Cybersecurity

RESEARCH Open Access

Inner product encryption from ring learning with errors

Shisen Fang*†, Shaojun Yang† and Yuexin Zhang

Abstract

The functional encryption scheme designed using the lattice can realize fine-grained encryption and it can resist quantum attacks. Unfortunately, the sizes of the keys and ciphertexts in cryptographic applications based on learning with errors are large, which makes the algorithm inefficient. Therefore, we construct a functional encryption for inner product predicates scheme by improving the learning with errors scheme of Agrawal et al. [Asiacrypt 2011], and its security relies on the difficulty assumption of ring learning with errors. Our construction can reduce the sizes of the keys and ciphertexts compared with the learning with errors scheme.

Keywords: Functional encryption, Inner product encryption, Lattices, Ring learning with errors

Introduction

Traditional public key encryption is “all or nothing” in accessing data, that is, a user can decrypt successfully or know nothing about the plaintexts. While the presentation of functional encryption (FE) (Boneh et al. 2011; O’Neill 2010) breaks through the restriction which is limited to only one user and has a single decryption result, and it can realize fine-grained encryption. As an extension of the traditional public key, the FE is the advanced cryptographic paradigm.

Two typical examples of FE are attribute-based encryption (ABE) (Goyal et al. 2006; Wang et al. 2019; Yun et al. 2018; Zhang and Wu 2017; Zhang et al. 2019) and predicate encryption (PE) (Attrapadung and Imai 2009; Agrawal et al. 2016; Boneh and Waters 2006; Blundo et al. 2010; Katz et al. 2008). In the (key-policy) ABE system, the secret key s is related to a predicate g and each ciphertext is related to an attribute I. A user who holds the secret key s is able to decrypt successfully if and only if $g(I) = 1$. So does for the PE system. However, there is an obvious difference between these two encryption systems. Namely, the attribute related with each ciphertext is revealed in the ABE system, while the attribute is hidden in the PE system.

ABE is an application of fuzzy identity-based encryption (Sahai and Waters 2005). In the ABE system (Agrawal et al. 2012; Ducas et al. 2014; Libert and Ţiţiu 2019; Yun et al. 2018; Zhang and Wu 2017; Zhang et al. 2019), data is encrypted on the basis of individual identity associated with a series of attributes. Hence, ABE is applicable in cloud storage to provide authorized data privacy. However, there are some issues to solve before applying ABE in practice. For example, when user’s attributes are altered, it is required for ABE supporting attribute revocation to change user’s access privilege timely and effectively. And in 2018, Liu et al. proposed a practical ABE scheme which can solve the aforementioned issue (Liu et al. 2018). ABE also has many other practical applications, such as network privacy (Baden et al. 2009), health record access-control (Camenisch et al. 2012), verifiable computation (Parno et al. 2011), forward-secure messaging (Green and Miers 2015) and so on.

In the PE system, the computation of inner product over $\mathbb{Z}_N$ about predicate was proposed by Katz et al. (2008) where N is a composite number. They also provide a construction about inner product predicate, called inner product encryption (IPE). Due to flexibleness and usefulness of IPE, a number of researchers have proposed

*Correspondence: shisenfang@outlook.com
†Shisen Fang and Shaojun Yang contributed equally to this work.
The Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, 350108, Fuzhou PR, China

schemes about IPE (Agrawal et al. 2011; Abdalla et al. 2020; Abdalla et al. 2015; Chen et al. 2018; Okamoto and Takashima 2015; Kurosawa and Phong 2017; Li et al. 2018; Tseng et al. 2020; Wang et al. 2019; Xagawa 2013).

For example, Chen et al. proposed two IPE schemes achieving both adaptive security and full attribute-hiding in the prime-order bilinear group (Chen et al. 2018). In 2018, Kwangsu et al. first proposed a two-input IPE scheme in composite-order bilinear groups (LEE 2018). And in 2019, Tomida et al. first constructed a multi-user and multi-challenge IPE scheme, which is constructible on a pairing-free group and secure under the matrix decisional Diffie-Hellman (MDDH) assumption (Tomida 2020). While in a pairing-based IPE system, the algorithm tends to be inefficient over computation since a lot of pairings (linear to vector lengths) are used during decryption. Therefore, in 2019, an IPE scheme proposed by Wei et al. with adaptive security based on the dual system encryption method requires only six bilinear pairs to decrypt (Wei and Gao 2019). In 2020, an IPE scheme proposed by Tseng et al. needs only one pairing computation to decrypt, which is the most efficient one in terms of the private key length and the number of pairings computation for decryption (Tseng et al. 2020).

As is known to all, compared with the conventional cryptography (designed based on certain hard problems), the lattice-based cryptography resists against the quantum attacks. What's more, a great number of lattice-based cryptographic schemes are based directly on two average-case problems, that is the small integer solution (SIS) problem and LWE problem. These two problems have been confirmed to support worst-case hardness guarantees in security.

In 2011, Agrawal et al. proposed the first lattice-based IPE scheme (Agrawal et al. 2011). To optimize the sizes of the public parameters and the ciphertexts, Xagawa et al. proposed an improved lattice-based IPE scheme (Xagawa 2013), Li et al. proposed an IPE scheme reducing the size by a factor of $\log \kappa$ compared with the work of reference (Xagawa 2013), where κ is a security parameter (Li et al. 2018), and Wang et al. proposed the first compact IPE scheme from learning with errors (LWE) in 2018 (Wang et al. 2018). Those schemes are constructed on the basis of the first lattice-based IPE scheme (Agrawal et al. 2011). In addition, Abdalla et al. constructed a multi-input FE scheme combining the access control functionality of ABE with the possibility of performing linear operations on the encrypted data and built identity-based functional encryption for inner products from lattices (Abdalla et al. 2020).

However, nearly all of IPE schemes based upon these two problems will suffer from either large key size or small message space. Although some researchers may improve the sizes of keys and ciphertexts of IPE schemes based on LWE problem to certain extent, they are still too large to be practical.

To acquire more efficiency in computation and confidence in security, we will provide a construction by adapting the scheme based on LWE (Agrawal et al. 2011) to ring-LWE (R-LWE). The R-LWE is an algebraic variant of LWE. In most practical applications, the n samples from the LWE distribution can be replaced by a sample from the R-LWE distribution, which will reduce the size of the public key by a factor of n. As is mentioned above, our construction is of theoretical value and practical significance.

Our construction

Our approach. We construct a functional encryption scheme for inner product predicates based on the R-LWE problem building on the ideas and techniques of the scheme in the reference (Agrawal et al. 2011). In our construction, we generate the secret key associated with the predicate g using of ring-SIS (R-SIS) and the ciphertext c associated with the attribute I using of R-LWE. The user then can decrypt successfully using the secret key when $g(I) = 1$.

It is necessary to simulate an experiment during the process of security proof, which allows the simulator to answer secret key queries whenever $g(I) = 0$. Similarly, just as the thought of proof in the reference (Agrawal et al. 2011), we make use of $m + 1$ R-LWE instances to generate a ciphertext that either decrypts correctly or decrypts to a random element in the message space $\mathcal{M}$ in this simulation. Therefore, we only need to use a weaker security model ("weak attribute hiding") in the security proof.

Our contribution. In this paper, we present an IPE scheme that is secure under the R-LWE hardness assumption. The scheme is at its core based on the LWE scheme of (Agrawal et al. 2011). Our scheme satisfies the slightly weaker notion considered by Okamoto and Takashima (2009) and Lewko et al. (2010).

Outline. The rest of the paper is organized as follows. In "Predicate encryption", we review some theoretical knowledge about predicate encryption. In "Preliminaries", we set some notations and provide some preliminaries about lattice theory and much more. In "A functional encryption scheme for inner product predicates", we describe concretely an IPE scheme and prove the correctness and security of the scheme. In "Conclusion" sections, we present some concluding remarks.

Predicate encryption

Let κ be the security parameter for the rest of this paper and let $n = n(\kappa)$ be a power of two. We first recall the following definition of predicate encryption proposed by Katz et al. (2008), which is based on the definition of searchable encryption proposed by Boneh and Waters (2006).

Definition 1 *((Katz et al. 2008), Definition 2.1). A (key-policy) predicate encryption scheme for the class of predicates $\mathcal{G}$ over the set of attributes Σ consists of four probabilistic polynomial-time (PPT) algorithms Setup, KeyGen, Enc, Dec such that:*

- *Setup: takes as input a security parameter κ and outputs a set of public parameters PP and a master secret key MK.*
- *KeyGen: takes as input the master secret key MK and a (description of a) predicate $g \in \mathcal{G}$. It outputs a key sk_g.*
- *Enc: takes as input the public parameters PP, an attribute $I \in \Sigma$, and a message $\boldsymbol{m}$ in some associated message space $\mathcal{M}$. It returns a ciphertext C.*
- *Dec: takes as input a secret key sk_g and a ciphertext C. It outputs either a message $\boldsymbol{m}$ or the distinguished symbol $\perp$.*

For correctness, we require that for all κ, (PP, MK) are generated by Setup (1^κ), for all $g \in \mathcal{G}$, any key sk_g is generated by KeyGen(sk, g) and for all $I \in \Sigma$, any ciphertext C is generated by Enc$(\text{PP}, I, \boldsymbol{m})$:

- If $g(I) = 1$, then Dec $\left(sk_g, C\right) = \boldsymbol{m}$.
- If $g(I) = 0$, then Dec $\left(sk_g, C\right) =\perp$ with all but negligible probability.

In this paper, the correctness proof satisfies a different correctness condition which is just as the correctness idea of the LWE scheme (Agrawal et al. 2011): when $C \leftarrow$ Enc$(\text{PP}, I, \boldsymbol{m})$ with probability 1, then $\boldsymbol{m} \leftarrow$ Dec(sk_g, C) if $g(I) = 1$, however, if $g(I) = 0$ then Dec $\left(sk_g, C\right)$ is computationally indistinguishable from a uniformly random element in the message space $\mathcal{M}$.

Next, we introduce several notations of security about the PE scheme. The basic concept of security is called payload hiding. It will guarantee that the ciphertext about the attribute I can hide all information associated with the message, unless one holds a secret key giving the explicit capability to decrypt. Namely, the adversary $\mathcal{A}$ holding the keys $sk_{g_1}, \cdots, sk_{g_l}$ cannot get any information about the message encrypted by any attribute I when satisfying $g_1(I) = \cdots = g_l(I) = 0$. A stronger notation of security is called attribute hiding. It requires that the ciphertext can hide all information associated with attribute I except the part which is leaked explicitly by one who holds the key. Namely, $\mathcal{A}$ who possesses the keys only can obtain the values of $g_1(I), \cdots, g_l(I)$. The last is an intermediate notion, weak attribute hiding, in which attribute hiding is guaranteed to hold only if $\mathcal{A}$ holds the keys that cannot recover the message. And our scheme satisfies the weak attribute hiding.

Definition 2 *((Katz et al. 2008), Definition 2.1). A predicate encryption scheme with respect to $\mathcal{G}$ and Σ is attribute hiding if for any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in the security parameter κ:*

1. *$\mathcal{A}\left(1^\kappa\right)$ outputs $I_0, I_1 \in \Sigma$.*
2. *Setup(1^κ) is run to generate PP and MK, and the adversary is given PP.*
3. *$\mathcal{A}$ may adaptively request keys for any predicates $g_1, \cdots, g_l \in \mathcal{G}$ subject to the restriction that $g_i(I_0) = g_i(I_1)$ for all i. In response, $\mathcal{A}$ is given the corresponding keys $sk_{g_i} \leftarrow$ KeyGen $\left(MK, g_i\right)$.*
4. *$\mathcal{A}$ outputs two equal-length messages $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$. If there is an i for which $g_i(I_0) = g_i\left(I_1\right) = 1$, then it is required that $\boldsymbol{m}_0 = \boldsymbol{m}_1$. A random bit b is chosen, and $\mathcal{A}$ is given the ciphertext $C \leftarrow$ Enc$(PP, I_b, \boldsymbol{m}_b)$.*
5. *$\mathcal{A}$ may continue to request keys for additional predicates, subject to the same restrictions as before.*
6. *$\mathcal{A}$ outputs a bit b', and succeeds if $b' = b$. The advantage of $\mathcal{A}$ is the absolute value of the difference between its success probability and $1/2$.*

By the above definition, we observe that there exists two relations among the three notations of security. One is that any scheme which is weak attribute hiding is payload hiding, the other is that any scheme which is attribute hiding is weak attribute hiding.

Preliminaries

Notation

If no special note, we use lowercase letters (e.g. a) to express polynomials, bold lowercase letters (e.g. $\boldsymbol{a}$) to express vectors, bold capital letters (e.g. $\mathbf{A}$) to express matrices, the arrows $\left(\text{e.g. } \vec{v}\right)$ to represent predicates or attributes. If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{A}'$ is an $m' \times n$ matrix, then $\left[\mathbf{A} \| \mathbf{A}'\right]$ represents an $\left(m + m'\right) \times n$ matrix formed by concatenating $\mathbf{A}$ and $\mathbf{A}'$. If $\boldsymbol{a}$ is a length m vector and $\boldsymbol{a}'$ is a length m' vector, then we denote $\left[\boldsymbol{a} | \boldsymbol{a}'\right]$ as a length $\left(m + m'\right)$ vector which is concatenated by $\boldsymbol{a}$ and $\boldsymbol{a}'$. Suppose to denote $\mathbf{S}$ as a basis of lattice Λ, then $\tilde{\mathbf{S}}$ denotes the Gram-Schmidt orthogonalization of $\mathbf{S}$.

For $n = n(\kappa) \in \mathbb{Z}^+$, we let $R_q = \mathbb{Z}_q[x]/f(x)$ be the integer polynomial ring modulo both $f(x)$ and q, where q is a prime and $f \in \mathbb{Z}[x]$ is a monic degree n polynomial. In particular, considering the security of our construction, we fix $f(x) = x^n + 1$ in the rest of paper. For $a \in R_q$, we denote $\|a\|$ as the Euclidean norm of a vector $a = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$ for $a_i \in \mathbb{Z}_q$. We define $rot_f(a) \in R_q^{n \times n}$ as the matrix whose i-th row is given by the coefficients of the polynomial $x^{i-1}a \mod f(x)$, for any $1 \le i \le n$. Note that for $a, b \in R_q$, $a \cdot b = \left(1, x, \cdots, x^{n-1}\right) \cdot rot_f(a)^T \cdot (b_0, b_1, \cdots, b_{n-1})^T = (a_0, a_1, \cdots, a_{n-1}) \cdot rot_f(b) \cdot \left(1, x, \cdots, x^{n-1}\right)^T$. The specific form of rot_f is given below:

$$rot_f(\boldsymbol{a}) = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ -a_{n-1} & a_0 & \cdots & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ -a_1 & -a_2 & \cdots & a_0 \end{bmatrix}.$$

Let $\mathbf{A} = rot_f(a)$, then the set $\Lambda^{\perp}(\mathbf{A}) = \left\{b \in \mathbb{Z}^n | b \cdot \mathbf{A} = 0 \mod q\right\}$ is an n-dimensional lattice. We extend that notation to the vector $\boldsymbol{a} \in R_q^m$ by applying rot_f component-wise. Namely, for $\boldsymbol{a} = (a_1, a_2, \cdots, a_m)$, $rot_f(\boldsymbol{a}) = \left[rot_f(a_1) \| rot_f(a_2) \| \cdots \| rot_f(a_m)\right]$.

We define the norm of a matrix $\mathbf{R} \in \{-1, 1\}^{m \times m}$ to be $\sup\{\|\mathbf{R}\boldsymbol{x}\| : \|\boldsymbol{x}\| = 1\}$. Then we recall the following result.

Lemma 1 *((Agrawal et al. 2011), Lemma A.1). Let $\mathbf{R}$ be an $m \times m$ matrix chosen at random from $\{-1, 1\}^{m \times m}$. Then $Pr\left\{\|\mathbf{R}\| > 12\sqrt{2m}\right\} < e^{-2m}$.*

Lattice

Now we remind some definitions and properties of lattice that we need to use in our system.

The m-dimension lattice Λ is generated by the set $\left\{\sum_{i=1}^{n} x_i \boldsymbol{b}_i \mid x_i \in \mathbb{Z}\right\}$ for n linearly independent vectors $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n \in \mathbb{R}^m$. That is to say, the lattice Λ is a full-rank discrete additive subgroup of $\mathbb{R}^m$. For $\boldsymbol{a} \in R_q^m, u \in R_q$, we define the ring setting as follows:

$$\Lambda_q(\boldsymbol{a}) := \left\{\boldsymbol{e} \in R_q^m : \exists s \in R_q, s.t.\ \boldsymbol{a}^T s = \boldsymbol{e}^T \mod q\right\},$$

$$\Lambda_q^{\perp}(\boldsymbol{a}) := \left\{\boldsymbol{e} \in R_q^m : \boldsymbol{a}\boldsymbol{e}^T = 0 \mod q\right\},$$

$$\Lambda_q^{u}(\boldsymbol{a}) := \left\{\boldsymbol{e} \in R_q^m : \boldsymbol{a}\boldsymbol{e}^T = u \mod q\right\}.$$

Next, we introduce the R-SIS (Lyubashevsky and Micciancio 2006; Peikert and Rosen 2006) and R-LWE (Lyubashevsky et al. 2010; Stehlé et al. 2009) as the ring-based variant of SIS and LWE respectively. They have been proven to be at least as hard as the shortest independent vectors problem (SIVP) and the decision version of the shortest vector problem (GapSVP). And there exists a reduction from the search version of R-LWE to the average-case decision R-LWE. If the probability that for all the polynomial-time adversaries $\mathcal{A}$ who solve the decision R-LWE is negligibly away from $\frac{1}{2}$, then we call that the decision R-LWE problem is infeasible.

Definition 3 *(Lyubashevsky and Micciancio 2006; Peikert and Rosen 2006, R-SIS$_{q,m,\beta}$) Given $\boldsymbol{a} = (a_1, \cdots, a_m) \in R_q^m$ a vector of m uniformly random polynomials, find a non-zero vector of small polynomial $\mathbf{e} = (e_1, \cdots, e_m) \in R_q^m$ such that $\boldsymbol{a}\boldsymbol{e}^T = \sum_{i=1}^{m} a_i e_i = 0 \mod q$, and $0 \le \|\boldsymbol{e}\| \le \beta$.*

Definition 4 *(Lyubashevsky et al. 2010; Stehlé et al. 2009, R-LWE Distribution) For $s \in R_q$ (the "secret") and an error distribution χ over R_q, a sample from the R-LWE distribution $A_{s,\chi}$ over $R_q^m \times R_q^m$ is generated by choosing $\boldsymbol{a} \leftarrow R_q^m$ uniformly at random, choosing $\boldsymbol{\eta} \leftarrow \chi^m$, and outputting $(\boldsymbol{a}, s \cdot \boldsymbol{a} + \boldsymbol{\eta})$.*

Definition 5 *(Lyubashevsky et al. 2010; Stehlé et al. 2009, R-LWE Search). For $s \in R_q$ and an error distribution χ over R_q. The search of version of the R-LWE is defined as follows: given access to arbitrarily many independent samples from $A_{s,\chi}$ for some arbitrary $s \in R_q$ and $\boldsymbol{\eta} \in \chi^m$, find s.*

Gaussian distribution. We denote $\rho_\sigma(a)$ as the standard n-dimensional Gaussian distribution with center 0 and the variance $\sigma > 0$, that is $\rho_\sigma(a) = \exp\left(-\pi \|a\|^2/\sigma^2\right)$. For any $\sigma \in \mathbb{R}^+$ and a lattice Λ as the subset of $\mathbb{Z}^n$, we define the lattice Gaussian distribution as $D_{\Lambda,\sigma}(a) = \frac{\rho_\sigma(a)}{\rho_\sigma(\Lambda)}$ where $\rho_\sigma(\Lambda) = \sum_{a' \in \Lambda} \rho_\sigma\left(a'\right)$. What's more, we denote the error distribution Ψ as the discrete Gaussian distribution $D_{\mathbb{Z}^n,\sigma}$ for some $\sigma > 0$. A sample from Ψ is a polynomial in R_q. We will use the following property referring to the Gaussian distribution in our construction.

Lemma 2 *((Micciancio and Regev 2004), Theorem 4.4) Let $n \in \mathbb{N}$. For any real number $\sigma = \omega\left(\sqrt{\log n}\right)$, we have $Pr_{\boldsymbol{a} \leftarrow D_{\mathbb{Z}^n,\sigma}}\left[\|\boldsymbol{a}\| > \sigma\sqrt{n}\right] \le 2^{-n+1}$.*

Sample algorithm

Now we introduce the following properties about sample algorithms. The **TrapGen** algorithm (Lai et al. 2015) is to generate the trapdoor for the R-LWE scheme. The algorithm **SampleLeft** (Agrawal et al. 2010; Cash et al. 2010) is used in our system, while the algorithm **SampleRight** (Agrawal et al. 2010) is used in the simulation during the proof of security.

We first recall the definition of the trapdoor in the ring setting.

Definition 6 *((Lai et al. 2015), Definition 2) Let $\boldsymbol{a} \in R_q^m, \boldsymbol{g} \in R_q^k$. A $\boldsymbol{g}$-trapdoor for $\boldsymbol{a}$ is a collection of linearly independent vectors of ring elements $\boldsymbol{T_a} \in R_q^{(m-k)\times k}$ such that $\boldsymbol{a}\begin{bmatrix} \boldsymbol{T_a} \\ \boldsymbol{I}_k \end{bmatrix} = h\boldsymbol{g}$ for some non-zero ring element $h \in R_q$. h is referred as the tag or label of the trapdoor. The* *quality of the trapdoor is measured by its largest singular value $s_1(\boldsymbol{T_a})$, which is computed as the largest singular value of the matrix obtained by interpreting $\boldsymbol{T_a}$ as a matrix in $\mathbb{Z}_q^{(m-k)n\times kn}$.*

Theorem 1 *((Lai et al. 2015)) Let q, m, n, k be positive integers with $q \geq 2$ and $m > k$. There exists a PPT algorithm **TrapGen** outputs a pair $\left(\boldsymbol{a} \in R_q^m, \boldsymbol{T_a} \in R_q^{(m-k)\times k}\right)$ such that $\boldsymbol{a}$ is statistically indistinguishable with the uniform distribution in R_q^m and the quality of the trapdoor $\boldsymbol{T_a}$ is measured by its largest singular value $s_1(\boldsymbol{T_a})$.*

By applying the definition and properties of rot_f to interpret a polynomial vector into a type of integer matrix, there are two efficient trapdoor delegation algorithms given as follows referring to the literature (Agrawal et al. 2010).

Algorithm 1 SampleLeft$(\boldsymbol{a}, \boldsymbol{b}, \mathbf{T}_{\boldsymbol{a}}, u, \sigma)$ (Agrawal et al. 2010)

Input: a vector $\boldsymbol{a} \in R_q^m$ with the trapdoor $\mathbf{T}_{\boldsymbol{a}}$, a vector $\boldsymbol{b} \in R_q^m$,
a polynomial $u \in R_q$ and a Gaussian parameter σ.
Output: a vector $\boldsymbol{e} \in R_q^{2m}$ satisfying $\boldsymbol{a}'\boldsymbol{e}^T = u$, where $\boldsymbol{a}' = [\boldsymbol{a}|\boldsymbol{b}]$.

Lemma 3 *((Agrawal et al. 2010), Theorem 3) Let $q > 2, m > 2\log q$ and $\sigma > \|\tilde{\boldsymbol{T}}_{\boldsymbol{a}}\|\omega\left(\sqrt{\log(2nm)}\right)$, then the algorithm **SampleLeft($\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{T_a}, u, \sigma$)** outputs a vector $\boldsymbol{e} \in R_q^{2m}$ distributed statistically close to $D_{\Lambda_q^u(\boldsymbol{a}'),\sigma}$ where $\boldsymbol{a}' = [\boldsymbol{a}|\boldsymbol{b}]$.*

Algorithm 2 SampleRight$(\boldsymbol{a}, \boldsymbol{b}, \mathbf{R}, \mathbf{T}_{\boldsymbol{b}}, u, \sigma)$ (Agrawal et al. 2010)

Input: a vector $\boldsymbol{a} \in R_q^m$, a polynomial $u \in R_q$, a vector $\boldsymbol{b} \in R_q^m$
with the trapdoor $\mathbf{T}_{\boldsymbol{b}}$, a matrix $\mathbf{R} \in \{-1, 1\}^{m\times m}$ and a Gaussian parameter σ.
Output: a vector $\boldsymbol{e} \in R_q^{2m}$ satisfying $\boldsymbol{a}'\boldsymbol{e}^T = u$ where $\boldsymbol{a}' = [\boldsymbol{a}|\boldsymbol{a}\mathbf{R} + \boldsymbol{b}]$.

Lemma 4 *((Agrawal et al. 2010), Theorem 4) Let $q > 2, m > 1$ and $\sigma > \|\tilde{\boldsymbol{T}}_{\boldsymbol{b}}\| \cdot \sqrt{nm} \cdot \omega(\log nm)$, then the algorithm **SampleRight($\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{R}, \boldsymbol{T_b}, u, \sigma$)** outputs a vector $\boldsymbol{e} \in R_q^{2m}$ distributed statistically close to $D_{\Lambda_q^u(\boldsymbol{a}'),\sigma}$ where $\boldsymbol{a}' = [\boldsymbol{a}|\boldsymbol{aR} + \boldsymbol{b}]$.*

Universal hash function

For a hash function h, define $\delta_h(x, y) = 1$ if $h(x) = h(y)$ and $\delta_h(x, y) = 0$ otherwise for $x, y \in X, x \neq y$. That is, $\delta_h(x, y) = 1$ if and only if the hashed values of x and y collide. For a finite set $\mathcal{H}$ of hash functions, define $\delta_{\mathcal{H}}(x, y) = \sum_{h\in\mathcal{H}} \delta_h(x, y)$. Hence, $\delta_{\mathcal{H}}(x, y)$ counts the number of hash functions in $\mathcal{H}$ under which x and y collide.

Definition 7 *((Roşca et al. 2017)) A (finite) family $\mathcal{H}$ of hash functions $h : X \to Y$ is universal if $Pr_{h\leftarrow U(\mathcal{H})}\left[\delta_h(x, y) = 1\right] = 1/|Y|$, for all $x, y \in X, x \neq y$.*

We will use the following variant of the leftover hash lemma which is necessary when presenting our construction.

Lemma 5 *((Roşca et al. 2017), Lemma 2.1) Let X, Y, Z denote finite sets and let $\mathcal{H}$ be a universal family of hash functions $h : X \to Y$. Let $f : X \to Z$ be arbitrary. Then for any random variable T taking values in X, we have: $\Delta((h, h(T), f(T)), (h, U(Y), f(T))) \leq \frac{1}{2}\sqrt{\gamma(T) \cdot |Y| \cdot |Z|}$, where $\gamma(T) = \max_{T'\in X} Pr\left[T = T'\right]$.*

Lemma 6 *Let q be a prime. For $\boldsymbol{R} \in \{-1, 1\}^{m\times m}$ and $\boldsymbol{a} \in R_q$, define $\Phi_{\boldsymbol{a}} : \{-1, 1\}^{m\times m} \to R_q^m$ by the rule: $\Phi_{\boldsymbol{a}}(\boldsymbol{R}) = \boldsymbol{aR}$. Then $\{\Phi_{\boldsymbol{a}}\}$ is universal.*

Proof We set $\boldsymbol{a} = (a_1, \cdots, a_m)$ and $\mathbf{R} = (r_{ij})$ where $a_i \in R_q$ and $r_{ij} \in \{-1, 1\}$ for $i, j \in \{1, \cdots m\}$. Then

$$\Phi_{\boldsymbol{a}}(\mathbf{R}) = (a_1, \cdots, a_m)\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mm} \end{pmatrix}$$

$$= \left(\sum_{i=1}^{m} a_i r_{i1}, \cdots, \sum_{i=1}^{m} a_i r_{im}\right).$$

Obviously, we need to prove $\Pr\left\{\left(\sum_{i=1}^{m} a_i r_{i1}, \cdots, \sum_{i=1}^{m} a_i r_{im}\right) = (y_1, \cdots, y_m)\right\} = \frac{1}{q^{nm}}$ for all $(y_1, \cdots, y_m) \in R_q^m$. Without loss of generality, we assume that $\sum_{i=1}^{m} a_i r_{i1} \neq 0$. Then by linearity, it suffices to prove that for all $y_1 \in R_q$, $\Pr\left\{\sum_{i=1}^{m} a_i r_{i1} = y_1\right\} = \frac{1}{q^n}$.

We write a_i as $a_{i0} + a_{i1}x + \cdots + a_{i,n-1}x^{n-1}$ and y_1 as $y_{10} + y_{11}x + \cdots + y_{1,n-1}x^{n-1}$ for $a_{ij}, y_{1j} \in \mathbb{Z}_q$. Then we calculate the following formula,

$$\begin{aligned}\sum_{i=1}^{m} a_i r_{i1} =& a_1 r_{11} + a_2 r_{21} + \cdots + a_m r_{m1} \\ =& r_{11} \sum_{j=0}^{n-1} a_{1j} x^j + \cdots + r_{m1} \sum_{j=0}^{n-1} a_{mj} x^j \\ =& \sum_{i=1}^{m} r_{i1} a_{i0} + \cdots + \sum_{i=1}^{m} r_{i1} a_{i,n-1} x^{n-1}.\end{aligned}$$

Since $y_{1j} \in \mathbb{Z}_q$, it follows that $\Pr\left\{\sum_{i=1}^{m} r_{i1} a_{ij} = y_{1j}\right\} = \frac{1}{q}$, which is equivalent to $\Pr\left\{\sum_{i=1}^{m} a_i r_{i1} = y_1\right\} = \frac{1}{q^n}$. Hence the hash function family is universal. □

A functional encryption scheme for inner product predicates

In this section, we first describe a new predicate encryption scheme and prove its correctness and security. We define our construction consisting of four PPT algorithms: setup, key generation, encryption and decryption algorithms. In this scheme, each secret key is associated with a predicate vector $\vec{v} \in \mathbb{Z}_q^l$ (for some fixed $l \geq 2$) and each ciphertext is associated with an attribute vector $\vec{w} \in \mathbb{Z}_q^l$. The decryption algorithm involves a condition that will decrypt successfully if and only if $\langle \vec{v}, \vec{w} \rangle = 0 \pmod q$. Therefore, we define the predicate associated with the secret key as $g_{\vec{v}}(\vec{w}) = 1$ when satisfying $\langle \vec{v}, \vec{w} \rangle = 0 \pmod q$, and $g_{\vec{v}}\left(\vec{w}\right) = 0$ otherwise.

The construction

Let $\kappa \in \mathbb{Z}^+$ and l be the length of predicate and attribute vectors. Let $m = m(\kappa, l)$, $q = q(\kappa, l)$ and $t = \lfloor \log q \rfloor$ be positive integers. Let α and σ be positive real Gaussian parameters. Let the error distribution $\chi = \lfloor D_{\alpha q} \rceil$ denote the discrete Gaussian distribution where each coefficient is sampled from $D_{\alpha q}$ and then rounded to nearest integer. The plaintext space is $\{0,1\}^n$, while the ciphertext space is $R_q^m \times \left\{R_q^m\right\}^{l(t+1)} \times R_q$.

FE.Setup$\left(1^\kappa, 1^l\right)$: Input a security parameter $\kappa \in \mathbb{Z}^+$ and a parameter l, do the following:

1. Using the algorithm **TrapGen** to obtain a vector $\boldsymbol{a} \in R_q^m$ together with the trapdoor $\mathbf{T}_{\boldsymbol{a}}$.
2. Choose $l \cdot (1+t)$ uniformly random vectors $\boldsymbol{a}_{i,\gamma} \in R_q^m$ for $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$.
3. Select a uniformly random polynomial $u \in R_q$.

Output the public parameters PP= $\left(\boldsymbol{a}, \left\{\boldsymbol{a}_{i,\gamma}\right\}_{i \in \{1,\cdots,l\}}, \gamma \in \{0,\cdots,t\}, u\right)$ and MK=$\mathbf{T}_{\boldsymbol{a}}$.

FE.KeyGen(PP, MK, $\vec{v}$): Input the public parameters PP, the master secret key MK and a predicate vector $\vec{v} \in \mathbb{Z}_q^l$, do:

1. For $i = 1, \cdots, l$, let $\hat{v}_i$ be the integer in $[0, q-1]$, which equals to $v_i \bmod q$. Let the binary decomposition of $\hat{v}_i$ as $\hat{v}_i = \sum_{\gamma=0}^{t} v_{i,\gamma} \cdot 2^\gamma$, where $v_{i,\gamma}$ are in $\{0,1\}$.
2. Define the vectors $\boldsymbol{a}'_{\vec{v}} := \sum_{i=1}^{l} \sum_{\gamma=0}^{t} v_{i,\gamma} \boldsymbol{a}_{i,\gamma}$ and $\boldsymbol{a}_{\vec{v}} := \left[\boldsymbol{a} | \boldsymbol{a}'_{\vec{v}}\right]$.
3. Using the master secret key MK=$\mathbf{T}_{\boldsymbol{a}}$ to compute $\boldsymbol{e} \leftarrow$ **SampleLeft**$\left(\boldsymbol{a}, \boldsymbol{a}'_{\vec{v}}, \mathbf{T}_{\boldsymbol{a}}, u, \sigma\right)$. Then $\boldsymbol{e}$ is a vector in R_q^{2m} satisfying $\boldsymbol{a}_{\vec{v}} \boldsymbol{e}^T = u$.

Output the secret key $sk_{\vec{v}} = \boldsymbol{e}$.

FE.Enc(PP, $\vec{w}$, $\boldsymbol{m}$): Input the public parameters PP, an attribute vector $\vec{w} \in \mathbb{Z}_q^l$ and a message $\boldsymbol{m}$, do:

1. Choose a uniformly random vector $\boldsymbol{b} \in R_q^m$.
2. Choose a uniformly polynomial $s \in R_q$.
3. Select a noise vector $\boldsymbol{\eta}$ from χ^m and a noise term η from χ.
4. Compute $\boldsymbol{c}_0 = s \cdot \boldsymbol{a} + 2\boldsymbol{\eta}$.
5. For $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$, do the following:
   (a) Pick a random matrix $\mathbf{R}_{i,\gamma} \in \{-1,1\}^{m \times m}$.
   (b) Calculate $\boldsymbol{c}_{i,\gamma} \leftarrow s \cdot \left(\boldsymbol{a}_{i,\gamma} + 2^\gamma w_i \boldsymbol{b}\right) + 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}$.
6. Compute $c' = us + \boldsymbol{m} + 2\eta$.

Output the ciphertext CT=$\left(\boldsymbol{c}_0, \left\{\boldsymbol{c}_{i,\gamma}\right\}_{i \in \{1,\cdots,l\}, \gamma \in \{0,\cdots,t\}}, c'\right)$.

FE.Dec(PP, CT, $sk_{\vec{v}}$): Input the public parameters PP, a secret key $sk_{\vec{v}}$ and a ciphertext CT, do:

1. Compute $\boldsymbol{c}_{\vec{v}} = \sum_{i=1}^{l} \sum_{\gamma=0}^{t} v_{i,\gamma} \boldsymbol{c}_{i,\gamma}$.
2. Let $\boldsymbol{c} = [\boldsymbol{c}_0 | \boldsymbol{c}_{\vec{v}}]$.

Output $\boldsymbol{m}' \leftarrow \left(c' - \boldsymbol{e} \cdot \boldsymbol{c}^T \mod f \mod q\right) \mod 2$.

Next, we need to show that our construction is correct for certain parameter choices and secure under R-LWE hardness assumption. The specific proof is as follows.

The correctness

Lemma 7 *Let the parameters q and α satisfy $q > 16\,(n + \lambda nm)$ and $\alpha < 8\left(\sqrt{n} + \lambda\sqrt{nm}\right)^{-1}$ where $\lambda = \left(1 + 12\sqrt{2ml}(t+1)\right)\sigma\sqrt{nm}$. When the **FE.KeyGen** algorithm returns the secret key, **FE.Enc** encrypts with probability 1 for all the plaintext **m**. If $\langle \vec{v}, \vec{w} \rangle = 0$, then we have **FE.Dec** $=$ **m** with overwhelming probability.*

Proof According to the decryption algorithm, we have,

$$\begin{aligned}
\boldsymbol{c}_{\vec{v}} &= \sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{c}_{i,\gamma} \\
&= \sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\left[s\cdot(\boldsymbol{a}_{i,\gamma}+2^{\gamma}w_i\boldsymbol{b})+2\boldsymbol{\eta}\cdot\mathbf{R}_{i,\gamma}\right] \\
&= \sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}s\boldsymbol{a}_{i,\gamma} + \sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}2\boldsymbol{\eta}\mathbf{R}_{i,\gamma}, \qquad (1)
\end{aligned}$$

the last equation holds because of $\langle\vec{v},\vec{w}\rangle = 0$.

By the above formula, we obtain,

$$\begin{aligned}
\boldsymbol{c} &= [\boldsymbol{c}_0|\boldsymbol{c}_{\vec{v}}] \\
&= \left[s\boldsymbol{a}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}s\boldsymbol{a}_{i,\gamma}\right] + \left[2\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}2\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right] \\
&= s\left[\boldsymbol{a}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{a}_{i,\gamma}\right] + \left[2\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}2\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right] \\
&= s\boldsymbol{a}_{\vec{v}} + \left[2\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}2\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right].
\end{aligned}$$

According to Lemma 3, we can get $\boldsymbol{a}_{\vec{v}}\boldsymbol{e}^T = u$ and $\boldsymbol{e}\cdot\boldsymbol{c}^T = us + 2\boldsymbol{e}\cdot\left[\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{\eta}\cdot\mathbf{R}_{i,\gamma}\right]^T$.

Finally, according to the third step of the decryption algorithm, we compute $\boldsymbol{m}'$ as

$$\begin{aligned}
&us + \boldsymbol{m} + 2\eta - us - 2\boldsymbol{e}\left[\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right]^T \\
&= \boldsymbol{m} + 2\left(\eta - \boldsymbol{e}\left[\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right]^T\right). \qquad (2)
\end{aligned}$$

If $\left\|\boldsymbol{m} + 2\left(\eta - \boldsymbol{e}\left[\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right]^T\right)\right\| < q/2$, centered reduction modulo q of $\boldsymbol{c}' - \boldsymbol{e}\cdot\boldsymbol{c}^T$ given us $\boldsymbol{m}+2\left(\eta - \boldsymbol{e}\left[\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{\eta}\mathbf{R}_{i,\gamma}\right]^T\right)$ (over the integers). Hence, in order to obtain $\boldsymbol{m} = \boldsymbol{m}'$, it suffices to certify $\left\|\boldsymbol{m} + 2\left(\eta - \boldsymbol{e}\left[\boldsymbol{\eta}\mid\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{\eta}\cdot\mathbf{R}_{i,\gamma}\right]^T\right)\right\| < q/2$.

We set $\boldsymbol{e} \in R_q^{2m}$ as $[\boldsymbol{e}_1|\boldsymbol{e}_2]$ for $\boldsymbol{e}_i \in R_q^m$. Then Eq. (2) can be rewritten as

$$\begin{aligned}
&\boldsymbol{m} + 2\eta - \left[\boldsymbol{e}_1\cdot 2\boldsymbol{\eta}^T + \boldsymbol{e}_2\cdot\left(\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\mathbf{R}_{i,\gamma}^T\cdot 2\boldsymbol{\eta}^T\right)\right] \\
=&\boldsymbol{m} + 2\eta - \left[\left(\boldsymbol{e}_1 + \boldsymbol{e}_2\cdot\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\mathbf{R}_{i,\gamma}^T\right)\cdot 2\boldsymbol{\eta}^T\right].
\end{aligned}$$

For $\eta \in \chi$ and $\boldsymbol{\eta} \in \chi^m$, we have $\|\eta\| < \alpha q\sqrt{n} + n$ and $\|\boldsymbol{\eta}\| < \alpha q\sqrt{nm} + nm$ with overwhelming probability because of the Gaussian tail bound. According to Lemma 1 and the triangle inequality, $\left\|\left(\boldsymbol{e}_1 + \boldsymbol{e}_2\cdot\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\mathbf{R}_{i,\gamma}^T\right)\cdot 2\boldsymbol{\eta}^T\right\|$ is not exceeding $2\lambda\left(\alpha q\sqrt{nm}+nm\right)$where$\lambda = \left(1+12\sqrt{2ml}(t+1)\right)\sigma\sqrt{nm}$. Thus we have $\left\|\boldsymbol{m} + 2\eta - \left[\left(\boldsymbol{e}_1 + \boldsymbol{e}_2\cdot\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\mathbf{R}_{i,\gamma}^T\right)\cdot 2\boldsymbol{\eta}^T\right]\right\| < \sqrt{n} + 2\left(\alpha q\sqrt{n} + n\right) + 2\lambda\left(\alpha q\sqrt{nm} + nm\right) < q/2$ with overwhelming probability when α and q satisfy the condition in the lemma.

If $\langle\vec{v},\vec{w}\rangle \neq 0$, $\sum_{i=1}^{l}\sum_{\gamma=0}^{t} 2^{\gamma}v_{i,\gamma}w_i s\cdot\mathbf{b}$ in the formula (1) is unequal to 0. Since $s \in R_q$ and $\mathbf{b} \in R_q^m$ are randomly chosen in the formula (1), the decryption algorithm cannot decrypt the message correctly. □

The security

To demonstrate the security, we introduce several security games to prove that the security of the scheme can be reduced to the hardness of R-LWE problem.

Theorem 2 *Suppose that $m \geq 3n\log q$. Then the above predicate encryption scheme is weakly attribute hiding under the R-LWE hardness assumption.*

Before introducing these security games, we define a simulation construction as following: alternative setup, key generation, and encryption algorithms.

Sim.Setup$(1^{\kappa}, 1^{l}, \vec{w}^*)$: Input a security parameter κ, a parameter l and an attribute vector $\vec{w}^* \in \mathbb{Z}_q^l$, do the following:

1. Select a uniformly random vector $\boldsymbol{a} \in R_q^m$ and polynomial $u \in R_q$.
2. Using the algorithm **TrapGen** to obtain a vector $\boldsymbol{b}^* \in R_q^m$ with a trapdoor $\mathbf{T}_{\boldsymbol{b}^*}$.
3. For $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$, choose random matrices $\mathbf{R}_{i,\gamma}^* \in \{-1,1\}^{m\times m}$ and set $\boldsymbol{a}_{i,\gamma} \leftarrow \boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^{\gamma}w_i^*\boldsymbol{b}^*$.

Output the public parameters and the master secret key PP= $\left(\boldsymbol{a}, \left\{\boldsymbol{a}_{i,\gamma}\right\}_{i\in\{1,\cdots,l\},\gamma\in\{0,\cdots,t\}}, u\right)$, MK= $\left(\vec{w}^*, \left\{\mathbf{R}^*_{i,\gamma}\right\}_{i\in\{1,\cdots,l\},\gamma\in\{0,\cdots,t\}}, \boldsymbol{b}^*, \mathbf{T}_{\boldsymbol{b}^*}\right)$.

Sim.KeyGen(PP, MK, $\vec{v}$): Input the public parameters PP, master secret key MK and a vector $\vec{v} \in \mathbb{Z}_q^l$, do:

1. If $\langle\vec{v}, \vec{w}\rangle = 0$, output $\perp$.
2. For $i = 1, \cdots, l$, let $\hat{v}_i$ be the integer in $[0, q-1]$ equals to $v_i \bmod q$. Write the binary decomposition of $\hat{v}_i$ as $\hat{v}_i = \sum_{\gamma=0}^{t} v_{i,\gamma} \cdot 2^\gamma$, where $v_{i,\gamma}$ are in $\{0,1\}$.
3. Define the vectors $\boldsymbol{a}'_{\vec{v}} := \sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\boldsymbol{a}_{i,\gamma}$ and $\boldsymbol{a}_{\vec{v}} := \left[\boldsymbol{a}|\boldsymbol{a}'_{\vec{v}}\right]$. Then it follows that

$$\boldsymbol{a}_{\vec{v}} = \left[\boldsymbol{a} \mid \boldsymbol{a}\left(\sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\mathbf{R}^*_{i,\gamma}\right) - \underbrace{\left(\sum_{i=1}^{l}\sum_{\gamma=0}^{t} 2^\gamma v_{i,\gamma} w_i^*\right)}_{\langle\vec{v},\vec{w}^*\rangle}\boldsymbol{b}^*\right].$$

4. Generate $\boldsymbol{e} \leftarrow$**SampleRight**$\left(\boldsymbol{a}, -\langle\vec{v}, \vec{w}^*\rangle\boldsymbol{b}^*, \sum_{i=1}^{l}\sum_{\gamma=0}^{t} v_{i,\gamma}\mathbf{R}^*_{i,\gamma}, \mathbf{T}_{\boldsymbol{b}^*}, u, \sigma\right)$.

Output the secret key $sk_{\vec{v}} = \boldsymbol{e}$.

Sim.Enc(PP, $\vec{w}$, $\boldsymbol{m}$, MK): The algorithm is the same as the **FE.Enc** algorithm, except:

1. In Step 1, the random vector $\boldsymbol{b}^* \in$ MK is used to replace the vector $\boldsymbol{b}$.
2. In Step 5(a), the random matrices $\mathbf{R}^*_{i,\gamma} \in$ MK are used to replace the matrices $\mathbf{R}_{i,\gamma}$ for $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$.

In order to prove Theorem 2, we consider a security game against the adversary $\mathcal{A}$ that plays the weak attribute hiding game as follows. The challenger $\mathcal{C}$ samples a bit $b \leftarrow \{0,1\}$ at the beginning of the game. $\mathcal{A}$ outputs two attribute vectors $\vec{w}_b$ for $b \in \{0,1\}$. $\mathcal{C}$ then runs the **FE.Setup** and **FE.KeyGen** algorithms to answer $\mathcal{A}$'s queries, and it also generates the ciphertext using the **FE.Enc** $(\vec{w}_b, \boldsymbol{m}_b)$ and sends it to $\mathcal{A}$. Finally $\mathcal{A}$ returns a bit b'. Our construction is secure if there is no probability polynomial time adversary $\mathcal{A}$ to output $b' = b$ with more probability that is non-negligibly away from $\frac{1}{2}$.

Next, we define a series of games which are statistically or computationally indistinguishable with the above security game against $\mathcal{A}$. What's more, according to the simulation scheme, $\mathcal{A}$ can only request keys when the predicate vector $\vec{v}$ satisfies $\langle\vec{v}, \vec{w}_b\rangle \neq 0$ for $b \in \{0,1\}$.

- Game 1: The challenger $\mathcal{C}$ runs the **FE.Setup** and **FE.KeyGen** to answer the adversary $\mathcal{A}$'s key queries. Then $\mathcal{C}$ computes the challenge ciphertext from **FE.Enc** $(\vec{w}_0, \boldsymbol{m}_0)$ and sends it to $\mathcal{A}$.
- Game 2: The challenger $\mathcal{C}$ runs the **Sim.Setup** $(\vec{w}^* = \vec{w}_0)$ and **Sim.KeyGen** to answer $\mathcal{A}$'s key queries. Then $\mathcal{C}$ computes the challenge ciphertext from **Sim.Enc** $(\vec{w}_0, \boldsymbol{m}_0)$ and sends it to $\mathcal{A}$.
- Game 3: The challenger $\mathcal{C}$ runs the **Sim.Setup** $(\vec{w}^* = \vec{w}_0)$ and **Sim.KeyGen** to answer $\mathcal{A}$'s key queries. Then $\mathcal{C}$ chooses uniformly the challenge ciphertext from the ciphertext space and sends it to $\mathcal{A}$.
- Game 4: The challenger $\mathcal{C}$ runs the **Sim.Setup** $(\vec{w}^* = \vec{w}_1)$ and **Sim.KeyGen** to answer $\mathcal{A}$'s key queries. Then $\mathcal{C}$ chooses uniformly the challenge ciphertext from the ciphertext space and sends it to $\mathcal{A}$.
- Game 5: The challenger $\mathcal{C}$ runs the **Sim.Setup** $(\vec{w}^* = \vec{w}_1)$ and **Sim.KeyGen** to answer $\mathcal{A}$'s key queries. Then $\mathcal{C}$ computes the challenge ciphertext from **Sim.Enc** $(\vec{w}_1, \boldsymbol{m}_1)$ and sends it to $\mathcal{A}$.
- Game 6: The challenger $\mathcal{C}$ runs the **FE.Setup** and **FE.KeyGen** to answer $\mathcal{A}$'s key queries. Then $\mathcal{C}$ computes the challenge ciphertext from **FE.Enc** $(\vec{w}_1, \boldsymbol{m}_1)$ and sends it to $\mathcal{A}$.

Lemma 8 *Assume that $m \geq 3n\log q$, then it follows that,*

(a) *At the view of the adversary $\mathcal{A}$, the Game 1 is statistically indistinguishable with the Game 2.*
(b) *At the view of the adversary $\mathcal{A}$, the Game 5 is statistically indistinguishable with the Game 6.*

Proof We prove (a) only because we can prove (b) with the same way.

Firstly, we demonstrate the public parameters and the ciphertext output by the **FE.Setup** and **FE.Enc** algorithms are statistically indistinguishable from those output by the **Sim.Setup** and **Sim.Enc** algorithms. That is, for every $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$, we need to argue the distributions of the set $\mathrm{E}_{i,\gamma}$ in Game 1 and Game 2 are statistically indistinguishable, where $\mathrm{E}_{i,\gamma}$ as the set $\left(\boldsymbol{a}, \left\{\boldsymbol{a}_{i,\gamma}, \boldsymbol{c}_{i,\gamma}\right\}\right)$.

In Game 1, the vector $\boldsymbol{a}$ is selected from the **TrapGen**. Then for all but a $2^{-\Omega(\kappa)}$ fraction of all $\boldsymbol{a}$ follow from uniformly distribution over R_q^m. While in Game 2, the vector $\boldsymbol{a}$ is sampled uniformly from R_q^m. Therefore, the distributions of $\boldsymbol{a}$ are statistically indistinguishable in both games.

Next, we discuss the joint distributions $\left\{\boldsymbol{a}_{i,\gamma}, \boldsymbol{c}_{i,\gamma}\right\}$ in the both games. In Game 1, the vector $\boldsymbol{a}_{i,\gamma}$ is sampled uniformly from the R_q^m and $\boldsymbol{c}_{i,\gamma}$ is equal to $s \cdot \left(\boldsymbol{a}_{i,\gamma} + 2^\gamma w_i^* \boldsymbol{b}^*\right) + 2\boldsymbol{\eta} \cdot \mathbf{R}^*_{i,\gamma}$, where $\mathbf{R}^*_{i,\gamma}$ is random independently in $\{-1,1\}^{m\times m}$ for every $i = 1, \cdots l, \gamma = 0, \cdots, t$

and $\boldsymbol{b}^*$ is uniformly selected from R_q^m. In Game 2, $\boldsymbol{a}_{i,\gamma}$ is calculated as $\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^*$, where $\mathbf{R}_{i,\gamma}^*$ is random independently in $\{-1,1\}^{m\times m}$ for every $i = 1, \cdots l, \gamma = 0, \cdots, t$, and $\mathbf{b}^*$ generated by **TrapGen** is statistically close to uniformly random in R_q^m, $\boldsymbol{c}_{i,\gamma}$ is equal to $s \cdot \left(\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^* + 2^\gamma w_i^* \boldsymbol{b}^*\right) + 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*$ where $\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^*$ is equal to the public parameter $\boldsymbol{a}_{i,\gamma}$.

Furthermore, according to Lemma 6, the function $\Phi_{\boldsymbol{a}}\left(\mathbf{R}_{i,\gamma}^*\right) = \boldsymbol{a}\mathbf{R}_{i,\gamma}^*$ is universal. Then it follows from that the statistical distance of the following two distributions is at most $\frac{1}{2}\left(\frac{1}{2^{m^2}} \cdot q^{2nm}\right)^{\frac{1}{2}} \leq \frac{1}{2}q^{-\frac{1}{2}nm}$ by Lemma 5, namely, $\left(\boldsymbol{a}, \boldsymbol{a}\mathbf{R}_{i,\gamma}^*, 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*\right) \approx_s \left(\boldsymbol{a}, \boldsymbol{a}_{i,\gamma}, 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*\right)$. Then for every fixed vector $\boldsymbol{b}^*$ and $\vec{w}^*$, it follows that $\left(\boldsymbol{a}, \boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^*, 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*\right) \approx_s \left(\boldsymbol{a}, \boldsymbol{a}_{i,\gamma}, 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*\right)$.

Since the matrix $\mathbf{R}_{i,\gamma}^*$ is chosen independently for every i, γ, the joint distributions of these quantities for all i, γ are also statistically close:

$$\left(\boldsymbol{a}, \left\{\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^*, 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*\right\}_{i,\gamma}\right) \approx_s \left(\boldsymbol{a}, \left\{\boldsymbol{a}_{i,\gamma}, 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^*\right\}_{i,\gamma}\right). \tag{3}$$

Next, we need to add two quantities which are statistically indistinguishable to the both sides of the formula (3). Then we can get the following by the conclusion that applying any function to two statistically indistinguishable ensembles produces statistically indistinguishable ensembles, that is, for every i and γ:

$$\left(\boldsymbol{a}, \left\{\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^*, \underbrace{s\left(\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^* + 2^\gamma w_i^* \boldsymbol{b}^*\right)}_{\text{add term}} + 2\boldsymbol{\eta}\mathbf{R}_{i,\gamma}^*\right\}\right)$$
$$\approx_s \left(\boldsymbol{a}, \left\{\boldsymbol{a}_{i,\gamma}, \underbrace{s\left(\boldsymbol{a}_{i,\gamma} + 2^\gamma w_i^* \boldsymbol{b}^*\right)}_{\text{add term}} + 2\boldsymbol{\eta}\mathbf{R}_{i,\gamma}^*\right\}\right).$$

By the above formula, the right side of the formula is the public parameters and the challenge ciphertext in Game 1, while the left side of the formula is the public parameters and the challenge ciphertext in Game 2. Hence, the public parameters and the challenge ciphertexts are statistically indistinguishable at the both games.

To complete the proof, we show that the secret keys output by **Sim.KeyGen** are statistically indistinguishable from those output by **FE.KeyGen** when given the public parameters and the challenge ciphertexts. In the two games, the secret key $\boldsymbol{e}$ follows from Gaussian distribution for Gaussian parameter σ, so the distributions of them are statistically indistinguishable when σ is sufficiently large. □

Lemma 9 *If the decision R-LWE problem is infeasible, then it follows that:*

(a) *At the view of the adversary $\mathcal{A}$, the Game 2 is computationally indistinguishable with the Game 3.*
(b) *At the view of the adversary $\mathcal{A}$, the Game 4 is computationally indistinguishable with the Game 5.*

Proof It suffices to prove (a). Given $m + 1$ R-LWE instances $\left(a_j, y_j\right)$ for $j = 0, \cdots, m$, in which we define either $y_j = s \cdot a_j + 2\eta_j$ for s is sampled uniformly from R_q and η_j is sampled from the discrete Gaussian χ, or $y_j \in R_q$ is uniformly random. We denote $\boldsymbol{c}_0 = (y_1, \cdots, y_m)$.

We consider a variant experiment, in which the challenger $\mathcal{C}$ runs the **Sim.Setup** $\left(\vec{w}^* = \vec{w}_0\right)$ and let $\boldsymbol{a} = (a_1, \cdots, a_m)$, $u = a_0$. Then $\mathcal{C}$ answers the queries of $\mathcal{A}$ using the **Sim.KeyGen** algorithm. Finally, for $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$, $\mathcal{C}$ computes $c' = y_0 + \boldsymbol{m}$, $\boldsymbol{c}_{i,\gamma} = \boldsymbol{c}_0\mathbf{R}_{i,\gamma}^*$ where $\mathbf{R}_{i,\gamma}^* \in \mathrm{MK}$ and sends $\left(\boldsymbol{c}_0, \left\{\boldsymbol{c}_{i,\gamma}\right\}, c'\right)$ to $\mathcal{A}$.

In Game 2, we observe that for $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$, the challenge ciphertext $\boldsymbol{c}_{i,\gamma}$ using the **Sim.Enc** as follows,

$\boldsymbol{c}_{i,\gamma} = s \cdot \left(\boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^* + 2^\gamma w_i^* \boldsymbol{b}^*\right) + 2\boldsymbol{\eta} \cdot \mathbf{R}_{i,\gamma}^* = (s \cdot \boldsymbol{a} + 2\boldsymbol{\eta})\,\mathbf{R}_{i,\gamma}^*$.

When $y_j = s \cdot a_j + 2\eta_j$, then $\boldsymbol{c}_{i,\gamma} = \boldsymbol{c}_0\mathbf{R}_{i,\gamma}^*$ in the variant experiment is identical to corresponding ciphertext in Game 2.

On the other hand, when y_j is uniformly random in R_q, then the simulated ciphertext is $\left(\boldsymbol{c}_0, \left\{\boldsymbol{c}_0\mathbf{R}_{i,\gamma}^*\right\}, c'\right)$ for $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$. By the Lemma 6, we know that the function $\Phi_{\boldsymbol{c}_0} = \boldsymbol{c}_0\mathbf{R}_{i,\gamma}^*$ is universal. Hence, by the variant of the leftover hash lemma (see Lemma 5), the statistical distance between the distribution of $\left(\boldsymbol{c}_0, \left\{\boldsymbol{c}_0\mathbf{R}_{i,\gamma}^*\right\}, c'\right)$ with the uniform distribution is bounded from $\frac{1}{2}q^{-\frac{1}{2}nm}$. While in the Game 3, the challenge ciphertext is selected uniformly from the ciphertext space. Therefore, the ciphertexts in the variant experiment and the Game 3 are statistically indistinguishable.

So we draw the conclusion that the statistical distance in the both games is negligible close under the hardness of R-LWE problem. □

Lemma 10 *The Game 3 and the Game 4 are statistically indistinguishable at the view of the adversary $\mathcal{A}$.*

Proof The only difference between the Game 3 and the Game 4 is the vector $\vec{w}^*$ which is used to calculate the public parameter $\boldsymbol{a}_{i,\gamma} = \boldsymbol{a}\mathbf{R}_{i,\gamma}^* - 2^\gamma w_i^* \boldsymbol{b}^*$, where $\boldsymbol{a}$ and $\mathbf{R}_{i,\gamma}^*$ are independent uniformly random samples. The function $\Phi_{\boldsymbol{a}} : \mathbf{R}_{i,\gamma}^* \rightarrow \boldsymbol{a}\mathbf{R}_{i,\gamma}^*$ is universal according to Lemma 6. For every $i \in \{1, \cdots, l\}$ and $\gamma \in \{0, \cdots, t\}$, $\left(\boldsymbol{a}, \boldsymbol{a}\mathbf{R}_{i,\gamma}^*\right)$ is statistically indistinguishable from $(\boldsymbol{a}, U)$ where U is uniformly random. For the value $C = 2^{\gamma} w_i^* \boldsymbol{b}^*$ associated with the fixed $\boldsymbol{b}^*$ and w_i^*, the distribution of $U - C$ is also uniformly random.

Therefore, we conclude that for all $i = 1, \cdots, l$ and $\gamma = 0, \cdots, t$, the distributions of $\boldsymbol{a}_{i,\gamma}$ in the both games are statistically indistinguishable. □

Proof of Theorem 2. Based on the Lemmas 8, 9 and 10, the Game 1 and Game 6 are statistically indistinguishable under the R-LWE hardness assumption. It indicates that there is no efficient adversary $\mathcal{A}$ that can win the security experiment.

Conclusion

We have constructed a new functional encryption scheme for inner product predicates from R-LWE problem. In our construction, firstly, we use setup algorithm to generate the public parameters and the master secret key. Secondly, we compute the secret key associated with the predicate vector $\vec{v}$ based on R-SIS problem using key generation algorithm. Thirdly, we calculate the ciphertext associated with the attribute vector $\vec{w}$ based on R-LWE problem using encryption algorithm. Finally, the user then can decrypt successfully using the secret key when $\langle \vec{v}, \vec{w} \rangle = 0$.

What's more, the n samples from the LWE distribution can be replaced by a sample from the R-LWE distribution, which will reduce the size of the public key by a factor of n. Hence, our scheme is more efficiency in computation than the scheme of the reference (Agrawal et al. 2011).

Some questions still remain. For example, one direction is to improve the security of our construction for researchers. Firstly, our scheme is secure under the R-LWE hardness assumption. While Rosca et al. proposed Middle-Product LWE (MP-LWE) problem as a variant of the LWE problem and proved a reduction from polynomial LWE to MP-LWE (Roşca et al. 2017). Hence, it is a open question to construct functional encryption schemes based on MP-LWE hardness assumption. Secondly, our scheme is weakly attribute hiding in security model. Therefore, we can try to construct a functional encryption scheme that is fully attribute hiding.

Acknowledgements
Not applicable.

Authors' contributions
This work is the original idea of Yang. All reporting including validating for correctness and security were performed by Fang. Correctness and security of the scheme were modified by Yang. Language modification of the article was completed by Zhang. All author(s) read and approved the final manuscript.

Funding
project is supported by the National Natural Science Foundation of China (11701089, 61822202, 61872089) and Science and Technology Program of Fujian Province, China (2019J01428).

Availability of data and materials
All data generated or analysed during this study are included in this published article.

Competing interests
The authors declare that they have no competing interests.

Received: 10 August 2020 Accepted: 29 September 2020
Published online: 10 November 2020

References

Abdalla M, Bourse F, De Caro A, Pointcheval D (2015) Simple functional encryption schemes for inner products. In: Katz J (ed). Public-Key Cryptography – PKC 2015. Springer, Berlin, Heidelberg. pp 733–751

Abdalla M, Catalano D, Gay R, Ursu B (2020) Inner-product functional encryption with fine-grained access control. IACR Cryptol ePrint Arch 2020:577

Agrawal S, Boneh D, Boyen X (2010) Efficient lattice (h)ibe in the standard model. In: Gilbert H (ed). Advances in Cryptology – EUROCRYPT 2010. Springer, Berlin, Heidelberg. pp 553–572

Agrawal S, Boyen X, Vaikuntanathan V, Voulgaris P, Wee H (2012) Functional encryption for threshold functions (or fuzzy ibe) from lattices. In: Fischlin M, Buchmann J, Manulis M (eds). Public-Key Cryptography-PKC 2015. Springer, Berlin, Heidelberg. pp 280–297

Agrawal S, Freeman DM, Vaikuntanathan V (2011) Functional encryption for inner product predicates from learning with errors. In: Lee DH, Wang X (eds). Advances in Cryptology – ASIACRYPT 2011. Springer, Berlin, Heidelberg. pp 21–40

Agrawal S, Libert B, Stehlé D (2016) Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw M, Katz J (eds). Springer, Berlin, Heidelberg. pp 333–362

Attrapadung N, Imai H (2009) Conjunctive broadcast and attribute-based encryption. In: Shacham H, Waters B (eds). Pairing-Based Cryptography – Pairing 2009. Springer, Berlin, Heidelberg. pp 248–265

Baden R, Bender A, Spring N, Bhattacharjee B, Starin D (2009) Persona: An online social network with user-defined privacy. ACM SIGCOMM Conf Appl Technol Architectures Protocol Comput Commun 39:135–146

Blundo C, Iovino V, Persiano G (2010) Predicate encryption with partial public keys. Cryptol Netw Secur 2010:476

Boneh D, Sahai A, Waters B (2011) Functional encryption: Definitions and challenges. In: Ishai Y (ed). Theory of Cryptography. Springer, Berlin, Heidelberg. pp 253–273

Boneh D, Waters B (2006) Conjunctive, subset, and range queries on encrypted data. IACR Cryptol ePrint Arch 2006:287

Camenisch J, Dubovitskaya M, Enderlein RR, Neven G (2012) Oblivious transfer with hidden access control from attribute-based encryption. In: Visconti I, De Prisco R (eds). Security and Cryptography for Networks. Springer, Berlin, Heidelberg. pp 559–579

Cash D, Hofheinz D, Kiltz E, Peikert C (2010) Bonsai trees, or how to delegate a lattice basis. In: Gilbert H (ed). Advances in Cryptology – EUROCRYPT 2010. Springer, Berlin, Heidelberg. pp 523–552

Chen J, Gong J, Wee H (2018) Improved inner-product encryption with adaptive security and full attribute-hiding. In: Peyrin T, Galbraith S (eds). Advances in Cryptology – ASIACRYPT 2018. Springer, Cham. pp 673–702

Ducas L, Lyubashevsky V, Prest T (2014) Efficient identity-based encryption over ntru lattices. In: Sarkar P, Iwata T (eds). Advances in Cryptology – ASIACRYPT 2014. Springer, Berlin, Heidelberg. pp 22–41

Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. ACM Conf Comput Commun Secur 89-98:89–98

Green MD, Miers I (2015) Forward secure asynchronous messaging from puncturable encryption. IEEE Comput Soc 2015:305–320

Katz J, Sahai A, Waters B (2008) Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart N (ed). Advances in Cryptology – EUROCRYPT 2008. Springer, Berlin, Heidelberg. pp 146–162

Kurosawa K, Phong L (2017) Anonymous and leakage resilient ibe and ipe. Des Codes Crypt 85:273–98

Lai RWF, Cheung HKF, Chow SSM (2015) Trapdoors for ideal lattices with applications. In: Lin D, Yung M, Zhou J (eds). Information Security and Cryptology. Springer, Cham. pp 239–256

LEE K (2018) Two-input functional encryption for inner products from bilinear maps. IEICE Trans Fundam Electron Commun Comput Sci E101.A:915–928
Lewko A, Okamoto T, Sahai A, Takashima K, Waters B (2010) Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert H (ed). Advances in Cryptology – EUROCRYPT 2010. Springer, Berlin, Heidelberg. pp 62–91
Li J, Zhang D, Lu X, Wang K (2018) Compact (targeted homomorphic) inner product encryption from lwe. In: Qing S, Mitchell C, Chen L, Liu D (eds). Information and Communications Security. Springer, Cham. pp 132–140
Libert B, Ţiţiu R (2019) Multi-client functional encryption for linear functions in the standard model from LWE. In: Steven DG, Shiho M (eds). Advances in Cryptology-ASIACRYPT 2019-25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III. Springer. pp 520–551
Liu Z, Jiang Z, Wang X, Yiu S (2018) Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating. J Netw Comput Appl 108:112–123
Lyubashevsky V, Micciancio D (2006) Generalized compact knapsacks are collision resistant. In: Bugliesi M, Preneel B, Sassone V, Wegener I (eds). Automata, Languages and Programming. Springer, Berlin, Heidelberg. pp 144–155
Lyubashevsky V, Peikert C, Regev O (2010) On ideal lattices and learning with errors over rings. In: Gilbert H (ed). Advances in Cryptology – EUROCRYPT 2010. Springer, Berlin, Heidelberg. pp 1–23
Micciancio D, Regev O (2004) Worst-case to average-case reductions based on gaussian measures. In: Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS. IEEE, Rome. pp 372–381. Proceedings - 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2004 ; Conference date: 17-10-2004 Through 19-10-2004
Okamoto T, Takashima K (2009) Hierarchical predicate encryption for inner-products. In: Matsui M (ed). Advances in Cryptology – ASIACRYPT 2009. Springer, Berlin, Heidelberg. pp 214–231
Okamoto T, Takashima K (2015) Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. Des Codes Cryptogr 77:725–771
O'Neill A (2010) Definitional issues in functional encryption. IACR Cryptol ePrint Arch 2010:556
Parno B, Raykova M, Vaikuntanathan V (2011) How to delegate and verify in public: Verifiable computation from attribute-based encryption. IACR Cryptol ePrint Arch 2011:597
Peikert C, Rosen A (2006) Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi S, Rabin T (eds). Theory of Cryptography. Springer, Berlin, Heidelberg. pp 145–166
Roşca M, Sakzad A, Stehlé D, Steinfeld R (2017) Middle-product learning with errors. In: Katz J, Shacham H (eds). Advances in Cryptology – CRYPTO 2017. Springer, Cham. pp 283–297
Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Cramer R (ed). Advances in Cryptology – EUROCRYPT 2005. Springer, Berlin, Heidelberg. pp 457–473
Stehlé D, Steinfeld R, Tanaka K, Xagawa K (2009) Efficient public key encryption based on ideal lattices. In: Matsui M (ed). Advances in Cryptology – ASIACRYPT 2009. Springer, Berlin, Heidelberg. pp 617–635
Tomida J (2020) Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. Theor Comput Sci 833:56–86
Tseng Y, Liu Z, Tso R (2020) Practical predicate encryption for inner product. IACR Cryptol ePrint Arch 2020:270
Wang Z, Fan X, Liu F-H (2019) Fe for inner products and its application to decentralized abe. In: Lin D, Sako K (eds). Public-Key Cryptography – PKC 2019. Springer, Cham. pp 97–127
Wang Z, Fan X, Wang M (2018) Compact inner product encryption from lwe. In: Qing S, Mitchell C, Chen L, Liu D (eds). Information and Communications Security. Springer, Cham. pp 141–153
Wei D, Gao H (2019) An inner product encryption scheme based on dual systems. Wuhan Univ J Nat Sci 24:125–133
Xagawa K (2013) Improved (hierarchical) inner-product encryption from lattices. In: Kurosawa K, Hanaoka G (eds). Public-Key Cryptography – PKC 2013. Springer, Berlin, Heidelberg. pp 235–252
Yun K, Wang X, Xue R (2018) Identity-based functional encryption for quadratic functions from lattices. In: Naccache D, Xu S, Qing S, Samarati P, Blanc G, Lu R, Zhang Z, Meddahi A (eds). Information and Communications Security. Springer, Cham. pp 409–425
Zhang D, Li J, Li B, Lu X, Xue H, Jia D, Liu Y (2019) Deterministic identity-based encryption from lattice-based programmable hash functions with high min-entropy. Secur Commun Netw 2019:1–12
Zhang L, Wu Q (2017) Adaptively secure hierarchical identity-based encryption over lattice. In: Yan Z, Molva R, Mazurczyk W, Kantola R (eds). Network and System Security. Springer, Cham. pp 46–58

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.