A Multivariate Public Key Encryption Scheme With Equality Test

XIAOYING SHEN, LICHENG WANG, HUIJUN ZHU, AND YUAN LIU

1State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China
2Nanyang Institute of Technology, Nanyang 473004, China

Corresponding author: Huijun Zhu (zhuhj1201@163.com)

This work was supported in part by the Shandong Provincial Key Research and Development Program of China under Grant 2018CXGC0701, in part by the National Natural Science Foundation of China (NSFC) under Grant 61972050, and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2019233 and Grant CX2019119.

ABSTRACT The public key encryption with equality test (PKEET) allows the cloud server to judge whether two different ciphertexts are generated by the same message without decryption. Through this technique, PKEET provides an effective solution for building secure outsourced databases, and has made some rich achievements. This paper combines multivariate public key encryption and equality test, and proposes the first multivariate public key encryption scheme with equality test (MPKEET), which inherits the advantages of both primitives. Moreover, the equality test algorithm proposed in this paper is based on a straight line. Compared with the schemes based on bilinear pairing, it is simpler and easier to implement. And our MPKEET scheme achieves desirable security, which can resist linearization equation attacks, differential attacks, XL attacks, Gröbner basis attacks and the attack of quantum computer, when appropriate parameters are selected.

INDEX TERMS Equality test, multivariate, public key encryption, security.

I. INTRODUCTION

As an extension of the classical public key encryption scheme, the public key encryption with equality test was proposed by Yang et al. [1], which can be used to judge whether the plaintexts encrypted by two different public keys are the same. To some extent, the proposed technology solves the problem of computing encrypted data of multiple public keys, enriches the retrieval method of the encrypted data in the cloud environment, and provides a broader application scenario for public key encryption.

Through this technology, some privacy preserving services for outsourced data could be achieved. Therefore, PKEET has been widely used in real applications since it was proposed. For example, in data archiving, the same data can be tagged for archiving by using ciphertext equality test technology. In a medical system, patients with the same disease can be screened out by using ciphertext equality test technology, and the most appropriate treatment for the disease can be selected by comparing treatment means with a curative effect. If an encrypted database and ciphertext equality test technology can be combined, cross table query of the encrypted database can be achieved.

Because of its wide applicability in practice, various PKEET constructions have been proposed [1]–[13], but their securities mostly depend on discrete logarithm problem and bilinear pairing, which can be solved in polynomial time by efficient Shor’s quantum algorithms [14]. Therefore, these public key cryptography algorithms are no longer secure in the era of quantum computing, the scheme proposed in this paper overcomes this shortcoming, and can resist the attack of quantum algorithm.

As one of the promising candidate scheme that can effectively resist the attack of quantum computer, the multivariate public key cryptosystem (MPKC) has attracted increasing attention in the academic world. In addition, because the MPKC schemes mostly operate on a small finite field, it is more efficient than traditional encryption schemes based on number theory, like RSA and ECC [15], which are well suited to resource-constrained computing devices, such as wireless sensor networks, radio frequency identification devices (RFID) and smart cards. Hence, this paper introduces the equality test into the multivariate public key encryption scheme for the first time, and proposes a novel multivariate...
public key encryption scheme with equality test (MPKEET), aiming to expand the application scenario of multivariate cryptography.

A. OUR CONTRIBUTIONS
To provide search functionality on multivariate public key encryption scheme, the construction of multivariate public key encryption with equality test has been proposed in this paper. Our main idea is to provide the additional function of the equality test to Ding et al.’s PMI+ scheme [16], which is considered secure at present [17]. Our contributions are summarized as follows.

- We introduce the idea of PKEET into the multivariate public key encryption scheme to enjoy the best-of-the-two-worlds for the first time. The proposed scheme integrates the advantages of these two cryptographic primitives, and fills in the gap of multivariate public key encryption algorithm in the field of equality test. The core work of this paper is to add the equality test algorithm to the classical multivariate public key encryption scheme, so that it can perform equality test on the ciphertexts encrypted by different public keys and ciphertexts encrypted by the same public key. This design enriches the search function of encrypted data.
- The encryption algorithm proposed in this paper operates on a small finite field, so it is more efficient than cryptography algorithms based on number theory. Compared with some existing public key encryption schemes with equality test, the encryption time of our scheme has been reduced by at least 94%.
- The equality test algorithm proposed in this paper is based on a straight line. This design is simpler and easier to implement. Compared with the schemes based on bilinear pairing in [4]–[6], [10] and [11], the time costs of our scheme has been reduced by at least 16.7%.
- Our MPKEET scheme achieves desirable security, and resists linearization equation attacks, differential attacks, XL attacks, Gröbner basis attacks and quantum computer attacks, when appropriate parameters are selected.

B. ORGANIZATION
The remainder of this paper is organized as follows: We discuss related work in section 2. The preliminaries and the system model are presented in section 3 and section 4, respectively. Then, a multivariate public key encryption scheme with equality test is proposed in section 5. In section 6, we give the detailed performance evaluations of our scheme. In section 7, we analyze the security of our proposed MPKEET. Finally, we conclude our work in the last section.

II. RELATED WORK
The development of big data has greatly facilitated people’s life. However, big data also faces many security risks in the collection, storage, and use of data. In response to these challenges, many valuable encryption schemes [18]–[21] have been proposed. Using these encryption algorithms, data is encrypted and then sent to the cloud for storage. When a user needs to find a file that contains a keyword, he should first download all ciphertexts data to local for decryption, and then keyword searches are performed on plaintext, but this operation not only wastes a lot of network overhead and storage overhead due to a lot of data that is not needed, but also requires users to pay a lot of computational overhead for decryption and search operations. Therefore, some schemes that allow users to retrieve ciphertext data through keywords are proposed.

Two related cryptographic primitives have been proposed: Public key encryption with keyword search (PEKS) [22] and public key encryption with equality test [1].

A. PKES
Public key searchable encryption (PKSE) is a kind of ciphertext searchable encryption system. Under the premise of protecting data confidentiality, PKES allows users to search ciphertext data that contains certain specific keyword. In 2004, Boneh et al. [22] proposed the concept of public key searchable encryption for the first time by using public key encryption technology and bilinear mapping. This construction solved the difficult problem of retrieving encrypted data in a specific environment. However, its communication mode is one-to-one, and keyword ciphertext can only be queried or decrypted by a specific user, which has many limitations in practice. Recently, Mamta et al. proposed attribute-based searchable encryption [23], [24] and IBE-based searchable encryption schemes [25]. These structures realize one-to-many communication, so that keyword ciphertext can be retrieved by multiple users, saving network storage space, and improving retrieval efficiency.

B. PKEET
The notion of public key encryption with equality test (PKEET) was firstly introduced by Yang et al. [1] as a new variant of searchable encryption. In 2011, Tang [2] proposed a public key encryption scheme with equality test to support fine-grained authorization, and introduced an authorization mechanism for the first time. Two users with their own public/private key pairs can issue tokens to an agent to authorize it to perform equality tests between their ciphertexts. But in this system, two users must come together to generate a token to allow agents to compare their ciphertexts. Although it provides users with strict control over who can compare their ciphertexts, in some situations, it may be a burden. Tang [3] proposed an all or nothing public key cryptosystem with equality test, which supported user-level authorization by specifying who can independently equality test between two ciphertexts. Subsequently, Tang [4] extended the fine-grained authorized public key encryption precaution with the equality test proposed in [2] to two-proxy
settings, which protected against the inevitable offline message recovery attacks.

Later, Ma et al. [5] proposed a public key encryption scheme with delegated equality test. This primitive allows only authorized parties to verify the equality of ciphertexts in scenarios involving multiple users, but it cannot work effectively due to its large number of bilinear mapping operations. Huang et al. [6] proposed a public key encryption scheme with authorization equality test, which enhanced privacy protection through receiver’s warrants and cipher-warrants. Lee et al. [7] proposed a polynomial time CCA2 attack on Huang et al.’s scheme and gave a solution.

In 2015, Ma et al. [8] proposed the concept of public key cryptography with equality test supporting flexible authorization, which supports four different types of authorization mechanisms. However, this work is based on bilinear pairing, which is a time-consuming operation compared with modular exponents. Lin et al. [9] proposed a public key encryption scheme with equality test, which supports flexible authorization, but does not require a bilinear pairing operation.

In 2016, Ma [10] proposed the first identity-based equality test encryption scheme. This allows equality tests to be carried out for one or more user’s ciphertexts, but the scheme uses too many hash-to-point operations and bilinear mapping operations, so it is inefficient and unsuitable for deployment on lightweight devices. In 2017, Wu et al. [11] proposed an efficient equality test encryption scheme based on bilinear pairings, which reduced the need for time-consuming hash-to-point functions.

In 2017, Wang et al. [12] combined a attribute-based encryption of ciphertext strategy with ciphertext equality test, proposed ciphertext policy attribute-based encryption with equality test, and proved the security of this scheme under the standard model. Zhu et al. [13] introduced attribute-based encryption into public key encryption with equality test, and proposed a new cloud storage scheme. This scheme supports fine-grained authorization, and can detect whether ciphertexts encrypted by different public keys contains the same plaintext information. However, as far as we know, there is a lack of an efficient MPKEET scheme which supports equality test without decryption. Achieving a secure MPKEET scheme is still an open problem. In this paper we add the equality text algorithm into an existing public key encryption scheme, so that it has more extensive application in practice.

### C. THE DIFFERENCE BETWEEN PKES AND PKEET

All of the schemes about PKES are usually for keyword searches, without the ability to directly compare ciphertext, and they have a single function and limited application scenarios. However, the public key encryption schemes with equality test can judge whether two ciphertexts encrypted by different users are generated by the same plaintext. Therefore, PKEET can be used for PKES, but PKES cannot be used for PKEET.

### III. PRELIMINARIES

In this section, we provide some basic concepts which will be utilized in our construction. For the convenience of readers, we summarize some symbols used in our scheme in Table 1.

### A. MULTIVARIATE QUADRATIC POLYNOMIAL SYSTEM

**Definition 1:** [26] A multivariate quadratic polynomial system is a set of nonlinear equations as follows:

$$
\begin{align*}
\mathbf{p}^{(1)} &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(1)} x_i x_j + \sum_{i=1}^{n} p_i^{(1)} x_i + p_0^{(1)}, \\
\mathbf{p}^{(2)} &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(2)} x_i x_j + \sum_{i=1}^{n} p_i^{(2)} x_i + p_0^{(2)}, \\
&\vdots \\
\mathbf{p}^{(l)} &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(l)} x_i x_j + \sum_{i=1}^{n} p_i^{(l)} x_i + p_0^{(l)}.
\end{align*}
$$

### TABLE 1. Symbols used in our MPKEET scheme.

| Symbol | Description |
|--------|-------------|
| n      | the number of variables |
| \(p^{(i)}\) | quadratic polynomials |
| \(p_{ij}\) | the coefficient of the quadratic term |
| \(p_i\) | the coefficient of the first order term |
| \(p_0\) | the constant term |
| l      | the number of equations |
| \(\tilde{Q}\) | the central map of PMI+ |
| \(\hat{Q}\) | a univariate mapping on \(\mathbb{R}\) |
| \(K\) | a finite field of order \(q\) and characteristic 2 |
| \(K^n\) | \(n\)-dimensional affine space |
| \(k\) | a security parameter |
| \(Z\) | set of integers |
| \(S, S'\) | two reversible affine transforms in the field \(K^n\) |
| \(T, T'\) | two reversible affine transforms in the field \(K^{n-a}\) |
| \(m\) | the plaintext |
| \(c\) | the ciphertext |
| \(M\) | the message space |
| \(g(z)\) | an irreducible polynomial of degree \(n\) |
| \(\theta\) | a nonnegative integer |
| \(\mathcal{R}\) | an extension field of degree \(n\) of \(K\) |
| \(r\) | the dimension of the internal perturbation |
| \(\varphi\) | an isomorphic mapping from \(\mathcal{R}\) to the vector space \(K^n\) |
| \(\tilde{Q}^{-1}\) | the inverse of \(\hat{Q}\) |
| \(z_i\) | a linear equation |
| \(Q^*\) | an internal perturbation mapping |
| \(P\) | a perturbation set |
| \(Q^+\) | an external perturbation mapping |
| \(Q_1, Q_2\) | the public key polynomial |
| \(H\) | a hash function |
Definition 2: (MQ Problem) [27] Given $l$ multivariate quadratic polynomials $p^{(i)}(x_1, \ldots, x_n), \ldots, p^{(i)}(x_1, \ldots, x_n)$ as shown in Eq. (1), find a vector $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ such that $p^{(i)}(\tilde{x}_1, \ldots, \tilde{x}_n) = 0$, for $\forall 1 \leq i \leq l$.

Remark 1: The MQ problem is proven to be NP hard [27] (even if all the equations are quadratic and the field is $\mathbb{F}_2$).

That is, this processes is unidirectional, and the inverse solution is not feasible.

B. MULTIVARIATE PUBLIC KEY CRYPTO SYSTEM

A multivariate public key cryptosystem (MPKC) is a public key cryptosystem whose public key is a set of multivariate quadratic polynomials. To build a MPKC based on the MQ problem, First, a central map $\tilde{Q} : \mathbb{K}^n \rightarrow \mathbb{K}^l$ is defined, and second, the central map is hidden by two invertible affine maps $S : \mathbb{K}^l \rightarrow \mathbb{K}^l$ and $T : \mathbb{K}^n \rightarrow \mathbb{K}^n$. The public key of the scheme is the composed map $Q = S \circ \tilde{Q} \circ T$. The private key consists of the two maps $S$ and $T$. Encryption only needs to substitute the plaintext $m$ into the public key polynomial $Q$ to evaluate the ciphertext $c = Q(m)$. The decryption process requires computing $m = T^{-1} \circ \tilde{Q}^{-1} \circ S^{-1}(c)$.

C. PERTURBED MATSUMOTO-IMAI-Plus (PMI+)

The PMI+ encryption system is a secure variant of the MI system. By adding an internal disturbance to the central mapping equations of the MI system to defeat linearized equation attack due to the special structure of the central mapping, and then adding an external disturbance to resist differential attacks, which also increases the complexity of attacks.

1) KEY GENERATION

Let $\mathbb{K}$ be a finite field of order $q$ with characteristic 2, $g(x) \in \mathbb{K}[x]$ be an irreducible polynomial of degree $n$ ($n > 0$), $\mathbb{K} = \mathbb{K}[x]/g(x)$ be an extension field of degree $n$ of $\mathbb{K}$, $\theta$ ($\theta < n$) is a nonnegative integer, and satisfies $\text{gcd}(g^\theta + 1, q^n - 1) = 1$. $r$ is a small random integer. Define $\varphi$ as $k$-linear mapping from $\mathbb{K}$ to the vector space $\mathbb{K}^n$:

$$\varphi(a_0 + a_1x + \ldots + a_{n-1}x^{n-1}) = (a_0, a_1, \ldots, a_{n-1}).$$

Define a polynomial map: Define a map $\tilde{Q} : \mathbb{K} \rightarrow \mathbb{K}^l$ by $\tilde{Q}(x) = x^l + \hat{q}$, and obtain the mapping on the vector space $\mathbb{K}^n : \tilde{Q}(x_1, x_2, \ldots, x_n) = \varphi \circ \tilde{Q} \circ \varphi^{-1}(x_1, x_2, \ldots, x_n)$.

- Add internal perturbation: Fix a small integer $r$ and define a system of linear equations

$$\begin{align*}
z_1(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \alpha_{i1}x_i + \beta_1, \\
z_2(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \alpha_{i2}x_i + \beta_2, \\
&\quad \vdots \\
z_r(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \alpha_{ir}x_i + \beta_r, \\
\end{align*}$$

where $z_i - \beta_i$ is linearly independent. Define a map $Z : \mathbb{K}^n \rightarrow \mathbb{K}^r$ by $Z(x_1, x_2, \ldots, x_n) = (z_1, z_2, \ldots, z_r)$. Randomly select $n$ quadratic polynomials $\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n$ with $z_1, z_2, \ldots, z_r$, and define a map $Q : \mathbb{K}^r \rightarrow \mathbb{K}^n$ by $Q(z_1, z_2, \ldots, z_r) = (\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_n)$. Then the internal disturbance map is $Q^*(x_1, x_2, \ldots, x_n) = \hat{Q} \circ Z(x_1, x_2, \ldots, x_n)$.

We add an internal perturbation into the map $Q^*$, that is, $\tilde{Q}(x_1, x_2, \ldots, x_n) = (Q^* + Q^*)(x_1, x_2, \ldots, x_n)$.

- Add external perturbation: Randomly select a multivariate polynomial $q_1, q_2, \ldots, q_d$ on $x_1, x_2, \ldots, x_n$ and define the external perturbation map $Q^* : \mathbb{K}^n \rightarrow \mathbb{K}^n$ as $Q^*(x_1, \ldots, x_n) = (q_1(x_1, \ldots, x_n), \ldots, q_d(x_1, \ldots, x_n))$.

An external perturbation is added to the map $\tilde{Q}$, namely, $Q = Q^* \parallel Q^+$, where $\parallel$ is the connection between the two parts.

From above, the central map of PMI+ is $\tilde{Q} = (Q^* + Q^+)$ $\parallel Q^+$, randomly choose two invertible affine map $T$ on $\mathbb{K}^{n+a}$ and $S$ on $\mathbb{K}^n$. The public key is $Q = T \circ \tilde{Q} \circ S$, and the secret key is $(\theta, \hat{Q}, Z, Q^+, T, S)$.

2) ENCRYPTION

For a given plaintext $m = (m_1, m_2, \ldots, m_n)$, the ciphertext can be obtained only by substituting $m$ into the public key polynomials $Q$.

$$c_1, c_2, \ldots, c_{n+a} = Q(m_1, m_2, \ldots, m_n)$$

3) DECRYPTION

Decryption of the ciphertext requires calculating

$$(m_1, m_2, \ldots, m_n) = S^{-1} \circ T^{-1}(c_1, c_2, \ldots, c_{n+a}).$$

D. PUBLIC KEY ENCRYPTION WITH EQUALITY TEST

A scheme of public key encryption with equality test [2] consists of the following six algorithms:

- Setup($k$): This algorithm takes a security parameter $k$ as input, and outputs the system parameter $pp$.
- KeyGen($pp$): This algorithm takes the system parameter $pp$ as input, and outputs the receiver’s public/private key pair ($pk$, $sk$).
- Enc($pk$, $m$): This algorithm takes a receiver’s public key $pk$ and a message $m$ as input, and outputs $c$ as the ciphertext.
- Dec($sk$, $c$): This algorithm takes a receiver’s private key $sk$ and a ciphertext $c$ as input, and outputs the corresponding message $m$.
- Auth($sk$): This algorithm takes a receiver’s private key $sk$ as input, and outputs the trapdoor $td$.
- Test($CA$, $tdA$, $CB$, $tdB$): Let $CA$ and $tdA$ be A’s ciphertext and trapdoor, respectively, and let $CB$ and $tdB$ be B’s ciphertext and trapdoor, respectively. This algorithm takes two (ciphertext, trapdoor) pairs ($CA$, $tdA$) and ($CB$, $tdB$) as input, and then
are encrypted by the encryption key to form the ciphertext

six algorithms:

the cloud server. The cloud server can only perform equality

obtain the encryption key, dataset

and Bob through the secure channel. After Alice and Bob

sk

be a message space,

parameters and generating the encryption key pkA, pkB and
decryption key skA, skB, and then sending them to the Alice
and Bob through the secure channel. After Alice and Bob
obtain the encryption key, dataset FA and FB which they own,
are encrypted by the encryption key to form the ciphertext
set CT A and CTB, and then CT A and CTB are uploaded to
the cloud server. The cloud server can only perform equality
tests between ciphertexts after obtaining the authorized secret
key from Alice and Bob, and it returns relevant plaintext
information according to the test results. The specific process
is shown in Figure 1.

V. THE PROPOSED SCHEME

This section describes a multivariate public key encryption
scheme with equality test. This scheme includes the following
six algorithms:

(1) Setup(14): Let k be a security parameter, M ∈ {0, 1}n
be a message space, K be a finite field of order q and
characteristic 2, g(x) ∈ K[x] be an irreducible polynomial of
degree n (n > 0) and K = K[x]/g(x) be an extension field
of degree n of K. This algorithm takes a security parameter k
as input and generates system parameters as follows:

(i) Define a polynomial mapping:
Let \( Q \) be a univariate mapping on \( \overline{K} \) :
\[
Q : x \mapsto y = x^{1+q^\theta},
\]
where \( 0 < \theta < n \), and the equation gcd\((q^\theta + 1, q^n - 1)\) = 1
is satisfied. Therefore, \( Q \) is an invertible map and its inverse
is given by
\[
Q^{-1}(x) = x^{t},
\]
where \( t(1 + q^\theta) = 1 \mod (q^n - 1) \). Let \( \varphi : \overline{K} \rightarrow \mathbb{K}^n \) be a
k-linear isomorphism:
\[
\varphi(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) = (a_0, a_1, \ldots, a_{n-1}).
\]

We can transform the mapping \( \hat{Q} \) into a quadratic poly-
nomial mapping \( \hat{Q}' : \mathbb{K}^n \rightarrow \mathbb{K}^n \) by using \( \varphi \) and its inverse
mapping \( \varphi^{-1} \); in brief,
\[
\hat{Q}'(x_1, x_2, \ldots, x_n) = \varphi \circ \hat{Q} \circ \varphi^{-1}(x_1, \ldots, x_n)
= (q_1, q_2, \ldots, q_n).
\]

where \( q_1, q_2, \ldots, q_n \in \mathbb{K}^n \). (ii) Define the internal perturba-
tion mapping:
Let \( r \) be a given small integer, and \( z_1, \ldots, z_r \) be a
set of randomly selected linear functions that depend
on \( x_1, x_2, \ldots, x_n \). In short,
\[
\begin{align*}
z_1(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \alpha_i x_i + \beta_1, \\
z_2(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \alpha_i x_i + \beta_2, \\
& \vdots \\
z_r(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \alpha_i x_i + \beta_r.
\end{align*}
\]

First, we define the mapping \( Z : \mathbb{K}^n \rightarrow \mathbb{K}^r \) to be
\[
Z(x_1, x_2, \ldots, x_n) = (z_1(x_1, \ldots, x_n), \ldots, z_r(x_1, \ldots, x_n)).
\]

Next, we randomly select \( n \) quadratic polynomials
\( \hat{q}_1, \ldots, \hat{q}_n \in \mathbb{K}[z_1, \ldots, z_r] \), and define a mapping
\( \hat{Q} : \mathbb{K}^r \rightarrow \mathbb{K}^n : \hat{Q}(z_1, \ldots, z_r) = (\hat{q}_1(z_1, \ldots, z_r), \ldots, \hat{q}_n(z_1, \ldots, z_r)) \).

Finally, the internal perturbation mapping is defined as
\[
Q'(x_1, x_2, \ldots, x_n) = \hat{Q} \circ Z(x_1, x_2, \ldots, x_n).
\]

Let \( P \) be the perturbation set, which is the set of
pairs \((\mu, \lambda)\), where \( \mu \) is the preimage of map \( \hat{Q} \), and \( \lambda \) is
the image of \( \mu \) under \( \hat{Q} \). We add internal perturbations to the
polynomial map \( Q' \) as follows:
\[
\hat{Q}(x_1, x_2, \ldots, x_n) = (Q' + Q^\lambda)(x_1, x_2, \ldots, x_n)
= (\hat{Q}_1, \hat{Q}_2, \ldots, \hat{Q}_n).
\]

(iii) Define the external perturbation mapping:
Randomly selecting \( a \) quadratic polynomial \( q_i^+ \)
\( x_1, x_2, \ldots, x_n \in \mathbb{K}[x_1, \ldots, x_n] \), and define an external pertur-
bation map
\[
Q^+ : \mathbb{K}^n \rightarrow \mathbb{K}^n : Q^+(x_1, x_2, \ldots, x_n) = (q_1^+ (x_1, \ldots, x_n), \ldots, q_n^+ (x_1, \ldots, x_n)).
\]
External perturbations are added to the map $\tilde{Q}$ to obtain
\begin{equation}
\tilde{Q} = \tilde{Q} \oplus Q^+ = (\tilde{Q}_1, \tilde{Q}_2, \ldots, \tilde{Q}_n, q_1^+, \ldots, q_n^+).
\end{equation}

(iv) Select a hash function $H: \{0, 1\}^n \rightarrow \mathbb{Z}_q$ and five hash functions
\begin{equation}
H_1, H_2, H_3, H_4: \mathbb{Z}_q \rightarrow \{0, 1\}^p,
\end{equation}
and
\begin{equation}
H_5: \mathbb{K}^{n+d} \rightarrow \{0, 1\}^n, p = \lceil n/4 \rceil.
\end{equation}

(v) Output the system parameters
\begin{equation}
p = (\mathbb{K}, \tilde{Q}, g(x), \phi, H, H_1, H_2, H_3, H_4, H_5).
\end{equation}

(2) KeyGen$(pp)$: This algorithm takes system parameters as input and outputs the public key $pk$ and private key $sk$.
Select randomly reversible affine transforms $S$, $T$, $T'$ in fields $\mathbb{K}^n$ and $\mathbb{K}^{n+d}$, respectively. To define two maps from domain $\mathbb{K}^n$ to $\mathbb{K}^{n+d}$,
\begin{equation}
Q_1(x_1, x_2, \ldots, x_n) = T \circ \tilde{Q} \circ S(x_1, x_2, \ldots, x_n)
= (\tilde{q}_1, \tilde{q}_2, \ldots, q_{n+d}).
\end{equation}
\begin{equation}
Q_2(x_1, x_2, \ldots, x_n) = T' \circ \tilde{Q} \circ S'(x_1, x_2, \ldots, x_n)
= (\tilde{q}'_1, \tilde{q}'_2, \ldots, q'_{n+d}).
\end{equation}

Taken together, the public key and private key are
\begin{equation}
(pk, sk) = ((Q_1, Q_2), (\theta, S, T, \tilde{Q}, Z, Q^+, S', T')).
\end{equation}

(3) Enc$(pk, m)$: This algorithm is used to encrypt a message $m = (m_1, \ldots, m_n)$ as follows:

(i) Construct a straight line and randomly select two points:
- Use $H_1, H_2, H_3$ and $H_4$ to generate two points
\begin{equation}
P_1 = (H_1(H(M)), H_2(H(M)))
\end{equation}
and
\begin{equation}
P_2 = (H_3(H(M)), H_4(H(M)));
\end{equation}
- Use $P_1$ and $P_2$ to construct a straight line $g_1(x) = ax + b$;
- Randomly select $x_1, x_2 \in \{0, 1\}^p$, to generate two points $(x_1, y_1), (x_2, y_2)$ on the line $g_1(x)$, where $x_1 \neq 0$ and $x_2 \neq 0$, if $x_1 = 0$ or $x_2 = 0$, reselect $x_1, x_2 \in \{0, 1\}^p$.

(ii) Randomly select reversible affine transforms $S$, $S'$ and $T$, $T'$ in fields $\mathbb{K}^n$ and $\mathbb{K}^{n+d}$, respectively, and compute
\begin{equation}
Q_1(m_1, \ldots, m_n) = (c_1, \ldots, c_{n+d}) = C_1,
\end{equation}
\begin{equation}
Q_2(x_1 \parallel y_1 \parallel x_2 \parallel y_2 \parallel H_5(C_1)) = (c_1', c_2', \ldots, c'_{n+d}) = C_2,
\end{equation}
and then output the ciphertext $c = (C_1, C_2)$.

(4) Dec$(sk, c)$: This algorithm decrypts the ciphertext by using the private key $sk$:

(i) Step 1: Make a transformation $T^{-1}$ for the ciphertext $(c_1, \ldots, c_{n+d})$, i.e.,
\begin{equation}
Y = T^{-1}(c_1, \ldots, c_{n+d}) = (y_1, \ldots, y_{n+d}),
\end{equation}
Step 2: Remove the external perturbation term from $Y$, that is to say, remove $\tilde{Y} = (y_{n+1}, y_{n+2}, \ldots, y_{n+a})$ and record the rest of the terms as $\tilde{Y} = (y_1, y_2, \ldots, y_n)$. Step 3: Calculate the inverse mapping of $\tilde{Q}$. For each $\mu$, there is a value of $\lambda$ that satisfies $\tilde{Q}(\mu) = \lambda$. After removing the internal perturbation, solve the inverse of $Q'$. Note that, for the inverse of $Q'$, it can be found by the inverse of $\tilde{Q}$. Specifically, for each $(\lambda, \mu) \in P$, compute
\begin{equation}
(y_{\lambda_1}, \ldots, y_{\lambda_n}) = \tilde{Q}^{-1}((y_1, y_2, \ldots, y_n) + \lambda),
\end{equation}
and
\begin{equation}
\tilde{\mu} = Z(y_{\lambda_1}, \ldots, y_{\lambda_n}),
\end{equation}
and check if $\tilde{\mu}$ is the same as the corresponding $\mu$. If it is not, discard it, if it is, go to the next step.

Step 4: Compute
\begin{equation}
X = (x_{\lambda_1}, \ldots, x_{\lambda_n}) = S^{-1}(y_{\lambda_1}, \ldots, y_{\lambda_n}),
\end{equation}
and $X$ is the plaintext.

(ii) Step 1: Make a transformation $T'^{-1}$ for the ciphertext $(c_1', \ldots, c'_{n+d})$, i.e.,
\begin{equation}
Y' = T'^{-1}(c_1', \ldots, c'_{n+d}) = (y_1', \ldots, y_{n+a}),
\end{equation}
Step 2: Remove external perturbation term from $Y'$, that is to say, remove $\tilde{Y}' = (y_{n+1}', y_{n+2}', \ldots, y_{n+a})$ and record the remaining terms as $\tilde{Y}' = (y_1', y_2', \ldots, y_n')$. Step 3: Calculate the inverse mapping of $Q'$. For each $(\lambda, \mu) \in P$, compute
\begin{equation}
(y'_{\lambda_1}, y'_{\lambda_2}, \ldots, y'_{\lambda_n}) = \tilde{Q}^{-1}(y_1', y_2', \ldots, y_n') + \lambda,
\end{equation}
and
\begin{equation}
\tilde{\mu}' = Z(y'_{\lambda_1}, y'_{\lambda_2}, \ldots, y'_{\lambda_n}),
\end{equation}
and then check whether $\tilde{\mu}'$ is the same as the corresponding $\mu$. If it is not, discard it, if it is, go to the next step.

Step 4: Compute
\begin{equation}
X' = (x'_{\lambda_1}, x'_{\lambda_2}, \ldots, x'_{\lambda_n}) = S'^{-1}(y'_{\lambda_1}, y'_{\lambda_2}, \ldots, y'_{\lambda_n}),
\end{equation}
then
\begin{equation}
X' = x_1 \parallel y_1 \parallel x_2 \parallel y_2 \parallel H_5(C_1).
\end{equation}
It is easy to compute $(x_1, y_1)$ and $(x_2, y_2)$.

(3) Verify the validity of the messages
\begin{itemize}
\item Generate $P_1 = (H_1(H(M)), H_2(H(M)))$ and $P_2 = (H_3(H(M)), H_4(H(M)))$.
\item Use $P_1$ and $P_2$ to construct a straight line $g_2(x) = cx + d$;
\item Verify the following equations
\begin{equation}
\begin{cases}
g_2(x_1) = y_1, \\
g_2(x_2) = y_2.
\end{cases}
\end{equation}
\end{itemize}

If each equation of Eq.(32) holds, then output the message $m$, otherwise, output $\bot$.

(5) Aut$(sk)$: This algorithm takes as a part of private key $(S', T')$ as input and outputs a trapdoor $td = (S', T')$. 

75468

VOLUME 8, 2020
Suppose there are two users Alice and Bob in the system. User Alice inputs an authorized private key $subsk_A$ and receives as output the trapdoor $td_A = (S'_A, T'_A)$. User Bob inputs an authorized private key $subsk_A$ and receives as output the trapdoor $td_B = (S'_B, T'_B)$. This algorithm takes trapdoors $(td_A, td_B)$ and ciphertexts $(c_A, c_B)$ as input, and outputs 1 or 0. The test process is as follows:

- Decrypt $c_A$ to obtain
  $$\left(x_{A_1}, y_{A_1}, x_{A_2}, y_{A_2}\right), \tag{33}$$
  and decrypt $c_B$ to obtain
  $$\left(x_{B_1}, y_{B_1}, x_{B_2}, y_{B_2}\right). \tag{34}$$
- Construct the straight line $g_A(x)$ by using $(x_{A_1}, y_{A_1})$ and $(x_{A_2}, y_{A_2})$. Construct the straight line $g_B(x)$ by using $(x_{B_1}, y_{B_1})$ and $(x_{B_2}, y_{B_2})$.
- If $g_A(x) = g_B(x)$, then output 1, otherwise output 0.

**VI. PERFORMANCE ANALYSIS**

In this section, we analyze the performance and space cost of our scheme. In this algorithm, the base field is $GF(2)$, so that the coefficients of the polynomials will be expressed by 1 bit.

1) **PUBLIC KEY SIZE**

The public key contains $2(n + a)$ quadratic polynomials, and each polynomial has $\frac{2(n+1)}{2}$ quadratic terms, $n$ linear terms and one constant term. The key size is approximately $O(n^3)$ bit. Typically, in our experiment, we select the parameters $(q, n, r, a, \theta) = (2, 84, 6, 14, 4)$, so the public key size is approximately 716380 bits.

2) **PRIVATE KEY SIZE**

The main parts of the private key are: the four linear transformations $S, T, S'$ and $T'$ and their inverses, which need $2n^2 + 2(n + a)^2$ bits; the $r$ linear function $z_j$, which is of size $r(n + 1)$ bits; the $n$ quadratic polynomials with $r$ variables, which need roughly $nr(r + 1)$ bits; and the $a$ quadratic polynomials with $n$ variables, which need roughly $an(n + 1)$ bits; The total is $O(n^2)$ bits. According to the same parameters as the public key, the private key size is approximately 87437 bits.

3) **ENCRYPTION COMPUTATIONAL COMPLEXITY**

For encryption, we need to compute the value of $2(n + a)$ quadratic polynomials. We can rewrite the quadratic polynomials in the following form:

$$\sum_{i=1}^{n} x_i(b_1 + \sum_{j=i}^{n} a_{ij}x_j) + c, \tag{35}$$

which allows us to compute the value of $2(n + a)$ polynomials faster than direct calculation. Therefore, we need roughly $n(n + 3)$ binary operations to compute the value of each polynomial. That is, $2n(n + a)(n + 3)$ binary operations are needed to encrypt a message, and it takes approximately 100ms to complete the Encryption.

4) **DECRYPTION COMPUTATIONAL COMPLEXITY**

The main part of the decryption process is step 3, where we need to calculate the values of $Q^{-1}$ and the values of $z_i 2^t$ times and compare them with the second components of the corresponding elements in $P$, this requires approximately $2'[4n + 3n - 1]$ binary operations and $2'$ modular exponentiations in finite field $GF(2^{84})$. The performing $L^{-1}$ in the first step and $S^{-1}$ in the last step require approximately $O(n^2)$ binary calculations. It takes approximately 4.68 seconds to complete the Decryption.

5) **EQUALITY TEST COMPUTATIONAL COMPLEXITY**

For the equality test, the tester first uses the trapdoor to decrypt part of the ciphertext. Second, constructing two straight lines requires 12 addition operations and 6 multiplication operations, and comparing the two lines requires only one addition and one multiplication operation. Therefore, the equality test algorithm needs 20 binary operations in addition to one decryption algorithm, and it takes approximately 4.70 seconds to complete the Equality test. In summary, the Table 2 provides a clear description of the performance of our scheme.

Moreover, in order to demonstrate the advantage of our MPKEET scheme, we compare it with some existing PKEET schemes ([4]–[6], [10], [11]) in Table 3, in terms of public key size, private key size, encryption time, decryption time, equality test time and the ability to resist quantum attacks. These experiments are performed on PC with Intel i5-4460 CPU @3.2 GHz and 4GB memory running Windows 7 64 bits system. Since our hardware environment and the security level is the same as Wang et al., we directly exploit the time cost of exponential and pairing operations given by Wang et al. [12], and estimate the time costs of algorithms of these schemes. In particular, the software used in ([4]–[6], [10], [11]) is VC++ 6.0, while our scheme use mathematical software Magma V2.12-16. If we convert it into VC++6.0, the performance of our scheme will be improved significantly.

As shown in Table 3, we compared efficiency, storage costs and security of our scheme with other existing schemes.

- **Storage costs:** Compared with other PKEET schemes, the key size of our scheme is large, it is a common disadvantage of post-quantum cryptography.
- **Efficiency:** The computation costs in the encryption are at least 94% less than that of other schemes, and test

---

**TABLE 2. Performance analysis of our MPKEET scheme.**

| Function | Time/Size |
|----------|-----------|
| Public key | 716380 bits |
| Secret key | 87437 bits |
| Encryption | 0.1s |
| Decryption | 4.68 s |
| Equality test | 4.70 s |
TABLE 3. The comparison of computational complexity.

| Scheme   | Public key size (bits) | Secret key size (bits) | Encryption (s) | Decryption (s) | Equality test (s) | Resist quantum |
|----------|------------------------|------------------------|----------------|----------------|--------------------|----------------|
| Tan12 [4]| 2544 bits             | 352 bits               | 4E_1 ≈ 3.24    | 2E_1 ≈ 1.62    | 4P ≈ 7.64          | No             |
| MZH15 [5]| 3512 bits             | 3512 bits              | 4E_1+1E_2+1P ≈ 6.05 | 1P+3E_1+1E_2 ≈ 5.24 | 4P+3E_2 ≈ 9.26     | No             |
| HTC15 [6]| 2544 bits             | 2512 bits              | 2E_1 ≈ 3.24    | 4E_1             | 4P+2E_2 ≈ 7.06     | No             |
| Ma16 [10]| 2544 bits             | 512 bits               | 4E_1+2E_2+2P ≈ 8.86 | 2P+2E_1 ≈ 5.44  | 4P ≈ 7.64          | No             |
| WZC17 [11]| 2544 bits           | 3512 bits              | 2E_2 ≈ 1.8     | 2P ≈ 3.82       | 2P+2E_2 ≈ 5.62     | No             |
| Our scheme | 7536 bits             | 8743 bits              | 1432368b ≈ 0.10 | 64H_{GF(2^84)} ≈ 4.68 | 64H_{GF(2^84)} ≈ 4.68 | Yes            |

(G_1, G_2, q): Groups G_1 and G_2 of prime order q for a pairing: e : G_1 × G_2 → G_2.
E_G1: The time-cost of a exponentiation in group G_1.
E_G2: The time-cost of a exponentiation in group G_2.
E_{GF(2^84)}: The time-cost of a exponentiation in group GF(2^84).
P: The time-cost of a pairing operation.
a: The time-cost of a binary operation.
| G_1 | Length of one element in G_1.
| G_2 | Length of one element in G_2.
| Z_q | Length of one random number in Z_q.

Security parameters: |G_1| = 272 bits, |G_2| = 1088 bits, |Z_q| = 256 bits (recommended by [12]).

Valuation basics: H_{GF(2^84)} = 0.811, H_{GF(2^96)} = 0.903, 1P+1.91s ([12])

phases of our scheme decrease by 16.7% as compared to that of the above scheme. The computation costs in the Decryption phase of our scheme are decrease by 10.7% and 13.97% as compared to that of Ma et al.’s scheme [5] and Ma’s scheme [10], respectively. In practice, the server only matches the same ciphertext without decryption. Therefore, our scheme achieves a better computation performance.

- Security: Except our scheme, other schemes can’t resist the attack of quantum computer.

VII. SECURITY ANALYSIS

The proposed MPKEET scheme of this paper adds an equality test algorithm to the traditional PMI+ scheme. Our core idea is to construct two points by using four one-way hash values of the message, and then determine a straight line from these two points. In the following process of encryption and decryption, we use the random selection of these two points to complete the design of the scheme. Therefore, when the attacker has trapdoor information, he can decrypt the randomly selected coordinates of two points, but due to the randomness of these two coordinates, he still cannot recover the message. In a case without a trapdoor, our scheme and the PMI+ system have the same security.

The security analysis of PMI+ multivariate systems can be divided into two categories: structural attacks and direct attacks. Structural attacks aim at the special structural characteristics of multivariate systems, and they mainly include linearized equation attack and differential attacks. Direct attacks start with public key polynomials of multivariate systems. The Gröbner basis algorithm and XL algorithm are commonly used as attack methods. It is generally believed that as long as the complexity of the attack scheme exceeds O(2^80), a scheme can resist such attacks. At present, the main method of multivariate cryptanalysis is experimental statistics. Aiming at the existing attack algorithms, attempt to experimentally determine the running time and memory requirements for a attack on PMI, and find the appropriate parameters through the experimental data.

A. RESISTING LINEARIZATION EQUATION ATTACKS

Linearization equation attacks were first proposed to attack the MI system. The idea is to obtain the following linear equation

\[ \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}x_{i}y_{j} + \sum_{i=1}^{n} b_{i}x_{i} + \sum_{j=1}^{m} c_{j}y_{j} + d = 0, \quad (36) \]

where \( X = \{x_1, x_2, \ldots, x_n\} \) and \( Y = \{y_1, y_2, \ldots, y_m\} \) are input vector and output vector of the public key. The attacker generates enough plaintext-ciphertext pairs \((X, Y)\) through the public-key equations, and substituted them into the equation \((36)\), and solved all its coefficients by using the Gaussian elimination method or its improved algorithm, so as to obtain the equivalent linear relation to achieve the purpose of cracking and decrypting the encryption scheme. In this paper, we randomly selected \((n+a+1)^2\) plaintext-ciphertext pairs and substituted them into equation \((36)\) to solve the linear system of equations with coefficients. It was found that there was only a zero solution. That is to say, the public key system did not contain a linear equation with the form of \((36)\), so it could resist the attack of linear equation.

B. RESISTING DIFFERENTIAL ATTACKS

Differential attacks were introduced by Fouque et al. [28] to analyzed PMI encryption system. The main ideas of the attack is to find a linear subspace of the plaintext space by calculating the differential of the central map, and limiting the public key to this space, so that all the internal disturbance terms become constant, and to then use the linearization attack method to recover the plaintext of the ciphertext.

An effective way to resist the differential attack is to disturb the polynomial system by adding randomly a small amount of Plus polynomials according to the Plus method [29], so that
the dimension of the kernel of the differential of the central map $Q$ is the same for nearly every vector in $\mathbb{F}^n$. As pointed out in [16], if $\gcd(\theta, n) = 1$, then at least 10 Plus polynomials be added to resist differential attack. Otherwise, $\gcd(\theta, n) + 10$ Plus polynomials be added to protect PMI+.

In this paper, the parameters are chosen as $(q, n, r, a, \theta) = (2, 84, 6, 14, 8)$. Therefore our MPKEET can resist differential attacks by adding 14 polynomial, because of $\gcd(\theta, n) + 10 = 14$.

\section*{C. RESISTING GRÖBNER BASIS ATTACKS}

The Gröbner basis is a classical algorithm for solving systems of multivariate polynomial equations, but it usually does not change whether the number of equations exceeds the number of variables. A rough estimate is given in reference [30], current computing equipment can only solve equations with no more than 15 variables in the finite field $GF(2^8)$, and the complexity is about $O(2^{30})$ when the number of variables reaches 20. At present, the fastest of such algorithms known are the F4 [31] and F5 [30] algorithms. According to [32], the attack complexity of F5 algorithm is about $2^{87.3n}$. Therefore, with the parameters $n \geq 92$, our scheme is secure against attacks using the Gröbner basis.

\section*{D. RESISTING XL ATTACKS}

The XL attack can be viewed as a combination of bounded degree Gröbner basis and linearization. The basic idea of this method is to generate a large number of higher order variants by multiplying each polynomial equation with all possible bounded order monomials, and generate a large number of higher order variants, and then linearize the expanded system. According to [33], the attack complexity of XL algorithm is about $\left(\frac{2^{2n}}{\omega^2}\right)^{\omega}$. When using the general elimination method $\omega = 3$, and when using the improved algorithm $\omega = 2.3766$, Therefore, Our scheme has an attack complexity greater than $2^{94}$.

Through the above analysis, it is concluded that the security of our scheme has the attack complexity $2^{90}$, with all the known attack methods of multivariate public key cryptography.

\section*{VIII. CONCLUSION}

In this paper, we first formalized the definition of multivariate public key encryption scheme with an equality test, which can be used for the outsourced computation of encrypted data. In contrast to the formulation in [1], our scheme allows users to specify who can perform the equality test between ciphertexts, this construction meets theoretical and practical security requirements. Moreover, we analyzed the algorithmic complexity of our scheme and showed that our scheme can resist linearization equation attacks, differential attacks, XL attack and Gröbner basis attacks, when appropriate parameters are selected. Compared to previous schemes with equality test, their securities depend on discrete logarithm problem, which can be solved in polynomial time by efficient shor’s quantum algorithms [14]. Therefore, these public key cryptography algorithms are no longer secure in the era of quantum computing, the scheme proposed in this paper overcomes this shortcoming, and can resist the attack of quantum algorithm.

\section*{REFERENCES}

[1] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, “Probabilistic public key encryption with equality test,” in Proc. Cryptogr. Track RSA Conf., in Lecture Notes in Computer Science, vol. 5985. San Francisco, CA, USA: Springer, Mar. 2010, pp. 119–131.

[2] Q. Tang, “Towards public key encryption scheme supporting equality test with fine-grained authorization,” in Proc. Australas. Conf. Inf. Secur. Privacy, in Lecture Notes in Computer Science, vol. 6812. Melbourne, VIC, Australia: Springer, Jul. 2011, pp. 389–406.

[3] Q. Tang, “Public key encryption supporting plaintext equality test and user-specified authorization,” Secure Commun. Netw., vol. 5, no. 12, pp. 1351–1362, Dec. 2012.

[4] Q. Tang, “Public key encryption schemes supporting equality test with authorisation of different granularity,” Int. J. Appl. Cryptogr., vol. 2, no. 4, pp. 304–321, 2012.

[5] S. Ma, M. Zhang, Q. Huang, and B. Yang, “Public key encryption with delegated equality test in a multi-user setting,” Comput. J., vol. 58, no. 4, pp. 986–1002, Apr. 2015.

[6] K. Huang, R. Tso, Y.-C. Chen, S. M. M. Rahman, A. Almogren, and A. Alamri, “PKE-AET: Public key encryption with authorized equality test,” Comput. J., vol. 58, no. 10, pp. 2686–2697, Oct. 2015.

[7] H. T. Lee, S. Ling, J. H. Seo, and H. Wang, “CCAA2 attack and modification of Huang et al.’s public key encryption with authorized equality test,” Comput. J., vol. 59, no. 11, pp. 1689–1694, Nov. 2016.

[8] S. Ma, Q. Huang, M. Zhang, and B. Yang, “Efficient public key encryption with equality test supporting flexible authorization,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 3, pp. 458–470, Mar. 2015.

[9] X. J. Lin, H. Qu, and X. Zhang, “Public key encryption supporting equality test and flexible authorization without bilinear pairings,” IACR Cryptol. ePrint Arch., 2016, p. 277.

[10] S. Ma, “Identity-based encryption with outsourced equality test in cloud computing,” Int. J. Secur. Commun. Netw., vol. 2, no. 4, pp. 389–402, Jun. 2016.

[11] L. Wu, Y. Zhang, K.-K.-R. Choo, and D. He, “Efficient and secure identity-based encryption scheme with equality test in cloud computing,” Future Gener. Comput. Syst., vol. 73, pp. 22–31, Aug. 2017.

[12] Q. Wang, L. Peng, H. Xiong, J. Sun, and Z. Qin, “Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing,” IEEE Access, vol. 6, pp. 760–771, 2018.

[13] H. Zhu, L. Wang, H. Ahmad, and X. Niu, “Key-policy attribute-based encryption with equality test in cloud computing,” IEEE Access, vol. 5, pp. 20428–20439, 2017.

[14] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM Rev., vol. 41, no. 2, pp. 303–332, Jan. 1999.

[15] A. L-T. Chen, M-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee, and B.-Y. Yang, “SSE implementation of multivariate PKCs on modern x86 CPUs,” in Proc. 11th Int. Workshop Cryptograph. Hardw. Embedded Syst., in Lecture Notes in Computer Science, vol. 5747. Lausanne, Switzerland: Springer, Sep. 2009, pp. 33–48.

[16] J. Ding and J. E. Gower, “Inoculating multivariate schemes against differential attacks,” in Proc. 9th Int. Conf. Theory Pract. Public-Key Cryptogr., in Lecture Notes in Computer Science, vol. 3958. New York, NY, USA: Springer, Apr. 2006, pp. 290–301.

[17] B. Lv, “Efficiency optimization and hardware implementation of multivariate public key scheme,” Ph.D. dissertation, South China Univ. Technol., Guangzhou, China, 2018.

[18] Z. Yu, C.-Z. Gao, Z. Jing, B. B. Gupta, and Q. Cai, “A practical public key encryption scheme based on learning parity with noise,” IEEE Access, vol. 6, pp. 31918–31923, 2018.

[19] P. K. Premkamal, S. K. Pasupuleti, and P. J. A. Alphonse, “Efficient escrow-free CP-ABE with constant size ciphertext and secret key for big data storage in cloud,” Int. J. Cloud Appl. Comput., vol. 10, no. 1, pp. 28–45, Jan. 2020.

[20] C. Li, Z. Zhang, and L. Zhang, “A novel authorization scheme for multimedia social networks under cloud storage method by using MA-CP-ABE,” Int. J. Cloud Appl. Comput., vol. 8, no. 3, pp. 32–47, Jul. 2018.
[21] Q. Zheng, X. Wang, M. K. Khan, W. Zhang, B. B. Gupta, and W. Guo, “A lightweight authenticated encryption scheme based on chaotic SCML for railway cloud service,” IEEE Access, vol. 6, pp. 711–722, 2018.

[22] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., in Lecture Notes in Computer Science, vol. 3027. Interlaken, Switzerland: Springer, May 2004, pp. 506–522.

[23] M. Dabra and B. B. Gupta, “An efficient KP design framework of attribute-based searchable encryption for user level revocation in cloud,” Concurrency Comput., Pract. Exper., p. e5291, Apr. 2019.

[24] M. Dabra, B. B. Gupta, and S. T. Ali, “ISEkFT: An IBE-based searchable encryption scheme with k-keyword fuzzy search trapdoor,” J. Inf. Technol. Res., vol. 12, no. 3, pp. 133–153, 2019.

[25] M. Dabra, B. B. Gupta, and S. T. Ali, “Dynamic policy attribute based encryption and its application in generic construction of multi-keyword search,” Int. J. E-Services Mobile Appl., vol. 11, no. 4, pp. 16–38, Oct. 2019.

[26] T. Matsumoto and H. Imai, “Public quadratic polynomial-tuples for efficient signature-verification and message-encryption,” in Proc. Workshop Theory Appl. Cryptograph. Techn., in Lecture Notes in Computer Science, vol. 330. Davos, Switzerland: Springer, May 1988, pp. 419–453.

[27] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman, 1979.

[28] P. Fouque, L. Granboulan, and J. Stern, “Differential cryptanalysis for multivariate schemes,” in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn., in Lecture Notes in Computer Science, vol. 3494. Aarhus, Denmark: Springer, May 2005, pp. 341–353.

[29] J. Patarin, L. Goubin, and N. Courtois, “C*+ and HM: Variations around two schemes of T. Matsumoto and H. Imai,” in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur., in Lecture Notes in Computer Science, vol. 1514. Beijing, China: Springer, Oct. 1998, pp. 35–49.

[30] J. C. Faugère, “A new efficient algorithm for computing Gröbner bases without reduction to zero (F₅),” in Proc. Int. Symp. Symbolic Algebr. Comput., 2002, pp. 75–83.

[31] J.-C. Faugère, “A new efficient algorithm for computing Gröbner bases (F₄),” J. Pure Appl. Algebra, vol. 139, nos. 1–3, pp. 61–88, 1999.

[32] B.-Y. Yang, J.-M. Chen, and N. T. Courtois, “On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis,” in Proc. 6th Int. Conf. Inf. Commun. Secur., in Lecture Notes in Computer Science, vol. 3269. Malaga, Spain: Springer, Oct. 2004, pp. 401–413.

[33] G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita, “Comparison between XL and Gröbner basis algorithms,” in Proc. 10th Int. Conf. Theory Appl. Cryptol. Inf. Secur., in Lecture Notes in Computer Science, vol. 3329. Jeju Island, South Korea: Springer, Dec. 2004, pp. 338–353.

XIAOYING SHEN received the M.S. degree from Northwest Normal University, in 2017. She is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. Her current research fields include modern cryptography, network security, and blockchain technology.

LICHENG WANG received the Ph.D. degree from Shanghai Jiao Tong University, in 2007. He is currently an Associate Professor at the Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include modern cryptography, network security, and trust management.

HUJUN ZHU received the B.S. degree from Luoyang Normal University, in 2007, the M.S. degree from Henan Polytechnic University, in 2010, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2018. She is currently an Associate Professor at the Nanyang Institute of Technology. Her research interests include modern cryptography, network security, and cloud computing.

YUAN LIU received the B.S. and M.S. degrees from Henan Polytechnic University, in 2014 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. Her research interests include modern cryptography and network security.