Multiparty Computation Based on Connectivity of Graphs

Liangliang Xiao [a], Mulan Liu [b1], Zhifang Zhang [b]

[a]Institute of Software, Chinese Academy of Sciences, Beijing, 100080, China.
[b]Key Laboratory of Mathematics Mechanization, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China.

Abstract

In this paper, we contribute the construction of practical perfect multiparty computation protocols based on the connectivity of graphs.

Key Words: Multiparty computation, Linear secret sharing scheme, Monotone span program, Connectivity of graphs, Multiplicative linear secret sharing scheme

1 Introduction

The secure multiparty computation problem is fundamental in cryptography and distributed computations. A solution of multiparty computation problem implies in principle a solution to any cryptographic protocol problem. After it was proposed by Yao [11] for two party case and Goldreich, Micali , Wigderson [7] for multiparty case, it has become an active and developing field of information security.

Since in reality many problems under network environment can be modelled into graphs, it suggests us to study multiparty computation based on graphs. Several works including [2] [8] [1] [10] [4] have been done to study secret sharing schemes based on some special properties of graphs, but there is few works about multiparty computation based on graphs. In [5], Cramer, Damgard, Maurer devise a generic construction of multiparty computation protocol from any linear secret sharing scheme. The efficiency of the construction strongly depends on the efficiency of the linear secret sharing scheme. Furthermore a dual technique is used to guarantee the linear secret sharing scheme to be multiplicative, which doubles the computation amount. In this paper, we consider the family of adversary structures based on the connectivity of graphs. First we construct ideal linear secret sharing schemes based on the connectivity of graphs. Then we prove the schemes are already multiplicative, hence the dual technique is not needed. Actually we devise an efficient algorithm to compute the recombination vector. At last we apply the ideal linear secret sharing schemes to devise the multiparty computation protocols which are as efficient as the well known ones against the threshold adversaries.

The paper is organized as follows. Section 1 is an introduction. In section 2, first we prove the adversary structures based on the connectivity of graphs are $Q2$, but not $Q3$, then we construct ideal linear secret sharing schemes to realize the corresponding access structures. In section 3 we

[1]Corresponding author Mulan Liu, E-mail: mlliu@amss.ac.cn.

prove the ideal linear secret sharing schemes constructed in section 2 are multiplicative. Actually, we devise an efficient algorithm to compute the recombination vector. At the end of section 3, we apply the ideal linear secret sharing schemes to devise the multiparty computation protocols based on connectivity of graphs. The conclusions are in section 4.

2 Secret Sharing Schemes Based on Connectivity of Graphs

In this section, first we give a special class of access structures based on connectivity of graphs, then we devise ideal linear secret sharing schemes to realize the access structures. In order to do this, we recall some basic concepts and results such as access structure and adversary structure, linear secret sharing and monotone span program. Throughout this paper we denote $\mathbb{K}$ as a finite field and $P = \{P_1, \cdots, P_n\}$ as the set of n participants.

2.1 Access Structure and Adversary Structure

An access structure, denoted by AS, is a collection of subsets of P satisfying the monotone ascending property: for any $A' \in AS$ and $A \in 2^P$ with $A' \subset A$, it holds that $A \in AS$. An adversary structure, denoted by $\mathcal{A}$, is a collection of subsets of P satisfying the monotone descending property: for any $A' \in \mathcal{A}$ and $A \in 2^p$ with $A \subset A'$, it holds that $A \in \mathcal{A}$. In this paper, we consider the complete situation, i.e. $\mathcal{A} = 2^P - AS$.

The sets in AS are called authorized sets and the sets in $\mathcal{A}$ are called adversary sets. The minimum access structure, denoted by AS_m, is defined as $\{A \in AS | \forall B \subsetneqq A \Rightarrow B \notin AS\}$ and the sets in AS_m are called minimum authorized sets. The maximum adversary structure, denoted by $\mathcal{A}_m$, is defined as $\{B \in \mathcal{A} | \forall A \supsetneqq B \Rightarrow A \notin \mathcal{A}\}$ and the sets in $\mathcal{A}_m$ are called maximum adversary sets. Note that AS, $\mathcal{A}$, AS_m, and $\mathcal{A}_m$ can be uniquely determined by one another.

2.2 Linear Secret Sharing Scheme and Monotone Span Program

Secret sharing was proposed by Shamir [9] and Blackley [3] independently. The definition is as follows. Suppose that S is the domain of secrets, R is the set of random inputs, and S_i is the domain of shares of P_i where $1 \leq i \leq n$. A perfect secret sharing scheme, $PSSS$ for short, is composed of the distribution function $\Pi : S \times R \to S_1 \times \cdots \times S_n$ and the reconstruction function: for any $A \in AS$, $Re|_A : (S_1 \times \cdots \times S_n)|_A = S_{i_1} \times \cdots \times S_{i_{|A|}} \to S$ such that the following two requirements are satisfied.

1. Correctness requirement: for any $A \in AS, s \in S, r \in R$, $Re|_A(\Pi(s,r)|_A) = s$.
2. Security requirement: for any $B \in \mathcal{A}$, $H(S|\Pi(S,R)|_B) = H(S)$.

In the following we only discuss perfect secret sharing schemes. A secret sharing scheme is linear if S, R, S_i are linear subspaces over $\mathbb{K}$ and the reconstruction function is linear [1]. A linear

secret sharing scheme, $LSSS$ for short, is called ideal if $S = \mathbb{K}$ and $dim_{\mathbb{K}}(S_i) = 1$ for $1 \leq i \leq n$.

Span programs were introduced by Karchmer and Wigderson [8] as a linear algebraic model of computation. In [1], the author prove the equivalence of devising linear secret sharing scheme realizing the access structure and constructing monotone span program computing the corresponding monotone Boolean function. Suppose $\mathbb{K}$ is a finite field, we denote $(\mathbb{K}, M, \overrightarrow{v}, \rho)$ as the monotone span program where M is a matrix, ρ is the map from the rows of M to the literal set $\{x_1, \cdots, x_n\}$, and $\overrightarrow{v}$ is the nonzero target vector. If M is an $n \times d$ matrix, then $\overrightarrow{v}$ is a d dimensional vector. By the tool of monotone span program, it is easy to prove the equivalence of devising a linear secret sharing scheme realizing the access structure AS and finding a finite field $\mathbb{K}$, positive integer $l \in \mathbb{N}$, linear subspaces $V_{P_i} \subset \mathbb{K}^l$, $1 \leq i \leq n$, such that $\bigcap_{A \in AS_m} \sum_{P_i \in A} V_{P_i} - \bigcup_{B \in \mathcal{A}_m} \sum_{P_i \in B} V_{P_i} \neq \phi$. In the following, the formula will be used to construct linear secret sharing scheme.

2.3 Access Structures Based on Connectivity of Graphs and its Realizations

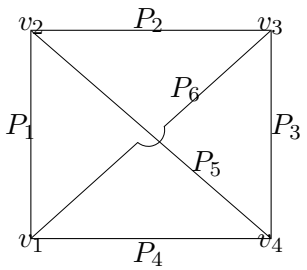
2.3.1 Access Structures Based on Connectivity of Graphs

Let m be a positive integer, $n = \begin{pmatrix} m \\ 2 \end{pmatrix}$, and $P = \{P_1, \cdots, P_n\}$ the set of participants. Let $G(V, E)$ be a undirected complete graph with the vertex set $V = \{v_1 \cdots, v_m\}$ and edge set $E = \{v_i v_j | i \neq j, 1 \leq i, j \leq m\}$. Suppose $f : P \to E$ is a bijection corresponding each participant with an edge. For any subset $A \subset P$, $G(V, E_A)$ is a spanning subgraph of $G(V, E)$ where $E_A = \{v_i v_j \in E | v_i v_j \in f(A)\}$. Define the access structure

$$AS = \{A \subset P | G(V, E_A) \text{ is a connected graph}\}. \tag{1}$$

Obviously AS satisfies the monotone ascending property since $G(V, E_A)$ is a spanning subgraph.

Example 2.1 *Let $m = 4$, $n = 6$, and $V = \{v_1, v_2, v_3, v_4\}$. Let $P = \{P_1, \cdots, P_6\}$, $f(P_1) = v_1v_2$, $f(P_2) = v_2v_3$, $f(P_3) = v_3v_4$, $f(P_4) = v_4v_1$, $f(P_5) = v_2v_4$, $f(P_6) = v_1v_3$. See the figure.*



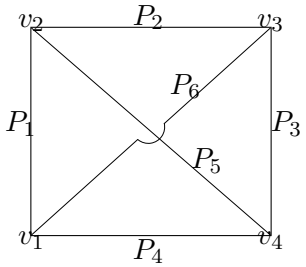
It's easy to have $AS_m = \{\{P_1, P_2, P_3\}, \{P_2, P_3, P_4\}, \{P_3, P_4, P_1\}, \{P_4, P_1, P_2\}, \{P_1, P_2, P_5\}, \{P_2, P_3, P_6\}, \{P_3, P_4, P_5\}, \{P_4, P_1, P_6\}, \{P_1, P_5, P_3\}, \{P_1, P_6, P_3\}, \{P_2, P_6, P_4\}, \{P_2, P_5, P_4\}, \{P_1, P_5, P_6\}, \{P_3, P_5, P_6\}, \{P_2, P_5, P_6\}, \{P_4, P_5, P_6\}\}$.

Proposition 2.1 *Suppose AS is given by (1) and $\mathcal{A} = 2^P - AS$ is the adversary structure. Then $\mathcal{A}$ is $Q2$, but not $Q3$* [2].

Proof: *Let $G(V, E')$ be an disconnected graph with $E' \subset E$. In order to prove $\mathcal{A}$ is $Q2$, it suffices to prove that $G(V, E - E')$ is a connected graph, that is, for every pair of vertices v and v', they are connected in the graph $G(V, E - E')$. Suppose the graph $G(V, E')$ has k connected components, $k \geq 2$. If the vertices v and v' are in different connected components of $G(V, E')$, then the edge $vv' \notin G(V, E')$. So the edge $vv' \in G(V, E - E')$ and it implies v and v' are connected in the graph $G(V, E - E')$. If the vertices v and v' are in the same connected component of $G(V, E')$, then we consider the vertex v'' in another connected component. We have v and v'' are connected, v' and v'' are connected in the graph $G(V, E - E')$. Hence v and v' are connected in the graph $G(V, E - E')$.*

Without loss of generality we can assume $|V| \geq 3$. It is equivalent to prove that there exist three disconnected subgraphs $G(V, E_1)$, $G(V, E_2)$, and $G(V, E_3)$ such that $G(V, E) = \bigcup_{i=1}^{3} G(V, E_i)$. Suppose v_1, v_2, and v_3 are three different vertices. Let $G(V, E_i)$ be the spanning subgraph of $G(V, E)$ obtained by deleting all the edges connected with the vertex v_i. Obviously $G(V, E_1)$, $G(V, E_2)$, and $G(V, E_3)$ are disconnected subgraphs and $G(V, E) = \bigcup_{i=1}^{3} G(V, E_i)$.

Example 2.2 *(following Example 2.1)*



Since $\mathcal{A}_m = \{\{P_1, P_3\}, \{P_2, P_4\}, \{P_5, P_6\}, \{P_1, P_2, P_6\}, \{P_2, P_3, P_5\}, \{P_3, P_4, P_6\}, \{P_4, P_1, P_5\}\}$, it's easy to verify that $\mathcal{A}$ is $Q2$ but not $Q3$.

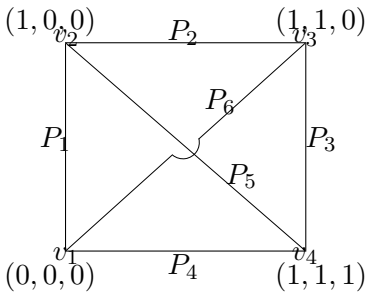
2.3.2 Ideal Linear Secret Sharing Scheme Realizing the Access Structure AS

Let $S = \mathbb{K}$ be a finite field with $|\mathbb{K}| > |\mathcal{A}_m|$ and $\overline{V} = \mathbb{K}^{m-1}$ be the $m-1$ dimensional linear space over $\mathbb{K}$. Select a basis of $\overline{V}$, say $\overrightarrow{v_1}, \cdots, \overrightarrow{v_{m-1}}$, and associate v_1 with $\overrightarrow{0}$, v_i with $\sum_{j=1}^{i-1} \overrightarrow{v_j}$, $2 \leq i \leq m$. Suppose $f(P_i) = vv'$, v is associated with the vector $\overrightarrow{v}$, and v' is associated with the vector $\overrightarrow{v'}$. let $V_{P_i} = span\{\overrightarrow{v} - \overrightarrow{v'}\}$.

Example 2.3 *(following Example 2.1)*

[2] $Q2$ means that for any $B, B' \in \mathcal{A}$, $B \cup B' \subsetneqq P$. $Q3$ means that for any $B, B', B'' \in \mathcal{A}$, $B \cup B' \cup B'' \subsetneqq P$.

Let $|\mathbb{K}| > 7$ *and* $\overline{V} = \mathbb{K}^3$. *Select* $\overrightarrow{v_1} = (1,0,0)$, $\overrightarrow{v_2} = (0,1,0)$, $\overrightarrow{v_3} = (0,0,1)$. *Associate vertex* v_1 *with* $(0,0,0)$, *vertex* v_2 *with* $(1,0,0)$, *vertex* v_3 *with* $(1,1,0)$, *vertex* v_4 *with* $(1,1,1)$.



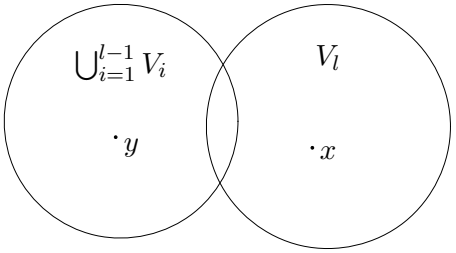
Let $V_{P_1} = span\{(1,0,0)\}$, $V_{P_2} = span\{(0,1,0)\}$, $V_{P_3} = span\{(0,0,1)\}$, $V_{P_4} = span\{(1,1,1)\}$, $V_{P_5} = span\{(0,1,1)\}$, $V_{P_6} = span\{(1,1,0)\}$.

Theorem 2.2 $\bigcap_{A\in AS_m} \sum_{P_i\in A} V_{P_i} - \bigcup_{B\in\mathcal{A}_m} \sum_{P_i\in B} V_{P_i} \neq \phi$

Proof: *First note that* $\sum_{i=1}^{n} V_{P_i} = \overline{V}$. *For any* $A \in AS_m$, $G(V, E_A)$ *forms a spanning tree of the graph* $G(V,E)$ *and adding any extra participant* P_i *to* A *will make a circle in the graph* $G(V, E_{A\cup\{P_i\}})$. *Since all the vectors* $\{\overrightarrow{v} - \overrightarrow{v'} | f(P_i) = vv'$, v *is associated with the vector* $\overrightarrow{v}$, *and* v' *is associated with the vector* $\overrightarrow{v'}\}$ *on a circle are linear dependent, it follows that* $\sum_{P_i\in A} V_{P_i} = \overline{V}$ *for any* $A \in AS_m$. *Hence* $\bigcap_{A\in AS_m} \sum_{P_i\in A} V_{P_i} - \bigcup_{B\in\mathcal{A}_m} \sum_{P_i\in B} V_{P_i} = \overline{V} - \bigcup_{B\in\mathcal{A}_m} \sum_{P_i\in B} V_{P_i}$.

For any $B \in \mathcal{A}_m$, *suppose* $G(V, E_B) = \bigcup_{i=1}^{l} G_i(V_i, E_i)$ *where* $G_i(V_i, E_i)$ *is the connected component and* $l \geq 2$. *Since* $dim_{\mathbb{K}} \sum_{P_i\in f^{-1}(E_i)} V_{P_i} = |V_i| - 1$, $dim_{\mathbb{K}} \sum_{P_i\in B} V_{P_i} \leq \sum_{i=1}^{l} dim_{\mathbb{K}} \sum_{P_i\in f^{-1}(E_i)} V_{P_i} = \sum_{i=1}^{l}(|V_i| - 1) = \sum_{i=1}^{l} |V_i| - l \leq m - l < m - 1$. *Hence* $\sum_{P_i\in B} V_{P_i} \subsetneqq \overline{V}$. *By the following lemma, the theorem is proved.*

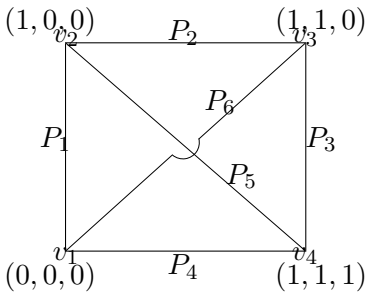
Lemma 2.3 *Suppose* $\overline{V}$ *is a linear space over the finite field* $\mathbb{K}$, *and* $V_i \subsetneqq \overline{V}$ *is a linear subspace,* $1 \leq i \leq l$. *If* $|\mathbb{K}| > l$, *then* $\bigcup_{i=1}^{l} V_i \subsetneqq \overline{V}$.



Proof: *According to reduce of absurdity and without loss of generality, we assume* $\bigcup_{i=1}^{l-1} V_i \subsetneqq \overline{V}$ *but* $\bigcup_{i=1}^{l} V_i = \overline{V}$. *Choose an element* x *in* $V_l - \bigcup_{i=1}^{l-1} V_i$ *and an element* y *in* $\bigcup_{i=1}^{l-1} V_i - V_l$, *consider the*

set of elements $\{x + \alpha \cdot y | \alpha \in \mathbb{K}\}$. *According to the Pigeonhole Principle, there exists* $\alpha_1 \neq \alpha_2$, *and* $V_{i_0} \in \{V_i | 1 \leq i \leq l\}$ *such that* $x + \alpha_1 \cdot y, x + \alpha_2 \cdot y \in V_{i_0}$. *It follows that* $x, y \in V_{i_0}$ *which contradicts to the choice of* x *and* y.

Example 2.4 *(following Example 2.3)*



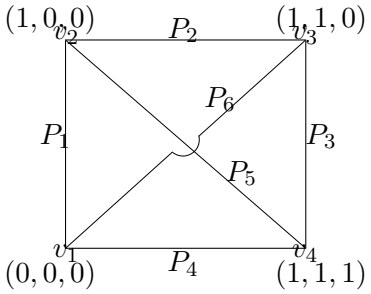
Note that $AS_m = \{\{P_1, P_2, P_3\}, \{P_2, P_3, P_4\}, \{P_3, P_4, P_1\}, \{P_4, P_1, P_2\}, \{P_1, P_2, P_5\}, \{P_2, P_3, P_6\}, \{P_3, P_4, P_5\}, \{P_4, P_1, P_6\}, \{P_1, P_5, P_3\}, \{P_1, P_6, P_3\}, \{P_2, P_6, P_4\}, \{P_2, P_5, P_4\}, \{P_1, P_5, P_6\}, \{P_3, P_5, P_6\}, \{P_2, P_5, P_6\}, \{P_4, P_5, P_6\}\}$,
$\mathcal{A}_m = \{\{P_1, P_3\}, \{P_2, P_4\}, \{P_5, P_6\}, \{P_1, P_2, P_6\}, \{P_2, P_3, P_5\}, \{P_3, P_4, P_6\}, \{P_4, P_1, P_5\}\}$,
$V_{P_1} = span\{(1, 0, 0)\}$, $V_{P_2} = span\{(0, 1, 0)\}$, $V_{P_3} = span\{(0, 0, 1)\}$, $V_{P_4} = span\{(1, 1, 1)\}$, $V_{P_5} = span\{(0, 1, 1)\}$, $V_{P_6} = span\{(1, 1, 0)\}$.
Let $\mathbb{K} = GF(p)$ *where* $p > 7$ *is a prime number, it's easy to verify that* $(1, 2, 3) \in \bigcap_{A \in AS_m} \sum_{P_i \in A} V_{P_i} - \bigcup_{B \in \mathcal{A}_m} \sum_{P_i \in B} V_{P_i}$.

As a direct result of Theorem 2.2, we have the following corollary.

Corollary 2.4 *There is an ideal linear secret sharing scheme realizing the access structure* AS.

Since $\bigcap_{A \in AS_m} \sum_{P_i \in A} V_{P_i} - \bigcup_{B \in \mathcal{A}_m} \sum_{P_i \in B} V_{P_i} \neq \phi$, we construct the monotone span program $(\mathbb{K}, M, \overrightarrow{v}, \rho)$ as follows. Suppose $f(P_i) = vv'$, v is associated with the vector $\overrightarrow{v}$, and v' is associated with the vector $\overrightarrow{v'}$. M is constituted by all the row vectors $\overrightarrow{v} - \overrightarrow{v'}$ for $1 \leq i \leq n$. ρ maps the row corresponding with P_i to x_i, and $\overrightarrow{v}$ can be any vector in $\bigcap_{A \in AS_m} \sum_{P_i \in A} V_{P_i} - \bigcup_{B \in \mathcal{A}_m} \sum_{P_i \in B} V_{P_i}$. By the method mentioned in [1], we can construct an ideal linear secret sharing scheme realizing AS.

Example 2.5 *(following Example 2.4)*



According to Example 2.4, $(\mathbb{K}, M, \overrightarrow{v}, \rho)$ *is the monotone span program where* $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\rho(i) = x_i$ *for* $1 \leq i \leq 6$, *and* $\overrightarrow{v} = (1, 2, 3)$. *We can construct an linear secret sharing scheme as follows [1].*

Distribution phase: suppose $s \in \mathbb{K}$ *is the secret, the dealer chooses randomly* $r_i \in \mathbb{K}$ *for* $1 \leq i \leq 2$, *computes* $M \cdot (s - 2r_1 - 3r_2, r_1, r_2)^\tau = (s - 2r_1 - 3r_2, r_1, r_2, s - r_1 - 2r_2, r_1 + r_2, s - r_1 - 3r_2)^\tau$ *and transmits the i-th row of the vector to* P_i *secretly where* τ *represents the transpose.*

Reconstruction phase: suppose A *is an authorized set and the participants in* A *want to recover the secret* s. *Without loss of generality, we assume* $A = \{P_1, P_2, P_3\}$. *Note that the row vectors in* M *associating with* x_1 *is* $(1, 0, 0)$, x_2 *is* $(0, 1, 0)$, x_3 *is* $(0, 0, 1)$. *The target vector* $\overrightarrow{v} = (1, 2, 3)$ *and* $(1, 2, 3) = 1(1, 0, 0) + 2(0, 1, 0) + 3(0, 0, 1)$. *Hence* P_1, P_2, P_3 *compute* $1(s - 2r_1 - 3r_2) + 2r_1 + 3r_2 = s$.

3 Multiparty Computation Protocols Based on Connectivity of Graphs

In this section, first we prove the ideal linear secret sharing schemes constructed in section 2 are multiplicative. Actually, we devise an efficient algorithm to compute the recombination vector. Then we apply the ideal linear secret sharing schemes to devise the multiparty computation protocols based on connectivity of graphs.

Since the access structures based on connectivity of graphs are $Q2$, it implies that any polynomial over $\mathbb{K}$ can be perfectly securely computed by a multiparty computation protocol against any adaptive and passive $\mathcal{A}$-adversary [5]. Since for computing a polynomial, it is enough to know how to compute the addition and multiplication of two elements. In the following, we only discuss how to compute addition and multiplication securely. Suppose s, s' are two secrets, Π is the distribution function. Let $\Pi(s, r) = (s_1, \cdots, s_n)$ and $\Pi(s', r') = (s'_1, \cdots, s'_n)$. A secret sharing scheme can be successfully applied to the construction of multiparty computation protocol if it has the additive property and multiplicative property, that is, $\Pi(s + s', r'') = (s_1 + s'_1, \cdots, s_n + s'_n)$ and ss'

can be obtained by the linear combination of $(s_1s'_1, \cdots, s_ns'_n)$. Suppose $ss' = \sum_{i=1}^{n} z_i \cdot s_is'_i$, then $\overrightarrow{z} = (z_1, \cdots, z_n)$ is called the recombination vector. Obviously the linear secret sharing scheme satisfies the additive property, but generally speaking it does not satisfy the multiplicative property.

For the basis $\overrightarrow{v_i} = \overrightarrow{e_{i-1}}$ of $\overrightarrow{V}$, we will prove in what follows that the ideal linear secret sharing scheme constructed in section 2 is multiplicative. Actually, we contribute an efficient algorithm to compute the recombination vector $\overrightarrow{z}$. Thus we can apply it to get a very efficient multiparty computation protocol. Notice that we associate vertex v_1 with $\overrightarrow{0}$, vertex v_i with $\sum_{j=1}^{i-1} \overrightarrow{e_j}$, $2 \leq i \leq m$. Hence all row vectors of the $n \times (m-1)$ matrix M are constituted by successive 1's and vice versa. Assume $M = (M_1, M_2, \cdots, M_{m-1})$ where $M_i = (m_{1i}, \cdots, m_{ni})^\tau$ is the i-th column of M. Let $M*$ be the matrix constituted by all the column vectors $M_i * M_j$, $1 \leq i \leq j \leq m-1$, where $M_i * M_j = (m_{1i}m_{1j}, \cdots, m_{ni}m_{nj})^\tau$. Note that $M*$ is a $n \times n$ matrix and $M* = (M_{i_1} * M_{j_1}, \cdots, M_{i_n} * M_{j_n})$.

Lemma 3.1 *The $n \times n$ matrix $M*$ is nonsingular.*

Proof: *We put the proof into the Appendix.*

Suppose $\overrightarrow{v}$ is the target vector. Let $N = \overrightarrow{v}^\tau \cdot \overrightarrow{v} = (a_{ij})_{1 \leq i,j \leq m-1}$, $\overrightarrow{v*} = (a_{i_1j_1}, \cdots, a_{i_nj_n})$. Consider the linear equation system $(M*)^\tau \cdot (z_1, \cdots, z_n)^\tau = \overrightarrow{v*}^\tau$ over $\mathbb{K}$, where $z_1, \cdots, z_n$ are variables. Since $M*$ is nonsingular, there is a solution which we still denoted by $(z_1, \cdots, z_n)$.

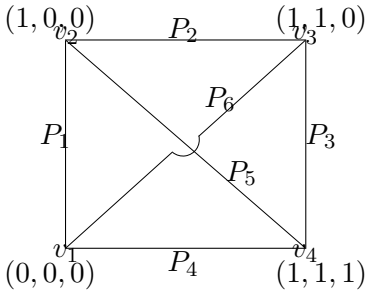
Theorem 3.2 $(\mathbb{K}, M, \overrightarrow{v}, \rho)$ *is multiplicative and $(z_1, \cdots, z_n)$ is the recombination vector.*

Proof: *Suppose $M = (M_1, \cdots, M_{m-1})$, $M* = (M_{i_1} * M_{j_1}, \cdots, M_{i_n} * M_{j_n})$, $N = \overrightarrow{v}^\tau \cdot \overrightarrow{v} = (a_{ij})_{1 \leq i \leq j \leq m-1}$, $\overrightarrow{v*} = (a_{i_1j_1}, \cdots, a_{i_nj_n})$ are constructed as above. Let s, s' be two secrets. Choose two vectors $\overrightarrow{y}, \overrightarrow{y'}$ satisfying $\overrightarrow{v} \cdot \overrightarrow{y}^\tau = s$, $\overrightarrow{v} \cdot \overrightarrow{y'}^\tau = s'$. Suppose $M \cdot \overrightarrow{y}^\tau = (s_1, \cdots, s_n)^\tau$ and $M \cdot \overrightarrow{y'}^\tau = (s'_1, \cdots, s'_n)^\tau$.*

Denote $\begin{bmatrix} z_1 & & \\ & \ddots & \\ & & z_n \end{bmatrix} = [z_1, \cdots, z_n]$, *for any elements $z_1, \cdots, z_n$ of $\mathbb{K}$, $\sum_{i=1}^{n} z_i \cdot s_is'_i = (s_1, \cdots, s_n) \cdot [z_1, \cdots, z_n] \cdot (s'_1, \cdots, s'_n)^\tau = \overrightarrow{y} M^\tau \cdot [z_1, \cdots, z_n] \cdot M \overrightarrow{y'}^\tau$. If $z_1, \cdots, z_n$ satisfy the equation $M^\tau \cdot [z_1, \cdots, z_n] \cdot M = \overrightarrow{v}^\tau \cdot \overrightarrow{v}$, then $\sum_{i=1}^{n} z_i \cdot s_is'_i = \overrightarrow{y}\,\overrightarrow{v}^\tau \cdot \overrightarrow{v}\,\overrightarrow{y}^\tau = ss'$. Hence it suffices to prove $M^\tau \cdot [z_1, \cdots, z_n] \cdot M = \overrightarrow{v}^\tau \cdot \overrightarrow{v} \Leftrightarrow (M*)^\tau \cdot (z_1, \cdots, z_n)^\tau = \overrightarrow{v*}^\tau$.*

Let $M = (b_{ij})$ and $M^\tau = (b_{ji})$. Then $M^\tau \cdot [z_1, \cdots, z_n] \cdot M = (b_{ji}) \cdot [z_1, \cdots, z_n] \cdot (b_{ij})$. For any $1 \leq i \leq j \leq m-1$, the (i,j)-th entry of $M^\tau \cdot [z_1, \cdots, z_n] \cdot M$ is $\sum_{k=1}^{n} z_k b_{ki} b_{kj} = (M_i * M_j)^\tau \cdot (z_1, \cdots, z_n)^\tau$. Furthermore, since $M^\tau \cdot [z_1, \cdots, z_n] \cdot M$ and $\overrightarrow{v}^\tau \cdot \overrightarrow{v}$ are symmetric matrix, it implies $M^\tau \cdot [z_1, \cdots, z_n] \cdot M = \overrightarrow{v}^\tau \cdot \overrightarrow{v} \Leftrightarrow$the entries of the upper triangle are equal. Thus it finishes the proof.

Example 3.1 *Suppose the graph $G(V, E)$ and the monotone span program $(\mathbb{K}, M, \overrightarrow{v}, \rho)$ are the same as in Example 2.5. Let's compute the recombination vector and verify $(\mathbb{K}, M, \overrightarrow{v}, \rho)$ is multiplicative.*



First we compute the recombination vector as follows. Note that $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\rho(i) = x_i$ *for* $1 \leq i \leq 6$, *and* $\overrightarrow{v} = (1, 2, 3)$. *Assume* $M_1 = (1, 0, 0, 1, 0, 1)^\tau$, $M_2 = (0, 1, 0, 1, 1, 1)^\tau$, $M_3 = (0, 0, 1, 1, 1, 0)^\tau$. *Let* $M* = (M_1 * M_1, M_2 * M_2, M_3 * M_3, M_1 * M_2, M_2 * M_3, M_1 * M_3) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$. *Let* $N = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot [1, 2, 3] = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} = (a_{ij})$ *where* a_{ij} *is the* (i, j)*-th entry of* N. *Therefore* $\overrightarrow{v*} = (a_{11}, a_{22}, a_{33}, a_{12}, a_{23}, a_{13})$. *Consider the linear equation system* $(M*)^\tau \cdot (z_1, z_2, z_3, z_4, z_5, z_6)^\tau = (a_{11}, a_{22}, a_{33}, a_{12}, a_{23}, a_{13})^\tau$, *i.e.* $\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 2 \\ 6 \\ 3 \end{bmatrix}$. *It has a unique solution* $(z_1, z_2, z_3, z_4, z_5, z_6)^\tau = (-1, -1, 3, 3, 3, -1)^\tau$.

In the following we verify $(z_1, \cdots, z_6)$ *is the recombination vector. Suppose* $s, s', r_i, r_i' \in \mathbb{K}$, $1 \leq i \leq 2$. *As the distribution phase in Example 2.5, the dealer computes* $(s_1, \cdots, s_6)^\tau = M \cdot (s - 2r_1 - 3r_2, r_1, r_2)^\tau = (s - 2r_1 - 3r_2, r_1, r_2, s - r_1 - 2r_2, r_1 + r_2, s - r_1 - 3r_2)^\tau$, $(s_1', \cdots, s_6')^\tau = M \cdot (s' - 2r_1' - 3r_2', r_1', r_2')^\tau = (s' - 2r_1' - 3r_2', r_1', r_2', s' - r_1' - 2r_2', r_1' + r_2', s' - r_1' - 3r_2')^\tau$. *It suffices to verify* $\sum_{i=1}^{6} z_i \cdot s_i s_i' = ss'$.

Compute $s_1 s_1' = (s - 2r_1 - 3r_2) \cdot (s' - 2r_1' - 3r_2') = ss' - 2sr_1' - 3sr_2' - 2r_1 s' + 4r_1 r_1' + 6r_1 r_2' -$

$3r_2s' + 6r_2r'_1 + 9r_2r'_2$.

Compute $s_2s'_2 = r_1r'_1$.

Compute $s_3s'_3 = r_2r'_2$.

Compute $s_4s'_4 = (s - r_1 - 2r_2) \cdot (s' - r'_1 - 2r'_2) = ss' - sr'_1 - 2sr'_2 - r_1s' + r_1r'_1 + 2r_1r'_2 - 2r_2s' + 2r_2r'_1 + 4r_2r'_2$.

Compute $s_5s'_5 = (r_1 + r_2) \cdot (r'_1 + r'_2) = r_1r'_1 + r_1r'_2 + r_2r'_1 + r_2r'_2$.

Compute $s_6s'_6 = (s - r_1 - 3r_2) \cdot (s' - r'_1 - 3r'_2) = ss' - sr'_1 - 3sr'_2 - r_1s' + r_1r'_1 + 3r_1r'_2 - 3r_2s' + 3r_2r'_1 + 9r_2r'_2$.

It can be easily verified that $\sum_{i=1}^{6} z_i \cdot s_is'_i = -(ss' - 2sr'_1 - 3sr'_2 - 2r_1s' + 4r_1r'_1 + 6r_1r'_2 - 3r_2s' + 6r_2r'_1 + 9r_2r'_2) - r_1r'_1 + 3r_2r'_2 + 3(ss' - sr'_1 - 2sr'_2 - r_1s' + r_1r'_1 + 2r_1r'_2 - 2r_2s' + 2r_2r'_1 + 4r_2r'_2) + 3(r_1r'_1 + r_1r'_2 + r_2r'_1 + r_2r'_2) - (ss' - sr'_1 - 3sr'_2 - r_1s' + r_1r'_1 + 3r_1r'_2 - 3r_2s' + 3r_2r'_1 + 9r_2r'_2) = ss'$.

As a result of this section, we apply the ideal linear secret sharing scheme to devise the protocol of computing addition and multiplication. It is similar to the one against the threshold adversary [6]. Assume the input values are s and s', determined by shares $s_1, \cdots, s_n$ and $s'_1, \cdots, s'_n$, respectively.

Addition For $i = 1, \cdots, n$, P_i computes $s_i + s'_i$. The shares $s_1 + s'_1, \cdots, s_n + s'_n$ determine $s + s'$.

Multiplication For $i = 1, \cdots, n$, P_i computes $s_is'_i = \widetilde{t_i}$.

Resharing step: P_i secretly shares $\widetilde{t_i}$, resulting in shares $t_{i1}, \cdots, t_{in}$, and sends t_{ij} to P_j.

Recombination step: For $j = 1, \cdots, n$, player P_j computes $t_j = \sum_{i=1}^{n} z_it_{ij}$, where $(z_1, \cdots, z_n)$ is the recombination vector. The shares $t_1, \cdots, t_n$ determine $t = ss'$.

4 Conclusions

In this paper we devise the ideal linear secret sharing schemes based on connectivity of graphs and prove they are multiplicative. Furthermore we devise an efficient algorithm to compute the recombination vector. We apply the ideal linear secret sharing schemes to devise the practical perfect multiparty computation protocols which are as efficient as the ones against the threshold adversaries. The method is different from the generic construction proposed by Cramer, Damgard, and Maurer and is more efficient for our case.

Appendix

Proof of lemma 3.1: Instead of the tedious but rigorous proof, a heuristic illustration is presented as follows.

Consider the case of $m = 5$, $n = 10$. Notice that all row vectors in M are constituted by successive 1's and vice versa, so M can be arranged as $\begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & \\ & 1 & & & \\ & 1 & 1 & & \\ & 1 & 1 & 1 & \\ & & 1 & & \\ & & 1 & 1 & \\ & & & 1 & \end{bmatrix}$ by row exchanges. We construct $M*$ by the rule of first adding the columns $M_i * M_{i+1}$ to M, then adding the columns $M_i *$ M_{i+2}, $M_i * M_{i+3}$, $\cdots$. Hence $M* = \left[\begin{array}{ccccc:ccccc:ccccc:cc} 1 & & & & & & & & & & & & & & \\ 1 & 1 & & & & 1 & & & & & & & & & \\ 1 & 1 & 1 & & & 1 & 1 & & & 1 & & & & & \\ 1 & 1 & 1 & 1 & & 1 & 1 & 1 & & 1 & 1 & & 1 & & \\ \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots & & \cdots & \cdots & & \cdots & & \\ & 1 & & & & & & & & & & & & & \\ & 1 & 1 & & & & 1 & & & & & & & & \\ & 1 & 1 & 1 & & & 1 & 1 & & & & 1 & & & \\ \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots & & \cdots & \cdots & & \cdots & & \\ & & 1 & & & & & & & & & & & & \\ & & 1 & 1 & & & & 1 & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots & & \cdots & \cdots & & \cdots & & \\ & & & 1 & & & & & & & & & & & \end{array}\right]$.

Suppose the i-th row of $M*$ is $\overrightarrow{u_i}$, $1 \le i \le 10$. Then $\overrightarrow{u_1} = \overrightarrow{e_1}$, $\overrightarrow{u_5} = \overrightarrow{e_2}$, $\overrightarrow{u_8} = \overrightarrow{e_3}$, $\overrightarrow{u_{10}} = \overrightarrow{e_4}$. Since $\overrightarrow{e_1} + \overrightarrow{e_2} + \overrightarrow{e_5} = \overrightarrow{u_2}$, $\overrightarrow{e_5} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$. Since $\overrightarrow{e_2} + \overrightarrow{e_3} + \overrightarrow{e_6} = \overrightarrow{u_6}$, $\overrightarrow{e_6} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$. Since $\overrightarrow{e_3} + \overrightarrow{e_4} + \overrightarrow{e_7} = \overrightarrow{u_9}$, $\overrightarrow{e_7} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$. Since $\sum_{i=1}^{3} \overrightarrow{e_i} + \sum_{i=5}^{6} \overrightarrow{e_i} + \overrightarrow{e_8} = \overrightarrow{u_3}$, $\overrightarrow{e_8} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$. Since $\sum_{i=2}^{4} \overrightarrow{e_i} + \sum_{i=6}^{7} \overrightarrow{e_i} + \overrightarrow{e_9} = \overrightarrow{u_7}$, $\overrightarrow{e_9} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$. Since $\sum_{i=1}^{10} \overrightarrow{e_i} = \overrightarrow{u_4}$, $\overrightarrow{e_{10}} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$. Hence $\overrightarrow{e_1}, \cdots, \overrightarrow{e_{10}} \in span\{\overrightarrow{u_1}, \cdots, \overrightarrow{u_{10}}\}$ which implies $M*$ is nonsingular.

References

[1] Beimel A., Secure Schemes for Secret Sharing and Key Distribution, PhD thesis, Technion - Israel Institute of Techonlogy, 1996.

[2] J. Benaloh and S. Rudich. Private communication, 1989.

[3] Blackley G.R., Safeguarding cryptographic keys, *Proc. of the 1979 AFIPS National Computer Conference*, 1979, 48:313-317.

[4] C. Blundo, A. De Santis, D. R. Stinson and U. Vaccaro, Graph decompositions and secret sharing schemes, J. Cryptology 8 (1995), 39-64. [Preliminary version appeared in "Advances in Cryptology – EUROCRYPT '92", R. A. Rueppel, ed., Lecture Notes in Computer Science 658 (1993), 1-24.]

[5] R. Cramer, I. Damgard, U. Maurer. General Secure Multi-Party Computation from any Linear Secret-Sharing Scheme. In: *Proc. EUROCRYPT '00*, Springer Verlag LNCS, vol 1807, pp. 316–334. Full version available from IACR eprint archive, 2000.

[6] R. Cramer, I. Damgard. Multiparty Computation, an introduction. CPT, *Lecture Notes, DAIMI*, 2002.

[7] O. Goldreich, S. Micali ,A. Wigderson. How to play ANY mental game. *Proceedings of the nineteenth annual ACM conference on Theory of computing*, pp.218-229, January 1987, New York, New York, United States.

[8] M. Karchmer and A. Wigderson. On span programs. In *Proc. 8th Ann. Symp. Structure in complexity Theory*, IEEE 1993, pp. 102-111.

[9] Shamir A., How to share a secret, *Communications of the ACM*, 1979, 22:612-613.

[10] H.-U. Sun and S.-P. Shieh, An efficient construction of perfect secret sharing schemes for graph-based access structures, Computers and Mathematics with Applictions 31 (1996), 129-135.

[11] A. Yao. Protocols for Secure Computation. *Proc. of IEEE FOGS '82*, pp. 160-164, 1982.