Elliptic Curve based Signcryption and its Multi-party Schemes

Yiliang HAN Xiaoyuan YANG
Key Lab. On Network and Information Security of Armed Police Force
Department of Electronic Technology, Engineering College of Armed Police Force
Xi'an, 710086, CHINA
yilianghan@hotmail.com

Abstract

Signcryption is a novel public key primitive to achieve the combined functionality of authentication and confidentiality in an efficient manner. A new Elliptic Curve Cryptosystems based Signcryption which combines ECDSA and PSCE-1 is presented in the paper. The signcryption scheme is a publicly verifiable scheme which can be verified by the third party after the specific recipient removes his key information. Analysis shows that the proposed scheme is secure against the adaptive chosen ciphertext attack. The signcryption saves the communication cost at least 1.25 times and enhances computation cost 1.19 times over ECDSA-then-PSCE-1. Compared with other signcryption schemes, such as Y.Zheng's ECSCS, the new signcryption uses a uniform elliptic curve cryptosystem platform instead of four kinds of cryptosystem components: hash function, keyed hash function, symmetric cipher and elliptic curve. While keeping high security and efficiency, the scheme can be implemented in software and hardware at low price because of above advantages. Base on the signcryption, a broadcast scheme for multiple recipients and a threshold scheme with key distributed generation for multiple senders are also proposed.

Keywords: *Threshold Cryptosystem. Signcryption. Distributed Key Generation*

1. Introduction

To avoid forgery and ensure confidentiality of a message, originator will use authentication and encryption. In common sense, the order of authentication and encryption can be divided into three classes [8]: authentication then encryption (AtE), encryption then authentication (EtA), encryption and authentication (E&A). The three methods in public key cryptosystems setting are: sign-then-encrypt, encrypt-then-sign, sign-and-encrypt.

The *Encrypt-then-Sign* is completely insecure against adaptive chosen ciphertext attack, even if the underlying encryption scheme is secure against adaptive chosen ciphertext attack. Because the sender's signature public key is easily malleable under adaptive chosen ciphertext. The common *Encrypt-and-Sign* paradigm cannot be *generically* secure because the signature part can reveal some information about the plaintext message, and this may be true even though the underlying signature scheme is unforgeable[5]. Though *Sign-then-Encrypt* is an appropriate composition, the high communication cost and computation cost hold its broad using.

Signcryption is a novel public key primitive to achieve the combined functionality of authentication and confidentiality in an efficient manner. It is more secure and more efficient than the traditional methods such as *Sign–then-Encrypt.* Y. Zheng proposed the conception of

signcryption and the first Discrete-Log based scheme SCS in 1997[4]. The proofs given by [5] showed that the SCS scheme was IND-CCA2 secure. J. Malone-Lee and W. Mao proposed a RSA based signcryption scheme TBOS and proved its IND-CCA2 security in 2003[10].

2. ECSCS Signcryption Scheme

Y. Zheng gave an signcryption scheme ECSCS in 1998[9]. The scheme will be described as following:

***Parameters public to all*:**

C— an elliptic curve over $GF(p^m)$, either with $p > 2^{150}$ and m=1or p=2 and m> 150 (public to all).

q— a large prime whose size is approximately of $|p^m|$ (public to all).

G— a point with order ***q***, chosen randomly from the points on *C* (public to all).

hash—a one-way hash function whose output has, say, at least 128 bits.

KH—a keyed one-way hash functions.

(***E***,***D***)—the encryption and decryption algorithms of a private key cipher.

***Keys*:**

$\boldsymbol{v_a}$—Alice's private key, chosen uniformly at random from [1,…,***q*** -1].

$\boldsymbol{P_a}$—Alice's public key ($\boldsymbol{P_a}=\boldsymbol{v_a}\boldsymbol{G}$, a point on ***C***).

$\boldsymbol{v_b}$—Bob's private key, chosen uniformly at random from [1,…,***q*** -1].

$\boldsymbol{P_b}$—Bob's public key ($\boldsymbol{P_b}=\boldsymbol{v_b}\boldsymbol{G}$, a point on ***C***).

Signcryption of *m* by Alice the Sender

$\boldsymbol{v} \in_R [1,\ldots, \boldsymbol{q}\text{-}1]$

$(\boldsymbol{k}_1, \boldsymbol{k}_2)= hash(\boldsymbol{v}\boldsymbol{P_b})$

$\boldsymbol{c}=E_{k1}(\boldsymbol{m})$

$\boldsymbol{r}=\boldsymbol{KH}_{k2}(\boldsymbol{m}, bind\_info)$

$\boldsymbol{s}=(\boldsymbol{v}/(\boldsymbol{r}+\boldsymbol{v_a})) \bmod \boldsymbol{q}$

(***c, r ,s***) will be sent to Bob.

Unsigncryption of (*c, r, s*) by Bob the Recipient

$\boldsymbol{u}= \boldsymbol{s}\boldsymbol{v_b} \bmod q$

$(\boldsymbol{k}_1, \boldsymbol{k}_2)= hash(\boldsymbol{u}\boldsymbol{P_a}+\boldsymbol{u}\boldsymbol{r}\boldsymbol{G})$

$\boldsymbol{m}= \boldsymbol{D}_{k1}(\boldsymbol{c})$

Accept ***m*** only if $\boldsymbol{KH}_{k2}(m, bind\_info)=\boldsymbol{r}$.

In contrast, ECSCS uses the same manner as the SCS scheme except using the DCDLP to replace DLP. Hence, ECSCS had the same security as SCS under the consumption that ECDLP is hard. ECSCS used four kinds of cryptosystem components: hash function, keyed hash function, symmetric cipher and elliptic curve. Especially, the encryption component was a symmetric cipher. So the scheme can be hardly called a perfect elliptic curve based signcryption. Because of the disadvantage, applications (software or devices) must contain four kinds of cryptosystem platforms can implement the scheme, which is not applicable in practice.

3. A New Signcryption scheme based on ECC

We propose the first signcryption scheme which is really based on Elliptic Curve Cryptosystem in this section.

There is a message m which will be signcrypted and sent to a specific recipient. Alice is a sender. Bob is a specific recipient.

3.1 Description of the new scheme

Choosing an elliptic curve $E(Fq)$ on a finite field Fq (q>max (n, s), is a prime number), G is a base point, ord(G)=l. 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 which is hard in our scheme. $H(\ )$ is a strong one way hash function.

Key generation: A random number $s_A \in \{1,\ldots,l\text{-}1\}$is the private key of Alice. Her public key is a point $P_A = s_A G$. Bob's private key is a random number $s_B \in \{1,\ldots,l\text{-}1\}$. His public key is a point $P_B = s_B G$.

Signcrypt: Alice will complete the following operations to signcrypt the message.

Step 1: Chooses $e \in \{1,\ldots,l\text{-}1\}$at random, and computes $r=H(m \| e)$.

Step 2: Computes $R=rG=(x_1, y_1)$.

Step 3: Computes $rP_B=(x_2, y_2)$.

Step 4: Computes $c=(m \| e) \oplus x_2$.

Step 5: Computes $y= r^{-1} (H(m)+ x_1 s_A) \mod p$.

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

Unsigncrypt: Bob can verify if the signcryption is sent by Alice.

Step1: Computes $s_B R=(x_2', y_2')$.

Step2: Computes $(m' \| e')= c \oplus x_2'$.

Step3: Computes $r'=H(m' \| e')$.Checks if $R \neq r'G$, rejects m'.

Step4: Computes y^{-1}.

Step5: Computes $u= y^{-1} H(m')$, $v= y^{-1} x_1$.

Step6: Computes $(x_1', y_1') =uG +vP_A$. Checks if $x_1 \neq x_1'$, rejects m', else return $m= m'$.

The check in step 3 guarantees that c is a legal ciphertext. Therefore an adversary has no chance to forge an illegal ciphertext and access the unsigncrypt oracle.

3.2 Security of the new scheme

3.2.1 Attack Model and Security Notions for Signcryption

The security of signcryption is different to common signature schemes. We use *random oracle* to describe the attack model for signcryption. We provide two oracles for adversary: *signcryption oracle* and *unsigncryption oracle*.

In common public key cryptosystems, adversary can encrypt a plaintext as his own because the public key is known to all. So the CPA (Chosen Plaintext Attack) is not meaningful. In the case of signcryption scheme, the private key of the sender is required in signcryption. So the adversary is not able to produce signcryptions on its own. We provide the adversary a signcryption oracle for the key of sender. We allow the adversary to choose the randomness inputs by the signcryption oracle, except for challenge signcryption.

Like common public key cryptosystems, signcryption must be unsigncrypted via recipient's private key. So we must provide the adversary an unsigncrytion oracle and allow him to access the unsigncryption oracle for the key of recipient.

The security notions of signcryption and the definition of IND-CCA2 security for

signcryption were given in [5][10].

A signcryption scheme is secure if the following conditions are satisfied:[11]

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.

Unforgeability: It is computationally infeasible for an adaptive attacker to masquerade Alice in creating a signcrypted text.

Confidentiality: It is computationally infeasible for an adaptive attacker to gain any partial information on the contents of a signcrypted text.

The following sub-sections are devoted to discussion of the security of the new signcryption scheme.

3.2.2Unforgeability of the scheme

Just like signature schemes, all of the people have chances to forge a Alice's signcryption. But there is a difference in the ability to forge signatures between the third party and the dishonest specific recipient. In a signature scheme, signer generates a signature using his private key and a secret random number which are confidential to others. In a signcryption scheme, signcryption is generated by the sender using receiver's public key as well as his private key and secret random number. A dishonest recipient has more power to forge, because only the recipient knows the responding private key. We will discuss the probabilities to forge a signcrypion for a dishonest recipient and the third party.

(a) Forged by a dishonest recipient.

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(R, c, y), Bob can use his private key s_B to decrypt the message $m=cx_2^{-1}$. Then the problem will turn into the verification of the signature (R, m, y) which is a normal ECDSA signature to m. ECDSA is known to be unforgeable against adaptive attacks. Therefore the signcryption scheme is unforgeable against adaptive attacks.

(b) Forged by the third Party.

Public keys of sender and recipient are known to a third party. We can also make the signcrypt and unsigncrypt algorithm public to him. Namely, the third party can access both of signcryption oracle and unsigncryption oracle for his randomness input except the challenge signcryption. Given a signcryption (R, c, y), adversary will try to forge a triplet (R', c', y') which will be considered as a legal signcrytion originated from Alice. If the adversary can obtain some information about the private key and secret random number of sender, he has opportunities to success. Under the assumption that $H()$ is a strong one-way function and ECDLP is hard, no information about e will reveal via $r=H(m \| e)$ and $R=rG$. If e is chosen uniformly at random, $y= r^{-1} (H(m)+ x_1 s_A)$ can guarantee the confidentiality of private key.

In this case, the advantage of an polynomial-time adversary $\mathbf{Adv}(A)=|2\mathbf{Pr}[(R, c, y)=(R', c', y')]-1|$ is a negligible function.

3.2.3 Non-repudiation of the scheme

Like others, the proposed signcryption scheme seems to lose the non-interactive repudiation settlement. Bob follows the unsigncryption procedure up until Step 3. Non-repudiation can be acquired. The triplet (R, $H(m')$, y) can be given to the third party and be verified the validity as a common ECDSA. If the triplet couldn't match the equation in Step 4, we will consider

that the signcryption is a fake.

3.2.4Confidentiality of the scheme

Firstly, we discuss the confidentiality of the encryption component in the scheme. Given the triplet($\boldsymbol{R}$, $\boldsymbol{c}$, $\boldsymbol{y}$), ($\boldsymbol{R}$, $\boldsymbol{c}$) is the ciphertext which generated as follow:

$$\begin{cases} \boldsymbol{r}=\boldsymbol{H}(\boldsymbol{m} \parallel \boldsymbol{e}). \\ \boldsymbol{R}=\boldsymbol{rG}=(\boldsymbol{x}_1, \boldsymbol{y}_1). \\ \boldsymbol{rP_B}=(\boldsymbol{x}_2, \boldsymbol{y}_2). \\ \boldsymbol{c}=(\boldsymbol{m} \parallel \boldsymbol{e}) \oplus \boldsymbol{x}_2 \mod \boldsymbol{p}. \end{cases}$$

The encryption component of the scheme is PSEC-1 which is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the elliptic curve decision Diffie-Hellman (EC-DDH) assumption [12]. Thus an adversary couldn't recover any information about message from ciphertext.

Secondly, the common *Encrypt-and-Sign* paradigm cannot be *generically* secure because the signature part can reveal some information about the plaintext message, and this may be true even though the underlying signature scheme is unforgeable[5].

In the scheme, the signature part is ($\boldsymbol{R}$, $\boldsymbol{y}$) which generated as follow:

$$\begin{cases} \boldsymbol{r}=\boldsymbol{H}(\boldsymbol{m} \parallel \boldsymbol{e}) \\ \boldsymbol{R}=\boldsymbol{rG}=(\boldsymbol{x}_1, \boldsymbol{y}_1) \\ \boldsymbol{y}=\boldsymbol{r}^{-1}(\boldsymbol{H}(\boldsymbol{m})+\boldsymbol{x}_1\boldsymbol{s}_A) \end{cases}$$

$\boldsymbol{H}(\ )$ is a strong one way function which guarantee that no information about message is revealed via $\boldsymbol{r}=\boldsymbol{H}(\boldsymbol{m} \parallel \boldsymbol{e})$ and $\boldsymbol{y}=\boldsymbol{r}^{-1}(\boldsymbol{H}(\boldsymbol{m})+\boldsymbol{x}_1\boldsymbol{s}_A)$. Thus, the signature part is confidentiality.

3.3 Efficiency of the new scheme

The most significant advantage of signcryption over Sign-then-Enc lies in the computation cost and communication cost[11]:

Cost(signcryption)<Cost(Signature)+Cost(Encryption).

In this section, the advantage in details will be shown.

The new signcryption scheme combines the ECDSA and PSEC-1. So we construct the ECDSA-then-PSEC-1 composition as an usual Sign-then-Enc scheme. The cost of two schemes will be compared in the subsection.

3.3.1 Communication Cost

Definition 1: In a cryptosystem, $|\boldsymbol{m}|$ denotes the length of plaintext, $|\boldsymbol{c}|$ denotes the length of all the information that must be transferred. ***Message Rate*** can be defined as following:

$$R_M = \frac{|m|}{|c|}.$$

It will be used to measure the communication efficiency of a cryptosystem in this section.

For the same plaintext $\boldsymbol{m}$, the length of all transferred information in signcryption is $|\boldsymbol{R}|+|\boldsymbol{c}|+|\boldsymbol{y}|$, while Sign-then-Enc is $|\boldsymbol{x}_1|+|\boldsymbol{y}|+|\boldsymbol{R}|+|\boldsymbol{c}|$ ($\boldsymbol{x}_1$ is x coordinate of a point). If $|\ \boldsymbol{q}\ |=192$, the signcryption can save 192 bit at least.

$$\frac{R_M(\text{Signcryption})}{R_M(\text{S}-\text{t}-\text{E})} = \frac{|c(\text{S}-\text{t}-\text{E})|}{|c(\text{Signcryption})|} = \frac{|x_1|+|y|+|R|+|c|}{|R|+|c|+|y|} \approx 1.25$$

The signcryption enhance the Message Rate 1.25 times over Sign-then-Enc.

3.3.2 Computation Cost

In the signcryption scheme, the number of computations of multiples of points is 6 (2 in signcryption and 4 in unsigncryption.). A multiple can be obtained in about 1.5|l| point additions[9]. Adapting fast computation, the computation cost for $aG + bP_A$ is (1+ 3/4|q|) point additions, or equivalently 1.17 point multiples. That is, the number of computations of multiples of points can reduced from 6 to 5.17.

In contrast, there are 7 multiples of points in Sign-then-Enc scheme (3 in signing and encryption and 4 in verifying and decryption.). The number can reduced from 7 to 6.17 too.

The numbers of other computations of the two schemes are equal. The signcryption scheme saves 1 multiples of points at least.

The signcryption enhance the computation efficiency 1.19 times over Sign-then-Enc.

4. The Scheme for Multiple Recipients

The scheme can be used to broadcast a message to multiple users in a secure and authenticated manner. Security, unforgeability, non-repudiation and consistency of a message are the major concerns with broadcasting to multiple recipients. Expect the above secure notions, we must prevent a particular recipient form being excluded from the group by a dishonest message originator. The traditional standard practice uses each recipient's public key encrypts the message-encryption key and attach the ciphertext to the signed and encrypted message. A RSA based scheme was given in RFC1421 [13]. Y. Zheng also gave a similar scheme using his SCS signcryption in 1998[11]. Using elliptic curve based signcryption proposed in section 2, we will give a elliptic curve based scheme for multiple recipients.

A message m will broadcasted to t recipients P_1, P_2,…, P_t through a multi-cast channel which allows all recipients will receive an identical part of a broadcast message.

Key generation: A random number $s_A \in \{1,\ldots,l\text{-}1\}$ is the private key of Alice. Her public key is a point $P_A = s_A G$. P_1, P_2,…, P_t. is t recipients. A random number $s_{Bi} \in \{1,\ldots,l\text{-}1\}$ is P_i's private key. The responding public key is a point $P_{Bi} = s_{Bi} G$. Where i =1,…,t. $H(\ )$ is a strong one-way hash function.

Signcrypt: Alice will complete the following operations to signcrypt the message.

Step 1: Chooses $k \in \{1,\ldots,l\text{-}1\}$ at random, and computes $h = H(m \| k)$.

Step 2: Combines the message and its authentication code via $w = m \| h$.

Step 3: Computes $c = w \oplus k$.

Step 4: Creates a signcryption of k for each recipient P_i, i =1,2,…,t.

(a) Chooses $e_i \in \{1,\ldots,l\text{-}1\}$ at random, and computes $r_i = H(k \| e_i)$.

(b) Computes $R_i = r_i G = (x_{1i}, y_{1i})$.

(c) Computes $r_i P_{Bi} = (x_{2i}, y_{2i})$.

(d) Computes $c_i = (k \| e_i) \oplus x_{2i} \mod p$.

(e) Computes $y_i = r_i^{-1} (H(w) + x_{1i} s_A) \mod p$.

Alice broadcasts (c, R_1, c_1, y_1,..., R_t, c_t, y_t) to all recipients.

Unsigncrypt:

Recipient P_i find (c,R_i, c_i, y_i) in (c, R_1, c_1, y_1,..., R_t, c_t, y_t) and verify if the signcryption is sent by Alice.

Step 1: Unsigncrypts k:

(a) Computes $s_{Bi} R_i = (x_{2i}', y_{2i}')$.

(b) Computes $k' \| e_i' = c_i \oplus x_2'^{-1}$.

(c) Checks if $R_i \neq H(k' \| e_i') G$, rejects c.

Step 2: Computes $w' = c \oplus k'$, and splits w' into m' and h'.

Step 3: Computes y_i^{-1}.

Step 4: Computes $u_i = y_i^{-1} H(w')$, $v_i = y_i^{-1} x_{1i}$.

Step 5: Computes $(x_{1i}', y_{1i}') = u_i G + v_i P_A$.

Checks if $x_1 = x_1'$ and $h = H(m' \| k')$, return $m = m'$, else reject m'.

Only if both of the checks success, P_i will accept m as a legal message originated from Alice.

5. The threshold Signcryption with Distributed Key Generation

When the sender is a group which consisted of n members, a threshold signcryption with distributed key generation will proposed in this section.

5.1 Verifiable Secret Sharing Scheme for Elliptic Curve

Since Shamir and Blakley presented the Secret Sharing Scheme independently in 1979[1]. Lots of investigations have been done in the topic. Verifiable Secret Sharing proposed by B.Chor, S.Goldwasser, Micali S. and B.Awerbuch in 1985 is a useful tool to resolve Multi-party computation problems[2]. Feldman proposed the first non-interactive VSS in 1987[14]. An implementation of Feldman'VSS on elliptic curve was described in [15]. Pedersen proposed a non-interactive and information-theoretic secure verifiable secret sharing scheme in 1991[16]. Pedersen'VSS based on Shamir's scheme is a (t, n) threshold. A trusted dealer is required to charge the whole process. In this section, the scheme will be described in an elliptic curve point group. We call it EC-Pedersen Scheme in the following section.

Choosing an elliptic curve $E(Fq)$ on a finite field Fq (q>max (n, s), is a prime number), G is a base point, ord(G)=l. Hence there is a subgroup generated by base point G. The secret to be shared is a private key $s \in Zq$, public key is $Q=sG$. A point H on $E(Fq)$ is generated by G.

Secret Splitting

Step 1: **Dealer** chooses $t \in \{1,\ldots,l-1\}$ at random, computes a commitment to s: $C_0 = C(s,r) = sG + rH$ and open it.

Step 2: **Dealer** chooses a secret polynomial $f(x) = (\sum_{i=0}^{t-1} f_i x^i) \bmod l$, computes $s_i = f(i)$ (i=1,..., n).

Set $f_0 = s$, it is the secret to be shared.

Dealer chooses $g_0,\ldots,g_{t-1} \in \{1,\ldots,l-1\}$, computes a commitment $C_i = C(f_i, g_i) = f_i G + g_i H$ (i=0,..., t-1).

Step 3: Let $g(x) = (\sum_{i=0}^{t-1} g_i x^i) \bmod l$, and let $r_i = g(i)$ (i=1,..., n).

Dealer computes a secret share (s_i, r_i)(i=1,..., n) and send the share to player P_i through a perfect private channel.

Dealer computes commitments $C_j = f_j G$ (j=0,..., t-1) which will be broadcasted to the whole group and be used to verify the shares later.

Sharers verify

When a player receives (s_i, r_i), he checks if $C(s_i, r_i) = \sum_{j=0}^{t-1} i^j C_j$.

If the test fails, r_i will be rejected because it is an illegal data.

Secrets reconstruct

Only t players out of group can reconstruct the secret polynomial by Lagrange Polynomial Interpolation as following

$$f(x) = \sum_{j=1}^{t} s_j \prod_{\substack{h=1 \\ h \neq j}}^{t} \frac{x-h}{j-h} \bmod l .$$

Let x=0, with the formula $s = \sum_{j=1}^{t} s_j \prod_{\substack{h=1 \\ h \neq j}}^{t} \frac{h}{h-j} \bmod l$, the secret s can be recovered.

The scheme can defend the attack launched by (n -1)/3 players.

5.2 A Secure Distributed Key Generation for ECC

The above scheme requires a trusted dealer to manage the whole process. Unfortunately, we can hardly to look for a trusted dealer in substantial circumstances. Distributed Key Generation (DKG) is a novel protocol which can distribute a secret in a group. Pedersen proposed the first DKG scheme in 1991[3]. R.Gennaro pointed out the scheme is insecurity and give a secure DKG scheme for Discrete-Log Cryptosystems in 1999[4]. We will give an implantation of the Gennaro's DKG scheme which will be called EC-DKG in the following. Under the assumption that ECDLP is hard, EC-DKG has the equal security as original scheme while more efficient.

1. Each player P_i performs a Pedersen-VSS of random number z_i as a dealer:

(a) P_i chooses two random polynomials $f_i(z), f'_i(z)$ over Z_q of degree t:

$$f_i(z) = \sum_{j=0}^{t-1} a_{ij} z^j \quad , f_i'(z) = \sum_{j=0}^{t-1} b_{ij} z^j$$

Let $z_i = z_{i0} = f_i(0)$. P_i broadcasts $C_{ik} = a_{ik}G + b_{ik}H$ for k=0,…,t-1. P_i computes the shares $s_{ij} = f_i(j)$, $s'_{ij} = f'_i(j)$ for j=1,2,…,n and sends s_{ij}, s'_{ij} to player P_j.

(b) Each player P_i verifies the shares he received from the other players.

For each i=1,…,n, P_j checks if $s_{ij}G + s'_{ij}H = \sum_{k=0}^{t-1} j^k (C_{ik})$ (1)

If the check fails for an index i, P_j broadcasts a *complaint* against P_i.

(c) Each player P_i who, as a dealer, received a complaint from player P_j broadcasts the values s_{ij}, s'_{ij} that satisfy Eq 1.

(d) Each player marks as *disqualified* any player that either received more than t complaints in Step **1b**, or answered to a complaint in Step **1c** with values that falsify Eq.1.

2. Each player then builds the set of non-disqualified players ***QGroup***.

3. The distributed secret value $x = \sum_{i \in QGroup} z_i$, but it is not explicitly computed by any party.

Each player P_i sets his share of the secret as $x_i = \sum_{j \in QGroup} s_{ji}$ and $x'_i = \sum_{j \in QGroup} s'_{ji}$.

4. Each player $i \in QGroup$ exposes $PK_i = z_i G$ via Feldman VSS:

(a) Each player P_i, $i \in QGroup$, broadcasts $A_{ik} = a_{ik} G$ for k=0,…,t-1.

(b) Each player P_j verifies the values broadcast by the other plalyers in ***QGroup***, namely, for each $i \in QGroup$, P_j checks if

$$s_{ij} G = \sum_{k=0}^{t-1} j^k A_{ik} \qquad (2)$$

If the check fails for an index i, P_i *complains* against by broadcasting the values s_{ij}, s'_{ij} that satisfy Eq. 2 but do not satisfy Eq. 2.

(c) For players P_i who receive at least one valid complaint , i.e. values which satisfy Eq. 1 AND NOT Eq. 2, the other players run the reconstruction phase of Pedersen-VSS to compute z_i, $f_i(z)$, A_{ik}, for k=0,…,t-1 in the clear. For all players in ***QGroup***, set $PK_i = A_{i0} = z_i G$.

Compute $PK = \sum_{i \in QGroup} PK_j$.

5.3 The Threshold Protocol of the Scheme with Distributed Key Generation

In this section, we will construct a threshold protocol about the above scheme, which is a multi-party secure computation problem.

5.3.1 Secret Distributed Generation

We use KDG scheme showed in section 5.2 to generate and verify the secret. In the protocol, ***Group***=$\{p_1', p_2', \ldots, p_n'\}$ is a group including n players which will generate share the secret s (a private key of ***Group***) and random number e distributed. ***QGroup***=$\{p_1, p_2, \ldots, p_t\}$ is a qualified subset of ***Group*** including t out of n players. All the players perform the protocol in section 5.2. At the end of the protocol, each player $P_i \in QGroup$ will have a private key share s_i and a secret number share e_i.

5.3.2 Signcryption with Distributed Key Generation

In this stage, ***QGroup*** will signcrypt a message for Bob. Bob verifies whether the signature is correct, then decrypt the cipher.

Threshold Signcryption generation

No subset of t-1 players of ***Group*** can generate a signcryption. Each player $P_i \in QGroup$ complete the following steps:

Step 1: Generates a uniform random number.

(a)Computes $E_i = e_i G$, which called random point, and broadcasts it.

(b) Chooses t random points, computes a point E by Lagrange Polynomial Interpolation as following:

$$E = \sum_{i=1}^{t} \left(\prod_{\substack{h=1 \\ h \neq j}}^{t} \frac{h}{h-j} \bmod l\right) e_i G = (x_e, y_e).$$

(c) Sets $e = x_e$. It will be used as a random number in signcryption.

Step 2: Compute $r=H(m \| e)$.

Step 3: Computes $R=rG=(x_1, y_1)$.

Step 4: Computes $rP_B=(x_2, y_2)$.

Step 5: Computes $c=(m \| e) \oplus x_2$.

Step 6: Computes $y_i = r^{-1}(H(m)+x_1 s_i) \mod p$.

The triplet (R, c, y_i) is the signcryption piece and will be sent to Bob.

Unsigncrypt: After received t pieces of signcryption generated by the ***QGroup***. Bob can verify if the signcryption is sent by ***Group***.

Step1: Computes the public key of ***QGroup*** via $PK = \sum_{i \in QGroup} PK_j$.

Step2: Chooses t pieces of signcryption, constructs y by Lagrange Polynomial Interpolation.

He can obtain a new triplet (R, c, y) which is a signcryption generated by ***QGroup***. Then he can unsigncrypt it follows the steps in section 2.

6. Conclusion

The proposed signcryption scheme combines ECDSA and PSEC-1. Both of the schemes are secure against the adaptive chosen ciphertext attack. The results in section 3.2 show that the attacks against confidentiality, unforgeablity and non-repudiation of the proposed scheme are equivalent to attacks against ECDSA and PSEC-1 separately. So the proposed scheme is secure against the adaptive chose ciphertext attack too.

Compared with ECSCS, the signcryption scheme proposed in this paper uses a uniform elliptic curve cryptosystem computation platform and a set of parameters. While Y. Zheng's ECSCS scheme uses four kinds of cryptography components: symmetrical cipher, hash function, keyed hash function and elliptic curve based computation. Though its computation cost is lightly loser than our signcryption, ECSEC's high prices in practice make it not applicable. In other word, an application (software or device) which must contain four kinds of cryptosystem platform can implement ECSCS. Hence the proposed scheme is more feasible than others.

References

[1] A. Shamir, “How to Share a Secret”, Communications of the ACM, 22(11), 1979:612-613

[2] Chor B.,Goldwasser S. Micali S. Awerbuch B. “Verifiable secret sharing and achieving simultaneity in the presence of faults.”, Proceedings of 26th IEEE symposium on foundations of computer science, 1985:151-160

[3] T.Pedersen. A threshold cryptosystem without a trusted party. Advances in Cryptology-EUROCRYPT’91, LNCS547. Springer-verlag,1991:522-526

[4] R.Gennaro, S.Jarecki, H. Krawezyk and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Advances in Cryptology-EUROCRYPT’99. Springer-verlag,1999:295-310

[5] J.Baek, R.Steinfeld and Y.Zheng. Formal Proofs for the Security of Signcryption. Public Key Cryptography 2002, LNCS2274, Springer-Verlag, 2002:80-98

[6] M.Bellare, A.Desai, D.Pointcheval and P.Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. Advances in Cryptology-CRYPTO’98, LNCS1462, Springer-Verlag,1998:26-45

[7] M.Bellare and C. Namprepre. Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm, Advances in Cryptologhy-Proceedings of ASIACRYPT’2000, LNCS1976, Springer-Verlag. 2000:531-545

[8] H. Krawczyk. The Order Of Encryption And Authentication For Protecting Communications (Or: How Secure Is SSL?). Advances In Cryptology-Proceedings of CRYPTO’2001, LNCS2139, Springer-Verlag. 2001:310-331

[9] Y. Zheng, H. Imai. How to construct efficient signcryption schemes on elliptic curves. Information Processing Letters 68, 1998: 227–233

[10] J. Malone-Lee and W. Mao. Two birds one stone: Signcryption using RSA . Topics in Cryptology - Cryptographers' Track, RSA Conference - 2003, LNCS 2612, Springer-Verlag, 2003:210-224.

[11]Y. Zheng. Signcryption and Its Applications in Efficient Public Key Solutions. Information Processing Letters, Vol.68, 1998: 227-233

[12] 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/contributions/psec.pdf

[13] J. Linn, Privacy enhancement for internet electronic mail: Part I: Message encryption and authentication procedures. RFC 1421 IETF 1993

[14] Feldman P. “A Practical Scheme for Mon-Interactive Verifiable Secret Sharing”, Proceedings of 28th IEEE symposium on Foundations of Computer Science, 1987:427-437

[15]Han Yiliang, Yang Xiaoyuan, Sun Jun, Li Delong, “Verifiable Threshold Cryptosystems Based on Elliptic Curve”, Proc of ICCNMC’2003, IEEE Computer Society, 2003:334-337

[16]T.P.Pedersen, “Distributed Provers with Applications to Undeniable Signatures”, Proc of Eurocrypt’91, Lecture Notes in Computer Science, LNCS 547, Berlin: Spring-Verlag, 1991: 221-238

Authors:

Yiliang Han: born in 1977, Lecturer of Engineering College of Armed Police Force, member of CCF. Research area: cryptology , intrusion tolerance. Have published 20 papers on Network Security and Privacy.

Xiaoyuan Yang: born in 1959, Chief professor of Key Lab. on Network and Information Security of Armed Police Force. Research area: cryptology and information security.