Journal of King Saud University – Computer and Information Sciences (2017) 29, 303–310

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Identity-based key-insulated aggregate signature scheme

P. Vasudeva Reddy *, P.V.S.S.N. Gopal

Department of Engineering Mathematics, Andhra University, Visakhapatnam, India

Received 11 December 2014; revised 5 July 2015; accepted 8 September 2015
Available online 31 October 2015

KEYWORDS

ID-based signature;
Key-insulated mechanism;
Aggregate signature;
Bilinear pairings;
CDH problem;
Unforgeability

Abstract Private key exposure can be the most devastating attack on cryptographic schemes; as such exposure leads to the breakage of security of the scheme as a whole. In the real world scenario, this problem is perhaps the biggest threat to cryptography. The threat is increasing with users operating on low computational devices (e.g. mobile devices) which hold the corresponding private key for generating signatures. To reduce the damage caused by the key exposure problem in aggregate signatures and preserve the benefits of identity-based (ID-based) cryptography, we hereby propose the first key-insulated aggregate signature scheme in ID-based setting. In this scheme the leakage of temporary private keys will not compromise the security of all the remaining time periods. The security of our scheme is proven secure in the random oracle paradigm with the assumption that the Computational Diffie–Hellman (CDH) problem is intractable. The proposed scheme allows an efficient verification with constant signature size, independent of the number of signers.

1. Introduction

In cryptographic schemes, the private keys are to be kept securely as the exposure of private keys leads to breakage in the security of the whole scheme. This problem can be the most devastating while performing cryptographic operations on low computational devices e.g. mobile phones, since it is easy to obtain the private information from a stolen device rather than breaking the hard problem on which the device's security is based. To overcome the problem of key exposure in cryptographic schemes and to reduce the damage caused by such problems, Dodis et al. (2002) came up with the idea of a key-insulated cryptosystem. Based on this idea some PKI based key-insulated signature schemes (Dodis et al., 2003; González-Deleito et al., 2004; Wang et al., 2004) appeared in the literature.

In Dodis et al. (2002), the lifetime of private keys is divided into discrete time periods $1, 2, \ldots, t$, including a physically secure but computationally limited device termed as helper or base, and the full fledged private key is separated into two parts: the temporary key and the helper key. The former performs the cryptographic operations on a powerful but insecure device, while the latter is stored on the helper. The user has to update their temporary key on every occasion, whereas the public key remains unaltered throughout the life time of the

* Corresponding author.
E-mail addresses: vasucrypto@yahoo.com (P. Vasudeva Reddy), gopalcrypto786@gmail.com (P.V.S.S.N. Gopal).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2015.09.003

system. In the beginning of each time period, the user combines the partial private key with the temporary private key for the previous period. Therefore, the exposure of the temporary private key in a time period will not let a forger to derive other temporary private keys for the remaining periods. Thus it is desirable to deal with the key exposure problem in ID-based cryptosystems.

The concept of ID-based cryptography was introduced by Shamir (1985). In this cryptosystem, the public key of a user can be obtained directly from their identity such as the I.P. address driving license number, etc. The user's private key is generated by a trusted authority termed as Key Generation Centre (KGC). Since then many signature schemes in the ID-based setting appeared in the literature but a practical ID-based encryption scheme using Weil pairing was devised by Boneh and Franklin (1985). Based on the work done by Boneh and Franklin (1985) some ID-based signature schemes using pairings appeared in the literature (Cha and Cheon, 2003; Gopal et al., 2012; Hess, 2002; Paterson, 2002).

Zhou et al. (2006) put forth the first ID-based key-insulated signature scheme, but it did not satisfy the strong key-insulated property. The strong key-insulated property means even if the helper is corrupted by a forger, the forger still would not be able to compute the private keys of a user. Weng et al. (2006a) and Li et al. (2006) proposed strong ID-based key-insulated signature schemes. Wu et al. (2012) proposed an ID-based key-insulated signature scheme that supports batch verifications. Later, some key-insulated signature schemes with special properties (Chen et al., 2014; Chen et al., 2011; Wang et al., 2013) in the random oracle paradigm and some schemes in the standard model (Wan, 2011; Weng et al., 2006b) appeared in the literature.

An aggregate signature is a digital signature obtained upon compressing n different signatures signed by n different users on n different messages. Such a signature effectively reduces the computational cost as well as the communication bandwidth. This type of compressed signature was introduced by Boneh et al. (2003). Since then many aggregate signature schemes in PKI as well as ID-based (Gentry and Ramzan, 2006; Wang et al., 2008; Xu et al., 2005; Yuan et al., 2014; Yu et al., 2011) settings appeared in the literature.

The key exposure in the low power devices such as mobile devices has become an unavoidable threat in insecure environments. If the private key of a participant in aggregate signature is compromised then the whole aggregation becomes untrustworthy. So, reducing the damage caused by key-exposure in aggregate signature is of great importance. To solve the key exposure problem in aggregate signatures and maintain the merits of aggregate signatures, Zhao et al. (2014) presented the concept of key-insulated aggregate signature scheme in PKI based setting.

In this paper we propose an ID-based key-insulated aggregate signature (IDKIAS) scheme using bilinear pairings over elliptic curves. This is the first key insulated aggregate signature scheme in ID-based setting. Our scheme is efficient in terms of computational and communication overhead, and is proven secure in the random oracle paradigm under CDH assumption.

The rest of the paper has been organized as follows: Section 2 deals with the preliminaries and computational hard problems. Notations and their description, syntax and security model for the proposed scheme are presented in Section 3. The proposed IDKIAS scheme and its schematic diagram are presented in Section 4. Proof of correctness and security proof of our scheme are presented in Section 5. Efficiency analysis of the proposed IDKIAS scheme is presented in Section 6, and finally, Section 7 deals with the conclusion.

2. Preliminaries

This section summarizes some fundamental concepts and necessary hard problems.

2.1. Bilinear map

Let $(G,+)$ and $(G_T,\cdot)$ be cyclic groups such that $|G| = |G_T| = q$, with P as a generator in G. A map $\hat{e} : G \times G \to G_T$ is called bilinear if it satisfies the following properties:

1. *Bilinear:* $\forall A, B \in G$, $\forall x, y \in Z_q^*$, $\hat{e}(xA, yB) = \hat{e}(A,B)^{xy}$.
2. *Non-degeneracy:* $\exists A \in G$, $\hat{e}(A,A) \neq 1$.
3. *Computable:* $\forall A, B \in G$, $\hat{e}(A,B)$ can be 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 (Barreto et al., 2002; Boneh and Franklin, 1985).

2.2. Complexity assumptions

In the following, we present some necessary hard problems on which the proposed scheme's security is based (Boneh and Boyen, 2004; Boneh and Franklin, 1985).

- *Computational Diffie–Hellman (CDH) problem:* $\forall x, y \in Z_q^*$, given $P, xP, yP \in G$ evaluate $xyP \in G$. For a polynomial-time adversary/forger $\mathcal{A}$, the advantage of $\mathcal{A}$ is defined as the running time T against the CDH problem in G as $Adv_{CDH}(T) = \Pr[\mathcal{A}(P, xP, yP) = xyP / P, xP, yP \in G]$.
- *Computational Diffie–Hellman (CDH) assumption:* For any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A},G}^{CDH}$ is negligibly small.
- *Decision Diffie–Hellman (DDH) problem:* $\forall x, y, z \in Z_q^*$, given $P, xP, yP, zP \in G$ decide whether $z = xy$. If so, the tuple (P, xP, yP, zP) is called a valid Diffie–Hellman tuple.
- *Gap Diffie–Hellman (GDH) group:* A group G is said to be a GDH group if there is a probabilistic polynomial time algorithm to evaluate the DDH problem but such algorithm do not exist to evaluate the CDH problem.

3. Syntax and security model of the proposed IDKIAS scheme

In this section, we present the notations and their description in Table 1, and then the syntax and security model for the proposed IDKIAS scheme.

In Table 1, we present the notations and their description used in the design of our proposed scheme

Table 1 Notations and their description used in the proposed scheme.

Notations	Description
$\{P_i\}_{i=1,2,\ldots,n}$	An aggregate group of signers
P_i	A signer in the aggregate group of signers $\{P_i\}_{i=1,2,\ldots,n}$
$\{M_i\}_{i=1,2,\ldots,n}$	An aggregate group of messages
M_i	A message in the aggregate group of messages $\{M_i\}_{i=1,2,\ldots,n}$
ID_i	The identity of a signer P_i
$d_{ID_i,0}$	The initial private key of ID_i
$HPK_{ID_i,t}$	The helper private key for ID_i in a time period 't'
$d_{ID_i,t}$	The temporary private key (updated key) for ID_i in a time period 't'
σ_i	A signature on the message M_i by the signer P_i with identity ID_i
$\{\sigma_i\}_{i=1,2,\ldots,n}$	An aggregate group of signatures
(σ, t)	An aggregate signature σ in a time period 't'

3.1. Syntax of the proposed IDKIAS scheme

An IDKIAS scheme involves the KGC, an aggregating set L of n users/signers $\{P_i\}_{i=1,2,\ldots,n}$ and an aggregate signature. The proposed IDKIAS scheme comprises seven polynomial time algorithms: Setup, Initial Private Key Generation, Key Update, Signature Generation, Signature Verification, Aggregate Signature and Aggregate Signature Verification.

1. *Setup:* It is a probabilistic algorithm performed by the KGC, takes a security parameter l and the total number of time periods t as input. The KGC returns the public parameters as *Params*, the system's master private key $\langle s \rangle$ and the user's helper private key $\langle hpk \rangle$. *Params* are made public, i.e. known to all, where as $\langle s \rangle$ and $\langle hpk \rangle$ are kept secret. *Params* are implicit input for the remaining algorithms.
2. *Initial Private Key Generation:* It is a probabilistic algorithm performed by the KGC. This algorithm takes the master private key $\langle s \rangle$, a user's identity $ID_i \in \{0,1\}^*$ for $i = 1, 2, \ldots, n$ and the *Params*, as input; and returns the user's initial private key $d_{ID_i,0}$ for the initial time period '0'.
3. *Key Update:* This algorithm consists of two deterministic algorithms.
   - *Helper Key Update:* This algorithm is performed by the helper, takes as input the helper private key $\langle hpk \rangle$, a user's identity ID_i, and a time period t; and returns the helper key $HPK_{ID_i,t}$ for the time period t.
   - *User Private Key Update:* This algorithm is performed by the user with identity ID_i takes as input, an update helper key $HPK_{ID_i,t}$, a temporary private key $d_{ID_i,t-1}$ for the time period $t-1$, and a time period t, and returns $d_{ID_i,t}$ as user's temporary private key for the time period t.
4. *Signature Generation:* For obtaining the signature on a message M_i, in a time period t, the user $ID_i \in L$ submits $ID_i, d_{ID_i,t}$, a message M_i, *Params* and a time period t, to this algorithm as input; and outputs (σ_i, t) as a valid signature.
5. *Signature Verification:* This algorithm takes a signature σ_i on a message M_i for a time period t by the user with identity ID_i as input; and verifies whether (σ_i, t) is valid or not. It outputs 'accept' if valid, or 'reject' otherwise.
6. *Aggregate Signature:* On receiving different n signatures $\{\sigma_i\}_{i=1,2,\ldots,n}$ along with n identities, message pairs $(ID_i, M_i)_{i=1,2,\ldots,n}$ for a time period t, as input; anyone among the signers or a third party, can output (σ, t) as an aggregate signature for a time period t by running this algorithm.
7. *Aggregate Signature Verification:* This algorithm takes an aggregate signature (σ, t) for a time period t, the n identities and message pairs $(ID_i, M_i)_{i=1,2,\ldots,n}$ as input; and verifies whether (σ, t) is valid or not. If valid, it outputs 'accept', else outputs 'reject'.

3.2. Security model of the proposed IDKIAS scheme

In this, we define the security model of the proposed IDKIAS scheme.

In this model, the following game is played between the forger $\mathcal{A}$ and the challenger $\mathcal{C}$. $\mathcal{A}$ is given a single ID and is permitted to pick identities of their choice except the challenge ID. $\mathcal{A}$ can access to the signing oracle with respect to the challenge ID.

The advantage of $\mathcal{A}$ denoted $Adv_{IDKIAS,\mathcal{A}}$ is the probability of success taken over the coin flips of $\mathcal{A}$ and the key extraction oracle.

Definition 1: In the aggregate model, the aggregate forger $\mathcal{A}$ is said to $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{KE}, q_S, \varepsilon, N, T)$ – break a N-user key-insulated aggregate signature scheme for a time period t if: $\mathcal{A}$ runs in time at most T, $\mathcal{A}$ makes at most $q_{H_1} + q_{H_2} + q_{H_3}$ hash queries, and q_{KE} private key extraction queries and q_S signature queries with $Adv_{IDKIAS,\mathcal{A}}$ is at least ε. An IDKIAS scheme is $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{KE}, q_S, \varepsilon, N, T)$ – key-insulated security against existential forgery under adaptively chosen message and ID attacks in the aggregate model if no such forger $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{KE}, q_S, \varepsilon, N, T)$ – breaks it.

Setup: $\mathcal{C}$ runs the setup algorithm to obtain the *Params*, system's master private key $\langle s \rangle$ and helper's private key $\langle hpk \rangle$. $\mathcal{A}$ is given a randomly generated identity ID_1. $\mathcal{C}$ forwards *Params*, to $\mathcal{A}$ and keeps $\langle s \rangle$ with itself.

Queries: Following are the queries made by $\mathcal{A}$ adaptively:

1. *Initial private key query:* $\mathcal{A}$ queries the initial private key for an identity $ID_i \neq ID_1$ of a user. $\mathcal{C}$ runs the initial private key generation algorithm to generate an initial private key corresponding to ID_i and sends it to $\mathcal{A}$.
2. *Helper key update query:* $\mathcal{A}$ sends a pair (ID_i, t) to $\mathcal{C}$ and queries for the helper key for a time period t. $\mathcal{C}$ runs the helper key update algorithm to generate a helper private key for ID_i, for a time period t and sends it to $\mathcal{A}$.
3. *User private key update query:* $\mathcal{A}$ sends a pair (ID_i, t) to $\mathcal{C}$ and queries for the user's private key for $ID_i \neq ID_1$ for a time period t. $\mathcal{C}$ runs the user's private key update algorithm to generate a temporary private key of user with identity ID_i for a time period t and sends it to $\mathcal{A}$.
4. *Signature query:* $\mathcal{A}$ sends a tuple (ID_i, M_i, t) to $\mathcal{C}$ and queries for the signature on M_i of ID_i in a time period t. $\mathcal{C}$ runs the signature generation algorithm and sends (σ_i, t) as the queried signature for a time period t to $\mathcal{A}$.

Forgery: Eventually, $\mathcal{A}$ outputs $n-1$ additional identities, $\{ID_i\}_{i=2,3,\ldots,n}$ where n is a game parameter and is at most N. Also, $\mathcal{A}$ can output an aggregate signature (σ^*, t^*) corresponding to the n identities, on the n respective messages $\{M_i\}_{i=1,2,\ldots,n}$ for a time period t^*. $\mathcal{A}$ wins if σ is a valid aggregate signature on messages $\{M_i\}_{i=1,2,\ldots,n}$ under $\{ID_i\}_{i=1,2,\ldots,n}$ respectively and $\mathcal{A}$ never requests for a signature on M_1 under ID_1.

$\mathcal{A}$'s advantage is defined as the probability of producing a forgery taken over the coin flips of $\mathcal{C}$ and $\mathcal{A}$. $\mathcal{A}$ wins the game, if the aggregate verification algorithm outputs '1' and the following conditions are valid:

(i) $Verify((\sigma^*, t^*), M^*, ID^*) = 1$.
(ii) The messages $\{M_i\}_{i=1,2,\ldots,n}$ are all distinct.
(iii) $\mathcal{A}$ never queried a signature on M_1^* under ID_1^* for a time period t^*.

4. The proposed ID-based key-insulated aggregate signature (IDKIAS) scheme

As mentioned in Section 3.2, the proposed IDKIAS scheme comprises seven polynomial time algorithms described as follows:

1. *Setup:* For a given security parameter l, the KGC runs this algorithm as follows:
   - Generates two cyclic groups $(G,+),(G_T,\cdot)$ such that $|G| = |G_T| = q \geqslant 2^l$.
   - Generates a generator $P \in G$ and an admissible bilinear map $\hat{e} : G \times G \to G_T$.
   - Generates two integers $s, hpk \in Z_q^*$ at random and computes $P_{pub} = sP, P_{hlp} = hpkP$ as the system's, helper's public keys respectively and also computes $g = \hat{e}(P_{pub}, P)$.
   - Picks hash functions $H_1, H_2 : \{0,1\}^* \to G$, and $H_3 : \{0,1\}^* \times G_T \to Z_q^*$.
   - Publishes the system's public parameters as $Params =< G, G_T, \hat{e}, q, P, P_{pub}, P_{hlp}, H_1, H_2, H_3, g >$ and keeps the system's private key $\langle s \rangle$ and helper's private key $\langle hpk \rangle$ with itself securely.

2. *Initial Private Key Generation:* When a user submits their identity ID_i to the KGC, it computes $d_{ID_i,0} = sH_1(ID_i) + hpkH_2(ID_i, 0)$. Then KGC sends $d_{ID_i,0}$ to the user as their initial private key and hpk to the helper as the helper private key via a secure channel.
3. *Key Update:*
   - *Helper Key Update:* At the start of the time period t, a helper computes a helper key $HPK_{ID_i,t} = hpk[H_2(ID_i, t) - H_2(ID_i, t-1)]$ and sends it to the user with identity ID_i.
   - *User Key Update:* Upon receiving the helper's key $HPK_{ID_i,t}$, the user ID_i updates their private key as $d_{ID_i,t} = HPK_{ID_i,t} + d_{ID_i,t-1}$. Subsequently, the user ID_i removes the two values $HPK_{ID_i,t}$ and $d_{ID_i,t-1}$.

4. *Signature Generation:* The user P_i submits their identity ID_i, $Params$, messages M_i, private key $d_{ID_i,t}$ for a time period t, as input to this algorithm and performs the following:
   - Picks an integer $r_i \in Z_q^*$ at random, and computes $U_i = g^{r_i} \in G_T$, $h_i = H_3(M_i, ID_i, U_i, t) \in Z_q^*$, and $V_i = h_i d_{ID_i,t} + r_i P_{pub} \in G$.
   - Outputs $\sigma_i = (U_i, V_i) \in G_T \times G$ as the signature on M_i by the user with ID_i, for the time period t.

5. *Signature Verification:* Any user can run this algorithm which takes the messages, identities, pairs $\langle M_i, ID_i \rangle$, and the signature σ_i as input. The verification for a time period t is done as follows:
   - Computes $h_i = H_3(M_i, ID_i, U_i, t) \in Z_q^*$.
   - Verifies $\hat{e}(P, V_i) = \hat{e}(P_{hlp}, h_i H_2(ID_i, t))\hat{e}(P_{pub}, h_i H_1(ID_i))U_i$ holds or not. If it holds, 'accept' the signature, 'reject' if not.

6. *Aggregate Signature:* Each user $\{P_i\}$, for $i = 1, 2, \ldots, n$, submits their signature (σ_i, t) in a time period t.Any designated user computes $U = \Pi_{i=1}^n U_i, V = \sum_{i=1}^n V_i$ and outputs $\sigma = (U, V) \in G_T \times G$ as the aggregate signature for the time period t.
7. *Aggregate Signature Verification:* Any user/verifier can verify the aggregate signature (σ, t) for a time period t as follows:
   - First computes $H_1(ID_i), H_2(ID_i, t)$ and $h_i = H_3(M_i, ID_i, U_i, t) \in Z_q^*$, for $i = 1, 2, \ldots, n$.
   - Verifies $\hat{e}(P, V) = \hat{e}(P_{hlp}, \sum_{i=1}^n h_i H_2(ID_i, t))\hat{e}$ $](P_{pub}, \sum_{i=1}^n h_i H_1(ID_i))U$ holds or not. If it holds, 'accept' the signature, if not 'reject'.

The proposed scheme can be depicted as a schematic diagram in Fig. 1 as follows:

5. Security of the proposed IDKIAS scheme

This section presents proof of correctness and security proof of the proposed IDKIAS scheme.

5.1. Proof of correctness

For single signature:

$$\begin{aligned}\hat{e}(P, V_i) &= \hat{e}(P, h_i d_{ID_i,t} + r_i P_{pub}) \\ &= \hat{e}(P, h_i[HPK_{ID_i,t} + d_{ID_i,t-1}] + r_i P_{pub}) \\ &= \hat{e}(P, h_i hpk H_2(ID_i, t))\hat{e}(P, h_i s H_1(ID_i))\hat{e}(P, r_i P_{pub}) \\ &= \hat{e}(P_{hlp}, h_i H_2(ID_i, t))\hat{e}(P_{pub}, h_i H_1(ID_i))U_i.\end{aligned}$$

For aggregate signature:

$$\begin{aligned}\hat{e}(P, V) &= \hat{e}\left(P, \sum_{i=1}^{n}(h_i d_{ID_i,t} + r_i P_{pub})\right) \\ &= \hat{e}\left(P, \sum(h_i[HPK_{ID_i,t} + d_{ID_i,t-1}] + r_i P_{pub})\right) \\ &= \hat{e}\left(P, \sum h_i hpk H_2(ID_i, t)\right)\hat{e}\left(P, \sum h_i s H_1(ID_i)\right) \\ &\quad \times \hat{e}\left(P, \sum r_i P_{pub}\right) \\ &= \hat{e}\left(P_{hlp}, \sum h_i H_2(ID_i, t)\right)\hat{e}\left(P_{pub}, \sum h_i H_1(ID_i)\right)U.\end{aligned}$$

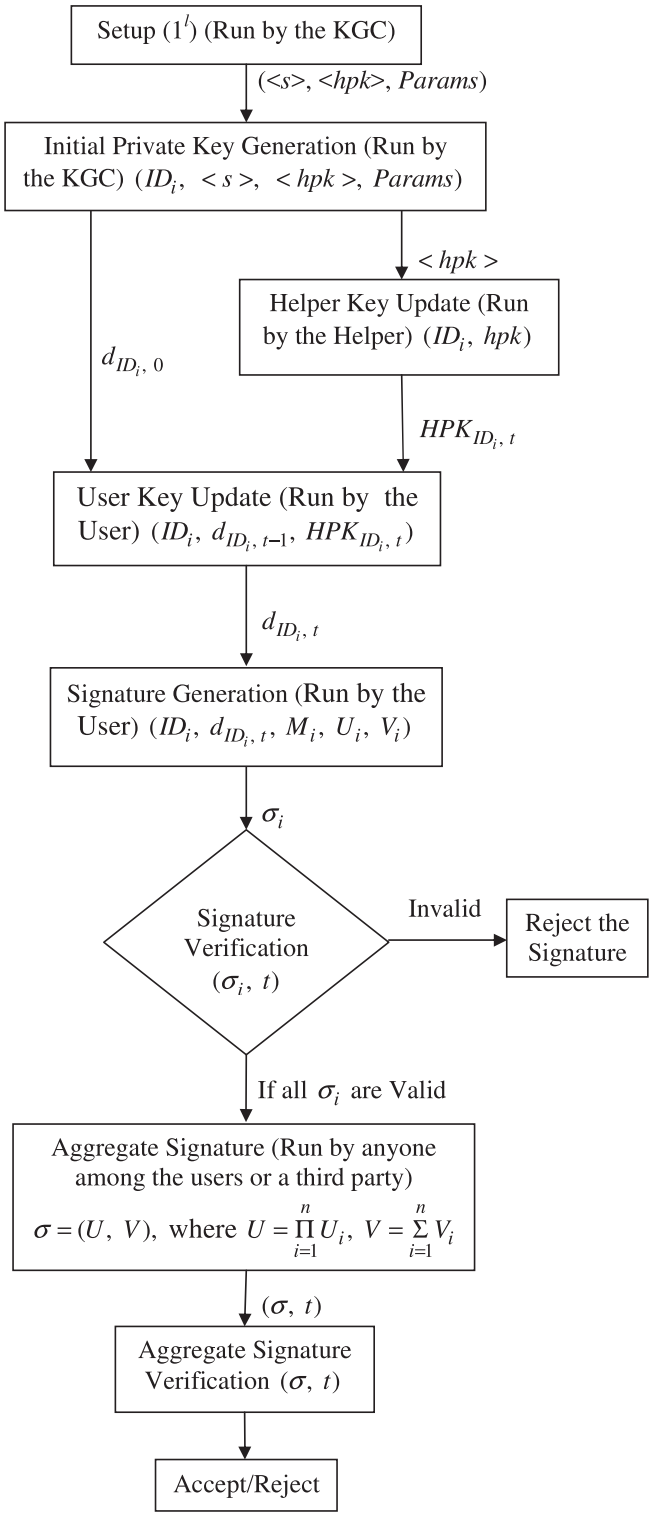

Figure 1 Schematic diagram of the proposed IDKIAS scheme.

5.2. Security proof

Theorem 1: Consider the GDH group G of prime order q. Then our IDKIAS scheme is $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{KE}, q_S, \varepsilon, N, T)$ – secure in G against existential forgery under chosen message and ID attacks in the aggregate model for any T and ε satisfying $T \leqslant T' - T_m(q_{H_1} + q_{H_2} + 2q_{KE} + 4q_S + 3N + 4)$, $\varepsilon \geqslant e(q_{KE} + q_S + N)\varepsilon'$. Here e denotes the base of natural logarithms and T_m denotes the time for computing a scalar multiplication in G.

Proof: Let $\mathcal{A}$ be a forger algorithm which breaks the proposed IDKIAS scheme. We show how to construct a T' – time algorithm $\mathcal{B}$ that solves the CDH instance in G with probability at least ε'. This will contradict the fact that G is a (T', ε') – GDH group.

Algorithm $\mathcal{B}$ is provided with $P, aP, bP \in G$ and its goal is to output $abP \in G$. For this, $\mathcal{B}$ simulates the challenger $\mathcal{C}$ and interacts with $\mathcal{A}$ as follows.

Setup: $\mathcal{B}$ chooses $P_{pub} = aP$ as system's overall public key and $P_{hlp} = hpkP$ as the helper's public key and provides $\mathcal{A}$ a randomly generated identity ID_1.

To respond to the queries to oracles H_1, H_2, H_3, helper key, initial key and sign made by $\mathcal{A}, \mathcal{B}$ proceed as follows:

– H_1 – **Queries:** $\mathcal{B}$ keeps a list L_1, which is empty initially, of tuples (ID_i, c_i, d_i, v_i) to respond to H_1 queries made by $\mathcal{A}$. Upon receiving a query for $ID \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:
1. If L_1 consists of the queried ID, then $\mathcal{B}$ responds with $H_1(ID) = v \in G$.
2. If not, $\mathcal{B}$ flips a coin $d \in \{0,1\}$ generated at random, which outputs '0' with probability μ and '1' with $1-\mu$. (μ is to be found later).
3. Now, $\mathcal{B}$ picks $c \in Z_q^*$ at random and computes $v = c(bP) \in G$, for $d = 0$ and $v = xP \in G$ for $d = 1$.
4. $\mathcal{B}$ adds (ID, c, d, v) to the list L_1 and returns $H_1(ID) = v \in G$ to $\mathcal{A}$.

– H_2 – **Queries:** $\mathcal{B}$ keeps a list L_2, which is empty initially. Upon receiving a query for the tuple (ID_i, t), made by $\mathcal{A}$, $\mathcal{B}$ verifies the list L_2 for this input. If L_2 is with the queried tuple, then $\mathcal{B}$ outputs the earlier defined value for this input. If not, $\mathcal{B}$ picks an integer $w \in Z_q^*$ at random, computes $H_2(ID_i, t) = wP \in G$ and inserts (ID_i, t, w, wP) in L_2.

– H_3 – **Queries:** $\mathcal{B}$ keeps a list L_3, which is empty initially, of tuples (ID_i, M_i, U_i, v', t) to respond to H_3 queries made by $\mathcal{A}$. Upon receiving a query on tuple (ID, M, U, t) from $\mathcal{A}$ to the H_3 oracle, $\mathcal{B}$ proceeds as follows:
1. If the list L_3 is with the queried tuple (ID, M, U, t), then $\mathcal{B}$ provides $H_3(ID, M, U, t) = v' \in Z_q^*$.
2. If not, $\mathcal{B}$ picks an integer $v' \in Z_q^*$ at random, inserts (ID, M, U, t, v') into L_3 and returns $H_3(ID, M, U, t) = v'$ to $\mathcal{A}$.

– *Initial key extraction queries:* $\mathcal{B}$ keeps a list L_4, which is empty initially. Upon receiving an initial private key query corresponding to ID_i made by $\mathcal{A}$, $\mathcal{B}$ recovers the respective tuple (ID_i, c_i, d_i, v_i) from L_1 and proceeds as follows:
1. It outputs 'failure' and halts, for $d = 0$.
2. If not, computes and returns $d_{ID_i,0} = cP_{pub} = a(cP) \in G$ to $\mathcal{A}$. Now, $\mathcal{B}$ inserts the tuple $(ID_i, d_{ID_i,0})$ in L_4.

– *Helper key extraction queries:* $\mathcal{B}$ keeps a list L_5, which is empty initially. Upon receiving a helper key query for ID_i in a time period t, made by $\mathcal{A}$, $\mathcal{B}$ retrieves the respective tuple (ID_i, t, w, wP) for L_2, computes $HPK_{ID_i,t} = wP_{hlp}$ and returns to $\mathcal{A}$.
Now, $\mathcal{B}$ inserts the tuple $(ID_i, HPK_{ID_i,t})$ in L_5.

– *Signature queries:* Upon receiving the signature query on a message M_i under ID_i for a time period t, from $\mathcal{A}$, $\mathcal{B}$ does the following:
1. $\mathcal{B}$ retrieves the tuple (ID_i, c_i, d_i, v_i) from L_1, picks an integer $k \in Z_q^*$ at random and computes $U = g^k$.
2. If L_3 contains (ID_i, M_i, U_i, t, v'), $\mathcal{B}$ picks $v'' \in Z_q^*$ and tries again, i.e. $\mathcal{B}$ inserts (ID_i, M_i, U_i, t, v'') into L_3. Now, $\mathcal{B}$ computes $V = (v'c_i + k_i)P_{pub} + v'wP_{hlp}$ and returns $\sigma = (U, V)$ to $\mathcal{A}$ as the queried signature.

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

$$\begin{aligned}\hat{e}(P,V) &= \hat{e}(P,(v'c_i+k_i)P_{pub}+v'wP_{hlp})\\ &= \hat{e}(P,v'c_iP_{pub})\hat{e}(P,k_iP_{pub})\hat{e}(P,v'wP_{hlp})\\ &= \hat{e}(aP,v'c_iP)\hat{e}(hpkP,v'wP)\hat{e}(aP,P)^{k_i}\\ &= \hat{e}(P_{pub},v'H_1(ID_i))\hat{e}(P_{hlp},v'H_2(ID_i,t))U.\end{aligned}$$

– *Forgery:* Eventually, $\mathcal{A}$ stops by conceding failure, as does $\mathcal{B}$ or returns an aggregate forgery σ^* for a time period t^*, on the set of messages.

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 $2 \leqslant i \leqslant n$. If not, $\mathcal{B}$ declares failure and stops. We have $H_1(ID_1^*) = c_1^*(bP)$, for $d_1^* = 0$ and $H_1(ID_i^*) = c_i^*P$, for $d_i^* = 1, i > 1$. This forged aggregate signature $\sigma*$ must satisfy $\hat{e}(P,V^*) = \hat{e}(P_{hlp}, \sum_{i=1}^n h_i^* H_2(ID_i^*, t))\hat{e}(P_{pub}, \sum_{i=1}^n h_i^* H_1(ID_i^*))U^*$.

Now $\mathcal{B}$ retrieves the n respective tuples $(ID_i^*, M_i^*, U_i^*, t^*, v_i'^*)$ from L_3 and computes $V_i^* = (v_i'^* c_i^* + k_i^*)P_{pub} + v_i'^* w_i^* P_{hlp}$ for $i > 1$, we have

$$\begin{aligned}\hat{e}(P,V_i^*) &= \hat{e}(P,(v_i'^*c_i^*+k_i^*)P_{pub}+v_i'^*w_i^*P_{hlp})\\ &= \hat{e}(aP,v_i'^*c_i^*P)\hat{e}(hskP,v_i'^*w_i^*P)\hat{e}(aP,P)^{k_i}\\ &= \hat{e}(P_{pub},v_i'^*H_1(ID^*))\hat{e}(P_{hlp},v_i'^*H_2(ID^*,t^*))U_i^*.\end{aligned}$$

implying σ_i^* is valid.

Now, $\mathcal{B}$ considers $V_1^* = V^* - \sum_{i=2}^n V_i^*$, and outputs

$$\begin{aligned}\hat{e}(P,V_1^*) &= \hat{e}(P,V^*-\sum_{i=2}^n V_i^*)\\ &= \hat{e}(P_{pub},v_1'^*c_1^*(bP))\hat{e}\big(P_{hlp},v_1'^*w_1^*P\big)\hat{e}(P,k_1^*P_{pub})\\ &= \hat{e}(P,v_1'^*c_1^*(abP)+v_1'^*w_1^*P_{hlp}+k_1^*P_{pub})\end{aligned}$$

$$\begin{aligned}&\Rightarrow V_1^* = v_1'^*c_1^*(abP)+v_1'^*w_1^*P_{hlp}+k_1^*P_{pub}\\ &\Rightarrow v_1'^*c_1^*(abP) = V_1^*-v_1'^*w_1^*P_{hlp}-k_1^*P_{pub}\\ &\Rightarrow abP = v_1'^*-1c_1^{-1*}(V_1^*-v_1'^*w_1^*P_{hlp}-k_1^*P_{pub}).\end{aligned}$$

This completes the description of algorithm $\mathcal{B}$.

For completing the proof of Theorem 1, we determine and prove that $\mathcal{B}$ solves the CDH problem, for given instance in G with non-negligible probability at least ε'. For this, we examine the required events for $\mathcal{B}$ to succeed, as mentioned below:

– E_1 : $\mathcal{B}$ does not abort while responding to any of $\mathcal{A}$'s key extraction queries.
– For one such query, the probability for $\mathcal{B}$ not to abort is $1-\mu$, since $\Pr[d_i = 0] = 1-\mu$. As $\mathcal{B}$ makes at most q_{KE} queries, the probability of $\mathcal{B}$ is $(1-\mu)^{q_{KE}}$, implying $\Pr[E_1] \geqslant (1-\mu)^{q_{KE}}$.
– E_2 : $\mathcal{B}$ does not abort while responding to any of $\mathcal{A}$'s signature queries.

Table 2 Notation and descriptions of various cryptographic operations and their costs.

Notation	Description
T_{mm}	Time needed to execute the modular multiplication operation
T_m	Time needed to execute the elliptic curve point multiplication (scalar multiplication in G): $T_m \approx 29T_{mm}$
T_p	Time needed to execute the bilinear pairing in G_T: $T_p \approx 87T_{mm}$
T_a	Time needed to execute the addition of two elliptic curve points (point addition in G): $T_a \approx 0.12T_{mm}$

- For one signature query the probability for $\mathcal{B}$ not to abort is $1-\mu$, since $\Pr[d_i = 0] = 1-\mu$. As $\mathcal{B}$ makes at most q_S queries, the probability of $\mathcal{B}$ is $(1-\mu)^{q_S}$, implying $\Pr[E_2/E_1] \geqslant (1-\mu)^{q_S}$.
- E_3 : $\mathcal{A}$ outputs a valid and nontrivial forged aggregate signature $\sigma = (U, V)$.
- If $\mathcal{B}$ does not abort while responding to the key extraction and signature queries, then the view of $\mathcal{A}$ is similar to its view in the real attack implying, $\Pr[E_3/E_1 \wedge E_2] \geqslant \varepsilon$.
- E_4 : Event E_3 took place, in addition, $d_1 = 0$ and $d_i = 1$ for $2 \leqslant i \leqslant n$, where for each i, d_i is the d-component of the tuple containing ID_i on the list L_1.
- The probability for $\mathcal{B}$ not to abort after $\mathcal{A}$ outputs a valid and nontrivial aggregate forgery is at least $(1-\mu)^{N-1}$, implying $\Pr[E_4/E_1 \wedge E_2 \wedge E_3] \geqslant (1-\mu)^{N-1}\mu$.
- $\mathcal{B}$ succeeds if all the events $\{E_i\}_{i=1,\ldots,4}$ occur and the success probability $\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4]$ can be computed as follows:

$$\begin{aligned}\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] &= \Pr[E_1]\Pr[E_2/E_1]\Pr[E_3/E_1 \wedge E_2] \\ &\quad \times \Pr[E_4/E_1 \wedge E_2 \wedge E_3] \\ &\geqslant (1-\mu)^{q_{KE}}(1-\mu)^{q_S}(1-\mu)^{N-1}\mu\varepsilon \\ &= (1-\mu)^{q_{KE}+q_S+N-1}\mu\varepsilon = f(\mu)(say)\end{aligned}$$

To maximize $f(\mu)$, differentiate $f(\mu)$ with respect to μ and equate it to 0. By doing so, we get μ as follows:

$$\begin{aligned}&(q_{KE}+q_S+N-1)(1-\mu)^{q_{KE}+q_S+N-1-1}(-1)\mu\varepsilon \\ &\quad +(1-\mu)^{q_{KE}+q_S+N-1}\varepsilon = 0 \\ &(1-\mu)^{q_{KE}+q_S+N-2}\varepsilon[(q_{KE}+q_S+N-1)(-\mu)+(1-\mu)] = 0 \\ &-\mu(q_{KE}+q_S+N)+\mu+1-\mu = 0 \\ &-\mu(q_{KE}+q_S+N) = -1 \\ &\Rightarrow \mu = \tfrac{1}{q_{KE}+q_S+N}.\end{aligned}$$

Algorithm $\mathcal{B}$ provides the correct output with probability at least

$$\left(1-\frac{1}{q_{KE}+q_S+N}\right)^{q_{KE}+q_S+N-1}\left(\frac{1}{q_{KE}+q_S+N}\right)\varepsilon \geqslant \varepsilon'.$$

For sufficiently large q_{KE} and q_S, we have $\frac{1}{e}\frac{\varepsilon}{(q_{KE}+q_S+N)} \geqslant \varepsilon'$.

The running time of algorithm $\mathcal{B}$, is the sum of the time taken to respond to different queries $q_{H_1}, q_{H_2}, q_{H_3}, q_{KE}, q_S$, made by $\mathcal{A}$ and transforming $\mathcal{A}$'s forgery in solving the CDH problem. From the above simulation we notice that there exists: $1T_m$ in each H_1 query, $1T_m$ in each H_2 query, $2T_m$ in each key extraction query, $4T_m$ in each signature query, $2T_m$ in the setup phase and $(3N+2)T_m$ in the forgery phase.

Therefore, the running time of $\mathcal{B}$ is $T \leqslant T' - T_m(q_{H_1}+q_{H_2}+2q_{KE}+4q_S+3N+4)$, as required. This concludes the proof of Theorem 1.

6. Efficiency analysis

In this section, we analyze the performance of our scheme with the related schemes in terms of computational and communication (signature length) cost point of view. In Table 2, we consider the time exhausting operations and their conversions (Barreto et al., 2002; Cao et al., 2010; Chung et al., 2007; He et al., 2011; Ren et al., 2007; Tan et al., 2010).

From the experimental results by Cao et al. (2010), He et al. (2011) and Ren et al. (2007), to achieve the 1024-bit RSA level security, we use the bilinear pairing (Tate pairing) defined over the supersingular elliptic curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2, q is a 160-bit Solinas prime number $q = 2^{159} + 2^{17} + 1$ and p is a 512-bit prime satisfying $p + 1 = 12qr$. The running time is calculated for different cryptographic operations in Cao et al. (2010), He et al. (2011), Ren et al. (2007) using MIRACAL (Shamus Software Ltd., 2013), a standard cryptographic library and implemented on a hardware platform PIV (Pentium-4) 3GHZ processor with 512-MB memory and a windows XP operating system.

Furthermore, Chung et al. (2007) indicate that the time needed to execute the elliptic curve scalar multiplication T_m is approximately $29T_{mm}$, where T_{mm} denotes, the time needed to execute the modular multiplication operation. It was also mentioned by Cao et al. (2010), He et al. (2011) that the time needed to execute one pairing based scalar multiplication T_m is approximately 6.38 ms, i.e. $T_m \approx 6.38$ ms, and the time needed to execute one bilinear pairing (Tate pairing) operation T_p is approximately 20.01 ms i.e. $T_p \approx 20.01$ ms and from the works proposed in Barreto et al. (2002), Tan et al. (2010) $1T_p \approx 3T_m \approx 87T_{mm}$. We summarize these results in Table 2.

Since the proposed scheme is the first key-insulated aggregate signature scheme in the ID-based setting, however, we compare our scheme with Zhao et al. (2014) key-insulated

Table 3 Comparison of our scheme with the existing scheme.

	Scheme	
	Zhao et al. (2014)	Our scheme
Size of aggregate signature	$(n+2)\|G\|$	$2\|G\|$
Aggregate signature phase	$2nT_m + (n-1)T_a \approx (58.12n - 0.12)T_{mm}$	$nT_m + (n-1)T_a \approx (29.12n - 0.12)T_{mm}$
Aggregate verification phase	$3T_p + (2n-1)T_a \approx (0.24n + 260.88)T_{mm}$	$3T_p + (2n-2)T_a \approx (0.24n + 260.76)T_{mm}$
Total cost	$\approx (58.36n + 260.76)T_{mm}$	$\approx (29.36n + 260.64)T_{mm}$
Cryptosystem	PKI-based	ID-based

aggregate signature scheme in PKI based setting. The comparison is summarized in Table 3.

From Table 3, the proposed scheme requires approximately $(29.36n + 260.64)T_{mm}$ operations for both signature generation and signature verification, whereas the scheme of (Zhao et al. 2014) requires approximately $(58.36n + 260.76)T_{mm}$ operations. Hence the proposed scheme reduced the computational cost approximately 50% than that of Zhao et al.'s (2014) scheme.

Moreover, the size of aggregate signature in the scheme (Zhao et al., 2014), grows linear with that of signers, but the proposed scheme has a constant signature size $2|G|$, which is independent of the number of signers which greatly reduces the communication complexity of the system.

7. Conclusion

This paper proposes the first and efficient scheme to reduce the damage caused by the key exposure problem in aggregate signatures in the ID-based setting. In this scheme, the exposure of temporary private keys for a time period will not compromise the security of the remaining periods. The proposed scheme is proven secure in the random oracle paradigm under the assumption that the CDH problem is hard. Our scheme requires constant (three) pairing operations for the verification of aggregate signature and the size of the aggregate signature is constant i.e. the signature size and the number of pairing operations are independent with the number of signers. Thus our IDKIAS scheme is efficient in terms of computational and communicational overhead.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions. Also the authors would like to thank the editors of the journal for their support.

References

Barreto, P.L.M., Kim, H., Lyn, B., Scott, M., 2002. In: Efficient Algorithms for Pairing based Cryptosystems, vol. 2442. Springer-Verlag, LNCS, pp. 354–368.

Boneh, D., Boyen, X., 2004. Efficient selective-ID secure identity based Encryption without random oracles. In: Eurocrypt'04, vol. 3027. Springer-Verlag, LNCS, pp. 223–238.

Boneh, D., Franklin, M., 1985. Identity based encryption from the Weil pairing. In: Advances in Cryptology-Crypto'01, vol. 2139. Springer-Verlag, LNCS, pp. 213–229.

Boneh, D., Gentry, C., Lynn, B., Shacham, H., 2003. Aggregate and verifiably encrypted signatures from bilinear maps. In: Eurocrypt '03, vol. 2656. Springer Verlag, LNCS, pp. 416–432.

Cao, X., Kou, W., Du, X., 2010. A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. Inf. Sci. 180 (15), 2895–2903.

Cha, J.C., Cheon, J.H., 2003. An identity-based signature scheme from gap Diffie–Hellman groups. In: Proceedings of PKC 2003, vol. 2567. Springer-Verlag, LNCS, pp. 18–30.

Chen, J., Long, Y., Chen, K., Guo, J., 2014. Attribute-based key-insulated signature and its applications. Inf. Sci. 275, 57–67.

Chen, J., Long, Y., Chen, K., Wang, Y., Li, X., 2011. An efficient threshold key-insulated signature scheme. J. Shanghai Jiaotong Univ. (Sci.) 16 (6), 658–662.

Chung, Y.F., Huang, K.H., Lai, F., Chen, T.S., 2007. ID-based digital signature scheme on the elliptic curve cryptosystem. Comput. Stand. Interfaces 29 (6), 601–604.

Dodis, Y., Katz, J., Xu, S., Yung, M., 2002. Key-insulated public key cryptosystems. In: EUROCRYPT'02, vol. 2332. Springer-Verlag, LNCS, pp. 65–82.

Dodis, Y., Katz, J., Xu, S., Yung, M., 2003. Strong key-insulated signature schemes. In: PKC'03, vol. 2567. Springer-Verlag, LNCS, pp. 130–144.

Gentry, C., Ramzan, Z., 2006. Identity-based aggregate signatures. In: PKC'06, vol. 3958. Springer-Verlag, LNCS, pp. 257–273.

González-Deleito, N., Markowitch, O., Dall'Olio, E., 2004. A new key-insulated signature scheme. In: ICICS'04, vol. 3269. Springer-Verlag, LNCS, pp. 465–479.

Gopal, P.V.S.S.N., Vasudeva Reddy, P., Gowri, T., 2012. New identity based signature scheme using bilinear pairings over elliptic curves. In: 3rd IEEE IACC-2013, pp. 362–367.

He, D., Chen, J., Hu, J., 2011. An ID-based proxy signature scheme without bilinear pairings. Ann. Telecommun. 66, 657–662.

Hess, F., 2002. In: Efficient Identity Based Signature Schemes Based on Pairings, vol. 2595. Springer-Verlag, LNCS, pp. 310–324.

Li, J., Zhang, F., Wang, Y., 2006. A strong identity based key-insulated cryptosystem. In: EUC Workshops 2006, vol. 4097. Springer-Verlag, LNCS, pp. 352–361.

Paterson, K.G., 2002. ID-based signatures from pairings on elliptic curves. IEEE Electron. Lett. 38 (18), 1025–1026.

Ren, K., Lou, W., Zeng, K., Moran, P.J., 2007. On broadcast authentication in wireless sensor networks. IEEE Trans. Wireless Commun. 6 (11), 4136–4144.

Shamir, A., 1985. Identity-based cryptosystems and signature schemes. In: Advances in Cryptology-Crypto'84, vol. 196. Springer-Verlag, LNCS, pp. 47–53.

Shamus Software Ltd. Miracl Library. Available at <http://certivox.org/display/EXT/MIRACL>.

Tan, S.H., Heng, S.H., Goi, B.M., 2010. Java implementation for pairing-based cryptosystems. In: Proc. of the Int. Conference in Computational Science and Its Applications (ICCSA'10), vol. 6019. Springer-Verlag, LNCS, pp. 188–198.

Wan, Z., 2011. A new identity-based parallel key-insulated signature scheme without random oracles. 4th IEEE Int. Symp. Comput. Intell. Des. 2, 27–30.

Wang, H., Cao, H., Li, D., 2013. Identity-based Key-insulated signcryption Scheme. J. Comput. Inf. Syst. 9 (8), 3067–3075.

Wang, Z., Chen, H.Y., Ye, D.F., Wu, Q., 2008. In: Practical Identity-Based Aggregate Signature from Bilinear Maps, vol. 13(6). Shangai Jiao Tong University Press, pp. 684–687.

Wang, J., Wu, Q., Wang, Y., 2004. A new perfect and strong key-insulated signature scheme. In: Proc. of China Crypt'2004, pp. 233–239.

Weng, J., Liu, S., Chen, K., Li, X., 2006a. Identity-based key-insulated signature with secure key-updates. In: INSCRYPT'06, vol. 1318. Springer-Verlag, LNCS, pp. 13–26.

Weng, J., Liu, S., Chen, K., Ma, C.S., 2006b. Identity-based key-insulated signature without random oracles. IEEE Int. Conf. Comput. Intell. Secur. 2, 1253–1258.

Wu, T.Y., Tseng, Y.M., Yu, C.W., 2012. ID-based key-insulated signature scheme with batch verifications and its novel application. Int. J. Innovative Comput. Inf. Control 8 (7(A)), 4797–4810.

Xu, J., Zhang, Z., Feng, D., 2005. ID-based aggregate signatures from bilinear pairings. In: Proc. of the 4th Int. Conference on Cryptology and Network Security, vol. 3810. Springer-Verlag, LNCS, pp. 110–119.

Yuan, Y., Zhan, Q., Huang, H., 2014. Efficient unrestricted identity-based aggregate signature scheme. PLoS One 9 (10), 1–8.

Yu, Y., Zheng, X., Sun, H., 2011. An identity based aggregate signature from pairings. J. Networks 6 (4), 631–637.

Zhao, H., Yu, Jia., Duan, S., Cheng, X., Hao, R., 2014. Key-insulated aggregate signature. Front. Comput. Sci. 8 (5), 837–846.

Zhou, Y., Cao, Z., Chai, Z., 2006. Identity based key-insulated signature. In: ISPEC'06, vol. 3903. Springer-Verlag, LNCS, pp. 226–234.