# An Efficient Protocol for the Problem of Secure Two-party Vector Dominance

Yingpeng Sang, Hong Shen, Zonghua Zhang
*School of Information Science*
*Japan Advanced Institute of Science and Technology*
*Asahidai, Tatsunokuchi, Ishikawa, Japan, 923-1211*
{yingpeng, shen, zonghua}@jaist.ac.jp

## Abstract

*The problem of secure two-party vector dominance requires the comparison of two vectors in an "all-or-nothing" way. In this paper we provide a solution to this problem based on the semi-honest model. It is reduced to the problem of privacy preserving prefix test, and an additive threshold homomorphic encryption is used to protect those privacies while computing the results of all of the prefix tests. Our solution has advantages of efficiency and security in comparison with other solutions.*

**Keywords** : *vector dominance, secure multiparty computation, homomorphic encryption.*

## 1 Introduction

Let $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_n)$ be two vectors. We say $A$ *dominates* $B$ (denoted by $A \succ B$), if $a_i > b_i$ for all $i = 1, ..., n$. If there is at least one $a_{i}^{'} \leq b_{i}^{'}$ $(1 \leq i^{'} \leq n)$, $A$ does not dominate $B$.

The problem of *Secure Two-party Vector Dominance* (STVD) was firstly defined in [1]: Alice (the owner of vector $A$) and Bob (the owner of vector $B$) want to know whether $A \succ B$ in an "all-or-nothing" way. "All-or-nothing" means that they should know whether $A \succ B$ or not while the following security requirements are met: 1) neither Alice nor Bob knows any element of the other's vector; 2)neither Alice nor Bob knows the relative ordering of any element pair $(a_i, b_i)$ (i.e., whether some $a_i \leq b_i$ or not). STVD can be looked as a multi-dimensional extension from Yao's millionaire problem [15], in which two millionaires want to know whose number of money is greater than the other's, without disclosing their numbers to each other.

STVD can be encountered in many applications [5], e.g., multi-commodity private bidding. In business-to-business bidding, a manufacturer may only want to deal with the supplier that can simultaneously satisfy the requirement for $n$ items because they have some coordinations in the production. This can be treated as a STVD problem: Alice want to buy $n$ items from Bob if she can securely know her bidding vector $A$ dominates Bob's price vector $B$ without disclosing anything other than whether they can have a deal.

In this paper we use 0-*encoding* to reduce STVD to a few cases of the *Prefix Testing* problem. The whole protocol of STVD is constructed on an additive homomorphic encryption scheme, Paillier's cryptosystem. Our protocol compares favorably in both efficiency and security with other solutions.

The remainder of this paper is organized as follows. Some related work is discussed in section 2. The preliminaries of our protocol are given in section 3. The building blocks are constructed in section 4, and our protocol is described in details in section 5. Comparisons of our protocol with other two solutions are given in section 6, and the paper is concluded in section 7.

## 2 Related Work

The problem of STVD is a special case of the general *Secure Multiparty Computation* (SMC) problem. Generally speaking, a SMC problem deals with computing any probabilistic function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than can be computed from that participant's input and output [10]. As Goldreich states in [9], the general secure multi-party computation problem is solvable in theory, but it's impractical to use the solutions derived by these general results for special cases of multi-party computation; special solutions should be developed for the problem of STVD for efficiency reasons.

There have been many protocols for the millionaire problem ([2],[7],[11],[12],[14],[15]). In [12] a very efficient protocol for this problem is proposed using a technique sim-

Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)
0-7695-2405-2/05 $20.00 © 2005 IEEE

ilar with the 0-encoding in this paper. However, it's not suitable to run their protocol $n$ rounds to check whether $A \succ B$, otherwise the ordering of every pair $(a_i, b_i)$ will be known by Alice and Bob. A more secure protocol should be developed.

Till now, there have been a few protocols for STVD. The protocol in [1] use a third party, and the third party is assumed not to collude with either Alice or Bob, otherwise the honest party will be cheated by the two colluders. To improve the security, the protocol in [5] doesn't employ the third party and is based on Yao's protocol for millionaire problem. In [5] Alice and Bob are assumed to be semi-honest. However, the complexity of Yao's protocol ([15]) depends on the range of the input, i.e., for inputs of $k$-bit numbers the complexity is $O(2^k)$. All of these protocols assumes that the inputs are non-negative integers. In this paper we keep on such an assumption.

## 3 Preliminaries

### 3.1 Semi-honest Model

In this paper we assume that Alice and Bob are semi-honest to each other. A formal definitions of semi-honest model can be found in [9]. A semi-honest party is assumed to follow the protocol exactly as what is prescribed by the protocol, except that it keeps a record of all its intermediate computations. Security in this model means that no player or coalition of players gains information which is not inherent in the output of the calculated function.

Semi-honest parties do constitute a model of independent interest ([9]). Because deviation from the specified program is difficult in many settings, general malicious behaviors, e.g., aborting the protocol or entering the protocol with an arbitrary input, may be infeasible for many users. However, it's easier to record the contents of some registers by the standard activities of the operating system, so some semi-honest behaviors may be feasible for the users: they may analyze their records of all intermediate computations so as to get any information other than the output.

### 3.2 Homomorphic Encryption

Our construction of STVD protocol is based on a Homomorphic Encryption (HE) scheme. A general definition of HE is given as follows [4]. Let $\varepsilon$ be a probabilistic encryption scheme. Let $M$ be the message space and $C$ the ciphertext space such that $M$ is a group under operation $\oplus$ and $C$ is a group under operation $\otimes$. $\varepsilon$ is a $(\oplus, \otimes)$-HE scheme if for any instance $E$ of the encryption scheme, given $c_1 = E_{r_1}(m_1)$ and $c_2 = E_{r_2}(m_2)$, there exists an $r$ such that

$$c_1 \otimes c_2 = E_{r_1}(m_1) \otimes E_{r_2}(m_2) = E_r(m_1 \oplus m_2)$$

$\varepsilon$ is *additive* when it's a $(+, \otimes)$ scheme, and *multiplicative* when it's a $(*, \otimes)$ scheme.

The HE scheme are also required to support secure $(2, 2)$-threshold decryption in our construction. The corresponding secret key is shared by both parties, and the decryption can only be performed by all parties acting together, but can't be performed by any single party.

There are a few cryptosystems that satisfy our requirements, and for efficiency we employ ElGamal encryption ([6]). We use a $(2, 2)$-threshold variant of ElGamal encryption ([4]) which is depicted as follows:

- Distributed Key Generation: Let $p$ and $q$ be large primes such that $p = 2q + 1$. $G_q$ denotes $\mathbb{Z}_p^*$'s unique multiplicative subgroup of order $q$, and $G_q$ has a generator of $g$. Alice's share of secret key is a random $s_A \in \mathbb{Z}_q$ and Bob's share of secret key is a random $s_B \in \mathbb{Z}_q$. Alice and Bob publish $g^{s_A}$ and $g^{s_B}$ respectively. The common public key is $(g, h, f)$ for which $h = g^{s_A + s_B}$, $f \in G_q$.
- Encryption: given a message $m \in \mathbb{Z}_q$, $E(m) = (x, y) = (g^\alpha, h^\alpha f^m)$ in which $\alpha$ is an random number in $\mathbb{Z}_q$.
- Decryption: given a ciphertext $c = (x, y)$, Alice and Bob publishes $x^{s_A}$ and $x^{s_B}$ respectively, $D(c) = y / x^{s_A + s_B}$.

In the above scheme, $D(c) = f^m$, but in this paper we only care about whether $m = 0$, i.e., whether $D(c) = 1$. The scheme has the following properties: 1) given two encryptions $E(m_1) = (x_1, y_1)$ and $E(m_2) = (x_2, y_2)$, we can efficiently get $E(m_1 + m_2) = (x_1 x_2, y_1 y_2)$; 2)given an encryption $E(m) = (x, y)$ and a number $a$, we can efficiently get $E(am) = (x^a, y^a)$.

## 4 Building Blocks

### 4.1 0-encoding

Suppose that Alice has a binary string $a_I ... a_2 a_1 \in \{0,1\}^I$ for $a$, Bob has a binary string $b_J ... b_2 b_1 \in \{0,1\}^J$ for $b$. $a, b \in \{0, ..., 2^{K+1} - 1\}$. $K$ is a security parameter for Alice and Bob so that $2^{K+1}$ won't leak any information about the range of their bit strings. We respectively add the prefix of '0..0' with length of $K - I$ and $K - J$ to $a$ and $b$ to have a $K$-bit $a$ and $b$.

If $a > b$, there must be a $i$ $(1 \le i \le K)$, which satisfies $a_K ... a_{i+1} = b_K ... b_{i+1}$, $a_i = 1$, and $b_i = 0$. If such a $b_i$

is substituted by 1, then $b_K...b_{i+1}1 = a_K...a_{i+1}a_i$. This is the main idea of 0-encoding on $b$, which is firstly shaped in [12]. In this section we describe it in a more general way and elicit more detailed properties: 1)0-encoding on $b$ has at most one prefix of $a$; 2)0-encoding on $b$ has only one prefix of $a$ iff $b < a$.

For a $n$-bit integer number $b$ whose binary string is $b_n b_{n-1}...b_1 \in \{0,1\}^n$, the *0-encoding* of $b$ is the set of $S_b^0$ such that

$$S_b^0 = \{1|b_n = 0\} \cup \{b_n b_{n-1}...b_{i+1}1 | b_i = 0, n-1 \geq i \geq 1\}$$

For example, given $b = (0101)_2$, $S_b^0 = \{1, 011\}$.

**Property 1:** *given two $K$-bit numbers $a$ and $b$, $S_b^0$ has a prefix of $a$, iff $a > b$.*

*Proof*: If $a > b$, there must be a $J$ for which $a_K...a_{J+1} = b_K...b_{J+1}$, $a_J = 1$ and $b_J = 0$, so $b_K...b_{J+1}1 \in S_b^0$ and $b_K...b_{J+1}1$ is a prefix of $a$. On the contrary, if $S_b^0$ has a prefix of $a$, e.g., $b_K...b_{J'}1 = a_K...a_{J'}a_{J'-1}$, then $b_K...b_{J'} = a_K...a_{J'}$, $b_{J'-1} = 0$, and $a_{J'-1} = 1$. So $a$ must be larger than $b$. □

**Property 2:** *given two $K$-bit numbers $a$ and $b$, $S_b^0$ has at most one prefix of $a$.*

*Proof*. From property 1 $S_b^0$ has a prefix of $a$ when $a > b$. Suppose that $S_b^0$ have two prefixes of $a$, such as:

$$a_K...a_{i+1}a_i = b_K...b_{i+1}1 \quad (1)$$

$$a_K...a_i a_{i-1}...a_{j-1}a_j = b_K...b_i b_{i-1}...b_{j-1}1 \quad (2)$$

From (1) we can have $a_i = 1$, $b_i = 0$; However, in (2) it's obvious that $a_i = b_i$, which contradicts $a_i \neq b_i$ in (1). Therefore, $S_b^0$ can't have two prefixes of $a$. It can also be induced that $S_b^0$ can't have more than two prefixes of $a$. □

**Property 3:** *given two $K$-bit numbers $a$ and $b$, $S_b^0$ hasn't a prefix of $a$, iff $a \leq b$.*

Property 3 can be easily deduced from property 1 and 2.

### 4.2 Privacy Preserving Prefix Test

By property 1, whether $S_{b_i}^0$ has a prefix of $a_i$ is a basic problem of STVD.

Suppose that the binary string of $a_i$ is $a_{iK}...a_{i1} \in \{0,1\}^K$, $b_i$ is $b_{iK}...b_{i1} \in \{0,1\}^K$, and $S_{b_i}^0 = \{b_i^j | 1 \leq j \leq K_i, K_i \in [1,K]\}$, $|b_i^j| = J_{ij}$ ($J_{ij} \in [1,K]$) and $b_i^j = b[i,j,J_{ij}]b[i,j,J_{ij}-1]...b[i,j,1] \in \{0,1\}^{J_{ij}}$. We also suppose that $a_i^{'}$ is the longest prefix of $a_i$ which ends up with '1', and $|a_i^{'}| = J_i$ ($J_i \in [1,K]$).

If Alice and Bob want to test whether $b_i^j$ is a prefix of $a_i$ while preserving their privacies on $a_i$ and $b_i^j$, they can follow the protocol of *Privacy Preserving Prefix Test (PPPT)*:

1. Randomly choosing $r[i,l,0]$ and $r[i,l,1]$ from $\mathbb{Z}_q$, Alice gets the following $2 \times K$ encryption matrix $Z_i$ based on $a_i$:

   $Z_i =$

   $$\begin{pmatrix} z[i,K,0] & ... & z[i,J_i,0] & z[i,J_i-1,0] & ... & z[i,1,0] \\ z[i,K,1] & ... & z[i,J_i,1] & z[i,J_i-1,1] & ... & z[i,1,1] \end{pmatrix}$$

   (a) if $K \geq l \geq J_i$ and $a_{il} = 0$, $z[i,l,0] = E(0)$, $z[i,l,1] = E(r[i,l,1])$.

   (b) if $K \geq l \geq J_i$ and $a_{il} = 1$, $z[i,l,0] = E(r[i,l,0])$, $z[i,l,1] = E(0)$.

   (c) if $(J_i - 1) \geq l \geq 1$, $z[i,l,0] = E(r[i,l,0])$, $z[1,l,1] = E(r[i,l,1])$.

   Alice sends $Z_i$ to Bob.

2. Based on $b_i^j$ and $Z_i$, Bob gets $y_{ij}$ as following:

   $$y_{ij} = z[i,K,b[i,j,J_{ij}]] \cdot z[i,K-1,b[i,j,J_{ij}-1]] \cdot \cdots \cdot z[i,K-J_{ij}+1,b[i,j,1]] \cdot E(0)$$

   Bob sends $y_{ij}$ to Alice.

3. Alice decrypts $y_{ij}$. If $D(y_{ij}) = 1$, then $b_i^j$ is a prefix of $a_i$; otherwise it's not a prefix of $a_i$.

**Theorem 1** *With overwhelming probability, PPPT is complete (if $b_i^j$ is a prefix of $a_i$, $D(y_{ij}) = 1$) and sound (if $b_i^j$ isn't a prefix of $a_i$, $D(y_{ij}) \neq 1$).*

*Proof:* If $b_i^j$ is a prefix of $a_i$, then $z[i,K,b[i,j,J_{ij}]] = z[i,K-1,b[i,j,J_{ij}-1]] = ... = z[i,K-J_{ij}+1,b[i,j,1]] = E(0)$, and $D(y_{ij}) = 1$ due to the homomorphic encryption.

If $b_i^j$ isn't a prefix of $a_i$, $y_{ij} = E(R_{ij})$. $R_{ij}$ is a random number in $\mathbb{Z}_q$, and with negligible probability $R_{ij}$ is 0 or the order of $f$ . Therefore, with overwhelming probability $D(y_{ij}) \neq 1$. □

## 5 The Protocol

From property 1 and 2, $A \succ B$ if and only if every $S_{b_i}^0$ ($1 \leq i \leq n$) has a unique prefix of the corresponding $a_i$. Therefore the main idea of our protocol for STVD is: 1)on every $b_i^j$ in $S_{b_i}^0$ for $1 \leq j \leq K_i$, PPPT is used to get $y_{ij} = E(R_{ij})$, and $y_i = E(\prod_{j=1}^{K_i} R_{ij})$. 2)$Y = \prod_{i=1}^{n} y_i = E(\sum_{i=1}^{n} \prod_{j=1}^{K_i} R_{ij})$ is got. If there is a $R_{ij} = 0$ for all $S_{b_i}^0$ ($1 \leq i \leq n$), then $D(Y) = 1$.

**Protocol** : The protocol for Secure Two-party Vector Dominance problem

**Input** : Two players, Alice and Bob, don't trust in each other. Alice has a private vector $A = (a_1, a_2, ..., a_n)$ and Bob has a private vector $B = (b_1, b_2, ..., b_n)$.

**Output** : The two players know whether $A \succ B$, without disclosing their vectors to each other or the ordering of any pair $(a_i, b_i)$ for $1 \leq i \leq n$.

**Step 1** Alice and Bob generate the public key and secret key following with the threshold cryptosystem in section 3.2. Every party holds the public key and it's own share of the secret key.

**Step 2** Bob generates his $S^0_{b_i}$ for $i = 1, ..., n$, as described in section 4.1.

**Step 3** For $j = 1, 2, ..., K$,

1. based on $a_1$, Alice gets an $2 \times K$ encryption matrix $Z_{1,j}$ as following:

$$Z_{1,j} = \begin{pmatrix} z[1,K,0] & ... & z[1,J_1,0] \\ z[1,K,1] & ... & z[1,J_1,1] \\ z[1,J_1-1,0] & ... & z[1,1,0] \\ z[1,J_1-1,1] & ... & z[1,1,1] \end{pmatrix}$$

   The pairs of $(z[1,l,0], z[1,l,1])$ are valued as following, with $r^0_{il}$ and $r^1_{il}$ randomly chosen from $\mathbb{Z}_q$ for every iteration of $j$:

   1.1 if $j = 1$,
   - i. if $K \geq l \geq J_1$ and $a_{1l} = 0$, $z[1,l,0] = E(0)$, $z[1,l,1] = E(r^1_{1l})$.
   - ii. if $K \geq l \geq J_1$ and $a_{1l} = 1$, $z[1,l,0] = E(r^0_{1l})$, $z[1,l,1] = E(0)$.
   - iii. if $(J_1 - 1) \geq l \geq 1$, $z[1,l,0] = E(r^0_{1l})$, $z[1,l,1] = E(r^1_{1l})$.

   1.2 if $j = 2, ..., K$, with $y_{1,j-1}$ from Bob,
   - i. if $K \geq l \geq J_1$ and $a_{1l} = 0$, $z[1,l,0] = (y_{1,j-1})^0 \cdot E(0)$, $z[1,l,1] = (y_{1,j-1})^{r^1_{1l}}$.
   - ii. if $K \geq l \geq J_1$ and $a_{1l} = 1$, $z[1,l,0] = (y_{1,j-1})^{r^0_{1l}}$, $z[1,l,1] = (y_{1,j-1})^0 \cdot E(0)$.
   - iii. if $(J_1 - 1) \geq l \geq 1$, $z[1,l,0] = (y_{1,j-1})^{r^0_{1l}}$, $z[1,l,1] = (y_{1,j-1})^{r^1_{1l}}$.

2. Alice sends $Z_{1,j}$ to Bob.

3. In every pair of $(z[1,l,0], z[1,l,1])$ for $l = 1, ..., J_{1j}$, Bob selects $z[1,l,b]$ if $b[1,j,l] = b$ $(b \in \{0,1\})$, and gets $y_{1,j}$ as following:

   3.1 if $1 \leq j \leq K_i$,

$$y_{1,j} = z[1,K,b[1,1,J_{1j}]] \cdot z[1,K-1,b[1,1,J_{1j}-1]] \cdots z[1,K-J_{1j}+1,b[1,1,1]] \cdot E(0)$$

   3.2 if $(K_i + 1) \leq j \leq K$, $y_{1,j} = (y_{1,j-1})^{r_j}$, $r_j$ is randomly chosen from $\mathbb{Z}_n$.

4. Bob sends $y_{1,j}$ to Alice.

**Step 4** For $i = 2, ..., n$, based on $a_i$ and $b_{i,j}$ $(j = 1, ..., K)$, Alice and Bob repeat step 3.

**Step 5** Alice and Bob get $Y = y_{1,K} \cdot y_{2,K} \cdots y_{n,K}$.

**Step 6** Alice and Bob combine their shares of the secret key, and then decrypt $Y$. If $D(Y) = 1$, $A \succ B$; otherwise $A$ doesn't dominate $B$.

**Theorem 2** *With overwhelming probability, the protocol for STVD is complete (if $A \succ B$, $D(Y) = 1$) and sound (if A doesn't dominate B, $D(Y) \neq 1$).*

*Proof:* If $A \succ B$, every $S^0_{b_i}$ has a prefix of $a_i$. In step 3, actually every $b^j_1$ could have selected elements from $Z_{1,1}$ on every $2 \leq j \leq K_i$ without adapting $Z_{1,1}$ to $Z_{1,j}$, and then got $y'_{1,j} = E(R_{1,j})$. If $S^0_{b_1}$ has a $b^{j'}_1$ that is the prefix of $a_1$, then $R_{1,j'} = 0$. However, to keep the privacy of $B$, $Z_{1,1}$ is changed into $Z_{1,j}$ for $2 \leq j \leq K_i$, and $K$ rounds are used to get the encrypted multiplication of $R_{1,j}$: $y_{1,K} = E(R_{1,1} \cdot R_{1,2} \cdots R_{1,K})$. It's still be true that $y_{1,K} = E(0)$ if $S^0_{b_1}$ has a prefix of $a_1$. In step 5, $Y = E(\sum_{i=1}^{n} \prod_{j=1}^{K} R_{i,j})$. If every $S^0_{b_i}$ has a prefix of $a_i$, $Y = E(0)$ and $D(Y) = 1$.

If $A$ doesn't dominate $B$, i.e., there is at least one $S^0_{b_{i'}}$ that hasn't a prefix of $a_{i'}$. From theorem 1, with overwhelming probability $E(R_{i',j}) \neq E(0)$ for $j = 1, ..., K$, and $y_{i',K} = E(\prod_{j=1}^{K} R_{i',j}) \neq E(0)$. Therefore, in step 5, overwhelmingly $Y \neq E(0)$ and $D(Y) \neq 1$.

**Theorem 3** *The protocol for STVD is secure for Alice and Bob when they are semi-honest parties.*

*Proof:*The security of STVD for semi-honest parties is based on the security of ElGamal's cryptosystem. At every step, what Alice and Bob get are all encryptions, and in the final step only $Y$ is decrypted. If $Y \neq E(0)$, $D(Y)$ is a random number in $\mathbb{Z}_q$, so both of two parties can learn no more information than $A$ doesn't dominate $B$.

## 6 Comparisons with other solutions

Suppose that $K$ is the security parameter, $n$ is the order of $A$ and $B$, the communication complexity of our protocol is $O(nK^2)$, and the computation complexity is $O(nK^2)$ modular multiplications for Alice and $O(nK)$ for Bob.

The solution in [5] has a $O(nK)$ communication complexity and $O(nK)$ computation complexity, but a third party is used in this protocol. To improve security, the solution in [1] doesn't use a third party but has a $O(n2^k)$ communication complexity and $O(n2^k)$ computation complexity. Our protocol achieves much lower complexity while keeping the stronger security.

## 7 Conclusions

We have provided an efficient and secure protocol for the problem of secure two-party vector dominance. The problem is reduced to the problem of privacy preserving prefix test. Homomorphic encryption is used to protect those privacies while computing the result of all of the prefix tests. We leave as an open problem to achieve high efficiency in the malicious model for STVD.

## Acknowledgment

This research is conducted as a program for the "Fostering Talent in Emergent Research Fields" in Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

[1] M.J. Atallah and W. Du. Secure Multi-party Computational Geometry. *Proceedings of the Seventh International Workshop on Algorithms and Data Structures*, Pages 165-179, Providence, Rhode Island, August, 2001.

[2] C. Cachin. Efficient Private Bidding and Auctions with an Oblivious Third Party. *Proceedings of the 6th ACM conference on Computer and communications security*, pages 120-127, Singapore, November, 1999.

[3] D. Chaum and T. P. Pedersen. Wallet Databases with Observers. *Advances in Cryptolog-CRYPTO'92*, volume 740 of LNCS, pages 89-105. Springer-Verlag, Berlin, 1993.

[4] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and Optimally Efficient Multi-authority Election Scheme. *Advances in Cryptology - Proceedings of the 14th Eurocrypt Conference*, volume 1233 of LNCS, pages 103-118. Springer, 1997.

[5] W. Du. A Study of Several Specific Secure Two-party Computation Problems, Ph.D. Thesis. Purdue University. 2000. Available From: htttp://www.cis.edu/ wedu/ Research/publication.html.

[6] T. ElGamal. A Public-key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT-31(4): 469-472, 1985.

[7] M. Fischlin. A Cost-effective Pay-per-multiplication Comparison Method for Millionaires. *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographers Track at RSA*, volume 2020 of LNCS, pages 457-472. Springer-Verlag, 2001.

[8] M. Freedman, K. Nissim and B. Pinkas. Efficient Private Matching and Set Intersection, *Advances in Cryptology- Proceedings of EUROCRYPT 2004*, volume 3027 of LNCS, pages 1-19. Springer-Verlag, 2004.

[9] O. Goldreich. Foundations of Cryptography: Volume 2. Cambridge University Press, 2001.

[10] S. Goldwasser. Multi-party Computations: Past and Present. *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, Pages 1-6, Santa Barbara, CA USA, August, 1997.

[11] I. Ioannidis and A. Grama. An Efficient Protocol for Yaos Millionaires Problem. *Proceedings of the 36th Hawaii Internatinal Conference on System Sciences 2003*, vol. 07, no. 7, page 205a, 2003.

[12] H. Y. Lin and W. G. Tzeng. An Efficient Solution to The Millionaires' Problem Based on Homomorphic Encryption. *Applied Cryptography and Network Security 2005 (ACNS 2005)*. Volume 3531 of LNCS. pages 456-466, 2005.

[13] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *In Eurocrypt '99*, volume 1592 of LNCS, pages 223-238, 1999.

[14] K. Peng, C. Boyd, E. Dawson and B. Lee. An Efficient and Verifiable Solution to the Millionaire Problem. *Proceedings of the 7th International Conference on Information Security and Cryptology (ICISC2004)*, volume 3506 of LNCS, pages 51-66.

[15] A.C. Yao, Protocols for Secure Computations. *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160-164, November 1982.