Cryptanalysis of a round optimal lattice-based multisignature scheme

Zi-Yuan Liu, Yi-Fan Tseng*, and Raylin Tso

Department of Computer Science, National Chengchi University, Taipei, Taiwan
{zyliu, yftseng, raylin}@cs.nccu.edu.tw

Abstract. Kansal and Dutta recently proposed a multisignature scheme at AFRICACRYPT 2020. This is the first lattice-based multisignature scheme that generates a multisignature in only a single round of interaction and supports public key aggregation. In this letter, we provide a cryptanalysis of this multisignature scheme and demonstrate that the scheme does not satisfy unforgeability requirements. We present an attack strategy to demonstrate that if an adversary obtains a sufficient number of signatures from a signer, he/she can recover the private key of the signer in polynomial time. We also uncover the root cause of the attack and provide a possible solution for this attack to aid future designs of secure multisignature schemes.

Keywords: Cryptanalysis · Multisignature · Lattices · Unforgeability

1 Introduction

The multisignature scheme, first introduced by Itakura and Nakamura [8], constitutes a digital signature scheme in which multiple users can cooperatively sign on a single message and the signature can be verified through the public keys of these users. In general, the size of a multisignature is more compact than the total size of the independent signatures of the users, constituting an advantage of multisignature schemes. Therefore, multisignature schemes are suitable for use in blockchain technology for reducing the size of transactions [2,10,13].

However, the security of the above multisignature schemes is based on the discrete logarithm problem. Shor [15,16] has discovered that there exists a quantum algorithm that can solve this hard problem. Therefore, with the gradual maturity of quantum computing technology, these multisignature schemes are becoming less secure. Accordingly, studies have proposed quantum-resistant multisignature schemes [3,6,7,14]. Nevertheless, these schemes are limited because they require multiple rounds of computation for signers to generate a signature. In addition, they do not support public key aggregation; specifically, to verify a signature, the verifier must store all of the signers' public keys and verify them individually.

To address the aforementioned limitations, Kansal and Dutta [9] recently proposed an optimal single-round multisignature scheme. The security of this

* Corresponding author.

scheme is based on the shortest integer solution (SIS) problem, a lattice hard problem; therefore, the scheme is considered to be resistant to quantum attacks. Furthermore, this scheme is equipped with a public key aggregation feature that enables users to aggregate all public keys into a single public key; the size of the aggregate key is the same as that of a single public key.

1.1 Contribution

The purpose of this letter is to demonstrate the limitations of the scheme proposed by Kansal and Dutta [9]. First, we reveal the flaw in the security proof of their scheme. Subsequently, we execute a forgery attack algorithm to demonstrate that their scheme does not meet unforgeability requirements. Specifically, we demonstrate that after obtaining a sufficient number of signatures provided by the same signer, our attack algorithm can retrieve the private key of the signer from the signatures. Additionally, in accordance with the parameter requirements stipulated by a previous study [9], we present an experimental evaluation to illustrate that our proposed attack is practical and useful under robust parameter settings for lattice-based cryptosystems. Finally, we present an analysis of the root cause of the attack and a possible solution to the problem, doing so to inspire new designs of improved multisignature schemes.

1.2 Organization

The remainder of this letter is organized as follows. Section 2 introduces the definition and security model of the concept of a multisignature. Section 3 presents a summary of the multisignature scheme proposed by Kansal and Dutta. Section 4 demonstrates first, the susceptibility of their scheme to forgery attacks and second, the usefulness of our proposed attack under robust parameter settings. Section 5 discusses the root cause of the attack and provides a possible solution. Finally, Section 6 concludes this letter.

2 Definition and security model of multisignature concept

This section provides a summary of the multisignature scheme presented in [9]. A multisignature scheme MS comprises four algorithms—namely a parameter generation algorithm pg, key generation algorithm kg, key aggregation algorithm kag, and verification algorithm vrf—and one protocol, namely an interactive signature generation protocol sg,

- $\mathsf{MS.pg}(1^\lambda) \to \mathcal{Y}$. This probabilistic polynomial time (PPT) algorithm takes a security parameter λ as its input and outputs a public parameter set $\mathcal{Y}$.
- $\mathsf{MS.kg}(\mathcal{Y}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$. This PPT algorithm takes a user i and the public parameter set $\mathcal{Y}$ as its inputs and outputs the public key pk_i and private key sk_i of user i.

- $\mathsf{MS.kag}(\mathcal{Y}, \mathcal{PK}) \rightarrow \mathsf{pkag}_{\mathcal{PK}}$. This deterministic algorithm takes the public parameter set $\mathcal{Y}$ and a set of public key of signers $\mathcal{PK}$ as its inputs and outputs an aggregated public key $\mathsf{pkag}_{\mathcal{PK}}$.
- $\mathsf{MS.sg}(\mathcal{Y}, \mathcal{PK}, \mathcal{SK}, M) \rightarrow \mathsf{msig}_{\mathcal{PK},M}$. Let $\mathcal{PK} = \{\mathsf{pk}_{i_1}, \cdots, \mathsf{pk}_{i_l}\}, \mathcal{SK} = \{\mathsf{sk}_{i_1}, \cdots, \mathsf{sk}_{i_l}\}$, and $I_{\mathsf{PK}} = \{i_1, \cdots, i_l\}$. In this protocol, each signer $i \in I_{\mathcal{PK}}$ uses $\mathcal{PK}$ along with its private key sk_i to generate a signature $\mathbf{T}_{i,M}$ for message M. The designated signer then aggregates all signatures $\mathbf{T}_{i,M}, i \in I_{\mathcal{PK}}$ into a multisignature $\mathsf{msig}_{\mathcal{PK},M}$.
- $\mathsf{MS.vrf}(\mathcal{Y}, \mathsf{msig}_{\mathcal{PK},M}) \rightarrow 0/1$. This deterministic algorithm takes the public parameter set $\mathcal{Y}$ and a multisignature $\mathsf{msig}_{\mathcal{PK},M}$ as its inputs and outputs 1 if $\mathsf{msig}_{\mathcal{PK},M}$ is valid. Otherwise, it outputs 0.

Definition 1 (Completeness of multisignature [9]). *A multisignature scheme* MS *meets completeness requirements if for any* $\mathcal{Y} \leftarrow \mathsf{MS.pg}(1^\lambda)$*, for any message* M*, and for any set of public keys* $\mathcal{PK} = \{\mathsf{pk}_1, \mathsf{pk}_2, \cdots, \mathsf{pk}_N\}$ *with a corresponding set of secret keys* $\mathcal{SK} = \{\mathsf{sk}_1, \mathsf{sk}_2, \cdots, \mathsf{sk}_N$*, where* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{MS.kg}(\mathcal{Y}, i)$ *for* $i = 1, 2, \cdots, N$*, the following holds.*

$$\begin{gathered} \textit{if } \mathsf{msig}_{\mathcal{PK},M} \leftarrow \mathsf{MS.sg}(\mathcal{Y}, \mathcal{PK}, \mathcal{SK}, M), \\ \textit{then } \mathsf{MS.vrf}(\mathcal{Y}, \mathsf{msig}_{\mathcal{PK},M}) = 1. \end{gathered}$$

For a multisignature scheme, the stipulated security requirement is that of unforgeability, which ensures that no PPT adversary can forge a multisignature with at least one honest signer [2]. This security requirement is modeled herein through an experiment $\mathsf{Exp}^{\mathsf{unforg}}_{\mathcal{F}}(\lambda)$ involving a simulator $\mathcal{S}$ and a forger $\mathcal{F}$, described as follows.

- **Setup** : The simulator $\mathcal{S}$ first generates system parameters $\mathcal{Y}$ and a challenge public key pk_{i^*} for user i^*. The simulator $\mathcal{S}$ then sends $(\mathcal{Y}, \mathsf{pk}_{i^*})$ to the forger $\mathcal{F}$.
- **Queries** : The forger $\mathcal{F}$ is allowed to query the signature oracle for $(M_i, \mathcal{PK}_i)$ in polynomial time, where M_i is a message and $\mathcal{PK}_i$ is a set of public keys with $\mathsf{pk}_{i^*} \in \mathcal{PK}$. The simulator $\mathcal{S}$ then returns a signature $\mathbf{T}_{i^*,M}$ on M.
- **Forgery** : The forger $\mathcal{F}$ outputs a forgery $\mathsf{msig}^*_{\mathcal{PK},M}$ on the message M for the set of public keys $\mathcal{PK}$. The simulator $\mathcal{S}$ returns 1 if the following conditions hold:
  1. $\mathsf{MS.vef}(\mathcal{Y}, \mathsf{msig}^*_{\mathcal{PK},M}) \rightarrow 1$,
  2. $\mathsf{pk}_{i^*} \in \mathcal{PK}$,
  3. M has not been queried in the signature oracle.

  Otherwise, $\mathcal{S}$ returns 0.

Definition 2 (Unforgeability of multisignature). *A multisignature scheme* MS *is unforgeable if* $\mathsf{Adv}^{\mathsf{unforg}}_{\mathcal{F}}(\lambda) = \mathsf{Pr}[\mathsf{Exp}^{\mathsf{unforg}}_{\mathcal{F}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$ *for all PPT adversaries* $\mathcal{F}$ *in the unforgeability experiment. Here* $\mathsf{negl}(\lambda)$ *is a negligible function in* λ*.*

3 Kansal and Dutta's multisignature scheme

In this section, we revisit the multisignature scheme proposed by Kansal and Dutta [9].

- $\mathsf{MS.pg}(1^\lambda) \to \mathcal{Y}$. In this algorithm, given a security parameter λ, the trusted key generation center executes the following steps to generate the public parameter set $\mathcal{Y}$:
  - choose $n = \mathcal{O}(\lambda), q = \mathcal{O}(n^3)$, and $m \geq 2n\lceil \log q \rceil$.
  - choose a standard deviation $\sigma = \Omega(\sqrt{n \log q} \log n)$.
  - select a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
  - choose three cryptographically secure hash functions: $H_0 : \{0,1\}^* \to \mathbb{Z}_q^{m \times n}, H_1 : \{0,1\}^* \to D_{\mathbb{Z}_q,\sigma}^{m \times n}$, and $H_2 : \{0,1\}^* \to D_{\mathbb{Z}_q,\sigma}^{n \times n}$. Here $D_{\mathbb{Z}_q,\sigma}^{k \times l} = \{\mathbf{W} \in \mathbb{Z}_q^{k \times l} : \|\mathbf{W}\| \leq \sigma\sqrt{k}\}$.
  - output the public parameter $\mathcal{Y} = (n, q, m, \sigma, \mathbf{A}, H_0, H_1, H_2)$.
- $\mathsf{MS.kg}(\mathcal{Y}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$. In this algorithm, given a public parameter $\mathcal{Y}$, each signer i executes the following steps to generate its public key pk_i and private key sk_i:
  - choose a short matrix $\mathbf{V}_i \in D_{\mathbb{Z}_q,\sigma}^{m \times m}$.
  - compute $\mathbf{Y}_i = \mathbf{A} \cdot \mathbf{V}_i \bmod q \in \mathbb{Z}_q^{n \times m}$.
  - set the public key $\mathsf{pk}_i = \mathbf{Y}_i \in \mathbb{Z}_q^{n \times m}$ and private key $\mathsf{sk}_i = \mathbf{V}_i \in D_{\mathbb{Z}_q,\sigma}^{m \times m}$.
- $\mathsf{MS.kag}(\mathcal{Y}, \mathcal{PK}) \to \mathsf{pkag}_{\mathcal{PK}}$. In this deterministic algorithm, given a public parameter $\mathcal{Y}$ and a set of public key $\mathcal{PK} = \{\mathsf{pk}_{i_1}, \mathsf{pk}_{i_2}, \cdots, \mathsf{pk}_{i_l}\}$, any user can compute an aggregated public key by computing the following equation:

$$\mathsf{pkag}_{\mathcal{PK}} = \sum_{i \in I_{\mathcal{PK}}} \mathsf{pk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \in \mathbb{Z}_q^{n \times n},$$

  where $I_{\mathcal{PK}} = \{i_1, i_2, \cdots, i_l\}$ is the index set of $\mathcal{PK}$.
- $\mathsf{MS.sg}(\mathcal{Y}, \mathcal{PK}, \mathcal{SK}, M) \to \mathsf{msig}_{\mathcal{PK},M}$. Given a public parameter $\mathcal{Y}$, a set of public keys $\mathcal{PK} = \{\mathsf{pk}_{i_1}, \mathsf{pk}_{i_2}, \cdots, \mathsf{pk}_{i_l}\}$ along with the corresponding private keys $\mathcal{SK} = \{\mathsf{sk}_{i_1}, \mathsf{sk}_{i_2}, \cdots, \mathsf{sk}_{i_l}\}$, and a message M, this interactive protocol executes the following steps to generate a multisignature:
  - each signer $i \in I_{\mathcal{PK}}$ computes a signature, as expressed in the following equation:

$$\begin{aligned} \mathbf{T}_{i,M} &= H_0(M, \mathcal{PK}) \\ &\quad + \mathsf{sk}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M), \end{aligned}$$

    where $\mathsf{sk}_i = \mathbf{V}_i, \mathsf{pk}_i = \mathbf{Y}_i$, and $I_{\mathcal{PK}}$ is the index set of $\mathcal{PK}$.
  - the designated signer first verifies whether each signature satisfies the following two conditions:
    * $\|\mathbf{T}_{i,M}\| \leq \|H_0(M, \mathcal{PK})\| + \sigma^3 m\sqrt{n}$.
    * $\mathbf{A} \cdot \mathbf{T}_{i,M} = \mathbf{A} \cdot H_0(M, \mathcal{PK}) + \mathbf{Y}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M)$.

- if the verification is successful, then the designated signer returns the multisignature $\mathsf{msig}_{\mathcal{PK},\mathcal{M}} = (\mathbf{T}_M, \mathsf{pkag}_{\mathcal{PK}}, I_{\mathcal{PK}}, M)$, where $\mathbf{T}_M = \sum_{i \in I_{\mathcal{PK}}} \mathbf{T}_{i,M} \bmod q$; otherwise, it returns $\perp$.

– $\mathsf{MS.vrf}(\mathcal{Y}, \mathsf{msig}_{\mathcal{PK},M}) \rightarrow 0/1$. In this algorithm, given a public parameter $\mathcal{Y}$ and a multisignature $\mathsf{msig}_{\mathcal{PK},M} = (\mathbf{T}_M, \mathsf{pkag}_{\mathcal{PK}}, I_{\mathcal{PK}}, M)$, the verifier outputs 1 if

- $\mathbf{A} \cdot \mathbf{T}_M = \mathbf{A} \cdot |I_{\mathcal{PK}}| \cdot H_0(M, \mathcal{PK}) + \mathsf{pkag}_{\mathcal{PK}} \cdot H_2(M)$ and
- $\|\mathbf{T}_M\| \leq |I_{\mathcal{PK}}| \cdot (\|H_0(M, \mathcal{PK})\| + \sigma^3 m\sqrt{n})$.

Otherwise, it outputs 0.

4 Cryptanalysis of Kansal and Dutta's multisignature scheme

4.1 Flaw in Kansal and Dutta's security proof

Although Kansal and Dutta presented a security proof to demonstrate that their multisignature scheme is unforgeable, we identified a flaw in their proof. Specifically, if an adversary $\mathcal{F}$ can output a forgery $\mathsf{msig}^*_{\mathcal{PK},M} = (\mathbf{T}^*_M, \mathsf{pkag}_{\mathcal{PK}}, I_{\mathcal{PK}}, M)$, the simulator $\mathcal{S}$ can then run the generalized forking lemma and obtain another forgery $\mathsf{msig}'_{\mathcal{PK},M} = (\mathbf{T}'_M, \mathsf{pkag}'_{\mathcal{PK}}, I_{\mathcal{PK}}, M)$. Kansal and Dutta claimed that because $\mathbf{A} \cdot \mathbf{T}^*_M = \mathbf{A} \cdot \mathbf{T}'_M \bmod q$ and $\|\mathbf{T}^*_M - \mathbf{T}'_M\| \leq \sigma^4 m\sqrt{n}$, the simulator $\mathcal{S}$ can obtain a solution $\mathbf{V}^* = \mathbf{T}^*_M - \mathbf{T}'_M$ to the SIS problem.

As mentioned in their proof, the two forgeries $\mathsf{msig}^*_{\mathcal{PK},M}$ and $\mathsf{msig}'_{\mathcal{PK},M}$ can be obtained under two different randomness settings $\rho = (\zeta, \mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_{t-1}, \mathbf{C}_t, \cdots, \mathbf{C}_{q_H})$ and $\rho' = (\zeta, \mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_{t-1}, \mathbf{C}'_t, \cdots, \mathbf{C}'_{q_H})$, respectively. Additionally, the outputs of the oracle H_1 are set according to the randomness settings. Specifically, $\mathbf{E}_j = H_1(\mathsf{pk}_j, \mathcal{PK}) = \mathbf{C}_q$ and $\mathbf{E}'_j = H_1(\mathsf{pk}_j, \mathcal{PK}) = \mathbf{C}_q$ for $j \in I_{\mathcal{PK}}$ and $q \in 1, \cdots, q_H$, respectively. According to the property of the generalized forking lemma [4], we determine that $\mathbf{E}_j$ is equal to $\mathbf{E}'_j$ for all $j \in I_{\mathcal{PK}}$ only in phases before the forking point. Therefore, we obtain

$$
\begin{aligned}
&\mathbf{A} \cdot \mathbf{T}^*_M \\
&= \mathbf{A} \cdot \sum_{i \in I_{\mathcal{PK}}} \mathbf{T}^*_{i,M} \\
&= \sum_{i \in I_{\mathcal{PK}}} \mathbf{A} \cdot H_0(M, \mathcal{PK}) + \mathbf{Y}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M) \\
&= \sum_{i \in I_{\mathcal{PK}}} \mathbf{A} \cdot H_0(M, \mathcal{PK}) + \mathbf{Y}_i \cdot \mathbf{E}_i \cdot H_2(M)
\end{aligned}
$$

and

$$\begin{aligned}
&\mathbf{A} \cdot \mathbf{T}'_M \\
&= \mathbf{A} \cdot \sum_{i \in I_{\mathcal{PK}}} \mathbf{T}'_{i,M} \\
&= \sum_{i \in I_{\mathcal{PK}}} \mathbf{A} \cdot H_0(M, \mathcal{PK}) + \mathbf{Y}_i \cdot H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M) \\
&= \sum_{i \in I_{\mathcal{PK}}} \mathbf{A} \cdot H_0(M, \mathcal{PK}) + \mathbf{Y}_i \cdot \mathbf{E}'_i \cdot H_2(M).
\end{aligned}$$

Hence, $\mathbf{A} \cdot \mathbf{T}^*_M \neq \mathbf{A} \cdot \mathbf{T}'_M \bmod q$; thus, $\mathbf{A} \cdot (\mathbf{T}^*_M - \mathbf{T}'_M) \neq 0 \bmod q$. Accordingly, we can conclude that $\mathbf{V}^* = \mathbf{T}^*_M - \mathbf{T}'_M$ is not a solution to the SIS problem.

4.2 Forgery attack on Kansal and Dutta's scheme

In this section, we demonstrate a forgery attack executed to break the unforgeable security of Kansal and Dutta's multisignature scheme. At a high level, we demonstrate that after obtaining a sufficient number of signatures from a same signer, the adversary can recover the private key of the signer. Subsequently, the adversary can use the obtained private key to adaptively sign any messages that can pass the verification process. Let $\mathcal{F}$ be the malicious adversary, given the challenge public key pk^*. Additionally, $\mathcal{F}$ can access the hash oracles and signature oracle.

- Let $t = \lceil m/n \rceil$; $\mathcal{F}$ first queries the signature oracle for t times for different $\mathcal{PK}$ on arbitrary messages (*i.e.*, for $i, j \in \{1, \cdots t\}$, and M_i can be equal to M_j).
- Then, without loss of generality, the signatures have the following relations:

$$\begin{aligned}
\mathbf{T}_{\mathsf{pk}^*, M_1} &= H_0(M_1, \mathcal{PK}_1) \\
&\quad + \mathsf{sk}^* \cdot H_1(\mathsf{pk}^*, \mathcal{PK}_1) \cdot H_2(M_1) \\
\mathbf{T}_{\mathsf{pk}^*, M_2} &= H_0(M_2, \mathcal{PK}_2) \\
&\quad + \mathsf{sk}^* \cdot H_1(\mathsf{pk}^*, \mathcal{PK}_2) \cdot H_2(M_2) \\
&\vdots \\
\mathbf{T}_{\mathsf{pk}^*, M_t} &= H_0(M_t, \mathcal{PK}_t) \\
&\quad + \mathsf{sk}^* \cdot H_1(\mathsf{pk}^*, \mathcal{PK}_t) \cdot H_2(M_t)
\end{aligned}$$

- For $i = 1, \cdots, t$, $\mathcal{F}$ computes the following matrices:

$$\begin{aligned}
\mathbf{O}_i &= \mathbf{T}_{\mathsf{pk}^*, M_i} - H_0(M_1, \mathcal{PK}_i) \in \mathbb{Z}_q^{m \times n} \\
\mathbf{P}_i &= H_1(\mathsf{pk}^*, \mathcal{PK}_t) \cdot H_2(M_i) \in \mathbb{Z}_q^{m \times n}
\end{aligned}$$

  and

$$\begin{aligned}
\mathbf{Q} &= [\mathbf{O}_1 | \mathbf{O}_2 | \cdots | \mathbf{O}_t] \in \mathbb{Z}_q^{m \times (nt)} \\
\mathbf{R} &= [\mathbf{P}_1 | \mathbf{P}_2 | \cdots | \mathbf{P}_t] \in \mathbb{Z}_q^{m \times (nt)},
\end{aligned}$$

such that $\mathbf{Q} = \mathsf{sk}^* \cdot \mathbf{R}$.
- $\mathcal{F}$ then defines two square matrices $\mathbf{Q}' \in \mathbb{Z}_q^{m \times m}$ and $\mathbf{R}' \in \mathbb{Z}_q^{m \times m}$ whose columns are the first m columns of $\mathbf{Q}$ and $\mathbf{R}$, respectively.
- Finally, $\mathcal{F}$ can recover the corresponding private key sk^* of the challenge public key pk^* by computing the following equation:

$$\mathsf{sk}^* = \mathbf{Q}' \cdot (\mathbf{R}')^{-1},$$

where $(\mathbf{R}')^{-1}$ is the inverse of $\mathbf{R}'$.

Section 5 details the attack and its root cause.

4.3 Experimental evaluation

This section presents an experimental evaluation to demonstrate that even under robust parameter settings, Kansal and Dutta's multisignature scheme is susceptible to our proposed attack. Table 2 presents information on our experimental platform. The algorithm was written using SageMath mathematical software[1] (version 9.1) and in Python language[2] (version 3.7). The source code of our algorithm can be found at `https://gist.github.com/cryptanalysis-MS/5741b817afcfe60af12f41465127d8a5`.

To illustrate that our proposed attack is effective under robust lattice parameter settings, we select three parameter settings from [11], as shown in Table 1. Moreover, these parameter settings meet the parameter requirements described in [9]:

- $q = \mathcal{O}(n^3)$,
- $m \geq 2n\lceil \log q \rceil$, and
- $\sigma = \Omega(\sqrt{n \log q} \log n)$.

.

Furthermore, to generate matrices with a discrete Gaussian distribution and width σ such that their norm is less than the required norm (*e.g.*, $\|\mathsf{sk}^*\| \leq \sigma\sqrt{m}$), our program recursively adopts SageMath's built-in function, namely *DiscreteGaussianDistributionIntegerSampler*. The results reveal that after obtaining at least $\lceil m/n \rceil$ number of signatures from the same signer for different $\mathcal{PK}$ on arbitrary messages, our proposed attack algorithm requires approximately 11.921, 35.625, 68.484, and 118.656 s to recover the signer's private key under our parameter settings.

5 Discussion

In this section, we first discuss the root cause of the attack and then provide possible solutions.

[1] https://www.sagemath.org/
[2] https://www.python.org/

Table 1. Time taken to recover signer's private key from the collected signatures under four parameter settings

Set	n	q	m	σ	**Time (s)**
1	128	2053	1953	152	11.921
2	192	4093	3194	211	35.625
3	256	4093	4259	256	68.484
4	320	4093	9643	433	118.656

Table 2. Information on experimental platform

Description	**Data**
CPU	Intel(R) Core(TM) i7-8700
CPU clock rate	3.2 GHz
CPU processor number	6
Operation system	Windows 10.0.18363
Random access memory	16.0 GB
Solid state disk	465.1 GB

In most lattice-based signature schemes [1,5,12], the main technique for hiding the information of the signer's private key entails adding "small" noise to the signature. Therefore, with the help of the corresponding public key, any verifier can verify the signature by eliminating the noise. However, in Kansal and Dutta's multisignature scheme, randomness is not added to generate the signature $\mathbf{T}_{i,M}$, except for the signer's private key.

Apparently, because $m \geq 2n\lceil \log q \rceil$, no right inverse matrix of $H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M) \in \mathbb{Z}_q^{m \times n}$ exists. Therefore, we cannot directly eliminate $H_1(\mathsf{pk}_i, \mathcal{PK}) \cdot H_2(M)$ to recover the private key sk_i. However, in our proposed attack algorithm—as described in Section 4—after obtaining $\lceil m/n \rceil$ signatures, any attacker can derive a square matrix $\mathbf{R}' \in \mathbb{Z}_q^{m \times m}$ and further recover the private key by computing the inverse of $\mathbf{R}'$.

Herein, we note that if at least two signatures $(\mathbf{T}_{\mathsf{pk}^*, M_i}, \mathbf{T}_{\mathsf{pk}^*, M_j})$, where $i, j \in \{1, \cdots, t\}$ and $i \neq j$, are generated from the same set of public keys $\mathcal{PK}$, then $\mathbf{R}'$ becomes a singular matrix. Accordingly, a simple solution to avoid such an attack is to restrict the signer to update their key pair after signing $\lceil m/n \rceil - 1$ numbers of signatures on the same set of public keys. To allow the signer to generate signatures at a number greater than the maximum allowable number of signatures, we may adopt the strategy underlying multisignature schemes [3,6,7,14], specifically by adding some noise that can be eliminated during verification. Nevertheless, the question of how to construct a secure and quantum-resistant single-round multisignature scheme still requires addressing.

6 Conclusion

In this letter, we present a cryptanalysis of the multisignature scheme proposed by Kansal and Dutta. We demonstrate that an adversary can easily recover a

signer's private key and further forge their signatures. Based on our experimental evaluation, we suggest for Kansal and Dutta's multisignature scheme to be used only when the signer does not need to generate an excessively large number of signatures.

Acknowledgment

This research was supported by the Ministry of Science and Technology, Taiwan (ROC), under Project Numbers MOST 108-2218-E-004-001-, MOST 108-2218-E-004-002-MY2, MOST 109-2218-E-011-007-.

References

1. Akleylek, S., Bindel, N., Buchmann, J., Krämer, J., Marson, G.A.: An efficient lattice-based signature scheme with provably secure instantiation. In: Pointcheval, D., Nitaj, A., T., R. (eds.) Progress in Cryptology – AFRICACRYPT 2016. Lecture Notes in Computer Science, vol 9646. pp. 44–60. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31517-1_3
2. Boneh, D., Drijvers, M., Neven, G.: Compact multi-signatures for smaller blockchains. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology – ASIACRYPT 2018. Lecture Notes in Computer Science, vol 11273. pp. 435–464. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03329-3_15
3. Damgård, I., Orlandi, C., Takahashi, A., Tibouchi, M.: Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. Cryptology ePrint Archive, Report 2020/1110 (2020), `https://eprint.iacr.org/2020/1110`
4. Drijvers, M., Edalatnejad, K., Ford, B., Kiltz, E., Loss, J., Neven, G., Stepanovs, I.: On the security of two-round multi-signatures. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 1084–1101. IEEE (2019). https://doi.org/10.1109/SP.2019.00050
5. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. **1**, 238–268 (2018). https://doi.org/10.13154/tches.v2018.i1.238-268
6. El Bansarkhani, R., Sturm, J.: An efficient lattice-based multisignature scheme with applications to Bitcoins. In: Foresti, S., Persiano, G. (eds.) Cryptology and Network Security. CANS 2016. Lecture Notes in Computer Science, vol 10052. pp. 140–155. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_9
7. Fukumitsu, M., Hasegawa, S.: A tightly-secure lattice-based multisignature. In: Proceedings of the 6th on ASIA Public-Key Cryptography Workshop. APKC '19. pp. 3–11. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3327958.3329542
8. Itakura, K., Nakamura, K.: A public-key cryptosystem suitable for digital multisignatures. NEC Research & Development (71), 1–8 (1983)
9. Kansal, M., Dutta, R.: Round optimal secure multisignature schemes from lattice with public key aggregation and signature compression. In: Nitaj, A., Youssef, A. (eds.) Progress in Cryptology - AFRICACRYPT 2020. Lecture Notes in Computer Science, vol 12174. pp. 281–300. Springer Cham (2020). https://doi.org/10.1007/978-3-030-51938-4_14

10. Le, D.P., Yang, G., Ghorbani, A.: A new multisignature scheme with public key aggregation for blockchain. In: 2019 17th International Conference on Privacy, Security and Trust (PST). pp. 1–7. IEEE (2019). https://doi.org/10.1109/PST47121.2019.8949046
11. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) Topics in Cryptology – CT-RSA 2011. Lecture Notes in Computer Science, vol 6558. pp. 319–339. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_21
12. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. Lecture Notes in Computer Science, vol 7237. pp. 738–755. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_43
13. Maxwell, G., Poelstra, A., Seurin, Y., Wuille, P.: Simple Schnorr multi-signatures with applications to Bitcoin. Des. Codes Cryptogr. **87**(9), 2139–2164 (2019). https://doi.org/10.1007/s10623-019-00608-x
14. Peng, C., Du, X.: New lattice-based digital multi-signature scheme. In: Qin, P., Wang, H., Sun, G., Z., L. (eds.) International Conference of Pioneering Computer Scientists, Engineers and Educators. ICPCSEE 2020. Communications in Computer and Information Science, vol 1258. pp. 129–137. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-7984-4_10
15. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE (1994). https://doi.org/10.1109/SFCS.1994.365700
16. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011