{s})$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\sqrt[4]{s})$.

Note that we do not have to calculate the basis ξ of $\mathbb{F}_{p^4}$ over $\mathbb{F}_{p^2}$ concretely (*ex.* $\sqrt{s+\sqrt{-1}}$), but we have to decide the form of ξ^2.

Square Root Extraction in $\mathbb{F}_{q^2}$

Let $\xi \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ and $\xi^2 = c \in \mathbb{F}_q$. We consider an algorithm which determines that $x = x_0 + x_1\xi \in \mathbb{F}_{q^2}$ and $x_0, x_1 \in \mathbb{F}_q$ are quadratic residue in $\mathbb{F}_{q^2}$ or not, and if so, outputs their square roots. Let $Nx = x_0^2 - x_1^2 c \in \mathbb{F}_q$ be the norm of x with respect to the extension $\mathbb{F}_{q^2}/\mathbb{F}_q$.

1. If $\chi_q(Nx) = 0$ then output 0, else if $\chi_q(Nx) = -1$, then outputs "NQR" and end.
2. If $\chi_q(Nx) = 1$, calculates $d = \sqrt{Nx} \in \mathbb{F}_q$.
3. If $\chi_q((x_0 + d)/2) = -1$, then let $d \leftarrow -d$, and calculates $y_0 = \sqrt{(x_0+d)/2} \in \mathbb{F}_q$.
4. Outputs $y = y_0 + (\frac{y_0 x_1}{x_0+d})\xi$ and end.

The more precise procedure in case of $q = p$, $p \equiv 3 \bmod 4$ is given as follows:

1. $Nx = x_0^2 - x_1^2 c$.
2. $d = (Nx)^{\frac{p+1}{4}}$.
3. If $d = 0$, then output 0 and end.
4. Else if $d^2 \neq Nx$, then output "NQR" and end.
5. Else
   5-1. $\tau = \frac{x_0+d}{2}$ $(= (x_0 + d) \times \frac{p+1}{2})$
   5-2. $\tau' = \tau^{\frac{p+1}{4}}$
   5-3. If $\tau'^2 = \tau$, then $y_0 = \tau'$, $y_1 = x_1\tau'/(x_0+d)$, else $y_0 = \frac{\xi}{\sqrt{-1}}(x_1\tau'/(x_0+d))$, $y_1 = \tau'/(\xi\sqrt{-1})$.
6. Output $[y_0, y_1]$ and end.

In the algorithm, when $x = x_0 + x_1\xi$ is a non-quadratic residue, for another non-quadratic constant $\eta = \alpha + \beta\xi$, $\sqrt{x\eta}$ has to be calculated. This is done by slightly modifying the above algorithm:

$\eta = \alpha + \beta\xi \in \mathbb{F}_{p^2}$: a constant
$\eta' = (N\eta)^{\frac{p+1}{4}}$

1. $Nx = x_0^2 - x_1^2 c$.

2. $d = (Nx)^{\frac{p+1}{4}}$.
3. If $d = 0$, then output 0 and end.
4. Else if $d^2 \neq Nx$, then $d \leftarrow d\eta'$, $x_0 \leftarrow \alpha x_0 + \beta x_1 c$, $x_1 \leftarrow \beta x_0 + \alpha x_1$.
5. Calculate
   5-1. $\tau = \frac{x_0+d}{2}$ $(= (x_0 + d) \times \frac{p+1}{2})$
   5-2. $\tau' = \tau^{\frac{p+1}{4}}$
   5-3. If $\tau'^2 = \tau$, then $y_0 = \tau'$, $y_1 = x_1\tau'/(x_0 + d)$, else $y_0 = \frac{\xi}{\sqrt{-1}}(x_1\tau'/(x_0 + d))$, $y_1 = \tau'/(\xi\sqrt{-1})$.
6. Output $[y_0, y_1]$ and end.

Integrating the above elemental algorithms, we can construct the algorithm by the norm from the quartic extension. We omit the precise description of the whole algorithm to eliminate redundancy.

As mentioned above, the proposed algorithm is surjective and the distribution of the image is expected to be "random". It is expected that each preimage has $(2/3)q$-elements in average by Proposition 1 applying to $\mathbb{F}_{q^2}$. If these properties can be shown, then the proposed algorithm is a strong candidate for a hash function with the range on elliptic curves. The detailed analysis is the subject of future investigation.

13. Conclusion

In this paper, we proposed an efficient algorithm for generating rational points on elliptic curves. This algorithm is available for general elliptic curves defined in the Weierstrass equation. The proposed method is based on the norm map from a quadratic extension field of the definition field. This method needs only one powering for determination of quadratic residuosity and a square root extraction, and at most 16 times multiplications in the definition field. Moreover, we investigated the security when the proposed algorithm is used as a hash function with the range in the group of rational points. In the indifferentiability setting, simple application of the algorithm can not achieve the desired security, however, two-step application of the algorithm for the quartic extension of the definition field might do well. The rigorous analysis remains to be addressed. In order to study deeper in these area, huge amount of deep results in mathematics are conducive and help us to find a breakthrough.

References

[1] Barreto, P.S.L. M. and Kim, H.Y.: Fast hashing onto elliptic curves over fields of characteristic 3, Cryptology ePrint Archive, Report 2001/098, 2001, available from: `http://eprint.iacr.org/2001/098`

[2] Blake, I., Seroussi, G. and Smart, N.: *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.

[3] Boneh, D., Shacham, H. and Lynn, B.: Short signatures from the Weil pairing, in: *Advances in Cryptology - ASIACRYPT '01*, LNCS 2248 (2001), Springer-Verlag, 514–532.

[4] Chevallier-Mames, B.: An Efficient CDH-based Signature Scheme With a Tight Security Reduction, in: *Advances in Cryptology - CRYPTO '05*, LNCS 3621 (2005), Springer-Verlag, 511–526,

[5] Cohen, H.: *A Course In Computational Algebraic Number Theory*, GTM 138, Springer-Verlag, 1993.

[6] Coron, J.-P., Dodis, Y., Malinaud, C. and Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function, in: *Advances in Cryptology - CRYPTO '05*, LNCS 3621 (2005), Springer-Verlag, 430–448.

[7] FIPS 180, Secure Hash Standard (SHS), Federal Information Processing Standards Publication 180, National Institute of Standards and Technology, (superseded by FIPS 180-1 and FIPS 180-2) 1995.

[8] FIPS 180-2, Secure Hash Standard (SHS), Federal Information Processing Standards Publication 180-2, National Institute of Standards and Technology, 2002.

[9] FIPS 186-2, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-2, National Institute of Standards and Technology, 2000.

[10] Galbraith, S.D.: Supersingular Curves in Cryptography, in: *Advances in Cryptology - ASIACRYPT '01*, LNCS 2248 (2001), Springer-Verlag, 495–513.

[11] Goh, E.J. and Jarechi, S.: A Signature Scheme as Secure as the Diffie-Hellman Problem, in: *Advances in Cryptology - EUROCRYPT '03*, LNCS 2248 (2003), Springer-Verlag, 401–415.

[12] Goldreich, O.: *Foundations of Cryptography: Volume 1 - Basic Tools*, Cambridge University Press, 2001.

[13] Hankerson, D., Menezes, A. and Vanstone, S.: *Guide to Elliptic Curve Cryptography*, Springer-Verlag, 2004.

[14] Koblitz, N.: Elliptic Curve Cryptosystems, in: *Mathematics of Computation*, **48** (1987), 203–209.

[15] Koblitz, N.: A Family of Jacobians Suitable for Discrete Log Cryptosystems, in: *Advances in Cryptology - CRYPTO '88*, LNCS 403 (1988), Springer-Verlag, 94–99.

[16] Kramer, K.: Arithmetic of Elliptic Curves Upon Quadratic Extension, in: *Transactions of the American Mathematical Society*, **264**, No.1 (1981), 121–135.

[17] Lenstra, A.K., Lenstra Jr., H.W., Manasse, M.S. and Pollard, J.M.: The number field sieve, in: *Proc. of STOC '90* (1990), 564–572.

[18] Lenstra, A.K. and Lenstra Jr., H.W. (eds.): *The Development of the Number Field Sieve*, LNM 1554, Springer-Verlag, 1993.

[19] Maurer, U., Renner, R. and Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology, in: *Theory of Cryptography - TCC '04*, LNCS 2951 (2004), Springer-Verlag, 21–39.

[20] Menezes, A., Oorschot, P.C.van and Vanstone, S.A.: *Handbook of Applied Cryptography*, CRC Press, 1996.

[21] Miller, V.: Uses of Elliptic Curves in Cryptography, in: *Advances in Cryptology – CRYPTO '85*, LNCS 218 (1986), Springer-Verlag, 417–426.

[22] Milne, J.S.: Abelian Varieties, in: *Arithmetic Geometry*, G. Cornell and J. H. Silverman Eds., Chap. V, 103–150, Springer-Verlag, 1986.

[23] Scott, M.: Deterministic hashing to points on IBE-friendly elliptic curves, 2005, available from: `ftp://ftp.computing.dcu.ie/pub/resources/crypto/note.pdf`

Hisayoshi Sato
Hitachi, Ltd., Systems Development Laboratory, 292, Yoshida-cho, Totsuka-ku, Yokohama, 244-0817, Japan.
E-mail: hisayoshi.sato.th(at)hitachi.com

Keisuke Hakuta
Hitachi, Ltd., Systems Development Laboratory, 292, Yoshida-cho, Totsuka-ku, Yokohama, 244-0817, Japan.
E-mail: keisuke.hakuta.cw(at)hitachi.com