NEW PERSPECTIVES ON LATTICE-BASED FUNCTIONAL ENCRYPTION FOR ANALYSIS ON ENCRYPTED DATA

Zongchen Sun and Leyou Zhang

School of Mathematics and Statistics
Xidian University
No. 266, Xinglong Section of Xifeng Road, Xi'an 710126, P. R. China
szcnnn@163.com; lyzhang@mail.xidian.edu.cn

Received September 2016; accepted December 2016

Abstract. *Traditional method of analysis on encrypted data is to get the plaintexts at first, which issues some inevitable security threats in the real life. A solution to this problem is functional encryption (FE) that enables direct computation on encrypted data. This novel paradigm breaks all-or-nothing access model. In this work, we discuss how to construct a succinct lattice-based functional encryption scheme by linear FE (short as LinFE) for inner products and show some new perspectives on the applications of it, such as information filtering, technology outsourcing and data analysis.*

Keywords: Functional encryption, Lattice, Learning with errors, Analysis on encrypted data

1. **Introduction.** The advent of cloud computing and the resulting demand for privacy-preserving require that we have to provide the new encryption technologies. Over the past decade, cryptographers have put forth some novel paradigms for public-key encryption (PKE): attribute-based encryption (ABE), predicate encryption (PE) and functional encryption (FE), etc. Especially, functional encryption breaks all-or-nothing access model, in which a secret key enables a user to learn a specific function of the encrypted data and nothing else. Obviously, the notion of functional encryption can be regarded as an "epitome" of the traditional PKE. Functional encryption provides both fine-grained access and computing on encrypted data, which has significantly advanced the state of the art in the field of cryptography.

According to the notion of functional encryption formalized by Boneh et al. [1], many constructions of functional encryption were proposed. Most of them focus on constructing FE for restricted classes of functions, such as point functions (or IBE) [2,3], threshold functions [4], Boolean formulas [5], inner product functions [6,7] and even regular languages [8]. However, the well-known genuine schemes for general circuits relied on indistinguishability obfuscation [9,10], which made them rely on either an exponential number of assumptions or a polynomial set of assumptions with exponential loss in the security reduction. In recent works [7,11], the problem of lattice-based FE for bounded collusions has been resolved without strong assumptions, such as indistinguishability obfuscation, and multilinear map.

However, the construction in [11] also has some disadvantages. In order to prevent an attacker getting the corresponding combination modulo p of master key components, the sateful FE scheme [11] requires that adversary only queries secret keys for some independent vectors. This assumption substantially underestimates the ability of the adversary. Inspired by Agrawal et al.'s work [11], we present a stateless FE for inner products that can be extended to a bounded collusion functional encryption for all circuits. In addition, the plaintext in our succinct FE scheme is not a vector that suffices for many concrete applications.

2. **Preliminaries.**

2.1. **Lattice.** Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_m] \in \mathbb{R}^{m\times m}$ be an $m \times m$ matrix whose columns are linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^m$. The m-dimensional full-rank lattice Λ generated by B is the set

$$\Lambda := \left\{ \mathbf{y} \in \mathbb{R}^m \text{ s.t. } \exists \mathbf{s} \in \mathbb{Z}^m,\ \mathbf{y} = B\mathbf{s} = \sum_{i=1}^{m} s_i b_i \right\}.$$

Definition 2.1. *For a prime q, $A \in \mathbb{Z}^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$ define:*

$$\Lambda_q^{\perp} := \{\mathbf{e} \in \mathbb{Z}^m | A\mathbf{e} = 0(\text{mod} q)\},\ \Lambda_q^{\mathbf{u}} := \{\mathbf{e} \in \mathbb{Z}^m | A\mathbf{e} = \mathbf{u}(\text{mod} q)\}.$$

Theorem 2.1. [3]. *Let $q \geq 2$ be odd and $m := \lceil 6n \log q \rceil$. There is a probabilistic polynomial time (PPT) algorithm TrapGen(q, n) that outputs a pair $(A \in \mathbb{Z}_q^{n\times m},\ T_A \in \mathbb{Z}^{m\times m})$ satisfying $\left|\left|\tilde{T}_A\right|\right| \leq O(\sqrt{n \log q})$ and $\left|\left|T_A\right|\right| \leq O(n \log q)$ with all but negligible in n.*

We need to sample short vectors in $\Lambda_q^{\mathbf{u}}(A)$ for some $\mathbf{u}$ in $\mathbb{Z}_q^n$ and define the following algorithms.

Algorithm *SamplePre* (A, T_A, u, σ):
Inputs:
a rank n matrix A in $\mathbb{Z}_q^{n\times m}$,
a "short" basis T_A of $\Lambda_q^{\perp}(A)$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$,
a Gaussian parameter $\sigma \geq \left|\left|\tilde{T}_A\right|\right| \omega\left(\sqrt{\log m}\right)$.
Output: a vector $\mathbf{x} \in \Lambda_q^{\mathbf{u}}(A)$ sampled from a distribution statistically close to $D_{\Lambda^{\mathbf{u}},\sigma}$,
whenever $D_{\Lambda^{\mathbf{u}},\sigma}$ denotes Discrete Gaussians (more details see [3]).

Theorem 2.2. [3]. *Let $q \geq 2$ and A be a matrix in $\mathbb{Z}_q^{n\times m}$ with $m > n$. Let T_A be a basis for $\Lambda_q^{\perp}(A)$ and $\sigma \geq \left|\left|\tilde{T}_A\right|\right| \omega\left(\sqrt{\log m}\right)$. Then for $\mathbf{u} \in \mathbb{Z}_q^n$:*

$$\Pr\left[x \sim D_{\Lambda_q^{\mathbf{u}}(a),\sigma} : ||\mathbf{x}|| > \sqrt{m}\sigma\right] \leq negl(n).$$

2.2. **The LWE hardness assumption.** The learning with errors (LWE) problem, a classic problem on lattice, was first defined by Regev [12]. The security of our scheme is based on the hardness of this problem.

Definition 2.2. *For a positive integer n, a prime q and a distribution χ over $\mathbb{Z}_q$, a (Z_q, n, χ)-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being, either a noisy pseudo-random sampler $\mathcal{O}_{\mathbf{s}}$ carrying some constant random secret key $\mathbf{s} \in \mathbb{Z}_q^n$, or, a truly random sampler $\mathcal{O}_{\$}$, whose behaviors are respectively as follows:*

- *$\mathcal{O}_{\mathbf{s}}$: outputs samples of the form*

$$(\mathbf{u}_i, v_i) = \left(\mathbf{u}_i, \mathbf{u}_i^T\mathbf{s} + x_i\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

where $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly distributed vector which keeps persistent across invocations, $x_i \in \mathbb{Z}_q$ is a fresh sample form χ, and $\mathbf{u}_i$ is uniformly sampled from $\mathbb{Z}_q^n$.

- *$\mathcal{O}_{\$}$: outputs samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$ uniformly at random.*

We call that an algorithm B decides the $(\mathbb{Z}_q, n, \chi) - LWE$ problem if

$$Adv_B^{(\mathbb{Z}_q,n,\chi)-LWE} = \left|\Pr\left[B^{\mathcal{O}_{\mathbf{s}}} = 1\right] - \Pr\left[B^{\mathcal{O}_{\$}} = 1\right]\right|$$

is non-negligible for a random $\mathbf{s} \in \mathbb{Z}_q^n$.

2.3. **Functional encryption.** A functional encryption [1] scheme for f consists of four algorithms, FE = (*FE.Setup*; *FE.Keygen*; *FE.Enc*; *FE.Dec*) which are defined as follows.

- *FE.Setup*(1^λ). It takes as input the unary representation of the security parameter and outputs the master public and secret keys (*mpk*; *msk*).
- *FE.Keygen*(*msk*; f). It takes as input the master secret key *msk* and a circuit f and outputs a corresponding secret key sk_f.
- *FE.Enc*(*mpk*; x). It takes as input the master public key *mpk* and message x and outputs a ciphertext *Ct*.
- FE.*Dec*(sk_f; *Ct*). It takes as input the secret key sk_f and a ciphertext *Ct* and outputs $f(x)$.

Definition 2.3. *A functional encryption scheme FE is correct if for all $f \in F$ and all $x \in X$.*

$$\Pr\left[\begin{array}{c}(mpk, msk) \leftarrow FE.Setup(1^\lambda); \\ FE.Dec(FE.keygen(msk, f),\ FE.Enc(mpk, x)) \neq f(x)\end{array}\right] = negl(\lambda).$$

3. The Proposed Functional Encryption Scheme.

3.1. **LinFE for inner products from LWE.** *LinFE.Setup*$(1^\lambda, 1^l)$. Given the security parameter λ and the length of plaintext l, it proceeds as follows.

- Use the algorithm *TrapGen* to generate a random $n \times m$ matrix A with a full-rank m-vector set $T_A \subseteq \Lambda_q^\perp(A)$.
- Select a uniformly random matrix $U \in \mathbb{Z}_q^{n\times l}$.
- Output the public parameters *mpk* and master secret key *msk* given by

$$mpk = (A, U),\ msk = T_A.$$

LinFE.KeyGen($\mathbf{y}$, *msk*). Given the master secret key *msk* and predicate vector $\mathbf{y} \in \mathbb{Z}_p^l$, it outputs a secret key $sk = \{\mathbf{e} \leftarrow SamplePre(A, T_A, U\mathbf{y}, \sigma)\} \in \mathbb{Z}_q^m$.

LinFE.Enc(*mpk*, $\mathbf{x}$). Given the public parameters *mpk* and a message $\mathbf{x} \in \mathbb{Z}_p^l$, it computes ciphertext *Ct* as follows.

- Choose a noise vector $\boldsymbol{\varepsilon}_0 \leftarrow \chi^m$, $\boldsymbol{\varepsilon}_1 \leftarrow \chi^l$, where χ denotes noise distribution.
- Choose a uniformly random vector $\mathbf{s} \in \mathbb{Z}_q^n$.
- Compute $\mathbf{c}_0 = A^\top\mathbf{s} + \boldsymbol{\varepsilon}_0$ and $\mathbf{c}_1 = U^\top\mathbf{s} + \boldsymbol{\varepsilon}_1 + \left\lfloor \frac{q}{mp^2} \right\rfloor \mathbf{x} \in \mathbb{Z}_q^l$.
- Return the ciphertexts $Ct = (\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^{m+l}$.

LinFE.Dec(*sk*, *Ct*). Given the receiver's *sk* and the ciphertexts *Ct*, it computes the evaluation $\mu' =< \mathbf{y},\ \mathbf{c}_1 > - < \mathbf{e},\ \mathbf{c}_0 >$ mod q and outputs the value μ that minimizes $\left|\left\lfloor \frac{q}{mp^2} \right\rfloor \mu - \mu'\right|$.

Correctness. For all $(mpk, msk) \leftarrow LinFE.Setup\left(1^\lambda\right)$, all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^l$, $sk \leftarrow LinFE.KeyGen$ and $Ct = LinFE.Enc$, we have that

$$\begin{aligned} \mu' &= < \mathbf{y},\ \mathbf{c}_1 > - < \mathbf{e},\ \mathbf{c}_0 > \bmod q \\ &\approx \mathbf{y}^\top \left(U^\top\mathbf{s} + \boldsymbol{\varepsilon}_1 + \left\lfloor \frac{q}{mp^2} \right\rfloor \mathbf{x}\right) - \mathbf{e}^\top \left(A^\top\mathbf{s} + \boldsymbol{\varepsilon}_0\right) \bmod q \\ &= \left\lfloor \frac{q}{mp^2} \right\rfloor \mathbf{y}^\top \cdot \mathbf{x} + \underbrace{\left(\mathbf{y}^\top\boldsymbol{\varepsilon}_1 - \mathbf{e}^\top\boldsymbol{\varepsilon}_0\right)}_{\textit{low-norm-noise} \in \mathbb{Z}_p} \bmod q. \end{aligned}$$

For appropriate parameters [3,11], the noise bound could be less than $\left\lfloor \frac{q}{4mp^2} \right\rfloor$, which suffices to guarantee decryption correctness.

TABLE 1. A comparison with the current available scheme [11]

scheme	*mpk*	*msk*	*sk*	*ciphertext*	*state*	*adversary*
[11]	$O(n^2 \log q)$ + $O(nl \log q)$	$O(nl)$	$O(n) + O(l)$	$O(n) + O(l)$	stateful	bounded
Ours	$O(n^2 \log q)$ + $O(nl)$	$O(n^2 \log^2 q)$	$O(n \log q)$	$O(n \log q) + O(l)$	stateless	regular

*Note that our *msk* and *sk* are independent of plaintext and the assumed adversary is stronger than [11] at the (slight) expense of the magnitude of other parameters i.e., $O(\log q)$.

Efficiency analysis. In Table 1, l denotes the length of plaintext and n denotes the row of matrix A. Bounded adversary means that he/she only queries secret keys for some controlled vectors.

Semantic security. Notice that the above scheme based on Regev PKE scheme is IND-FE-CPA under LWE assumption. The proof of our scheme is analogous to that in [12]. Due to space limitations, we omit the details.

3.2. **FE for regular circuit.** In this section, we describe how to convert our LinFE into an FE scheme for circuits. We refer the reader to [13] for more details about randomized encodings (RE).

FE.Setup(1^λ). It invokes *LinFE.Setup* and returns (*mpk*, *msk*).

FE.KeyGen(*msk*, f). Given the master secret key *msk* and a circuit f, it works as follows.

- Encode f by a sequence of degree 3 polynomials $P_1, \ldots, P_k$.
- Linearize each polynomial P_i and let $P_i{}'$ be its vector of coefficients, such as $P_i' = (a, b, c, d)$.
- Output $FE.sk_f = \{sk_i = LinFE.KeyGen(msk, P_i')\}_{i \in [k]}$.

FE.Enc(*mpk*, x). Given the master public key and plaintext x, it outputs the ciphertext $Ct_x = LinFE.Enc(x^3, x^2, x, 1)$.

FE.Dec(*mpk*, sk_f, CT_x). Given a secret key sk_f and ciphertext Ct_x for message x, it works as follows.

- Compute $\{P_i(x)\}_{i \in [k]} = \{LinFE.Dec(Ct_x, sk_i)\}_{i \in [k]}$.
- Run the decoder for the randomized encoding to recover $f(x)$.

Semantic security and Correctness. We refer the reader to [14] the definition of no-adaptive simulation (NA-SIM) secure.

Theorem 3.1. *Let the underlying scheme LinFE and RE be NA-SIM, for any family F of polynomial-size circuits, the FE scheme described above is q-AD-SIM-secure against a bounded number of collusions.*

We briefly sketch the proof. Let *RE.Sim* and *LinFE.Sim* be the simulator guaranteed by the security of RE and LinFE scheme respectively. Given secret key queries f_i, the corresponding secret keys $FE.sk_i$ and the values $f_i(x)$ for $i \in [q^*]$, $q^* \leq q$, our simulator *FE.Sim* works as follows (more details see [11,15]).

- For each $i \in [q^*]$, invoke $RE.Sim\,(f_i(x))$ to learn $(P_1^i(x), \ldots, P_k^i(x))$.
- Output $Ct_x = LinFE.Sim\left(\{P_1^i(x), \ldots, P_k^i(x), FE.sk_i\}_{i \in [q^*]}\right)$.

Obviously, the correctness of our FE scheme follows from the correctness of LinFE and RE.

4. **New Perspectives on Applications.** Today public key encryption is an invaluable tool and has been used in secure web communication (e.g., HTTPS and SSL), voice traffic, storage systems, etc. Functional encryption can be viewed as a generalization of many

advances in public-key encryption over the past decade, in which the sender can determine who can decrypt a specific ciphertext and how much information receivers can get about the plaintext. Let the function f be an identity function such that functional encryption is actually equal to general PKE. Here, we discuss the three examples, closely related to our research in functional encryption.

4.1. **Information filtering.** Since the amount of data (good or bad) in our world has been exploding, information filtering is an indispensable technology to prevent "spam" flood. However, lots of information is encrypted in network or cloud, which incurs some problems for information filtering. In addition, the ciphertexts are public, which can be intentional tampered. To ensure network security, some regulators have to momently monitor the encrypted data, but cannot snoop the personal private data. Functional encryption provides a great solution. In this model, a data owner will leverage partial information, i.e., $f(x)$, where the function f should be regulated or public and cannot reveal anything about the data itself. This cryptosystem can be embedded in many instantiations, such as search engine tool, and email filtering. Note that description about spam filtering in [13] is incomplete because the recipient cannot obtain message (also can get $f(x)$ from functional encryption). A simple solution is to add another traditional PKE, which is a real encryption system. Here FE scheme is just a testing tool (Figure 1).

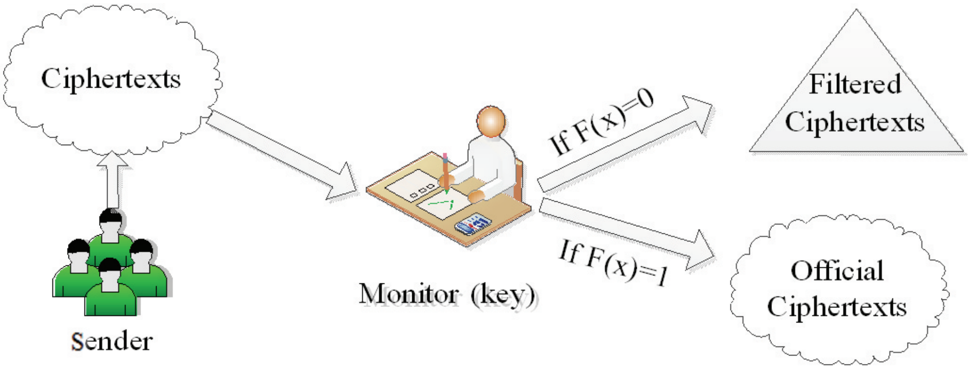

FIGURE 1. A simplified test tool of information filtering

4.2. **Technology outsourcing.** Suppose that two companies are business partners and they need to provide technical support for each other. However, they are not willing to reveal the commercial secret, i.e., their core technology. For instance, company A owned the kernel code x needs to provide company B with related service about this code. In this case, there is a trusted authority who can generate a derived secret key sk_f associated with f. Company A holding sk_f can compute $f(x)$ from an encryption of any data x, where $f(x)$ is the result of running this kernel code. It should be noted that f must be well-designed by the owner; otherwise some attackers may obtain the secret x.

4.3. **Data analysis.** We take the example of smart grid privacy protection. Smart grids not only bring numerous advantages in terms of energy consumption reduction, but also lead to an enhancement in the ability of monitoring. Such management might reveal their personal habits and behavior, which electrical appliances they are using at each moment, whether they are at home or not, and so on. Consequently, customer privacy has to be protected while at the same time smart meters can send detailed energy measurement reports on customers. In our scenario, an energy supplier (ES) receives electricity measurements (encrypted individual values Ct) from smart meters. It can obtain secret key sk_f for desired but constrained f from the key distribution center (KDC) and then decrypt the ciphertext to get $f(x)$. At last, ES will adjust the electrical power system by the evaluation of $f(x)$.

5. **Conclusions and Open Problems.** In this paper, a stateless FE for inner products is proposed based on LWE assumption. Then a simple method is introduced to convert the proposed scheme into an FE scheme for regular circuit based on randomized encodings. Furthermore, we discuss some new perspectives on the applications of the proposed schemes, such as information filtering, technology outsourcing and secret data analysis. However, the construction of f may be a block for practical application scenarios in the real life. In addition, how to construct practical FE scheme for randomized functionalities without making any additional strong assumptions is still an open problem.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China under Grants (61472307, 61100165, and 61100231), and Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JM6004). The authors will also gratefully acknowledge the helpful comments and suggestions of the reviewers.

REFERENCES

[1] D. Boneh, A. Sahai and B. Waters, Functional encryption: Definitions and challenges, *Theory of Cryptography*, vol.6597, pp.253-273, 2011.

[2] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology – CRYPTO 2001*, CA, USA, pp.213-229, 2001.

[3] S. Agrawal, D. Boneh and X. Boyen, Efficient lattice (H) IBE in the standard model, *Advances in Cryptology – EUROCRYPT 2010*, Riviera, French, pp.553-572, 2010.

[4] A. Sahai and B. Waters, Fuzzy identity-based encryption, *Advances in Cryptology – EUROCRYPT 2005*, Aarhus, Denmark, pp.457-473, 2005.

[5] V. Goyal, O. Pandey, A. Sahai and B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, *Proc. of the 13th ACM Conference on Computer and Communications Security*, pp.89-98, 2006.

[6] S. Agrawal, D. M. Freeman and V. Vaikuntanathan, Functional encryption for inner product predicates from learning with errors, *Advances in Cryptology – ASIACRYPT 2011*, Seoul, South Korea, pp.21-40, 2011.

[7] M. Abdalla, F. Bourse, A. De Caro and D. Pointcheval, Simple functional encryption schemes for inner products, *Public-Key Cryptography – PKC 2015*, Gaithersburg, USA, pp.733-751, 2015.

[8] B. Waters, Functional encryption for regular languages, *Advances in Cryptology – CRYPTO 2012*, Santa Barbara, USA, pp.218-235, 2012.

[9] S. Garg, C. Gentry, S. Halevi et al., Candidate indistinguishability obfuscation and functional encryption for all circuits, *The 54th Annual Symposium on Foundations of Computer Science*, pp.40-49, 2013.

[10] B. Barak, S. Garg, Y. T. Kalai, O. Paneth and A. Sahai, Protecting obfuscation against algebraic attacks, *Advances in Cryptology – EUROCRYPT 2014*, Copenhagen, Denmark, pp.221-238, 2014.

[11] S. Agrawal, B. Libert and D. Stehlé, *Fully Secure Functional Encryption for Inner Products, from Standard Assumptions*, IACR Cryptology ePrint Archive, https://eprint.iacr.org/2015/608.pdf, 2015.

[12] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, *Proc. of the 37th Annual ACM Symposium on Theory of Computing*, pp.84-93, 2005.

[13] D. Boneh, A. Sahai and B. Waters, Functional encryption: A new vision for public-key cryptography, *Communications of the ACM*, vol.55, no.11, pp.56-64, 2012.

[14] A. O'Neill, *Definitional Issues in Functional Encryption*, IACR Cryptology ePrint Archive, http://eprint.iacr.org/2010/556.pdf, 2010.

[15] S. Gorbunov, V. Vaikuntanathan and H. Wee, Functional encryption with bounded collusions via multi-party computation, *Advances in Cryptology – CRYPTO 2012*, Santa Barbara, USA, pp.162-179, 2012.