IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INFORMATION SECURITY SOLUTIONS FOR TELEMEDICINE APPLICATIONS

Received January 8, 2018, accepted February 13, 2018, date of publication February 27, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2809684

A Short Linearly Homomorphic Proxy Signature Scheme

QUN LIN[1], JIN LI[2], ZHENGAN HUANG[2], WENBIN CHEN[2], AND JIAN SHEN[3]

[1]Institute of Mathematics and Statistics, Hanshan Normal University, Chaozhou 521041, China
[2]School of Computer Science, Guangzhou University, Guangzhou 510006, China
[3]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Corresponding author: Jin Li (jinli71@gmail.com)

The work of J. Li was supported in part by the Natural Science Foundation of Guangdong Province for Distinguished Young Scholars under Grant 2014A030306020, in part by the Guangzhou Scholars Project for Universities of Guangzhou under Grant 1201561613, in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2015B010129015, in part by the National Natural Science Foundation of China under Grant 61472091, in part by the National Natural Science Foundation for Outstanding Youth Foundation under Grant 61722203, and in part by the State Key Laboratory of Cryptology, Beijing, China. The work of Z. Huang was supported in part by the National Natural Science Foundation of China under Grant 61702125 and in part by the Scientific Research Foundation for Post-doctoral Researchers of Guangzhou under Grant gdbsh2016020. The work of W. Chen was supported by the Program for Innovative Research Team in Education Department of Guangdong Province under Grant 2015KCXTD014 and Grant 2016KCXTD017.

ABSTRACT Linearly homomorphic signature schemes allow the performance of linear computations on authenticated data. They are important primitives for many applications, such as electronic voting, smart grids, electronic health records, and so on. Proxy signature schemes allow an original signer to delegate his/her signing power to a proxy signer, so that the proxy signer can sign on behalf of the original signer. Therefore, a signature scheme offering both of the above signatures' properties is very desirable. In this paper, we construct the first linearly homomorphic proxy signature scheme, so the proxy signer can produce a linearly homomorphic signature on behalf of the original signer. The scheme is provably secure in the random oracle model. Moreover, the length of signature is short and constant. Linearly homomorphic proxy signature scheme can be used in applications, such as electronic business and cloud computing.

INDEX TERMS Homomorphic signatures, proxy signature, bilinear pairings, random oracle.

I. INTRODUCTION

The conception of homomorphic signatures was originally proposed by Johnson *et al.* [1] in 2002. Homomorphic signature schemes are important primitives and allow to validate computation over authenticated data [9], [20]–[22], [28], [29], [32]–[34]. Informally, a signer holding a dataset $\{\mathbf{V}^{(i)}\}_{i=1}^{l}$ can produce corresponding signatures $\sigma_i = Sign(SK, \mathbf{V}^{(i)})$ for $i = 1, \ldots, l$ and store the signed dataset on a remote server. Later the server can publicly compute a succinct valid signature σ on $\mathbf{V} = \mathbf{f}(\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(l)})$. A keynote feature of homomorphic signatures is that the homomorphic signature σ can be computed without needing to know the original secret key. In the last years, various types of homomorphic signature schemes have been proposed. The first schemes proposed were only suitable for performing linear computations on authenticated data [2]–[7]. Then solutions have been developed to support polynomial functions [8], [9], [16]. Now, without any restrictions on the functions themselves, leveled fully homomorphic signature schemes have been designed [17], [18]. Homomorphic signature schemes can be employed in electronic business and cloud computing [10]–[15].

The concept of proxy signatures was first introduced by Mambo *et al.* [23] in 1996. Proxy signature schemes enable an original signer to delegate his/her signing capability to a proxy signer, and then the proxy signer can sign a message on behalf of the original signer. In 2012, Boldyreva *et al.* [24] gave the definition of proxy signatures in detail and formalized a model of security for proxy signature schemes. Furthermore, they specified the adversary's capabilities and goals. In their model, a public key infrastructure setting (PKI) is also assumed, where each entity holds a public and secret key pair. As usual, each user can sign messages using the signing algorithm of a standard digital signature scheme. A provably secure proxy signature scheme was also proposed in this model. Although the scheme is lack of efficiency, it can pay attention to the importance of security model [25]–[27], [30], [31], [35], [36]. According to the

delegation level, proxy signature schemes can be divided into full delegation, partial delegation and delegation by warrant. Proxy signature schemes have shown to be useful in many applications. Nowadays, various types of proxy signature schemes have been proposed, and they can offer the mixed natures of signatures, such as strong proxy signatures [37], proxy ring signatures [38], [39], and proxy blind signatures [40], [41]. But up to our knowledge, there are no schemes which combine the natures of linearly homomorphic signatures and proxy signatures.

In this paper, the concept of linearly homomorphic proxy signatures **(LHPS)** is proposed for the first time. It means that the proxy signer can produce linearly homomorphic signatures on behalf of the original signer. Suppose Alice wants to produce linearly homomorphic signatures, but she may be on vocation. So she can delegate Bob to generate signatures on behalf of her. Anyone can verify the validity of the signatures and perform linear computations on authenticated data. In a word, linearly homomorphic proxy signatures can combine the natures of linearly homomorphic signatures and proxy signatures.

A. OUR CONTRIBUTIONS

In this paper, we introduce the notion and security model of linearly homomorphic proxy signature schemes, and design a new LHPS scheme from bilinear pairings. We prove that our scheme is secure in the random oracle model. Moreover, our signature is an element of a circle group, so its length is very short.

B. 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 linearly homomorphic proxy signature schemes and the security model. Section 3 gives a new linearly homomorphic proxy signature, and Section 4 gives the security and efficiency analysis 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 bilinear groups which satisfy $|G_1|=|G_2|=q$ for some prime number q.

$e : G_1 \times G_1 \rightarrow G_2$ 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: If g is a generator of G_1, then $e(g, g)$ is a generator of G_2. In other words, $e(g, g) \neq 1$;

(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).** Given a random generator $g \in G_1$, if there exists no probabilistic polynomial-time (PPT) algorithm A that on input (g, g^x, g^y) outputs g^{xy} with non-negligible probability, we say that the CDH assumption holds in G_1. Here the probability is taken over the uniform choices of $x, y \leftarrow Z_q{}^*$ and the internal coin tosses of A.

B. BLS SHORT SIGNATURE SCHEME

The BLS short signature scheme proposed in [19] 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 bilinear map $e : G_1 \times G_1 \rightarrow G_2$ and a full-domain hash function $H : \{0, 1\}^* \rightarrow G_1$, and g is a random generator of G_1.

KeyGen: The secret key is $x \in Z_q^*$, and the public key is $PK = g^x$.

Sign: Given a secret key x, and a message m, compute the signature $\sigma = H(m)^x$.

Verify: Given a public key PK, a message m and a signature σ, verify if the equation $e(\sigma, g) = e(H(m), PK)$ holds, this algorithm outputs 1; otherwise it outputs 0.

The security of the BLS short signature scheme is based on the CDH assumption. We refer to [19] for more details.

C. LINEARLY HOMOMORPHIC PROXY SIGNATURE

Definition 2: **(LHPS)**. A linearly homomorphic proxy signature (LHPS) scheme consists of six algorithms :(**Setup, KeyGen, Delegation, PSign, PVerify, Combine**). The algorithms are defined as follows:

- **Setup:** This algorithm takes a security parameter λ and an integer l as input, and returns the string params, which denotes the common scheme parameters. Notice that l denotes an upper bound for the number of messages signed in each file.
- **KeyGen:** This algorithm takes the system parameters as input and returns a secret/public key pair (SK, PK) for a user in the system.
- **Delegation:** The original signer A creates a warrant m_{ω_B} related to the proxy signer B, then interacts with B by a series of interactive algorithms forming the delegation protocol. As a result of the interaction, the final output of the protocol is a proxy key S_p that the proxy signer B uses to produce proxy signatures on behalf of the original signer A.
- **PSign:** On input a proxy key S_p, a file identifier $\tau \in \{0, 1\}^\lambda$, and a message vector $\mathbf{V}$, this algorithm outputs the proxy signature σ.
- **PVerify:** Given the public key PK_0 for the original signer A, the public key PK_B for the proxy signer B, a warrant m_{ω_B}, a file identifier τ, a message vector $\mathbf{V}$ and a proxy signature σ, it outputs 1 (accept) or 0 (reject).
- **Combine:** Given PK_0, PK_B, a warrant m_{ω_B}, a file identifier τ, and a set of tuple $\{(f_i, \sigma_i)\}_{i=1}^{l}$, 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 $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \ldots, \mathbf{V}^{(l)}$*)*.

Correctness. For correctness, we require:
(1) For all message vector $\mathbf{V}$ and all file identifier $\tau \in \{0,1\}^\lambda$, if $\sigma \leftarrow$ **PSign**$(S_p, \tau, \mathbf{V})$, then

$$\textbf{PVerify}(PK_0, PK_B, m_{\omega_B}, \tau, \mathbf{V}, \sigma) = 1.$$

(2) For all $\tau \in \{0,1\}^\lambda$, and all sets of triples $\{(f_i, \sigma_i, \mathbf{V}^{(i)})\}_{i=1}^{l}$, if it holds that

$$\textbf{PVerify}(PK_0, PK_B, m_{\omega_B}, \tau, \mathbf{V}^{(i)}, \sigma_i) = 1$$

for all $i = 1, \ldots, l$, then

$$\textbf{PVerify}(PK_0, PK_B, m_{\omega_B}, \tau, \sum_{i=1}^{l} f_i \mathbf{V}^{(i)},$$
$$\textbf{Combine}(PK_0, PK_B, m_\omega, \tau, \{(f_i, \sigma_i)\}_{i=1}^{l})) = 1.$$

Security Model: We should consider two types of unforgeability in LHPS: **Delegation unforgeability** and **Linearly homomorphic proxy signature unforgeability.** Delegation unforgeability means that if the adversary does not obtain the targeted delegation from the original signer, it is hard to output a forgery of the targeted proxy signature. Linearly homomorphic proxy signature unforgeability means that, except the proxy signer, anyone else including the origin signer cannot generate a valid linearly homomorphic proxy signature on behalf of the proxy signer. So we can divide the adversaries into the following two types: **Type I**, the adversary A has the key pair (SK_j, PK_j) for the proxy signer j, but it can not obtain the delegation from the original signer; **Type II**, the adversary obtains the delegation from the original signer, but it has no secret key of the challenged proxy signer.

Now we introduce the security models in detail as follows:

Definition 3 (Delegation Unforgeability): A LHPS = (**Setup, KeyGen, Delegation, PSign, PVerify, Combine**) has delegation unforgeability if the advantage of any PPT Type I adversary A in the following security game is negligible in the security parameter λ (Note that in this model, the adversary A does not obtain the targeted delegation from the original signer).

The challenger C sets $(SK_0, PK_0) \leftarrow$ **KeyGen** (1^λ) as the secret/public key for the original signer, then gives PK_0 and the system parameter *params* to A. The *params* define a message space and a signature space. Note that l denotes an upper bound for the number of messages signed in each file.
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:

1) **KR queries.** Given a key pair (SK_i, PK_i), C first checks if (SK_i, PK_i) is a valid key pair. If it is true, then (SK_i, PK_i) is stored in a list. Otherwise, C rejects and outputs a special symbol $\perp$.
2) **DE queries.** Given PK_0 and any registered public key PK_i, C returns a warrant m_{ω_i} and a delegation key S_{w_i} corresponding to the warrant m_{ω_i}.
3) **Signing queries.** Given any registered public key PK_i, PK_0, a file identifier $\tau \in \{0,1\}^\lambda$, and a message vector $\mathbf{V}$, C outputs the signature σ including the corresponding warrant m_{ω_i}.

Output: A outputs a public key PK_j, PK_0, a warrant m_{ω_j}, a file identifier $\tau^* \in \{0,1\}^\lambda$, a message vector $\mathbf{V}^*$ and a signature σ^*.

The adversary wins if **PVerify**$(PK_0, PK_j, m_{w_j}, \tau^*, \mathbf{V}^*, \sigma^*) = 1$, and it must satisfy that the public key PK_j does not appear in DE queries and signing queries.

The advantage of the adversary is the probability that he wins the above game.

Definition 4 (Linearly Homomorphic Proxy Signature Unforgeability): A LHPS = (**Setup, KeyGen, Delegation, PSign, PVerify, Combine**) has linearly homomorphic proxy signature unforgeability if the advantage of any PPT Type II adversary A in the following security game is negligible in the security parameter λ (Note that in this model, the adversary A has no secret key of the challenged proxy signer).

The challenger C sets $(SK_0, PK_0) \leftarrow$ **KeyGen** (1^λ) as the secret/public key for the original signer and $(SK^*, PK^*) \leftarrow$ **KeyGen** (1^λ) as the secret/public key for the targeted proxy signer. Then C gives (SK_0, PK_0), PK^* and the system parameter *params* to A. The *params* define a message space and a signature space. Similarly, l denotes an upper bound for the number of messages signed in each file.
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:

1) **KR queries.** Given a key pair (SK_i, PK_i), C first checks if (SK_i, PK_i) is a valid key pair. If it is true, then (SK_i, PK_i) is stored in a list. Otherwise, C rejects and outputs a special symbol $\perp$.
2) **Signing queries.** Given the targeted public key PK^*, PK_0, a warrant m_{ω^*}, a file identifier $\tau \in \{0,1\}^\lambda$, and a message vector $\mathbf{V}$, C outputs the signature σ.

Output: A outputs the targeted public key PK^*, PK_0, a warrant m_{ω^*}, a file identifier $\tau^* \in \{0,1\}^\lambda$, a message vector $\mathbf{V}^* \neq \mathbf{0}$ and a signature σ^*.

The adversary wins if **PVerify**$(PK_0, PK^*, m_{\omega^*}, \tau^*, \mathbf{V}^*, \sigma^*) = 1$, and the file identifier τ^* does not appear in signing queries.

The advantage of the adversary is the probability that he wins the above game.

III. THE PROPOSED SCHEME

In this section, we propose a provably secure linearly homomorphic proxy signature from bilinear pairings. And our proxy signature scheme belongs to delegation by warrant. So the original signer makes a warrant m_ω before delegation. The warrant m_ω contains some explicit information including the description of the delegation relation.

Now we construct the new scheme as follows:

1) **Setup:** Let (G_1, G_2) be bilinear groups satisfying $|G_1| = |G_2| = q$ for some prime number q and g be the generator of G_1. The bilinear map is given by e: $G_1 \times G_1 \rightarrow G_2$. Define two hash functions

$H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \times Z \to G_1$. H_1 and H_2 will be viewed as random oracles in our security proof. Let $[l] = \{1, \ldots, l\}$, $[N] = \{1, \ldots, N\}$. The security parameter is λ. The system parameter $params = (G_1, G_2, q, g, e, \lambda, H_1, H_2)$. Assume that $params$ defines the file identifier space $\mathcal{ID}$, where $|\mathcal{ID}| = poly(\lambda)$.

2) **KeyGen:** The original signer chooses a secret key $x_0 \in Z_q^*$ and the public key is $PK_0 = g^{x_0}$. The secret key of the proxy signer B is $x_B \in Z_q^*$ and the corresponding public key is $PK_B = g^{x_B}$.
3) **Delegation:** The original signer generates a standard warrant m_ω related to the proxy signer B, computes $S_\omega = H_1(m_w)^{x_0}$, and sends S_ω to B. Then B verifies the equation $e(S_\omega, g) = e(H_1(m_w), PK_0)$. If the equation holds, then the proxy signer B gets the proxy key $S_p = (x_B, S_\omega)$.
4) **PSign:** Given a message vector $\mathbf{V} = (v_1, \ldots, v_N) \in Z_q^N$, a proxy key S_p, and a file identifier $\tau \in \mathcal{ID}$, it returns $\perp$ if $\sum_{j\in[N]} v_j = 0$. Otherwise, the proxy signer B can compute

$$\sigma = S_\omega^{\sum_{j\in[N]} v_j} \cdot (\prod_{j\in[N]} H_2(\tau, j)^{v_j})^{x_B}$$

5) **PVerify:** Given PK_0, PK_B, a warrant m_ω, a file identifier τ, a message vector $\mathbf{V} = (v_1, \ldots, v_N) \in Z_q^N$, and a proxy signature σ, return 0 if $\sum_{j\in[N]} v_j = 0$. Otherwise, the verifier checks if

$$e(\sigma, g) = e(H_1(m_w), PK_0)^{\sum_{j\in[N]} v_j} \cdot e(\prod_{j\in[N]} H_2(\tau, j)^{v_j}, PK_B)$$

If the equation holds, output 1; otherwise, output 0.
6) **Combine:** Given PK_0, PK_B, a warrant m_ω, a file identifier τ and a set of tuple $\{(f_i, \sigma_i)\}_{i=1}^l$, this algorithm outputs a signature $\sigma \leftarrow \prod_{i\in[l]} \sigma_i^{f_i}$.

Correctness:

Given PK_0, PK_B, a warrant m_ω, a file identifier τ, a message vector $\mathbf{V} = (v_1, \ldots, v_N) \in Z_q^N$ and a proxy signature σ, if $\sigma \leftarrow$ **PSign**$(S_p, \tau, \mathbf{V})$, the correctness of the scheme can be verified by the following equations:

$$\begin{aligned} e(\sigma, g) &= e(S_\omega, g)^{\sum_{j\in[N]} v_j} \cdot e((\prod_{j\in[N]} H_2(\tau, j)^{v_j})^{x_B}, g) \\ &= e(H_1(m_w), PK_0)^{\sum_{j\in[N]} v_j} \\ &\quad \cdot e(\prod_{j\in[N]} H_2(\tau, j)^{v_j}, PK_B) \end{aligned}$$

Furthermore, given $\tau \in \mathcal{ID}$ and all sets of triples $\{(f_i, \sigma_i, \mathbf{V}_i)\}_{i=1}^l$, if $\sigma_i \leftarrow$ **PSign**$(S_p, \tau, \mathbf{V}^{(i)})$, then by our definition of **Combine**, we have $\sigma \leftarrow \prod_{i\in[l]} \sigma_i^{f_i}$.

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

$$\begin{aligned} e(\sigma_i, g) = e(H_1(m_w), PK_0)^{\sum_{j\in[N]} v_j^{(i)}} \\ \cdot e(\prod_{j\in[N]} H_2(\tau, j)^{v_j^{(i)}}, PK_B). \end{aligned}$$

So by the bilinear property, we have

$$\begin{aligned} e(\sigma, g) &= \prod_{i\in[l]} e(\sigma_i, g)^{f_i} \\ &= e(H_1(m_w), PK_0)^{\sum_{i\in[l]}\sum_{j\in[N]} f_i v_j^{(i)}} \\ &\quad \cdot e(\prod_{j\in[N]} H_2(\tau, j)^{\sum_{i\in[l]} f_i v_j^{(i)}}, PK_B) \\ &= e(H_1(m_w), PK_0)^{\sum_{j\in[N]}\sum_{i\in[l]} f_i v_j^{(i)}} \\ &\quad \cdot e(\prod_{j\in[N]} H_2(\tau, j)^{\sum_{i\in[l]} f_i v_j^{(i)}}, PK_B) \\ &= e(H_1(m_w), PK_0)^{\sum_{j\in[N]} v_j} \\ &\quad \cdot e(\prod_{j\in[N]} H_2(\tau, j)^{v_j}, PK_B) \end{aligned}$$

This completes the proof.

IV. PROPOSED SCHEME ANALYSIS

Theorem 1: Assuming that the type I adversary makes at most q_{H_1}, q_{H_2}, q_D and q_S queries to the H_1, H_2, Delegation and Signing oracles, respectively, the signature scheme has delegation unforgeability on adaptively chosen-message attacks in the random oracle model if the CDH assumption holds in G_1.

Proof: Supposing C is a challenger and A is an adversary, C is given (g, g^x, g^y) in order to output g^{xy}. In this model, the adversary A does not obtain the delegation from the original signer.

First, C runs A on input $PK_0 = g^x$ as the public key of the original signer, then sends the system $params$=$(G_1, G_2, q, g, e, \lambda, H_1, H_2, PK_0)$ to the adversary A and responses as follows:

Key registration queries: We assume that the number of users in the game is q_{H_1}. When A requests to register a new user i by outputting pair (x_i, PK_i), C verifies if they are valid key pairs, then adds (x_i, PK_i) to the **Key-List**.

H_1-**queries:** Assuming w.l.o.g A makes q_{H_1} times to H_1-queries and gets the warrant m_{ω_i} for $1 \le i \le q_{H_1}$ from C before these queries, C randomly chooses $s \in [1, q_{H_1}]$ and $t_i \in Z_q^*$ for $1 \le i \le q_{H_1}$, where s is the targeted proxy signer's number. When A queries m_{ω_i} to H_1-oracle, C answers $H_1(m_{w_i}) = g^{t_i}$ if $i \ne s$; Otherwise, $H_1(m_{w_i}) = g^y$ if

$i = s$. Then C adds $(m_{\omega_i}, g^{t_i}, t_i)_{i \neq s}$ to the H_1-**List**; If $i = s$, C adds $(m_{\omega_s}, g^y, *)$ to the H_1-**List** (Note that $*$ means the corresponding value is unknown).

H_2-**queries:** Assume that A makes q_{H_2} times to H_2-queries. When A queries (τ, j) to H_2- oracle, C randomly chooses $\alpha_{\tau,j} \in Z_q^*$ for $1 \leq j \leq N$ and answers $H_2(\tau, j) = g^{\alpha_{\tau,j}}$. Then C adds $((\tau, j), g^{\alpha_{\tau,j}}, \alpha_{\tau,j})$ to the H_2-**List**.

Delegation queries: When A requests delegation for user i, assuming w.l.o.g A has requested H_1-**queries** on m_{ω_i}, C checks the H_1-**List** and computes $S_{\omega_i} = (g^{t_i})^x = PK_0^{t_i}$ if $i \neq s$; Otherwise, C aborts if $i = s$. Then C adds (i, S_{ω_i}) to the **DG-List**.

Signing queries: Given PK_0, PK_i, a warrant m_{ω_i}, a file identifier $\tau \in \mathcal{ID}$, and a message vector $\mathbf{V} = (v_1, \ldots, v_N) \in Z_q^N$ such that $\sum_{j\in[N]} v_j \neq 0$, assuming w.l.o.g A has requested the above corresponding queries for user i, C checks the **Key-List**, H_2-**List**, **DG-List** and responses as follows:

- if $i \neq s$, C answers the signature

$$\sigma = S_{\omega_i}^{\sum_{j\in[N]} v_j} \cdot \Big(\prod_{j\in[N]} H_2(\tau, j)^{v_j}\Big)^{x_i} = PK_0^{t_i \sum_{j\in[N]} v_j} \cdot \Big(\prod_{j\in[N]} PK_i^{\alpha_{\tau,j} v_j}\Big)$$

- if $i = s$, C aborts.

Output: A outputs a a signature σ^* on a message vector $\mathbf{V}^* = (v_1^*, \ldots, v_N^*) \in Z_q^N$ with respect to PK_0, PK_i, a warrant m_{ω_i}, and a file identifier τ^* such that $\sum_{j\in[N]} v_j^* \neq 0$ and

$$\mathbf{PVerify}(PK_0, PK_i, m_{\omega_i}, \tau^*, \mathbf{V}^*, \sigma^*) = 1$$

- If $i \neq s$, C aborts.
- Otherwise, it holds that σ^* can satisfy the verification equation

$$e(\sigma^*, g) = e(H_1(m_{w_s}), PK_0)^{\sum_{j\in[N]} v_j^*} \cdot e\Big(\prod_{j\in[N]} H_2(\tau^*, j)^{v_j^*}, PK_s\Big)$$

Assuming w.l.o.g A has requested H_1-queries on m_{ω_s} and H_2-queries on τ^*, C checks H_1-**List**, and gets $H_1(m_{\omega_s}) = g^y$. Furthermore, C gets $H_2(\tau^*, j) = g^{\alpha_{\tau^*,j}}$ for $j \in [N]$ from H_2-**List**. Then we have

$$\begin{aligned} e(\sigma^*, g) &= e(g^y, g^x)^{\sum_{j\in[N]} v_j^*} \cdot e(g^{\sum_{j\in[N]} \alpha_{\tau^*,j} v_j^*}, PK_s) \\ &= e(S_{\omega_s}^{\sum_{j\in[N]} v_j^*}, g) \cdot e(PK_s^{\sum_{j\in[N]} \alpha_{\tau^*,j} v_j^*}, g) \\ &= e(S_{\omega_s}^{\sum_{j\in[N]} v_j^*} \cdot PK_s^{\sum_{j\in[N]} \alpha_{\tau^*,j} v_j^*}, g) \end{aligned}$$

So by the non-degenerate property, we have

$$\sigma^* = S_{\omega_s}^{\sum_{j\in[N]} v_j^*} \cdot PK_s^{\sum_{j\in[N]} \alpha_{\tau^*,j} v_j^*}$$

C can compute $g^{xy} = S_{\omega_s} = (\sigma^* / \prod_{j\in[N]} PK_s^{\alpha_{\tau^*,j} v_j^*})^{1/\sum_{j\in[N]} v_j^*}$, then the CDH problem is solved.

We analyze the probability of success for C. There are three sceneries in which C will abort. Assume **E1** means that $i = s$ in Delegation queries; **E2** means that $i = s$ in Signature queries; **E3** means that $i \neq s$ in Output period.

We have $\Pr[\mathbf{E1}] = \frac{q_D}{q_{H_1}}$, $\Pr[\mathbf{E2}] = \frac{q_S}{q_{H_1}}$, $\Pr[\mathbf{E3}] = 1 - \frac{1}{q_{H_1}}$. So if A is an adversary with success probability ε, C can solve the CDH problem with probability $(1 - \frac{q_D}{q_{H_1}})(1 - \frac{q_S}{q_{H_1}})\frac{1}{q_{H_1}}\varepsilon$.

This completes the proof. ■

Theorem 2: Assuming that the type II adversary makes at most q_{H_1}, q_{H_2} and q_S queries to the H_1, H_2 and Signing oracles, respectively, the scheme has linearly homomorphic proxy signature unforgeability on adaptively chosen-message attacks in the random oracle model if the CDH assumption holds in G_1.

Proof: Assuming C is a challenger and A is an adversary, C is given (g, g^x, g^y) in order to output g^{xy}. In this model, the adversary A does not obtain the secret key of the challenged proxy signer.

First, C runs A on input $PK^* = g^x$ as the public key of the targeted proxy signer. Moreover, C chooses a random integer $x_0 \in Z_q^*$ and sets $PK_0 = g^{x_0}$, then sends the system parameter *params*=$(G_1, G_2, q, g, e, \lambda, H_1, H_2, x_0, PK_0, PK^*)$ to A and responses as follows:

Key registration queries: When A requests to register a new user i by outputting pair (x_i, PK_i), C verifies if they are valid key pairs, and then adds (x_i, PK_i) to the **Key-List**.

H_1-**queries:** Assuming A makes q_{H_1} times to H_1-queries, C randomly chooses $t_i \in Z_q^*$ for $1 \leq i \leq q_{H_1}$. When A queries m_{ω_i} to H_1-oracle, C answers $H_1(m_{w_i}) = g^{t_i}$. Then C adds (m_{w_i}, g^{t_i}, t_i) to the H_1-**List**.

H_2-**queries:**Assume that A makes q_{H_2} times to H_2-queries. C randomly chooses $s \in [N]$ and a file identifier $\tau^* \in \{0, 1\}^\lambda$ as the target. When A queries (τ, j) to H_2- oracle, C randomly chooses $\alpha_{\tau,j} \in Z_q^*$ for $1 \leq j \leq N$ and answers $H_2(\tau, j) = g^{\alpha_{\tau,j}}$ if $(\tau, j) \neq (\tau^*, s)$; Otherwise, $H_2(\tau^*, s) = g^y$. Then C adds $((\tau, j), g^{\alpha_{\tau,j}}, \alpha_{\tau,j})_{(\tau,j)\neq(\tau^*,s)}$ to the H_2-**List**. If $(\tau, j) = (\tau^*, s)$, then C adds $((\tau^*, s), g^y, *)$ to the H_2-**List** (Note that $*$ means the corresponding value is unknown).

Signing queries: Given PK_0, PK^*, a warrant m_{ω^*}, a file identifier $\tau \in \mathcal{ID}$, and a message vector $\mathbf{V} = (v_1, \ldots, v_N) \in Z_q^N$, assuming w.l.o.g A has requested H_1-queries on m_{ω^*} and H_2-queries on τ, C checks and gets the corresponding t^* from H_1-**List**. Furthermore, if $\tau \neq \tau^*$, C gets the corresponding $\alpha_{\tau,j}$ for $j \in [N]$ from H_2-**List**. Then C responses as follows:

- if $\tau \neq \tau^*$, C answers the signature

$$\sigma = S_{\omega^*}^{\sum_{j\in[N]} v_j} \cdot \Big(\prod_{j\in[N]} H_2(\tau, j)^{v_j}\Big)^x = PK_0^{t^* \sum_{j\in[N]} v_j} \cdot PK^{*\sum_{j\in[N]} \alpha_{\tau,j} v_j}$$

- if $\tau = \tau^*$, C aborts.

Output: A outputs a a signature σ^* on a message vector $\mathbf{V}^* = (v_1^*, \ldots, v_N^*) \in Z_q^N$ with respect to PK_0, PK^*, a warrant m_{ω^*}, and a file identifier $\overline{\tau}$ such that $\mathbf{V}^* \neq \mathbf{0}$ and

$$\mathbf{PVerify}(PK_0, PK^*, m_{\omega^*}, \overline{\tau}, \mathbf{V}^*, \sigma^*) = 1$$

- If $\overline{\tau} \neq \tau^*$ or $v_s^* = 0$, C aborts.
- Otherwise, it holds that σ^* can satisfy the verification equation

$$e(\sigma^*, g) = e(H_1(m_{\omega^*}), PK_0)^{\sum_{j\in[N]} v_j^*} \cdot e(\prod_{j\in[N]} H_2(\tau^*, j)^{v_j^*}, PK^*)$$

Assuming w.l.o.g A has requested H_1-queries on m_{ω^*} and H_2-queries on τ^*, C checks H_1-**List**, H_2-**List** and gets the corresponding values. Then we have

$$\begin{aligned} e(\sigma^*, g) &= e(H_1(m_{\omega^*}), g^{x_0})^{\sum_{j\in[N]} v_j^*} \cdot e(g^{\sum_{j\neq s} \alpha_{\tau^*,j} v_j^*} g^{y v_s^*}, g^x) \\ &= e(S_{\omega^*}^{\sum_{j\in[N]} v_j^*}, g) \cdot e(PK^{*\sum_{j\neq s} \alpha_{\tau^*,j} v_j^*} (g^{xy})^{v_s^*}, g) \\ &= e(S_{\omega^*}^{\sum_{j\in[N]} v_j^*} \cdot PK^{*\sum_{j\neq s} \alpha_{\tau^*,j} v_j^*} (g^{xy})^{v_s^*}, g) \end{aligned}$$

So by the non-degenerate property, we have

$$\sigma^* = S_{\omega^*}^{\sum_{j\in[N]} v_j^*} \cdot PK^{*\sum_{j\neq s} \alpha_{\tau^*,j} v_j^*} (g^{xy})^{v_s^*}$$

C can compute $g^{xy} = (\sigma^* / (S_{\omega^*}^{\sum_{j\in[N]} v_j^*} \cdot PK^{*\sum_{j\neq s} \alpha_{\tau^*,j} v_j^*}))^{\frac{1}{v_s^*}}$, then the CDH problem is solved.

We analyze the probability of success for C. There are three sceneries in which C will abort. Assume **E1** means that $\tau = \tau^*$ in Signature queries; **E2** means that $\overline{\tau} \neq \tau^*$ in Output period, **E3** means that $v_s^* = 0$ in Output period.

We have $\Pr[\mathbf{E1}] = \frac{q_S}{poly(\lambda)}$, $\Pr[\mathbf{E2}] = 1 - \frac{1}{poly(\lambda)}$, $\Pr[\mathbf{E3}] = \frac{1}{q}$. So if A is an adversary with success probability ε, C can solve the CDH problem with probability $(1 - \frac{q_S}{poly(\lambda)})\frac{1}{poly(\lambda)}(1 - \frac{1}{q})\varepsilon$.

This completes the proof. ■

Efficiency Analysis: Our scheme has a series of advantages. Firstly, the delegation uses the BLS short signature scheme, so the certification is an element in group G_1. Secondly, our linearly homomorphic proxy signature is also an element in group G_1, so the length of the signature is very short. Our scheme is more efficient and suitable for low-bandwidth communication environments. Finally, the verification of the new proxy signature only requires three pair computations, so it is efficient and practical.

V. CONCLUSION

In this paper, we formally introduce the concept of linearly homomorphic proxy signatures, which allows a proxy signer to produce linearly homomorphic signatures on behalf of the original signer. Moreover, we give the formal security definition and design a linearly homomorphic proxy signature. Then we prove the signature is secure against existentially forgery on adaptively chosen-message attacks in the random oracle model based on the CDH assumption. The length of our signature scheme is very short, so our scheme is suitable for low-bandwidth communication environments. Linearly homomorphic proxy signature schemes can be used in applications such as electronic business and cloud computing.

ACKNOWLEDGMENT

This work was done while the first author was visiting Prof. Jin Li at Guangzhou University.

REFERENCES

[1] R. Johnson, D. Molnar, D. Song, and D. Wagner, "Homomorphic signature schemes," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 2271. New York, NY, USA: Springer, 2002, pp. 244–262.

[2] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 5443. Berlin, Germany: Springer, 2009, pp. 68–87.

[3] N. Attrapadung and B. Libert, "Homomorphic network coding signatures in the standard model," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 6571. Berlin, Germany: Springer, 2011, pp. 17–34.

[4] N. Attrapadung, B. Libert, and T. Peters, "Efficient completely context-hiding quotable and linearly homomorphic signatures," in *Public Key Cryptography*. 2013, pp. 386–404.

[5] W. Chen, H. Lei, and K. Qi, "Lattice-based linearly homomorphic signatures in the standard model," *Theor. Comput. Sci.*, vol. 634, pp. 47–54, Jun. 2016.

[6] D. M. Freeman, "Improved security for linearly homomorphic signatures: A generic framework," in *Public Key Cryptography—PKC*. Berlin, Germany: Springer, 2012, pp. 697–714.

[7] B. Libert, T. Peters, M. Joye, and M. Yung, "Linearly homomorphic structure-preserving signatures and their applications," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 2013, pp. 289–307.

[8] D. Boneh and D. Freeman, "Homomorphic signatures for polynomial functions," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632. Berlin, Germany: Springer, 2011, pp. 149–168.

[9] D. Catalano, D. Fiore, and B. Warinschi, "Homomorphic signatures with efficient verification for polynomial functions," in *Proc. Int. Cryptol. Conf.*, 2014, pp. 371–389.

[10] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018, doi: 10.1016/j.cose.2017.08.007.

[11] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," in *Cluster Computing*. New York, NY, USA: Springer, 2017, pp. 1–10, doi: 10.1007/s10586-017-0849-9.

[12] D. Kalra and D. Ingram, "Electronic health records," in *Information Technology Solutions for Healthcare*. London, U.K.: Springer, 2006, pp. 135–181.

[13] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 425–437, Feb. 2015.

[14] P. Li *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generat. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.

[15] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, and D. S. Wong, "L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing," *Knowl.-Based Syst.*, vol. 79, pp. 18–26, May 2015.

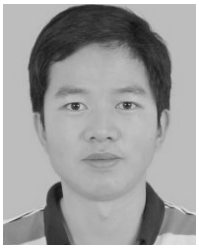
[16] R. Hiromasa, Y. Manabe, and T. Okamoto, "Homomorphic signatures for polynomial functions with shorter signatures," in *Proc. 30th Symp. Cryptograph. Inf. Secur.*, Kyoto, Japan, 2013, pp. 1–8.

[17] X. Boyen, X. Fan, and E. Shi, "Adaptively secure fully homomorphic signatures based on lattices," IACR Cryptol. ePrint Arch., Tech. Rep. 916, 2014.

[18] D. Wichs, "Leveled fully homomorphic signatures from standard lattices," IACR Cryptol. ePrint Arch., Tech. Rep. 2014/897, 2014. [Online]. Available: http://eprint.iacr.org/

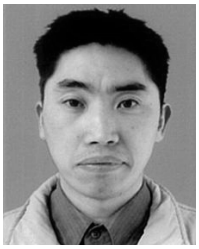
[19] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 2248. Berlin, Germany: Springer, 2001, pp. 514–532.
[20] J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters, "Computing on authenticated data," *J. Cryptol.*, vol. 28, no. 2, pp. 351–395, 2015.
[21] N. Attrapadung, B. Libert, and T. Peters, "Computing on authenticated data: New privacy definitions and constructions," in *Advances in Cryptology—ASIACRYPT 2012*. Berlin, Germany: Springer, 2012, pp. 367–385.
[22] D. Boneh and D. M. Freeman, "Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures," in *Public Key Cryptography—PKC 2011*. Berlin, Germany: Springer, 2011, pp. 1–16.
[23] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proc. 3rd ACM Conf. Comput. Commun. Secur. (CCS)*, 1996, pp. 48–57.
[24] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *J. Cryptol.*, vol. 25, no. 1, pp. 57–115, 2012.
[25] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li, "Insight of the protection for data security under selective opening attacks," *Inf. Sci.*, vols. 412–413, pp. 223–241, Oct. 2017.
[26] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1615–1625, Jun. 2014.
[27] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, May 2015.
[28] D. Catalano, "Homomorphic signatures and message authentication codes," in *Security and Cryptography for Networks*. Berlin, Germany: Springer, 2014, pp. 514–519.
[29] D. Catalano, D. Fiore, and B. Warinschi, "Adaptive pseudo-free groups and applications," in *Advances in Cryptology—EUROCRYPT 2011*. Berlin, Germany: Springer, 2011, pp. 207–223.
[30] W. Chen *et al.*, "Inapproximability results for the minimum integral solution problem with preprocessing over ℓ_∞ norm," *Theor. Comput. Sci.*, vol. 478, pp. 127–131, Mar. 2013.
[31] W. Chen *et al.*, "An improved lower bound for approximating the minimum integral solution problem with preprocessing over ℓ_∞ norm," *J. Combinat. Optim.*, vol. 30, no. 3, pp. 447–455, 2015.
[32] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," *Int. J. Inf. Coding Theory*, vol. 1, no. 1, pp. 3–14, Mar. 2009.
[33] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin, "Secure network coding over the integers," in *Public Key Cryptography—PKC 2010*. Berlin, Germany: Springer, 2010, pp. 142–160.
[34] S. Lee, M. Gerla, H. Krawczyk, K. W. Lee, and E. A. Quaglia, "Performance evaluation of secure network coding using homomorphic signature," in *Proc. Int. Symp. Netw. Coding (NetCod)*, Jul. 2011, pp. 1–6.
[35] J. Li *et al.*, "Secure distributed deduplication systems with improved reliability," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3569–3579, Dec. 2015.
[36] J. Li, Q. Lin, C. Yu, X. Ren, and P. Li, "A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram," *Soft Comput.*, vol. 22, no. 1, pp. 47–65, 2016.
[37] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications," in *Proc. SCIS*, Jan. 2001, pp. 603–608.
[38] M. Asaar, M. Salmasizadeh, and W. Susilo, "A short identity-based proxy ring signature scheme from RSA," *Comput. Standards Interfaces*, vol. 38, pp. 144–151, Feb. 2015.
[39] J. Li, X. Chen, T. H. Yuen, and Y. Wang, "Proxy ring signature: Formal definitions, efficient construction and new variant," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 2. Guangzhou, China, 2006, pp. 1259–1264.
[40] F. Zhang, R. Safavi-Naini, and C. Lin, "New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairing," IACR Cryptol. ePrint Arch., Tech. Rep. 2003/104, 2003.
[41] X.-Q. Cai, Y.-H. Zheng, and R.-L. Zhang, "Cryptanalysis of a batch proxy quantum blind signature scheme," *Int. J. Theor. Phys.*, vol. 53, no. 9, pp. 3109–3115, 2014.

QUN LIN received the B.S. degree from the Department of Atmosphere Science, Nanjing University, in 1999, and the M.S. degree from the School of Mathematics and Computational Science, Sun Yat-sen University, in 2005. He is currently with the Institute of Mathematics and Statistics, Hanshan Normal University. His research interests include public-key cryptography and information security.



JIN LI received the B.S. degree in mathematics from Southwest University, Chongqing, China, in 2002, and the M.S. degree in mathematics and the Ph.D. degree in information security from Sun Yat-sen University, Guangzhou, China, in 2004 and 2007, respectively. He served as a Senior Research Associate for the Korea Advanced Institute of Technology, Daejeon, South Korea, and the Illinois Institute of Technology, Chicago, IL, USA, from 2008 to 2010. He is currently a Professor with Guangzhou University, Guangzhou, China. He has authored over 40 papers in international conferences and journals. His research interests include design of secure protocols in cloud computing (secure cloud storage, encrypted keyword search, and outsourcing computation) and cryptographic protocols.

ZHENGAN HUANG received the B.S. and M.S. degrees from the Department of Mathematics, Sun Yat-sen University, in 2009 and 2011, respectively, and the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, in 2015. He served as a Security Engineer for Huawei Technologies Co., Ltd. from 2015 to 2016. He currently holds a post-doctoral position with Guangzhou University. His research interests include public-key cryptography and information security.



WENBIN CHEN received the M.S. degree in mathematics from the Institute of Software, Chinese Academy of Science, in 2003, and the Ph.D. degree in computer science from North Carolina State University, USA, in 2010. He is currently an Associate Professor with Guangzhou University. His research interests include theoretical computer science, such as lattice-based cryptography, algorithm design and analysis, and computational complexity.

JIAN SHEN received the M.E. and Ph.D. degrees in computer science from Chosun University, South Korea, in 2009 and 2012, respectively. Since 2012, he has been a Professor with the Nanjing University of Information Science and Technology, Nanjing, China. His research interests include public key cryptography, secure data sharing, and data auditing in cloud.