Asymmetric Commutative Encryption Scheme Based Efficient Solution to the Millionaires' Problem

Meng Liu*, Priyadarsi Nanda†, Xuyun Zhang‡, Chi Yang§, Shui Yu¶, Jianxin Li‖
*School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China
†School of Electrical and Data Engineering, University of Technology Sydney, Australia
‡Department of Electrical and Computer Engineering, University of Auckland, New Zealand
§School of Computing and Information Technology, University of Wollongong, Australia
¶School of Information Technology, Deakin University, Burwood, Australia
‖School of Computer Science and Software Engineering, University of Western Australia, Australia
Email: liumeng@sdu.edu.cn, Priyadarsi.Nanda@uts.edu.au, xuyun.zhang@auckland.ac.nz,
chiyangit@gmail.com, syu@deakin.edu.au, jianxin.li@uwa.edu.au

***Abstract*—Secure multiparty computation (SMC) is an important scheme in cryptography and can be applied in various real-life problems. The first SMC problem is the millionaires' problem which involves two-party secure computation. Because the efficiency of public key encryption scheme appears less than symmetric encryption scheme, most existing solutions based on public key cryptography to this problem is inefficient. Thus, a solution based on the symmetric encryption scheme has been proposed. Although it is claimed that this approach can be efficient and practical, we discover that there exist several severe security flaws in this solution. In this paper, we analyze the vulnerability of existing solutions, and propose a new scheme based on the Decisional Diffie-Hellman hypothesis (DDH). Our solution also uses two special encodings (0-encoding and 1-encoding) generated by our modified encoding method to reduce the computation cost of modular multiplications. Extensive experiments are conducted to evaluate the efficiency of our solution, and the experimental results show that our solution can be much more efficient and be approximately 8000 times faster than the solution based on symmetric encryption scheme for a 32-bit input and short-term security. Moreover, our solution is also more efficient than the state-of-the-art solution.**

I. Introduction

Secure multiparty computation (SMC) was first proposed by Yao in 1982 [1]. The goal of SMC is to enable parties to jointly compute a function over their inputs without revealing these private inputs. For example, a given number of parities p_1, p_2, ..., p_n, all participants have a private input data, respectively d_1, d_2, ..., d_n. They want to compute the value of a public function f on n variables (d_1, d_2, ..., d_n). An SMC protocol is secure if no participant can learn more than what he/she can learn from his/her own input from the public function and the result. SMC appears as an essential problem in cryptography and its solutions have been utilized in cooperative scientific computation, data mining, privacy-preserving clustering [2], bidding and auction in e-commerce [3], [4], secure computational geometry [5]–[8], set intersection [9], [10], secure statistical analysis [11], privacy-preserving image retrieval [12]–[14] and secure data aggregation in smart metering systems [15].

The first SMC problem is Yao's Millionaires' problem. It is a secure two-party computations problem and it has served as an important building blocks in some solutions [12], [16]–[20]. The problem discusses two millionaires, Alice and Bob, who want to know which of them is richer without disclosing their actual wealth. This problem is analogous to a more general problem where Alice and Bob have their private inputs, x and y, and they want to determine the predicate $x > y$ without revealing the actual values of x and y. The first solution to millionaires' problem is presented by Yao himself. For the n-bit numbers x and y, it needs 1 time public key encryption, 2^n times public key decryptions, 2^n times modular operation, and at least $2^{2n}/2$ times verifications. So it is exponential in time and space and too expensive to be practical.

Many other solutions have been proposed to solve Millionaires' problem. Ioannidis et al. [21] use 1-out-of-2 oblivious transfer scheme to construct a protocol that runs n times of the OT scheme, where n is the length of the private inputs. The implementation of the 1-out-of-2 oblivious transfer based on public key cryptography needs 4 times public key encryptions and 2 decryptions. Let $N = 2^n$ be the maximal value of the input, it needs $4 \log N$ public key encryptions and $2 \log N$ public key decryptions. Lin et al. [22] propose a two-round protocol to solve the Millionaires' Problem. Their protocol uses multiplicative homomorphic encryption scheme and is more efficient than an additive one practically. It can save computation time and communication bandwidth in practicality. Let p be the modulo prime, the solution in [22] takes $5n \log p$ modular multiplications. And the size of exchanged messages is $6n \log p$ bits in [22].

Shundong et al. [23] propose a symmetric cryptographic solution to the millionaires' problem based on set-inclusion problems using a commutative encryption scheme and claims that it is more efficient for practical applications than known solutions and is capable of greatly reducing the computational cost. Unfortunately, we have discovered that the solution has some security flaws and is not more efficient than our protocol based on public key cryptography when the size of the input is large.

Veugen et al. [24] have analyzed the state-of-the-art comparison protocols. In terms of execution time, they point that Damgard's solution [25] developed on the basis of the dedicated DGK homomorphic encryption scheme outperforms the other protocols. Nevertheless, this solution has an initialization time of approximately 150 seconds for medium term security.

Our contributions can be summarized as follows:

1) We analyze some security flaws of Shundong's symmetric cryptographic solution to millionaires' problem and this solution is not more efficient and practical than some previous solutions.

2) We introduce a new solution based on the Decisional Diffie-Hellman hypothesis as well as the set intersection problem to the millionaires' problem. Experimental results show that our solution is more efficient and practical.

3) We further improve our solution by reducing the size of the random secret keys without compromise security. Consequently, our solution is also more efficient than the state-of-the-art solution.

The rest of the paper is organized as follows. Section II provides some discussions on Shundong's symmetric cryptographic solution to millionaires' problem. In Section III we propose our solution to Yao's millionaires' problem. In Section IV, we demonstrate security analysis and proof to our solution. Section V possesses experiment results as well as analyses. Ultimately, the paper will be concluded in Section VI.

II. Some Discussions on Shundong's symmetric cryptographic solution to millionaires' problem

Shundong's symmetric cryptographic solution to millionaires' problem is proposed based on a private set-inclusion problem. This problem can be formally defined as follows: Alice has a private number x, and Bob has a private set $X = \{x_1, x_2, x_3, ..., x_n\}$. Alice and Bob need to know whether $x \in X$ without disclosing their private data either x or X to the counterpart. It can be solved with a commutative encryption scheme that has been made for the purpose of determining whether the two numbers are equal [26]. The commutative encryption scheme can be either an asymmetric encryption scheme or a symmetric encryption scheme. In fact, the set-inclusion problem is a special case of the private intersection problem. Agrawal et al. [27] propose a solution with a commutative encryption based on the Decisional Diffie-Hellman hypothesis to solve the private intersection problem. Neither of the two parties could learn the other party's information outside of the intersection because of lacking necessary key information. A similar protocol is also proposed by Li et al. based on public key cryptography [28]. However, the two solutions suffer the same questions of more computational complexity and can also reveal $|X|$. Consequently, Shundong et al. introduce a new solution based on symmetric encryption scheme, which is a commutative encryption scheme. And the new solution can be efficient and maintains the privacy of $|X|$ [23].

In simple terms, a commutative encryption scheme must satisfy that $E_a(E_b(x)) = E_b(E_a(x))$, where E is a encryption function and a and b are two specified keys. First, the formal protocol with a commutative encryption to the set-inclusion problem has been defined and found in [23]. Through the applications of a commutative scheme, a symmetric encryption solution based on set-inclusion Protocol to the set-inclusion problem has also been proposed in reference [23]. Shundong et al. have analyzed the security of their protocol and proved that it is secure in reference [23]. But in fact, their protocol exhibits some important security drawbacks.

We discover a definition flaw in Shundong's Protocol. If the cardinality $|U|$ of the set U is even, Bob could not determine the subset A from $\{X, \bar{X}\}$ because the cardinality of the set X could equal to the cardinality of the set $\bar{X}$ according to Protocol 1. A simply solution is that an element $y \notin U$ can be added into the set X. So the cardinality of the new set X' is odd and it will not influence the result. For simplicity, this flaw will not be considered for later discussion.

There exists another important security drawback in Shundong's Protocol. It is found by Xie et al. [9]. They find that Alice could easily explore Bob's whole set X if Alice has known the set U. For each element $e \in U$, Alice can easily find r'_i such that $e \oplus r'_i = x \oplus r_i$. When Alice receives the two sets of D and $\pi(E) = \{e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(t)}\}$ from Bob in step 5, instead of computing $\{b_1 \oplus s_1 \oplus r_1, b_1 \oplus s_2 \oplus r_2, ..., b_t \oplus s_t \oplus r_t\}$, she could compute $G' = D \oplus R' = \{b_1 \oplus s_1 \oplus r'_1, b_1 \oplus s_2 \oplus r'_2, ..., b_t \oplus s_t \oplus r'_t\}$. If $|\pi(E) \cap G'| = 1$, then $e \in B$. So Alice can determine the set X according to the result of Shundong's protocol.

Since Shundong's symmetric encryption solution to the set-inclusion problem has some drawbacks, their symmetric solution to millionaires' problem is also broken.

III. Our solution to Yao's millionaires' problem

Some previous work based on homomorphic encryption have studied and proposed some efficient protocols to Yao's millionaires' problem. Blake et al. [29] use the additive homomorphic Paillier cryptosystem to construct a two-round protocol to Yao's millionaires' problem. The computation cost is $O(n \log N)$ and the communication cost is $O(n \log N)$. Lin et al. [22] also propose a two-round protocol for solving the Millionaires' Problem using the multiplicative homomorphic encryptions and a special coding for the private inputs. Since multiplicative homomorphic encryption scheme is more efficient than an additive one practically, their solution saves computation time and communication bandwidth in practicality. The ElGamal encryption scheme is a multiplicative homomorphic encryption scheme with the scalaring property. And the Paillier encryption scheme is an additive homomorphic encryption scheme. For efficiency of computation, they modify the scheme so that each decryption takes 1 modular exponentiation without affecting the security of the scheme. Shundong et al. propose to use the XOR operation as the symmetric commutative function and the solution can sharply reduce the computational overhead. Unfortunately, it does have some security flaws. Generally, we have two policies to reduce the computational overhead. The first one is to employ a symmetric encryption scheme, and the second one is to employ an asymmetric encryption scheme but we can greatly reduce the computational number of modular multiplications.

A. 0-encoding and 1-encoding

The main idea reducing the computational number of modular multiplications is to reduce the scale of the set intersection problem. Lin et al. [22] use two special encodings, 0-encoding and 1-encoding.

Let $x = x_n x_{n-1} ... x_1 \in \{0,1\}^n$ be a binary string of length n. The 0-encoding of x is the set S_x^0 of binary string x such that

$$S_x^0 = \{x_n x_{n-1} ... x_{i+1} 1 | x_i = 0, 1 \leq i \leq n\}$$

The 1-encoding of x is the set S_x^1 of binary string such that

$$S_x^1 = \{x_n x_{n-1} ... x_i | x_i = 1, 1 \leq i \leq n\}$$

Both S_x^1 and S_x^0 have at most n elements.

We can encode x into its 1-encoding S_x^1 and y into its 0-encoding S_y^0.

Theorem 1: x is greater than y if and only if S_x^1 and S_y^0 have a common element.

The proof of theorem 1 and more information about 0-encoding and 1-encoding can be found in [22].

We give an example. Let $x = 10 = 1010_2$ and $y = 6 = 0110_2$ of length 4 (we fill in the leading zeros). We have $S_x^1 = \{1, 101\}$ and $S_y^0 = \{1, 0111\}$. Since $S_x^1 \cap S_y^0 \neq \emptyset$, we have $x > y$. And if $x = 6 = 0110_2$ and $y = 10 = 1010_2$,we have $S_x^1 = \{01, 011\}$ and $S_y^0 = \{11, 1011\}$.Since $S_x^1 \cap S_y^0 = \emptyset$, we have $x \leq y$.

In order to construct our solution, we redefine a new 0-encoding and 1-encoding.

Definition 1: Let $x = x_n x_{n-1} ... x_1 \in \{0,1\}^n$ be a binary string of length n. The 0-encoding of x is the set S_x^0 of binary numbers x such that

$$S_x^0 = \{x_n x_{n-1} ... x_{i+1} 1 \underbrace{0_{i-1}, ..., 0_2, 0_1}_{i-1} | x_i = 0, 1 \leq i \leq n\}$$

The 1-encoding of x is the set S_x^1 of binary numbers such that

$$S_x^1 = \{x_n x_{n-1} ... x_i \underbrace{0_{i-1}, ..., 0_2, 0_1}_{i-1} | x_i = 1, 1 \leq i \leq n\}$$

Both S_x^1 and S_x^0 have at most n elements.

We also give an example. Let $x = 10 = 1010_2$ and $y = 6 = 0110_2$ of length 4. We have $S_x^1 = \{1000_2, 1010_2\} = \{8, 10\}$ and $S_y^0 = \{1000_2, 0111_2\} = \{8, 7\}$. Since $S_x^1 \cap S_y^0 \neq \emptyset$, we have $x > y$. And if $x = 6 = 0110_2$ and $y = 10 = 1010_2$,we have $S_x^1 = \{0100_2, 0110_2\} = \{4, 6\}$ and $S_y^0 = \{1100_2, 1011_2\} = \{1100_2, 1011_2\} = \{12, 11\}$. Since $S_x^1 \cap S_y^0 = \emptyset$, we have $x \leq y$.

B. Our commutative encryption scheme

We propose a commutative encryption scheme solution to Yao's millionaires' problem. The commutative encryption scheme is constructed based on the Decisional Diffie-Hellman hypothesis. Our commutative encryption scheme requires 1 modular exponentiation for each party, so it is more efficient than multiplicative homomorphic encryption scheme.

Definition 2: Let M denote a message space and K denote a key space. A commutative encryption function is a computable (in polynomial time) and bijection function $f : M \times K \to M$ that satisfies that we have $f_b \circ f_a(m)$=$f_a \circ f_b(m)$, for a given $m \in M$, any $a, b \in F$.

Fact 1: Let M be the group of quadratic residues modulo a prime p, where p is a large 'safe' prime number, i.e., both p and $q = (p-1)/2$ are large primes. Let K be $\{1, 2, ..., q-1\}$. According to Decisional Diffie-Hellman hypothesis, the power function

$$f_e(m) \equiv m^e \bmod p$$

is commutative encryption function.

(1) $f_b \circ f_a(m) = (m^a \bmod p)^b \bmod p$=$m^{ab} \bmod p = (m^b \bmod p)^a \bmod p = f_a \circ f_b(m)$

(2) Each of the powers f_e is a bijection.

The DDH is a computational hardness assumption about a certain problem of discrete logarithms in cyclic groups. So the security of f_e depends on the computational difficulty of discrete logarithm problem. Let h denote a public collision-free hash function.

Protocol 1: Our solution to Yao's millionaires' problem

Inputs:

Alice: x and a large safe prime p, where $0 < x < 2^n, n = \lfloor \log(p-1)/2 \rfloor$

Bob: y and a large safe prime p, where $0 < y < 2^n, n = \lfloor \log(p-1)/2 \rfloor$

Output:

whether $x > y$.

1. Alice generates a random secret key a, where a is a large number and $a < (p-1)/2$, then Alice computes S_x^1 and

for each $s \in S_x^1$ computes $S = S \cup \{f_a(h(s)^2)\}$.

Alice prepares $l_1 = n - |S_x^1|$ random numbers z_j and combines them to the set S, where $z_j \in Z_q^*$, $1 \leq j \leq l_1$ and each z_j is unique.

Alice generates a random permutation of S, expressed by $\pi_1(S)$ and sends $\pi_1(S)$ to Bob.

2. Bob generates a random secret key b, where b is a large number and $b < (p-1)/2$, then Bob gets $\pi_1(S)$ and computes $G = f_b(\pi_1(S))$.

Bob computes S_y^0 and

for each $r \in S_y^0$ computes $R = R \cup f_b(h(r)^2)$.

Bob prepares $l_2 = n - |S_y^0|$ random numbers z_j and combines them to the set R, where $z_j \in Z_q^*$, $1 \leq j \leq l_1$ and each z_j is unique.

Bob generates a random permutation of R, expressed by $\pi_2(R)$ and sends $\pi_2(R)$ to Alice.

Bob generates a random permutation of G, expressed by $\pi_3(G)$ and sends $\pi_3(G)$ to Alice.

3. Alice gets $\pi_2(R)$ and computes $H = f_a(\pi_2(R))$.

If $|H \cap \pi_3(G)| = 1$, Bob concludes that $x > y$ and $x \leq y$ otherwise. Bob tells Alice the result.

Our solution is constructed based on the Decisional Diffie-Hellman hypothesis. Let p be the quadratic residues modulo prime. Let n be the length of the private inputs of x and y. The most time-consuming computation is modular multiplications, so we will only count the cost of modular multiplications. Our solution takes no more than $4n \log \frac{p}{2}$ modular multiplications and the solution in [22] takes $5n \log p$ modular multiplications. The size of exchanged messages in our solution is no more than $3n \log p$ bits and it is $6n \log p$ bits in [22].

IV. Security Analysis

Goldreich [30] presents a security evaluation benchmark based on the simulation paradigm, which has been widely used to prove the secure of a multiparty computation solution.

A. The Semi-Honest Model

We suppose that both of the parties in our solution to Yao's millionaires' problem are semi-honest. A protocol is private in the semi-honest model if each party is unable to conclude the private input data of another party from the final and his/her collected intermediate computation results. And our solution is privacy preserving in a semi-honest setting.

B. Formulation of Privacy

Goldreich [30] proposes the privacy definition of secure multiparty computation to study the security of multiparty computation schemes. Let $f = (f_1, f_2)$ be a probabilistic polynomial-time functionality and Π be a two-party protocol for computing f. The $view$ of the first party during an execution of Π on the input (x, y), denoted by $\text{view}_1^{\Pi}(x, y)$, is $(x, r^1, m_1^1, ..., m_t^1)$, where r^1 represents the outcome of the first party's internal coin tosses, and m_i^1 represents the i-th message it has received. The output of the first party during an execution of Π on the input (x, y), denoted by $\text{output}_1^{\Pi}(x, y)$, is implicit in the party's view of the execution. The view and output of the second party can be defined analogously.

Definition 3: For a functionality f, Π privately computes f if there exist probabilistic polynomial-time algorithms, denoted by S_1 and S_2 such that

$$\begin{aligned}\{(S_1(x, f_1(x, y)), f_2(x, y))\}_{x,y} &\overset{c}{\equiv} \\ \{(\text{view}_1^{\Pi}(x, y), \text{output}_2^{\Pi}(x, y))\}_{x,y}&,\end{aligned} \tag{1}$$

and

$$\begin{aligned}\{(S_2(y, f_2(x, y)), f_1(x, y))\}_{x,y} &\overset{c}{\equiv} \\ \{(\text{view}_2^{\Pi}(x, y), \text{output}_1^{\Pi}(x, y))\}_{x,y}&,\end{aligned} \tag{2}$$

where $\overset{c}{\equiv}$ denotes computational indistinguishability, $\text{view}_1^{\Pi}(x, y)$ and $\text{view}_2^{\Pi}(x, y)$, $\text{output}_1^{\Pi}(x, y)$ and $\text{output}_2^{\Pi}(x, y)$, are related random variables, defined as a function of the same random execution.

C. Security Analysis on Our Solution

Notice that in this protocol $f_1(x, y) = f_2(x, y) = x > y$ or $f_1(x, y) = f_2(x, y) = x \leq y$, and the $view$ of a party is defined by $(x, r, m_1, m_2, ...)$, where x is the party's input, r is the private coin tosses, and m_i is the i-th message it received.

Suppose that $f_1(x, y) = f_2(x, y) = x > y$. We can construct simulator S_1 as follows:

S_1 receives $(x, f_1(x, y))$ as its input, and simulates $\text{view}_1^{\Pi}(x, y)$ is satisfied by eq. 1.

1. S_1 first generates a random secret key b', where b' is a large number and $b' < (p-1)/2$. And then S_1 randomly constructs a number y' such that $|S_x^1 \cap S_{y'}^0| = 1$.

2. S_1 prepares $l_2' = n - |S_{y'}^0|$ random numbers z_j.

3. According to protocol 1, S_1 computes S_x^1, $S_{y'}^0$, S and R'.

4. S_1 computes $\pi_1(S)$, and $\pi_2(R')$.

5. S_1 computes G', H' and $\pi_3(G')$.

Let $S_1(x, x > y) = \{x, a, b', S_x^1, S_{y'}^0, S, R', \pi_1(S), \pi_2(R'), G', \pi_3(G'), H', \ |H' \cap \pi_3(G')| = 1\}$. Since $\text{view}_1^{\Pi}(x, y) = \{x, a, S_x^1, S, \pi_1(S), \ \pi_2(R'), \pi_3(G'), H', \ |H' \cap \pi_3(G')| = 1\}$. So it shows that

$$\{(S_1(x, f_1(x, y)), f_2(x, y))\}_{x,y}$$
$$\overset{c}{\equiv} \{(\text{view}_1^{\Pi}(x, y), \text{output}_2^{\Pi}(x, y))\}_{x,y}.$$

Simulator S_2 for simulating $\text{view}_2^{\Pi}(x, y)$, such that

$$\{(S_2(x, f_2(x, y)), f_1(x, y))\}_{x,y}$$
$$\overset{c}{\equiv} \{(\text{view}_2^{\Pi}(x, y), \text{output}_1^{\Pi}(x, y))\}_{x,y}.$$

Similarly, if $f_1(x, y) = f_2(x, y) = x \leq y$, we can construct two simulators S_1 and S_2, such that

$$\{(S_1(x, f_1(x, y)), f_2(x, y))\}_{x,y}$$
$$\overset{c}{\equiv} \{(\text{view}_1^{\Pi}(x, y), \text{output}_2^{\Pi}(x, y))\}_{x,y},$$

and

$$\{(S_2(x, f_2(x, y)), f_1(x, y))\}_{x,y}$$
$$\overset{c}{\equiv} \{(\text{view}_2^{\Pi}(x, y), \text{output}_1^{\Pi}(x, y))\}_{x,y}.$$

V. Experimental Results and Analysis

A. Experiment Settings

In order to demonstrate the fact efficiency of our solution, we implemented our protocol using Python 2.7 based on Charm-Crypto [31] which depends on a few open-source C math libraries including OpenSSL, GMP (GNU Multiple Precision Arithmetic Library) and PBC (Pairing-based Cryptography Library). We builded the Charm-Crypto based on GMP 6.0.0 [32] not using the side-channel silent mpz_powm_sec function. And all of the experiments have been carried out on a machine running the Ubuntu subsystem in Windows 10 System with an Intel i5-4690 Processor at 3.50GHz and 8GB RAM. The asymmetric cryptographic key lengths have been chosen according to the current NIST standard from 1024 to 8192 bits.

B. Comparision with the Solution based on Symmetric Commutative Encryption Scheme

Because the efficiency of public key encryption schemes appears less than 0.1% of symmetric encryption schemes, no solution developed on the bases of the public key cryptography to Yao's millionaires' problem can be efficient [23]. Let

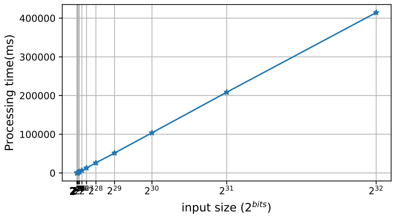

Fig. 1. Processing time of Shundong's Protocol vs. input size(2^{bits})

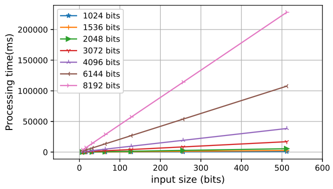

Fig. 2. Processing time of Protocol 1 vs. input size(bits)

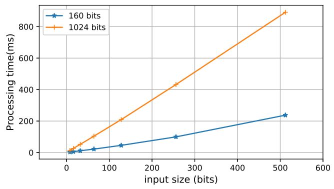

Fig. 3. Comparison of Processing time of Protocol 1 while the key size is 160 bits and 1024 bits

TABLE I
COMPARISON OF PROCESSING TIME (MS) BETWEEN OUR AND SHUNDONG'S PROTOCOL

Input size (bits)	Protocol 1 for 7 kinds of modulus size(bits)							XOR in Shundong's Protocol
	1024	1536	2048	3072	4096	6144	8192	
8	14	39	85	265	601	1680	3560	0.02
16	26	78	171	529	1199	3360	7106	5.419
32	52	154	341	1056	2403	6724	14,361	414,472.559
64	104	308	681	2112	4805	13,441	28,674	-
128	209	617	1363	4223	9614	26,871	571,77	-
256	432	1270	2734	8462	19,215	53,785	114,074	-
512	891	2520	5512	16,977	38,510	107,623	228,183	-

N indicate the maximal value of the input in Shundong's protocol and its bit size is $n = \lceil \log N \rceil$. Shundong's Protocol takes $4N$ XOR operation. When N is a small number, It is inevitable that Shundong's protocol is more efficient than our protocol. For the demonstration of the fact computation cost of the solution based on symmetric commutative encryption scheme and asymmetric commutative encryption scheme, we implemented our protocol 1 and XOR operation in Shundong's protocol in accordance with the previous experiment settings.

We test the performance of XOR in Shundong's Protocl and results have been presented in Fig. 1. The input size is chosen from 8 to 32 bits. As evident from the Fig. 1, it can be clearly observed that, with the linear increase in the size of an input, the processing time of Shundong's Protocol is increased linearly.

The Diffie-Hellman "group" is used for public cryptographic schemes. These groups are approximately as strong as a symmetric key. We choose 1024 (Group 2), 1536(Group 5), 2048(Group 14), 3072(Group 15), 4096(Group 16), 6144(Group 17) and 8192(Group 18) bits Diffie-Hellman groups to test the performance of our Protocol 1. The results of processing time are summarized in Table I.

As suggested by Table I and Fig. 2, we can observe that as the size of an input increases linearly the cost is almost increased linearly for the same modulus size.

The security level of our solution is determined by the modulus size, and it is more secure with the longer modulus size and the cost also is more expensive. Accordingly, we can determine the modulus size in accordance with security strength. From Table I, without considering the intersection operation, we can observe the fact that the computation cost of the XOR operation in Shundong's protocol is very small if the size of the private input is also small, otherwise the cost is large, for instance when the input size bit is 32. When the input size bit is 16, the cost for the computation of XOR operation is 5.419 ms. Thus, we can conclude the cost to be no less than 355,139.584 ($5.419 * 2^{16}$) ms and it is actually 414,472.559 ms. And we can continue to conclude the cost to be no less than 56,448 years when the input size bit is 64. So Shundong's protocol is too expensive to be practical when the input size is large. Thus, our solution to Yao's millionaires' problem is more efficient and practical than the solution based on symmetric commutative encryption scheme.

C. Comparision with the Solution based on DGK Encryption Scheme

It should be taken into notice that the parameter p is a large safe prime chosen in Protocol 1, i.e., both p and $q = (p-1)/2$ are large primes. In fact, the parameter p could be such a prime that $p-1$ has a sufficiently large prime factor q. "sufficiently large" means that the size of q is at least 160 bits, i.e., $q > 2^{160}$. For short-term security, the 2^{160} setting is imposed by the lower-bound requirement of the index computation attack algorithm called λ-method or kangaroo method for solving the discrete logarithm problem proposed by Pollard [33]. Thus, the random secret keys size setting for a and b in Protocol 1 can be 160 bits without having degenerated the underlying intractable discrete logarithm problem [34]. The performance of our solution to Yao's millionaires' problem can be further improved. The improved results are summarized in Fig. 3 while $a \approx 2^{160}$ and $b \approx 2^{160}$. It should be taken into notice that the processing time of Protocol 1 for $a \approx 2^{160}$ and $b \approx 2^{160}$ may have been reduced by 4 times while the parameter size of p is 1024 bits.

In terms of processing time, Damgard's solution [25] based on the dedicated DGK homomorphic encryption scheme is the state-of-the-art comparison protocol. We also compare the performance of the state-of-the-art comparison protocol based on DGK homomorphic encryption scheme with our solution based on symmetric commutative encryption scheme. For a 24-bit input and medium-term security, the results in Table II suggest that our solution outperforms Damgard's solution in terms of processing time. Owed to the fact that DGK homomorphic encryption scheme requires one complex modular exponentiation, it can be computed in advance during idle times. The processing time has also been presented in Table II under the assumption that h^r can be pre-computed during idle times. As evident from the Table II, it can be observed that our solution also outperforms Damgard's solution with precomputations in terms of processing time.

TABLE II
COMPARISON RESULTS OF PROCESSING TIME (MS) FOR A 24-BIT INPUT AND MEDIUM-TERM SECURITY

solutions	our solution	Damgard's solution	Damgard's solution with precomputations
processing time	24.31	132.52	39.99

VI. CONCLUSIONS

In this paper, we discuss some drawbacks in Shundong's symmetric cryptographic solution to millionaires' problem and introduce a new solution based on the set intersection problem. Our solution based on the Decisional Diffie-Hellman hypothesis is an asymmetric commutative encryption scheme. We also use two special encodings to reduce the computation cost of modular multiplications and the scale of the set intersection problem. To compare the fact computation cost of the solution based on symmetric commutative encryption scheme and asymmetric commutative encryption scheme, we implement XOR operation in Shundong's protocol and our protocol. It is found that Shundong's protocol is not more efficient than our protocol when the size of the input is large. Moreover, experimental results show that our solution is more efficient and practical than the state-of-the-art comparison solution.

REFERENCES

[1] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, ser. SFCS '82. Washington, DC, USA: IEEE Computer Society, 1982, pp. 160–164.

[2] S. Jha, L. Kruger, and P. McDaniel, "Privacy preserving clustering," in *Proceedings of the 10th European Conference on Research in Computer Security*, ser. ESORICS'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 397–417.

[3] D.-H. Shih, H.-Y. Huang, and D. C. Yen, "A secure reverse vickrey auction scheme with bid privacy," *Information Sciences*, vol. 176, no. 5, pp. 550–564, mar 2006.

[4] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter *et al.*, "Secure multiparty computation goes live," in *Financial Cryptography and Data Security*. Springer, 2009, pp. 325–343.

[5] B. Yang, A. Sun, and W. Zhang, "Secure two-party protocols on planar circles," *Journal of Information*, vol. 8, no. 1, pp. 29–40, 2011.

[6] B. Yang, Z. Shao, and W. Zhang, "Secure two-party protocols on planar convex hulls," *Journal of Information and Computational Science*, vol. 9, no. 4, pp. 915–929, 2012.

[7] L. Liu, C. Wu, and S. Li, "Two privacy-preserving protocols for point-curve relation," *Journal of Electronics (China)*, vol. 29, no. 5, pp. 422–430, 2012.

[8] B. YANG, C.-H. YANG, Y. YU, and D. XIE, "A secure scalar product protocol and its applications to computational geometry," *Journal of Computers*, vol. 8, no. 8, pp. 2018–2026, 2013.

[9] Q. Xie and U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users," in *Proceedings of the 2011 Ninth Annual International Conference on Privacy, Security and Trust (PST)*. IEEE, 2011, pp. 252–259.

[10] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient robust private set intersection," *Int. J. Appl. Cryptol.*, vol. 2, no. 4, pp. 289–303, Jul. 2012.

[11] W. Du, Y. S. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *Proceedings of the 4th SIAM International Conference on Data Mining*, vol. 233. Lake Buena Vista, Florida, 2004, pp. 222–233.

[12] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "Epcbir: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Information Sciences*, 2016.

[13] Z. A. Abduljabbar, H. Jin, A. Ibrahim, Z. A. Hussien, M. A. Hussain, S. H. Abbdal, and D. Zou, "Secure biometric image retrieval in iot-cloud," in *2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, 2016, pp. 1–6.

[14] L. Zhang, T. Jung, K. Liu, X. Y. Li, X. Ding, J. Gu, and Y. Liu, "Pic: Enable large-scale privacy preserving content-based image search on cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[15] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in iot-enabled smart metering systems," *Future Generation Computer Systems*, 2017.

[16] M. J. Atallah and W. Du, "Secure multi-party computational geometry," in *Proceedings of the 7th International Workshop on Algorithms and Data Structures*, ser. WADS '01. London, UK, UK: Springer-Verlag, 2001, pp. 165–179.

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

[18] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, Jan 2013.

[19] S. Garg, P. Mohassel, and C. Papamanthou, "Tworam: Efficient oblivious ram in two rounds with applications to searchable encryption," in *Proceedings, Part III, of the 36th Annual International Cryptology Conference on Advances in Cryptology — CRYPTO 2016 - Volume 9816*. New York, NY, USA: Springer-Verlag New York, Inc., 2016, pp. 563–592.

[20] P. Rindal and M. Rosulek, "Faster malicious 2-party secure computation with online/offline dual execution," in *Proceedings of the 25th USENIX Security Symposium*, 2016, pp. 297–314.

[21] I. Ioannidis and A. Grama, "An efficient protocol for yao's millionaires' problem," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*. IEEE, 2003, pp. 6–9.

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

[23] L. Shundong, W. Daoshun, D. Yiqi, and L. Ping, "Symmetric cryptographic solution to yao's millionaires' problem and an evaluation of secure multiparty computations," *Information Sciences*, vol. 178, no. 1, pp. 244–255, 2008.

[24] T. Veugen, F. Blom, S. J. A. de Hoogh, and Z. Erkin, "Secure comparison protocols in the semi-honest model," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1217–1228, Oct 2015.

[25] I. Damgard, M. Geisler, and M. Kroigard, "Homomorphic encryption and secure comparison," *International Journal of Applied Cryptography*, vol. 1, no. 1, pp. 22–31, Feb. 2008.

[26] R. Fagin, M. Naor, and P. Winkler, "Comparing information without leaking it," *Commun. ACM*, vol. 39, no. 5, pp. 77–85, May 1996.

[27] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003, pp. 86–97.

[28] L. Shundong, S. Tiange, and D. Yiqi, "Secure multi-party computation of set-inclusion and graph-inclusion," *Journal of Computer Research and development*, vol. 10, pp. 1647–1653, 2005.

[29] I. Blake and V. Kolesnikov, "Strong conditional oblivious transfer and computing on intervals," in *Advances in Cryptology - ASIACRYPT 2004*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3329, pp. 515–529.

[30] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge university press, 2009, vol. 2.

[31] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[32] GNU. The gnu multiple precision arithmetic library. [Online]. Available: https://gmplib.org/

[33] J. M. Pollard, "Monte Carlo methods for index computation mod p," *Mathematics of Computation*, vol. 32, pp. 918–924, 1978.

[34] M. Wenbo, "Modern cryptography: theory and practice," *Publisher: Prentice Hall PTR, Copyright: Hewlett Packard*, 2004.