Short Signcryption Scheme for the Internet of Things

Xuanwu Zhou1,2, Zhigang Jin1, Yan Fu1,2, Huaiwei Zhou3, Lianmin Qin3
1 School of Electronics and Information Engineering, Tianjin University, Tianjin 300072, China
2 Command College of the Chinese Armed Police Forces, Tianjin 300250, China
3 Information Technology Research Center, Huarong Corporation, Yantai 265400, China
E-mail: schwoodchow@163.com

Keywords: internet of things, signcryption, provable security, distributed key management, system efficiency

Received: June 23, 2011

Signcryption is an effective cryptographic primitive, which simultaneously fulfils both the functions of encryption and signature with much lower cost than traditional schemes; it is an ideal method to provide confidentiality and unforgeability and ensure secure data storage and transmission in the IOT (Internet of things). In the paper, we propose a publicly verifiable short signcryption scheme S-ECSC for the Internet of things based on elliptic curves cryptosystem; and prove the provable security of S-ECSC under the Random Oracle model, including confidentiality in IND-CCA2 model, unforgeability in UF-CMA model and non-repudiation security. As per the efficiency analysis, S-ECSC achieves an average 80% reduction in computation cost compared with typical discrete logarithm, RSA based signcryption schemes, and has the lowest communication cost in Elgamal type signcryptions. With its superiority in efficiency and security, S-ECSC proves to be more suitable for resource-restricted environment in IOT and better satisfies the requirement of secure protocols in IOT, such as key management, secure routing, etc. At last, we take key generating and distributing protocol of distributed key management in IOT as an application example, and analyse the method and importance to apply S-ECSC into secure protocols in IOT.

Povzetek: Članek opisuje šifrirno shemo za internet stvari.

1 Introduction

The concept of IOT (Internet of Things) was first put forward by Ashton of the former MIT Auto-ID Center in 1999 when he was working on RFID (Radio Frequency Identification). Presently, the most widely-accepted definition of IOT is as follows [1, 2, 3, 4]. IOT is a self-configuring network in which things are connected with network according to certain protocols with RFID, ultra-red sensor, GPS (Global Positioning System), laser scanner, etc to interchange and transmit data, and ultimately achieve intelligent identification, positioning, tracing, supervision and management. IOT is the new direction of future computer and communication technology, and is regarded as the third landmark in the development of information technology after computer science and Internet.

According to the function classification, the hierarchical structure of IOT is composed of application layer, network layer and sensor layer. The basic function of network layer is secure and reliable interconnection between things via wire-based or wireless technology, in which the secure and dynamic interconnection via wireless network has been the overwhelming trend. In wireless network technology, many researchers have focused on IEEE802.11 WLAN (Wireless Local Area Network), which is mainly composed of wireless Ad hoc network, WSN (Wireless Sensor Network) and WMN (Wireless Mesh Network). As a new wireless network, IEEE802.11 WLAN proves to be suitable for commercial, medical, domestic, military, and other applications with its superiority, such as inexpensiveness, adaptability and reliability, etc. In the Internet of things, IEEE802.11 WLAN has been playing an increasingly important role in secure and reliable connection between different objects. Whereas, the distributed network management and restricted network resources in IEEE802.11 WLAN have rendered many problems as to the security of confidentiality, integrity, non-repudiation and availability for data storage and transmission in IOT. Besides, security measures designed for traditional network, which has relatively abundant network resources, fixed connection, stable topology, special routing and comprehensive network service, are not completely applicable to wireless network environment in the Internet of thing. Therefore, it is of great necessity to design special security technology, protocols and corresponding algorithms for the secure and dynamic wireless communication in the Internet of things.

The confidentiality and integrity of message is the basic requirement for secure communication in IOT; in the symmetric setting, efforts focused on the composition of symmetric key encryption and message authentication code (MAC). In asymmetric settings, the composition method of “signature-then-encryption” has been
employed. But these have all proved impractical not only for the insecurity in case of arbitrary schemes but also for the low efficiency regarding application into resource-restricted environment in IOT, which results from the sum cost of encryption and signature.

In 1997, Zheng proposed a cryptographic primitive “signcryption” [5], which simultaneously fulfills the integrated function of public encryption and digital signature with a computing and communication cost significantly smaller than that required by the “signature-then-encryption” method. Since then, signcryption has been a focus of cryptography as an ideal method to simultaneously provide confidentiality and unforgeability and many researchers have explored the application of signcryption in different security protocols [6, 7, 8, 9, 10, 11]. The study of signcryption algorithms suitable for IOT network environment and its application in IOT security schemes is an important direction in cryptography; it is more of a requirement from the rapid development of the Internet of things than just a requirement from the theoretical or applied cryptography research.

In order to improve the security and efficiency of communication in the Internet of things, we propose a publicly verifiable short signcryption scheme S-ECSC for the Internet of things based on elliptic curves cryptosystem; and prove the provable security of S-ECSC under the Random Oracle model, including confidentiality in IND-CCA2 model, unforgeability in UF-CMA model and non-repudiation security. At last, we take key generating and distributing protocol for different terminals of distributed key management in IOT as an application example, and analyse the method and importance in the application of S-ECSC into secure protocols in IOT. Compared with other typical discrete logarithm, RSA and elliptic curves based signcryption schemes, S-ECSC is more suitable for resource-restricted environment in IOT communication with its superiority in computing and communication cost and can better satisfy the requirement of secure protocols in IOT, such as key management, secure routing, etc.

2 Short Signcryption Scheme on Elliptic Curves

First, we pin down the basic notions concerning signcryption which will facilitate the design and analysis of the short signcryption scheme.

2.1 Basic Notions in Signcryption

Definition 2.1.1 (Elliptic Curve) An elliptic curve $E(F_q)$ over finite field $F_q$ is a sextuple: $T = (q, a, b, P, l, h)$, where $P = (x_p, y_p)$ is the base point of $E(F_q)$, prime $l$ is the order of $P$. As to $t \in Z_l^*, \ Q \in E(F_q), \ Q = tG$ denotes multiple double additions on elliptic curve. $O$ is the point at infinity, satisfying $lP = O$ and $G + O = G$ for any point $G \in E(F_q)$.

Definition 2.1.2 (ECDLP, Elliptic Curve Discrete Logarithm Problem). ECDLP is the following computation:

$$x \leftarrow \text{ECDLP}(Q, P).$$

$P$ is a base point and $Q \in \langle P \rangle, \ x \in Z_l^*, \ Q = xP$.

Definition 2.1.3 (Signcryption Scheme) A signcryption scheme $\Sigma = (\text{GC}, \text{GK}, \text{SC}, \text{USC})$ consists of the following algorithms:

1. Probabilistic common parameters generation algorithm $\text{GC}(1^k)$ takes security parameter $1^k$ as input and returns a sequence of common parameters such as description of computational groups and hash functions.

2. Key generation algorithm $\text{GK}(1^k)$, which is also probabilistic, takes identity and security parameter as input and returns secret/public key-pair $(sk_{ID}, pk_{ID})$.

$$sk_{ID}, pk_{ID} \leftarrow \text{GK}(1^k).$$

3. Signcryption algorithm $\text{SC}(sk_A, pk_B, m)$ that takes sender's secret key $sk_A$, receiver's public key $pk_B$ and message $m \in \text{SP}_m$ ($\text{SP}_m$ is the message space) as input and returns signcryption text $C$ or $\bot$ (a reject symbol). It is also probabilistic algorithm.

$$C \cup \{\bot\} \leftarrow \text{SC}(sk_A, pk_B, m).$$

4. Deterministic unsigncryption algorithm $\text{USC}(sk_B, pk_A, C)$ takes as input receiver's secret key $sk_B$, sender's public key $PK_A$ and signcryption text $C$, and returns either message $m$ or $\bot$.

$$m \cup \{\bot\} \leftarrow \text{USC}(sk_B, pk_A, C).$$

If the signcryption scheme is publicly verifiable, it is composed of an additional public verification algorithm $PV$.

5. Deterministic public verification algorithm $PV(pk_A, pk_B, C, R)$ takes as input public key pair $(pk_B, pk_A)$, signcryption text $C$ and parameter $R$, and returns either “true” or $\bot$.

“true” $\cup \{\bot\} \leftarrow PV(pk_A, pk_B, C, R)$.

2.2 S-ECSC Signcryption Algorithm

Short signcryption scheme $S-ECSC = (\text{GC}, \text{GK}, \text{SC}, \text{USC}, PV)$

Common parameters generation

$\text{GC}(1^k) = \{\text{On input } (1^k) : \}$

$$K : E(F_q) \rightarrow \{0, 1\}^{|z|}, \ H : \{0, 1\}^* \rightarrow Z_l^*, \ (K, H, T) \leftarrow \text{GC}(1^k).$$
\[ T = (q, a, b, P, l, h), \text{where } P = (x_P, y_P) \text{ is the base point of } E(F_q), \text{ord}(P) = l \text{ is a prime, } O \text{ is the point at infinity.} \]

**Key pair generation**

\[ GK(\ A, 1^k) = \text{"On input } (A, 1^k): \]

\[ sk_A \leftarrow Z^* \cap PK_A = sk_A P \neq O, \]
\[ (sk_A, PK_A) \leftarrow \ldots \]

\[ GK(\ B, 1^k) = \text{"On input } (B, 1^k): \]

\[ sk_B \leftarrow Z^* \cap PK_B = sk_B P \neq O, \]
\[ (sk_B, PK_B) \leftarrow \ldots \]

**Signcryption**

\[ SC(\ sk_A, PK_B, m) = \text{"On input } (sk_A, PK_B, m): \]

If \( sk_A \not\in Z^* \) or \( PK_B \not\in P \) return \( \perp \),
\[ r \leftarrow Z^* \cap R = (x_R, y_R) \leftarrow rPK_B, \]
\[ \sigma \leftarrow K(R), c \leftarrow E_{\sigma}(m), \]
\[ h \leftarrow H(\ m \oplus PK_A \oplus PK_B \oplus R), \]
\[ s = (hsk_A + r) \mod l, \]
\[ C = (c, h, s): \]

Symmetric encryption scheme \( \mathcal{Y} = (E, D) \) is an encryption scheme with passive indistinguishability defined in Definition 3.2.8.

**Unsigncryption**

\[ USC(\ sk_B, PK_A, C) = \text{"On input } (sk_B, PK_A, C): \]

If \( sk_B \not\in Z^* \) or \( PK_A \not\in P \) return \( \perp \),

Parse \( C \) into \( (c, h, s) \),

If \( h, s \not\in Z^* \) or \( c \not\in SP \) return \( \perp \),

Else \( I = sp - hPK_A \), \( R = sk_B I = (x_R, y_R) \),
\[ \sigma \leftarrow K(R), m \leftarrow D_{\sigma}(c), \]
\[ h' \leftarrow H(\ m \oplus PK_A \oplus PK_B \oplus R), \]

If \( h = h' \) return \( m \), else return \( \perp \).

**Public Verification**

If controversy arises between signcryption senders and receiver, a trusted third party can solve the repudiation. The third party evaluates the following formula after signcryption receiver publishing \( (R, C) \).

\[ PV(\ PK_A, PK_B, C, R) = \]

\[ \text{"On input } PK_A, PK_B, C, R: \]

Parse \( C \) into \( (c, h, s) \),

\[ \sigma \leftarrow K(R), m \leftarrow D_{\sigma}(c), \]
\[ h' \leftarrow H(\ m \oplus PK_A \oplus PK_B \oplus R), \]

If \( h = h' \) return \text{true}, else return \( \perp \).

### 3 Provable Security of S-ECSC

In this section, we will analyse the provable security of the signcryption in random oracle model, including confidentiality, unforgeability and non-repudiation.

#### 3.1 Correctness of S-ECSC

**Definition 3.1.1** Message space \( Message(\ sk_A, PK_B) \) is the set of all \( m \) associated to each private/public key pair \( (sk_A, PK_B) \) output by \( GK(\ ID, 1^k) \) for which \( SC(\ sk_A, PK_B, m) \) never returns \( \perp \).

**Definition 3.1.2** A signcryption scheme \( \Sigma = (GC, GK, SC, USC) \) is correct if \( USC(\ sk_A, PK_A, C) = m \) for any private/public key pair \( (sk_A, PK_B) \) output by \( GK(\ ID, 1^k) \), any message \( m \in Message(\ sk_A, PK_B) \), and any \( C \neq \perp \) that might be output by \( SC(\ sk_A, PK_B, m) \).

**Theorem 3.1.1** S-ECSC is correct for any private/public key pair \( (sk_A, PK_B) \) output by \( GK(\ ID, 1^k) \), any message \( m \in Message(\ sk_A, PK_B) \) and any \( C \neq \perp \) that might be output by \( SC(\ sk_A, PK_B, m) \).

**Proof of correctness:** Obviously, the signcryption scheme S-ECSC is correct if and only if \( USC(\ SC(\ sk_A, PK_B, m)) = m \).

As per the formula in the scheme,
\[ sk_B = sk_B(sP - hsk_A P) = sk_B(s - hsk_A) P = sk_B P = rPK_B = R = (x_R, y_R), \]
\[ \sigma \leftarrow K(R), \]
\[ m \leftarrow D_{\sigma}(c), \]
\[ h' \leftarrow H(\ m \oplus PK_A \oplus PK_B \oplus R), \]
\[ m = h', \text{ } m \leftarrow USC(\ sk_B, PK_A, C). \]

Thus \( USC(\ SC(\ sk_A, PK_B, m)) = m \), the short signcryption S-ECSC is correct, as desired.

#### 3.2 Confidentiality of S-ECSC

**Definition 3.2.1** Computational Elliptic Curve Problem (CECP). Let \( T = (q, a, b, P, l, h) \) be an elliptic curve and \( AC \) an attacker on CECP, CECP is defined as the following:

**Experiment** \( EXP_{CECP}(AC) \)

\[ d, e \leftarrow Z^*, \]
\[ D = dP, E = eP, \]
\[ F \in \langle P \rangle \leftarrow AC^T(D, E). \]

If \( F = deP \) return 1 else return 0.

Note that \( F = deP = dE = eD \). (1)
**Definition 3.2.2** Decisional Elliptic Curve Problem (DECP). Let \( T = (q, a, b, P, l, h) \) be an elliptic curve and \( AD \) an attacker on DECP. DECP is defined as the following:

**Experiment** \( EXP_{T}^{DECP} (AD) \)

- \( b \leftarrow \{0, 1\} \).
- If \( b = 0 \) \( \exists e_{0} : d, e, f \leftarrow Z_{l}^{*} \).
- If \( b = 1 \) \( \exists e_{1} : d, e \leftarrow Z_{l}^{*}, f = de(\text{mod} l) \).
- \( D = dP, E = eP, F = fP \).
- \( b' \leftarrow AD^{\tau} (D, E, F) \).
- If \( b' = b \) return 1 else return 0.

**Definition 3.2.3** DECP Oracle \( O_{DECP}^{T} \). Let \( T = (q, a, b, P, l, h) \) be an elliptic curve, DECP Oracle is defined as the following:

- \( O_{DECP}^{T} \leftarrow \text{on input } (P, D, E, F) \)
- If \( D, E, F \not< P \) return \( \bot \), else
- If \( DECP (D, E, F') = 1 \) return 1,
- If \( DECP (D, E, F') = 0 \) return 0.

**Definition 3.2.4** Elliptic Curve Gap Problem (ECGP). Let \( T = (q, a, b, P, l, h) \) be an elliptic curve and \( AEGC \) an attacker on ECGP, let’s consider the following experiment:

**Experiment** \( EXP_{T}^{ECGP} (AEGC) \)

- \( d, e \leftarrow Z_{l}^{*} \).
- \( F = fP \leftarrow AEGC_{T}^{O_{DECP}^{T}(\sigma)} (d, e) \).
- If \( f = de(\text{mod} l) \) return 1 else return 0.

The ECGP advantage of ECGP is defined as \( Adv_{T}^{ECGP} (AEGC) = Pr (EXP_{T}^{ECGP} (AEGC) = 1) \).

**Hypothesis 3.2.1** (ECGP is hard). Given elliptic curve \( T \) and secure parameter \( l \), the probability of solving ECGP in time \( t \) is \( \xi (1^{t}, T) \) which is negligible, that is

\[
\xi (1^{t}, T) = Pr (d, e \leftarrow Z_{l}^{*}, Q = xP : EXP_{T}^{ECGP} (AEGC) = 1). \]

**Definition 3.2.5** Left-or-right signcryption oracle. Let \( \Sigma = (GC, GK, SC, USC) \) be a signcryption scheme, a left-or-right signcryption oracle is defined as follows.

- **Oracle** \( SC_{t, PK_{a}} (LR(m_0, m_1, b)) \)
  - “On input \( (m_0, m_1) \):
    - \( b \in \{0, 1\}, m_0, m_1 \in SP_{m} \),
    - \( C \leftarrow SC (sk_{d}, PK_{b}, m_0) \),
  - Return \( C \).”

**Definition 3.2.6** Confidentiality of signcryption. Let \( ASC \) be an algorithm against the confidentiality of signcryption oracle and returns a bit. We consider the following experiment:

**Experiment** \( EXP_{ASC}^{ind-cca2} (ASC) \)

- \( (K, H, T) \leftarrow GC (1^{t}) \)
- \( (s_{k}, PK_{a}) \leftarrow GK (A, 1^{k}) \)
- \( (s_{k}, PK_{b}) \leftarrow GK (B, 1^{k}) \)
- \( C' \leftarrow SC_{s_{k}, PK_{a}} (LR(m_0, m_1, b)), b \leftarrow \{0, 1\} \)
- \( b' \leftarrow ASC_{s_{k}, PK_{a}} (LR(m_0, m_1, b)) \)
- If \( ASC \) queried \( USC (sk_{d}, PK_{b}) \) on a signcryption text previously returned by \( SC_{s_{k}, PK_{a}} (LR(m_0, m_1, b)) \) then return 0.
- If \( b' = b \) return 1 else return 0.

The IND-CCA2 advantage of ASC is defined as

\[
\delta (k) = Adv^{ind-cca2}_{ASC} \left( ASC \right) = Pr (EXP_{ASC}^{ind-cca2} (ASC) = 1). \]

A signcryption scheme is indistinguishable under adaptive chosen cipher-text attack if the IND-CCA2 advantage of any attacker ASC with reasonably restricted resources (time-complexity, frequency and length of queries) is negligible.

**Definition 3.2.7** Left-or-right Encryption Oracle. Let \( Y = (E, D) \) be the symmetric encryption algorithm in the signcryption scheme, a left-or-right encryption Oracle is defined as:

- **Oracle** \( E_{\sigma} (LR(m_0, m_1, b)) \)
  - “On input \( (m_0, m_1) \):
    - \( b \in \{0, 1\}, m_0, m_1 \in SP_{m} \),
    - \( C \leftarrow E_{\sigma} (m_0) \),
  - Return \( C \).”

**Definition 3.2.8** Passive Indistinguishability. Let \( AI \) be an algorithm against the passive indistinguishability of symmetric encryption scheme \( Y \), which has access to a left-or-right encryption oracle and returns a bit. We consider the following experiment:

**Experiment** \( EXP_{AI}^{pl} (AI) \)

- \( C' \leftarrow E_{\sigma} (LR(m_0, m_1, b)) \),
- \( b \leftarrow \{0, 1\}, b' \leftarrow AI_{E_{\sigma}(LR(b))} \),
- If \( b' = b \) return 1 else return 0.

The \( pl \) advantage of \( AI \) is defined as

\[
\nu (k) =Adv^{pl}_{AI} (AI) = Pr (EXP_{AI}^{pl} (AI) = 1). \]

An encryption scheme is passively indistinguishable if the \( pl \) advantage of any attacker \( AI \) with reasonably restricted resources (time-complexity, frequency and length of queries) is negligible.

**Hypothesis 3.2.2** (Ideal Hash Function). Hash function has the property of Random Oracle. Namely, the outputs of hash function are randomly and uniformly distributed.

**Theorem 3.2.1** If there exists an algorithm \( ASC \) against the IND-CCA2 property of signcryption scheme \( \Sigma \) in time \( t \) with non-negligible advantage \( \delta (k) \), using \( q_{SC} \)
queries to its signcryption oracle and \((q_H,q_M)\) queries to its random oracles. Then we can thus formulate an AECG attacker on ECGP with non-negligible advantage \(\xi \cdot T^t\) in time \(t\), using \(q_{SC}\) queries to its signcryption oracle and \(q_H\) queries to its random oracles.

**Proof of confidentiality:** In our proof, the random oracles \(K\) and \(H\) are replace by the random oracle simulators with two types of “query-answer” lists. For example, \(Sim_K\) simulates random oracle \(K\) with two types of “query-answer” lists \(L_1^K\) and \(L_2^K\). \(L_2^K\) consists of simple “query-answer” ( \((R,\sigma)\) entries from \(K\), while \(L_2^K\) consists of special input-output entries \((PK_b \oplus \Omega_i \oplus (?,\sigma))\) which implies \(\sigma = K(\Omega^{ks})\) with the implicit input \(\Omega^{ks}\) stored and denoted as “\(\sigma\)’.”

**Sim** \((L_1^K,R)\) on input \((L_1^K,R)\):

**Else** if \(1 \leftarrow DECP(D,PK_b,R)\) return \(\perp\) \(\Box\).

**Else if** \(1 \leftarrow DECP(\Omega_i,PK_b,R)\) return \(\sigma_i\). //there is an entry \((PK_b \oplus \Omega_i \oplus (?,\sigma))\) in \(L_1^K\)

**Else if** \(R = R_i\) , return \(\sigma_i\) //there is an entry \((R_i,\sigma_i)\) in \(L_1^K\),

\[\sigma_i \leftarrow \{0,1\}^{l_{c_{PK_i}}} , R_i \leftarrow R, \]

\[\text{Add} \ (R_i,\sigma_i) \text{ into } L_1^K . \]

**If** \(EXP_{SCG}^{ind-cca2}(ASC)\) makes queries to random oracle \(O^{H}\), AECG will reply with simulator \(Sim_H\).

**Sim** \((L_1^H,h)\) on input \((L_1^H,h)\): // \(h = (m \oplus PK_A \oplus PK_b \oplus R)\)

If \(1 \leftarrow DECP(D,PK_b,R)\) return \(\perp\),

**Else if** \(1 \leftarrow DECP(\Omega_i,PK_b,R)\) and \(m \oplus PK_A \oplus PK_b = m_i \oplus (PK_A_i) \oplus (PK_b_i)\) return \(\sigma_i\) //there is an entry \((\Omega_i \oplus m_i \oplus (PK_b_i)) \oplus ? \oplus \Omega_i \oplus ? \oplus m_i \oplus (PK_A_i)\).

Else if \(h = h_i\), return \(h_i\) //there is an entry \((h_i,h_i)\) in \(L_1^H\),

\[h_i \leftarrow \{0,1\}^{l_{c_{PK_i}}} , h_i \leftarrow h , \text{Add} \ (h_i,h_i) \text{ into } L_1^H . \]

**If** \(EXP_{SCG}^{ind-cca2}(ASC)\) makes queries to random oracle \(O^{SC}\), AECG will reply with simulator \(Sim_{SC}\).

**Sim** \((L_1^K,L_1^H,PK_A,PK_b,m)\) =

“On input \((L_1^K,L_1^H,PK_A,PK_b,m)\): // \(h = (m \oplus PK_A \oplus PK_b \oplus R)\)

\[\sigma \leftarrow \{0,1\}^{l_{c_{PK_i}}} , c \leftarrow E_{\sigma}(m), \]

\[h \leftarrow Z_i^\ast , s \leftarrow {Z_i^\ast} , \]

\[\Omega \leftarrow sP-hPK_A, \Omega_i \leftarrow \Omega, \]

\[\sigma_i \leftarrow \sigma , m_i \leftarrow m, h_i \leftarrow h , \]

**Add entry** \((\Omega_i,\sigma_i)\) into \(L_2^H\),

**Add entry** \((m \oplus PK_A \oplus PK_b \oplus (?,h_i))\) into \(L_2^H\),

\[C = (c,h,s) \text{ and return } C . \]

If \(EXP_{SCG}^{ind-cca2}(ASC)\) makes queries to \(O^{USC}\), AECG will reply with simulator \(Sim_{USC}\).

Sim \((L_1^K,L_1^H,D,PK_A,PK_b,C)\) =

“On input \((L_1^K,L_1^H,D,PK_A,PK_b,C)\):

**Parse** \(C\) into \((c,h,s)\), \(\Omega \leftarrow sP-hPK_A\),

If \(\Omega = D\) return \(\perp\),

**Else if** \(\Omega \oplus (\Omega_i,\sigma_i)\) into \(L_1^H\), \(m \leftarrow D_{\sigma}(c)\),

or \(\Omega \oplus (\Omega_i,\sigma_i)\) into \(L_1^H\), \(m \leftarrow D_{\sigma}(c)\),

**Add entry** \((m_i \oplus (PK_A_i) \oplus (PK_b_i)) \oplus \Omega \oplus \sigma_i\) into \(L_2^H\),

**Add entry** \((PK_b_i) \oplus (PK_A_i) \oplus PK_b\) into \(L_2^H\),

Then \(h' \leftarrow h\),

**Else** \(h = h_i\) return \(\perp\).

Based on Theorem 3.2.1 we formulate an AECG attacker on ECGP; apparently contradicting Hypothesis 3.2.1, thus prove the confidentiality of the improved signcryption scheme.

The AECG attacker on ECGP is formulated as follows.

**AECG** \((T,D,E)\) =

“On input \((T,D,D = dP,E = eP)\):

\[h^* , s^* \leftarrow \{Z_i^\ast \} , PK_A \leftarrow (h^*)^{-1}(s^*P + D) , \]

\[PK_b \leftarrow E \sigma \leftarrow \{Z_i^\ast \} , \]

\[C = (c^*,h^*,s^*) \leftarrow SC_{sk,b,PK_b} (LR(m_i,m_i,b)) , \]

Short Signcryption Scheme for... Informatica 35 (2011) 521–530 525
\[ c^* \leftarrow E_{\sigma^*}(m_i), \text{ and the random oracle queries } O^* \text{ are replaced with random oracle simulator } Sim\, O^*. \]

If \( SC_{sk,PK} \) has ever queried \( Sim\, K(R) = \perp \),
Halt and return \( R \).
If \( SC_{sk,PK} \) has ever queried \( Sim\, H(h) = \perp \),
Halt and return \( h \) (the rightmost \(|R| \) bits of \( h \)).

\[ EXP_{SGC}^{ind-cca2}(ASC) = \perp. \]

II random oracle queries \( O^* \) are also replaced with random oracle simulator \( Sim\, O^* \).

If \( EXP_{SGC}^{ind-cca2}(ASC) \) has ever queried \( Sim\, K(R) = \perp \),
Halt and return \( R \).
If \( EXP_{SGC}^{ind-cca2}(ASC) \) has ever queried \( Sim\, H(h) = \perp \),
Halt and return \( h \) (the rightmost \(|R| \) bits of \( h \)).
\[ b \leftarrow \text{EXP}_{SGC}^{ind-cca2}(ASC), \]
Return \( R \).

Let \( ASC \) be an attacker against IND-CCA2 security of signcryption in time \( t \), using \( q_{sc} \) queries to its signcryption oracle, \( q_{usc} \) queries to its unsigncryption oracle and \( (q_k,q_h) \) queries to its random oracles. \( AECG \) is an attacker against ECGP security of elliptic curve in time \( t' \), using \( q_{igp}\) queries to its DECP Oracle \( O_{DECP}^{\text{DECP}} \). \( AI \) is an attacker against PI security of the symmetric key encryption in time \( t'' \). From the AECG algorithm formulated above, the following bound holds.

More details about the probability proof of the theorem can be found in [5, 12, 13, 14].

\[ Adv_{SGC}^{ind-cca2}(t,q_{sc},q_{usc},q_k,q_h) \leq 2 Adv_{ECP}^{\text{DECP}}(t',q_{igp}) + 2 Adv_{T}^{\text{DECP}}(t''') + \]
\[ q_{sc}\left(\frac{q_k + q_h + q_{sc} + q_{usc} + 2}{2^{t}(t')^{-1}}\right) + q_{h} + 2q_{usc} \]
\[ \Rightarrow Adv_{SGC}^{ind-cca2}(t,q_{sc},q_{usc},q_k,q_h) \leq q_{sc}\left(\frac{q_k + q_h + q_{sc} + q_{usc} + 2}{2^{t}(t')^{-1}}\right) - \]
\[ q_{h} + 2q_{usc} - Adv_{T}^{\text{DECP}}(t''') \]
\[ \leq Adv_{ECP}^{\text{DECP}}(t',q_{igp}). \]

As \( ASC \) and \( AECG \) are reasonably resource bounded,
\[ q_{sc}\left(\frac{q_k + q_h + q_{sc} + q_{usc} + 2}{2^{t}(t')^{-1}}\right) - q_{h} + 2q_{usc} \text{ is negligible.} \]

On account of all the above analyses, if the IND-CCA2 security of signcryption will be broken by \( ASC \) with non-negligible advantage, so will the ECPG security of elliptic curve by \( AECG \) with non-negligible advantage. Therefore, \( S-ECSC \) achieves confidentiality in the IND-CCA2 model, as desired.

### 3.3 Unforgeability of \( S-ECSC \)

**Definition 3.3.1** Unforgeability of Signcryption. Let \( \Sigma = (GC, GK, SC, USC) \) be a signcryption scheme, and let \( A \) be an algorithm that has access to a signcryption oracle and returns a pair of strings. We consider the following experiment:

**Experiment** \( EXP_{SGC}^{\text{uf cma}}(A) \)

\[ (sk_k,PK_k) \leftarrow GK(A,k), \]
\[ (sk_k,PK_k) \leftarrow GK(B,k), \]
\[ (m,C') \leftarrow S-ECSC_{sk,PK,sc}(PK_A,PK_B), \]
If the following are true return 1 else return 0:
1. \( m \leftarrow USC((sk_k,PK_k),C') \),
2. \( m \in \text{Message}(sk_A,PK_B), \)
3. \( m \) is not a query of \( A \) to its signcryption oracle.

The UF-CMA advantage of \( A \) is defined as

\[ Adv_{SGC}^{\text{uf cma}}(A) = \Pr \{ EXP_{SGC}^{\text{uf cma}}(A) = 1 \}. \]

To be specific, the UF-CMA advantage can be concluded as a function \( \varepsilon(k) \) defined by

\[ \varepsilon(k) = \Pr \{ (sk_k,PK_k) \leftarrow GK(A,k), \]
\[ (sk_k,PK_k) \leftarrow GK(B,k), \]
\[ (m,C') \leftarrow S-ECSC_{sk,PK,sc}(PK_A,PK_B), \]
\[ m \leftarrow USC((sk_k,PK_k),C') \}. \]

A signcryption is un-forgettable under chosen message attack if the UF-CMA advantage of any attacker \( A \) with reasonably restricted resources (time-complexity, frequency and length of queries) is negligible.

**Hypothesis 3.3.1** (ECDLP is hard). Let \( T \) be an elliptic curve, and let \( A \) be an algorithm that has access to a elliptic curve oracle and returns a string. We consider the following experiment:

**Experiment** \( EXP_{T}^{\text{ECDLP}}(A) \)

\[ x \leftarrow Z_t \]
\[ x' \leftarrow A^T(P,Q). \]
If $x' = x$ return 1 and return 0 otherwise.

The ECDLP advantage of $A$ is defined as

$$Adv_T^{ECDLP}(A) = Pr(\exp_T^{ECDLP}(A) = 1).$$ (11)

Given elliptic curve $T$ and secure parameter $l^k$, the probability of solving ECDLP in time $t$ is $\delta(1^k, T)$ which is negligible, that is

$$\delta(1^k, T) = Pr[1^k, x \leftarrow Z_1^*, Q \leftarrow xP : \exp_T^{ECDLP}(A) = 1].$$ (12)

**Definition 3.3.2** Gap Elliptic Curve Discrete Logarithm (GECDL). Let $T = (q, a, b, P, l, h)$ be an elliptic curve and $AGL$ an attacker on GECDL. $O_{DECP}$ is DECP Oracle, let’s consider the following experiment:

**Experiment** $\exp_T^{GECDL}(AGL)$

$$d \leftarrow Z_1^*, D \leftarrow dP,$$

$$d' \leftarrow AGL^{O_{DECP}(\omega)}(D),$$

If $d' = d$ return 1 else return 0.

The GECDL advantage of $AGL$ is defined as

$$Adv_T^{GECDL}(AGL) = Pr(\exp_T^{GECDL}(AGL) = 1).$$ (13)

**Hypothesis 3.3.2** (GECDL is hard). Given elliptic curve $T$ and secure parameter $l^k$, the probability of solving GECDL in time $t$ is negligible, that is

$$\delta(1^k, T) = Pr[1^k, d \leftarrow Z_1^*, d' \leftarrow AGL^{O_{DECP}(\omega)}(D), \exp_T^{GECDL}(AGL) = 1].$$ (14)

**Proof of unforgeability:** Let $ASC$ be an attacker against UF-CMA security of signcryption executing in time $t$, using $q_{usc}$ queries to its signcryption oracle, $q_{usc}$ queries to its unsigncryption oracle and $(q_k, q_h)$ queries to its random oracles. $AGL$ is an attacker against GECDL security of elliptic curve executing in time $t'$, using $q_{O_{DECP}}$ queries to its DECP Oracle $O_{DECP}$. From the algorithm formulated above, the following bound holds. Similarly, more details about the probability proof of the theorem can be found in [12, 13, 14].

$$Adv_{SGC}^{O_{DECP}(\omega)}(t, q_{usc}, q_{usc}, q_k, q_h) \leq 2\sqrt{Adv_T^{GECDL}(t', q_{O_{DECP}}) + \frac{2\varepsilon q_{usc}(q_k + q_h + q_{usc}) + q_k + 1}{2lT'(t'-1)}}.$$ (15)

As $ASC$ is reasonably resource bounded,

$$\Rightarrow \frac{2\varepsilon q_{usc}(q_k + q_h + q_{usc}) + q_k + 1}{2lT'(t'-1)}$$ is negligible

$$\Rightarrow Adv_{SGC}^{O_{DECP}(\omega)}(t, q_{usc}, q_k, q_h) \leq 2\sqrt{Adv_T^{GECDL}(t', q_{O_{DECP}})}.$$ (16)

If the UF-CMA security of signcryption will be broken by $ASC$ with non-negligible advantage, so will the GECDL security of elliptic curve by $AGL$ with non-negligible advantage. Therefore, $S$-ECSC achieves unforgeability in the UF-CMA model, as desired.

### 3.4 Nonrepudiation of $S$-ECSC

**Definition 3.4.1** Non-repudiation of signcryption. It is computationally feasible for a third party to settle a dispute between signcryption sender and receiver in an event where sender denies the fact that he is the originator of signcryption.

**Definition 3.4.2** Relation Map. A relation is a map defined as

$$\mathcal{R}^{H}_{E, \pi}: \{0,1\} \times \{0,1\} \rightarrow \{0,1\}.$$ (17)

For every string $x \in \{0,1\}^*$, random oracle $H \in \mathcal{R}^{H}_{E, \pi}$, it satisfies

$$\mathcal{R}^{H}_{E, \pi}(x, x) = \mathcal{R}^{H}_{E, \pi}(x, 0^*) = 0.$$ (18)

Besides, $\mathcal{R}^{H}_{E, \pi}$ must be computable by a deterministic polynomial time algorithm $A^{H}_{\pi}(x, y, E, \pi)$. A malleability adversary $s$ is a pair of probabilistic polynomial time algorithms $(P, Q)$ with access to random oracle $H \in \mathcal{R}^{H}_{E, \pi}$.

The security notion of non-malleability for encryption scheme was introduced by Dolev, Dwork and Naor[15]. In this section, we generalize non-malleability into a more comprehensive security notion applicable to signcryption as well.

**Definition 3.4.3** Non-malleability of Signcryption. A signcryption scheme $\Sigma = (GC, GK, SC, USC)$ is non-malleable if any adversary can not by witnessing signcryption generating of a message $m$ or querying a signcryption oracle, produce the signcryption text of a related message $m'$.

To be specific, a signcryption scheme is non-malleable if for every relation $\mathcal{R}$ and every malleability adversary $s = (P, Q)$, there is a deterministic time algorithm $Q'$ so that $|\tau_k - \tau'_{s_k}|$ defined as follows is negligible.

$$\tau_k = Pr[H \leftrightarrow 2^\omega : (SC, USC) \leftrightarrow K(1^k), \pi \leftrightarrow P^{H}(SC); x \leftrightarrow \pi^{H}(1^k), \beta \leftrightarrow SC^{H}(x), \beta \leftrightarrow Q^{H}(SC, \pi, \beta), R^{H}_{E, \pi}(x, USC^{H}(\beta') = 1),$$ (19)

$$\tau'_{s_k} = Pr[H \leftrightarrow 2^\omega : (SC, USC) \leftrightarrow K(1^k), \pi \leftrightarrow P^{H}(SC); x \leftrightarrow \pi^{H}(1^k), \beta \leftrightarrow Q^{H}(SC, \pi, \beta), R^{H}_{E, \pi}(x, USC^{H}(\beta') = 1)].$$ (20)

**Theorem 3.4.1** The short signcryption scheme $S$-ECSC achieves non-repudiation security.

**Proof of non-repudiation:** In signcryption schemes, unforgeability implies non-repudiation if there is no duplication of the signcryption text. If the signcryption scheme is forgeable or malleable, the signcryption generator will have opportunity to repudiate.
In S-ECSC, the map \( K : E(F_q) \rightarrow \{0,1\}^l(c) \) and \( H : \{0,1\}^r \rightarrow Z^*_n \) are both unique, distinct (\( m_1, r \)) and (\( m_2, r \)) will generate different signcryption text \( C = (c, h, s) \). Furthermore, the scheme can be reinforced by state padding. The state padding not only ensures different signcryption scheme for distinct (\( m_1, r \)) but for the same original message (\( m_1, r \)) with different state information. Thus, the above signcryption scheme satisfies: as to \( |\tau(k) - \tau(k)| \) for every \( c \) there is a \( k \), such that \( |\tau(k) - \tau(k)| \leq k^{-\epsilon} \) for every \( k \geq k_\epsilon \). Thus the signcryption text \( C \) produced by \( SC \) (\( \text{sk}_A, \text{PK}_A, m \)) is not duplicable, and with the unforgeability proof of S-ECSC in UF-CMA model in section 3.3, we can come to the conclusion that S-ECSC achieves non-repudiation security, as desired.

## 4 Efficiency of S-ECSC

In this section, the short signcryption scheme S-ECSC will be compared with other typical schemes including discrete logarithm based signcryption SCS [5], B&D [16], KCDSA[17], SC-DSA[18] and RSA based signcryption TBO[19] and elliptic curve based scheme ECSC[20] and ECGSC[21].

In these schemes, such computing as modular exponential, modular inverse and elliptic curve addition, elliptic curve scalar multiplication should be taken into consideration for computing complexity, while computing cost of modular addition, modular multiplication, hash, symmetric encryption/decryption are negligible. To ensure the security of the basic cryptographic primitives, the minimum security parameters of these cryptosystems recommended for the current practice are as follows: for DLP, \( |p| = 1024\)bits, \( |q| = 160\)bits. For RSA, \( |N| = 1024\)bits; for ECC, \( |q| = 131\)bits (79, 109 may also be chosen), \( |l| = 160\)bits. The length of secure hash function is 128bits.

### Table 1: Comparison of computing and communication cost

<table>
<thead>
<tr>
<th>Schemes</th>
<th>GC\textsc{+}GK</th>
<th>SC</th>
<th>USC</th>
<th>EC</th>
<th>PV</th>
<th>Length of C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCS</td>
<td>2E</td>
<td>1E</td>
<td>1E+1</td>
<td>E</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>B&amp;D</td>
<td>2E</td>
<td>2E</td>
<td>2E+1</td>
<td>E</td>
<td>0</td>
<td>2E</td>
</tr>
<tr>
<td>KCDSA</td>
<td>2E</td>
<td>2E</td>
<td>3E</td>
<td>E</td>
<td>2E</td>
<td></td>
</tr>
<tr>
<td>SC-DSA</td>
<td>2E</td>
<td>2E</td>
<td>3E+1</td>
<td>E</td>
<td>2E</td>
<td></td>
</tr>
<tr>
<td>TBOS</td>
<td>2E+2E</td>
<td>2E</td>
<td>3E+1</td>
<td>E</td>
<td>2E</td>
<td>[</td>
</tr>
<tr>
<td>ECSCS</td>
<td>2E+2E</td>
<td>2E</td>
<td>2E</td>
<td>0</td>
<td>2E</td>
<td>[\frac{</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schemes</th>
<th>SC</th>
<th>USC</th>
<th>EC</th>
<th>PV</th>
<th>Length of C</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-ECSC</td>
<td>2kP+1l</td>
<td>2kP+1l</td>
<td>2l</td>
<td></td>
<td>[</td>
</tr>
</tbody>
</table>

Notes of notations: 1. GC denotes the common parameters generation algorithm, GK denotes the keys generation algorithm; SC denotes the signcryption algorithm; USC denotes the unsigncryption algorithm; EC denotes the extra computation to accomplish public verifiability; PV denotes the public verification by a third party. Length of C denotes the length of signcryption text. 2. E denotes modular exponential; I denotes modular inverse; \( K \) denotes scalar multiplication on elliptic curve; \( \theta \) denotes there is no relevant computation. 3. |D(\cdot)| denotes the block length of block cipher. |l| denotes the outputs length of secure hash function. |K\.absolute_value| denotes the length of key hash function in SCS, the same as |h|, \( |LH(\cdot)| \) denotes the length of hash function with long message digest, much larger than |h|.

**Remark 1.** (Comparison with DLP based signcryption schemes). SCS is the fastest scheme in all of the four DLP based schemes (SCS, B&D, KCDSA and SC-DSA). Based on the result of Koblitz and Menezes [22], the computing cost in key generation in our scheme is 1/8 of that in SCS; signcryption operation in ours is about 1/8 of that in SCS, and unsigncryption is about 1/8 of that in SCS. To sum up, S-ECSC reduces about 87%computing cost compared with SCS.

**Remark 2.** (Comparison with RSA based signcryption scheme). As per the result of [22], the computing cost in key generation in our scheme is about 1/8 of that in TBOS; signcryption operation in ours is about 1/16 of that in TBOS, and unsigncryption is about 1/8 of that in TBOS, achieving a total 89%computing cost reduction over TBOS.

**Remark 3.** (Comparison with other ECC based schemes). The computing cost in key generation are the same; signcryption cost in ours is slightly lower than that in ECSCS while unsigncryption of ECSCS is slightly lower than ours, total resulting an equal computing cost, yet ECSCS proves to be unsuitable for public verifying. Although ECGSC is publicly verifiable, its computing cost in signcryption and unsigncryption is much larger than S-ECSC, resulting in a much higher total computing cost.

**Remark 4.** (Comparison of communication cost). As per the comparison of signcryption text length, except for RSA based TBOS signcryption, S-ECSC has the lowest communication cost in Elgamal type signcryption schemes.

Therefore, we may come to the conclusion that our short signcryption scheme S-ECSC has the highest efficiency and the lowest communication cost in all of the publicly verifiable schemes.
5 Application of $S$-ECSC in Secure Communication of IOT

In order to achieve confidentiality and integrity for secret key in the Internet of things, a secure channel should be established for the distribution and transmission in key management schemes. Meanwhile, the special network environment in IOT, such as wireless connection, microterminals and restricted resources, makes it necessary to design schemes of high efficiency which can fulfill the same function with much smaller computing and communication cost than traditional schemes. With its superiority in computing and communication, $S$-ECSC greatly improves the efficiency in key management in IOT in terms of key distributing time and bandwidth resources and better satisfies the requirement of secure protocols in key management. In this section, we will take the key management schemes in [23,24,25] as an example and propose a key applying and distributing scheme in key management of IOT based on $S$-ECSC.

With this $S$-ECSC based scheme, we analyse the method and importance to apply $S$-ECSC in secure wireless communication for the Internet of things. The key applying and distributing protocol in key management of IOT based on $S$-ECSC is as follows.

1) Initializing

PKG selects the system parameter, including the parameters in $S$-ECSC.

$$\forall k \in Z^*_i, PK_A = sk_A P \neq 0, (sk_A, PK_A)$$

is the private/public key pair for one of the distributed terminals $A$. $sk_B \in Z^*_i, PK_B = sk_B P \neq 0$, $(sk_B, PK_B)$ is the private/public key pair of PKG.

2) Key applying

Step1: Terminal $A$ encodes the applying request data \{ID$_A$, Message\} into plaintext $m \in$ Message ($sk_A, PK_B$), and applies the signcryption algorithm on $m$.

$$C \leftarrow SC (sk_A, PK_B, m).$$

Then signcryption text $C$ will be transmitted to PKG.

Step2: PKG applies the unsigncryption algorithm on signcryption text $C$.

$$m \leftarrow USC (sk_B, PK_A, C).$$

Thus, PKG recovers plaintext $m$, and simultaneously fulfills authentication on identity of terminal $A$ and examines the integrity of message $m$.

3) Key generating and distributing

Step1: PKG generates secret key $k \in$ Message ($sk_B$, $PK_A$) with the key generating algorithm $KG(1^k)$, and applies the signcryption algorithm on $k$.

$$C' \leftarrow SC (sk_B, PK_A, k).$$

Then the signcryption text $C'$ will be transmitted to terminal $A$.

Step2: Terminal $A$ applies the unsigncryption algorithm on $C'$.

$$k \leftarrow USC (sk_A, PK_B, C').$$

Thus, $A$ recovers secret key $k$, and fulfills authentication on identity of PKG and examines the integrity of secret key $k$.

With the application of $S$-ECSC in key applying and distributing, the above scheme achieves secure and efficient transmission of terminal secret key via the public channel of IOT. The scheme fulfills the integrated functions of encryption and digital signature in a single step and simultaneously achieves confidentiality, integrity and non-repudiation for the secret terminal key and other signcrypted message; whereas, the computing and communication cost is significantly smaller than traditional schemes.

6 Conclusions

The study of signcryption algorithms suitable for IOT network environment and its application in IOT security schemes is an important direction in cryptography: it is more of a requirement from the rapid development of the Internet of things than just a requirement from the theoretical or applied cryptography research. In the paper, we propose a publicly verifiable short signcryption scheme $S$-ECSC suitable for secure communication in the Internet of things; and prove the provable security of $S$-ECSC under the Random Oracle model, including confidentiality, unforgeability and non-repudiation security. At last, we take key generating and distributing protocol for different terminals of distributed key management in IOT as an example, and analyze the method and importance in the application of $S$-ECSC into secure protocols in IOT.

Compared with other typical discrete logarithm, RSA and elliptic curve based signcryption schemes; $S$-ECSC achieves about 87% reduction in computing cost than DLP signcryption schemes and about 89% reduction compared with RSA schemes. And it has the lowest communication cost in the Elgamal type schemes. Therefore, security schemes based on $S$-ECSC are most suitable for such circumstances as with restricted computation ability and integrated space, circumstances with limited bandwidth yet requiring for high-speed operation. Besides, the computational problems ECGP and GECDL in the paper can also be basis of security proof for other elliptic curve based schemes.

Acknowledgement

The authors should thank the anonymous reviewers for their constructive advice and comments to the paper, with which we can greatly improve our work.

References


