An Efficient Identity-based Signcryption from Lattice

Xiufeng Zhao and Xiang Wang

Information Engineering University, Zhengzhou China
zhao_xiu_feng@163.com, dixonwx@163.com

Abstract

Signcryption is a cryptographic primitive that can provide valid solution for authentication and confidentiality. In this paper, an efficient identity-based signcryption scheme from lattice was proposed. Our scheme based on Macciancio and Piekert’s trapdoor generation method and delegate algorithm for lattice. Our main idea is that establish the identity-based system master key via trapdoor generation method for lattice, and extract user’s key by trapdoor delegate algorithm. With the aid of random oracle, we prove that the proposed scheme has CCA security under the learning with errors (LWE) assumption and unforgeability against adaptive chosen messages attacks under the small integer solution (SIS) assumption. Comparison with signature-then-encryption approach shows that our scheme is efficient.

Keywords: id-based cryptography, signcryption, lattice, LWE, SIS

1. Introduction

In CRYPTO1997, Zheng [1] proposed a new cryptographic primitive: signcryption, which can perform digital signature and public key encryption simultaneously at lower computational costs and communication overheads than signature-then-encryption approach, it can provide both confidentiality and authentication. Recently, lattice has emerged as a possible alternative to number theory. Ajtai [2] first gave the idea of trapdoor function for lattice and showed that the average-case hardness of some lattice problem is equivalent to its worst-case hardness. Additionally, lattice-based cryptography is believed to be secure against quantum computer attacks. Li et al., [3] first proposed a lattice-based signcryption scheme, and proved that the proposed scheme secure under lattice problems.

Motivation The concept of identity-based cryptosystem was firstly proposed by Shamir [4] in 1984. In Crypto 2001, Boneh et al., [5] proposed identity-based encryption from Weil pairing. Later, identity-based signcryption has been increasingly researched because of the simplicity of a public key management [6, 7, 8]. Our goal is to design an identity-based signcryption scheme from lattice hard problems.

Contribution We present the first identity-based signcryption scheme from lattice, and proved that the proposed scheme has the indistinguishability against adaptive chosen ciphertext attacks under the learning with errors (LWE) assumption and unforgeability against adaptive chosen message attacks under the small integer solution (SIS) assumption in the random oracle model. Comparison with sign-then-encrypt approach shows that our scheme is efficient.

Outline of the Paper We introduce the preliminary work about lattice in Section 2. In Section 3, we review the formal model of identity-based signcryption. In Section 4 we describe our new identity-based signcryption scheme from lattice. We compare our protocol with other approaches in Section 5 and conclude the paper in Section 6.
2. Preliminaries

2.1. Lattice and Hard Problems

An $m$-dimensional lattice $\Lambda$ is a discrete additive subgroup of $\mathbb{Z}^m$. For some $k \leq m$, called the rank of the lattice, $\Lambda$ is generated as the set of all integer combinations of some $k$ linearly independent basis vector $B = \{b_1, ..., b_k\}$, i.e., $\Lambda = \{Bz : z \in \mathbb{Z}^k\}$. For positive integers $n$ and $q$, let $A \in \mathbb{Z}_q^{n \times m}$ be arbitrary and define the following full-rank $m$-dimensional $q$-ary lattices:

$$\Lambda^\perp (A) = \{z \in \mathbb{Z}^m : Az = 0 \mod q\}$$

$$\Lambda^\perp (A^T) = \{z \in \mathbb{Z}^m : \exists s \in \mathbb{Z}_q^n, s.t. z = A^T s \mod q\}$$

For any $u \in \mathbb{Z}_q^n$ admitting an integral solution to $AX = u \mod q$, define coset (or “shifted” lattice)

$$\Lambda_u^\perp (A) = \{z \in \mathbb{Z}^m : Az = u \mod q\} = \Lambda^\perp (A) + X.$$  

Now, we give the conception of Gaussians on lattices. For any $s > 0$, the Gaussian function on $\mathbb{Z}^n$ cantered at $c$ with parameter $s$ is defined as

$$\forall x \in \mathbb{Z}^n, \rho_{s,c}(x) = \exp\left(-\pi \frac{\|x - c\|^2}{s^2}\right).$$

For any vector $c \in \mathbb{Z}_q^n$, real $s > 0$, and $n$-dimensional lattice $\Lambda$, the discrete Gaussian distribution over $\Lambda$ is defined as

$$\forall x \in \Lambda, D_{\Lambda,s,c}(x) = \frac{\rho_{s,c}(x)}{\rho_{s,c}(\Lambda)},$$

where $\rho_{s,c}(\Lambda) = \sum_{x \in \Lambda} \rho_{s,c}(x)$.  

Let $q \geq 2$ be an integer and let $\chi$ be a probability distribution on $\mathbb{Z}_q$. For an integer dimension $n$ and a vector $s \in \mathbb{Z}_q^n$, define $A_{s,x}$ to the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q^n$ of the variable $(a, a^T s + x)$, where $a \leftarrow \mathbb{Z}_q^n$ is uniform and $x \leftarrow \chi$ are independent, and all operations are performed in $\mathbb{Z}_q$.

**Definition 1 (LWE)** The goal of standard LWE problem is to find $s \in \mathbb{Z}_q^n$ (with overwhelming probability) given access to any desired $\text{poly}(n)$ number of samples from $A_{s,x}$ for some arbitrary $s$.

Regev [9] and Peikert [10] respectively showed that, for certain moduli $q$ and Gaussian error distributions $\chi$, the $\text{LWE}_{s,x}$ problem is hard as solving several standard worst-case lattice problems.
Definition 2 (SIS) Given an integer \( q \), a matrix \( A \in \mathbb{Z}_q^{n \times m} \), and a real \( \beta \), the small integer solution problem (in the \( l_2 \) norm) is to find a nonzero integer vector \( e \in \mathbb{Z}^n \) such that
\[
A e \equiv 0 \mod q \quad \text{and} \quad \|e\|_2 \leq \beta.
\]

Using Gaussian techniques, Micciancio and Regev [11] showed that the SIS problem is as hard (on the average) as approximating certain worst-case problems on lattices to within small factors.

2.2. Trapdoors for Lattice

Eurocrypt 2012, Micciancio and Peikert [12] gave a new notion of trapdoor for hard random lattice, which is simultaneously simple, efficient, tighter, faster, and small. They also gave algorithms for inverting LWE, randomly sampling SIS preimages, and securely delegating trapdoors, the following theorem summarizes the main results.

Definition 3 Let \( A \in \mathbb{Z}_q^{n \times m} \) and \( G \in \mathbb{Z}_q^{n \times w} \) be matrixes with \( m \geq w \geq n \). A trapdoor for \( A \) is a matrix \( R \in \mathbb{Z}_q^{(m-w) \times w} \) such that
\[
A = RHG
\]
for some invertible matrix \( H \in \mathbb{Z}_q^{n \times n} \). \( H \) is viewed as the tag or label of the trapdoor. In this paper, we let \( H = I \).

Theorem 1 [12] There is an efficient randomized algorithm \( \text{MR.GenTrap}(1^n, 1^n, q) \) that, gave any integers \( n \geq 1 \), \( q \geq 2 \), and sufficiently large \( m = O(n \log q) \), outputs a party-check matrix \( A \in \mathbb{Z}_q^{n \times m} \) and a trapdoor \( R \) such that the distribution of \( A \) is \( \text{negl}(n) \)-far from uniform. Moreover, there are efficient algorithms \( \text{MR.Invert} \), \( \text{MR.SampleD} \) and \( \text{MR.DelTrap} \) that with overwhelming probability over all random choices, do the following:

- For any \( b = A^T s + e \), where \( s \in \mathbb{Z}_q^n \) is arbitrary and either \( \|e\| < q \sqrt{O(n \log q)} \) or \( e \leftarrow D_{\mathbb{Z}_q^n} \) for \( 1/\alpha \geq 1/\sqrt{n \log q} \cdot \omega(\sqrt{\log n}) \), the deterministic algorithm \( \text{MR.Invert}(R, A, b) \) outputs \( s \) and \( e \).
- For any \( u \leftarrow \mathbb{Z}_q^n \) and large enough \( s = O(\sqrt{n \log q}) \), the randomized algorithm \( \text{MR.SampleD}(R, A, u, s) \) samples from a distribution within \( \text{negl}(n) \) statistical distance of \( D_{\mathbb{Z}_q^n} \).
- For any valid input \( A' = A \parallel H \), and \( H' \), algorithm \( \text{MR.DelTrap}(A', 1, s') \) outputs a trapdoor \( R' \) for \( A' \) with tag \( H' \), whose distribution is the same for any discrete Gaussian sampling over coset of \( \Lambda = \mathbb{Z}^n \) with parameter \( s' \geq \eta_e(A) \), and \( s_1(R') \leq s'(\sqrt{m} + \sqrt{w}) \) except with negligible probability, where \( s_1(R') \) denotes the maximal singular value of \( R' \).
3. Formal Model of Identity-based Signcryption

3.1. Generic Scheme

An identity-based signcryption scheme consists of the following algorithms.

- **Setup**: Given a security parameter \( n \), PKG generates a master key \( s \) and common parameters \( \text{params} \). \( \text{params} \) is made public while \( s \) is kept secret.

- **Extract**: Given an identity \( id \), the PKG runs this algorithm to generate the private key \( s_{id} \) associated with \( id \) and transmits it to user \( id \) via a secure channel.

- **Signcrypt**: To send a message \( m \) to Bob whose identity is \( id_B \), Alice with identity \( id_A \) obtains a ciphertext \( \sigma \) by computing \( \text{Signcrypt}(m, s_{id_A}, id_B) \).

- **Unsigncrypt**: After Bob receives the ciphertext \( \sigma \), he computes \( \text{Unsigncrypt}(\sigma, s_{id_B}, id_B) \) and obtains the message \( m \) or the symbol \( ⊥ \) indicating that the ciphertext is invalid.

3.2. Security Notions

In this section, we will recall Malone-Lee's [13] security models for id-based signcryption scheme.

**Definition 4** An id-based signcryption scheme is said to have the indistinguishability against adaptive chosen ciphertext attacks property (IND-IDSC-CCA) if no polynomially bounded adversary has a non-negligible advantage in the following game.

**Setup** The challenger \( C \) runs the Setup algorithm with a security parameter \( k \) and obtains common parameters \( \text{params} \) and a master key \( s \). He sends \( \text{params} \) to the adversary \( A \) and keeps \( s \) secret.

**Phase 1** The adversary \( A \) performs a polynomial bounded number of queries. These queries may be made adaptively, i.e., each query may depend on the answers to the previous queries.

- Key extraction queries: \( A \) produces an identity \( id \) and receives the extracted private key \( s_{id} = \text{Extract}(id) \).

- Signcryption queries: \( A \) produces two identities \( id_i, id_j \) and a plaintext \( m \). \( C \) computes \( s_{id_i} = \text{Extract}(id_i) \) and \( \sigma = \text{Signcrypt}(m, s_{id_i}, id_i) \) and sends \( \sigma \) to \( A \).

- Unsigncryption queries: \( A \) produces two identities \( id_i, id_j \) and a ciphertext \( \sigma \). \( C \) generates the private key \( s_{id_j} \) and sends the result of \( \text{Unsigncrypt}(m, s_{id_j}, id_j) \) to \( A \). This result may be the symbol \( ⊥ \) if \( \sigma \) is an invalid ciphertext.

**Challenge** \( A \) chooses two plaintexts, \( m_0 \) and \( m_1 \), and two identities, \( id_A \) and \( id_B \), on which he wishes to be challenged. He cannot have asked the private key corresponding \( id_B \) in the first stage. \( C \) chooses randomly a bit \( γ \), computes \( \sigma = \text{Signcrypt}(m_γ, s_{id_A}, id_B) \) and sends it to \( A \).
Phase 2 A asks a polynomial number of queries adaptively again as in the first stage. It is not allowed to extract the private key corresponding to \( id_A \) and it is not allowed to make an unsigncryption query for \( \sigma \) under \( id_A \).

Guess Finally, A produces a bit \( \gamma \)' and wins the game if \( \gamma = \gamma \). A ’s advantage is defined as
\[
Adv(A) = 2 \Pr[\gamma' = \gamma] - 1.
\]

Definition 5 An id-based signcryption scheme is said to be secure against an existential forgery for adaptive chosen message attacks (EUF-IDSC-CMA) if no polynomial bounded adversary has a non-negligible advantage in the following game.

(1) The challenger C runs the Setup algorithm with a security parameter \( k \) and obtains common parameters \( \text{params} \) and a master key \( s \). He sends \( \text{params} \) to the adversary A and keeps \( s \) secret.
(2) The adversary A performs a polynomial bounded number of queries adaptively just like in the previous definition.
(3) Finally, A produces a new triple \( (\sigma^*, id_A^*, id_B^*) \), where the private key of \( id_A^* \) was not asked in the second stage and wins the game if the result of \( \text{Unsigncrypt}(\sigma^*, id_A^*, id_B^*) \) is not the \( \bot \) symbol.

4. The Signcryption Scheme

In this section, we present an efficient identity-base signcryption scheme from lattice. Parameters set as \( q = \text{poly}(n) \), \( m = O(n \log q) \), \( w = n \lceil \log q \rceil \), \( s = O\left(\sqrt{n \log q}\right) \). Our identity-based signcryption scheme consists of the following algorithms:

System setup: Let \( H_1 : \{0,1\}^l \rightarrow Z_q^{m \times w} \) be a hash function maps identity to a matrix, \( H_2 : \{0,1\}^* \rightarrow Z_q^w \) be a hash function maps message and a random string to a vector, \( H_3 : \{0,1\}^* \rightarrow \{0,1\}^l \), where \( l \) is the message length. Given the security parameter \( n \), the KGC runs the algorithm \( \text{MR.GenTrap}(1^n, 1^n, q) \), generates \( A \in \mathbb{Z}_q^{n \times m} \) and \( R \in \mathbb{Z}_q^{w \times w} \) satisfy \( A \begin{bmatrix} R \\ I \end{bmatrix} = G \), where \( m = m - w \), \( G \) is a public primitive matrix. Finally, KGC keeps \( R \) as the master secret key and publishes system parameters \( \text{params} = \{A, G, H_1, H_2, H_3\} \).

Extract: Given a public identity \( id \in \{0,1\}^l \), the KGC runs \( \text{MR.DelTrap}(A_{id} = A \parallel H_1(id), I, s') \), where \( s' \geq \eta_q(A^\perp(A)) \), outputs \( R_{id} \in Z_q^{w \times w} \) as associated private key, which satisfy \( A \parallel H_1(id) \begin{bmatrix} R_{id} \\ I \end{bmatrix} = G \). Suppose identity of sender is \( id_s \) and secret key of sender is \( R_s \), the identity of receiver is \( id_r \) and secret key of receiver is \( R_r \).
**Signcrypt:** The sender wants to send message \( m \) to the receiver, he follows the following steps:

1. Choose \( r \leftarrow \{0,1\}^k \) randomly;
2. Run the algorithm \( \text{MR.Sampled} (R, A \parallel H_1(id_i), H_2(m \parallel r), s) \), output \( \delta \in \mathbb{Z}_q^{n \times w} \) sampled from \( D_{\text{A}[H_1(id_i), s]} \) with trapdoor \( R \), for \( A \parallel H_1(id_i) \);
3. Randomly choose a vector \( s \leftarrow \mathbb{Z}_q^n \), compute \( c = m \oplus H_3(s, \delta) \);
4. Compute \( u = (A \parallel H_1(id_i)) \cdot s + \delta \).

The signcryption ciphertext is \( \sigma = (r, c, u) \).

**Unsigncrypt:** When receiving the ciphertext \( \sigma = (r, c, u) \), the receiver follows the following steps:

1. Run the algorithm \( \text{MR.Invert} (R, A \parallel H_1(id_i), u) \), output \( (s, \delta) \);
2. Recover the message by computing \( m = c \oplus H_3(s, \delta) \);
3. Accept if \( r \leftarrow \{0,1\}^k \), \( \|s\| \leq s \cdot \sqrt{m + w} \) and \( (A \parallel H_1(id_i)) \cdot \delta = H_2(m \parallel r) \); otherwise reject.

**Security analysis**

**Theorem 2** In the random oracle model, if an IND-IDSC-CCA adversary \( A \) is able to distinguish two valid ciphertexts during the game defined in definition 5 with a non-negligible advantage \( \varepsilon \) and asking at most \( q_{hi}, (i = 1, 2, 3) \) random oracle queries, \( q_e \) extract queries, \( q_{sc} \) signcryption queries and \( q_{usc} \) unsigncryption queries, then there exists a solver \( C \) that can be solve LWE problem with advantage \( \frac{(q_e - 1)(q_{sc} - 1)(q_{usc} - 1)}{q_e q_{sc} q_{usc} q_{hi}} \varepsilon \).

**Proof:** Suppose challenger \( C \) receives a random LWE instance \( (\widetilde{A}, \widetilde{u} = \widetilde{A} \cdot \widetilde{s} + \widetilde{\delta}) \), his goal is to compute \( \widetilde{s} \) and \( \widetilde{\delta} \). \( C \) will run the adversary as a subroutine and act as challenger in the IND-IDSC-CCA game.

**Setup** Suppose \( \widetilde{A} = \widetilde{A}_1 \parallel \widetilde{A}_2 \), the challenger \( C \) let \( A = \widetilde{A}_1 \),

\[ \text{params} = \{A, G, H_1, H_2, H_3\} \], and send public parameters \( \text{params} \) to the adversary \( A \), where \( \{H_1, H_2, H_3\} \) viewed as random oracle.

**Phase 1** \( C \) maintains lists \( L_1, L_2 \) and \( L_3 \) for simulating hash oracles \( H_1, H_2 \) and \( H_3 \), respectively, and answers the queries as follows:

**Random Oracle queries**

- **H_1 queries:** For an \( H_1(id_i) \) query \( (1 \leq i \leq q_{hi}) \), the challenger \( C \) first checks if the value \( H_1(id_i) \) was previously defined for the input identity \( id_i \). If it was, the previously defined value is returned. Otherwise, \( C \) chooses a matrix \( R_{id_i} \in \mathbb{Z}_q^{n \times w} \) from distribution
\[ D \text{, where } D \text{ is the subgaussian with parameter } s > 0 \text{, computes} \]
\[ h_{i_1} = H(id_{i_1}) = G - A R_{id_{i_1}} \text{, and inserts } (id_{i_1}, R_{id_{i_1}}, h_{i_1}) \text{ into } L_1 \text{, returns } h_{i_1} \text{ to adversary.} \]

Especially, for an \( H_i(id_j) (1 \leq j \leq q_{H1}, j \neq i) \) query, the challenger defined \( h_{i_1} = \overline{A}_2 \), and inserts \( (id_j, R_{id_j}, h_{i_1}) \) into \( L_1 \), and returns \( h_{i_1} \) to adversary.

- **\( H_2 \) queries:** For an \( H_2(m \parallel r) \) query \((1 \leq i \leq q_{H2})\), the challenger \( C \) first checks if the value \( H_2 \) was previously defined for the input \((m, r)\). If it was, the previously defined value is returned. Otherwise, \( C \) chooses \( h_{i_2} \in \mathbb{F}_q \) randomly, inserts \((m, r, h_{i_2})\) into \( L_2 \), and returns \( h_{i_2} \) to adversary.

- **\( H_3 \) queries:** For an \( H_3(s_i \parallel \delta_i) \) query \((1 \leq i \leq q_{H3})\), the challenger \( C \) first checks if the value \( H_3 \) was previously defined for the input \((s_i, \delta_i)\). If it was, the previously defined value is returned. Otherwise, \( C \) chooses \( h_{i_3} \in \{0,1\}^j \) randomly, inserts \((s_i, \delta_i, h_{i_3})\) into \( L_3 \), and returns \( h_{i_3} \) to adversary.

**Extract queries** When the adversary asks for a private key corresponding to an identity \( id_j (1 \leq j \leq q) \), \( C \) checks for list \( L_{H1} \), if it was previously defined for the input \( id_j \), then returns \( R_{id_j} \). Otherwise, \( C \) chooses a matrix \( R_{id_j} \in \mathbb{F}_q^{m \times w} \) from distribution \( D \), where \( D \) is the subgaussian with parameter \( s > 0 \), computes \( h_{i_1} = H(id_j) = G - A R_{id_j} \), and inserts \((id_j, R_{id_j}, h_{i_1})\) into \( L_1 \), and returns \( R_{id_j} \) to the adversary.

**Signcryption queries** At some time, the adversary performs a signcryption query for a plaintext \( m \) with signcryption \( id_A \) and unsigncryption \( id_B \). If \( id_A \neq id_j \), the challenger \( C \) first generates a private key for \( id_A \) by extract query described above, and returns \( signcrypt(m, R_{id_A}, id_B) \) to answer the adversary’s query; else, the simulation abort.

**Unsigncryption queries** If the adversary submits an unsigncryption query for a ciphertext \( \sigma = (r, c, u) \) with signcryption \( id_A \) and unsigncryption \( id_B \). If \( id_B \neq id_j \), the challenger \( C \) first generates a private key for \( id_B \) by extract query described above, and returns \( unsigncrypt(\sigma, id_A, R_{id_B}) \) to answer the adversary’s query.

**Challenge** After a polynomial bounded number of queries, the adversary chooses identities \( id_A^* \) and \( id_B^* \), which he wishes to be challenged. Then the adversary submits two messages \( m_0 \) and \( m_1 \) to the challenger. \( C \) randomly chooses two binary strings \( r^* \leftarrow \{0,1\}^k \) and \( c^* \leftarrow \{0,1\}^j \), and sets \( u^* = \overline{u} \). Finally, \( C \) returns \( \sigma^* = (c^*, r^*, u^*) \) to the adversary.

**Phase 2:** \( \wedge \) can ask a polynomial bounded number of queries adaptively again as in Phase 1 with the restriction that it is not allowed to extract the private key corresponding to \( id_B^* \) and it is not allowed to make an unsigncryption query for \( \sigma^* \) under \( id_B^* \).
**Guess:** A produces a result $\gamma'$, which is ignored by $C$. $C$ just looks into the list $L_c$ for tuples of the form $(s_i, \delta_i, h_{s_i})$. For each of them, $C$ checks whether $\bar{u} = \overline{A}' s_i + \delta_i$ holds. If this equation holds, $C$ stops and outputs $(s_i, \delta_i)$ as a solution of the challenge LWE problem. If no tuple of this kind satisfies the equation, $C$ stops and outputs “failure”.

**Probability of success.** Now, we assess $C$'s probability of success. For the simulation is perfect without aborting, we require the following conditions fulfilled:

1. Extract queries: $id \neq id$;
2. Signcryption queries: $id_a \neq id$;
3. Unsigncryption queries: $id_b \neq id$;

Then the probability of $C$ not aborting is

$$\Pr\{\text{abort}\} = \left( 1 - \frac{1}{q_e} \right) \left( 1 - \frac{1}{q_{sc}} \right) \left( 1 - \frac{1}{q_{sc}} \right) \frac{1}{q_{H_1}} \frac{(q_e - 1)(q_{sc} - 1)(q_{ase} - 1)}{q_e q_{sc} q_{ase} q_{H_1}} .$$

If the simulation does not abort, the adversary will win the game in definition 5 with probability at least $\varepsilon$. Then $C$ can solve for the LWE problem instance with probability

$$\frac{(q_e - 1)(q_{sc} - 1)(q_{ase} - 1)}{q_e q_{sc} q_{ase} q_{H_1}} \varepsilon .$$

**Theorem 3** In the random oracle model, if an EUF-IDSC-CMA forger $F$ is able to forge a valid signcryption ciphertexts during the game defined in definition 6 with an advantage non-negligible advantage $\varepsilon$ and asking at most $q_{H_i}$ ($i = 1, 2, 3$) random oracle queries, $q_e$ extract queries, $q_{sc}$ signcryption queries and $q_{ase}$ unsigncryption queries, then there exists a solver $C$ that can be solve SIS problem with advantage $\frac{(q_e - 1)(q_{ase} - 1)}{q_e q_{ase} q_{H_1}} \varepsilon$.

**Proof:** Suppose challenger $C$ receives a random SIS problem instance $\bar{A} \cdot X = 0 \mod q$, his goal is to compute $X$ such that $\|X\| \leq \beta$. $C$ will run the forger as a subroutine and act as challenger in the EUF-IDSC-CMA game.

$C$ first sets the public parameters using the setup algorithm described in the previous proof. Then the forger can perform a polynomial bounded number queries including random oracle queries, private key extraction queries, signcryption queries, and unsigncryption queries. The challenger $C$ answers the forger in the same way as that of Theorem 2 except $H_2$ query. When asked $H_2(m_i, \| r_i \) (1 \leq i \leq q_{H_2})$ query, $C$ answers as follows:

$C$ randomly chooses $\delta_{m_i}$, s.t. $\| \delta_{m_i} \| \leq s \cdot \sqrt{m + w}$, sets

$$\bar{A} \cdot \delta_{m_i} = H_2(m_i, \| r_i \) . \quad (1)$$
C stores \((m_j, r_j, \delta_M, h(m_j || r_j))\) to \(L_2\), and returns \(H_2(m_j, r_j)\) to forgery \(F\).

Finally, if \(C\) does not aborts, the forger will return a new ciphertext \(\sigma^*\) on message \(m^*\), where \(m^*\) has never been queried under identity \(id_A^*\) and \(id_B^*\). Now, \(C\) can unsigncrypt \(\sigma^*\) and obtain \((r^*, m^*, \delta^*)\) by extracting private key of \(id_B^*\), satisfy \(\|\delta^*\| \leq s \cdot \sqrt{m + w}\) and

\[
(A || H_1(id_A^*)) \cdot \delta^* = H_2(m^* || r^*). 
\] (2)

Before forging signcryption \(\sigma^*\), \(F\) queried \(H_2\) on message \(m^*\) and \(r^*\), so \(C\) looks into the list \(L_2\) for the tuple \((m^*, r^*, \delta_M^*, h(m^* || r^*))\), such that

\[
\overline{A} \cdot \delta_M^* = H_2(m^* || r^*). 
\] (3)

If \(id_A^* = id_j\), \(C\) computes (2) subtracts (3), obtains the following equation:

\[
(A || H_1(id_j)) \cdot (\delta^* - \delta_M^*) = 0. 
\] (4)

i.e. \(\overline{A} \cdot (\delta^* - \delta_M^*) = 0\), and sets \(X = \delta^* - \delta_M^*\) to be the solution for the SIS problem \(\overline{A}X = 0 \mod q\).

**Probability of success.** Now, we assess \(C\)'s probability of success. For the simulation is perfect without aborting, we require the following conditions fulfilled:

1. Extract queries: \(id \neq id_j\);
2. Signcryption queries: \(id_A \neq id_j\);
3. Unsigncryption queries: \(id_B \neq id_j\);
4. Forgery ciphertext: \(id_A^* = id_j\).

According the EUF-IDSC-CMA game in definition 6, the condition (4) holds implies that the condition (1) holds. Then the probability of \(C\) not aborting is

\[
\Pr[\text{abort}] = \left(1 - \frac{1}{q_{sc}}\right) \cdot \left(1 - \frac{1}{q_{asc}}\right) \cdot \frac{1}{q_{H1}} = \frac{(q_{sc} - 1)(q_{asc} - 1)}{q_{sc}q_{asc}q_{H1}}. 
\]

If the simulation does not abort, the forger \(F\) will win the game in definition 6 with probability at least \(\varepsilon\). Then \(C\) can solve for the SIS problem instance with probability

\[
\frac{(q_{sc} - 1)(q_{asc} - 1)}{q_{sc}q_{asc}q_{H1}} \varepsilon. 
\]

**5. Comparison**

**Comparison with signature-then-encryption approach.** Firstly, we compare our scheme with the signature-then-encryption approach, as show in Table 1. Suppose the sender want to
send 1 bit message to the receiver in a confidential and authentic way by IBS scheme [14] and GPV IBE scheme[15] respectively. The signature length is \( k + (m + 1)l_q \), where \( l_q \) is the number of bits required to represent an element of \( \mathbb{Z}_q \). GPV IBE scheme encrypting one bit generates \((m + 1)l_q\) bits ciphertext. Therefore, the ciphertext length of signature-then-encryption approach is \((k + (m + 1)l_q)(m + 1)l_q\), where, \( l_q \) is the number of bits required to represent an element of \( \mathbb{Z}_q \). The secret key size of sender is \((m + 1)^2 l_q\), and the secret key size of receiver is \(ml_q\). As far as our identity-based signcryption scheme, the secret key size of both sender and receiver is \( n^2 \lceil \log q \rceil l_q \), the ciphertext length is \( k + l + (m + n \lceil \log q \rceil)l_q \). Accordingly, our scheme is an efficient identity-based signcryption scheme.

**Comparison with other identity-based signcryption scheme.** Secondly, we compare our identity-based signcryption scheme to the signcryption scheme based lattice [3], including the size of secret key, computation cost and ciphertext length, as show in Table 1, where \( H \) denotes hash function operation, \( M \) denotes matrix-vector multiplication operation, XOR denotes exclusive or operation. LWE.Invert denotes solving LWE problem, and SampleD denotes Gaussian preimage sampling from a discrete Gaussian over a desired coset of \( \Lambda \). From Table 1, we can conclude that our identity-based signcryption scheme has advantage over secret key size and simplicity of key management, and the computational cost is equivalent.

**Table 1. Performance Comparison**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Secret Key Size</th>
<th>Ciphertext Length</th>
<th>Computation Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign-then-Encrypt approach</td>
<td>sender: ((m + 1)^2 l_q)</td>
<td>signature: (k + (m + 1)l_q)</td>
<td>SampleD+4H+M</td>
</tr>
<tr>
<td>Id-based signcryption[14]</td>
<td>receiver: (ml_q)</td>
<td>ciphertext: ((k + (m + 1)l_q)(m + 1)l_q)</td>
<td>(3(k + (m + 1)l_q)) M</td>
</tr>
<tr>
<td>Id-based encryption[15]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signcryption[3]</td>
<td>(ml_q)</td>
<td>(k + l + ml_q)</td>
<td>SampleD+LWE.Invert+4H+2M+2XOR</td>
</tr>
<tr>
<td>Our scheme</td>
<td>(n^2 \lceil \log q \rceil l_q)</td>
<td>(k + l + (m + n \lceil \log q \rceil)l_q)</td>
<td>SampleD+LWE.Invert+8H+2M+2XOR</td>
</tr>
</tbody>
</table>

**6. Conclusion**

We present the first identity-based signcryption scheme from lattice. The scheme can be proven CCA security under LWE assumption and SUF-CMA secure under SIS assumption. The challenge work is that how to achieve identity-based signcryption from lattice without random oracle.

**Acknowledgements**

This work is supported by the National Natural Science Foundation of China under Grant No. 61173139.
References


Authors

XiuFeng Zhao received the B.S. degree from Qufu Teacher University in 2000, and M.S. degree in Northwestern Polytechnical University in 2003. In 2012, she received her Ph.D. degree in Shandong University. She is now a teacher of Information Engineering University. Her recent research directions are lattice cryptography and provable security.

Xiang Wang received B.S. degree, M.S. degree and Ph.D. degree from Information Engineering University. He is a teacher in the Information Engineering University. His research field is public key cryptosystem.