Optimized Implementation of General Secret Sharing Scheme

Lein Harn and Ching-Fang Hsu*

Department of Computer Science Electrical Engineering

University of Missouri-Kansas City

Kansas City, 64110, MO, USA

Email: harnl@umkc.edu

* Computer School

Central China Normal University

430079, Wuhan, China

Email: cherryjingfang@gmail.com

May 14, 2014

<Abstract>

Secret sharing (SS) is one of the most important cryptographic primitives used for data outsourcing. The (t, n) SS was introduced by Shamir and Blakley separately in 1979. The secret sharing policy of the (t, n) threshold SS is far too simple for many applications because it assumes that every shareholder has equal privilege to the secret or every shareholder is equally trusted. Ito et al. introduced the concept of a general secret sharing scheme (GSS). In a GSS, a secret is divided among a set of shareholders in such a way that any "qualified" subset of shareholders can access the secret, but any "unqualified" subset of shareholders cannot access the secret. The secret access structure of GSS is far more flexible than threshold SS. In this paper, we propose an optimized implementation of GSS. Our proposed scheme first uses Boolean logic to derive two important subsets, one is called Min which is the *minimal positive access subset* and the other is called Max which is the *maximal negative access subset*, of a given general secret sharing structure. Then, conditions of parameters of a GSS are established based on these two important subsets. Furthermore, integer linear/non-linear programming is used to optimize the size of shares of a GSS. The complexity of linear/non-linear programming is $O(n)$, where n is the number of shares generated by the dealer. This proposed design can be applied to implement GSS based on any classical SS. We use two GSSs, one is based on Shamir's weighted SS (WSS) using linear polynomial and the other is based on Asmuth-Bloom's SS using Chinese Remainder Theorem (CRT), to demonstrate our design. In comparing with existing GSSs, our proposed scheme is more efficient and can be applied to all classical SSs.

Keywords: General secret sharing; Chinese remainder theorem; Secret sharing policy; Monotone function; Integer optimization; Minimal positive access subset; Maximal negative access subset.

1. Introduction

The (t,n) SS was introduced by Shamir [1] and Blakley [2] independently in 1979. In a (t,n) SS, a dealer divides a secret *s* into *n* shares and *s* is shared among a set of *n* shareholders, $U=\{U_1,U_2,...,U_n\}$, in such a way that any *t* or more than *t* shareholders can reconstruct the secret *s*; while fewer than *t* shareholders cannot reconstruct the secret *s*. In Shamir's (t,n) SS, a dealer generates *n* shares based on a linear polynomial having degree $t-1$. Secret reconstruction is based on Lagrange interpolating formula using any *t* or more than *r* private shares. Shamir's (t,n) SS is unconditionally secure. There are other types of SSs. For example, Blakely's scheme [2] is based on Geometry, Mignotte's scheme [3] and Asmuth-Bloom's scheme [4] are based on Chinese remainder theorem (CRT), and McEliece et al. scheme [5] is based on Reed-Solomon codes.

The weighted (t,n) secret sharing scheme (WSS) was originally proposed by Shamir [1]. In a WSS, each share of a shareholder has a positive weight. The secret can be recovered if the overall weight of shares is equal to or larger than the threshold; but the secret cannot be recovered if the overall weight of shares is smaller than the threshold value. In fact, Shamir's (t,n) SS is a special type of WSSs in which the weight of all shares is the same. One simple way to implement a WSS using Shamir's (t,n) SS is to assign multiple shares to each shareholder according to his/her weight. There are some papers to discuss properties and characteristics of a WSS. For example, Morillo et al. [6] discussed the property of information rate of a WSS. Beimel et al. [7] characterized all weighted threshold access structures that are ideal. They showed that a weighted threshold access structure is ideal if and only if it is a hierarchical threshold access structure, or a tripartite access structure, or a composition of two ideal weighted threshold access structures that are defined on smaller sets of users.

The secret sharing policy of the (*t*, *n*) threshold SS is far too simple for many applications because it assumes that every shareholder has equal privilege to the secret. Complicated sharing policies, in which shareholders have different privileges, can also be realized by other general SSs [8, 9]. Ito et al. [8] have introduced the concept of general secret sharing (GSS). In a GSS, a secret is divided among a set of shareholders, U, in such a way that any "qualified" subset of U can access the secret, but any "unqualified" subset of U cannot access the secret. Benaloh et al. [9] have shown that there is a correspondence between the set of general secret sharing functions and the set of monotone functions. Saito et al. [8] have introduced the *cumulative array technique* and used it to construct a GSS based on monotone access structures. In their scheme, multiple shares are needed for each shareholder. Benaloh et al. [9] have represented the access instances using formulae. According to monotone access instances of a secret, a set of formulae on a set of variables is used to share the secret. Their scheme shares the same problem as scheme proposed by Saito et al. That is, multiple shares are needed for each shareholder. Harn et al. [10] have proposed an *l*-span generalized SS in which the shares can be repeatedly used for *r* times to reconstruct *r* different secrets. However, the security of their scheme is based on RSA assumption. The *cumulative map* is a simple realization of the multiple assignment map based on a (t,n) SS [11, 12, 13] which utilized a GSS based on a WSS. However, the GSSs constructed by the cumulative map are inefficient. Iwamoto et al. [14] proposed an optimal multiple assignments based on integer programming to optimize the size of shares. The complexity of solving an integer programming problem is related to the cardinality of the constraint variables set. However, the number of variables in the integer programming is $O(2^n)$, where n is the number of shares generated by the dealer. Li et al. [15] proposed a method to reduce the number of constraint variables in the integer programming problem. Srinathan et al. [16] have considered the problem of non-perfect secret sharing (NSS) over general secret sharing policy and defined generalized monotone span programs (MSP) to facilitate the design of NSS schemes. However, their approach captures and addresses only NSS schemes that are linear. In 2007, Xu et al. [17] have studied new operations on secret sharing policy to construct large MSPs from small MSPs and proposed new design of GSS. Recently, Guo et al. [18] have proposed a scheme based on the key-lock-pair mechanism. The share of each shareholder is a pair of column vectors corresponding to the key-lock-pair. However, the number of elements of column vectors is determined by the number of terms in the secret access structure. Iftene [19] has proposed a GSS using CRT for special types of general access structures such as the compartmented and the weighted threshold SSs.

In this paper, we propose an optimized implementation of GSS. Our proposed scheme first uses Boolean logic to derive two important sets, one is called Min which is the *minimal positive access subset* and the other is called Max which is the *maximal negative access subset*, of a given general secret sharing structure. Then, conditions of parameters of a GSS are established based on these two important subsets. Furthermore, integer linear/non-linear programming is used to optimize the size of shares of a GSS. The complexity of linear/non-linear programming is $O(n)$, where n is the number of shares generated by the dealer. This proposed design can be applied to implement GSS based on any classical SS. We use two GSSs, one is based on Shamir's weighted SS (WSS) using linear polynomial and the other is based on Asmuth-Bloom's SS using Chinese Remainder Theorem (CRT), to demonstrate our design. In comparing with existing GSSs, our proposed scheme is more efficient and can be applied to all classical SSs. Here, we summarize the contributions of our paper.

- We propose an optimized design to implement a GSS based on any classical SS.
- For any given general secret sharing policy, Boolean logic is used to derive Min and Max, then parameters of a GSS are determined based on Min and Max.

- Integer linear/non-linear programming can be used to minimize the size of shares. The complexity in the integer/non-linear programming is $O(n)$.
- Two GSSs, one is polynomial-based Shamir's WSS and the other is CRT-based Asmuth-Bloom's SS, are used to demonstrate our design.
- Our design can be generalized to implement a GSS based on any classical SS.

The rest of this proposal is organized as follows. In the next section, we introduce some preliminaries including CRT, Mignotte's and Asmuth-Bloom SSs based on CRT. In Section 3, we introduce our design to implement a GSS. In Section 4, we demonstrate the optimized implementation of a GSS based on Shamir's WSS using linear polynomial. In Section 5, we demonstrate the optimized implementation of a GSS based on Asmuth-Bloom's SS using CRT. Conclusion is given in Section 6.

2. Preliminaries

In this section, we introduce some preliminaries including CRT, Mignotte's and Asmuth-Bloom schemes based on CRT.

2.1 Chinese Remainder Theorem (CRT) [20]

Given following system of equations as

$$\begin{aligned} x &= s_1 \bmod p_1; \\ x &= s_2 \bmod p_2; \\ &\,\cdot \\ &\,\cdot \\ &\,\cdot \\ x &= s_t \bmod p_t, \end{aligned}$$

there is one unique solution as $x = \sum_{i=1}^{t} \frac{N}{p_i} \cdot y_i \cdot s_i \bmod N,$ *where* $\frac{N}{p_i} \cdot y_i \bmod p_i = 1,$ *and* $N = p_1 \cdot p \ \cdot \ldots \cdot p_t,$ *if all moduli are pairwise coprime (i.e.,* $\gcd(p_i, p_j) = 1,$ *for every* $i \neq j$*).*

CRT has been used in RSA decryption to speed-up the decryption process. With the knowledge of prime decomposition of RSA composite integer and using CRT, the complexity of RSA decryption is reduced by a factor of $\frac{1}{4}$. CRT can also be used in the SS. Each of the shares is represented in a congruence, and the solution of the system of congruences using CRT is the secret to be recovered. SS based on CRT uses, along with CRT, a special sequence of integers that guarantee the impossibility of recovering the secret from a set of shares with less than a certain cardinality. In the next subsections, we will review two most well-known SSs based on CRT.

2.2 Review of Mignotte's (t,n) *SS [3]*

Share generation: A sequence consists of pairwise coprime positive integers, $p_1 < p_2 < \ldots < p_n$, with $p_{n-t+2} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t$, where p_i is the public information associated with each shareholder, U_i. For this given sequence, the dealer chooses the secret s as an integer in the set $Z_{p_{n-t+2}\cdot\ldots\cdot p_n, p_1\cdot p_2\cdot\ldots\cdot p_t}$ (i.e., $Z_{p_{n-t+2}\cdot\ldots\cdot p_n, p_1\cdot p_2\cdot\ldots\cdot p_t}$ is referred to the range $(p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n, p_1 \cdot p_2 \cdot \ldots \cdot p_t)$). We call the range, $Z_{p_{n-t+2}\cdot\ldots\cdot p_n, p_1\cdot p_2\cdot\ldots\cdot p_t}$, ***secure secret range,*** in which the secret should be selected by the dealer in order to enforce the secret sharing policy.

Share for the shareholder, U_i, is generated as $s_i = s \bmod p_i, i = 1,2,\ldots,n$. s_i is sent to shareholder, U_i, secretly.

{Remark 1} *The numbers in the secure secret range,* $Z_{p_{n-t+2}\cdot\ldots\cdot p_n, p_1\cdot p_2\cdot\ldots\cdot p_t}$, *are integers upper bounded by* $p_1 \cdot p_2 \cdot \ldots \cdot p_t$, *which is the smallest product of any* t *moduli, and lower bounded by* $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n$, *which is the largest product of any* $t-1$ *moduli. The secret,* s, *selected in this range can ensure that (a) the secret can be recovered with any* t *or more than* t *shares (i.e., the product of their moduli must be either equal to or larger than the upper bound,* $p_1 \cdot p_2 \cdot \ldots \cdot p_t$*), and (b) the secret cannot be obtained with fewer than* t *shares (i.e., the product of their moduli must be either equal to or smaller than the lower bound,* $p_{n-t+2} \cdot \ldots \cdot p_n$*). Thus, the secret of a* (t,n) *threshold SS should be selected from the secure secret range. In a GSS, it has a general secret sharing policy. The secure secret range should be determined according to the secret sharing policy such that the policy can be enforced. The secure secret range specified in a* (t,n) *threshold SS is no longer valid for a GSS having different secret sharing policy.*

Secret reconstruction: Given t distinct shares, for example, $\{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$, the secret s can be reconstructed by solving the following system of equations as

$$\begin{aligned} x &= s_{i_1} \bmod p_{i_1}; \\ x &= s_{i_2} \bmod p_{i_2}; \\ &. \\ &. \\ &. \\ x &= s_{i_t} \bmod p_{i_t}. \end{aligned}$$

Using the standard CRT, a unique solution x is given as $x = \sum_{r=1}^{t} \frac{N}{p_{i_r}} \cdot y_{i_r} \cdot s_{i_r} \bmod N$, where $N = p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_t}$, and $\frac{N}{p_{i_r}} \cdot y_{i_r} \bmod p_{i_r} = 1$.

We want to point out that Mignotte's (t,n) threshold SS is not a perfect SS since any share substantially decreases the entropy of the secret.

2.3 Review of Asmuth-Bloom (t,n) ***SS [4]***

Share generation: In Asmuth-Bloom (t,n) SS, the dealer selects p_0 and a sequence of pairwise coprime positive integers, $p_1 < p_2 < ... < p_n$, such that $p_0 \cdot p_{n-t+2} \cdot ... \cdot p_n < p_1 \cdot p_2 \cdot ... \cdot p_t$, and $\gcd(p_0, p_i) = 1, i = 1,2,...,n,$ where p_i is the public information associated with each shareholder, U_i. For this given sequence, the dealer chooses the secret s as an integer in the set Z_{p_0}. The dealer selects an integer, α, such that $s + \alpha p_0 \in Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot ... \cdot p_n, p_1 \cdot p_2 \cdot ... \cdot p_t}$. We want to point out that the value, $s + \alpha p_0$, needs to be in the *secure secret range*, $Z_{p_{n-t+2} \cdot p_{n-t+3} \cdot ... \cdot p_n, p_1 \cdot p_2 \cdot ... \cdot p_t}$; otherwise, the value, $s + \alpha p_0$, can be obtained with fewer than t shares. However, in the original paper [4], it specifies that the value, $s + \alpha p_0$, is in the set, $Z_{p_1 \cdot p_2 \cdot ... \cdot p_t}$. This range is different from the *secure secret range*. In other words, if $s + \alpha p_0$ is selected to be smaller than the lower bound of the *secure secret range* (i.e., but it is still in the range, $Z_{p_1 \cdot p_2 \cdot ... \cdot p_t}$), then the value, $s + \alpha p_0$, can be obtained with fewer than t shares. It is obvious that this situation violates one of the security requirements of the (t,n) SS.

Share for the shareholder, U_i, is generated as $s_i = s + \alpha p_0 \bmod p_i$, and s_i is sent to shareholder, U_i, secretly, for $i = 1,2,...,n$.

Secret reconstruction: Given a subset of t distinct shares, for example, $\{s_{i_1}, s_{i_2}, ..., s_{i_t}\}$, the secret s can be reconstructed by solving the following system of equations as

$$\begin{aligned} x &= s_{i_1} \ mod\ p_{i_1}; \\ x &= s_{i_2} \ mod\ p_{i_2}; \\ &. \\ &. \\ &. \\ x &= s_{i_t} \ mod\ p_{i_t}. \end{aligned}$$

Using standard CRT, a unique solution x is given as $x = \sum_{r=1}^{t} \frac{N}{p_{i_r}} \cdot y_{i_r} \cdot s_{i_r} \bmod N$, where $N = p_{i_1} \cdot p_{i_2} \cdot ... \cdot p_{i_t}$, and $\frac{N}{p_{i_r}} \cdot y_{i_r} \bmod p_{i_r} = 1$. Then, the secret s can be recovered by computing $s = x \bmod p_0$.

Asmuth and Bloom showed that the entropy of the secret decreases "not too much" when $t-1$

shares are known. Interest readers can refer to the original paper [4] for detailed discussion. Asmuth-Bloom's SS can be generalized to take more than t shares in the secret reconstruction. For example, when there are j (i.e., $t < j \le n$) shareholders with their shares, $\{s_1, s_2, ..., s_j\}$, participated in the secret reconstruction, the secret, s, can be reconstructed using standard CRT to find a unique solution x for the system of j equations.

3. Proposed Scheme

In the following, we give a definition of GSS.

Definition 1. ***General secret sharing (GSS).*** *A secret,* s, *is shared according to a given secret sharing policy by a group of* n *shareholders* $U = \{U_1, U_2, ..., U_n\}$. *A GSS is a method of breaking the secret,* s, *into* n *shares,* $s_1, s_2, ..., s_n$, *with* s_i *secretly distributed to* U_i *such that*

(1) if $A \subseteq U$ *is a qualified subset of shareholders, called positive access instance, according to the secret sharing policy, then the secret,* s, *can be reconstructed from shares,* $\{s_i | u_i \in A\}$.

(2) if $A \subseteq U$ *is not a qualified subset of shareholders, called negative access instance, according to the secret sharing policy, then the secret,* s, *cannot be reconstructed from shares,* $\{s_i | u_i \in A\}$.

The set F of all positive access instances is called *positive access structure* of the secret sharing policy and the set N of all negative access instances is called *negative access structure* of the secret sharing policy. Suppose the positive access structure of a given sharing policy is F. The corresponding negative access structure is N with $N \cup F = 2^U$ and $N \cap F = \phi$, where 2^U is the power set of the shareholder set $U = \{U_1, U_2, ... U_n\}$, the symbol "$\cap$" is logic AND, and the symbol "$\cup$" is logic OR.

The positive access structure of a GSS has the monotone increasing property. That is, if $B \in F$, and $B \subseteq C \in U$, then $C \in F$. Similarly, the negative access structure which is the logically complement of the positive access structure of a GSS has the monotone decreasing property. That is, if $B \in N$, and $C \subseteq B \subseteq U$, then $C \in N$.

Definition 2. ***Minimal positive access subset and maximal negative access subset.*** *A secret is shared by a set of shareholders according to a given secret sharing policy, where F is the positive access structure and N is the negative access structure, such that*

(1) the subset, $Min \in F$, *is the minimal positive access subset if for every* $C \subseteq Min$ *but* $C - \{U_i | U_i \in C\} \notin F$; *and*

(2) the subset, $Max \in N$, *is the maximal negative access subset if for every* $C \subseteq Max$ *but* $C \cup \{U_i | U_i \in U - C\} \notin N$.

Both Min and Max can be used to characterize the secret sharing policy completely. For a GSS, if the secret can be recovered by shareholders specified in Min, the secret can be recovered by any positive access instance. Similarly, if the secret cannot be recovered by shareholders specified in $Miax$, the secret cannot be recovered by any negative access instance. One contribution of our proposed scheme is to determine both Min and Max of a general secret sharing policy using Boolean algebra.

Let us assume that there are four shareholders, $U=\{A,B,C,D\}$. If the positive access structure is $F=\{(B\cap D)\cup(B\cap C\cap D)\cup(A\cap B\cap D)\cup(A\cap B\cap C\cap D)\}$, then $Min=\{B\cap D\}$. The 4 positive access instances in Min can be represented using the Karnaugh map by marking cells with “1s”. The Boolean function, $f=(A'\cap B\cap C'\cap D)\cup(A'\cap B\cap C\cap D)\cup(A\cap B\cap C'\cap D)\cup(A\cap B\cap C\cap D)=B\cap D$, corresponds to these “1s” in the Karnaugh map.

AB\CD	00	01	11	10
00				
01		1	1	
11		1	1	
10				

Similarly, If the negative access structure is $N=\{(\varphi)\cup(B)\cup(D)\cup(B\cap D)\}$, then $Miax=\{B\cap D\}$. The 4 negative access instances in Max can be represented using the Karnaugh map by marking cells with “0s”. The Boolean function, $f'=(A'\cap B'\cap C'\cap D')\cup(A'\cap B'\cap C'\cap D)\cup(A'\cap B\cap C'\cap D')\cup(A'\cap B\cap C'\cap D)=A'\cap C'$, corresponds to these “0s” in the Karnaugh map.

AB\CD	00	01	11	10
00	0	0		
01	0	0		
11				
10				

In the following discussion, we consider a general positive access structure, $F=\{(B\cap C\cap D)\cup(A\cap B)\cup(A\cap B\cap D)\cup(A\cap B\cap C\cap D)\cup(A\cap B\cap C)\}$. The Karnaugh map corresponding to this positive access structure and negative access structure is given below.

AB\CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	1	1	1	1
10	0	0	0	0

The Boolean functions, $f=(A'\cap B\cap C\cap D)\cup(A\cap B\cap C'\cap D')\cup(A\cap B\cap C'\cap D)\cup(A\cap B\cap C\cap D)\cup(A\cap B\cap C\cap D')),$ and $f'=\{(A'\cap B'\cap C'\cap D')\cup(A'\cap B'\cap C'\cap D)\cup(A'\cap B'\cap C\cap D)\cup(A'\cap B'\cap C\cap D')\cup(A'\cap B\cap C'\cap D')\cup(A'\cap B\cap C'\cap D)\cup(A'\cap B\cap C\cap D')\cup(A\cap B'\cap C'\cap D')\cup(A\cap B'\cap C'\cap D)\cup(A\cap B'\cap C\cap D)\cup(A\cap B'\cap C\cap D')),$ represent the positive access structure and negative access structure separately, where A' is the logically complement of the Boolean variable, $A.$ Boolean algebra can be used to simplify Boolean functions to obtain $f=(A\cap B)\cup(B\cap C\cap D)$ and $f'=(B')\cup(A'\cap D')\cup(A'\cap C').$ Thus, we obtain $Min=\{(A\cap B)\cup(B\cap C\cap D)\}$ and $Max=\{(A\cap C\cap D)\cup(B\cap C)\cap(B\cap D)\}.$

In our proposed scheme, we first determine both Min and Max of a secret sharing policy and then establish conditions of parameters of a GSS based on both Min and Max.

3.1 Deriving Min and Max from a general secret sharing policy

We use examples to demonstrate our scheme.

Example 1. Assume that there are four shareholders, $U=\{A,B,C,D\}$, and the secret sharing policy can be expressed by the following positive access structure, $F=\{(A\cap B\cap C)\cup(A\cap B\cap D)\cup(B\cap C\cap D)\cup(A\cap C\cap D)\cup(A\cap B\cap C\cap D)\}.$ The Boolean functions, $f=(A\cap B\cap C)\cup(A\cap B\cap D)\cup(A\cap B\cap D)\cup(A\cap C\cap D)\cup(A\cap B\cap C\cap D)$ and $f'=(A'\cap B')\cup(A'\cap C')\cup(A'\cap D')\cup(B'\cap C')\cup(B'\cap D')\cup(C'\cap D'),$ correspond to the positive access structure and negative access structure, respectively. The Karnaugh map corresponding to this positive access structure and negative access structure is given below.

AB\CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0

Karnaugh map simplification is used to derive $Min=\{(A\cap B\cap C)\cup(A\cap B\cap D)\cup(B\cap C\cap D)\cup(A\cap C\cap D)\}$ and $Max=\{(C\cap D)\cup(B\cap D)\cup(B\cap C)\cup(A\cap D)\cup(A\cap C)\cup(A\cap B)\}$.

In this example assume that each shareholder has the same weight. Since $Min=\{(A\cap B\cap C)\cup(A\cap B\cap D)\cup(B\cap C\cap D)\cup(A\cap C\cap D)\}$, it specifies that any 3 or more than 3 shareholders can recover the secret. Furthermore, since $Max=\{\ (C\cap D)\cup(B\cap D)\cup(B\cap C)\cup(A\cap D)\cup(A\cap C)\cup(A\cap B)\}$, it specifies that any 2 or fewer than 2 shareholders cannot recover the secret. In other words, the secret sharing policy of this example is a $(3,4)$ SS. This example demonstrates that our proposed scheme can be used to derive both Min and Max of a threshold secret sharing scheme.

Example 2. Assume that there are four shareholders, $U=\{A,B,C,D\}$, and the secret sharing policy can be expressed by the following positive access structure, $F=\{(A\cap B)\cup(A\cap B\cap C)\cup(B\cap C)\cup(B\cap C\cap D)\cup(A\cap C\cap D)\cup(A\cap B\cap C\cap D)\}$. The Boolean functions,, $f=(A\cap B)\cup(A\cap B\cap C)\cup(B\cap C)\cup(B\cap C\cap D)\cup(A\cap C\cap D)\cup(A\cap B\cap C\cap D)$ and $f'=(B'\cap C')\cup(B'\cap D')\cup(A'\cap B')\cup(A'\cap C')$, correspond to the positive access structure and negative access structure, respectively. The Karnaugh map corresponding to this positive access structure and negative access structure is given below.

AB\CD	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	1	1	1
10	0	0	1	0

Karnaugh map simplification is used to derive $Min=\{(A\cap B)\cup(B\cap C)\cup(A\cap C\cap D)\}$ and $tMax=\{(A\cap C)\cup(A\cap D)\cup(B\cap D)\cup(C\cap D)\}$. In Sections 4 and 5, Min and Max of this example will be used to determine optimal size of shares of a GSS.

3.2. Establishing conditions of parameters

The outline of our proposed design is to use both Min and Max of a given secret sharing policy to determine the parameters of a GSS. Since in a secret sharing policy, Min includes all minimal positive access instances and Max includes all negative access instances, parameters satisfying Min ensure all positive access instances in F to recover the secret and parameters satisfying Max ensure all negative access instances in N not to recover the secret. In other words, our proposed scheme greatly simplifies the implementation of a GSS by only considering parameters satisfying both Min and Max. The secret of a general secret sharing policy needs to be recovered by the shareholders specified in Min and not to be recovered by shareholders specified in Max. Conditions of parameters of a GSS satisfying both Min and Max are established and solved which can enforce the secret sharing policy. In the following, we use examples to demonstrate our scheme.

<u>3.2.1 Implementing a GSS based on Shamir's WSS</u>

Example 3. We want to implement the secret sharing policy in Example 1 based on Shamir's WSS. We first give the following definition of a weighted (t,n) SS.

Definition 3. ***Weighted*** (t,n) ***secret sharing scheme (WSS).*** *A secret,* s, *is shared by a group of* n *shareholders* $U=\{U_1,U_2,...,U_n\}$ *having a threshold* t. *A WSS is a method of breaking the secret,* s, *into* n *shares,* $s_1,s_2,...,s_n$, *with* s_i *secretly distributed to* U_i *such that*

(1) if $A\subseteq U$ *is a qualified subset of shareholders, having the overall weight of shares equal to or larger than the threshold* t, *then the secret,* s, *can be reconstructed from shares.*

(2) if $A\subseteq U$ *is not a qualified subset of shareholders, having the overall weight of shares smaller than the threshold* t,*, then the secret,* s, *cannot be reconstructed from shares.*

Shamir's (t,n) SS is a special type of WSSs in which the weight of all shares is the same. One simple way to implement a WSS using Shamir's (t,n) SS is to assign multiple shares to each shareholder according to his/her weight.

In Shamir's WSS, there are different weights of shares. The secret can be recovered if the overall weight of shares is equal to or more than the threshold and the secret cannot be recovered if the overall weight of shares is less than the threshold. Therefore, following parameters of Shamir's WSS need to be determined in order to implement a general secret sharing policy, (a) the threshold of the secret of a WSS, and (b) the weights of shareholders.

Each item in Min/Max is a positive/negative access instance which may involve multiple shareholders. For example, if $U_A \cap U_B$ is a positive access instance in Min, it means that U_A and U_B together can recover the secret. To satisfy this requirement, the additive sum of weights of their shares, w_A+w_B, needs to be larger than or equal to the threshold, t, of the secret. When both U_A and U_B present their shares, the additive sum of weights of their shares, w_A+w_B, is larger than or equal to the threshold of the secret so the secret can be recovered from their shares. On the other hand, if $U_C \cap U_D$ is a negative access instance in Max, it means that U_C and U_D together cannot recover the secret. To satisfy this requirement, the additive sum of weights of their shares, w_C+w_D, needs to be smaller than the threshold of the secret. When both U_C and U_D present their shares, their additive sum, w_C+w_D, is smaller than the threshold so the secret cannot be recovered from their shares. If we use $min\{Min\}$ to represent the minimal additive sum of weights in Min and $max\{Max\}$ to represent the maximal additive sum of weights in Max, the threshold, t, of the secret should be selected from $max\{Max\} < t \le min\{Min\}$ in order to satisfy both security requirements. Parameters of a GSS need to be determined to satisfy this inequality relation in order to enforce the secret sharing policy exactly.

If we assume that the weight of all shares is 1. Then, from Example 1, $Min = \{(A \cap B \cap C) \cup (A \cap B \cap D) \cup (B \cap C \cap D) \cup (A \cap C \cap D)\}$ and $Max = \{(C \cap D) \cup (B \cap D) \cup (B \cap C) \cup (A \cap D) \cup (A \cap C) \cup (A \cap B)\}$. We can obtain $min\{Min\} = 3$ and $max\{Max\} = 2$. Then, from the condition, $max\{Max\} < t \le min\{Min\}$,we have $t = 3$. In fact, this is a $(3,4)$ SS in which the threshold is 3 and any 3 or more than 3 shares can recover the secret; but fewer than 3 shares cannot recover the secret. In this example, we demonstrate that both Min and Max can be used to determine the parameters (i.e., the threshold) of a secret sharing policy (i.e., a threshold SS). In Section 4, we will demonstrate our scheme for a general secret sharing policy.

3.2.2 Implementating a GSS based on Asmuth-Bloom SS

In CRT-based Asmuth-Bloom SS, each shareholder has different modulus. The secret can be recovered if the product of moduli of shareholders is larger than the lower bound of the *secure secret range* and the secret cannot be recovered if the product of moduli of shareholders is smaller than *secure secret range*. Therefore, following parameters of a GSS need to be determined, (a) the *secure secret range*, and (b) the moduli of shareholders.

Each item in Min/Max is a positive/negative access instance which may involve multiple shareholders. For example, if $U_A \cap U_B$ is a positive access instance in Min, it means that U_A and U_B together can recover the secret. To satisfy this requirement, the shifted value, $s + \alpha p_0$, needs to be smaller than their moduli product, $p_A p_B$. When both U_A and U_B present their shares, their moduli product, $p_A p_B$, is larger than the shifted value so the shifted value can be recovered from their shares. On the other hand, if $U_C \cap U_D$ is a negative access instance in Max, it means that U_C and U_C together cannot recover the secret. To satisfy this requirement, the shifted value needs to be larger than their moduli product, $p_C p_D$. When both U_C and U_D present their shares, their moduli product $p_C p_D$ is smaller than the shifted value so the shifter value cannot be recovered from their shares. If we use $min\{Min\}$ to represent the minimal moduli product in Min and $max\{Max\}$ to represent the maximal moduli product in Max, the shifted value should be selected from $max\{Max\} < s + \alpha p_0 < min\{Min\})$ in order to satisfy both security requirements. Parameters of a GSS need to be determined to satisfy this inequality relation in order to enforce the secret sharing policy exactly. We use the following example to demonstrate our scheme.

Example 4. We want to implement the secret sharing policy in Example 1 based on Asmuth-Bloom SS. From Example 1, we have $Min=\{(A\cap B\cap C)\cup(A\cap B\cap D)\cup(B\cap C\cap D)\cup(A\cap C\cap D)\}$ and $Max=\{(C\cap D)\cup(B\cap D)\cup(B\cap C)\cup(A\cap D)\cup(A\cap C)\cup(A\cap B)\}$. Recall that the secret sharing policy specified in this example is a $(3,4)$ SS. Any 3 or more than 3 shareholders can recover the secret, but fewer than 3 shareholders cannot recover the secret s. The moduli products in Min of Asmuth-Bloom $(3,4)$ SS can be represented as $\{p_{r_i}\cdot p_{r_{i+1}}.p_{r_{t+2}} \mid \forall p_{r_i},p_{r_{i+1}},p_{r_{t+2}}\in\{p_A,p_B,p_C,p_D\}\}$ and the moduli products in Max can be represented as $\{p_{r_i}p_{r_{i+1}} \mid \forall p_{r_i},p_{r_{i+1}}\in\{p_A,p_B,p_C,p_D\}\}$. To satisfy the secret sharing policy, the *secure secret range* can be specified as $max\{Max\}<s+\alpha p_0<min\{Min\}$. However, if all moduli satisfy $p_A<p_B<p_C<p_D$ such that $p_0\cdot p_C\cdot p_D<p_A\cdot p_B\cdot p_C$, then we have $min\{Min\}=p_A\cdot p_B\cdot p_C$ and $max\{Max\}=p_C\cdot p_D$. Thus, we have $p_C\cdot p_D<s+\alpha p_0<p_A\cdot p_B\cdot p_C$, which is the *secure secret range* as we described in Section 2.3.

In this example, we demonstrate that both Min and Max can be used to determine the parameters (i.e., the *secure secret range*) of a secret sharing policy (i.e., a threshold SS). In Section 5, we will demonstrate our scheme for a general secret sharing policy.

4. Optimized Implementation of GSS Based on Shamir's WSS

In this section, we show how to use linear integer programming to optimize the size of shares of a GSS.

4.1 Proposed Scheme

We use following example to demonstrate our scheme.

Example 5. We want to implement the secret sharing policy in Example 2 based on Shamir's WSS. From Example 2, we have $Min=\{(A\cap B)\cup(B\cap C)\cup(A\cap C\cap D)\}$ and $Max=\{(A\cap C)\cup(A\cap D)\cup(B\cap D)\cup(C\cap D)\}$. Assume that the weights of shares of shareholders, A, B, C and D are $w_A,w_B,w_C,w_D,$ respectively. If we use $min\{Min\}$ to represent the minimal additive sum of weights in Min and $max\{Max\}$ to represent the maximal additive sum of weights in $Max,$ the threshold, $t,$ of the secret should be selected from $max\{Max\}<t\le min\{Min\}$. In other words, the upper bound of the threshold is $min\{w_A+w_B,w_B+w_C,w_A+w_C+w_D\}$ and the lower bound of the threshold is $max\{w_A+w_C,w_A+w_D,w_B+w_D,w_C+w_D\}$. Since the additive sum of weights of shareholders of each item in Min should be larger than the additive sum of weights of shareholders of each item in $Max,$ we can establish following inequality conditions.

From $w_A+w_B>\{w_A+w_C,w_A+w_D,w_B+w_D,w_C+w_D\}\Rightarrow w_B>w_C,w_B>w_D,w_A>w_D$.

From $w_B+w_C>\{w_A+w_C,w_A+w_D,w_B+w_D,w_C+w_D\}\Rightarrow w_B>w_A,w_B+w_C>w_A+w_D,w_C>w_D,w_B>w_D$.

From $w_A + w_C + w_D > \{ w_A + w_C , w_A + w_D , w_B + w_D , w_C + w_D \} \Rightarrow w_A + w_C > w_B$.

In summary, we have $w_B > w_C > w_D, w_B > w_A > w_D, w_B + w_C > w_A + w_D , w_A + w_C > w_B$.

Linear integer programming can be used to determine these weights. It is well known that solving linear integer problems belong to the class of NP-hard optimization problems [21]. The complexity of solving an integer programming problem is related to the cardinality of the constraint variables set. The number of variables in the integer programming proposed by Iwamoto et al. [14] is $O(2^n),$ where n is the number of shares generated by the dealer; but, in our proposed scheme, the number of variables is $O(n),$ In most GSSs, since the number of shareholders is limited to be a small integer (say 10), this can greatly simplify the processing time to determine the variables.

The objective function, $O,$ in a linear programming can be set up as, $O = \min(\{\max\{w_1, w_2, \ldots, w_n\}),$ to minimize the largest weight among all weights of shareholders, or $O = min(\sum_{i=1}^{n} w_i),$ to minimize the sum of all weights of shareholders. In this given example, if we set $O = min(w_A + w_B + w_C + w_D),$ we can obtain $w_A = 3, w_B = 4, w_C = 2, w_D = 1$ and $t = 6$. A Shamir's WSS can be used to implement this GSS by setting the threshold of the secret to be 6 and assigning $w_A = 3, w_B = 4, w_C = 2, w_D = 1,$ to shareholders, A, B, C and D, respectively. The dealer follows Shamir's $(6,10)$ SS to select a polynomial having degree 5 and generates 3 shares for A, 4 shares for B, 2 shares for C and 1 share for D. Secret can be recovered when there are 6 or more than 6 shares available. The secret reconstruction follows classical Shamir's (t,n) SS. We illustrate our proposed scheme in Figure 1.

{Remark 2} *If contradict conditions, such as $w_A > w_B$ and $w_B > w_A$, derived from our scheme, there have no solutions for weights of shareholders. However, this situation will never be occurred. This is because the inequality conditions are derived from a given secret sharing policy. For a given positive access structure, we obtain both Min and Max. These two subsets are logically complemented to each other. These two subsets will never produce any contradict condition. In other words, if we derive $w_A > w_B$, this implies that A is a minimal positive access instance in Min and B is a maximal negative access instance in $SetMax$. On the other hand, if we derive $w_B > w_A$, this implies B is a minimal positive access instance in Min and A is a maximal negative access instance in Max. These two contradicted results cannot be occurred in a secret access policy.*

4.2 Security Analysis

In our proposed GSS, the threshold, $t,$ of the secret is determined from $,max\{ Max \} < t \leq min\{ Min \}.$ For any positive access instance in a minimal positive access subset, the additive sum of weights of this access instance is larger than or equal to the threshold (i.e., $t \le min\{Min\}$). Thus, the secret can be recovered from shares of any positive access instance in F.. On the other hand, for any negative access instance in a maximal negative access subset, the additive sum of weights of this negative access instance is smaller than the threshold (i.e., $t > max\{Max\}$). Thus, the secret cannot be recovered from shares of any negative access instance in N.

5. Optimized Implementation of GSS Based on Asmuth-Bloom SS

5.1 Proposed Scheme

In our proposed CRT-based GSS, the dealer follows the same procedure as Asmuth-Bloom (t, n) SS to select a secret, s, in the set Z_{p_0}. Then, the dealer selects an integer, α, and linearly shifts the secret as $s + \alpha p_0$, into *the secure secret range.* The share of shareholder, U_i, is computed as $s + \alpha p_0 \bmod p_i$. In the following, we first explain how to determine the *secure secret range* of the shifted value, $s + \alpha p_0$, such that it can enforce the secret sharing policy. Then, we will explain how to determine moduli of shareholders.

Let us assume that there are *n* shareholders and each shareholder, U_i, has a public modulus, p_i. For a general secret access structure, following Section 3.1, we can derive Min and Max. Each item in Min / Max is an access instance which involve multiple shareholders. For example, if $U_A \cap U_B$ is one minimal positive instance in Min, it means that U_A and U_B together can recover the secret. To satisfy this requirement, the shifted value, $s + \alpha p_0$, needs to be smaller than their moduli product, $p_A p_B$. When both U_A and U_B present their shares, their moduli product, $p_A p_B$, is larger than the shifted value so the shifted value can be recovered from their shares. On the other hand, if $U_C \cap U_D$ is one maximal negative access instance in Max, it means that U_C and U_C together cannot recover the secret. To satisfy this requirement, the shifted value needs to be larger than their moduli product, $p_C p_D$. When both U_C and U_D present their shares, their moduli product $p_C p_D$ is smaller than the shifted value so the shifter value cannot be recovered from their shares. If we use $min\{Min\}$ to represent the minimal moduli product in Min and $max\{Max\}$ to represent the maximal module product in Max, the shifted value should be selected from the *secure secret range*, as $(max\{Max\}, min\{Min\})$ $(i.e., max\{Max\} < s + \alpha p_0 < min\{Min\})$ in order to satisfy both security requirements. We use the following example to demonstrate our design.

Example 6. We want to implement the secret sharing policy in Example 2 based on Asmuth-Bloom (t, n) SS.

From Example 2, we have $Min = \{(A \cap B) \cup (B \cap C) \cup (A \cap C \cap D)\}$ and $Max = \{(A \cap C) \cup (A \cap D) \cup (B \cap D) \cup (C \cap D)\}$. Assume that the moduli of A, B, C and D are p_A, p_B, p_C, p_D, respectively. The upper bound of the *secure secret range* is $min\{p_A p_B, p_B p_C, p_A p_C p_D\}$ and the lower bound of the *secure secret range* is $max\{ p_A p_C, p_A p_D, p_B p_D, p_C p_D \}$. Since every moduli product in $SetMin$ should be larger than every moduli product in $SetMax$, we have following inequality conditions.

From $p_A p_B > \{ p_A p_C, p_A p_D, p_B p_D, p_C p_D \} \Rightarrow p_B > p_C, p_B > p_D, p_A > p_D$.

From $p_B p_C > \{ p_A p_C, p_A p_D, p_B p_D, p_C p_D \} \Rightarrow p_B > p_A, P_B P_C > P_A P_D, p_C > p_D, p_B > p_D$.

From $p_A p_C p_D > \{ p_A p_C, p_A p_D, p_B p_D, p_C p_D \} \Rightarrow p_B > p_C, p_A p_C > p_B$.

In summary, we obtain the following inequality conditions, $p_B > p_C > p_{D,} p_B > p_A > p_{D,} p_A p_C > p_B$.

We can use non-linear integer programming to determine these moduli. The objective function, O, can be set up to minimize the size of shares. For example, if $O = min(\{max\{ p_1, p_2, \ldots, p_n \}),$ it minimizes the largest modulus in all moduli of shareholders, or $O = min(\sum_{i=1}^{n} p_i),$ it minimizes the sum of moduli of shareholders. The product form of variables can be easily avoided and changes the problem into a linear integer optimization [22]. More research works are needed to solve our proposed non-linear integer programming. We illustrate our proposed scheme in Figure 2.

{Remark 3} *Following the same discussion in Remark 2, the contradict conditions, such as $p_A > p_B$ and $p_B > p_A$, cannot be occurred in our scheme.*

5.2 Security

In our proposed GSS, the shifted value, $s + \alpha p_0$, is from the *secure secret range* $i.e., max\{ Max \} < s + \alpha p_0 < min\{ Min \}$. For any positive access instance, the moduli product of this access instance is larger than the shifted value (i.e., $> min\{ Min \}$). Thus, the shifted value can be recovered from shares of this access instance. On the other hand, for any negative access instance, the moduli product of this negative access instance is smaller than the shifted value (i.e., $< max\{ Max \}$). Thus, the shifted value cannot be recovered from shares of this negative access instance.

Let us analyze the security of a different situation. We assume that $U_A \cap U_B \cap U_C$ is one positive access instance in the minimal positive access subset. Shareholders, U_A and U_B, with their shares together can recover a value, $s'' \in (0.p_A p_B)$. The "real" shifted value, $s' = s + \alpha p_0$, selected in the *secure secret range*, $max\{ Max \} < s + \alpha p_0 < min\{ Min \}$, is different from $s'' \in (0.p_A p_B)$ since $s'' \in (0.p_A p_B) < max\{ SetMax \}$. Shareholders, U_A and U_B, together cannot recover the "real" shifted value from s''. However, there exists the following relation between s' and s''. That is, $s''' + \lambda p_0 = s' \in (max\{ Max \}, min\{ Min \})$. The number of possible values of λ which can shift s'' into the *secure secret range*, $max\{ Max \} < s + \alpha p_0 < min\{ Min \}$, is $\dfrac{min\{ Min \} - max\{ Max \}}{p_A p_B}$.

However, there is only one λ corresponding to the "real" shifted value. Since the modulus, p_0, satisfies

$p_0 \cdot max\{ Max \} < min\{ Min \},$ we have $\frac{min\{ Min \} - max\{ Max \}}{p_A p_B} > \frac{min\{ Min \} - max\{ Max \}}{max\{ Max \}} > \frac{p_0 \cdot max\{ Max \} - max\{ tMax \}}{max\{ Max \}} > p_0 - 1.$ Thus, the collection of possible values of λ is no less than the collection of possible values of the secret $s.$ No useful information is leaked from the collection of shares in this case.

6. Conclusion

The secret sharing policy of the (*t*, *n*) threshold SS is far too simple for many applications because it assumes that every shareholder has equal privilege to the secret. A general design to implement a GSS based on any classical SS is proposed in this paper. The dealer first determines Min and $Miax$ from a secret sharing policy and then use this information to obtains inequality conditions of parameters of a GSS. Integer optimization is used to minimize the size of shares. Two GSSs, one is based on WSS using linear polynomial and the other is based on Asmuth-Bloom (t, n) SS using CRT, are included to demonstrate our design.

References

[1] A. Shamir, “How to share a secret,” in *Commun. Assoc. Comp. Mach.*, vol. 22, no. 11, pp. 612-613, 1979.

[2] G. R. Blakley, “Safeguarding cryptographic keys,” in *Proceedings of AFIPS'79 Nat. Computer Conf.*, vol. 48, AFIPS Press, pp. 313-317, 1979.

[3] M. Mignotte, “How to share a secret,” in *Proc. of the Workshop on Cryptography,* Springer, Heidelberg, pp. 371-375, 1983.

[4] A. Asmuth and J. Bloom, “A modular approach to key safeguarding,” in *IEEE Transactions on Information Theory,* vol. IT-29, no. 2, pp. 208-210, 1983.

[5] R.J. McEliece and D.V. Sarwate, “On sharing secrets and Reed-Solomon codes,” in *Communications of the ACM,* vol. 24, no. 9, pp. 583-584, 1981.

[6] P. Morillo, C. Padró, G. Sáez and J.L. Villar, “Weighted threshold secret sharing schemes,” in *Information Processing Letters,* vol. 70, pp. 211-216, 1999.

[7] A. Beimel, T. Tassa and E. Weinreb, “Characterizin g ideal weighted threshold secret sharing,” in *J. Kilian, editor, Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005*, Cambridge, MA, USA, February 10-12, 2005, LNCS 3378, Springer-Verlag, pp. 600-619, 2005.

[8] M. Ito, A. Saito and T. Nishizeki, “Secret sharing scheme realizing general access structure,” in *Proceedings of the IEEE Global Telecommunications Conference, Globecom ’87*, pp. 99-102. IEEE Press, 1987.

[9] J. Benaloh and J. Leichter, “Generalized secret sharing and monotone functions,” in S. Goldwasser (Ed.):*Advanced in Cryptology-CRYPTO’ 88,* LNCS, vol. 403, Springer-Verlag, pp. 27-35, 1989.

[10] L. Harn and H. Lin, “An *l*-span generalized secret sharing scheme,” in E,F, Bnckell (Ed.): *Advances in Cryptology- CRYPTO '92*, LNCS, vol. 740, Springer-Verlag, pp. 558-565, 1993.

[11] M. Itoh, A. Saito, and T. Nishizeki, “Secret sharing scheme realizing general access structure,” *IEEE Globecom*, pp. 99-102, 1987.

[12] M. Itoh, “Secret sharing scheme realizing general access structure,” *IEICE Trans. Fundamentals*, vol. J71-A, no. 8, pp. 1592-1598, 1988, (in Japanese).

[13] M. Itoh, “Multiple assignment scheme for sharing secret,” *J. of Cryptology*, vol. 6, pp. 15–20, 1993.

[14] M. Iwamoto, H. Yamamoto, and H. Ogawa, “Optimal multiple assignments based on integer programming in secret sharing schemes with general access structures,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, pp. 101-112, 2007.

[15]] Q. Li, X. X. Li, D. Zheng, and K. F. Chen, “Optimal multiple assignments with (m, m)-scheme for general access structures,” *Cryptology ePrint Archive*, Report 2012/007, 2012.

[16] K. Srinathan, N. T. Rajan and C. P. Rangan, “Non-perfect secret sharing over general access structures,” in A. Menezes, P. Sarkar (Eds.): *INDOCRYPT 2002*, LNCS, vol. 2551, Springer-Verlag, pp. 409-421, 2002.

[17] J. Xu and X. Zha, “Secret sharing schemes with general access structure based on MSPs,” in *J. of Communications*, vol. 2, no. 1, pp. 52-33, 2007.

[18] C. Guo and C- C Chang , “A construction for secret sharing scheme with general access structure,” in *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, 2013.

[19] S. Iftene, “General secret sharing based on the Chinese remainder theorem with applications in e-voting,” in *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67-84, 2007.

[20] H. Cohen, A Course in Computational Algebraic Number Theory, 4th ed., ser. Graduate Texts in Mathematics, Springer-Verlag, 2000.

[21] K. Genova and V. Guliashki, “Linear integer programming methods and approaches–a survey,” in Cybernetics and Information Technologies, vol. 11, no. 1, 2011.

[22] M. Pioro and D. Medhi, Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan Kaufmann Publishers (an imprint of Elsevier). ISBN13: 978-0-12-557189-0.

<u>Parameters generation:</u>

Inputs: $s, n, F = Secret\ sharing\ policy, O = Objective\ function.$

1. For the given positive access structure, $F,$ derive the minimal positive access subset, $Min,$ and maximal negative access subset, $Max.$
2. Establish inequality conditions of weights of shareholders based on relations of additive sums of weights in the subsets, $Min,$ and $Max.$
3. Solve weights, $w_i,$ for $i = 0,1,,\ldots,n,$ using linear optimization with objective function, $O,$ and subject to conditions obtained in Step 2, and $\max\{Max\} < t \le \min\{Min\}.$

Outpits: $t, w_1, \ldots, w_n.$

<u>Share generation:</u>

Generate shares following Shamir's (t,n) threshold SS. The dealer selects a polynomial with degree $t-1$ and generates shares for shareholders. Each shareholder has multiple shares according to his/her weight. Each share is sent to shareholder secretly.

<u>Secret reconstruction:</u>

Given any *t* or more than *t* distinct shares, the secret can be reconstructed following the classical Shamir's secret reconstruction.

Figure 1. Proposed GSS based on WSS.

Parameters generation:

Inputs: $s, p_0, n, F = \mathit{Secret\ sharing\ policy}, O = \mathit{Objective\ function.}$

1. For the given positive access structure, F, derive the minimal positive access subset, $Min,$ and maximal negative access subset, $Max.$
2. Establish inequality conditions of moduli of shareholders based on relations of moduli products in the subsets, $Min,$ and $Max.$
3. Solve moduli, p_i, for $i = 0,1,,\ldots,n,$ with $\gcd(p_i, p_j) = 1, i \neq j,$ using nonlinear optimization with objective function, $O,$ and subject to conditions obtained in Step 2, $max\{ p_A p_C, p_A p_D, p_B p_D, p_C p_D \} < s + \alpha p_0 < min\{ p_A p_B, p_B p_C, p_A p_C p_D \},$ $p_0 \cdot max\{ Max \} < min\{ Min \},$ and $p_0 > s > 0. \cdot$
4. Solve the linearly shifted value, $s' = s + \alpha p_0 \in (max\{ Max \}, min\{ Min \}).$

Outpits: $s', p_1, \ldots, p_n.$

Share generation:

Share for the shareholder, U_i, is generated as $s_i = s + \alpha p_0 \bmod p_i$. s_i is sent to shareholder, U_i, secretly.

Secret reconstruction:

Given any subset of distinct shareholders satisfying the positive access structure, the secret value, $s' = s + \alpha p_0$, can be reconstructed by using the standard CRT. Then, the secret s can be recovered by computing $s = s' \bmod p_0$.

Figure 2. Proposed GSS based on CRT.