RESEARCH ARTICLE

Security Analysis of the Unrestricted Identity-Based Aggregate Signature Scheme

Kwangsu Lee, Dong Hoon Lee*

Center for Information Security Technologies, Korea University, Seoul, Korea

* donghlee@korea.ac.kr

OPEN ACCESS

Citation: Lee K, Lee DH (2015) Security Analysis of the Unrestricted Identity-Based Aggregate Signature Scheme. PLoS ONE 10(5): e0128081. doi:10.1371/journal.pone.0128081

Academic Editor: Yongtang Shi, Nankai University, CHINA

Received: January 14, 2015

Accepted: April 15, 2015

Published: May 18, 2015

Copyright: © 2015 Lee, Lee. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. 2010-0029121). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Aggregate signatures allow anyone to combine different signatures signed by different signers on different messages into a short signature. An ideal aggregate signature scheme is an identity-based aggregate signature (IBAS) scheme that supports full aggregation since it can reduce the total transmitted data by using an identity string as a public key and anyone can freely aggregate different signatures. Constructing a secure IBAS scheme that supports full aggregation in bilinear maps is an important open problem. Recently, Yuan et al. proposed such a scheme and claimed its security in the random oracle model under the computational Diffie-Hellman assumption. In this paper, we show that there is an efficient forgery on their IBAS scheme and that their security proof has a serious flaw.

1 Introduction

Aggregate signature schemes allow anyone to combine n different signatures on different n messages signed by different n signers into a short aggregate signature. The main advantage of aggregate signature schemes is the reduction of signature communication and storage requirements by the compression of multiple signatures into a single one. The applications of aggregate signature schemes include secure routing protocols, public-key infrastructure systems, and sensor networks. Boneh et al. [1] proposed the first aggregate signature scheme allowing anyone to combine different signatures in bilinear groups and proved its security in the random oracle model. Subsequently Lysyanskaya et al. [2] constructed a sequential aggregate signature scheme in which signatures can be combined sequentially, and Gentry and Ramzan [3] proposed a synchronized aggregate signature scheme in which all signers should share synchronized information. There are many additional aggregate signature schemes with different properties [4–13].

Although aggregate signature schemes can reduce the size of signatures by their aggregating them, they usually cannot reduce the total amount of transmitted data significantly since a verifier should additionally retrieve the public keys of all signers. Therefore, an important issue in aggregate signature schemes is reducing the size of public keys [7, 10–12]. An ideal solution to this problem is the use of an identity-based aggregate signature (IBAS) scheme since it uses an already known identity string as a user's public key [3]. However, there is only one IBAS scheme with full aggregation proposed by Hohenberger et al. [9] in multilinear maps.

Multilinear maps are attractive tools for cryptographic constructions, but they are currently impractical since they are based on leveled homomorphic encryption schemes [14]. There are some IBAS schemes in bilinear maps, but those schemes only support sequential or synchronized aggregation [3, 6, 8]. Therefore, constructing an IBAS scheme featuring full aggregation in bilinear maps is an important open problem.

The main challenge when devising an IBAS scheme with full aggregation is aggregating the random values of all signers such that each one hides the private key in the signature [3]. For this reason, current IBAS schemes can only aggregate the randomness of all signers through synchronized or sequential aggregation [3, 6]. Additionally, designing a secure IBAS scheme is not an easy task. In fact, even the original IBAS scheme of Boldyreva et al. [6] was broken by Hwang et al. [15] and had to be corrected later. Recently, Yuan et al. [16] proposed an IBAS scheme featuring full aggregation in bilinear maps and claimed its security in random oracle models. The authors first proposed an IBS scheme in bilinear maps and from that constructed an IBAS scheme. To demonstrate the security of their IBS scheme, the authors claimed that its security could be proven under the computational Diffie-Hellman (CDH) assumption by using Forking Lemma in the random oracle model.

In this paper, we show that the IBS and IBAS schemes of Yuan et al. [16] are not secure at all. First, we show that there is a universal forgery algorithm against their IBS scheme which outputs a forged signature by using two valid signatures. This forgery algorithm also applies to their IBAS scheme. One may wonder whether our forgery algorithm contradicts the security claims of their schemes or not. To explain this, we then show that the security proof of Yuan et al.'s IBS scheme has a serious flaw. The proof essentially relies on the condition that two forged signatures obtained using Forking Lemma have the same randomness. However, we reveal that the forged signatures of an adversary cannot satisfy this condition since the signatures of Yuan et al.'s IBS scheme are publicly re-randomized.

This paper is organized as follows: In Section 2, we review bilinear groups and the IBS and IBAS schemes of Yuan et al. In Section 3, we present a universal forgery against the IBS scheme. In Section 4, we analyze the security proof of Yuan et al.'s IBS scheme and show that it has a serious flaw.

2 The IBAS Scheme of Yuan et al

In this section, we review bilinear groups, complexity assumptions, and the IBS and IBAS schemes of Yuan et al. [16]. For more details, refer to [16].

2.1 Bilinear Groups and Complexity Assumptions

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p . Let g be a generator of $\mathbb{G}$. A bilinear map is a map $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e are all efficiently computable.

Assumption 2.1 (Computational Diffie-Hellman, CDH). *Let $\mathbb{G}$ be a cyclic group of prime order p and g be a generator of $\mathbb{G}$. The CDH assumption is that if the challenge tuple $D = (p, \mathbb{G}, g, g^a, g^b)$ is given, no probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ can compute $g^{ab} \in \mathbb{G}$ with more than a negligible advantage. The advantage of $\mathcal{A}$ is defined as $\text{Adv}_{\mathcal{A}}^{\text{CDH}}(\lambda) = \Pr[\mathcal{A}(D) = g^{ab}]$ where the probability is taken over random choices of $a, b \in \mathbb{Z}_p$.*

2.2 The Identity-Based Signature Scheme

The IBS scheme of Yuan et al. [16] consists of the algorithms, **Setup**, **GenKey**, **Sign**, and **Verify**, which are described as follows:

- **Setup**(1^λ): This algorithm takes as input a security parameter 1^λ . It generates bilinear groups $\mathbb{G}, \mathbb{G}_T$ of prime order p . Let g be a random generator of $\mathbb{G}$. It chooses random exponents $s_1, s_2 \in \mathbb{Z}_p^*$ and two cryptographic hash functions $H_1: \{0,1\}^* \rightarrow \mathbb{G}$ and $H_2: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. It outputs a master key $MK = (s_1, s_2)$ and public parameters $PP = ((p, \mathbb{G}, \mathbb{G}_T, e), g, g_1 = g^{s_1}, g_2 = g^{s_2}, H_1, H_2)$.
- **GenKey**(ID, MK, PP): This algorithm takes as input an identity $ID \in \{0,1\}^*$, the master key $MK = (s_1, s_2)$, and the public parameters PP . It outputs a private key $SK_{ID} = (D_1 = H_1(ID)^{s_1}, D_2 = H_1(ID)^{s_2})$.
- **Sign**(M, SK_{ID}, PP): This algorithm takes as input a message $M \in \{0,1\}^*$, a private key $SK_{ID} = (D_1, D_2)$, and the public parameters PP . It selects a random exponent $r \in \mathbb{Z}_p^*$ and computes $h = H_2(ID \| M)$. It outputs a signature $\sigma = (U = g^r, V = D_1^h \cdot g_1^r, W = D_2 \cdot g_2^r)$.
- **Verify**(σ, ID, M, PP): This algorithm takes as input a signature $\sigma = (U, V, W)$, an identity $ID \in \{0,1\}^*$, a message $M \in \{0,1\}^*$, and the public parameters PP . It computes $h = H_2(ID \| M)$ and checks whether $e(V, g) \stackrel{?}{=} e(H_1(ID)^h \cdot U, g_1)$ and $e(W, g) \stackrel{?}{=} e(H_1(ID) \cdot U, g_2)$. If both equations hold, then it outputs 1. Otherwise, it outputs 0.

Claim 2.2 ([16]) *The above IBS scheme is existentially unforgeable under chosen message attacks in the random oracle model if the CDH assumption holds.*

Remark 2.3 *The original IBS and IBAS schemes of Yuan et al. are described in the additive notation in bilinear groups. However, in this paper, we use the multiplicative notation for notational simplicity.*

Remark 2.4 *The signatures of Yuan et al.'s IBS scheme are publicly re-randomized. If $\sigma = (U, V, W)$ is a valid signature, then a re-randomized signature $\sigma' = (U \cdot g^r, V \cdot g_1^r, W \cdot g_2^r)$ is also valid where r is a random exponent in $\mathbb{Z}_p^*$.*

2.3 The Identity-Based Aggregate Signature Scheme

The IBAS scheme consists of the algorithms, **Setup**, **GenKey**, **Sign**, **Verify**, **Aggregate**, and **AggVerify**. The **Setup**, **GenKey**, **Sign**, and **Verify** algorithms of Yuan et al.'s IBAS scheme are the same as their IBS scheme counterparts. The additional algorithms of the IBAS scheme of Yuan et al. [16] are described as follows:

- **Aggregate**($\sigma_1, \sigma_2, S_1, S_2, PP$): This algorithm takes as input a signature $\sigma_1 = (U_1, V_1, W_1)$ on a multiset $S_1 = \{(ID_{1,1}, M_{1,1}), \dots, (ID_{1,n_1}, M_{1,n_1})\}$ of identity and message pairs, a signature $\sigma_2 = (U_2, V_2, W_2)$ on a multiset $S_2 = \{(ID_{2,1}, M_{2,1}), \dots, (ID_{2,n_2}, M_{2,n_2})\}$ of identity and message pairs, and the public parameters PP . It outputs an aggregate signature $\sigma = (U = U_1 \cdot U_2, V = V_1 \cdot V_2, W = W_1 \cdot W_2)$ on the multiset $S = S_1 \cup S_2$.
- **AggVerify**(σ, S, PP): This algorithm takes as input an aggregate signature $\sigma = (U, V, W)$, a multiset $S = \{(ID_1, M_1), \dots, (ID_n, M_n)\}$ of identity and message pairs, and the public parameters PP . It computes $h_i = H(ID_i \| M_i)$ for $i = 1, \dots, n$ and checks whether $e(V, g) \stackrel{?}{=} e(\prod_{i=1}^n H_1(ID_i)^{h_i} \cdot U, g_1)$ and $e(W, g) \stackrel{?}{=} e(\prod_{i=1}^n H_1(ID_i) \cdot U, g_2)$. If both equations hold, then it outputs 1. Otherwise, it outputs 0.

Claim 2.5 ([16]) *The above IBAS scheme is existentially unforgeable under chosen message attacks in the random oracle model if the underlying IBS scheme is unforgeable under chosen message attacks.*

3 Forgery Attacks on the IBAS Scheme

In this section, we show that the IBS and IBAS schemes of Yuan et al. are not secure at all by presenting an efficient forgery algorithm. In fact, our forgery algorithm is universal since anyone who has two valid signatures on the same identity and different messages can generate a forged signature on the same identity and any message of its choice.

Lemma 3.1 *There is a probabilistic polynomial-time (PPT) algorithm $\mathcal{F}$ that can forge the IBS scheme of Yuan et al. with non-negligible probability if $\mathcal{F}$ makes just two signature queries.*

Proof. The basic idea of our forgery attack is that if a forger obtains two valid signatures on an identity, then he can derive another valid signature by computing a linear combination of those signatures with carefully chosen scalar values. A forgery algorithm $\mathcal{F}$ is described as follows:

1. $\mathcal{F}$ randomly selects a target identity ID^* and two different messages M_1 and M_2 . It obtains a signature $\sigma_1 = (U_1, V_1, W_1)$ on the pair (ID^*, M_1) and a signature $\sigma_2 = (U_2, V_2, W_2)$ on the pair (ID^*, M_2) from the signing oracle.
2. It randomly selects a target message M^* for a forged signature. Next, it computes $h_1 = H_2(ID^* || M_1)$, $h_2 = H_2(ID^* || M_2)$, and $h^* = H_2(ID^* || M^*)$. It computes two exponents δ_1, δ_2 that satisfy the following equation

$$\begin{bmatrix} h_1 & h_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} h^* \\ 1 \end{bmatrix} \mod p.$$

Note that δ_1 and δ_2 can be computed using linear algebra since the determinant $h_1 - h_2$ of the left matrix is not zero if $h_1 \neq h_2$.

3. Finally, $\mathcal{F}$ outputs a forged signature σ^* on the identity and message pair (ID^*, M^*) as

$$\sigma^* = (U^* = U_1^{\delta_1} \cdot U_2^{\delta_2}, V^* = V_1^{\delta_1} \cdot V_2^{\delta_2}, W^* = W_1^{\delta_1} \cdot W_2^{\delta_2}).$$

To finish the proof, we should show that the forger $\mathcal{F}$ outputs a (forged) signature with non-negligible probability and that the forged signature passes the verification algorithm. We know that $\mathcal{F}$ always outputs a signature if $h_1 \neq h_2$. Because H_2 is a collision-resistant hash function and $M_1 \neq M_2$, $h_1 \neq h_2$ is of non-negligible probability. Now we should show that the forged signature is correct according to the verification algorithm. Let r_1 and r_2 be the randomness of σ_1 and σ_2 respectively. The correctness of the forged signature is easily verified as follows:

$$\begin{aligned} U^* &= U_1^{\delta_1} \cdot U_2^{\delta_2} = g^{r_1 \delta_1 + r_2 \delta_2} = g^{r^*}, \\ V^* &= V_1^{\delta_1} \cdot V_2^{\delta_2} = (H_1(ID^*)^{s_1} g_1^{r_1})^{\delta_1} \cdot (H_1(ID^*)^{s_1} g_1^{r_2})^{\delta_2} \\ &= H_1(ID^*)^{s_1 (h_1 \delta_1 + h_2 \delta_2)} g_1^{r_1 \delta_1 + r_2 \delta_2} = H_1(ID^*)^{s_1 h^*} g_1^{r^*}, \\ W^* &= W_1^{\delta_1} \cdot W_2^{\delta_2} = (H_1(ID^*)^{s_2} g_2^{r_1})^{\delta_1} \cdot (H_1(ID^*)^{s_2} g_2^{r_2})^{\delta_2} \\ &= H_1(ID^*)^{s_2 (h_1 \delta_1 + h_2 \delta_2)} g_2^{r_1 \delta_1 + r_2 \delta_2} = H_1(ID^*)^{s_2 h^*} g_2^{r^*} \end{aligned}$$

where the randomness of the forged signature is defined as $r^* = r_1 \delta_1 + r_2 \delta_2 \mod p$. This completes the proof.

Corollary 3.2 *There is a PPT algorithm $\mathcal{F}$ that can forge the IBAS scheme of Yuan et al. with non-negligible probability if $\mathcal{F}$ makes just two signature queries.*

The proof of this corollary is trivial from the proof of the previous Lemma since the IBAS scheme uses the IBS scheme as its underlying signature scheme. We omit the proof.

4 Our Analysis of the Security Proof

From the forgery algorithm presented in the previous section, it is evident that the IBS and IBAS schemes of Yuan et al. are not secure. However, Yuan et al. [16] claimed that their IBS scheme is secure in the random oracle model under the CDH assumption. In this section, we analyze the security proof of Yuan et al. and show that it has a critical flaw.

4.1 The Original Proof

In this subsection, we briefly review the security proof of Yuan et al.'s IBS scheme [16] that solves the CDH problem by using Forking Lemma [17, 18].

Suppose there is an adversary $\mathcal{A}$ that outputs a forged signature of the IBS scheme with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the CDH problem using $\mathcal{A}$ when it is given a challenge tuple $D = ((p, \mathbb{G}, \mathbb{G}_T, e), g, g^a, g^b)$ is described as follows:

- **Setup:** $\mathcal{B}$ chooses a random exponent $s_2 \in \mathbb{Z}_p^*$ and maintains an H_1 -list and an H_2 -list for random oracles. It implicitly sets $s_1 = a$ and publishes the public parameters $PP = ((p, \mathbb{G}, \mathbb{G}_T, e), g, g_1 = g^a, g_2 = g^{s_2}, H_1, H_2)$.
- **Hash Query:** If this is an H_1 hash query on an identity ID_i , then $\mathcal{B}$ handles this query as follows: If the identity ID_i already appears in the H_1 -list, then it responds with the value in the list. Otherwise, it picks a random coin $c \in \{0, 1\}$ with $\Pr[c = 0] = \delta$ for some δ and proceeds as follows: If $c = 0$, then it chooses $t_i \in \mathbb{Z}_p^*$ and sets $Q_i = (g^b)^{t_i}$. If $c = 1$, then it chooses $t_i \in \mathbb{Z}_p^*$ and sets $Q_i = g^{t_i}$. Next, it adds (ID_i, t_i, c, Q_i) to the H_1 -list and responds to $\mathcal{A}$ with $H_1(ID_i) = Q_i$.
If this is an H_2 hash query on an identity ID_i and a message M_i , then $\mathcal{B}$ handles this query as follows: If the tuple (ID_i, M_i) already appears in the H_2 -list, then it responds with the value in the list. Otherwise, it randomly chooses $h_i \in \mathbb{Z}_p^*$, adds (ID_i, M_i, h_i) to the H_2 -list, and responds with $H_2(ID_i || M_i) = h_i$.
- **Private-Key Query:** $\mathcal{B}$ handles a private key query for an identity ID_i as follows: It first retrieves (ID_i, t_i, c, Q_i) from the H_1 -list. If $c = 0$, then it aborts the simulation since it cannot create a private key. Otherwise, it creates the private key $SK_{ID_i} = (D_1 = g_1^{t_i}, D_2 = g_2^{t_i})$ and responds to $\mathcal{A}$ with SK_{ID_i} .
- **Signature Query:** $\mathcal{B}$ handles a signature query on an identity ID_i and a message M_i as follows: It randomly chooses $r' \in \mathbb{Z}_p^*$ and computes $h = H_2(ID_i || M_i)$. Next, it responds to $\mathcal{A}$ with a signature $\sigma = (U = g^{r'} \cdot H_1(ID_i)^{-h}, V = g_1^{r'}, W = H_1(ID_i)^{s_2} \cdot U^{s_2})$.
- **Output:** $\mathcal{A}$ finally outputs a forged signature $\sigma^* = (U^*, V^*, W^*)$ on an identity ID^* and a message M^* .

To solve the CDH problem, $\mathcal{B}$ retrieves the tuple (ID^*, t^*, c^*, Q^*) from the H_1 -list. If $c^* \neq 0$, then it aborts since it cannot extract the CDH value. Otherwise, it obtains two valid signatures $\sigma_1^* = (U_1^*, V_1^*, W_1^*)$ and $\sigma_2^* = (U_2^*, V_2^*, W_2^*)$ on the same identity and message tuple (ID^*, M^*) such that $U_1^* = U_2^*$ and $h_1^* \neq h_2^*$ by applying Forking Lemma. That is, it replays $\mathcal{F}$ with the same random tape but with a different choice of the random oracle H_2 . If $U_1^* = U_2^*$, then we

have the following equation

$$\begin{aligned} V_1^* \cdot (V_2^*)^{-1} &= H_1(ID^*)^{s_1 h_1^*} g_1^{s_1 r^*} \cdot (H_1(ID^*)^{s_1 h_2^*} g_1^{s_1 r^*})^{-1} \\ &= H_1(ID^*)^{s_1(h_1^* - h_2^*)} = (g^{ab})^{t^*(h_1^* - h_2^*)}. \end{aligned}$$

Thus, $\mathcal{B}$ can compute the CDH value as $(V_1^* \cdot (V_2^*)^{-1})^{1/(t^*(h_1^* - h_2^*))}$ if $h_1^* \neq h_2^* \pmod p$.

4.2 A Non-Extractable Forgery

To extract the CDH value from forged signatures by applying Forking Lemma, it is essential for the simulator to obtain two valid signatures σ_1^* and σ_2^* such that $U_1^* = U_2^*$ and $h_1^* \neq h_2^*$. By replaying the forgery with the same random tape, but with a different choice of random oracle H_2 , it is possible for the simulator to obtain two valid signatures $\sigma_1^* = (U_1^*, V_1^*, W_1^*)$ and $\sigma_2^* = (U_2^*, V_2^*, W_2^*)$ with $h_1^* \neq h_2^*$ because of Forking Lemma. However, we show that the probability of $U_1^* = U_2^*$ can be negligible if the forgery is clever.

Lemma 4.1 *If there is a PPT algorithm $\mathcal{A}$ that can forge the IBS scheme of Yuan et al., then there is another PPT algorithm $\mathcal{F}$ that can forge the IBS scheme with almost identical probability, but the simulator of Yuan et al. cannot extract the CDH value from the forged signatures of $\mathcal{F}$.*

Proof. The basic idea of this proof is that anyone can re-randomize the signature of Yuan et al.'s IBS scheme by using the public parameters. In this case, even though a simulator uses the same random tape for Forking Lemma, a forgery outputs a forged signature σ^* on an identity ID^* and a message M^* after re-randomizing it by using the information $h^* = H_2(ID^* \| M^*)$. Let $H': \{0,1\}^* \rightarrow \mathbb{Z}_p$ be a collision resistant hash function that is not modeled as the random oracle. A new forgery $\mathcal{F}$ that uses $\mathcal{A}$ as a sub-routine is described as follows:

1. $\mathcal{F}$ is first given PP and runs $\mathcal{A}$ by giving PP . $\mathcal{F}$ also handles the private key and signature queries of $\mathcal{A}$ by using his own private key and signature oracles.
2. $\mathcal{A}$ outputs a forged signature $\sigma' = (U', V', W')$ on an identity ID^* and a message M^* .
3. $\mathcal{F}$ computes $h^* = H_2(ID^* \| M^*)$ and $h' = H'(U' \| h^*)$, and then re-randomizes the forged signature of $\mathcal{A}$ as

$$\sigma^* = (U^* = U' \cdot g^{h'}, V^* = V' \cdot g_1^{h'}, W^* = W' \cdot g_2^{h'}).$$

4. Finally, $\mathcal{F}$ outputs a forged signature $\sigma^* = (U^*, V^*, W^*)$ on an identity ID^* and M^* .

To finish the proof, we should show that the forged signature of $\mathcal{F}$ is correct and the simulator of Yuan et al. cannot extract the CDH value from the forged signatures by using Forking Lemma. Let r' be the randomness of σ' . The correctness of the forged signature is easily checked as follows

$$\begin{aligned} U^* &= U' \cdot g^{h'} = g^{r' + h'} = g^{r^*}, \\ V^* &= V' \cdot g_1^{h'} = H_1(ID^*)^{s_1 h^*} g_1^{r' + h'} = H_1(ID^*)^{s_1 h^*} g_1^{r^*}, \\ W^* &= W' \cdot g_2^{h'} = H_1(ID^*)^{s_2 h^*} g_2^{r' + h'} = H_1(ID^*)^{s_2 h^*} g_2^{r^*} \end{aligned}$$

where $h^* = H_2(ID^* \| M^*)$ and $r^* = r' + h'$. To extract the CDH value from the forged signature of $\mathcal{F}$ by using Forking Lemma, the simulator of Yuan et al. should obtain two valid signatures $\sigma_1^* = (U_1^*, V_1^*, W_1^*)$ and $\sigma_2^* = (U_2^*, V_2^*, W_2^*)$ on the same identity and message pair (ID^*, M^*)

such that $U_1^* = U_2^*$ and $h_1^* \neq h_2^*$ after replaying $\mathcal{F}$ with the same random tape but with a different choice of the hash oracle H_2 . Let $\sigma_1^* = (U_1^*, V_1^*, W_1^*)$ and $\sigma_2^* = (U_2^*, V_2^*, W_2^*)$ be the two valid signatures obtained from $\mathcal{F}$ by using Forking Lemma. Let $\sigma_1' = (U_1', V_1', W_1')$ and $\sigma_2' = (U_2', V_2', W_2')$ be the original signatures before the re-randomization of $\mathcal{F}$. If $h_1^* \neq h_2^*$, then we have $h_1' \neq h_2'$ with non-negligible probability since H' is a collision-resistance hash function and the inputs of the hash function are different. From $h_1' \neq h_2'$, we have $U_1^* \neq U_2^*$ with non-negligible probability since U_1' and U_2' are re-randomized with difference values $g^{h_1'}$ and $g^{h_2'}$ respectively. Therefore, the event of obtaining two valid signatures only occurs with negligible probability. This completes our proof.

4.3 Discussions

From the above analysis, we know that the security of the original IBS scheme of Yuan et al. cannot be proven under the CDH assumption by applying Forking Lemma since the forger can easily re-randomize the signatures. To fix this problem, we may modify the IBS scheme to compute $h = H_2(U||ID||M)$ instead of computing $h = H_2(ID||M)$ where U is the first element of a signature. In this case, the forger cannot re-randomize the signature of the modified IBS scheme since U is an input of H_2 . Unfortunately, this modified IBS scheme cannot be used to construct an IBAS scheme since the U from the individual signatures cannot be aggregated. Note that if each U is aggregated, then a verifier cannot check the validity of an aggregate signature since each U is not included in the aggregate signature. Therefore, there is no easy solution to the problem.

5 Conclusion

In this paper, we showed that the IBAS scheme of Yuan et al. that achieves both full aggregation and constant pairing computation is not secure at all. We presented an efficient forgery algorithm against the IBS scheme and showed that their security proof of the scheme has a serious flaw. To invalidate the security of their IBS scheme, we showed that a linear combination of two valid signatures is also a new signature. Since the security of their IBAS scheme is based on the security of their IBS scheme, their IBAS scheme is also insecure. Therefore, constructing an IBAS scheme with full aggregation in bilinear maps still remains an important open problem.

Author Contributions

Conceived and designed the experiments: KL DHL. Performed the experiments: KL. Analyzed the data: KL. Contributed reagents/materials/analysis tools: KL. Wrote the paper: KL DHL.

References

1. Boneh D, Gentry C, Lynn B, Shacham H (2003) Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham E, editor, EUROCRYPT 2003. Springer, volume 2656 of *Lecture Notes in Computer Science*, pp. 416–432.
2. Lysyanskaya A, Micali S, Reyzin L, Shacham H (2004) Sequential aggregate signatures from trapdoor permutations. In: Cachin C, Camenisch J, editors, EUROCRYPT 2004. Springer, volume 3027 of *Lecture Notes in Computer Science*, pp. 74–90.
3. Gentry C, Ramzan Z (2006) Identity-based aggregate signatures. In: Yung M, Dodis Y, Kiayias A, Malkin T, editors, PKC 2006. Springer, volume 3958 of *Lecture Notes in Computer Science*, pp. 257–273.
4. Lu S, Ostrovsky R, Sahai A, Shacham H, Waters B (2006) Sequential aggregate signatures and multi-signatures without random oracles. In: Vaudenay S, editor, EUROCRYPT 2006. Springer, volume 4004 of *Lecture Notes in Computer Science*, pp. 465–485.

5. Ahn JH, Green M, Hohenberger S (2010) Synchronized aggregate signatures: new definitions, constructions and applications. In: ACM Conference on Computer and Communications Security. pp. 473–484.
6. Boldyreva A, Gentry C, O'Neill A, Yum DH (2007) Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In: Ning, P, di Vimercati, SDC, Syverson, PF, editors, ACM Conference on Computer and Communications Security. ACM, pp. 276–285. Full version available at <http://eprint.iacr.org/2007/438>.
7. Schröder D (2011) How to aggregate the cl signature scheme. In: Atluri V, Díaz C, editors, ESORICS 2011. Springer, volume 6879 of *Lecture Notes in Computer Science*, pp. 298–314.
8. Gerbush M, Lewko AB, O'Neill A, Waters B (2012) Dual form signatures: An approach for proving security from static assumptions. In: Wang X, Sako K, editors, ASIACRYPT 2012. Springer, volume 7658 of *Lecture Notes in Computer Science*, pp. 25–42.
9. Hohenberger S, Sahai A, Waters B (2013) Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In: Canetti R, Garay JA, editors, CRYPTO 2013. Springer, volume 8042 of *Lecture Notes in Computer Science*, pp. 494–512.
10. Lee K, Lee DH, Yung M (2013) Sequential aggregate signatures with short public keys: Design, analysis and implementation studies. In: Kurosawa K, Hanaoka G, editors, PKC 2013. Springer, volume 7778 of *Lecture Notes in Computer Science*, pp. 423–442.
11. Lee K, Lee DH, Yung M (2013) Aggregating cl-signatures revisited: Extended functionality and better efficiency. In: Sadeghi AR, editor, FC 2013. Springer, volume 7859 of *Lecture Notes in Computer Science*, pp. 171–188.
12. Lee K, Lee DH, Yung M (2013) Sequential aggregate signatures made shorter. In: MJJ Jr, Locasto ME, Mohassel P, Safavi-Naini R, editors, ACNS 2013. Springer, volume 7954 of *Lecture Notes in Computer Science*, pp. 202–217.
13. Lu S, Ostrovsky R, Sahai A, Shacham H, Waters B (2013) Sequential aggregate signatures, multisignatures, and verifiably encrypted signatures without random oracles. *J Cryptology* 26: 340–373. doi: [10.1007/s00145-012-9126-5](https://doi.org/10.1007/s00145-012-9126-5)
14. Garg S, Gentry C, Halevi S (2013) Candidate multilinear maps from ideal lattices. In: Johansson T, Nguyen PQ, editors, EUROCRYPT 2013. Springer, volume 7881 of *Lecture Notes in Computer Science*, pp. 1–17.
15. Hwang JY, Lee DH, Yung M (2009) Universal forgery of the identity-based sequential aggregate signature scheme. In: Li W, Susilo W, Tupakula UK, Safavi-Naini R, Varadharajan V, editors, ASIACCS 2009. ACM, pp. 157–160.
16. Yuan Y, Zhan Q, Huang H (2014) Efficient unrestricted identity-based aggregate signature scheme. *PLoS ONE* 9: e110100. doi: [10.1371/journal.pone.0110100](https://doi.org/10.1371/journal.pone.0110100) PMID: [25329777](https://pubmed.ncbi.nlm.nih.gov/25329777/)
17. Pointcheval D, Stern J (2000) Security arguments for digital signatures and blind signatures. *J Cryptology* 13: 361–396. doi: [10.1007/s001450010003](https://doi.org/10.1007/s001450010003)
18. Bellare M, Neven G (2006) Multi-signatures in the plain public-key model and a general forking lemma. In: Juels A, Wright RN, di Vimercati SDC, editors, ACM CCS 2006. ACM, pp. 390–399.