Efficient Signature Scheme with Batch Verifications in Identity-Based Framework

P.V.S.S.N. Gopal, P. Vasudeva Reddy, and T. Gowri

In group-oriented applications, it is often required to verify a group of signatures/messages. The individual verification of signed messages in such applications comes at a high cost in terms of computations and time. To improve computational efficiency and to speed up the verification process, a batch verification technique is a good alternative to individual verification. Such a technique is useful in many real-world applications, such as mail servers, e-commerce, banking transactions, and so on. In this work, we propose a new, efficient identity-based signature (IDS) scheme supporting batch verifications. We prove that the proposed IDS scheme and its various types of batch verifications is tightly related to the Computational Diffie–Hellman problem under a random oracle paradigm. We compare the efficiency of the proposed scheme with related schemes that support batch verifications.

Keywords: Identity-based signature, batch verification, bilinear pairings, CDH problem, unforgeability.

I. Introduction

In group-oriented applications and multicast communications, it is often required to verify a group of signatures/messages. However, individual verification of signed messages comes at a high cost in terms of computations and time. Verification of signatures on a batch basis is essential in many real-world applications, such as mail servers, e-commerce, e-voting, banking transactions, and so on. To improve the verification process and minimize the verification time, a signature scheme with batch verifications — one that verifies multiple signatures simultaneously as a whole — is needed.

The notion of batch cryptography was introduced by Fiat [1] in 1989. Fiat proposed a modified version of RSA suitable for batch signature generation. Many public key infrastructure (PKI)–based schemes with batch verifications have been proposed in the literature [2]–[4]. Bellare and others' scheme [4] gave a systematic approach for batch verification and presented three generic methods for batching modular exponentiations: the random subset test, the small exponents test, and the bucket test.

To overcome the task of maintaining certificate libraries used for revoking, storage, and distribution of certificates that require huge communication overload within a PKI-based setting, Shamir [5], in 1984, devised a new paradigm called identity-based cryptography (IBC). Under such a paradigm, the public key of a user can be directly derived from the user's personal information, such as a telephone number, an e-mail address, and so on. The corresponding private key is issued by a trusted authority, termed a key generation center (KGC). Since 1984, many encryption and signature schemes have been constructed in identity-based settings, but the most usable and

Manuscript received Oct. 6, 2014; revised Oct. 8, 2015; accepted Dec. 9, 2015.

P.V.S.S.N. Gopal (gopalcrypto786@gmail.com) and P. Vasudeva Reddy (corresponding author, vasucrypto@yahoo.com) are with the Department of Engineering Mathematics, AUCE (A), Andhra University, Andhra Pradesh, India.

T. Gowri (gowri3478@yahoo.com) is with the Department of Electrical and Computer Engineering, GITAM University, Andhra Pradesh, India.

practical encryption scheme was devised by Boneh and Franklin [6] in 2001 using Weil pairing over elliptic curves. Based on Boneh and Franklin's work in [6], many signature schemes in identity-based settings have been proposed [7]–[10].

However, so far, little attention has been paid to design signature schemes that support batch verification. To facilitate the wide use of identity-based signature (IDS) schemes in real applications such as e-commerce, electronic payment systems, and e-government, it is necessary and important to study the design of secure and efficient batch verifications for ID-based signature schemes.

The first IDS scheme supporting batch verifications using pairings over elliptic curves was proposed by Yoon and others [11] in 2004. Based on the number of messages and signers, the authors in [11] classified multiple signatures into the following types:

- Type 1: Multiple users sign on a single message.
- Type 2: A single user signs on multiple messages.
- Type 3: Multiple users sign on multiple messages, where every message is signed by a different user.

Cao and others [12], in 2006, showed that the scheme in [11] is not secure, since an adversary can deceive a verifier to accept an invalid signature. In the same year, Cui and others [13] proposed an IDS scheme supporting batch verifications of Types 2 and 3 above with a different key construction. In 2007, Chiang and others [14] proved that the scheme in [13] is insecure.

Zhang and others [15], in 2008, proposed an efficient identity-based batch verification scheme for vehicular sensor networks using elliptic curves. Tseng and others [16] discussed the twelve schemes of Cha and Cheon [7], such as signature schemes, and obtained an efficient IDS scheme that supported different types of batch verifications, in 2009. Hwang and others [17] proposed a new, efficient batch verification for an IDS scheme using pairings, in 2015. Ren and others [18], in 2015, proposed an efficient batch verification scheme for detecting illegal signatures without pairings over elliptic curves.

The schemes in [11] and [16] require a linear number of pairing operations with that of signers for a batch verification of Type 3. As discussed in [19] and [20], the security reductions of the schemes in [11] and [16] are not tight, since these schemes use the forking lemma [21] to prove their security.

In this paper, we propose a new, efficient IDS scheme supporting batch verifications. This scheme uses bilinear pairings over elliptic curves and is secure under a random oracle paradigm with the assumption that the Computational Diffie–Hellman (CDH) problem is hard. The proposed IDS scheme provides tight reductions due to the fact that its security is not proven through use of the forking lemma [21].

The rest of this paper is organized as follows. Section II presents some preliminaries, including bilinear maps and complexity assumptions. The proposed IDS scheme is depicted in Section III. A security analysis of the IDS scheme is presented in Section IV. Batch verifications of the proposed IDS scheme are introduced in Section V. A security analysis of the batch verifications of the proposed IDS scheme is presented in Section VI. A complexity analysis for the batch verifications of the proposed IDS scheme is presented in Section VII, and Section VIII concludes our work.

II. Preliminaries

This section summarizes some fundamental concepts and necessary hard problems.

1. Bilinear Map

Let $(G, +)$ and $(G_T, \cdot)$ be additive and multiplicative cyclic groups, respectively, of the same prime order q; that is, $|G| = |G_T| = q$. Let P be a generator in G. A map $\hat{e}: G \times G \to G_T$ is *bilinear* if the following properties are satisfied:

- Bilinear: For all $A, B \in G$, and for any $x, y \in Z_q^*$, $\hat{e}(xA, yB) = e(A, yB)^x = \hat{e}(xA, B)^y = e(A, B)^{xy}$.
- Non-degeneracy: There is an element in G, say $A \in G$, such that $\hat{e}(A, A) \neq 1$.
- Computable: For any $A, B \in G$, the map $\hat{e}(A, B)$ is computable using an efficient algorithm.

Upon making suitable variations in the Weil or Tate pairing, one can obtain such maps on elliptic curves over a finite field [6], [22].

2. Notations and their Descriptions

Table 1 illustrates the notations and their descriptions used in the proposed scheme.

3. Computational Problems

In the following, we present some computationally hard problems on which the proposed scheme's security is based.

A. CDH Problem

For a given $x, y \in Z_q^*$ and CDH tuple, $P, xP, yP \in G$, the CDH problem is to find $xyP \in G$. Given an adversary $\mathcal{A}$, the advantage of this adversary, $\mathrm{Adv}(\mathcal{A})$, to solve the CDH problem in G in polynomial time with running time t is defined as follows:

$$\mathrm{Adv}_{\mathrm{CDH},\,t}(\mathcal{A}) = \Pr[\mathcal{A}(P, xP, yP) = xyP / x,\ y \in Z_q^*].$$

ETRI Journal, Volume 38, Number 2, April 2016
http://dx.doi.org/10.4218/etrij.16.0114.1193

Table 1. Notations and their descriptions.

Notations	Description
$(G, +)$	A cyclic group under addition
$(G_T, \cdot)$	A cyclic group under multiplication
q	Order of the groups G and G_T
$\hat{e}$	A symmetric bilinear map defined from $G \times G$ to G_T
P	A generator of the group G
ID	The identity of a user
P_{pub}	The system's overall public key
H_1	A cryptographic hash function defined by $H_1 : \{0, 1\}^* \rightarrow G$
H_2	A cryptographic hash function defined by $H_2 : \{0, 1\}^* \times G_T \rightarrow Z_q^*$
Q_{ID}	The public key of a user with identity ID
d_{ID}	The private key of a user with identity ID
σ	A signature on the message m made by the signer with identity ID

B. CDH Assumption

For any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage, $\mathrm{Adv}_{\mathrm{CDH},\,t}(\mathcal{A})$, is negligibly small.

III. New IDS Scheme

In this section, we present our new IDS scheme. This scheme comprises four algorithms: System Setup, Key Extract, Signature Generation, and Signature Verification. Detailed functionalities of these algorithms are as follows.

1. System Setup

For a given security parameter $l \in Z^+$, the KGC runs this algorithm to generate the following system parameters:

- Generates additive and multiplicative cyclic groups, say $(G, +)$ and $(G_T, \cdot)$, respectively, of the same prime order q; that is, $|G| = |G_T| = q$.
- Generate a generator $P \in G$ and an admissible bilinear map $\hat{e}$ such that $\hat{e} : G \times G \rightarrow G_T$.
- Generate an integer $s \in Z_q^*$ at random and compute $P_{pub} = sP$, $g = \hat{e}(P_{pub}, P)$.
- Picks cryptographic hash functions $H_1 : \{0, 1\}^* \rightarrow G$ and $H_2 : \{0, 1\}^* \times G_T \rightarrow Z_q^*$.
- Publishes the system's public parameters as Params = $< G, G_T, q, \hat{e}, P, P_{pub}, H_1, H_2, g >$ and keeps $<s>$ securely as the master private key.

2. Key Extract

This algorithm, run by the KGC, receives an identity $ID \in \{0, 1\}^*$ of a user and then computes $Q_{ID} = H_1(ID)$ and $d_{ID} = sQ_{ID} \in G$. The KGC securely transmits (Q_{ID}, d_{ID}) to the user with identity "ID." The user keeps d_{ID} securely and makes Q_{ID} public.

3. Signature Generation

The user provides the following information as input for this algorithm: identity ID, private key d_{ID}, Params, and message $m \in \{0, 1\}^*$. The computations performed are as follows:

- Select an integer $r \in Z_q^*$ at random and compute $U = g^r \in G_T$, $h = H_2(m, ID, U) \in Z_q^*$, and $V = hd_{ID} + rP_{pub} \in G$.
- Generate the signature on message m of the user with identity ID as $\sigma = (U, V) \in G_T \times G$.

4. Signature Verification

Any *verifier* can run this algorithm, which takes the signature σ on a message m by a user with identity ID as input. The verification is done as follows:

- Compute the hash value $h = H_2(m, ID, U) \in Z_q^*$.
- Verify the validity of the equation $\hat{e}(P, V) = \hat{e}(P_{pub}, hQ_{ID})U$. If it is valid, then accept the signature; else, reject the signature.

IV. Security Analysis

This section presents a proof of correctness and a security reduction of the proposed IDS scheme under an adaptively chosen message and ID attack under a random oracle paradigm.

1. Proof of Correctness

The following equation shows that the proposed IDS scheme is correct; the verification equation is valid:

$$\begin{aligned}\hat{e}(P, V) &= \hat{e}(P, hd_{ID} + rP_{pub}) \\ &= \hat{e}(P, hd_{ID})\hat{e}(P, rP_{pub}) \\ &= \hat{e}(sP, hQ_{ID})\hat{e}(sP, P)^r \\ &= \hat{e}(P_{pub}, hQ_{ID})\, U.\end{aligned}$$

2. Security Reduction

In the following, we prove that the proposed scheme is unforgeable under chosen message and identity attacks under a random oracle paradigm, with the assumption that the CDH

problem is hard.

Theorem 1. Let $\mathcal{A}$ be a probabilistic polynomial time forger who forges the proposed IDS scheme with non-negligible advantage. Then, there is an algorithm $\mathcal{B}$ that can output the given CDH instance $(P, aP, bP) \in G$ with a non-negligible advantage in probabilistic polynomial time.

Proof. Let $\mathcal{A}$ be a forger who breaks the proposed IDS scheme. We show that by using $\mathcal{A}$ one can construct an algorithm $\mathcal{B}$ that can solve the CDH problem. Algorithm $\mathcal{B}$ is given (P, aP, bP) as a random instance of the CDH problem in G; that is, its goal is to output $abP \in G$. Algorithm $\mathcal{B}$ simulates an original signer to obtain a valid signature from $\mathcal{A}$, and by doing so, it can solve the CDH problem.

A. Setup/Queries

Algorithm $\mathcal{B}$ sets $P_{\text{pub}} = aP$ as the system's overall public key and provides $\mathcal{A}$ with Params. At any time, $\mathcal{A}$ may make queries to oracles H_1, H_2, key extract, and signature. We presume that prior to any query from key extract, both a signature query and a H_1 query have already been made on an identity ID. To respond to these queries, algorithm $\mathcal{B}$ does the following:

- H_1- queries: Algorithm $\mathcal{B}$ keeps a list, L_1, which is empty initially of tuples, (ID, c, d, v) to respond to H_1- queries. Upon receiving a query from the H_1 oracle for $\text{ID} \in \{0, 1\}^*$, made by $\mathcal{A}$, algorithm $\mathcal{B}$ proceeds as follows:
  - (i) If L_1 consists of the queried ID, then algorithm $\mathcal{B}$ responds with $H_1(\text{ID}) = v \in G$.
  - (ii) If not, then algorithm $\mathcal{B}$ flips a coin $d \in \{0, 1\}$ generated at random, such that $\Pr[d = 0] = 1/(q_K + 1)$. Here, q_K denotes a query made to the key extraction oracle.
  - (iii) Now, algorithm $\mathcal{B}$ picks a random integer $c \in Z_q^*$ and computes $v = c(bP) \in G$ for $d = 0$, and $v = cP \in G$ for $d = 1$.
  - (iv) Algorithm $\mathcal{B}$ adds (ID, c, d, v) to list L_1 and returns $H_1(\text{ID}) = v \in G$ to $\mathcal{A}$.
- H_2- queries: Algorithm $\mathcal{B}$ keeps a list L_2 of tuples, (m, ID, U, w), which is empty initially. To respond to H_2 queries made by $\mathcal{A}$ on tuple (m, ID, U), algorithm $\mathcal{B}$ proceeds as follows:
  - (i) If L_2 contains queried tuple (m, ID, U), then algorithm $\mathcal{B}$ provides $H_2(m, \text{ID}, U) = w \in Z_q^*$.
  - (ii) If not, then algorithm $\mathcal{B}$ picks a random integer $w \in Z_q^*$, inserts (m, ID, U, w) in L_2 and returns $H_2(m, \text{ID}, U) = w \in Z_q^*$ to $\mathcal{A}$.

B. Key Extraction Queries

Upon receiving the private key queries on an identity ID by $\mathcal{A}$, algorithm $\mathcal{B}$ retrieves the respective tuple (ID, c, d, v) from L_1 and does the following:

1) It outputs "failure" and then halts, for $d = 0$.
2) If $d = 1$, then it computes and returns $d_{\text{ID}} = cP_{\text{pub}} = c(aP) = a(cP) \in G$ to $\mathcal{A}$.

C. Signature Queries

Upon receiving the signature query on a message m under ID from $\mathcal{A}$, algorithm $\mathcal{B}$ retrieves the H_1 oracle and obtains the tuple (ID, c, d, v) from L_1. Algorithm $\mathcal{B}$ then selects a random integer $x \in Z_q^*$ and computes $U = g^x$. In addition, if the list L_2 contains the tuple (m, ID, U, w), then $\mathcal{B}$ chooses $w' \in Z_q^*$ and tries again; that is, $\mathcal{B}$ adds (m, ID, U, w') to L_2. Now, $\mathcal{B}$ computes $V = (wc + x)P_{\text{pub}}$ and returns $\sigma = (U, V)$ to $\mathcal{A}$ as the queried signature.

The responses to signature queries are valid, as well the output σ. This can be seen from the following:

$$\begin{aligned}\hat{e}(P, V) &= \hat{e}\left(P, (wc + x)P_{\text{pub}}\right) \\ &= \hat{e}(P, wcP_{\text{pub}})\hat{e}(P, xP_{\text{pub}}) \\ &= \hat{e}(aP, wcP)\hat{e}(aP, xP) \\ &= \hat{e}(P_{\text{pub}}, wQ_{\text{ID}})U.\end{aligned}$$

D. Forgery

Eventually, $\mathcal{A}$ stops by conceding failure or returns a forgery σ on m under ID. Algorithm $\mathcal{B}$ obtains (ID, c, d, v) from L_1, declares failure if $d = 1$, and stops. If not, then it computes $Q_{\text{ID}} = c(bP)$ for $d = 0$. The forged signature σ must satisfy $\hat{e}(P, V) = \hat{e}(P_{\text{pub}}, wQ_{\text{ID}})U$. Now, $\mathcal{B}$ retrieves the respective tuple (m, ID, U, w) from L_2 and computes $V = (wc + x)P_{\text{pub}}$; thus, we have

$$\begin{aligned}\hat{e}(P_{\text{pub}}, wQ_{\text{ID}})U &= \hat{e}\left(P_{\text{pub}}, w(c(bP))\right)\hat{e}(P_{\text{pub}}, P)^x \\ &= \hat{e}(P_{\text{pub}}, wcbP + xP) \\ &= \hat{e}(P, wcabP + xP_{\text{pub}}) \\ &= \hat{e}(P, V) \\ &\Rightarrow V = wcabP + xP_{\text{pub}}.\end{aligned}$$

Now, $\mathcal{B}$ outputs abP as a solution to the CDH instance by computing $abP = w^{-1}c^{-1}(V - xP_{\text{pub}})$. This concludes the description of algorithm $\mathcal{B}$. ■

V. Batch Verifications of Proposed IDS Scheme

This section presents batch verifications of different types for

the proposed IDS scheme. To verify a k-batch signature, $\{(\mathrm{ID}_i, m_i, \sigma_i)\}_{i=1, 2, \ldots, n}$, for $n \le k$, the verifier uses the following batch verify algorithms:

1) For Type 2 batch verifications: In this case, we have $\mathrm{ID} = \mathrm{ID}_1 = \ldots = \mathrm{ID}_n$. The verifier computes $Q_{\mathrm{ID}_i} = H_1(\mathrm{ID}_i) \in G$ and $h_i = H_2(m_i, U_i)$, for $i = 1, 2, \ldots, n$. In addition, the verifier computes $U = \prod_{i=1}^{n} U_i$, where $U_i = g^{r_i}$. The Type 2 batch verification algorithm outputs "1" if the following equation holds; otherwise, it outputs "0":

$$\hat{e}\left(P, \sum_{i=1}^{n} V_i\right) = \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} h_i Q_{\mathrm{ID}}\right) U.$$

2) For Type 3 (or 1) batch verifications: The verifier first computes $Q_{\mathrm{ID}_i} = H_1(\mathrm{ID}_i) \in G$ and $h_i = H_2(m_i, U_i)$, for $i = 1, 2, \ldots, n$. In addition, the verifier computes $U = \prod_{i=1}^{n} U_i$, where $U_i = g^{r_i}$. The Type 3 (or 1) batch verification algorithm outputs "1" if the following equation holds; otherwise, it outputs "0":

$$\hat{e}\left(P, \sum_{i=1}^{n} V_i\right) = \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} h_i Q_{\mathrm{ID}_i}\right) U.$$

One can verify that the batch verifications of the proposed IDS scheme are correct as shown below.

Proof of Correctness. For Type 2 batch verifications, we have the following:

$$\begin{aligned}\hat{e}\left(P, \sum_{i=1}^{n} V_i\right) &= \hat{e}\left(P, \sum_{i=1}^{n} (h_i d_{\mathrm{ID}} + r_i P_{\mathrm{pub}})\right) \\ &= \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} h_i Q_{\mathrm{ID}}\right) \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} r_i P\right). \\ &= \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} h_i Q_{\mathrm{ID}}\right) U.\end{aligned}$$

For Type 3 (or 1) batch verifications, we have the following:

$$\begin{aligned}\hat{e}\left(P, \sum_{i=1}^{n} V_i\right) &= \hat{e}\left(P, \sum_{i=1}^{n} (h_i d_{\mathrm{ID}_i} + r_i P_{\mathrm{pub}})\right) \\ &= \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} (h_i Q_{\mathrm{ID}_i} + r_i P)\right) \\ &= \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} (h_i Q_{\mathrm{ID}_i})\right) \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} r_i P\right) \\ &= \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} h_i Q_{\mathrm{ID}_i}\right) U.\end{aligned}$$

VI. Security Analysis of Batch Verifications of Proposed IDS Scheme

In this section, we will show that the proposed IDS scheme provides k-batch existential unforgeability against adaptive chosen message and ID attacks.

Definition 1. The proposed k-batch IDS scheme offers existential unforgeability under adaptively chosen message and ID attacks if there is no probabilistic polynomial time adversary/forger $\mathcal{A}$ with non-negligible advantage in the following game played between $\mathcal{A}$ and a challenger, $\mathcal{C}$:

1) Setup: This phase is similar to the one in Theorem 1.
2) Queries: Forger $\mathcal{A}$ makes similar queries as in Theorem 1.
3) k-batch forgery: For some integer $n \le k$, the forger $\mathcal{A}$ outputs n signatures $(\mathrm{ID}_i, m_i, \sigma_i)$, for $i = 1, 2, \ldots, n$. Note that there exists at least one index i such that ID_i is not asked the extract query and (ID_i, m_i) in the key extraction oracle and a tuple (ID_i, m_i) is also not asked in the sign query; that is, the forger $\mathcal{A}$ owns at most $(n - 1)$ private keys of n identities. Forger $\mathcal{A}$ wins the game if the batch verify algorithm outputs "1." The advantage of the forger $\mathcal{A}$ is as the probability that $\mathcal{A}$ wins.

1. Security of k-Batch Signature for Type 2

A security proof for Type 2 batch verifications of the proposed IDS scheme is presented below.

Theorem 2. Let $\mathcal{A}$ be a probabilistic polynomial-time forger who can forge the Type 2 k-batch signature of the proposed IDS scheme with a non-negligible advantage under a random oracle paradigm. Then, there is an algorithm $\mathcal{B}$ that can output the given CDH instance with non-negligible advantage in probabilistic polynomial-time.

Proof. Assume that $\mathcal{A}$ is a forger who can forge a Type 2 k-batch signature under adaptively chosen message and ID attacks with a non-negligible advantage. As in Theorem 1, we show that there exists an algorithm $\mathcal{B}$ that solves the given instance of the CDH problem using $\mathcal{A}$. Algorithm $\mathcal{B}$ runs the *setup* algorithm to obtain the public and private keys. The public key is sent to $\mathcal{A}$. As discussed in Theorem 1, $\mathcal{A}$ issues queries and is answered by $\mathcal{B}$.

Algorithm $\mathcal{B}$ obtains the corresponding tuple $(\mathrm{ID}_i, c_i, d_i, v_i)$ from list L_1, declares failure if $d = 1$, and stops. If not, it computes $Q_{\mathrm{ID}} = c(bP)$ for $d = 0$.

The signature $\sigma = (U, V)$ must satisfy the equation $\hat{e}\left(P, \sum_{i=1}^{n} V_i\right) = \hat{e}\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} h_i Q_{\mathrm{ID}}\right) U$.

Now, $\mathcal{B}$ recovers the corresponding tuple $(m_i, \mathrm{ID}, U_i, w_i)$ from list L_2 and computes $V_1 = (w_1 c + x_1) P_{\mathrm{pub}}$. Consider

$\hat{e}(P_{\text{pub}},\ w_1 Q_{\text{ID}})\hat{e}(P_{\text{pub}},\ x_1 P) = \hat{e}(aP,\ w_1 c(bP) + x_1 P)$

$\hat{e}(P,\ w_1 c(abP) + x_1 P_{\text{pub}}) = \hat{e}(P,\ V_1).$

$\Rightarrow V_1 = w_1 c(abP) + x_1 P_{\text{pub}} \Rightarrow w_1 c(abP) = V_1 - x_1 P_{\text{pub}}.$

Now, $\mathcal{B}$ outputs abP as a solution to the CDH instance by computing $abP = w_1^{-1} c^{-1}(V_1 - x_1 P_{\text{pub}})$. ■

2. Security of *k*-Batch Signature for Types 1 and 3

In the following, we prove the security of batch verifications of Types 1 and 3 of the proposed IDS scheme. Notice that a Type 1 batch verification is a subcase of Type 3. Thus, it is enough to prove the security of a *k*-batch signature of Type 3.

Theorem 3. Let $\mathcal{A}$ be a probabilistic polynomial-time forger who can forge a Type 3 *k*-batch signature of the proposed IDS scheme with a non-negligible advantage under a random oracle paradigm. Then, there is an algorithm $\mathcal{B}$ that can output the given CDH instance with non-negligible advantage in probabilistic polynomial-time.

Proof. Let ID_i, for $i = 1, 2, \ldots, n$, denote the identities of distinct signers participating in a signing. From Definition 1, an adversary owns at most $(n - 1)$ private keys of n signers. Assume that there exists a probabilistic polynomial-time adversary $\mathcal{A}$ that can forge a *k*-batch signature of the proposed IDS scheme of Type 3 for adaptively chosen message and *ID* attacks with a non-negligible advantage.

As in Theorem 1, there exists a probabilistic polynomial-time algorithm $\mathcal{B}$ that returns a forged *k*-batch signature of Type 3, σ on messages $\{m_i\}$ under $\{\text{ID}_i\}$, for $i = 1, 2, \ldots, n$, and $\mathcal{A}$ must not have requested a signature on m_1 under ID_1.

Algorithm $\mathcal{B}$ obtains (ID_i, c_i, d_i, v_i) from L_1 and continues if $d_1 = 0$ and $d_i = 1$ for $2 \le i \le n$. If not, then $\mathcal{B}$ declares failure and stops. We have $Q_{\text{ID}_1} = c_1(bP)$ for $d_1 = 0$ and $Q_{\text{ID}_i} = c_i P$ for $d_i = 1$, $i > 1$. The forged Type 3 *k*-batch signature σ must satisfy the equation $\hat{e}\left(P, \sum_{i=1}^{n} V_i\right) = \hat{e}\left(P_{\text{pub}}, \sum_{i=1}^{n} w_i Q_{\text{ID}_i}\right) U$.

Now, $\mathcal{B}$ retrieves the n respective tuples (ID_i, m_i, U_i, w_i) from L_2 and computes $V_i = (w_i c_i + x_i) P_{\text{pub}}$, for $i > 1$; thus, we have $\hat{e}(P, V_i) = \hat{e}(P, (w_i c_i + x_i) P_{\text{pub}}) = \hat{e}(P_{\text{pub}}, w_i Q_{\text{ID}_i}) U_i$, which implies σ_i is valid. Now, $\mathcal{B}$ considers $V_1 = V - \sum_{i=2}^{n} V_i$, and outputs

$$\hat{e}(P,\ V_1) = \hat{e}\left(P,\ V - \sum_{i=2}^{n} V_i\right) = \hat{e}(P,\ w_1 c_1 abP + k_1 P_{\text{pub}}).$$

$$\Rightarrow\ V_1 = w_1 c_1 abP + x_1 P_{\text{pub}} \Rightarrow w_1 c_1 abP = V_1 - x_1 P_{\text{pub}}.$$

Now, $\mathcal{B}$ outputs abP as a solution to the CDH instance by computing $abP = w_1^{-1} c_1^{-1}(V_1 - x_1 P_{\text{pub}})$. ■

VII. Complexity Analysis

In this section, we present the complexity issues and compare the computational efficiency of the proposed IDS scheme supporting batch verifications with related schemes. For comparison, we consider the time-consuming operations. According to [23] and [24], $1T_{\text{p}} \approx 1200t_{\text{m}}$, $1T_{\text{m}} \approx 29t_{\text{m}}$, and $1T_{\text{a}} \approx 0.12t_{\text{m}}$, where T_{a} denotes the time for evaluating a point addition in G, T_{m} denotes the time for evaluating a point scalar multiplication over G, T_{p} denotes the time to compute one pairing operation, and t_{m} denotes the time to perform a modular multiplication in Z_q^*. An efficiency comparison of the proposed IDS scheme supporting batch verifications with related schemes; Yoon and others [11]; and Tseng and others [16] is presented in Table 2.

Compared with the other operations, the pairing evaluation is the most costly in terms of time. Despite the fact that much research has taken place to speed up the pairing computation [22], it is still time consuming. The proposed IDS scheme is efficient when compared to the schemes in [11] and [16] for batch verifications of Types 2 and 3. In particular, for batch verifications of Type 3, the pairing operations in the schemes in [11] and [16] grow linearly with that of the signers, whereas the proposed IDS scheme requires a constant number (only two) of pairing operations irrespective of the number of signers, which reduces greatly the computational complexity. Hence, the proposed IDS scheme supporting batch verifications is more efficient than the related existing schemes.

VIII. Conclusion

In this paper, we have proposed a new, efficient IDS scheme using bilinear pairings supporting batch verifications. We have proved that various types of batch verifications for the

Table 2. Efficiency comparison.

Scheme	Type 2 batch verifications	Types 3 batch verifications
Yoon and others [11]	$2T_p + nT_m + (2n-2)T_a$ $= (29.24n + 2399.76)t_m$	$(n+1)T_p + nT_m + (2n-2)T_a$ $= (1229.24n + 1199.76)t_m$
Tseng and others [16]	$2T_p + nT_m + (2n-2)T_a$ $= (29.24n + 2399.76)t_m$	$(n+1)T_p + nT_m + (2n-2)T_a$ $= (1229.24n + 1199.76)t_m$
Proposed scheme	$2T_p + nT_m$ $= (29n + 2400)t_m$	$2T_p + nT_m + (n-1)T_a$ $= (29.12n + 2399.88)t_m$

proposed IDS scheme are unforgeable under a random oracle paradigm with the assumption that the CDH problem is intractable. In addition, a security reduction of the proposed IDS scheme and its batch verifications has been obtained without the use of a forking lemma [21], and so is tightly related to the CDH problem. For batch verifications of Type 3, the proposed IDS scheme requires a constant number of pairing operations, which greatly improves the computational efficiency. In summary, the performance of our scheme is good, which makes the scheme applicable in practice. Both the security and high efficiency of the batch verifications mean that it is possible to apply them in environments where computational issues are seen as the main constraints, such as in ad-hoc networks. In future, we will extend our batch verification schemes for various forms of anonymous authentication, such as group signatures, e-cash, e-voting, intelligent cars to control traffic, and anonymous credentials.

References

[1] A. Fiat, "Batch RSA," *Adv. Cryptology*, Santa Barbara, CA, USA, Aug. 20–24, 1989, pp. 175–185.

[2] C.H. Lim and P.J. Lee, "Security of Interactive DSA Batch Verification," *IEEE Electron. Lett.*, vol. 30, no. 19, 1994, pp. 1592–1593.

[3] S.M. Yen and C.S. Laih, "Improved Digital Signature Suitable for Batch Verification," *IEEE Trans. Comput.*, vol. 44, no. 7, 1995, pp. 957–959.

[4] M. Bellare, J.A. Garay, and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures," *Int. Conf. Theory Appl. Cryptographic Techn.*, Espoo, Finland, May 31–June 4, 1998, pp. 236–250.

[5] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Adv. Cryptology*, Santa Barbara, CA, USA, Aug. 19–22, 1984, pp. 47–53.

[6] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Adv. Cryptology*, Santa Barbara, CA, USA, Aug. 19–23, 2001, pp. 213–229.

[7] J.C. Cha and J.H. Cheon, "An Identity-Based Signature Scheme from Gap Diffie-Hellman Groups," *Int. Workshop Practice Theory Public Key Cryptography*, Miami, FL, USA, Jan. 6–8, 2003, pp. 18–30.

[8] F. Hess, "Efficient Identity Based Signature Schemes Based on Pairings," *Sel. Areas Cryptography*, St. John's, Canada, Aug. 15–16, 2002, pp. 310–324.

[9] K.G. Paterson, "ID-Based Signatures from Pairings on Elliptic Curves," *IEEE Electron. Lett.*, vol. 38, no. 18, 2002, pp. 1025–1026.

[10] P.V.S.S.N. Gopal, P.V. Reddy, and T. Gowri, "New Identity Based Signature Scheme Using Bilinear Pairings over Elliptic Curves," *IEEE Int. Adv. Comput. Conf.*, Ghaziabad, India, Feb. 22–23, 2013, pp. 361–365.

[11] H. Yoon, J.H. Cheon, and Y. Kim, "Batch Verifications with ID-Based Signatures," *Int. Conf. Inf. Security Cryptology*, Seoul, Rep. of Korea, Dec. 2–3, 2004, pp. 233–248.

[12] T. Cao, D. Lin, and R. Xue, "Security Analysis of Some Batch Verifying Signatures from Pairings," *Int. J. Netw. Security*, vol. 3, no. 2, 2006, pp. 138–143.

[13] S. Cui, P. Duan, and C.W. Chan, "An Efficient Identity-Based Signature Scheme with Batch Verifications," *Int. Conf. Scalable Inf. Syst.*, Hong Kong, China, May 29–June 1, 2006, vol. 152, no. 22.

[14] H.F. Chiang, S.M. Yen, and H.C. Lin, "Security Analysis of Batch Verification on Identity-Based Signature Schemes," *WSEAS Int. Conf. Comput.*, Crete Island, Greece, July 26–28, 2007, pp. 50–55.

[15] C. Zhang et al., "An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks," *IEEE Conf. Comput. Commun.*, Phoenix, AZ, USA, Apr. 15–17, 2008, pp. 816–824.

[16] Y.M. Tseng, T.Y. Wu, and J.D. Wu, "Towards Efficient ID-Based Signature Schemes with Batch Verifications from Bilinear Pairings," *IEEE Int. Conf. Availability, Rel. Security*, Fukuoka, Japan, Mar. 16–19, 2009, pp. 935–940.

[17] J.Y. Hwang et al., "New Efficient Batch Verification for an Identity-Based Signature Scheme," *Security Commun. Netw.*, vol. 8, no. 15, Oct. 2015, pp. 2524–2535.

[18] Y. Ren et al., "An Efficient Batch Verifying Scheme for Detecting Illegal Signatures," *Int. J. Netw. Security*, vol. 17, no. 4, Jan. 2015, pp. 463–470.

[19] E.J. Goh and S. Jarecki, "A Signature Scheme as Secure as the Diffie-Hellman Problem," *Int. Conf. Theory Appl. Cryptographic Techn.*, Warsaw, Poland, May 4–8, 2003, pp. 401–415.

[20] J. Katz and N. Wang, "Efficiency Improvements for Signature Schemes with Tight Security Reductions," *ACM Conf. Comput. Commun. Security*, Washington, DC, USA, Oct. 27–30, 2003, pp. 155–164.

[21] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," *J. Cryptology*, vol. 13, no. 3, June 2000, pp. 361–396.

[22] P.S.L.M. Barreto et al., "Efficient Algorithms for Pairing-Based Cryptosystems," *Adv. Cryptology*, Santa Barbara, CA, USA, Aug. 18–22, 2002, pp. 354–369.

[23] N. Koblitz, A. Menezes, and S. Vanstone, "The State of Elliptic Curve Cryptography," *Des., Codes, Cryptography*, vol. 19, no. 2, Mar. 2000, pp. 173–193.

[24] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, USA: CRC Press, LLC, 1996.

P.V.S.S.N. Gopal received his MS degree in applied mathematics and MPhil in (commutative algebra) mathematics from Pondicherry University, Puducherry, India, in 2001 and 2008, respectively. He received his PhD degree in applied mathematics from Andhra University, Visakhapatnam, India, in 2015. His research interests include abstract algebra, linear algebra, number theory, and elliptic curve cryptography. He is a lifetime member of the Cryptology Research Society of India.

P. Vasudeva Reddy received his MS and PhD degrees in mathematics from Sri Venkateswara University, Tirupati, India, in 1998 and 2006, respectively. He received his MTech degree in computer science and technology-networks from Andhra University, Visakhapatnam, India, in 2010. He is currently working as a professor with the Department of Engineering Mathematics, Andhra University. His research interests include algebra & number theory applications and cryptography. He has several publications in national and international reputed journals. He is an associate editor for the International Journal of Cryptography and Security. He is a member of the International Association of Engineers, and lifetime member of both the Cryptology Research Society of India and the Indian Mathematical Society.

T. Gowri received her BTech degree in electronics and communications engineering from Nagarjuna University, Guntur, India, in 2000 and her MTech degree in electronics and communications engineering from Jawaharlal Nehru Technological University (A), Anantapur, India, in 2006. She is currently working as an assistant professor with the Department of Electronics and Communication Engineering, Gandhi Institute of Technology and Management University, Visakhapatnam, India. Her research interests include computer electronics, digital signal processing, and information security.

ETRI Journal, Volume 38, Number 2, April 2016
http://dx.doi.org/10.4218/etrij.16.0114.1193