An Efficient and Verifiable Solution to the Millionaire Problem

Kun Peng[1], Colin Boyd[1], Ed Dawson[1], and Byoungcheon Lee[1,2]

[1] Information Security Research Centre,
IT Faculty, Queensland University of Technology
{k.peng, c.boyd, e.dawson}@qut.edu.au
http://www.isrc.qut.edu.au
[2] Joongbu University, Korea
sultan@joongbu.ac.kr

Abstract. A new solution to the millionaire problem is designed on the base of two new techniques: zero test and batch equation. Zero test is a technique used to test whether one or more ciphertext contains a zero without revealing other information. Batch equation is a technique used to test equality of multiple integers. Combination of these two techniques produces the only known solution to the millionaire problem that is correct, private, publicly verifiable and efficient at the same time.

Keywords: millionaire problem, efficiency, verifiability, zero test, batch equation.

1 Introduction

In the millionaire problem, two millionaires want to compare their richness without revealing their wealth. This problem can be formulated as a comparison of two ciphertexts without decrypting them. Many solutions to the millionaire problem [3, 6, 8, 9, 10, 13, 16, 15, 24, 19, 20, 4, 5, 31, 12, 2] have been proposed. However, none of them are both verifiable and efficient.

In this paper, a new solution to the millionaire problem is proposed. This new solution is based on two new techniques: zero test and batch equation. The zero test is an interactive multiparty protocol which takes as input one or more ciphertexts and outputs 0 if at least one ciphertext is an encryption of 0, and outputs 1 otherwise. Batch equation allows equality between multiple pairs to be checked simultaneously, using randomised inputs. A circuit to solve the millionaire problem is reduced to a zero test with the help of homomorphism of the employed encryption algorithm and batch equation. Then the zero test is performed by some participants (e.g. the two millionaires themselves) without revealing their wealth. This scheme is the only known correct, private, publicly verifiable and efficient solution to the millionaire problem.

The structure of the rest of this paper is as follows. In Section 2, the millionaire problem is introduced and drawbacks of the currently existing solutions are

C. Park and S. Chee (Eds.): ICISC 2004, LNCS 3506, pp. 51–66, 2005.

pointed out. In Section 3, fundamental cryptographic primitives to be employed in this paper are recalled. In Section 4 and Section 5, an original cryptographic primitive — zero test — and a theorem about batch equation are proposed. In Section 6 and Section 7, a novel solution to the millionaire problem is presented and analysed. In Section 8, the paper is concluded.

2 The Millionaire Problem

In the millionaire problem, two millionaires want to compare who is richer without revealing their wealth. So they encrypt their wealth and the two ciphertexts should be compared. Some participants (often the two millionaires themselves) are employed to process the two ciphertexts and find out which contains a larger message. Any solution to the millionaire problem must be correct, private and verifiable according to the following standards.

- Correctness: If every participant is honest, the correct result is obtained.
- Privacy: After the computation, different entities' knowledge about the two messages is as follows:
  - a millionaire: his wealth, the result and what can be deduced from them;
  - others: at most the result and what can be deduced from it.
- Verifiability: Each participant can verify that the other participants are honest in their computation.

Fischlin [12] argued that the participants have no motivation to deviate from the protocol if they are input providers (millionaires). To support his claim, he gave an example, the "flighting ticket" problem and solved it as an application of the millionaire problem. However, verifiability is necessary to a solution to the millionaire problem in a general sense and needed in many of its applications like auctions. Even in the "flighting ticket" problem, motivation to deviate cannot be completely omitted and verifiability may still be needed.

Most current solutions to the millionaire problem are general-purpose and can deal with other applications than the millionaire problem, while some (like [12]) only deal with the millionaire problem. Two methods have been used to solve the millionaire problem. The first method is based on encrypted truth tables. Namely, a truth table of each logic gate in a circuit is encrypted and the rows in every table are shuffled, so that each gate can be evaluated with its inputs and output in ciphertext. The second method is based on logic homomorphism of encryption schemes. As special encryption algorithms are designed to be homomorphic in regard of the logical relation in the gates in the circuit, the evaluation can be realized by computing the ciphertexts of the inputs to the function without the help of any truth table.

The recent schemes employing the first method include [24], [20], [10], [19] and [5]. In [24], a circuit is generated by an authority AI and sent to another authority A, who uses it to process the ciphertext inputs. A hash function is employed in the truth tables to link their inputs to their outputs. Oblivious

transfer is employed to submit the inputs to the function confidentially. Correctness of the circuit is guaranteed by a cut-and-choose mechanism. Correctness of the computation is guaranteed by one-wayness of the hash function and an assumption that AI and A do not conspire. As the oblivious transfer primitive employed in [24] is not verifiable, AI can modify the inputs to the circuit without being detected. This problem was fixed by Juels and Szydlo [20]. They design a primitive called verifiable 1-out-of-2 oblivious transfer, which is slightly less efficient than the 1-out-of-2 oblivious transfer in [24], but prevents AI from cheating alone. Other drawbacks of [24] are (1) the cut-and-choose mechanism to guarantee circuit correctness is highly inefficient in communication as a few circuits must be transported from AI to A; (2) correctness of the auction relies on trust that the two authorities do not collude and is not publicly verifiable[1]; (3) the oblivious transfer used for bid submission is not efficient (both in computation and communication).

In [10], [19] and [5], correctness of circuit and evaluation can be publicly verified with help of public-key cryptology. So the costly cut-and-choose mechanism is removed and correctness is not based on any trust. However, public-key cryptology is much less efficient in computation than the hash function computation in [24] and [20]. As the number of gates in a circuit is not small and construction, evaluation and the corresponding validity verification in each gate requires hundreds of exponentiations, an extremely high computational cost.

The recent schemes employing the second method include [31], [2] and [12]. The schemes in [31] and [12] limit the computation to a two-input-provider one-participant situation and employ a technique called non-interactive cryptocomputing, where one input provider encrypts his input and the other input provider acts as the participant to perform the computation on the encrypted input. The scheme in [2] is a multiparty version of [31]. In [31], NOT and OR gates are used to construct the circuits while Goldwasser-Micali encryption or ElGamal encryption are extended to be NOT and OR homomorphic to calculate NOT and OR logic in ciphertext. Extension of logic homomorphism is implemented by expanding and combining ciphertexts. This expansion and combination mechanism brings two problems. The first problem is that the distribution of the expression of the final result is dependent on the circuit (namely the input of the participant), which violates privacy. The second problem is that the length of ciphertexts increases quickly as the computations go on, which brings a heavy burden on computation and communication. The efficiency pressure is so great, that depth of the circuit (thus number and length of the inputs) is strictly limited in [31]. The scheme in [2] also has these two drawbacks. Non-interactive

[1] Although it is said in [24] that "A naive verification procedure is to require the auctioneer to publish the tables and garbled input values of the circuit (signed by the AI), and allow suspecting bidders to simulate its computation", this verification procedure violates the basic rule in [24] that AI alone cannot know the bids. So another verification method based on "signed 'translation' table" in [24] has to be employed. Soundness of this verification method is based on an assumption that AI does not reveal the 'translation' table to A.

cryptocomputing brings a third problem to [31]: lack of verifiability. Although Sander *et al* suggested the usage of a fault tolerance mechanism, it requires to run the non-interactive cryptocomputing protocol many times, so is impractical in efficiency. To overcome the first two problems in [31], Fischlin [12] proposed a non-interactive cryptocomputing protocol, which sacrifices generality in [31] and only deals with the millionaire problem. In [12], NOT, XOR and AND gates are used to construct the circuits while Goldwasser-Micali encryption (which is NOT and XOR homomorphic) is extended to be AND homomorphic. Ciphertext expansion in [12] is not continual and does not bring the influence of the circuit to the distribution of the expression of the final result. Although the first two problems are overcome, [12] lacks verifiability too. As it employs non-interactive cryptocomputing, there is no practical verification mechanism.

So far, there is not any correct, private, verifiable and efficient solution to the millionaire problem. A correct, private, verifiable and efficient solution to the millionaire problem will be designed in this paper. The new technique employs the second method, but in a novel way.

3 Fundamental Primitives

Two fundamental primitives to be used in this paper are introduced in this section.

3.1 Mix Network

A mix network [18, 29, 30] mixes a number of encrypted inputs to the same number of outputs, while the link between the inputs and the outputs is kept secret. A mix network is usually composed of some mixing servers, each of which re-encrypts (or decrypts) and permutes the inputs in turn. The following two properties are usually required.

1. Correctness: the plaintexts of outputs must be a permutation of the plaintexts of the inputs.
2. Privacy: the permutation between the inputs and the outputs is unknown.

Correctness of a mix network must be publicly verifiable. There are two methods to verify correctness of a mix network.

1. Global verification: after all the servers have finished their mixing and the outputs are decrypted, a final verification is performed on the outputs in plaintext.
2. Individual verification: immediately after each server's mixing, he has to prove that his mixing is correct and the proof is verified instantly.

Usually, mix network with global verification is more efficient, but global verification requires that some parties must know the plaintexts in the inputs and can only be performed after the outputs are decrypted.

3.2 Modified ElGamal Encryption

ElGamal encryption is modified slightly as follows to be additive homomorphic.

- Integers p and q are large primes, such that $p = 2q+1$. Integer g is a generator of the cyclic subgroup of order q in Z_p^*.
- The private key is an integer x in Z_q and the public key is $(p,\ g,\ y = g^x \bmod p)$.
- Encryption: a message m in Z_q is encrypted into $c = (a, b) = (g^r \bmod p,\ g^m y^r \bmod p)$ where r is randomly chosen from Z_q.
- Decryption: given a ciphertext $c = (a, b)$, firstly $d = b/a^x \bmod p$ is calculated, then a search is performed to find $m = \log_g d$.

As in this paper decryption is only employed to test whether a ciphertext contains a zero or not, the search becomes a comparison of d and 1 ($m = 0$ iff $d = 1$).

In the rest of this paper,

- unless specified all the computations are performed modulo p;
- when c_1 and c_2 are two modified ElGamal ciphertexts and $c_1 = (a_1, b_1)$, $c_2 = (a_2, b_2)$
  - $c_1 c_2 = (a_1 a_2, b_1 b_2)$;
  - $c_1/c_2 = (a_1 a_2^{-1}, b_1 b_2^{-1})$;
  - $c_1^\gamma = (a_1^\gamma, b_1^\gamma)$.

4 A Building Block—Zero Test

Zero test is a new technique to test whether one or more ciphertexts contain a zero without decrypting them. The employed encryption algorithm $E()$ must be semantically secure[2] and additive homomorphic: $E(m_1)E(m_2) = E(m_1 + m_2)$. The corresponding decryption function is $D()$, which must be a distributed decryption function. The modified ElGamal encryption in Section 3.2 is employed in this paper[3]. In this modified ElGamal encryption, an exponentiation with the message as its exponent is encrypted using normal ElGamal encryption, so it is additive homomorphic. As ElGamal encryption is based on DL problem, distributed key generation [11, 27, 14] without any trusted party can be implemented efficiently. Although usually a search of logarithm is needed in the decryption function of the modified ElGamal encryption, the costly search is not

[2] Roughly, an encryption algorithm is said to be semantically secure if given m_0, m_1 and $c = E(m_k)$ where $k = 0$ or 1, the difference between the probability that k can be correctly guessed and 0.5 is negligible. See [23–Page 306] for more formal definition.

[3] Paillier encryption [25] with distributed decryption could be used, but distributed generation of an encryption system based on factorization problem is highly inefficient.

necessary in this paper, where any decryption is only performed to test whether the encrypted message is a known certain value, say zero. So application of the modified ElGamal encryption in this paper is efficient.

4.1 Simple Zero Test

We start with a simple case: to test a single ciphertext. Given a ciphertext c, it is required to test whether $D(c) = 0$ without revealing $D(c)$. The test is denoted as simple zero test $ZT(c)$, which outputs 0 if $D(c) = 0$, 1 if $D(c) \neq 0$. This technique is similar to a equality test technique in [22]. However, a multiparty test is used here while 2 two-party test is used in [22]. Suppose the private key is shared by some authorities $A_1, A_2, \ldots, A_m$ with a threshold secret sharing. The simple zero test is as follows.

1. Randomization
   Each A_l chooses a random integer r_l from $\{2, 3, \ldots, q-1\}$ and calculates $c_l = c_{l-1}^{r_l}$ to randomise the ciphertext where $c_0 = c$. It is publicly verifiable that $c_l \neq 1$ ($a_l \neq 1$ is verified where $c_l = (a_l, b_l)$) and $c_l \neq c_{l-1}$, so it is ensured that $r_l > 1$. Each A_l proves c_l is an exponentiation of c_{l-1} for $l = 1, 2, \ldots, m$ using a proof of equality of logarithms [7]: $\log_{a_{l-1}} a_l = \log_{b_{l-1}} b_l$.
2. Decryption
   The authorities cooperate to calculate $d = D(c_m)$ and prove the correctness of the distributed decryption using Chaum-Pedersen proof of equality of logarithms in [7] (see [27, 28] for details of distributed ElGamal decryption and its correctness verification). The output of the zero test is then as follows.

$$ZT(c) = \begin{cases} 0 & \text{if } d = 0 \\ 1 & \text{if } d \neq 0 \end{cases} \tag{1}$$

Theorem 1. *$ZT()$ is correct (if $D(c) = 0$, then $ZT(c) = 0$) and sound (if $D(c) \neq 0$, then $ZT(c) = 1$).*

Proof: As the correctness proof of randomization (proof of equality of logarithms [7]) is sound (if A_l does not know r_l such that $c_l = c_{l-1}^{r_l}$, he can pass the verification with only a negligible probability), $c_m = c^{\prod_{l=1}^{m} r_l}$ with an overwhelmingly large probability (in regard to the length of the challenge in the proof of knowledge of logarithm in [32] or in the proof of equality of logarithms in [7]) if the randomization is verified to be valid.

As the proof of correctness of decryption (Chaum-Pedersen proof of equality of logarithms in [7]) is sound (incorrect decryption can pass the decryption verification with only a negligible probability), $ZT(c) = D(c_m)$ with an overwhelmingly large probability (in regard to the length of the challenge in the Chaum-Pedersen proof of equality of logarithms in [7]) if the decryption is verified to be valid. So $ZT(c) = D(c^{\prod_{l=1}^{m} r_l})$ with an overwhelmingly large probability.

As $E()$ is additive homomorphic, $D(c^{\prod_{l=1}^{m} r_l}) = D(c) \prod_{l=1}^{m} r_l$. So $ZT(c) = D(c) \prod_{l=1}^{m} r_l$ if the whole verification succeeds. So, if $D(c) = 0$, $ZT(c) = 0$,

therefore $ZT()$ is correct. As it is publicly verifiable that $c_m \neq 1$, it is guaranteed $\prod_{l=1}^{m} r_l \neq 0 \bmod q$. So if $D(c) \neq 0$, then $ZT(c) \neq 0$. □

Theorem 2. *$ZT()$ is private. (No information about $D(c)$ is revealed except whether it is zero if at least one participant conceals his mixing and the number of dishonest authorities is not over the sharing threshold.)*

Proof: The correctness proof of randomization (proof of equality of logarithms in [7]) is special honest-verifier zero knowledge, so r_l is kept secret in the proof. Moreover, if the number of dishonest authorities is not over the threshold, none of $c_0, c_1, \ldots c_{m-1}$ can be decrypted. So the only revealed information about $D(c)$ is $D(c)\prod_{l=1}^{m} r_l$, from which the cooperation of all the authorities is necessary to deduce $D(c)$. Moreover, $D(c)\prod_{l=1}^{m} r_l$ is uniformly distributed in the message space of the encryption algorithm. So, if the number of dishonest authorities is not over the sharing threshold (thus no ciphertext but c_m can be decrypted and at least one authority A_l chooses r_l randomly), $D(c)\prod_{l=1}^{m} r_l$ is uniformly distributed and independent of $D(c)$. Therefore, if the number of dishonest authorities is not over the sharing threshold, no information about $D(c)$ is revealed except whether it is zero. □

Simple zero test is publicly verifiable as both randomization and distributed decryption are publicly verifiable.

4.2 Complex Zero Test

As $ZT()$ can only test whether a single ciphertext is an encryption of zero, it cannot work when there are more than one ciphertext to test (to be zero or not). In a complex zero test, it is required to test whether there is at least one encryption of zero in multiple ciphertexts without revealing any other information about the messages encrypted in the ciphertexts. Suppose ciphertexts c_i for $i = 1, 2, \ldots, n$ are the encrypted inputs to test. The complex zero test is denoted as $ZM(c_1, c_2, \ldots, c_n)$, which returns 0 iff there is at least one encryption of zero in c_i for $i = 1, 2, \ldots, n$. The test $ZM(c_1, c_2, \ldots, c_n)$ is implemented as follows where the decryption key is shared by authorities A_1, A_2, ..., A_m.

1. Mix network
   The authorities act as mixing servers and set up a mix network to mix c_i for $i = 1, 2, \ldots, n$ to ciphertexts c'_i for $i = 1, 2, \ldots, n$. The mix network must be correct and private. As it is not desired to decrypt the outputs c'_i for $i = 1, 2, \ldots, n$ in this paper, correctness of the mixing must be publicly verifiable without decrypting them. So mix networks with global verification [26, 17, 29] cannot be employed although they are very efficient. Among the mix networks employing individual verification, [18] and [30] are good choices here. Both of them are efficient and their correctness and privacy are strong enough for many applications including zero test. [30] is more efficient than [18], but achieves weaker privacy.[4]

[4] Shuffling in groups and batch verification of validity of shuffling are employed in [30]. The grouping operation leads to high efficiency, but weakens privacy a little.

2. Simple zero tests
   The authorities then cooperate to perform $ZT(c'_i)$ for $i = 1, 2, \ldots, n$ one by one until a zero is found in one simple zero test or all the n simple tests finish. The output of the zero test is then as follows.

$$ZM(c_1, c_2, \ldots, c_n) = \begin{cases} 0 & \text{if a zero is found in one simple zero test} \\ 1 & \text{if no zero is found after all the simple tests finish} \end{cases} \quad (2)$$

Theorem 3. *$ZM()$ is correct and sound ($ZM(c_1, c_2, \ldots, c_n) = 0$ iff there is at least one encryption of zero in c_i for $i = 1, 2, \ldots, n$).*

Proof: Equation (2) indicates that $ZM(c_1, c_2, \ldots, c_n) = 0$ iff $ZT(c'_i) = 0$ for some i in $\{1, 2, \ldots, n\}$. As each $ZT()$ is correct and sound, $ZT(c'_i) = 0$ iff c'_i encrypts a zero. So $ZM(c_1, c_2, \ldots, c_n) = 0$ iff c'_i encrypts a zero for some i in $\{1, 2, \ldots, n\}$.

As the employed mix network ([18] or [30]) is correct, $\{D(c'_1), D(c'_2), \ldots, D(c'_n)\} = \{D(c_1), D(c_2), \ldots, D(c_n)\}$. So $ZM(c_1, c_2, \ldots, c_n)$ returns zero iff $D(c_i) = 0$ for some i in $\{1, 2, \ldots, n\}$. □

Theorem 4. *$ZM()$ is private. (No information about $D(c_i)$ for $i = 1, 2, \ldots, n$ is revealed except whether there is at least one zero among them if at least one authority conceals his mixing and the number of dishonest authorities is not over the sharing threshold.)*

Proof: As $ZT()$ is private, no information about $D(c'_i)$ for $i = 1, 2, \ldots, n$ is revealed in $ZM()$ except whether at least one of them is zero and the index of the first zero among them (if there is at least one zero). As the employed mix ([18] or [30]) network is private, no link is known between $c_1, c_2, \ldots, c_n$ and $c'_1, c'_2, \ldots, c'_n$ if at least one authority conceals his mixing and the number of dishonest authorities is not over the sharing threshold. So no information about $D(c'_i)$ for $i = 1, 2, \ldots, n$ is revealed in $ZM()$ except whether at least one of them is zero. □

Complex zero test is publicly verifiable as both the mix network and simple zero test are publicly verifiable.

5 Batch Equation

To apply zero test to the millionaire problem, the following theorem about batch equation is necessary. Batch equation is a technique to test equality of each pair of integers in multiple pairs. The idea is similar to the so called "batch verification" [1]. However, unlike batch verification, no zero-knowledge proof or verification is involved in batch equation.

Theorem 5. *When there exists $y_i \neq z_i \bmod q$ with any i in $\{1, 2, \ldots, n\}$, $\sum_{i=1}^{n} y_i t_i = \sum_{i=1}^{n} z_i t_i \bmod q$ with a probability no more than 2^{-T} if q is a prime, t_i is randomly chosen from $\{0, 1, 2, \ldots, 2^T - 1\}$ for $i = 1, 2, \ldots, n$ and $q \geq 2^T$.*

To prove Theorem 5, a lemma is proved first.

Lemma 1. *Suppose* $t_1, t_2, \ldots, t_{v-1}, t_{v+1}, t_{v+2}, \ldots, t_n$ *are constant. If* q *is a prime,* $y_v \neq z_v \bmod q$, $q \geq 2^T$ *and* $\sum_{i=1}^{n} y_i t_i = \sum_{i=1}^{n} z_i t_i \bmod q$, *then there is only one possible* T*-bit solution for* t_v.

Proof: If this lemma is not correct, then the following two equations can be satisfied simultaneously where $y_v \neq z_v \bmod q$, $|t_v| = |\hat{t}_v| = T$ and $t_v \neq \hat{t}_v$.

$$\sum_{i=1}^{n} y_i t_i = \sum_{i=1}^{n} z_i t_i \bmod q \tag{3}$$

$$(\sum_{i=1}^{v-1} y_i t_i) + y_v \hat{t}_v + \sum_{i=v+1}^{n} y_i t_i = (\sum_{i=1}^{v-1} z_i t_i) + z_v \hat{t}_v + \sum_{i=v+1}^{n} z_i t_i \bmod q \tag{4}$$

Subtracting (4) from (3) yields

$$y_v(t_v - \hat{t}_v) = z_v(t_v - \hat{t}_v) \bmod q$$

So

$$(y_v - z_v)(t_v - \hat{t}_v) = 0 \bmod q$$

Note that $t_v - \hat{t}_v \neq 0 \bmod q$ because $q \geq 2^T$ and $|t_v| = |\hat{t}_v| = T$.
As q is a prime, $y_v - z_v = 0 \bmod q$. A contradiction to the statement $y_v \neq z_v \bmod q$ is found. Therefore, the lemma is correct. □

Proof of Theorem 5: Lemma 1 implies that among the $(2^T)^n$ possible combinations of $\{t_1, t_2, \ldots, t_n\}$ in $\{0, 1, 2, \ldots, 2^T - 1\}^n$, at most $(2^T)^{n-1}$ of them can satisfy $\sum_{i=1}^{n} y_i t_i = \sum_{i=1}^{n} z_i t_i \bmod q$ when $y_v \neq z_v \bmod q$. So if $y_v \neq z_v \bmod q$ and t_i are randomly chosen from $\{0, 1, 2, \ldots, 2^T - 1\}$ for $i = 1, 2, \ldots, n$, then $\sum_{i=1}^{n} y_i t_i = \sum_{i=1}^{n} z_i t_i \bmod q$ is satisfied with probability no more than 2^{-T}. □

6 Solution to the Millionaire Problem

The millionaire problem is solved in a circuit to compare two ciphertexts to determine which one contains a larger message without decrypting them. The circuit is implemented through three levels of computation as shown in Statement (5), which is true iff $D(c_1) > D(c_2)$.

$$\begin{aligned}(D(c_{1,1}) = 1 \wedge D(c_{2,1}) = 0) \vee (D(c_{1,1}) = D(c_{2,1}) \wedge D(c_{1,2}) = 1 \wedge D(c_{2,2}) = 0) \\ \vee \ \ldots \ \vee \ (D(c_{1,1}) = D(c_{2,1}) \wedge D(c_{1,2}) = D(c_{2,2}) \\ \wedge \ldots \wedge D(c_{1,L-1}) = D(c_{2,L-1}) \wedge D(c_{1,L}) = 1 \wedge D(c_{2,L}) = 0)\end{aligned} \tag{5}$$

At the innermost level, there are tests of bit equality and tests of bit difference, which can be implemented with the help of homomorphism of the employed encryption algorithm. At the middle level, there are computations of "AND"

logic, which can be implemented with the help of batch equation and homomorphism of the employed encryption algorithm. At the outermost level, there are computations of "OR" logic, which can be implemented using zero test.

Suppose the two messages are encrypted bit by bit as $c_1 = (c_{1,1}, c_{1,2}, \ldots, c_{1,L})$ and $c_2 = (c_{2,1}, c_{2,2}, \ldots, c_{2,L})$ where the most significant bit is on the left. The solution is as follows.

1. The participants (e.g. the two millionaires) corporately and randomly choose t_i from $\{0, 1, 2, \ldots, 2^T - 1\}$ for $i = 1, 2, \ldots, n$. For example, one millionaire randomly chooses $t_{1,i}$ for $i = 1, 2, \ldots, n$ and publishes $H(t_{1,i})$ for $i = 1, 2, \ldots, n$ while the other randomly chooses $t_{2,i}$ for $i = 1, 2, \ldots, n$ and publishes $H(t_{2,i})$ for $i = 1, 2, \ldots, n$ where $H()$ is a one-way and collision-resistant hash function. Then they publish $t_{i,1}$, $t_{i,2}$ for $i = 1, 2, \ldots, n$ and calculate $t_i = t_{i,1} + t_{i,2} \bmod 2^T$ for $i = 1, 2, \ldots, n$.
2. The participants act as the authorities in $ZM()$ and perform

$$ZM\ (\ c_{1,1}/(E(1)c_{2,1})\ ,\ (c_{1,1}/c_{2,1})^{t_1}(c_{1,2}/(E(1)c_{2,2}))^{t_2}, \ldots \\ (\prod_{i=1}^{L-1}(c_{1,i}/c_{2,i})^{t_i})\quad (c_{1,L}/(E(1)c_{2,L}))^{t_L}\ ) \tag{6}$$

where the modified ElGamal encryption in Section 3.2 is employed and $E(1) = (1, g)$. Then $D(c_1)$ is declared to be larger than $D(c_2)$ iff Statement (6)=0.

Theorem 6. *The solution to the millionaire problem through Statement (6) is a correct and sound with an overwhelmingly large probability ($D(c_1) > D(c_2)$ iff Statement (6)=0 with an overwhelmingly large probability).*

Proof: $D(c_1) > D(c_2)$ iff Statement (5) is true. According to additive homomorphism of the modified ElGamal encryption, Statement (5) is equivalent to

$$\begin{aligned} &D(c_{1,1}/(E(1)c_{2,1})) = 0\ \vee\ (D(c_{1,1}/c_{2,1}) = 0 \wedge D(c_{1,2}/(E(1)c_{2,2})) = 0)\ \vee \\ &\ldots\ \vee\ (D(c_{1,1}/c_{2,1}) = 0 \wedge D(c_{1,2}/c_{2,2}) = 0 \wedge \ldots \\ &\wedge D(c_{1,L-1}/c_{2,L-1}) = 0 \wedge D(c_{1,L}/(E(1)c_{2,L})) = 0) \end{aligned} \tag{7}$$

According to Theorem 5, with an overwhelmingly large probability Statement (7) is equivalent to

$$\begin{aligned} &D(c_{1,1}/(E(1)c_{2,1})) = 0\ \vee\ t_1 D(c_{1,1}/c_{2,1}) + t_2 D(c_{1,2}/(E(1)c_{2,2})) = 0\ \vee \\ &\ldots\ \vee\ t_1 D(c_{1,1}/c_{2,1}) + t_2 D(c_{1,2}/c_{2,2}) + \ldots \\ &+ t_{L-1} D(c_{1,L-1}/c_{2,L-1}) + t_L D(c_{1,L}/(E(1)c_{2,L})) = 0 \end{aligned} \tag{8}$$

According to additive homomorphism of the modified ElGamal encryption, Statement (8) is equivalent to

$$\begin{aligned} &(D(c_{1,1}/(E(1)c_{2,1})) = 0\ \vee\ D((c_{1,1}/c_{2,1})^{t_1}(c_{1,2}/(E(1)c_{2,2}))^{t_2}) = 0 \\ &\vee\ \ldots\ \vee\ D((\prod_{i=1}^{L-1}(c_{1,i}/c_{2,i})^{t_i})(c_{1,L}/(E(1)c_{2,L}))^{t_L}) = 0 \end{aligned} \tag{9}$$

So $D(c_1) > D(c_2)$ iff one of the L clauses in Statement (9) is true. As $ZM()$ is correct and sound method to test whether there is any zero encrypted in some ciphertexts, Statement (9) can be evaluated through Statement (6). Therefore, $D(c_1) > D(c_2)$ iff Statement (6)=0 with an overwhelmingly large probability. □

Theorem 7. *The solution to the millionaire problem through Statement (6) is private.*

Proof: The computations in Statement (6) before the zero test are in ciphertext and involves no decryption, so are private. The computation in $ZM()$ is private as proved in Theorem 4. So the computations in Statement (6) are private. □

7 Analysis

Suppose the two millionaires act as the participants, the cost of the solution of the millionaire problem includes:

- $(L + 2)(L - 1)$ short exponentiations (T-bit exponent)
- $2L$ divisions;
- about $16L$ full-length exponentiations for the mix network in [18] or $2(2L + k(4k - 2))$ full-length exponentiations for the mix network in [30] where k is a small parameter unrelated to L;
- average of $L/2$ simple zero tests: $2L$ full-length exponentiations, L proofs of equality of logarithms (costing $2L$ full-length exponentiations) and $L/2$ distributed decryptions and validity proof of decryptions (costing $3L$ full-length exponentiations).

A comparison between the new solution to the millionaire problem and solutions based on the existing solutions is provided in Table 1. The schemes in [10] and [2] are similar to [19] and [31] respectively, so are not analysed separately. In the schemes in Table 1, only [12] and our proposed scheme provided a concrete circuit to solve the millionaire problem. In [12], $L + L(L - 1)$ two-input gates are needed, while in our scheme, $L + L(L - 1)/2$ two-input gates are needed. In the other schemes, at least $7L$ two-input gates are needed as analysed in [21]. In [31], the number of gates should be larger than $7L$ as only "NOT" and "OR" gates are used. Even if only $7L$ two-input gates are employed in [31], ciphertext expansion bring an intolerable cost for encryption and communication. As each gate has only two inputs, those $7L$ gates must be in $\log_2 7L$ levels. So in [31] at least $8^{\log_2 7L} = 343L^3$ ciphertexts must be transmitted. It is pointed out in [12] that at least L^4 multiplications are needed in [31]. In [12], $L(L + 1)/2$ "AND" gates are needed, so $\lambda L(L + 3)/2$ encryptions (each costly 1.5 multiplication in average) and $\lambda L(L + 1)/2$ multiplications are needed for "AND" gates. There are $L(L + 1)/2$ multiplications for "XOR" and "NOT" computation in [12]. In this analysis, the number of multiplications are accounted in computation and transportation of integers with significant length (several hundred bits or longer)

is accounted in regard of communication where K is the bit length of full-length exponent (e.g 1024 bits). One full-length exponentiations is regarded as $1.5K$ multiplications and computation of the product of n short exponentiations is regarded as $n + 0.5nT$ multiplications.

In the example in Table 1, $K = 1024$ and $L = 100$. Let t, the number of cuttings in [20] and λ, the parameter in [12] be 40. The parameter k in the mix network [30] is set to be 5, which is big enough to provide strong privacy.

The analysis in Table 1 indicates that both [12] and the proposed solution can efficiently solve the millionaire problem with short inputs. Compared to the proposed schemes, the scheme in [12] has a drawback: lack of verifiability. Is it possible to overcome the drawback and make the scheme in [12] verifiable? A naive method is to employ the shuffling-then-decryption technique from the proposed scheme to [12]. However, this method is infeasible. Firstly, non-interactive cryptocomputing implies that validity of the circuit is not verifiable. As the circuit is dependent on the participant's input, revealing the circuit violates privacy of the the participant's input. Even if non-interactive cryptocomputing is replaced by two-party computation in [12] and the circuit is independent on any input, verifiability still cannot be practically achieved in [12]. Distributed key generation for Goldwasser-Micali encryption (distributed generation of a secret factorization) is much more costly than distributed key generation for ElGamal encryption and distributing the Goldwasser-Micali private key between the two participants without any trusted party is highly inefficient. Moreover, before the encrypted result is output for decryption, the $L\lambda$ ciphertexts in it must be shuffled using a re-encryption mix, otherwise privacy is violated. Although it may be possible to design a verifiable re-encryption mix for Goldwasser-Micali ciphertexts[5], the large-scale shuffling ($L\lambda = 4000$ when $L = 100$ and $\lambda = 40$) and proof-verification of validity of the shuffling is too impractical. On the other hand, the proposed solution to the millionaire problem can be modified to non-interactive cryptocomputing to improve its efficiency. After the modification, distributed decryption and the costly proof operations can be omitted, so that the proposed scheme becomes much more efficient by sacrificing verifiability. Without verifiability, the proposed scheme becomes more efficient than the scheme in [12]. In summary, the new solution achieves the best trade-off between security and efficiency and provides the only general, correct, private, verifiable and efficient solution for the millionaire problem.

8 Conclusion

A new solution to the millionaire problem is designed to achieve correctness, privacy, verifiability and high efficiency, which have never been achieved simultaneously before. In the future, the possibility of using the techniques in this paper to compute other functions will be investigated.

[5] There is no such shuffling or mix at present.

Table 1. Comparison of solutions of the millionaire problem

	Correctness	Privacy	Verifiability	Computation		Communication	
				cost	example	cost	example
[24]	Yes	Yes	No	$\geq 10KLt$	≥ 40960000	$\geq 37Lt + 2t$	≥ 148080
[20]	Yes	Yes	Yes	$\geq 10KLt$	≥ 40960000	$\geq 37Lt + 2t$	≥ 148080
[19]	Yes	Yes	Yes	average $3110KL + 6$	318464006	average $1626L + 6$	162606
[5]	Yes	Yes	Yes	average $\geq 2693KL$	≥ 275763200	$\geq 1543L$	≥ 154300
[31]	Yes	No	No	L^4	> 100000000	$\geq 343L^3$	≥ 343000000
[12]	Yes	Yes	No	$(1.5\lambda L(L+3)) + (\lambda+1)L(L+1))/2$	516050	$L(\lambda+2)$	4200
Proposed scheme using [18] mix	Yes	Yes	Yes	average $(L+2)(L-1)(1+0.5T) + 25KL$	2772058	average $25L$	2500
Proposed scheme using [30] mix	Yes	Yes	Yes	average $(L+2)(L-1)(1+0.5T)$ $+K(2L + 2(5.5L + 4k(k-2)))$	1666138	average $2L + 2(5.5L + 2k^2)$	1400

Acnowledgement

This paper is sponsored by DP 0345458 ARC DISCOVERY/ 2003-2005 and LX 0346868.

References

1. Riza Aditya, Kun Peng, Colin Boyd, and Ed Dawson. Batch verification for equality of discrete logarithms and threshold decryptions. In *Second conference of Applied Cryptography and Network Security, ACNS 04*, volume 3089 of *Lecture Notes in Computer Science*, pages 494–508, Berlin, 2004. Springer-Verlag.
2. D. Beaver. Minimal-latency secure function evaluation. In *EUROCRYPT '00, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 335–350. Springer, 2000.
3. Michael Ben-Or, Shafi Goldwasser, Joe Killian, , and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC1988*, pages 113–131, 1988.
4. Christian Cachin. Efficient private bidding and auctions with an oblivious third party. In *the 6th ACM Conference on Computer and Communications Security*, 1999. Available at http://www.tml.hut.fi/∼helger/crypto/link/protocols/auctions.html.
5. Christian Cachin and Jan Camenisch. Optimistic fair secure computation (extended abstract). In *CRYPTO '00*, pages 94–112, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science 1880.
6. D. Chaum, I. B. Damgård, and J. van de Graaf. Multiparty computations ensuring privacy of each party's input and correctness of the result. In *CRYPTO '87*, pages 87–119, Berlin, 1987. Springer-Verlag. Lecture Notes in Computer Science Volume 293.
7. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO '92*, pages 89–105, Berlin, 1992. Springer-Verlag. Lecture Notes in Computer Science Volume 740.
8. David Chaum, Claude Crepeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC1988*, pages 11–19, 1988.
9. Ronald Cramer, Ivan Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. Efficient multiparty computations secure against an adaptive adversary. In *EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 1999.
10. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT '01, Innsbruck, Austria, May 6-10, 2001, Proceeding*, volume 2045 of *Lecture Notes in Computer Science*, pages 280–299. Springer, 2001.
11. P Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science*, pages 427–437, 1987.
12. Marc Fischlin. A cost-effective pay-per-multiplication comparison method for millionaires. In *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings*, volume 2020 of *Lecture Notes in Computer Science*, pages 457–472. Springer, 2001.

13. Matthew K. Franklin and Stuart Haber. Joint encryption and message-efficient secure computation. *Journal of Cryptology 9(4)*, pages 217–232, 1996.
14. R Gennaro, S Jarecki, H Krawczyk, and T Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *EUROCRYPT '99*, pages 123–139, Berlin, 1999. Springer-Verlag. Lecture Notes in Computer Science Volume 1592.
15. Rosario Gennaro, Michael O. Rabin, and Tal Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing, PODC'98*, pages 101 – 111, 1987.
16. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC 1987*, pages 218–229, 1987.
17. Philippe Golle, Sheng Zhong, Dan Boneh, Markus Jakobsson, and Ari Juels. Optimistic mixing for exit-polls. In *ASIACRYPT '02*, pages 451–465, Berlin, 2002. Springer-Verlag. Lecture Notes in Computer Science Volume 1592.
18. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. In *Public Key Cryptography 2003*, pages 145–160, Berlin, 2003. Springer-Verlag. Lecture Notes in Computer Science Volume 2567.
19. M Jakobsson and A Juels. Mix and match: Secure function evaluation via ciphertexts. In *ASIACRYPT '00*, pages 143–161, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1976.
20. A. Juels and M. Szydlo. An two-server auction protocol. In *Proc. of Financial Cryptography*, pages 329–340, 2002.
21. Kaoru Kurosawa and Wakaha Ogata. Bit-slice auction circuit. In *7th European Symposium on Research in Computer Security, ESORICS2002*, volume 2502 of *Lecture Notes in Computer Science Volume 2339*, pages 24 – 38, Berlin, 2002. Springer-Verlag.
22. H. Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In *ASIACRYPT '03*, volume 2894 of *Lecture Notes in Computer Science*, pages 416–433, Berlin, 2003. Springer.
23. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC press, 1996.
24. Moni Naor, Benny Pinkas, and Reuben Sumner. Privacy perserving auctions and mechanism design. In *ACM Conference on Electronic Commerce 1999*, pages 129–139, 1999.
25. P Paillier. Public key cryptosystem based on composite degree residuosity classes. In *EUROCRYPT '99*, pages 223–238, Berlin, 1999. Springer-Verlag. Lecture Notes in Computer Science Volume 1592.
26. C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *EUROCRYPT '93*, pages 248–259, Berlin, 1993. Springer-Verlag. Lecture Notes in Computer Science Volume 765.
27. Torben P. Pedersen. A threshold cryptosystem without a trusted party. In *EUROCRYPT '91*, pages 522–526, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science Volume 547.
28. Torben P. Pedersen. *Distributed Provers and Verifiable Secret Sharing Based on the Discrete Logarithm Problem*. PhD thesis, Computer Science Department, Aarhus University,Aarhus, Denmark, 1992.

29. Kun Peng, Colin Boyd, Edward Dawson, and Kapali Viswanathan. Efficient implementation of relative bid privacy in sealed-bid auction. In *The 4th International Workshop on Information Security Applications, WISA2003*, volume 2908 of *Lecture Notes in Computer Science*, pages 244–256, Berlin, 2003. Springer-Verlag.
30. Kun Peng, Colin Boyd, Edward Dawson, and Kapali Viswanathan. A correct, private and efficient mix network. In *2004 International Workshop on Practice and Theory in Public Key Cryptography*, pages 439–454, Berlin, 2004. Springer-Verlag.
31. Tomas Sander, Adam Young, and Moti Yung. Non-interactive cryptocomputing for NC^1. In *40th Annual Symposium on Foundations of Computer Science, New York, NY, USA, FOCS '99*, pages 554–567, 1999.
32. C Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology, 4, 1991*, pages 161–174, 1991.