Security Analysis of Batch Verification on Identity-based Signature Schemes

HAN-FEI CHIANG, SUNG-MING YEN, HSI-CHUNG LIN
National Central University
Dept. of Computer Science and Information Eng.
Lab. of Cryptography and Information Security
Jhong-Li, TAIWAN 32001, R.O.C.

Abstract: Batch verification can improve efficiency when the verifier has to verify a great deal of signatures. With the rising interests on pairing-based cryptography, many researches on identity-based signatures have been proposed. Furthermore, some new researches on enhancing performance of verifying identity-based signatures by a batch verification have also been reported. Another possible weaker but faster batch checking of signatures, named as the batch screening, has been considered in the literature. This paper considers the security of recent batch verification schemes, and a new attack on one identity-based signature scheme with batch screening is proposed. Then, we point out that a previous attack on another identity-based signature scheme with batch screening is inappropriate due to the misunderstanding of the difference between batch screening and verification

Key–Words: Batch verification, Digital signature, Identity-based cryptography, Signature screening.

1 Introduction

Identity-based (ID-based) cryptography has been one of the most active fields in cryptology research since it was proposed by Shamir in 1984 [1] as it provides a simple and efficient alternative to traditional certificate-based public-key cryptography (PKC). In traditional PKC, a user's public key should be certified for identity relationship by a trusted third party before being used correctly. In contrast, ID-based cryptography aims to simplify or even to ignore the certificate issue and management problems in public key infrastructure (PKI) deployment. The main advantage of ID-based schemes is that no certificate is necessary because a user's public key can be derived directly from his identity information. Owing to the extensive simplification of public key management, ID-based schemes have attracted extensive research interests.

Numerous ID-based cryptographic schemes, such as ID-based signature (IBS), have been proposed recently due to the advances of pairing-based cryptography [2]. However, to the best of our knowledge, the pairing operations used in ID-based cryptography are still computationally expensive. In order to improve the performance of these pairing-based IBS schemes, batch verification seems to be an urgent and critical research topic.

Batch verification of signatures was firstly[1] proposed by Yen *et al.*, [4] and Naccache *et al.*, [5] independently. Instead of validating each individual signature separately, batch verification validates multiple signatures in a batch and thus improves the computational efficiency of signature verification substantially. In order to achieve high security service, given a batch instance consisting of multiple signatures, a batch verification should reject it with extremely high probability if there exists any invalid signature. Some other important researches of batch verification for signature schemes using modular exponentiations have been proposed in [6, 7].

In the pairing-based signature schemes proposed in [8, 9], batch verification of signatures from a same singer was considered. The first formal and detailed investigation on batch verification for IBS schemes was proposed by Yoon *et al.* in 2004 [10] in which batch verifications are classi-

[1] Batch operation for signatures was firstly considered by Fiat in [3], but only batch signature *generation* was taken into account in his work.

fied into three categories according to the number of signers and messages included in a batch.

Type 1 Multiple signatures on a single message generated by multiple signers.

Type 2 Multiple signatures on multiple messages generated by a single signer.

Type 3 Multiple signatures on multiple messages generated by multiple signers where each message is signed by a distinct user.

Furthermore, the authors of [10] defined attack models of all these three types of batch verifications for IBS schemes and showed that a previous IBS scheme proposed by Cha and Cheon [11] is secure of Type 2, but it is insecure of Type 1 and Type 3. Subsequently, Yoon *et al.* proposed a new IBS scheme which is secure in all batch verifications[2] of Types 1, 2, and 3 with security proofs. Yoon *et al.*'s scheme is the first IBS scheme being capable of batch screening (verification) of signatures signed by multiple signers, however the pairing operations increase linearly with the number of signers and messages.

Motivated by Yoon *et al.*'s work, in 2006 Cui *et al.* proposed an IBS scheme without *map-to-point* hash function, and it was claimed to be secure in batch screening (verification) of Types 2 and 3 (and Type 1) [12]. Comparing with Yoon *et al.*'s scheme [10], Cui *et al.*'s scheme is much more efficient since only constant number of pairing operations are necessary for the batch screening of Type 3.

In 2006, Cao *et al.* [13] proposed two attacks where Attack 1 is against Yoon *et al.*'s [10] and Zhang *et al.*'s [8, 9] pairing-based schemes, and Attack 2 is against Zhang *et al.*'s schemes [8, 9]. Cao *et al.* also considered a randomization technique as a countermeasure against the attacks. In fact, the idea of the countermeasure is identical to the well-known *small exponent test* proposed in [4, 5, 6, 7] but without emphasizing the effect of exponents with small bit length since only computationally expensive pairing operation is considered in efficiency analysis for IBS schemes.

As aforementioned claim, both Yoon *et al.*'s and Cui *et al.*'s schemes were in fact developed based on the notion of "*signature screening*" (by Bellare *et al.* [6]) instead of exact "batch verification". In signature screening, what the verifier cares is whether every message of the input signatures has been authenticated *at some point of time* rather than the validity of each individual message-signature pair. Screening approach provides a "weak but fast" batch verification of signatures. Actually, in these two schemes [10, 12], the authors analyzed the security of their schemes according to "screening" purpose despite the misusage of terminology of batch verification.

The main contribution of this paper is that Cui *et al.*'s IBS scheme will be shown to be insecure for *batch screening* of Type 3. By the way, we found that the attack proposed by Cao *et al.* to Yoon *et al.*'s IBS scheme with screening is inappropriate due to the misunderstanding of the difference between batch screening and verification.

2 Preliminary Background

2.1 Bilinear Maps

Suppose that $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive and multiplicative groups respectively, and both $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of the same prime order q. Let P be an arbitrary generator of $\mathbb{G}_1$. Assume that the discrete logarithm problem is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties:

1. **Bilinear** $\forall\ X, Y \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q, e(aX, bY) = e(X, Y)^{ab}$.
2. **Non-degenerate** $\exists\ X, Y \in \mathbb{G}_1$ such that $e(X, Y) \neq 1$.
3. **Efficient** There exists an efficient algorithm to compute $e(X, Y)$ for all $X, Y \in \mathbb{G}_1$.

2.2 ID-based Signature Schemes and Batch Verification

In general, an ID-based signature scheme includes four algorithms: Setup, Extract, Sign, and Verify.

Setup A Private Key Generator (PKG) sets up the system parameters **Param** and the system master key K_s. Then, PKG publishes **Param** to the users.

Extract For each identity ID, PKG generates the private key D_{ID} corresponding to ID using K_s and **Param**.

Signing A user ID uses **Param** and D_{ID} to produce a message-signature pair (m, σ, ID) on a message m.

[2] In fact, Yoon *et al.*'s scheme only achieves the weaker notion of screening to be discussed below. The usage of terminology batch verification might be a misleading.

Verify Given a message-signature pair (m, σ, ID), a verifier checks the validity of this signature using **Param** and ID (or function of ID) as the public key of this identity.

In a batch verification process, the **Verify** algorithm will be replaced by a **Batch Verification** algorithm [10].

Batch Verification Given multiple signatures $\sigma_1, \sigma_2, \ldots, \sigma_k$ on messages $m_1, m_2, \ldots, m_k$ and the corresponding signers' identities $ID_1, ID_2, \ldots, ID_k$ respectively, a verifier can check the validity of these signatures simultaneously. If $m_1 = m_2 = \ldots = m_k$, it is called the batch verification of Type 1. On the other hand, if $ID_1 = ID_2 = \ldots = ID_k$, it is called the batch verification of Type 2. For the most complex case, in the batch verification of Type 3, each distinct message is signed by a distinct signer.

2.3 Attack Model for Batch Verification

Yoon *et al.* in their work formalized the attack model for ID-based signature schemes with batch verification [10]. In this model, a forger $\mathcal{F}$ is called a k-batch forger of Type i (i=1,2, or 3) when he executes the following game.

Setup A k-batch forger $\mathcal{F}$ is given the system parameters **Param**.

Queries $\mathcal{F}$ can access the hash, **Extract**, and **Signing** oracles by his choices. $\mathcal{F}$ acquires the hash value of his queries, the private keys of his chosen ID's and the signatures of his chosen ID's and messages.

Outputs Finally, $\mathcal{F}$ outputs $ID_1, \ldots, ID_n$ and messages $m_1, \ldots, m_n$ and the corresponding signatures $\sigma_1, \ldots, \sigma_n$ of Type i, where $i = 1, 2, 3$ and $n \leq k$.

$\mathcal{F}$ wins if the output batch instance passes the batch verification process within polynomial time bound with non-negligible probability and there exists one index j such that ID_j has not been queried to the **Extract** oracle and the message m_j has not been queried to the **Signing** oracle.

The above definition stands for the declaration in [10] that they dealt with "signature screening" instead of batch verification. There is a kind of attack in which every message of a valid batch instance outputted by the adversary has been queried previously while some corresponding signatures are invalid. This is the only kind of attack which is considered successful in batch verification, but it is excluded in batch screening. Under the above attack model, this kind of attack is regarded as an unsuccessful one since all of the message has been queried previously, hence the model apparently dose not capture the definition of batch verification. On the other hand, the attack model restricts successful attacks to a specific situation in which at least one message has not been queried previously. This restriction completely rules out the kind of attack discussed above and thus the attack model corresponds to batch screening.

3 Security Analysis of Cui *et al.*'s IBS Batch Verification Scheme

A special type of hash function which maps message or identity information to a point on elliptic curve, named as the *map-to-point* function, is widely employed in pairing-based schemes. Since current constructions of map-to-point function are probabilistic and computationally expensive, Zhang *et al.* [14] suggested a more efficient pairing-based IBS scheme without the requirement of any map-to-point function. Motivated by [14], Cui *et al.* recently proposed a new IBS scheme without map-to-point function and base the security of their scheme on a special and complex assumption named "generalized k-CAA" [12]. It was claimed that Cui *et al.*'s IBS scheme is more efficient than all previous schemes and it is capable of batch verification of Type 2 and Type 3. Security proof of Type 3 batch verification is also given to support their claim. In this section, a practical attack against Cui *et al.*'s IBS scheme with Type 3 batch verification is proposed and the incompleteness of their security proof which enables the proposed attack is also pointed out.

3.1 Brief Review of Cui *et al.*'s IBS Scheme

Suppose that the trust authority (**TA**) chooses an arbitrary generator $P \in \mathbb{G}_1$ and randomly selects his master secret $x \in \mathbb{Z}_q^*$, then he computes $P_{pub} = xP$. The system parameters are (P, P_{pub}, ω, H), where $\omega = e(P, P)$ and $H : \{0,1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$ is a one-way hash function.

The private key extract procedure of Cui *et al.*'s IBS scheme follows the idea used in [15].

Given an identity $id \in \mathbb{Z}_q^*$, **TA** computes $S_{id} = \frac{1}{x+id}P$ as the private key of the identity id. Procedures of singing, verification, and batch verification of Type 3 are described as follows.

Signing To sign a message $m \in \{0,1\}^*$, the signer with private key S_{id} randomly selects $s \in \mathbb{Z}_q^*$, and computes

$$r = \omega^s,\ u = H(m,r),\ v = (u+s)S_{id},$$

where the resulting signature of m is the pair (r, v, id).

Verification Given a message m, a signature (r, v, id) and the system parameters (P, P_{pub}, ω, H), the verifier first computes $u = H(m,r)$ and then accepts (r, v, id) as a valid signature if

$$\omega^u r = e(P_{pub} + idP, v).$$

The verifier rejects (r, v, id), otherwise.

Batch Verification of Type 3 Suppose that there are at most λ message-signature pairs $(m_1, r_1, v_1, id_1), (m_2, r_2, v_2, id_2), \ldots, (m_\lambda, r_\lambda, v_\lambda, id_\lambda)$ where all the messages $m_1, m_2, \ldots, m_\lambda$ are distinct. The verifier computes $u_i = H(m_i, r_i), \forall i = 1, \ldots, \lambda$, and accepts all these signatures if the following equation holds:

$$w^{\sum_{i=1}^{\lambda} u_i} \prod_{i=1}^{\lambda} r_i = e(P_{pub}, \sum_{i=1}^{\lambda} v_i) e(P, \sum_{i=1}^{\lambda} id_i v_i). \tag{1}$$

3.2 The Proposed Attack against Cui *et al.*'s Scheme

Suppose the forger wants to forge two signatures signed respectively by id_1 and id_2 which can pass the Type 3 batch verification. The forger randomly chooses $s_1, s_2 \in \mathbb{Z}_q^*$ and computes $r_1 = \omega^{s_1}, r_2 = \omega^{s_2}$ and $u_1 = H(m_1, r_1), u_2 = H(m_2, r_2)$ as usual. Then, the forger computes

$$a = \frac{u_1 + u_2 + s_1 + s_2}{id_1 - id_2},\ v_1 = aP,\ v_2 = -aP.$$

Finally, the forger outputs a batch instance consisting of two message-signature pairs (m_1, r_1, v_1, id_1), (m_2, r_2, v_2, id_2), where both id_1 and id_2 have not been queried by the **Extract** oracle, and both m_1 and m_2 have not been queried by the **Signing** oracle previously. Obviously, the forged batch instance can pass the Type 3 batch verification procedure in Eq (1) since

$$\begin{aligned} & e(P_{pub}, v_1 + v_2) e(P, id_1 v_1 + id_2 v_2) \\ = \ & e(P, x(v_1 + v_2) + id_1 v_1 + id_2 v_2) \\ = \ & e(P, x(aP + (-aP)) + (id_1 - id_2)aP) \\ = \ & e(P, (u_1 + u_2 + s_1 + s_2)P) \\ = \ & e(P, P)^{u_1 + u_2 + s_1 + s_2} \\ = \ & \omega^{u_1 + u_2} r_1 r_2. \end{aligned}$$

The verifier might detect the proposed attack since $v_1 + v_2 = O$ where O stands for the *point at infinity* of the curve. However, the forger can insert some valid message-signature pairs which he collected (also possibly via **Signing** queries) previously. That is, following the attack game defined in [12], a k-batch forger $\mathcal{F}$ could win the attack game by outputting the following batch instance:
$(m_1, r_1, v_1, id_1), (m_2, r_2, v_2, id_2), (m_3, r_3, v_3, id_3), \ldots, (m_k, r_k, v_k, id_k)$
where $(m_3, r_3, v_3, id_3), \ldots, (m_k, r_k, v_k, id_k)$ are known valid message-signature pairs and $(m_1, r_1, v_1, id_1), (m_2, r_2, v_2, id_2)$ are forged as described previously. In this case, $v_1 + v_2 + v_3 + ... + v_k = O$ happens with extremely small possibility and hence hard for the verifier to detect the attack.

Recall that the security model used in the security analysis of Cui *et al.*'s scheme in [12] captures merely signature screening which is a security notion weaker than batch verification. The proposed attack shows that Cui *et al.*'s batch verification procedure cannot verify batch instances, and in fact even can not screen batch instances.

3.3 Flaw of the Security Proof of Cui *et al.*'s Scheme

A security proof of Cui *et al.*'s Type 3 batch verification procedure is presented in [12]. However, only limited forging case is considered in that proof. Namely, the authors merely considered the case where only one forged signature appears in the batch instance. In fact, the attack we proposed is a possible and practical case, but that proof does not includ it.

In the **Output** stage of the proof reduction, the authors of [12] claimed that:

1. In order to win the attack game, a k-batch forger $\mathcal{F}$ will finally output a valid batch instance $(m_1, r_1, v_1, id_1), \ldots, (m_k, r_k, v_k, id_k)$

which can pass the batch verification of Type 3, where id_k and m_k had not been queried before.

2. In addition, $(m_1, r_1, v_1, id_1), \ldots,$ $(m_{k-1}, r_{k-1}, v_{k-1}, id_{k-1})$ must pass the batch verification as well.

From the above two statements, an algorithm that can generate one single valid message-signature pair (i.e., (m_k, r_k, v_k, id_k)) of Cui *et al.*'s IBS scheme can be readily constructed, and then a solution of generalized k-CAA can be outputted with non-negligible probability.

However, the second statement is in fact an unreasonable assumption and there is no reason supporting that $(id_1, m_1, r_1, v_1), \ldots, (id_{k-1}, m_{k-1}, r_{k-1}, v_{k-1})$ will satisfy the batch verification equation. Furthermore, the second statement will limit the security proof to the specific case where only one forged signature is included in the whole batch instance. Apparently, there are many other possible cases where two or more forged signatures are included in the batch instance. Our attack is an example of a valid batch instance with $(k-2)$ valid message-signature pairs, plus 2 forged ones. Due to the previous reasons, we conclude that Cui's security proof of Type 3 batch verification is incomplete.

4 Discussion on the Security Analysis by Cao *et al.*

In [13], Cao *et al.* reviewed three pairing-based batch verification schemes proposed by Yoon *et al.* [10] and Zhang *et al.* [8, 9]. Afterward, the authors proposed two attacks against those three batch verification schemes. In their attacks, Attack 1 is against those three schemes and Attack 2 is against schemes proposed in [8, 9]. In this section, we will show that **Attack 1** is not a successful attack to Yoon *et al.*'s batch verification scheme.

In Yoon's signature scheme, an identity ID_i outputs a signature $\sigma_i = (V_i, U_i)$ on message m_i in **Signing** stage. To verify k multiple signatures of Type 3, the verifier checks whether following equation holds:

$$e(P, \sum_{i=1}^{k} V_i) = \prod_{i=1}^{k} e(Q_i, U_i + h_i P_{pub}), \qquad (2)$$

where P_{pub} is the system parameter, Q_i is a function of ID_i and h_i is a function of m_i and U_i.

The Attack 1 proposed by Cao *et al.* is very simple. Suppose that (m_1, V_1, U_1, ID_1), (m_2, V_2, U_2, ID_2), $\ldots$, (m_k, V_k, U_k, ID_k) are valid message-signature pairs. The forger randomly chooses $k-1$ values $V'_1, V'_2, \ldots, V'_{k-1}$ and finally solves the equation

$$V'_1 + V'_2 + \ldots + V'_{k-1} + V'_k = V_1 + V_2 + \ldots + V_k$$

to obtain the value of V'_k. Then, the forger takes multiple message-signature pairs
$(m_1, V'_1, U_1, ID_1), (m_2, V'_2, U_2, ID_2), \ldots,$
(m_k, V'_k, U_k, ID_k)
as the input instance of batch verification.

Obviously, this forged instance can satisfy Eq (2) while almost every individual signature is invalid. However, this attack is beyond the scope of the attack model defined in [10]. In the attack model, a forger is regarded as a winner only if there exists at least one index i such that ID_i and the corresponding message m_i have not been queried before. In the Attack 1, the forger simply modifies the value of V_i while keeping all the messages as before. Since all the messages have been queried previously, this attack dose not make sense under the attack model. This misunderstanding of attack is due to the reason that in [10] and [12], the authors misused the terminology of batch verification. In [10, 12], according to their attack model, "signature screening" rather than "batch verification" is considered.

Regarding Zhang *et al.*'s schemes [8, 9], the authors neither defined an attack model for batch verification nor specified clearly whether their scheme is designed for the purpose of screening or batch verification.

Regardless of the misunderstanding makde in [13], Cao *et al.* also considered a randomization technique as a countermeasure against the attacks. However, the idea of the countermeasure is identical to the well-known *small exponent test* proposed in the open literature [4, 5, 6, 7]. This technique can make those schemes be more secure such that batch verification can be achieved. This generic method might also be applied to repair Cui *et al.*'s scheme and make it be more secure than screening. More clearly, the batch verifier uses random factors in the batch verification process, so that the attacker cannot forge invalid signatures to pass the checking.

5 Conclusions

Yoon *et al.*'s IBS scheme with batch verification can achieve to verify multiple signatures signed by multiple signers simultaneously, say to achieve the Type 3 batch verification. However, their scheme is inefficient since the verifying cost increases with the number of signers. In 2006, Cui *et al.* have proposed a more efficient IBS scheme with Type 3 batch verification. Unfortunately, this scheme is insecure with Type 3 batch verification due to the attack proposed in this paper.

We also point out that the attack proposed by Cao *et al.* against Yoon *et al.*'s scheme dose not work under the attack model because Yoon *et al.* have limited their design signature screening. However, a well-known randomization technique can enhance the security of Yoon *et al.*'s and many other schemes.

Acknowledgements: This research was supported in part by the National Science Council R.O.C. under contract NSC 95-2221-E-008-045-MY2.

References:

[1] A. Shamir, "Identity-base cryptosystems and signature schemes," *Advances in Cryptology – Crypto '84*, LNCS 196, pp. 47–53, Springer-Verlag, 1985.

[2] D. Boneh, B. Lynn, and H. Shacham, "Short signature form weil pairing," *Proc. of Asiacrypt '01*, LNCS 2248, pp. 514–532, Springer-Verlag, 2001.

[3] A. Fiat, "Batch RSA," *Advances in Cryptology – Crypto '89*, LNCS 435, pp. 175–185, Springer-Verlag, 1990.

[4] S.M. Yen and C.S. Laih, "Improved digital signature suitable for batch verification," *IEEE Trans. on Computers*, vol. 44, no. 7, pp. 957–959, 1995.

[5] D. Naccache, D. M'raihi, S. Vaudenay, and D. Raphaeli, "Can D.S.A. be improved? complexity trade-offs with the digital signature standard," *Advances in Cryptology – Eurocrypt '94*, LNCS 950, pp. 77–85, Springer-Verlag, 1994.

[6] M. Bellare, J. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," *Advances in Cryptology – Eurocrypt '98*, LNCS 1403, pp. 236–250, Springer-Verlag, 1998.

[7] J. H. Cheon and D. H. Lee, "Use of sparse and/or complex exponents in batch verification of exponentiations," *IEEE Trans. on Computers*, vol. 55, no. 12, pp. 1536–1542, 2006.

[8] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," *Proc. of ACISP '03*, LNCS 2727, pp. 312–323, Springer-Verlag, 2003.

[9] F. Zhang, R. Safavi-Naini, and W. Susilo, "Efficient verifiably encrypted signature and partially blind signature from bilinear pairings," *Proc. of Indocrypt '03*, LNCS 2904, pp. 191–204, Springer-Verlag, 2003.

[10] H. Yoon, J. H. Cheon, and Y. Kim, "Batch verifications with ID-based signatures," *Information Security and Cryptology – ICISC '04*, LNCS 3506, pp. 233–248, Springer-Verlag, 2005.

[11] J. Cha and J. Cheon, "An ID-based signature from gap-Diffie-Hellman groups," *Public Key Cryptography – PKC '03*, LNCS 2567, pp. 18–30, Springer-Verlag, 2003.

[12] S. Cui, P. Duan and C. W. Chan, "An efficient identity-based signature scheme with batch verifications," *Proc. of the First International Conference on Scalable Information Systems – INFOSCALE '06*, vol. 152, ACM press, 2006.

[13] T. Cao, D. Lin and R. Xue, "Security analysis of some batch verifying signatures from pairings," *International Journal of Network Security*, vol.3, no.2, pp.112–117, 2006.

[14] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," *Public Key Cryptography – PKC '04*, LNCS 2947, pp. 277–290, Springer-Verlag, 2004.

[15] S. Mitsunari, R. Sakai, and M. Kasahara, "A new traitor tracing," *IEICE Trans. Fundamentals*, vol. E85-A, no. 2, pp. 481–484, 2002.