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 crypto-researchers 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 $\alpha \in Z_q^*$ in secret.

[Encryption] Given $m \in G_q$, pick $r \in Z_q^*$ 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 $k \in Z_q^*$, 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

$$x > y \Longleftrightarrow S_x^1 \cap S_y^0 \neq \varnothing$$

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

$$S_s^0 = \{s_n s_{n-1} \cdots s_{i+1} 1 \mid s_i = 0, 1 \leq i \leq n\}$$

$$S_s^1 = \{s_n s_{n-1} \cdots s_{i+1} s_i \,|\, s_i = 1, 1 \le i \le n\}$$

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

Example 2. $x = 101110$, $y = 101101$. $S_x^1 = \{1, 101, 1011, 10111\}$, $S_y^0 = \{11, 10111\}$. Since $S_x^1 \cap S_y^0 = \{10111\} \ne \varnothing$, 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 $T[i,j], i \in \{0,1\}, 1 \le j \le n$, and

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

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 $E(1)_i \ne E(1)_j$ if $i \ne j$. For example, if $x = x_n x_{n-1} \cdots x_1 = 1101 \cdots 1$, then T is generated as follows

	n	$n-1$	$n-2$	$n-3$	$\cdots$	1
0	$E(r_n)$	$E(r_{n-1})$	$\underline{E(1)_{n-2}}$	$E(r_{n-3})$	$\cdots$	$E(r_1)$
1	$\underline{E(1)_n}$	$\underline{E(1)_{n-1}}$	$E(r_{n-2})$	$\underline{E(1)_{n-3}}$	$\cdots$	$\underline{E(1)_1}$

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

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

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

Using the exponential lifting transform, he obtains c_t' from c_t. Let $|S_y^0| = \lambda$. He obtains $c_1', c_2', \cdots, c_\lambda'$. Let $l = n - \lambda$. He picks l random $z_j = (a_j, b_j) \in G_q^2, 1 \le j \le l$ 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 $\hat{t} = \hat{t}_n \hat{t}_{n-1} \cdots \hat{t}_i = x_n x_{n-1} \cdots x_i$. Hence,

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

Therefore, $E(1)_n E(1)_{n-1} \cdots E(1)_i = E(1)$, $D(c_{\hat{t}}) = D(E(1)) = 1$.

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

Table 1: Lin-Tzeng solution

Alice		Bob
Pick $\alpha \leftarrow Z_q^*$, compute $h \leftarrow g^{-\alpha}$.		
	$\xrightarrow{p,q,g,h}$	
For $x = x_n x_{n-1} \cdots x_1$, construct $T = \{T[i,j]\}_{0 \leq i \leq 1, 1 \leq j \leq n}$, where $T[x_i, i] = E(1)_i,\ T[\bar{x}_i, i] = E(r_i),\ \forall\, r_i \in G_q$.	$\xrightarrow{T}$	For $y = y_n y_{n-1} \cdots y_1$, compute $S_y^0 = \{y_n y_{n-1} \cdots y_{i+1} 1 \,\vert\, y_i = 0, 1 \leq i \leq n\}$. Let $\vert S_y^0 \vert = \lambda$, $l = n - \lambda$. For $\forall\, t = t_n t_{n-1} \cdots t_i \in S_y^0$, compute $c_t = T[t_n, n] \cdots \times T[t_i, i]$. Exponentially lift c_t to c'_t. Pick $z_j = (a_j, b_j), 1 \leq j \leq l$, construct a sequence $z_1, \cdots, z_l, c'_1, \cdots, c'_\lambda$. Permutate it to obtain
	$\xleftarrow{\hat{c}_1, \cdots, \hat{c}_n}$	$\hat{c}_1, \cdots, \hat{c}_n$.
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 2: Modified Lin-Tzeng solution

Alice	(p,q,g)	Bob
$x = x_n x_{n-1} \cdots x_1$		$y = y_n y_{n-1} \cdots y_1$
Pick $\alpha \leftarrow Z_q^*$, compute $h \leftarrow g^{-\alpha}$.		Pick $\beta \leftarrow Z_q^*$, compute $\widetilde{h} \leftarrow g^{-\beta}$.
	$\xrightarrow{h}$	
	$\xleftarrow{\widetilde{h}}$	
Construct $T = \{T[i,j]\}_{0 \le i \le 1, 1 \le j \le n}$, where $T[x_i, i] = E(1)_i,\ T[\bar{x_i}, i] = E(r_i),\ \forall\, r_i \in G_q$.		Construct $\widetilde{T} = \{\widetilde{T}[i,j]\}_{0 \le i \le 1, 1 \le j \le n}$, where $\widetilde{T}[y_i, i] = E(1)_i,\ \widetilde{T}[\bar{y_i}, i] = E(\widetilde{r}_i),\ \forall\, \widetilde{r}_i \in G_q$.
	$\xrightarrow{T}$	
	$\xleftarrow{\widetilde{T}}$	
Compute $S_x^0 =$ $\{x_n x_{n-1} \cdots x_{i+1} 1 \mid x_i = 0, 1 \le i \le n\}$. Let $\lvert S_x^0 \rvert = \lambda$, $l = n - \lambda$. For $\forall\, t = t_n t_{n-1} \cdots t_i \in S_x^0$, compute $c_t = T[t_n, n] \cdots \times T[t_i, i]$. Exponentially lift c_t to c'_t. Pick $z_j = (a_j, b_j), 1 \le j \le l$, construct a sequence $z_1, \cdots, z_l, c'_1, \cdots, c'_\lambda$. Permutate it to obtain $\hat{c}_1, \cdots, \hat{c}_n$.		Compute $S_y^0 =$ $\{y_n y_{n-1} \cdots y_{i+1} 1 \mid y_i = 0, 1 \le i \le n\}$. Let $\lvert S_y^0 \rvert = \lambda'$, $l' = n - \lambda'$. For $\forall\, s = s_n s_{n-1} \cdots s_i \in S_y^0$, compute $d_s = T[s_n, n] \cdots \times T[s_i, i]$. Exponentially lift d_s to $\widetilde{d}_s$. Pick $\widetilde{z}_j = (\widetilde{a}_j, \widetilde{b}_j), 1 \le j \le l'$, construct a sequence $\widetilde{z}_1, \cdots, \widetilde{z}_{l'}, \widetilde{d}_1, \cdots, \widetilde{d}_{\lambda'}$. Permutate it to obtain $\hat{d}_1, \cdots, \hat{d}_n$.
	$\xrightarrow{\hat{c}_1}$	
	$\xleftarrow{\hat{d}_1}$	
	$\vdots$	
	$\xrightarrow{\hat{c}_n}$	
	$\xleftarrow{\hat{d}_n}$	
Use α to compute $m_i = D(\hat{d}_i), 1 \le i \le n$. If there is $m_i = 1$, then $x > y$.		Use β to compute $\widetilde{m}_j = D(\hat{c}_j), 1 \le j \le n$. If there is no $\widetilde{m}_j = 1$, then $x > y$.

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: $S_x^0 = \{x_n x_{n-1} \cdots x_{i+1} 1 \,|\, x_i = 0, 1 \le i \le n\}$, $S_x^1 = \{x_n x_{n-1} \cdots x_{i+1} x_i \,|\, x_i = 1, 1 \le i \le n\}$. If $x \neq y$, then

$$|S_x^0 \cap S_y^1| = 1 \text{ or } 0.$$

Proof. If $a, b \in S_x^0 \cap S_y^1, a < b$, then a must be of the form

$$a = x_n \cdots x_{i+1} 1 = y_n \cdots y_{i+1} y_i \in \{0,1\}^{n+1-i}, \text{ and } x_i = 0, y_i = 1,$$

for some index i. On the one hand, by the definition of S_x^0, we know that b must be of the form

$$b = x_n \cdots x_{i+1}\ 0\ x_{i-1} \cdots x_{j+1}\ 1 \in \{0,1\}^{n+1-j}, x_j = 0$$

for some index j. On the other hand, by the definition of S_y^1, b must be of the form

$$b = y_n \cdots y_{i+1}\ 1\ y_{i-1} \cdots y_j \in \{0,1\}^{n+1-j}, y_j = 1.$$

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 $\hat{d}_1, \cdots, \hat{d}_n$, and denote the sequence obtained by Alice as $\hat{c}_1, \cdots, \hat{c}_n$. Alice and Bob exchange $\hat{d}_1, \cdots, \hat{d}_n$ and $\hat{c}_1, \cdots, \hat{c}_n$ using the oblivious transfer protocol. We here stress that the transferred $\hat{d}_1, \cdots, \hat{d}_n$ and $\hat{c}_1, \cdots, \hat{c}_n$ are *not recognizable.* Any party, say, Alice, can cheat the other party by transferring an arbitrary sequence $\bar{c}_1, \cdots, \bar{c}_n$. If Bob honestly transfer $\hat{d}_1, \cdots, \hat{d}_n$, 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

[1] I. Blake, V. Kolesnikov: Strong conditional oblivious transfer and computing on intervals. ASIACRYPT2004, LNCS, vol. 3329, pp. 515-529. Springer-Verlag (2004)

[2] I. Damgard, S. Meldgaard, and J. B. Nielsen. Perfectly secure oblivious RAM without random oracles. In 8th Theory of Cryptography Conference - TCC 2011, volume 6597 of LNCS, pages 144-163. Springer (2011)

[3] Even S., Goldreich O., Lempel A.: A randomized protocol for signing contracts. Commun. ACM 28(6), 637-647 (1985)

[4] M. Fischlin: A cost-effective pay-per-multiplication comparison method for millionaires. CT-RSA 2001, LNCS, vol. 2020, pp. 457-472. Springer-Verlag (2001)

[5] M. T. Goodrich, M. Mitzenmacher, O. Ohrimenko, and R. Tamassia. Privacy-preserving group data access via stateless oblivious RAM simulation. In 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 157-167. ACM-SIAM (2011)

[6] Y. Huang, D. Evans, J. Katz, and L. Malka. Faster secure two-party computation using garbled circuits. In 20th USENIX Security Symposium (2011)

[7] I. Ioannidis, A. Grama: An efficient protocol for Yao's millionaires' problem. In Proceedings of the 36th Hawaii Internatinal Conference on System Sciences 2003 (2003)

[8] E. Kushilevitz, S. Lu, and R. Ostrovsky. On the (in)security of hash-based oblivious RAM and a new balancing scheme. In 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 143-156. ACM-SIAM (2012)

[9] H. Lin, W. Tzeng: An efficient solution to the millionaires' problem based on homomorphic encryption. ACNS2005, LNCS, vol. pp. 456-466. Springer-Verlag, (2005)

[10] B. Pinkas and T. Reinman. Oblivious RAM revisited. In Advances in Cryptology - Crypto 2010, volume 6223 of LNCS, pages 502-519. Springer (2010)

[11] Rabin, M.: How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University (1981)

[12] A. Salomma, Public-key Cryptography, Springer-Verlag (1990)

[13] B. Schoenmakers, P. Tuyls: Pratical two-party computation based on the conditional gate. ASIACRYPT2004, LNCS, vol. 3329, pp. 119-136. Springer-Verlag (2004)

[14] E. Shi, T.-H. H. Chan, E. Stefanov, and M. Li. Oblivious RAM with o((log n)3) worst-case cost. In Advances in Cryptology - Asiacrypt 2011, volume 7073 of LNCS, pages 197-214. Springer (2011)

[15] A. Yao. Protocols for secure computations. In Proceedings of 23th Annual Symposium on Foundations of Computer Science (FOCS1982), pp. 160-164. IEEE (1982)