Theoretical Computer Science 634 (2016) 47–54

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Lattice-based linearly homomorphic signatures in the standard model

Wenbin Chen [a,*], Hao Lei [b], Ke Qi [a]

[a] Department of Computer Science, Guangzhou University, PR China
[b] Huawei Technologies Co., Ltd., PR China

ARTICLE INFO

Article history:
Received 6 April 2015
Received in revised form 9 March 2016
Accepted 5 April 2016
Available online 11 April 2016
Communicated by E. Kushilevitz

Keywords:
Lattice
Linearly homomorphic signature

ABSTRACT

At present, there are some linearly homomorphic signatures in the standard model, whose security is based on the RSA assumption and Diffie–Hellman assumption. But there are still no lattice-based linearly homomorphic signature in the standard model. How to construct it? It is proposed as a open problem by Freeman [12]. In this paper, we propose the first lattice-based linearly homomorphic signature in the standard model, which settle this open problem.

1. Introduction

Given n-dimensional vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in a finite field $\mathbb{F}_p$, a linearly homomorphic signature algorithm signs each vector and produces one signature for every vector. The linear homomorphic property means that anyone can produce a signature for any vector $\mathbf{v} \in \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$, where $\text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} = \{c_1\mathbf{v}_1 + \ldots + c_k\mathbf{v}_k | c_1, \ldots, c_k \in \mathbb{F}_p\}$. The formal definition of linearly homomorphic signature will be given in Section 2.

In 2002, Johnson et al. propose the first linearly homomorphic signature scheme [15]. Boneh et al. apply the linearly homomorphic signature scheme to the network coding protocols for authenticating packets [7], whose construction is based on bilinear groups and it authenticates linear functions on vectors over large prime fields. Based on RSA assumption, Gennaro et al. propose a linearly homomorphic signature which authenticates linear functions on vectors over the integers [13]. Based on lattice assumptions, Boneh and Freeman use the k-SIS tool to construct a linear homomorphic signature scheme over binary fields [5]. Furthermore, based on ideal lattices, Boneh and Freeman propose a polynomial homomorphic signature scheme which authenticates polynomial functions on data [6].

All previous homomorphic signatures are proven secure in the random oracle model. Those linearly homomorphic signatures proven secure in the standard model are the scheme of Attrapadung et al. [4] and the scheme of Catalano et al. [8]. The linearly homomorphic signature scheme of Attrapadung et al. [4] uses bilinear groups of composite order and is based on the identity-based encryption scheme of Lewko and Waters [16]. Catalano et al.'s signature scheme is based on the adaptive pseudo-free groups [9]. These two signature schemes are linearly homomorphic over the integers or over $\mathbb{F}_p$ for some large p. On the other hand, their security is based on that the RSA assumption holds or the discrete logarithm problem is hard.

* Corresponding author.
E-mail address: cwb2011@gzhu.edu.cn (W. Chen).

http://dx.doi.org/10.1016/j.tcs.2016.04.009
0304-3975/

At present, there are not any linearly homomorphic signatures schemes in the standard model which are based on lattice. There are also not signatures schemes in the standard model which are linearly homomorphic over a small field such as $\mathbb{F}_2$.

Our contributions.

In this paper, we settle above open problems. We propose the first secure linearly homomorphic signature scheme in the stand model which is based on lattice and authenticates vectors over small field including $\mathbb{F}_2$.

2. Definitions of linearly homomorphic signatures

Notation. Given a function $f(n)$, if it is $O(n^{-c})$ for all $c > 0$, it is called *negligible*. A negligible function of n is denoted as $negl(n)$. Given a function $f(n)$, if it is $O(n^c)$ for some $c > 0$, it is called *polynomial*. A polynomial function of n is denoted as $poly(n)$. If the probability that an event occurs is $1 - negl(n)$, we say it occurs with *overwhelming probability*.

The formal definition of homomorphic signature is given as follows.

Definition 2.1 *(Homomorphic signature adapted from [6]).* A homomorphic signature scheme $\mathcal{S}$ consists of a tuple of probabilistic, polynomial-time algorithms $(Setup, Sign, Verify, Evaluate)$ with the following functionality:

Setup$(1^n, k)$. Given a security parameter n and a maximum data set size k, this algorithm outputs a public key pk and a secret key sk. The public key pk defines a message space $\mathcal{M}$, a signature space Σ, and a set $\mathcal{F}$ of admissible functions $f : \mathcal{M}^k \to \mathcal{M}$.

Sign(sk, τ, m, i). Given a secret key sk, a tag $\tau \in \{0,1\}^n$, a message $m \in \mathcal{M}$ and an index $i \in \{1, \ldots, k\}$, this algorithm outputs a signature $\sigma \in \Sigma$.

Verify(pk, τ, m, σ, f). Given a public key pk, a tag $\tau \in \{0,1\}^n$, a message $m \in \mathcal{M}$, a signature $\sigma \in \Sigma$ and a function $f \in \mathcal{F}$, this algorithm outputs either 0 (reject) or 1 (accept).

Evaluate$(pk, \tau, f, \vec{\sigma})$. Given a public key pk, a tag $\tau \in \{0,1\}^n$, a function $f \in \mathcal{F}$, and a tuple of signatures $\vec{\sigma} \in \Sigma^k$, this algorithm outputs a signature $\sigma' \in \Sigma$.

Let $\pi_i : \mathcal{M}^k \to \mathcal{M}$ be the function $\pi_i(m_1, \ldots, m_k) = m_i$ that projects onto the i-th component. We require $\pi_1, \ldots, \pi_k \in \mathcal{F}$ that for every pk output by **Setup**$(1^n, k)$.

For correctness, we require that for each (pk, sk) output by **Setup**$(1^n, k)$, the following hold:

1. Let $\tau \in \{0,1\}^n$, $m \in \mathcal{M}$, $i \in \{1, \ldots, k\}$, if $\sigma \leftarrow$ **Sign**(sk, τ, m, i), then **Verify**$(pk, \tau, m, \sigma, \pi_i) = 1$.
2. Let $\tau \in \{0,1\}^n$, $\vec{m} = (m_1, \ldots, m_k) \in \mathcal{M}^k$, $f \in \mathcal{F}$, $i \in \{1, \ldots, k\}$, if $\sigma_i \leftarrow$ **Sign**(sk, τ, m, i), then **Verify**$(pk, \tau, f(\vec{m}), Evaluate(pk, \tau, f, (\sigma_1, \ldots, \sigma_k)), f) = 1$.

The above signature is called $\mathcal{F}$-homomorphic, or homomorphic with respect to $\mathcal{F}$.

The formal definition of linearly homomorphic signature is given as follows.

Definition 2.2. A linearly homomorphic signature scheme is a homomorphic signature scheme where the message space $\mathcal{M}$ consists of n-dimensional vectors over a ring R and the set of admissible function $\mathcal{F}$ consists of R-linear function from $(R^n)^k$ to R. We represent a linear function $f(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \Sigma_{i=1}^{i=k} c_i \mathbf{v}_i$ as the vector $(c_1, \ldots, c_k) \in R^k$

The unforgeability of linearly homomorphic signature is given as follows.

Definition 2.3 *(Unforgeability adapted from [6]).* For a linearly homomorphic signature scheme $\mathcal{LS} = (Setup, Sign, Verify, Evaluate)$, we consider the following game:

Setup: The challenger runs $Setup(1^n, k)$ to obtain (pk, sk) and gives pk to $\mathcal{A}$. The public key defines a message space $\mathcal{M}$, a signature space Σ, and a set $\mathcal{LF}$ of linearly functions $f : \mathcal{M}^k \to \mathcal{M}$.

Queries: Proceeding adaptively, $\mathcal{A}$ specifies a sequence of data sets $\vec{m}_i \in \mathcal{M}^k$. For each i, the challenger chooses τ_i uniformly from $\{0,1\}^n$ and gives to $\mathcal{A}$ the tag τ_i and the signatures $\sigma_{ij} \leftarrow Sign(sk, \tau_i, m_{ij}, j)$ for $j = 1, \ldots, k$.

Output: $\mathcal{A}$ outputs a tag $\tau^* \in \{0,1\}^n$, a message $m^* \in \mathcal{M}$, a function $f^* \in \mathcal{F}$, and a signature $\sigma^* \in \Sigma$. The adversary wins if $Verify(pk, \tau^*, m^*, \sigma^*, f^*) = 1$, and either

(1) $\tau^* \neq \tau_i$ for all i (a type 1 forgery), or
(2) $\tau^* = \tau_i$ for some i but $m^* \neq f^*(\vec{m}_i)$ (a type 2 forgery).

The advantage of $\mathcal{A}$ is defined to be the probability that $\mathcal{A}$ wins the security game.

$\mathcal{LS}$ is called $(t(n), \epsilon(n))$-unforgeable if there is no $t(n)$-time adversary $\mathcal{A}$ with advantage at least $\epsilon(n)$ in the game.