Hindawi
Security and Communication Networks
Volume 2017, Article ID 3413567, 7 pages
https://doi.org/10.1155/2017/3413567

Research Article

Security Analysis of a Certificateless Signature from Lattices

Seunghwan Chang,[1] Hyang-Sook Lee,[2] Juhee Lee,[1] and Seongan Lim[1]

[1]*Institute of Mathematical Sciences, Ewha Womans University, Seoul 120-750, Republic of Korea*
[2]*Department 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. We point out flaws in the security arguments and present attack algorithms that are successful in the strong Type 1 adversarial model using the weak properties of the hash function.

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 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a positive real number γ, the (q, m, γ)-SIS problem for $\mathbf{A}$ is to find a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\mathbf{Av} = 0 \pmod q$ and $\|\mathbf{v}\| \le \gamma$.

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 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a vector $\mathbf{u} \in \mathbb{Z}^n$, and a positive real number γ', the (q, m, γ')-ISIS problem for $(\mathbf{A}, \mathbf{u})$ is to find a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\mathbf{Av} = \mathbf{u} \pmod q$ and $\|\mathbf{v}\| \le \gamma'$.

For any polynomial-bounded m, γ and for any prime $q \ge \gamma \cdot \omega(\sqrt{n \log n})$, 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 *TrapGen* and *SampleMat* are building blocks of Tian-Huang scheme. We skip the details of each algorithms in this paper.

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

(ii) For $(\mathbf{A}, \mathbf{T}_A) \leftarrow TrapGen(q, n, m, \gamma)$, $(\mathbf{D} \in \mathbb{Z}_q^{n\times k}) \leftarrow SampleMat(\mathbf{A}, \mathbf{T}_A, \gamma', u)$ if and only if each of the column vectors of $\mathbf{D}$ is a solution of (q, m, γ')-ISIS problem for $(\mathbf{A}, \mathbf{u})$.

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

The scheme consists of the following seven algorithms:

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

(a) computes $(\mathbf{A}, \mathbf{T}_A) \leftarrow TrapGen(q, n, m_1)$,

(b) chooses a random matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m_2}$,

(c) chooses two secure hash functions

$$F : \{0,1\}^* \longrightarrow \mathbb{Z}_q^{n\times k},$$
$$H : \mathbb{Z}_q^{2n} \times \{0,1\}^* \longrightarrow \left\{\mathbf{v} \in \{-1,0,1\}^k : \|\mathbf{v}\|_1 \le \lambda\right\}, \quad (1)$$

(d) outputs the master secret $msk = \mathbf{T}_A$ and the public parameters $params = \{\mathbf{A}, \mathbf{B}, F, H\}$.

(ii) PartialPrivateKeyExtract(id, msk, $params$): on input $(\mathsf{id}, msk = \mathbf{T}_A, params)$, the KGC obtains $\mathbf{D}_{\mathsf{id}} \in \mathbb{Z}^{m_1\times k} \leftarrow SampleMat(\mathbf{A}, \mathbf{T}_A, s, F(\mathsf{id}))$ and sends it to the user with id. Upon receiving $\mathbf{D}_{\mathsf{id}}$, the user with id checks the correctness of $\mathbf{D}_{\mathsf{id}}$ by verifying if $\mathbf{AD}_{\mathsf{id}} = F(\mathsf{id})$ and $\|\mathbf{D}_{\mathsf{id}}\| \le s\sqrt{m_1}$. If so, the user sets his partial private key as $psk_{\mathsf{id}} = \mathbf{D}_{\mathsf{id}}$.

(iii) SetSecretValue(id, $params$): on input $(\mathsf{id}, params)$, the user with id selects a random matrix $\mathbf{S}_{\mathsf{id}} \in \mathbb{Z}^{m_2\times k}$ satisfying $\|\mathbf{S}_{\mathsf{id}}\|_\infty \le b$ and outputs his secret value $x_{\mathsf{id}} = \mathbf{S}_{\mathsf{id}}$.

(iv) SetPrivateKey(id, x_{id}, psk_{id}, $params$): on input $(\mathsf{id}, x_{\mathsf{id}} = \mathbf{S}_{\mathsf{id}}, psk_{\mathsf{id}} = \mathbf{D}_{\mathsf{id}}, params)$, the user with id outputs his full private key $\mathrm{SK}_{\mathsf{id}} = (psk_{\mathsf{id}}, x_{\mathsf{id}}) = (\mathbf{D}_{\mathsf{id}}, \mathbf{S}_{\mathsf{id}}) \in \mathbb{Z}^{m_1\times k} \times \mathbb{Z}^{m_2\times k}$.

(v) SetPublicKey(id, $\mathrm{SK}_{\mathsf{id}}$, $params$): on input $(\mathsf{id}, \mathrm{SK}_{\mathsf{id}} = (\mathbf{D}_{\mathsf{id}}, \mathbf{S}_{\mathsf{id}}), params)$, the user with id outputs his public key $\mathrm{PK}_{\mathsf{id}} = \mathbf{BS}_{\mathsf{id}}$.

(vi) CLSign(id, $\mathrm{SK}_{\mathsf{id}}$, m, $params$): on input $(\mathsf{id}, \mathrm{SK}_{\mathsf{id}} = (\mathbf{D}_{\mathsf{id}}, \mathbf{S}_{\mathsf{id}}), \mathsf{m}, params)$, the user with id

(a) selects $\mathbf{y}_1 \leftarrow D_\sigma^{m_1}$ and $\mathbf{y}_2 \leftarrow D_\sigma^{m_2}$ and sets $\mathbf{y} = \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \in \mathbb{Z}^m$.

(b) sets $\mathbf{h} = H\left(\begin{bmatrix}\mathbf{Ay}_1\\ \mathbf{By}_2\end{bmatrix}, \mathsf{m}\right) \in \{-1,0,1\}^k$ and $\mathbf{c} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}_{\mathsf{id}}\end{bmatrix}\mathbf{h} \in \mathbb{Z}^m$.

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

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

(vii) CLVfy(id, $\mathrm{PK}_{\mathsf{id}}$, σ, m, $params$): on input $(\mathsf{id}, \mathrm{PK}_{\mathsf{id}} = \mathbf{BS}_{\mathsf{id}}, \mathsf{sig} = (\mathbf{h}, \mathbf{z}), \mathsf{m}, params)$, the algorithm outputs 1 only if $\|\mathbf{z}\| \le 2\sigma\sqrt{m}$, $\mathbf{h} = H\left(\begin{bmatrix}\mathbf{Az}_1\\ \mathbf{Bz}_2\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ \mathrm{PK}_{\mathsf{id}}\end{bmatrix}\mathbf{h}, \mathsf{m}\right)$, where $\mathbf{z} = \begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix}$.

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 $(\mathbf{h}, \mathbf{z})$ on m.

$$\begin{bmatrix} \mathbf{Az}_1 \\ \mathbf{Bz}_2 \end{bmatrix} - \begin{bmatrix} F(\mathsf{id}) \\ \mathrm{PK}_{\mathsf{id}} \end{bmatrix} \mathbf{h} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \mathbf{z} - \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{\mathsf{id}} \\ \mathbf{S}_{\mathsf{id}} \end{bmatrix} \mathbf{h} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} (\mathbf{z} - \mathbf{c}) = \begin{bmatrix} \mathbf{Ay}_1 \\ \mathbf{By}_2 \end{bmatrix}. \tag{2}$$

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 $\mathrm{PK}_{\mathsf{id}}$ of the user id because no certificate of $\mathrm{PK}_{\mathsf{id}}$ is given. Therefore, replacing the public key $\mathrm{PK}_{\mathsf{id}}$ 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 $\mathrm{PK}_{\mathsf{id}}$ 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 in this paper 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 $\mathscr{C}$ and a Type 1 adversary $\mathscr{A}$.

[CL-EUF-Game 1]

[Initialization] the challenger runs $(msk, mpk, params) \leftarrow \mathrm{setup}(k)$ and sends $(params, mpk)$ to $\mathscr{A}$.

[Queries] $\mathscr{A}$ can request the following queries adaptively to the challenger $\mathscr{C}$.

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

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

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

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

StrongSign Query. For the requested $(\mathsf{id}, x_{\mathsf{id}}, m_i)$, $\mathscr{C}$ returns the signature σ_i such that

$$\mathrm{CLVfy}\left(\mathsf{id}, \mathrm{PK}_{\mathsf{id}}, \mathsf{sig}_i, m_i, params\right) = 1. \tag{3}$$

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

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

(2) the pair $(\mathsf{id}^*, x_{\mathsf{id}^*}, m^*)$ has never been requested to the StrongSign oracle, where x_{id^*} is the corresponding secret of $\mathrm{PK}_{\mathsf{id}^*}$,

(3) $\mathrm{CLVfy}(\mathsf{id}^*, \mathrm{PK}_{\mathsf{id}^*}, \mathsf{sig}^*, m^*, params) = 1$.

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

Definition 3. For any polynomial-time strong Type I 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 $\mathscr{C}$.

Suppose there is a polynomial-time strong Type 1 adversary $\mathscr{A}$ 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 $\mathscr{C}$ can solve the $(q, m_1, 4\sigma\sqrt{m} + 2s\lambda\sqrt{m_1})$-SIS problem with nonnegligible probability using $\mathscr{A}$. Now, we give a brief review of how the challenger $\mathscr{C}$ solves a given SIS problem by using the successful strong Type 1 adversary; for the full details, see [9]. Suppose that a specific $(q, m_1, 4\sigma\sqrt{m} + 2s\lambda\sqrt{m_1})$-SIS problem with matrix $\mathbf{A}$ is given to $\mathscr{C}$.

First, the challenger $\mathscr{C}$ simulates the security game with the adversary $\mathscr{A}$ for $params = \{\mathbf{A}, \mathbf{B}, F, H\}$ with a randomly chosen $\mathbf{B} \in \mathbb{Z}_q^{n \times m_2}$ and two secure hash functions H and F. Even though the challenger $\mathscr{C}$ does not know the corresponding trapdoor T_A, $\mathscr{C}$ 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 $\mathscr{C}$. The challenger $\mathscr{C}$ also records the list $L_H = \{(\mathbf{v}_i, \mu_i, \mathbf{h}_i)\}$ corresponding to id_i of H-oracle query as a random oracle. Finally, $\mathscr{A}$ outputs a signature forgery $\mathsf{sig}^* = (\mathbf{h}^*, \mathbf{z}^*)$ on a message m^* for an identity id^* and $\mathbf{P}_{\mathsf{id}^*}$ with nonnegligible probability.

To solve the given SIS problem for the matrix $\mathbf{A}$, the challenger $\mathscr{C}$ reruns the adversary $\mathscr{A}$ with the same random tape but different outputting sequence of H-oracle. The general forking lemma assures that $\mathscr{A}$ will output a new forgery sig'^* on a message m^* for an identity id^* and $\mathbf{P}_{\mathsf{id}^*}$ such that $\mathbf{h}^* \neq \mathbf{h}'$ and

$$\begin{bmatrix} \mathbf{A}\mathbf{z}_1^* \\ \mathbf{B}\mathbf{z}_2^* \end{bmatrix} - \begin{bmatrix} F(\mathsf{id}^*) \\ P_{\mathsf{id}^*} \end{bmatrix} \mathbf{h}^* = \begin{bmatrix} \mathbf{A}\mathbf{z}_1' \\ \mathbf{B}\mathbf{z}_2' \end{bmatrix} - \begin{bmatrix} F(\mathsf{id}^*) \\ P_{\mathsf{id}^*} \end{bmatrix} \mathbf{h}', \quad (4)$$

where $\mathbf{z}^* = \begin{bmatrix} \mathbf{z}_1^* \\ \mathbf{z}_2^* \end{bmatrix}$ and $\mathbf{z}' = \begin{bmatrix} \mathbf{z}_1' \\ \mathbf{z}_2' \end{bmatrix}$.

In particular, we see that $\mathbf{A}\mathbf{z}_1^* - F(\mathsf{id}^*)\mathbf{h}^* = \mathbf{A}\mathbf{z}_1' - F(\mathsf{id}^*)\mathbf{h}'$. By inserting $F(\mathsf{id}^*) = \mathbf{A}\mathbf{D}_{\mathsf{id}^*}$, we have

$$\mathbf{A}\left(\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{D}_{\mathsf{id}^*}\left(\mathbf{h}' - \mathbf{h}^*\right)\right) = 0. \quad (5)$$

Since $\|\mathbf{z}_1^*\|, \|\mathbf{z}_1'\| \leq 2\sigma\sqrt{m}$ and $\|\mathbf{D}_{\mathsf{id}^*}\mathbf{h}'\|, \|\mathbf{D}_{\mathsf{id}^*}\mathbf{h}^*\| \leq s\lambda\sqrt{m_1}$ with overwhelming probability, we can see that

$$\left\|\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{D}_{\mathsf{id}^*}\mathbf{h}' - \mathbf{D}_{\mathsf{id}^*}\mathbf{h}^*\right\| \leq 4\sigma\sqrt{m} + 2s\lambda\sqrt{m_1}. \quad (6)$$

The fact $\mathbf{h}^* \neq \mathbf{h}'$ and the nonuniqueness of the solution of ISIS problem for $(\mathbf{A}, F(\mathsf{id}^*))$ yields

$$\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{D}_{\mathsf{id}^*}\mathbf{h}' - \mathbf{D}_{\mathsf{id}^*}\mathbf{h}^* \neq 0 \quad (7)$$

with probability at least 1/2. Therefore, $\mathbf{x} = \mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{D}_{\mathsf{id}^*}\mathbf{h}' - \mathbf{D}_{\mathsf{id}^*}\mathbf{h}^*$ is a solution of the given SIS problem to $\mathscr{C}$.

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 1 adversary. Then we give two successful strong Type 1 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 $H : \mathbb{Z}_q^{2n} \times \{0,1\}^* \to \{-1, 0, 1\}^k$ 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* $\mathbf{P}_{\mathsf{id}} = \mathbf{BS}_{\mathsf{id}}$ *and* $\mathbf{AD}_{\mathsf{id}} = F(\mathsf{id})$. *Consider a signature* $\mathsf{sig} = (\mathbf{h}, \mathbf{z})$ *on a message* m, *where* $\mathbf{h} = H\left(\begin{bmatrix} \mathbf{A}\mathbf{y}_1 \\ \mathbf{B}\mathbf{y}_2 \end{bmatrix}, \mathsf{m}\right)$. *Then* sig *is valid under* $(\mathsf{id}, \mathbf{P}_{\mathsf{id}})$ *if the following holds:*

(1) $\|\mathbf{z}\| \leq 2\sigma\sqrt{\mathsf{m}}$.

(2) $\mathbf{z} = \begin{bmatrix} \mathbf{D}_{\mathsf{id}} \\ \mathbf{S}_{\mathsf{id}} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$.

Proof. By (2) we have

$$\begin{aligned} \begin{bmatrix} \mathbf{A}\mathbf{z}_1 \\ \mathbf{B}\mathbf{z}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \left(\begin{bmatrix} \mathbf{D}_{\mathsf{id}} \\ \mathbf{S}_{\mathsf{id}} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \right) \\ &= \begin{bmatrix} \mathbf{A}\mathbf{D}_{\mathsf{id}} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}\mathbf{S}_{\mathsf{id}} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{A}\mathbf{y}_1 \\ \mathbf{B}\mathbf{y}_2 \end{bmatrix} \\ &= \begin{bmatrix} F(\mathsf{id}) \\ \mathbf{P}_{\mathsf{id}} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{A}\mathbf{y}_1 \\ \mathbf{B}\mathbf{y}_2 \end{bmatrix}, \end{aligned} \quad (8)$$

and so

$$\mathbf{h} = H\left(\begin{bmatrix} \mathbf{A}\mathbf{y}_1 \\ \mathbf{B}\mathbf{y}_2 \end{bmatrix}, \mathsf{m}\right) = H\left(\begin{bmatrix} \mathbf{A}\mathbf{z}_1 \\ \mathbf{B}\mathbf{z}_2 \end{bmatrix} - \begin{bmatrix} F(\mathsf{id}) \\ \mathbf{P}_{\mathsf{id}} \end{bmatrix} \mathbf{h}, \mathsf{m}\right). \quad (9)$$

□

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

Lemma 6. *Let* $\mathbf{P}_{\mathsf{id}} = \mathbf{BS}_{\mathsf{id}}$ *and* $\mathbf{AD}_{\mathsf{id}} = \mathbf{F}(\mathsf{id})$. *If a signature* $\mathsf{sig} = (\mathbf{h}, \mathbf{z})$ *on a message* m *is valid under* $(\mathsf{id}, \mathbf{P}_{\mathsf{id}})$ *where* $\mathbf{h} = \mathbf{H}\left(\begin{bmatrix} \mathbf{A}\mathbf{y}_1 \\ \mathbf{B}\mathbf{y}_2 \end{bmatrix}, \mathsf{m}\right)$, *then with overwhelming probability we have*

$$\mathbf{z} = \begin{bmatrix} \mathbf{D}_{\mathsf{id}} \\ \mathbf{S}_{\mathsf{id}} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}. \quad (10)$$

By Lemmas 5 and 6, we see that the validity of a signature $(\mathbf{z}, \mathbf{h})$ on the message m under $(\mathsf{id}, \mathbf{P}_{\mathsf{id}})$ for $\mathbf{P}_{\mathsf{id}} = \mathbf{BS}_{\mathsf{id}}$ and $\mathbf{AD}_{\mathsf{id}} = \mathbf{F}(\mathsf{id})$ is (computationally) equivalent to the following:

(1) $\|\mathbf{z}\| \leq 2\sigma\sqrt{\mathsf{m}}$,

(2) $\mathbf{z} = \begin{bmatrix} \mathbf{D}_{\mathsf{id}} \\ \mathbf{S}_{\mathsf{id}} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$ for $\mathbf{h} = \mathbf{H}\left(\begin{bmatrix} \mathbf{A}\mathbf{y}_1 \\ \mathbf{B}\mathbf{y}_2 \end{bmatrix}, \mathsf{m}\right)$.

Now we analyze some cryptographic properties concerning

$$\begin{aligned}\mathbf{h} &= H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right)\\ &= H\left(\begin{bmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\mathbf{z}_2\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ P\end{bmatrix}\mathbf{h}, \mathsf{m}\right),\end{aligned} \tag{11}$$

where $H : \mathbb{Z}_q^{2n} \times \{0,1\}^* \to \{-1,0,1\}^k$ is a collision-resistant hash function.

Theorem 7. *Suppose that we have functions*

$$H : \mathbb{Z}_q^{2n} \times \{0,1\}^* \to \{-1,0,1\}^k,$$
$$H' : \mathbb{Z}^{m_1} \times \mathbb{Z}^{m_2} \times \{0,1\}^* \to \{-1,0,1\}^k,$$

such that

$$H'(\mathbf{y}_1, \mathbf{y}_2, \mathsf{m}) = H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right). \tag{12}$$

For any given $\mathbf{y}_1, \mathsf{m}, \mathbf{h} = H'(\mathbf{y}_1, \mathbf{y}_2, \mathsf{m})$*, one can efficiently compute* $\mathbf{y}_2$ *if the following data are known:*

$$\left(\mathbf{S}_{\mathsf{id}}, \mathbf{z} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}_{\mathsf{id}}\end{bmatrix}\mathbf{h} + \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix}\right). \tag{13}$$

Proof. Given $\mathbf{h} = H'(\mathbf{y}_1, \mathbf{y}_2, \mathsf{m})$ together with $(\mathbf{S}_{\mathsf{id}}, \mathbf{z} = \left[\begin{smallmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}_{\mathsf{id}}\end{smallmatrix}\right]\mathbf{h} + \left[\begin{smallmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{smallmatrix}\right])$, one can recover $\mathbf{y}_2$ by computing and taking the second component of

$$\mathbf{z} - \begin{bmatrix}\mathbf{0}\\ \mathbf{S}_{\mathsf{id}}\end{bmatrix}\mathbf{h} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\mathbf{h} + \mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix}. \tag{14}$$

□

Theorem 7 can be interpreted as asserting that the function H' cannot acquire one-wayness in the presence of the (additional) data $\left(\mathbf{S}_{\mathsf{id}}, \mathbf{z} = \left[\begin{smallmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}_{\mathsf{id}}\end{smallmatrix}\right]\mathbf{h} + \left[\begin{smallmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{smallmatrix}\right]\right)$ for $\mathbf{h} = H'(\mathbf{y}_1, *, \mathsf{m})$, even though H' is closely related to a secure hash function H. We note that this additional data is always available to any strong Type 1 adversary against the Tian-Huang scheme by requesting ReplacePublicKey queries and a StrongSign query. In fact, we will use this to design a successful strong Type 1 attack on the Tian-Huang scheme in Section 3.2.1.

Theorem 8. *Suppose we have functions*

$$H : \mathbb{Z}_q^{2n} \times \{0,1\}^* \to \{-1,0,1\}^k,$$
$$H'' : \mathbb{Z}_q^{m_1} \times \mathbb{Z}_q^{m_2} \times \mathbb{Z}_q^{n\times \mathrm{k}} \times \{0,1\}^* \to \{-1,0,1\}^k,$$

such that

$$H''(\mathbf{z}_1, \mathbf{z}_2, \mathbf{P}, \mathsf{id}, \mathsf{m}) = H\left(\begin{bmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\mathbf{z}_2\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ \mathbf{P}\end{bmatrix}\mathbf{h}, \mathsf{m}\right). \tag{15}$$

For any given preimage $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{P}, \mathsf{id}, \mathsf{m})$ *of* $\mathbf{h} = H''(*)$*, one can efficiently compute a second preimage of* $\mathbf{h}$.

Proof. Suppose we are given $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{P}, \mathsf{id}, \mathsf{m})$ such that $H''(\mathbf{z}_1, \mathbf{z}_2, \mathbf{P}, \mathsf{id}, \mathsf{m}) = \mathbf{h}$. Choose a $\mathbf{S}' \in \mathbb{Z}^{m_2 \times k}$ such that $\mathbf{S}'\mathbf{h} \neq 0$ and $\mathbf{B}\mathbf{S}' \neq 0$. Compute $\mathbf{z}_2' \coloneqq \mathbf{z}_2 + \mathbf{S}'\mathbf{h}$ and $\mathbf{P}' \coloneqq \mathbf{P} + \mathbf{B}\mathbf{S}'$. Then

$$\begin{aligned}H''\left(\mathbf{z}_1, \mathbf{z}_2', \mathbf{P}', \mathsf{id}, \mathsf{m}\right) &= H\left(\begin{bmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\mathbf{z}_2'\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ \mathbf{P}'\end{bmatrix}\mathbf{h}, \mathsf{m}\right)\\ &= H\left(\begin{bmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\left(\mathbf{z}_2 + \mathbf{S}'\mathbf{h}\right)\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ \mathbf{P} + \mathbf{B}\mathbf{S}'\end{bmatrix}\mathbf{h}, \mathsf{m}\right)\\ &= H\left(\begin{bmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\mathbf{z}_2\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ \mathbf{P}\end{bmatrix}\mathbf{h}, \mathsf{m}\right) = \mathbf{h}.\end{aligned} \tag{16}$$

□

By Theorem 8, the function H'' cannot be collision-resistant even if H is. In Section 3.2.2 we show how to utilize a second preimage of $\mathbf{h}$ to design a successful strong Type 1 attack on the Tian-Huang scheme.

Theorems 7 and 8 show that the functions $\mathbf{h} = H\left(\left[\begin{smallmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{smallmatrix}\right], \mathsf{m}\right)$ and $\mathbf{h} = H\left(\left[\begin{smallmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\mathbf{z}_2\end{smallmatrix}\right] - \left[\begin{smallmatrix}F(\mathsf{id})\\ \mathbf{P}\end{smallmatrix}\right]\mathbf{h}, \mathsf{m}\right)$ cannot be a secure hash function even if the underlying function 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

$$\mathbf{h} = H'(\mathbf{y}_1, \mathbf{y}_2, \mathsf{m}) = H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right). \tag{17}$$

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

$$\begin{aligned}\mathbf{h} &= H''(\mathbf{z}_1, \mathbf{z}_2, \mathbf{P}, \mathsf{id}, \mathbf{m})\\ &= H\left(\begin{bmatrix}\mathbf{A}\mathbf{z}_1\\ \mathbf{B}\mathbf{z}_2\end{bmatrix} - \begin{bmatrix}F(\mathsf{id})\\ \mathbf{P}\end{bmatrix}\mathbf{h}, \mathsf{m}\right).\end{aligned} \tag{18}$$

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 $\mathbf{h} = H'(\mathbf{y}_1, \mathbf{y}_2, \mathsf{m})$ as in Theorem 7.

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

Step 1. $\mathscr{A}$ choose any $\mathbf{S}'_{\mathsf{id}} \in \mathbb{Z}^{m_2 \times k}$, to be used as a new secret value for id, and sets $\mathbf{P}'_{\mathsf{id}} := \mathbf{B}\mathbf{S}'_{\mathsf{id}}$.

Step 2. $\mathscr{A}$ submits a query ReplacePublicKey($\mathsf{id}, \mathbf{P}'_{\mathsf{id}}$), so that the public key corresponding to id is now $\mathbf{P}'_{\mathsf{id}} = \mathbf{B}\mathbf{S}'_{\mathsf{id}}$.

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

$$\mathsf{sig} = (\mathbf{z}, \mathbf{h}) = \left(\mathbf{c} + \mathbf{y}, H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right)\right), \tag{19}$$

where

(i) $\mathbf{y} = \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix}$; we may assume that $\mathbf{y}_2 \neq \mathbf{0}$; that is, there is $1 \leq \ell \leq m_2$ such that the ℓth component $y_{2\ell}$ of $\mathbf{y}_2$ is nonzero. (So there is $\varepsilon \in \{1, -1\}$ such that

$$\left\|\varepsilon\mathbf{e}_\ell + \mathbf{y}_2\right\| \leq \left\|\mathbf{y}_2\right\|, \tag{20}$$

where $\mathbf{e}_\ell$ is a standard unit vector, with all components zero except for the ℓth, which is one.)

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

(iii) $\mathbf{z} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}'_{\mathsf{id}}\end{bmatrix}\mathbf{h} + \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix}$.

Step 4. $\mathscr{A}$ computes $\mathbf{t} := \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix} \leftarrow \mathbf{z} - \begin{bmatrix}\mathbf{0}\\ \mathbf{S}'_{\mathsf{id}}\end{bmatrix}\mathbf{h}$; note that $\mathbf{t} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\mathbf{h} + \mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} = \begin{bmatrix}\mathbf{z}_1\\ \mathbf{y}_2\end{bmatrix}$.

Step 5. $\mathscr{A}$ chooses $\mathbf{S}^*_{\mathsf{id}} \in \mathbb{Z}^{m_2 \times k}$ so that $\mathbf{S}^*_{\mathsf{id}}\mathbf{h} = \varepsilon\mathbf{e}_\ell \in \mathbb{Z}^{m_2}$; for instance, one can simply let $\mathbf{S}^*_{\mathsf{id}}$ to be the matrix whose entries are all zero except for the (ℓ, s)-entry, which is set to be εh_s (recall that $h_s \neq 0$ from Step 3). $\mathscr{A}$ computes $\mathbf{P}^*_{\mathsf{id}} = \mathbf{B}\mathbf{S}^*_{\mathsf{id}}$ and submits a query ReplacePublicKey($\mathsf{id}, \mathbf{P}^*_{\mathsf{id}}$).

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

(i) $\mathbf{h}^* \leftarrow \mathbf{h}$;

(ii) $\mathbf{z}^* \leftarrow \begin{bmatrix}\mathbf{t}_1\\ \mathbf{S}^*_{\mathsf{id}}\mathbf{h} + \mathbf{t}_2\end{bmatrix}$.

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

(i) Noting $\mathbf{h}^* = \mathbf{h} = H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right)$, we have

$$\begin{aligned}\mathbf{z}^* &= \begin{bmatrix}\mathbf{t}_1\\ \mathbf{S}^*_{\mathsf{id}}\mathbf{h} + \mathbf{t}_2\end{bmatrix} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\mathbf{h} + \mathbf{y}_1\\ \mathbf{S}^*_{\mathsf{id}}\mathbf{h} + \mathbf{y}_2\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}^*_{\mathsf{id}}\end{bmatrix}\mathbf{h}^* + \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix}.\end{aligned} \tag{21}$$

(ii) Since $\mathbf{z}^* = \begin{bmatrix}\mathbf{z}_1\\ \mathbf{S}^*_{\mathsf{id}}\mathbf{h} + \mathbf{y}_2\end{bmatrix} = \begin{bmatrix}\mathbf{z}_1\\ \varepsilon\mathbf{e}_\ell + \mathbf{y}_2\end{bmatrix}$, we have

$$\begin{aligned}\left\|\mathbf{z}^*\right\|^2 &= \left\|\mathbf{z}_1\right\|^2 + \left\|\varepsilon\mathbf{e}_\ell + \mathbf{y}_2\right\|^2 \leq \left\|\mathbf{z}_1\right\|^2 + \left\|\mathbf{y}_2\right\|^2 = \left\|\mathbf{z}\right\|^2\\ &\leq 2\sigma\sqrt{\mathsf{m}}.\end{aligned} \tag{22}$$

3.2.2. Attack Algorithm 2. The idea of the attack is that a preimage of $\mathbf{h} = H''(\mathbf{z}_1, \mathbf{z}_2, \mathbf{P}, \mathsf{id}, \mathbf{m})$ can be obtained from any (eavesdropped) valid signature and one can compute a second preimage of $\mathbf{h}$ as in Theorem 8. The adversary only eavesdrops and makes one ReplacePublicKey query. A strong Type 1 adversary $\mathscr{A}$ proceeds as follows.

Step 1. $\mathscr{A}$ starts with a(n eavesdropped) valid signature $\mathsf{sig} = (\mathbf{h}, \mathbf{z})$ on a message m under the public key $\mathbf{P}_{\mathsf{id}} = \mathbf{B}\mathbf{S}_{\mathsf{id}}$, where

(i) $\mathbf{h} = (h_1, \ldots, h_k) = H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right)$;

(ii) $\mathbf{z} = \begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix} = \begin{bmatrix}z_{11}\\ \vdots\\ z_{1m_1}\\ z_{21}\\ \vdots\\ z_{2m_2}\end{bmatrix} = \mathbf{c} + \mathbf{y}$;

(iii) $\mathbf{c} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}_{\mathsf{id}}\end{bmatrix}\mathbf{h}$.

We may assume that $\mathbf{z}_2 \neq \mathbf{0}$; that is, there is $1 \leq \ell \leq m_2$ such that $z_{2\ell}$ is nonzero. So there is $\varepsilon \in \{1, -1\}$ such that

$$\left\|\varepsilon\mathbf{e}_\ell + \mathbf{z}_2\right\| \leq \left\|\mathbf{z}_2\right\|, \tag{23}$$

where $\mathbf{e}_\ell$ is a unit vector, with all zero components except for the ℓth, which is one. The adversary $\mathscr{A}$ sets $\mathbf{z}'_2 = \varepsilon\mathbf{e}_\ell + \mathbf{z}_2$. We may also assume that $\mathbf{h} \neq \mathbf{0}$, say $h_s \neq 0$ for some $1 \leq s \leq k$.

Step 2. $\mathscr{A}$ sets $\mathbf{S}'_{\mathsf{id}} \in \mathbb{Z}^{m_2 \times k}$ to be the matrix whose entries are all zeros except for the (ℓ, s)-entry, which is εh_s. $\mathscr{A}$ computes

(i) $\mathbf{P}^*_{\mathsf{id}} \leftarrow \mathbf{P}_{\mathsf{id}} + \mathbf{B}\mathbf{S}'_{\mathsf{id}}$;

(ii) $\mathbf{z}^* = \begin{bmatrix}\mathbf{z}^*_1\\ \mathbf{z}^*_2\end{bmatrix} \leftarrow \begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}'_2\end{bmatrix}$.

Note that $\mathbf{z}^* = \begin{bmatrix}\mathbf{0}\\ \mathbf{S}'_{\mathsf{id}}\end{bmatrix}\mathbf{h} + \mathbf{z} \neq \mathbf{z}$ and $\mathbf{P}_{\mathsf{id}} = \mathbf{B}\mathbf{S}_{\mathsf{id}}$ with $\mathbf{S}_{\mathsf{id}}$ unknown to $\mathscr{A}$.

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

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

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

(i) Noting that $\mathbf{h} = H\left(\begin{bmatrix}\mathbf{A}\mathbf{y}_1\\ \mathbf{B}\mathbf{y}_2\end{bmatrix}, \mathsf{m}\right)$, we have

$$\begin{aligned}\mathbf{z}^* &= \begin{bmatrix}\mathbf{0}\\ \mathbf{S}'_{\mathsf{id}}\end{bmatrix}\mathbf{h} + \mathbf{z} = \begin{bmatrix}\mathbf{0}\\ \mathbf{S}'_{\mathsf{id}}\end{bmatrix}\mathbf{h} + \mathbf{c} + \mathbf{y}\\ &= \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}'_{\mathsf{id}} + \mathbf{S}_{\mathsf{id}}\end{bmatrix}\mathbf{h} + \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} = \begin{bmatrix}\mathbf{D}_{\mathsf{id}}\\ \mathbf{S}^*_{\mathsf{id}}\end{bmatrix}h + \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix}.\end{aligned} \tag{24}$$

(ii) Since $\mathbf{z}^* = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{S}'_{\mathrm{id}}\mathbf{h}+\mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1 \\ \varepsilon\mathbf{e}_\ell+\mathbf{z}_2 \end{bmatrix}$, we have

$$\begin{aligned} \|\mathbf{z}^*\|^2 &= \|\mathbf{z}_1\|^2 + \|\varepsilon\mathbf{e}_\ell + \mathbf{z}_2\|^2 \le \|\mathbf{z}_1\|^2 + \|\mathbf{z}_2\|^2 = \|\mathbf{z}\|^2 \\ &\le 2\sigma\sqrt{\mathsf{m}}. \end{aligned} \tag{25}$$

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

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT 2003*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, Berlin, Germany, 2003.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[3] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 333–342, Bethesda, Md, USA, June 2009.

[4] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, article no. 34, 2009.

[5] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pp. 197–206, ACM, Victoria, Canada, May 2008.

[6] V. Lyubashevsky and D. Micciancio, "Asymptotically efficient lattice-based digital signatures," in *Proceedings of the 5th IACR Theory of Cryptography Conference (TCC '08)*, pp. 37–54, New York, NY, USA, March 2008.

[7] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Advances in Cryptology—EUROCRYPT 2012*, vol. 7237 of *Lecture Notes in Computer Science*, pp. 738–755, Springer, Berlin, Germany, 2012.

[8] M. Tian and L. Huang, "Efficient identity-based signature from lattices," in *ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2–4, 2014. Proceedings*, vol. 428 of *IFIP Advances in Information and Communication Technology*, pp. 321–329, Springer, Berlin, Germany, 2014.

[9] M. Tian and L. Huang, "Certificateless and certificate-based signatures from lattices," *Security and Communication Networks*, vol. 8, no. 8, pp. 1575–1586, 2015.

[10] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[11] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, "Certificateless signature: a new security model and an improved generic construction," *Designs, Codes and Cryptography. An International Journal*, vol. 42, no. 2, pp. 109–126, 2007.

[12] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Proceedings of the 12th Australasian Conference on Information Security and Privacy (ACISP '07)*, pp. 308–322, Townsville, Australia, July 2007.

[13] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signatures: new schemes and security models," *The Computer Journal*, vol. 55, no. 4, pp. 457–474, 2012.