JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 953-964 (2017)
DOI: 10.6688/JISE.2017.33.4.6

Efficient Encrypted Data Comparison Through a Hybrid Method*

XING-XIN LI, YOU-WEN ZHU+ AND JIAN WANG
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing, 210016 P.R. China
E-mail: lixingxin93@163.com; {zhuyw; wangjian}@nuaa.edu.cn

With the development of smart city, user data are all transformed in encrypted form to protect user privacy. Secure comparison in encrypted form is the fundamental operation of many secure encrypted data analysis tasks, such as secure *k*-NN query and classification, Bayesian classification. Thus, it is important to achieve an efficient secure encrypted data comparison scheme. Recently, some methods have been put forward to support secure comparison over encrypted data. Nevertheless, the existing solutions are still inefficient in practical. In this paper, we propose a novel encrypted data comparison protocol based on a hybrid approach of Paillier cryptosystem and garbled circuits. Our scheme reveals nothing about encrypted data and comparison result, and is provably secure under semi-honest model. Additionally, our proposed protocol can achieve higher efficiency, compared with the state-of-the-art scheme. Finally, we indicate the security and efficiency of our scheme by theoretical analysis and experiment evaluations.

Keywords: secure comparison, cloud computing, homomorphic cryptosystem, garbled circuits, privacy preserving

1. INTRODUCTION

Smart city [26] aims to provide more efficient, sustainable, competitive, productive, open and transparent place to live. Recently, smart city has gained much attention both in academia and business. Smart city uses information and communication technologies to enhance all aspects of the city, from local economy, transport and traffic management to quality of citizen life and e-governance [26, 27]. Cloud computing plays an important role in smart city. Most of services offered in smart city are based on cloud computing. Users can interact with these services through heterogeneous networks [30, 31], which will disclose many user data to cloud servers or attackers. User data contain sensitive private information and thus should not be revealed to cloud server. Otherwise, citizens might refrain from using smart city services. For example, user may provide his phone number, personal preferences to server in recommender system [18]. If the server obtains the private information, it can infer more sensitive information about user based on some priori knowledges. Encrypting data can effectively protect the privacy of sensitive data. However, encryption makes it difficult to implement statistical data analysis over the user data, which is the essential task in many data mining applications. How to complete the data analysis task without hampering data privacy is considered as a big challenge [6].

Since smart city collates significant amounts of data about their citizens, it is neces-

Received July 2, 2016; revised August 5, 2016; accepted August 30, 2016.
Communicated by Zhe Liu.
* This work is supported by the Fundamental Research Funds for the Central Universities (No. NS2016094).
+ Corresponding author.

sary to develop some efficient protocols to handle these datasets. Especially if the dataset is in encrypted form, this work becomes more difficult. Many works have been done to solve this problem. Some works focus on secure query over outsourced database. For example, Elmehdwi *et al.* [4] construct secure *k*-NN query scheme over encrypted dataset based on Paillier homomorphic encryption. Zhu *et al.* propose secure *k*-NN query [22] with a novel combination of random matrix transformation and additively homomorphic encryption. Xu *et al.* [28] propose secure *k*-NN query scheme over outsourced database based on oblivious RAM. Besides, Liu *et al.* [23-25] propose some protocols for the efficient implementation of public key encryption. Ren *et al.* [29] construct an efficient mutual verifiable provable data possession scheme to protect outsourced data integrity. In this paper, we focus on one of fundamental operations in analyzing encrypted data, secure comparison. For example, secure *k*-NN query [4, 22], secure multi-keyword ranked search [19-21] all need to invoke secure comparison to complete each computing tasks. Secure comparison takes two encrypted integers as input and outputs encrypted comparison result without allowing users to learn anything about the plaintext hidden in the encrypted input and the comparison result.

So far, researchers have proposed various methods to solve the problem of secure comparison over encrypted data. Existing secure comparison works are based on homomorphic cryptosystem and garbled circuits. Recently, in [17], Samanthula *et al.* use secure bit-decomposition [16] to convert encrypted integer into encryptions of bits, and then propose a novel secure bitwise comparison protocol based on Paillier cryptosystem to compare these encryptions of bits. Though their protocol can securely compare encrypted data, it involves a large amount of time-consuming computations in secure bit-decomposition and bitwise comparison, which makes the protocol computationally inefficient. For example, it needs $3l$ encryptions, l+1 decryptions to convert encrypted integer into encryptions of bits and $14l$ encryptions, $7l$ decryptions to securely compare these encryptions of bits, when comparing two encrypted l-bit values. Thus, in this paper, we focus on the problem of secure comparison without bit-decomposition. In [5], Erkin *et al.* propose a secure comparison protocol without bit decomposition (denoted as SC_h). Their work can just complete comparison between two encrypted values using bitwise comparison protocol. Note that bitwise comparison protocol still takes many time-consuming operations on encrypted data, we prefer to overcome this weakness on efficiency by using Yao's garbled circuits. Yao's garbled circuits provide an efficient solution for comparing two private data without leaking each private data. Many works have been proposed to compare encrypted data using comparison garbled circuits, such as [17], denoted as SC_g. Through the analysis in our paper, however, we find that SC_g is still time consuming. As secure comparison needs to be performed frequently when analyzing encrypted data, more efficient protocols for secure comparison over encrypted data are required.

To overcome the weakness of existing schemes, we propose a novel secure comparison protocol SCED based on a hybrid approach of homomorphic encryption and garbled circuits in this paper. Our main contributions in this paper are as follows.

1. We present SCED, a secure comparison protocol over encrypted data. Our SCED needs less encryptions, decryptions and smaller garbled circuits compared to existing works based on homomorphic encryption or garbled circuits.

2. We theoretically prove our protocol is secure under semi-honest model. Besides, we analyze the complexity of our protocol, which indicates that our scheme is efficient than existing works as shown in section 6.
3. We conduct experiments to evaluate the performance of our protocol and compare it with existing works. The experiment results are consistent with our theoretical analysis.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, we introduce Paillier cryptosystem and Yao's garbled circuits, which will be used in our protocol. Section 4 introduces our system model and framework. Section 5 proposes our new efficient comparison protocol SCED. Section 6 provides security and complexity analysis of our SCED. Section 7 utilizes experiments to evaluate our execution cost and compare it with existing works. At last, Section 8 concludes the paper.

2. RELATED WORK

Many protocols have been proposed to implement secure comparison over encrypted data. As mentioned in Section 1, these existing protocols are based on homomorphic cryptosystem or garbled circuits.

2.1 Secure Comparison based on Homomorphic Cryptosystem

Damgard *et al.* [3] propose a protocol for secure comparison of integers based on homomorphic encryption. In their work, they first propose a homomorphic encryption scheme (called DGK encryption). Then they propose a bitwise comparison protocol based on DGK encryption. Garay *et al.* [7] propose two new solutions to the secure integer comparison problem by using threshold homomorphic cryptosystem. In [17], Samanthula *et al.* propose a novel bitwise comparison protocol based on Paillier encryption. However, the inputs of these comparison protocols are all encryptions of individual bits rather than simple encrypted integers. Secure bit-decomposition protocol [2] can convert an encrypted integer into encryptions of individual bits, making it possible to compare two encrypted integers through secure bitwise comparison protocol. However, secure bit-decomposition protocol involves expensive computations, which makes the whole secure comparison of encrypted integers computationally inefficient.

Nishide *et al.* [14] construct an efficient comparison protocol without depending on the bit-composition protocol. Although their comparison protocol is based on secret sharing, it can also be used to achieve comparison of encrypted data. Their comparison protocol is efficient than the comparison protocol relying on bit-decomposition. Erkin *et al.* propose a comparison protocol without bit-composition in [5] using DGK encryption and bitwise comparison protocol proposed in [3]. However, secure bitwise comparison protocol still takes too much expensive computation.

2.2 Secure Comparison based on Garbled Circuits

Yao's garbled circuits are very efficient in processing integer comparison. In [1].

Chun *et al.* present a secure comparison protocol by using add and comparison circuits. They first share encrypted data between two parties, and then use garbled circuits to obtain the comparison result with a random value. Finally, the protocol outputs encrypted comparison result using the homomorphic property of Paillier encryption. Liu *et al.* [13] propose a secure minimum protocol selecting minimum from multiple encrypted values. Their approach is based on homomorphic encryption and Yao's garbled circuits, yielding one to two orders of magnitude improvement in running time compared to existing works based on homomorphic encryption. In this paper, we find their protocol involves lager garbled circuits, such that their protocol is still time consuming.

3. PRELIMINARIES

3.1 Paillier Cryptosystem

Paillier cryptosystem [15] is an additively homomorphic public key encryption algorithm. The main steps involved in Paillier cryptosystem include:

Key Generation: Select two large prime numbers p, q and a generator g of $Z^*_{N^2}$. The public key is $pk = (N, g)$, where $N = p*q$ and the secret key is $sk = (p, q)$. Here, $L(x) = (x - 1)/N$.

Encryption: Let m be a number in plaintext space Z_N. Select a random r from Z^*_N, the ciphertext of m is

$$E_{pk}(m) = g^m \times r^N \bmod N^2. \tag{1}$$

Decryption: Given a cipheretext $E_{pk}(m)$, using the secret key sk, compute $\varphi = lcm(p-1, q-1)$ and the plaintext hidden in $E_{pk}(m)$ is

$$m = \frac{L(E_{pk}(m)^{\varphi} \bmod N^2)}{L(g^{\varphi} \bmod N^2)} (\bmod N). \tag{2}$$

As a probabilistic homomorphic encryption, Paillier cryptosystem has following properties:

Homomorphic addition: Given two ciphertexts $E_{pk}(m_1)$, $E_{pk}(m_2)$ and $k \in Z_N$, it has

$$E_{pk}(m_1) * E_{pk}(m_2) = E_{pk}(m_1 + m_2), \tag{3}$$

$$E^k_{pk}(m_1) = E_{pk}(km_1). \tag{4}$$

Semantic security: Given a set of ciphertexts, an adversary cannot deduce any information about the plaintexts.

3.2 Yao's Garbled Circuits

Garbled circuits [9] allow two semi-honest parties to compute an arbitrary function

without leaking any information about any input. There are two parities in garbled circuits, called constructor and evaluator respectively. Constructor prepares a garbled version of a circuit, while evaluator obliviously computes the output of the circuit without learning any intermediate values. We refer readers to [11] for more details.

Two different ADD garbled circuits will be adopted in this paper to construct our secure comparison protocol. One ADD circuit called ADD1 takes two l-bit integers x and y as input, and outputs an l-bit integer z, such that $z = x+y \bmod 2^l$. Another ADD circuit called ADD2 takes two l-bit integers x and y as input, and outputs an (l+1)-bit integer z, such that $z = x + y$.

4. SYSTEM MODEL AND FRAMEWORK

4.1 System Model

In this paper, we assume there are two non-colluding users, denoted as CS_1 and CS_2. CS_1 has two encrypted data $E_{pk}(x)$ and $E_{pk}(y)$ ($0 \le x, y < 2^l$) and x, y are not known to both CS_1 and CS_2, while CS_2 has the secret key sk. CS_1 wants to obtain the relationship of x and y in encrypted form, denoted as $E_{pk}(\delta)$, $\delta = 1$ if $x \ge y$, otherwise $\delta = 0$. Since x, y are available to CS_1 only in encrypted form, CS_1 needs to obtain $E_{pk}(\delta)$with the help of CS_2. Note that both CS_1 and CS_2 should not learn any information about δ, x and y during the comparison process. Besides, we assume these two non-colluding users are semi-honest (*i.e.*, honest-but-curious) [8], that is each participant correctly follows the protocol, but tries to learn more information than the output using what he obtains during the protocol excution.

4.2 Framework

SC_h in [5] not only achieves secure comparison over encrypted data but also gives a framework to securely compare encrypted data. This framework can be concluded as three main steps.

1. CS_1 computes $E_{pk}(z) = E_{pk}(x-y+2^l)$. Then CS_1 chooses a random (l+k+1)-bit value $r_1 \in Z_N$, where k is a security parameter, and $l+k+1 \ll \log_2 N$. CS_1 computes $E_{pk}(d) = E_{pk}(z)*E_{pk}(r_1) = E_{pk}(z+r_1)$, and sends it to CS_2. Then CS_2 decrypts $E_{pk}(d)$, obtains $d \equiv z+r_1$.
2. CS_1 obtains the encrypted value $E_{pk}(z \bmod 2^l)$ with the help of CS_2.
3. CS_1 gets the comparison result $E_{pk}(\delta) = E_{pk}^{2^{-l}}(z) * E_{pk}^{-2^{-l}}(z \bmod 2^l)$.

The core of this framework is step 2, *e.g.* securely computing $E_{pk}(z \bmod 2^l)$, while $r_1 \bmod 2^l$ is held by CS_1, and $d \bmod 2^l$ is held by CS_2. In SC_h, CS_1 firstly needs to get an encryption $E_{pk}(\lambda)$ of a binary value indicating whether $d \bmod 2^l < r_1 \bmod 2^l$ using secure bitwise comparison, and then computes $E_{pk}(z \bmod 2^l)$ as follows.

$$E_{pk}(z \bmod 2^l) = E_{pk}(d \bmod 2^l) * E_{pk}^{-1}(r_1 \bmod 2^l) * E_{pk}^{2^l}(\lambda) \tag{5}$$

However, secure bitwise comparison protocol takes many encryptions and decryp-

tions, thus the cost is too expensive. For example, secure bitwise comparison proposed in [17] needs 14l encryptions and 7l decryptions, when comparing two l-bit encrypted values. In this paper, we use garbled circuits to efficiently complete step 2 and propose a novel solution to securely compute $E_{pk}(z \bmod 2^l)$ based on ADD garbled circuit. Then, we construct our novel secure comparison protocol based on this framework. Details are described in next section.

5. THE PROPOSED PROTOCOL

In this section, we propose a novel secure comparison protocol SCED based on a hybrid approach of homomorphic cryptosystem and garbled circuits, which effectively combines the idea of SC_h and the property of garbled circuits. In our SCED, we use a more efficient approach to securely compute $E_{pk}(z \bmod 2^l)$ instead of secure bitwise comparison protocol. Our new approach is based on the following property.

Property 1: Given two l-bit integers x and y as the input of an add circuit, we can remove the final carry bit and get an l-bit output s, such that $s \equiv (x + y) \bmod 2^l$.

We combine this property with garbled circuits, efficiently computing the value $E_{pk}(z \bmod 2^l)$. Furthermore, we propose our new secure comparison protocol SCED shown in Protocol 1, which needs less encryptions, decryptions and smaller garbled circuits. Details about SCED are described as follows.

CS_1 firstly selects a random (l+k+1)-bit value $r \in Z_N$, where k is a security parameter and $l+k+1 \ll log_2 N$, and computes

$$E_{pk}(a) = E_{pk}(x) * E_{pk}^{-1}(y) * E_{pk}(r_1 + 2^l) = E_{pk}(z + r_1) \tag{6}$$

where $z = x - y + 2^l$. Then CS_1 sends $E_{pk}(a)$ to CS_2. After receiving $E_{pk}(a)$, CS_2 decrypts it using the secret key sk and obtains $a = z + r_1$. Note that CS_2 cannot learn anything about the value z due to the random value r_1 masking it.

After this, CS_1, CS_2 compute ($r_1 \bmod 2^l$), ($a \bmod 2^l$) respectively. It's easy for CS_1 and CS_2 to finish this computation since r_1 and a are known to CS_1, CS_2 respectively. Then, CS_1 as the circuit evaluator further constructs a garbled circuit based on the follow steps.

Add ($a \bmod 2^l$) and ($-r_1 \bmod 2^l$) using an add circuit ADD1. Note that the result $\gamma = z \bmod 2^l$, as part of the circuit, is not known to CS_1, CS_2.

CS_1 selects a random (l+k)-bit value r_2 and adds r_2 to γ using an add circuit ADD2. The circuit outputs the computation result $\beta = \gamma + r_2$ to CS_2 (*i.e.*, the circuit evaluator).

After requiring β, CS_2 returns $E_{pk}(b)$ to CS_1 where $E_{pk}(b) = E_{pk}((a-\beta)/2^l)$. Finally, CS_1 obtains the output $E_{pk}(\delta)$ by computing

$$E_{pk}(\delta) = E_{pk}(b) * E_{pk}^{-1}((r_1 - r_2)/2^l). \tag{7}$$

The correctness of SCED can be proved as follows. As $0 \le x, y < 2^l$, $z = x - y + 2^l$ is a positive (l+1)-bit value. The most significant bit (denoted as z_l) of z can be used to indicate the relationship of x and y. That is $z_l = 1$ if $x \ge y$, otherwise $z_l = 0$. The value z_l can

be computed as $z_l = 2^{-l}(z-(z \bmod 2^l))$. However, the value $E_{pk}(z \bmod 2^l)$ cannot easily be computed since the value x, y are available to CS_1 only in encrypted form. CS_1 needs to compute $E_{pk}(z \bmod 2^l)$ with the help of CS_2.

Thus, CS_1 firstly chooses a random $(l+k+1)$-bit value $r_1 \in Z_N$ and computes $E_{pk}(a) = E_{pk}(z+r_1)$. Then, SC_1 sends $E_{pk}(a)$ to CS_2. CS_2 decrypts $E_{pk}(a)$ to obtain $a = z+r_1$. Since r_1 is restricted to $(l+k+1)$–bit and $l+k+1 \ll \log_2 N$, $z+r_1$ will not lead to an overflow ($z+r_1 > N$), that is $a \equiv z+ r_1$. Thus we have $z \bmod 2^l = (a \bmod 2^l - r_1 \bmod 2^l) \bmod 2^l$. We use an add circuit ADD1 with inputs $(a \bmod 2^l)$ and $(-r_1 \bmod 2^l)$ to get the value $(z \bmod 2^l)$. Then we additively blind $(z \bmod 2^l)$ using another add circuit ADD2 and a random value r_2 chosen by CS_1. The final output of this garbled circuit is $\beta = z \bmod 2^l + r_2$, which is only known to the garbled circuit evaluator CS_2. CS_2 then computes $E_{pk}(b) = E_{pk}((a-\beta)/2^l)$ and returns it to CS_1. CS_1 calculates the final comparison result as follows.

$$\begin{aligned} E_{pk}(\delta) &= E_{pk}(b) * E_{pk}^{-1}((r_1 - r_2)/2^l) \\ &= E_{pk}((z + r_1 - (z \bmod 2^l) - r_2)/2^l - (r_1 - r_2)/2^l) \\ &= E_{pk}((z - (z \bmod 2^l))/2^l) \\ &= E_{pk}(z_l) \end{aligned} \tag{8}$$

Protocol 1: Secure Comparison Protocol Over Encrypted Data (SCED)

Input: CS_1 has two encrypted values $E_{pk}(x)$, $E_{pk}(y)$ without knowing plaintext x and y ($0 \le x, y < 2^l$). CS_2 has the secret key sk.

Output: CS_1 obtains the encrypted comparison result $E_{pk}(\delta)$, where $\delta = 1$ if $x \ge y$, and 0 otherwise. Note that both CS_1 and CS_2 should learn nothing about x, y and δ during the comparison process.

1: CS_1 selects a random $(l+k+1)$-bit value r_1, computes $E_{pk}(a) = E_{pk}(x) * E_{pk}^{-1}(y) * E_{pk}(r_1 + 2^l) = E_{pk}(z + r_1)$, sends $E_{pk}(a)$ to CS_2

2: CS_2 decrypts $E_{pk}(a)$, and gets $a = z+ r_1$

3: CS_1, CS_2 compute $(r_1 \bmod 2^l)$, $(a \bmod 2^l)$ respectively, then construct a garbled circuit based on the follow steps
- Add $(a \bmod 2^l)$ and $(-r_1 \bmod 2^l)$ using an add circuit ADD1 and outputs γ, as part of the circuit
- CS_1 selects a random $(l + k)$-bit value r_2, adds r_2 to γ using an add circuit ADD2.
- The circuit outputs the computation result $\beta = \gamma + r_2$ to CS_2

4: CS_2 gets the result of the garbled circuit. Then he computes $E_{pk}(b) = E_{pk}((a-\beta)/2^l)$ and sends it to CS_1.

5: CS_1 receives $E_{pk}(b)$ and obtains the output $E_{pk}(\delta) = E_{pk}(b) * E_{pk}^{-1}((r_1 - r_2)/2^l)$

6. EVALUATION

6.1 Security Analysis

In this section, we formally prove the security of our SCED protocol under semi-honest model. That is each participant is assumed to correctly follow the stated steps and

CS_1, CS_2 do not collude with each other. We consider the view of CS_1 and CS_2 in our SCED as follows.

CS_1: In SCED protocol, CS_1 can only get the encrypted value $E_{pk}(b)$ and final output $E_{pk}(\delta)$. CS_1 cannot learn any useful information from these ciphertexts, because all these values are encrypted using Paillier homomorphic encryption and Paillier is semantically secure.

CS_2: In SCED, CS_2 has the secret key and can decrypt $E_{pk}(a)$ to get the plaintext a. However, CS_2 cannot get anything about x, y from value a due to the random value r_1 masking $x-y+2^l$, giving it 2^{-k} statistical security. Besides, CS_2 can get the value $\beta = z \bmod 2^l + r_2$, which is still no useful for CS_2 since r_2 additively blinds the value $z \bmod 2^l$.

Besides, the security of garbled circuits has been formally proved in [12]. In all, our SCED leaks no useful information about x, y and δ, thus our SCED protocol is secure.

6.2 Computational Complexity

Table 1 lists the computational complexity of existing approaches SC_h, SC_g and our SCED. The detailed analysis about these protocols is as follows.

SC_h invokes secure bitwise comparison protocol to compute $E_{pk}(z \bmod 2^l)$. In this paper, we prefer to use the bitwise comparison protocol proposed in [17], which is based on Paillier encryption. The total computation cost in SC_h is $14l+2$ encryptions, $7l$ decryptions.

SC_g needs 4 encryptions, 2 decryptions and a garbled circuit consisting of three ADD2 garbled circuits and one comparison garbled circuit to get the encrypted comparison result. Considering the free-XOR technique [11], the computation cost of garbled circuits only depends on the number of non-XOR gates in the circuit. Using garbled circuits designed in [10], the number of non-XOR gates in the add circuit and comparison circuit with two l-bit as input are all l. The total computation cost in SC_g is 4 encryptions, 2 decryptions and $4l+4k+1$ non-XOR gates.

In SCED protocol, we use both homomorphic encryption and Yao's garbled circuits to achieve the comparison of two encrypted data. Only 3 encryptions, 1 decryption and two different add garbled circuits need to be performed in our SCED protocol. Considering that ADD1 takes two l-bit numbers $a \bmod 2^l$, $-r_1 \bmod 2^l$ as input and ADD2 takes two $(l+k)$-bit numbers as input, the computation cost of two different add garbled circuits is $2l+k$ non-XOR gates. Thus, the computation cost of SCED protocol is 3 encryptions, 1 decryption and $2l+k$ non-XOR gates.

Table 1. Computational complexity of existing approaches and ours.

Approach	Computation Cost		
	Encryption	Decryption	non-XOR gates
SC_h in [5]	$14l+2$	$7l$	0
SC_g in [17]	4	2	$4l+4k+1$
SCED	3	1	$2l+k$

7. EXPERIMENT

In this section, we analyze the performance of our proposed protocol and compare it with existing works SC_h, SC_g. In this experiment, we fix the Paillier encryption key size K=1024, and implement Yao's garbled circuit based on FasterGC [9] (the fastest known implementation for garbled circuits). Besides, we set security parameter k=20 to guarantee a sufficient statistical security. We conduct our experiment on a machine with 3.30 GHz CPU and 8GB RAM

We compare the performance of our protocol with SC_h and SC_g by varying the bit length l and the results are shown in Fig. 1. In Fig. 1, when l varies from 10 to 50, the running time of SC_h increases from 2.14 to 6.73 seconds, while the running time of our SCED protocol increases slightly from 0.69 to 0.96 seconds. It is clearly that our SCED is more efficient than SC_h. For example, when l=30, SCED's time (0.83 seconds) is about 16% of SC_h's time (5.21 seconds), and when l=50, SCED's time (0.96 seconds) is about 2% of SC_h's time (6.73 seconds).

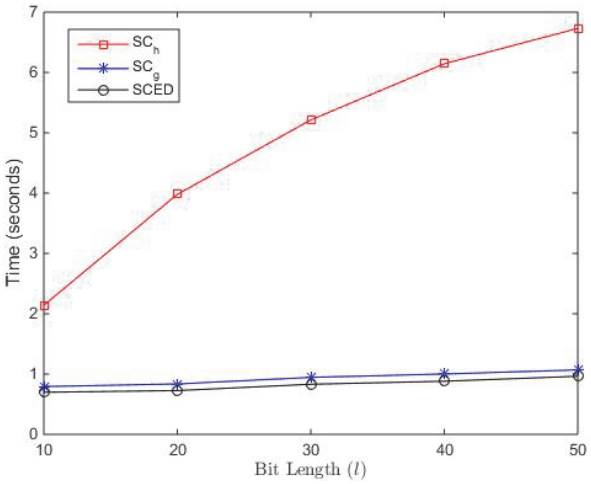


Fig. 1. Computation time of SC_h, SC_g and SCED.

Besides, in Fig. 1, when l varies from 10 to 50, the running time of SC_g increases from 0.79 to 1.07 seconds. We note that secure comparison needs to be done many times in some encrypted data analysis tasks. Though the running time improvement between SCED and SC_g is small, this will lead to more efficient encrypted data analysis tasks.

Above all, the computational cost of our SCED protocol is more efficient than SC_g and SC_h, which is consistent with our theoretical analysis in Table 1.

8. CONCLUSIONS

In this paper, we focused on the problem of secure comparison over encrypted data, which is one of fundamental operations in analyzing encrypted data. Existing secure

comparison protocols required a large amount of time-consuming operations and were inefficient. We proposed a novel protocol SCED for secure comparison over encrypted data, which effectively combined the idea of SC_h with the property of garbled circuits. We theoretically proved our protocol SCED is secure under semi-honest model. Additionally, through theoretical analysis and simulation experiments, we evaluated the computation cost of our scheme and compared it with existing works, which shows that our SCED is more efficient than the state-of-the-art ones.

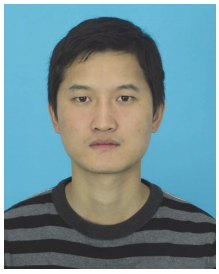
REFERENCES

1. H. Chun, Y. Elmehdwi, F. Li, P. Bhattacharya, and W. Jiang, "Outsourceable two-party privacy-preserving biometric authentication," in *Proceedings of the 9th ACM Symposium on Information*, *Computer and Communications Security*, 2014, pp. 401-412.
2. I. Damgard, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *Proceedings of Theory of Cryptography Conference*, 2006, pp. 285-304.
3. I. Damgard, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for online auctions," in *Proceedings of Australasian Conference on Information Security and Privacy*, 2007, pp. 416-430.
4. Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure *k*-nearest neighbor query over encrypted data in outsourced environments," in *Proceedings of IEEE 30th International Conference on Data Engineering*, 2014, pp. 664-675.
5. Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy preserving face recognition," in *Proceedings of International Symposium on Privacy Enhancing Technologies Symposium*, 2009, pp. 235-253.
6. Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Transactions on Information Forensics and Security*, Vol. 7, 2012, pp. 1053-1066.
7. I. Garay, B. Schoenmakers, and J. Villegas, "Practical and secure solutions for integer comparison," in *Proceedings of International Workshop on Public Key Cryptography*, 2007, pp. 330-342.
8. A. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, UK, 2009.
9. Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits," in *Proceedings of USENIX Security Symposium*, Vol. 201, 2011, pp. 1-16.
10. V. Kolesnikov, A.-R. Sadeghi, and T. Schneider, "Improved garbled circuit building blocks and applications to auctions and computing minima," in *Proceedings of International Conference on Cryptology and Network Security*, 2009, pp. 1-20.
11. V. Kolesnikov and T. Schneider, "Improved garbled circuit: Free XOR gates and applications," in *Proceedings of International Colloquium on Automata*, *Languages and Programming*, 2008, pp. 486-498.
12. Y. Lindell and B. Pinkas, "A proof of security of Yao's protocol for two-party computation," *Journal of Cryptology*, Vol. 22, 2009, pp. 161-188.

13. A. Liu, K. Zhengy, L. Liz, G. Liu, L. Zhao, and X. Zhou, "Efficient secure similarity computation on encrypted trajectory data," in *Proceedings of the 31st IEEE International Conference on Data Engineering*, 2015, pp. 66-77.
14. T. Nishide and K. Ohta, "Multiparty computation for interval, equality, and comparison without bit-decomposition protocol," in *Proceedings of International Workshop on Public Key Cryptography*, 2007, pp. 343-360.
15. P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, 1999, pp. 223-238.
16. B. K. Samanthula, H. Chun, and W. Jiang, "An efficient and probabilistic secure bit-decomposition," in *Proceedings of the 8th ACM SIGSAC symposium on Information, Computer and Communications Security*, 2013, pp. 541-546.
17. B. K. Samanthula, W. Jiang, and E. Bertino, "Privacy-preserving complex query evaluation over semantically secure encrypted data," in *Proceedings of European Symposium on Research in Computer Security*, 2014, pp. 400-418.
18. T. Ma, J. Zhou, M. Tang, and Y. Tian, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Transactions on Information and Systems*, Vol. 98, 2015, pp. 902-910.
19. Z. Fu, K. Ren, J. Shu, and X Sun, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, 2016, pp. 2546-2559.
20. Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, Vol. 98, 2015, pp. 190-200.
21. Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, 2016, pp. 340-352.
22. Y. W. Zhu, Z. Q. 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, 2016, pp. 1-12.
23. Z. Liu, J. Großschadl, and I. Kizhvatov, "Efficient and side-channel resistant RSA implementation for 8-bit AVR microcontrollers," in *Proceedings of the 1st International Workshop on the Security of the Internet of Things*, Vol. 10, 2010, pp. 51-60.
24. Z. Liu, J. Großschadl, and D. S. Wong, "Low-weight primes for lightweight elliptic curve cryptography on 8-bit processors," in *Proceedings of International Conference on Information Security and Cryptology*, 2013, pp. 217-235.
25. Z. Liu, H. Seo, J. Großschal, and H. Kim, "Efficient implementation of NIST – compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes," *IEEE Transactions on Information Forensics and Security*, Vol. 11, 2016, pp. 1385-1397.
26. R. Kitchin, "The real-time city? big data and smart urbanism," *GeoJournal*, Vol. 79, 2014, pp. 1-14.
27. A. Martínez-Ballesté, P. A. Pérez-Martínez, and A. Solanas, "The pursuit of citizens' privacy: a privacy-aware smart city is possible," *IEEE Communications Magazine*, Vol. 51, 2013, pp. 136-141.
28. X. Rui, K. Morozov, Y. J. Yang, J. Y. Zhou, and T. Takagi. "Privacy-preserving *k*-nearest neighbour query on outsourced database," in *Proceedings of Australasian*

Conference on Information Security and Privacy, LNCS 9722, 2016, pp. 181-197.
29. Y. J. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, Vol. 16, 2015, pp. 317-323.
30. J. Shen, H. W. Tan, J. Wang, J. W. Wang, and S. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *Journal of Internet Technology*, Vol. 16, 2015, pp. 171-178.
31. S. D. Xie and Y. X. Wang, "Construction of tree network with limited delivery latency in homogeneous wireless sensor networks," *Wireless Personal Communications*, Vol. 78, 2014, pp. 231-246.

Xing-Xin Li (李兴鑫) received the B.S. degree in Nanjing University of Aeronautics and Astronautics in 2014. He is currently pursuing the M.S. degree at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include cloud computing and data privacy.



You-Wen Zhu (朱友文) received his Ph.D. degree in Computer Science from University of Science and Technology of China in 2012. He is currently an Associate Professor at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include information security and data privacy.

Jian Wang (王箭) received the Ph.D. degree in Nanjing University in 1998. He is currently a Professor at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include cryptographic protocol and malicious tracking.