On a Relation Between Verifiable Secret Sharing Schemes and a Class of Error-Correcting Codes

Ventzislav Nikov[1] and Svetla Nikova[2]

[1] Department of Mathematics and Computing Science,
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands
v.nikov@tue.nl

[2] Department Electrical Engineering, ESAT/COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10,
B-3001 Heverlee-Leuven, Belgium
svetla.nikova@esat.kuleuven.ac.be

Abstract In this paper we try to shed a new insight on Verifiable Secret Sharing Schemes (VSS). We first define a new "metric" (with slightly different properties than the standard Hamming metric). Using this metric we define a very particular class of codes that we call *error-set codes*, based on a set of forbidden distances which is a monotone decreasing set. Next we redefine the packing problem for the new settings and generalize the notion of error-correcting capability of the error-set codes accordingly (taking into account the new metric and the new packing). Then we consider burst-error interleaving codes proposing an efficient burst-error correcting technique, which is in fact the well known VSS pair-wise checking protocol and we prove the error-correcting capability of the error-set interleaving codes.

Using the known relationship from [8] between a Monotone Span Program (MSP) and a generator matrix of the code generated by the suitable set of vectors, we prove that the error-set codes in fact has the allowed (opposite to forbidden) distances of the dual access structure of the access structure that the MSP computes. This relation gives us an efficient construction for them and as a consequence we establish a link between Secret Sharing Schemes (SSS) and the error-set codes.

Further we give a necessary and sufficient condition for the existence of linear SSS (LSSS), to be secure against (Δ, Δ_A)-adversary expressed in terms of an error-set code. Finally, we present necessary and sufficient conditions for the existence of a VSS scheme, based on an error-set code, secure against (Δ, Δ_A)-adversary.

Our approach is general and covers all known linear VSS. It allows us to establish the minimal conditions for security of VSSs. Our main theorem states that the security of a scheme is equivalent to a pure geometrical (coding) condition on the linear mappings describing the scheme. Hence the security of all known schemes, e.g. all known bounds for existence of VSS including the recent result of Fehr and Maurer [10], can be expressed as certain (geometrical) coding conditions.

1 Preliminaries

The concept of *secret sharing* was introduced by Shamir [19] as a tool to protect a secret from getting exposed or from getting lost. It allows a so-called *dealer* to share a secret among the members of a set $\mathcal{P}$, which are usually called *players* or *participants*, in such a way that only certain specified subsets of players are able to reconstruct the secret (if needed) while smaller subsets have no information about this secret at all (in a strict information theoretic sense).

We call the groups who are allowed to reconstruct the secret *qualified* and the groups who should not be able to obtain any information about the secret *forbidden*. The set of qualified groups is denoted by Γ and the set of forbidden groups by Δ. Denote the participants by $P_i,\ 1 \le i \le n$ and the set of all players by $\mathcal{P} = \{P_1, \ldots, P_n\}$. The set Γ is called *monotone increasing* if for each set A in Γ also each set containing A is in Γ. Similarly, Γ is called *monotone decreasing*, if for each set A in Δ also each subset of A is in Δ. A monotone increasing set Γ can be efficiently described by the set Γ^- consisting of the minimal elements in Γ, i.e. the elements in Γ for which no proper subset is also in Γ. Similarly, the set Δ^+ consists of the maximal sets in Δ. The tuple (Γ, Δ) is called an *access structure* if $\Gamma \cap \Delta = \emptyset$. If the union of Γ and Δ is equal to $2^{\mathcal{P}}$ (so Γ is equal to Δ^c, the complement of Δ), then we say that access structure (Γ, Δ) is *complete* and we denote it just by Γ. In the sequel we shall only consider complete, monotone access structures.

The dual $\Gamma^\perp$ of an access structure Γ, defined on $\mathcal{P}$, is the collection of sets $A \subseteq \mathcal{P}$ such that $\mathcal{P} \setminus A = A^c \notin \Gamma$.

It is common to model cheating by considering an *adversary* $\mathcal{A}$ who may corrupt some of the players. The adversary is characterized by particular subset Δ_A of Δ, called *adversary and privacy structure* [12] respectively, which is itself monotone decreasing structure. The players which belong to Δ are called also *curious* and the players which belong to Δ_A are called *corrupt* or *bad*.

One can distinguish between *passive* and *active* corruption, see Fehr and Maurer [10] for recent results. Passive corruption means that the adversary obtains the complete information held by the corrupt players, but the players execute the protocol correctly. Active corruption means that the adversary takes full control of the corrupt players. Active corruption is strictly stronger than passive corruption. Both passive and active adversaries may be *static*, meaning that the set of corrupt players is chosen once and for all before the protocol starts, or *adaptive* meaning that the adversary can at any time during the protocol choose to corrupt a new player based on all the information he has at the time, as long as the total set is in Δ_A.

Denote the complement $\Gamma_A = 2^{\mathcal{P}} \setminus \Delta_A = \Delta_A^c$. Its dual access structure $\Gamma_A^\perp$ should be called the honest (or good) players structure, since for any set A of corrupt players, i.e. in Δ_A, the complement $A^c = \mathcal{P} \setminus A$ is the set of honest players and vise versa. Note that the set $\{A^c : A \in \Delta_A\}$ is the dual access structure $\Gamma_A^\perp$.

Some authors [11] consider also *fail-corrupt* players. To fail-corrupt a player means that the adversary may stop the communication from and to that player at an arbitrary time during the protocol. Once a player is caused to fail, he will not recover the communication. However, the adversary is not allowed to read the internal data of a fail-corrupt player, unless the player is also passively corrupted at the same time. The collection of fail-corrupt players is denoted by $\Delta_F \subseteq \Delta$. Generally we will not consider such kind of corruption, so unless it is exact mentioned we will assume that the adversary cannot fail-corrupt the players.

Definition 1. *[10] An $(\Delta, \Delta_A, \Delta_F)$-adversary is an adversary who can (adaptively) corrupt some players passively and some players actively, as long as the set A of actively corrupt players and the set B of passively corrupt players satisfy both*

$$A \in \Delta_A \quad \text{and} \quad (A \cup B) \in \Delta$$

Additionally the adversary could fail-corrupt some players in Δ_F. When $\Delta_F = \emptyset$ we will denote it by (Δ, Δ_A), in case $\Delta_A = \Delta$ we will simply say Δ_A-adversary.

This model is known as *mixed adversary* model. Note that in case of Secret Sharing Schemes we have $\Delta_A = \emptyset$, while for Verifiable Secret Sharing Schemes we have $\Delta_A \neq \emptyset$. In the threshold case we write instead of $(\Delta, \Delta_A, \Delta_F)$-adversary simply (k, k_a, k_f)-adversary. Recently Hirt and Maurer [12] introduced the notion of $\mathcal{Q}^2(\mathcal{Q}^3)$ adversary structure.

Definition 2. *[12] For a given set of players $\mathcal{P}$ and an adversary structure Δ_A, we say that the adversary structure is $\mathcal{Q}^\ell$ if no ℓ sets in Δ_A cover the full set $\mathcal{P}$ of players.*

Definition 3. *[17] For any two monotone* decreasing *sets Δ_1, Δ_2 operation $\uplus$, called element-wise union, is defined as follows: $\Delta_1 \uplus \Delta_2 = \{A = A_1 \cup A_2; A_1 \in \Delta_1, A_2 \in \Delta_2\}$. For any two monotone* increasing *sets Γ_1, Γ_2 operation $\uplus$ is defined as follows: $\Gamma_1 \uplus \Gamma_2 = \{A = A_1 \cup A_2; A_1 \notin \Gamma_1, A_2 \notin \Gamma_2\}^c$.*

Definition 4. *A secret sharing scheme based on an access structure (Γ, Δ) is a pair* (Share and Reconstruct) *of protocols (phases) namely, the* sharing phase*, where the players share a secret $s \in \mathcal{K}$, and the* reconstruction phase*, where the players try to reconstruct s, such that the following two properties hold:*

- Privacy: *The players of any set $B \in \Delta$ learn nothing about the secret s as a result of the sharing phase.*
- Correctness: *The secret s could be computed by any set of players $A \in \Gamma$.*

Recall that the SSS is called perfect if and only if $\Delta^c = \Gamma$.

2 A class of Error-Correcting "Codes"

Let $\mathbb{F}$ be a finite field and let the set of secrets for the dealer $\mathcal{D}$ be $\mathcal{K} = \mathbb{F}^{p_0}$. We will only consider the case $p_0 = 1$, even though many of the considerations remain valid in the general case too. Associate with each player P_i $(1 \leq i \leq n)$ a positive integer p_i such that the sets of possible shares for player P_i is given by $\mathbb{F}^{p_i}$. Denote by $p = \sum_{i=1}^{n} p_i$ and by $N = p_0 + p$. For the sake of simplicity one could assume that $p_i = 1$ for $0 \leq i \leq n$ in that case $p = n$ and $N = n + 1$ hold.
Now we will recall some definitions from the theory of error-correcting codes. Any non-empty subset $\mathcal{C}$ of $\mathbb{F}^N$ is called a code, the parameter N is called the *length* of the code. Each vector in $\mathcal{C}$ is called *codeword* of $\mathcal{C}$.
The *Hamming sphere* (or *ball*) $B_e(\mathbf{x})$ of radius e around a vector $\mathbf{x}$ in $\mathbb{F}^N$ is defined by $B_e(\mathbf{x}) = \{\mathbf{y} \in \mathbb{F}^N : d(\mathbf{x}, \mathbf{y}) \leq e\}$.

One of the basic coding theory problems is the so-called *Sphere Packing Problem:* given N and e, what is the maximum number of non-intersecting spheres of radius e that can be placed in $\mathbb{F}^N$, the N-dimensional Hamming space?
Sphere packing is related to *error correction*. The centers of these spheres are at distance at least $2e+1$ apart from each other and constitute a *code*; these centers are called *codewords* and each corresponds to a possible message that one may want to transmit. Assume now that one of these messages is transmitted and that at most e coordinates are corrupt during the transmission. To decode, i.e., to decide which of the messages was actually sent, compute the Hamming distance between the received vector and all the centers. Since at most e errors occurred, the transmitted word will still be the nearest center, and all errors can be corrected.
Define the *minimum distance* of a code $\mathcal{C} \subseteq \mathbb{F}^N$ as the smallest of all distances between different codewords in $\mathcal{C}$, i.e.

$$d_{min} = \min_{\mathbf{a},\mathbf{b} \in \mathcal{C},\ \mathbf{a} \neq \mathbf{b}} d(\mathbf{a}, \mathbf{b}) \tag{1}$$

It follows from this definition that a code with minimum distance d_{min} can correct $\lfloor (d_{min}-1)/2 \rfloor$ errors, since spheres with this radius are disjoint (see [16, p.10, Theorem 2]). If d_{min} is even the code can *detect* $d_{min}/2$ errors, meaning that a received word can not have distance $d_{min}/2$ to one codeword and distance less that $d_{min}/2$ to another one. However it may have distance $d_{min}/2$ to more codewords.
Something more actually can be said. Code $\mathcal{C}$ can decode errors and *erasures* simultaneously. An erasure is an ambiguously received coordinate (the value is not 0 or 1 but undecided). Let $\mathcal{C}$ be a code of length N with minimum distance d_{min} and let $e = \lfloor (d_{min} - 1)/2 \rfloor$. Then the code can correct b errors and c erasures as long as $2b + c < d_{min}$ (for more details see [6]). In other words, we should be able to retrieve the transmitted codeword if during the transmission at most c of the symbols in the word are erased and at most b received symbols are incorrect.
If $\mathcal{C}$ is a T-dimensional subspace of $\mathbb{F}^N$, then the code $\mathcal{C}$ is *linear* and is denoted by $[N, T, d_{min}]$. Set $\mathcal{C}^\perp = \{\mathbf{y} \mid \langle \mathbf{y}, \mathbf{x} \rangle = 0 \text{ for all } \mathbf{x} \in \mathcal{C}\}$. The set $\mathcal{C}^\perp$ is an $(N-T)$-dimensional linear subspace of $\mathbb{F}^N$ and is called the *dual code* of $\mathcal{C}$.
There are two methods to determine a linear code $\mathcal{C}$: a *generator matrix* and a *parity check matrix*. A *generator matrix* of a linear code $\mathcal{C}$ is any $T \times N$ matrix G whose rows form a basis for $\mathcal{C}$. A generator matrix H of $\mathcal{C}^\perp$ is called a *parity check matrix* for $\mathcal{C}$. Clearly, the matrix H is of size $(N-T) \times N$. Hence $\mathbf{x} \in \mathcal{C}$ if and only if $H\mathbf{x}^T = \mathbf{0}$, or in other words $HG^T = GH^T = 0$ holds.
When a *sender* want to send a message (sometimes called *information vector*) say $\mathbf{x}$ to the *receiver* he calculates a codeword of the code by multiplying the information vector with the generator matrix, e.g. $\mathbf{y} = \mathbf{x}G$. The codeword $\mathbf{y}$ is transmitted to the receiver. The receiver decodes the word $\mathbf{z}$ he received, which is the codeword plus errors, i.e. $\mathbf{z} = \mathbf{y} + \mathbf{err}$, if the number of errors is less than a certain number (the error-correcting capabilities of the code). Recall that for each codeword $\mathbf{y}$ the equality $H\mathbf{y}^T = \mathbf{0}$ holds, hence $H\mathbf{z}^T = \mathbf{err}$ (called *syndrome*) hold.

Let for two vectors $\mathbf{x} = (\mathbf{x^0}, \mathbf{x^1}, \dots, \mathbf{x^n})$ and $\mathbf{y} = (\mathbf{y^0}, \mathbf{y^1}, \dots, \mathbf{y^n})$ in $\mathbb{F}^N$, where $\mathbf{x^i}, \mathbf{y^i} \in \mathbb{F}^{p_i}$, the set $\delta_p(\mathbf{x}, \mathbf{y})$ is defined by $\delta_p(\mathbf{x}, \mathbf{y}) = \{i : \mathbf{x^i} \neq \mathbf{y^i}\}$. The *p-support* of vector $\mathbf{x}$, denoted by $\sup_p(\mathbf{x})$, is defined by $\sup_p(\mathbf{v}) = \{i \;:\; \mathbf{v^i} \neq \mathbf{0}\}$. Hence $\delta_p(\mathbf{x}, \mathbf{y}) = \sup_p(\mathbf{x} - \mathbf{y}) \subseteq \{0, \dots, n\}$. Considering the properties of the p-support of a vector, we notice some similarities to the properties of the norm.

(1). $\sup_p(\mathbf{x}) = \emptyset$ if and only if $\mathbf{x} = \mathbf{0}$,
(2). $\sup_p(j\mathbf{x}) = \sup_p(\mathbf{x})$ if $j \neq 0$, and
(3). $\sup_p(\mathbf{x} + \mathbf{z}) \subseteq \sup_p(\mathbf{x}) \cup \sup_p(\mathbf{y})$.

In his paper [10] Fehr and Maurer pointed out that $\delta_p(\mathbf{x}, \mathbf{y})$ behaves like a metric, as for all vectors $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^N$ one has that

(1). $\delta_p(\mathbf{x}, \mathbf{x}) = \emptyset$,
(2). $\delta_p(\mathbf{x}, \mathbf{y}) = \delta_p(\mathbf{y}, \mathbf{x})$ (symmetry), and
(3). $\delta_p(\mathbf{x}, \mathbf{z}) \subseteq \delta_p(\mathbf{x}, \mathbf{y}) \cup \delta_p(\mathbf{y}, \mathbf{z})$,

but actually they do not explore this property. Our first step is to use $\delta_p(\mathbf{x}, \mathbf{y})$ instead of the Hamming distance and to explore the properties of the so defined space.

Let Δ be a monotone decreasing collection of subsets of players. Then $B_\Delta(\mathbf{x})$, the Δ-*neighborhood* of pseudo-radii in Δ centered around the vector $\mathbf{x} \in \mathbb{F}^N$, is defined as follows:

$$B_\Delta(\mathbf{x}) = \{\mathbf{y} \in \mathbb{F}^N : \delta_p(\mathbf{x}, \mathbf{y}) \in \Delta\}$$

In the special case when Γ is an *e-threshold* access structure ($\Delta = \{A : |A| \leq e\}$), the Δ-neighborhood $B_\Delta(\mathbf{x})$ is in fact the Hamming sphere $B_e(\mathbf{x})$. Now we can generalize the classical sphere packing problem:

Generalized Sphere Packing Problem: Given N and Δ, what is the maximum number of non-intersecting Δ-neighborhoods that can be placed in the N-dimensional space?

As usual we will call any non-empty subset $\mathcal{C}$ of $\mathbb{F}^N$ a code. For a code $\mathcal{C}$ the *set of possible (allowed) distances* is defined by

$$\Gamma(\mathcal{C}) = \{A : \text{ there exist } \mathbf{a}, \mathbf{b} \text{ in } \mathcal{C},\ \mathbf{a} \neq \mathbf{b} \text{ such that } \delta_p(\mathbf{a}, \mathbf{b}) \subseteq A\} \tag{2}$$

and the *set of forbidden distances* is defined by

$$\Delta(\mathcal{C}) = \Gamma(\mathcal{C})^c. \tag{3}$$

It is easy to see that $\Delta(\mathcal{C})$ is monotone decreasing and that $\Gamma(\mathcal{C})$ is monotone increasing. Let us call the so-defined codes *"error-set codes"*. For the classical error-correcting codes we have $\Delta(\mathcal{C}) = \{A : |A| < d_{min}\}$. In fact we can define the set of *minimal* codeword support differences as

$$\begin{aligned}\Gamma(\mathcal{C})^- = \{A : &\text{there exist } \mathbf{a}, \mathbf{b} \text{ in } \mathcal{C},\ \mathbf{a} \neq \mathbf{b} \text{ such that } \delta_p(\mathbf{a}, \mathbf{b}) = A \\ &\text{but, there is no } \mathbf{c}, \mathbf{d} \in \mathcal{C},\ \mathbf{c} \neq \mathbf{d},\ \delta_p(\mathbf{c}, \mathbf{d}) \subsetneq A\}.\end{aligned} \tag{4}$$

We will focus our attention only on linear codes, even though many of the considerations remain valid in non-linear settings too. Hence using the relation

between δ_p and $\sup_p$ we could redefine the notion *minimal codeword* (introduced by Massey [14] and generalized by Van Dijk [8]) as follows: The codeword $\mathbf{x}$ in $\mathcal{C}$ is *minimal* if $\sup_p(\mathbf{x})$ in $\Gamma(\mathcal{C})^-$.
As noted before, the packing problem is fundamental in error correction. The natural question that arises now is how the new packing problem is related to the theory of error-correcting codes?
In coding theory, any subset of coordinates is equally likely to be in error (and/or erasure). In the model we consider here some subsets of coordinates (those in $\Delta(\mathcal{C})$) are assumed to be more likely in error than others (those in $\Gamma(\mathcal{C})$). A well-studied model where this situation arises is the so-called *bursty* channel, in which errors occur in clusters. Another related approach are the so-called D-codes [9] which have restricted (to some interval) inner distance distribution.
Now we will prove that the error-set codes have similar error-correcting capabilities as the classical codes have.

Theorem 1. *An error-set code $\mathcal{C}$ with set of forbidden distances $\Delta(\mathcal{C})$ can correct all errors in Δ if and only if $\Delta \uplus \Delta \subseteq \Delta(\mathcal{C})$.*

Proof. First we will prove that the centers of a new sphere packing constitute a code $\mathcal{C}$ with set of possible distances $\Gamma(\mathcal{C}) \subseteq \Gamma \uplus \Gamma$ (and thus $\Delta \uplus \Delta \subseteq \Delta(\mathcal{C})$). Indeed, let $\mathbf{a}, \mathbf{b}$ be any two distinct centers of $\mathcal{C}$. Any two sets $A, B \in \Delta$ are in the Δ-neighborhoods of say $\mathbf{a}$, resp. $\mathbf{b}$. Since these neighborhoods are non-intersecting we have that $A \cup B \subset \delta_p(\mathbf{a}, \mathbf{b})$. Hence $\delta_p(\mathbf{a}, \mathbf{b}) \notin \Delta \uplus \Delta$. Conversely, suppose that $\delta_p(\mathbf{a}, \mathbf{b}) \in \Delta \uplus \Delta$. Then there exist $A, B \in \Delta$, such that $A \cup B = \delta_p(\mathbf{a}, \mathbf{b})$. By the "triangle inequality" we have that $\delta_p(\mathbf{a}, \mathbf{b}) \subseteq \delta_p(\mathbf{a}, \mathbf{x}) \cup \delta_p(\mathbf{x}, \mathbf{b})$, and equality holds if $\delta_p(\mathbf{a}, \mathbf{x}) \cap \delta_p(\mathbf{b}, \mathbf{x}) = \emptyset$. Now it is easy to see that there exists $\mathbf{x}$ such that $A \cup B = \delta_p(\mathbf{a}, \mathbf{b}) = \delta_p(\mathbf{a}, \mathbf{x}) \cup \delta_p(\mathbf{b}, \mathbf{x})$ and $\delta_p(\mathbf{a}, \mathbf{x}) \subseteq A$, $\delta_p(\mathbf{b}, \mathbf{x}) \subseteq B$. This contradicts the fact that any Δ-neighborhoods of $\mathbf{a}$ and $\mathbf{b}$ are non-intersecting. $\square$

Example 1. Consider the special case with threshold access structure, so $\Delta = \{A : |A| \leq e\}$. Write as above $B_\Delta(\mathbf{x}) = B_e(\mathbf{x})$ (the usual Hamming sphere). Now $\Delta \uplus \Delta = \{A : |A| \leq 2e\} = \Delta(\mathcal{C})$ and so $\Gamma(\mathcal{C}) = \{A : |A| \geq 2e+1\}$. Hence the minimum distance of $\mathcal{C}$ is $d_{min} = 2e+1$. In this case, Theorem 1 is equivalent to the classical error-correcting theorem [16, 6].

Remark 1. Assume that a codeword from $\mathcal{C}$ was sent and that some subset of errors $A \in \Delta$ occurred during the transmission. To decode the received vector $\mathbf{z}$, we compute the Δ-neighborhood $B_\Delta(\mathbf{z})$ and check which codeword is in this Δ-neighborhood. In fact, since the error-pattern is a set A in Δ and $\Delta \uplus \Delta \subseteq \Delta(\mathcal{C})$, there will be only one codeword in the Δ-neighborhood of $\mathbf{z}$ and so we can correct the errors.
Something more actually is true: we can decode errors and erasures simultaneously in the generalized setting too.
Let $\mathcal{C}$ be a code of length N with set of forbidden distances $\Delta(\mathcal{C})$. Suppose that $\Delta \uplus \Delta \subseteq \Delta(\mathcal{C})$. Then $\mathcal{C}$ can correct all errors in Δ. Moreover, for any $\Delta_c, \Delta_b \subseteq \Delta$ such that $\Delta_c \uplus \Delta_c \uplus \Delta_b \subseteq \Delta(\mathcal{C})$, the code $\mathcal{C}$ can correct all errors in Δ_c and erasures in Δ_b. In fact, the decoding method coincides with the classical method of decoding errors and erasures, see [6] for example.

3 A Burst-Correcting Technique

We will call a *burst* any error pattern consisting of several sub-vectors $\mathbf{x^i}$ of $\mathbf{x} = (\mathbf{x^0}, \mathbf{x^1}, \ldots, \mathbf{x^n})$, which are not necessarily consequtively ordered.
First, we will present a standard burst-error correcting technique, which uses error-correcting codes. The idea is to change the order of the coordinates of several consecutive codewords in such a way that a burst is spread out over the various codewords. Let $\mathcal{C}$ be a code of length n and let ℓ be some positive integer. Consider an $\ell \times n$ matrix which has codewords in $\mathcal{C}$ as their rows. Read this matrix column-wise from top to bottom starting with the leftmost column. The resulting codewords have length $n\ell$ and form a so-called *interleaved code* derived from $\mathcal{C}$ at depth ℓ. If $\mathcal{C}$ can correct e-errors then the interlieved code can correct bursts of length $e\ell$.
Let $\mathcal{C}$ be an error-set code of length N, with a set of forbidden distances $\Delta(\mathcal{C})$ and generator $d \times N$ matrix G. The sender wants to send an information matrix $X \in \mathbb{F}^{d\times d}$ (assume for the sake of simplicity that X is symmetric). Note that X could be asymmetric too, in which case X and X^T are encoded. Thus the sender calculates the (array) codeword Y as $Y = XG$, ($Y \in \mathbb{F}^{N\times N}$). Then applying the interleaving approach the sender reads the matrix column-wise. From now on we will consider only intelieved codes at depth d.

Theorem 2. *Let $\mathcal{C}$ be an error-set code of length N, with set of forbidden distances $\Delta(\mathcal{C})$. Then the interleaving error-set code derived from $\mathcal{C}$ of length N can efficiently correct all burst-errors in Δ if and only if $\Delta \uplus \Delta \subseteq \Delta(\mathcal{C})$.*

Proof. Since every row in the array codeword is a codeword of the error-set code $\mathcal{C}$ and the errors are spread we can correct them row by row (see Theorem 1). On the other hand we will show that the known VSS technique called "pair-wise" checking, provides efficient detection of inconsistency in cases with excess of information. Moreover this technique has an additional advantage that all checks can be performed privately (which is of great importance in SSS).
The "pair-wise" technique is applied as follows. The receiver calculates a symmetric consistency $n \times n$ matrix, verifying the equation $G^T Y = G^T XG = Y^T G$. In other words he puts on entry (i, j) - 1 if $G_i^T Y_j = Y_i^T G_j$ and 0 otherwise. Using the consistency matrix (as in the VSS protocols, e.g. [5]) and assuming an error pattern in Δ occurs it is easy to find a set in $\Gamma(\mathcal{C})$ which is consistent, therefore uniquely define the codeword. □

Remark 2. The interleaving error-set code derived from $\mathcal{C}$ of length N can efficiently correct the burst-error patterns in Δ_c and burst-erasure patterns Δ_b if and only if $\Delta_c \uplus \Delta_c \uplus \Delta_b \subseteq \Delta(\mathcal{C})$.

4 SSS as an Example of a Particular Class of "Codes"

First we give a formal definition of a Monotone Span Program.

Definition 5. *[13] A* Monotone Span Program *(MSP) $\mathcal{M}$ is a quadruple $(\mathbb{F}, M, \boldsymbol{\varepsilon}, \psi)$, where $\mathbb{F}$ is a finite field, M is a matrix (with m rows and $d \leq m$*

columns) over $\mathbb{F}$, $\psi : \{1, \ldots, m\} \to \{1, \ldots, n\}$ *is a surjective function and* $\boldsymbol{\varepsilon}$ *is a fixed non-zero vector, called* target vector*, e.g. column vector* $(1, 0, \ldots, 0) \in \mathbb{F}^d$. *The size of* $\mathcal{M}$ *is the number* m *of rows and is denoted as size*$(\mathcal{M})$.

As ψ labels each row with a number i from $[1, \ldots, m]$ that corresponds to player $P_{\psi(i)}$, we can think of each player as being the "owner" of one or more rows. Also consider a "function" φ from $[P_1, \ldots, P_n]$ to $[1, \ldots, m]$ which gives for every player P_i the set of rows owned by him (denoted by $\varphi(P_i)$). In some sense φ is "inverse" of ψ. For any set of players $B \subseteq \mathcal{P}$ consider the matrix consisting of rows these players own in M, i.e. $M_{\varphi(B)}$. As is common, we shall shorten the notation $M_{\varphi(B)}$ to just M_B. The reader should stay aware of the difference between M_B for $B \subseteq \mathcal{P}$ and for $B \subseteq \{1, \ldots, m\}$.
An MSP is said *to compute* a (complete) access structure Γ when $\boldsymbol{\varepsilon} \in \operatorname{im}(M_A^T)$ if and only if A is a member of Γ. In other words, the players in A can reconstruct the secret precisely if the rows they own contain in their linear span the target vector of $\mathcal{M}$, and otherwise they get no information about the secret. In other words there exists a so-called *recombination vector* (column) $\boldsymbol{\lambda}$ such that $M_A^T \boldsymbol{\lambda} = \boldsymbol{\varepsilon}$ hence $\langle \boldsymbol{\lambda}, M_A(s, \boldsymbol{\rho})^T \rangle = \langle M_A^T \boldsymbol{\lambda}, (s, \boldsymbol{\rho})^T \rangle = \langle \boldsymbol{\varepsilon}, (s, \boldsymbol{\rho})^T \rangle = s$ for any secret s and any random vector $\boldsymbol{\rho}$. It is easy to check that the vector $\boldsymbol{\varepsilon} \notin \operatorname{im}(M_B^T)$ if and only if there exists a $\boldsymbol{k} \in \mathbb{F}^d$ such that $M_B \boldsymbol{k} = \mathbf{0}$ and $\boldsymbol{k}_1 = 1$.
We stress here that

$$\begin{aligned} A \in \Gamma &\iff \exists \;\; \boldsymbol{\lambda} \in \mathbb{F}^{|\varphi(A)|} \;\; \text{such that} \;\; M_A^T \boldsymbol{\lambda} = \boldsymbol{\varepsilon} \\ B \notin \Gamma &\iff \exists \;\; \boldsymbol{k} \in \mathbb{F}^d \;\; \text{such that} \;\; M_B \boldsymbol{k} = \mathbf{0} \;\; \text{and} \;\; \boldsymbol{k}_1 = 1. \end{aligned} \tag{5}$$

The first property guaranties *correctness* and the second *privacy* of the SSS. Technically the property (5) means that when we consider the restricted matrix M_A for some subset A of $\mathcal{P}$, the first column is linearly dependent to the other columns if and only if $A \notin \Gamma$. Sometimes we will slightly change the first property rewriting it in the following way:

$$A \in \Gamma \iff \exists \;\; \boldsymbol{\lambda} \in \mathbb{F}^m \;\; \text{such that} \;\; M_A^T \boldsymbol{\lambda} = \boldsymbol{\varepsilon} \;\; \text{and} \;\; \sup_p(\boldsymbol{\lambda}) \subseteq A. \tag{6}$$

The latest in fact is the same vector $\boldsymbol{\lambda}$ as in (5), but expanded with zeroes.

Definition 6. *([8, Definition 3.2.2]) Let* $\Gamma^- = \{X_1, \ldots, X_r\}$. *Then the set of vectors* $C = \{\mathbf{c^i} \in \mathbb{F}^m : 1 \leq i \leq r\}$ *is said to be* suitable *for the access structure* Γ *if* C *satisfies the following properties called* $g(\Gamma)$ *respectively* $d^-(\Delta)$.

- $sup_p(\mathbf{c^i}) = X_i$ *for* $1 \leq i \leq r$;
- *For any vector* $(\mu_1, \ldots, \mu_r)$ *in* $\mathbb{F}^r$*, such that* $\sum_{i=1}^r \mu_i \neq 0$*, there exists a set* $X \in \Gamma = \Delta^c$ *satisfying* $X \subseteq sup_p(\sum_{i=1}^r \mu_i \mathbf{c^i})$.

It is easy to verify that the minimal codewords defined by Massey [14] are particular case of the more general notion suitable set.
In the next theorem Van Dijk provides an important link between a parity check matrix of a code generated as a span of suitable vectors and an MSP matrix.

Theorem 3. *([8, Theorem 3.2.5, Theorem 3.2.6]) Let* $\Gamma^- = \{X_1, \ldots, X_r\}$. *Consider a set of vectors* $C = \{\mathbf{c^i} : 1 \leq i \leq r\}$. *Let* H *be a parity check matrix* *of the code generated by the linear span of the vectors* $(1, \mathbf{c^i})$ $1 \leq i \leq r$ *and let* H *be of the form* $H = (\boldsymbol{\varepsilon} \mid H')$ *(This can be assumed without loss of generality). Then the MSP with the matrix* M *defined by* $M^T = H'$ *computes the access structure* Γ *if and only if the set of vectors* C *is suitable for* Γ*.*

There is a tight connection between an access structure and its dual. It turns out that the codes generated by the corresponding sets of suitable vectors are orthogonal.

Theorem 4. *([8, Theorem 3.5.4]) Let* $\Gamma^- = \{X_1, \dots, X_r\}$ *be an access structure and* $(\Gamma^\perp)^- = \{Z_1, \dots, Z_t\}$ *be its dual. Then there exists a suitable set* $C = \{\mathbf{c^i} : 1 \leq i \leq r\}$ *for* Γ *if and only if there exists a suitable set* $C^\perp = \{\mathbf{h^j} : 1 \leq j \leq t\}$ *for* $\Gamma^\perp$*.*
Suppose there exists a suitable set C *for* Γ *and a suitable set* $C^\perp$ *for* $\Gamma^\perp$*. Let* $\mathcal{C}$ *be the code defined by the linear span of vectors* $\{(1, \mathbf{c^i}) : 1 \leq i \leq r\}$ *and let* $\mathcal{C}^\perp$ *be the code defined by the linear span of vectors of* $\{(1, \mathbf{h^j}) : 1 \leq j \leq t\}$*. Then the codes* $\mathcal{C}$ *and* $\mathcal{C}^\perp$ *are orthogonal to each another.*

Lemma 1. *Let* $\Gamma^- = \{X_1, \dots, X_r\}$ *be the access structure computed by MSP* $\mathcal{M}$*. Also let* $\boldsymbol{\lambda^i} \in \mathbb{F}^m$ *be the recombination vectors that corresponds to* X_i *see (5) and (6). Then the set of vectors* $C = \{\boldsymbol{\lambda^i} : 1 \leq i \leq r\}$ *defines a suitable set of vectors for the complete access structure* Γ*.*

Theorem 5. *[18] Let* $\mathcal{M}$ *be an MSP program computing* Γ*, and* $\mathcal{M}^\perp$ *be an MSP computing the dual access structure* $\Gamma^\perp$*. Let code* $\mathcal{C}^\perp$ *has the parity check matrix* $H^\perp = (\boldsymbol{\varepsilon} \mid (M^\perp)^T)$ *and code* $\mathcal{C}$ *has the parity check matrix* $H = (\boldsymbol{\varepsilon} \mid M^T)$*. Then for any MSP* $\mathcal{M}$ *there exists an MSP* $\mathcal{M}^\perp$ *such that* $\mathcal{C}$ *and* $\mathcal{C}^\perp$ *are dual.*

McEliece and Sarwate [15] reformulated the Shamir's scheme in terms of Reed-Solomon codes instead of in terms of polynomials, adding in this way error-correcting properties. The general relationship between linear codes and secret sharing schemes was established by Massey [14], Blakley and Kabatianskii [2]. In fact, the coding theoretic approach can be reformulated as the vector space construction, which was introduced by Brickel in [3]. This approach was generalized to the so-called generalized vector space construction by Van Dijk [8]. Two approaches of constructing secret sharing schemes based on linear codes could be distinguished.
The first one uses an $[n, k+1, d_{min}]$ linear code $\overline{\mathcal{C}}$. Let $\overline{G}$ be a generator matrix of $\overline{\mathcal{C}}$, so it is $(k+1) \times n$ matrix. The dealer $\mathcal{D}$ chooses a random information vector $\mathbf{x} \in \mathbb{F}^{k+1}$, subject to $\mathbf{x}_1 = s$ - the secret. Then he calculates the codeword $\mathbf{y}$ corresponding to this information vector as $\mathbf{y} = \mathbf{x}\overline{G}$, $(\mathbf{y} \in \mathbb{F}^n)$. Then $\mathcal{D}$ gives $\mathbf{y}_j$ to player P_j to be his share.
The second approach uses an $[N = n+1, k+1, d_{min}]$ linear code $\widetilde{\mathcal{C}}$. Let $\widetilde{G}$ be a generator matrix of $\widetilde{\mathcal{C}}$, so it is $(k+1) \times (n+1)$. The dealer $\mathcal{D}$ calculates the codeword $\mathbf{y}$ as $\mathbf{y} = \mathbf{x}\widetilde{G}$, $(\mathbf{y} \in \mathbb{F}^N)$, from a random information vector $\mathbf{x} \in \mathbb{F}^{k+1}$, subject to $\mathbf{y}_0 = s$ - the secret. Then $\mathcal{D}$ gives $\mathbf{y}_j$ to player P_j to be his share.
The two kinds of approaches seem different but are related. In the first approach all the shares form a *complete* codeword of the code, while in the second one

all the shares form only part of a codeword. But as Van Dijk [8] proved one can simply transform the matrices of the codes, setting $\widetilde{G} = (\boldsymbol{\varepsilon} \mid \overline{G})$. Hence one can consider the code $\overline{\mathcal{C}}$ to be obtained from the code $\widetilde{\mathcal{C}}$ by *puncturing* i.e. by deleting a coordinate [16].
Now we will generalize these approaches to our error-set codes. We will denote the codes and their generator matrices for the first (and the second) approaches by $\overline{\mathcal{C}}$ and $\overline{G}$ ($\widetilde{\mathcal{C}}$ and $\widetilde{G}$, respectively). Let $\overline{\mathcal{C}}$ be a code of length p, with set of forbidden distances $\Delta(\overline{\mathcal{C}})$ and with $d \times p$ generator matrix $\overline{G}$. Analogously let $\widetilde{\mathcal{C}}$ be a code of length N, with set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$ and with $d \times N$ generator matrix $\widetilde{G}$. Recall that $\widetilde{G} = (\boldsymbol{\varepsilon} \mid \overline{G})$ holds.

Lemma 2. *Let $\mathcal{M} = (\mathbb{F}, M, \boldsymbol{\varepsilon}, \psi)$ be an MSP computing an access structure Γ. Let $\widetilde{\mathcal{C}}$ be an error-set code of length N, with a set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$ and with $d \times N$ generator matrix $\widetilde{G}$ of the form $\widetilde{G} = (\boldsymbol{\varepsilon} \mid M^T)$. Then $\Delta(\widetilde{\mathcal{C}}) = \Delta^{\perp} \uplus \{\mathcal{D}\}$.*

Proof. Let $\mathcal{M}$ be an MSP computing an access structure Γ and $\mathcal{M}^{\perp}$ be its dual MSP. Using $\overline{G} = M^T$ and $\overline{G}^{\perp} = (M^{\perp})^T$ compute the codes $\widetilde{\mathcal{C}}$ and $\widetilde{\mathcal{C}}^{\perp}$. Van Dijk [8] proved that codes $\widetilde{\mathcal{C}}$ and $\widetilde{\mathcal{C}}^{\perp}$ are orthogonal to each other. Moreover Van Dijk showed (see Definition 6 and Theorems 3 and 4) that matrix $\widetilde{G} = (\boldsymbol{\varepsilon} \mid M^T)$ is generated by vectors $(1, \mathbf{h^j})$ where $\mathbf{h^j}$ are suitable vectors for the dual access structure $\Gamma^{\perp}$. It turns out that the codes $\widetilde{\mathcal{C}}$ and $\widetilde{\mathcal{C}}^{\perp}$ are even dual (see Theorem 5). Thus by Lemma 1 and Definition 6 we have that $supp_p(\mathbf{h^j}) \in (\Gamma^{\perp})^-$. In other words the suitable vectors for $\Gamma^{\perp}$ are the *minimal* codewords for the code $\widetilde{\mathcal{C}}$, see definition (4). Hence we have $\Delta(\widetilde{\mathcal{C}}) = \Delta^{\perp} \uplus \{\mathcal{D}\}$ to be the set of forbidden distances for the code generated by $\widetilde{G}$. □

Example 2. In the threshold case $\widetilde{G}$ can be the generator matrix of the extended Reed-Solomon MDS code $[n+1, k+1, n-k+1]$, since $\overline{G}^T$ can be the Vandermonde matrix with rows $(1, \alpha, \alpha^2, \dots, \alpha^k)$. In other words the extended Reed-Solomon code can be used to generate an (k, n) threshold scheme

Note 1. Lemma 2 gives an efficient way to construct error-set codes using MSPs. Note that we do not require any relation between Δ and Δ_A (or for k and k_a).

Now using the results of Theorem 1 and Lemma 2 we obtain the following corollary.

Corollary 1. *An error-set code $\widetilde{\mathcal{C}}$ corrects Δ_A (k_a in the threshold case) errors and one erasure (e.g. $\{\mathcal{D}\}$) if and only if $\Delta_A \uplus \Delta_A \subseteq \Delta^{\perp}$ (analogously $2k_a < n-k$).*

Remark 3. The main difference between error-set codes and SSS is that the SSS provides *privacy*, meaning that $\Delta \supseteqq \Delta_A$ (or $k \geq k_a$).

5 VSS as an Example of a Particular Class of burst "Codes"

A formal definition of VSS is as follows.

Definition 7. *A* Verifiable Secret Sharing *scheme secure against* (Δ, Δ_A)*-adversary* $\mathcal{A}$*, is a pair* (Share-Detect, Reconstruct) *of protocols (phases). At the beginning of the* Share *phase the dealer inputs a secret* $s \in \mathcal{K}$*, at the end of* Share *phase each participant* P_i *is instructed to output a Boolean value* ver_i*. At the end of* Reconstruct *phase each participant is instructed to output a value in* $\mathcal{K}$*. The protocol is unconditionally secure if the following properties hold:*

- Termination (Acceptance of good players): *If a good player* P_i *outputs* $ver_i = 0$ *at the end of Share then every good player outputs* $ver_i = 0$*; Moreover if the dealer* $\mathcal{D}$ *is good, then* $ver_i = 1$ *for every good player* P_i*;*
- Correctness (Verifiability): *If a group of good players* P_i *output* $ver_i = 1$ *at the end of Share, then at this time a value* $s' \in \mathcal{K}$ *has been fixed and and the end of Reconstruct all good players will output the same value* s' *and moreover if the dealer is not corrupt* $s' = s$*.*
- Privacy (Unpredictability): *If* $|\mathcal{K}| = q$*, the secret* s *is chosen randomly from* $\mathcal{K}$*, and the dealer is good, then any forbidden coalition cannot guess at the end of Share the value* s *with probability better than* $1/q$*.*

Note that an SSS with error-correcting capabilities could be considered as an VSS with *honest dealer*, since the *robustness* is guaranteed using the interleaving technique.

Therefore we will first revisit the standard approaches described in the literature used to build SSS from codes employing the interleaving technique.

The *first* approach uses an $[n, k+1, d_{min}]$ linear code $\overline{\mathcal{C}}$. Let $\overline{G}$ be a generator matrix of $\overline{\mathcal{C}}$, so its size is $(k+1) \times n$. Now the dealer $\mathcal{D}$ chooses a random information matrix $X \in \mathbb{F}^{(k+1)\times(k+1)}$, except that s (the secret) is in its upper-left corner. Then $\mathcal{D}$ calculates the (array) codeword Y corresponding to this information matrix $Y = X\overline{G}$, $(Y \in \mathbb{F}^{n\times n})$. Note that the rows in Y are the usual codewords of $\overline{\mathcal{C}}$. Using the interleaving approach the dealer $\mathcal{D}$ gives columns $Y_{(j)}$ to the player P_j as his share. Note that the first coordinate in $Y_{(j)}$ corresponds to the first codeword which encodes the secret.

The *second* approach is very similar. Now $\widetilde{\mathcal{C}}$ is an $[N = n+1, k+1, d_{min}]$ linear code. Let $\widetilde{G}$ be a generator matrix of $\widetilde{\mathcal{C}}$, so it is a $(k+1) \times (n+1)$ matrix. The dealer $\mathcal{D}$ calculates the (array) codeword Y as $Y = X\widetilde{G}$, $(Y \in \mathbb{F}^{N\times N})$, from a random information matrix $X \in \mathbb{F}^{(k+1)\times(k+1)}$, except that s (the secret) is in the upper-left corner of Y. Again applying the interleaving approach the dealer $\mathcal{D}$ gives columns $Y_{(j)}$ to player P_j as his share. Note that the first coordinate in $Y_{(j)}$ corresponds to the first codeword which encodes the secret. The zero column $Y_{(0)}$ is the dealer's share.

It is straightforward to generalize these two approaches to error-set codes. In this case $\overline{\mathcal{C}}$ is a code of length p, with a set of forbidden distances $\Delta(\overline{\mathcal{C}})$ and $\overline{G}$ is a $d \times p$ matrix. Analogously $\widetilde{\mathcal{C}}$ is a code of length N, with a set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$ and $\widetilde{G}$ is a $d \times N$ matrix. Recall that $\widetilde{G} = (\boldsymbol{\varepsilon} \mid \overline{G})$ holds. Then $X \in \mathbb{F}^{d\times d}$ and $Y \in \mathbb{F}^{p\times p}$ for the first approach and $Y \in \mathbb{F}^{N\times N}$ for the second. Note that X could be symmetric or asymmetric.

The sharing procedure we have just described coincides with the sharing procedures of the standard VSS protocols [1, 5]. Note that the shares in these

protocols are distributed in exactly the same way using the interleaving technique.
We will say that the VSS (with honest dealer) is *based on code* $\widetilde{\mathcal{C}}$. Now we will translate the results of Lemma 2 and Corollary 1 into the VSS language.

Proposition 1. *Let $\widetilde{\mathcal{C}}$ be an error-set code of length N, with a set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$. Let consider VSS (with honest dealer) based on this code and $(\Delta, \Delta_A, \Delta_F)$-adversary ($(k, k_a, k_f)$-adversary).*

- Correctness:
  Then VSS (with honest dealer) based on this code satisfy the correctness property in Definition 7 if and only if the code $\widetilde{\mathcal{C}}$ is able to correct burst-error pattern in Δ_A (k_a in threshold case) and burst-erasure pattern in $\Delta_F \uplus \{\mathcal{D}\}$ ($k_f + 1$), i.e. $\Delta_A \uplus \Delta_A \uplus \{\mathcal{D}\} \uplus \Delta_F \subseteq \Delta(\widetilde{\mathcal{C}})$ ($2k_a + k_f + 1 < d_{min}$).
- Privacy:
  Then VSS (with honest dealer) based on this code satisfy the correctness property in Definition 7 if and only if the code $\widetilde{\mathcal{C}}$ has $\Delta(\widetilde{\mathcal{C}})$ as the set of forbidden distances, i.e. $\Delta(\widetilde{\mathcal{C}}) = \Delta^{\perp} \uplus \{\mathcal{D}\}$ ($d_{min} = n - k + 1$).

Proof. The result for a (Δ, Δ_A)-adversary follows directly from Lemma 2 and Corollary 1.
It is straightforward to extend the model considering also *fail-corrupt* players. Recall that to fail-corrupt a player means that the adversary may stop the communication from and to that player at an arbitrary moment during the protocol. From a coding point of view these players are erasures, so the bounds are extended naturally to $\mathcal{P} \notin \Delta_A \uplus \Delta_A \uplus \Delta \uplus \Delta_F$ ($2k_a + k + k_f < n$). □

In coding theory the *Sender* is always assumed to be honest, while in VSS the Dealer could be corrupt. We could simulate the inproper behavior of the dealer in the following way.
Let $\widetilde{\mathcal{C}}$ be a code of length N, with set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$ and $\widetilde{G}$ be a $d \times N$ generator matrix for the code. The sender chooses information matrix $X \in \mathbb{F}^{d \times d}$ (using the first approach). Then he computes the array codeword $Y \in \mathbb{F}^{N \times N}$ by $Y = X\widetilde{G}$. But instead of distributing the columns of Y to the players as their shares, the dealer introduces a burst-error pattern (not necessarily in Δ_A) obtaining matrix Z from Y in this way. Then he distributes Z as shares. Since after receiving their shares the corrupt players could handle wrong ones (i.e. introducing another burst-error pattern in Δ_A) in the reconstruction phase we simulate this behavior as *retransmitting* Z to $\widetilde{Z}$. Since we are able to correct only the error-patterns in Δ_A, we need to apply twice the decoding algorithm (pair-wise checking protocol) in order to correct the errors. But even then we have the problem that the sender could introduce errors not from Δ_A and that the errors he introduced together with the errors that the corrupt players introduced could be not from Δ_A. What the share-detection phase in the VSS protocols (e.g. [5]) achieves more is that the dealer is forced (by the accusation-broadcast mechanism) to defend himself if inconsistent information (not in Δ_A) is found. Thus the honest players have (maybe after being broadcasted by the dealer) consistent shares. This could be simulated by the assumption that Z and

Y differ in an error pattern which is a subset of the error pattern between Z and $\widetilde{Z}$. Therefore the difference between Y and $\widetilde{Z}$ is a error pattern from Δ_A. This immediately gives the following requirements for the code in this *retransmitting scenario*.

Theorem 6. *Let $\widetilde{\mathcal{C}}$ be an error-set code of length N, with a set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$. Let consider VSS based on this code and $(\Delta, \Delta_A, \Delta_F)$-adversary ($(k, k_a, k_f)$-adversary).*

- Correctness:
  Then VSS based on this code satisfy the correctness property in Definition 7 if and only if the code $\widetilde{\mathcal{C}}$ is able to correct burst-error pattern in Δ_A (k_a in threshold case) and burst-erasure pattern in $\Delta_F \uplus \{\mathcal{D}\}$ ($k_f + 1$), i.e. $\Delta_A \uplus \Delta_A \uplus \{\mathcal{D}\} \uplus \Delta_F \subseteq \Delta(\widetilde{\mathcal{C}})$ ($2k_a + k_f + 1 < d_{min}$).
- Privacy:
  Then VSS based on this code satisfy the correctness property in Definition 7 if and only if the code $\widetilde{\mathcal{C}}$ has $\Delta(\widetilde{\mathcal{C}})$ as the set of forbidden distances, i.e. $\Delta(\widetilde{\mathcal{C}}) = \Delta^{\perp} \uplus \{\mathcal{D}\}$ ($d_{min} = n - k + 1$).

The last bounds coincide with the well known bounds in [1, 11, 5, 10], namely, $\Delta_A \uplus \Delta_A \uplus \Delta_F \subseteq \Delta^{\perp}$ or equivalently $\mathcal{P} \notin \Delta_A \uplus \Delta_A \uplus \Delta_F \uplus \Delta$ (in the threshold case the bound becomes $2k_a + k_f + k < n$).

We recall the following notations from [16].

Definition 8. *[16]*

- *Code $\mathcal{C}$ is called* weakly self-dual *if and only if $\mathcal{C} \subsetneqq \mathcal{C}^{\perp}$,*
- *Code $\mathcal{C}$ is called* self-dual *if and only if $\mathcal{C} = \mathcal{C}^{\perp}$.*

Remark 4. How are dual codes and dual access structures linked?

- From Definition 8 for weakly self-dual codes it follows that there exists a non-invertible matrix D such that $DH = G$, where G and H are the generator and parity check matrices of the code $\mathcal{C}$.
- When $\widetilde{\mathcal{C}}$ ($\overline{\mathcal{C}}$) is weakly self-dual code, i.e. $\widetilde{\mathcal{C}} \subsetneqq \widetilde{\mathcal{C}}^{\perp}$, we have $\Gamma(\widetilde{\mathcal{C}}) \subsetneqq \Gamma(\widetilde{\mathcal{C}}^{\perp})$, but from Theorems 3 and 4, it follows that $\Gamma(\widetilde{\mathcal{C}}) = \Gamma^{\perp}$ and $\Gamma(\widetilde{\mathcal{C}}^{\perp}) = \Gamma$. Hence we have $\Gamma^{\perp} \subsetneqq \Gamma$, i.e. Γ is a $\mathcal{Q}^2$ access structure.
- Taking again into account Theorems 3 and 4, i.e. that $\overline{H} = (M^{\perp})^T$, and $\overline{G} = M^T$ we obtain that $D(M^{\perp})^T = M^T$, for some non-invertible matrix D. This implies that $\Gamma^{\perp} \subsetneqq \Gamma$, i.e. Γ is a $\mathcal{Q}^2$ access structure. Note that $\overline{G}\ \overline{H}^T = 0$ implies that $M^T M^{\perp} = \overline{E}$, where $\overline{E}$ is a zero matrix except for the entry in the upper left corner which is 1.

On the other hand for dual-codes we obtain $\Gamma = \Gamma^{\perp}$, i.e. the access structure is self-dual, and $D(M^{\perp})^T = M^T$ for some invertible matrix D ($M^T M^{\perp} = \overline{E}$ holds).

Several interesting open questions arise:

- Whether for any $\mathcal{Q}^2$ access structure Γ there exists a weakly self-dual error-set code with set of allowed distances Γ.

- Whether for any self-dual access structure Γ there exists a self-dual error-set code with set of allowed distances Γ.
- One could define the *weight/distance distribution* of an error-set code, as well as its *weight enumerator* (see [16]). It is interesting to check wether Mac Williams theorem [16] for the weight enumerators of a code and its dual hold in this setting.
- It is well known that for a given access structure Γ (and correspondingly MSP $\mathcal{M}$) the numbers $p_0, p_1, \ldots, p_n$ are the players individual information rate. Appling the powerful invariant theory to the weight enumerator of self-dual error-set codes (access structures) it would be interesting to investigate which numbers are suitable and which are not.

References

1. M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness theorems for Non- Cryptographic Fault-Tolerant Distributed Computation, *Proc. ACM STOC'88*, 1988, 1-10.
2. G. Blakley, G. Kabatianskii. Linear Algebra Aproach to Secret Sharing Schemes, LNCS 829, 1994, pp. 33-40.
3. E. Brickell. Some ideal secret sharing schemes, *J. of Comb. Math. and Comb. Computing* 9, 1989, pp. 105-113.
4. D.Chaum, C.Crepeau and I.Damgard, Multi-Party Unconditionally Secure Protokols, *Proc. ACM STOC'88*, 1988, 11-19.
5. R. Cramer, I. Damgard, U. Maurer. General Secure Multi-Party Computation from any Linear Secret Sharing Scheme, *EUROCRYPT'2000*, LNCS 1807, 2000, pp. 316-334.
6. G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, Covering Codes, *Elsevier Science*, Amsterdam, 1997.
7. B. Chor, S. Goldwasser, S. Micali and B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, *Proc. of the IEEE 26th Annual Symp. on Foundations of Computer Science* 1985, pp. 383-395.
8. M. van Dijk, Secret Key Sharing and Secret Key Generation, *Ph.D. Thesis*, 1997, TU Eindhoven.
9. P. Delsarte. The Hamming space viewed as an association scheme, *Proc. of the 23rd Symposium on Information Theory in the Benelux*, 2002, pp. 329-380.
10. S. Fehr, U. Maurer, Linear VSS and Distributed Commitments Based on Secret Sharing and Pairwise Checks, *Proc. CRYPTO 2002*, LNCS, Springer-Verlag, vol. 2442, pp. 565-580.
11. M. Fitzi, M. Hirt and U. Maurer, Trading Correctness for Privacy in Unconditional Multi-Party Computation, *CRYPTO 98*, LNCS, Springer-Verlag, vol. 1462, 1998, pp. 121-136.
12. M. Hirt, U. Maurer. Player Simulation and General Adversary Structures in Perfect Multiparty Computation, *J. of Cryptology* 13, 2000, pp. 31-60.
13. M. Karchmer, A. Wigderson, On Span Programs, *Proc. of 8-th Annual Structure in Complexity Theory Conference*, San Diego, California, 18-21 May 1993. IEEE Computer Society Press, pp. 102-111.
14. J. L. Massey, Minimal codewords and secret sharing, *Proc. 6-th Joint Swedish-Russian Int. Workshop on Inform. Theory* 1993, 276-279.
15. R. J. McEliece, D. V. Sarwate, On Sharing secrets and Reed-Solomon codes, *Commun. ACM* 24, 1981, pp. 583-584.
16. F. J. Mac Williams, N. J. A. Sloane, The Theory of Error-Correcting Codes, *Elsevier Science*, Amsterdam, 1988.
17. V. Nikov, S. Nikova, B. Preneel, J. Vandewalle, Applying General Access Structure to Proactive Secret Sharing Schemes, *Proc. of the 23rd Symposium on Information Theory in the Benelux*, May 29-31, 2002, Universite Catholique de Lovain (UCL), Lovain-la-Neuve, Belgium, pp. 197-206, *Cryptology ePrint Archive*: Report 2002/141.
18. V. Nikov, S. Nikova, B. Preneel. Upper Bound for the Size of Monotone Span Programs, *ISIT 2003*, 2003, pp. 284.
19. A. Shamir, How to share a secret, *Commun. ACM* 22, 1979, pp. 612-613.