Secret Sharing Schemes and Linear Codes

Hakan Özadam[1,2], Ferruh Özbudak[1,2] and Zülfükar Saygı[2,3]
[1]Department of Mathematics and [2]Institute of Applied Mathematics,
Middle East Technical University, İnönü Bulvarı, 06531, Ankara, Turkey
[3]Department of Mathematics, Faculty of Arts and Sciences,
TOBB Economics and Technology University, Söğütözü 06530, Ankara, Turkey
Email: {ozhakan,ozbudak,saygi}@metu.edu.tr

***Abstract*—The study of secret sharing schemes was independently initiated by Shamir and Blakely in 1979. Since then several other secret sharing schemes were introduced. Many of those schemes are (n,k) threshold systems. In this work we give a survey on secret sharing schemes using linear codes. We analyze Shamir's scheme with an emphasis on the access structure. We explain Massey's linear secret sharing scheme and mention its superiority to (n,k) threshold systems.**

***Index Terms*—Secret sharing, linear codes, (n,k) threshold schemes**

I. Introduction

Imagine that a group of scientists came up with a formula which must be kept secret. Therefore the formula is encrypted and the key is stored in some place which is hopefully safe. The formula is so important for the scientists that they can not stand loosing it. For this reason they make several copies of the encryption key and store it in different locations. But this would increase the risk of key being stolen. What they actually need is somehow split the key into pieces, store the pieces in different locations and be able to reconstruct the key even if some of the pieces are missing. If a piece is lost or stolen, they can recover the key from the remaining pieces yet the stolen piece would not be sufficient to recover the key.

As a second example, consider that there is a deadly weapon whose production is an automated process controlled by a computer. The product plan has been loaded to the computer and the computer controls the production by executing the instructions given in the plan. The plan is kept as encrypted. Before each production the plan is decrypted and after the production, decrypted plan is deleted. This guarantees that the production can only be initiated by authorized people. This weapon is so dangerous that no one is allowed to start its production alone. Therefore the decryption key must be distributed to the officials such that the production can be started only when an authorized group decides to do so. This can be made possible if there is a software loaded to the computer which can split the original encryption key into pieces and reconstruct the key using some of these pieces. In this scenario, we assume that the computer does not store the original encryption key permanently and it works perfectly secure and reliable.

It is easily seen that these two real-world problems are of the same kind which is called the problem of secret sharing. More explicitly, in a secret sharing scheme, there are shareholders, a trusted party and a secret (encryption key in the above cases). The trusted party divides the secret into pieces so that only authorized groups of shareholders are able to recover the secret.

A trivial solution to this problem is the following. The secret is given to a trusted third party (which is possibly a computer programme). It encrypts the data with a key of length say 1024 bits. Assuming there are 4 shareholders, the key bits are divided into 4 blocks, each having 256 bits. Each block B_i is given to a shareholder P_i.

When shareholders come together, they give their shares (key pieces) to the trusted party and recover the secret. If there are 3 or less shareholders then at least 256 bits of the key are unavailable so decryption can not be done.

Before discussing the drawback of this scenario, we need the notion of entropy. Let X be a random variable with alphabet $\mathcal{X}$ and a probability mass function $p_X(x) = Pr\{X = x\}$. The entropy of X is defined as $H(X) = -\sum_{x\in\mathcal{X}} p_X(x)log_2 p_X(x)$. Let Y be another random variable with alphabet $\mathcal{Y}$ and a probability mass function $p_Y(y) = Pr\{Y = y\}$. Regarding the pair (X, Y) as another random variable with probability mass function $p_{XY}(x, y)$, the conditional entropy $H(Y|X)$ is defined as $H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x, y)log_2 p_Y(y|x)$. The mutual information $I(X;Y)$ between X and Y is defined as $I(X;Y) = H(Y) - H(Y|X)$.

The main disadvantage of such a system is that it leaks the secret. In our example , a person who wants to reveal the secret and has absolutely no knowledge of the key has to crack a 1024-bit key, but a shareholder already knows 256 bits of this key and when 2 shareholders come together and combine their keys they need to crack a 512-bit key to get the secret. In information theoretic terms this means $H(K) > H(K|K_1, K_2)$, or equivalently $I(K; K_1, K_2) > 0$. As a consequence unauthorized groups of shareholders are serious threats for the security of this system.

This problem can be remedied as follows, let K_1, K_2, K_3 be three random bit sequences each of which is 1024 bits long and K be the original encryption key (secret). Let $K_4 = K_1 \oplus K_2 \oplus K_3 \oplus K$ where $\oplus$ stands for bitwise XOR. Then the keys K_1, K_2, K_3, K_4 are distributed to the shareholders P_1, P_2, P_3, P_4. Since K_1, K_2, K_3 are chosen randomly when any 3 shareholders come together they ab-

solutely have no information about the secret that is $H(K) = H(K|K_{i_1}, K_{i_2}, K_{i_3})$ where $\{i_1, i_2, i_3\} \subset \{1, 2, 3, 4\}$. In other words the secret does not leak. In [10], a more detailed discussion of secret sharing schemes from an information theoretic point of view can be found..

Though, leakage issue is handled, this system brings other problems. If the shareholder P_4 is missing (died, ill etc.) then it will be impossible to recover the secret for P_1, P_2, P_3.

In [12], the author considers the following problem: Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?

In [16], the minimal solution is given as 462 locks in total and 252 keys per scientist and these numbers exponentially increase as the number of scientists increase.

Imagining that the secret is residing in a cabinet and encryptions correspond to locks, if many locks and many keys are used as in the previous problem, leakage and recoverability problems can be solved. However, as Shamir points out in [16], such a solution would require enormous number of locks and keys and hence is not practical.

According to our discussion, when distributing a secret, it is sensible to require that the following conditions are satisfied:

1) Secret does not leak, that is to say any group of unauthorized shareholders should have absolutely no information about the secret (I(Secret; Insufficient Shares) $= 0$). In the literature, such schemes are called **perfect** secret sharing schemes.
2) It is possible to recover the secret even if some of the shareholders are missing.
3) The sizes of the pieces given to shareholders should not be too large when compared to the original key and encryption/decryption can be done in a feasible amount of time.

In 1979, Shamir introduced an algebraic secret sharing scheme [16] satisfying all the above constraints. This scheme is discussed in detail in the next section.

The same year, Blakely introduced another secret sharing scheme [4] which is based on a simple geometric idea. Two non-parallel lines in the plane intersect at one point and one line itself does not provide any information about the point of intersection. More generally, t non-parallel hyperplanes in t dimensional space intersect at one point and any $t-1$ or less hyperplanes give no information about the point of intersection. In Blakely's scheme, there are $n > t$, shareholders and shares are n non-parallel hyperplanes of t dimensional space. Each plane is given to a shareholder. When any t out of n shareholders combine their shares, they can recover the secret by computing the point of intersection.

Asmuth and Bloom introduced a secret sharing scheme which is essentially different from that of Shamir's and Blakely's schemes. They used an arithmetic approach based on Chinese Remainder Theorem. For details see [3].

TABLE I
POSSIBLE INTERPOLATIONS

y	p(x)
0	$3*x^2+2*x+6$
1	$6*x^2+5$
2	$2*x^2+5*x+4$
3	$5*x^2+3*x+3$
4	x^2+x+2
5	$4*x^2+6*x+1$
6	$4*x$

II. SHAMIR'S SECRET SHARING SCHEME

Shamir's system is based on the following observation: Let $\mathbb{F}_q$ be a finite field, $(x_i, y_i) \in \mathbb{F}_q \times \mathbb{F}_q, \quad i = 1, \ldots, k$, where all $x_i s$ are distinct. There is a unique polynomial $p(x)$ of degree $k-1$ such that $p(x_i) = y_i$ for $i = 1, \ldots, k$. And there are q polynomials of degree $k-1$ satisfying $p(x_i) = y_i$ for $i = 1, \ldots, k-1$.

Assume that we represent the secret as an element a of $\mathbb{F}_q$. We pick a random polynomial $p(x) \in \mathbb{F}_q[x]$ such that $p(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$ where $a = a_0$. Say we'll distribute the secret to n shareholders. We pick any $n \geq k$ distinct elements from $\mathbb{F}_q$. The shares are calculated as pairs $K_i = (x_i, p(x_i)) = (x_i, y_i)$. In order to recover the secret, any k out of n shareholders come together. They give their shares to the trusted party and then the polynomial $p(x)$ can be constructed by Lagrange interpolation. There are other efficient algorithms (see, for example [11]) for polynomial interpolation.

A. Example

Let there be 4 shareholders P_1, P_2, P_3, P_4, suppose that we are working in $\mathbb{F}_7$ and the secret is 2. We need to choose a polynomial whose constant term is 2. We pick $p(x) = x^2 + x + 2 \in \mathbb{F}_7[x]$. Next we randomly choose 4 elements of $\mathbb{F}_7$, let them be $1, 2, 4, 5$. Now we put $p(1) = 4$, $p(2) = 1$, $p(4) = 1$, $p(5) = 4$. Then the shares are $K_1 = (1, 4)$, $K_2 = (2, 1)$, $K_3 = (4, 1)$, $K_4 = (5, 4)$. When P_1, P_2 and P_4 combine their shares they can reconstruct $p(x)$ as $4(x-2)(x-5)/(1-2)(1-5) + 2(x-1)(x-5)/(2-1)(2-5) + 4(x-1)(x-2)/(5-1)(5-2) = x^2 + x + 2$.

But if P_1 and P_2 combine their shares, to uniquely determine $p(x)$, they need to fix a third point $x \in \mathbb{F}_7$ and interpolate for different values of y, where $p(x) = y$. Say P_1 and P_2 fixed $x = 5$. Then they need to consider the 7 possible values of y to obtain $p(x)$. In Table I, we present the resulting interpolations for all possible values of y. Clearly, what they do here is an exhaustive search in $\mathbb{F}_7$.

Generalizing this observation, consider that $k-1$ shareholders come together to reveal the secret. In order to make a precise interpolation, a k^{th} (x_j, y_j) pair is needed. For a fixed x_j, there are q possibilities for y_j and hence there are q different equally likely polynomials. Using Lagrange Interpolation, it can easily be shown that each polynomial has a distinct constant term (as in Table I). So we see that Shamir's scheme does not leak the secret.

B. Drawbacks

One of the main disadvantages of this system is that it has a very simple access structure. Any k out of n participants are able to recover the secret. Such systems are called **(n,k) threshold systems**. Note that Blakely's [4] and Asmuth-Bloom's [3] schemes are also (n,k) threshold schemes. However in some cases we require that some shareholders should be more privileged than others. For examples; executives of a company with a CEO and military officers having different ranks. In such cases it may well be expected that CEO and any other executive or else at least 4 executives are required to recover the secret. Clearly (n,k) threshold systems do not have such capabilities.

Another important disadvantage is coming from the dense access structure of the system. The following example which is taken from [1] will help us better understand the situation. Suppose we have a vault with a $(6, 3)$ access structure. That is to say, any 3 out of 6 shareholders can open the vault. One day, the vault is found empty. Only one of the shareholders provided an alibi proving his innocence. There are $\binom{5}{3} = 10$ other possible betrayal scenarios. Having only this information it will be extremely difficult to solve the case for the investigators. Now suppose that the vault had the following sparse access structure $\{1, 2, 4\}, \{3, 4, 5\}, \{1, 3, 6\}, \{2, 5, 6\}$. And consider that the shareholder 1 has an alibi. This would leave only two possibilities: either 3,4 and 5 or 2,5 and 6 are guilty. In any case, 5 is guilty and hopefully he can be convinced to give the names of his companions.

These observations motivated the invention of other secret sharing schemes having better access structures than (n, k) threshold schemes. In the proceeding section we will see that, secret sharing schemes based on linear codes are one of those schemes having nice access structures.

III. Secret Sharing Using Linear Codes

Throughout this section, we follow the denitions and notations given in [8], [9], [13].

In [14] authors deduce that Shamir's secret sharing scheme is very similar to a special type of Reed-Solomon codes. Later on it has been realized [13] that coding theory can be applied to secret sharing.

Here, the idea is the following. Suppose we can encode our secret into a codeword $(D_1, \ldots, D_n)$. If we know sufficiently many $D_i s$ then using the error correction mechanism we can find the remaining $D_i s$ and recover the secret. This observation suggests that error-correcting codes can be used to design secret sharing schemes.

Massey invented a secret sharing scheme [13] using this idea. A similar scheme was introduced by Brickell, for details see [5]. Xiaoqing and Zhiguo proposed another secret sharing scheme [17] which is also based on linear codes but their scheme does not require a trusted third party which is not the case in previous ones.

Now we explain in detail the secret sharing scheme introduced by Massey [13]. This scheme is also referred as linear secret sharing scheme since it is based on linear codes. Let $\mathbf{C} \subset \mathbb{F}_q^n$ be a k-dimensional linear code with generator matrix $G = [\mathbf{g}_0, ..., \mathbf{g}_{n-1}]$. Throughout this document we will assume that G has no zero column. In this scenario the secret s is an element of $\mathbb{F}_q$ and there are $n-1$ shareholders and a dealer (trusted party).

In order to determine the shares, the dealer chooses $\mathbf{t} \in \mathbf{C}$, $\mathbf{t} = (t_0, ..., t_{n-1})$ such that $t_0 = s$. He can choose such a $\mathbf{t}$ by first picking randomly a vector $\mathbf{u} = (u_0, ..., u_{k-1}) \in \mathbb{F}_q^k$ such that $s = \mathbf{u}\mathbf{g}_0$. Such a $\mathbf{u}$ can be chosen in q^{k-1} ways. Now $\mathbf{t}$ can be computed as $\mathbf{t} = \mathbf{u}G$. Shares are $\{t_1, ..., t_{n-1}\}$ and G is known by all shareholders.

Our assumption that G can not have any zero column is sensible because if a column $\mathbf{g}_i$ were zero then clearly t_i which is the share of the i^{th} participant would be zero. Hence this shareholder would not participate at all.

Note that if $\mathbf{c}$=(0,...,0,1,0,...,0) $\in \mathbf{C}^\perp$ where 1 occurs in j^{th} position then since $\mathbf{cv} = 0\ \forall \mathbf{v} \in \mathbf{C}$ we get that j^{th} component of every $v \in \mathbf{C}$ is 0. This implies $g_j = 0$ but this contradicts our assumption hence such a $\mathbf{c}$ can not exist.

As stated in [8] if $\mathbf{g}_0, \mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly dependent then the secret can be recovered by first solving the linear equation

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

after finding x_js, the secret can be computed as

$$t_0 = \mathbf{u}\mathbf{g}_0 = \sum_{j=1}^{m} x_j \mathbf{u}\mathbf{g}_{i_j} = \sum_{j=1}^{m} x_j t_{i_j}.$$

From now on we will assume that this is the only way to recover the secret for any set of shares.

Let $\mathbf{C} \subset \mathbb{F}_q^n$ be a linear code over $\mathbb{F}_q$.

Definition III.1 (Support of a Vector)**.** The support of a vector $\mathbf{v} \in \mathbb{F}_q^n$ is defined to be

$$\{0 \le i \le n-1 : v_i \neq 0\}$$

Definition III.2. The vector $\mathbf{v}_1 \in \mathbb{F}_q^n$ is said to *cover* $\mathbf{v}_2 \in \mathbb{F}_q^n$ if the support of $\mathbf{v}_1$ contains that of $\mathbf{v}_2$.

Definition III.3 (Minimal Vector)**.** The vector $\mathbf{v} \neq 0$ is called minimal if it only covers its scalar multiples.

Definition III.4 (Minimal Codeword)**.** A codeword whose first component is 1 and only covers its scalar multiples is called a minimal codeword.

Obviously every minimal codeword is a minimal vector whereas converse is not the case. For a codeword $\mathbf{v} = (1, v_1, ..., v_{n-1}) \in \mathbf{C}^\perp$ not all v_j 0 , the secret can be recovered as $\mathbf{vt} = t_0 + v_1 t_1 + ... + v_{n-1} t_{n-1} = 0$ ($\mathbf{v} \in \mathbf{C}^\perp$, $\mathbf{t} \in \mathbf{C}$). Therefore, t_0 is obtained as $t_0 = -(v_1 t_1 + ... + v_{n-1} t_{n-1})$ where $t_1, \ldots, t_{n-1}$ are the shares. Moreover if there is $\mathbf{w} \in \mathbf{C}^\perp$ such that $\mathbf{w} = (1, 0, ..., 0, w_{l_1}, 0, ..., w_{l_m}, 0, ..., 0)$ not all w_{l_j} 0 then t_0 can be recovered as $\mathbf{wt} = t_0 + w_{l_1} t_{l_1} + ... + w_{l_m} t_{l_m} = 0$.

This observation suggests that the minimal access sets can be determined by the codewords in the dual code $\mathbf{C}^{\perp}$ whose first entry is 1 having maximum number of 0 components namely, minimal codewords. It turns out that the access structure of this scheme is completely determined by the minimal codewords.

We now state a well known result which gives the correspondence between linearly dependent column vectors of G and codewords in $\mathbf{C}^{\perp}$.

Proposition III.5. The columns $\mathbf{g}_{i_1}, ..., \mathbf{g}_{i_k}$ of G are linearly dependent if and only if there exists a codeword $\mathbf{c} = (0, .., 0, c_{i_1}, 0, ..., 0, c_{i_m}, 0, ...0) \neq 0$ in $\mathbf{C}^{\perp}$.

Proof: Let $g_{i_r s}$ denote the s^{th} component of the column vector $\mathbf{g}_{i_r}$. If $\mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly dependent then there exists $c_{i_1}, ..., c_{i_m}$ not all zero, $c_{i_1}\mathbf{g}_{i_1} + ... + c_{i_m}\mathbf{g}_{i_m} = 0$. Then obviously $c_{i_1}g_{i_1 s} + ... + c_{i_m}g_{i_m s} = 0$ for all $s = 0, ..., m-1$. Now pick any $\mathbf{v} \in \mathbf{C}$. Since $\mathbf{v}$ is a linear combination of the row vectors of G we have $\mathbf{v} = \sum_{j=0}^{m-1} a_j(g_{1j}, ..., g_{(m-1)j})$. For $\mathbf{c} = (0, ..., 0, c_{i_1}, 0, ..., 0, c_{i_m,0,...,0})$ we have
$\mathbf{vc} =$
$\sum_{j=0}^{m-1} a_j(g_{1j}, ..., g_{(m-1)j})(0, ..., 0, c_{i_1}, 0, ..., 0, c_{i_m}, 0, ..., 0)$
$= \sum_{j=0}^{m-1} a_j(c_{i_1}g_{i_1 j} + ... + c_{i_m}g_{i_m j}) = \sum_{j=0}^{m-1} a_j 0 = 0$ which is to say that $c \in \mathbf{C}^{\perp}$. Conversely if $0 \neq \mathbf{c} \in \mathbf{C}^{\perp}$ then $\mathbf{vc} = 0$ and by similar arguments it follows that $\mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly dependent. ■

Proposition III.6 ([13])**.** Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be two distinct minimal codewords of $\mathbf{C}$. Then $\mathbf{c}_1$ and $\mathbf{c}_2$ can not be nonzero in the same components.

Proof: Assume the contrary, since they are distinct there exist a component (say j^{th} one) that they differ. Then $\mathbf{c}_1 = (1, ..., a, ...)$, $\mathbf{c}_2 = (1, ..., b, ...)$ where a and b are j^{th} components and $a \neq b$ also $a \neq 0$, $b \neq 0$. Then $\mathbf{c}_3 = \mathbf{c}_2 - a^{-1}b\mathbf{c}_1$ is also a codeword of $\mathbf{C}$. Note that support of $\mathbf{c}_3$ is a proper subset of support of $\mathbf{c}_1$ and $\mathbf{c}_2$. Also notice that first component of $\mathbf{c}_3$ is nonzero. Thus $\mathbf{c}_4 = (1 - a^{-1}b)\mathbf{c}_3$ is a codeword whose first component is 1, $\mathbf{c}_4$ is not a scalar multiple of c_1 and c_2 also c_1 and c_2 covers $\mathbf{c}_4$. This contradicts the fact that $\mathbf{c}_1$ and $\mathbf{c}_2$ are minimal codewords. ■

Clearly, any group of shareholders including an authorized access group can recover the secret. Therefore we are only interested in sets of shareholders where any proper subset of them are unable to recover the secret. This motivates the following definitions.

Definition III.7 (Minimal Access Set)**.** A set of participants is called a minimal access set if they can recover the secret by combining their shares but any of its proper subsets can not do so.

Definition III.8 (Access Structure)**.** The access structure of a secret sharing scheme is the set of all minimal access sets.

In some cases there are participants which exist in every minimal access set i.e. recovering the secret is impossible without them. Such participants are called **dictatorial participants**.

A secret sharing scheme is **democratic of degree t** if every group of t shareholders is in the same number of minimal access sets.

Before establishing the correspondence between minimal codewords and minimal access sets, we give the following well known result.

Proposition III.9. If $\{P_{i_1}, ..., P_{i_m}\}$ is a minimal access set then the corresponding columns $\mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly independent.

Proof: For a minimal access set $\{P_{i_1}, ..., P_{i_m}\}$ suppose that $\mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly dependent. Then

$$\lambda_1\mathbf{g}_{i_1} + ... + \lambda_m\mathbf{g}_{i_m} = 0 \qquad \text{(III.1)}$$

where not all λ_j are 0. Without loss of generality assume $\lambda_1 \neq 0$. So $\mathbf{g}_{i_1}$ can be computed by solving the linear equation (III.1). Therefore the participants $\{P_{i_2}, ..., P_{i_m}\}$ can learn the share of P_{i_1} by combining their shares and hence they can recover the secret which is a contradiction. ■

We are now ready to state the correspondence between minimal codewords and minimal access sets.

Proposition III.10 ([1])**.** There is a 1-1 correspondence between minimal codewords and minimal access sets in the sense that for every minimal access set $\{P_{i_1}, ..., P_{i_m}\}$ there exists a unique minimal codeword
$\mathbf{c} = (1, 0, ..., 0, c_{i_1}, 0, ..., 0, c_{i_m}, 0, ..., 0) \in \mathbf{C}^{\perp}$ such that $c_{i_j} \neq 0$ for $j = 1, ..., m$ and vice versa.

Proof: If $\{P_{i_1}, ..., P_{i_m}\}$ is a minimal access set then the columns $\mathbf{g}_0, \mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly dependent. By proposition III.5, there exist
$\mathbf{a} = (a_0, 0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_m}, 0, ..., 0) \in \mathbf{C}^{\perp}$. $a_0 \neq 0$, otherwise
$(0, 0, ..., 0, a_{i_1}, 0, ..., 0, a_{i_m}, 0, ..., 0) \in \mathbf{C}^{\perp}$ which gives a contradiction by proposition III.5 and III.9.
Set $\mathbf{c} = a_0^{-1}\mathbf{a} = (1, 0, ..., 0, c_{i_1}, 0, ..., 0, c_{i_m}, 0, ..., 0)$. If $c_{i_j} = 0$ for some $j \in \{1, ..., m\}$ then it follows from proposition III.5 that $\{P_{i_1}, ..., P_{i_{j-1}}, P_{i_{j+1}}, ..., P_{i_m}\}$ can recover the secret which is a contradiction. If $\mathbf{c}$ is not a minimal vector then $\mathbf{c}$ covers a vector $\bar{\mathbf{c}} \neq 0$ and $c \neq \lambda\bar{\mathbf{c}}$ for any scalar λ. $\bar{\mathbf{c}}_0 \neq 0$ by proposition III.9. Consider $\bar{\bar{\mathbf{c}}} = \bar{c}\mathbf{c}_0 - \bar{\mathbf{c}} \neq 0$, $\mathbf{c}$ covers $\bar{\bar{\mathbf{c}}}$ and $\bar{\bar{\mathbf{c}}}_0 = 0$. So $\bar{\bar{\mathbf{c}}}$ is of the form $(0, 0, ..., 0, \bar{\bar{c}}_{i_1}, 0, ..., 0, \bar{\bar{c}}_{i_m}, 0, ...0) \in \mathbf{C}^{\perp}$. This is a contradiction by proposition III.5 and III.9 and hence $\mathbf{c}$ is a minimal codeword. Uniqueness of $\mathbf{c}$ is immediate from III.6. For the second part of the proof, if $\mathbf{c} = (1, .., 0, c_{i_1}, 0, ..., 0, c_{i_m}, 0, ...0)$ is a minimal codeword then $\mathbf{g}_0, \mathbf{g}_{i_1}, ..., \mathbf{g}_{i_m}$ are linearly dependent by proposition III.5. So the set of participants $\{P_{i_1}, ..., P_{i_m}\}$ can recover the secret. If any proper subset of this can recover the secret then $\{P_{i_1}, ..., P_{i_{j-1}}, P_{i_{j+1}}, ..., P_{i_m}\}$ can also recover the secret. This implies the existence of a nonzero codeword which $\mathbf{c}$ properly covers. This contradicts the minimality of $\mathbf{c}$. Thus $\{P_{i_1}, ..., P_{i_m}\}$ is a minimal access set. ■

Ding and Yuan pointed out (see [8]) that if every nonzero codeword of $\mathbf{C}^{\perp}$ is minimal then the secret sharing scheme based on $\mathbf{C}$ has a quite interesting and desired access structure as follows.

Proposition III.11. ([8]) Let $\mathbf{C}$ be an [n,k;q] code, let $G = [\mathbf{g}_0, ..., \mathbf{g}_{n-1}]$ be its generator matrix. If each nonzero codeword of $\mathbf{C}$ is a minimal vector, then in the secret sharing scheme based on $\mathbf{C}^{\perp}$, there are altogether q^{k-1} minimal access sets. In addition we have the following:

1) If $\mathbf{g}_i$ is a multiple of $\mathbf{g}_0$, $1 \leq i \leq n-1$, then participant P_i must be in every minimal access set. In other words P_i is a dictatorial participant.
2) If $\mathbf{g}_i$ is not a multiple of $\mathbf{g}_0$, $1 \leq i \leq n-1$, then participant P_i must be in $(q-1)q^{k-2}$ out of q^{k-1} minimal access sets.

This makes the following problem a very interesting one: given a linear code $\mathbf{C}$, determine all the minimal codewords of $\mathbf{C}$. This problem is called the **Covering Problem** (cf. [9]).

A. Example

Assume we are working in $\mathbb{F}_3$ and there are 4 shareholders denoted by $P1, P_2, P_3, P_4$. We first determine a parity check matrix of our code

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 1 & 2 \end{bmatrix}$$

Let $\mathbf{C}$ be the code we are working with. Then $\mathbf{H}$ is a generator of $\mathbf{C}^{\perp}$. A generator matrix for $\mathbf{C}$ is

$$G = \begin{bmatrix} 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = [\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4].$$

Suppose that the secret to be shared is 2. We pick a codeword $\mathbf{t}$ from $\mathbf{C}$ whose first component is 2. Say $\mathbf{t} = (22021)$. The shares are distributed as $K_1 = 2, K_2 = 0, K_3 = 2, K_4 = 1$ where K_i is the share of P_i.
$\mathbf{C}^{\perp} = \{(11121), (00000), (10112), (20221), (22212), (02021), (01012), (12100), (21200)\}$.
The minimal vectors of $\mathbf{C}^{\perp}$ are $(10112), (20221), (02021), (01012), (12100), (21200)$. The minimal codewords are $(10112), (12100)$. The first codeword tells us that the columns $\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2$ are linearly dependent and any proper subset of them is linearly independent and therefore $\{P_2, P_3, P_4\}$ is a minimal access set. Similarly the second codeword indicates that $\{P_1, P_2\}$ is another minimal access set. Note that, since P_2 is in every minimal access set, P_2 is a dictatorial participant.

$\{P_2, P_3, P_4\}$ can recover the secret by first solving the system of equations given by $g_0 = a_2 g_2 + a_3 g_3 + a_4 g_4$. The solution yields $a_2 = 2$, $a_3 = 2$, $a_4 = 1$. Now combining the shares as $a_2 K_2 + a_3 K_3 + a_4 K_4 = 2$, they recover the secret.

Suppose that in the absence of P_4, P_2 and P_3 combine their shares to reconstruct the secret. Here what they need to do is looking for codewords whose 2^{nd} component is 0 and 3^{rd} component is 2 (indexing starts form 0). There are three codewords in $\mathbf{C}$ having this property, they are (00022),(11020) and (22021). The 0^{th} component of each codeword is different. Consequently, combining their shares P_2 and P_3 can not obtain any information about the secret.

IV. Recent Advances

Using perfect nonlinear mappings over finite fields, in [6] authors constructed linear secret sharing schemes and determined their access structure.

Definition IV.1. Let $f : A \to B$ be a function. Let $P_f = \max_{0 \neq a \in A} \max_{b \in B} \frac{|f(b+a)-f(b)|}{|A|}$. f is called perfect nonlinear if $P_f = \frac{1}{|B|}$.

In general, $P_f \geq \frac{1}{|B|}$

Let $\Pi(x) : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ be a perfect nonlinear function. For any $a, b \in \mathbb{F}_{p^m}$ define $f_{a,b}(x) = Tr_{p^m/p^h}(a\Pi(x) + bx)$ we define a linear code $\mathcal{C}_\pi$ over $\mathbb{F}_{p^h}$ as

$\mathcal{C}_\pi = \{c_{a,b} = (f_{a,b}(\gamma_1), \ldots, f_{a,b}(\gamma_{p^m-1}))|\ a, b \in \mathbb{F}_{p^m}\}$

where $\gamma_1, \ldots, \gamma_{p^m-1}$ are all nonzero elements of $\mathbb{F}_{p^m}$.

We denote a linear code which is a subspace of $\mathbb{F}_q^n$ with dimension k and minimum nonzero Hamming weight d by an $[n, k, d; q]$ code.

Theorem IV.2 ([6])**.** Let $\Pi : \mathbb{F}_{p^m} \to \mathbb{F}_{p^m}$ be a perfect nonlinear function with $\Pi(0) = 0$. Then the code $\mathcal{C}_\pi$ has parameters $[p^m - 1, 2m/h, d; p^h]$. Let $G = [\mathbf{g}_0, \ldots, \mathbf{g}_{p^m-2}]$ denote a generator matrix of $\mathcal{C}_\pi$. If $p^h < (p^{m/2}+1)/2$, then in the secret sharing based on $\mathcal{C}_\pi^{\perp}$, the total number of participants is $p^m - 2$, and there are all together p^{2m-h} minimal access sets. Moreover,

1) When $d^{\perp} = 2$, the structure is as follows.
   If $\mathbf{g}_i$ is a multiple of $\mathbf{g}_0$, $1 \leq i \leq p^m - 2$, then participant P_i must be in every minimal access set. In other words P_i is a dictatorial participant
   If $\mathbf{g}_i$ is not a multiple of $\mathbf{g}_0$, $1 \leq i \leq p^m - 2$, then participant P_i must be in $(p^h - 1)p^{2m-2h}$ out of p^{2m-h} minimal access sets.
2) When $d^{\perp} \geq 3$, we have the following structure.
   For any fixed $1 \leq t \leq min\{(2m/h) - 1, d^{\perp} - 2\}$ every group of t participants is involved in $(p^h - 1)^t p^{2m-(t+1)h}$ out of p^{2m-h} minimal access sets.

The next theorem tells us that it is also possible to construct democratic secret sharing schemes of degree 1 using perfect nonlinear mappings, regardless of minimum Hamming weight.

Theorem IV.3 ([6])**.** Let $m > 1$, and let $\mathcal{C}_\pi$ be the $[p^m - 1, 2m/h, d; p^h]$ code from the perfect nonlinear function $\Pi(x) = x^{p^k+1}$. If $p^h < (p^{m/2} + 1)/2$, then in the secret sharing scheme based on $\mathcal{C}_\pi^{\perp}$, the total number of participants is $p^m - 2$, and there are altogether p^{2m-h} minimal access sets. In addition, every participant is involved in $(p^h - 1)p^{2m-h}$ minimal access sets.

In [9], authors use the following result to construct linear secret sharing schemes. Their construction potentially admits

dictatorial participants.

Proposition IV.4 ([2] and [8])**.** In an [n,k;q] code **C**, let w_{min} and w_{max} be the minimum and maximum nonzero weights respectively. If

$$\frac{w_{min}}{w_{max}} > \frac{q-1}{q} \qquad \text{(IV.1)}$$

then each nonzero codeword of **C** is a minimal vector.

Briefly, this proposition tells us that if the weights of codewords are close enough to each other then every nonzero codeword is a minimal vector.

Theorem IV.5. [9] Let p be an odd prime and let $q = p^k$. Suppose $N|q-1$ and $nN = q-1$. Let θ be a primitive nth root of unity in $\mathbb{F}_q$, $\gamma \in \mathbb{F}_q*$, and $\beta \in \mathbb{F}_q$ with $\gamma^n \neq \beta^n$. Define $c(\xi) = (Tr(\xi(\gamma-\beta)), Tr(\xi(\gamma\theta-\beta)), \ldots, Tr(\xi(\gamma\theta^{n-1}-\beta)))$ where $Tr(\xi)$ is the trace function from $\mathbb{F}_q$ to $\mathbb{F}_p$. Define **C** as $\mathbf{C} = \{c(\xi) : \xi \in \mathbb{F}_q\}$. If

$$N - 1 < \frac{(p-1)q - p^2}{(2p-1)(p-1)\sqrt{q}}$$

then all nonzero codewords of **C** are minimal. Furthermore, in the secret sharing scheme based on $\mathbf{C}^{\perp}$, the set of all dictatorial participants is given by

$$\left\{ i : 1 \leq i \leq n-1 \ and \ \frac{\gamma\theta^i - \beta}{\gamma - \beta} \in \mathbb{F}_p \right\}$$

Each of the other participants is involved in $(p-1)p^{k-2}$ minimal access sets.

As we mentioned, McEliece and Sarwate pointed out the relationship with Shamir's scheme and Reed-Solomon codes in [14]. In [15] authors take this idea one step further and use MDS codes to construct (n,k) threshold schemes. Their scheme has the advantage of detecting and identifying cheaters who are participants giving incorrect shares to the dealer.

In [7], authors use algebraic geometric codes over elliptic curves to construct linear secret sharing schemes. They deduce that many of the resulting codes give MDS linear secret sharing schemes.

V. Conclusion

Although classical secret sharing schemes and their extensions are efficient and secure, their access structure do not meet our needs under certain circumstances. We have seen how error-correcting codes can be used to construct secret sharing schemes. The access structure of Massey's linear secret sharing scheme turned out to be very useful and interesting. Characteristics of this access structure depend on the underlying dual code and its analysis make some coding theoretic problems interesting like the covering problem.

References

[1] R. Anderson, C. Ding, T. Helleseth, and T. Klove. How to build robust shared control systems. *Designs, Codes and Cryptography*, 15:111–124, November 1998.

[2] A. Ashikhmin and A. Barg. Minimal vectors in linear codes. *IEEE Transactions on Information Theory*, 44(5):2010–2017, September 1998.

[3] C. Asmuth and J. Bloom. A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, 29:208–210, March 1983.

[4] G.R. Blakely. Safeguarding cryptographic keys. In *National Computer Conference*, pages 313–317, 1979.

[5] E. F. Brickell. Ideal secret sharing schemes. In *Advances in Cryptology-Eurocrypt89 (Lecture Notes in Computer Science)*, volume 434, pages 468–475. Springer, 1990.

[6] C. Carlet, C. Ding, and J. Yuan. Secret sharing schemes from three classes of linear codes. *IEEE Transactions on Information Theory*, 51:2089–2102, June 2005.

[7] H. Chen and R. Kramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In *CRYPTO 2006*, pages 521–536. Springer-Verlag, 2006.

[8] C. Ding and J. Yuan. Covering and secret sharing with linear codes. In *Discrete Mathematics and Theoretical Computer Science (Lecture Notes in Computer Science)*, volume 2731, pages 11–25. Springer-Verlag, 2003.

[9] C. Ding and J. Yuan. Secret sharing schemes from three classes of linear codes. *IEEE Transactions on Information Theory*, 52:206–212, January 2006.

[10] E. D. Karnin, J. W. Greene, and M. E. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29(1):35–41, January 1983.

[11] D. Knuth. *The Art of Computer Programming*. Addison-Wesley, 1969.

[12] C.L. Liu. *Introduction to Combinatorial mathematics*. McGraw-Hill, 1968.

[13] J.L. Massey. Minimal codewords and secret sharing. In *6th Joint Swedish-Russian Workshop On Information Theory*, pages 276–279, 1993.

[14] R.J. McEliece and D.V. Sarwate. On sharing secrets and reed-solomon codes. *Communications of the ACM*, 24:583–584, 1981.

[15] J. Pieprzyk and X.M. Zhang. Ideal threshold schemes from mds codes. *Discrete Mathematics and Theoretical Computer Science*, 6(2):471–482, 2004.

[16] A. Shamir. How to share a secret. *Communications of the ACM*, 24:612–613, September 1979.

[17] T. Xiaoqing and W. Zhiguo. New secret sharing scheme based on linear code. *Applied Mathematics - A Journal of Chinese Universities*, 19(2):160–166, June 2004.