An ID-based linearly homomorphic signature scheme and its application in blockchain

Qun Lin, Hongyang Yan, Zhengan Huang, Wenbin Chen, Jian Shen, and Yi Tang*

***Abstract*—Identity-based cryptosystems mean that public keys can be directly derived from user identifiers, such as telephone numbers, email addresses, and social insurance number etc.. So they can simplify key management procedures of certificate-based public key infrastructures and can be used to realize authentication in blockchain. Linearly homomorphic signature schemes allow to perform linear computations on authenticated data. And the correctness of the computation can be publicly verified. Although a series of homomorphic signature schemes have been designed recently, there are few homomorphic signature schemes designed in identity-based cryptography. In this paper, we construct a new ID-based linear homomorphic signature scheme, which avoids the shortcomings of the use of public-key certificates. The scheme is proved secure against existential forgery on adaptively chosen message and ID attack under the random oracle model. ID-based linearly homomorphic signature schemes can be applied in e-business and cloud computing. Finally, we show how to apply it to realize authentication in blockchain.**

***Index Terms*—ID-based signature, Homomorphic signature, Bilinear pairings, Random oracle.**

I. INTRODUCTION

NOWADAYS, people have paid attention to the importance of information security [3], [4], [5], [6], [9], [10]. The public-key cryptography plays a critical role in information security. As we know, certificate-based cryptosystems are most widely deployed public-key cryptosystems. And they require that the authenticated public-key certificate of an entity be obtained in order to encrypt information for the entity. So these certificates need to be generated in large and distributed to many users in communities. Furthermore, the certificates need to be verified frequently. So the management of public-key certificates is cumbersome. In order to avoid the shortcomings of the use of public-key certificates, Shamir introduced the concept of identity-based cryptography in 1984 [1]. The idea is to derive public keys directly from user identifiers, such as telephone numbers, email addresses, and social insurance number etc.. Moreover, the corresponding private key is generated by a combination of the user's public key and the system-level secret key of a central authority that is named as Private Key Generator or PKG for short. Since then, the research on ID-based cryptography has made great progress , such as ID-based signature schemes [11], [28], [29], ID-based encryption schemes [30], [31], ID-based key agreement schemes [32], [33].

In 2002, the conception of homomorphic signature was originally proposed by Johnson et al. [2]. The notion of homomorphic signature is an important primitive and allows to validate computation over authenticated data [39], [40], [41]. Informally, a user Alice can sign l messages $\{m_i\}_{i=1}^{l}$ and produce the signatures $\{\sigma_i\}_{i=1}^{l}$, which can be verified exactly as ordinary signatures. The homomorphic property provides the special feature that given $\sigma_1, ..., \sigma_l$ and some function $f : M^l \to M$, anyone can compute a signature σ on the value $f(m_1, ..., m_l)$ without knowledge of the secret signing key Sk. Homomorphic signature schemes can be employed in electronic business and cloud computing [16], [18], [21], [27], [36]. Nowadays, there are many types of homomorphic signatures, such as the linearly homomorphic signature schemes [7], [8], [13], [20], [23], [26], the homomorphic schemes supporting polynomial functions [12], [14], [15], and the leveled fully homomorphic signature schemes [17], [19]. But these schemes belong to certificate-based cryptosystems. Up to our knowledge, there are few homomorphic signature schemes [34], [35], [37] designed in identity-based cryptography. And the schemes in [34], [35] focus on network coding which can prevent malicious nodes to produce the pollution attacks. The scheme in [37] is designed over lattices, and it is not efficient. Since the management of public-key certificates is cumbersome in the certificate-based cryptosystems, it is meaningful to design homomorphic signature schemes in identity-based cryptosystems.

Our contributions. In this paper, the concept and security model of ID-based linearly homomorphic signature are proposed. It means that the signer can produce a linearly homomorphic signature in identity-based cryptosystems. Moreover, we use bilinear groups as the underlying tool to design an ID-based linearly homomorphic signature. The new scheme is proved secure against existential forgery on adaptively chosen message and ID attack in the random oracle model, and it can combine the natures of linearly homomorphic signature and identity-based cryptosystems.

Q. Lin is with Institute of Mathematics and Statistics, Hanshan Normal University, Chaozhou 521041, China (email: 190952020@qq.com).

H. Yan is with College of Computer and Control Engineering, Nankai University, Tianjin 300071, China (email: hyang.yan@foxmail.com).

Z. Huang is with School of Computer Science, Guangzhou University, Guangzhou 510006, China (email: zhahuang.sjtu@gmail.com).

W. Chen is with School of Computer Science, Guangzhou University, Guangzhou 510006, China (email: cwb2011@gzhu.edu.cn).

J. Shen is with School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (email: s_shenjian@126.com).

Y. Tang is with School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China (email: ytang@gzhu.edu.cn). Corresponding author.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2809426, IEEE Access

Organization. The rest of the paper is organized as follows. Section 2 contains some preliminaries about bilinear maps, the short signature scheme proposed by Boneh, Lynn, and Shacham (BLS), as well as the framework of ID-based linear homomorphic signature schemes and the security model. Section 3 gives a new ID-based linear homomorphic signature, and Section 4 gives the security proof of the scheme. Finally, Section 5 concludes this paper.

II. PRELIMINARIES

A. Bilinear groups

In this section, we briefly review the facts about bilinear maps. Let (G_1,G_2) be two cyclic groups of prime order q, in which group operations are efficiently computable.

$e : G_1 \times G_1 \to G_2$ is a bilinear map with the following properties:
(1) Bilinear: $\forall g, h \in G_1, \forall a, b \in Z_q$, $e(g^a, h^b) = e(g, h)^{ab}$;
(2) Non-degenerate: There exist $g, h \in G_1$, such that $e(g, h) \neq 1$;
(3) Computable: For all $g, h \in G_1$, there exists an efficient algorithm to compute $e(g, h)$.

Now we introduce the Computational Diffie-Hellman assumption in G_1.

Definition 1: **(CDH).** Let $g \in G_1$ be a random generator, $x, y \leftarrow {Z_q}^*$ be taken over the uniform choices, and λ be the security parameter. We define the advantage of an adversary A in solving the Computational Diffie-Hellman problem as

$$\mathbf{ADV}_A^{cdh}(\lambda) = \Pr\left[A(g, g^x, g^y) = g^{xy}\right],$$

where the probability is taken over the uniform choices of x, y and the internal coin tosses of A. If for every probabilistic polynomial-time (PPT) algorithm A, the $\mathbf{ADV}_A^{cdh}(\lambda)$ is negligible, we say that the CDH assumption holds in G_1.

B. BLS short signature scheme

The BLS short signature scheme proposed in [22] consists of the following algorithms: a key generation algorithm **KeyGen**, a signature generation algorithm **Sign** and a signature verification algorithm **Verify**. And it uses a full-domain hash function $H : \{0,1\}^* \to G_1$. $e : G_1 \times G_1 \to G_2$ is a bilinear map, and g is a random generator of G_1.
KeyGen: Pick a random $x \in Z_q^*$ as the secret key, and computer the public key $PK = g^x$.
Sign: Given a secret key x, and a message m, this algorithm outputs the signature $\sigma = H(m)^x$.
Verify: Given a public key PK, a message m, and a signature σ, if the equation $e(g, \sigma) = e(PK, H(m))$ holds, this algorithm outputs 1; otherwise it outputs 0.

The security of BLS short signature scheme is based on the CDH assumption. We refer to [22] for more details.

C. ID-based signature scheme

Definition 2: **(ID-based signature scheme [11])**. An ID-based signature scheme is a tuple of four PPT algorithms **(Setup, Extract, Sign, Verify)**. The algorithms are defined as follows:

- **Setup:** This algorithm takes as input a security parameter λ and outputs a secret/public key pair (x, P_{pub}) for the PKG.
- **Extract:** This algorithm takes as input the secret key x, the params and an user's identity ID, and returns a private key D_{ID} corresponding to ID in the system.
- **Sign:** Given the private key D_{ID} and a message m, this algorithm outputs a signature σ for m.
- **Verify:** Given the signer's identity ID, a message m and a signature σ, this algorithm outputs *1* if σ is a valid signature for m; otherwise, output *0*.

Correctness. For all message m, if $\sigma \leftarrow$ **Sign**(D_{ID}, m), then

$$\textbf{Verify}(ID, m, \sigma) = 1.$$

Existential Unforgeability. An ID-based signature scheme is unforgeable against adaptively chosen-message and ID attacks if no polynomial time algorithm *A* has a non-negligible advantage against a challenger *C* in the following game:

- *C* runs **Setup** of the scheme and sends the system parameters to *A*.
- *A* issues the following queries:
  (1) **Extract queries**. Given an identity *ID*, *C* outputs the private key corresponding to *ID*.
  (2) **Signing queries**. Given an identity *ID* and a message m, *C* returns a signature for m.
- *A* outputs (ID^*, m, σ). Then *A* wins the game if **Verify**$(ID^*, m, \sigma) = 1$, the identity ID^* does not appear in Extract queries and (ID^*, m) does not appear in Signing queries.

D. ID-based linearly homomorphic signature

Definition 3: **(ID-based linearly homomorphic signature scheme).** An ID-based linearly homomorphic signature scheme is a tuple of five PPT algorithms **(HSetup, HExtract, HSign, HVerify, HEval)**. The algorithms are defined as follows:

- **HSetup:** This algorithm takes as input a security parameter λ, an upper bound l for the number of messages signed in each file and an integer N denoting the length of vectors to be signed. It outputs a secret/public key pair (x, P_{pub}) for the PKG.
- **HExtract:** This algorithm takes as input the secret key for the PKG, the params and an user's identity ID, and returns a secret key D_{ID} corresponding to ID in the system.
- **HSign:** Given the secret key D_{ID}, a message vector **v**, and a file identifier τ, this algorithm outputs a signature σ.
- **HVerify:** Given the signer's identity ID, a message vector **v**, a file identifier τ, and a signature σ, this algorithm outputs *1* if σ is a valid signature for **v**; otherwise, output *0*.
- **HEval:** Given the signer's identity ID, a file identifier τ, and a set of tuples $\{(f_i, \sigma_i)\}_{i=1}^l$, this algorithm outputs a signature σ (Note that σ is intended to be a signature on

$\sum_{i=1}^{l} f_i \mathbf{v}^{(i)}$, where $\mathbf{v}^{(i)}$ denotes the i-th vector in the list of vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(l)}$).

Correctness. For correctness, we require:
(1) For all message vector $\mathbf{v}$ and all file identifier $\tau \in \{0,1\}^\lambda$, if $\sigma \leftarrow$ **HSign**$(D_{ID}, \mathbf{v}, \tau)$, then **Hverify**$(ID, \mathbf{v}, \tau, \sigma) = 1$.
(2) For all $\tau \in \{0,1\}^\lambda$ and all sets of triples $\{(f_i, \sigma_i, \mathbf{v}^{(i)})\}_{i=1}^{l}$, if **Hverify**$(ID, \mathbf{v}^{(i)}, \tau, \sigma_i) = 1$ holds for all i, then **Hverify**$(ID, \sum_{i=1}^{l} f_i \mathbf{v}^{(i)}, \tau,$ **HEval**$(ID, \tau, \{(f_i, \sigma_i)\}_{i=1}^{l})) = 1$.

Security model:

An ID-based linear homomorphic signature is unforgeable against adaptively chosen-message and ID attack if the advantage of any PPT adversary A in the following game is negligible in the security parameter λ.

Setup: The challenger C sets a secret/public key pair (x, P_{pub}) for the PKG, and gives P_{pub} to the adversary A.

Queries: The adversary's attack capabilities are modelled by providing it access to a series of oracles, so A can ask a polynomial number of queries as follows:

- **HExtract queries.** Given an identity *ID*, C outputs the secret key corresponding to *ID*.
- **HSigning queries**. A asks for a new signature on an identity *ID*, a message vector $\mathbf{v}$ and a file identifier $\tau \in \{0,1\}^\lambda$. The challenger C runs the algorithm **HSign** to compute a signature σ for $\mathbf{v}$. Finally C chooses a handle h from a proper set, stores $(h, (\sigma, ID, \mathbf{v}, \tau))$ in a table T and returns h to A.
- **Derivation queries**. A chooses a set of handles $h = (h_1, ..., h_l)$ and a vector of coefficients $\mathbf{f} = (f_1, ..., f_l)$. Then the challenger C checks $\{(h_i, (\sigma_i, ID, \mathbf{v}^{(i)}, \tau_i,))\}_{i=1,...,l}$ from table T and returns $\perp$ if any of these does not exist or if $\tau_i \neq \tau_j$ for some $i, j \in \{1, ..., l\} (i \neq j)$. Else, C computes $\mathbf{v} = \sum_{i=1}^{l} f_i \mathbf{v}^{(i)}$, σ =**HEval**$(ID, \tau_1, \{(f_i, \sigma_i)\}_{i=1}^{l})$, picks a handle h, stores $(h, (\sigma, ID, \mathbf{v}, \tau_1))$ in the table T and returns h to A.
- **Reveal queries**. A chooses a handle h. If this handle does not exist in the table T, the challenger C returns $\perp$. Otherwise, C checks the corresponding record $(h, (\sigma, ID, \mathbf{v}, \tau))$ from the table T and sends $(\sigma, ID, \mathbf{v}, \tau)$ to A. Next it adds $(h, (\sigma, ID, \mathbf{v}, \tau))$ to a different table T^*.

Output: A outputs an identity ID^*, a message vector $\mathbf{v}^*$, a file identifier τ^* and a signature Q^*.

The adversary wins if **Hverify**$(ID^*, \mathbf{v}^*, \tau^*, Q^*) = 1$, the identity ID^* does not appear in HExtract queries, and it must satisfy one of the following conditions:

1) The file identifier $\tau^* \neq \tau_k$ for all τ_k that appears in the table T^* and $\mathbf{v}^* \neq \mathbf{0}$.
2) The file identifier $\tau^* = \tau_k$ for some file identifier τ_k that appears in the table T^*, but $\mathbf{v}^* \notin V_k$, where V_k denotes the subspace spanned by all vectors $\{\mathbf{v}^{(i)}\}_{i=1,...,l_k}$ queried with the same file τ_k that appears in T^*, with $0 < l_k \leq l$.

We define the advantage $\mathbf{ADV}_A^{lhs}(\lambda)$ of an adversary against an ID-based linearly homomorphic signature scheme as the probability of A winning the above game.

Definition 4: **(Unforgeability of ID-based Linearly Homomorphic Signatures).** An ID-based linearly homomorphic signature scheme is secure against chosen-message and ID attack if $\mathbf{ADV}_A^{lhs}(\lambda)$ in the above relevant game is negligible for any PPT adversaries .

III. The proposed scheme

In this section, we propose a provably secure ID-based linearly homomorphic signature from bilinear pairings. The new scheme is defined as follows:

1) **HSetup:** Let (G_1,G_2) be bilinear groups such that $|G_1|$=$|G_2|$=q for some prime number q. A bilinear map is given by e: $G_1 \times G_1 \rightarrow G_2$. Define two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : \{0,1\}^* \times \{0,1\}^* \rightarrow G_1$. H_1 and H_2 will be viewed as random oracles in our security proof. Choose a generator g of G_1 and $x \in Z_q^*$, then set $P_{pub} = g^x$. Let $[l] = \{1, ..., l\}$, $[N] = \{1, ..., N\}$. The security parameter is λ. **S=(Setup,Extract,Sign,Verify)** is a standard ID-based signature scheme, such as the scheme proposed in [11]. Compute $(x_s, P_{pub_s}) \leftarrow$ **Setup**(1^λ) as the secret/public key of PKG for **S**. The master secret key is $x_h = (x_s, x)$, and the master public key is $P_{pub_h} = (G_1, G_2, q, g, P_{pub_s}, P_{pub}, e, H_1, H_2)$.
2) **HExtract:** Given an identity *ID*, the algorithm generates $D_{ID}^{(1)} \leftarrow$ **Extract**(x_s, ID) and $D_{ID}^{(2)} \leftarrow H_1(ID)^x$, then outputs $D_{ID} \leftarrow (D_{ID}^{(1)}, D_{ID}^{(2)})$, which is the secret key associated to the identity *ID*.
3) **Hsign:** Suppose this algorithm has stored a list L of all previously returned identifiers τ with the related information (r, w, σ_1) defined below. Take the secret key D_{ID}, an identity *ID*, a message vector $\mathbf{v} = (v_1, ..., v_N) \in Z_q^N$ and a file identifier $\tau \in \{0,1\}^\lambda$ as input, this algorithm responses according to the type of τ in input:
   - **If** τ appears in L, retrieve the associated (r, w, σ_1) from L.
   - **Otherwise**, choose $r \in Z_q^*$ randomly, set $w \leftarrow g^r$, $\sigma_1 \leftarrow$ **Sign**$(D_{ID}^{(1)}, (\tau, w))$, and store this information in L.

   Then choose $s \in Z_q^*$ randomly, and compute

   $$\sigma_2 = D_{ID}^{(2)\sum_{j \in [N]} v_j} \cdot (H_1(ID)^s \prod_{j \in [N]} H_2(\tau, j)^{v_j})^r.$$

   Finally, output $Q \leftarrow (w, \sigma_1, \sigma_2, s)$ as a signature for a message vector $\mathbf{v} = (v_1, ..., v_N)$.
4) **HVerify:** Given an identity *ID*, a file identifier τ, a message vector $\mathbf{v} = (v_1, ..., v_N) \in Z_q^N$ and a signature $Q = (w, \sigma_1, \sigma_2, s)$, the verifier checks that:

   $$\textbf{Verify}(ID, \sigma_1, (\tau, w)) = 1,$$

$$e(\sigma_2, g) = e(H_1(ID), P_{pub})^{\sum\limits_{j\in[N]} v_j} \cdot e(H_1(ID)^s \prod_{j\in[N]} H_2(\tau, j)^{v_j}, w)$$

If both of the above equations hold, output 1; otherwise, output 0.

5) **HEval:** Given an identity *ID*, a file identifier τ, and a set of tuples $\{(f_i, \sigma_i)\}_{i=1}^{l}$ such that $Q_i = (w^{(i)}, \sigma_1^{(i)}, \sigma_2^{(i)}, s^{(i)})$, this algorithm checks if $w^{(i)}$ are not all equal, then output $\perp$. Else, compute $s \leftarrow \sum\limits_{i\in[l]} f_i s^{(i)}$, $\sigma_2 \leftarrow \prod\limits_{i\in[l]} {\sigma_2^{(i)}}^{f_i}$. Then output $Q = (w^{(1)}, \sigma_1^{(1)}, \sigma_2, s)$.

Correctness:

Given an identity *ID*, a file identifier τ, a message vector $\mathbf{v} = (v_1, ..., v_N) \in Z_q^N$ and a signature $Q = (w, \sigma_1, \sigma_2, s) \leftarrow$ **HSign**$(D_{ID}, ID, \mathbf{v}, \tau)$, the correctness of the scheme can be verified by the following equations:

$$\textbf{Verify}(ID, \sigma_1, (\tau, w)) = 1,$$

$$\begin{aligned} e(\sigma_2, g) &= e(D_{ID}^{(2)}, g)^{\sum\limits_{j\in[N]} v_j} \cdot e((H_1(ID)^s \prod_{j\in[N]} H_2(\tau, j)^{v_j})^r, g) \\ &= e(H_1(ID), P_{pub})^{\sum\limits_{j\in[N]} v_j} \cdot e(H_1(ID)^s \prod_{j\in[N]} H_2(\tau, j)^{v_j}, w) \end{aligned}$$

Moreover, given $\tau \in \{0,1\}^\lambda$ and all sets of triples $\{(f_i, Q_i, \mathbf{v}^{(i)})\}_{i=1}^{l}$ such that $Q_i = (w^{(i)}, \sigma_1^{(i)}, \sigma_2^{(i)}, s^{(i)})$, where $w^{(1)} = w^{(2)} = ... = w^{(l)}$, by our definition of **HEval**, we have

$$s \leftarrow \sum_{i\in[l]} f_i s^{(i)}, \qquad \sigma_2 \leftarrow \prod_{i\in[l]} {\sigma_2^{(i)}}^{f_i}.$$

Then the signature is $Q = (w^{(1)}, \sigma_1^{(1)}, \sigma_2, s)$.

Now, we only need to check that Q is a signature on the $\mathbf{v} = (v_1, ..., v_N) = \sum\limits_{i\in[l]} f_i \mathbf{v}^{(i)}$, where $\mathbf{v}^{(i)}$ denotes the i-th vector in the list of vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(l)}$. Suppose $\mathbf{v}^{(i)} = (v_1^{(i)}, ..., v_N^{(i)})$, by correctness of individual signature, we have

$$\textbf{Verify}(ID, \sigma_1^{(1)}, (\tau, w^{(1)})) = 1,$$

and for $i = 1, \cdots, l$,

$$\begin{aligned} e(\sigma_2^{(i)}, g) &= e(H_1(ID), P_{pub})^{\sum\limits_{j\in[N]} v_j^{(i)}} \cdot e(H_1(ID)^{s^{(i)}} \prod_{j\in[N]} H_2(\tau, j)^{v_j^{(i)}}, w^{(1)}). \end{aligned}$$

So by the bilinear property, we have

$$\begin{aligned} e(\sigma_2, g) &= \prod_{i\in[l]} e(\sigma_2^{(i)}, g)^{f_i} \\ &= e(H_1(ID), P_{pub})^{\sum\limits_{i\in[l]} \sum\limits_{j\in[N]} f_i v_j^{(i)}} \cdot e(H_1(ID)^{\sum\limits_{i\in[l]} f_i s^{(i)}} \prod_{j\in[N]} H_2(\tau, j)^{\sum\limits_{i\in[l]} f_i v_j^{(i)}}, w^{(1)}) \\ &= e(H_1(ID), P_{pub})^{\sum\limits_{j\in[N]} \sum\limits_{i\in[l]} f_i v_j^{(i)}} \cdot e(H_1(ID)^s \prod_{j\in[N]} H_2(\tau, j)^{\sum\limits_{i\in[l]} f_i v_j^{(i)}}, w^{(1)}) \\ &= e(H_1(ID), P_{pub})^{\sum\limits_{j\in[N]} v_j} \cdot e(H_1(ID)^s \prod_{j\in[N]} H_2(\tau, j)^{v_j}, w^{(1)}) \end{aligned}$$

Both of the verification equations hold, so Q is a signature on the $\mathbf{v} = \sum\limits_{i\in[l]} f_i \mathbf{v}^{(i)}$.

This completes the proofs.

IV. Proposed Scheme Analysis

Theorem 1: Assuming **S=(Setup,Extract,Sign,Verify)** is an ID-based signature scheme unforgeable under adaptive chosen message and ID attack, H_1 and H_2 are viewed as random oracles, and the CDH assumption holds, the scheme described above is an ID-based linear homomorphic signature scheme secure against chosen-message and ID attack.

Proof: As usual, the proof proceeds by contradiction. Assuming there exists an adversary *A* that has non-negligible probability in winning the above game, we will show how to build a simulator *B* that breaks the security of the underlying ID-based signature scheme **S** or the CDH assumption. Let $(ID^*, \mathbf{v}^*, \tau^*, Q^* = (w^*, \sigma_1^*, \sigma_2^*, s^*))$ be a valid forgery returned by the adversary. It must hold that **Hverify**$(ID^*, \mathbf{v}^*, \tau^*, Q^*) = 1$ and the identity ID^* does not appear in HExtract queries. According to the game definition, we distinguish two types of forgery as follows:

- **Type 1:** File identifier $\tau^* \neq \tau_j$ for all τ_j that appears in any Reveal queries and $\mathbf{v}^* \neq \mathbf{0}$.
- **Type 2:** $\tau^* = \tau_j$ for some file identifier τ_j that appears in any Reveal queries, but $\mathbf{v}^* \notin V_j$, where V_j denotes the subspace spanned by all vectors $\{\mathbf{v}_i\}_{i=1,...,l_j}$ queried with the same file τ_j in the Reveal queries, with $0 < l_j \leq l$.

First, the simulator flips a coin $b \leftarrow \{0, 1\}$ randomly. If $b = 0$, it guesses that the adversary will produce a Type 1 forgery. Otherwise, it guesses that the adversary will return a Type 2 forgery. Notice that with probability at least $\frac{1}{2}$ the guess is correct.

Type 1 Simulation:

Assuming *B* is a simulator, *A* is an adversary for our scheme, and C_s is a challenger for the underlying ID-based signature scheme **S**. In this case, *B* has guessed that *A* will return a Type 1 forgery. We describe *B* that uses *A* to break the security of **S**. First, *B* receives P_{pub_s} from C_s, sets $x \leftarrow Z_q^*$, $P_{pub} \leftarrow g^x$, and initializes an empty table L as described in the description of the above scheme. Then B sends the master public key $(G_1, G_2, q, g, P_{pub_s}, P_{pub}, e, H_1, H_2)$ to A.

HExtract queries. When A requests the secret key for user ID, B uses its Extracting oracle for **S** to get $D_{ID}^{(1)}$, and can easily compute $D_{ID}^{(2)}$ as in the real case. Then B sets $D_{ID} \leftarrow (D_{ID}^{(1)}, D_{ID}^{(2)})$ and sends D_{ID} to A.

HSigning queries. When A asks for a new signature on an identity *ID*, a message vector **v** and a file identifier τ, B checks if τ does not appear in L, then chooses $r \in Z_q^*$ randomly, sets $w \leftarrow g^r$, uses its Signing oracle for **S** to compute σ_1 for (τ, w), and stores (r, w, σ_1) in L. Otherwise, it retrieves the associated (r, w, σ_1) from L. Then B can easily compute (σ_2, s) (i.e. the remaining parts of each signature) as in the real case. And B sets $Q \leftarrow (w, \sigma_1, \sigma_2, s)$. Note that Q is associated with a new handle h and B stores $(h, (Q, ID, \mathbf{v}, \tau))$ in a table T.

Derivation and Reveal queries. B proceeds as the real oracle.

Output. Finally, the adversary A is supposed to output a forgery $(ID^*, \mathbf{v}^*, \tau^*, Q^*)$ such that $Q^* = (w^*, \sigma_1^*, \sigma_2^*, s^*)$ and **Hverify**$(ID^*, \mathbf{v}^*, \tau^*, Q^*) = 1$. According to the definition of Type 1 forgery, the file identifier $\tau^* \neq \tau_j$ for all τ_j that appears in any Reveal queries and the identity ID^* does not appear in HExtract queries. So B can output $((\tau^*, w^*), \sigma_1^*)$ as a forgery for **S**.

Let us analyze B's probability of success. It is straightforward to see that if the adversary has advantage ϵ in forging the signature scheme, then B has probability at least $\epsilon/2$ in breaking the security of **S**.

Type 2 Simulation:

Assuming B is a simulator and A is an adversary, B is given (g, g^x, g^y) in order to output g^{xy}.

In this case, B has guessed that A will return a Type 2 forgery. We describe B that uses A to break the CDH assumption. First, B runs $(x_s, P_{pub_s}) \leftarrow$ **Setup**(1^λ) as the secret/public key of PKG for **S**, sets $P_{pub} \leftarrow g^x$, and initializes an empty table L as described in the description of the above scheme. Then B sends the master public key $(G_1, G_2, q, g, P_{pub}, P_{pub_s}, e, H_1, H_2)$ to A and responses as follows:

H_1-**queries.** Assuming A makes H_1-queries at most q_{H_1} times, B randomly chooses $\eta \in [1, q_{H_1}]$ as the target ID's number. Denote by ID_k the input of the k-th query made by A and chooses $t_k \in Z_q^*$ uniformly at random. When A queries ID_k to H_1-oracle, B answers $H_1(ID_k) = g^{t_k}$ if $k \neq \eta$; Otherwise, $H_1(ID_k) = g^y$ if $k = \eta$. Then B adds $(ID_k, g^{t_k}, t_k)_{k \neq \eta}$ to the H_1-**List**; If $k = \eta$, B adds $(ID_\eta, g^y, *)$ to the H_1-**List** (Note that $*$ means the corresponding value is unknown).

H_2-**queries.** Assuming A makes H_2-queries at most q_{H_2} times, B randomly chooses $\alpha_j, \beta_j \in Z_q^*$. When A queries to H_2-oracle, B responds to A as $H_2(\tau, j) = (g^y)^{\alpha_j} g^{\beta_j}$. Then B adds $((\tau, j), g^{y\alpha_j} g^{\beta_j}, \alpha_j, \beta_j)$ to the H_2-**List**.

HExtract queries. When A requests secret key for user ID_k, assuming w.l.o.g A has requested H_1-**queries** on ID_k, B checks the H_1-**List** and computes $D_{ID_k}^{(2)} = (g^{t_k})^x = (P_{pub})^{t_k}$ if $k \neq \eta$; Otherwise, B aborts if $k = \eta$. Furthermore, B generates $D_{ID_k}^{(1)} \leftarrow$ **Extract**(x_s, ID). Then B sends $D_{ID_k} \leftarrow (D_{ID_k}^{(1)}, D_{ID_k}^{(2)})$ to A and adds $(ID_k, D_{ID_k})_{k \neq \eta}$ to the **SK-List**.

HSigning queries. We will use the assumption that A only queries the HSigning oracle on independent vectors for each file identifier τ. Given ID_k, a file identifier τ, and an index $i \in [l]$, if A requests a signature on the i-th message vector $\mathbf{v} = (v_1, ..., v_N) \in Z_q^N$ from file τ, B answers as follows:

I. if $k \neq \eta$, assuming A has requested HExtract queries on ID_k and H_2-queries on τ, B checks the H_1-**List**, H_2-**List** and **SK-List**, gets the corresponding D_{ID_k}, and responses with the following.

1) if τ does not appear in L, it chooses fresh $r \in Z_q^*$ randomly, set $w \leftarrow g^r$, $\sigma_1 \leftarrow$ **Sign**$(D_{ID_k}^{(1)}, (\tau, w))$, and stores (τ, r, w, σ_1) in L.
2) Otherwise, it retrieves the corresponding (r, w, σ_1) from L.

Then choose $s \in Z_q^*$ randomly, and compute

$$\sigma_2 = D_{ID_k}^{(2)\sum_{j\in[N]} v_j} \cdot (H_1(ID_k)^s \prod_{j\in[N]} H_2(\tau, j)^{v_j})^r$$

So the signature is $Q = (w, \sigma_1, \sigma_2, s)$. It is easy to check that Q is valid. Note that Q is associated with a new handle h and stored $(h, (Q, ID_k, \mathbf{v}, \tau))$ in a table T.

II. if $k = \eta$, assuming A has requested H_2-queries on τ, B checks the H_2-**List**, and responses with the following.

1) if τ does not appear in L, B chooses fresh $r \in Z_q^*$ randomly, sets $w = P_{pub}^r = g^{rx}$, $\sigma_1 \leftarrow$ **Sign**$(D_{ID_\eta}^{(1)}, (\tau, w))$, and stores (τ, r, w, σ_1) in L.
2) Otherwise, it retrieves the corresponding (r, w, σ_1) from L.

Then compute

$$s \leftarrow -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j, \qquad \sigma_2 \leftarrow w^{\sum_{j\in[N]} \beta_j v_j}$$

So the signature is $Q = (w, \sigma_1, \sigma_2, s)$. It is not hard to see that the signature Q is correct. Because

$$\textbf{Verify}(ID_\eta, \sigma_1, (\tau, w)) = 1$$

Furthermore, $s = -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j$, so

$$\sum_{j\in[N]} v_j + rs + r\sum_{j\in[N]} \alpha_j v_j = rs + \sum_{j\in[N]} (1 + r\alpha_j) v_j = 0$$

Hence we have

$$\begin{aligned} & D_{ID_\eta}^{(2)\sum_{j\in[N]} v_j} \cdot (H_1(ID_\eta)^s \prod_{j\in[N]} H_2(\tau, j)^{v_j})^{rx} \\ = \; & g^{xy\sum_{j\in[N]} v_j} \cdot (g^{ys} g^{y\sum_{j\in[N]} \alpha_j v_j} g^{\sum_{j\in[N]} \beta_j v_j})^{rx} \\ = \; & g^{xy(\sum_{j\in[N]} v_j + rs + r\sum_{j\in[N]} \alpha_j v_j)} \cdot g^{rx\sum_{j\in[N]} \beta_j v_j} \\ = \; & g^{rx\sum_{j\in[N]} \beta_j v_j} = w^{\sum_{j\in[N]} \beta_j v_j} = \sigma_2 \end{aligned}$$

Then Q is associated with a new handle h and B stores $(h, (Q, ID_\eta, \mathbf{v}, \tau))$ in T.

Finally, the signature Q is not directly returned to A but associated with a new handle h.

Derivation and Reveal queries. B proceeds as the real oracle.

Output. Finally, the adversary A is supposed to output a forgery $(ID^*, \mathbf{v}^*, \tau^*, Q^*)$ such that $\mathbf{v}^* = (v_1^*, ..., v_N^*)$, $Q^* = (w^*, \sigma_1^*, \sigma_2^*, s^*)$ and $\mathbf{Hverify}(ID^*, \mathbf{v}^*, \tau^*, Q^*) = 1$.

- If $ID^* \neq ID_\eta$, B aborts.
- If $ID^* = ID_\eta$ with the probability $\frac{1}{q_{H_1}}$, then it proceeds as follows.

Given the file identifier τ^*, B gets the corresponding (r, w) from the table L. Note that $Q^* = (w^*, \sigma_1^*, \sigma_2^*, s^*)$ is a Type 2 forgery, so $w^* = w = g^{rx}$ (Otherwise, if $w^* \neq w$, then $((\tau^*, w^*), \sigma_1^*)$ is a forgery for $\mathbf{S}$). And (σ_2^*, s^*) can satisfy the second verification equation

$$\begin{aligned} & e(\sigma_2^*, g) \\ = \; & e(H_1(ID_\eta), P_{pub})^{\sum_{j\in[N]} v_j^*} \\ & \cdot e(H_1(ID_\eta)^{s^*} \prod_{j\in[N]} H_2(\tau^*, j)^{v_j^*}, w^*) \end{aligned}$$

Assuming w.l.o.g A has requested H_1-**queries** on ID_η and H_2-**queries** on τ^*, B checks H_1-**List**, and gets $H_1(ID_\eta) = g^y$. Furthermore, B gets $H_2(\tau^*, j) = (g^y)^{\alpha_j} g^{\beta_j}$ for $j \in [N]$ from H_2-**List**. Then we have

$$\begin{aligned} & e(\sigma_2^*, g) \\ = \; & e(g^y, g^x)^{\sum_{j\in[N]} v_j^*} \cdot e(g^{y(s^* + \sum_{j\in[N]} \alpha_j v_j^*)} g^{\sum_{j\in[N]} \beta_j v_j^*}, g^{rx}) \\ = \; & e(D_{ID_\eta}^{(2)\,\sum_{j\in[N]} v_j^*}, g) \\ & \cdot e(D_{ID_\eta}^{(2)\,r(s^* + \sum_{j\in[N]} \alpha_j v_j^*)} w^{\sum_{j\in[N]} \beta_j v_j^*}, g) \\ = \; & e(D_{ID_\eta}^{(2)\,(rs^* + \sum_{j\in[N]} (1 + r\alpha_j) v_j^*)} \cdot w^{\sum_{j\in T} \beta_j v_j^*}, g) \end{aligned}$$

So by the non-degenerate property, we have

$$\sigma_2^* = D_{ID_\eta}^{(2)\,(rs^* + \sum_{j\in[N]} (1 + r\alpha_j) v_j^*)} \cdot w^{\sum_{j\in T} \beta_j v_j^*}$$

If $s^* \neq -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j^*$, then it holds that

$$rs^* + \sum_{j\in[N]} (1 + r\alpha_j) v_j^* \neq 0$$

So B can compute

$$g^{xy} = D_{ID_\eta}^{(2)} = \left(\frac{\sigma_2^*}{w^{\sum_{j\in[N]} \beta_j v_j^*}}\right)^{\frac{1}{rs^* + \sum_{j\in[N]} (1 + r\alpha_j) v_j^*}}$$

Then the CDH problem is solved.

Now we only need to show that $s^* = -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j^*$ with probability $\frac{1}{q}$. We use a technique analysis similar to that in the analysis of [25]. According to the above assumption, we know that A only queries the HSigning oracle on independent vectors for each file identifier τ. Since all of the signed message vectors are $N-$dimensional vectors, we assume w.l.o.g. that A makes at most $N-1$ HSigning queries for the file identifier τ^* (Otherwise, the signed message vectors $\{\mathbf{v}^{(1)}, ..., \mathbf{v}^{(N)}\}$ for file identifier τ^* will compose a maximal linear independent group. So B can simulate HSigning queries for itself). Assume that A makes exactly $N-1$ Reveal queries with the file τ^* and gets $Q_i = (w^{(i)}, \sigma_1^{(i)}, \sigma_2^{(i)}, s^{(i)})$ for $\mathbf{v}^{(i)} = (v_1^{(i)}, ..., v_N^{(i)}), i \in \{1, ..., N-1\}$, where $w^{(1)} = w^{(2)} = ... = w^{(N-1)}$. B can check the table L and retrieve the corresponding r. And A has $N-1$ values $s_1, ..., s_{N-1} \in Z_q^*$ such that $s_i = -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j^{(i)}$ for $i \in \{1, ..., N-1\}$. So we have

$$\begin{cases} -s_1 = (\frac{1}{r} + \alpha_1) v_1^{(1)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(1)} \\ -s_2 = (\frac{1}{r} + \alpha_1) v_1^{(2)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(2)} \\ ...... \\ -s_{N-1} = (\frac{1}{r} + \alpha_1) v_1^{(N-1)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(N-1)} \end{cases}$$

Furthermore, according to the definition of Type 2 forgery, $\mathbf{v}^* \notin \text{span}\,(\mathbf{v}^{(1)}, ..., \mathbf{v}^{(N-1)})$. And A only queries the HSigning oracle on independent vectors for τ^*, so $\{\mathbf{v}^{(1)}, ..., \mathbf{v}^{(N-1)}\}$ in Reveal queries for τ^* composes a linear independent group. Then $\mathbf{v}^{(1)}, ..., \mathbf{v}^{(N-1)}, \mathbf{v}^{(*)}$ for τ^* will be N linear independent vectors.

Assume A can output the forgery $Q^* = (w^*, \sigma_1^*, \sigma_2^*, s^*)$ for the message vector $\mathbf{v}^* = (v_1^*, ..., v_N^*)$ such that $s^* = -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j^*$, then combined with the above equations, we have

$$\begin{cases} -s_1 = (\frac{1}{r} + \alpha_1) v_1^{(1)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(1)} \\ -s_2 = (\frac{1}{r} + \alpha_1) v_1^{(2)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(2)} \\ ...... \\ -s_{N-1} = (\frac{1}{r} + \alpha_1) v_1^{(N-1)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(N-1)} \\ -s^* = (\frac{1}{r} + \alpha_1) v_1^{(*)} + ... + (\frac{1}{r} + \alpha_N) v_N^{(*)} \end{cases}$$

Notice that $\mathbf{v}^{(1)}, ..., \mathbf{v}^{(N-1)}, \mathbf{v}^{(*)}$ for τ^* are N linear independent vectors. So

$$\begin{vmatrix} v_1^{(1)} & ... & v_N^{(1)} \\ v_1^{(2)} & ... & v_N^{(2)} \\ ... & & \\ v_1^{(N-1)} & ... & v_N^{(N-1)} \\ v_1^{(*)} & ... & v_N^{(*)} \end{vmatrix} \neq 0$$

According to Cramer's Rule, A can get $\{\frac{1}{r} + \alpha_j\}_{j\in[N]}$ and recover the values $\{\alpha_j\}_{j\in[N]}$. But the random numbers $\{\alpha_j\}_{j\in[N]}$ are independent of A's view. So we conclude that the equation $s^* = -\sum_{j\in[N]} (\frac{1}{r} + \alpha_j) v_j^*$ holds randomly and the probability is $\frac{1}{q}$.

Let us analyze B's probability of success. It is not hard to see that if the adversary has advantage ϵ in forging the signature scheme, then B has probability at least $\frac{\epsilon}{2q_{H_1}}(1 - \frac{1}{q})$ in solving the CDH problem.

This completes the proof.

■

V. Application in Data Analysis and Blockchain

Recently, blockchain has attracted more and more attention from both academy and industry because of its decentralization. The blockchain has been widely applied in the smart contract and financial transactions, and data forensics in IoT as well. In this section, we will show how to use the homomorphic signature scheme in the construction in the IoT for the data and computation authentication. With the advent of blockchain, more and more users are using the blockchain to store their personal data in blockchain as an access control service because of the properties of unforgettability in the blockchain. When the volume of data grows, the data utilization will be valuable for data analysis. As we know, big data transactions have been common for the promotion of data service. However, with the utilization of the third-party service provider such as cloud computing, the big data computation such as data mining and machine learning over the data can be performed at the third-party's side. As a result, the user can get the result from the cloud server while relieving the computation overhead. Though the advantages of the cloud computing techniques, another issue arises, that is, how to achieve the characteristic of authentication of the original data and the results. In more details, if the cloud server return a wrong computation result that is not computed from the user's data, it will be difficult for the receiver to detect.

To overcome this challenge, we show how to use the homomorphic signature to achieve the data authentication and guarantee the correctness of the computation results.

At first, the users involved in our system generate their own public keys for the blockchain system, that is, the virtual identity for our homomorphic signature. To upload the data in the cloud or distributed storage nodes while providing the outsourced computation service, the user first generates the signature for all the data stored in the nodes. Then, the data and the signature will be uploaded to the storage nodes in the network. Furthermore, the pointer to the data, including the access information will be computed and stored in the blockchain. To access the data, the users first access and get the pointer from the blockchain. If they are allowed to access the data, they can further download and compute the data. To reduce the communication and computation overhead, the users may ask the nodes to compute the data with any function they provide. The reason is that all the data are signed with homomorphic signature and any function can be compute with this type of signature.

After the computation, the users will be able to get the computation results as well as the aggregate homomorphic signature for this computation results. After that, the users are able to verify the computation results with the virtual identity. If the signature is valid, the users accept the computation result from the nodes. Otherwise, it means that the computation results are invalid. With the blockchain techniques, it also guarantees that the data owner or service provides such as the nodes can get a fair payment after they provide the data or computation.

The security of the above scheme can be easily analyzed with the property of blockchain and homomorphic signature. With the technique of blockchain, any user, including the data owner and the nodes cannot change the data information and their digital signature. Furthermore, the fairness can be guaranteed with the blockchain without a centralized party. With the technique of the homomorphic signature, data computation can be performed while keeping the authentication. The results from the nodes can be verified by checking the signature.

VI. Conclusion

In this paper, we first formally introduce the concept and security model of ID-based linearly homomorphic signature, then design a new ID-based linearly homomorphic signature scheme. The scheme allows a signer to produce linearly homomorphic signature and avoids the shortcomings of the use of public-key certificates. Moreover, the scheme is proved secure against existential forgery on adaptively chosen message and ID attack under the random oracle model. ID-based linearly homomorphic signature schemes can be applied in e-business, cloud computing and blockchain.

Acknowledgment

The third author was supported by National Natural Science Foundation of China (No. 61702125), and Scientific Research Foundation for Post-doctoral Researchers of Guangzhou (No. gdbsh2016020). The fourth author was partly supported by the Program for Innovative Research Team in Education Department of Guangdong Province Under Grant No.2015KCXTD014 and No.2016KCXTD017.

References

[1] A. Shamir. *Identity-based cryptosystems and signature schemes.* Crypto. 1984, 84: 47-53.
[2] R. Johnson, D. Molnar, D. Song, et al. *Homomorphic signature schemes.* CT-RSA. 2002, 2271: 244-262.
[3] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li. *Insight of the Protection for Data Security under Selective Opening Attacks.* Information Sciences, 2017, Volumes 412-413: 223-241.
[4] J. Li, X. Chen, M. Li, J. Li, P. Lee, W. Lou. *Secure Deduplication with Efficient and Reliable Convergent Key Management.* IEEE Transactions on Parallel and Distributed Systems, 2014, 25(6): 1615-1625.
[5] W. Chen, L. Peng, J. Wang, F. Li, M. Tang, W. Xiong, S. Wang. *Inapproximability Results for the Minimum Integral Solution Problem with Preprocessing over infinity Norm.* Theoretical Computer Science, Volume 478, 25 March 2013, Pages 127-131.
[6] W. Chen, L. Peng, J. Wang, F. Li, M. Tang, W. Xiong, S. Wang. *An Improved Lower Bound for Approximating the Minimum Integral Solution Problem with Preprocessing over infinity Norm.* Journal of Combinatorial Optimization, 2015, 30(3): 447-455.
[7] D. Freeman. *Improved security for linearly homomorphic signatures: A generic framework.* In Public Key Cryptography-PKC 2012, Springer, 2012: 697-714.
[8] D. Boneh, D. Freeman, J. Katz, et al. *Signing a linear subspace: Signature schemes for network coding.* Public Key Cryptography. 2009, 5443: 68-87.
[9] J. Li, Y. Li, X. Chen, P. Lee, W. Lou. *A Hybrid Cloud Approach for Secure Authorized Deduplication.* IEEE Transactions on Parallel and Distributed Systems. 2015, 26(5): 1206-1216.
[10] J. Li, X. Chen, X. Huang, S. Tang and Y. Xiang, M. Hassan and A. Alelaiwi. *Secure Distributed Deduplication Systems with Improved Reliability.* IEEE Transactions on Computers. 2015, 64(12): 3569-3579.
[11] J. Choon, J. Cheon. *An identity-based signature from gap Diffie-Hellman groups.* International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2003: 18-30.
[12] D. Boneh, D. Freeman. *Homomorphic signatures for polynomial functions.* Eurocrypt. 2011, 6632: 149-168.
[13] N. Attrapadung, B. Libert. *Homomorphic network coding signatures in the standard model.* Public Key Cryptography. 2011, 6571: 17-34.
[14] D. Catalano, D. Fiore, and B. Warinschi. *Homomorphic signatures with efficient verification for polynomial functions.* In Advances in Cryptology-CRYPTO 2014, Springer, 2014: 371-389.
[15] R. Hiromasa, Y. Manabe, T. Okamoto. *Homomorphic signatures for polynomial functions with shorter signatures.* In The 30th Symposium on Cryptography and Information Security, Kyoto. 2013.
[16] J. Li, Y. Zhang, X. Chen, Y. Xiang. *Secure attribute-based data sharing for resource-limited users in cloud computing.* Computers and Security, 2018, 72: 1-12. DOI:10.1016/j.cose.2017.08.007

[17] X. Boyen, X. Fan, E. Shi. *Adaptively secure fully homomorphic signatures based on lattices.* IACR Cryptology ePrint Archive, 2014, 916.

[18] P. Li, J. Li, Z. Huang, C. Gao, W. Chen, K. Chen. *Privacy-preserving outsourced classification in cloud computing.* Cluster Computing, 2017:1-10. DOI: 10.1007/s10586-017-0849-9.

[19] S. Gorbunov, V. Vaikuntanathan, D. Wichs. *Leveled fully homomorphic signatures from standard lattices*. Cryptology ePrint Archive, Report 2014/897, 2014. http://eprint.iacr.org/.

[20] N. Attrapadung, B. Libert, T. Peters. *Efficient completely context-hiding quotable and linearly homomorphic signatures.* Public Key Cryptography. 2013: 386-404.

[21] J. Li, J. Li, X. Chen, C. Jia and W. Lou. *Identity-based Encryption with Outsourced Revocation in Cloud Computing.* IEEE Transactions on Computers. 2015, 64(2): 425-437.

[22] D. Boneh, B. Lynn, H. Shacham. *Short signatures from the Weil pairing.* Advances in Cryptology-ASIACRYPT 2001, LNCS 2248, 514-532.

[23] D. Boneh, D. Freeman. *Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures.* Public Key Cryptography. 2011, 6571: 1-16.

[24] D. Catalano, A. Marcedone, O. Puglisi. *Authenticating Computation on Groups: New Homomorphic Primitives and Applications.* International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2014: 193-212.

[25] N. Attrapadung, B. Libert, T. Peters. *Computing on authenticated data: new privacy definitions and constructions.* Advances in Cryptology-ASIACRYPT 2012, 2012: 367-385.

[26] W. Chen, H. Lei, K. Qi. *Lattice-Based Linearly Homomorphic Signatures in the Standard Model.* Theoretical Computer Science, 2016, Vol 634. pp: 47-54.

[27] P. Li, J. Li, Z. Huang, T. Li, C. Gao, S. Yiu, K. Chen. *Multi-key privacy-preserving deep learning in cloud computing.* Future Generation Computer Systems, 2017, 74: 76-85.

[28] J. Wu, Y. Tseng, S. Huang. *Leakage-resilient id-based signature scheme in the generic bilinear group model.* Security and communication Networks, 2016, 9(17): 3987-4001.

[29] P. Sarde, A. Banerjee. *A Secure ID-based Proxy Signature Scheme from Bilinear Pairings.* International Journal of Computer Applications, 2015, 124(9).

[30] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. SIAM Journal on Computing, 2003, 32(3): 586-615.

[31] C. Meshram. *An efficient ID-based cryptographic encryption based on discrete logarithm problem and integer factorization problem.* Information Processing Letters, 2015, 115(2): 351-358.

[32] G. Schmid, F. Rossi. *Implementing identity-based key agreement in embedded devices*. International Conference on Pervasive and Embedded Computing and Communication Systems. IEEE, 2016: 1-7.

[33] P. Sarkar, M. Chowdhury. *Inductive Hierarchical Identity Based Key Agreement with Pre-deployment Interactions*. International Conference on Applications and Techniques in Information Security. Springer Singapore, 2016: 106-114.

[34] Y. Zhang, Y. Jiang, B. Li, et al. *An Efficient Identity-Based Homomorphic Signature Scheme for Network Coding.* International Conference on Emerging Internetworking, Data and Web Technologies. Springer, Cham, 2017: 524-531.

[35] S. Sadrhaghighi, S. Khorsandi. *An identity-based digital signature scheme to detect pollution attacks in intra-session network coding.* Information Security and Cryptology (ISCISC), 2016 13th International Iranian Society of Cryptology Conference on. IEEE, 2016: 7-12.

[36] J. Li, Z. Liu, X. Chen, et al. L-EncDB: *A Lightweight Framework for Privacy-Preserving Data Queries in Cloud Computing.* Knowledge-based Systems. 2015, 79: 18-26.

[37] F. Wang, K. Wang, B. Li, and Y. Gao. *Leveled Strongly-Unforgeable Identity-Based Fully Homomorphic Signatures.* ISC 2015, LNCS, 9290, 2015: 42-60.

[38] D. Catalano. *Homomorphic signatures and message authentication codes.* In Security and Cryptography for Networks, Springer, 2014: 514-519.

[39] D. Catalano, D. Fiore, and B. Warinschi. *Adaptive pseudo-free groups and applications.* In Advances in Cryptology-EUROCRYPT 2011, Springer, 2011: 207-223.

[40] D. Charles, K. Jain, and K. Lauter. *Signatures for network coding.* International Journal of Information and Coding Theory, 2009, 1(1): 3-14.

[41] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin. *Secure network coding over the integers.* In Public Key Cryptography-PKC 2010, Springer, 2010: 142-160.