Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Applied Mathematics and Computation 167 (2005) 870–881
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

An efficient signcryption scheme with forward secrecy based on elliptic curve

Ren-Junn Hwang *, Chih-Hua Lai, Feng-Fu Su

Department of Computer Science and Information Engineering, Tamkang University, 151 Ying-Chuan Road, Tamsui, Taipei Hsien 25137, Taiwan, ROC

Abstract

An efficient signcryption scheme based on elliptic curve is proposed in this paper. The signcryption scheme combines digital signature and encryption functions. The proposed scheme takes lower computation and communication cost to provide security functions. It not only provides message confidentiality, authentication, integrity, unforgeability, and non-repudiation, but also forward secrecy for message confidentiality and public verification. In the proposed scheme, the judge can verify sender's signature directly without the sender's private key when dispute occurs. Our scheme can be applied to mobile communication environment more efficiently because of the low computation and communication cost.

Keywords: Elliptic curve; Digital signature; Signcryption

1. Introduction

Message security and sender's authentication for communication in the open channel is a basic and important technology of Internet. For keeping message

* Corresponding author.
E-mail address: junhwang@ms35.hinet.net (R.-J. Hwang).

0096-3003/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2004.06.124

confidential and unforged, the sender uses a digital signature algorithm with his private key to sign the message, and encrypts the message and digital signature using a symmetric encryption algorithm using a randomly chosen secret key. The sender uses a public key encryption algorithm with the recipient's public key to encrypt this secret key as envelope. Then, the sender sends the envelope and cipher text to the recipient. After the recipient receives the cipher text and envelope, the recipient uses his private key to decrypt the envelope to get secret key and decrypts cipher text to get plain text and signature by using this secret key. Finally, the recipient verifies the message based on this signature. This method is named signature-then-encryption scheme which is shown in Fig. 1.

Zheng [1] first proposed a new cryptography technique named "Signcryption" which combines the functions of digital signature and encryption

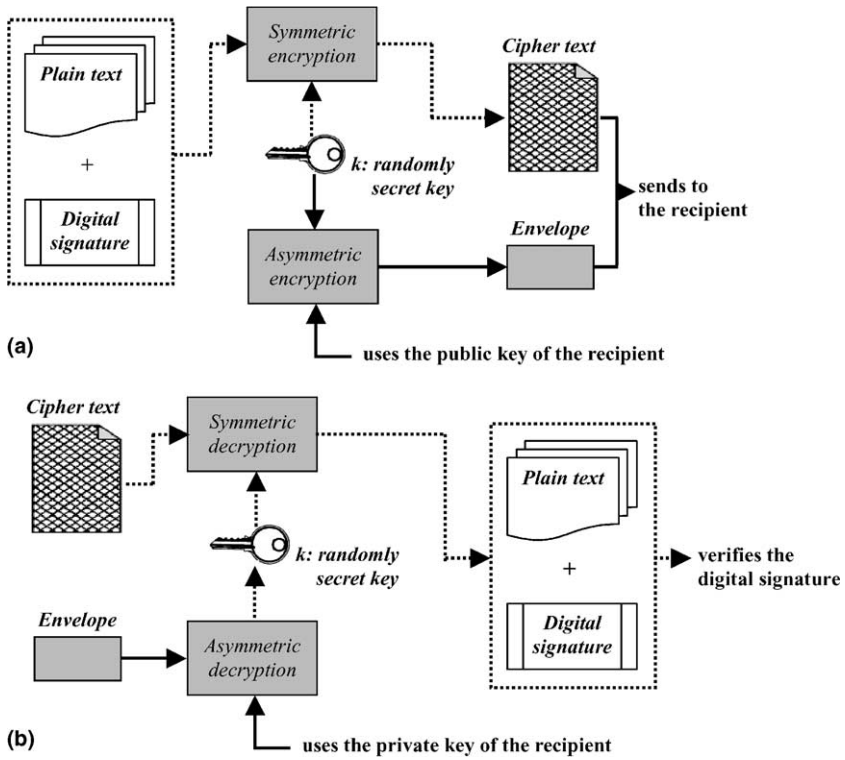

Fig. 1. Signature-then-encryption scheme. (a) The sender selects randomly secret key to encrypt plain text and digital signature. (b) The recipient decrypts envelope and cipher text to get plain text and digital signature.

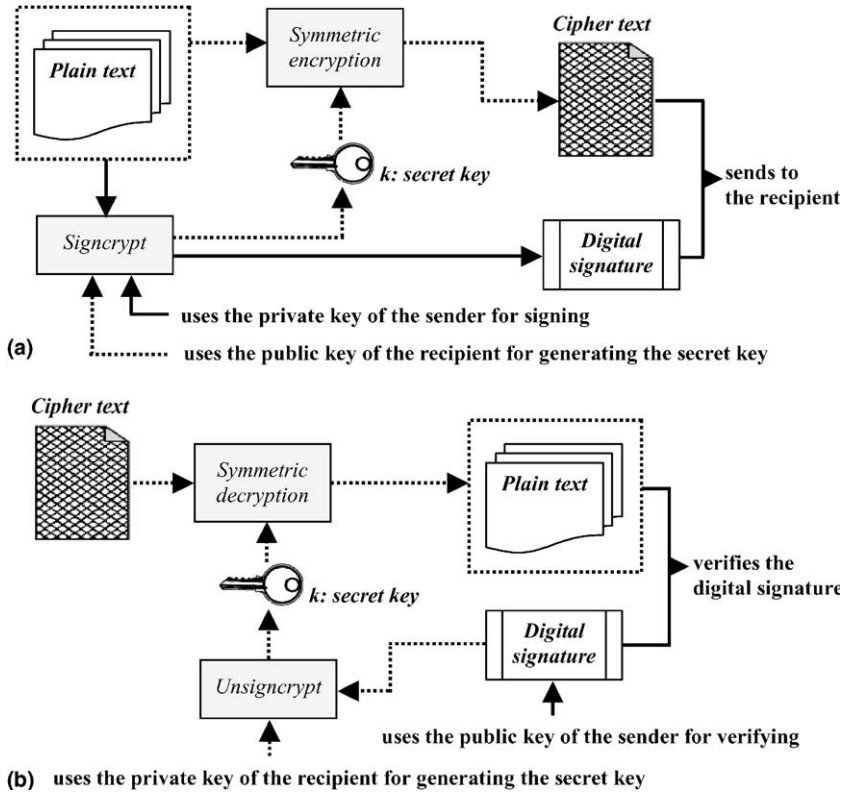

Fig. 2. Signcryption scheme. (a) Signcryption phase. (b) Unsigncryption phase.

algorithm for authentication and confidentiality. In the signcryption scheme, the sender uses the recipient's public key to derive a secret key for symmetric encryption. After the recipient receives the cipher text and digital signature, he uses his private key to derive the same secret key. The signcryption scheme is shown in Fig. 2.

Zheng [2] proposed another signcryption scheme based on elliptic curve, which saves about 58% computational cost and saving about 40% communication cost than signature-then-encryption scheme based on elliptic curve. Jung et al. [3] showed that Zheng's [1] scheme does not provide forward secrecy of message confidentiality when the sender's private key disclosed. They also proposed a new signcryption based on discrete logarithm problem (DLP for short) with forward secrecy. In Jung's scheme, even attacker obtains the sender's private key, he cannot get the corresponding original message yet that sender had sent. However, in those research results [1–3], when dispute occurs, the judge cannot directly verify the signature because of not knowing the recipient's

private key [4]. Bao and Deng [4] enhanced Zheng's [1] signcryption that the judge can verify signature without the recipient's private key. Gamage et al. [5] modified Zheng's [1] signcryption that anyone can verify the signature of cipher text. Their scheme only verifies the cipher text to protect confidentiality of message in firewall application.

In this paper, we propose a new efficient signcryption scheme based on elliptic curve. It provides not only confidentiality, authentication, integrity, unforgeability and non-repudiation, but also forward secrecy of message confidentiality and public verification. By forward secrecy of message confidentiality function, although the private key of the sender is disclosed, it does not affect the confidentiality of previous messages. By the public verification function, a judge directly verifies signature of original message without the sender's private key when dispute occurs. It enhances the justice of judge. In addition, our scheme saves great amount of computational cost especially for sender. In our signcryption scheme, we just need two elliptic curve point multiplications, one modular multiplication, one modular addition, and one one-way hash function computation for sender, which is more efficient than the previous schemes [3–5]. The lower computation cost make our scheme can be applied to the lower computational power device like mobile device more efficiently.

The structure of this paper is organized as follows. Section 2 introduces the new signcryption scheme. Section 3 analyses its security properties. We make the comparisons among the previous schemes and ours in Section 4. Finally, Section 5 makes some conclusions.

2. The proposed signcryption scheme with forward secrecy

The signcryption scheme with forward secrecy provides the security requirements [6]: message confidentiality, authentication, integrity, unforgeability, non-repudiation. The secret key of the proposed scheme is only related to recipient's private key. It protects the confidentiality of message even if the private key of the sender disclosed. It is the forward secrecy function that is provided by our signcryption. In addition, the proposed scheme provides publicly verifiable function. By the publicly verifiable function, the judge can verify sender's signature directly without the sender's private key when dispute occurs. The proposed scheme spends lower time in computation, especially for sender. It contains four phases: initialization phase, signcryption phase, unsigncryption phase and judge verification phase. In the initialization phase, system generates and publishes domain parameters of elliptic curve, and each user generates his own private key and the related public key. Each user should get the certification of his public key from the certificate authority (CA). In the signcryption phase, the sender Alice signs and encrypts a message. Then

she sends the signcrypted text to the recipient Bob. In the unsigncryption phase, the recipient Bob derives secret key to decrypt plain text. He also verifies the signature. In the judge verification phase, a judge decides whether the sender Alice sent the signcrypted message or not, when dispute occurs. We describe these four phases in the following.

Initialization phase. In this phase, we should select and publish some parameters as follows:

- q a large prime number, where $q > 2^{160}$.
- a, b two integer elements which are smaller than q and satisfy $4a^3 + 27b^2 \bmod q \neq 0$.
- F the selected elliptic curve over finite field q: $y^2 = x^3 + ax + b \bmod q$.
- G a base point of elliptic curve F with order n.
- O a point of F at infinite.
- n the order of point G, where n is a prime, $n \times G = O$ and $n > 2^{160}$. (The symbol "×" denotes the elliptic curve point multiplication [7].)
- H a one-way hash function.
- $E_k(\cdot)/D_k(\cdot)$ symmetric encryption/decryption algorithm with private key k such as DES or AES.

The sender Alice randomly selects an integer d_A as her private key and $d_A < n$. She computes her public key $U_A = d_A \times G$. The recipient Bob also selects private key d_B and public key $U_B = d_B \times G$ by the same way as Alice. They need to get a certificate of their public key from the certificate authority.

Signcryption phase. Assume that Alice wants to send a message M to Bob. Alice generates digital signature (R, s) of message M and uses the symmetric encryption algorithm and secret key k to encrypt M. Let C be the cipher text. Alice generates the signcrypted text (C, R, s) in the following steps.

Step 1: Verifies Bob's public key U_B by using his certificate.
Step 2: Randomly selects an integer r, where $r < n$.
Step 3: Computes $R = r \times G = (r_1, r_2)$.
Step 4: Computes $K = r \times U_B = (k, l)$.
Step 5: Uses the symmetric encryption algorithm to generate cipher text $C = E_k(M)$, where the secret k is generated in Step 4.
Step 6: Uses the one-way hash function to generate $h = H(M \| r_1)$, where r_1 is generated in Step 3.
Step 7: Computes $s = d_A - h \cdot r \bmod n$.
Step 8: Sends the signcrypted text (C, R, s) to Bob.

Unsigncryption phase. Bob receives the signcrypted text (C, R, s). He decrypts cipher text C by performing symmetric decryption algorithm with secret key k. He also verifies the signature. Bob gets the plain text as follows.

Step 1: Verifies Alice's public key U_A by using her certificate.
Step 2: Computes $K = d_B \times R = (k, l)$.
Step 3: Uses a symmetric decryption algorithm to generate plain text $M = D_k(C)$, where the secret key k is computed in Step 2.
Step 4: Uses the one-way hash function to compute $h = H(M||r_1)$, where r_1 is the x-coordinate value of the point R.
Step 5: Verifies $s \times G + h \times R$ is equal to U_A or not. If it is true then accept M is correct plain text which is sent by Alice; otherwise reject M.

Judge verification phase. By some reasons, we need the trusted third party such as judge to decide that the sender Alice sent M to the recipient Bob. In our scheme, the recipient Bob only provides (M, R, s) to the judge, when dispute occurs. The judge decides whether the sender Alice ever sent the message to the recipient Bob or not based on (M, R, s). The judge performs the following steps to make the decision.

Step 1: Verifies Alice's public key U_A by using her certificate.
Step 2: Uses the one-way hash function to generate $h = H(M||r_1)$.
Step 3: If $s \times G + h \times R$ equal to U_A then the sender Alice sent (M, R, s) to the recipient Bob really; otherwise she did not send this message to the recipient Bob.

3. The security functions of the proposed scheme

The proposed scheme provides seven security functions: message confidentiality, authentication, integrity, unforgeability, non-repudiation, forward secrecy of message confidentiality and public verification. This section proves these results. Most of these results are based on two problems: the elliptic curve discrete logarithm problem (ECDLP for short) [7] and the elliptic curve Diffie–Hellman problem (ECDHP for short) [8]. Up to now, both of these problems are hard [7,8]. The ECDLP is defined in Definition 1. Definition 2 defines ECDHP. Boneh and Lipton [9] show that the ECDLP and the ECDHP are equivalent if the best algorithm for the ECDLP is fully exponential computational time complexity.

Definition 1 (*The Elliptic Curve Discrete Logarithm Problem*). Let P and Q be two points of an elliptic curve with order n and n is a prime. The

point $Q = k \times P$ where $k < n$. Given these two points P and Q, find the correct k of Q.

Up to now, it is computational infeasible to generate k from P and Q [7].

Definition 2 (*The Elliptic Curve Diffie–Hellman Problem*). Let G be a base point of an elliptic curve with a prime order n and $P = c \times G$ and $Q = d \times G$. Given two points P and Q without c and d, find another point $K = c \cdot d \times G$.

As the ECDLP, ECDHP is a computational infeasible problem [8].

(1) *Confidentiality*. In our scheme, if the attacker wants to derive the original message, he must get the secret key k. The secret key k is the x-coordinate value of point K. There are many cases that the attacker can try to derive the secret key k. However, we show that possible ways to generate secret key k is equal to solve the ECDLP or ECDHP. By Definitions 1 and 2, this two problems are computational infeasible.

Case 1: Assume that the attacker tries to compute point K from Eq. (1), he should derive secret parameter r from Eq. (2). However, the attacker just know the point U_B and G, he has to solve the ECDLP firstly. It is computational infeasible that the attacker tries to get the point K from Eqs. (1) and (2).

$$K = r \times U_B, \tag{1}$$

$$R = r \times G. \tag{2}$$

Case 2: The attacker can get R and U_B easily. Assume that the attacker tries to compute K from Eq. (3) based on $R = r \times G$ and $U_B = d_B \times G$. According to Definition 2, he has to solve the ECDHP by this way.

$$K = r \cdot d_B \times G. \tag{3}$$

Case 3: The attacker gets the recipient's public key U_B easily. If he tries to generate d_B from $U_B = d_B \times G$, then he has to solve ECDLP as Case 1. Therefore, the attacker is hard to get K from Eq. (4) because no knowledge about d_B.

$$K = d_B \times R. \tag{4}$$

Case 4: The attacker can try to get K from Eq. (5). However, he must generate correct $h = H(M\|r_1)$ for message M in this case. Because the one-way hash function is collision resistant, the attacker cannot derive the correct h and h^{-1} without knowing original message M and r_1. Therefore, he cannot derive point K from Eq. (5).

$$K = (h^{-1} \cdot (d_A^{-s})) \times U_B. \tag{5}$$

(2) *Authentication* (The proposed scheme provides authentication property). In our proposed scheme, the recipient and the judge can use the sender's public key U_A with its certificate to authenticate the validity of the sender. When the recipient decrypted the cipher text C to get the plain text M, he can use Eq. (6) to authenticate the correctness of the received message. If the equation is hold, the recipient is sure that the received message does not modify in the transmission process. Therefore, the proposed scheme provides the authentication of the sender's identity and the transmitted message.

$$U_A = s \times G + h \times R. \tag{6}$$

(3) *Integrity* (The proposed scheme provides integrity). In our proposed scheme, the recipient can verify whether the received message is the original one that was sent by the sender or not. By Steps 6 and 7 of the signcryption phase, the sender computes and sends s to the recipient. The parameter s is generated as Eq. (6), where $h = H(M\|r_1)$ and M is the original message. If the attacker changes the original cipher text C as C', the related message is changed to M'. Let $h' = H(M'\|r_1)$. By the property of one-way hash function, it is computational infeasible for the attacker to modify C as C' such that $h = H(M\|r_1)$ is equal to $h' = H(M'\|r_1)$. Furthermore, the attacker does not get d_A and r, he cannot compute the correct s' from s and R, such that $s' = d_A - h' \cdot r \bmod n$. So, if the C is altered, the recipient can verify that the original message is altered in the unsigncryption phase.

(4) *Unforgeability* (The proposed scheme provides unforgeability). In our scheme, the attacker cannot forge valid (M, R, s) without the private key of sender. Assume that the attacker tries to forge a valid (M', R, s') from a previous (M, R, s) that he eavesdropped. The (M', R, s') has to satisfy Eq. (7). The attacker must generate h' and s' from Eqs. (8) and (9) for the message M'. However, the attacker or the recipient does not get the correct secret parameter r, he cannot generate the correct s'. If the attacker or the recipient wants to derive the randomly secret parameter r from $R = r \times G$, he should solve the ECDLP firstly, it is computational infeasible. Therefore, our proposed scheme satisfies unforgeability.

$$U_A = s' \times G + h' \times R, \tag{7}$$

$$h' = H(M'\|r_1), \tag{8}$$

$$s' = s + h \cdot r - h' \cdot r \bmod n. \tag{9}$$

(5) *Non-repudiation* (The proposed scheme provides the non-repudiation property). When dispute occurs for sender and recipient, the recipient can send (M, R, s) to the judge for settling whether the original message M sent by sender or not. In Judge Verification phase, the judge can determine the signature is generated by the sender if Eq. (6) is hold, because of only the sender can use

her own private key d_A to generate correct signature s. According to the previous analysis about unforgeability, we show that anybody without the private key d_A cannot forge the correct signature of message as the sender. In other words, our proposed scheme satisfies non-repudiation property.

(6) *Forward secrecy of message confidentiality*. The forward secrecy of message confidentiality means that the sender's long-term private key d_A is compromised, but the attacker still cannot recover any previous message M from (C, R, s) which is a signcrypted text that the sender sent to somebody before. In our proposed scheme, if the attacker tries to derive the plaintext M, he has to decrypt its cipher text C by its corresponding secret key k. However, without original message M, he cannot use Eq. (10) to compute h or the inverse value of h under modulo n. In addition, if he wants to derive r from R, he should solve the computational infeasible problem ECDLP. Anybody even got sender's private key d_A and signcrypted text (C, R, s), who still cannot compute the point K using Eq. (11). In other words, he cannot decrypt signcrypted text to get the previous message M. Therefore, our proposed scheme provides forward secrecy of message confidentiality even if the sender's private key divulged.

$$h = H(M \| r_1), \tag{10}$$

$$K = r \times U_B = h^{-1} \cdot (d_A - s) \times U_B. \tag{11}$$

(7) *Public verification* (The proposed scheme provides the publicly verifiable function). When dispute occurs, our scheme provides public verification as well as Bao and Deng's scheme [4]. Because the sender's public key U_A with its certificate is associated to his own private key d_A. Anybody who obtains the (M, R, s) and sender's public key U_A, he can use Eq. (6) to settle whether the originator of the message M send by the sender or not without the assistance of sender's private key in Judge verification phase of our proposed scheme. Our proposed scheme provides the public verification properties.

4. Discussion

Up to now, there are some research results related signcryption such as: Bao and Deng's scheme [4], Gamage et al.'s scheme [5], Jung et al.'s scheme [3], Zheng's schemes [1,2]. This section shows the comparisons among these results and ours.

4.1. Security properties

Only our proposed scheme and Jung et al.'s [3] scheme can provide forward secrecy of message confidentiality. And only our proposed scheme and Bao–Deng's scheme [4] can make public verification of the signature with orig-

Table 1
Make the comparisons based on security properties

	Confidentiality	Integrity	Unforgeability	Non-repudiation	Forward secrecy	Public verification
Our scheme	Yes	Yes	Yes	Directly	Yes	Yes
Zheng [1]	Yes	Yes	Yes	Another protocol	No	No
Zheng [2]	Yes	Yes	Yes	Another protocol	No	No
Bao and Deng [4]	Yes	Yes	Yes	Directly	No	Yes
Gamage et al. [5]	Yes	Yes	Yes	Directly	No	Yes
Jung et al. [3]	Yes	Yes	Yes	Another protocol	Yes	No

inal message. Gamage et al. [5] also provides public verification with cipher text in firewall application. But in [1–3], the judge has to engage in an interactive zero-knowledge proof protocol for the non-repudiation of the sender. Table 1 summarizes all of the comparisons of security properties among our proposed scheme and the other signcryption schemes. The "Directly" mark of non-repudiation column of Table 1 means that the proof of non-repudiation need no help of any other protocols.

4.2. Computational cost

In our proposed scheme, we try to reduce sender's computational cost. Table 2 shows the comparisons of computational cost of sender and recipient among our signcryption scheme and others. The proposed scheme does not take any inverse computation for sender and recipient, and just need two

Table 2
Make the comparisons based on the computational cost

	Entity	ECPM	ECPA	EXP	DIV	MUL	ADD	HASH
Our scheme	Sender	2	–	–	–	1	1	1
	Recipient	3	1	–	–	–	–	1
Zheng [1]	Sender	–	–	1	1	–	1	2
	Recipient	–	–	2	–	2	–	2
Zheng [2]	Sender	1	–	–	1	1	1	2
	Recipient	2	1	–	–	2	–	2
Bao and Deng [4]	Sender	–	–	2	1	–	1	3
	Recipient	–	–	3	–	1	–	3
Gamage et al. [5]	Sender	–	–	2	1	–	1	2
	Recipient	–	–	3	–	1	–	2
Jung et al. [3]	Sender	–	–	2	1	–	1	2
	Recipient	–	–	3	–	1	–	2

ECPM = the number of elliptic curve point multiplication operation. ECPA = the number of elliptic curve point addition operation. EXP = the number of modular exponentiation operation. DIV = the number of modular division (inverse) operation. MUL = the number of modular multiplication operation. ADD = the number of modular addition operation. HASH = the number of one-way or keyed one-way hash function operation.

Table 3
Make the comparisons based on the average computational time of major operations

Various schemes	Sender Average computational time (ms)	Recipient Average computational time (ms)
Our scheme	$2 \times 83 = 166$	$3 \times 83 = 249$
Zheng [1]	$1 \times 220 = 220$	$2 \times 220 = 440$
Zheng [2]	$1 \times 83 = 83$	$2 \times 83 = 166$
Bao and Deng [4]	$2 \times 220 = 440$	$3 \times 220 = 660$
Gamage et al. [5]	$2 \times 220 = 440$	$3 \times 220 = 660$
Jung et al. [3]	$2 \times 220 = 440$	$3 \times 220 = 660$

elliptic curve point multiplication for sender. It is more efficient than the others [3–5].

In the same secure level, the elliptic curve multiplication only needs 83 ms and the modular exponentiation operation needs 220 ms for average computational time in the Infineon's SLE66CUX640P security controller [10]. Therefore, we consider about the most computational time for elliptic curve multiplication and modular exponentiation operation showed in Table 3.

Although our proposed scheme has lower performance than Zheng's scheme [2], our proposed scheme provides added functions including the forward secrecy of message confidentiality and public verification while Zheng's scheme [2] did not provide. Besides, judges can arbitrate non-repudiation of the sender without another protocol in our proposed scheme.

5. Conclusions

This paper proposed an efficient signcryption based on elliptic curve with forward secrecy and publicly verifiable. The proposed signcryption scheme with forward secrecy satisfies the message confidentiality of previous encrypted message even if the sender divulged his private key inattentively. Furthermore, the trusted third party judges the signature of sender without the additional interactive protocol. In a word, the proposed scheme not only provides the security properties of message confidentiality, authentication, integrity, unforgeability and non-repudiation, but also forward secrecy of message confidentiality and publicly verifiable function. Our scheme saves more computational cost for sender to suit the application of restricted computational device like mobile device.

References

[1] Y. Zheng, Digital signcryption or how to achieve Cost(Signature & Encryption) $\ll$ Cost(Signature) + Cost(Encryption), in: Advances in Cryptology—Crypto'97LNCS 1294, Springer-Verlag, 1997, pp. 165–179.

[2] Y. Zheng, H. Imai, How to construct efficient signcryption schemes on elliptic curves, Information Processing Letters 68 (1998) 227–233.
[3] H.Y. Jung, K.S. Chang, D.H. Lee, J.I. Lim, Signcryption schemes with forward secrecy, Proceeding of WISA 2 (2001) 403–475.
[4] F. Bao, R.H. Deng, A signcryption scheme with signature directly verifiable by public key, in: Proceedings of PKC'98LNCS 1431, Springer-Verlag, 1998, pp. 55–59.
[5] C. Gamage, J. Leiwo, Y. Zheng, Encrypted message authentication by firewalls, in: Proceedings of 1999 International Workshop on Practice and Theory in Public Key Cryptography (PKC'99), 1–3 March, 1999, Kamakura, JapanLNCS 1560, Springer-Verlag, 1999, pp. 69–81.
[6] J. Beak, R. Steinfeld, Y. Zheng, Formal proofs for the security of signcryption, in: Proceedings of PKC'02LNCS 2274, Springer-Verlag, 2002, pp. 81–98.
[7] D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA), International Journal of Information Security 1 (1) (2001) 36–63.
[8] Certicom Research, Standards for efficient cryptography, SEC 1: elliptic curve cryptography, Standards for efficient cryptography group (SECG), September 20, 2000.
[9] D. Boneh, R.J. Lipton, Algorithms for black-box fields and their application to cryptography, in: Advances in Cryptography: Crypto '96, 1996, pp. 283–297.
[10] L. Batina, S.B. Örs, B. Preneel, J. Vandewalle, Hardware architectures for public key cryptography, Integration the VLSI Journal 34 (1–2) (2003) 1–64.