New Signature Schemes Giving Message Recovery Based on EC-KCDSA

Dae Hyun YUM, Sang Gyoo SIM, Pil Joong LEE

Department of Electronic and Electrical Engineering, POSTECH

{daehyun, sim}@oberon.postech.ac.kr, pjl@postech.ac.kr

Abstract

New signature schemes giving message recovery based on EC-KCDSA are introduced. These signature schemes can be efficiently used with established EC-KCDSA systems.

I. Introduction

While the RSA[9] signature scheme has a message recovery feature, the ElGamal[2] signature scheme does not have a message recovery feature. The first ElGamal-type signature scheme having the message recovery feature was introduced by Nyberg and Rueppel [6, 7, 8]. Their schemes, called NR-signature, on finite fields were shown to have security flaws[1, 5, 7]. However, no practical attack on EC-NR (NR-signature on elliptic curves) was known to date.

In this paper, we will introduce new signature schemes having the message recovery feature based on EC-KCDSA, called EC-KCDSA/MR1 and EC-KCDSA/MR2. Since EC-KCDSA/MR1 and EC-KCDSA/MR2 are motivated by both EC-KCDSA and EC-NR, we expect that EC-KCDSA/MR1 and EC-KCDSA/MR2 can be efficiently used with deployed EC-KCDSA systems.

II. EC-KCDSA

In this section, we will review EC-KCDSA [3], briefly. The following notations will be used in the remainder of this paper.

Hash(·) - Hash function

G - An elliptic curve point of prime order n

x_A - The private signature key of entity A is a random integer in the interval [1, n-1]

Y_A - The public verification key of entity A is the elliptic curve point $Y_A = x_A^* G$, where $x_A^* = x_A^{-1} \bmod n$

z_A - Entity A's hash-code of certification data

$\oplus$ - XOR operator

$||$ - Concatenation

Signature generation

To sign a message M, an entity A selects a random integer k in the interval [1, n-1] and computes the signature (r, s) as follows:

$r = \text{Hash}(kG)$

$e = r \oplus \text{Hash}(z_A||M)$

$s = x_A(k - e) \pmod n$

Signature verification

To verify A's signature (r, s), entity B executes the following steps:

$e = r \oplus \text{Hash}(z_A \| M)$

$R = sY_A + eG$

$r' = \text{Hash}(R)$

If $r' = r$, B accepts (r, s) as A's valid signature.

III. EC-KCDSA/MR1

EC-KCDSA/MR1 is a signature scheme giving message recovery based on EC-KCDSA.

A given message M is split into the recoverable part M_{rec} and the non-recoverable part M_{clr}. The message representative d of M is consisted of M_{rec} and some redundancy. A signer can select the type of redundancy as natural redundancy, added redundancy or both. The message with natural redundancy means that the message includes redundancy naturally or that redundancy of the message is verifiable implicitly in some applications. The message with added redundancy may be constructed by the hash token of the message or the recoverable message.

If a signer use added redundancy, then the types of redundancy shall be fixed as either short redundancy or long redundancy. While, short redundancy is used if the entire message is recoverable from the signature, long redundancy is used if the partial message is recoverable from the signature. Agreement on these choices among the users is essential for the purpose of the operation of the digital signature mechanism giving message recovery.

A binding data *bind* can contain z_A and M_{clr}, where M_{clr} is the non-recoverable part of message.

Signature generation

To sign a message representative (inputted data) d, an entity A selects a random integer k in the interval $[1, n-1]$ and computes the signature (r, s) as follows:

$r = \text{Hash}(kG) \oplus d$

$e = r \oplus \text{Hash}(bind)$

$s = x_A(k - e) \pmod n$

Signature verification

To verify A's signature (r, s), an entity B executes the following steps:

$e = r \oplus \text{Hash}(bind)$

$R = sY_A + eG$

$d = r \oplus \text{Hash}(R)$

If d satisfies the redundancy conditions, B accepts (r, s) as A's valid signature.

By using equivalence class in [4, 8], we can show the security of EC-KCDSA/MR1 has the same level as that of EC-KCDSA.

In EC-KCDSA, if we let the message representative d as $\text{Hash}(z_A \| M)$, the message (r, s) is obtained as $(r, s) = (\text{Hash}(kG), x_A(k - r \oplus d))$. In EC-KCDSA/MR1, we assume there is not the non-recoverable part, i.e., we ignore *Bind* and Hash(*Bind*). Then, the signature of EC-KCDSA/MR1 becomes $(r^*, s^*) = (\text{Hash}(kG) \oplus d, x_A(k - r^*))$. The signature (r, s) of EC-KCDSA can be transformed into a signature $(r \oplus d, s)$ of EC-KCDSA/MR1 and the signature (r^*, s^*) of EC-KCDSA/MR1 can be transformed into a signature $(r^* \oplus d, s^*)$ of EC-KCDSA. Consequently, EC-KCDSA and EC-KCDSA are strongly equivalent.

IV. EC-KCDSA/MR2

While EC-KCDSA/MR1 is strictly based on EC-KCDSA, EC-KCDSA/MR2 adopts some flavors of EC-NR.

Signature generation

To sign an inputted data d, an entity A selects a random integer k in the interval $[1, n-1]$ and computes the signature (r, s) as follows:

$\Pi = \text{Hash}(kY_A)$

$r = d \oplus \Pi \oplus \text{Hash}(bind)$

$s = k - x_A \cdot r \pmod n$

Signature verification

To verify A's signature (r, s), an entity B executes the following steps:

$R = sY_A + rG$

$\Pi = \text{Hash}(R)$

$d = r \oplus \Pi \oplus \text{Hash}(bind)$

If d satisfies the redundancy conditions, B accepts (r, s) as A's valid signature.

EC-NR uses a user key pair as $(x_A, Y_A = x_A G)$. However, another key pair $(x_A, Y_A = x_A^* G)$ also can be used (we denote this scheme as EC-NR*). When using the latter key pair, signature generation and verification steps become as follows:

$\Pi = \pi(kY_A)$

$r = d + \Pi \pmod n$

$s = k - x_A r \pmod n$

$R = sY_A + rG$

$\Pi = \pi(R)$

$d = r - \Pi \pmod n$

Here, $\pi(\cdot)$ is a mapping converting an elliptic curve point into an integer.

EC-NR* and EC-KCDSA/MR2 have the same equation computing s. However, EC-KCDSA uses 'Hash(·)' and 'XOR operator $\oplus$' instead of '$\pi(\cdot)$' and '+ mod n', respectively. We expect that these changes enhance the security of EC-KCDSA/MR2. In addition, implementation of Hash(·) is independent of the underlying field, while implementation of $\pi(\cdot)$ depends on the underlying fields, i.e., GF(p), GF(2^m) or GF(p^m).

V. Conclusions

We proposed new message recovery signature schemes based on EC-KCDSA. Currently, we are working on proving the security of these signature schemes in the random oracle model and the generic group model.

References

[1] Chen-Chi Lin and Chi-Sung Laih, "Cryptanalysis of Nyberg-Rueppel's Message Recovery Scheme," IEEE Communications Letters, Vol. 4, No. 7, pp. 231-232, 2000.

[2] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, IT-31, 4, pp. 469-472, 1985.

[3] C. H. Lim and P. J. Lee, "A study on the proposed Korean Digital Signature Algorithm," Asiacrypt'98, LNCS 1514, pp.175-186, 1998.

[4] A. Miyaji, "A Message Recovery Signature Scheme Equivalent to DSA over Elliptic Curve," Asiacrypt'96, LNCS 1163, pp. 1-14, 1996.

[5] A. Miyaji, "Another Countermeasure to Forgeries over Message Recovery Signature," IEICE Trans., Fundamentals, Vol. E80-A, No. 11, pp. 2192-2200, 1997.

[6] 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.

[7] K. Nyberg and R. A. Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem," Eurocrypt'94, LNCS 0950, pp. 182-193, 1995.

[8] K. Nyberg and R. A. Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem," Designs Codes and Cryptography, Vol. 7, pp. 61-81, 1996.

[9] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of ACM, 21, 2, pp. 120-126, 1978.