An Efficient and Provably-Secure Identity-based Signcryption Scheme for Multiple PKGs *

Zhengping Jin¹, †, Huijuan Zuo², ¹, Hongzhen Du¹, ³ and Qiaoyan Wen¹

¹ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
† E-mail: zhpjin@yahoo.cn
² Institute of Mathematics and Information, Hebei Normal University, Shijiazhuang 050016, China
³ Mathematics Department, Baoji University of Arts and Sciences, Baoji 721007, China

Abstract

In this paper, based on the scheme proposed by Barreto et al in ASIACRYPT 2005, an identity-based signcryption scheme in multiple Private Key Generator (PKG) environment is proposed, which mitigates the problems referred to users’ private keys escrow and distribution in single PKG system. For security of the scheme, it is proved to satisfy the properties of message confidentiality and existential signature-unforgeability, assuming the intractability of the $q$-Strong Diffie-Hellman problem and the $q$-Bilinear Diffie-Hellman Inversion problem. For efficiency, compared with the state-of-the-art signcryption schemes of the same kind, our proposal needs less pairing computations and is shown to be the most efficient identity-based signcryption scheme for multiple PKGs up to date.

1. Introduction

The concept Identity-based Cryptography was first introduced by Shamir [12] in 1984. Its basic idea is that the users can choose arbitrary strings, such as their email addresses or other online identifies, as their public keys, and the corresponding private keys are created by binding the identity with a master private key of a trusted authority (called private key generation, or PKG for short). This eliminates much of the overhead associated with key management in conventional public key infrastructure. However, as pointed out in [13], it is unrealistic in practice to set up a single global private key generator mainly because of the inherent key escrow problem, i.e., the PKG knows all its users’ private keys. Another flaw of this single PKG cryptographic system is that when distributing the users’ private keys, so many secure channels between the PKG and its users are required. In order to mitigate these problems, Wang and Cao [13] proposed the multiple PKGs environment for identity-based cryptographic systems. Exactly speaking, in this multiple PKGs environment, sharing the common global system parameters, each administrator domain maintains its own domain PKG which generates its own master private key, computes and publishes the corresponding master public key, then generates and distributes private keys for the registered users within its domain. A reasonable requirement for this environment is that users could securely communicate with each other no matter whether they belong to the same domain or not. Actually, the original idea of the multiple PKGs environment was first suggested in an authenticated key agreement protocol [5]. Later, a practical identity-based encryption scheme in multiple PKGs environment was presented in [13], and two identity-based signcryption schemes for multiple PKGs were respectively designed in [7, 8].

Signcryption, first proposed by Zheng [15], is a cryptographic primitive that performs digital signa-
ture and public key encryption in a single operation, at lower computational costs and communication over-heads than that of doing both operations sequentially. The first identity-based signcryption scheme was proposed by Malone-Lee [10], but it was pointed out by Libert and Quisquater [9] that this scheme was not semantically secure. Then, an identity-based signcryption scheme was designed by Boyen [4], which provides several strong security properties of confidentiality, non-reputation, etc. Later, Boyen’s scheme was improved in efficiency in [1, 6], where Barreto et al’s proposal [1], to the best of our knowledge, happens to be the most efficient identity-based signcryption scheme up to date.

In this paper, based on the scheme proposed by Barreto et al [1], we present an identity-based signcryption scheme for multiple PKGs. Our proposal is proved to satisfy the security of message confidentiality and existential signature-unforgeability, assuming the intractability of the $q$-Strong Diffie-Hellman problem and the $q$-Bilinear Diffie-Hellman Inversion problem which were formalized by Boneh and Boyen [2, 3, 1]. Thanks to the original advantages of Barreto et al’s scheme, our identity-based signcryption scheme for multiple PKGs achieves high efficiency in implementation, which is better than the homogeneous schemes proposed in [7] and [8], and happens to be the most efficient identity-based signcryption scheme for multiple PKGs up to date.

This paper is organized as follows. Some preliminaries are given in section 2. The formal model and some security notions for our identity-based signcryption scheme for multiple PKGs are constructed in section 3. The details of the proposed scheme are elaborated in section 4. We prove compliance of our implementation with the security model and analyze the efficiency of our scheme in section 5. Finally, in section 6, we draw some conclusions.

2. Preliminaries

In this section, we describe the basic security theoretic concepts of bilinear map groups and the hard problems underlying our proposed algorithms.

Let $k$ be a security parameter and $p$ be a $k$-bit prime number. Let $G_1, G_2$ be cyclic additive groups generated by $G_1, G_2$ respectively, both of whose orders are $p$. Let $G_T$ be a cyclic multiplicative group of the same order.

We say that $(G_1, G_2, G_T)$ are bilinear map groups if there exists a bilinear map $e : G_1 \times G_2 \rightarrow G_T$ satisfying:

1. Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $(P, Q) \in G_1 \times G_2, a, b \in \mathbb{Z}_p$.

2. Non-degeneracy: There exists $P \in G_1, Q \in G_2$ such that $e(P, Q) \neq 1_{G_T}$.

3. Computability: For all $(P, Q) \in G_1 \times G_2, e(P, Q)$ is efficiently computable.

4. There exists an efficient, publicly computable (but not necessarily invertible) isomorphism $\psi : G_2 \rightarrow G_1$ such that $\psi(1_{G_2}) = G_1$.

The computational assumptions for the security of our scheme were previously formalized by Boneh and Boyen [2, 3, 1] and are recalled in the following definition.

Definition 1. Let $(G_1, G_2, G_T)$ be bilinear map groups with generators $P \in G_1$ and $Q \in G_2$. Two hard problems are described as follows:

$q$-Strong Diffie-Hellman problem ($q$-SDHP).

Given a $(q + 2)$-tuple $(P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^q Q)$, find a pair $(c, \frac{1}{c^\alpha} P)$ with $c \in \mathbb{Z}_p^*$. 

$q$-Bilinear Diffie-Hellman Inversion problem ($q$-BDHIP).

Given a $(q + 2)$-tuple $(P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^q Q)$, compute $e(P, Q)^{1/\alpha} \in G_T$.

3. Formal Model of Identity-based Signcryption

3.1. Generic Scheme

An identity-based signcryption scheme for multiple PKGs consists of five algorithms: Global-Setup, Domain-Setup, Key-Extraction, Signcryption and Unsigncryption, whose functions are specified as follows.

Global-Setup. Given a security parameter $k$, the globally trusted third-party outputs the system’s global public parameters $\text{params}$.

Domain-Setup. Given the common parameters $\text{params}$, each domain PKG$i$ generates its domain master private key $s_i$ and the corresponding domain public key $Q_{\text{pub}}$.

Key-Extraction. Given an identity ID$_U$, the domain PKG$_{i_U}$ computes the corresponding private key $S_{i_U}$, then secretly transmits it to its owner.

Signcryption. Given a message $M$ and a receiver’s identity ID$_B$, the sender obtains the ciphertext $c$ by computing $\text{Signcryption}(M, S_{i_U}, \text{ID}_B)$.

Unsigncryption. Given a ciphertext $c$ and a sender’s identity ID$_A$, the intended receiver computes $\text{Unsigncryption}(c, \text{ID}_A, \text{ID}_B)$, then returns a message $M$ and its valid signature, or outputs a distinguished symbol if $c$ does not decrypt into a message bearing the signer ID$_A$’s signature.
For consistency, we of course require that if \( \sigma = \text{Signcryption}(M, S_{ID_A}, ID_B) \), then \( M \) should be a part of \( \text{Unsigncryption}(\sigma, ID_A, S_{ID_B}) \).

3.2. Security Notions

The formal security notions for identity-based signcryption schemes have been defined in [4, 6, 1], where Barreto et al [1] mainly considered the property of message confidentiality and the existentially signature-unforgeable security against adaptive chosen message and ciphertext attacks. We modify their definitions slightly to adapt for our identity-based signcryption scheme for multiple PKGs.

**Definition 2.** An identity-based signcryption scheme for multiple PKGs, briefly called \( \text{IBSCMP} \), is said to satisfy the **Message Confidentiality** property (or the indistinguishability against adaptive chosen ciphertext attacks property: IND-IBSCMP-CCA) if no probabilistic polynomial time (PPT) adversary has a non-negligible advantage in the following game.

1. The challenger \( C \) runs the **Global-Setup** and **Domain-Setup** algorithms on input of a security parameter \( k \) and sends the system’s global-wide public parameter \( params \) and each domain-wide master public key \( K_{pub} \), to the adversary \( A \).

2. In a find stage, \( A \) performs a polynomially bounded number of the following queries.

   - Key-extraction queries: \( A \) chooses an identity \( ID_U \) under \( PKG_{KU} \), then \( C \) computes \( S_{ID_U} = \text{Key-Extraction}(ID_U) \) and sends \( S_{ID_U} \) to \( A \).

   - Signcryption queries: \( A \) chooses a pair of identities \( (ID_S, ID_R) \) and a plaintext \( M \), then \( C \) returns to \( A \) a ciphertext \( \sigma = \text{Signcryption}(M, S_{ID_A}, ID_R) \), where \( S_{ID_A} = \text{Key-Extraction}(ID_S) \).

   - Unsigncryption queries: \( A \) chooses a pair of identities \( (ID_S, ID_R) \) and a ciphertext \( \sigma \), then \( C \) returns the result of \( \text{Unsigncryption}(\sigma, ID_S, S_{ID_B}) \) to \( A \), where \( S_{ID_B} = \text{Key-Extraction}(ID_R) \).

3. \( A \) produces two equal length plaintexts \( M_0, M_1 \) and a pair of identities \( (ID_S^0, ID_R^0) \), where the private key of \( ID_B \) can not have been extracted. \( C \) computes \( \sigma^* = \text{Signcryption}(M_b, S_{ID_A}, ID_R) \) for a random bit \( b \in \{0, 1\} \) and sends \( \sigma^* \) to \( A \).

4. In the guess stage, \( A \) makes new queries as in the find stage. This time, neither the key-extraction query on \( ID_R \) nor the unsigncryption query on \( \sigma^* \) could be asked.

5. \( A \) finally outputs a bit \( b' \) and wins the game if \( b' = b \).

The advantage of \( A \) is defined as \( 2Pr[b' = b] - 1 \).

**Definition 3.** \( \text{IBSCMP} \) is said to be **Existentially Signature-unforgeable** against adaptive chosen message and ciphertext attacks (ESUF-IBSCMP-CMA) if no PPT adversary can succeed in the following game with a non-negligible advantage.

1. The challenger \( C \) and the adversary \( A \) respectively does the same as step 1 and 2 of the game in definition 2.

2. \( A \) produces a triple \( (\sigma^*, ID_S^*, ID_R^*) \), where the private key of \( ID_S^* \) was not previously asked. \( A \) wins the game if the result of \( \text{Unsigncryption}(\sigma^*, ID_S^*, S_{ID_B}) \) is a valid message-signature pair \( (M^*, s^*) \) such that no signcryption query involved \( M^* \), \( ID_S^* \) and some receiver \( ID_R^* \) (possibly different from \( ID_R^* \)) and resulted in a ciphertext \( \sigma' \) whose decryption under the private key \( S_{ID_R^*} \) is the alleged forgery \( (M^*, s^*, ID_S^*) \).

The advantage of the adversary \( A \) is its probability of victory.

We note that, in both of above definitions, inside attacks are considered.

4. An Identity-based Signcryption Scheme for Multiple PKGs

In this section, modifying Barreto et al’s signcryption scheme proposed in [1], we design an identity-based signcryption scheme in multi-PKG environment as follows.

**Global-Setup.** Given \( k \in \mathbb{Z}^+ \), the globally trusted third-party does the following:

1. Chooses bilinear map groups \( (G_1, G_2, G_T) \) of prime order \( p > 2^k \) and generators \( G_2 \in G_2, G_1 = \psi(G_2) \in G_1 \), and computes \( g = e(G_1, G_2) \in G_T \).

2. Picks three hash functions \( H_1 : \{0, 1\}^* \times G_2 \rightarrow \mathbb{Z}_p^* \), \( H_2 : \{0, 1\}^* \times G_T \rightarrow \mathbb{Z}_p^* \) and \( H_3 : G_T \rightarrow \{0, 1\}^n \).

The global public parameters are \( \text{params} := \{G_1, G_2, G_T, G_1, G_2, g, \psi, H_1, H_2, H_3\} \).

**Domain-Setup.** Given global public parameters, each domain PKG, does the following:

1. Randomly chooses \( s_i \in \mathbb{Z}_p^* \) as the domain master private key and keeps it secret.

2. Calculates \( K_{pub} = s_i G_2 \) as the domain master public key and publishes \( K_{pub} \).

**Key-Extraction.** Given an identity \( ID_U \), this algorithm outputs the private key \( S_{ID_U} \) under the domain PKG \( K_{KU} \), where \( S_{ID_U} = H_{1,ID_U}(K_{pub} || s_{ID_U}) G_2 \in G_2 \).

**Signcryption.** Given a message \( M \in \{0, 1\}^* \) and a receiver’s identity \( ID_R \), the sender does as follows:

1. Randomly picks \( x \in \mathbb{Z}_p^* \), computes \( r = g^x \) and \( c = M \oplus H_3(r) \in \{0, 1\}^n \).

2. Sets \( h = H_2(M, r) \in \mathbb{Z}_p^* \).

3. Computes \( S = (x + h) \psi(S_{ID_A}), T = x(H_1(ID_B, K_{pub_R}) G_1 + \psi(K_{pub_R})) \).
The ciphertext is $\sigma = (c, S, T) \in \{0, 1\}^n \times G_1 \times G_1$.

**Unsigncryption.** Given a ciphertext $\sigma = (c, S, T)$ and some sender’s identity ID$_A$, the intended receiver does the following:
1. Computes $r = e(T, S_{ID_B})$, $M = c \oplus H_2(r)$, and $h = H_2(M, r)$.
2. Returns the message $M$ and the signature $(h, S) \in Z_q^* \times G_1$ if $r = e(S, H_1(1|ID_A, K_{pub_A})G_2 + K_{pub_A})^{\gamma-h}$ and the $\pm$ symbol otherwise.

It is easy to see that the proposed scheme is consistent.

## 5. Security Results and Efficiency Comparisons

The following theorems claim the security of our proposal under the same irreflexivity assumption as Boyen’s scheme [4]: the signcryption algorithm is assumed to disallow messages from being addressed to the same identity as authored them.

**Theorem 1.** Let $A$ be an adversary against the IND-IBSCMP-CCA security of our scheme. If $A$ has an advantage after running $t'$ for running $t$, making at most $q_{b_1}, q_s, q_u$ queries to $H_1$ ($i = 1, 2, 3$), the signcryption oracle and the unsigncryption oracle respectively, then we have an algorithm $B$ that solves the $q$-BDHIP for $q = q_{b_1}$ with probability

$$
\epsilon' \geq \frac{\epsilon}{q_{b_1}(2q_{b_2} + q_{b_3})} \left(1 - \frac{q_s}{2q} + \frac{q_u}{2q} + \frac{\epsilon}{q_{b_1}}\right)
$$

and within a time $t' < t + O(q_s + q_u)t_p + O(q_{b_2})t_\alpha + O(q_{b_3})t_{\gamma}$, where $k$ is the security parameter whereas $t_p, t_\alpha$ and $t_{\gamma}$ are respectively the costs of a pairing computation, a multiplication in $G_2$ and an exponentiation in $G_T$.

**Proof.** Algorithm $B$ takes a random instance $(P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^q Q)$ of the $q$-BDHIP as input, and attempts to extract $e(P, Q)^{1/\alpha}$ by running $A$ as a subroutine and acting as $A$’s challenger in the game of definition 2.

At first, $B$ does some preparations for interacting with $A$. It randomly selects $R_1, \ldots, R_q$ from $Z_p^*$ and $\xi \in \{1, 2, \ldots, q\}$. For $i \in \{1, 2, \ldots, q\}$, it computes $x_i = x_j - R$ where $x_j = R_q$. Then it sets up generators $G_2 \in G_2$, $G_1 = \psi(G_2) \in G_1$ and another element $X = \alpha G_2 \in G_2$ such that it knows $q - 1$ pairs $(R_i, I_i = \frac{1}{\alpha G_2})$, $i = \{1, 2, \ldots, q\} \backslash \{\xi\}$. To do so,

1. It expands $f(z) = \prod_{i=1, i \neq \xi}^q (z + R_i)$ to obtain $c_0, \ldots, c_{q-1} \in Z_p^*$ so that $f(z) = \sum_{j=0}^{q-1} c_j z^j$, and sets generators $G_2 = \sum_{j=0}^{q-1} c_j (\alpha Q) = f(\alpha)Q \in G_2$ and $G_1 = \psi(G_2) = f(\alpha)P \in G_1$. Another element $X \in G_2$ is fixed to $X = \sum_{j=0}^{q-1} c_j - 1 (\alpha Q) = \alpha Q$ although $B$ does not know $\alpha$.
2. For $i \in \{1, \ldots, q\} \backslash \{\xi\}$, it expands $f_i(z) = \frac{f(z)}{z + R_i} = \prod_{k=1, k \neq \xi, k \neq i}^q (z + R_k)$ to get $d_0, d_1, \ldots, d_{q-2} \in Z_p^*$ such that $f_i(z) = \sum_{j=0}^{q-2} d_j z^j$, and computes the pair $(R_i, I_i)$ by calculating $I_i = \sum_{j=0}^{q-2} d_j (\alpha Q) = f_i(\alpha)Q = \frac{f(\alpha)}{z + R_i} Q = \frac{1}{\alpha G_2}$. Subsequently, $B$ randomly selects $y_j \in Z_p^*$ for $j = 1, 2, \ldots, N$, where $N(q < q)$ is the total number of domain PKGs. The domain-wide master public key of PKG$_j$ is chosen as $K_{pub} = -X - (x_1 + y_j)G_2 = (-\alpha - x_1 - y_j)G_2$ so that its unknown private key is implicitly set to $s_j = -\alpha - x_1 - y_j \in Z_p^*$. For $i \in \{1, \ldots, q\}, j \in \{1, \ldots, N\}$, let $H_{i,j} = x_i + y_j$, then $B$ has pairs $(H_{i,j}, -I_i) = (H_{i,j}, \frac{1}{\alpha G_2})$, $i \in \{1, \ldots, q\}, j \in \{1, \ldots, N\}$.

$B$ then initializes a counter $l$ to 1 and starts $A$ on input of $(G_1, G_2, K_{pub_1}, \ldots, K_{pub_N})$. During the game, $A$ will consult $B$ for answers to the random oracles $H_1, H_2$ and $H_3$, and $B$ generates these answers randomly, but to maintain the consistency and to avoid collision, $B$ keeps three lists $L_1, L_2, L_3$ respectively to store the answers.

- $H_1$-queries on input $IND_U$ under some PKG, say PKG$_j$ : if it is the first time for $IND_U$ to query $H_1$, $B$ returns $H_j$, stores the information $(IND_U, K_{pub_j}, H_{i,j})$ in $L_1$, and increments $l$; otherwise, returns the corresponding value $H_{i,j}$ in $L_1$.
- $H_2$-queries on input $(M, r)$: $B$ returns the defined value if it exists and a random value $h_2 \in Z_p^*$ otherwise. To anticipate possible subsequent unsigncryption requests, $B$ additionally simulates $H_3$ on its own to obtain $h_3 = H_3(r) \in \{0, 1\}^n$, $c = M \oplus h_3, \gamma = r \cdot e(G_1, G_2)^{h_2}$, and stores the information $(M, r, h_2, c, \gamma)$ in $L_2$.
- $H_3$-queries on input $r \in G_T$: $B$ returns the previously assigned value if it exists and a random value $h_3 \in \{0, 1\}^n$ otherwise. In the latter case, the input $r$ and the response $h_3$ are stored in $L_3$.
- Key-extraction queries on input $IND_U$ under some PKG$_j$: if $i = \xi$, then $B$ fails. Otherwise, it knows that $H_1(IND_U, K_{pub_j}) = H_{i,j}$ and returns $-I_i = \frac{1}{\alpha G_2}$.
Signcryption queries for a plaintext \( M \) and identities \((ID_S, ID_R) = (ID_\mu, ID_\nu)\) under PKG_\mu and PKG_\nu respectively, \( \mu, \nu \in \{1, \ldots, q\} \), \( i_\mu, i_\nu \in \{1, \ldots, N\} \): We assume that \( \mu = \xi \) (and hence \( \nu \neq \xi \) by the irreflexivity assumption), because otherwise \( B \) knows the sender’s private key \( S_{ID_\nu} = -I_\mu \) and answers the query according to the Signcryption algorithm. Thus \( B \) randomly chooses \( \lambda, h \in \mathbb{Z}_2^* \) and computes \( S = \lambda \psi(S_{ID_\nu}) = -\lambda \psi(I_\mu), T = \lambda \psi(H_{\mu,i_\mu} G_2 + K_{pub_\mu}) = h \psi(H_{\nu,i_\nu} G_2 + K_{pub_\nu}) \) in order to obtain the desired equality \( r = e(T, S_{ID_\nu}) = e(S, H_{\xi,i_\nu} G_2 + K_{pub_\mu}) e(G_1, G_2)^{-h} = e(\psi(S_{ID_\nu}), H_{\xi,i_\nu} G_2 + K_{pub_\nu}) \). \( B \) checks if \( H_2(M, r) \) to \( h \) (\( B \) fails if \( H_2 \) is already defined but this only happens with probability \( \frac{q^2 + q_2}{q} \)). At last, \( B \) returns the ciphertext \( \sigma = (M \oplus H_3(r), S, T) \).

Unsigncryption queries on a ciphertext \( \sigma = \langle c, S, T \rangle \) for identities \((ID_S, ID_R) = (ID_\mu, ID_\nu)\) under PKG_\mu and PKG_\nu respectively: we assume that \( \nu = \xi \) for similar reasons as in signcryption queries, hence \( \mu \neq \xi \) by the irreflexivity assumption. Therefore, \( B \) has the sender’s private key \( S_{ID_\mu} \) and also knows that, for all valid ciphertexts, \( \log_{S_{ID_\mu}} (\psi^{-1}(S) - h S_{ID_\mu}) = \log_\psi(Q_{ID_\mu})(T) \), where \( h = H_2(M, r) \) is obtained in the Signcryption algorithm and \( Q_{ID_\nu} = H_{\nu,i_\nu} G_2 + K_{pub_\nu} \).

Hence, \( e(T, S_{ID_\nu}) = e(\psi(Q_{ID_\nu}), \psi^{-1}(S) - h S_{ID_\mu}) = e(\psi(Q_{ID_\nu}), \psi^{-1}(S)) e(\psi(S_{ID_\mu}), S_{ID_\mu})^{-h} (**) \) \( = e(S(Q_{ID_\mu}), \psi(Q_{ID_\nu}), S_{ID_\nu})^{-h}. \) Thus, \( B \) computes \( \gamma = e(S(Q_{ID_\mu}), S_{ID_\nu}) \), where \( Q_{ID_\mu} = H_{\mu,i_\mu} G_2 + K_{pub_\mu} \), and searches through list \( L_2 \) for entries of the form \((\mu, r, h_{\mu,i_\mu}^{(c, \gamma)}), i \) indexed by \( i \in \{1, \ldots, q_2\} \). If none is found, \( \sigma \) is rejected. Otherwise, each one of them is further examined for the corresponding indexes, \( B \) checks if \( e(T, S_{ID_\mu}) = e(S(Q_{ID_\mu}), \psi(Q_{ID_\nu}), S_{ID_\nu})^{-\gamma/\alpha} \), (***) meaning that (***) is satisfied. The pairings are computed only once and at most \( q_2 \) exponentiations are needed. If the unique \( i \in \{1, \ldots, q_2\} \) satisfying (***) is detected, the matching pair \((M_\mu, (h_{\mu,i_\mu}, S_\nu)) \) is returned. Otherwise, \( \sigma \) is rejected. Overall, an inappropriate rejection occurs with probability smaller than \( \frac{q_2}{q} \) across the game.

At the challenge phase, \( A \) outputs messages \((M_0, M_1)\) and identities \((ID_S, ID_R)\) under PKG_\nu and PKG_\nu respectively for which she never obtained \( ID_R \)'s private key. If \( ID_R = ID_\xi \), \( B \) aborts. Otherwise, it randomly picks \( \beta \in \mathbb{Z}_2^* \) and \( c \in \{0, 1\} \) and \( S \in \mathcal{G}_1 \) to return the challenge \( \sigma^* = \langle c, S, T \rangle \) where \( T = -\beta G_1 \) in \( G_1 \). If we define \( \rho = \beta/\alpha \) and since \( s_\mu = -\alpha - \xi - y_\mu \) and \( s_{\mu, i_\mu} = \rho H_{\xi,i_\mu} G_1 + \rho \psi(K_{pub_\mu}) \) we define \( T = -\beta G_1 = -\alpha \rho G_1 = (H_{\xi,i_\mu} + s_{\mu, i_\mu}) P G_1 \) and the guess stage, her view is simulated as before and her eventual output is ignored. Standard arguments can show that a successful \( A \) is very likely to query \( H_2 \) or \( H_3 \) on the input \( e(G_1, G_2)^c \) if the simulation is indistinguishable from a real attack environment.

To produce a result, \( B \) fetches a random entry \((M, r, h_2, c, \gamma)\) or \((r, h_3)\) from the lists \( L_2 \) or \( L_3 \). With probability \( \frac{1}{2q_2 + q_3} \), the chosen entry will contain the right element \( r = e(G_1, G_2)^c = e(P, Q)^f(\alpha)\beta/\alpha \), where \( f(z) = \sum_{i=0}^{q-2} c_i z^i \) is the polynomial for which \( G_2 = f(\alpha)Q \). At last, the \( q \)-BDHP solution can be extracted by \( e(P, Q)^{1/\alpha} = r^{(c_0^2 - c_0)} \eta - (c_0) \) where \( \pi = e\left(\sum_{i=0}^{q-2} c_i z^i, c_0 Q\right) = e(G_1, \sum_{j=0}^{q-2} c_j z^j Q) \).

In an analysis of \( B \)'s advantage, we note that it succeeds in above game if and only if all of the following independent events happen:

- \( E_1: ID_\xi \) is challenged, which implies that no key-extraction query is made on \( ID_\xi \).
- \( E_2: \) There is no collision on \( H_2 \) in a signcryption query. \( E_3: \) No valid ciphertext is rejected.
- \( E_4: B \) selects the correct element from \( L_2 \) or \( L_3 \) at the last phase.

Clearly, \( \Pr[E_1] = \frac{1}{q_2}, \Pr[E_2] = \frac{1}{q_1 + q_2}, \Pr[E_3] = \frac{1}{2q_2 + q_3} \). Thus,

\[
\epsilon' = \epsilon \cdot \Pr[E_1 \land E_2 \land E_3 \land E_4] \\
\geq \epsilon \frac{1}{q_1 (2q_2 + q_3)} \left( 1 - q_1 \frac{q_2 + q_3}{2k} \right) \left( 1 - \frac{q_1}{2k} \right)
\]

On the other hand, \( B \)'s workload is dominated by \( O(q_2^2) \) multiplications in the preparation phase, \( O(q_1 + q_2) \) pairing calculations and \( O(q_1 q_2) \) exponentiations in \( \mathcal{G}_1 \) in its emulation of the signcryption and unsigncryption oracles, thus it totally takes a time \( \epsilon' < t + O(q_1 + q_2) \rho + O(q_1^2) \mu + O(q_1 q_2) \tau \).

\( \square \)

**Theorem 2.** Let \( A \) be an adversary against the ESUF-IBSCMP-CMA security of our scheme. If \( A \) produces a forgery with probability \( \epsilon \geq 10q_1 (q_1 + 1)(q_1 + q_2) / (2k^2 - 1) \) after running a time \( t \) and making at most
$q_h$, queries to random oracles $H_i$ ($i = 1, 2, 3$), $q_s$ signcryption queries and $q_u$ unsigncryption queries, then we have an algorithm $B$ that is able to solve the $q$-SDHP for $q = q_{hg}$ in expected time $t'$

$$
\leq 120668 q_h q_{g2} t'^{1+O\left(\log q\right)} \exp\left(-\frac{1}{2 q^2}\right) + O\left(q^{1+\epsilon} t_m\right),
$$

where $t_p, t_e, t_m$ denote the same quantities as in theorem 1.

**Proof.** Algorithm $B$ takes a random instance $(P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^q Q)$ of the $q$-Strong Diffie-Hellman problem as input, and attempts to extract a pair $(\omega, \frac{1}{\omega} - P), \omega \in Z_p^*$ by running $A$ as a subroutine and acting as $A$’s challenger in the game of definition 3.

At first, we will show that a forger in the ESUF-IBSCMP-CMA game implies a forger in a chosen-message and given identity attack.

Before it, we need some preparations. $B$ randomly selects $R_i \in Z_p^*$ for $i \in \{1, \ldots, q - 1\}$. With the technique used in the proof of theorem 1, it sets up generators $G_2 \in G_2, G_1 = \psi(G_2) \in G_1$ and another element $X = \alpha G_2 \in G_2$ such that it knows $q - 1$ pairs $(R_i, \frac{1}{\alpha G_2})$ for $i \in \{1, \ldots, q - 1\}$. Subsequently, $B$ randomly selects $y_j \in Z_p^*$ for $j = 1, 2, \ldots, N$, where $N(< q)$ is the total number of domain PKGs.

The domain-wide master public key of PKG, is chosen as $K_{puby} = X - y_j G_2 = (\alpha - y_j) G_2$ so that its unknown private key is implicitly set to $s_j = \alpha - y_j \in Z_p^*$. For $i \in \{1, \ldots, q - 1\}$ and $j \in \{1, \ldots, N\}$, let $H_i, j = R_i + y_j$, then we have pairs $(H_i, j, \frac{1}{H_i + y_j} G_2)$.

$B$ then initializes a counter $t$ to 1 and randomly chooses an identity $I_1 \in \{0, 1\}^*$ under PKG, $j \in \{1, \ldots, N\}$ as the sender’s identity of a challenge to some forger in a chosen-message and given identity attack against our scheme. Now we describe the oracles that $B$ needed for answering necessary consultations. To maintain the consistency and to avoid collision, $B$ keeps three lists $L_1, L_2, L_3$ respectively to store the random answers of the random oracles $H_1, H_2$ and $H_3$.

- **$H_1$-queries on output identity $ID_U$ under some PKG, if it is the first time for $ID_U$ to query $H_1$, $B$ returns $(\omega + y_j) (\omega$ is randomly chosen from $Z_p^*)$ if $ID_U = I_D^*$ and returns $H_{\omega, \psi}$ if $ID_U \neq I_D^*$, then stores the answer in $L_1$ and increments $\omega$; otherwise, returns the corresponding value in existed information.

- **$H_2$-queries on output $(M, r)$ and $H_3$-queries for an input $r \in G_2$ are exactly the same as those proposed in the proof of theorem 1.

- **Key-extraction queries on an input $ID_o$ under some PKG, if $ID_o = I_D^*$, then $B$ fails. Otherwise, it knows that $H_1(ID_o, K_{puby}) = H_{v, j}$ and returns $\frac{1}{v, j} G_2 \in G_2$.

- **Signcryption queries for a plaintext $M$ and identities $(ID_S, ID_B) = (ID_{\mu}, ID_{\nu})$ under PKG, and PKG, respectively, $\mu, \nu \in \{1, \ldots, q\}$, $\mu, \nu \in \{1, \ldots, N\}$: we assume that $ID_{\mu} = ID_{\nu} = I_D^*$ for the same reason proposed in the signcryption query during the proof of theorem 1, and hence $ID_{\nu} \neq I_D^*$ by the irreflexivity assumption. $B$ randomly chooses $\lambda, h \in Z_p^*$, and computes $S = \lambda \psi(\mu, ID_{\nu}) = \lambda \psi(\frac{1}{\mu, \psi}, G_2), T = \lambda \psi((\omega + y_{\mu}) G_2 + K_{puby}) - h \psi(\psi(\omega + y_{\mu}) G_2 + K_{puby})$ in order to obtain the equality $r = e(T, S_{ID_{\nu}}) = e(S, (\omega + y_{\mu}) G_2 + K_{puby}) e(G_1, G_2)^{-h} = e(\psi(S_{ID_{\nu}}), (\omega + y_{\mu}) G_2 + K_{puby}) e(G_1, G_2)^{-h}$ before patching the hash value $H_2(M, r)$ to $h$ ($B$ fails if $H_2$ is already defined but this only happens with probability at most $\frac{2 \log q}{T}$). At last, the ciphertext $\sigma = (M \oplus H_3(r), \widetilde{S}, T)$ is returned.

- **Unsigncryption queries on a ciphertext $\sigma = (c, S, T)$ for identities $(ID_S, ID_B) = (ID_{\mu}, ID_{\nu})$ under PKG, and PKG, respectively: we assume that $ID_{\nu} = I_D^*$ (and hence $ID_{\nu} \neq I_D^*$ by the irreflexivity assumption) because otherwise B knows the receiver’s private key and can normally run the Unsigncryption algorithm. Therefore, $B$ has the sender’s private key $S_{ID_{\mu}} = \frac{1}{\mu, \psi} G_2$ and also knows that, for all valid ciphertexts, $\log_{S_{ID_{\nu}}}(\psi^{-1}(S) - h S_{ID_{\nu}}) = \log_{\psi(\psi(\mu, ID_{\nu}))}(T)$, where $h = H_2(M, r)$ is obtained in the Signcryption algorithm and $Q_{ID_{\mu}} = (\omega + y_{\mu}) G_2 + K_{puby}$. Hence, what $B$ should do is the same as that described in the unsigncryption query during the proof of theorem 1.

Now we show how to design an algorithm $F$ in a chosen-message and given identity attack to our scheme by running the ESUF-IBSCMP-CMA attacker $A$ that makes $q_h$ queries to random oracles $H_i$ ($i = 1, 2, 3$), $q_s$ signcryption queries and $q_u$ queries to unsigncryption oracle. For any $ID$ under some PKG, our algorithm $F$ is as follows:

1. $F$ chooses $l \in \{1, \ldots, q\}$ randomly. Denote by $(ID_j, K_{pubj})$ the input of the $j$-th query to $H_1$ asked by $A$. $F$ sets $(Q_{ID_j}, K_{pubj})$ to $(ID, K_{pub})$ if $j = l$ and $(ID_j, K_{pubj})$ otherwise.

2. $F$ runs $A$ with the given system parameters and responds to $A$’s queries to $H_1, H_2, H_3$, signcryption oracle and unsigncryption oracle by taking the place of $A$’s input $ID_j$ with $ID_j$ and running corresponding oracles.

3. Denote the output of $A$ as $(ID_{out}, m, \sigma)$. If $ID_{out} = ID$ and $(ID, m, \sigma)$ is valid, then $F$ outputs $(ID, m, \sigma)$; otherwise, $F$ fails.
We can see that the advantage of $F$ is $e^* \geq (1 - \frac{1}{q^2}) \frac{1}{q_0} e \geq 10(q_k + 1)(q_g + q_h_o)/2^k$, and its running time is almost the same as $A$ needs.

Subsequently, $B$ runs $F$ which is a forgery in a chosen-message and given identity attack to our scheme as a subroutine instead of $A$ and attempts to solve the $q$-SDHP. It is noted that $F$ can extract private keys associated to any identity but ID* by querying the key-extraction oracle. Therefore, thanks to the irreflexivity assumption, it is able to extract clear message-signature pairs from ciphertexts produced by the forger as it knows the private key of the receiving identity. Thus, we just consider the message-signature pairs which are decrypted from ciphertexts produced by the forger and could be seen as ID*'s valid signature. Applying the forking lemma [11, Theorem 13], from the forger $F$, we can build an algorithm $F'$ that replays $F$ a sufficient number of times on the input $(ID^*, K_{pub_i})$ to obtain two suitable forgeries which can be decrypted into two valid message-signature pairs $(m, r, h_1, S_1), (m, r, h_2, S_2)$ with $h_1 \neq h_2$, in expected time $t^* \leq 120686q_{h_i}^1/\epsilon$. Since both forgeries satisfy the verification equation, we obtain

$$e(S_1, Q_{ID^*})g^{-h_1} = e(S_2, Q_{ID^*})g^{-h_2},$$

where $Q_{ID^*} = (\omega + y_i)G + K_{pub_1} = (\omega + \alpha)G_2$. Then it comes that

$$e((h_1 - h_2)^{-1}S_1 - S_2, Q_{ID^*}) = e(G_1, G_2),$$

and hence $(h_1 - h_2)^{-1}(S_1 - S_2) = \frac{1}{\omega + \alpha}G_1$. Proceeded as in [2], $B$ can extract $\frac{1}{\omega + \alpha}P$ from $\frac{1}{\omega + \alpha}G_1$ as follows: it first obtains $\gamma_{-1}, \gamma_0, \ldots, \gamma_{q-2} \in \mathbb{F}_p^*$ for which

$$f(z) = \prod_{i=0}^{q-2} (z + R_i)$$

obtained at the preparation phase and eventually computes $\frac{1}{\omega + \alpha}G_1 = \sum_{i=0}^{q-2} \gamma_i \psi(a^iQ)) = \frac{1}{\omega + \alpha}P$. Thus, $B$ gets the pair $(\omega, \frac{1}{\omega + \alpha}P)$.

It finally comes that, since $F$ makes a forgery in a time $t$ with probability $e^ \geq (1 - \frac{1}{q^2}) \frac{1}{q_0} e \geq 10(q_k + 1)(q_g + q_h_o)/2^k$, $B$ solves the $q$-SDHP in time less than $120686q_{h_i}^1q_{h_o}^2 + \frac{t + O((q_k + q_h_o)q_g + q_h_o)}{e(1 - 1/2^k)} + O(q^2t_m)$, where the last term accounts for the cost of the preparation phase.

In the rest of this section, let us analyze the efficiency of our identity-based signcryption scheme for multiple PKGs. Table 1 summaries the number of relevant basic operations underlying several identity-based signcryption schemes for multiple PKGs, namely, $G_2$ exponentiations, scalar point multiplications, and pairing evaluations. It is noted that the computation of the pairing is the most time-consuming in pairing based cryptographic schemes [14]. It is easy to see from Table 1 that our proposal needs less pairing computations, so it is more efficient than that of Li et al.'s [8] and Lal-Kushwah’s [7].

<table>
<thead>
<tr>
<th></th>
<th>Li-Hu-Zhang</th>
<th>Lal-Kushwah</th>
<th>ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC exp mult pairing</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>USC exp mult pairing</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Note: SC – Signcryption; USC – Unsigncryption

6. Conclusion

We have proposed a new identity-based signcryption scheme for multiple PKGs, which is proved to be indistinguishable against adaptive chosen ciphertext attacks and signature-unforgeable against adaptive chosen plaintext and ciphertext attacks based on some computational assumptions. Compared with the state-of-the-art signcryption schemes of the same kind, our scheme needs less pairing computations and is the most efficient identity-based signcryption scheme in multiple PKGs environment up to date.

References


