Improvement of Lin-Tzeng Solution to Yao’s Millionaires Problem and Its Cheating Advantage Analysis

Zhengjun Cao$^1$, Lihua Liu$^2$

Abstract

In 2005, Lin and Tzeng proposed a solution to Yao’s Millionaires problem in the setting of semi-honest parties. At the end of the protocol only the party (Alice) who is responsible for setting up the system parameters knows the outcome. It does not specify how to have the other party (Bob) know the result. In this note, we present an improvement of the Lin-Tzeng solution. It requires that Alice and Bob alternately perform the original protocol twice. Under the reasonable assumption that a participator does not deviate from the prescribed steps before he/she obtains the outcome, Alice and Bob can almost simultaneously obtain the result. To the best of our knowledge, it is the first time to show that one participator has only an advantage of $\ln n/n$ possibility to cheat the other in the reasonable setting.

Keywords. Multiplicative homomorphic encryption; Semi-honest assumption; Exponentially lifting transform.

1 Introduction

Yao’s Millionaires problem is a special example of the general secure multiple-party computation. In the scenario, Alice has a secret integer $x$ and Bob has a secret integer $y$. They want to know $x > y$ or $x \leq y$, without leaking any information of $x$ and $y$. Yao [15] first presented a solution to the problem. Since then, the solutions [1, 4, 7, 12, 13] have been proposed. These solutions used additive or XOR homomorphic encryption schemes. In 2005, Lin and Tzeng [9] proposed a solution based on the ElGamal encryption which is probabilistic and multiplicative homomorphic. In 2011, Y. Huang et al [6] proposed a fast secure two-party computation using garbled circuits. Among these solutions, Lin-Tzeng protocol is more easily to accept by cryptoresearchers because it is directly based on mathematically intractable problems.

In this note, we shall point out that at the end of Lin-Tzeng solution only the party (Alice) who is responsible for setting up the system parameters knows the outcome. They did not

$^1$Department of Mathematics, Shanghai University, Shanghai, China.

$^2$Department of Mathematics, Shanghai Maritime University, Shanghai, China. liulh@shmtu.edu.cn
specify how to have the other party (Bob) know the result. We stress that in the scenario it is not preferable to require that Alice honestly tells Bob the result. We present an improvement of the Lin-Tzeng solution. It requires that Alice and Bob alternately perform the original protocol twice. At last, Alice and Bob can almost simultaneously obtain the result. It is the first time to show that one participator has only an advantage of $\ln n/n$ possibility to cheat the other in the reasonable setting. We also remark that it is no use to introduce an obvious transfer protocol for exchanging the sequences obtained by the participators.

2 Preliminary

Yao’s Millionaires problem: Alice has an integer $x$, and Bob has an integer $y$. They want to know $x \leq y$, or $x > y$, without leaking any information of $x$ and $y$.

**Multiplicative homomorphic encryption.** An encryption $E$ is multiplicative homomorphic if for arbitrary $m_1, m_2$ it satisfies $E(m_1) \times E(m_2) = E(m_1 \times m_2)$.

The ElGamal encryption can be described as follows.

[Setup] Pick $p = 2q + 1$, where $p$ and $q$ are two big primes. Let $G_p$ denote the group (multiplicative) of the residues modulo $p$. Let $G_q$ denote a subgroup of order $q$ which is generated by $g$. Publish $p, q, g, h = g^\alpha$ and keep $\alpha \in \mathbb{Z}_q^*$ in secret.

[Encryption] Given $m \in G_q$, pick $r \in \mathbb{Z}_q^*$ and compute $c = E(m) = (a, b) = (g^r, mh^r)$.

[Decryption] Given $c = (a, b)$, compute $D(c) = b \times a^\alpha = m$.

It is easy to check that ElGamal encryption is multiplicative homomorphic. In fact, we have

$$E(m_1) \times E(m_2) = (g^{r_1}, m_1 h^{r_1}) \times (g^{r_2}, m_2 h^{r_2}) = (g^{r_1 + r_2}, (m_1 \times m_2) h^{r_1 + r_2}) = E(m_1 \times m_2)$$

If $c = E(1) = (a, b)$, then the exponentially lifting transform $c' = c^k = (a^k, b^k)$, where $k \in \mathbb{Z}_q^*$, has the property that $D(c') = D(c) = 1$. This is due to that $b' \times (a')^\alpha = m^k = 1^k = 1$. Notice that the Lin-Tzeng solution depends on this property.

3 Lin-Tzeng solution revisited

The Lin-Tzeng scheme uses the binary representations of $x$ and $y$ to define two sets $S_x^1$ and $S_y^0$. It then proves that

$$x > y \iff S_x^1 \cap S_y^0 \neq \emptyset$$

Let $s = s_n s_{n-1} \cdots s_1 \in \{0, 1\}^n$ be a string of length $n$. The sets $S_y^0$ and $S_x^1$ are defined as follows:

$$S_y^0 = \{s_n s_{n-1} \cdots s_{i+1} 1 \mid s_i = 0, 1 \leq i \leq n \}$$
Let \( S^1 = \{s_n s_{n-1} \cdots s_{i+1} s_i \mid s_i = 1, 1 \leq i \leq n\} \)

Example 1. \( s = 101101, S^0_s = \{11, 10111\}, S^1_s = \{1, 101, 1011, 101101\}\). Actually, \(|S^0_s| = k\), where \( k \) is the number of bit 0 contained in the string \( s \). \(|S^1_s| = l\), where \( l \) is the number of bit 1 contained in the string \( s \).

Example 2. \( x = 101110, y = 101101, S^1_x = \{1, 101, 1011, 10111\}, S^0_y = \{11, 10111\}\). Since \( S^1_x \cap S^0_y = \{10111\} \neq \emptyset \), we have \( x > y \).

We now describe the Lin-Tzeng solution.

1. Alice is responsible for setting up the parameters of ElGamal encryption. She then picks \( \alpha \in Z_q^* \) and computes \( h = g^{-\alpha} \). Publish \( p, q, g, h \).
2. Alice uses the binary representation \( x = x_n x_{n-1} \cdots x_1 \in \{0, 1\}^n \) to construct a \( 2 \times n \) table \( T \) where \( T[i, j], i \in \{0, 1\}, 1 \leq j \leq n \), and

\[
T[x_i, i] = E(1)_i, \quad T[\bar{x}_i, i] = E(r_i), \quad r_i \in G_q.
\]

Send \( T \) to Bob.

Notice that \( E(1)_i \) denotes the ciphertext of unit 1 encrypted by ElGamal encryption which is placed in the \( i \)-th column. Since each column has a ciphertext of unit 1, it is confusing to simply use the notation \( E(1) \) as the original [9]. Due to that ElGamal encryption is probabilistic, we have \( E(1)_i \neq E(1)_j \) if \( i \neq j \). For example, if \( x = x_n x_{n-1} \cdots x_1 = 1101 \cdots 1 \), then \( T \) is generated as follows

\[
\begin{array}{ccccccc}
 n & n-1 & n-2 & n-3 & \cdots & 1 \\
 0 & E(r_n) & E(r_{n-1}) & E(1)_{n-2} & E(r_{n-3}) & \cdots & E(r_1) \\
 1 & E(1)_n & E(1)_{n-1} & E(r_{n-2}) & E(1)_{n-3} & \cdots & E(1)_1 \\
\end{array}
\]

Since the randomness of \( E(r_i), E(1)_i, 1 \leq i \leq n \), Bob can not determine the position of \( E(1)_i, 1 \leq i \leq n \).

3. Bob uses the binary representation \( y = y_n y_{n-1} \cdots y_1 \) to construct the set \( S^0_y \). For \( t = t_n t_{n-1} \cdots t_i \in S^0_y \), he looks up the table \( T \) for \( T[t_j, j], i \leq j \leq n \), and computes

\[
c_t = T[t_n, n] \times T[t_{n-1}, n-1] \cdots T[t_i, i]
\]

Using the exponential lifting transform, he obtains \( c'_t \) from \( c_t \). Let \( |S^0_y| = \lambda \). He obtains \( c'_1, c'_2, \cdots, c'_\lambda \). Let \( l = n - \lambda \). He picks \( l \) random \( z_j = (a_j, b_j) \in G_q^2, 1 \leq j \leq l \) and construct the following sequence

\[
z_1, \cdots, z_l, c'_1, \cdots, c'_\lambda
\]

Randomly permutate the sequence to obtain \( \hat{c}_1, \cdots, \hat{c}_n \). Send the resulting sequence to Alice.
4. Alice computes \( m_i = D(\hat{c}_i), 1 \leq i \leq n \). If there exists \( m_i = 1 \), then she concludes that \( x > y \). Otherwise, \( x \leq y \).

**Correctness.** If \( S_x^1 \cap S_y^0 = \hat{t} \), then \( \hat{t} = \hat{t}_n \hat{t}_{n-1} \ldots \hat{t}_i = x_n x_{n-1} \ldots x_i \). Hence,

\[
\begin{align*}
c_i &= T[\hat{t}_n, n] \times T[\hat{t}_{n-1}, n-1] \times \cdots \times T[\hat{t}_i, i] \\
&= T[x_n, n] \times T[x_{n-1}, n-1] \times \cdots \times T[x_i, i] \\
&= E(1)_n E(1)_{n-1} \cdots E(1)\end{align*}
\]

Therefore, \( E(1)_n E(1)_{n-1} \cdots E(1) = E(1), D(c_i) = D(E(1)) = 1 \).

The following Table 1 summarizes the steps of Lin-Tzeng solution.

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick ( \alpha \leftarrow Z_q^* ), compute ( h \leftarrow g^{-\alpha} ).</td>
<td>( p; q; g; h ) → 99K</td>
</tr>
</tbody>
</table>
| For \( x = x_n x_{n-1} \ldots x_1 \), construct \( T = \{T[i, j] \mid 0 \leq i \leq 1, 1 \leq j \leq n, T = \{T[i, j] \mid 0 \leq i \leq 1, 1 \leq j \leq n, \text{ where} \} \), compute \( T[x_i, i] = E(1)_i, T[\bar{x}_i, i] = E(r_i) \). \( \forall r_i \in G_q \). \( T \rightarrow \) For \( y = y_n y_{n-1} \ldots y_1 \), compute \( S_y^0 = \{y_n y_{n-1} \ldots y_{i+1} \mid y_i = 0, 1 \leq i \leq n\} \). \( \lambda, l = n - \lambda \). \( \lambda, l \) or \( \lambda, l \).
| Exponentially lift \( c_t \) to \( c'_t \). \( c_t = T[t_n, n] \times \cdots \times T[t_i, i] \). | Pick \( z_j = (a_j, b_j), 1 \leq j \leq l \), construct a sequence \( z_1, \ldots, z_l, c'_1, \ldots, c'_\lambda \). \( c'_1, \ldots, c'_\lambda \).  \( \hat{c}_1, \ldots, \hat{c}_n \) \rightarrow \( \hat{c}_1, \ldots, \hat{c}_n \). |
| Compute \( m_i = D(\hat{c}_i), 1 \leq i \leq n \). If there is \( m_i = 1 \), then \( x > y \). |  |

### 4 Improvement of Lin-Tzeng solution and its cheating advantage analysis

At the end of the Lin-Tzeng protocol, it does not specify how to have Bob know the outcome. The usual measure is to require that Alice honestly tells Bob the result. But in the scenario it is not preferable because Alice can cheat Bob in the stage. In the original scheme, the semi-honest requirement is not explicitly specified. We think the following explicit assumption is more reasonable.
Semi-honest assumption: A participator does not deviate from the prescribed steps before he/she obtains the outcome. Once he/she obtains the result, he/she will try to cheat the other participator.

4.1 Improvement

As a modification, we suggest having Alice and Bob alternately perform the original protocol twice. To keep them in symmetric positions, they should agree to the system parameters \((p, q, g)\) for ElGamal encryption. See the following table 2 for the other steps.

<table>
<thead>
<tr>
<th>Alice</th>
<th>(x = x_nx_{n-1} \cdots x_1)</th>
<th>((p, q, g))</th>
<th>Bob</th>
<th>(y = y_ny_{n-1} \cdots y_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick (\alpha \leftarrow Z_q^*), compute (h \leftarrow g^{-\alpha}).</td>
<td>Pick (\beta \leftarrow Z_q^*), compute (h \leftarrow g^{-\beta}).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Construct (T = {T[i, j]}_{0 \leq i, 1 \leq j \leq n}), where (T[x_i, i] = E(1)_i), (T[\bar{x}_i, i] = E(r_i)), (\forall r_i \in G_q).</td>
<td>Construct (\bar{T} = {\bar{T}[i, j]}_{0 \leq i, 1 \leq j \leq n}), where (\bar{T}[y_i, i] = E(1)_i), (\bar{T}[\bar{y}_i, i] = E(\bar{r}_i)), (\forall \bar{r}_i \in G_q).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compute (S_0^x = {x_nx_{n-1} \cdots x_{i+1}</td>
<td>x_i = 0, 1 \leq i \leq n}).</td>
<td>Compute (S_0^y = {y_ny_{n-1} \cdots y_{i+1}</td>
<td>y_i = 0, 1 \leq i \leq n}).</td>
<td></td>
</tr>
<tr>
<td>Let (</td>
<td>S_0^x</td>
<td>= \lambda), (l = n - \lambda) for all (t), (t_n t_{n-1} \cdots t_i \in S_0^x), compute (c_i = T[t_n, i] \cdots \times T[t_i, i]) to (c'_i).</td>
<td>Let (</td>
<td>S_0^y</td>
</tr>
<tr>
<td>Pick (z_j = (a_j, b_j), 1 \leq j \leq l), construct a sequence (z_1, \cdots, z_l, c'<em>1, \cdots, c'</em>\lambda).</td>
<td>Pick (\bar{z}_j = (\bar{a}_j, \bar{b}_j), 1 \leq j \leq l'), construct a sequence (\bar{z}<em>1, \cdots, \bar{z}</em>{l'}, \bar{d}<em>1, \cdots, \bar{d}</em>{\lambda'}).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permutate it to obtain (\hat{c}_1, \cdots, \hat{c}_n).</td>
<td>Permutate it to obtain (\hat{d}_1, \cdots, \hat{d}_n).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use (\alpha) to compute (m_i = D(\hat{d}_i), 1 \leq i \leq n). If there is (m_i = 1), then (x &gt; y).</td>
<td>Use (\beta) to compute (m_j = D(\hat{c}_j), 1 \leq j \leq n). If there is no (m_j = 1), then (x &gt; y).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.2 Cheating advantage analysis

Due to the unbalanced positions of two participators, the works [1, 4, 7, 12, 13] did not consider the advantage of that one participator cheats the other party. In the above modification, it is easy to see that the positions of Alice and Bob are symmetric. Having the explicit semi-honest assumption, we now estimate cheating advantages.

Lemma 1. Suppose $x = x_n x_{n-1} \cdots x_1 \in \{0, 1\}^n$ and $y = y_n y_{n-1} \cdots y_1 \in \{0, 1\}^n$. The sets $S_x^0$ and $S_x^1$ are defined as follows: $S_x^0 = \{x_n x_{n-1} \cdots x_{i+1} | x_i = 0, 1 \leq i \leq n\}$, $S_x^1 = \{x_n x_{n-1} \cdots x_{i+1} x_i | x_i = 1, 1 \leq i \leq n\}$. If $x \neq y$, then $|S_x^0 \cap S_y^1| = 1$ or 0.

Proof. If $a, b \in S_x^0 \cap S_y^1, a < b$, then $a$ must be of the form $a = x_n \cdots x_{i+1}1 = y_n \cdots y_{i+1}y_i \in \{0, 1\}^{n-i}$, and $x_i = 0, y_i = 1$, for some index $i$. On the one hand, by the definition of $S_x^0$, we know that $b$ must be of the form $b = x_n \cdots x_{i+1}0 x_{i-1} \cdots x_{j+1}1 \in \{0, 1\}^{n-j}, x_j = 0$ for some index $j$. On the other hand, by the definition of $S_y^1$, $b$ must be of the form $b = y_n \cdots y_{i+1}1 y_{i-1} \cdots y_j \in \{0, 1\}^{n-j}, y_j = 1$.

This leads to a contradiction. 

After Alice and Bob obtain $\hat{c}_1, \cdots, \hat{c}_n$ and $\hat{d}_1, \cdots, \hat{d}_n$, respectively, they should alternately exchange $\hat{c}_i, \hat{d}_i$ one by one. In the step of exchanging $\hat{c}_i$ and $\hat{d}_i$, Bob only has the advantage of $1/(n+1-i)$ possibility to cheat Alice because $|S_x^0 \cap S_y^1| = 1$ or 0. On average, we have the following bound

$$\frac{1/n + 1/(n-1) + \cdots + 1}{n} \approx \ln n/n.$$ 

That is, Bob has an advantage of $\ln n/n$ possibility to cheat Alice. So does Alice. Therefore, due to the randomness of two sequences $\hat{c}_1, \cdots, \hat{c}_n$ and $\hat{d}_1, \cdots, \hat{d}_n$, Alice and Bob will almost simultaneously obtain the result. By the way, both two protocols in [1] and [4] do not specify how to have Bob know the result, and not estimate the cheating advantage. To the best of our knowledge, it is the first time to show that one participator has only an advantage of $\ln n/n$ possibility to cheat the other in the reasonable setting.

5 Further discussion

To prevent Alice from cheating Bob, one might introduce an oblivious transfer protocol [2, 5, 8, 10, 11, 14] into the original scheme. Concretely, it requires that Alice and Bob alternatively
execute the original protocol twice. Denote the sequence obtained by Bob as \( \hat{d}_1, \ldots, \hat{d}_n \), and denote the sequence obtained by Alice as \( \hat{c}_1, \ldots, \hat{c}_n \). Alice and Bob exchange \( \hat{d}_1, \ldots, \hat{d}_n \) and \( \hat{c}_1, \ldots, \hat{c}_n \) using the oblivious transfer protocol. We here stress that the transferred \( \hat{d}_1, \ldots, \hat{d}_n \) and \( \hat{c}_1, \ldots, \hat{c}_n \) are not recognizable. Any party, say, Alice, can cheat the other party by transferring an arbitrary sequence \( \bar{c}_1, \ldots, \bar{c}_n \). If Bob honestly transfers \( \hat{d}_1, \ldots, \hat{d}_n \), then he will be cheated. Thus, the primitive of oblivious transfer is not applicable to the Lin-Zeng solution.

We here point out that in most reasonable applications of OT, the transferred messages must be recognizable for the receiver, or the sender is willing to disclose some messages to the receiver. The property has been explicitly specified in the earlier works by Rabin [11], Even, Goldreich and Lempel [3]. It stressed that:

The notion of a “recognizable secret message” plays an important role in our definition of OT. A message is said to be a recognizable secret if, although the receiver cannot compute it, he can authenticate it once he receives it.

The notion of a recognizable secret message is evidently relevant to the study of cryptographic protocols, in which the sender is reluctant to send the message while the receiver wishes to get it. In such protocols, it makes no sense to consider the transfer of messages that are either not secret (to the receiver) or not recognizable (by the receiver).

In symmetric case, such as signing contracts, both two participators can easily verify the correctness of the received messages. In unsymmetric case, such as a database manager plays the role of the sender and a client plays the role of the receiver, it is usual that the sender is willing to disclose some messages to the receiver. To sum up, if the transferred messages are not recognizable then the receiver can not decide which message to retrieve. But it is a pity that in most cases the transferred messages are not recognizable.

6 Conclusion

We improve the Lin-Tzeng solution to Yao’s Millionaires problem by having the two participators alternately perform the original Lin-Zeng protocol twice. We specify the assumption that a participator does not deviate from the prescribed steps before he/she obtains the outcome. We also estimate the cheating advantage under the reasonable assumption.

References


