SCIS'97-12B

SCIS'97 The 1997 Symposium on
Cryptography and Information Security
Fukuoka, Japan, January 29-February 1, 1997
The Institute of Electronics,
Information and Communication Engineers

楕円曲線上のエルガマル署名のトラップドアアルゴリズム

A trapdoor generating algorithm over elliptic curve ElGamal signature

宮地 充子

Atsuko Miyaji

松下電器産業株式会社

門真市門真 1006

Matsushita Electric Industrial Co., LTD.

E-mail : miyaji@isl.mei.co.jp

1006, KADOMA, KADOMA-SHI, OSAKA

あらまし Eurocrypt'96 において Bleichenbacher によりあるエルガマル署名に対する攻撃が提案された．Bleichenbacher-攻撃はトラップドアが構成できることが必須である．Bleichenbacher-攻撃を楕円曲線に適用する方法が筆者によって検討されているが，楕円曲線上のエルガマル署名に対しては常に適用できるわけでなく，また攻撃に必須のトラップドアも容易に構成できないなど，楕円曲線特有の性質が現れている．トラップドアの概念は鍵供託などへの利用など有効に使われる可能性もあり構成方法は重要である．本論文では，楕円曲線上のエルガマル署名に対するトラップドアを構成するアルゴリズムを提案するとともにそのアルゴリズムの応用について議論する．

abstract At Eurocrypt'96 Bleichenbacher proposed an attack over ElGamal signature. A trapdoor is necessary for Bleichenbacher-attack. We have investigated how to apply Bleichenbacher-attack to ElGamal signature over elliptic curves. Bleichenbacher-attack does not necessarily work ElGamal over elliptic curves and the trapdoor cannot be constructed well. The conception of a trapdoor might be used for a constructive purpose such as Key-Escrow system. So we are interested how to generate a trapdoor over elliptic curves. In this paper, we propose a new trapdoor generating algorithm and investigate how to apply the algorithm.

1 Introduction

The ElGamal signature scheme([4]) was proposed in 1985, which is based on the difficulty of the discrete logarithm problem(DLP). On the other hand, the NIST Digital Signature Algorithm(DSA)([10]) which is one of variants of ElGamal signature was proposed in 1991. There had not been remarkable differences between ElGamal and DSA, but in 1996 Bleichenbacher proposed an attack which works on ElGamal but not on DSA. Bleichenbacher-attack can be avoided by modifying ElGamal signature slightly. However Bleichenbacher-attack is important from the viewpoint of of security equivalence between signature schemes([12, 9]): the attack indicates that ElGamal is not equivalent to DSA. Furthermore Bleichenbacher-attack points out the existence of a trapdoor over ElGamal signature: whoever knows a trapdoor for ElGamal signature can generate any user's valid signature on any message. The trapdoor is constructed on a basepoint of a definition field $\mathbb{F}_p$. As for ElGamal signature, the trapdoor generating algorithm is also proposed([2]).

In the case of ElGamal signatures over elliptic curves, called ECElG in this paper, the author investigates how Bleichenbacher-attack is applied to ECElG([9]). It is shown that all ECElG is not necessarily vulnerable to Bleichenbacher-attack. A trapdoor generating algorithm over ECElG is also reported in [9]. However the algorithm is not so efficient since it is a combination of choosing a random elliptic curve suitable for ECElG and checking whether a basepoint of the elliptic curve satisfies the trapdoor condition. In fact, there has not been an effective method of constructing elliptic curves which is suitable for cyprotsystems and, at the same time, has a designated basepoint. Such a method would be useful for constructing elliptic curves with a trapdoor over ElGamal signature. A trapdoor for ElGamal signature can be used for a constructive purpose like "proxy signature": an authority who knows the trapdoor can generate a valid signature instead of any user. Therefore we take interest in constructing a trapdoor algorithm over elliptic curves.

In this paper, we propose an effective method of constructing elliptic curves suitable for cyprotsystems with a designated basepoint. Using this method, we propose an effective trapdoor generating algorithm over ECElG. We also discuss the running time. Furthermore we investigate how to apply the method to construct elliptic curves suitable for cryptosystems.

This paper is organized as follows. Section 2 summarizes ElGamal signature, DSA, and Bleichenbacher-attack over ElGamal. Section 3 summarizes ElGamal signature and DSA over elliptic curves. Section 4 shows Bleichenbacher-attack over elliptic curves and presents a new trapdoor generating algorithm over elliptic curves. Section 4.3 investigates how to apply the algorithm to elliptic curve cryptosystems.

2 ElGamal, DSA and Bleichenbacher-attack

This section summarizes ElGamal, DSA, and Bleichenbacher-attack. We assume that, in any signature schemes, the trusted authority uses a common system parameters: a large prime p, a large prime factor q of $p-1$ and a basepoint $g \in \mathbb{F}_p = GF(p) = \{0, \ldots, p-1\}$ whose order is q. These system parameters are known to all users. The signer Alice has a secret key x_A and publishes its corresponding public key $y_A = g^{x_A} \pmod{p}$. The original ElGamal signature([4]) uses a generator of $\mathbb{F}_p^* = \{1, \ldots, p-1\}$ as a basepoint. However for practical purposes([13, 12]), we use the above basepoint in $\mathbb{F}_p$. Here we summarize how each signature scheme is defined for $m \in \mathbb{F}_p^*$, where m is typically a hashed value of a message.

ElGamal

Alice chooses a random number $k \in \mathbb{F}_q^*$, and computes $r_1 = g^k \pmod{p}$ and $r_1' = r_1 \pmod{q}$. Then she computes $s \in \mathbb{F}_q^*$ from

$$sk = m + r_1' x_A \pmod{q}. \tag{1}$$

Here if $s = 0$, then she chooses the random number k again. Of course such a probability is negligibly small. Then the triplet $(m; (r_1, s))$ constitutes the signed message. The signature verification is done by checking that $(r_1, s) \in \mathbb{F}_p^* \times \mathbb{F}_q^*$ and the next equation,

$$r_1^s = g^m y_A^{r_1'} \pmod{p}. \tag{2}$$

We make the sign $+$ of r_1' in Equation (1) coincide with that of DSA since the following discussion holds regardless of signs.

DSA

Alice chooses a random number $k \in \mathbb{F}_q^*$, and computes $r_1 = g^k \pmod{p}$ and $r_1' = r_1 \pmod{q}$. Then she computes $s \in \mathbb{F}_q^*$ from Equation (1). Here if $r_1' = 0$ or $s = 0$, then she chooses the random number k again. Then the triplet $(m; (r_1', s))$ constitutes the signed message. The signature verification is done by checking $(r_1', s) \in \mathbb{F}_q^* \times \mathbb{F}_q^*$ and the next equation,

$$r_1' = (g^{m/s} y_A^{r_1'/s} \pmod{p}) \pmod{q}. \tag{3}$$

Here we summarize Bleichenbacher-attack([2]) over ElGamal.

Bleichenbacher-attack over ElGamal:

Assume that a forger knows $\beta \in \mathbb{F}_p^*$ and $t \in \mathbb{F}_q^*$ such as $\beta = 0 \pmod{q}$ and $\beta^t = g \pmod{p}$. Then he can generate any user's valid signature on any $m \in \mathbb{F}_p$ as follows. For $\forall m \in \mathbb{F}_p^*$, he sets $r_1 = \beta$ and $s = tm \pmod{q}$. Then (r_1, s) is a valid signature on m for any user since (r_1, s) satisfies the verification equation 2 as follows: $g^m y_A^{r_1} r_1^{-s} = g^m g^{-tm/t} = 1$.

In the case of DSA-signature, such $r_1 = \beta$ is already removed. Therefore DSA is strong against the attack. For practical purposes, it might be significant to remove the case of $r_1 = 0$ from ElGamal signature. Importantly, it indicates the security differences between ElGamal and DSA([9]). Furthermore the attack points out the existence of trapdoor over ElGamal signature. For a given $\mathbb{F}_p$ and g, it would be difficult to find the trapdoor β and t. However, an authority might generate $\mathbb{F}_p$ and g with a trapdoor β and t. We summarize a trapdoor generating algorithm([2]):

1. set $\mathbb{F}_p$, a large prime $q|p-1$, and $p-1 = qn$,
2. find $\beta = lq$ ($l \in \{1, \cdots, n-1\}$) such that the order of β is q,
3. set a basepoint $g = \beta^t$ for $1 < t < q-1$.

We usually set a prime p to be 768 or 1,024 bits and a prime q to be 160 bits or more. The algorithm does not construct directly a definition field with a designated basepoint. Therefore this algorithm is not so efficient. In fact we might repeat n trials on the average before finding a suitable β since the number of elements with the order q is q. Let us change the algorithm to the following: choose a random β with the order q and check whether β is divisible by q. Even with the algorithm, we might repeat q trials on the average before finding a suitable β since the number of elements divisible by q is n. However a trapdoor can be used for a constructive purpose like proxy signature. Therefore we would like to construct a definition field with a designated basepoint in order to construct ElGamal signature with a trapdoor.

3 ECElG and ECDSA

The ElGamal-type signatures can be constructed in other groups, as long as DLP is hard. So ElGamal and DSA can be also constructed on an elliptic curve,the number of elements divisible by q is n and the number of elements divisible by q is n and the number of elements divisible by q is n and which are called ECElG and ECDSA in this paper. In this section we summarize how ECElG and ECDSA are defined for a message $m \in \mathbb{F}_p^*$.

We assume that the trusted authority chooses an elliptic curve $E/\mathbb{F}_p$(p is a large prime) and a basepoint $G \in E(\mathbb{F}_p)$ with a large prime order q. The signer Alice has a secret key x_A and publishes the corresponding public key $Y_A = x_A G$. The following discussion also holds in the case of $E/\mathbb{F}_{2^r}$.

ECElG

Alice chooses a random number $k \in \mathbb{F}_q^*$, and computes

$$R_1 = kG, \tag{4}$$

in E. Then she sets $r_1' = x(R_1) \pmod q$ and computes $s \in \mathbb{F}_q^*$ from Equation (1), where $x(R_1)$ denotes the x-coordinate of R_1. Here if either $x(R_1) = 0$ or $s = 0$, then she chooses the random number k again. Then the triplet $(m; (R_1, s))$ constitutes the signed message. The signature verification is done by checking $x(R_1) \in \mathbb{F}_p^*$, $s \in \mathbb{F}_q^*$, and the next equation in E,

$$sR_1 = mG + r_1' Y_A, \tag{5}$$

where $r_1' = x(R_1) \pmod q$.

ECDSA

Alice chooses a random number $k \in \mathbb{F}_q^*$, computes Equation (4), and sets

$$r_1' = x(R_1) \pmod q. \tag{6}$$

Then she computes $s \in \mathbb{F}_q^*$ from Equation (1). Here if either $r_1' = 0$ or $s = 0$, then she chooses another random number k again. Then the triplet $(m; (r_1', s))$ constitutes the signed message. The signature verification is done by checking $r_1', s \in \mathbb{F}_q^*$ and the next equation,

$$r_1' = x(\frac{m}{s}G + \frac{r_1'}{s}Y_A) \pmod q. \tag{7}$$

4 Bleichenbacher-attack over elliptic curves

In Section 2 we saw that a trapdoor generating algorithm is necessary for Bleichenbacher-attack, which needs a method to construct $\mathbb{F}_p$ with a designated basepoint. In the case of elliptic curves it is reported that Bleichenbacher-attack works for ECElG that is not equivalent to ECDSA([9]). Furthermore two trapdoor generating algorithms are presented in [9]. However the algorithms are not so efficient since they cannot construct directly an elliptic curve with a designated basepoint.

The conception of a trapdoor can be used for a constructive purpose like a proxy signature. Therefore we get interested in constructing a trapdoor over elliptic curves.

This section presents a new trapdoor generating algorithm by using a feature of elliptic curves after summarizing Bleichenbacher-attack over ECElG.

4.1 Bleichenbacher-attack over ECElG

We assume $q < p$ since Bleichenbacher-attack works over ECElG only in the case of $q < p$ ([9]). Then the attack is as follows. Assume that a forger knows $B \in E(\mathbb{F}_p)$ and $t \in \mathbb{F}_q^*$ such as $x(B) \in \mathbb{F}_p^*$, $x(B) = 0 \pmod q$, and $tB = G$. For $m \in \mathbb{F}_p^*$, he sets $R_1 = B$ and $s = tm \pmod q$. Then (R_1, s) is a valid Alice's signature on m since (R_1, s) satisfies the verification equation 5 as follows:

$$mG + x(R_1)Y_A - sR_1 = mG - tm/tG = \mathcal{O}.$$

A natural-trial trapdoor generating algorithm is as follows([9]):

A natural-trial trapdoor generating algorithm over elliptic curves

1. *Construct an elliptic curve with a nearly-prime-order*
   set $E/\mathbb{F}_p$, a large prime $q|\#E(\mathbb{F}_p)$ and $\#E(\mathbb{F}_p) = nq$.

2. *Check the trapdoor condition*
   find $B \in E(\mathbb{F}_p)$ with $x(B) = lq$ $(l \in \{1, \cdots, n-1\})$ such that the order of B is q. If fails, go to step 1.

3. *Set a basepoint*
   set a basepoint $G = tB$ for $1 < \forall t < q - 1$.

4. *Output an elliptic curve with a trapdoor*
   output E and G with a trapdoor t and B.

The above natural-trial trapdoor generating algorithm over elliptic curves seems cannot construct directly an elliptic curve with a designated basepoint in the same way as the case of finite fields. In the case of elliptic curves we can take p and q whose sizes are almost the same since there does not exist any sub-exponential algorithm to compute the logarithm on elliptic curves chosen suitably([11]). Therefore it may often occur that an elliptic curve of step 1 fails in step 2. So we must repeat step 1 until the elliptic curve of step 1 succeeds in step 2. However, step 1 does not necessarily run fast([14]). It would be preferable to construct an elliptic curve with the designated basepoint.

Another trapdoor generating algorithm presented in [9] is also a combination of choosing a random elliptic curve and checking the existence of a designated point. Therefore this algorithm has the same problem as the above algorithm.

4.2 A new trapdoor generating algorithm

Here we show a new trapdoor generating algorithm over elliptic curves by using a feature that any elliptic curves has many isomorphic elliptic curves.

A new trapdoor generating algorithm

1. Choose an elliptic curve $E/\mathbb{F}_p$

   $$y^2 = x^3 + ax + b (a, b \in \mathbb{F}_p)$$

   such that $\#E(\mathbb{F}_p)$ is divisible by a large prime q and that q is a quadratic residue modulo p. Here we set $u \in \mathbb{F}_p$ such that $u^2 = q \pmod{p}$.

2. Choose $R = (r_x, r_y) \in E(\mathbb{F}_p)$ such that the order of R is q and that r_x is a quadratic residue modulo p. Such R can be constructed a systematicly by finding only one

order-q point: set any $R = (r_x, r_y)$ with the order q. If r_x is a quadratic residue modulo p then output R. If not, compute $2R = (r_{2x}, r_{2y})$ and check whether r_{2x} is a quadratic residue modulo p. If not, compute $3R = (r_{3x}, r_{3y})$ and check r_{3x}. Continue the computation of $nR = (r_{nx}, r_{ny})$ until r_{nx} is a quadratic residue modulo p. Here we set $l \in \mathbb{F}_p$ such that $l^2 = r_x \pmod{p}$.

3. Choose $1 < \forall t < q$ and computes

$$tR = G = (g_x, g_y).$$

Then the order of G is q since t is relatively prime to q.

4. Define an isomorphism φ from E to E_q as follows

$$\varphi : E(\mathbb{F}_p) \ni (x, y) \rightarrow (\frac{q}{r_x}x, \frac{uq}{lr_x}y) \in E_\varphi(\mathbb{F}_p),$$

where $E_\varphi/\mathbb{F}_p : \ y^2 = x^3 + a(\frac{q}{r_x})^2x + b(\frac{q}{r_x})^3$. Then output the elliptic curve E_φ and the basepoint $\varphi(G)$ with a trapdoor $\varphi(R)$ and t.

The above E_φ and $\varphi(G)$ generated in Algorithm 1 has a trapdoor $\varphi(G)$ and t as follows.

Theorem 1 *An elliptic curve E_φ and a basepoint $\varphi(G)$ constructed by the above Algorithm has a trapdoor $\varphi(R)$ and t. That is, the x-coordinate of $\varphi(R)$ is q and $\varphi(G) = t\varphi(R)$.*

proof: We show the elliptic curve E_φ has a trapdoor. Since φ is isomorphism and $\varphi(\mathcal{O}) = \mathcal{O}$, φ is homomorphism([15]). Therefore

1. $\varphi(G) = \varphi(tR) = t\varphi(R)$;
2. both the order of $\varphi(R)$ and $\varphi(G)$ are q.

Furthermore $\varphi(G) = (\frac{q}{r_x}g_x, \frac{uq}{lr_x}g_y)$, and $\varphi(R) = (q, \frac{uq}{lr_x}r_y)$. So the x-coordinate of $\varphi(R)$ is q. This means that $\varphi(R)$ and t is a trapdoor on the elliptic curve E_φ and the basepoint $\varphi(G)$.

Note that the existence of the trapdoor cannot be recognized easily by E_φ and $\varphi(G)$. The coefficients of E_φ are not necessarily divisible by q since the coefficients $a(\frac{q}{r_x})^2$ and $b(\frac{q}{r_x})^3$ are represented in modulo p. Furthermore if we choose a suitable t such as $\frac{q}{r_x}g_x, \frac{uq}{lr_x}g_y > p$, then both x- and y-coordinate of $\varphi(G)$ are not necessarily divisible by q since they are represented in modulo p.

We discuss the running time of the above trapdoor generating algorithm. The above Algorithm requires the next two processes (in addition to the usual process of generating an elliptic curve and a basepoint suitable for cryptosystems): finding q that is a quadratic residue modulo p and finding R of which the x coordinate is a quadratic residue modulo p. Each condition is satisfied by two trials on the average. To sum up, the running time

of the above trapdoor generating algorithm is the same as that of constructing the usual elliptic curves for cryptosystems.

There are many isomorphism for any elliptic curves. From the above algorithm, we see that a suitable isomorphism can set the x-coordinate of a basepoint of elliptic curve cryptosystems to a random quadratic residue number modulo p. In the next section we will show another application of isomorphism defined in Algorithm 2.

4.3 Application of isomorphism over elliptic curves

This section investigates the constructive application of isomorphism over elliptic curves. Some applications require elliptic curves with a basepoint of a small coordinate in order to achieve a fast operation and decrease the size of system parameters([8]). For this purpose, the isomorphism φ defined in Algorithm 1 is useful as follows.

Algorithm constructing elliptic curve with a small basepoint

1. Choose an elliptic curve $E/\mathbb{F}_p$

$$y^2 = x^3 + ax + b(a, b \in \mathbb{F}_p)$$

such that $\#E(\mathbb{F}_p)$ is divisible by a large prime q.

2. Choose $G = (g_x, g_y) \ni E(\mathbb{F}_p)$ such that the order of G is q and that g_x is a quadratic residue modulo p.

Here we set $l \in \mathbb{F}_p$ such that $l^2 = g_x \pmod{p}$.

3. Define an isomorphism φ from E to E_q as follows

$$\varphi : E(\mathbb{F}_p) \ni (x, y) \rightarrow (\frac{1}{g_x}x, \frac{1}{lg_x}y) \in E_\varphi(\mathbb{F}_p),$$

where $E_\varphi/\mathbb{F}_p : \ y^2 = x^3 + a(\frac{1}{g_x})^2x + b(\frac{1}{g_x})^3$. Then output the elliptic curve E_φ and the basepoint $\varphi(G) = (1, \frac{g_y}{lg_x})$.

By the above algorithm, we can construct an elliptic curve and a basepoint with the x-coordinate equal to 1.

Here we discuss the security of EDLP on E_φ and $\varphi(G)$. In step 1 we choose a random E with $q|\#E(\mathbb{F}_p)$ for a quadratic residue q. The only condition added to a usual elliptic curve for EDLP is that q is a quadratic residue. Therefore EDLP on (E, G) in step 1 is the same security as any EDLP. On the other hand, EDLP on $(E_\varphi, \varphi(G))$ is the same security as EDLP on (E, G) in step 1 since $(E_\varphi, \varphi(G))$ is isomorphic to (E, G). To sum up, EDLP on $(E_\varphi, \varphi(G))$ is the same security as any EDLP.

Next we discuss the performance of $(E_\varphi, \varphi(G))$. As for the public key size, the public key size is reduced by 25% since a user's public key mainly consists of a basepoint's x-coordinate, two parameters of an elliptic curve and the user's public key's x-coordinate.

As for the speed of computation of kG, the computation of addition to G, which is required in the computation of kG, is reduced by 10%.

5 Conclusion

In this paper, we have shown a new trapdoor generating algorithm over ECB-attack. We have also shown the running time is the same as that of a usual construction of an elliptic curve and a basepoint suitable for cryptosystems. This algorithm uses a suitable isomorphism over elliptic curves. As an application of the isomorphism we have presented an algorithm constructing elliptic curve with a small basepoint.

Acknowledgements
The author wishes to thank Makoto Tatebayashi for helpful advice.

参考文献

[1] A. O. L. Atkin and F. Morain, "Elliptic curves and primality proving", *Research Report 1256, INRIA*, Juin 1990. Submitted to Math. Comp.

[2] D. Bleichenbacher, "Generating ElGamal signatures without knowing the secret key" to appear in *Advances in Cryptology-Proceedings of EUROCRYPT'96.*

[3] W. Diffie and M. Hellman, "New directions in cryptography" *IEEE Trans. Inform. Theory*, Vol. IT-22 (1976), 644-654.

[4] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. Inform. Theory*, Vol. IT-31 (1985), 469-472.

[5] G. Harper, A. Menezes and S. Vanstone, "Public-key cryptosystems with very small key lengths", *Advances in Cryptology-Proceedings of Eurocrypt'92*, Lecture Notes in Computer Science, **658**(1993), Springer-Verlag, 163-173.

[6] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, **48**(1987), 203-209.

[7] V. S. Miller, "Use of elliptic curves in cryptography", *Advances in Cryptology-Proceedings of Crypto'85*, Lecture Notes in Computer Science, **218**(1986), Springer-Verlag, 417-426.

[8] A. Miyaji, "Elliptic curve over F_p suitable for cryptosystems", *Advances in Cryptology-Proceedings of AUSCRYPT'92*, Lecture Notes in Computer Science, **718**(1993), Springer-Verlag, 479-491.

[9] A. Miyaji, "A message recovery signature scheme equivalent to DSA over elliptic curves ", *Advances in Cryptology-Proceedings of ASIACRYPT'96*, Lecture Notes in Computer Science, **1163**(1996), Springer-Verlag, 1-14.

[10] "Proposed federal information processing standard for digital signature standard (DSS)", *Federal Register*, v. 56, n. 169, 30 Aug 1991, 42980-42982.

[11] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, 80-89, 1991.

[12] K. Nyberg and R. A. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem", *Designs Codes and Cryptography*, **7**(1996), 61-81.

[13] C. P. Schnorr, "Efficient identification and signatures for smart cards", *Advances in cryptology-Proceedings of Crypto'89*, Lecture Notes in Computer Science, **435**(1989), Springer-Verlag, 239-252.

[14] R. Schoof, "Elliptic curves over finite fields and the computation of square roots mod p", *Mathematics of Computation*, vol. 44(1985), 483-494.

[15] J. H. Silverman, *The Arithmetic of Elliptic Curves*, GTM106, Springer-Verlag, New York, 1986.