Peer-to-Peer Networking and Applications (2020) 13:2334–2341
https://doi.org/10.1007/s12083-020-00907-4

Functional encryption with application to machine learning: simple conversions from generic functions to quadratic functions

Huige Wang[1,2] · Kefei Chen[3,4] · Yuan Zhang[5] · Yunlei Zhao[2]

Received: 30 September 2019 / Accepted: 19 March 2020 / Published online: 13 April 2020

Abstract

Functional encryption (FE) and predicate encryption (PE) can be utilized in deploying and executing machine learning (ML) algorithms to improve efficiency. However, most of existing FE and PE algorithms only consider generic functions. Actually, quadratic-functions-based FE and PE can be used to further reduce the computation costs significantly. In this paper, we present a functional encryption scheme for quadratic functions from those for generic functions. In our constructions, ciphertexts are associated with a pair of vectors $(\mathsf{x}, \mathsf{y}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m$, private keys are associated with a quadratic function, and the decryption of ciphertexts $CT_{(\mathsf{x},\mathsf{y})}$ with a private key sk_{F}, where F is a $n \times m$-dimensional matrix, recovers $(\mathsf{x})^{\top}\mathsf{F}\mathsf{y} \in \mathbb{Z}_q$. Compared with Baltico et al.'s FEs for quadratic functions (at Crypto 2017), our schemes could obtain almost the same ciphertexts size of $O((n+m)\log q)$ as their schemes (in contrast to $O(n)$ in Baltico et al.'s schemes), and the computation for quadratic functions in our scheme does not rely on bilinear maps, while their schemes must rely on this assumption. In particular, our schemes under the standard assumptions achieve adaptive security, while Baltico et al.'s scheme only obtains selective security. Moreover, beyond the MDDH and GGM assumptions, our schemes allow for instantiations under standard assumptions such as LWE, LPN, and etc.

Keywords Quadratic functions · Functional encryption · Adaptive security · Generic conversions · Machine learning

1 Introduction

Applying emerging ML algorithms with the integration of cloud computing [38] in reality has already brought large benefits for people [21, 30]. For example, in cloud-assisted eHealth systems [19, 36, 37], with these ML algorithms, a medical institution can train a cloud server (which is subject to a cloud service provider) to deploy ML models on the server. After that, the cloud server is able to provide healthcare services for users (e.g., patients) without requiring the participation of the medical institution. By doing so, the medical institution can outsource the healthcare services to the cloud server and enables the users to leverage the services in an efficient and convenient way.

Despite the conveniences and benefits brought by ML algorithms, critical security and privacy concerns [14, 31] in training the cloud server [33] and requesting services from it have been raised seriously [22]. Specifically, in most ML algorithms, it is inevitable to leak training data and key information to the cloud server during the deploying ML models, and the cloud server is able to extract users privacy when it provides services for users. As a consequence, a malicious cloud server (or a malicious insider working at the service provider) can illegally gain profits from the leaked training data and users' privacy, and the security and reliability of the system cannot be guaranteed [16].

To protect the training data and users' privacy [13, 24, 32] against the cloud server, several privacy-preserving ML algorithms [11, 15, 23] are proposed. Most of them are constructed on secure multi-party computation (SMC) and homomorphic encryption (HE). However, due to the low efficiency to perform SMC and HE, both the users (including those who outsource services to the cloud server) and the cloud server, who execute these algorithms, bear high costs in terms of computation and communication.

To improve the efficiency while remaining the functionalities of ML algorithms, emerging cryptographic primitives

This article is part of the Topical Collection: *Special Issue on Security and Privacy in Machine Learning Assisted P2P Networks*
Guest Editors: Hongwei Li, Rongxing Lu and Mohamed Mahmoud

✉ Huige Wang
wanghuige@fudan.edu.cn

Extended author information available on the last page of the article.

[17, 18], such as functional encryption (FE) [12, 27, 28] and predicate encryption (PE), are employed. Recent literatures [8, 34] have shown the great improvement of efficiency in deploying and executing ML algorithms by utilizing FE and PE.

From the perspective of technique, a functional encryption for a functionality F defined over a key space K and a message space $\mathcal{M}$, performs the computation of $F(K; M)$ from the key sk_K, associated to a key $K \in \mathcal{K}$, and a ciphertext CT_M which encrypts the message $M \in \mathcal{M}$. Since the original functional encryption was proposed in a long line of researches on it have been spurred. Predicate encryption is a special functional encryption where ciphertexts $CT_{X,M}$ are associated with a plaintext M and an attribute X, secret keys sk_P are associated with a boolean predicate P, and the decryption of a ciphertext $CT_{X,M}$ recovers the encrypted plaintext M with a private key sk_P, if and only if $P(X) = 1$. However, in existing functional encryption and predicate encryptions, only generic functions are considered but not quadratic functions, and thus result in lower efficiency.

Our contributions. In this paper, we present a simple transformation from functional encryption that supports generic functions to one that supports quadratic functions (that include inner product function). In our scheme, ciphertexts $CT_{(\mathsf{x},\mathsf{y})} \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m$ are associated with a pair of vectors (x, y) (when performing encryption, (x, y) can be seen as a string of length $(n + m)\log q$), private keys sk_{F} are associated with a quadratic function F (where $\mathsf{F} \in \mathbb{Z}_q^{n\times m}$ can be seen as the key of F), and the decryption of a ciphertext $CT_{(\mathsf{x},\mathsf{y})}$ with a private key sk_{F} recovers $(\mathsf{x})^\top \mathsf{F}\mathsf{y} \in \mathbb{Z}_q$, where $q > 2^\lambda$ is a prime number, and n, m is positive natural number. The ciphertexts in our schemes have the improved size of length $O((n + m)\log q)$ rather than $O(nm \log q)$ which is the case in [1] where they proposed an FE scheme for inner product functions (our scheme can implement the inner product functionality if taking the key matrix F as the identity matrix I).

In particular, our schemes remove the bilinear maps that is used to implement the quadratic functionality. For the instantiations, if we take the FE schemes proposed by Gorbunov et al. in [10] as the underlying FE schemes, then, beyond the MDDH and GGM assumptions, our schemes also allow for instantiations under other standard assumptions such as the decisional diffie-hellman (DDH) [6], RSA [7], learning with errors (LWE) [25], and learning parity with noise (LPN) [35]. This is because, the semantic secure public-key encryption schemes that is used to build their FE schemes can be instantiated under these assumptions. Of course, if we adopt the FE schemes in [29] and [9] as the underlying FE schemes, then we can again obtain instantiations from non-standard assumptions such as indistinguishability obfuscator (IO) [29] and multilinear maps [9].

Overview of Our FE for Quadratic Functions. Our schemes work over prime fields $\mathbb{Z}_q$ such that $q > 2^\lambda$ is a prime number and λ is the security parameter used in our schemes. They are quite efficient in communication size: public key and private key has flexible length changed with the constructions of the underlying FE schemes for generic functions. The ciphertexts in our schemes obtain comparable size as that in [4]. The ideas for designing the generic transformation is very simple where we only use a sufficiently-expressive FE scheme for generic functions, beyond which, any additional assumptions are not introduced. Our both schemes (in the public-key and secret-key setting) could be proved adaptively secure, where security is guaranteed even for messages that are adaptively chosen at any point in time, under the the same security assumption of the underlying FE schemes. In the following, we will highlight some of the core ideas in our schemes.

Now, we first introduce the functionality that our schemes support. Specifically, the functionality refers to that for an instantiation expressed in the form of a pair of vectors $(\mathsf{x}, \mathsf{y}) \in \mathcal{M}$ encrypted in a ciphertext $CT_{(\mathsf{x},\mathsf{y})}$, and a function F presented as a matrix $\mathsf{F} \in \mathcal{K}$, the decryption for the ciphertext $CT_{(\mathsf{x},\mathsf{y})}$ under the private key sk_{F} with which the function F is associated with, allows to compute a quadratic function value $(\mathsf{x})^\top \mathsf{F}\mathsf{y} \in \mathcal{Y}$, where the function F is defined as $F : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$.

The first thing we think about is to encrypt the pair of vectors (x, y) into a ciphertext $CT_{(\mathsf{x},\mathsf{y})}$ under the underlying FE scheme FE for generic functions which are computable by a polynomially-size circuit $\mathcal{G}$. Toward finding a decryption method, we first observe that, given $CT_{(\mathsf{x},\mathsf{y})}$ and a private key sk_{G} (associated with the circuit class $\mathcal{G}$), under the FE scheme FE, we can compute the function value $(\mathsf{x})^\top \mathsf{F}\mathsf{y} \in \mathcal{Y}$. However, without any extra processing, this is obviously infeasible. In order to achieve this goal, we embed the computation of $\mathsf{x})^\top \mathsf{F}\mathsf{y}$ in the circuit $\mathcal{G}[\mathsf{F}]$ with the function F hardwired in. Then take the pair (x, y) as the inputs of the circuit $\mathcal{G}$ and endow the circuit the functionality of computing $\mathsf{x})^\top \mathsf{F}\mathsf{y}$ with the hardwired function F and the pair (x, y). By running the decryption, we finally get a desired result.

Concurrent and Independent work. In concurrent and independent work, Lin [20], and Ananth and Sahai [3] present constructions of private-key functional encryption schemes for degree-D polynomials based on D-linear maps. If taking $D = 2$, these schemes support quadratic polynomials from bilinear maps. In 2017, Baltico et al. [4] also propose constructions of functional encryption (both in the private-key and public-key settings) for quadratic functions from the MDDH and GGM assumptions under the existence of bilinear maps. Their GGM-based schemes are proved adaptively secure but their MDDH-based schemes only achieve selective security. In comparison to these works, our schemes have the advantage of working without pairings and can be proved adaptively secure only under the same security of the underlying FE schemes which is easy to achieve, since such security can be obtained by many existing methods.

2 Preliminaries

In this section, we introduce some notations and cryptographic building blocks used in this paper.

2.1 Notations

Throughout the paper, $\mathbb{N}$ denotes the set of natural numbers and $\lambda \in \mathbb{N}$ denotes the security parameter. Let $y \leftarrow A(x_1, \cdots; R)$ denote the operation of running algorithm A on inputs $x_1, \cdots$ and coins R to output y. For simplicity, we write $y \leftarrow A(x_1, \cdots; R)$ as $y \leftarrow_{\$} A(x_1, \cdots)$ with implied coins. If $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \cdots, n\}$. We call a function $negl$ negligible in λ if $negl(\lambda) \in \lambda^{-\omega(1)}$ and a function $poly$ a polynomial if $poly \in \lambda^{\mathcal{O}(1)}$. If x denotes a vector, then $|\mathsf{x}|$ denotes the number of components in x and x_i denotes the $i-th$ component of the vector x. If P denotes circuit, then we use notation $\mathsf{P}[z](\cdot)$ to emphasize the fact that the value z is hard-coded into P.

In this paper, for security definition and proofs we use a code-based game playing framework in [5, 26]. A game G has a main procedure, and possibly other procedure. G begins by executing the main procedure which runs an adversary A after some initialization. A can make oracle calls permitted by G. When A finishes execution, G continues to execute with A's output. By $\mathsf{G}^A \Rightarrow y$, we denote the event that G executes with A to output y. Generally, we abbreviate $\mathsf{G}^A \Rightarrow true$ or $\mathsf{G}^A \Rightarrow 1$ as G, and boolean flags and sets are initialized to false and $\emptyset$ respectively.

Furthermore, given a matrix of scalars $\mathsf{F} = (f_{i,j}) \in \mathbb{Z}_q^{n \times m}$ and two vectors of $\mathsf{a} \in \mathbb{Z}_q^n, \mathsf{b} \in \mathbb{Z}_q^m$, one can efficiently compute

$$\mathsf{a}^\top \mathsf{F} \mathsf{b} = \sum_{i \in [n], j \in [m]} f_{i,j} a_i b_j .$$

2.2 Quadratic function

Let $n, m \in \mathbb{N}^+$ be positive integers, $q > 2^\lambda$ be a prime number. We let the message space $\mathcal{M} := \mathbb{Z}_q^n \times \mathbb{Z}_q^m$ be a pair of vectors (x, y). The key space consists of matrices $\mathcal{K} := \mathbb{Z}_q^{n \times m}$, every key $K \in \mathcal{K}$ is a matrix $\mathsf{F} = (f_{i,j})$ and the output space is $\mathcal{Y} := \mathbb{Z}_q$. The functionality $F(K, M)$ is the one that computes the value $\mathsf{x}^\top \mathsf{F} \mathsf{y} \in \mathbb{Z}_q$, where $K = \mathsf{F}$ and $M = (\mathsf{x}, \mathsf{y}) \in \mathcal{M}$.

3 Functional encryption

In the following, we review the definition of functional encryption from [4].

Functionality. In our scheme, we will use the class of functionalities $F : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ where $\mathcal{K}$ denotes the key space, $\mathcal{M}$ denotes the message space, and $\mathcal{Y}$ denotes the output space of the function F and these spaces are defined respectively as Section 3.

Definition 1 (Functional Encryption) A functional encryption scheme FE for a functionality F in the public-key setting (resp., in the private-key setting which adds the boxed parameter) consists of a tuple of algorithms FE=($\mathsf{FE.Setup}$, $\mathsf{FE.KeyGen}$, $\mathsf{FE.Enc}$, $\mathsf{FE.Dec}$) that works as follows.

$\mathsf{FE.Setup}(1^\lambda, F)$. On input a security parameter 1^λ and a functionality F, the algorithm $\mathsf{FE.Setup}(1^\lambda, F)$ outputs a master public key mpk and a master secret key msk.

$\mathsf{FE.KeyGen}(msk, K)$. On input a master secret key msk and a functionality key $K \in \mathcal{K}$, the algorithm $\mathsf{FE.KeyGen}(msk, K \in \mathcal{K})$ outputs a private key sk_K.

$\mathsf{FE.Enc}(mpk, \boxed{\text{msk}}, M)$. On input a master public key mpk and a message $M \in \mathcal{M}$, the algorithm $\mathsf{FE.Enc}(mpk, M)$ outputs a ciphertext CT in the public-key setting. While, in the private-key setting, the algorithm will additionally take a master secret key msk as input, similarly hereinafter.

$\mathsf{FE.Dec}(sk_K, CT)$. On input a private key sk_K and a ciphertext CT, the algorithm $\mathsf{FE.Dec}(sk_K, CT)$ outputs a $y \in \mathcal{Y} \cup \{\perp\}$.

Correctness. For correctness, it is required that for all queries $K \in \mathcal{K}$, and all message $M \in \mathcal{M}$, if $sk_K \leftarrow \mathsf{FE.KeyGen}(msk, K)$ and $\mathsf{FE.Enc}(mpk, \boxed{\text{msk}}, M)$, then it holds with overwhelming probability that $\mathsf{FE.Dec}(sk_K, CT) = F(K, M)$ when $F(K, M) \neq \perp$.

Definition 2 (Adaptive Indistinguishable-Based Security) For the adaptive indistinguishable-based chosen-plaintext (a-IND-CPA) security, we use $\mathsf{aINDCPA}_{\mathcal{A},F}^{\mathsf{FE},b}(\lambda)$ (see Fig. 1) to denote a-IND-CPA game between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. We define the advantage of $\mathcal{A}$ in game $\mathsf{aINDCPA}_{\mathcal{A},F}^{\mathsf{FE},b}(\lambda)$ as $\mathsf{Adv}_{\mathsf{FE},F,\mathcal{A}}^{\mathsf{aINDCPA}}(\lambda) = \Pr[\mathsf{aINDCPA}_{\mathcal{A},F}^{\mathsf{FE},0}(\lambda) = 1] - \Pr[\mathsf{aINDCPA}_{\mathcal{A},F}^{\mathsf{FE},1}(\lambda) = 1]$.

Definition 3 (Selective Indistinguishable-Based Security) For the selective indistinguishable-based chosen-plaintext (s-IND-CPA) security, where the challenge messages are required to deliver before the master public key and key queries. We use $\mathsf{sINDCPA}_{\mathcal{A},F}^{\mathsf{FE},b}(\lambda)$ (see Fig. 2) to denote s-IND-CPA game between a PPT adversary $\mathcal{A}$ and a

Fig. 1 Adaptive IND-CPA Experiment for FE

$\mathsf{aINDCPA}^{\mathsf{FE},b}_{\mathcal{A},F}(\lambda)$	$\mathsf{KeyGenO}(msk, K)$:
$(mpk, msk) \leftarrow \mathsf{FE.Setup}(1^{\lambda})$	Return $sk_K \leftarrow \mathsf{FE.KeyGen}(msk, K)$
$b' \leftarrow \mathcal{A}^{\mathsf{KeyGenO}(msk,\cdot),\mathsf{EncO}(mpk,\boxed{\mathrm{msk}},\cdot,\cdot)}(mpk)$	$\mathsf{EncO}(mpk, \boxed{\mathrm{msk}}, M_0, M_1)$:
Return b'.	Return $\mathsf{FE.Enc}(mpk, \boxed{\mathrm{msk}}, M_b)$

challenger $\mathcal{C}$. We define the advantage of $\mathcal{A}$ in game $\mathsf{sINDCPA}^{\mathsf{FE},b}_{\mathcal{A},F}(\lambda)$ as $\mathsf{Adv}^{\mathsf{sINDCPA}}_{\mathsf{FE},F,\mathcal{A}}(\lambda) = \Pr[\mathsf{sINDCPA}^{\mathsf{FE},0}_{\mathcal{A},F}(\lambda) = 1] - \Pr[\mathsf{sINDCPA}^{\mathsf{FE},1}_{\mathcal{A},F}(\lambda) = 1]$.

In the above two experiments, we require that if for all key queries $\{K\}$ and message queries $\{(M_0, M_1)\}$ made by the adversary $\mathcal{A}$, then we have $F(K, M_0) = F(K, M_1)$. Obviously, the adaptive security implies selective security.

4 Construction of functional encryption for quadratic functions

In this section, we present a functional encryption for quadratic functions (QFE) with (adaptive) IND-CPA security.

4.1 Construction

In the following, we present a generic construction for functional encryption for quadratic functions. Let QFE=(QFE.Setup, QFE.KeyGen, QFE.Enc, QFE.Dec) denote the scheme over the functional space $F : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ (see Section 2.2 for its functionalities) and message space $\mathcal{M} = \mathbb{Z}_q^n \times \mathbb{Z}_q^m$, where $\mathcal{K} = \mathbb{Z}_q^{n\times m}$ and $\mathcal{Y} = \mathbb{Z}_q$. In particular, our scheme uses the following building block.

- A functional encryption scheme FE=(FE.Setup, FE.KeyGen, FE.Enc, FE.Dec) for function family $\mathscr{G}$.

The construction is described as follows.

$\mathsf{QFE.Setup}(1^{\lambda}, F)$. On input a security parameter 1^{λ} and a function family F, the setup algorithm first samples $(fmk, fsk) \leftarrow \mathsf{FE.Setup}(1^{\lambda})$ and then sets the master public key and master secret key as $mpk = fmk$ and $msk = fsk$. It finally outputs (mpk, msk).

$\mathsf{QFE.KeyGen}(msk, \mathsf{F})$. On input a master secret key msk and a function key $\mathsf{F} \in \mathcal{K}$, the key generation algorithm first parses $msk = fsk$ and then constructs a circuit $\mathcal{G}[\mathsf{F}] \in \mathscr{G}$ with key F hardwired in it (where the construction of $\mathcal{G}$ is shown in Fig. 3). Then it computes $sk_{\mathcal{G}} \leftarrow \mathsf{FE.KeyGen}(fsk, \mathcal{G})$ and sets $sk_{\mathsf{F}} = sk_{\mathcal{G}}$. Finally, it outputs the private key sk_{F}.

$\mathsf{QFE.Enc}(mpk, \boxed{\mathrm{msk}}, (\mathsf{x}, \mathsf{y}))$. On input a master public key mpk and a message (x, y), this algorithm first parses $mpk = fmk$, and then computes $CT \leftarrow \mathsf{FE.Enc}(fmk, (\mathsf{x}, \mathsf{y}))$ (Note that (x, y) could be seen as a string of length $(n + m)\log q$). Finally, it outputs the ciphertext CT.

$\mathsf{QFE.Dec}(sk_{\mathsf{F}}, CT)$. On input a private key sk_{F} and a ciphertext CT, the algorithm computes $y = \mathsf{FE.Dec}(sk_{\mathsf{F}}, CT)$ and finally outputs y.

Correctness. The correctness of the functional encryption scheme for quadratic polynomial follows by the correctness of the functional encryption for general functionalities F. Namely, for all $(mpk, msk) \leftarrow \mathsf{QFE.KeyGen}(msk, \mathsf{F})$, all $(\mathsf{x}, \mathsf{y}) \in \mathcal{M}$, all $sk_{\mathsf{F}} \leftarrow \mathsf{QFE.KeyGen}(msk, \mathsf{F})$ and $CT \leftarrow \mathsf{QFE.Enc}(mpk, (\mathsf{x}, \mathsf{y}))$, where $sk_{\mathsf{F}} = sk_{\mathcal{G}}$, if the circuit $\mathcal{G}[\mathsf{F}]$ satisfies the functionality in Fig. 3, then

$$\mathsf{QFE.Dec}(sk_{\mathsf{F}}, CT) = \mathsf{FE.Dec}(sk_{\mathcal{G}[\mathsf{F}]}, CT) = \mathsf{x}^{\top}\mathsf{F}\mathsf{y}.$$

4.2 Security

The security of the scheme QFE follows the following theorem.

Theorem 1 *If the functional encryption scheme* FE *for function family* $\mathscr{G}$ *is adaptively (resp. selectively) secure (see definitions 2 and 3 for the details), the functional encryption scheme* QFE *for quadratic function* F *is also adaptively (resp. selectively) secure.*

Proof Adaptive security. We prove the adaptive security of the scheme QFE via two games below, then go to the details of the proof by proving that the two games are computationally indistinguishable.

Fig. 2 Selective IND-CPA Experiment for FE

$\mathsf{sINDCPA}^{\mathsf{FE},b}_{\mathcal{A},F}(\lambda)$	$\mathsf{KeyGenO}(msk, K)$:
$(M_0, M_1) \leftarrow \mathcal{A}(1^{\lambda})$, where $M_0, M_1 \in \mathcal{M}$	Return $sk_K \leftarrow \mathsf{FE.KeyGen}(msk, K)$
$(mpk, msk) \leftarrow \mathsf{FE.Setup}(1^{\lambda})$	
$CT^* \leftarrow \mathsf{EncO}(mpk, \boxed{\mathrm{msk}}, M_0, M_1)$	$\mathsf{EncO}(mpk, \boxed{\mathrm{msk}}, M_0, M_1)$:
$b' \leftarrow \mathcal{A}^{\mathsf{KeyGenO}(msk,\cdot)}(mpk, CT^*)$	Return $\mathsf{FE.Enc}(mpk, \boxed{\mathrm{msk}}, M_b)$
Return b'.	

Fig. 3 Circuit $\mathcal{G}[\mathsf{F}]$

```
Constants : $\mathsf{F}$
Input : $(\mathsf{x}, \mathsf{y})$
1. Compute $y = \mathsf{x}^\top \mathsf{F}\mathsf{y}$. \\ equivalent to computing $y = \sum_{i\in[n]j\in[m]}(f_{i,j}x_iy_j)$
2. Output $y$.
```

G_0 : This game is the original adaptive security game of the scheme QFE except that the challenge ciphertext encrypts the message $(\mathsf{x}_0, \mathsf{y}_0)$.

G_1 : This game is the same as G_0 except that the challenge ciphertext encrypts the message $(\mathsf{x}_1, \mathsf{y}_1)$. □

If there exists an adversary $\mathcal{A}_{\mathsf{QFE}}$ that can break the adaptive security of the scheme QFE, then there exists an adversary $\mathcal{A}_{\mathsf{FE}}$ that can break the adaptive security of the scheme FE. In the following, we use the adversary $\mathcal{A}_{\mathsf{QFE}}$ to construct the adversary $\mathcal{A}_{\mathsf{FE}}$. Let $\mathcal{B}$ be the challenger of the scheme FE. Assume that before proceeding the reduction, the adversary $\mathcal{A}_{\mathsf{FE}}$ has received the master public key fpk from its challenger.

Setup phase In this phase, the adversary $\mathcal{A}_{\mathsf{FE}}$ first sets the master public key $mpk = fpk$ for the scheme QFE, then it sends mpk to $\mathcal{A}_{\mathsf{QFE}}$.

Key query phase When the adversary $\mathcal{A}_{\mathsf{QFE}}$ makes a key query of the form $\mathsf{F} \in \mathcal{K}$, the adversary $\mathcal{A}_{\mathsf{FE}}$ first constructs a circuit $\mathcal{G}[\mathsf{F}] \in \mathscr{G}$ with F hardwired in it. Then it delivers $\mathcal{G}[\mathsf{F}]$ to its challenger $\mathcal{B}$ and from which it gets the private key $sk_{\mathcal{G}[\mathsf{F}]}$. Finally, it sets the private key as $sk_{\mathsf{F}} = sk_{\mathcal{G}[\mathsf{F}]}$ and sends sk_{F} to the adversary $\mathcal{A}_{\mathsf{QFE}}$.

Challenge phase When $\mathcal{A}_{\mathsf{QFE}}$ makes a challenge query $((\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1)) \in \mathcal{M}$ such that $(\mathsf{x}_0)^\top \mathsf{F}\mathsf{y}_0 = (\mathsf{x}_1)^\top \mathsf{F}\mathsf{y}_1$ for all queries F that the adversary makes in the key query phase before the challenge phase, the adversary $\mathcal{A}_{\mathsf{FE}}$ delivers the pair $((\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1))$ to its challenger, from which it gets the challenge ciphertext CT^* (which is generated by $CT^* \leftarrow \mathsf{FE.Enc}(fpk, (\mathsf{x}_b, \mathsf{y}_b))$, where $b \in \{0, 1\}$ is chosen randomly by the challenger $\mathcal{B}$). Finally, the adversary $\mathcal{A}_{\mathsf{FE}}$ sends CT^* to $\mathcal{A}_{\mathsf{QFE}}$.

Key query phase The adversary $\mathcal{A}_{\mathsf{QFE}}$ makes more private key queries of the form $\mathsf{F} \in \mathcal{K}$ as above but with the restriction $(\mathsf{x}_0)^\top \mathsf{F}\mathsf{y}_0 = (\mathsf{x}_1)^\top \mathsf{F}\mathsf{y}_1$ for all $((\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1))$ that $\mathcal{A}_{\mathsf{QFE}}$ queries in the challenge phase.

Guess phase $\mathcal{A}_{\mathsf{QFE}}$ eventually outputs a bit $b' \in \{0, 1\}$, and the experiment outputs the same bit.

For any stateful adversary $\mathcal{A}_{\mathsf{QFE}}$, if the challenger $\mathcal{B}$ encrypts the message $(\mathsf{x}_0, \mathsf{y}_0)$, the adversary $\mathcal{A}_{\mathsf{FE}}$ perfectly simulates the game G_0 for $\mathcal{A}_{\mathsf{QFE}}$; when the challenger $\mathcal{B}$ encrypts the message $(\mathsf{x}_1, \mathsf{y}_1)$, the adversary $\mathcal{A}_{\mathsf{FE}}$ perfectly simulates the game G_1 for $\mathcal{A}_{\mathsf{QFE}}$. Therefore, by Definition 2, the advantage that the adversary $\mathcal{A}_{\mathsf{QFE}}$ distinguishes games G_0 and G_1 (i.e., the adversary $\mathcal{A}_{\mathsf{QFE}}$ breaks the adaptive security of the scheme QFE) is equal to the advantage that the adversary $\mathcal{A}_{\mathsf{FE}}$ breaks the adaptive security of the scheme FE. Thus the theorem follows.

Selective security. The proof of the selective security of the scheme QFE is similar to that of the adaptive security but with the difference that the reduction relies on the selective security of the scheme FE where the challenge messages must be decided before the setup and key generation. For simplicity, we omit the details about the proof.

5 Instantiations

In this section, we describe how to instantiate the underlying functional encryption scheme FE for generic functions used to construct our resulting FE schemes for quadratic functions (see Section 4.1).

The underlying FE schemes for generic functions required by our FE schemes for quadratic functions can be built from a wide range of assumptions. For instance, they can be instantiated under the standard assumptions such as decisional diffie-hellman (DDH) [6], RSA [7], learning with errors (LWE) [25], and learning parity with noise (LPN) [35], and the non-standard assumptions such as intractable problems on composite order multilinear maps [9] and indistinguishability obfuscator (IO) [29]. Particularly, these instantiable FE schemes under the standard assumptions can be taken from the general-purpose public-key FE schemes proposed by Gorbunov et al. in [10]. As their schemes are based on the existence of semantically-secure public-key encryption (PKE) and pseudorandom generators (PRG). To our knowledge, there are already many PKE schemes proposed based on various standard assumptions (DDH [6], RSA [7], LWE [25], and LPN [35]). Therefore, our resulting schemes can also be instantiated from these assumptions. For those FE schemes under IO and multilinear maps, their constructions can be taken directly from [29] and [9] respectively. In particular, if these underlying FEs are selectively secure, we can use these techniques proposed by Ananth et al. in [2] (i.e., generic transformations from selective security to adaptive security for FE), to convert them into adaptive ones. Furthermore, from this constructions, we can see that the underlying FE schemes that satisfy lightweight circuit is ok (due to the lightweight computation $\mathsf{x}^\top \mathsf{F}\mathsf{y}$ only existed in the circuit $\mathcal{G}[\mathsf{F}]$).

Table 1 Comparisons with [4]

Schemes	# of SK	# of PK
SK-FE in SXDH [4]	$2\log q$	$\|\mathsf{bgp}\|$
PK-FE in SXDH [4]	$2\log q$	$2(n+m)\log q$
PK-FE in GGM [4]	$(nm+2)\log q$	$\|\mathsf{bgp}\| + (n+m+1)\log q$
SK-FE in our paper	|SK| of FE	|PK| of FE
PK-FE in our paper	|SK| of FE	|PK| of FE

6 Comparisons

In this section, we give the performance analysis and comparison between our FE scheme and that by Baltico et al. in [4] in terms of private key size, master public key size, ciphertext size, and etc. in Tables 1 and 2. From the two Tables, we can see that the size of public key and private key in our schemes (both in secret-key settings and public-key settings) has flexible length changed with the varied constructions of the underlying FEs for generic functions used in our schemes. The ciphertexts in our both schemes have comparable size with those in [4]. In particular, different from [4], where the computations of the quadratic functions must rely on the bilinear map, while ours do not. Moreover, beyond the MDDH and GGM assumptions, our schemes can also be instantiated from other assumptions such as the DDH, RSA, LWE and LPN assumptions, while Baltico et al.'s schemes [4] can only be instantiated by SXDH and GGM. Furthermore, all our schemes are provably secure against adaptive adversaries, while the schemes of Baltico et al.'s [4] from the MDDH assumption are only proved selectively secure, and under the same assumption, their encryption scheme in the secret-key setting is a deterministic encryption.

Notations in Tables 1 and 2. $n, m \in \mathbb{N}^+$, $k \in \mathbb{N}^*$; $\log q$: size of an element in group; "N": NO; "Y": YES; "# of SK": size of private key; "# of PK": size of master public key; "# of CT": size of ciphertext; "BP": the number of bilinear pairings needed in decryption algorithm; "R/D": deterministic encryption or randomized encryption; "security": selective security or adaptive security ; "|SK| of FE": size of private key in FE for circuit; "|PK| of FE": size of master public key in FE for circuit; "|bgp|": description of bilinear group setting, where $\mathsf{bgp} = (q, G_1, G_2, G_T, e, g_1, g_2)$.

Table 2 Comparisons with [4]

Schemes	# of CT	security	BP	R/D
SK-FE in SXDH [4]	$2(n+m)\log q$	selective	mn+2	D
PK-FE in SXDH [4]	$(6n+6m+2)\log q$	selective	2mn+2	R
PK-FE in GGM [4]	$2(n+m+1)\log q$	adaptive	2	R
SK-FE in our paper	$O((m+n)\log q)$	adaptive	0	R
PK-FE in our paper	$O((m+n)\log q)$	adaptive	0	R

7 Conclusions

In this paper, we present a simple framework that transforms generic functions to quadratic functions for functional encryption and provided a concrete scheme, which can be utilized to construct efficient privacy-preserving machine learning algorithms. The proposed scheme is built on the proposed framework, could be constructed based on the standard assumptions (such as DDH, LWE, LPN etc.) and be proved to achieve adaptive security. Compared with existing schemes, the proposed scheme is more scalable and provides a stronger security guarantee. Moreover, the future is that we intend to propose the functional encryption for randomized functionalities with application to machine learning under standard assumptions.

Acknowledgments The first author is supported by the National Key Research and Development Program of China (Grant No. 2017YFB0802000) and the National Natural Science Foundation of China (Grant Nos. NSFC61702007, NSFC61572318) and Other Foundations (Grant Nos. 2019M661360, gxbjZD27,KJ2018A0533, XWWD201801, ahnis20178002, KJ2017A519, 16ZB0140). The second author is supported by the National Key Research and Development Program of China (Grant No. 2017YFB0802000) and the National Natural Science Foundation of China (Grant No. U1705264). The fourth author is supported in part by National Key Research and Development Program of China (Grant No. 2017YFB0802000), the National Natural Science Foundation of China (Grant Nos. 61877011, 61472084, U1536205), Shanghai Innovation Action Project (Grant No. 16DZ1100200), Shanghai Science and Technology Development Funds (Grant No. 16JC1400801), and Shandong Provincial Key Research and Development Program of China (Grant Nos. 2017CXG0701, 2018CXGC0701).

References

1. Agrawal S, Kumarasubramanian A, Prabhakaran M, Sahai A (2015) On the practical security of inner product functional encryption. In: Katz J. (ed) Advance in PKC 2015, vol 9020. Springer, Berlin Heidelberg, pp 777–798
2. Ananth P, Brakerski Z, Segev G, Vaikuntanathan V (2015) From selective to adaptive security in functional encryption. In: Gennaro R, Robshaw M (eds) Advance in CRYPTO, vol 2015, pp 657–677
3. Ananth P, Sahai A (2017) Projective Arithmetic Functional Encryption and Indistinguishability Obfuscation from Degree-5 Multilinear Maps. In: Coron JS., Nielsen J (eds) Advances in EUROCRYPT 2017, vol 10210, Springer, Berlin Heidelberg. pp 152–181
4. Baltico CEZ, Catalano D, Fiore D, Gay R (2017) Practical functional encryption for quadratic functions with applications to predicate encryption. In CRYPTO 2017:67–98
5. Bellare M, Rogaway P (2006) The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In:

Vaudenay S (ed) EUROCRYPT 2006. vol 4004. Springer, Heidelberg, pp 409–426
6. Boneh D (1998) The decision di e-hellman problem. In: proceedings of the 3rd Algorithmic Number Theory Symposium volume 1423, pages 48–63 Lecture Notes in Computer Science
7. Boneh D (1999) Twenty years of attacks on the rsa cryptosystem. In: Notices of the American Mathematical Society, pp 203–213
8. Ehsan H, Hassan T, Mehdi G (2017) Cryptodl: Deep neural networks over encrypted data. arXiv:1711.05189
9. Garg S, Gentry C, Halevi S, Zhandry M (2016) Functional encryption without obfuscation, In TCC2016, 480–511
10. Gorbunov S, Vaikuntanathan V, Wee H (2012) Functional encryption with bounded collusions via multi-party computation. In: Reihaneh Safavi-Naini, Ran Canetti (eds) editors, Advances in Cryptology CRYPTO 2012, vol 7417. Springer, Berlin Heidelberg, pp 162–179
11. Graepel T, Kristin L, Michael N (2012) Ml confidential: Machine learning on encrypted data. In: International Conference on Information Security and Cryptology, pages 1–21. Springer
12. Wang H, Chen K, Qin B et al (2018) LR-RRA-CCA secure functional encryption for randomized functionalities from trapdoor HPS and LAF. Sci China Inf Sci 61:058101. https://doi.org/10.1007/s11432-017-9120-4
13. Hao M, Li H, Luo X, Xu G, Yang H, Liu S (2019) Efficient and privacy-enhanced federated learning for industrial artificial intelligence 1–1. IEEE Trans Industrial Inform. to appear, https://doi.org/10.1109/ TII.2019.2945367
14. Jiang W, Li H, Xu G, Wen M, Dong G, Lin X (2019) Ptas: Privacy-preserving thin-client authentication scheme in blockchain-based pki. Future Generation Comput Syst 96:185–195
15. Jiang XQ, Kim M, Lauter K, Song YS (2018) Secure outsourced matrix computation, and Application to neural networks. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications security pages 1209–1222
16. Keith B, Vladimir I, Ben K, Antonio M, Brendan MH, Sarvar P, Daniel R, Aaron S, Karn S (2017) Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp 1175–1191
17. Li H, Liu D, Dai Y, Luan TH, Yu S (2018) Personalized search over encrypted data with efficient and secure updates in mobile clouds. IEEE Trans Emerg Topic Comput 6(1):97–109
18. Li H, Yang Y, Dai Y, Yu S, Xiang Y Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data, pages 1–1, 2017, to appear. https://doi.org/10.1109/TCC.2017.2769645 IEEE Transactions on Cloud Computing
19. Li HW, Yang Y, Dai YS, Bai J, Yu S, Xiang Y (2017) Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data IEEE Transactions on Cloud Computing accepted
20. Lin H (2016) Indistinguishability obfuscation from ddh on 5-linear maps and locality-5 prgs, In Cryptology ePrint Archive, Report 2016/1096., 2016. http://eprint.iacr.org/2016/1096
21. Miao YB, Liu XM, Choo KKR, Deng H, Li JG, Li HW, Ma JF (2019) Privacy-preserving attribute-based keyword search in shared multi-owner setting. IEEE Trans Dependable Sec Comput Accepted. https://doi.org/10.1109/TDSC.2019.2897675
22. Mohassel P, Zhang YP (2017) Secureml: a system for scalable privacy-preserving machine learning. In: In 2017 IEEE Symposium on Security and Privacy (S&P), pp 19–38
23. Ran G, Nathan D, Kim L, Kristin L, Michael N, John W (2016) Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning, pp 201–210
24. Ren H, Li HW, Dai YS, Yang K, Lin XD (2018) Querying in internet of things with privacy preserving: Challenges, solutions and opportunities. IEEE Network 32(6):144–151
25. Sun XC, Li B, Lu XH, Fang FY (2015) Cca secure public key encryption scheme based on lwe without gaussian sampling. In Inscrypt 2015:361–378
26. Shacham H, Ristenpart T, Shrimpton T (2011) Careful with composition: Limitations of the indiferentiability framework. In: Paterson KG (ed) EUROCRYPT 2011, volume 6632 of LNCS, pages 487–506 pringer
27. Wang HG, Chen KF, Joseph KL, Hu ZY (2018) Leakage-resilient chosen-ciphertext secure functional encryption from garbled circuits, In ISPEC2018, 119–140
28. Wang HG, Zhang Y, Chen K, Sui GY, Zhao YL, huang XY (2019) Functional broadcast encryption with applications to data sharing for cloud storage Information Sciences
29. Waters B (2015) A punctured programming approach to adaptively secure functional encryption. In Advances in CRYPTO 2015:678–697
30. Li GWXHW, Liu S, Yang K, Lin XD (2019) Verifynet: Secure and verifiable federated learning. IEEE Trans Inform Forensics Secur Accepted. https://doi.org/10.1109/TIFS.2019.2929409
31. Xu G, Li H, Liu S, Wen M, Lu R (2019) Efficient and privacy-preserving truth discovery in mobile crowd sensing systems. IEEE Trans Vehicular Technol 68(4):3854–3865
32. Xu G, Li H, Ren H, Yang K, Deng RH (2019) Data security issues in deep learning: Attacks, countermeasures and opportunities. IEEE Commun Mag 57(11):116–122
33. Xu GW, Li HW, Dai YS, Yang K, Lin XD (2018) Enabling efficient and geometric range query with access control over encrypted spatial data. IEEE Trans Inform Forensics Secur 14(4):870–885
34. Xu RH, James JB, Lin C (2019) Cryptonn: Training neural networks over encrypted data. arXiv:1904.07303
35. Yu Y, Zhang J (2016) Cryptography with auxiliary input and trapdoor from constant-noise lpn. In CRYPTO 2016:214–243
36. Zhang Y, Xu CX, Li HW, Yang K, Zhou JY, Lin XD (2018) Healthdep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems. IEEE Trans Industrial Inform 14(9):4101–4112
37. Zhang Y, Xu CX, Lin XD, Shen XM (2019) Blockchain-based public integrity verification for cloud storage against procrastinating auditors. IEEE Transactions on Cloud Computing accepted
38. Zhang Y, Xu CX, Ni JB, Li HW, Shen XM (2019) Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2019.2923222

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

Affiliations

Huige Wang[1,2] · Kefei Chen[3,4] · Yuan Zhang[5] · Yunlei Zhao[2]

Kefei Chen
kfchen@hznu.edu.cn

Yuan Zhang
zy_loye@126.com

Yunlei Zhao
ylzhao@fudan.edu.cn

[1] Department of Computer, Anhui Science and Technology University, Bengbu, 233030, China

[2] School of Computer Science, Fudan University, Shanghai, 200433, China

[3] Department of Mathematics, Hangzhou Normal University, Hangzhou, 311121, China

[4] Westone Cryptologic Research Center, Beijing 100070, China

[5] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China