Ciphertext Comparison, a New Solution to the Millionaire Problem

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

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

Abstract. A new cryptographic protocol —ciphertext comparison— can compare two ciphertexts without revealing the two encrypted messages. Correctness of the comparison can be publicly verified. This technique provides an efficient and publicly verifiable solution to the famous millionaire problem. It is the first solution to the millionaire problem to output a precise result (the two messages are equal or which is larger). Privacy in this new solution is achieved with an overwhelmingly large probability and strong enough in practice.

Keywords: Ciphertext comparison, the millionaire problem, efficiency.

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. The millionaire problem is an intensively studied problem in multiparty computation. Since this problem was raised by Yao [17], many multiparty computation schemes [13, 12, 6, 11, 3, 16, 2, 8, 15] have been proposed, each of which can be applied to the millionaire problem. However, none of the currently known multiparty computation schemes provides an efficient and verifiable solution to the millionaire problem. Moreover, all the existing solutions to the millionaire problem only output one bit. So they output an imprecise result (whether a message is larger than the other or no larger than the other), while a precise result should indicate a message is larger than the other or equal to the other or smaller than the other. In addition, many of the existing schemes have various other problems like lack of verifiability.

A new protocol proposed in this paper, ciphertext comparison, can efficiently implement comparison of two encrypted messages without revealing them. A distributed homomorphic encryption algorithm is employed to encrypt the two messages. The ciphertext comparison technique outputs $a(m_1 - m_2)$ where a is a random and secret integer. Parameter choice guarantees that $a(m_1 - m_2)$ indicates the comparison result, but does not reveal any information about the two

S. Qing et al. (Eds.): ICICS 2005, LNCS 3783, pp. 84–96, 2005.

messages with an overwhelmingly large probability except which one is larger. The whole protocol is publicly verifiable. Correctness, privacy, robustness, public verifiability and high efficiency are achieved simultaneously for the first time in solving the millionaire problem. Moreover, a precise result is output in this new solution.

The rest of this paper is organised as follows. In Section 2, the millionaire problem and its previous solutions are recalled. In Section 3, new primitives needed in this paper are proposed and proved to be secure. In Section 4, the new ciphertext comparison protocol is described. In Section 5, the new ciphertext comparison protocol is analysed and compared against previous solutions to the millionaire problem. The paper is concluded in Section 6.

In the rest of this paper, the following symbols are used.

- $||$ stands for concatenation.
- $\lfloor x \rfloor$ is the largest integer no more than x.
- $PKN(x_1, x_2, \ldots, x_n | cond)$ stands for the proof of knowledge of a set of integers $x_1, x_2, \ldots, x_n$ satisfying a given condition *cond*.

2 The Millionaire Problem and Related Work

The millionaire problem was raised by Yao [17]. In the millionaire problem, two millionaires want to compare who is richer without revealing their wealth. This problem can be formulated as a comparison of two encrypted messages without revealing them. Some participants (sometimes the two millionaires themselves) are employed to solve the problem without revealing the two messages. The following four properties are often desired in a solution protocol to the millionaire problem.

- Correctness: If the two ciphertexts are decrypted and then compared, the result is the same as the protocol outputs.
- Precision: A precise result must be output to indicate exactly which of the three possibilities (whether a message is smaller than or equal to or larger than the other) occurs.
- Public verifiability: Each participant can be publicly verified to honestly follow the protocol.
- Privacy: After the computation, no information about the two messages is revealed except the comparison result.

Solutions to the millionaire problem always employ multiparty computation. In multiparty computation, multiple participants compute a function with encrypted inputs and determine the result of the function without revealing the inputs. They usually employ an evaluation circuit consisting of some logic gates to compute the function in ciphertext. Usually the decryption key of the employed encryption algorithm is shared among the participants, so that privacy of the encrypted inputs can be protected with an assumption that the number of malicious participants is not over a threshold. In all the known existing multiparty computation solutions, only an imprecise result is output (a precise result should indicate one of the three possible results).

According to the computation in every gate in the circuit, they can be divided into two methods. The first method is based on encrypted truth tables. Namely, the rows in the truth table of each logic gate in the circuit are encrypted and shuffled, so that any legal encrypted input to each gate can be matched to an encrypted output without being revealed. The second method is based on logic homomorphism of certain encryption schemes. As special encryption algorithms homomorphic in regard to the logic gates in the circuit are employed, the evaluation can be implemented by computing in ciphertext without the help of any truth table. The recent schemes employing the first method include [13], [12], [6], [11] and [3]. The recent schemes employing the second method include [16], [2], [8] and [15]. None of them provides a correct, precise, private, verifiable and efficient solution to the millionaire problem. Such a solution will be designed in this paper. The new technique is called ciphertext comparison. It employs the second method, but in a novel manner.

3 Preliminary Work

Three cryptographic primitives are presented in this section and will be applied to the new ciphertext comparison protocol. All multiplications in this section are with a modulus N^2 where N is the Paillier composite [14].

3.1 Proof of Knowledge of N^{th} Root mod $Z^*_{N^2}$

Proof of knowledge of root was proposed by Guillou and Quisquater [10], in which an honest verifier ZK proof of knowledge of v^{th} root with a composite modulus n was presented and proved to be secure. A variation of the proof protocol of knowledge of root in [10] is described here. In the protocol in Figure 1, a specific setting is employed: knowledge of N^{th} root modulo N^2 must be proved where N is a Paillier composite. The protocol is used to prove the knowledge of x, the N^{th} root of y and an integer in $Z^*_{N^2}$, where P and V stand for prover and verifier. This proof protocol is consistent with the Paillier setting and can be applied to verify the validity of Paillier encryption. Correctness of this protocol is straightforward. Namely, when the prover knows a N^{th} root of y, he can pass the verification. Since the setting is different from the original protocol in [10], it must be proved that the new protocol is sound with Paillier setting.

$P \rightarrow V:$ $b = r^N$ where r is randomly chosen from Z_N.
$V \rightarrow P:$ e, where $\|e\| = 160$.
$P \rightarrow V:$ $w = rx^e$
Verification: $w^N = by^e$

Fig. 1. Proof of Knowledge of N^{th} Root

Theorem 1. *The proof protocol of knowledge of N^{th} root in Figure 1 is specially sound if N is correctly generated. More precisely, if the prover can provide correct responses to two different challenges with a same commitment, he can calculate a N^{th} root of y efficiently.*

Proof: If the prover can provide responses w_1 and w_2 to a commitment b and two different challenges e_1 and e_2 where $e_1 > e_2$, such that

$$w_1^N = by^{e_1} \tag{1}$$

$$w_2^N = by^{e_2} \tag{2}$$

then (1) divided by (2) yields

$$(w_1/w_2)^N = y^{e_1-e_2}$$

According to the Euclidean algorithm, integers α and β can be found, such that $\beta(e_1 - e_2) = \alpha N + gcd(N, e_1 - e_2)$. As $N = pq$ is correctly generated, p and q are primes and the length of p and q is much longer than $|e_1 - e_2|$, so $gcd(N, e_1 - e_2) = 1$. So

$$(w_1/w_2)^{\beta N} = y^{\beta(e_1-e_2)} = y^{\alpha N+1}$$

Namely,

$$y = ((w_1/w_2)^{\beta}/y^{\alpha})^N$$

So, $(w_1/w_2)^{\beta}/y^{\alpha}$ is a N^{th} root of y. Note that the prover can calculate α and β efficiently from N and $e_1 - e_2$ using Euclidean algorithm. Therefore, the prover can get a N^{th} root of y efficiently. □

Theorem 2. *The proof protocol of knowledge of N^{th} root in Figure 1 is honest verifier zero knowledge.*

Proof: A simulator with no knowledge of any N^{th} root of y can choose e and w randomly and calculate $b = w^N/y^e$. Thus a simulated transcript composed of uniformly distributed b, e and w is obtained. The proof transcript generated by a prover with knowledge of an N^{th} root of y and an honest verifier (who chooses the challenge randomly and independently) is also composed of uniformly distributed e, w, b. These two transcripts are indistinguishable. So these two proof transcripts are indistinguishable. □

According to Theorem 1 and Theorem 2, this proof protocol is a so-called Σ-protocol [7]. So according to Damgard's analysis in [7], this proof is sound (the probability that a prover without the knowledge of a N^{th} root of y can pass the verification in this protocol is no more than 2^{-160}) and private (the prover's knowledge of N^{th} root of y is not revealed). Hash function $H()$ can be employed to generate the challenge as $e = H(y||b)$, so that the protocol becomes non-interactive. In the rest of this paper, non-interactive proof of knowledge of N^{th} root is applied. If $H()$ can be seen as a random oracle, security is not compromised in the non-interactive proof.

3.2 Proof of Knowledge of 1-Out-of-2 N^{th} Root mod $Z^*_{N^2}$

The proof protocol in Figure 2 is a combination of the proof of N^{th} root in Figure 1 and the proof of partial knowledge [5] to prove the knowledge of x, the N^{th} root of y_1 or y_2, integers in $Z^*_{N^2}$. For simplicity, it is supposed without losing generality $x^N = y_2$. Correctness of this protocol is straightforward. Namely, when the prover knows a N^{th} root of either y_1 or y_2, he can pass the verification. As the proof of knowledge of N^{th} root modulo N^2 in Section 3.1 and the partial proof technique in [5] are both specially sound and honest verifier ZK, this protocol is also specially sound and honest verifier ZK. Namely, the probability that a prover without the knowledge of a N^{th} root of y_1 or y_2 can pass the verification in this protocol is no more than 2^{-160} the prover's knowledge of N^{th} root of y_1 or y_2 is not revealed. Moreover, this protocol can also be extended to be non-interactive without compromising its security when a hash function regarded as a random oracle is used to generate the challenge e.

1. The prover chooses r, w_1 and e_1 randomly from Z^*_N, $Z^*_{N^2}$ and $\{0,1\}^{160}$ respectively. He calculates $b_1 = w_1^N y_1^{e_1}$ and $b_2 = r^N$.
2. The verifier randomly chooses a 160-bit challenge e.
3. The prover calculates $e_2 = e - e_1$ and $w_2 = r/x^{e_2}$.
4. The prover publishes e_1, w_1, e_2 and w_2. Anybody can verify $e = e_1 + e_2$ and $b_1 = w_1^N y_1^{e_1}$ and $b_2 = w_2^N y_2^{e_2}$.

Fig. 2. Proof of Knowledge of 1-out-of-2 N^{th} Root

3.3 A Combined Proof of Equality of Exponents and Knowledge of N^{th} Root

Let g_1, g_2, y_1 and y_2 be in $Z^*_{N^2}$. The proof protocol in Figure 3 is used to prove $PKN(x, r_1, r_2 \mid x \in Z, r_1 \in Z^*_N, r_2 \in Z^*_N, y_1 = g_1^x r_1^N, y_2 = g_2^x r_2^N)$. Correctness of this protocol is straightforward. Namely, if the prover knows x, r_1, r_2 and follows the protocol, the verifier will accept his proof. Soundness of this protocol seems at first to be straightforward if it is regarded as a combination of proof of equality of logarithms [4] and proof of knowledge of N^{th} root in Section 3.1, both

1. The prover chooses $v \in Z_N$, $u_1 \in Z^*_N$ and $n_2 \in Z^*_N$ randomly and calculates $\gamma = g_1^v u_1^N$ and $\theta = g_2^v u_2^N$. He sends γ and θ to the verifier.
2. The verifier randomly chooses a 160-bit challenge e and sends it to the prover.
3. The prover calculates $z_1 = v - ex$, $z_2 = u_1/r_1^e$, $z_3 = u_2/r_2^e$ and sends them to the verifier.
4. The verifier verifies $\gamma = g_1^{z_1} z_2^N y_1^e$ and $\theta = g_2^{z_1} z_3^N y_2^e$. He accepts the proof only if these two equations are correct.

Fig. 3. Combined Proof of Equality of Exponent and Knowledge of N^{th} Root

of which are sound. However, in this protocol, g_1 and g_2 may be in two different cyclic groups with different orders. As the proof of equality of logarithms in [4] can only be applied to prove equality of logarithms in a same group or two groups with a same order, it cannot be applied here. To the authors' knowledge, the only technique to prove equality of logarithms in groups with different orders was proposed by Bao [1]. However, his technique is only sound (passing his verification guarantee two logarithms in different groups with different orders are equal with a very large probability) but not correct (lots of equal logarithm pairs in the two groups cannot pass the verification with a very large probability) so can only be applied to his special application — a verifiable encryption scheme. As our protocol must be both correct and sound, our technique is different from his in that equality of exponents instead of equality of logarithms is proved. Namely, it is not required in our scheme that the two exponents are equal with two different moduluses. It is enough that the two exponents are equal without any modulus. Soundness of our protocol is proved in Theorem 3.

Theorem 3. *The proof protocol in Figure 3 is specially sound. More precisely, if the prover can provide correct responses for two different challenges to a same commitment, he can efficiently calculate* x, r_1 *and* r_2, *such that* $x \in Z$, $r_1 \in Z_N^*$, $r_2 \in Z_N^*$, $y_1 = g_1^x r_1^N$, $y_2 = g_2^x r_2^N$ *if* N *is correctly generated.*

Proof: If the prover can provide two sets of responses $z_{1,1}$, $z_{2,1}$, $z_{3,1}$ and $z_{1,2}$, $z_{2,2}$, $z_{3,2}$ for two different challenges e_1 and e_2 and the same commitment pair γ, θ, such that

$$\gamma = g_1^{z_{1,1}} z_{2,1}^N y_1^{e_1} \tag{3}$$

$$\theta = g_2^{z_{1,1}} z_{3,1}^N y_2^{e_1} \tag{4}$$

$$\gamma = g_1^{z_{1,2}} z_{2,2}^N y_1^{e_2} \tag{5}$$

$$\theta = g_2^{z_{1,2}} z_{3,2}^N y_2^{e_2} \tag{6}$$

(3) divided by (5) yields

$$g_1^{z_{1,1}} z_{2,1}^N y_1^{e_1} = g_1^{z_{1,2}} z_{2,2}^N y_1^{e_2}$$

So,

$$g_1^{z_{1,1}-z_{1,2}} (z_{2,1}/z_{2,2})^N = y_1^{e_2-e_1}$$

(4) divided by (6) yields

$$g_2^{z_{1,1}-z_{1,2}} (z_{3,1}/z_{3,2})^N = y_2^{e_2-e_1}$$

According to the Euclidean algorithm, integers α and β can be found, such that $\beta(e_1 - e_2) = \alpha N + gcd(N, e_1 - e_2)$. So

$$g_1^{\beta(z_{1,1}-z_{1,2})} (z_{2,1}/z_{2,2})^{\beta N} = y_1^{\alpha N + gcd(N, e_1-e_2)}$$

and

$$g_2^{\beta(z_{1,1}-z_{1,2})} (z_{3,1}/z_{3,2})^{\beta N} = y_2^{\alpha N + gcd(N, e_1-e_2)}$$

As $N = pq$ and the p and q are primes with length much longer than $|e_1 - e_2|$ (N is a correctly generated Paillier composite), $gcd(N, e_1 - e_2) = 1$. So,

$$g_1^{\beta(z_{1,1}-z_{1,2})}((z_{2,1}/z_{2,2})^{\beta}/y_1^{\alpha})^N = y_1 \tag{7}$$

and

$$g_2^{\beta(z_{1,1}-z_{1,2})}((z_{3,1}/z_{3,2})^{\beta}/y_2^{\alpha})^N = y_2 \tag{8}$$

Note that the prover can efficiently calculate α and β easily from N and $e_1 - e_2$ using Euclidean algorithm. Therefore, the prover can get $x = \beta(z_{1,1} - z_{1,2})$, $r_1 = (z_{2,1}/z_{2,2})^{\beta}/y_1^{\alpha}$ and $r_2 = (z_{3,1}/z_{3,2})^{\beta}/y_2^{\alpha}$ efficiently, such that $x \in Z$, $r_1 \in Z_N^*$, $r_2 \in Z_N^*$, $y_1 = g_1^x r_1^N$, $y_2 = g_2^x r_2^N$. □

Theorem 4. *The proof protocol in Figure 3 is honest verifier zero knowledge.*

This theorem can be proved like Theorem 2.

According to Theorem 3 and Theorem 4, the proof protocol in Figure 3 is sound (the probability that a prover without the required knowledge can pass the verification in this protocol is no more than 2^{-160}) and private (the prover's secret knowledge is not revealed). A hash function $H()$ can be employed to generate the challenge as $e = H(y_1||y_2||\gamma||\theta)$, so that the protocol becomes non-interactive. In the rest of this paper, the non-interactive version of this proof is applied. If $H()$ can be seen as a random oracle, security is not compromised in the non-interactive proof. Note that this protocol does not guarantee the secret knowledge x is smaller than $order(g_1)$ or $order(g_2)$. That is why we say that equality of exponents instead of equality of logarithms is included in this protocol.

4 Ciphertext Comparison

Suppose two L-bit messages m_1 and m_2 encrypted in c_1 and c_2 respectively are to be compared. The main idea of the comparison is comparing $F(m_1)$ and $F(m_2)$ where $F()$ is a monotonely increasing one-way function. Based on this idea, a comparison technique $Com(c_1, c_2)$ can be designed, such that $Com(c_1, c_2) = 1$ if $m_1 > m_2$; $Com(c_1, c_2) = 0$ if $m_1 = m_2$; $Com(c_1, c_2) = -1$ if $m_1 < m_2$. The comparison procedure is as follows.

1. An additive homomorphic encryption algorithm with encryption function $E()$ is employed, such that $E(x_1 + x_2) = E(x_1)E(x_2)$ and $E(ax) = E(x)^a$ for any messages x, x_1, x_2 and factor a. The public key is published while the private key is shared by participants $A_1, A_2, \ldots, A_m$. The message space of the encryption algorithm is $\{0, 1, \ldots, N-1\}$, where $2^{L+mL'} < \lfloor N/2 \rfloor$ and L' is a security parameter.
2. m_i is encrypted into $c_i = E(m_i)$ for $i = 1, 2$. It is proved that c_i is an encryption of a message with L bits without revealing the message for $i = 1, 2$.

3. Each A_l chooses a_l so that $a_l \in \{0, 1, \ldots, 2^{L'} - 1\}$ and calculates $c'_l = c'^{a_l}_{l-1}$ for $i = 1, 2$ where $c'_0 = c_1/c_2$. A_l proves $\log_{c'_{l-1}} c'_l < 2^{L'}$ without revealing a_l for $l = 1, 2, \ldots, m$.
4. The authorities cooperate to decrypt c'_m.

$$Com(c_1, c_2) = \begin{cases} 1 & \text{if } 0 < D(c'_m) \leq \lfloor N/2 \rfloor \\ 0 & \text{if } D(c'_m) = 0 \\ -1 & \text{if } D(c'_m) > \lfloor N/2 \rfloor \end{cases} \quad (9)$$

Any distributed additive homomorphic encryption algorithm can be employed in this ciphertext comparison. In this section, the ciphertext comparison protocol is described in detail based on distributed Paillier (see [9]).

4.1 Bit Encryption and Its Validity Verification

Messages m_1 and m_2 must be encrypted in a special way such that it is publicly verifiable from their encryptions that they are in the range $\{0, 1, \ldots, 2^L - 1\}$. So the following encryption-by-bit method is employed.

- Paillier encryption with distributed decryption (see [9]) is employed such that the parameters N, m, L' and L satisfy $2^{L+mL'} < (N-1)/2$ where m is the number of participants and N is the Paillier composite.
- Binary representation of m_i is a vector $(m_{i,1}, m_{i,2}, \ldots, m_{i,L})$ for $i = 1, 2$ where $m_{i,j} \in \{0, 1\}$ and $m_i = \sum_{j=1}^{L} m_{i,j} 2^{j-1}$.
- Component $m_{i,j}$ is encrypted with Paillier encryption to $c_{i,j} = g^{m_{i,j}} r_{i,j}^N \bmod N^2$ for $i = 1, 2$ and $j = 1, 2, \ldots, L$ where $r_{i,j}$ is randomly chosen from Z_N^*.
- The encrypted vectors $(c_{i,1}, c_{i,2}, \ldots, c_{i,L})$ for $i = 1, 2$ are published.
- The encryptor (millionaire or more generally message provider) proves that each $c_{i,j}$ is an encryption of 0 or 1 by providing a proof of knowledge of $c_{i,j}^{1/N}$ or $(c_{i,j}/g)^{1/N}$ for $i = 1, 2$ and $j = 1, 2, \ldots, L$. Proof of knowledge of 1-out-of-2 N^{th} root in the Paillier setting described in Section 3.2 is employed in the proof.
- Anybody can verify validity of $c_{i,j}$ for $i = 1, 2$ and $j = 1, 2, \ldots, L$. If the verification is passed, two ciphertexts $c_i = \prod_{j=1}^{L} c_{i,j}^{2^{j-1}} \bmod N^2 = g^{\sum_{j=1}^{L} m_{i,j} 2^{j-1}} r_i^N \bmod N^2 = g^{m_i} r_i^N \bmod N^2$ for $i = 1, 2$ are formed for comparison where $r_i = \prod_{j=1}^{L} r_{i,j}^{2^{j-1}} \bmod N$.

Only if the two ciphertexts are verified to be L bits long, can they be compared.

4.2 The Comparison Function

The authorities $A_1, A_2, \ldots, A_m$ compare $c_1 = g^{m_1} r_1^N \bmod N^2$ and $c_2 = g^{m_2} r_2^N \bmod N^2$ and validity of the comparison can be publicly verified.

1. a_l is selected randomly from $\{0, 1, \ldots, 2^{L'} - 1\}$ while its validity is guaranteed by bit encryption and its validity verification.

(a) A_l chooses a_l randomly from $\{0, 1, \ldots, 2^{L'} - 1\}$ with binary representation $(a_{l,1}, a_{l,2}, \ldots, a_{l,L'})$ where $a_l = \sum_{j=1}^{L'} a_{l,j} 2^{j-1}$. He keeps them secret and publishes $d_{l,j} = g^{a_{l,j}} t_{l,j}^N \bmod N^2$ for $j = 1, 2, \ldots, L'$ where $t_{l,j}$ is randomly chosen from Z_N^*.
(b) A_l proves that each $d_{l,j}$ contains 0 or 1 by providing a proof of knowledge of N^{th} root of either $d_{l,j}$ or $d_{l,j}/g$ modulus N^2 as proposed in Section 3.2.
(c) Anybody can verify the validity of $d_{l,j}$ for $j = 1, 2, \ldots, L'$ and calculates $d_l = \prod_{j=1}^{L'} d_{l,j}^{2^{j-1}} \bmod N^2$, which is a commitment of a_l.

Only if d_l for $l = 1, 2, \ldots, m$ are verified to be valid, the comparison continues.

2. Each A_l performs $c'_l = c'^{a_l}_{l-1} s_l^N \bmod N^2$ for $l = 1, 2, \ldots, m$ where $c'_0 = c_1/c_2 \bmod N^2$ and s_l is randomly chosen from Z_N^*. A_l has to give a proof

$$PKN(a_l, t_l, s_l \mid a_l \in Z, t_l \in Z_N^*, s_l \in Z_N^*,\ d_l = g^{a_l} t_1^N \bmod N^2, \\ c'_l = c'^{a_l}_{l-1} s_l^N \bmod N^2) \tag{10}$$

where $t_l = \prod_{j=1}^{L'} t_{l,j}^{2^{j-1}} \bmod N$. This proof can be implemented using the combined proof of equality of exponent and knowledge of N^{th} root proposed in Section 3.3 and is called *monotone proof*. Only if the verification is passed, the comparison continues.

3. The authorities corporately compute $Com(c_1, c_2)$ by decrypting c'_m.
4. Result of comparison

$$Com(c_1, c_2) = \begin{cases} 1 & \text{if } 0 < D(c'_m) \le (N-1)/2 \\ 0 & \text{if } D(c'_m) = 0 \\ -1 & \text{if } D(c'_m) > (N-1)/2 \end{cases} \tag{11}$$

5 Analysis

The ciphertext comparison technique is analysed and compared against the previous solutions to millionaire problem in this section.

5.1 Security and Efficiency Analysis

Theorem 5. *The proposed ciphertext comparison protocol is correct and sound. More precisely, assume it is infeasible for any A_l to find s and t, such that $t \neq 1 \bmod N$ and $g^s t^N = 1 \bmod N^2$, then iff $m_1 < m_2 < 2^L$, $Com(E(m_1), E(m_2)) = -1$; iff $m_1 = m_2 < 2^L$, $Com(E(m_1), E(m_2)) = 0$; iff $m_2 < m_1 < 2^L$, $Com(E(m_1), E(m_2)) = 1$.*

Proof: As the combined proof of equality of exponents and knowledge of N^{th} root in Section 3.3 is sound, *monotone proof* (Formula (10)) guarantees that A_l

knows a'_l, t'_l and s_l such that $a'_l \in Z$, $t'_l \in Z^*_N$, $s_l \in Z^*_N$, $d_l = g^{a'_l} t'^N_1 \bmod N^2$ and $c'_l = c'^{a'_l}_{l-1} s^N_l \bmod N^2$. As Paillier encryption algorithm is additive homomorphic,

$$\begin{gathered} D(c'_m) = D((c_1/c_2)^{\prod_{l=1}^m a'_l}) = \\ D(c_1^{\prod_{l=1}^m a'_l}/c_2^{\prod_{l=1}^m a'_l}) = D(c_1^{\prod_{l=1}^m a'_l}) - D(c_2^{\prod_{l=1}^m a'_l}) \bmod N \\ = D((g^{m_1} r_1^N)^{\prod_{l=1}^m a'_l}) - D((g^{m_2} r_2^N)^{\prod_{l=1}^m a'_l}) \bmod N \\ = D(g^{m_1 \prod_{l=1}^m a'_l} (r_1^{\prod_{l=1}^m a'_l})^N) - D(g^{m_2 \prod_{l=1}^m a'_l} (r_2^{\prod_{l=1}^m a'_l})^N) \bmod N \end{gathered}$$

In Section 4.1, validity of $d_{l,j}$ for $j = 1, 2, \ldots, L'$ is proved by A_l using the proof of knowledge of 1-out-of-2 N^{th} root proposed in Section 3.2. As the proof of knowledge of 1-out-of-2 N^{th} root is sound, it is guaranteed that A_l knows $a_{l,j}$ and $t_{l,j}$ for $j = 1, 2, \ldots, L'$, such that $t_{l,j} \in Z^*_N$, $a_{l,j} \in \{0, 1\}$ and $d_{l,j} = g^{a_{l,j}} t^N_{l,j} \bmod N^2$. As $d_l = \prod_{j=1}^{L'} d_{l,j}^{2^{j-1}} \bmod N^2$, A_l knows $a_l = \sum_{j=1}^{L'} a_{l,j} 2^{j-1}$ and $t_l = \prod_{j=1}^{L'} t_{l,j}^{2^{j-1}} \bmod N$, such that $t_l \in Z^*_N$, $a_l < 2^{L'}$ and $d_l = g^{a_l} t^N_l \bmod N^2$. Therefore, $g^{a'_l} t'^N_1 = g^{a_l} t^N_l \bmod N^2$. Namely, $g^{a'_l - a_l} (t'_l/t_l)^N = 1 \bmod N^2$. As a result, $t'_l = t_l \bmod N$, otherwise A_l can find t'_l/t_l and $a'_l - a_l$, such that $t'_l/t_l \neq 1 \bmod N$ and $g^{a'_l - a_l} (t'_l/t_l)^N = 1 \bmod N^2$, which is contradictory to the assumption that it is infeasible to find s and t, such that $t \neq 1 \bmod N$ and $g^s t^N = 1$ without knowledge of factorization of N. So $a_l = a'_l \bmod order(g)$. Therefore,

$$\begin{gathered} D(c'_m) = D(g^{m_1 \prod_{l=1}^m a_l} (r_1^{\prod_{l=1}^m a_l})^N) - D(g^{m_2 \prod_{l=1}^m a_l} (r_2^{\prod_{l=1}^m a_l})^N) \bmod N \\ = D(E(m_1 \prod_{l=1}^m a_l)) - D(E(m_2 \prod_{l=1}^m a_l)) \bmod N \\ = m_1 \prod_{l=1}^m a_l - m_2 \prod_{l=1}^m a_l \bmod N \end{gathered}$$

Since it has been publicly verified that $2^{L+mL'} < (N-1)/2$, $m_i \in \{0, 1, \ldots, 2^L - 1\}$ and $a_l \in \{0, 1, \ldots, 2^L - 1\}$ for $l = 1, 2, \ldots, m$, function $F(m_i) = m_i \prod_{l=1}^m a_l$ is monotonely increasing and smaller than $\lfloor N-1 \rfloor/2$. Therefore, if $m_1, m_2 < 2^L$,

$$D(c'_m) \begin{cases} \in (0, (N-1)/2] & \text{iff } m_1 > m_2 \\ = 0 & \text{iff } m_1 = m_2 \\ > (N-1)/2 & \text{iff } m_1 < m_2 \end{cases} \tag{12}$$

□

The assumption that it is infeasible for A_l to find s and r, such that $r \neq 1 \bmod N$ and $g^s r^N = 1 \bmod N^2$ without knowledge of factorization of N is correct because it seems reasonable to assume that given a constant z it is infeasible to find x and y, such that $f_1(x) f_2(y) = z$ where $f_1()$ and $f_2()$ are one-way functions. As factorization of N is kept secret to any single authority, both $f_1(x) = g^x \bmod N^2$ and $f_2(y) = y^N \bmod N^2$ are one-way functions to A_l. Moreover, if this assumption is incorrect, any Paillier ciphertext can be decrypted into multiple different messages, which is contradictory to the wide belief that Paillier encryption is secure. Therefore, this assumption is reliable.

The encryption verification in Section 4.1 is private as illustrated in Section 3.2. Privacy of the comparison in Section 4.2 is analysed as follows. If A_l does not reveal a_l, it is computationally infeasible for any other party to get any information about a_l from d_l as Paillier encryption is secure when the number of dishonest authorities is not over the threshold. Validity proof of d_l is private as it is a proof of knowledge of 1-out-of-2 N^{th} root proposed in Section 3.2. *Monotone proof* (Formula (10)) is private as it is a combined proof of knowledge of 1-out-of-2 N^{th} root and equality of exponents proposed in Section 3.3. So a_l is not revealed in these two proofs. So $m_1 - m_2$ is not revealed from $D(c'_m)$, which is equal to $\prod_{l=1}^{m} a_l(m_1 - m_2) \bmod N$. Therefore, none of m_1, m_2 or $m_1 - m_2$ are revealed. However, when $D(c'_m)$ is too near to the boundaries of its value domain (in $(0, 2^L)$, $(N-2^L, N)$, $(2^{mL'}, 2^{L+mL'})$ or $(N-2^{L+mL'}, N-2^{mL'})$) partial information is revealed from $m_1 - m_2$. The revelation of partial information is demonstrated in Table 1. An example is given in Table 1, where N is 1024 bits long (according to the widely accepted security standard) and $L = 40$ (large enough for practical applications). As illustrated in Table 1, $D(c'_m)$ is usually far away from the boundaries and is in the four special ranges with an overwhelmingly small probability. So, the ciphertext comparison protocol is private with an overwhelmingly large probability. Therefore, the whole ciphertext comparison protocol is private with an overwhelmingly large probability.

Table 1. Partial information revelation from $m_1 - m_2$

	Phenomenon	Revelation	Probability	
			value	example
Case 1	$D(c'_m) \in (0, 2^L)$	$m_1 - m_2 \in (0, D(c'_m)]$	$2^L/N$	2^{-984}
Case 2	$D(c'_m) \in (N - 2^L, N)$	$m_2 - m_1 \in (0, D(c'_m)]$	$2^L/N$	2^{-984}
Case 3	$D(c'_m) \in (2^{mL'}, 2^{L+mL'})$	$m_1 - m_2 \in [D(c'_m)/2^{mL'}, 2^L)$	$2^L/N$	2^{-984}
Case 4	$D(c'_m) \in (N - 2^{L+mL'}, N - 2^{mL'})$	$m_2 - m_1 \in [D(c'_m)/2^{mL'}, 2^L)$	$2^L/N$	2^{-984}
Totally			$2^{L+2}/N$	2^{-982}

As the encryption verification in Section 4.1 is a proof of knowledge of 1-out-of-2 N^{th} root proposed in Section 3.2, it is sound, namely a ciphertext containing an invalid message can pass the verification with a negligible probability. So the ciphertext comparison protocol is robust. As each step in the ciphertext comparison protocol is publicly verifiable, public verifiability is achieved. As no complex circuit is used, the ciphertext comparison scheme is quite efficient.

5.2 Comparison

A comparison between the new solution to the millionaire problem and the existing solutions is provided in Table 2, where the modular multiplications are counted in regard to computation and transportation of integers with significant length (e.g. 1024 bits long) is counted in regard to communication. The

schemes in [6] and [2] are similar to [11] and [16] respectively, so are not analysed separately. K is the full-length of exponent, t is the cutting factor in the cut-and-choose mechanism in [13] and [12], λ is a parameter in [8] and T is a parameter in [15]. An example is used in Table 2, where fair values are chosen for the parameters: $|N| = 1024$, $K = 1024$, $m = 3$, $L = 100$, $L' = 10$, $t = 40$, $\lambda = 40$ and $T = 20$. According to this comparison, the proposed ciphertext comparison technique is the only efficient, publicly verifiable, private and precise solution to millionaire problem.

Table 2. Property comparison

	Public	Precise	Computation		Communication	
	verifiability	result	cost	example	cost	example
[13]	No	No	$\geq 15KLt$	$\geq$ 40960000	$\geq 37Lt + 2t$	$\geq$ 148080
[12]	Yes	No	$\geq 15KLt$	$\geq$ 40960000	$\geq 37Lt + 2t$	$\geq$ 148080
[11]	Yes	No	average $4665KL + 6$	477696006	average $1626L + 6$	162606
[3]	Yes	No	average $\geq 4039.5KL$	$\geq$ 413644800	$\geq 1543L$	$\geq$ 154300
[16]	No	No	$> L^4$	$>$ 100000000	$\geq 343L^3$	$\geq$ 343000000
[8]	No	No	$(1.5\lambda L(L+3)) + (\lambda+1)L(L+1))/2$	516050	$L(\lambda+2)$	4200
[15]	Yes	No	$(L+2)(L-1)(1+0.5T) + 37.5KL$	4052058	$25L$	2500
Proposed	Yes	Yes	$1.5K(5mL' + 8m + 10L)$	1803264	$5mL' + 4m + 10L$	1162

6 Conclusion

A new cryptographic technique —ciphertext comparison— is proposed to compare two ciphertexts and determine which contains a larger message. This new technique is the only efficient and publicly verifiable solution to the millionaire problem. It is also the only precise solution to the millionaire problem. In the new scheme privacy of the two messages is protected with an overwhelmingly large probability.

References

1. Feng Bao. An efficient verifiable encryption scheme for encryption of discrete logarithms. In *the Smart Card Research Conference, CARDIS'98*, volume 1820 of *Lecture Notes in Computer Science*, pages 213–220, Berlin, 1998. 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, Berlin, 2000. Springer.
3. Christian Cachin and Jan Camenisch. Optimistic fair secure computation (extended abstract). In *CRYPTO '00*, volume 1880 of *Lecture Notes in Computer Science*, pages 94–112, Berlin, 2000. Springer-Verlag.
4. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105, Berlin, 1992. Springer-Verlag.

5. R. Cramer, I. B. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, Berlin, 1994. Springer-Verlag.
6. 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, Berlin, 2001. Springer.
7. Ivan Damgård and Ronald Cramer. On $\sum$-protocols. *Cryptologic Protocol Theory*, 2002. Available as `http://www.daimi.au.dk/~ivan/Sigma.ps`.
8. 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, Berlin, 2001. Springer.
9. Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern. Sharing decryption in the context of voting or lotteries. In *Financial Cryptography 2000*, pages 90–104, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science 1962.
10. L. C. Guillou and J. J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231, Berlin, 1989. Springer-Verlag.
11. M Jakobsson and A Juels. Mix and match: Secure function evaluation via ciphertexts. In *ASIACRYPT '00*, volume 1976 of *Lecture Notes in Computer Science*, pages 143–161, Berlin, 2000. Springer-Verlag.
12. A. Juels and M. Szydlo. A two-server, sealed-bid auction protocol. In *The Sixth International Conference on Financial Cryptography 2002*, volume 2357 of *Lecture Notes in Computer Science*, pages 72–86, Berlin, 2002. Springer-Verlag.
13. Moni Naor, Benny Pinkas, and Reuben Sumner. Privacy perserving auctions and mechanism design. In *ACM Conference on Electronic Commerce 1999*, pages 129–139, 1999.
14. P Paillier. Public key cryptosystem based on composite degree residuosity classes. In *EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Berlin, 1999. Springer-Verlag.
15. Kun Peng, Colin Boyd, Ed Dawson, and Byoungcheon Lee. An efficient and verifiable solution to the millionaire problem. In *Pre-Proceedings of ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 315–330, Berlin, 2004. Springer-Verlag.
16. 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.
17. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *IEEE Symposium on Foundations of Computer Science 1982, FOCS 1982*, pages 160–164, 1992.