Security Analysis of a Certificateless Signature from Lattices

Seunghwan Chang,1 Hyang-Sook Lee,2 Juhee Lee,1 and Seongan Lim1

1Institute of Mathematical Sciences, Ewha Womans University, Seoul 120-750, Republic of Korea
2Department of Mathematics, Ewha Womans University, Seoul 120-750, Republic of Korea

Correspondence should be addressed to Juhee Lee; juhee1108@gmail.com

Received 1 July 2016; Revised 30 November 2016; Accepted 18 December 2016; Published 26 January 2017

Academic Editor: Pino Caballero-Gil

Tian and Huang proposed a lattice-based CLS scheme based on the hardness of the SIS problem and proved, in the random oracle model, that the scheme is existentially unforgeable against strong adversaries. Their security proof uses the general forking lemma under the assumption that the underlying hash function $H$ is a random oracle. We show that the hash function in the scheme is neither one-way nor collision-resistant in the view of a strong Type 1 adversary. This means that the hash function defined from $H$ cannot be modelled as a random oracle and this indicates critical flaws in the security argument.

We show that the CLS scheme is insecure against strong Type 1 adversaries by providing effective attack algorithms. The attack algorithms, which are successful in the strong Type 1 adversarial model, are based on the weak properties of the hash function that we have found.

1. Introduction

The notion of certificateless signature (CLS) has been introduced by Al-Riyami and Paterson [1] in 2003 as a variant of identity-based signature (IBS) to eliminate the key escrow problem inherent in IBS and assuage the certificate management of regular signatures. To solve the key escrow problem, a user’s private key in a CLS scheme is not generated by the KGC alone. Instead, it is a combination of a secret from the KGC and one chosen by the user. More precisely, each user has two secrets: a secret value generated by the user and a partial private key produced by the KGC, who holds the master key. Signing requires both secrets. Since the KGC does not have access to the secret value generated by the user, the key escrow problem can be solved.

Lattice-based signature schemes are an important alternative for the current number-theoretic signature schemes and are emerging as a promising candidate for postquantum cryptography on the basis of Shor’s work [2]. There are many lattice-based cryptosystems, including encryption schemes [3, 4]. There have been various attempts to construct lattice-based CLS schemes.

The first lattice-based signature scheme was proposed by Gentry et al. [5] as a Hash-and-Sign signature scheme and its security is based on the hardness SIS problem on the average case. Lyubashevsky and Micciancio [6] proposed a lattice-based one-time signature scheme in 2008. Lyubashevsky [7] proposed a lattice-based signature by extending the scheme of Lyubashevsky and Micciancio [6] in the framework of Fiat-Shamir. Since then, a number of lattice-based signature schemes have been proposed in the context of PKI. Tian and Huang [8] proposed a IBS scheme following the framework of Lyubashevsky [7].

In 2015, Tian and Huang [9] proposed a lattice-based CLS scheme and proved under the SIS assumption that the scheme is existentially unforgeable against strong adversaries, in the random oracle. In this paper, we discuss security flaws in the CLS scheme of Tian and Huang by scrutinizing misuses of the hash function in the security arguments. The security proof of the scheme is given in random oracle model and uses the general forking lemma under the assumption that the underlying hash function $H$ is a random oracle. We show that the hash function is neither one-way nor collision-resistant in the view of a strong Type 1 adversary. This means that the hash function defined from $H$ cannot be modelled as a random oracle and this indicates critical flaws in the security argument.

We show that the CLS scheme is insecure against strong Type 1 adversaries by providing effective attack algorithms. The attack algorithms, which are successful in the strong Type 1 adversarial model, are based on the weak properties of the hash function that we have found.
The rest of the paper is organized as follows. In Section 2 we give preliminaries on CLS schemes and review the CLS scheme of Tian and Huang as well as their security proofs. In Section 3, we analyze the security arguments of Tian and Huang and point out security flaws. We show that the scheme is insecure under strong Type 1 attack. We draw our conclusion in Section 4.

2. Review of the Certificateless Signature Scheme of Tian and Huang

In 2015, Tian and Huang proposed a CLS scheme (from hereon in referred to simply as “Tian-Huang scheme”) and they claimed that their scheme is provably secure in the strong Type 1 adversarial model by assuming the hardness of the SIS problem [9]. In this section, we review some basics of Tian-Huang scheme and their security proof.

2.1. Some Basics of SIS Problem. The security of Tian-Huang scheme is based on the hardness of the SIS problem. The SIS problem can be stated as follows.

Definition 1. Given a positive integer \( q \), a matrix \( A \in \mathbb{Z}_q^{n \times m} \), and a positive real number \( \gamma \), the \((q, m, \gamma)\)-SIS problem for \( A \) is to find a nonzero vector \( v \in \mathbb{Z}^m \) such that \( Av \equiv 0 \) (mod \( q \)) and \( \|v\| \leq \gamma \).

One of the related problems of the SIS problem is the Inhomogeneous-SIS (ISIS) problem that can be described as follows.

Definition 2. Given a positive integer \( q \), a matrix \( A \in \mathbb{Z}_q^{n \times m} \), a vector \( u \in \mathbb{Z}^n \), and a positive real number \( \gamma' \), the \((q, m, \gamma')\)-ISIS problem for \((A, u)\) is to find a nonzero vector \( v \in \mathbb{Z}^m \) such that \( Av = u \) (mod \( q \)) and \( \|v\| \leq \gamma' \).

For any polynomial-bounded \( m, \gamma \) and for any prime \( q \geq \gamma \cdot o(\sqrt{n \log n}) \), it is known by Micciancio and Regev [10] that solving SIS on average is hard as approximating some intractable lattice problem.

The polynomial-time algorithms \( \text{TrapGen} \) and \( \text{SampleMat} \) are building blocks of Tian-Huang scheme. We skip the details of each algorithms in this paper.

(i) \((A, T_A) \leftarrow \text{TrapGen}(q, n, m, \gamma)\) if and only if \( T_A \in \mathbb{Z}_q^{n \times m} \) is of full rank and each column vector of \( T_A \) is a solution of \((q, m, \gamma)\)-SIS problem for the matrix \( A \in \mathbb{Z}_q^{n \times m} \).

(ii) \((A, T_A) \leftarrow \text{TrapGen}(q, n, m, \gamma), (D \in \mathbb{Z}_q^{m \times k}) \leftarrow \text{SampleMat}(A, T_A, \gamma', u)\) if and only if each of the column vectors of \( D \) is a solution of \((q, m, \gamma')\)-ISIS problem for \((A, u)\).

2.2. A Description of Tian-Huang Scheme [9]. In this subsection we give a brief description of Tian-Huang scheme; for the full details, see [9]. Major public parameters of the scheme are the following:

(i) \( n \): security parameter.

(ii) \( q \): prime number.

(iii) \( M \): a positive real number.

(iv) \( m, m_1, m_2, b, k, \lambda, s, \sigma \): positive integers with \( m = m_1 + m_2 \), \( m_1 > 2m \log q \), \( m_2 > 64 + n \log q/(2b + 1) \) and \( s = \Omega(\sqrt{n \log q}) \), \( \sigma = 12\lambda m \).

The scheme consists of the following seven algorithms:

(i) \( \text{Setup}(n \in \mathbb{Z}) \): on input the security parameter \( n \), the KGC

(a) computes \((A, T_A) \leftarrow \text{TrapGen}(q, n, m)\),

(b) chooses a random matrix \( B \in \mathbb{Z}_q^{n \times m} \),

(c) chooses two secure hash functions

\[ F : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{m \times k}, \]
\[ H : \mathbb{Z}_q^{m \times k} \times \{0, 1\}^* \rightarrow \{v \in \{−1, 0, 1\}^k : \|v\|_1 \leq \lambda\}, \]

(d) outputs the master secret \( msk = T_A \) and the public parameters \( \text{params} = \{A, B, F, H\} \).

(ii) \( \text{PartialPrivateKeyExtract}(id, msk, \text{params}) \): on input \((id, msk = T_A, \text{params})\), the KGC obtains \( D_{id} \in \mathbb{Z}_{m \times k} \leftarrow \text{SampleMat}(A, T_A, s, F(id)) \) and sends it to the user with id. Upon receiving \( D_{id} \), the user with id checks the correctness of \( D_{id} \) by verifying if \( AD_{id} = F(id) \) and \( \|D_{id}\| \leq s/\sqrt{m} \). If so, the user sets his partial private key as \( psk_{id} = D_{id} \).

(iii) \( \text{SetSecretValue}(id, \text{params}) \): on input \((id, \text{params})\), the user with id selects a random matrix \( S_d \in \mathbb{Z}_q^{m \times k} \) satisfying \( \|S_d\|_{\infty} \leq b \) and outputs his secret value \( x_{id} = S_d \).

(iv) \( \text{SetPrivateKey}(id, x_{id}, psk_{id}, \text{params}) \): on input \((id, x_{id} = S_d, psk_{id} = D_{id}, \text{params})\), the user with id outputs his full private key \( SK_{id} = (psk_{id}, x_{id}) = (D_{id}, S_d) \in \mathbb{Z}_q^{m \times k} \times \mathbb{Z}_q^{m \times k} \).

(v) \( \text{SetPublicKey}(id, SK_{id}, \text{params}) \): on input \((id, SK_{id} = (D_{id}, S_d), \text{params})\), the user with id outputs his public key \( PK_{id} = BS_{id} \).

(vi) \( \text{CLSign}(id, SK_{id}, m, \text{params}) \): on input \((id, SK_{id} = (D_{id}, S_d), m, \text{params})\), the user with id

(a) selects \( y_1 \leftarrow D_{id}^m \) and \( y_2 \leftarrow D_{id}^s \) and sets \( y = \left[ \begin{array}{c} y_1 \\ y_2 \end{array} \right] \in \mathbb{Z}_q^m \).

(b) sets \( h = H \left( \left[ \begin{array}{c} Ay_1 \\ Bz \end{array} \right] \right) \in \{−1, 0, 1\}^k \) and \( c = \left[ \begin{array}{c} D_{id} \\ S_d \end{array} \right] h \in \mathbb{Z}_q^m \).

(c) computes \( z = c + y \in \mathbb{Z}_q^m \).

(d) outputs the signature \( \text{sig} = (h, z) \) on \( m \) with probability \( \min(1, D_{id}^m(z)/MD_{id}^s(z)) \). If nothing is outputted, repeat this algorithm.

(vii) \( \text{CLVfy}(id, PK_{id}, m, \text{params}) \): on input \((id, PK_{id} = BS_{id}, \text{sig} = (h, z), m, \text{params})\), the algorithm outputs 1 only if \( \|z\| \leq 2\sigma \sqrt{m} \), \( h = H \left( \left[ \begin{array}{c} Az \\ Bz \end{array} \right] \right) \in \{−1, 0, 1\}^k \), \( m \), where \( z = \left[ \begin{array}{c} z_1 \\ z_2 \end{array} \right] \).
Correctness. The correctness of Tian-Huang scheme is clear from the fact that the following holds for any correctly computed key pairs and a signature (h, z) on m.

$$\begin{bmatrix} A z_1 \\ B z_2 \end{bmatrix} - \begin{bmatrix} F(\text{id}) \\ \text{PK}_{\text{id}} \end{bmatrix} h = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \left( \begin{bmatrix} 0 \\ h \end{bmatrix} - \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} D_{\text{id}} \\ S_{\text{id}} \end{bmatrix} h \right)$$

$$= \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} (z - c) = \begin{bmatrix} A y_1 \\ B y_2 \end{bmatrix}.$$  \hspace{1cm} (2)

2.3. Security Model for CLS Schemes. The most general security notion of regular signature is the existential unforgeability under an adaptively chosen message attack. It is extended to IBS, namely, existential unforgeability under an adaptively chosen message and an adaptively chosen-ID attack, where an adversary can choose its messages and its identities adaptively. The security notion of CLS is similar to that of IBS, but it is more complicated from the following facts:

(i) The KGC should be considered as an adversary because it is not a trusted party.

(ii) There is no way to authenticate the public key PK_{id} of the user id because no certificate of PK_{id} is given. Therefore, replacing the public key PK_{id} of the user id is allowed.

Such issues necessitate considering two types of adversaries in CLS, namely, Type 1 and Type 2 adversaries. The Type 1 adversary models outside attacker which is allowed to replace any user’s public key. The Type 2 adversary models a malicious KGC which is allowed to obtain the master secret msk. For each type, the adversary is also given access to the signing oracle for any messages for any identities of its chosen. However, none of Type 1 and Type 2 adversaries are allowed to replace PK_{id} and obtain msk at the same time.

In 2007, Hu et al. [11] defined formal security models and Huang et al. [12, 13] also defined formal security models in which the adversaries can be categorized into normal, strong, super adversaries (ordered based on their attack powers), which are accessed different sign oracles. Because our focus is the Type 1 security of Tian-Huang scheme against strong adversary, we describe the Type 1 security model of CLS against strong adversary, only. The Type 1 security of a CLS against strong adversary is formalized by using the following security game, CL-EUF game between the challenger $\mathcal{C}$ and a Type 1 adversary $\mathcal{A}$.

[CL-EUF-Game 1]

[Initialization] the challenger runs $(\text{msk}, \text{mpk}, \text{params}) \leftarrow \text{setup}(k)$ and sends $(\text{params}, \text{mpk})$ to $\mathcal{A}$. $\mathcal{A}$ can request the following queries adaptively to the challenger $\mathcal{C}$.

CreateUser Query. For the requested identity id $\in \{0, 1\}^*$, if id has already been created, nothing is to be carried out. Otherwise, the oracle runs the algorithms PartialPrivateKeyExtract(id, params, msk), SetSecretValue(id, params), SetPrivateKey(id, x_{id}, ps_{id}, params), and SetPublicKey(id, x_{id}, params) to generate ps_{id}, x_{id}, SK_{id}, and PK_{id}. In both cases, PK_{id} is returned.

RevealSecretValue Query. For the requested identity id $\in \{0, 1\}^*$, the oracle returns the corresponding secret value x_{id}.

ReplacePublicKey Query. For the requested (id, PK_{id}), the oracle replaces the public key PK_{id} of the original user with PK_{id}' and returns the replaced (id, PK_{id}'). The replacement will be recorded.

RevealPartialPrivateKey Query. For the requested id $\in \{0, 1\}^*$, the challenger $\mathcal{C}$ returns the corresponding partial private key ps_{id}.

StrongSign Query. For the requested (id, x_{id}, m_{i}), $\mathcal{C}$ returns the signature $\sigma_i$ such that CLVfy(id, PK_{id}, sig_{id}, m_{i}, params) = 1.

[Forgery] finally, $\mathcal{A}$ outputs sig^* on a message m^* for an identity id*. We say that the adversary $\mathcal{A}$ wins the [CL-EUF-Game 1] if

(1) id^* has never been requested to the RevealPartialPrivateKey oracle,

(2) the pair (id^*, x_{id^*}, m^*) has never been requested to the StrongSign oracle, where x_{id^*} is the corresponding secret of PK_{id^*}.

(3) CLVfy(id^*, PK_{id^*}, sig^*, m^*, params) = 1.

The security of CLS against strong Type 1 adversaries is defined as follows.

Definition 3. For any polynomial-time strong Type 1 adversary, if the probability of the adversary win CL-EUF-Game 1 is negligible, then we say the CLS scheme is existentially unforgeable against strong Type 1 adversaries under adaptive chosen message and chosen identity attacks.

Note that, for the security of CLS, one should consider both of Type 1 and Type 2 adversaries. However, we believe that the description of Type 1 security of CLS is enough to read the ideas of this paper and we omit the description of Type 2 security of CLS in this paper.

2.4. Summary of the Security Proof of Tian-Huang Scheme in [9]. The security proof of Tian-Huang scheme in [9] is given in random oracle model where the hash functions $F$ and $H$ are viewed as random oracles and controlled by the challenger $\mathcal{C}$.

Suppose there is a polynomial-time strong Type 1 adversary $\mathcal{A}$ that requests CreateUser, RevealPartialPrivateKey, RevealSecretValue, ReplacePublicKey, StrongSign, and $F$ and $H$ queries and outputs a forged signature for Tian-Huang scheme with nonnegligible probability. Tian and Huang
proved that the challenger $\mathcal{C}$ can solve the $(q,m_1,4\sigma\sqrt{m} + 2\lambda \sqrt{m_1})$-SIS problem with nonnegligible probability using $\mathcal{A}$. Now, we give a brief review of how the challenger $\mathcal{C}$ solves a given SIS problem by using the successful strong Type 1 adversary; for the full details, see [9]. Suppose that a specific $(q,m_1,4\sigma\sqrt{m} + 2\lambda \sqrt{m_1})$-SIS problem with matrix $A$ is given to $\mathcal{C}$.

First, the challenger $\mathcal{C}$ simulates the security game with the adversary $\mathcal{A}$ for params $= \{A, B, F, H\}$ with a randomly chosen $B \in \mathbb{Z}_p$ and two secure hash functions $H$ and $F$. Even though the challenger $\mathcal{C}$ does not know the corresponding trapdoor $T_{\alpha, \sigma}$, $\mathcal{C}$ can respond to Create-User-Oracle query or Extract-Partial-Private-Key-Oracle query correctly by using the hash function $F$ as a random oracle which is controlled and recorded by $\mathcal{C}$. The challenger $\mathcal{C}$ also records the list $L_H = \{(y_i, \mu_i, h_i)\}$ corresponding to id, of $H$-oracle query as a random oracle. Finally, $\mathcal{A}$ outputs a signature forgery $\text{sig}^* = (h^*, z^*)$ on a message $m^*$ for an identity id$^*$ and $P_{id}$, such that $h^* \neq h'$ and

$$\begin{bmatrix} Az_z \rbrack - F(id^*) - h = \begin{bmatrix} Az_z \rbrack - F(id^*) - h', \end{bmatrix} \text{holds:} \begin{bmatrix} F(id^*) - h^* - h' \end{bmatrix} = 0. \tag{5}$$

where $z^* = \begin{bmatrix} z \rbrack_{1 \times 2}^T$ and $z' = \begin{bmatrix} z \rbrack_{1 \times 2}^T$.

In particular, we see that $Az_z - F(id^*)h^* = Az_z - F(id^*)h'$. By inserting $F(id^*) = AD_{id}$, we have

For $|z| \leq 2\sigma\sqrt{m}$ and $|D_{id} - h|, |D_{id} - h'| \leq s\lambda \sqrt{m}$, with overwhelming probability, we can see that

$$|z^* - z + D_{id} - h - D_{id} - h'| \leq 4\sigma\sqrt{m} + 2\lambda \sqrt{m}. \tag{6}$$

The fact $h^* \neq h'$ and the nonuniqueness of the solution of the SIS problem for $(A, F(id^*))$ yields

$$Az_z - F(id^*) - h' = 0 \tag{7}$$

with probability at least $1/2$. Therefore, $x = z^* - z + D_{id} - h' \neq D_{id} - h'$ is a solution of the given SIS problem to $\mathcal{C}$.

Remark 4. The security proof above is based on the forking lemma assuming the underlying hash function $H$ can be modelled as a random oracle. However, as we will see in the next section, for the strong Type I adversary, the specific hash function in the scheme, which is related but has a different property from the given $H$, is neither one-way nor collision-resistant. This means that the hash function defined from $H$ cannot be modelled as a random oracle. Therefore, we see that there is a critical flaw in their security proof. In fact, we present successful strong Type 1 adversarial algorithms on the scheme in the next section.

3. Main Results

In this section, we discuss the flaws that we have found in the arguments of their security proof against a strong Type I adversary. Then we give two successful strong Type I attack algorithms.

3.1. Analysis of Cryptographic Usage of Hash Functions in the Tian-Huang Scheme. The Tian-Huang scheme uses a collision-resistant hash function $H : \mathbb{Z}_p^2 \rightarrow \{0,1\}^s \rightarrow \{−1,0,1\}^s$ in an essential way in the security proof. In this section, we discuss the usage of the hash function and analyze how the security arguments utilize its cryptographic properties incorrectly.

Lemma 5. Let $P_{id} = BS_{id}$ and $AD_{id} = F(id)$. Consider a signature $\text{sig} = (h, z)$ on a message $m$, where $h = H\left(\begin{bmatrix} Ay \rbrack \end{bmatrix}_{1 \times 2}, m\right)$. Then sig is valid under $(id, P_{id})$ if the following holds:

1. $|z| \leq 2\sigma\sqrt{m}$
2. $z = \begin{bmatrix} D_{id} \rbrack_{1 \times 2} h + \begin{bmatrix} y \rbrack_{1 \times 2}$. \tag{8}

Proof. By (2) we have

$$\begin{bmatrix} Az_z \rbrack - F(id^*) - h' = \begin{bmatrix} Az_z \rbrack - F(id^*) - h', \end{bmatrix} \text{holds:} \begin{bmatrix} F(id) - h^* - h' \end{bmatrix} = 0. \tag{9}$$

The converse is true with overwhelming probability if the hash function $H$ is collision-resistant.

Lemma 6. Let $P_{id} = BS_{id}$ and $AD_{id} = F(id)$. If a signature $\text{sig} = (h, z)$ on a message $m$ is valid under $(id, P_{id})$ where $h = H\left(\begin{bmatrix} Ay \rbrack \end{bmatrix}_{1 \times 2}, m\right)$, then with overwhelming probability we have

$$z = \begin{bmatrix} D_{id} \rbrack_{1 \times 2} h + \begin{bmatrix} y \rbrack_{1 \times 2}. \tag{10}$$

By Lemmas 5 and 6, we see that the validity of a signature $(z, h)$ on the message $m$ under $(id, P_{id})$ for $P_{id} = BS_{id}$ and $AD_{id} = F=id$ is (computationally) equivalent to the following:

1. $|z| \leq 2\sigma\sqrt{m}$
2. $z = \begin{bmatrix} D_{id} \rbrack_{1 \times 2} h + \begin{bmatrix} y \rbrack_{1 \times 2} for h = H\left(\begin{bmatrix} Ay \rbrack \end{bmatrix}_{1 \times 2}, m\right)$.
For any given function \( H \) such that a StrongSign query. In fact, we will use this to design a Huang scheme by requesting ReplacePublicKey queries and available to any strong Type 1 adversary against the Tian-Huang scheme. Theorem 7 can be interpreted as asserting that the functions \( H' \) and \( H'' \) is a one-way and collision-resistant if some additional data is known. Moreover we see that such additional information is always available to any strong Type 1 adversaries against the Tian-Huang scheme. In other words, none of the functions \( H' \) and \( H'' \) is a secure hash function in the view of strong Type 1 adversaries against the Tian-Huang scheme. Tian and Huang claimed that their CLS scheme is provably secure against strong Type 1 adversaries. The arguments of the proof are based on the fact that \( H' \) and \( H'' \) are cryptographically secure hash functions and they can be modelled as a random oracle, which are assumed by the authors from the secure choice of \( H \). By Theorems 7 and 8, however, we see that this is not correct under the strong Type 1 adversarial model and so are their security proofs. In fact, we present two successful strong Type 1 attacks in the sections that follow.

### 3.2. Strong Type 1 Attacks on the Tian-Huang Scheme

We present two attack algorithms on the Tian-Huang scheme. The first attack is successful by considering the hash function in the scheme as

\[
H''(z_1, z_2, P, \text{id}, m) = H\left( \begin{bmatrix} A_{z_1} \\ B_{z_2} \end{bmatrix} - \begin{bmatrix} F(\text{id}) \\ P \end{bmatrix} \right), \tag{18}
\]

The second attack is successful by considering the hash function in the scheme as

\[
h = H''(z_1, z_2, P, \text{id}, m) = H\left( \begin{bmatrix} A_{z_1} \\ B_{z_2} \end{bmatrix} - \begin{bmatrix} F(\text{id}) \\ P \end{bmatrix} \right). \tag{18}
\]

**3.2.1. Attack Algorithm 1.** The idea of the attack is that a strong Type 1 adversary, by requesting ReplacePublicKey queries and a StrongSign query, is always able to obtain the data
needed to compute the preimage of $h = H'(y_1, y_2, m)$ as in Theorem 7.

A strong Type 1 adversary $\mathcal{A}$ proceeds as follows.

Step 1. $\mathcal{A}$ chooses any $S'_{id} \in \mathbb{Z}^{m_1 \times k}$, to be used as a new secret value for id, and sets $P'_{id} = BS'_{id}$.

Step 2. $\mathcal{A}$ submits a query $\text{ReplacePublicKey}(\text{id}, P'_{id})$, so that the public key corresponding to id is now $P'_{id} = BS'_{id}$.

Step 3. $\mathcal{A}$ submits a query $\text{StrongSign}(m, \text{id}, S'_{id})$ to obtain a signature

$$\text{sig} = (z, h) = \left( c + y, H \left( \left[ \begin{array}{c} A_{y_1} \\ By_{y_2} \end{array} \right], m \right) \right),$$

where

(i) $y = \left[ \begin{array}{c} y_1 \\ y_2 \end{array} \right]$; we may assume that $y_2 \neq 0$; that is, there is $1 \leq \ell \leq m_2$ such that the $\ell$th component $y_{2\ell}$ of $y_2$ is nonzero. (So there is $e \in \{1, -1\}$ such that

$$\|e e_{\ell} + y_2\| \leq \|y_2\|,$$

where $e_{\ell}$ is a standard unit vector, with all components zero except for the $\ell$th, which is one.)

(ii) $h \in \{-1, 0, 1\}^k$; we may assume that $h \neq 0$; that is, $h_{i,j} \neq 0$ for some $1 \leq s \leq k$.

(iii) $z = \left[ \begin{array}{c} D_{y_1} \\ 0 \end{array} \right] h + \left[ \begin{array}{c} y_1 \\ y_2 \end{array} \right]$.

Step 4. $\mathcal{A}$ computes $t = \left[ \begin{array}{c} t_1 \\ t_2 \end{array} \right] \leftarrow z - \left[ \begin{array}{c} S'_{id} \\ y_2 \end{array} \right] h$; note that $t = \left[ \begin{array}{c} D_{h+y_{1\ell}} \\ y_{2\ell} \end{array} \right] = \left[ \begin{array}{c} z_{1\ell} \\ z_{2\ell} \end{array} \right]$.

Step 5. $\mathcal{A}$ chooses $S_{id} \in \mathbb{Z}^{m_1 \times k}$ so that $S_{id} h = e e_{\ell} \in \mathbb{Z}^{m_1}$; for instance, one can simply let $S_{id}$ be the matrix whose entries are all zero except for the $(\ell, s)$-entry, which is set to be $e_{h_{s}}$ (recall that $h_{s} \neq 0$ from Step 3). $\mathcal{A}$ computes $P''_{id} = BS''_{id}$ and submits a query $\text{ReplacePublicKey}(\text{id}, P''_{id})$.

Step 6. $\mathcal{A}$ forges a signature $\text{sig}^* = (h^*, z^*)$ by computing

(i) $h^* \leftarrow h$;

(ii) $z^* \leftarrow \left[ \begin{array}{c} t_1 \\ S'_{id} h + t_2 \end{array} \right]$.

Validity of the Forged Signature. The validity of $\text{sig}^*$ as a signature on $m$ under $(\text{id}, P''_{id})$ can be checked using Lemma 5:

(i) Noting $h^* = h = H \left( \left[ \begin{array}{c} A_{y_1} \\ By_{y_2} \end{array} \right], m \right)$, we have

$$z^* = \left[ \begin{array}{c} t_1 \\ S'_{id} h + t_2 \end{array} \right] = \left[ \begin{array}{c} D_{y_1} h + y_1 \\ S_{id} h + y_2 \end{array} \right]$$

$$= \left[ \begin{array}{c} D_{id} \\ S_{id} \end{array} \right] h^* + \left[ \begin{array}{c} y_1 \\ y_2 \end{array} \right].$$

(ii) Since $z^* = \left[ \begin{array}{c} S'_{id} h + t_2 \\ S_{id} h + t_2 \end{array} \right]$, we have

$$\|z^*\|^2 = \|z_{1\ell}\|^2 + \|e e_{\ell} + y_{2\ell}\|^2 \leq \|z_{1\ell}\|^2 + \|y_{2\ell}\|^2 = \|z\|^2$$

$$\leq 2\sigma \sqrt{m}.$$

3.2.2. Attack Algorithm 2. The idea of the attack is that a preimage of $h = H''(z_1, z_2, P, \text{id}, m)$ can be obtained from any (eavesdropped) valid signature and one can compute a second preimage of $h$ as in Theorem 8. The adversary only eavesdrops and makes one ReplacePublicKey query. A strong Type 1 adversary $\mathcal{A}$ proceeds as follows.

Step 1. $\mathcal{A}$ starts with a(n eavesdropped) valid signature $\text{sig} = (h, z)$ on a message $m$ under the public key $P_{id} = BS_{id}$, where

(i) $h = (h_1, \ldots, h_k) = H \left( \left[ \begin{array}{c} A_{y_1} \\ By_{y_2} \end{array} \right], m \right)$;

(ii) $z = \left[ \begin{array}{c} z_{11} \\ z_{12} \\ \vdots \\ z_{1m_2} \end{array} \right] = c + y$;

(iii) $c = \left[ \begin{array}{c} D_{y_1} h + y_1 \\ S'_{id} h + c + y \end{array} \right]$.

We may assume that $z_{2\ell} \neq 0$; that is, there is $1 \leq \ell \leq m_2$ such that $z_{2\ell}$ is nonzero. So there is $e \in \{1, -1\}$ such that

$$\|e e_{\ell} + z_{2\ell}\| \leq \|z_{2\ell}\|,$$

where $e_{\ell}$ is a unit vector, with all zero components except for the $\ell$th, which is one. The adversary $\mathcal{A}$ sets $z_2 = e e_{\ell} + z_{2\ell}$. We may also assume that $h \neq 0$, say $h_{s} \neq 0$ for some $1 \leq s \leq k$.

Step 2. $\mathcal{A}$ sets $S'_{id} \in \mathbb{Z}^{m_1 \times k}$ to be the matrix whose entries are all zero except for the $(\ell, s)$-entry, which is $e_{h_{s}}$. $\mathcal{A}$ computes

(i) $P_{id} \leftarrow P_{id} + BS_{id}$;

(ii) $z^* = \left[ \begin{array}{c} D_{y_1} \\ S'_{id} h + c + y \end{array} \right]$.

Note that $z^* = \left[ \begin{array}{c} S'_{id} h + c + y \end{array} \right] = \left[ \begin{array}{c} D_{id} \\ S_{id} \end{array} \right] h + z \neq z$ and $P_{id} = BS_{id}$ with $S_{id}$ unknown to $\mathcal{A}$.

Step 3. $\mathcal{A}$ submits a query $\text{ReplacePublicKey}(\text{id}, P'_{id})$; note that $\mathcal{A}$ does not know the secret value $S'_{id} = S_{id} + S'_{id}$ corresponding to $P'_{id}$.

Step 4. $\mathcal{A}$ returns forged $\text{sig}^* = (h, z^*)$ as a signature on $m$ under $(\text{id}, P'_{id})$.

Validity of the Forged Signature. The validity of $\text{sig}^*$ as a signature on $m$ under $(\text{id}, P'_{id})$ can be checked using Lemma 5:

(i) Noting $h^* = h = H \left( \left[ \begin{array}{c} A_{y_1} \\ By_{y_2} \end{array} \right], m \right)$, we have

$$z^* = \left[ \begin{array}{c} 0 \\ S'_{id} h + z \end{array} \right] = \left[ \begin{array}{c} 0 \\ S_{id} \end{array} \right] h + z \left[ \begin{array}{c} y_1 \\ y_2 \end{array} \right] = \left[ \begin{array}{c} D_{id} \\ S_{id} + S'_{id} \end{array} \right] h + \left[ \begin{array}{c} y_1 \\ y_2 \end{array} \right].$$
(ii) Since $z^* = \left[ \frac{z_1}{S_{hv}, z_2} \right] = \left[ \frac{z_1}{e_r, z_2} \right]$, we have
\[
\|z^*\|^2 = \|z_1\|^2 + \|e_r^c + z_2\|^2 \leq \|z_1\|^2 + \|z_2\|^2 = \|z\|^2 \leq 2\sigma \sqrt{m}.
\]

(25)

4. Conclusion

In this paper, we showed that the hash function used in Tian-Huang's scheme is not a secure hash function in the presence of a strong Type 1 adversary even though the function is defined from a cryptographically secure hash function. Such weakness of the hash function admits successful strong Type 1 attacks on their scheme. The security proof of the Tian-Huang scheme was done under the assumption that the hash function is a random oracle, which requires cryptographically security properties. It seems that to improve security argument one needs to make more careful use of the hash function in the simulation of the security game.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Hyang-Sook Lee was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (no. 2015R1A2A1A15054564). Juhee Lee was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (no. NRF-2012R1A1A3015819).

References
