IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 9, 2018, accepted April 8, 2018, date of publication April 16, 2018, date of current version May 24, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2827025

Secure "Ratio" Computation and Efficient Protocol for General Secure Two-Party Comparison

LINMING GONG[1], SHUNDONG LI[2], CHUNYING WU[3], AND DAOSHUN WANG[4], (Member, IEEE)

[1]Shaanxi Key Laboratory of Clothing Intelligence, National and Local Joint Engineering Research Center for Advanced Networking and Intelligent Information Service, School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China
[2]School of Computer Science, Shaanxi Normal University, Xi'an 710062, China
[3]School of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 710062, China
[4]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Corresponding authors: Linming Gong (glmxinjing@163.com) and Shundong Li (shundong@snnu.edu.cn)

This work was supported in part by the Ph.D. Science Foundation of Polytechnic University under Grant 107020331, in part by the Nature Science Foundation of China under Grant 61272435, Grant 61373020, Grant U1536102, and Grant U1536116, and in part by the China Mobile Research Fund Project under Grant MCM20170407.

ABSTRACT Many privacy protections in distributed setting are based on secure comparison, according to which two distrusted users try to jointly determine whether $a > b$, $a = b$, or $a < b$. Performing "secure comparison" is fundamental to achieve widespread acceptance of privacy protection for distributed setting users. Surprisingly, however, no attention has been paid to secure comparison on fractions, to the best of our knowledge. In this paper, we first present an efficient system for computing $((ka + k_1)/(kb + k_1))$, which implies the potential relationship ($a > b$, $a = b$, and $a < b$) between a and b, where k, k_1, and b belong to the party who doesn't have the secret of the system. Based on this system, two distrusted users in distributed setting can learn whether $a > b$, $a = b$, or $a < b$ in one execution, without disclosing a and b to each other. Then, we develop two efficient protocols for secure comparison on integers and secure comparison on fractions, respectively. Our schemes, based on homomorphic encryption, are cryptographically secure. We prove that these protocols are secure using simulation paradigm. Our approaches can be used in many secure multi-party computation protocols that involve fractions, rational numbers, and integers. They can also be more convenient to solve some secure multiparty computational geometry problems that often involve ratio evaluation.

INDEX TERMS Secure ratio computation, millionaires' problem, homomorphic encryption, secure computation.

I. INTRODUCTION

In applications of network communications and signal processing, we (participants) are often interested in a particular scenario wherein a party $\mathbf{P}_c$ seeks the cooperation of another party $\mathbf{P}_s$ to perform a secure comparison task (Secure Two-Party Comparison, STPC).

STPC problem can be described as follows. Alice and Bob want to jointly determine the potential relationship ($>, =$ or $<$) of their confidential numbers without revealing this data to each other nor to anyone else. However, they still want to learn the relationship of their private data. It is the fundamental to secure multi-party computation (SMC), and it plays an important role in many applications, such as price negotiations in electronic auction or bidding systems [1]–[9], [22], privacy-preserving computational geometry [10]–[13], privacy preserving data mining [14]–[19], and private set intersection [20], [21]. As a sub-protocol (to be called) of many SMC protocols, the reuse of private information of the involved two parties with no or negligible leakage is all-important. Since many SMC protocols involve a large number of instances of secure comparison, even a minor efficiency gain in secure comparison will bring about significant performance improvements [22].

Related Works: The secure comparison problem starts from the millionaires' problem that was presented by Yao [23]. It is a cryptographic solution for determining "greater than" between two numbers. The complexity of this solution is exponential in the number of bits of the involved numbers. Before long, Cachin [24] put forward a protocol based on the Φ-hiding assumption with constant-round communication complexity, which needs a third party. Thereafter, many

efficient comparison protocols without any third party have been constructed [22], [25]–[29]. Ioannidis and Grama [26] designed a protocol to solve the secure comparison problem based on the 1-out-of-2 oblivious transfer, which takes d-round communication (where d is the length of the private inputs), and is restricted by the security parameter of the oblivious transfer scheme. Fischlin [25] put forward a secure comparison protocol with 2-round communication based on the Goldwasser-Micali encryption scheme. Blake and Kolesnikov [22] and Lin and Tzeng [27] constructed secure comparison protocols with 2-round communication based on the Paillier addition homomorphic cryptosystem and the ElGamal multiplication homomorphic encryption scheme. However, all these protocols with 2-round communication can only solve the greater than secure comparison problem and determine the relation of two integers. Luo *et al.* [28] proposed a secure comparison protocol by geometric method, which takes $24 \log n + 8d + 4$ modular multiplications. However, this protocol can only detemine the relation of two real numbers. Veugen *et al.* [29] gave a comprehensive analysis only on the state-of-the-art integer comparison protocols for a two party setting in the semi-honest security protocol. However, all these analyzed protocols can only compare the numbers less than $\lfloor \log \frac{n}{3} \rfloor$. Li and Wang [30] presented an efficient secure comparison protocol based on homomorphic encryption and the Fundamental Theorem of Arithmetic, which can only determine $>$ or $\leq$ of two integers in one execution. Nakai *et al.* [31] developed several efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations, which solve the "greater than" comparison problem (on integers) in a new perspective. Hezaveh *et al.* [33] put forward an efficient solution to the socialist millionaires' problem, which can only solve "equality" comparison problem (on integers) in an efficient way. Grigoriev *et al.* [35] solved Yao's millionaires' problem (i.e. greater than comparison on integers) utilizing public-key encryption without computational assumptions. Liu *et al.* [32] proposed a solution to secure comparison (on integers) based on Paillier homomorphic encryption scheme. Li *et al.* [34] put forward an efficient solution for the general millionaires' problem based on Paillier homomorphic encryption scheme, which can solve comparison problem (on rational numbers).

All those above protocols have solved the secure comparison problem well. However, no method can be found to solve the millionaires' problem precisely in the fraction field. In this paper, we propose protocols to solve this problem. Our protocols are highly efficient and can precisely figure out the three possible relationships $(>, =, <)$ between two (integer, or rational, or fraction) numbers in one execution.

Contributions: It is quite common in practical applications that the STPC problem is not confined to the integer field. Indeed, private inputs that comprise rational numbers or fractions are more common in general cases. Therefore, a more generalized precise solution to the millionaires' problem is absolutely necessary for the broader field. It is noted that rational numbers comprise integer numbers, and rational numbers can be transformed into fractions but the reverse is not true. Therefore, the solution to the millionaires' problem on fractions is more flexible and adaptable.

We study this problem utilizing the "ratio" approach, which can avoid the wrong understanding that the encryption of the difference between two numbers can be evaluated via homomorphic operation of Paillier encryption scheme. Refer to the analysis in paper [38], the encryption function $f(x, y) = (1 + kn)^x y^n \bmod n^2$ is bijective if and only if $(x \in Z_n) \wedge (y \in Z_n^*)$. That is to say, Paillier encryption scheme can not be used to encrypt negative numbers. In order to solve the secure comparison with a correct utilization of Paillier encryption scheme, we seek to figure out the relationship between two private numbers through evaluating a relationship-ratio. Where, the problem of comparing two fractions can be reduced to computing a relationship-ratio that reflects the relationship between two integers.

Our main contributions are as follows.

(1) We provide a "ratio" method for solving general secure comparison problem with a correct utilization of Paillier encryption scheme;

(2) We solve the millionaires' problem precisely for fractions;

(3) Our protocols have high efficiency that it can figure out the three potential relationships $(<, =, >)$ between two private numbers from two distrusted parties in one execution;

(4) To achieve a lower computational complexity, our protocols entrust the time-consuming exponent calculation "$r^n \bmod n^2$" to cloud-server in data pretreatment phase. We employ an equivalent but far more efficient way ($r_x^n \bmod n^2 = R_i^{\ell_1} \cdot R_j^{\ell_2} \bmod n^2$, where $\ell_1 + \ell_2$) instead of time-consuming exponent calculation: $r^n \bmod n^2$.

The rest of this paper is organized as follows. Section 2 introduces the building blocks. Section 3 describes the secure relationship-ratio computing system and shows its correctness and security. Section 4 gives the solution to secure comparison with encrypted integers. Section 5 gives the solution to secure comparison with encrypted fractions. Section 6 compares the performance of different protocols based on the cryptographic computational problem, "decisional composite residuosity". Section 7 gives our conclusions.

II. BUILDING BLOCKS

In this section, we describe essential building blocks used in secure comparison (SC) protocols: definitions about two-party secure computation in **Section 2.1**, homomorphic encryption (HM) in **Section 2.2** and analogous decryption in **Section 2.3**.

A. DEFINITIONS ABOUT TWO-PARTY SECURE COMPUTATION

Through out this paper, we define security for two party protocols in the presence of semi-honest adversaries.

The following definitions are formed according to [36] and [37].

Definition 1 (Ideal Protocol): Assume that there exists a completely Trusted Third Party (TTP). Assisted by TTP, a secure two-party computation protocol can be performed as follows. Alice and Bob transform their inputs x and y to TTP. TTP evaluates $f(x, y)$ independently, and sends the output to Alice and Bob once he completes the evaluation. Because there is no way for Alice and Bob to obtain additional information other than $f(x, y)$. Such a simple protocol is the highest private protocol of secure two-party evaluation, and the privacy of any practical secure two-party evaluation protocol cannot outperform this protocol.

Definition 2 (Semi-Honest Participant): According to [36], a semi-honest participant follows the protocol trustily as specified. However, it may try to learn more information than allowed by looking at the transcript of messages that it received and its internal state(s).

Definition 3 (Simulation-Based Security [36]): Let $f = (f_A, f_B) : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a two-party functionality that can be evaluated by Alice and Bob in probabilistic polynomial time and let Π be a two-party protocol for computing f. Given the input of the protocol, the view of Alice and Bob in an execution of Π on inputs (a, b) is

$$view_d^{\pi}(a,b) = (a, r_d, m_{d1}, m_{d2}, \cdots, m_{dk}),$$

where $d \in \{A, B\}$ and r_d is the content of Alice's or Bob's internal random tape, and m_{di} represents the i-th message that she or he received. The output of the Alice in an execution of Π on (a, b) is denoted by $Output_A^{\Pi}(a, b)$ and can be computed from $View_A^{\pi}(a, b)$. Similarly, the output of Bob in an execution of Π on (a, b) is denoted by $Output_B^{\Pi}(a, b)$ and can be computed from $View_B^{\pi}(a, b)$.

Let f and Π be as above. Protocol Π is said to securely compute f in the presence of semi-honest adversaries if there exist probabilistic polynomial-time algorithms $\mathcal{S}_A$ and $\mathcal{S}_B$ such that

$$\{(\mathcal{S}_A(a, f_A(a,b)), f_B(a,b))\}_{a,b} \overset{c}{\equiv} \{(view_A^{\pi}(a,b), output_B^{\pi}(a,b))\}_{a,b} \tag{1a}$$

$$\{(f_A(a,b), \mathcal{S}_B(a, f_B(a,b)))\}_{a,b} \overset{c}{\equiv} \{(output_A^{\pi}(a,b), view_B^{\pi}(a,b))\}_{a,b} \tag{1b}$$

where $\overset{c}{\equiv}$ represents indistinguishability on computation.

B. HOMOMORPHIC PUBLIC ENCRYPTION (HPKE)

Intuitively, a public key encryption scheme is homomorphic if given two ciphertexts $C_1 = E_{pk}(m_1; r_1)$ and $C_2 = E_{pk}(m_2; r_2)$, it is possible to efficiently compute $E_{pk}(m_1 \circledR_p m_2; r)$ by evaluating $C_1 \circledR_c C_2$, where $\circledR_p$ and $\circledR_c$ are operations that carried out in plaintext space ($\mathcal{M}$) and cipher space ($\mathcal{C}$), respectively. We abuse notation and use $E_{pk}(m)$ to denote the random variable induced by $E_{pk}(m; r)$, where r is chosen uniformly. We have the following formal definition,

Definition 4 (Homomorphic Public Encryption (HPKE)): A public encryption $(\mathcal{G}, E, D)$ is homomorphic if for all N and all (pk, sk) output by key generating algorithm $\mathcal{G}(1^N)$, and for every $m_1, m_2 \in \mathcal{M}$ it holds that

$$\{pk, C_1 = E_{pk}(m_1), C_2 = E_{pk}(m_2), C_1 \circledR_c C_2\} \equiv \{pk, C_1 = E_{pk}(m_1), C_2 = E_{pk}(m_2), E_{pk}(m_1 \circledR_p m_2)\} \tag{2}$$

where $C_1, C_2 \in \mathcal{C}$.

The Paillier Cryptosystem: Paillier put forward three encryption schemes [38], where Schemes 1 (see Figure 1) is homomorphic.

Encryption:
- plaintext $m < n$
- select a random $r < n$
- cipherext $C = g^m r^n \bmod n^2$

Decryption:
- $L(\mu) = \frac{\mu - 1}{n}$
- ciphertext $C < n^2$
- plaintext $m = \frac{L(C^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)} \bmod n$
- where p, q are two big primes with equal length, $n = pq$, $\lambda = \text{lcm}(p-1, q-1)$, $g = 1 + kn$ $(k \in Z_n^*)$

FIGURE 1. Paillier's encryption scheme 1.

This scheme is semantically secure, assuming hardness of the decisional composite residuosity problem. Note that, the Paillier Scheme 1 has two additively homomorphic properties

$$E(m_1 + m_2) = E(m_1) \cdot E(m_2), \tag{3a}$$

$$(E(m_2))^{m_1} = E(m_1 m_2), \tag{3b}$$

which plays an important role in secure computing and computing on encrypted data.

Definition 5 (Decisional Composite Residuosity (DCR) Problem): Set $\mathcal{D}$ be a distinguisher, and let the sets D_{Ran} and D_{Cr} are two distributions:

$$D_{Ran} = \{(n, \mathcal{R}) = (n, R) | \ \mathcal{R} \overset{R}{\leftarrow} Z_{n^2}\}$$
$$D_{Cr} = \{(n, \mathcal{R}) = (n, r^n \bmod n^2) | \ \mathcal{R} \leftarrow r^n \bmod n^2\};$$

where τ is the secure parameter and $Adv_{\mathcal{D}}(\tau)$ is the advantage of a distinguisher $\mathcal{D}$ in distinguishing distributions D_{Ran} and D_{Cr}. Given a distribution $(n, \mathcal{R}) \in \{D_{Ran}, D_{Cr}\}$, the distinguishing result on $(n, \mathcal{R})$ from a distinguisher $\mathcal{D}$ is denoted as $\mathcal{D}(n, \mathcal{R}) = D_{Ran}$ or $\mathcal{D}(n, \mathcal{R}) = D_{Cr}$. Then, $Adv_{\mathcal{D}}(\tau)$ can be expressed as

$$Adv_{\mathcal{D}}(\tau) = |Pr[\mathcal{D}(n, \mathcal{R}) = D_{Ran}] - Pr[\mathcal{D}(n, \mathcal{R}) = D_{Cr}]|.$$

This is a well-known intractable problem. In other word, for any probabilistic, polynomial-time algorithm $\mathcal{D}$, there is a negligible function $\delta(\tau)$ such that

$$Adv_{\mathcal{D}}(\tau) \leq \delta(\tau).$$

C. PRE-COMPUTATION WITH THE AID OF CLOUD

In order to improve computation efficiency, we assume that both Alice and Bob should develop pre-computation with the aid of cloud prior to their undergoing secure comparison protocols (see in **Section 3** and **Section 4**) as follows.

Alice or Bob selects $r_{i_A}, r_{i_B} \in Z_n^*$ ($r_{i_A}, r_{i_B} < n$) respectively, and both entrust $R_{i_\Psi} = r_{i_\Psi}^n \bmod n^2$ ($\Psi \in \{A, B\}$) to cloud. After receiving all R_{i_Ψ}, Alice or Bob permutes R_{i_Ψ} and stores them in a collection which is denoted by R_Ψ. When encrypting a message using Paillier scheme, one can employ an equivalent but far more efficient way to evaluate $r^n \bmod n^2$ as follows. Firstly, one randomly selects several members from R_Ψ, for example, $R_i, R_j \in R$ as the seeds, then he or she computes $r_x^n \bmod n^2 = R_i^{\ell_1} \cdot R_j^{\ell_2} \bmod n^2$, where ℓ_1, ℓ_2 and $\ell_1 + \ell_2$ are small integers. This does not lead to information leakage, which is guaranteed by the semantic security of Paillier encryption scheme.

D. FIGURE OUT RELATIONSHIP BETWEEN TWO NUMBERS BY EVALUATING RELATIONSHIP-RATIO

Note that the relationship between α, β (where $\alpha, \beta \in Z_n^+$) can be determined by computing $\frac{K\alpha+K_1}{K\beta+K_1}$, that is $\alpha > \beta$ if $\frac{K\alpha+K_1}{K\beta+K_1} > 1$, $\alpha = \beta$ if $\frac{K\alpha+K_1}{K\beta+K_1} = 1$, and $\alpha < \beta$ if $\frac{K\alpha+K_1}{K\beta+K_1} < 1$. We can define function

$$\mathcal{P}(\frac{\alpha}{\beta}) = \begin{cases} +1 & \frac{K\alpha+K_1}{K\beta+K_1} > 1 \\ 0 & \frac{K\alpha+K_1}{K\beta+K_1} = 1 \\ -1 & \frac{K\alpha+K_1}{K\beta+K_1} < 1, \end{cases}$$

to figure out the relationship between α and β, where $\frac{K\alpha+K_1}{K\beta+K_1}$ is called `relationship-ratio`.

III. SECURE RELATIONSHIP-RATIO COMPUTING SYSTEM

A. DESCRIPTION OF SECURE RELATIONSHIP-RATIO COMPUTING SYSTEM

Without loss of generality, assuming the communicating parties, Alice and Bob have secrets related to a and b, respectively. Alice and Bob try to jointly evaluate the relationship of their private data by performing a secure relationship-ratio computing system, but they do not want to disclose their secrets to each other.

In what follows, we give the description of secure relationship-ratio computing system, which is composed of three random algorithms: **Key-Generation**, **Encrypted Relationship-Ratio Development** and **Relationship-Ratio Computation**. The system is denoted as Π(**Key** $-$ **Gen**, **Enc** $-$ **RRD**, **RR** $-$ **Computation**):

Key-Gen: Alice runs the key generation algorithm to generate her public-key $(n,\ 1+n)$ and private key $\lambda = lcm(p-1, q-1)$.

Enc-RRD: (1) Alice randomly selects $r_a \in Z_n^*$ and encrypts her secret a into $C_{(a,1+n)}$ as follows.

$$C_{(a,1+n)} = (1+n)^a r_a^n \bmod n^2.$$

Then Alice sends it to Bob.

(2) After receiving $C_{(a,1+n)}$, Bob works as follows.

– Selects a random number $r_b \in Z_n^*$ and encrypts his private information b into $C_{(b,1+n)}$ as follows.

$$C_{(b,1+n)} = (1+n)^b (r_b)^n \bmod n^2.$$

– Selects $K \in Z_{n-1}^+$, $K_1, K_2 \in Z_n^+$, $r_b, r_{b1}, r_{b2}, r_{b3} \in Z_n^*$ and evaluates:

$$\begin{aligned}
C_{(K_2,1+K_1n)} &= (1+K_1\cdot n)^{K_2} r_{b1}^n \bmod n^2, \\
C_{(a,1+K_1n)} &= C_{(a,1+n)}^{K_1} \bmod n^2 \\
&= ((1+K_1\cdot n)^a r_a^{K_1 n} \bmod n^2, \\
C_{(b,1+K_1n)} &= C_{(b,1+n)}^{K_1} \bmod n^2 \\
&= ((1+K_1\cdot n)^b r_b^{K_1 n} \bmod n^2, \\
C_{(Ka,1+K_1n)} &= C_{(a,1+K_1n)}^{K} \bmod n^2 \\
&= ((1+K_1\cdot n)^{Ka} r_a^{KK_1 n} \bmod n^2, \\
C_{(Kb,1+K_1n)} &= C_{(b,1+K_1n)}^{K} \bmod n^2 \\
&= ((1+K_1\cdot n)^{Kb} r_b^{KK_1 n} \bmod n^2, \\
&C_{(Ka+K_2,1+K_1n)} \\
&= C_{(Ka,1+K_1n)} \cdot C_{(K_2,1+K_1n)} \cdot r_{b2}^n \bmod n^2 \\
&= ((1+K_1\cdot n)^{Ka+K_2} r_a^{KK_1 n} r_{b1}^n r_{b2}^n \bmod n^2, \\
&C_{(Kb+K_2,1+K_1n)} \\
&= C_{(Kb,1+K_1n)} \cdot C_{(K_2,1+K_1n)} \\
&\quad \cdot r_{b3}^n \bmod n^2 \\
&= ((1+K_1\cdot n)^{Kb+K_2} r_b^{KK_1 n} r_{b1}^n r_{b3}^n \bmod n^2.
\end{aligned}$$

Then Bob sends the pair $(C_{(Ka+K_2,1+K_1n)}, C_{(Kb+K_2,1+K_1n)})$ to Alice.

RR-Computation: Set $\mathcal{L}(\chi) = \chi - 1$, where $\chi \in Z_{n^2}$. Owing to Carmichael's theorem, knowing with $(C_{(Ka+K_2,1+K_1n)}, C_{(Kb+K_2,1+K_1n)})$, Alice can obtain a fraction $0 < \frac{Ka+K_2}{Kb+K_2} < n^2$ by computing

$$\begin{aligned}
&\frac{\mathcal{L}(C_{(Ka+K_2,1+K_1n)}^{\lambda} \bmod n^2)}{\mathcal{L}(C_{(Kb+K_2,1+K_1n)}^{\lambda} \bmod n^2)} \\
&= \frac{\mathcal{L}((((1+K_1\cdot n)^{Ka+K_2} r_a^{KK_1 n} r_{b1}^n r_{b2}^n \bmod n^2)^{\lambda} \bmod n^2)}{\mathcal{L}((((1+K_1\cdot n)^{Kb+K_2} r_b^{K_1 n} r_{b1}^n r_{b3}^n \bmod n^2)^{\lambda} \bmod n^2)} \\
&= \frac{\lambda(Ka+K_2)K_1}{\lambda(Kb+K_2)K_1} \\
&\equiv \frac{Ka+K_2}{Kb+K_2}.
\end{aligned}$$

Note that the process used to evaluate $\frac{\mathcal{L}(C_{(Ka+K_2,1+K_1n)}^{\lambda} \bmod n^2)}{\mathcal{L}(C_{(Kb+K_2,1+K_1n)}^{\lambda} \bmod n^2)}$ is analogous to the decryption of Paillier, while the output of this process is a real number $0 < \mathbb{R} = \frac{Ka+K_2}{Kb+K_2} < n^2$ rather

than an integer on Z_n. We call it `Relationship-Ratio Computation` in following sections.

`Relationship out of relationship-ratio computing`. Obviously, one can figure out the relationship of a and b through determining the relationship of $\frac{Ka+K_2}{Kb+K_2}$ and 1 as follows.

$$\frac{a}{b} = \frac{Ka}{Kb} \begin{cases} > 1 & \frac{Ka+K_2}{Kb+K_2} > 1 \\ = 1 & \frac{Ka+K_2}{Kb+K_2} = 1 \\ < 1 & \frac{Ka+K_2}{Kb+K_2} < 1. \end{cases}$$

Because of this, we define this process as `relationship out of relationship-ratio computing`.

B. SECURITY ANALYSIS

Theorem 1: If the DCR problem is intractable, then system Π *has indistinguishable encryptions under adaptive chosen plaintext attacks.*

Proof: Recall that the DCR challenger works as follows: it generates (p, q, n), and chooses a random number $r \in Z_n$ and f uniformly from $\{0, 1\}$, sets $\mathcal{R} = r^n \bmod n^2$ if $f = 0$ and $\mathcal{R} = R$ if $f = 1$, and finally gives $(n, (n, \mathcal{R}))$ to the attacker.

Let Π(**Key − Generation**, **Homomorphicencryption**, **Analogousdecryption**) be our developed system. Let $\mathcal{A}$ be a polynomial-time algorithm attacking Π($\mathcal{A}$ could be external or internal). We may construct an algorithm $\mathcal{B}$ to solve the DCR problem as follows.

Algorithm $\mathcal{B}$

1: Receives $(n, (n, \mathcal{R}))$ from the DCR challenger;
2: Let $pk = (n, n + 1)$;
3: Sends 1^n and pk to $\mathcal{A}$;
4: Receives two messages m_0 and m_1 from $\mathcal{A}$;
5: Chooses $d \in \{0, 1\}$ uniformly;
6: Let $C^* = (n, n + 1, (1 + n)^{m_d} \cdot \mathcal{R} \pmod{n^2})$ and sends C^* to $\mathcal{A}$;
7: Let d' denote the output of d guessed by $\mathcal{A}$;
8: Outputs f' (If $d = d'$ and then set $f' = 0$. If $d \neq d'$, then set $f' = 1$).

Algorithm $\mathcal{B}$ runs in polynomial time because $\mathcal{A}$ runs in polynomial time. Consequently, the operations in $\mathcal{G}(1^k)$ can be performed in polynomial time. By Bayes theorem, we can evaluate the probability that $\mathcal{B}$ wins the DCR security game as follows.

$$\begin{aligned} &Pr[f = f'] \\ &= Pr[f = 0]Pr[f = f'|f = 0] + Pr[f = 1]Pr[f = f'|f = 1] \\ &= \frac{1}{2}Pr[f' = 0|f = 0] + \frac{1}{2}Pr[f' = 1|f = 1] \\ &= \frac{1}{2}Pr[d = d'|f = 0] + \frac{1}{2}Pr[d \neq d'|f = 1]. \end{aligned} \quad (4)$$

When $f = 0$, the DCR challenger sets $\mathcal{R} = r^n \bmod n^2$, and thus the view that $\mathcal{B}$ presents to $\mathcal{A}$ is identical to that of the actual IND-CPA secure game against Π. Therefore, the probability that $d = d'$ given $f = 0$ is the same as the probability that $\mathcal{A}$ wins the IND-CPA secure game against Π, i.e.,

$$Pr[d = d'|f = 0] = \frac{1}{2} + \delta. \quad (5)$$

When $f = 1$, the DRC challenger sets $\mathcal{R} = R$. Since R is uniformly selected from Z_{n^2}. It follows that $(1 + n)^{m_d} \cdot \mathcal{R} \pmod{n^2}$ is uniformly distributed on the group Z/n^2Z. Moreover, the random variables m_0, m_1 and d are jointly independent. Hence, pk and C^* reveal no information about d, so the guess d' output by $\mathcal{A}$ must be independent of d. Since d is either 0 or 1, each with a probability of $\frac{1}{2}$, it follows that

$$Pr[d = d'|f = 1] = \frac{1}{2}. \quad (6)$$

From (4), (5) and (6), it follows that

$$Pr[f = f'] = \frac{1}{2}(\frac{1}{2} + \delta) + \frac{1}{2} \times \frac{1}{2} = \frac{1}{2} + \frac{1}{2}\delta. \quad (7)$$

Thus, $\mathcal{B}$ wins the DCR security game with advantage:

$$\begin{aligned} &\left|Pr[f = f'] - \frac{1}{2}\right| \\ &= \left|(\frac{1}{2} + \frac{1}{2}\delta) - \frac{1}{2}\right| \\ &= \frac{\delta}{2}. \end{aligned} \quad (8)$$

By **Definition 5**, algorithm $\mathcal{B}$ might win the DCR security game with only a negligible advantage, so $\frac{\delta}{2}$ must be negligible, which implies that δ is also negligible. Therefore, algorithm $\mathcal{A}$ has only a negligible advantage δ in the IND-CPA game against Π. □

Theorem 2: If the system Π *has indistinguishable encryptions under adaptive chosen plaintext attacks, then Alice and Bob can securely figure out the relationship between a and b through system* Π.

Proof: For Alice, knowing λ and K_1, she can figure out the relationship between a and b using `relation out of fuzzy computing`, but cannot obtain any additional information about K, b, K_2 and K_1.

For Bob, due to the sematic security of the system Π, $C_{(a,1+n)} = (1 + n)^a r_a^n \bmod n^2$ is computationally indistinguishable from $r^n \bmod n^2$ (r is a randomness from Z_n^*).

Hence, Alice and Bob can securely figure out the relationship between a and b through system Π. □

IV. SECURE COMPARISON WITH ENCRYPTED INTEGER INPUTS

In this section, utilizing the secure relationship-ratio computing system, we first propose a new protocol (denoted as SCEII) for securely comparing two integers. Then, we analyze its security, correctness and efficiency in the standard model (simulation-based security).

A. DESCRIPTION OF PROTOCOL SCEII

Based on pre-computation, homomorphic operation and with the aid of cloud, we design an efficient protocol for securely comparing integers in semi-honest setting. To securely compare their integer numbers (Alice with X and Bob with Y), Alice and Bob proceed the protocol SCEII as follows.

Step 1. Alice runs the key generation algorithm to generate her keys, and then she publishes the public-key $(n,\ 1+n)$ but keeps λ as her private key.

Step 2. Alice firstly selects a random number $R_{1_A} \in R_A$, and then encrypts her private X by her own public-key as follows.

$$c_X = (1 + X \cdot n) \cdot R_{1_A} \bmod n^2. \tag{9}$$

Then Alice sends c_X to Bob.

Parallel operation. Before receiving c_X, Bob works as follows.

- Selects $r_Y \in Z_{n-1}^+$, $r_{XY} \in Z_n^+$, $R_{1_B}, R_{2_B} \in R_B$ randomly and evaluates

$$c_{r_{XY}} = R_{1_B}(1 + r_{XY} \cdot n) \bmod n^2. \tag{10a}$$

$$c_{Y+r_{XY}} = R_{2_B}(1 + n(r_Y \cdot Y + r_{XY})) \bmod n^2. \tag{10b}$$

- Selects $\gamma_1, \gamma_2, \ldots, \gamma_{2\ell_1-2} \in Z_n^+$ ($\ell_1 \geq 2$) such that half of the numbers $\frac{\gamma_1}{\gamma_2}, \frac{\gamma_3}{\gamma_4}, \ldots, \frac{\gamma_{2\ell_1-3}}{\gamma_{2\ell_1-2}}$ are greater than 1 but the remaining half are less than 1.
- For $1 \leq i \leq \ell_1 - 1$, randomly chooses $k_i \in Z_n$ and $R_{(2i+1)_B}, R_{(2i+2)_B} \in R_B$, and evaluates ciphertext pairs as follows.

$$c_{\gamma_{2i+1}} = (1 + k_i \cdot \gamma_{2i+1} \cdot n \bmod n^2) \cdot R_{(2i+1)_B} \bmod n^2. \tag{11a}$$

$$c_{\gamma_{2i+2}} = (1 + k_i \cdot \gamma_{2i+2} \cdot n \bmod n^2) \cdot R_{(2i+2)_B} \bmod n^2. \tag{11b}$$

Step 3. After receiving c_X, Bob works as follows.

① Selects a random number $R_Y \in R$, and computes $c_{X+r_{XY}}$ using the homomorphism of Paillier cryptosystem as follows.

$$c_{X+r_{XY}} = (c_X)^{r_Y} \cdot c_{r_{XY}} \cdot R_Y \pmod{n^2}. \tag{12}$$

② Implements an permutation on these above ℓ_1 ciphertext pairs $((c_{X+r_{XY}}, c_{Y+r_{XY}}), (c_{\gamma_1}, c_{\gamma_2}), (c_{\gamma_3}, c_{\gamma_4}), \ldots, (c_{\gamma_{2\ell-3}}, c_{\gamma_{2\ell-2}}))$, and denotes this permutation as $(c_1, c_2), (c_3, c_4), \ldots, (c_{2\ell_1-1}, c_{2\ell_1})$, then sends them to Alice.

Step 4. After receiving $(c_1, c_2), (c_3, c_4), \ldots, (c_{2\ell_1-1}, c_{2\ell_1})$, Alice evaluates

$$\mathbf{Sign} = \prod_{i=1}^{\ell_1} P\left(\frac{\mathcal{L}(c_i^\lambda)}{\mathcal{L}(c_{i+1}^\lambda)}\right), \quad \text{where } P(\mathcal{X}) = \begin{cases} +1 & \mathcal{X} > 1 \\ 0 & \mathcal{X} = 1 \\ -1 & \mathcal{X} < 1. \end{cases} \tag{13}$$

and sends **Sign** to Bob.

Step 5. After receiving **Sign**, by

$$\mathbf{R}_{\ell_1\mathbf{e}} = \begin{cases} X > Y & \mathbf{Sign} = +1 \\ X = Y & \mathbf{Sign} = 0 \\ X < Y & \mathbf{Sign} = -1 \end{cases} \quad \left(\text{where } \frac{\ell_1}{2} \text{ is even or } \ell_1 = 1\right)$$

Or

$$\mathbf{R}_{\ell_1\mathbf{o}} = \begin{cases} X < Y & \mathbf{Sign} = +1 \\ X = Y & \mathbf{Sign} = 0 \\ X > Y & \mathbf{Sign} = -1 \end{cases} \quad \left(\text{where } \frac{\ell_1}{2} \text{ is odd}\right)$$

Bob could figure out the relationship of X and Y.

B. ANALYSIS OF CORRECTNESS

Theorem 3: Given the function

$$P(\triangle) = \begin{cases} +1 & \triangle > 1 \\ 0 & \triangle = 1 \\ -1 & \triangle < 1 \end{cases}$$

*and ℓ_1 ciphertext pairs which is developed by Bob as Protocol **3.1**, Alice can educe the relationship of X and Y ($X > Y$,$X < Y$ or $X = Y$) through analogous decryption and the following equation*

$$\mathbf{Sign} = \prod_{i=1}^{\ell_1} P\left(\frac{\mathcal{L}(c_i^\lambda)}{\mathcal{L}(c_{i+1}^\lambda)}\right). \tag{14}$$

Proof: First, we list two mathematical facts support this theorem as follows.

Fact 1: Given $X, Y, r_{XY}, r_Y \in Z_n^+$, if $\frac{X}{Y}$ is used to detemine the relationship of X and Y, $\frac{r_Y X}{r_Y Y}$ and $\frac{r_Y X + r_{XY}}{r_Y Y + r_{XY}}$ would keep the relationship of X and Y.

Fact 2: Since the function

$$P(\triangle) = \begin{cases} +1 & \triangle > 1 \\ 0 & \triangle = 1 \\ -1 & \triangle < 1 \end{cases}$$

is shared between Alice and Bob, if Bob interfuses a fraction F_N with ω random fractions $(\chi_1, \chi_2, \ldots, \chi_\omega)$ selected randomly beforehand such that half of them are greater than 1, and then sends these $\omega + 1$ fraction to Alice; Alice can figure out that $F_N > 1$, $F_N < 1$ or $F_N = 1$, but cannot figure out which one is F_N.

This is because that half of $\chi_1, \chi_2, \ldots, \chi_\omega$ are greater than 1 but the remaining half are less than 1, so we have

$$R'_{mem} = \prod_{i=1}^{\omega} P(\chi_i) = \begin{cases} +1 & \frac{\omega}{2} \text{ is even} \\ -1 & \frac{\omega}{2} \text{ is odd.} \end{cases}$$

If we add one more fraction to these ω fractions, then we obtain

$$R'_{mem} \times P(\chi_{\omega+1}) = \prod_{i=1}^{\omega} P(\chi_i) \times P(\chi_{\omega+1}) = \begin{cases} +1 & \chi_{\omega+1} > 1 \\ 0 & \chi_{\omega+1} = 1 \\ -1 & \chi_{\omega+1} < 1 \end{cases} \quad (\frac{\omega}{2} \text{ is even})$$

$$R'_{mem} \times P(\chi_{\omega+1}) = \prod_{i=1}^{\omega} P(\chi_i) \times P(\chi_{\omega+1}) = \begin{cases} +1 & \chi_{\omega+1} < 1 \\ 0 & \chi_{\omega+1} = 1 \\ -1 & \chi_{\omega+1} > 1 \end{cases} \quad (\frac{\omega}{2} \text{ is odd})$$

Based on **Fact 1**, through equations (5b), (7) and (8) in **Section 3** and analogous decryption, we obtain

$$P(\frac{\mathcal{L}(c_{r_Y X + r_{XY}}^{\lambda})}{\mathcal{L}(c_{r_Y Y + r_{XY}}^{\lambda})}) = P(\frac{\mathcal{L}(c_{r_Y X}^{\lambda})}{\mathcal{L}(c_{r_Y Y}^{\lambda})}) = P(\frac{\mathcal{L}(c_X^{\lambda})}{\mathcal{L}(c_Y^{\lambda})}) \qquad (15)$$

Based on **Fact 1** and **Fact 2**, through equation (8) and analogous decryption, Alice can determine the relationship of X and Y. After receiving **Sign**, Bob can also know the relationship of X and Y by function $P(\triangle)$. However, he can not evaluate any other information about Alice by function $P(\triangle)$. So the protocol SCEII is correct. □

C. ANALYSIS OF SECURITY

Theorem 4: Protocol SCEII can be used to compare X and Y securely.

Proof: Protocol SCEII is developed to make Alice and Bob figure out the relationship of X and Y($X > Y$, $X = Y$ or $X < Y$) while keeping the privacy of X and Y. The only insecure factor in this comparison protocol is whether it has additional information leakage of the inputs or not. Next, we prove that there is no additional information leakage in the process of comparison.

For Alice's privacy, we construct a simulator $\mathcal{S}_B$ which simulates the protocol by selecting a random X' as the input of Alice, and letting Y as the input of Bob. The view generated by $\mathcal{S}_B$ is $(Y, c_{X'})$ and the view in the real execution is (Y, c_X). $c_{X'}$ and c_X are indistinguishable, which is guaranteed by the fact that Alice's private data transferred to Bob is encrypted by her own public-key $(n, 1+n)$ and the sematic security of the system Π. Thus $\mathcal{S}_B(c_{X'}, Y)$ and the real view $\text{View}_B^{\Pi}(c_X, Y)$ are indistinguishable.

For Bob's privacy, we construct a simulator $\mathcal{S}_A$ to simulate the view of Alice without the private input of Bob. We need the view generated by $\mathcal{S}_A$ being indistinguishable from the view of Alice in the real execution. $\mathcal{S}_A$ simulates as follows.

The input of $\mathcal{S}_A$ are the comparison result **Sign** $\in \{-1, 0, 1\}$ and Alice's private input X. $\mathcal{S}_A$ encrypts X into a ciphertext c_X for the first step. For the second step, Bob evaluates $c_{X+r_{XY}}$ as equation **(8)** and uses it to develop ℓ_1 ordered ciphertext pairs (c_1, c_2), (c_3, c_4), $\ldots$, $(c_{2\ell_1-1}, c_{2\ell_1})$ corresponding to $2\ell_1$ random numbers. The view generated by $\mathcal{S}_A$ is $(X, (c_1, c_2), (c_3, c_4), \ldots, (c_{2\ell_1-1}, c_{2\ell_1}),$ **Sign**).

Becuase $c_{X+r_{XY}}$ is the encryption of the addition of X and r_{XY}, which is evaluated using homomorphic operation on the ciphertext c_X, and $c_{X+r_{XY}}$ is one of $c_1, c_2, \ldots, c_{2\ell_1}$, so the distribution is identical to that of the real execution. □

D. ANALYSIS OF EFFICIENCY

Theorem 5 (Efficiency): Protocol SCEII is three-round and takes at most $2\ell_1 \log n + 2d + 4\ell_1$ modular multiplications.

Proof: Alice sends messages to Bob in **Step 2** and **Step 4**, and Bob sends messages to Alice in **Step 3**. Thus, protocol SCEII is three-round. We neglect the cost of generating the public key in **Step 1** since this can be done in the setup stage. The cost of pre-evaluation with the aid of cloud(s) prior to protocol SCEII are also neglected. In **Step 2**, Alice conducts one encryption: $c_X = (1 + X \cdot n) \cdot R_{1_A} \bmod n^2$, which takes 2 modular multiplications; At the same time, Bob conducts $2\ell_1 - 2$ encryptions, which takes $4\ell_1 - 4$ modular multiplications. In **Step 4**, Bob needs to conduct one modular exponentiations. In **Step 4**, Alice does ℓ_1 analogous decryptions, which takes $2\ell_1 \log n$ modular multiplications. To compare fairly, we convert all operations to the number of modular multiplications [27]. So to privately and cooperatively figure out the relationship of X and Y by protocol SCEII, in total, $2\log n + 4\ell_1 + 4$ modular multiplications and one modular exponentiations should be taken. Computing $c_{X+r_{XY}} = (c_X)^{r_Y} \cdot c_{r_{XY}} \cdot R_Y \pmod{n^2}$ takes at most $2d + 2$ modular multiplications [28]. Overall, protocol SCEII needs $2\ell_1 \log n + d + 4\ell_1$ modular multiplications (see as Table 1). □

TABLE 1. The overload of protocol SCEII.

operations	conducted times by Alice (or Bob)	convert to the number of modular multiplications
Enc	Alice: 1+Bob: $2\ell_1 - 2$	Alice: 2 + Bob:$4\ell_1 - 4$
Ana-Dec	Alice: ℓ_1	$2\ell_1 \log n$
Mod-Exp	Bob: 1	2d+2
total	$(2\ell_1 - 1)$+ℓ_1+1	$2\ell_1 \log n + 2d + 4\ell_1$

**d is the length of the private inputs.
**Enc: is the abbreviation of encryption.
**Ana-Dec: is the abbreviation of analogous decryption.
**Mod-Exp: is the abbreviation of modular exponentiation.
**total: stands for "Enc + Ana-Dec + Mod-Exp".

V. SECURE COMPARISON ON FRACTIONS

In this section, based on the secure relationship-ratio computing system, we first present a new cloud-assisted protocol(denoted as SCOF) for securely comparing fractions. Then, we conduct analysises on its security, correctness and efficiency. To securely and efficiently compare their fractions (Alice with $\frac{a}{b}$ and Bob with $\frac{c}{d}$), Alice and Bob proceed the protocol SCEII as follows.

A. DESCRIPTION OF PROTOCOL SCOF

Step ❶. Alice runs the key generation algorithm to generate her keys, then publishes the public-key $(n,\ 1+n)$ but keeps λ as her private key.

Step ❷. Alice first expresses her private fraction $\frac{a}{b}$ as (a, b), then randomly chooses $R_{Aa}, R_{Ab} \in R_A$ and encrypts a, b as follows.

$$C_a = (1 + a \cdot n) \cdot R_{Aa} \bmod n^2 \tag{16a}$$
$$C_b = (1 + b \cdot n) \cdot R_{Ab} \bmod n^2 \tag{16b}$$

And sends (C_a,C_b) to Bob.

Before receiving (C_a, C_b), Bob needs to do as follows

- Randomly chooses $r_\Delta \in Z_n^+$, $\mathcal{R}_{\mathbf{B_1}} \in R_B$and computes

$$C_{r_\Delta} = (1 + r_\Delta \cdot n) \cdot \mathcal{R}_{\mathbf{B_1}} \pmod{n^2}. \tag{17}$$

- Chooses $\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_{2\ell-2} \in Z_n^+$ $(\ell_2 \geq 1)$ such that half of the numbers $\frac{\Upsilon_1}{\Upsilon_2}, \frac{\Upsilon_3}{\Upsilon_4}, \ldots, \frac{\Upsilon_{2\ell_2-3}}{\Upsilon_{2\ell_2-2}}$ are greater than 1 but the remaining half are less than 1.
- Randomly chooses $k_j \in Z_n$ (where$1 \leq j \leq \ell_2 - 1$) and $\mathcal{R}_{\mathbf{B_{2j+1}}}, \mathcal{R}_{\mathbf{B_{2j+2}}} \in R_B$, and evaluates $\ell_2 - 1$ ciphertext pairs as follows.

$$C_{\Upsilon_{2j+1}} = (1+\kappa_i \cdot \Upsilon_{2j+1} \cdot n \bmod n^2) \cdot \mathcal{R}_{\mathbf{B_{2j+1}}} \bmod n^2 \tag{18a}$$
$$C_{\Upsilon_{2j+2}} = (1+\kappa_i \cdot \Upsilon_{2j+2} \cdot n \bmod n^2) \cdot \mathcal{R}_{\mathbf{B_{2j+2}}} \bmod n^2 \tag{18b}$$

Step ❸. After receiving (C_a, C_b), Bob works as follows.

① First expresses his private fraction $\frac{c}{d}$ as (c, d) and evaluates

$$C'_a = (C_a)^d \bmod n^2 \tag{19a}$$
$$C'_b = (C_b)^c \bmod n^2 \tag{19b}$$

② Randomly chooses $\mathcal{R}_1, \mathcal{R}_2 \in R_B$ and computes $C_{ad+r_\Delta}, C_{bc+r_\Delta}$ as follows.

$$C_{ad+r_\Delta} = C'_a \cdot C_{r_\Delta} \cdot \mathcal{R}_1 \pmod{n}^2 \tag{20a}$$
$$C_{bc+r_\Delta} = C'_b \cdot C_{r_\Delta} \cdot \mathcal{R}_2 \pmod{n}^2 \tag{20b}$$

③ Implements a permutation on these ciphertext pairs $(C_{ad+r_\Delta},\ C_{bc+r_\Delta}), (C_{\Upsilon_3},\ C_{\Upsilon_4}), \ldots, (C_{\Upsilon_{2\ell_2-1}},\ C_{\Upsilon_{2\ell_2}})$, and denotes this permutation as $(C_1, C_2), (C_3, C_4), \ldots, (C_{2\ell_2-1}, C_{2\ell_2})$, and sends them to Alice.

Step ❹. After receiving (C_j, C_{j+1}), Alice evaluates **S***ign* as follows.

$$\mathbf{S}ign = \prod_{j=1}^{\ell_2} P\left(\frac{\mathcal{L}(C_j^\lambda)}{\mathcal{L}(C_{j+1}^\lambda)}\right), \quad \text{where } P(y) = \begin{cases} +1 & y > 1 \\ 0 & y = 1 \\ -1 & y < 1. \end{cases} \tag{21}$$

And sends it to Bob.

TABLE 2. The overload of protocol SCOF.

operations	conducted times by Alice (or Bob)	convert to the number of modular multiplications
Enc	Alice: 2 + Bob: $2\ell_2 - 2$	Alice: 4 + Bob: $4\ell_2 - 4$
Ana-Dec	Alice: ℓ_2	$2\ell_2 \log n$
Mod-Exp	Bob: 1	2d + 2
total	$2\ell_2 + \ell_2 + 1$	$2\ell_2 \log n + 2d + 4\ell_2$+2

**d is the length of the private inputs.
**Enc: is the abbreviation of encryption.
**Ana-Dec: is the abbreviation of analogous decryption.
**Mod-Exp: is the abbreviation of modular exponentiation.
**total: stands for "Enc + Ana-Dec + Mod-Exp".

Step ❺. After receiving **S***ign*, Bob can figure out the relationship of $\frac{a}{b}$ (belonged to Alice) and $\frac{c}{d}$ (belonged to Bob) by

$$\mathbf{R}_{\ell_{2\mathbf{e}}} = \begin{cases} \frac{a}{b} > \frac{c}{d} & \mathbf{S}ign = +1 \\ \frac{a}{b} = \frac{c}{d} & \mathbf{S}ign = 0 \\ \frac{a}{b} < \frac{c}{d} & \mathbf{S}ign = -1 \end{cases} \quad (\text{where } \frac{\ell_2}{2} \text{ is even or } \ell_2 = 1)$$

Or

$$\mathbf{R}_{\ell_{2\mathbf{o}}} = \begin{cases} \frac{a}{b} < \frac{c}{d} & \mathbf{S}ign = +1 \\ \frac{a}{b} = \frac{c}{d} & \mathbf{S}ign = 0 \\ \frac{a}{b} > \frac{c}{d} & \mathbf{S}ign = -1 \end{cases} \quad (\text{where } \frac{\ell_2}{2} \text{ is odd}).$$

B. ANALYSIS OF CORRECTNESS

Fact 3: Given $\frac{X_1}{Y_1}$ *and* $\frac{X_2}{Y_2}$ $(X_1, Y_1, X_2, Y_2 \in Z_n^+)$, $\frac{X_1 Y_2}{X_2 Y_1}$ *could be used to determine the relationship of* $\frac{X_1}{Y_1}$ *and* $\frac{X_2}{Y_2}$, *and* $\frac{X_1 Y_2 + r_F}{X_2 Y_1 + r_F}$ $(r_F \in Z_n^+)$ *would keep the relationship of* $\frac{X_1}{Y_1}$ *and* $\frac{X_2}{Y_2}$.

Based on **Fact 3**, through equations (14a), (14b), (15a) and (15b) in **Section 5.1** and analogous decryption $\frac{\mathcal{L}(C^\lambda_{ad+r_\Delta})}{\mathcal{L}(C^\lambda_{bc+r_\Delta})}$ (or $\frac{\mathcal{L}(C^\lambda_{bc+r_\Delta})}{\mathcal{L}(C^\lambda_{ad+r_\Delta})}$), we can draw a conclusion that

$$P\left(\frac{\mathcal{L}(C^\lambda_{ad+r_\Delta})}{\mathcal{L}(C^\lambda_{bc+r_\Delta})}\right) = P\left(\frac{\mathcal{L}(C^\lambda_{ad})}{\mathcal{L}(C^\lambda_{bc})}\right). \tag{22}$$

Based on **Fact 1** and **Fact 3**, through equation (16) and analogous decryption, Alice could know the relationship of $\frac{a}{b}$ and $\frac{c}{d}$, but she cannot compute any additional information about Bob. This is because Alice can only obtain ℓ_2 random fractions through analogous decryption. After receiving **Sign**, Bob can also figure out the relationship of $\frac{a}{b}$ and $\frac{c}{d}$ by $\mathbf{R}_{\ell_{2\mathbf{e}}}$ or $\mathbf{R}_{\ell_{2\mathbf{o}}}$. However, he cannot evaluate any additional information about Alice by this work. So the correctness is completed.

TABLE 3. Performance comparison.

Protocol	Comparison Range	Date Types	Total Computing Cost	Technique
Lin[27]	d	integer	$5d\log n + 4d - 6$	encoding
Luo[28]	–	–	–	difference computing
Li[30]	d	integer	$2d\log n + d - 1$	encoding
Veugen[29]	–	–	–	difference computing
Liu[32]	–	–	–	difference computing
Li[34]	–	–	–	difference computing
SCEII	$n-1$	integer	$2\ell_1\log n + 2d + 4\ell_1$	ratio computing
SCOF	$n-1$	fractions	$2\ell_2\log n + 4d + 4\ell_2 + 4$	ratio computing

**computation cost is measured in the number of modular multiplication and communication cost is measured in round.
**d is the length of the private inputs.
**$\mathcal{P}$ is the modulus in ElGamal encryption[11] and we may set $\log\mathcal{P} = 512$.
**n is the modulus in Paillier's homomorphic encryption scheme[11] and we may set $\log n = 2048$.
**– represents computing difference between two numbers with the wrong utilization of Paillier encryption scheme.

C. ANALYSIS OF SECURITY

Theorem 6: Protocol SCOF can be used to compare $\frac{a}{b}$ and $\frac{c}{d}$ securely.

Proof: Protocol SCOF is designed to make Alice and Bob know $\frac{a}{b} > \frac{c}{d}$, $\frac{a}{b} = \frac{c}{d}$ or $\frac{a}{b} < \frac{c}{d}$ while keeping the privacy of $\frac{a}{b}$ and $\frac{d}{c}$. The only insecure factor in this comparison protocol is whether it has additional information leakage of the inputs or not. Next we show that there is no additional information leakage in the process of comparison.

Assuming Π is the protocol SCOF that we developed to compute function $F = (F_1, F_2) : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ on inputs $(a,b),(c,d)$. The view of Alice is denoted as $View_{\mathbb{A}}^{\Pi}((a,b),(c,d))$ and her output is denoted as $Output_{\mathbb{A}}^{\Pi}((a,b),(c,d))$. Similarly, the view and output of Bob are denoted by $View_{\mathbb{B}}^{\Pi}((a,b),(c,d))$ and $Output_{\mathbb{B}}^{\Pi}((a,b),(c,d))$, respectively.

For Alice's privacy, we construct a simulator $\mathcal{S}_{\mathbb{B}}$ which simulates the protocol by selecting a randomness $\frac{a'}{b'}$ as the input of Alice, and letting $\frac{c}{d}$ as the input of Bob. The view generated by $\mathcal{S}_{\mathbb{B}}$ is $((C_c, C_d),(C_{a'}, C_{b'}))$ and the view in the real execution is $((C_c, C_d),(C_a, C_b))$. $(C_{a'}, C_{b'})$ and (C_a, C_b) are indistinguishable, which is guaranteed by the fact that Alice's private data transferred to Bob is encrypted by his own public-key $(n,\ 1+n)$ and the sematic security of the system Π. Therefore, $\mathcal{S}_{\mathbb{B}}((C_{a'}, C_{b'}),(C_c, C_d))$ and the real view $\text{View}_{\mathbb{B}}^{\Pi}((C_a, C_b),(C_c, C_d))$ are indistinguishable. Namely, the simulator $\mathcal{S}_{\mathbb{B}}$ constructed as above makes

$$\{F_1((a',b'),(c,d)), \mathcal{S}_{\mathbb{B}}((c,d), F_2((a',b'),(c,d)))\}_{(a,b),(c,d)} \overset{c}{\equiv} \{Output_{\mathbb{A}}^{\Pi}((a,b),(c,d)), View_{\mathbb{B}}^{\Pi}((a,b),(c,d))\}_{(a,b),(c,d)} \tag{23}$$

For Bob's privacy, we construct a simulator $\mathcal{S}_{\mathbb{A}}$ to simulate the view of Alice without the private input of Bob. We need the view generated by $\mathcal{S}_{\mathbb{A}}$ being indistinguishable from the view of Alice in the real execution. $\mathcal{S}_{\mathbb{A}}$ simulates as follows.

The input of $\mathcal{S}_{\mathbb{A}}$ are the comparing result **Sign** $\in \{-1, 0, 1\}$ and Alice's private input $\frac{a}{b}$. $\mathcal{S}_{\mathbb{A}}$ expresses $\frac{a}{b}$ as an ordered pair (a,b) for the first step; and then it encrypts a and b into ciphertexts C_a and C_b. In the third step, Bob evaluates $C_{ad'+r_{\triangle'}}$ and $C_{bc'+r_{\triangle'}}$ as **Step ❸** in Section **4.1** and ciphertext pairs $(C_{\Upsilon_3},\ C_{\Upsilon_4}), \ldots, (C_{\Upsilon_{2\ell_2-1}},\ C_{\Upsilon_{2\ell_2}})$ corresponding to $2\ell_2 - 2$ random numbers as **Step ❷** in Section **4.1**. The view generated by $\mathcal{S}_{\mathbb{A}}$ is $((a,b),(C'_1, C'_2),\ (C'_3, C'_4), \ldots,\ (C'_{2\ell_2-1}, C'_{2\ell_2}),$ **Sign**).

Becuase $C_{ad'+r_{\triangle'}}$ and $C_{bc'+r_{\triangle'}}$ are evaluated by homomorphic operation, where Alice only know a, b and some dubious information that the ciphertext pair $(C_{ad'+r_{\triangle'}}, C_{bc'+r_{\triangle'}})$ is one of $(C'_1, C'_2),\ (C'_3, C'_4), \ldots,\ (C'_{2\ell_2-1}, C'_{2\ell_2})$, so the distribution is identical to that in the real execution. Namely, the simulator $\mathcal{S}_{\mathbb{A}}$ constructed as above makes

$$\{\mathcal{S}_{\mathbb{A}}((a,b), F_1((a,b),(c',d'))), F_2((a,b),(c',d'))\}_{(a,b),(c,d)} \overset{c}{\equiv} \{View_{\mathbb{A}}^{\Pi}((a,b),(c,d)), Output_{\mathbb{B}}^{\Pi}((a,b),(c,d))\}_{(a,b),(c,d)} \tag{24}$$

□

D. ANALYSIS OF EFFICIENCY

Theorem 7 (Efficiency): Protocol SCOF is three-round and takes at most $2\ell_2\log n + 4d + 2$ modular multiplications.

Proof: Alice transfers messages to Bob in **Step ❷** and **Step ❹**, and Bob transfers message to Alice in **Step ❸**. Thus, protocol SCOF is three-round. We neglect the cost of generating a public key in **Step ❶** since this can be done in the setup stage. The cost of pre-evaluation with the aid of cloud(s) prior to protocol SCOF are also neglected. In **Step ❷**, Alice conducts two encryptions $C_a = (1 + a\cdot n)\cdot R_{Aa} \bmod n^2$, $C_b = (1 + b\cdot n)\cdot R_{Ab} \bmod n^2$; At the same time, Bob needs to develop $2\ell_1 - 2$ encryptions. In **Step ❸**, Bob conducts two modular exponentiations. In **Step ❹**, Alice conducts ℓ_2 analogous decryptions. To compare fairly, we convert all operations to the number of modular multiplications [27]. Hence, to compare $\frac{a}{b}$ and $\frac{c}{d}$ by protocol SCOF, in total, $2\ell_2$ encryption and ℓ_2 analogous decryptions and two modular exponentiations should be taken. For our homomorphic-encryption-analogous-decryption system, each encryption requires 2 modular multiplications, and each decryption also requires $2\log n$ modular multiplications. Computing $C_{ad+r_\Delta} = C'_a \cdot C_{r_\Delta} \cdot \mathcal{R}_1 \pmod n^2$ or $C_{bc+r_\Delta} = C'_b \cdot C_{r_\Delta} \cdot \mathcal{R}_2 \pmod n^2$ takes at most $2d+2$ modular multiplications [28]. Overall, protocol SCOF needs $2\ell_2\log n + 4d + 4\ell_2 + 2$ modular multiplications (see as Table 2). □

VI. PERFORMANCE ANALYSES

Except protocols [28], [32], [34], the existing solutions to the millionaires' problem are all for integers, thus they cannot be compared with our protocols in an objective manner. However, for a wrong utilization that using Paillier encryption scheme to directly encrypt an negative number, protocols [28], [29], [32], [34] have failed to solve what as they claimed.

We compare the protocols [27]–[30], [32], [34] based on the same cryptographic computational problem, DCR, as our protocols. To compare fairly, we convert all operations of these protocols (based on DCR) to the number of modular multiplications, respectively. In comparison with previous methods [27], [30], although protocol SCEII is 3-round, it reduces the number of modular computation significantly. Moreover, protocol SCEII can determine whether $x > y, x < y$ or $x = y$ (where x, y are integers less than $n - 1$)in one execution, and protocol SCOF can figure out whether $x > y, x < y$ or $x = y$ (whre x, y are fractions less than $n - 1$) in one execution, while previous Li's method [30] can only solve the greater than problem. In comparison with Li [30], our method cannot only be used to securely compare two integers but also two fractions. On the efficiency, in comparison with previous methods [27], [30], SCEII and SCOF have considerable advantages on total computing cost when $\ell_1, \ell_2 \in \{0, 2\}$. The detailed comparisons with previous methods are listed as Table 3.

VII. CONCLUSION

Comparing information privately is the fundamental to secure multiparty computation and plays an important role in developing practical SMC protocols. Although there are several protocols claimed that the millionaires' problem for rational numbers had been solved, these solutions are supported by a wrong understanding that Paillier encryption scheme could encrypt a negative number (However, the encryption function $f(x, y) = (1 + kn)^x y^n \bmod n^2$ is bijective if and only if $(x \in Z_n) \wedge (y \in Z_n^*)$ by reference to the detailed analysis in paper [38], namely, Paillier encryption scheme can not be directly used to encrypt negative numbers). In fact, the existing solutions are all designed for integers and they fail to meet the needs of many practical applications. In this study, we extend the millionaires' problem for fractions, where we skilfully combine "ratio" method and propose efficient protocols to obtain solutions. We have shown the security of these protocols. In future research, we aim to extend the solution to the millionaires' problem further to solve more practical SMC problems that involve ratio computation.

REFERENCES

[1] S. Mawet, "Negotiations using secure multi-party computation," Ph.D. dissertation, Dept. Inst. Inf. Commun. Technol., Electron. Appl. Math., Univ. Catholique Louvain, Louvain-la-Neuve, Belgium, 2015.

[2] W. Li, M. Larson, and C. Hu, "Secure multi-unit sealed first-price auction mechanisms," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3833–3843, 2016.

[3] A. Thapa, W. Liao, M. Li, P. Li, and J. Sun, "SPA: A secure and private auction framework for decentralized online social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 8, pp. 2394–2407, Aug. 2016.

[4] A. Aly and V. M. Van, "Practically efficient secure single-commodity multi-market auctions," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2016, pp. 110–129.

[5] H. Huang, X.-Y. Li, Y.-E. Sun, H. Xu, and L. Huang, "PPS: Privacy-preserving strategyproof social-efficient spectrum auction mechanisms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1393–1404, May 2015.

[6] M. Larson, R. Li, C. Hu, W. Li, X. Cheng, and R. Bie, "A bidder-oriented privacy-preserving VCG auction scheme," in *Proc. Int. Conf. Wireless Algorithms, Syst.*, 2015, pp. 284–294.

[7] A. Abdelhadi, H. Shajaiah, and C. Clancy, "A multitier wireless spectrum sharing system leveraging secure spectrum auctions," *IEEE Trans. Cogn. Commun. Netw.*, vol. 1, no. 2, pp. 217–229, Jun. 2015.

[8] Q. Huang, Y. Gui, F. Wu, Q. Zhang, and G. Chen, "A general privacy-preserving auction mechanism for secondary spectrum markets," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1881–1893, Jun. 2016.

[9] J. A. Montenegro and J. Lopez, "A practical solution for sealed bid and multi-currency auctions," *Comput. Secur.*, vol. 45, pp. 186–198, Sep. 2014.

[10] S. D. Li and Y. Q. Dai, "Secure two-party computational geometry," *J. Comput. Sci. Technol.*, vol. 20, no. 2, pp. 258–263, 2005.

[11] L. Shundong, W. Chunying, W. Daoshun, and Y. Dai, "Secure multiparty computation of solid geometric problems and their applications," *Inf. Sci.*, vol. 282, pp. 401–413, Oct. 2014.

[12] L. Yonglong, H. Liusheng, Z. Hong, and C. Guoliang, "A secure protocol for determining whether a point is inside a convex polygon," *Chin. J. Electron.*, vol. 15, no. 4, pp. 578–582, 2006.

[13] B. Mu and S. Bakiras, "Private proximity detection for convex polygons," *Tsinghua Sci. Technol.*, vol. 21, no. 3, pp. 270–280, Jun. 2016.

[14] X. Yi, F. Y. Rao, and E. Bertino, "Privacy-preserving association rule mining in cloud computing," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, 2015, pp. 439–450.

[15] S. Rane and P. T. Boufounos, "Privacy-preserving nearest neighbor methods: Comparing signals without revealing them," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 18–28, Mar. 2013.

[16] X. S. Wang, Y. Huang, and Y. Zhao, H. Tang, X. Wang, and D. Bu, "Efficient genome-wide, privacy-preserving similar patient query based on private edit distance," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. ACM*, 2015, pp. 492–503.

[17] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.

[18] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1200–1210, May 2014.

[19] A. Abidin, A. Aly, and S. Cleemput, "An MPC-based privacy-preserving protocol for a local electricity trading market," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, 2016, pp. 615–625.

[20] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2004, pp. 1–19.

[21] M. J. Freedman, C. Hazay, and K. Nissim, "Efficient set intersection with simulation-based security," *J. Cryptol.*, vol. 29, no. 1, pp. 115–155, 2016.

[22] I. F. Blake and V. Kolesnikov, "Strong conditional oblivious transfer and computing on intervals," in *Advances in Cryptology—ASIACRYPT*. Berlin, Germany: Springer, 2004, pp. 515–529.

[23] A. C. Yao, "Protocols for secure computations," in *Proc. FOCS*, vol. 82. 1982, pp. 160–164.

[24] C. Cachin, "Efficient private bidding and auctions with an oblivious third party," in *Proc. 6th ACM Conf. Comput. Commun. Secur.*, 1999, pp. 120–127.

[25] M. Fischlin, "A cost-effective pay-per-multiplication comparison method for millionaires," in *Topics in Cryptology—CT-RSA*. Berlin, Germany: Springer, 2001, pp. 457–471.

[26] I. Ioannidis and A. Grama, "An efficient protocol for Yao's millionaires' problem," in *Proc. IEEE 36th Annu. Hawaii Int. Conf.*, Jan. 2003, p. 6.

[27] H. Y. Lin and W. G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," *Applied Cryptography and Network Security*. Berlin, Germany: Springer, 2005, pp. 456–466.

[28] Y. Luo, L. Huang, and W. Yang, "An efficient protocol for private comparison problem," *Chin. J. Electron.*, vol. 18, no. 2, pp. 205–209, 2009.

[29] T. Veugen, F. Blom, and S. J. A. de Hoogh, "Secure comparison protocols in the semi-honest model," *IEEE J. Sel.Topics Signal Process.*, vol. 9, no. 7, pp. 1217–1228, Oct. 2015.
[30] S. D. Li and D.-S. Wang, "Efficient secure multiparty computation based on homomorphic encryption," *Acta Electronica Sinica*, vol. 41, no. 4, pp. 798–803, 2013.
[31] T. Nakai, Y. Tokushige, Y. Misawa, M. Iwamoto, and K. Ohta, "Efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, 2016, pp. 500–517.
[32] X. Liu, S. Li, and X. B. Chen, "Efficient solutions to two-party and multi-party millionaires' problem," *Secur. Commun. Netw.*, vol. 39, May 2017, Art. no. 5207386.
[33] M. Hezaveh and C. Adams, "An efficient solution to the socialist millionaires' problem," in *Proc. IEEE 30th Can. Conf. Elect. Comput. Eng. (CCECE)*, Apr./May 2017, pp. 1–4.
[34] S. Li, Y. Guo, S. Zhou, J. Dou, and D. Wang, "Efficient protocols for the general millionaires' problem," *Chin. J. Electron.*, vol. 26, no. 4, pp. 696–702, 2017.
[35] D. Grigoriev, L. B. Kish, and V. Shpilrain, "Yao's millionaires' problem and public-key encryption without computational assumptions," *Int. J. Found. Comput. Sci.*, vol. 28, no. 04, pp. 379–389, 2017.
[36] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[37] Y. Aumann and Y. Lindell, "Security against covert adversaries: Efficient protocols for realistic adversaries," *J. Cryptol.*, vol. 23, no. 2, pp. 281–343, 2010.
[38] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 1999, pp. 223–238.

LINMING GONG received the Ph.D. degree from the School of Computer Science, Shaanxi Normal University. He is currently a Teaching Fellow with the School of Computer Science, Xi'an Polytechnic University, Xi'an, China. His current research interests include applied cryptography, secure multiparty computation, computer and network security, mobile and wireless communication security, and privacy-preserving data mining.

SHUNDONG LI received the Ph.D. degree from the School of Computer Science, Xi'an Jiaotong University. He was an Associate Professor with the School of Computer Science, Beijing Normal University, Beijing, China. He is currently a Professor of computer science with Shaanxi Normal University, Xi'an, China. His current research interests include secure multiparty computation, computer and network security, and privacy-preserving data mining.

CHUNYING WU received the Ph.D. degree from the School of Computer Science, Shaanxi Normal University. She is currently an Associate Professor of computer science with Shanghai Normal University, Shanghai, China. Her current research interests include applied cryptography and secure multiparty computation.

DAOSHUN WANG received the Ph.D. degree from the College of Mathematics, Sichuang University. He is currently an Associate Professor of computer science, Tsinghua University, Beijing, China. His current research interests include applied cryptography, secret sharing, and computer and network security.