離散対数問題に基づくメッセージ復元型署名の弱点１

宮地 充子

松下電器産業 (株)　情報通信研究所
〒 571 大阪府 門真市 門真 1006 番
Email:miyaji@isl.mei.co.jp

最近，Nyberg-Rueppel により離散対数問題に基づくメッセージ復元型署名が提案された．本論文では，この署名が攻撃されることを示すとともに，その回避方法について述べる．

離散対数問題，メッセージ復元型署名，攻撃

Weakness in Message recovery signature schemes based on discrete logarithm problems 1

Atsuko Miyaji

Information and Communications Technology Laboratory
Matsushita Electric Industrial Co.,LTD.
1006, Kadoma, Kadoma-shi, Osaka, 571, Japan
Email:miyaji@isl.mei.co.jp

Nyberg and Rueppel recently proposed a new digital signature scheme with message recovery feature([7, 8]). Some variants and their applications on it are discussed([3]). One of their effective applications is the authenticated key exchange. We show these signatures are vulnerable to a known-message attack. This attack becomes serious especially in the authenticated key exchange if a generic chosen-message attack(nonadaptive) is assumed: a forger can exchange an authenticated key successfully. Furthermore we analyze how to defend the attack. We also investigate how to apply the attack to the message recovery signature on an elliptic curve([4, 5]).

discrete logarithm problems, message-recovery signature, attack

1 Introduction

Two signature schemes have received widespread attention: the RSA signature([9]) which is based on the difficulty of factoring large numbers and the ElGamal signature([1]) which is based on the difficulty of taking discrete logarithms. For the ElGamal signature scheme, many variants have been proposed([10, 6]). Here we call these variants of ElGamal based signatures EG-signatures. The main advantage of the RSA signature scheme for EG-signatures has been the feature of massage recovery: a message can be conveyed within the signature and be recovered at the verifier's site. So the message need not be sent along with the signature. The signature with the message recovery feature can be also used in the text hashing mode.

Nyberg and Rueppel recently proposed a new ElGamal based signature scheme with message recovery feature([8]). This signature scheme is used to various applications: identity-based public key cryptosystems, multisignature schemes, and key exchange protocols. Especially the application for key exchange protocol can achieve the authenticated key establishment in one pass transaction.

This paper discusses the security of Nyberg-Rueppel's signature scheme in accordance with [2], in which the notion of the "secure" signature is classified strictly by considering the kinds of attacks and forgeries. In this classification, the most powerful attack possible for an enemy is characterized as "adaptive chosen-message attack": an enemy can use the real signer as "an oracle" (where each chosen message may depend on the signature of previously chosen message). On the other hand, the least damage for a real signer is characterized as "existential forgery": an enemy forges a signature for at least one message over which he has no control. So a signature scheme, which resists an existential forgery using an adaptive chosen-message attack, is considered the most secure signature scheme.

This paper shows, for the first time, that an forger can generate a bogus signature in Nyberg-Rueppel's signature scheme by the following two forgery attacks.

1. an existential forgery using a known-message attack: given a valid signature of a known message, an enemy can forge a valid signature of another different message without the knowledge of the secret key.

2. a universal forgery using a generic chosen-message attack (nonadaptive): given a valid signature of a chosen message which depends only on the system parameters (where the chosen-message is independent of any previous signature), an enemy can forge a valid signature of any message chosen a priori without the knowledge of the secret key.

We also show that a forger can impersonate and exchange an authenticated key successfully with the second attack. Furthermore we analyze why these attacks are made possible and how to improve the signature scheme to avoid the attacks.

The paper is organized as follows. Section 2 summarizes the EG-signatures and the message recovery signature schemes. Section 3 describes the above attacks for the message recovery signature schemes. Section 4 analyzes the attacks and shows how to defend the attack. Section 5 discusses how the attacks are applied for the message recovery signature schemes on an elliptic curve.

2 Message recovery signature scheme

In this section, first we summarize ElGamal based signature scheme. Next we will describe Nyberg-Rueppel's idea by showing one of the message recovery signature schemes. Here we call it NR(p)-signature.

2.1 ElGamal based signature scheme

The trusted authority chooses system parameters, that are a large prime p, a large integer factor q of $p-1$ and an element $g \in \mathbb{Z}_p^*$ whose order is q. Those system parameters are known to all users.

The signer Alice has a secret key x_A and publishes its corresponding public key $y_A = g^{x_A}$. The Alice's signature (r_1, s) of a message $m \in \mathbb{Z}_p^*$ is computed as follows. First she chooses a random number $k \in \mathbb{Z}_q$, and computes

$$\begin{aligned} r_1 &= g^k \pmod p \quad (1)\\ r_1' &= r_1 \pmod q \\ ak &\equiv b + cx_A \pmod q, \quad (2)\end{aligned}$$

where (a, b, c) is a permutation of $(\pm m, \pm r_1', \pm s)$. Then she sends (r_1, s) along with the message m. The signature verification is done by checking the next equation,

$$r_1^a = g^b y_A^c \pmod p. \quad (3)$$

The original ElGamal signature and DSA signature ([6]) are essentially based on the case of $(a, b, c) = (s, m, r_1')$. The coefficient (a, b, c) is further generalized to the Meta-ElGamal signature scheme([3]). The following discussion also holds in the Meta-ElGamal signature.

2.2 Message recovery signature scheme

Here we describe briefly one of the message recovery signature scheme, NR(p)-signature. The Alice's signature (r_2, s) of a message $m \in \mathbb{Z}_p^*$ is computed as follows. First she chooses a random number $k \in \mathbb{Z}_q$, and computes

$$\begin{aligned} r_1 &= g^k \pmod p \quad (4)\\ r_2 &= r_1^{-1} m \pmod p \quad (5)\\ r_2' &= r_2 \pmod q \\ s &\equiv k - x_A r_2' \pmod q. \quad (6)\end{aligned}$$

Then she sends only (r_2, s). The message can be recovered by computing $m = g^s y_A^{r_2'} r_2 \pmod p$ with Alice's public key y_A.

The message recovery signature scheme can be derived generally from EG-signatures replacing m (resp. r_1') by 1 (resp. r_2') in Equation (2). Therefore the signature equation in the message recovery signatures is generally of the form

$$ak \equiv b + cx_A \pmod q, \quad (7)$$

where (a, b, c) is a permutation of $(\pm 1, \pm r_2', \pm s)$. We call this general signature schemes MR(p)-signatures. The description leads to the following six equations if we neglect the $\pm$ signs.

$$\begin{aligned} sk &\equiv 1 + r_2' x_A \pmod q \quad (8)\\ r_2' k &\equiv 1 + s x_A \pmod q \quad (9)\\ k &\equiv s + r_2' x_A \pmod q \quad (10)\\ sk &\equiv r_2' + x_A \pmod q \quad (11)\\ r_2' k &\equiv s + x_A \pmod q \quad (12)\\ k &\equiv r_2' + s x_A \pmod q \quad (13)\end{aligned}$$

NR(p)-signature uses Equation (10) since only Equation (10) does not need inverses both in the signature generation and verification.

The message recovery signature scheme can be suitably used for the key exchange protocol. The key exchange with the message recovery signature scheme can achieve an authenticated and shared secret key by a non-interactive procedure. We describe the details below.

Authenticated key exchange

Let x_B be Bob's secret key and $y_B = g^{x_B} \pmod p$ be his public key. Alice establishes a secret session key with Bob in the following steps:

1. generates a random and secret $K \in \mathbb{Z}_q$.
2. computes $m = g^K \pmod p$.
3. generates the signature (r_2, s) of m with Alice's secret key x_A by a procedure of MR(p)-signatures.
4. sends (r_2, s) to Bob.
5. computes the session key using Bob's public key y_B

$$K_{AB} = y_B^K = g^{x_B K} \pmod p.$$

When receiving (r_2, s) from Alice, Bob

1. recovers m from (r_2, s) with y_A, according to the message recovery procedure of MR(p)-signatures.
2. computes the session key using Bob's secret key x_B

$$K_{AB} = m^{x_B} = g^{K x_B} \pmod p.$$

3 Forgery against NR(p)-signature

Now we describe the forgery protocol against NR(p)-signature. Assume that a forger gets Alice's signature (r_2, s) for a message m. Then the forger can compute a signature $(\tilde{r_2}, \tilde{s})$ for a message $\tilde{m}$ without the knowledge of Alice's secret key by the following procedure:

1. computes $\tilde{r_1} = (mr_2^{-1})g^{-1} = r_1 g^{-1} = g^{k-1} \pmod p$.
2. sets a message $\tilde{m} = mg^{-1} \pmod p$, $\tilde{r_2} = r_2$ and $\tilde{s} = s - 1$.
3. sends $(\tilde{r_2}, \tilde{s})$ as a signature of $\tilde{m}$.

We see that $(\tilde{r_2}, \tilde{s})$ is a valid signature of $\tilde{m}$ since

$$\begin{aligned} g^{\tilde{s}} y_A^{\tilde{r_2}^{\theta}} \tilde{r_2} &= g^{s-1} y_A^{r_2^{\theta}} r_2 \\ &= g^s y_A^{r_2^{\theta}} r_2 g^{-1} \\ &= mg^{-1} \\ &= \tilde{m} \pmod p. \end{aligned}$$

By this procedure, a forger can make the signature on a message mg^{-1}. This attack can be easily extended to a stronger attack: a forger can generate a signature for any message in a subset $S_{m,g} = \{mg^{-n} | n \in \mathbb{Z}_q\}$, within one time known-message attack. Furthermore if we assume a generic chosen-message attack, the forger can generate the signature of any intentional message.

We will show the most serious scenario by this attack: a forger can impersonate Alice and exchange an authenticated key successfully with Bob.

Applying the attack to key exchange

We assume that a forger obtains Alice's signature of a message $m \in S_g = \{g^l | l \in \mathbb{Z}_q\}$ that he chooses. Clearly the subspace S_g is generic since it depends only the system parameter. Note that a valid signature is required only for one message in the subspace S_g. Here we set the message $m = g^l$ and the signature (r_2, s). Then the forger selects a random $x \in \mathbb{Z}_q$, generates the signature $(\tilde{r_2}, \tilde{s})$ for $\tilde{m} = mg^{x-l}$ by the above procedure and sends $(\tilde{r_2}, \tilde{s})$ to Bob. Next the forger computes,

$$K = y_B^x = g^{x_B x},$$

where y_B is Bob's public key. On the other hand, Bob receives $(\tilde{r_2}, \tilde{s})$ and first recovers $m = g^x$ using Alice's public key. Next he computes the shared key K using his secret key x_B, where

$$K = m^{x_B} = g^{x x_B}.$$

Bob is certain that m is the Alice's authentic message since m is recovered with Alice's public key, according to the message recovery procedure. Therefore Bob believes that K is the Alice's authenticated key. The forger can impersonate Alice and send a message to Bob successfully.

We have shown the generic chosen-message attack in the key exchange scenario. In the same way, a forger can use the generic chosen-message attack in the message-sending scenario. Note that the scope of this attack is all the message space $\mathbb{Z}_p^*$.

4 Analyzing the attack

The easiest method to avoid the attack in Section 3 would be to use NR(p)-signature for the hash value of a message: a message is also sent along with the signature and its hash value is recovered. Apparently this method spoils the message recovery feature. In this section, we analyze the attack and improve the method to avoid the attack, while maintaining the message recovery feature. We will also show which schemes of MR(p) and EG-signatures are vulnerable to this attack.

4.1 Essence of the attack

The attack consists of two essential attacks. One is an existential forgery using a known-message attack: given a valid signature of a known message, an enemy can forge a valid signature of another different message. The other is a universal forgery using a chosen-message attack (nonadaptive): given a valid signature of a message chosen beforehand, an enemy can forge a valid signature of any message chosen a priori. The essence of the former forgery is that a different $\tilde{r_1}$ satisfying the signature congruence-equation can be constructed by modifying the original r_1. That of the latter forgery is that the chosen-message can be constructed using the homomorphism feature of a function in the signature procedure. Let us call these two attacks a congruence-equation attack and a homomorphism attack respectively. The following discussion will show why we call them as above.

4.2 Congruence-equation attack

The congruence-equation attack is divided into two cases: the attack using the basepoint "g" and the attack using Alice's public key "y_A". The attack in Section 3 is the former case. We will discuss each case.

The congruence-equation attack using the basepoint

Assume that a forger gets Alice's signature (r_2, s) for a message m in MR(p)-signatures. Here we call $r_1 (= g^{\mathsf{k}})$ a commitment, where k is unknown to the forger. Then the forger can always construct a new commitment $\tilde{r_1} = r_1/g = g^{\mathsf{k}-1}$. He does not know the correct discrete logarithm of $\tilde{r_1}$ but more importantly he knows it is equal to the value subtracted by 1 from the discrete logarithm of r_1. First he converts Equation(7) standing for the original $m(= r_1 r_2), r_2, s$ and k to that for the new $\tilde{m}(= \tilde{r_1}\tilde{r_2}), \tilde{r_2}, \tilde{s}$ and $k-1$, maintaining the original signature congruence-equation: he tries to find $(\tilde{m}, \tilde{r_2}, \tilde{s})$ satisfying the following equation,

$$a(k-1) \equiv (b-a) + cx_{\mathsf{A}} \pmod{q}, \tag{14}$$

where $(a, b-a, c)$ is a permutation of $(1, \tilde{r_2}', \tilde{s})$. In fact such $(\tilde{r_2}, \tilde{s})$ is the valid signature of $\tilde{m} = \tilde{r_1}\tilde{r_2}$ since they satisfy the signature equation for a new commitment $\tilde{r_1} = g^{\mathsf{k}-1}$.

Let us go back to the discussion of Equation (14). We see that either $\tilde{r_2}$ or $\tilde{s}$ must be kept the same as the original r_2 or s since two coefficients a and c are fixed. Therefore we see that the congruence-equation attack succeeds if and only if we use the schemes of $b = s$ or $b = r_2$ in MR(p)-signatures (7): the cases of (10) and (12), or the cases of (11) and (13) respectively. First we investigate the case of $b = s$. Then we set $\tilde{r_2} = r_2$. A new $\tilde{m}$ can be set as

$$\tilde{m} = \tilde{r_1} r_2 = g^{\mathsf{k}-1} r_2 = m/g,$$

using the original r_2. Note that $\tilde{m}$ can be represented by m and g. Then he can generate the signature of $\tilde{m} = m/g$ by setting $\tilde{s} = s - 1$ or $\tilde{s} = s - r_2'$ in the case of Equation (10) or (12). Next we investigate the case of $b = r_2$ in MR(p)-signatures. Then we set $\tilde{s} = s$ and $\tilde{r_2} = r_2' - s \pmod{q}$ (resp. $\tilde{r_2} = r_2' - 1 \pmod{q}$) in the case of Equation (11) (resp. (13)). Then $(\tilde{r_2}, \tilde{s})$ is a valid signature of $\tilde{m} = \tilde{r_1}\tilde{r_2}$. Note that, in this case of setting $\tilde{s} = s$ (i.e. $\tilde{r_2} \neq r_2$), $\tilde{m}$ cannot be represented only by m and the known data like g and y_{A}.

In the case of EG-signature, we investigate whether there exists $(\tilde{m}, \tilde{s})$ satisfying Equation (14), where $(a, b-a, c)$ is a permutation of $(\tilde{m}, \tilde{r_1}', \tilde{s})$. Note that also in this case, a new commitment $\tilde{r_1}$ is set to $\tilde{r_1} = r_1/g$. As we have seen in the case of MR(p)-signature, two coefficients a and c must be fixed. Namely two variables of $(\tilde{m}, \tilde{r_1}, \tilde{s})$ must be kept the same as the original. Since $\tilde{r_1} \neq r_1$, he must keep $\tilde{m}$ the same as m. Therefore he cannot forge a message in EG-signatures.

The congruence-equation attack using Alice's public key

The above attack uses a power of a basepoint g in order to modify the commitment original $r_1 = g^{\mathsf{k}}$. Considering the signature equation (7), we will see that a forger can also use a power of Alice's public key $y_{\mathsf{A}} = g^{\mathsf{x_A}}$.

In this case, the forger constructs a different commitment $\tilde{r_1} = r_1/y_{\mathsf{A}} = g^{\mathsf{k}-\mathsf{x_A}}$. He does not know the correct discrete logarithm of $\tilde{r_1}$ but more importantly he knows it is equal to the value subtracted by x_{A} from the discrete logarithm of r_1. Since the following discussion is almost the same as the above case of "the attack using the basepoint", we will describe briefly. First he converts Equation(7) standing for the original $m(= r_1 r_2), r_2, s$ and k to that for the new $\tilde{m}(= \tilde{r_1}\tilde{r_2}), \tilde{r_2}, \tilde{s}$ and $k - x_{\mathsf{A}}$: he tries to find $(\tilde{m}, \tilde{r_2}, \tilde{s})$ satisfying the following equation,

$$a(k - x_{\mathsf{A}}) \equiv b + (c-a)x_{\mathsf{A}} \pmod{q}, \tag{15}$$

where $(a, b, c-a)$ is a permutation of $(1, \tilde{r_2}', \tilde{s})$. Therefore the congruence-equation attack in this case succeeds if and only if we use the schemes of $c = s$ or $c = r_2'$ in MR(p)-signature: the cases of signature equations (9) and (13), or (8) and (10) respectively. In the case of $c = s$, he can generate the signature of

$$\tilde{m} = \tilde{r_1} r_2 = g^{\mathsf{k}-\mathsf{x_A}} r_2 = m/y_{\mathsf{A}}$$

by setting $\tilde{r_2} = r_2$ and $\tilde{s} = s - r'_2$ (resp. $\tilde{s} = s - 1$) in the case of Equation (9) (resp. (13)). Note that $\tilde{m}$ can be represented by m and y_{A}. On the other hand in the case of $c = r'_2$, he sets $\tilde{s} = s$ and $\tilde{r_2} = r'_2 - s \pmod q$ (resp. $\tilde{r_2} = r'_2 - 1 \pmod q$) in the case of Equation (8) (resp. (10)). Then $(\tilde{r_2}, s)$ is a valid signature of $\tilde{m} = \tilde{r_1}\tilde{r_2}$. Note that, in this case of setting $\tilde{s} = s$ (i.e. $\tilde{r_2} \neq r_2$), $\tilde{m}$ cannot be represented only by m and the known data like g and y_{A}. As for EG-signatures, this attack are not applicable in the same reason as the above.

The important point of the congruence-equation attack is that a forger can construct the signature equation standing for a new commitment $\tilde{r_1}$ by converting the original signature equation while maintaining its congruity. Therefore the congruence-equation attack succeeds if and only if exactly two coefficients of the converted signature equation (14) and (15) are kept the same as the original coefficients (i.e. the cases of $b \neq 1$ and the cases of $c \neq 1$ in Equation (7)). So all the signature equation (8)$\sim$(13) are vulnerable to this attack by using either (14) or (15). On the other hand, EG-signatures are strong against the congruence-equation attack. But the congruence equation attack is "existential forgery". So this is typically avoided by introducing an additional step: adding redundancy to the message before it is signed and through checking the redundancy after recovery.

4.3 Homomorphism attack

The homomorphism attack extends the congruence-equation attack to forge any message. We cannot avoid the homomorphism attack by adding redundancy. As we have seen in the previous section, all the cases (8) $\sim$ (13) in MR(p)-signatures are vulnerable to the congruence-equation attack. First we investigate which cases of the congruence-equation attack are extended to the homomorphism attack.

We divide the congruence-equation attack into two cases: one case is keeping $\tilde{r_2}$ the same as the original and another case is not. The cases of (10) and (12), or (9) and (13) can set $\tilde{r_2} = r_2$ in "the attack using the basepoint" or in "the attack using Alice's public key" respectively. On the other hand, (8) and (11) cannot set $\tilde{r_2} = r_2$ in both the attack using the basepoint and Alice's public key. There is serious difference between them. In the former case of setting $\tilde{r_2} = r_2$, a forged message $\tilde{m}$ is independent of the original signature (r_2, s). In fact an enemy can forge any message $\tilde{m}$ belonging to a subset $S_{\mathsf{m,g}}$ or $S_{\mathsf{m,y_A}} = \{my_{\mathsf{A}}^{-\mathsf{n}} | n \in \mathbb{Z}_{\mathsf{q}}\}$ with one pair of (m, r_2, s). These forged message subsets $S_{\mathsf{m,g}}$ and $S_{\mathsf{m,y_A}}$ are independent of the original signature (r_2, s): they are determined only by a message m and a known data like the basepoint or the public key. The bogus signature on each message is given for each (9), (10), (12) and (13) as follows,

$$\begin{aligned}
\{(my_{\mathsf{A}}^{-\mathsf{n}}, r_2, s - nr'_2) \quad &| \quad my_{\mathsf{A}}^{-\mathsf{n}} \in S_{\mathsf{m,y_A}}\},\\
\{(mg^{-\mathsf{n}}, r_2, s - n) \quad &| \quad mg^{-\mathsf{n}} \in S_{\mathsf{m,g}}\},\\
\{(mg^{-\mathsf{n}}, r_2, s - nr'_2) \quad &| \quad mg^{-\mathsf{n}} \in S_{\mathsf{m,g}}\}, \text{and}\\
\{(my_{\mathsf{A}}^{-\mathsf{n}}, r_2, s - n) \quad &| \quad my_{\mathsf{A}}^{-\mathsf{n}} \in S_{\mathsf{m,y_A}}\}.
\end{aligned}$$

On the other hand in the latter case $\tilde{r_2} \neq r_2$, $\tilde{m}$ depends on both the original m and the signature (r_2, s) as we have seen in Section 4.2.

It is necessary for extending the congruence-equation attack that the forged $\tilde{m}$ can be represented only by the original m and the known data as we will see below. So the homomorphism attack is serious in the cases of $a \neq s$ in Equation(7), that are Equations (10) and (12) in the congruence-equation attack using the basepoint, and, (9) and (13) in congruence-equation attack using the public key, where (10) is just the signature equation in NR(p)-signature.

We will investigate why a forger can preselect the forged message subset in the cases of (9), (10), (12) and (13). For simplicity, we deal only with the cases of (10) and (12). The following discussion also holds in the other cases.

Here we assume that a forger wants to generate the signature $(\tilde{r_2}, \tilde{s})$ of an intentional message $\tilde{m} \in \mathbb{Z}_p$. First a forger tries to construct m which enables him to forge a message $\tilde{m}$, whatever the Alice's signature for m is. This condition is written as follows:

$$\tilde{m}(= \tilde{r_2}\tilde{r_1}) = \exists\varphi(m, g, p, q, y_A), \tag{16}$$

where φ is a suitable function to $\mathbb{Z}_p$ such that $m = \varphi^{-1}(\tilde{m}, g, p, q, y_A)$ exists. This means that $\tilde{m}$ is independent of the parameters k, r_2 and s which the signer Alice can take arbitrarily. Then he sets

$$m = \varphi^{-1}(\tilde{m}, g, p, q, y_A)$$

and has this message m signed by Alice. For example in the cases (10) and (12), a chosen-message m for an intentional $\tilde{m}$ is constructed by setting $m = \varphi^{-1}(\tilde{m}, g)$, where

$$\tilde{m} = r_2\tilde{r_1} = r_2 r_1 g^{-1} = mg^{-1} = \varphi(m, g).$$

After getting the Alice's signature on the above m, he can forge an intentional message $\tilde{m} = mg^{-1}$. Generally any message $m \in S_{\tilde{m},g} - \{\tilde{m}\}$ will do to forge an intentional $\tilde{m}$. Of course we need only one valid signature on such a chosen-message m.

Next we investigate the condition that $\tilde{m}$ is not denoted by $\exists\varphi$ in Equation (16). Since the relation between m and $\tilde{m}$ is determined by Equation (5), we change Equation (5) so as for $\tilde{m}$ not to be represented only by m and g. First we change Equation (5) to

$$r_2 = f(r_1)^{-1}m \pmod{p}, \tag{17}$$

where $f : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$ is a map known to all users. Note that the congruence-equation attack still holds: a forger can generate the signature on $\tilde{m} = r_2 f(r_1/g)$ using a known-message and the signature (m, r_2, s). So we must find f such that it prevents this further attack, the homomorphism attack, from forging an intentional message. The relation between m and the forged $\tilde{m}$ is represented as the following equation

$$\tilde{m} = r_2 f(g^{k-1}) = mf(g^{k-1})/f(g^k). \tag{18}$$

If f is a homomorphism, then Equation (18) leads

$$\tilde{m} = m/f(g) = \varphi(m, g).$$

So he can forge an intentional $\tilde{m}$ after having $m = \tilde{m}f(g)$ signed by Alice. In MR(p)-signatures, we can regard a map f as an identity map. Since an identity map is a homomorphism, (10) and (12) in MR(p)-signatures are vulnerable to the homomorphism attack. Therefore we see that the homomorphism attack succeeds if and only if the term k is cancelled in Equation (18): f has a homomorphism-like feature. So we must set f such that the term k cannot be cancelled in Equation (18). Then the above φ does not exist between $\tilde{m}$ and m. Here is one example.

Example: Define

$$\begin{array}{rccc} f: & \mathbb{Z}_p & \longrightarrow & \mathbb{Z}_p \\ & x & \longrightarrow & x + g. \end{array}$$

Then Equation (18) is

$$\tilde{m} = mf(g^{k-1})/f(g^k) = m(g^{k-2} + 1)/(g^{k-1} + 1).$$

So the term k is not cancelled. Therefore a forger cannot generate the signature of an intentional message. Apparently f has a good feature that the computation amount is negligible.

There are two points in the homomorphism attack: one is that the forged message by the congruence-equation attack can be represented by the original message and the known data, and the other is that the map f has the homomorphism-like feature. Therefore the cases of (8) and (11) are strong against the homomorphism attack, where (8) is the signature equation which adds the message recovery feature to the original ElGamal signature scheme. On the other hand Equation(9), (10), (12) and (13) are vulnerable to the homomorphism attack. As we have discussed in Section 2, Equation (10) is optimal with regard to the computation amount: it does not need inversions both in the signature generation and verification. But this signature equation cannot avoid the congruence-equation attack and the more serious attack, the homomorphism attack. In order to avoid the homomorphism attack, we have seen that we must change Equation (5) to Equation (17) using a suitable function like the above example. Then an enemy can not forge a signature on an intentional message.

5 Message recovery signature on an elliptic curve

The ElGamal based signature schemes can be constructed over an elliptic curve. So the message recovery feature can be added to ElGamal based signature on an elliptic curve. We will see how the previous attacks are applied to the elliptic curve message recovery signature. First we describe the message recovery signature using an elliptic curve. In this case the system parameters are: an elliptic curve $E/\mathbb{Z}_{\mathsf{p}}$, a basepoint $G \in E(\mathbb{Z}_{\mathsf{p}})$ and the order q of G. The signer Alice has a secret key x_{A} and publishes the corresponding public key $Y_{\mathsf{A}} = x_{\mathsf{A}}G$. The procedure for Alice to make a signature of $m \in \mathbb{Z}_{\mathsf{p}}^*$ is as follows. First she picks a random number $k \in \mathbb{Z}_{\mathsf{p}}$, and computes

$$\begin{aligned} R_1 &= kG = (r_{1\mathsf{x}}, r_{1\mathsf{y}}), && (19)\\ r_2 &= r_{1\mathsf{x}}^{-1} m \pmod{p}, && (20)\\ r_2' &= r_2 \pmod{q}, \\ ak &\equiv b + cx_{\mathsf{A}} \pmod{q}, && (21) \end{aligned}$$

where (a, b, c) is a permutation of $(\pm 1, \pm r_2', \pm s)$ and Equation (19) is computed in E. Then she outputs the signature (r_2, s) to Bob.

The message can be recovered by computing

$$m = x(\frac{b}{a}G + \frac{c}{a}Y_{\mathsf{A}})r_2 \pmod{p},$$

where $\frac{\mathsf{b}}{\mathsf{a}}G + \frac{\mathsf{c}}{\mathsf{a}}Y_{\mathsf{A}}$ is computed in E and $x(\frac{\mathsf{b}}{\mathsf{a}}G + \frac{\mathsf{c}}{\mathsf{a}}Y_{\mathsf{A}})$ denotes the x-coordinate of $\frac{\mathsf{b}}{\mathsf{a}}G + \frac{\mathsf{c}}{\mathsf{a}}Y_{\mathsf{A}}$.

Here we call the above general message recovery signature schemes with the elliptic curves MRE(p)-signatures. The case of $(a, b, c) = (1, s, r_2')$ is the elliptic curve version of NR(p)-signature. Here we call it NRE(p)-signature.

Let us apply the attack in Section 3 to MRE(p)-signature. For the congruence-equation attack, the same discussion on MR(p)-signature also follows in MRE(p)-signature. Therefore all MRE(p)-signatures are vulnerable to the congruence-equation attack. So we have to add redundancy to a message before it is signed.

For the homomorphism attack, we must investigate the cases of $a \neq s$ in Equation (21) in the same way as MR(p)-signature. There is a difference between MRE(p)-signature and MR(p)-signature in the term r_2. In MR(p)-signature, the commitment r_1 is related to r_2 directly in Equation (5) (i.e. f is an identity map in Equation (17)). On the other hand, in MRE(p)-signature, R_1 is not related to r_2 directly. In fact, the map f in Equation (17) is the x-coordinate function as follows,

$$r_2 = x(R_1)^{-1} m \pmod{p}.$$

The x-coordinate function on E, whatever an elliptic curve E is chosen, has not homomorphism-like feature:

$$x((k-1)G)/x(kG) = x(kG-G)/x(kG) = \varphi(k, G),$$

where the term k is not cancelled in φ. Therefore all MRE(p)-signatures are strong against the homomorphism attack.

As we have seen in Section 2, the most effective signature equation is used in NR(p)-signature. Also NRE(p)-signature has the same advantage. Both NR- and NRE(p)-signatures are vulnerable to the congruence-equation attack. For the homomorphism attack, NRE(p)-signature are strong. But in NR(p)-signature, we have seen that we have to change Equation (5) in order to preventing the homomorphism attack.

6 Conclusion

We have shown Nyberg-Rueppel's signature is vulnerable to two forgery attacks: the congruence-equation attack and the homomorphism attack. We have applied these attacks to MR(p)-, EG- and MRE(p)-signatures generally and shown that:
1. all the cases of (8) $\sim$ (13) in MR(p)-signatures are vulnerable to the congruence-equation attack.
2. the cases of (9), (10), (12) and (13) are vulnerable to the homomorphism attack, where (10) is the case of the NR(p)-signature.
3. the cases of (8) and (11) are strong against the homomorphism attack, where (8) is the case of adding the message recovery feature to the original ElGamal signature scheme.
4. the homomorphism attack becomes invalid by changing Equation (5) to (17) using a suitable function.
5. EG-signatures are strong against both the congruence-equation attack and the homomorphism attack.
6. MRE(p)-signatures are vulnerable to the congruence-equation attack.
7. MRE(p)-signatures are strong against the homomorphism attack.

Acknowledgements
The author would like to thank Tatsuaki Okamoto for helpful conversations. The author wishes to thank Makoto Tatebayashi and Natsume Matsuzaki for helpful advice.

References

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

[2] S. Goldwasser, S. Micali and R. L. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks", *SIAM J. Computing*, 17, 2(1988), 281-308.

[3] P. Horster, M. Michels and H. Petersen "Meta-Message Recovery and Meta-Blind signature schemes based on the discrete logarithm problem and their applications", *Advances in Cryptology-Proceedings of Asiacrypt'94*, Lecture Notes in Computer Science, to appear.

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

[5] 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.

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

[7] K. Nyberg and R. A. Rueppel "A new signature scheme based on the DSA giving message recovery", *Proceedings of 1st ACM Conference on Computer and Communications Security*, 1993.

[8] K. Nyberg and R. A. Rueppel "Message recovery for signature schemes based on the discrete logarithm problem", *Advances in Cryptology-Proceedings of Eurocrypt'94* , Lecture Notes in Computer Science, to appear.

[9] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, vol.21, No.2(1978), 120-126.

[10] 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.