JOINT SIGNATURE AND ENCRYPTION ON ELLIPTIC CURVE *

HAN Yiliang, YANG Xiaoyuan, ZHANG Jian

Key Lab. of Network and Information Security, A.P.F
Department of Electronic Technology, Engineering College of A.P.F
Wujing Road, Xi'an, Shaanxi, 710086, China
E-mail:Yilianghan@hotmail.com

Abstract: The known solutions to achieve confidentiality and authentication simultaneously fail to provide verifiability using standard elliptic curve signature [10][1]. An elliptic curve based signcryption scheme SC-ECDSA was proposed. A one time padding cipher with a session key generated by respective secret knowledge was constructed to encrypt messages. The standardized signature scheme ECDSA was used to sign and verify. Proof shows that SC-ECDSA matches the three security notions: unforgeability, non-repudiation and confidentiality (provable CUF-CPA). For typical security parameters, SC-ECDSA saves 80% computation time and 4.7% message expansion than other schemes. It presents a 29.2% reduction in computation time and a 6.9% reduction in message expansion than traditional *Sign-then-Encrypt*.

Keyword: signcryption; authenticated encryption; ECDSA.

1. Introduction

How to transmit a message confidentially and authentically is the essential security issue for information systems. To avoid forgery and keep private, originator will use authentication and encryption. Though *Sign-then-Encrypt* used in PEM (Privacy Enhanced Mail) and PGP (Pretty Good Privacy) is an appropriate composition, the high computation costs prevent it from using widely. Zheng has proposed the notion of signcryption, which is a novel public key primitive to achieve the combined functionality of authentication and confidentiality in an efficient manner [9][10]. A signcryption verified publicly was given [1]. The standardization is one of the crucial factors for practical uses of signcryption. Some schemes based on standardized signature were proposed also, such as SC-KCDSA [8], SC-DSA [7] and TBOS [4]. But designing a signcryption scheme which provides verifiability using standard elliptic curve signature is an open problem. The work is motivated by this. The paper proposed the first signcryption based on ECDSA [3] (Elliptic Curve Digital Signature Algorithm), one of the most widely used standard signature.

2. ECDSA-Verifiable Signcryption

Choosing an elliptic curve $E(Fq)$ on a finite field Fq (q is a prime number), G is a base point, $\mathrm{ord}(G)=n$. Hence, there is a subgroup generated by base point G. Choosing a secret number $s \in Zq$, we can compute $Q=sG$ easily. Computing s via Q and G is an ECDLP (elliptic curve discrete logarithm problem) which is too hard to be resolved at present. $H(.)$ and $LH(.)$ are hash functions. $MAC_k(.)$ is a message authentic function with key k.

There is a message $\boldsymbol{m}$ which will be signcrypted by originator ***Alice*** and sent to a specific recipient ***Bob***. A signcryption scheme is specified by three algorithms: Key Generation, Signcryption and Unsigncryption.

Key Generation. A random number $s_A \in \{1,\ldots,n\text{-}1\}$ is the private key of ***Alice***. Her public key is a point $P_A= s_AG$. ***Bob***'s private key is a random number $s_B \in \{1,\ldots, n\text{-}1\}$. His public key is a point $P_B= s_BG$.

Signcryption. ***Alice*** completes the following actions:

1. Chooses $k \in \{1,\ldots,n\text{-}1\}$ randomly.
2. Computes $R=kG=(x_1, y_1)$, and sets $r = x_1 \bmod q$.
3. Computes $kP_B=(x_2, y_2)$, $Kenc=LH(x_2)$, $(Kmac, Ksig)=H(y_2)$.
4. Computes $s=k^{-1}(H(m||Bind_{A,B}||Ksig)+ rs_A) \bmod n$.
5. Computes $e=MAC_{Kmac}(m)$.
6. Computes $\boldsymbol{c}= (m||e)\oplus Kenc$.

The triplet (c, R, s) is the signcryption text and will be sent to ***Bob***.

Unsigncryption. ***Bob*** verifies as follows:

1. Computes $s_BR=(x_2,y_2)$, $Kenc=LH(x_2)$, $(Kmac, Ksig)=H(y_2)$.
2. Computes $(m'||e) = c\oplus Kenc$.

*

3. Computes $e'= MAC_{Kmac}(m')$. If the $e\neq e'$, rejects m'.
4. Computes $u= s^{-1}H(m\|Bind_{A,B}\|Ksig)$, $v= s^{-1}r$.
5. Computes $(x_1', y_1')=uG+vP_A$. If $x_1\neq x_1'$ or $y_1\neq y_1'$, rejects m', else returns $m= m'$.

The triplet $(H(m\|Bind_{A,B}\|Ksig),R, s)$ is a ECDSA signature text on message $H(m\|Bind_{A,B}\|Ksig)$. The third party can verify in ECDSA manners.

3. Security of SC-ECDSA

A signcryption scheme is secure if the following conditions are satisfied [9]:

- **Unforgeability:** It is computationally infeasible for an adaptive attacker to masquerade ***Alice*** in creating a signcrypted text.
- **Non-repudiation:** It is computationally feasible for a third party to settle a dispute between ***Alice*** and ***Bob*** in an event where ***Alice*** denies the fact that she is the originator of a signcrypted text with ***Bob*** as its recipient.
- **Confidentiality:** It is computationally infeasible for an adaptive attacker to gain any partial information on the contents of a signcrypted text.

3.1. *Unforgeability of SC-ECDSA*

Dishonest ***Bob*** is the most powerful attacker to forge a signcryption, because he is the only person who knows the private key s_B which is required to directly verify a signcryption from ***Alice***. Given a signcryption text (c,R, s), ***Bob*** can use his private key s_B to decrypt the c , and obtain (m,R,s).Then the problem will turn into the verification of the normal ECDSA signature $(R, H(m\|Bind_{A,B}\|Ksig), s)$. ECDSA is known to be unforgeable against adaptive attacks. Therefore the signcryption scheme is unforgeable against adaptive attacks. Under the assumption that hash function has property of Random Oracle and ECDLP is hard enough, the advantage of a polynomial-time adversary $\mathrm{Adv}(A)=|2\Pr[(R, c, y)=(c',R', s')]-1|$ is a negligible function. So, SC-ECDSA is secure against all of known forge attacks.

3.2. *Non-repudiation of SC-ECDSA*

The target of non-repudiation is to prevent ***Alice*** from denying the signcryption she sent. Non-repudiation of SC-ECDSA is achieved through verification of the triplet $(H(m\|Bind_{A,B}\|Ksig), R, s)$ publicly, while that of ECDSA is achieved through verification of the triplet $(H(m),r, s)$. Unforgeability implies non-repudiation if there is no duplication of the signcryption. If the signcryption scheme is malleable or forgeable, ***Alice*** will have opportunity to deny. But SC-ECDSA is unforgeable. So non-repudiation can be achieved when no repudiation signcryption exits.

J.Stern, D.Pointcheval and J.Malone-Lee found that ECDSA is a *duplicate signature*, because the map f: $R\rightarrow r$ is not unique. The two symmetrical point has the same x-coordinate: $R =(x_R, y_R)$, $-R =(x_R, -y_R)$, so the same signature (r, s) can be got by (m_1,R,s) and $(m_2,-R, s)$ [6]. While there is no duplication of signcryption exits in SC-ECDSA, because the map f: $R\rightarrow R$ is unique.

Hence, SC-ECDSA has the stronger non-repudiation than ECDSA.

3.3. *Confidentiality of SC-ECDSA*

Let F be a family of functions with domain $\{0,1\}^{l'}$ and range $\{0,1\}^{l''}$. The OTP(One Time Padding, stream ciphers that xor data with a (pseudo) random pad) encryption under f of plaintext x is performed by choosing $r\in_R\{0,1\}^{l'}$ and computing $c=f(r)\oplus x$. The ciphertext is the pair (r,c). If f is chosen at random, we get perfect secrecy against chosen-plaintext attacks. We denote this scheme by $\mathrm{OTP}_\$$. Let MAC be a message authentic function family with n bits outputs, and k a key to a member of that family. Then the definition of $\mathrm{AtE}(\mathrm{OTP}_\$,\mathrm{MAC})$ composition is given as: (i) computes $t=MAC_k(x)$; (ii) appends t to x; (iii) outputs the OTP encryption under f of the concatenated $c=f(r)\oplus(x\|t)$.

Lemma 1. If MAC is a message authentic function family that resists one-query attacks, AtE $(\mathrm{OTP}_\$, MAC)$ will be CUF-CPA security [2].

We will construct an encryption scheme ENC in $\mathrm{AtE}(\mathrm{OTP}_\$,MAC)$ manner which works on message m. $LH(.)$ is a hash function with $l'+|n|$ bits outputs. H is a hash function with l bits outputs.

Encryption. ***Alice*** completes the following operation:

1. Selects the number $k\in\{1,\ldots,n-1\}$ at random.
2. Computes $R=kG=(x_1, y_1)$.
3. Computes $kP_B =(x_2, y_2)$, $Kenc=LH(x_2)$, $(Kmac, Ksig)=H(y_2)$.
4. Computes $e=MAC_{Kmac}(m)$.
5. Computes $c=(m\|e)\oplus Kenc$.

(c, R) is the ciphertext and will be sent to ***Bob***.

Decryption: ***Bob*** completes the following operation after receiving ciphertext:

1. Computes $s_BR=(x_2, y_2)$.$Kenc=LH(x_2)$, $(Kmac, Ksig)=H(y_2)$.

2. Computes $(m'||e)= c\oplus Kenc$.
3. Computes $e'=MAC_{Kmac}(m')$. If $e=e'$, returns $m=m'$, else rejects it.

Theorem 1. The *ENC* is a semantic security scheme, i.e. in the sense of CUF-CPA.

Proof. Defining two functions: (i) $x(R)=R_x$ denotes the operation of computing *x*-coordinate of a point *R*; (ii) $E(\boldsymbol{x})=R$ denotes the operation of embedding $\boldsymbol{x}$ into an elliptic curve. Let $r=x(R)=x_1$ and $R=kG$. The value of *r* is random because of the same property of *k*. Let $f(.)=LH(x(s_B E(.)))$. Function $f(.)$ is private and random, because s_B is private and random. While $f(r)=LH(x(s_B E(r))) =LH(x(s_B E(\boldsymbol{x}_1))) =LH(x(s_B R)) =LH(x_2)=Kenc$, which is the encryption key in *ENC*. *Kmac* is the authentic key that can be computed by both the sender and recipient. Hence, *ENC* is a composition in AtE($OTP_\$$,*MAC*) manner.$H(.)$ is a hash function which achieves the IND-CMA security.

Then, *ENC* is CUF-CPA. ?

Theorem 2. *ENC* and SC-ECDSA have the same security property of confidentiality.

Proof. All of public data for SC-ECDSA as follows: *q*, *n*, *G*, $P_A=s_A G$, $P_B=s_B G$, $R=kG$, *c*, *s*. The attacker can compute $H(m||Bind_{A,B}||Ksig)G=rP_A-sR$, where $r=x_1=x(R)$, let $h=H(m||Bind_{A,B}||Ksig)$.

An adaptive attacker to *ENC* can obtain the following public data: *q*, *n*, *G*, $P_B=s_B G$, $R=kG$, *c*. Giving the value of *hG* will not reduce the complexity of attacking *ENC*, because *hG* hides all of the information of message *m* under the assumption of *Random Oracle*. Suppose that A_{ENC} is an adversary for *ENC* which works on (q, n, G, $P_B=s_B G$, $R=kG$, c, hG) and outputs partial information $\tilde{m}$ of message *m*. A_{SC} is an adversary for SC-ECDSA which works on (q, n, G, $P_A=s_A G$, $P_B=s_B G$, $R=kG$, c, s) and outputs partial information $\tilde{m}$ of message *m*.

There is a deterministic polynomial time algorithm $A_{SC}(q, n, G, P_A=s_A G, P_B=s_B G, R=kG, c, s)$:

1. Computes $hG=rP_A - sR$.
2. $\tilde{m}=A_{ENC}(q, n, G, P_B=s_B G, R=kG, c, hG)$.

If A_{ENC} gets any partial information of message *m*, so does A_{SC}.

There is a deterministic polynomial time algorithm $A_{ENC}(q, n, G, P_B=s_B G, R=kG, c, hG)$:

1. Selects $s\in\{1,\ldots,l\}$ randomly.
2. Computes $P_A=r^{-1}sR - r^{-1}hG$.
3. $\tilde{m}=A_{SC}(q, n, G, P_A, P_B=s_B G, R=kG, c, s)$.

If A_{SC} gets any partial information of message *m*, so does A_{ENC}.

SC-ECDSA is CUF-CPA. ?

4. Efficiency of SC-ECDSA

The advantage of SC-ECDSA over *Sign-then-Encrypt* composition and other signcryption schemes in details will be shown. We will construct the ECDSA-then-PSEC-1[5] composition as a usual *Sign-then-Encrypt* scheme.

4.1. *Computation Cost*

In public key cryptosystems, computing modular multiplication, modular exponential, modular inverse and multiples of points on elliptic curve consumes the most of computational resources, while the costs of addition, hash, encrypt\decrypt in symmetric cryptosystem is negligible.

Table 1.Comparison of computation cost

	KG	S	U	AC	VP
SCS[9]	2*E*	1*E*+1*I*	2*E*	/	/
ECSCS[10]	2*k*P	1*k*p+1*I*	2*k*P	/	/
Bao&Deng[1]	2*E*	2*E*+1*I*	3*E*	0	2*E*
KCDSA[8]	2*E*	2*E*	3*E*	save *r*,*s* or 3*E*	2*E*
SC-DSA[7]	2*E*	2*E*+2*I*	3*E*+1*I*	save *r*,*s* or 2*E*+1I	2*E*+1*I*
StE	2*k*P	3*k*P+1*I*	4*k*P+1*I*	0	2*k*P+1*I*
SC-ECDSA	2*k*P	2*k*P+1*I*	3*k*P+1*I*	0	2*k*P+1*I*

Note: a. KG denotes key generation; S denotes signcrytpion; U denotes unsigncryption; AC denotes additional computation; VP denotes verify publicly.

b. *E* denotes modular exponential; *I* denotes modular inverse; *k*P denotes multiples of points on elliptic curve.

c. The secure parameter of DLP based schemes (e.g. DSA): $|p|$=1024bits, $|q|$=160bits. The secure parameter of ECDLP based schemes (e.g. ECDSA): $|n|$=160bits [3].

Remark 1. (Compared with DLP based signcryption schemes). We only compare the computation cost of SC-ECDSA and SCS, because SCS is the fastest scheme in all of the four DLP based schemes (SCS, Bao&Deng, KCDSA and SC-DSA). By the result of [3], the computation cost of key generation operation in SC-ECDSA is 1/8 of that in SCS; signcryption operation in

SC-ECDSA is 1/4 of that in SCS, and unsigncryption is 1/5 of that in SCS. SC-ECDSA saves computational cost 80% over others.

Remark 2. (Compared with other ECDLP based schemes). There are three ECDLP based schemes: ECSCS, ECDSA-then-PSEC-1(StE) and SC-ECDSA. The computation cost of SC-ECDSA is slightly higher than that of ECSCS which has the flaw of being verified publicly. The cost of signcrytion operation in SC-ECDSA is 2/3 of Sign-then-Encrypt. The cost of unsigncryption operation in SC-ECDSA is 3/4 of Sign-then-Encrypt. This represents a 29.2% reduction in average computational cost.

To sum up, SC-ECDSA has the highest efficiency in all of the schemes which have the same function.

4.2. *Communication Cost*

Definition 1. ***Data Expended Rate***. In a signcryption scheme S on plaintext m, C_{Σ} denotes all of the signcryption text, **Data Expended Rate** can be defined as DR(S) = ($|C_{\Sigma}|-|m|$) / $|C_{\Sigma}|$, where $|m|$ denotes the length of message ***m***.

Secure parameters of cryptographic primitive: $|p|$=1024bits, $|q|$=160bits (DLP based schemes e.g. DSA); $|n|$=160bits (ECDLP based schemes e.g. ECDSA); the block length of block cipher is 64bits (e.g. DES, IDEA et al); the secure hash function outputs at least 160bits message digest. Comparison results are given in Table 2.

Table 2. Comparison of Data Expended Rate

	M	C_{Σ}	DR
SCS[9]	$\|D(.)\|$	$D(.)\|+\|KH(.)\|+\|q\|$	84%
ECSCS[10]	$\|D(.)\|$	$\|D(.)\|+\|h\|+\|n\|$	84%
Bao&Deng[1]	$\|D(.)\|$	$\|D(.)\|+\|h(.)\|+\|q\|$	84%
KCDSA[8]	$\|D(.)\|$	$\|D(.)\|+\|h(.)\|+\|q\|$	84%
SC-DSA[7]	$\|D(.)\|$	$\|D(.)\|+2\|q\|$	84%
StE	$\|n\|$	$7\|n\|$	86%
SC-ECDSA	$\|n\|$	$5\|n\|$	80%

SC-ECDSA saves communication 6.9% over sign-then-encrypt and 4.7% over others.

5. Implementation Issue

Avoiding the hybrid cryptosystems used in other schemes makes SC-ECDSA be implemented in software and hardware at a low cost. While Zheng's ECSCS uses four kinds of cryptography components: symmetrical cipher, hash function, keyed hash function and elliptic curve based computation. In other word, an application (software or device) that must contain four kinds of cryptosystem paradigms can implement ECSCS. Hence, SC-ECDSA scheme is more feasible than others.

We have implemented SC-ECDSA. Test platform as follows:

Complier: gcc (GNU C Compiler, version 2.91.60)
CPU: Intel Pentium IV 2.4GHz
RAM: 128Mbytes
Key length: 160bits

It costs about 9ms to signcrypt and 12ms to unsignrypt.

6. Conclusion

The signcryption scheme proposed in the paper has the following advantages: 1. based on a standard signature algorithm ECDSA; 2. computation cost and message expansion are less than that of traditional approach and other signcryption; 3. it is a provable secure scheme; 4. it is feasible in practice.

References

1. F.Bao, R.H.Deng, A Signcryption Scheme with Signature Directly Verifiable by Public Key, *In Public Key Cryptography'98, LNCS1431*,pp.55-59
2. H.Krawczyk, The Order of Encryption and Authentication for Protecting Communications (or: How secure is SSL?), *In Advances in Cryptoloty-Crypto'01,LNCS2139*, pp.310-331
3. N.Koblitz, A.Menezes, S.Vanstone, the State of Elliptic Curve Cryptography. *Designs, Codes and Cryptography*, vol.19,pp.173-193, Oct.2000
4. J.Malone-Lee and W. Mao. Two birds one stone: Signcryption using RSA. *Topics in Cryptology - Cryptographers' Track, RSA Conference-2003, LNCS 2612*, pp.210-224
5. T.Okamoto, E. Fujisaki H. Morita. *PSEC: Provably Secure Elliptic Curve Encryption Scheme.* Submission to IEEE P1363a (1998, March), http:// grouper.ieee.org/groups/ 1363/P1363a/
6. J. Stern,D.Pointcheval, J.Malone-Lee et al , Flaws in Applying Proof Methodologies to Signature Schemes, *In Advances in Cryptology-Crypto'02, LNCS 2442*, pp.93–110
7. J.B.Shin, K.Lee, K.Shim, New DSA-Verifiable Signcryption Schemes. *In ICISC'02, LNCS2587*, pp.35-47
8. D.H.Yum, P.J.Lee, New Signcryption Schemes based on KCDSA. *In ICISC'01, LNCS2288,* pp.305-317
9. Y.Zheng, Digital signcryption or how to achieve cost (signature &encryption) << cost (signature) + cost (encryption), (Extended abstract), *In Advances in Cryptoloty-Crypto'97, LNCS1294*, pp.165-179

10. Y.Zheng, H. Imai. How to construct efficient signcryption schemes on elliptic curves. I*nformation Processing Letters*,Vol.5, pp.227-233,May 1998