離散対数問題に基づくメッセージ復元型署名の弱点 2

宮地 充子

松下電器産業 (株)　マルチメディア開発センター
〒 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とその変型が提案された ([7, 8])．彼らは 2 つの攻撃の存在を指摘しているが，これらが全変型にどのように適応されるかについては未検討であった．著者は [4] において，新たな攻撃を 2 つ指摘するとともに，これら 2 つの攻撃及び彼らにより指摘されていた 1 つの攻撃がどのように全変型に適応されるかについて述べた．本論文では，この署名の新たな攻撃をさらに 2 つ示すとともに，それがどのように全変型に適応されるか述べる．また彼らにより指摘されていた残りの一つの攻撃の適応性についても述べる．

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 2

Atsuko Miyaji

Multimedia Development Center
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 ElGamal-type digital signature scheme with message recovery feature and its six variants([7, 8]). They also pointed out two forgeries against some of their signatures. But they did not investigate explicitly how to apply these forgeries to all variants including elliptic curves. The author presented new two forgeries and investigated deeply how to apply the two forgeries and already presented one forgery on all variants([4]). In this paper, we present the further two forgeries and investigate explicitly how to apply the two forgeries and the other presented forgery on all variants.

discrete logarithm problems, message-recovery signature, attack

1 Introduction

The RSA signature([9]), which is based on the difficulty of factoring, has a message recovery feature. On the other hand, the ElGamal signature([2]) and its six variants([10, 1]), which are based on the difficulty of the discrete logarithm problem, do not have a message recovery feature. Here we call them EG-signatures. Recently Nyberg and Rueppel proposed a method to add the message recovery feature to all EG-signatures([7, 8]). The Nyberg-Rueppel's signatures can achieve the authenticated key exchange in one pass transaction. They also pointed out two forgeries against their signatures, which are called the signature-equation attack using basepoint, and the recovery-equation attack using basepoint in this paper. However they did not investigate explicitly how to apply these forgeries to all variants including elliptic curves. In fact, they did not investigate how to apply the recovery-equation attack using basepoint on an elliptic curve message recovery scheme.

Recently the author showed further two forgeries against Nyberg-Rueppel signatures ([4]). Furthermore she investigated how to apply the two forgeries and the signature-equation attack using basepoint to all Nyberg-Rueppel's signatures.

In this paper, we show the further new two forgeries. We also investigate how to apply the two forgeries to all Nyberg-Rueppel's signatures including elliptic curves. Furthermore we investigate how to apply the recovery-equation attack using basepoint on elliptic curve variants.

The paper is organized as follows. Section2 summarizes EG-signatures and the message recovery signature schemes. Section3 describes the above attacks for Nyberg-Rueppel's signatures. Section4 discusses how the attacks are applied for the message recovery signature schemes on the elliptic curves.

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

$$r_1 = g^k \pmod{p} \tag{1}$$

$$\begin{aligned} r_1' &= r_1 \pmod q \\ ak &\equiv b + cx_A \pmod q, \end{aligned} \tag{2}$$

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,

$$r_1^a = g^b y_A^c \pmod p. \tag{3}$$

The original ElGamal signature ([2]) and DSA signature ([1]) are essentially based on the case of $(a, b, c) = (s, m, r_1')$.

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

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

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. We call (5) the message-mask equation.

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

$$ak \equiv b + cx_A \pmod q, \tag{7}$$

where (a, b, c) is a permutation of $(\pm 1, \pm r_2', \pm s)$. The message is recovered by computing the recovery equation

$$m = g^{b/a} y_A^{c/a} r_2 \pmod p. \tag{8}$$

We call this general signature schemes MR(p)-signatures. The description leads to the following six equations if we neglect the $\pm$ signs.

$$\begin{aligned} sk &\equiv 1 + r_2' x_A \pmod q && (9) \\ r_2' k &\equiv 1 + s x_A \pmod q && (10) \\ k &\equiv s + r_2' x_A \pmod q && (11) \\ sk &\equiv r_2' + x_A \pmod q && (12) \\ r_2' k &\equiv s + x_A \pmod q && (13) \\ k &\equiv r_2' + s x_A \pmod q && (14) \end{aligned}$$

NR(p)-signature uses Equation (11) since only Equation (11) does not need inverses both in the signature generation and verification.

3 Forgery against MR(p)-signature

First we describe new two forgery protocols against NR(p)-signature. Next we investigate how to apply them on all MR(p)-signatures.

3.1 The redundancy attack

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 way:

1. computes $mr_2^{-1} = r_1 (= g^k) \pmod{p}$.
2. chooses any number $n \in \mathbb{Z}_p$ such that $\tilde{r_2} = r_2' + nq \neq r_2$. (There are about $|p/q|$ variants.)
3. sets a message $\tilde{m} = \tilde{r_2} r_1 \pmod{p}$ and $\tilde{s} = s$.
4. sends $(\tilde{r_2}, \tilde{s})$ as a signature of $\tilde{m}$.

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

$$g^{\tilde{s}} y_A^{\tilde{r_2}} \tilde{r_2} = g^s y_A^{r_2} \tilde{r_2} = g^k \tilde{r_2} = \tilde{m} \pmod{p}.$$

The redundancy attack utilizes the next facts: 1. there is redundancy between the real value $r_2 \in \mathbb{Z}_p$ determined in Equation (5) and the necessary value r_2' for Equation (7); 2. the redundancy enables a forger to construct a new message $\tilde{m}$ and the valid signature $(\tilde{r_2}, \tilde{s})$ by setting $\tilde{r_2} \equiv r_2 \pmod{q}$ (and $\tilde{r_2} \neq r_2$), $\tilde{s} = s$ and $\tilde{m} = r_1 \tilde{r_2}$ (using Equation (5)). Since these are facts in all MR(p)-signatures, they all are vulnerable to the redundancy attack.

From the above discussion, the methods to avoid the redundancy attack are as follows: 1. limit r_2 to $0 < r_2 < q$ in signature generation by setting $p \approx q$ (which may require repeated trials of the random number k) and reject the signature in message recovery if $r_2 \geq q$.
2. change the message space $\mathbb{Z}_p$ to $\mathbb{Z}_q$ (that is $\mathbb{Z}_q$-message recovery signature) so that a forger cannot construct a new $\tilde{m}$ in Equation (5). Then r_2 has only to be determined in $\mathbb{Z}_q$. So Equation (5) can be replaced

$$r_2 = (r_1 \pmod{q})^{-1} m \pmod{q}. \tag{15}$$

The most effective application, the authenticated key exchange, requires $\mathbb{Z}_p$-message recovery feature. Therefore only the former method is preferable. Generally we set $|p| \gg |q|$ in order to reduce the signature size. So the former method avoids the redundancy attack but actually increases the signature size.

Next we apply the redundancy attack to EG-signatures. In this case, the attack tries to do fact1 and 2 by changing $\tilde{r_2}$ in each fact to $\tilde{r_1}$ and the signature equation (7) in fact1 to (2) with a given message and the signature (r_1, s, m). For the fact1, the above discussion

also holds in EG-signatures. So a forger can take a new $\tilde{r_1}$ which does not equal r_1 in $\mathbb{Z}_p$ but equal r_1 in $\mathbb{Z}_q$. On the other hand, the fact2 is invalid for EG-signatures since the relation equation between m and a commitment r_1, the message-mask equation (5), is not required in EG-signatures. Therefore all EG-signatures are strong against the redundancy attack.

3.2 The recovery-equation attack using y_A

The recovery-equation attack using y_A is constructed as well by changing the function of g in the attack presented in [8] to y_A. This forgery can compute a signature (r_2, s) on a message of the form $m = My_A^e$ for any chosen $M \in \mathbb{Z}_p$ without ever seeing any signature and Alice's secret key. However all MR(p)-signatures are not necessarily vulnerable to the recovery-equation attack. We deal with the signature equation (7) and investigate which schemes are vulnerable.

The recovery equation for the above message $m = My_A^e$ is

$$r_2 = (My_A^e)g^{-b/a}y_A^{-c/a}, \tag{16}$$

and we search its solution r_2, s, and e. First we set

$$r_2 = Mg^U y_A^V, \tag{17}$$

for any chosen $U, V \in \mathbb{Z}_q$. Next we investigate each signature scheme.
1. the schemes $a = r_2'$ (i.e. (10) and (13))
Then we can solve $U = -b/a$ for b if and only if $b = s$. Therefore the scheme (10) is strong. But in scheme (13), we can forge m by setting $s = -Ur_2'$ and $e = V + 1/r_2'$.
2. the schemes $a = s$ (i.e. (9) and (12))
Then we can solve $U = -b/a$ for c in each case. In fact, in scheme (9) we can forge m by setting $s = -1/U$ and $e = V + r_2'/s$. In scheme (12), we can forge m by setting $s = -r_2'/U$ and $e = V + 1/s$
3. the schemes $a = 1$ (i.e. (11) and (14))
Then we can solve $U = -b/a$ for b if and only if $b = s$. Therefore the scheme (14) is strong. But in scheme (11), we can forge m by setting $s = -U$ and $e = V + r_2'$.

To sum up, the signature schemes (9), (11), (12), and (13) are vulnerable to the recovery-equation attack using y_A, but (10) and (14) are strong.

4 Message recovery signature on 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 attack is 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 $E/\mathbb{Z}_p$, a basepoint $G \in E(\mathbb{Z}_p)$ and the order q of G.

The signer Alice has a secret key x_A and publishes the corresponding public key $Y_A = x_A G$. The procedure for Alice to make a signature of $m \in \mathbb{Z}_p^*$ is as follows. First she picks a random number $k \in \mathbb{Z}_p$, and computes

$$\begin{aligned} R_1 &= kG = (r_{1x}, r_{1y}), && (18)\\ r_2 &= r_{1x}^{-1} m \pmod{p}, && (19)\\ r_2' &= r_2 \pmod{q}, \\ ak &\equiv b + cx_A \pmod{q}, && (20) \end{aligned}$$

where (a, b, c) is a permutation of $(\pm 1, \pm r_2', \pm s)$ and Equation (18) is computed in E. Then she outputs the signature (r_2, s) to Bob.

The message can be recovered by computing

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

where $\frac{b}{a}G + \frac{c}{a}Y_A$ is computed in E and $x(\frac{b}{a}G + \frac{c}{a}Y_A)$ denotes the x-coordinate of $\frac{b}{a}G + \frac{c}{a}Y_A$. We call the above elliptic curve schemes MRE(p)-signatures. We investigate how to apply the redundancy attack, the recovery-equation attack using basepoint and public key on MRE(p)-signatures.

The redundancy attack

In MRE(p)-signature, the size $|p|$ of an elliptic curve E/F_p chosen carefully can be reduced to the size $|q|$ of the order of the basepoint G ([5]). Therefore MRE(p)-signatures avoid this attack easily by limiting r_2 to $0 < r_2 < q$ in signature generation and rejecting the signature in message recovery if $r_2 \geq q$. Of course it may require repeated trials of the random number k in signature generation. In order to avoid the repeated trials, we suggest to use an elliptic curve over $\mathbb{Z}_p$ with p-elements ([3]). The redundancy attack has no impact on $E/\mathbb{Z}_p$ with p-elements at all since the order q of the basepoint G is equal to p.

The recovery-equation attack using basepoint and public key

The recovery-equation attack using basepoint or public key forges a type of message, a power of basepoint or a power of public key. Since in MRE(p)-signatures these are not $\mathbb{Z}_p$-elements, such message does not exist. Therefore both the recovery-equation attack using basepoint and public key do not work for MRE(p)-signatures at all. If a forger forces to apply the attack using basepoint, then he sets

$$r_2 = M\, x(eG) x(uG + vY_A)^{-1}, \qquad (21)$$

for $u, v \in \mathbb{Z}_q$ and must solve $b/a = u$ and $c/a = v$ for s. Apparently it is impossible! The same also holds in the attack using public key.

5 Conclusion

We have shown two forgeries, the redundancy attack and the recovery-equation attack using public key, for Nyberg-Rueppel's signature. We have applied these attacks to MR(p)-

and MRE(p)-signatures generally. Especially for MRE(p)-signatures, we have applied the recovery-equation attack using basepoint as well. The results are as follows:
1. all MR(p)-signatures are vulnerable to the redundancy attack.
2. the redundancy attack becomes invalid by setting the size of basepoint to be same as that of the finite field.
3. EG-signatures are strong against the redundancy attack.
4. MRE(p)-signatures are strong against the redundancy attack.
5. the schemes (9), (11), (12), and (13) in MR(p)-signatures are vulnerable to the recovery-equation attack using public key.
6. the schemes (10) and (14) are strong against the recovery-equation attack using public key.
7. MRE(p)-signatures are strong against the recovery-equation attack using public key and basepoint.

References

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

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

[3] A. Miyaji, "On ordinary elliptic curves", *Advances in Cryptology-Proceedings of ASIACRYPT'91*, Lecture Notes in Computer Science, **739**(1993), Springer-Verlag, 460-469.

[4] A. Miyaji, "Weakness in message recovery signature schemes based on discrete logarithm problems 1", *IEICE Japan Tech. Rep.*, ISEC95-11, 1994.

[5] 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.

[6] "Specification for a digital signature standard", National Institute for Standards and Technology, Federal Information Standard Publication XX, draft (1991).

[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, **950**(1995), Springer-Verlag, 182-193.

[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.