Secrecy of Two-Party Secure Computation*

Yi-Ting Chiang, Da-Wei Wang**, Churn-Jung Liau, and Tsan-sheng Hsu***

Institute of Information Science Academia Sinica, Taipei, 115, Taiwan
{ytc, wdw, liaucj, tshsu}@iis.sinica.edu.tw

Abstract. Privacy protection has become one of the most important issues in the information era. Thus, many protocols have been developed to achieve the goal of cooperatively accomplishing a computational task without revealing the participants' private data. Practical protocols, however, do not guarantee perfect privacy protection, as some degree of privacy leakage is allowed during the computation process for the sake of efficient resource consumption, e.g., the number of random bits required and the computation time. Although there are metrics for measuring the amount of resource consumption, as far as we know, there are no effective metrics that measure the degree of privacy leakage. Without such metrics, however, it is difficult to compare protocols fairly. In this paper, we propose a framework based on linear algebra and information theory to measure the amount of privacy leakage in protocols. This framework can be used to analyze protocols that satisfy certain algebraic properties. We use it to analyze three two-party scalar product protocols. The framework might also be extendable to the analysis of other protocols.

Keywords: Privacy Analysis, Private Computation, Scalar Product.

1 Introduction

Privacy protection is one of the most pressing issues in the information era. The massive databases spread over the Internet are gold mines for some and, at the same time, one of the greatest threats to privacy for others. How to cooperatively accomplish a computational task without revealing participants' private input has therefore gained a lot of attention and the development of efficient solutions is now an active research area. In theory [11,7], it is possible to securely compute almost any function without revealing anything, except the output. Unfortunately, the theoretical results are not readily applicable to real applications due to their high computational complexity.

Most theoretical approaches adopt a computationally indistinguishable view of secrecy and try to find provable secure solutions, but such a definition leaves

* Supported in part by Taiwan Information Security Center.

** Corresponding Author: Joint-appointment faculty member of National Yang Ming University, Taiwan. Supported in part by NSC (Taiwan) Grant 93-2213-E-001-031.

*** Supported in part by NSC (Taiwan) Grants 92-2213-E-001-005 and 93-2213-E-001-001.

S. Jajodia and D. Wijesekera (Eds.): Data and Applications Security 2005, LNCS 3654, pp. 114–123, 2005.

little room to quantify secrecy. Meanwhile, in application oriented studies, researchers usually take an intuitive approach to the definition of secrecy and try to prove the secrecy of protocols by refuting possible attacks. However, being intuitive, this approach cannot actually prove the security of protocols per se. It can only be argued that refuting possible attacks preserves some security. There is a gap between the theoretical and intuitive approaches in terms of provable secrecy. Although, privacy is a basic human right, it is not the only one. When multi-party private computation is applied to the public sector, sometimes privacy must be compromised to accommodate other important social values. It can also be applied to the private sector, such as in a business setting. For example, two (or more) companies might want to compute a function cooperatively; however, neither of them wants to share their private information. In both public sector and private sector applications, it would be beneficial to be able to quantify secrecy so that some tradeoff, for example, between secrecy and computational efficiency, could be made. In [5], similar arguments are presented about ideal secrecy and acceptable secrecy. In this paper, we propose an information theoretical framework toward a quantifiable definition of secrecy for multi-party private computation.

The remainder of this paper is organized as follows. We give a short review of related works in Section 2. In Section 3, we present our formal framework. In Section 4, we analyze several scalar product protocols to demonstrate our model and summarize the results. Finally, in Section 5, we present our conclusions and a short discussion about possible extensions of our model. We also indicate the direction of future work.

2 Related Work

Secure two-party computation was first studied by Yao [11] and extended to the multi-party case by Goldreich et al [7]. Through a sequence of effort, a satisfactory definitional treatment was developed and precise proofs for security were provided . A full description of these developments can be found in [6]. The general construction approach is as follow. To securely compute a function, it is first converted to a combinatorial circuit. For each gate in the circuit, all parties run a protocol to compute the result of that gate. Both the input and the output of the gate are shared randomly and the final output is also shared randomly among all parties, after which each party can exchange its share of the information to compute the final result. Although, this general construction approach is impressive, it implies that both the size of the circuit and the number of parties involved dominate the size, i.e., complexity, of the protocol. Note that the size of the circuit is related to the size of the input. Therefore, the approach is not a feasible solution for a real world problem with a large input and/or a large number of parties [9].

The high cost of the general approach for large problems has motivated researchers to look for efficient solutions for specific functions and many protocols have already been developed to solve specific problems. There are specific pro-

tocols for general computation primitives, such as, scalar products [1,10], set union and set intersection cardinality [8], and private permutation [2]. In addition, there are protocols for specific application domains, for example, data mining, computational geometry, statistical analysis, etc. An excellent survey of secure multi-party computation problems can be found in [3].

Almost all the approaches mentioned above are based on the notion of ideal secrecy, as indicated in [5]. In that paper the authors ask if it would be possible to lower the security requirement from an ideal level to an acceptable level so that an efficient protocol could be developed. We extend their work by quantifying the security level within an information theoretical framework.

3 Framework

In multi-party private computation, n players cooperate to compute a function, and each player holds some private input that is part of the parameters for computing the function. The goal is to compute the function and maintain the secrecy of each party's private input. Given a protocol, P, we use X_i^P to denote the private input of party i, and msg_i^P to denote the message received by party i. We use information theory to model the amount of information revealed after running P. Before running P, each party has no information about other parties' private input. However, after running P, each party may know something about some of the other parties' private inputs because of new information gathered during the execution of P. Let $H_i^P = H(X_i^P)$ denote the entropy of random variable X_i^P, and $H_{ij}^P = H(X_i^P|msg_j^P)$ denote the entropy of random variable X_i^P given msg_j^P. The conditional entropy corresponds to the intuitive idea of the amount of information (uncertainty) of X_i^P from party j's perspective after receiving msg_j^P.

We define the degree of secrecy of protocol P as $\min_{i,j}(H_{ij}^P/H_i^P)$, or $\min_{i,j}(H_{ij}^P)$; and call the former *relative secrecy* and the latter *absolute secrecy*. When comparing different protocols, we believe that relative secrecy is a better notion, since it is normalized to a number between zero and one, where one indicates perfect secrecy, and zero means no secrecy at all. However, for some specific applications, where the number of players and the types of private input are fixed, absolute secrecy gives the user a direct measurement of the degree of uncertainty that each private input contains after executing the protocol. Obviously we assume the existence of private communication channels between any two parties. To model the case of a broadcast channel, we simply replace msg_i^P with msg^P, where msg^P denotes the complete record of messages broadcast during the execution of the protocol. It is worth mentioning that our model can be extended to model situations such as parties forming a coalition, where there is asymmetry among data elements in the private inputs and among the parties. We do not try to describe such a general model here, as the extension might detract from our main points. In a later work, we hope to extend our model to a multi-party setting.

4 Analysis of the Protocols

4.1 Preliminaries

In this paper, we analyze the degree of secrecy of three two-party scalar product protocols, each of which has two players, Alice and Bob, who have private input X_A and X_B respectively. The private input of each player is an n dimensional vector. After running the protocol, Alice and Bob receive the numbers u and v respectively, such that $u + v$ is the inner product of X_A and X_B, i.e., $X_A \cdot X_B$. Let $*$ be the matrix product operator, and X_B^T be the transpose of X_B. Then, $u + v = X_A \cdot X_B = X_A * X_B^T$. Hereafter, we assume that $X_A, X_B \in GF(p)^n$, where $GF(p)$ is a Galois field of order p, and p is a prime number. We also assume that both parties are semi-honest, i.e., they both follow the protocol and do not deliberately deviate from it to get more information. Instead, they only deduce information from messages they receive.

We first list some facts from information theory.

Fact 1

1. $H(X|msg) = H(X, R|msg) - H(R|X, msg)$.
2. *If R is a function of X and msg, then* $H(R|X, msg) = 0$ *and* $H(X|msg) = H(X, R|msg)$.
3. *If* $H(R|X, msg) \neq 0$ *and* $H(X|R, msg) = 0$, *then* $H(X|msg) = H(R|msg) - H(R|X, msg)$.

Let V and C be two random sources. If it is known that some functional dependency exists between V and C, then knowing information about C reveals information about V. That is, the entropy of V is reduced. For the case where $V, C \in GF(p)^n$ and A is a matrix, we get the following:

Proposition 1. *Let $V, C \in GF(p)^n$ be two vectors with all elements uniformly randomly selected from $GF(p)$, and let A be an $m \times n$ matrix with all its elements in $GF(p)$. If there exists a functional dependency $A * V = C$ and $rank(A) = k$, then $H(V|C) = (n - k) \log p$.*

Proof: By $A * V = C$, let W_1 and W_2 be two vector spaces with ordered bases α and β such that there is a linear transformation T: $W_1 \rightarrow W_2$. $[T]_\beta^\alpha = A$. Since $rank(A) = k$, if C is known, we can find a vector space $U \subseteq W_1$ such that the dimension of U is $n - k$ and $V \in U$. Let $s = (s_1, \ldots, s_{n-k})$ be an ordered basis of U. Then V can be expressed in the form:

$$V = a_1 s_1 + a_2 s_2 + \cdots + a_{n-k} s_{n-k}.$$

Thus, $H(V|C) = H(a_1, \ldots, a_{n-k}) = (n - k) \log p$. ■

The following lemma can be derived directly from the above proposition.

Lemma 1. *Let $A * V = C$ be a linear system of equations in $GF(p)$. If there are k linear independent equations in $A * V = C$, that is, $rank(A) = k$, and n unknowns in V, then $H(V|C) = (n - k) \log p$.*

We now describe and analyze three scalar product protocols. In our analysis, let $\mathbf{I}_i$ be an $i \times i$ identity matrix, and $\mathbf{0}_{i \times j}$ be an $i \times j$ zero matrix.

4.2 Analysis of Protocol 1

The protocol is as follows. First Alice and Bob agree to an $n*n$ invertible matrix, M, and a positive integer, k, that is not larger than n.

Scalar Product Protocol 1 *[5]*

Alice		*Bob*
1. Compute $X'_A = X_A * M$.		*Compute* $X'_B = (M^{-1} * X_B^T)^T$.
Let $X'_A = [x_{A_1}, \ldots, x_{A_n}]$,		*Let* $X'_B = [x_{B_1}, \ldots, x_{B_n}]$,
$\bar{X}_A = [x_{A_1}, \ldots, x_{A_k}]$,		$\bar{X}_B = [x_{B_1}, \ldots, x_{B_k}]$,
$\underline{X}_A$=$[x_{A_{k+1}}, \ldots, x_{A_n}]$		$\underline{X}_B = [x_{B_{k+1}}, \ldots, x_{B_n}]$
2.	*Alice* $\xrightarrow{\bar{X}_A}$ *Bob*	
	Alice $\xleftarrow{\underline{X}_B}$ *Bob*	
3. $u = \underline{X}_A * \underline{X}_B^T$		$v = \bar{X}_A * \bar{X}_B^T$

Let U be a matrix whose column vectors are the leftmost k column vectors of matrix M, and let V be a matrix whose row vectors are the last $n-k$ row vectors of matrix M^{-1}. We organize messages received by Alice and Bob in a matrix form and use Lemma 1 to derive the conditional information of each private input after the other party receives the messages sent during the protocol.

- Alice receives the message $msg_A = \{\underline{X}_B\} = \{V * X_B^T\}$. Thus, $V * X_B^T = \underline{X}_B$ and $rank(V) = n - k$. By Lemma 1, $H(X_B|msg_A) = k \log p$.
- Similarly, Bob receives the message $msg_B = \{\bar{X}_A\} = \{U * X_A\}$. Hence, $U * X_A = \bar{X}_A$ and $rank(U) = k$. By lemma 1, $H(X_A|msg_B) = (n-k) \log p$.

Based on the above discussion, we have the following lemma.

Lemma 2. *In Protocol 1, the degree of secrecy for Alice is* $\frac{H(X_A|msg_B)}{H(X_A)} = \frac{(n-k)\log p}{n \log p} = (n-k)/n$, *and for Bob is* $\frac{H(X_B|msg_A)}{H(X_B)} = \frac{k \log p}{n \log p} = k/n$. *The degree of secrecy for Protocol 1 is* $\min(\frac{H(X_A|msg_B)}{H(X_A)}, \frac{H(X_B|msg_A)}{H(X_B)}) = \min(k, n-k)/n \leq \frac{1}{2}$.

Remarks: In [5], it is mentioned, but not formally explained, that M should be invertible and k should be selected as $k = \lceil n/2 \rceil$. From our analysis, we know that selecting M to be invertible and $k = \lceil n/2 \rceil$ maximizes the degree of secrecy. It is also mentioned in [5] the selection of M should avoid the case where $\bar{X}_A = [x_{A_1}, \ldots, x_{A_k}]$; for example, that the selection of $M = \mathbf{I}_n$ is one of the bad cases. However, in our framework, picking $M = \mathbf{I}_n$ and picking M to be any invertible matrix are identical in terms of the degree of secrecy. Institutively, the advice mentioned above indicates that, the case where an individual value is fully revealed is definitely more serious than the cases where individual values are partially revealed, even though the total information remains are the same. The conflict will be resolved when our model is extended to consider asymmetry among the data elements of private inputs.

4.3 Analysis of Protocol 2

This protocol assumes the existence of a semi-honest party, C. In other words, C does not collude with Alice or Bob. First C generates two $1 \times n$ random matrices, R_a and R_b; and then randomly picks two integers, r_a and r_b, such that $r_a + r_b = R_a * R_b^T$. C sends R_a and r_a to Alice, and R_b and r_b to Bob.

Scalar Product Protocol 2 *[4,5]*

Alice		*Bob*
1. $X'_A = X_A + R_a$		$X'_B = X_B + R_b$
2.	*Alice* $\xrightarrow{X'_A}$ *Bob*	
	Alice $\xleftarrow{X'_B}$ *Bob*	
3.		*Bob generates a random value* v*, and computes* $s = X'_A * X_B^T + r_b - v$
4.	*Alice* $\xleftarrow{s}$ *Bob*	
5. $u = s - (R_a * X_B'^T) + r_a$		

Because, in this protocol, the commodity party C generates random variables without receiving any message, C gets no information about the private inputs of Alice and Bob.

Alice receives the message $msg_A = \{X'_B, r_a, s\}$ in Protocol 2, where

- $X'_B = \mathbf{I}_n * X_B^T + \mathbf{I}_n * R_b^T + 0 \cdot r_b + 0 \cdot v$,
- $r_a = \mathbf{0}_{1\times n} * X_B^T + R_a * R_b^T - 1 \cdot r_b + 0 \cdot v$, and
- $s = X'_A * X_B^T + \mathbf{0}_{1\times n} * R_b^T + 1 \cdot r_b - 1 \cdot v$.

Since $H(R_b|X_B, msg_A)$, $H(r_b|X_B, msg_A)$, and $H(v|X_B, msg_A)$ are all 0, we have $H(X_B|msg_A)$=$H(X_B, R_b, r_b, v|msg_A)$. Let $A_1 = \begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n & 0 & 0 \\ \mathbf{0}_{1\times n} & R_a & -1 & 0 \\ X'_A & \mathbf{0}_{1\times n} & 1 & -1 \end{bmatrix}$, $Z_1 = \begin{bmatrix} X_B^T \\ R_b^T \\ r_b \\ v \end{bmatrix}$, and $C_1 = \begin{bmatrix} X_B'^T \\ r_a \\ s \end{bmatrix}$.

Note that $rank(A_1) = n + 2$, $A_1 * Z_1 = C_1$, and C_1 is essentially msg_A. $H(X_B|msg_A) = n \log p$ by Lemma 1.

Bob gets the message $msg_B = \{r_b, X'_A\}$ in Protocol 2, where

- $\mathbf{I}_n * X_A^T + \mathbf{I}_n * R_a^T + 0 \cdot r_a = X_A'^T$, and
- $\mathbf{0}_{1\times n} * X_A^T + R_b * R_a^T - 1 \cdot r_a = r_b$.

Let $A_2 = \begin{bmatrix} \mathbf{I}_n & \mathbf{I}_n & 0 \\ \mathbf{0}_{1\times n} & R_b & -1 \end{bmatrix}$, $Z_2 = \begin{bmatrix} X_A^T \\ R_a^T \\ r_a \end{bmatrix}$, and $C_2 = \begin{bmatrix} X_A'^T \\ r_b \end{bmatrix}$.

It is easy to verify that $H(R_a|X_A, msg_B) = 0$, $H(r_a|X_A, msg_B) = 0$, and $rank(A) = n + 1$. Thus, $H(X_A|msg_B) = H(X_A, R_a, r_b|msg_B)$. We know $A_2 *$

$Z_2 = C_2$, and C_2 is essentially msg_B. By Lemma 1, $H(X_A|msg_B) = ((2n+1) - (n+1)) \log p = n \log p$.

Based on the above discussion, we have the following lemma.

Lemma 3. *The degree of secrecy for Protocol 2 is* $\min(\frac{H(X_A|msg_B)}{H(X_A)}, \frac{H(X_B|msg_A)}{H(X_B)})$*=1.*

4.4 Analysis of Protocol 3

This protocol assumes M is a public $n \times n$ matrix, m is a publicly known constant that is at most n, and $rank(M) = k$. Without loss of generality, we assume that n can be evenly divided by m, and $q = n/m$.

Scalar Product Protocol 3 *[10]*

Alice		*Bob*
1. Generate a $1 \times n$ random matrix R. Let D be an $m \times n$ matrix whose elements are d_{ij}, where $d_{ij} = \begin{cases} 1, if\ j \in [(i-1) \cdot q + 1, i \cdot q] \\ 0, otherwise \end{cases}$ Define $X'_A = (X_A^T + M * R^T)^T$ and $Q = D * R^T$.		
2.	*Alice* $\xrightarrow{Q, X'_A}$ *Bob*	
3.		*Let $s = X'_A * X_B^T$, and generate a $1 \times m$ random matrix $R' = [r'_1, \ldots, r'_m]$. Let $W = [w_1, \ldots, w_n]$ be a $1 \times n$ matrix, where $w_{(i-1)\times q+j} = r'_i$, $\forall i \in [1, m]$ and $\forall j \in [1, q]$. Let $X'_B = X_B * M + W$.*
4.	*Alice* $\xleftarrow{X'_B, s}$ *Bob*	
5. Note that $s = X'_A * X_B^T$ $= X'_A * X_B^T + R * W^T - R * W^T$ $= X_A * X_B^T + R * X_B'^T - R * W^T$. Since Alice knows X'_B, she can get $u = X_A * X_B - R * W^T$.		*Bob can compute $v = R' * Q$. Notes that $R \cdot W^T = R' * Q$.*

Alice receives the message $msg_A = \{X_B'^T, s\}$, where

- $X_B'^T = M^T * X_B^T + \mathbf{I}_n * W^T$, and
- $s = X'_A * X_B^T + \mathbf{0}_{1\times n} * W^T$.

Note that there are only m unknowns, $r'_1, \ldots, r'_m$, in W. Let $A = \begin{bmatrix} M^T & \mathbf{I}_n \\ X'_A & \mathbf{0}_{1\times n} \end{bmatrix}$, $Z = \begin{bmatrix} X_B^T \\ W^T \end{bmatrix}$, and $C = \begin{bmatrix} X_B'^T \\ s \end{bmatrix}$.

We know that $H(W|X_B, msg_A) = 0$, $rank(A) = n+1$, $A * Z = C$, and C is essentially msg_A. By Lemma 1, $H(X_B|msg_A) = (n+m-(n+1))\log p = (m-1)\log p$.

Bob receives the message $msg_B = \{X'_A, Q\}$ from Alice in Protocol 3, where

- $X_A'^T = \mathbf{I}_n * X_A^T + M * R^T$, and
- $Q = \mathbf{0}_{m\times n} * X_A^T + D * R^T$.

In Bob's case, $H(R|X_A, msg_B)$ may not be 0 if $rank(M) = k \neq n$. On the other hand, $H(X_A|R, msg_B) = 0$, even if k is not equal to n. So we have $H(X_A|msg_B) = H(R|msg_B) - H(R|X_A, msg_B)$.

We first compute $H(X_A, R|msg_B)$. Let $A_1 = \begin{bmatrix} \mathbf{I}_n & M \\ \mathbf{0}_{m\times n} & D \end{bmatrix}$, $Z_1 = \begin{bmatrix} X_A^T \\ R^T \end{bmatrix}$, and $C_1 = \begin{bmatrix} X_A'^T \\ Q \end{bmatrix}$.

It is clear that $rank(A) = n+m$, $A_1 * Z_1 = C_1$, and C_1 is essentially msg_B. By Lemma 1, $H(R|msg_B) = H(X_A, R|msg_B) = (2n-(n+m))\log p = (n-m)\log p$.

To compute $H(R|X_A, msg_B)$, X_A can be treated as a constant vector. Therefore, let $A_2 = \begin{bmatrix} M \\ D \end{bmatrix}$, $Z_2 = \begin{bmatrix} R^T \end{bmatrix}$, and $C_2 = \begin{bmatrix} X_A'^T - X_A^T \\ Q \end{bmatrix} = \begin{bmatrix} M * R \\ Q \end{bmatrix}$. From $A_1 * Z_1 = C_1$, we can derive $A_2 * Z_2 = C_2$.

Let $rank(A_2) = e$. From Lemma 1, $H(R|X_A, msg_B) = (n-e)\log p$. As a result, $H(X_A|msg_B) = H(R|msg_B) - H(R|X_A, msg_B) = ((n-m)-(n-e))\log p = (e-m)\log p$.

Note that $e \leq n$, m is an integer, and $\min(m-1, e-m) \leq \min(m-1, n-m) \leq (n-2)/2$.

Based on the above discussion, we have the following lemma.

Lemma 4. *The degree of secrecy for Protocol 3 is:*

$$\min\left(\frac{H(X_A|msg_B)}{H(X_A)}, \frac{H(X_B|msg_A)}{H(X_B)}\right) = \min\left(\frac{e-m}{n}, \frac{m-1}{n}\right) \leq \frac{1}{2} - \frac{1}{n} < \frac{1}{2}.$$

Remarks: In our analysis, Protocol 3 achieves its maximum level of secrecy when $m = \frac{n+1}{2}$ and $rank(A_2) = n$ However, we require that $m = n/q$ for some integer q, and m to be an integer. When n is even and $m = n/2$, the protocol achieves its maximum level of secrecy. This provides a guideline for choosing M and m.

5 Conclusion and Future Works

In this paper, we propose the measurement of secrecy in the information theoretical sense, and use our model to analyze three two-party scalar product

protocols. The results are summarized in Table 1. We note that although Protocol 2 achieves the highest level of security with the least complexity, i.e., random bits, communication cost, and computational efforts, it requires a semi-honest third party, which may be costly to implement in real applications. Protocol 3 may be slightly more secure than Protocol 1.

Table 1. Summary of results

	Protocol 1	Protocol 2	Protocol 3
random bits	0	$(2n+1)\lceil \log p \rceil$	$(m+n)\lceil \log p \rceil$
communication cost	$O(n \log p)$	$O(n \log p)$	$O(n \log p)$
computational complexity	$O(n^2)$	$O(n)$	$O(n^2)$
degree of secrecy	$\leq \frac{1}{2}$	1	$\leq \min(\frac{n-m}{n}, \frac{m-1}{n})$ $\leq \frac{1}{2} - \frac{1}{n}$
comments	requires a $n \times n$ inevitable matrix	requires a semi-honest third party	achieve max secrecy when $m = \lfloor n/2 \rfloor$

We consider that maintaining secrecy is an important factor in multi-party private computation, but it is not the sole goal. Thus, a tradeoff among computational complexity, communication complexity, and secrecy can be explored. The theoretical existential proof of solutions for multi-party private computation is elegant and impressive; however, it is not practical for real world, large-scale applications. For real applications, perfect secrecy is an ideal situation, but adequate secrecy is sometimes sufficient. Being able to quantify the secrecy preserved by protocols is important in deciding if an adequate secrecy level can be achieved. In this paper, we have proposed the use of an information theoretical framework to measure the secrecy of protocols. Furthermore, we have analyzed three two-party scalar protocols to demonstrate the efficacy of our approach.

Finally, there are two interesting research directions worthy of further study. First, it would interesting and challenging to develop general analysis methodologies. So far, we have only investigated the linearly dependent relationship between secret input and messages. More tools are needed to analyze more complex protocols. The second interesting direction would be to explore possible tradeoffs between secrecy and other performance related measurements.

References

1. M. J. Atallah and W. Du. Secure multi-party computational geometry. *Lecture Notes in Computer Science*, 2125:165–179, 2000.
2. W. Du and M. J. Atallah. Privacy-preserving cooperative statistical analysis. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 102–110, New Orleans, Louisiana, USA, December 2001.

3. W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In *New Security Paradigms Workshop*, pages 11–20, Cloudcroft, New Mexico, USA, September 2001.
4. W. Du and Z. Zhan. Building decision tree classifier on private data, 2002.
5. W. Du and Z. Zhan. A practical approach to solve secure multi-party computation problems. In *Proceedings of New Security Paradigms Workshop*, Virginia Beach, virginia, USA, September 2002.
6. O. Goldreich. *Foundations of Cryptography Volume II Basic Aplications*. Cambridge, 2004.
7. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or: A completeness theorem for protocols with honest majority. In *Proc. 19th ACM Symposium on Theory of Computing*, pages 218–229, 1987.
8. M. Kantarcoglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1026–1037, 2004.
9. Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay — a secure two-party computation system. In *Proceedings of the 13th Symposium on Security, Usenix*, pages 287–302, 2004.
10. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 639–644, July 2002.
11. A. C. Yao. How to generate and exchange secrets. In *Proceedings of the 27rd Annual IEEE Symposium on Foundations of Computer Science*, pages 162–167, November 1986.