Construction of Multiplicative Monotone Span Program*

Yuenai Chen [1], Chunming Tang [1,2]

[1] School of Mathematics and Information Sciences, Guangzhou University, Guangzhou 510006, China
[2] Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong Higher Education Institutes, Guangzhou University, 510006, China

Abstract. Multiplicative monotone span program is one of the important tools to realize secure multiparty computation. It is essential to construct multiplicative monotone span programs for secure multiparty computations. For any access structure, Cramer et al. gave a method to construct multiplicative monotone span programs, but its row size became double, and the column size also increased. In this paper, we propose a new construction which can get a multiplicative monotone span program with the row size less than double without changing the column size.

Keywords: secure multiparty computation, monotone span program, multiplicative

1 Introduction

Secure multiparty computation can be defined as the problem of n participants $P = \{P_1, \cdots, P_n\}$ want to compute an agreed function $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ in a secure way, where the participant P_i takes the input x_i, and receives the output y_i for $1 \leq i \leq n$. Here the security means guaranteeing the correctness of the output as well as the privacy of each participant's input, even though some participants cheat.

Any a computable function can be assumed to be a polynomial over finite field[1−3]. Thus secure multiparty computation problem is transformed into how to compute the addition and multiplication in a security way. As we are known, linear secret sharing scheme can be used to realize the secure multiparty computation[4,5]. According to the addition homomorphism of linear secret sharing scheme, the addition can be easily compute. For multiplication, Cramer

[2] *This work was supported by the Foundation of National Natural Science of China under Grant No. 11271003, Guangdong Provincial Natural Science Foundation (China) under Grant No. S2012010009950, High Level Talents Project of Guangdong, and Science Research Project of Education Bureau in Guangzhou under Grant No. 2012A004.

introduces a multiplicative monotone span program[6], which can convert the product of the two main keys to a linear combination of each participant's subkey product which can be computed individually. Therefore, the multiplication is converted into a linear computation, which can be safely computed. It is essential for secure multiparty computation that how to construct a multiplicative monotone span program. Cramer gives a construction[6], which makes the row size of the monotone span program to double, and the column size also increases. The row size of the monotone span program refers to the data diffusion of the linear secret sharing scheme, while the column size determines the computational complexity when reconstructing keys. Considering this, we propose a new construction of multiplicative monotone span program which can make the row size less than double without changing the column size.

2 Basic Knowledge

Suppose $P = \{1, \cdots, n\}$ is the set of all players in a secret sharing schemeThe access structure Γ over P is the collection of some subsets of P. Γ has monotonicity, if $A \in \Gamma$, $A \subset B$, then $B \in \Gamma$.

An *linear secret sharing scheme*(LSSS) is defined over a finite field $\mathbb{F}$, and the secret s to be distributed is an element in $\mathbb{F}$. The dealer distributes the secret s among players in P according to an distributed function $\Pi : S \times R \to s_1 \times \cdots \times s_n$ which is linear, i.e., $\Pi(s, r) = (s_1 \cdots, s_n)$, where $r \in R$ is random input. Each player i receives from the dealer a share s_i.

An LSSS is said to *realize* Γ, if 1) for any $A \in \Gamma$, the players in A can recover the secret s, then A is called *qualified* subset; 2) For any $B \notin \Gamma$, the players in B can not get any information about the secret s, then B is called *unqualified* subset. If we use R to represent the collection of all unqualified subsets, then R is called *adversary structure*. If $A \in \Gamma$, and for any $C \subset A$, $C \notin \Gamma$, then A is called the *minimal qualified subset*. Once the minimal qualified subsets are set up, Γ is uniquely determined; And vice versa. Therefore, we use Γ to represent the collection of all minimal qualified subsets in the rest of this paper. Similarly, If $D \notin \Gamma$, but for any $E \supset D$, $E \in \Gamma$, then D is called the *maximal unqualified subset*, and we use R to represent the collection of all maximal unqualified subsets. R and Γ are uniquely determined by each other.

Most proposed secret sharing schemes are linear, but the concept of an LSSS was first considered in its full generality by Karchmer and Wigderson[7] who introduced the equivalent notion of monotone span program(MSP). The span program is a linear algebraic model for computing boolean functions[7]. For any access structure Γ, it can be represented by a Boolean function as follow:

For any $A \in \Gamma$, let $\lambda_A = (\lambda_1, \cdots, \lambda_n)$, where $\lambda_i = 1$ if and only if $i \in A$, otherwise $\lambda_i = 0$. Now we can define the Boolean function $f_\Gamma : \{0,1\}^n \to \{0,1\}$ over access structure Γ that for any $\lambda \in \{0,1\}^n$, $f_\Gamma(\lambda) = 1$ if and only if there is $A \in \Gamma$ such that $\lambda_A = \lambda$, otherwise $f_\Gamma(\lambda) = 0$. For simplicity reasons, we denote $f(A) = 1$ if and only if $A \in \Gamma$, otherwise $f(A) = 0$.

Definition 2.1[8] *An MSP is given by a quadruple* $(\mathbb{F}, M, \varphi, \varepsilon)$*, where* $\mathbb{F}$ *is a field,* M *is a* $(d \times l)$*-matrix over* $\mathbb{F}$ *with a* ***labeling*** $\varphi : \{1, \cdots, d\} \to \{1, \cdots, n\}$*, which assigns to every row of* M *a player in* P*, and* ε *is a fixed non-zero vector, called the* ***target vector****. Without loss of generality, we take* $\varepsilon = (1, \cdots, 1)$*. The* ***row size*** *of a monotone span program is the number of rows of matrix* M*, and the* ***column size*** *is the number of columns of* M*, denoted by* ***Rsize*** *and* ***Csize*** *respectively.*

As we are known, MSP can be used to realize an LSSS. Suppose $M = (\mathbf{m}_1, \cdots, \mathbf{m}_d)^\top$, where $\mathbf{m}_i \in \mathbb{F}^l$ is the i-th row vector of M. In order to determine the shares of secret s, the dealer chooses a random vector $\mathbf{u} = (u_1, \cdots, u_l) \in \mathbb{F}^l$ such that $s = \mathbf{u}\varepsilon^\top$, and the share $s_i = \mathbf{u}\mathbf{m}_i^\top$ is assigned to a player according to φ.

Definition 2.2[8] *We say the MSP* $(\mathbb{F}, M, \varphi, \varepsilon)$ ***accepts*** *the subset* $A \subset P$ *if and only if* $\varepsilon \in span(M_A)$*; otherwise we say it* ***rejects*** A*.*

M_A represents the sub-matrix of M, which is composed by all the row vectors which are assigned to players in A. Suppose $M_A = (\mathbf{m}_{i_1}, \cdots, \mathbf{m}_{i_t})^\top$ $span(M_A) = span\{\mathbf{m}_{i_1}, \cdots, \mathbf{m}_{i_t}\}$ represents the space generated by the vectors $\mathbf{m}_{i_j}$, $j = 1, \cdots, t$.

If the MSP accepts A, then the secret can be recovered by first solving the linear equation

$$\varepsilon = \sum_{j=1}^{t} x_j \mathbf{m}_{i_j}$$

after finding x_js, the secret can be computed as

$$s = \mathbf{u}\varepsilon^\top = \sum_{j=1}^{t} x_j \mathbf{u}\mathbf{m}_{i_j}^\top = \sum_{j=1}^{t} x_j s_{i_j}$$

Definition 2.3[8] *An MSP* $(\mathbb{F}, M, \varphi, \varepsilon)$ ***computes*** *the Boolean function* $f_\Gamma : \{0,1\}^n \to \{0,1\}$ *if it accepts exactly those sets* $A \in \Gamma$*. Then we say the MSP* ***realize*** *access structure* Γ*.*

Actually, MSPs and LSSSs are in natural 1-1 correspondence. Since it is more convenient to use MSP than LSSS when describing some issues, so in this paper, we use *multiplicative* MSP(M-MSP) instead of multiplicative LSSS to discuss its construction.

As we have mentioned, M-MSP is very important to multi-party computation, therefore, it is critical to construct an M-MSP for any access structure. Now we introduce the definition and determination of M-MSP given by Cramer[6] and Zhifang Zhang[9] respectively.

Assume $(\mathbb{F}, M, \varphi, \varepsilon)$ computes the Boolean function f_Γ, for any two vectors $x = (x_1, \cdots, x_d)^\top$ and $y = (y_1, \cdots, y_d)^\top$ of d dimensions over $\mathbb{F}^d$, defines the operation "$\otimes$" as follow:

$$x \otimes y = (x_i y_i, x_i y_j + x_j y_i | 1 \le i < j \le d, i \ne j, \varphi(i) = \varphi(j))^\top,$$

Rewrite the vectors $x = (x_1, \cdots, x_d)^\top$ and $y = (y_1, \cdots, y_d)^\top$ as

$$x = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1d_1} \\ \vdots \\ x_{n1} \\ \vdots \\ x_{nd_n} \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ 1 \\ \vdots \\ n \\ \vdots \\ n \end{matrix}, y = \begin{pmatrix} y_{11} \\ \vdots \\ y_{1d_1} \\ \vdots \\ y_{n1} \\ \vdots \\ y_{nd_n} \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ 1 \\ \vdots \\ n \\ \vdots \\ n \end{matrix}$$

Where $x_{i1}, \cdots, x_{id_i}$ and $y_{i1}, \cdots, y_{id_i}$ are labeled by participant i for $1 \leq i \leq n$, and $d = \sum_{i=1}^{n} d_i$, then

$$x \otimes y = (x_{ij}y_{ij}, x_{ks}y_{kt} + x_{kt}y_{ks} | 1 \leq i, j \leq d, 1 \leq k \leq n, 1 \leq s < t \leq d_k),$$

obviously, the dimension of $x \otimes y$ is $d^* = \sum_{i=1}^{n} \frac{d_i(d_i+1)}{2}$.

Definition 2.4[6] *An MSP $(\mathbb{F}, M, \varphi, \varepsilon)$ is said to be **multiplicative**, if there is a vector z called the recombination vector of dimension $d^* = \sum_{i=1}^{n} \frac{d_i(d_i+1)}{2}$ such that, for any $s, s' \in \mathbb{F}$ and any vectors $y, y' \in \mathbb{F}^l$ satisfying $\varepsilon y^\top = s$ and $\varepsilon y'^\top = s'$, we have*

$$ss' = \langle z, My^\top \otimes My'^\top \rangle$$

It is difficult to determine whether an MSP is multiplicative using definition, therefor, Zhifang Zhang proposed a method to determine whether an MSP is multiplicative or not, which can be carried out less difficult[9].

Denotes $M = (M_1, \cdots, M_l)$, where M_i is the i-th column of M, i.e. $M_i = (m_{1i}, m_{2i}, \cdots, m_{di})^\top$. Let

$$M^* = (M_1 \otimes M_1, \cdots, M_1 \otimes M_l, M_2 \otimes M_2, \cdots, M_2 \otimes M_l, \cdots, M_l \otimes M_l)$$

where M^* is composed of all vectors which are the results of operations of "$\otimes$" of any two columns, which can be the same, of M. Obviously, M^* is $d^* \times \frac{l(l+1)}{2}$ matrix. Similarly, let

$$\varepsilon^* = (v_1v_1, \cdots, v_1v_l, v_2v_2, \cdots, v_2v_l, \cdots, v_lv_l),$$

then the dimension of vector ε^* is $\frac{l(l+1)}{2}$.

Theorem 2.5[9] *An MSP $(\mathbb{F}, M, \varphi, \varepsilon)$ is multiplicative if and only if linear equation system $\varepsilon^* = zM^*$ has solution, and z is the recombination vector.*

Now for a given MSP, we can decide if it is multiplicative or not according to the linear equation system $\varepsilon^* = zM^*$, if the linear equation system $\varepsilon^* = zM^*$ has a solution, then it is; otherwise, it is not. Therefore, how to construct a multiplicative MSP if it is not?

It is well know that Cramer et al. have given a construction[6], which is in general not optimal (in terms of the number of rows and columns) due to

their construction doubles the number of rows, and also increases the number of columns. In addition, Cramer et al.'s construction is applied to any "given" span program, that is, there must be a preparatory given span program, and construct an M-MSP based on it.

Now we propose a new construction for any given access structure according to solve a quadratic equation system. The idea is to transform a non multiplicative MSP to be multiplicative by adding rows to it one-by-one, without changing the access structure. However, the computation complexity of this method is very large due to the solve of the quadratic equations. Even so, our construction is still meaningful, because it may generate an M-MSP with small row size than double increased, moreover, it need not change the column size.

In section 3, we will briefly introduce the construction of linear code for any access structure, as it is a tool for our construction of M-MSP. Then in section 4, we'll first to review Cramer et al.'s construction of M-MSP, then give our construction. Finally, we'll give an example to show how it works.

3 The Optimal Linear Code for Access Structure Γ

Definition 3.1[10] *A **linear code** C of length $n+1$ over $\mathbb{F}_q$ is simply a linear subspace of $\mathbb{F}_q^{n+1}$. If C has dimension k, then C is generated by the rows of a $k \times (n+1)$ matrix $G = (\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_n)$ of rank k, which is called a **generating matrix** of C.*

Now we explain the LSSS introduced by Massey[10]. In this scenario the secret s is an element of $\mathbb{F}_q$ and there are n players and a dealer.

In order to determine the shares, the dealer chooses randomly and uniformly a vector $\mathbf{u} = (u_1, \cdots\cdots, u_k) \in \mathbb{F}_q^k$ such that $s = \mathbf{ug}_0$ and the share is $s_i = \mathbf{ug}_i$ for $i = 1, \cdots, n$.

As stated in [14], if $\mathbf{g}_0, \mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_t}$ are linearly dependent, then the secret can be recovered by first solving the linear equation

$$\mathbf{g}_0 = \sum_{j=1}^{t} x_j \mathbf{g}_{i_j}$$

after finding x_js, the secret can be computed as

$$s = \mathbf{ug}_0 = \sum_{j=1}^{t} x_j \mathbf{ug}_{i_j} = \sum_{j=1}^{t} x_j s_{i_j}.$$

We say the linear code C *realize* access structure Γ, if for any $A \in \Gamma$, we have $\mathbf{g}_0 \in span\{\mathbf{g}_i | i \in A\}$.

Given a linear code C, it uniquely decides an access structure; But for any given access structure, there are many (not unique) linear codes to realize it. However, we care about the linear code which has the shortest length called the optimal linear code. In [11], we have given a method to find the optimal linear code. Now we briefly review it and for the details please read paper [11].

Suppose we are given an access structure $\Gamma = \{S_1, \cdots, S_m\}$, where $S_i \subset [1, n]$ for $1 \leq i \leq m$. we define $m \times (n+1)$ matrix H with the following form:

$$H = \begin{pmatrix} 1 & h_{11} & h_{12} & \cdots & h_{1n} \\ 1 & h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & h_{m1} & h_{m2} & \cdots & h_{mn} \end{pmatrix}$$

where $h_{ij} \neq 0$ if $j \in S_i$, else $h_{ij} = 0$ for $1 \leq i \leq m$, $1 \leq j \leq n$.

Suppose we have found the corresponding adversary structure of Γ which is $R = \{R_1, \cdots, R_l\}$. Define

$$G = \begin{pmatrix} 1 & g_{11} & g_{12} & \cdots & g_{1n} \\ 1 & g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & g_{l1} & g_{l2} & \cdots & g_{ln} \end{pmatrix}$$

where $g_{ij} = 0$ if $j \in R_i$, and g_{ij} is unknown for all $j \notin R_i$.

Theorem 3.2 *There is a linear code to realize access structure $\Gamma = \{S_1, \cdots, S_m\}$ over $\mathbb{F}$ if and only if the system of quadratic equations $GH^\top = 0$ has a solution.*

If the system $GH^\top = 0$ has a solution, then we get an ideal linear code, because the size of each share is equal to the size of the secret. The prove of this theorem is given by [11].

Example 1. Find a linear code over $\mathbb{F}_q^7$ for $\Gamma = \{(1,2,3),(3,4,5),(3,5,6)\}$.

First, we find out the corresponding adversary structure $R = \{(1,2,4,5,6),(1,3,4,6),(2,3,4,6),(1,3,5),(2,3,5)\}$. Then

$$H = \begin{pmatrix} 1 & h_{11} & h_{12} & h_{13} & 0 & 0 & 0 \\ 1 & 0 & 0 & h_{23} & h_{24} & h_{25} & 0 \\ 1 & 0 & 0 & h_{33} & 0 & h_{35} & h_{36} \end{pmatrix}, G = \begin{pmatrix} 1 & 0 & 0 & g_{13} & 0 & 0 & 0 \\ 1 & 0 & g_{22} & 0 & 0 & g_{25} & 0 \\ 1 & g_{31} & 0 & 0 & 0 & g_{35} & 0 \\ 1 & 0 & g_{42} & 0 & g_{44} & 0 & g_{46} \\ 1 & g_{51} & 0 & 0 & g_{54} & 0 & g_{56} \end{pmatrix}.$$

According to theorem 3.2, we need to solve the following system of equations:

$$\begin{cases} 1+h_{13}g_{13}=0 \\ 1+h_{12}g_{22}=0 \\ 1+h_{11}g_{31}=0 \\ 1+h_{12}g_{42}=0 \\ 1+h_{11}g_{51}=0 \\ 1+h_{23}g_{13}=0 \\ 1+h_{25}g_{25}=0 \\ 1+h_{25}g_{35}=0 \\ 1+h_{24}g_{44}=0 \\ 1+h_{24}g_{54}=0 \\ 1+h_{33}g_{13}=0 \\ 1+h_{35}g_{25}=0 \\ 1+h_{35}g_{35}=0 \\ 1+h_{36}g_{46}=0 \\ 1+h_{36}g_{56}=0 \end{cases}$$

It is straightforward to find a general solution: $h_{13}=h_{23}=h_{13}$, $h_{25}=h_{35}$, $g_{31}=g_{51}=-h_{11}^{-1}$, $g_{22}=g_{42}=-h_{12}^{-1}$, $g_{44}=g_{54}=-h_{24}^{-1}$, $g_{25}=g_{35}=-h_{25}^{-1}$, $g_{46}=g_{56}=-h_{36}^{-1}$, $g_{13}=-h_{13}^{-1}$.

If the system $GH^{\top}=0$ has not any solutions, there is an algorithm to generate an optimal linear code(in terms of the length of the code).

Algorithm 3.3: *The optimal linear code for Γ.*

1. Add one column to matrixes G and H respectively, we obtain two matrixes G_1 and H_1 with $n+2$ columns. We emphasis that the new column is the i-th column of G_1 and H_1 respectively, then the i-th column has the same form with the $(i+1)$-th column in G_1 and H_1 respectively for every $1\leq i\leq n$.
   There exists a linear code with length $n+2$ realizing Γ if the system of quadratic equations $G_1H_1^{\top}=0$ has a solution. There is an output which is a linear code realizing Γ.
2. If there does not exist solution of $G_1H_1^{\top}=0$, two columns are added up in matrixes G and H which are changed into two matrixes G_2 and H_2 with $n+3$ columns respectively. New two columns have same forms with two columns or one column of G and H respectively.
   There exists a linear code with length $n+3$ realizing Γ if the system of quadratic equations $G_2H_2^{\top}=0$ has a solution. There is an output which is a linear code realizing Γ.
3. Suppose there does not exist solution of $G_iH_i^{\top}=0$, where matrixes G_i and H_i are obtained by being added up i columns in matrixes G and H respectively. $i+1$ columns are added up in matrixes G and H which are changed into two matrixes G_{i+1} and H_{i+1} with $n+i+2$ columns respectively. New $i+1$ columns have same forms with $i+1$ columns, or i columns, $\ldots$, or one column of G and H respectively.
   There exists a linear code with length $n+i+2$ realizing Γ if the system of quadratic equations $G_{i+1}H_{i+1}^{\top}=0$ has a solution. There is an output which is a linear code realizing Γ.

4. Repeating the step 3, and obtaining a linear code realizing Γ until the system of quadratic equations $G_{i+1}H_{i+1}^{\top} = 0$ has a solution for some i.

As we have mentioned, for any access structure Γ, there are many linear codes to realize it. Similarly, there are many MSPs to realize a given access structure Γ. And we also care about the optimal MSP with the smallest row size and column size, even the optimal M-MSP. In [12], we have proven that there is an 1-1 correspondence between the linear code C with the generated matrix G and MSP $(\mathbb{F}, M, \varphi, \varepsilon)$ realizing the same access structure Γ, and the transform is $M = (G\backslash\mathbf{1})^{\top}$, where $\mathbf{1}$ is the left-most column of G which is the all one column. Therefore, it is the same to use linear code as to use MSP.

4 The Construction of Multiplicative Monotone Span Program

4.1 Cramer et al.'s construction

To our knowledge, the methods to construct M-MSP are not so much. In paper [13], there was an ideal construction for access structure defined by the connectivity of graph; For general access structure, Cramer et al. gave a construction[6], which doubles the row size of the MSP, and also increases the column size.

Cramer et al.'s construction is based on the following idea:

Suppose $(\mathbb{F}, M_1, \varphi_1, \varepsilon_1)$ and $(\mathbb{F}, M_2, \varphi_2, \varepsilon_2)$ are the MSPs computing Boolean functions f_1 and f_2 respectively, where M_1 is $d \times l_1$ matrix and M_2 is $d \times l_2$ matrix. Let

$$\widetilde{M} = \begin{pmatrix} M_1 & 0_{d\times(l_2-1)} \\ c_1 0_{d\times(l_1-1)} & M_2\backslash c_1 \end{pmatrix}$$

$\widetilde{\varphi}$ is defined by: the labels of the first d rows of $\widetilde{M}$ are the same with $(\mathbb{F}, M_1, \varphi_1, \varepsilon_1)$, and the labels of the last d rows are the same with $(\mathbb{F}, M_2, \varphi_2, \varepsilon_2)$. If $M_1^{\top} M_2 = \varepsilon_1^{\top}\varepsilon_2$, then $(\mathbb{F}, \widetilde{M}, \widetilde{\varphi}, \widetilde{\varepsilon})$ is M-MSP computing Boolean function $f_1 \vee f_2$.

Given a access structure Γ, suppose $(\mathbb{F}, M, \varphi, \varepsilon)$ is the MSP computing Boolean function f_Γ. Take the dual access structure $\Gamma^* = \{A \subset P | P - A \notin \Gamma\}$. Let z_0 be a solution of system $zM = \varepsilon$, and $z_1, \cdots, z_{d-l}$ be a set of basis of solution space of system $zM = \mathbf{0}$. Let

$$M_* = (z_0^{\top}, z_1^{\top}, \cdots, z_{d-l}^{\top})$$

Obviously, we have $M^{\top} M_* = \varepsilon^{\top}\varepsilon_*$, where $\varepsilon = (1, 0, \cdots, 0)$, $\varepsilon_* = (1, 0, \cdots, 0)$. Therefor, $(\mathbb{F}, M_*, \varphi, \varepsilon_*)$ is the MSP computing Boolean function f_{Γ^*}.

Thus, we can construct $\widetilde{M}$ according to $(\mathbb{F}, M, \varphi, \varepsilon)$ and $(\mathbb{F}, M_*, \varphi, \varepsilon_*)$, and obtain an MSP $(\mathbb{F}, \widetilde{M}, \widetilde{\varphi}, \widetilde{\varepsilon})$ computing $f_\Gamma \vee f_{\Gamma^*}$. If the adversary structure R meets Q^2 condition, then $\Gamma^* \subseteq \Gamma$, therefor, $f_\Gamma \vee f_{\Gamma^*} = f_\Gamma$.

From the matrix $\widetilde{M}$ we find that the row size becomes double, and the column size also increases.

4.2 our construction

Considering Cramer et al.'s construction doubles the row size, we propose a new construction for any given access structure according to solve a quadratic equations system. The idea is to transform a non M-MSP to be multiplicative by adding rows to it one-by-one, without changing the access structure. We use an algorithm to describe the new construction:

Algorithm 4.1: *Construction of Multiplicative Monotone Span Program*

(1) According to access structure Γ and adversary structure R, we obtain two matrixes of H and G respectively, by executing algorithm 3.3, we obtain the optimal linear code G;
(2) using the correspondence between linear code and MSP, we get an MSP $(\mathbb{F}, M, \varphi, \varepsilon)$ according to $M = (G \backslash \mathbf{1})^{\top}$;
(3) compute M^* and ε^* according to M and ε, if there is a solution of linear equation system $zM^* = \varepsilon^*$, then M has multiplication, output M;
(4) If there is no solution of $zM^* = \varepsilon^*$, add one row to M and denote by M_1, if the new row is the i-th row of M_1, then it has the same form as well as the label with the $(i+1)$-th row of M_1, $1 \leq i \leq n$. Compute M_1^* according to M_1, if there is a solution of system $zM_1^* = \varepsilon^*$, then M_1 has multiplication, output M_1;
(5) If there is no solution of $zM_1^* = \varepsilon^*$, then add two rows to M and denote by M_2, the two new rows have the same forms as well as labels with one or two rows of M. Compute M_2^* according to M_2, if there is a solution of system $zM_2^* = \varepsilon^*$, then M_2 has multiplication, output M_2;
(6) If there is no solution of $zM_i^* = \varepsilon^*$, then add $i+1$ rows to M and denote by M_{i+1}, the $i+1$ new rows have the same forms as well as labels with 1 or $2 \cdots$ or $i+1$ rows of M. Compute M_{i+1}^* according to M_{i+1}, if there is a solution of system $zM_{i+1}^* = \varepsilon^*$, then M_{i+1} is multiplicative, output M_{i+1};
(7) Repeat the step (6) until the system $zM_{i+1}^* = \varepsilon^*$ has solution.

As we know, not all access structures have M-MSP to realize it. Only if the access structure satisfy the Q^2 condition which means for any $A, B \in R$, $A \cup B \neq P$, there is M-MSP to realize it.

Proof. The output(if have) of this algorithm is multiplicative and realizing the same access structure Γ. Because 1) add rows to M is similar to add columns to G which has been proven that it won't change the access structure in paper [11]; 2) The output matrix satisfy the determination theorem 2.5.

Theorem 4.2 *If the adversary structure R does't satisfy the Q^2 condition, there must be two columns in M which is obtained by the step (2) of the above algorithm, such that the result of the two columns' operation "$\otimes$" is zero vector.*

It is easy to account for this result. Actually, G is constructed according to adversary structure R. If R does't satisfy the Q^2 condition, then there are $A, B \in R$ such that $A \cup B = P$. Suppose $g_A = (1, \cdots, g_{si}, \cdots, g_{sj}, \cdots)$ and $g_B = (1, \cdots, g_{ti}, \cdots, g_{tj}, \cdots)$ is the corresponding rows of A and B in G, If $i \notin A$, then $i \in B$, and $g_B = (1, \cdots, \underbrace{0}_{i}, \cdots, g_{tj}, \cdots)$; If $j \notin B$, then $j \in A$, and $g_A =$

$(1, \cdots, g_{si}, \cdots, \underbrace{0}_{j}, \cdots)$. Because $M = (G \backslash \mathbf{1})^{\top}$, and $(\cdots, g_{si}, \cdots, \underbrace{0}_{j}, \cdots)^{\top} \otimes (\cdots, \underbrace{0}_{i}, \cdots, g_{tj}, \cdots) = \mathbf{0}$, therefore, the theorem is established.

Remarks We have several remarks about this algorithm:

1. If R does't satisfy Q^2 condition, then the algorithm has no output, that is, there is no multiplicative MSP, this conclusion is supported by theorem 4.2.
2. When R meets Q^2 condition, the algorithm must have output during finite steps, i.e. there must be a recombination vector z such that $zM_i^* = \varepsilon^*$. Actually, when adding rows to M, it is just an increase of column vectors in coefficient matrix ${M_i^*}^{\top}$ of system $zM_i^* = \varepsilon^*$ according to the construction of M^*, while the rows number won't change. From the knowledge of linear algebra, it only can increase the rows rank of coefficient matrix ${M_i^*}^{\top}$ by adding rows to matrix M, but not reduce. When matrix ${M_i^*}^{\top}$ has full rows rank, then system $zM_i^* = \varepsilon^*$ must has solution.
3. It is reasonable to add rows to M but not columns. If we add columns to M, the only result is, we increase some equations basis on the system of $zM^* = \varepsilon^*$. If the system $zM^* = \varepsilon^*$ has no solution, then won't be the new system which is added some columns to M.
4. The monotone span program obtained by algorithm 4.1 realize the same access structure. This is because: fix matrix M, then matrix G is fixed, and system $GH^{\top} = 0$ become linear equations system on variable h. If $GH^{\top} = 0$ has solution, after we add some rows, which have the same form with some rows of M, to M, that is, add some columns to G as well as H, it won't change the property that linear equations system has solution on variable h, actually, let the added columns in H be zero vector, the new system $GH^{\top} = 0$ still has solution. Finally we obtain a monotone span program realize the same access structure.

4.3 Example

Example 2. Let $P = \{1, 2, 3, 4\}$, $\Gamma = \{(1, 2), (1, 3), (1, 4), (2, 3, 4)\}$, $R = \{(1), (2, 3), (2, 4), (3, 4)\}$, and R is Q^2, the matrixes H and G are:

$$H = \begin{pmatrix} 1 & h_{11} & h_{12} & 0 & 0 \\ 1 & h_{21} & 0 & h_{23} & 0 \\ 1 & h_{31} & 0 & 0 & h_{34} \\ 1 & 0 & h_{42} & h_{43} & h_{44} \end{pmatrix}, G = \begin{pmatrix} 1 & 0 & g_{12} & g_{13} & g_{14} \\ 1 & g_{21} & 0 & 0 & g_{24} \\ 1 & g_{31} & 0 & g_{33} & 0 \\ 1 & g_{41} & g_{42} & 0 & 0 \end{pmatrix}.$$

Since there is a contradiction in the system:

$$\begin{cases} 1 + & h_{11}g_{21} & = 0 \\ 1 + & h_{21}g_{21} & = 0 \\ 1 + & h_{11}g_{31} & = 0 \\ 1 + & h_{31}g_{31} & = 0 \\ 1 + & h_{44}g_{24} & = 0 \\ 1 + h_{31}g_{21} + & h_{34}g_{24} & = 0 \end{cases}$$

so $GH^\top = 0$ has no solution. Add one column to H and G respectively, then we get

$$H_1 = \begin{pmatrix} 1 & h'_{11} & h_{11} & h_{12} & 0 & 0 \\ 1 & h'_{21} & h_{21} & 0 & h_{23} & 0 \\ 1 & h'_{31} & h_{31} & 0 & 0 & h_{34} \\ 1 & 0 & 0 & h_{42} & h_{43} & h_{44} \end{pmatrix}, G_1 = \begin{pmatrix} 1 & 0 & 0 & g_{12} & g_{13} & g_{14} \\ 1 & g'_{21} & g_{21} & 0 & 0 & g_{24} \\ 1 & g'_{31} & g_{31} & 0 & g_{33} & 0 \\ 1 & g'_{41} & g_{41} & g_{42} & 0 & 0 \end{pmatrix}.$$

Now the system of $G_1 H_1^\top = 0$ has solutions. Note that if we add one column in the second, or third, or forth column to H and G respectively, the equations system $G_1 H_1^\top = 0$ has no solution.

We take one solution over F_{11} as follow:

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 4 & 4 & 4 \\ 1 & 0 & -1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

let $M = (G_1 \backslash \mathbf{1})^\top$, then

$$M = \begin{pmatrix} 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 \\ 4 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 \\ 4 & 1 & 0 & 0 \end{pmatrix}.$$

The label ϕ is: $\phi(1) = \phi(2) = 1$, $\phi(3) = 2$, $\phi(4) = 3$, $\phi(5) = 4$; Now we'll determine if the system $zM^* = \varepsilon^*$ has a solution, the coefficient matrix and the extended matrix of system is

$$(M^{*\top}, \varepsilon^{*\top}) = \begin{pmatrix} 0 & 0 & 0 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

Applying elementary row transformation to this matrix we find the system $zM^* = \varepsilon^*$ has no solution. Add one row to M which is the 5-th row in M_1:

$$M_1 = \begin{pmatrix} 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 \\ 4 & 0 & 0 & 1 \\ 4 & 0 & 1 & 0 \\ x & y & 0 & 0 \\ 4 & 1 & 0 & 0 \end{pmatrix}.$$

and H_1 is

$$H_1 = \begin{pmatrix} 1 & h'_{11} & h_{11} & h_{12} & 0 & 0 & 0 \\ 1 & h'_{21} & h_{21} & 0 & h_{23} & 0 & 0 \\ 1 & h'_{31} & h_{31} & 0 & 0 & h'_{34} & h_{34} \\ 1 & 0 & 0 & h_{42} & h_{43} & h'_{44} & h_{44} \end{pmatrix}$$

Let $h'_{34} = h'_{44} = 0$ in order to system $G_1 H_1^\top = 0$ still has solution. Now the coefficient matrix and the extended matrix become

$$(M_1^{*\top}, \varepsilon^{*\top}) = \begin{pmatrix} 0 & 0 & 0 & 5 & 5 & x^2 & 5 & 8x \\ 0 & 0 & 0 & 0 & 0 & xy & 4 & x+4y \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & y^2 & 1 & 2y \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

Take $x = y = 1$, we can verify the system $zM_1^* = \varepsilon^*$ has solution, take one solution $z = (-2, -2, 1, 3, 3, 0, 3, 0)$, then M_1 is multiplicative. □

4.4 The Comparison between Our Construction and Cramer's

From the above example, we can see that our construction has advantage in the size of monotone span program:

\	$Rsize$	$Csize$	$Rsize'$	$Csize'$
$ours$	5	4	6	4
$cramer$	5	4	10	↑

While in the computational complexity of the algorithm, Cramer's construction is easier to realize compared with our's. Therefore, how to reduce the complexity of the algorithm is still a big problem.

5 Conclusion

Multiplicative monotone span program is important to the secure multi-party computation when using linear secret sharing scheme to realize it. In this paper, we give a new construction of multiplicative monotone span program, which can make the size become smaller compare with Cramer et al's, with the tools of linear code. Unfortunately, the computational complexity of our construction is very large, therefore, reducing the complexity will be our future work.

References

1. Pippenger N.,Fischer M.J. Relations among complexity measures. Journal of the ACM,1979,26361-381.
2. Schnorr C.P. The network complexity and the Turing machine complexity of finite functions. Acta Informatica,1976, (7):95-107.
3. Zwick U. Boolean Circuit Complexity, Fall Semester 1994/5, Lecture 3, Tel Aviv, 1994. http://www.cs.tau.ac.il/
4. Cramer R.,Damgard I.,Maurer U. General Secure Multi-Party Computation from any Linear Secret-Sharing Scheme Proc. EUROCRYPTO'00, Springer-Verlag LNCS, vol 1807:316-334.
5. Zhang Z. Secret Sharing and Secure Multiparty Computation. Academy of Mathematics and Systems Science PhD thesis, 2007
6. Cramer R.,Damgard I.,Maurer U. General Secure Multi-Party Computation from any Linear Secret-Sharing Scheme[C]//Proc.EUROCRYPT'00. 2000:316-334.
7. M.Karchmer,A.Wigderson. On span programs [J]. In Proceedings of the Eighth Annual Structure in Complexity Theory Conference,IEEE,1993: 102-111.
8. Radomirović Saša. Investigations Into Span Programs With Multiplication. Institute for theoretical Computer Science Diploma Thesis, ETH Zürich, 1998.
9. Zhang Z. Multiplicative Monotone Span Program. Journal of Graduate University of Chinese Academy of Sciences, 2006,236827-832.
10. J.L.Massey. Minimal codewords and secret sharing [J]. Proc. 6th Joint Swedish-Russian Workshop on Information Theory, August 22-27, 1993: 276-279.
11. C. Tang, S. Gao and C. Zhang. The Optimal Linear Secret Sharing Schemes for any Given Access Structure. Journal of Systems Science & Complexity, Vol.26, No.4, 2013, pp.634-649.
12. Yuenai Chen, Chunming Tang, Shuguang Dai, The Greatest Lower Bound of Monotone Span Programs. CHINACRYPT2011:159-163
13. Zhang Z. Monotone span program with multiplication. Graduate School of the Chinese Academy of Sciences, 2006,6(23):827-832.
14. C.Ding, J.Yuan. Covering and secret sharing with linear codes. In Discrete Mathematics and Theoretical Computer Science(Lecture Notes in Computer Science), 2003,2731:11-25.