

Error Correcting and Complexity Aspects of Linear Secret Sharing Schemes

Yvo Desmedt*, Kaoru Kurosawa**, and Tri van Le*

*Florida State University
Tallahassee, FL 32306-4530
email: {desmedt,levan}@cs.fsu.edu

** Ibaraki University,
Nakanarusawa, Hitachi, Ibaraki, Japan
e-mail: kurosawa@cis.ibaraki.ac.jp

Abstract. Linear secret sharing schemes and general access structures have played a key role in modern cryptography. Cramer-Damgård-Maurer recently proved that any linear secret sharing scheme over a *finite field* can be a verifiable one. We give a simple proof based on error-correcting codes. Our proof allows us to generalize the Cramer-Damgård-Maurer's result to linear schemes over modules, which played an important role in threshold cryptography, i.e. any existing linear secret sharing scheme over a *module* can be changed into a *verifiable* one. We then reflect on another aspect of linear secret sharing. While there has been lots of research on bounds in general access secret sharing schemes, little has been done on the computational complexity aspects. In this paper we also demonstrate that verifying whether a linear scheme is a secret sharing scheme for a given access structure is coNP-complete. The later result relates to the problem cheating sharedealer, the dual problem of secret sharing.

Keywords: secret sharing, VSS, modules, error correcting codes.

1 Introduction

In a secret sharing scheme [4, 19, 12] a dealer D distributes a secret s to n participants P_1, \dots, P_n in such a way that any qualified subset (access sets) can recover s , but any non-qualified subsets (nonaccess set) has no information on s . The family of access sets is called an access structure and is denoted by Γ . An access structure Γ is called monotone if $A_1 \in \Gamma$ and $A_1 \subseteq A_2$ imply $A_2 \in \Gamma$.

Secret sharing have played an important role in theoretical as well as practical modern cryptography. An example is its use in secure distributed computation, e.g. [11, 2, 5] and key escrow [17] is an example of its practical use.

A secret sharing scheme for Γ exists if and only if Γ is monotone. Further, it is known that linear secret sharing schemes can realize any monotone access

structure (see e.g. [20]). The recent work of Cramer-Damgård-Maurer [6] has made linear secret sharing even more important by demonstrating that any linear secret sharing scheme can be verifiable, a very important property. It guarantees that the dealer did not cheat.

This paper studies two aspects linear secret sharing schemes. We investigate a complexity theoretic aspect and an error correcting code one.

1.1 Complexity Theoretic Aspect

A linear secret sharing scheme is realized by a matrix G of size $n \times (h + 1)$ over GF_q such that the share of P_i , with respect to secret $s \in GF_q$, is a subset of rows of column vector $v = G(s, a_1, \dots, a_h)^T$, where a_i 's are random elements of GF_q . While it is known in [20] that any monotone access structure Γ can be realized by a linear secret sharing scheme G , the inverse problem of recovering the access structure Γ from the scheme G is quite hard. In fact, we show that the easier problem of deciding whether the access structure of G is Γ is still *coNP*-complete. This problem is very important because to design a secret sharing scheme, one is often given an access structure Γ , and the purpose is to design a scheme corresponding to this access structure. However, in a verifiable linear secret sharing scheme such as [6], the verifiability aspect only guarantees that the dealer computed the shares according to the matrix G , it does not ensure in any way that the matrix G indeed corresponds to access structure Γ . So a crooked designer may exploit by designing a special matrix G so that some secret set A' of participants will be able to obtain the shared secret illegally without anyone noticing.

1.2 Error Correcting Codes Aspect

Suppose that a dealer D gives v_i to participant P_i as his share. Then we can consider $C = \{(v_1, \dots, v_n)\}$ as an error correcting code. McEliece and Sarwate [16] pointed out that Shamir's (k, n) secret sharing scheme [19] is equivalent to a Reed-Solomon code [18]. For any ideal (k, n) secret sharing scheme, Karnin, Greene and Hellman [15] showed that

$$d = n - k + 1,$$

where d is the minimum Hamming distance of C .

In this paper, we show another error correcting codes aspect of linear secret sharing schemes.

A VSS (verifiable secret sharing scheme) is a protocol in which D proves that (v_1, \dots, v_n) is consistent. We consider a model in which there is a public board and are secure communication channels between any two players. It is assumed that there exists an infinitely powerful adversary who may corrupt a dealer D and a nonaccess set.

Ben-Or, Goldwasser and Wigderson [2] showed a (k, n) threshold VSS. Cramer, Damgård and Maurer [6] showed a VSS for any linear secret sharing scheme.

However, their proof of correctness is rather complex. We show a very simple proof of the validity of Cramer-Damgård-Maurer's VSS from a viewpoint of error correcting codes.

In a linear secret sharing scheme, we view $C = \{(v_1, \dots, v_n)\}$ as a *linear* error correcting code. Therefore, there exists a parity check matrix H . This implies (v_1, \dots, v_n) is a codeword if and only if

$$H \cdot (v_1, \dots, v_n)^T = \mathcal{O}.$$

Our proof is based on this observation.

Our alternative proof also allows us to generalize the Cramer-Damgård-Maurer VSS. First, we show that a variant of the Ben-Or, Goldwasser and Wigderson VSS scheme is a special case of our generalization. Secondly, we present VSS schemes for several homomorphic secret sharing schemes. These are useful for distributed RSA digital signature schemes (called threshold cryptography), where the RSA exponent is not in a field.

2 Linear Secret Sharing Schemes

In a linear secret sharing scheme, there exists a $n \times (h + 1)$ matrix

$$G = \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_n \end{pmatrix} \quad (1)$$

over GF_q . For a secret s , D chooses a random vector (a_1, \dots, a_h) and computes

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = G \begin{pmatrix} s \\ a_1 \\ \vdots \\ a_h \end{pmatrix}. \quad (2)$$

D gives a subset of v_1, \dots, v_n to P_i as his share. For simplicity, however, we assume that the share of P_i is v_i . It is easy to verify that this simplification has no impact on the correctness and generality of our arguments. Indeed, let us say that P_1 has v_1 and v_2 , then P_1 plays the roles of P_1 and P_2 . So, there is no difficulty.

For this secret sharing scheme, $\{P_{i_1}, \dots, P_{i_k}\}$ is an access set if and only if $(1, 0, \dots, 0)$ is expressed as a linear expression of $(\mathbf{g}_{i_1}, \dots, \mathbf{g}_{i_k})$. We denote the access structure of this secret sharing scheme by Γ_M .

3 VSS for Linear Secret Sharing Scheme

We consider a model in which there is a public board and there are secure communication channels between any two players. We also consider infinitely powerful adversaries. An adversary may corrupt a dealer D and a nonaccess set.

Definition 1. We say that (v_1, \dots, v_n) is consistent if there exists some (s, a_1, \dots, a_h) which satisfies eq.(2).

A VSS is a protocol in which D proves that (v_1, \dots, v_n) is consistent. Cramer, Damgård and Maurer showed a VSS for linear secret sharing schemes. Their VSS is given as follows.

1. D chooses an $(h+1) \times (h+1)$ symmetric matrix $R = \{r_{ij}\}$ such that $r_{1,1} = s$ randomly.
2. He computes

$$\begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} = G \cdot R \quad (3)$$

and gives \mathbf{b}_i to P_i .

3. Each P_i computes

$$\begin{pmatrix} u_{i1} \\ \vdots \\ u_{in} \end{pmatrix} = G \cdot \mathbf{b}_i^T$$

and gives u_{ij} to P_j for all $j \neq i$.

4. Each P_j checks if

$$u_{ij} = \mathbf{g}_i \cdot \mathbf{b}_j^T$$

for all $i \neq j$.

5. If there is some inconsistency, D is requested to broadcast u_{ij} or \mathbf{b}_j .
6. Suppose that D is accepted. Then let $v_i = b_{i,0}$ be the share of P_i . where

$$\mathbf{b}_i = (b_{i,0}, \dots, b_{i,h}).$$

4 Simple Proof of VSS Based on Error Correcting Codes

In this section, we show a very simple proof of the validity of the VSS of Cramer, Damgård and Maurer from the viewpoint of error correcting codes.

From eq.(1), we can consider that $C = \{(v_1, \dots, v_n)\}$ is a linear error correcting code and G is a generator matrix. Let H be a parity check matrix of G . Then (v_1, \dots, v_n) is a codeword if and only if

$$H \cdot (v_1, \dots, v_n)^T = \mathcal{O}.$$

We first prove this formally. Without loss of generality, we can assume that all the columns of G are independent. Let M be an $n \times (n-h-1)$ matrix such that the blockmatrix $(G \ M)$ is nonsingular. Let

$$(G \ M)^{-1} = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix}, \quad (4)$$

where H_2 is a $(n-h-1) \times n$ matrix.

Theorem 1. (v_1, \dots, v_n) is consistent (that is, there exists some (s, a_1, \dots, a_h) which satisfies eq.(2)) if and only if

$$H_2 \cdot (v_1 \dots v_n)^T = \mathcal{O}.$$

(Proof) Suppose that there exists some (s, a_1, \dots, a_h) which satisfies eq.(2). Then eq.(2) can be written as $(v_1 \ v_2 \ \dots \ v_n)^T = (G \ M) \cdot (s \ \dots \ a_h \ 0 \ \dots \ 0)^T$. Therefore, we have $(G \ M)^{-1} \cdot (v_1 \ v_2 \ \dots \ v_n)^T = (s \ \dots \ a_h \ 0 \ \dots \ 0)^T$. Hence,

$$\begin{pmatrix} H_1 \\ H_2 \end{pmatrix} \cdot (v_1 \ v_2 \ \dots \ v_n)^T = (s \ \dots \ a_h \ 0 \ \dots \ 0)^T.$$

Then

$$H_2 \cdot (v_1 \ \dots \ v_n)^T = (0 \ \dots \ 0)^T.$$

The converse part is obtained by following the argument backward.

Q.E.D.

Suppose that P_i is given $\mathbf{b}'_i = (b_{i0}, \dots, b_{ih})$ from the dealer. Define

$$X \triangleq \begin{pmatrix} \mathbf{b}'_1 \\ \vdots \\ \mathbf{b}'_n \end{pmatrix} \cdot G^T.$$

If D is honest, then

$$X = G \cdot R \cdot G^T$$

from eq.(3). Note that this X is a symmetric matrix because R is symmetric. The VSS of Cramer, Damgård and Maurer is equivalent to checking if X is symmetric.

We next show a simple proof the validity of this check by using the above Theorem.

Theorem 2. X is symmetric if and only if $(b_{1,0}, \dots, b_{n,0})$ is consistent.

(Proof) First suppose that D is honest. In this case, $\mathbf{b}'_i = \mathbf{b}_i$. Therefore,

$$X = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} \cdot G^T = G \cdot R \cdot G^T.$$

Hence, X is symmetric because R is symmetric.

Next suppose that $X = X^T$. Let

$$B \triangleq \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix}.$$

Then we can write that

$$X = (B \ \mathcal{O}) \cdot \begin{pmatrix} G^T \\ M^T \end{pmatrix}.$$

Hence, since X is symmetric,

$$(B \ \mathcal{O}) \cdot \begin{pmatrix} G^T \\ M^T \end{pmatrix} = (G \ M) \cdot \begin{pmatrix} B^T \\ \mathcal{O} \end{pmatrix}.$$

Now we have

$$(G \ M)^{-1}(B \ \mathcal{O}) = \begin{pmatrix} B^T \\ \mathcal{O} \end{pmatrix} \cdot \begin{pmatrix} G^T \\ M^T \end{pmatrix}^{-1} = \begin{pmatrix} * \\ \mathcal{O} \end{pmatrix}^{-1}.$$

Substitute eq.(4) into the above equation. Then we have

$$H_2 \cdot B = \mathcal{O}.$$

This means that $(b_{1,0}, \dots, b_{n,0})$ is consistent from Theorem 1.

Q.E.D.

Since $v_i = b_{i,0}$, this completes the proof.

5 Generalizations

Our interpretation of Cramer-Damgård-Maurer's VSS scheme allows generalizations. Any secret sharing scheme satisfying an algebraic description as in Section 2 has a VSS similar to the one of Cramer-Damgård-Maurer.

5.1 Extension to VSS of Ben-Or, Goldwasser and Wigderson

First we show that a variant of the Ben-Or, Goldwasser and Wigderson (k, n) threshold VSS scheme [2] is a special case of our interpretation of the Cramer-Damgård-Maurer's VSS.

By slightly modifying the Ben-Or-Goldwasser-Wigderson VSS scheme, it can be described as follows:

1. D chooses a random symmetric polynomial

$$f(x, y) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} r_{i,j} x^i y^j$$

over GF_q such that $f(0, 0) = s$, where s is a secret and $r_{i,j} = r_{j,i}$.

2. He computes

$$b_i(x) = f(x, i)$$

and gives $b_i(x)$ to P_i .

- Each P_i computes

$$u_{ij} = b_i(j)$$

and gives u_{ij} to P_j for all $j \neq i$.

- Each P_j checks if

$$u_{ij} = b_j(i)$$

for all $i \neq j$.

- If there is some inconsistency, D is requested to broadcast u_{ij} or $b_i(x)$.
- Suppose that D is accepted. Then let $v_i = b_i(0)$ be the share of P_i .

This protocol is a special case of Sec.3 such that $R = (r_{i,j})$ and

$$G = \begin{pmatrix} 1, 1, 1, \dots, 1 \\ 1, 2, 2^2, \dots, 2^{k-1} \\ \vdots \\ 1, n, n^2, \dots, n^{k-1} \end{pmatrix}.$$

Therefore, the validity of VSS is proved as a special case of Sec.4. ($f(x, y)$ is not symmetric in their original VSS.)

5.2 Threshold cryptography

Introduction The description of Section 2 can easily be generalized to commutative modules. This allows us to make verifiable secret sharing schemes for linear secret sharing schemes over commutative modules. Such schemes have been developed for threshold cryptography. Examples of such secret sharing schemes are the Desmedt-Frankel [9] and the one by Blackburn-Burmester-Desmedt-Wild [3]. These schemes have been developed for threshold cryptography, see e.g. [7]. These allow, for example, distributed RSA digital signature schemes, where the RSA exponent is not in a field.

In Section 2 the entries of the matrix G and the codewords (v_1, v_2, \dots, v_n) belong to the vector space $GF(q)$. We now use a module approach [13] instead of a vectorspace approach. Let R be a commutative ring and K be a finite Abelian group such that it is an R -module. Now, assume that the entries of G are over the commutative ring R and the entries of a codeword over the finite Abelian group K . We finally assume that G is of full rank, this means that $h + 1$ rows form an invertible matrix.

Our setting implies that $s, a_1, \dots, a_h, v_1, v_2, \dots, v_n$ are elements of K .

VSS for Linear Secret Sharing Scheme over modules Cramer-Damgård-Maurer's VSS scheme can easily be extended to commutative modules, as we now explain. We use the notation of Section 3.

Step 1 of the Cramer-Damgård-Maurer scheme is the only step that needs to be replaced in a module setting, as follows:

- D chooses an $(h + 1) \times (h + 1)$ symmetric two-dimensional array $R' = \{r_{ij}\}$ where $r_{i,j} \in_R K$ such that $r_{1,1} = s$.

We use the notation R' to avoid confusion with the notation R for ring.

Note that the operation $G \cdot R'$ makes sense since all the multiplication with entries in R' (elements of K) are scalar multiplications. However a multiplication of a “two-dimensional array” R'_1 with another “two-dimensional” array R'_2 makes no sense, since this would require a secondary internal operation in K , which may not exist. This is the reason algebraists define matrices with entries in a ring. That is why we call R' a “two-dimensional” array. In our case, all multiplications with R' are properly defined since they involve scalar multiplications with the entries of R' .

The new Step 1 implies that the entries of \mathbf{b}_i and \mathbf{u}_i are in K .

Adapting the proof of the theorem First of all the proof that the participants learn nothing about the secret is a straightforward adaptation of the proof in [6].

First of all we need to demonstrate that there exists an $n \times (n - h - 1)$ matrix M over the ring R such that $(G M)$ is invertible. We use standard techniques from error correcting codes for this. Since G is of full rank, we can write

$$G = P \cdot \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \quad (5)$$

where P is an $n \times n$ permutation matrix and G_1 is an invertible $(h + 1) \times (h + 1)$ matrix. Now consider the blockmatrix:

$$G' = \begin{pmatrix} G_1 & \mathcal{O} \\ G_2 & I \end{pmatrix},$$

where \mathcal{O} is an $(h + 1) \times (n - h - 1)$ zero matrix over the ring R and I is an $(n - h - 1) \times (n - h - 1)$ identity matrix over the ring R . Obviously G' is invertible since G_1 is invertible. Since P is invertible $P * G'$ is also and by taking $(G M) = P * G'$ we have obtained the desired matrix M .

To adapt our proofs of Theorems 1 and 2, we view the module as a bimodule [14] where the effect of the scalar is the same whether we multiply with a left or right scalar. So, when $a \in R$ and $k \in K$, we can write the scalar operation as $a \cdot k$ and as $k \cdot a$ and $a \cdot k = k \cdot a$. It is trivial to see that any R -module can be used as such a bimodule.

We now explain how to read the proofs of Theorems 1 and 2 in our general setting. In these proof, take the entries of X and the zero entries of \mathcal{O} as elements of K . We view X and \mathcal{O} as “two-dimensional arrays”, as explained in Section 5.2. The rest then generalizes trivially.

Impact It is not too difficult to verify that several secret sharing schemes satisfy the module structure. Examples are the threshold secret sharing schemes and the general access secret sharing scheme in [9]. Indeed, the first ones are generalizations of Shamir’s linear secret sharing scheme [19] where K has been

adapted from the original key space to make it an appropriate module. The general access secret sharing scheme in [9] is an adaptation of [20], a secret sharing scheme linear over $GF(q)$ and so also linear over a module. When the group is Abelian in [8] and [3], a closer inspection of the schemes shows that these are linear over a module.

So, the generalization of Cramer-Damgård-Maurer works for all these schemes.

6 Access Structure Matching Problem

In this section, we prove that the problem of deciding whether the access structure of G is Γ is coNP-complete. Here we assume that each access structure Γ is given by a monotone boolean circuit f_Γ such that $f_\Gamma(A) = 1$ if and only if $A \in \Gamma$.

We also prove that the problem of deciding whether two given monotone boolean circuits are equivalent is a coNP-complete problem. The later problem is interesting in its own since most problems involving properties of monotone boolean circuits are not hard. For example, the well-known CIRCUIT-SAT problem is NP-hard for general circuits, but are quite easy for monotone ones. Another example is that whil-known CIRCUIT-SAT problem is NP-hard for general circuits, but are quite easy for monotone ones. Another example is that whiltions are already known, proving such similar result for general circuit would be a major result. Thus the general feeling is that monotone boolean circuits are much easier to handle. However, we show here that these circuits are rich enough that even deciding their equivalence is coNP-hard. First we recall the NP-complete vertex-cover problem.

Let $G = (V, E)$ be an undirected graph, a subset $S \subseteq V$ of vertices is called a vertex cover of G if for all edge (u, v) of G , u or v is in S . It is well known that the problem of deciding whether a given undirected graph G has a vertex cover of size at most k is a NP-complete problem [10].

In the following we say that a linear secret sharing scheme realized by a matrix G is simply a linear secret sharing scheme G . First we need the following facts.

Fact 1 ([1]) *Given an access structure Γ , one can build in polynomial time a linear secret sharing scheme G such that $\Gamma_G \equiv \Gamma$, where the term “polynomial time” is with respect to the description length of boolean circuit f_Γ .*

Fact 2 ([21]) *Let $T_{k,m}(b)$ be the k -out-of- m threshold function:*

$$T_{k,m}(b_1, \dots, b_m) = \begin{cases} 1, & \text{if } \sum_{i=1}^m b_i > k \\ 0, & \text{otherwise.} \end{cases}$$

Then $T_{k,m}(b_1, \dots, b_m)$ can be represented as a monotone boolean circuit of size $O(m^3)$.

Proof. Our proof is different from that in [21]. First we observe that if b_1, \dots, b_m were sorted in descending order then $T_{k,m}$ can be simply as $T_{k,m} = b_1 \wedge \dots \wedge b_{k+1}$. Therefore it is enough to show that the bit sorting circuit $S_m(b_1, \dots, b_m)$, which output m input bits b_1, \dots, b_m in descending order, can be constructed from *AND* and *OR* gates in $O(m^3)$ time. First if $m = 2q$, then we can recursively construct S_m as follows:

1. Let $(y_1, \dots, y_q) = S_q(b_1, \dots, b_q)$, and $(z_1, \dots, z_q) = S_q(b_{q+1}, \dots, b_m)$.
2. For $1 \leq l \leq m$, let $s_l = \bigvee_{i=0}^l (y_1 \wedge \dots \wedge y_i \wedge z_1 \wedge \dots \wedge z_{l-i})$.
3. Let $S_m(b_1, \dots, b_m) = (s_1, \dots, s_m)$.

Next if $m = 2q + 1$, then we construct: $S_m(b_1, \dots, b_m) = \Pi_m(S_{m+1}(b_1, \dots, b_m, 0))$, where $\Pi_m : \{0, 1\}^{m+1} \rightarrow \{0, 1\}^m$ is the projection onto the first m coordinates. The case $m = 1$ is trivial. Noting that s_l is the l^{th} largest bit, it is not hard to see that this construction takes $O(m^3)$ steps and produces a sorting circuit of size $O(m^3)$. *Q.E.D.*

We now formally state and prove our theorems.

Theorem 3. *Given two access structures Γ_1, Γ_2 as two boolean circuits, deciding whether Γ_1 and Γ_2 are equivalent is a coNP-complete problem.*

Proof. Let (G, k) be an instance of the Vertex Cover (VC) problem. We will construct in polynomial time two monotone boolean circuits Γ_1, Γ_2 such that:

$$(G, k) \in VC \Leftrightarrow \Gamma_1 \not\equiv \Gamma_2.$$

Indeed, let m be the number of nodes of the graph $G = (V, E)$. Without loss of generality, we can assume that $1 \leq k \leq m$. First, let Γ_1 be the following boolean formula:

$$\Gamma_1(b_1, b_2, \dots, b_m) = \bigwedge_{(u,v) \in E} (b_u \vee b_v),$$

and let Γ_2 be the following boolean circuit:

$$\Gamma_2(b_1, b_2, \dots, b_m) = T_{k,m}(b_1, \dots, b_m) \bigwedge \Gamma_1(b_1, \dots, b_m).$$

By Lemma 2, $T_{k,m}(b_1, \dots, b_m)$ can be constructed in $O(m^3)$ steps. Hence Γ_1, Γ_2 are two monotone boolean circuits that can be constructed in polynomial time. We now need to show that $(G, k) \in VC \Leftrightarrow \Gamma_1 \not\equiv \Gamma_2$, thus reduce the VC problem to our problem.

(\Rightarrow) Assume that $(G, k) \in VC$. Let $G = (V, E)$. By definition of VC, there exists a vertex cover $S \subseteq V$ such that $\forall (u, v) \in E : (u \in S) \vee (v \in S)$, and that $|S| \leq k$. For $i = 1, 2, \dots, m$, let $b_i = 1$ if $i \in S$, and $b_i = 0$ otherwise. Because S is a vertex cover of G , for all edge $(u, v) \in E$, either u or v belongs to S . Therefore $\Gamma_1(b_1, b_2, \dots, b_m) = 1$. Further, we also have $\Gamma_2(b_1, b_2, \dots, b_m) = 0$ because $\sum_{i=1}^m b_i = |S| = k$. This proves that $\Gamma_1 \not\equiv \Gamma_2$.

(\Leftarrow) Assume that $\Gamma_1 \not\equiv \Gamma_2$. Then there exists $(b_1, \dots, b_m) \in \{0, 1\}^m$ such that $\Gamma_1(b_1, \dots, b_m) \neq \Gamma_2(b_1, \dots, b_m)$. Since we know from the definition of Γ_1, Γ_2 that $\forall x \in \{0, 1\}^m : \Gamma_1(x) \geq \Gamma_2(x)$. Hence we must have $\Gamma_1(b_1, \dots, b_m) = 1$ and $\Gamma_2(b_1, \dots, b_m) = 0$. Since $\Gamma_2 = \Gamma_1 \wedge T_{k,m}$, these two imply that $T_{k,m}(b_1, \dots, b_m) = 0$. Now let $S = \{i \mid b_i = 1\}$. Because $\Gamma_1 = 1$, for all edge $(u, v) \in E$, $b_u \vee b_v = 1$. Therefore either b_u or b_v is in S . We thus deduce that S is a vertex cover of G . Further $T_{k,m} = 0$ implies that $|S| \leq k$. Therefore $(G, k) \in VC$.

In the above we have shown that $(G, k) \in VC \Leftrightarrow \Gamma_1 \not\equiv \Gamma_2$. Since VC is NP-complete, we conclude that deciding whether $\Gamma_1 \equiv \Gamma_2$ is coNP-hard. Since checking if $\Gamma_1 \not\equiv \Gamma_2$ can be done in NP by giving a particular boolean vector b where $\Gamma_1(b) \neq \Gamma_2(b)$, we obtain that deciding whether $\Gamma_1 \equiv \Gamma_2$ is indeed coNP-complete. *Q.E.D.*

Corollary 1. *Given an access structure Γ and a linear secret sharing scheme G , deciding if $\Gamma_G \equiv \Gamma$ is a coNP-complete problem.*

Proof. From Theorem 1 one can reduce the coNP-complete problem of deciding whether $\Gamma_1 \equiv \Gamma_2$ in Theorem 3 to the above problem by constructing a linear secret sharing scheme G such that $\Gamma_G \equiv \Gamma_1$, and then we have $\Gamma_G \equiv \Gamma_2 \Leftrightarrow \Gamma_1 \equiv \Gamma_2$. Hence deciding whether $\Gamma_G \equiv \Gamma$ is coNP-hard. There is one technical difficulty nevertheless: The original theorem in [1] only constructs G for access structure Γ_1 given as monotone boolean formula, not monotone boolean circuit. However, this is not really a problem because if we look carefully at the proof of our theorem 3, it only requires that Γ_2 to be a circuit, and Γ_1 can be a formula. Thus we can also apply our theorem 3 here. Now it is not hard to show that this problem is in coNP. Hence we have shown that it is coNP-complete. *Q.E.D.*

Besides the new complexity bound, there is a security consequence of this theorem. While work in secret sharing schemes have mainly concentrated on dealing with cheating shareholders, the dual problem of cheating sharedealer also has practical importance. This paper has demonstrated that such situation is possible and non-trivial, since a crooked designer may design a special matrix G so that some secret set A' of participants will be able to obtain the shared secret illegally, yet no one else knows that the constructed matrix G has hidden feature. .

References

1. J. C. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology, Proc. of Crypto'88 (Lecture Notes in Computer Science 403)*, pages 27–35. Springer-Verlag, 1990. Santa Barbara, California, U.S.A., August 11–15.
2. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 1–10, May 2–4, 1988.

3. S. R. Blackburn, M. Burmester, Y. Desmedt, and P. R. Wild. Efficient multiplicative sharing schemes. In U. Maurer, editor, *Advances in Cryptology — Eurocrypt '96, Proceedings (Lecture Notes in Computer Science 1070)*, pp. 107–118. Springer-Verlag, 1996. Zaragoza, Spain, May 12–16.
4. G. R. Blakley. Safeguarding cryptographic keys. In *Proc. Nat. Computer Conf. AFIPS Conf. Proc.*, pp. 313–317, 1979. vol.48.
5. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pp. 11–19, May 2–4, 1988.
6. R. Cramer, I. Damgård, and U. M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In B. Preneel, editor, *Advances in Cryptology — Eurocrypt 2000, Proceedings (Lecture Notes in Computer Science 1807)*, pp. 316–334. Springer-Verlag, 2000. Bruges, Belgium, May 14–18.
7. A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *Proceedings of the twenty-sixth annual ACM Symp. Theory of Computing (STOC)*, pp. 522–533, May 23–25, 1994. Montréal, Québec, Canada.
8. Y. Desmedt, G. Di Crescenzo, and M. Burmester. Multiplicative non-abelian sharing schemes and their application to threshold cryptography. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology — Asiacrypt '94, Proceedings (Lecture Notes in Computer Science 917)*, pp. 21–32. Springer-Verlag, 1995. Wollongong, Australia, November/December, 1994.
9. Y. G. Desmedt and Y. Frankel. Homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM Journal on Discrete Mathematics*, 7(4), pp. 667–679, November 1994.
10. M. Garay and D. Johnson. *Computers and Intractability: A guide to NP-completeness*. W. H. Freeman and Company, New York, 1979.
11. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth annual ACM Symp. Theory of Computing, STOC*, pp. 218–229, May 25–27, 1987.
12. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE Global Telecommunications Conf., Globecom'87*, pp. 99–102. IEEE Communications Soc. Press, 1987.
13. N. Jacobson. *Basic Algebra I*. W. H. Freeman and Company, New York, 1985.
14. N. Jacobson. *Basic Algebra II*. W. H. Freeman and Company, New York, 1989.
15. E. D. Karnin, J. W. Greene, and M. Hellman. On secret sharing systems. *IEEE Tr. Inform. Theory*, 29(1), pp. 35–41, January 1983.
16. R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Comm. ACM*, 24(9), pp. 583–584, September 1981.
17. S. Micali. Fair public-key cryptosystems. In E. F. Brickell, editor, *Advances in Cryptology — Crypto '92, Proceedings (Lecture Notes in Computer Science 740)*, pp. 113–138. Springer-Verlag, 1993. Santa Barbara, California, U.S.A., August 16–20.
18. I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *SIAM Journal on Applied Mathematics*, 8, pp. 300–304, 1960.
19. A. Shamir. How to share a secret. *Commun. ACM*, 22, pp. 612–613, November 1979.
20. G. J. Simmons, W. Jackson, and K. Martin. The geometry of shared secret schemes. *Bulletin of the Institute of Combinatorics and its Applications*, 1, pp. 71–88, 1991.
21. I. Wegener. *The Complexity of Boolean Functions*. J. Wiley, New York, 1987.