Hindawi
Security and Communication Networks
Volume 2018, Article ID 6909703, 9 pages
https://doi.org/10.1155/2018/6909703

Research Article

FTP: An Approximate Fast Privacy-Preserving Equality Test Protocol for Authentication in Internet of Things

Youwen Zhu,[1,2] Yue Zhang,[1,3] Jiabin Yuan,[1] and Xianmin Wang[4]

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China*
[4]*School of Computer Science, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Xianmin Wang; xianmin@gzhu.edu.cn

Received 26 April 2018; Accepted 30 August 2018; Published 18 October 2018

Academic Editor: Laurence T. Yang

Privacy-preserving string equality test is a fundamental operation of many algorithms, including privacy-preserving authentication in Internet of Things (IoT). Existing secure equality test schemes can theoretically achieve string equality comparison and preserve the private strings. However, they suffer from heavy computation and communication cost, especially while the strings are of hundreds of bits or longer, which is not suitable for IoT applications. In this paper, we propose an approximate Fast privacy-preserving equality Test Protocol (FTP), which can securely complete string equality test and achieve high running efficiency at the cost of little accuracy loss. We strictly analyze the accuracy of our proposed scheme and formally prove its security. Additionally, we leverage extensive simulation experiments to evaluate the running cost, which confirms our high efficiency; for instance, our proposed FTP can securely compare two 256-bit strings within 0.7 seconds on ordinary laptops.

1. Introduction

In recent years, with the growth of privacy concern, privacy-preserving computation [1–4] receives increasing attention, since various privacy-preserving computation schemes can support computation on private data while keeping the privacy of the involved data. Sensitive data collection and analysis over the encrypted data become the current trend [5–12]. Based on this situation, Privacy-preserving Equality Test (PET) aims at securely comparing two binary strings which are privately held by two parties. That is, by PET scheme, two participants can securely work out whether their binary strings are exactly equal or not; meanwhile each participant can obtain no useful information about the private binary string of the other participant; even two strings are the same. PET is a significant basic building of many privacy-preserving schemes, such as privacy-preserving authentication [13–15], secure comparison of biological characteristics [16–18], privacy-preserving machine learning [19–21], secure cost comparison in wireless network [22, 23], privacy-preserving threshold schema in recommendation systems [3], attribute comparison in attribute-based encryption [24–26], and secure query in cloud [27, 28]. For example, Internet-of-Things (IoT) applications may authenticate users in privacy-preserving manner. For completing the authentication, a user needs to submit his/her authentication credential to IoT system, and the system decides whether the user is legal or not by comparing the user's authentication credential with authentication information stored in the system database. As privacy concern, the user cannot reveal his/her authentication credential to the system, and the latter can just access them in encrypted form. Meanwhile, to protect the privacy of the IoT system, any user cannot learn useful information of database stored in the IoT system. This dilemmatic problem can be solved by employing a PET protocol.

As its wide applications, several works have devoted to PET recently. Nateghizad et al.'s scheme [29], denoted as NEL16, is the state-of-the-art approach to achieve PET, which is also the most efficient PET method up to now. NEL16 can be viewed as an improved method of Lipmaa and Toft's PET scheme in [30] denoted as LT13. In LT13 [30], Lipmaa and Toft compute the Hamming distance of

two private binary strings in encrypted form. Then, they generate a Lagrange interpolating polynomial that outputs 0 if the input equals 0 and outputs 1 otherwise. Finally, the comparison result is figured out in encrypted form by securely evaluating the Lagrange interpolating polynomial with encrypted Hamming distance as input. Compared to LT13, NEL16 further computes the number of "1" of binary representation of the Hamming distance in encrypted form and uses the number of "1", instead of the Hamming distance, to evaluate the Lagrange interpolating polynomial. Suppose binary representation of the Hamming distance has t bits. The number of "1" must be not bigger than t, which can be represented by using just $\lceil \log_2 t \rceil$ bits. While $t \geqslant 2$, it always has $t > \lceil \log_2 t \rceil$. Thus, NEL16 requires a lower-degree Lagrange polynomial and can reduce running time. However, NEL16 still cannot achieve practical running efficiency, since computing the number of "1" in encrypted form is also time-consuming. As shown in [29], while implementing them on a Linux machine of 64-bit microprocessor and 8 GB RAM to compare two 256-bit binary strings, LT13 and NEL16 both cost tens of seconds. Therefore, existing PET schemes still suffer from low efficiency.

In this paper, we propose a new PET scheme, named Fast privacy-preserving equality Test Protocol (FTP), which has high efficiency at the cost of little error rates. In FTP, we randomly convert the original binary strings into shorter ones, then the shorter binary strings are securely compared to decide whether the original ones are the same, by which we can dramatically reduce both computation cost and communication overheads. Although FTP just compares shorter strings, we can ensure the comparison result is exactly correct if the original binary strings are the same or they have an odd number of different bits, and the comparison result has low false-positive rates while they have an even number of different bits. For data privacy, our proposed FTP can achieve provable security, and no private information is disclosed throughout the protocol. In general, our main contributions in this paper can be summarized as follows:

(i) We propose a Fast privacy-preserving equality Test Protocol, named FTP, which can achieve much high running efficiency than the state-of-the-art PET schemes. FTP can guarantee an exactly correct comparison result while the involved binary strings are the same or have an odd number of different bits and has a low false-positive rate if the compared strings have an even number of different bits.

(ii) We formally prove the security of FTP and can guarantee no privacy is disclosed throughout the proposed protocol.

(iii) We strictly analyze the accuracy loss of FTP and leverage extensive experiments to evaluate the running cost. The results indicate that FTP is highly accurate and can dramatically reduce running cost.

The rest of this paper is organized as follows. In Section 2, we describe preliminaries and system model. In Section 3, we present our approximate fast privacy-preserving equality test in detail and theoretically analyze its accuracy loss. In Section 4, we formally prove the security of our scheme, evaluate our running efficiency, and compare our scheme with previous ones. In Section 5, we simply review the related work. At last, we conclude this paper in Section 6.

2. System Model and Preliminaries

2.1. Paillier Encryption System. In [31], Paillier proposes a probabilistic public key encryption scheme with semantic security (Indistinguishability under Chosen-Plaintext Attack, IND-CPA). Its steps are concisely described as follows.

Key Generation. Select two large enough primes p and q. Then, the secret key sk is $\lambda = \mathtt{lcm}(p-1, q-1)$, i.e., the least common multiple of $p-1$ and $q-1$. The public key pk is (n, g), where $n = pq$ and $g \in \mathbb{Z}^*_{n^2}$ such that $\mathtt{gcd}(\mathtt{L}(g^\lambda \bmod n^2), n) = 1$, that is, the maximal common divisor of $\mathtt{L}(g^\lambda \bmod n^2)$, and n equals 1. Here, $\mathtt{L}(x) = (x-1)/n$.

Encryption. Let m_0 be a number in plaintext space $\mathbb{Z}_n$. Select a random $r \in \mathbb{Z}^*_n$ as the secret parameter, then the ciphertext of m_0 is $c_0 = g^{m_0} r^n \bmod n^2$.

Decryption. Let $c_0 \in \mathbb{Z}_{n^2}$ be a ciphertext. The plaintext hidden in c_0 is

$$m_0 = \frac{\mathtt{L}\left(c_0^\lambda \bmod n^2\right)}{\mathtt{L}\left(g^\lambda \bmod n^2\right)} \bmod n. \tag{1}$$

In Paillier encryption system, it obviously has

$$\begin{aligned} &\mathtt{E}_{pk}(m_1, r_1) * \mathtt{E}_{pk}(m_2, r_2) \\ &\quad = \mathtt{E}_{pk}(m_1 + m_2, r_1 * r_2) \bmod n^2 \end{aligned} \tag{2}$$

where $\mathtt{E}_{pk}(m, r)$ denotes the encrypted result of m using public key pk and random secret parameter r. That is, the product of ciphertexts of m_1 and m_2 is a ciphertext of $m_1 + m_2$. Thus, Paillier encryption scheme is additively homomorphic. Further, for any $k \in \mathbb{Z}_n$, there is

$$\mathtt{E}_{pk}(m, r)^k = \mathtt{E}_{pk}\left(k * m, r^k\right) \bmod n^2, \tag{3}$$

i.e., the k-th power of $\mathtt{E}_{pk}(m, r)$ is a ciphertext of $k * m$.

Besides, Paillier cryptosystem has the self-blinding property as it is a probabilistic encryption. For any plaintext $m \in \mathbb{Z}_n$, it has $\mathtt{E}_{pk}(m, r_1) * r_2^n = \mathtt{E}_{pk}(m, r_1 r_2) \bmod n^2$ and $m = \mathtt{D}_{sk}(\mathtt{E}_{pk}(m, r_1)) = \mathtt{D}_{sk}(\mathtt{E}_{pk}(m, r_1))$, in which $\mathtt{D}_{sk}(\cdot)$ denotes the corresponding decryption function.

Paillier encryption system is a significant secure basic tool of our scheme, which will be utilized to encrypt private data and support necessary computation. For simplicity, we use $[\![m]\!]$ to denote the ciphertext of m encrypted by Paillier cryptosystem, while the random parameter r is no need to be pointed out.

2.2. System Model. In this paper, we consider privacy-preserving user authentication in IoT. A user (named Bob) submits a u-bit authentication credential to system (named Alice), and the system decides whether the user is legal or not by comparing Bob's authentication credential with the authentication information stored in the system database.

As privacy concern, Bob cannot reveal the authentication credential and authentication result to Alice, and Alice just obtains them in encrypted form. Meanwhile, to protect the privacy of Alice, Bob cannot learn any information of Alice's database. This dilemmatic problem can be seen as a privacy-preserving equality test (PET) problem as follows.

Privacy-Preserving Equality Test (PET) Problem. PET involves two parties: Alice and Bob. Alice privately hold u-bit binary strings $x = (x_1, x_2, \cdots, x_u)$ and Bob $y = (y_1, y_2, \cdots, y_u)$. Here, x and y can be also considered as two integers that belong to $[0, 2^u - 1]$. Besides, Bob has a public key pair (pk, sk) of Paillier encryption system, where pk is public key and sk is secret key. They want to securely compare x and y such that only Alice obtains the comparison result in encrypted form; i.e., Alice gains $[\![\theta]\!]$ in which

$$\theta = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Additionally, x should be privately kept to Alice throughout the protocol, and Bob's private string y cannot be disclosed to Alice or anybody else. Neither Alice nor Bob can learn the real value of θ.

2.3. Security Model. In this paper, we assume that the participants Alice and Bob are semihonest. It means that each participant follows the protocol correctly but records all the received information in the protocol to infer as much information about the private data of the other participant as possible. In [32], Goldreich gives a formal definition of security against semihonest adversaries, which can be described as follows.

Definition 1 (privacy under semi-honest model [32]). Let $f(x, y)$ be a functionality and $f_1(x, y)$ (resp. $f_2(x, y)$) denote the first (resp., second) element of $f(x, y)$. Let Π be a two-party protocol for computing $f(x, y)$ such that the first (resp., second) party obtains $f_1(x, y)$ (resp., $f_2(x, y)$). The view of the first (resp., second) party during an execution of Π on (x, y), denoted as $VIEW_1(x, y)$ (resp., $VIEW_2(x, y)$), is $(x, r, m_1, \cdots, m_t)$ (resp., $(y, r, m_1, \cdots, m_t)$), where x (resp., y) represents the input of the first (resp., second) party, r represents its random number, and m_i represents the i-th message it has received. We say that protocol Π privately computes function f, i.e., Π is secure against semihonest adversaries, if there exist probabilistic polynomial-time algorithms S_1 and S_2, such that

$$S_1(x, f_1(x, y)) \overset{c}{\equiv} (VIEW_1(x, y)) \tag{5}$$

$$S_2(y, f_2(x, y)) \overset{c}{\equiv} (VIEW_2(x, y)) \tag{6}$$

where $\overset{c}{\equiv}$ represents computational indistinguishability.

2.4. Design Goal. For PET problem shown in Section 2.2, we aim at proposing a new solution to achieve the following security and performance goals.

(i) ***High Accuracy***. The protocol should arrive at a correct output with high probability while both participants exactly follow the protocol steps. That is, the solution should be of high accuracy to output a correct comparison result.

(ii) ***Input Privacy***. Throughout the protocol, each bit of the private inputs x and y should be known to its owner only. That it, any useful information about x cannot be disclosed to Bob, and y cannot be revealed to Alice.

(iii) ***Result Privacy***. Both users cannot get the value of result θ in plaintext, and only Alice can obtain the encrypted output $[\![\theta]\!]$ which is encrypted by Bob's public key.

(iv) ***Efficiency***. The protocol needs to employ a sublinear number of public key encryption and decryption such that it can achieve high running efficiency even while x and y are of hundreds of bits.

2.5. Review of LT13 Scheme. In this following, we will simply introduce the previous PET schemes LT13 [30].

Generally, LT13 consists of two stages: (1) Computing the encrypted Hamming distance $[\![d]\!]$ between x and y such that only Alice learns $[\![d]\!]$. During the first stage, Bob uses the public key pk to encrypt his private bit y_i for $i = 1$ to u and sends each $[\![y_i]\!]$ to Alice. Then, based on (7) and the additively homomorphic property of Paillier encryption scheme,

$$x_i \oplus y_i = \begin{cases} 1 - y_i, & \text{if } x = 1, \\ y_i, & \text{if } x = 0. \end{cases} \tag{7}$$

Alice can obtain the encrypted Hamming distance $[\![d]\!] = \prod_{i=1}^{u} [\![x_i \oplus y_i]\!]$ where $[\![x_i \oplus y_i]\!] = [\![1]\!] * [\![y_i]\!]^{-1}$ if $x_i = 1$ and $[\![x_i \oplus y_i]\!] = [\![y_i]\!]$ if $x_i = 0$.

(2) Computing the final result $[\![\theta]\!]$ which is also known to Alice only. To this end, they first select a u-degree public Lagrange interpolation polynomial $F(z) = \sum_{i=0}^{u} \alpha_i z^i$ that satisfies $F(1) = 1, F(2) = F(3) = \cdots = F(u + 1) = 0$.

Namely, we can correctly attain the output by setting $\theta = F(d + 1)$, since $0 \leqslant d \leqslant l$. Second, Alice sets $[\![D]\!] = [\![d]\!] * [\![1]\!]$, i.e., $D = d + 1$, and $[\![w]\!] = [\![D]\!]^S$ where $S = R^{-1} \bmod n$, R is randomly selected from $\mathbb{Z}_n^*$, and n is the large integer in the public key. After that, $[\![w]\!]$ will be sent to Bob, who decrypts w, encrypts w^i, and returns the ciphertext $[\![w^i]\!]$ to Alice for $i = 2, 3, \cdots, u$. Finally, Alice can gain $[\![D^i]\!] = [\![w^i]\!]^{R^i}$ and $[\![F(D)]\!] = [\![\alpha_0]\!] * [\![D]\!]^{\alpha_1} * \prod_{i=2}^{u} [\![D^i]\!]$, which is exactly $[\![\theta]\!]$ because $\theta = F(d + 1) = F(D)$.

As can be seen, for a larger l, LT13 needs more computation and communication cost. While $u = 256$, LT13 uses tens of seconds [29], which is far away from being practical. In this paper, we will introduce a new PET scheme which can reduce the number of invoking Paillier encryption system and thus dramatically lessen running cost at the expense of small accuracy loss.

3. Privacy-Preserving Equality Test

Assume $r = (r_1, r_2, \cdots, r_u)$ is a uniform random vector from $\{1, -1\}^u$. For two binary strings $x = (x_1, x_2, \cdots, x_u)$ and $y = (y_1, y_2, \cdots, y_u)$, if setting $\widetilde{d} = \sum_{i=1}^{u} r_i (x_i - y_i)^2$, we have the following observations. Here, we use Δ_{xy} to denote the number of different bits of x and y. It is easy to say $0 \leqslant \Delta_{xy} \leqslant u$.

Observation 1. If $x = y$, then $\widetilde{d}$ always equals 0.

Proof. While $x = y$, each $x_i - y_i$ equals 0; thus $\widetilde{d} \equiv 0$. □

Observation 2. If $x \neq y$ and Δ_{xy} is odd, then it must be $\widetilde{d} \neq 0$.

Proof. Without loss of generality, we assume the first Δ_{xy} bits of x and y are different from each other. That is, $x_i \neq y_i$ for $i = 1$ to Δ_{xy}, and $x_i = y_i$ for $i = \Delta_{xy} + 1$ to u.

In this case, $(x_i - y_i)^2 = 1$ for $i = 1$ to Δ_{xy} and $(x_i - y_i)^2 = 0$ for $i = \Delta_{xy} + 1$ to u. Then, $\widetilde{d} = \sum_{i=1}^{\Delta_{xy}} r_i$, in which each $r_i \in \{1, -1\}$. Since Δ_{xy} is odd, the number of $r_i = 1$ is impossibly equal to that of $r_i = -1$.

Therefore, it must be $\widetilde{d} = \sum_{i=1}^{\Delta_{xy}} r_i \neq 0$, which completes the proof. □

Observation 3. If $x \neq y$ and Δ_{xy} is even, suppose $\Delta_{xy} = 2k$ (obviously, $0 < k \leqslant \lfloor u/2 \rfloor$), then $\widetilde{d} = 0$ with the probability $(2k)!/(4^k * (k!)^2)$, and correspondingly $\widetilde{d} \neq 0$ with the probability $(1 - (2k)!/(4^k * (k!)^2))$.

Proof. Without loss of generality, we assume that $x_i \neq y_i$ for $i = 1$ to $2k$ and $x_i = y_i$ for $i = 2k + 1$ to u. Then, $(x_i - y_i)^2 = 1$ for $i = 1$ to $2k$ and $(x_i - y_i)^2 = 0$ for $i = 2k + 1$ to u. Further, we have $\widetilde{d} = \sum_{i=1}^{2k} r_i$.

Let λ_1, λ_{-1} denote the number of $r_i = 1$, $r_i = -1$, respectively, for $i = 1$ to $2k$. Then, $\widetilde{d} = \lambda_1 - \lambda_{-1}$. Hence, $\widetilde{d} = 0$ iff $\lambda_1 = \lambda_{-1}$. As r_i is uniformly randomly selected from $\{1, -1\}$, the probability of $\lambda_1 = \lambda_{-1}$ is $\binom{2k}{k} / 2^{2k}$, where $\binom{2k}{k}$ denotes $2k$ choose k. Then, the probability of $\widetilde{d} \neq 0$ is $\binom{2k}{k} / 2^{2k} = (2k)!/(4^k * (k!)^2)$. Accordingly, the probability of $\widetilde{d} \neq 0$ is

$$1 - \frac{\binom{2k}{k}}{2^{2k}} = 1 - \frac{(2k)!}{4^k * (k!)^2}. \tag{8}$$

It completes the proof. □

Observations 1, 2, and 3 show we can approximatively determine $x = y$ by comparing $\widetilde{d}$ and 0. Besides, we have

$$\begin{aligned} \widetilde{d} &= \sum_{i=1}^{u} r_i (x_i - y_i)^2 = \sum_{i=1}^{u} r_i (x_i - (2x_i - 1) y_i) \\ &= \sum_{i=1}^{u} r_i x_i - \sum_{i=1}^{u} (r_i (2x_i - 1)) y_i, \end{aligned} \tag{9}$$

since $x_i^2 = x_i$ and $y_i^2 = y_i$. Then, we can get an approximative scheme for securely comparing x and y with high efficiency as follows.

Basic Approach. Alice selects u numbers $r_i \in \{1, -1\}$ uniformly at random and computes $A = \sum_{i=1}^{u} r_i x_i$. Then, Alice sets a u-bit binary vector $s = (s_1, s_2, \cdots, s_u)$ where $s_i = 1$ if $r_i(2x_i - 1) = 1$ and $s_i = 0$ otherwise and sends s to Bob. While receiving s, Bob can locally compute $B = \sum_{i=1}^{u} (2s_i - 1) y_i$. Finally, Alice and Bob utilize LT13 [30] to securely compare private numbers A and B such that Alice gains $[\![A = B]\!]$, i.e., Alice obtains $[\![1]\!]$ if $A = B$ and $[\![0]\!]$ otherwise.

As $x_i \in \{0, 1\}$, it has $(2x_i - 1) \in \{1, -1\}$ and $r_i(2x_i - 1) \in \{1, -1\}$. For the security Bob cannot learn any information about x_i from s, because r_i is uniformly randomly selected from $\{1, -1\}$. Since $2s_i - 1 = r_i(2x_i - 1)$, thus $B = \sum_{i=1}^{u} (2s_i - 1) y_i = \sum_{i=1}^{u} (r_i(2x_i - 1)) y_i$. Hence, $\widetilde{d} = 0$ iff $A = B$. That is, the basic scheme substantially determines $x = y$ by checking $\widetilde{d} = 0$. We will analyze accuracy of the basic approach in Theorem 2.

Due to $|A|, |B| \leqslant u$; thus A and B can be represented by using $\lceil \log_2 u \rceil + 1$ bits, in which $\lceil \log_2 u \rceil$ bits represent the value of $|A|, |B|$ and one bit is used to denote their sign. While $u \geqslant 4$, it always is $\lceil \log_2 u \rceil + 1 < u$. For example, when $u = 256$, we have $\lceil \log_2 u \rceil + 1 = 9$. Therefore, our basic scheme can dramatically reduce the running cost.

Theorem 2. *For Alice and Bob's binary strings $x = (x_1, x_2, \cdots, x_u)$ and $y = (y_1, y_2, \cdots, y_u)$, when $x \neq y$, let the probability of $\Delta_{xy} = i$ be q_i for $i = 1, 2, \cdots, u$, where $0 \leqslant q_i \leqslant 1$ and $\sum_{i=1}^{u} q_i = 1$. Namely, q_i denotes the condition probability $\Pr(\Delta_{xy} = i \mid x \neq y)$. For simplicity, suppose each q_i is identical, i.e., each $q_i = 1/u$. Then, for the basic scheme, we have*

(1) *if $x = y$, the basic approach always arrives at a correct result, i.e., Alice always gains $[\![1]\!]$.*

(2) *if $x \neq y$, the basic approach returns a false result (i.e., Alice gains $[\![1]\!]$) with the condition probability $\Pr(\widetilde{d} = 0 \mid x \neq y)$ in average, and correspondingly the basic scheme returns a correct result (i.e., Alice gains $[\![0]\!]$) with the probability $\Pr(\widetilde{d} \neq 0 \mid x \neq y) = 1 - \Pr(\widetilde{d} = 0 \mid x \neq y)$. Besides, it has*

$$\Pr\left(\widetilde{d} = 0 \mid x \neq y\right) = \sum_{i=1}^{\lfloor u/2 \rfloor} \frac{(2i)!}{u * 4^i * (i!)^2}. \tag{10}$$

To simplify, we use E_0 to denote the probability $\Pr(\widetilde{d} = 0 \mid x \neq y)$, i.e., $E_0 = \Pr(\widetilde{d} = 0 \mid x \neq y)$.

Proof. (1) If $x = y$, it always has $\widetilde{d} = 0$ according to Observation 1. As $\widetilde{d} = A - B$, then $A = B$ holds. Thus, Alice will gain $[\![1]\!]$, i.e., the basic scheme will get an exactly correct result.

(2) If $x \neq y$, the correct result is $[\![0]\!]$. Thus, the basic will correctly complete the comparison only when $A \neq B$. According to (9), we have $A \neq B$ iff $\widetilde{d} \neq 0$. Hence, in this situation, the probability that the basic scheme returns a correct result equals the condition probability $\Pr(\widetilde{d} \neq 0 \mid x \neq y)$. Correspondingly, the basic scheme returns a false

result with the probability $\Pr(\widetilde{d} = 0 \mid x \neq y)$. Since Δ_{xy} may be 1 to u while $x \neq y$, then $\Pr(\widetilde{d} = 0 \mid x \neq y) = \sum_{i=1}^{u} \Pr(\widetilde{d} = 0 \mid \Delta_{xy} = i)\Pr(\Delta_{xy} = i \mid x \neq y)$. On account of Observation 2, if Δ_{xy} is odd, it always has $\widetilde{d} \neq 0$, i.e., the probability $\Pr(\widetilde{d} = 0 \mid \Delta_{xy} = i) = 0$ for each odd i. Besides each $\Pr(\Delta_{xy} = i \mid x \neq y) = 1/u$; therefore,

$$\begin{aligned}
&\Pr\left(\widetilde{d} = 0 \mid x \neq y\right) \\
&= \sum_{i=1}^{\lfloor u/2 \rfloor} \Pr\left(\widetilde{d} = 0 \mid \Delta_{xy} = 2i\right) \Pr\left(\Delta_{xy} = 2i \mid x \neq y\right) \\
&= \sum_{i=1}^{\lfloor u/2 \rfloor} \frac{\Pr\left(\widetilde{d} = 0 \mid \Delta_{xy} = 2i\right)}{u}.
\end{aligned} \tag{11}$$

Observation 3 has shown $\Pr(\widetilde{d} = 0 \mid \Delta_{xy} = 2i) = (2i)!/(4^i * (i!)^2)$. As a result,

$$\Pr\left(\widetilde{d} = 0 \mid x \neq y\right) = \sum_{i=1}^{\lfloor u/2 \rfloor} \frac{(2i)!}{u * 4^i * (i!)^2}, \tag{12}$$

which completes the proof. □

Theorem 3. *Let the functions $g()$, $f()$ be*

$$\begin{aligned}
g(k) &= \frac{(2k)!}{4^k * (k!)^2}, \\
f(u) &= \sum_{i=1}^{\lfloor u/2 \rfloor} \frac{(2i)!}{u * 4^i * (i!)^2},
\end{aligned} \tag{13}$$

where $k \geqslant 1$, $u \geqslant 16$, and i and u are integers. We have

(1) *$g(k) > g(k+1)$ for any $i \geqslant 1$,*
(2) *if u is even, then $f(u) > f(u+1)$ and $f(u) > f(u+2)$,*
(3) *if u is odd, then $f(u) > f(u+2)$.*

Proof. (1) According to the setting, we have

$$\begin{aligned}
g(k+1) - g(k) &= \frac{(2k+2)!}{4^{k+1} * ((k+1)!)^2} - \frac{(2k)!}{4^k * (k!)^2} \\
&= \frac{(2k+2)! - (2k)! * 4 * (k+1)^2}{4^{k+1} * ((k+1)!)^2} \\
&= \frac{(2k)! * \left((2k+2)(2k+1) - (2k+2)^2\right)}{4^{k+1} * ((k+1)!)^2}
\end{aligned} \tag{14}$$

Then,

$$g(k+1) - g(k) = \frac{(2k)! * (2k+2) * (-1)}{4^{k+1} * ((k+1)!)^2} < 0, \tag{15}$$

Therefore, $g(k) > g(k+1)$ holds.
(2) It is easy to say

$$f(u) = \sum_{i=1}^{\lfloor u/2 \rfloor} \frac{(2i)!}{u * 4^i * (i!)^2} = \sum_{i=1}^{\lfloor u/2 \rfloor} \frac{g(i)}{u}. \tag{16}$$

If u is even, assume $u = 2 * v$, then $v = \lfloor u/2 \rfloor = \lfloor (u+1)/2 \rfloor$ and $v + 1 = \lfloor (u+2)/2 \rfloor$. Hence,

$$\begin{aligned}
f(u+1) - f(u) &= \sum_{i=1}^{v} \frac{g(i)}{u+1} - \sum_{i=1}^{v} \frac{g(i)}{u} = \sum_{i=1}^{v} \frac{-g(i)}{u(u+1)} \\
&< 0.
\end{aligned} \tag{17}$$

Besides,

$$\begin{aligned}
f(u+2) - f(u) &= \sum_{i=1}^{v+1} \frac{g(i)}{u+2} - \sum_{i=1}^{v} \frac{g(i)}{u} \\
&= \frac{1}{2}\left(\sum_{i=1}^{v+1} \frac{g(i)}{v+1} - \sum_{i=1}^{v} \frac{g(i)}{v}\right) \\
&= \frac{1}{2}\left(\frac{g(v+1)}{v+1} - \sum_{i=1}^{v} \frac{g(i)}{v(v+1)}\right) \\
&= \frac{1}{2v(v+1)}\left(v * g(v+1) - \sum_{i=1}^{v} g(i)\right).
\end{aligned} \tag{18}$$

Since $g(k) > g(k+1)$, we have $v * g(v+1) - \sum_{i=1}^{v} g(i) < 0$. Thus, $f(u+2) - f(u) < 0$.

As a result, $f(u) > f(u+1)$ and $f(u) > f(u+2)$ both are proved.

(3) If u is odd, assume $u = 2 * v + 1$, then $v = \lfloor u/2 \rfloor$ and $v + 1 = \lfloor (u+2)/2 \rfloor$. Then,

$$f(u+2) - f(u) = \sum_{i=1}^{v+1} \frac{g(i)}{u+2} - \sum_{i=1}^{v} \frac{g(i)}{u}. \tag{19}$$

Since $(u+2)/(u+1) < u/(u-1)$, we have

$$\begin{aligned}
f(u+2) - f(u) < \frac{u+2}{u+1} * \sum_{i=1}^{v+1} \frac{g(i)}{u+2} - \frac{u}{u-1} \\
* \sum_{i=1}^{v} \frac{g(i)}{u}.
\end{aligned} \tag{20}$$

That is,

$$f(u+2) - f(u) < \sum_{i=1}^{v+1} \frac{g(i)}{u+1} - \sum_{i=1}^{v} \frac{g(i)}{u-1}. \tag{21}$$

Because $v = \lfloor (u-1)/2 \rfloor$ and $v+1 = \lfloor (u+1)/2 \rfloor$, it has $f(u+1) = \sum_{i=1}^{v+1} (g(i)/(u+1))$ and $f(u-1) = \sum_{i=1}^{v} (g(i)/(u-1))$. Besides, $u-1$ is an even integer. We have proved $f(u-1) > f(u-1+2) = f(u+1)$. Thus,

$$f(u+2) - f(u) < f(u+1) - f(u-1) < 0. \tag{22}$$

Consequently, $f(u) > f(u+2)$ is correct. It completes the proof. □

As we can see, $\Pr(\widetilde{d} = 0 \mid x \neq y) = f(u)$. Theorem 3 shows that $\Pr(\widetilde{d} = 0 \mid x \neq y)$ has a downward trend, as u increases. While $u = 16$, the error probability $E_0 = \Pr(\widetilde{d} = 0 \mid x \neq y) = 0.146$ which may be too high for real applications.

Input: Alice privately holds a u-bit binary string $x = (x_1, x_2, \cdots, x_u)$, and Bob holds a private u-bit binary string $y = (y_1, y_2, \cdots, y_u)$ and the public key pair (pk, sk) of Paillier encryption scheme where pk and sk are public key and private key, respectively.
Output: Alice obtains $[\![\theta]\!]$ where $\theta = (x = y)?1 : 0$, and Bob learns nothing.

1: Alice generates two random u-dimension vectors $r, R \in \{1, -1\}^u$. For each $i = 1$, let r_i and R_i denote the i-th dimension of r and R, respectively.
2: Alice computes two u-dimension binary vectors $s, S \in \{0, 1\}^u$ such that their i-th bits s_i and S_i satisfy (1) $s_i = 1$ if $r_i(2x_i - 1) = 1$, otherwise $s_i = 0$. (2) $S_i = 1$ if $R_i(2x_i - 1) = 1$, otherwise $S_i = 0$. Alice sends s and S to Bob.
3: Alice computes $a = \sum_{i=1}^{u} r_i x_i$ and $A = \sum_{i=1}^{u} R_i x_i$. Then, Alice utilizes $\lceil \log_2 u \rceil + 1$ bits to represent a, A, respectively. The first bit denotes the sign where 0 denotes positive and 1 denotes negative, and the latter $\lceil \log_2 u \rceil$ bits represents the their absolute value. Let $t = 2(\lceil \log_2 u \rceil + 1)$. We use $x' = (x'_1, x'_2, \cdots, x'_t)$ to denote the $2(\lceil \log_2 u \rceil + 1)$ bits of a and A, in which the first $\lceil \log_2 u \rceil + 1$ bits correspond to a, and the latter $\lceil \log_2 u \rceil + 1$ bits correspond to A.
4: Bob computes $b = \sum_{i=1}^{u} (2s_i - 1) y_i$ and $B = \sum_{i=1}^{u} (2S_i - 1) y_i$. Then, Bob utilizes $\lceil \log_2 u \rceil + 1$ bits to represent b, B, respectively. The first bit denotes the sign where 0 denotes positive and 1 denotes negative, and the latter $\lceil \log_2 u \rceil$ bits represents the their absolute value. Similarly, $y' = (y'_1, y'_2, \cdots, y'_t)$ is used to denote the $2(\lceil \log_2 u \rceil + 1)$ bits of b and B, in which the first $\lceil \log_2 u \rceil + 1$ bits correspond to b, and the latter $\lceil \log_2 u \rceil + 1$ bits correspond to B.
5: For $i = 1$ to t, Bob uses his public key pk to encrypt each private bit y'_i, and sends $[\![y'_i]\!]$ to Alice.
6: Alice computes encrypted Hamming distance $[\![d]\!] = \prod_{i=1}^{t} [\![x'_i \oplus y'_i]\!]$, where $[\![x'_i \oplus y'_i]\!] = [\![1]\!] * [\![y'_i]\!]^{-1}$ if $x'_i = 1$ and $[\![x'_i \oplus y'_i]\!] = [\![y'_i]\!]$ if $x'_i = 0$.
7: Alice and Bob select a t-degree public Lagrange interpolation polynomial $F(z) = \sum_{i=0}^{t} \alpha_i z^i \bmod n$ in which n is the large integer in the public key, such that $F(z)$ satisfies $F(1) = 1, F(2) = F(3) = \cdots = F(t + 1) = 0$. Namely, we can correctly attain the output by setting $\theta = F(d + 1)$, since $0 \leqslant d \leqslant l$.
8: Alice sets $[\![D]\!] = [\![d]\!] * [\![1]\!]$, i.e. $D = d + 1$, and $[\![w]\!] = [\![D]\!]^{\lambda}$ where $\lambda = R^{-1} \bmod n$ and R is randomly selected from $\mathbb{Z}_n^*$. After that, Alice sends $[\![w]\!]$ to Bob.
9: Bob decrypts w, encrypts w^i, and returns the ciphertext $[\![w^i]\!]$ to Alice for $i = 2, 3, \cdots, t$.
10: Alice computes $[\![D^i]\!] = [\![w^i]\!]^{R^i}$, and further gains the final output $[\![\theta]\!] = [\![F(D)]\!] = [\![\alpha_0]\!] * [\![D]\!]^{\alpha_1} * \prod_{i=2}^{t} [\![D^i]\!]$.

Protocol 1: FTP: approximate Fast privacy-preserving equality Test Protocol.

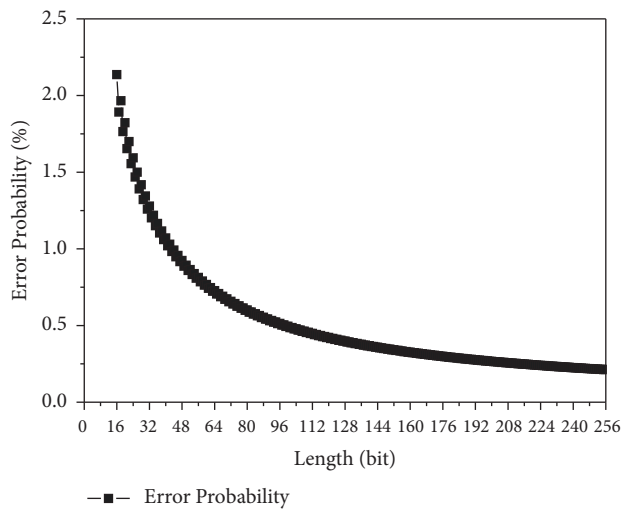

Figure 1: E_0^2: the error probability of using double A and B, while x and y are inequal.

We can reduce the probability by generating multiple A and B with different random vector r, which can exponentially reduce the error probability. For example, if we use double A and B, then the error probability will be E_0^2. Even when $u = 16$, the error probability will be 0.021 around. Figure 1 shows the error probability while u ranges from 16 to 256. It indicates our error probability will be smaller than 1% while the bit length is larger than 42. When using our scheme to compare two 256-bit binary strings, the error probability will be about 0.21% only.

In general, the details of our scheme with double A and B are formally shown in Protocol 1. First, Alice randomly generates $r, R \in \{1, -1\}^u$ and shares $r_i(2x_i - 1), R_i(2x_i - 1)$ with Bob. Second, Alice locally computes $a = \sum_{i=1}^{u} r_i x_i$ and $A = \sum_{i=1}^{u} R_i x_i$, and Bob gains $b = \sum_{i=1}^{u} r_i(2x_i - 1) y_i$ and $B = \sum_{i=1}^{u} R_i(2x_i - 1) y_i$. They decide $x = y$ iff $a = b$ and $A = B$. Third, they use $2(\lceil \log_2 u \rceil + 1)$-bit x' and y' to represent (a, A) and (b, B), respectively. Finally, Alice and Bob utilize the similar methods of LT13 to securely compare x' and y' such that Alice gains $[\![x' = y']\!]$.

4. Analysis Evaluation

4.1. Security. We prove the security of our proposed scheme FTP through the following Theorem 4.

Theorem 4. *Our proposed scheme FTP discloses nothing useful about the privacy of input values and the final result.*

Proof. We will discuss the view of Alice and Bob, respectively.

In our scheme FTP, Alice receives $[\![y'_i]\!]$, $[\![w^i]\!]$ for $i = 1, 2 \cdots, 2(\lceil \log_2 u \rceil + 1)$. Based on IND-CPA security of Paillier encryption system [31], Alice can learn nothing useful about y'_i and w^i. Thus, Bob's private data can be securely preserved.

Throughout FTP, Bob learns just s, S, and w. For each bit s_i in s, it has $s_i = 1$ if $r_i(2x_i - 1) = 1$; otherwise $s_i = 0$. That

is, $(2x_i - 1) = (2s_i - 1)r_i$. Each $r_i \in \{1, -1\}$ is unknown to Bob, and we can simply assume $\Pr(r_i = 1) = \Pr(r_i = 0) = 0.5$ for the view of Bob. Hence, for any s_i, conditional probability $\Pr(x_i = 1 \mid s_i) = \Pr(x_i = 0 \mid s_i) = 0.5$, which means Bob can learn nothing about x_i from s_i. Similarly, it is provable that S discloses nothing about x_i. For w, based on the additive homomorphic property, we have $w = D * R^{-1} \bmod n$. As R is randomly selected from $\mathbb{Z}_n^*$, Bob can infer no information about D from w. In general, s, S, and w reveal nothing of Alice's private data.

To sum up, the privacy of Alice and Bob both can be preserved in our scheme FTP, which completes the proof. □

4.2. Computation and Communication Cost. In this section, we will analyze the computation complexity and communication overheads of our proposed FTP in detail.

Computation Complexity. Since simple addition and multiplication are much cheaper than encryption, decryption, and ciphertext multiplication of Paillier cryptosystem, we will ignore the simple addition and multiplication in the protocol. Throughout FTP, Bob encrypts each y_i' and w^i for $i = 1$ to $2(\lceil \log_2 u \rceil + 1)$ and decrypts one times to gain w. Alice uses $[\![y_i']\!]$ and $[\![w^i]\!]$ to compute $[\![D]\!]$ and $[\![\theta]\!]$, which requires ciphertext multiplication $2(\lceil \log_2 u \rceil + 1)$ times. In total, both Bob and Alice just employ Paillier encryption system $O(\log u)$ times.

Communication Overheads. In our scheme FTP, Alice and Bob need to transmit s, S, $[\![y_i']\!]$ and $[\![w^i]\!]$ for $i = 1$ to $2(\lceil \log_2 u \rceil + 1)$. If each ciphertext is β-bit, then the total communication overheads are $2u + 2\beta(\lceil \log_2 u \rceil + 1)$. While $u = 256$ and we set the public key of Paillier encryption system to be 2048 bits, the communication overheads will be 37376 bits.

4.3. Experiment Results. We implement our scheme and two existing efficient algorithms: LT13 and NEL16, using C language. During executing our scheme, we utilize GMP library [33] and Paillier library [34] with key size of 2048 bits. All experiments are performed on an Apple computer with macOS Sierra 10.12.6, Intel Core i5 1.6GHz CPU and 4 GB memory. Alice and Bob communicate through the socket where ping time is about 0.81 seconds.

Figure 2 shows the runtime of LT13, NEL16, and our scheme FTP while the compared string is of 16 to 256 bits. As can be seen, FTP can dramatically reduce the running time compared to LT13 and NEL16. When the length u is 256, LT13 costs about 25 seconds, NEL16 takes 6 seconds around, and FTP just needs 0.6 seconds. While the length is larger, the advantage of FTP will be more salient. The main reason is that we transform the original x, y into x', y' which is much shorter than the original ones. More importantly, our transformation just involves simple addition and multiplication and can be completed rapidly. In FTP, Paillier encryption system is employed only to securely compare x' and y'. Therefore, FTP can reduce the running cost, especially when u is large. If the bit length is smaller than 16, FTP has no significant advantages on running time, and LT13 or NEL16 is suitable for the short-string equality comparison scenario.

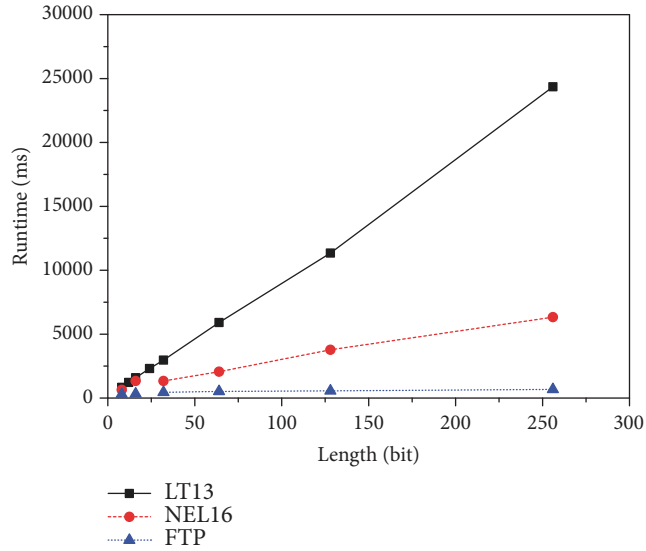

Figure 2: Runtime comparison of LT13, NEL16, and our scheme FTP.

4.4. Improvement. Though our scheme FTP can reduce the cost, it still takes u bits to transmit the vector s or S. We can further improve the scheme to avoid transmitting s or S. Let $H() : \{0,1\}^* \longrightarrow \{0,1\}^u$ be a pseudorandom function. Alice and Bob, in advance, select a constant pad. While they decide to compare the private binary vectors, they can separately generate a random binary string $H(time \parallel pad)$ where $time$ denotes the time they decide to implement the protocol and $\parallel$ denotes concatenation. Then, they set $s_i = H_i$ in which H_i denotes the i-th bit of $H(time \parallel pad)$. Since $s_i = (r_i(2x_i - 1) + 1)/2$ and $(2x_i - 1)^2 = 1$, Alice can locally get $r_i = (2H_i - 1)(2x_i - 1)$. Thus, Alice and Bob can compute A and B, respectively. By this method, Alice need not to send the vector s again. For S, they can preestablish another constant pad' and use it avoid transmitting S by a similar method.

5. Related Work

Privacy-preserving string equality test is one of secure multiparty computation (SMC) problems, and it has wide applications in various privacy-preserving scenes [35–38]. Up to now, a big number of works can be utilized to achieve privacy-preserving string equality test. We simply discuss the previous schemes as follows.

In 1982, Yao [39] proposes the first SMC problem, Millionaire problem and gives a secure solution. After that, garbled circuits method [32, 40] is put forward to securely evaluate a general function. Nevertheless, the general approach is too expensive and can just theoretically solve the problem. Scalar product protocol (also known as dot product protocol) focuses on computing the scalar product of two private vectors with privacy-preservation. Privacy-preserving string equality test can be achieved by invoking scalar product protocol. We thus review the main solutions of scalar product protocol. In [41], Vaidya et al. proposed a scalar product protocol based on algebraic transformation. By

using homomorphic encryption, two solutions for securely computing dot product of private vectors are given in [42] and [43], respectively. A polynomial secret sharing-based scalar product protocol is presented by Shaneck and Kim [44]. Nevertheless, the schemes either are not provably secure or have heavy computation and communication overheads. Recently, Zhu et al. propose two efficient solutions for secure scalar product protocol [45, 46], which can be utilized to securely compute the Hamming distance of two private strings but cannot support the distance comparison. Cheng et al. [47] review the approaches to secure Internet of Things in a quantum world. In [48], Li et al. leverage Paillier encryption to achieve secure comparison protocol, based on which they also propose a secure SVM classification scheme. Nevertheless, the comparison scheme in [48] focuses on securely figuring out the bigger one from two private integers but cannot directly support the equality comparison problem investigated in this paper.

In [30], Lipmaa and Toft propose a secure string equality test scheme based on Paillier encryption scheme [31]. While comparing u-bit strings, Lipmaa and Toft's scheme requires $O(u)$ encryption of Paillier encryption system and thus is time-consuming. Nateghizad et al. [29] improve Lipmaa and Toft's scheme by reducing the degree of Lagrange interpolation polynomial. As yet, the number of invoking Paillier encryption in Nateghizad et al.'s solution is also linear with u, which is not suitable for a large u either. In general, the existing privacy-preserving string equality test schemes are still far away from being practical.

6. Conclusions

In this paper, we considered efficient and privacy-preserving authentication in IoT applications. To this end, we proposed a new privacy-preserving equality test protocol, which can securely complete string equality test and achieve high running efficiency at the cost of little accuracy loss. We strictly analyzed the accuracy of our proposed scheme and formally proved our security. Additionally, we leveraged extensive simulation experiments to evaluate the running cost, which confirms our high efficiency.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by the National Key Research and Development Program of China (no. 2017YFB0802300), the Natural Science Foundation of China (no. 61602240), the Natural Science Foundation of Jiangsu Province of China (no. BK20150760), Research Fund of Guangxi Key Laboratory of Cryptography and Information Security (no. GCIS201723), and Postgraduate Research & Practice Innovation Program of Jiangsu Province (no. KYCX18_0305).

References

[1] R. Cramer, "Introduction to secure computation," in *Lectures on Data Security*, pp. 16–62, 1999.

[2] Y. Zhu, Z. Huang, and T. Takagi, "Secure and controllable k-NN query over encrypted cloud data with key confidentiality," *Journal of Parallel and Distributed Computing*, vol. 89, pp. 1–12, 2016.

[3] T. R. Hoens, M. Blanton, A. Steele, and N. V. Chawla, "Reliable medical recommendation systems with patient privacy," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, p. 67, 2013.

[4] J. Shi, C. Chen, and S. Zhong, "Privacy preserving growing neural gas over arbitrarily partitioned data," *Neurocomputing*, vol. 144, pp. 427–435, 2014.

[5] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[6] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, and Z.-H. Zhan, "Secure data uploading scheme for a smart home system," *Information Sciences*, vol. 453, pp. 186–197, 2018.

[7] J. Li, X. Chen, S. S. Chow, Q. Huang, D. S. Wong, and Z. Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," *Journal of Network and Computer Applications*, vol. 112, pp. 89–96, 2018.

[8] W. Chen, Z. Chen, N. F. Samatova, L. Peng, J. Wang, and M. Tang, "Solving the maximum duo-preservation string mapping problem with linear programming," *Theoretical Computer Science*, vol. 530, pp. 1–11, 2014.

[9] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Gao, "Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures," *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[10] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[11] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, "A Covert Channel Over VoLTE via Adjusting Silence Periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018.

[12] L. Fan, X. Lei, N. Yang, T. Q. Duong, and G. K. Karagiannidis, "Secure Multiple Amplify-and-Forward Relaying with Cochannel Interference," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 8, pp. 1494–1505, 2016.

[13] S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data," in *Proceedings of the 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS '13)*, pp. 1–8, 2013.

[14] Y. Liu, C. Cheng, T. Gu, T. Jiang, and X. Li, "A Lightweight Authenticated Communication Scheme for Smart Grid," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 836–842, 2016.

[15] Q. Lin, J. Li, Z. Huang, W. Chen, and J. Shen, "A short linearly homomorphic proxy signature scheme," *IEEE Access*, vol. 6, pp. 12966–12972, 2018.

[16] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 190–209, 2011.

[17] Y. Luo, *Efficient Anonymous Biometric Matching in Privacy-Aware Environments*, University of Kentucky, 2014.

[18] J. Li, Q. Lin, C. Yu, X. Ren, and P. Li, "A QDCT- and SVD-based color image watermarking scheme using an optimized encrypted binary computer-generated hologram," *Soft Computing*, vol. 22, no. 1, pp. 47–65, 2018.

[19] C. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack," *Information Sciences*, vol. 444, pp. 72–88, 2018.

[20] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018.

[21] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN Hybrid Method for Short Utterance Speaker Recognition," *IEEE Transactions on Industrial Informatics*, vol. 99, pp. 1–8, 2018.

[22] S. Zhong and Y. Zhang, "How to select optimal gateway in multi-domain wireless networks: Alternative solutions without learning," *IEEE Transactions on Wireless Communications*, vol. 12, no. 11, pp. 5620–5630, 2013.

[23] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient Merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655–663, 2014.

[24] K. Xue, Y. Xue, J. Hong et al., "RAAC: Robust and Auditable Access Control with Multiple Attribute Authorities for Public Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 953–967, 2017.

[25] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[26] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, 2018.

[27] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 99, pp. 97–109, 2016.

[28] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, and P. Hong, "Two-Cloud Secure Database for Numeric-Related SQL Range Queries with Privacy Preserving," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1596–1608, 2017.

[29] M. Nateghizad, Z. Erkin, and R. L. Lagendijk, "Efficient and secure equality tests," in *Proceedings of the 8th IEEE International Workshop on Information Forensics and Security, (WIFS '16)*, pp. 1–6, IEEE, 2016.

[30] H. Lipmaa and T. Toft, "Secure equality and greater-than tests with sublinear online complexity," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pp. 645–656, Springer, Riga, Latvia, 2013.

[31] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '99)*, vol. 99, pp. 223–238, Springer, 1999.

[32] O. Goldreich, *Foundations of Cryptography*, vol. 2, Cambridge University Press, 2009.

[33] "The GNU multiple precision arithmetic library," http://gmplib.org/.

[34] Paillier library, http://acsc.cs.utexas.edu/libpaillier/.

[35] H. Li, Y. Yang, Y. Dai, J. Bai, S. Yu, and Y. Xiang, "Achieving Secure and Efficient Dynamic Searchable Symmetric Encryption over Medical Cloud Data," *IEEE Transactions on Cloud Computing*, 2017.

[36] Y. Zhu, Y. Zhang, X. Li, H. Yan, and J. Li, "Improved collusion-resisting secure nearest neighbor query over encrypted data in cloud," *Concurrency and Computation: Practice and Experience*, Article ID e4681, 2018.

[37] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 127–139, 2015.

[38] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in Internet of Things with Privacy Preserving: Challenges, Solutions and Opportunities," *IEEE Network*, pp. 1–8, 2018.

[39] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pp. 160–164, 1982.

[40] A. C.-C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th Annual Symposium on Foundations of Computer Science (FOCS '86)*, pp. 162–167, Toronto, Canada, 1986.

[41] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 639–644, 2002.

[42] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, "On private scalar product computation for privacy-preserving data mining," in *Proceedings of the 7th International Conference on Information Security and Cryptology*, vol. 3506 of *Lecture Notes in Computer science*, pp. 104–120, 2004.

[43] A. Amirbekyan and V. Estivill-Castro, "A new efficient privacy-preserving scalar product protocol," in *Proceedings of the Sixth Australasian Conference on Data Mining and Analytics*, vol. 70, pp. 209–214, 2007.

[44] M. Shaneck and Y. Kim, "Efficient cryptographic primitives for private data mining," in *Proceedings of the 43rd Annual Hawaii International Conference on System Sciences*, pp. 1–9, 2010.

[45] Z. Youwen, T. Tsuyoshi, and H. Liusheng, "Efficient secure primitive for privacy preserving distributed computations," in *Proceedings of the 7th International Workshop on Security (IWSEC '12)*, Lecture Notes in Computer Science, pp. 233–243, 2012.

[46] Y. Zhu, Z. Wang, B. Hassan, Y. Zhang, J. Wang, and C. Qian, "Fast Secure Scalar Product Protocol with (almost) Optimal Efficiency," in *Proceedings of the 11th EAI International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '15)*, pp. 234–242, 2015.

[47] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, "Securing the Internet of Things in a Quantum World," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 116–120, 2017.

[48] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the Soundness and Security of Privacy-Preserving SVM for Outsourcing Data Classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.