Secret Sharing Schemes and Error Correcting codes

Svetla Nikova

Katholieke Universiteit Leuven, Belgium and
University of Twente, The Netherlands

Secret Sharing Schemes

- **Secret Sharing Schemes** (SSS)
  protect secrecy and integrity of information (secret s).
- It allows a so called *dealer* $\mathcal{D}$ to share the secret among
- Set of entities, usually called *players* $\mathcal{P} = \{P_1, \ldots, P_n\}$.
- Assume that the secret s and all shares are elements of a finite field $\mathbb{F}$.
- Shamir and Blakley independently introduced SSS in 1979.

Secret Sharing Schemes

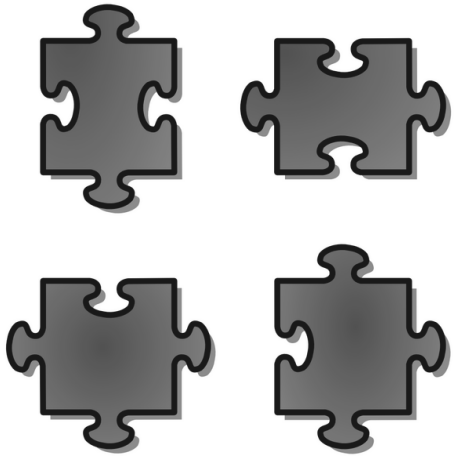
- **Secret Sharing Schemes** (SSS)
  protect secrecy and integrity of information (secret s).
- It allows a so called *dealer* $\mathcal{D}$ to share the secret among
- Set of entities, usually called *players* $\mathcal{P} = \{P_1, \ldots, P_n\}$.
- Assume that the secret s and all shares are elements of a finite field $\mathbb{F}$.
- Shamir and Blakley independently introduced SSS in 1979.

Secret Sharing Schemes

- **Secret Sharing Schemes** (SSS)
  protect secrecy and integrity of information (secret s).
- It allows a so called *dealer* $\mathcal{D}$ to share the secret among
- Set of entities, usually called *players* $\mathcal{P} = \{P_1, \ldots, P_n\}$.
- Assume that the secret s and all shares are elements of a finite field $\mathbb{F}$.
- Shamir and Blakley independently introduced SSS in 1979.

Secret Sharing

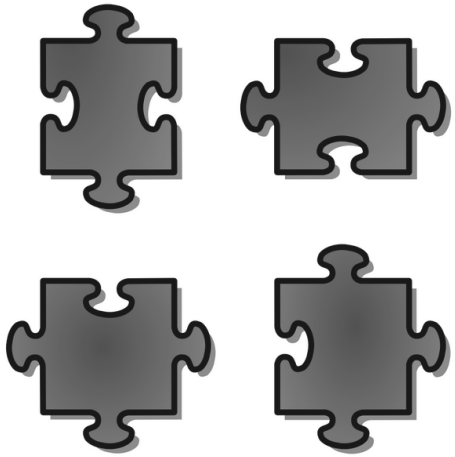
- Secret sharing: participants hold shares of a secret.



- **Forbidden** groups can learn nothing about the secret. Δ - the collection of all forbidden groups - **monotone increasing**.
- **Qualified** groups can recover the secret. Γ - the collection of all qualified groups - **monotone decreasing**.

Secret Sharing

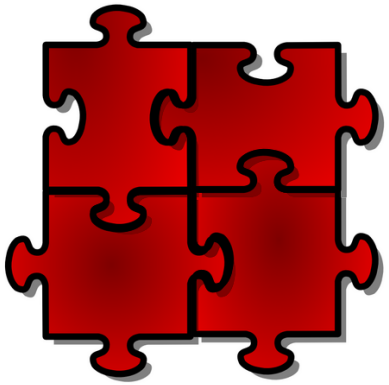
- Secret sharing: participants hold shares of a secret.



- **Forbidden** groups can learn nothing about the secret. Δ - the collection of all forbidden groups - **monotone increasing**.
- **Qualified** groups can recover the secret. Γ - the collection of all qualified groups - **monotone decreasing**.

Secret Sharing

- Secret sharing: participants hold shares of a secret.



- **Forbidden** groups can learn nothing about the secret. Δ - the collection of all forbidden groups - **monotone increasing**.
- **Qualified** groups can recover the secret. Γ - the collection of all qualified groups - **monotone decreasing**.

Secret Sharing Schemes

- **Threshold** SSS - notation (k, n):
  - **Privacy:** any subset of players of size at most $k-1$ should get no information about the secret.
  - **Reconstruction:** any subset of players of size at least k is allowed to reconstruct the secret.
- **Ramp** SSS - notation (k, t, n):
  - **Privacy:** any subset of players of size at most $k-1$ should get no information about the secret.
  - **Reconstruction:** any subset of players of size at least t is allowed to reconstruct the secret.
  - The subsets of players of size between k and $t-1$ might get some information about the secret.

Secret Sharing Schemes

- **Threshold** SSS - notation (k, n):
  - **Privacy:** any subset of players of size at most $k-1$ should get no information about the secret.
  - **Reconstruction:** any subset of players of size at least k is allowed to reconstruct the secret.
- **Ramp** SSS - notation (k, t, n):
  - **Privacy:** any subset of players of size at most $k-1$ should get no information about the secret.
  - **Reconstruction:** any subset of players of size at least t is allowed to reconstruct the secret.
  - The subsets of players of size between k and $t-1$ might get some information about the secret.

Shamir's SSS:

- **Share**
  - The dealer D associates with any player P_i a number α_i and broadcasts this information;
  - The dealer D chooses a private, random polynomial $f(x)$ of degree $k-1$ subject to $f(0) = s$;
  - The dealer D computes $s_i = f(\alpha_i)$ and sends it privately to the player P_i.
- **Reconstruct**
  - At least k players pooling their shares s_i jointly reconstruct $f(x)$;
  - Compute $f(0)$.

Shamir's SSS:

- **Share**
  - The dealer D associates with any player P_i a number α_i and broadcasts this information;
  - The dealer D chooses a private, random polynomial $f(x)$ of degree $k-1$ subject to $f(0) = s$;
  - The dealer D computes $s_i = f(\alpha_i)$ and sends it privately to the player P_i.
- **Reconstruct**
  - At least k players pooling their shares s_i jointly reconstruct $f(x)$;
  - Compute $f(0)$.

Access Structures

- (Γ, Δ) is called an **access structure** if $\Gamma \cap \Delta = \varnothing$.
- If $\Gamma \cup \Delta = P(\mathcal{P})$ then it is said that the access structure is **complete** and is denoted by Γ. The SSS is called **perfect**.
- The SSS is called **ideal** if every player has only one share.
- The tuple $(\Gamma^{\perp}, \Delta^{\perp})$ is defined on $\mathcal{P}$ as follows

$$\Gamma^{\perp} = \{A : \mathcal{P} \setminus A \in \Delta\} \quad \text{and} \quad \Delta^{\perp} = \{A : \mathcal{P} \setminus A \in \Gamma\}.$$

  $(\Gamma^{\perp}, \Delta^{\perp})$ is called the **dual access structure** of (Γ, Δ).
- Note that $(n-k-1, n)$ SSS is dual to (k, n) SSS.

Access Structures

- (Γ, Δ) is called an **access structure** if $\Gamma \cap \Delta = \varnothing$.
- If $\Gamma \cup \Delta = P(\mathcal{P})$ then it is said that the access structure is **complete** and is denoted by Γ. The SSS is called **perfect**.
- The SSS is called **ideal** if every player has only one share.
- The tuple $(\Gamma^{\perp}, \Delta^{\perp})$ is defined on $\mathcal{P}$ as follows

$$\Gamma^{\perp} = \{A : \mathcal{P} \setminus A \in \Delta\} \quad \text{and} \quad \Delta^{\perp} = \{A : \mathcal{P} \setminus A \in \Gamma\}.$$

  $(\Gamma^{\perp}, \Delta^{\perp})$ is called the **dual access structure** of (Γ, Δ).
- Note that $(n-k-1, n)$ SSS is dual to (k, n) SSS.

Monotone Span Programs

Definition (KW93)

A **Monotone Span Program** (MSP) $\mathcal{M}$ is a quadruple $(\mathbb{F}, M, \varepsilon, \psi)$, 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 labeling function and $\varepsilon = (1, 0, \ldots, 0)^T \in \mathbb{F}^d$ is called **target vector**.

- An MSP is said **to compute** a (complete) access structure Γ when $\varepsilon \in \mathrm{im}(M_A^T) \iff A$ is a member of Γ.
- A is **accepted** by $\mathcal{M} \iff A \in \Gamma$, otherwise we say A is **rejected** by $\mathcal{M}$.

Monotone Span Programs

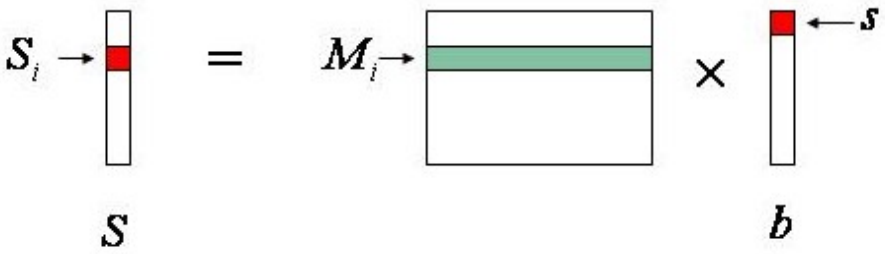
Definition (KW93)

A **Monotone Span Program** (MSP) $\mathcal{M}$ is a quadruple $(\mathbb{F}, M, \varepsilon, \psi)$, 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 labeling function and $\varepsilon = (1, 0, \ldots, 0)^T \in \mathbb{F}^d$ is called **target vector**.

- An MSP is said **to compute** a (complete) access structure Γ when $\varepsilon \in \mathrm{im}(M_A^T) \iff A$ is a member of Γ.
- A is **accepted** by $\mathcal{M} \iff A \in \Gamma$, otherwise we say A is **rejected** by $\mathcal{M}$.

Linear algebraic view on Shamir's SSS

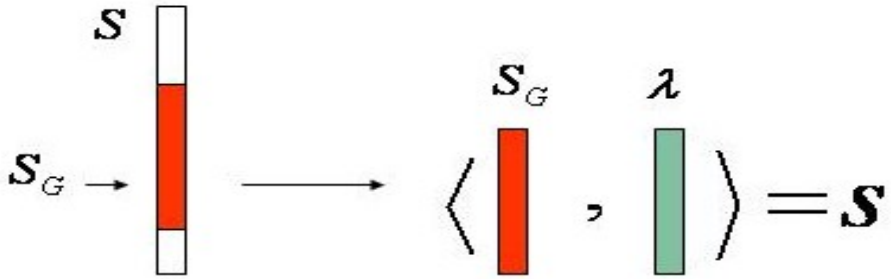
- **Share**
  - The dealer D associates with any player P_i a number α_i and constructs an $n \times k$ Vandermonde matrix, M, which is made public.
  - The dealer D chooses a private, random vector b of length k, setting its first coordinate b_1 to s.
  - The dealer D computes $S_i = M_i b$ and sends it privately to the player P_i.



Linear algebraic view on Shamir's SSS

- **Reconstruct**
  - From a collection G of at least k shares S_i, the corresponding players compute λ such that $M_G^T \lambda = \varepsilon$.
  - Let $Mb = S$ (hence $M_G b = S_G$), then

$$s = \langle b, \varepsilon \rangle = \langle b, M_G^T \lambda \rangle = \langle M_G b, \lambda \rangle = \langle S_G, \lambda \rangle$$



Example of Secret Sharing Scheme

$$MSP * (secret, \mathbf{random}) = \mathbf{shares}$$

$$\Gamma^- = \{14, 34, 24, 23\}, \qquad \Delta^+ = \{13, 12, 4\}.$$

$$\begin{pmatrix} 0 & 0 & 1 \\ \hline 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} s \\ r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} r_2 \\ \hline s + r_1 \\ \hline r_1 + r_2 \\ r_2 \\ \hline s + r_2 \\ r_1 \end{pmatrix}$$

Example of Secret Sharing Scheme

$$MSP * (secret, \mathbf{random}) = \mathbf{shares}$$

$$\Gamma^- = \{14, 34, 24, 23\}, \qquad \Delta^+ = \{13, 12, 4\}.$$

$$\begin{pmatrix} 0 & 0 & 1 \\ \hline 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} s \\ r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} r_2 \\ \hline s + r_1 \\ \hline r_1 + r_2 \\ r_2 \\ \hline s + r_2 \\ r_1 \end{pmatrix}$$

Example of Secret Sharing Scheme

$$MSP * (secret, \mathbf{random}) = \mathbf{shares}$$

$$\Gamma^- = \{14, 34, 24, 23\}, \qquad \Delta^+ = \{13, 12, 4\}.$$

$$\begin{pmatrix} 0 & 0 & 1 \\ \hline 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} s \\ r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} r_2 \\ \hline s + r_1 \\ \hline r_1 + r_2 \\ r_2 \\ \hline s + r_2 \\ r_1 \end{pmatrix}$$

Example of Secret Sharing Scheme

$$MSP * (secret, \mathbf{random}) = \mathbf{shares}$$

$$\Gamma^- = \{14, 34, 24, 23\}, \qquad \Delta^+ = \{13, 12, 4\}.$$

$$\begin{pmatrix} 0 & 0 & 1 \\ \hline 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} s \\ r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} r_2 \\ \hline s + r_1 \\ \hline r_1 + r_2 \\ r_2 \\ \hline s + r_2 \\ r_1 \end{pmatrix}$$

Error Correcting Codes

- Any non-empty subset $\mathcal{C}$ of $\mathbb{F}^n$ is called a code, n - the *length* of the code. Each vector in $\mathcal{C}$ is called *codeword* of $\mathcal{C}$;
- *Minimum distance* of a code $\mathcal{C}$:
  $d_{min} = \min_{\mathbf{a},\mathbf{b}\in\mathcal{C},\ \mathbf{a}\neq\mathbf{b}} d(\mathbf{a},\mathbf{b})$;
- Code $\mathcal{C}$ with min distance d_{min} can correct $e \leq \lfloor (d_{min}-1)/2 \rfloor$ errors;
- Code $\mathcal{C}$ can correct b errors and c erasures as long as $2b + c < d_{min}$;
- Two methods to determine a linear code $\mathcal{C}$:
  a *generator matrix* G and a *parity check matrix* H.

Error Correcting Codes

- Any non-empty subset $\mathcal{C}$ of $\mathbb{F}^n$ is called a code, n - the *length* of the code. Each vector in $\mathcal{C}$ is called *codeword* of $\mathcal{C}$;
- *Minimum distance* of a code $\mathcal{C}$: $d_{min} = \min_{\mathbf{a},\mathbf{b}\in\mathcal{C},\ \mathbf{a}\neq\mathbf{b}} d(\mathbf{a},\mathbf{b})$;
- Code $\mathcal{C}$ with min distance d_{min} can correct $e \leq \lfloor (d_{min}-1)/2 \rfloor$ errors;
- Code $\mathcal{C}$ can correct b errors and c erasures as long as $2b + c < d_{min}$;
- Two methods to determine a linear code $\mathcal{C}$: a *generator matrix* G and a *parity check matrix* H.

Error Correcting Codes

- Any non-empty subset $\mathcal{C}$ of $\mathbb{F}^n$ is called a code, n - the *length* of the code. Each vector in $\mathcal{C}$ is called *codeword* of $\mathcal{C}$;
- *Minimum distance* of a code $\mathcal{C}$: $d_{min} = \min_{\mathbf{a},\mathbf{b}\in\mathcal{C},\ \mathbf{a}\neq\mathbf{b}} d(\mathbf{a},\mathbf{b})$;
- Code $\mathcal{C}$ with min distance d_{min} can correct $e \leq \lfloor (d_{min}-1)/2 \rfloor$ errors;
- Code $\mathcal{C}$ can correct b errors and c erasures as long as $2b + c < d_{min}$;
- Two methods to determine a linear code $\mathcal{C}$: a *generator matrix* G and a *parity check matrix* H.

MDS codes

- Let n denote code length, k dimension and d minimum distance. Notation $[n, k, d]$ code $\mathcal{C}$.
- Dual of an $[n, k, d]$ code $\mathcal{C}$ is an $[n, n-k, d^{\perp}]$ code $\mathcal{C}^{\perp}$.
- Singleton bound for an $[n, k, d]$ code: $d \leq n - k + 1$.
- Codes that satisfy the bound with equality are MDS codes, i.e. $[n, k, n+1-k]$ codes.
- Singleton bound (equivalent form) $d + d^{\perp} \leq n + 2$, equality if and only if MDS code.
- The dual of an $[n, k, d]$ MDS code is an $[n, n-k, k+1]$ MDS code.
- Any k columns of a generator matrix of MDS code $\mathcal{C}$ are linearly independent.

MDS codes

- Let n denote code length, k dimension and d minimum distance. Notation $[n, k, d]$ code $\mathcal{C}$.
- Dual of an $[n, k, d]$ code $\mathcal{C}$ is an $[n, n-k, d^{\perp}]$ code $\mathcal{C}^{\perp}$.
- Singleton bound for an $[n, k, d]$ code: $d \leq n - k + 1$.
- Codes that satisfy the bound with equality are MDS codes, i.e. $[n, k, n+1-k]$ codes.
- Singleton bound (equivalent form) $d + d^{\perp} \leq n + 2$, equality if and only if MDS code.
- The dual of an $[n, k, d]$ MDS code is an $[n, n-k, k+1]$ MDS code.
- Any k columns of a generator matrix of MDS code $\mathcal{C}$ are linearly independent.

MDS codes

- Let n denote code length, k dimension and d minimum distance. Notation $[n, k, d]$ code $\mathcal{C}$.
- Dual of an $[n, k, d]$ code $\mathcal{C}$ is an $[n, n-k, d^{\perp}]$ code $\mathcal{C}^{\perp}$.
- Singleton bound for an $[n, k, d]$ code: $d \leq n - k + 1$.
- Codes that satisfy the bound with equality are MDS codes, i.e. $[n, k, n+1-k]$ codes.
- Singleton bound (equivalent form) $d + d^{\perp} \leq n + 2$, equality if and only if MDS code.
- The dual of an $[n, k, d]$ MDS code is an $[n, n-k, k+1]$ MDS code.
- Any k columns of a generator matrix of MDS code $\mathcal{C}$ are linearly independent.

MDS codes and Shamir's Secret Sharing

- Linear Secret Sharing Schemes - use linear operations
- Example: Reed-Solomon codes. A message $(a_0, a_1, \ldots, a_{k-1})$ defines a polynomial $f(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}$. The codeword is $(f(1), f(2), \ldots, f(n))$. An $[n, k]$ Reed-Solomon code can correct up to $n - k$ erasures.
- Shamir's scheme is a Reed-Solomon code: a secret $f(0)$ is encoded as a codeword $(f(0), f(1), f(2), ..., f(n))$. The missing shares correspond to erasures in the code.
- Thus an $[n + 1, k]$ Reed-Solomon code defines a $(k - 1, n)$ threshold scheme.
- In fact, every (k, n) linear threshold secret sharing scheme is equivalent to some $[n + 1, k + 1]$ MDS code.

MDS codes and Shamir's Secret Sharing

- Linear Secret Sharing Schemes - use linear operations
- Example: Reed-Solomon codes. A message $(a_0, a_1, \ldots, a_{k-1})$ defines a polynomial $f(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}$. The codeword is $(f(1), f(2), \ldots, f(n))$. An $[n, k]$ Reed-Solomon code can correct up to $n - k$ erasures.
- Shamir's scheme is a Reed-Solomon code: a secret $f(0)$ is encoded as a codeword $(f(0), f(1), f(2), ..., f(n))$. The missing shares correspond to erasures in the code.
- Thus an $[n + 1, k]$ Reed-Solomon code defines a $(k - 1, n)$ threshold scheme.
- In fact, every (k, n) linear threshold secret sharing scheme is equivalent to some $[n + 1, k + 1]$ MDS code.

MDS codes and Shamir's Secret Sharing

- Linear Secret Sharing Schemes - use linear operations
- Example: Reed-Solomon codes. A message $(a_0, a_1, \ldots, a_{k-1})$ defines a polynomial $f(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}$. The codeword is $(f(1), f(2), \ldots, f(n))$. An $[n, k]$ Reed-Solomon code can correct up to $n - k$ erasures.
- Shamir's scheme is a Reed-Solomon code: a secret $f(0)$ is encoded as a codeword $(f(0), f(1), f(2), ..., f(n))$. The missing shares correspond to erasures in the code.
- Thus an $[n + 1, k]$ Reed-Solomon code defines a $(k - 1, n)$ threshold scheme.
- In fact, every (k, n) linear threshold secret sharing scheme is equivalent to some $[n + 1, k + 1]$ MDS code.

Two approaches for constructing SSS from codes

The first approach uses an $[n, k+1, d]$ linear code $\overline{\mathcal{C}}$ with generator matrix $\overline{G}$ ($\mathbb{F}^{(k+1)\times n}$). 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 $\mathbf{y} = \mathbf{x}\overline{G}$, ($\mathbf{y} \in \mathbb{F}^n$). $\mathcal{D}$ gives $\mathbf{y}_j$ to player P_j to be his share.

Theorem (Brickell 89)

Let $\overline{G}$ be a generator matrix of an $[n, k+1, d]$ linear code. In a secret sharing scheme based on $\overline{G}$ as described above a set of shares belonging to players $A \subset \mathcal{P}$ determines the secret s if and only if the vector ε is a linear combination of the columns in the generator matrix $\overline{G}$ with indices in A.
Furthermore, the secret-sharing is perfect.

Two approaches for constructing SSS from codes

The first approach uses an $[n, k+1, d]$ linear code $\overline{\mathcal{C}}$ with generator matrix $\overline{G}$ ($\mathbb{F}^{(k+1)\times n}$). 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 $\mathbf{y} = \mathbf{x}\overline{G}$, ($\mathbf{y} \in \mathbb{F}^n$). $\mathcal{D}$ gives $\mathbf{y}_j$ to player P_j to be his share.

Theorem (Brickell 89)

Let $\overline{G}$ be a generator matrix of an $[n, k+1, d]$ linear code. In a secret sharing scheme based on $\overline{G}$ as described above a set of shares belonging to players $A \subset \mathcal{P}$ determines the secret s if and only if the vector ε is a linear combination of the columns in the generator matrix $\overline{G}$ with indices in A.
Furthermore, the secret-sharing is perfect.

Two approaches for constructing SSS from codes

The second approach uses an $[n+1, k+1, d]$ linear code $\widetilde{\mathcal{C}}$ with generator matrix $\widetilde{G}$ $(\mathbb{F}^{(k+1)\times(n+1)})$. 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.

Theorem (Massey 93)

Let $\widetilde{G}$ be a generator matrix of an $[n+1, k+1, d]$ linear code. In a secret sharing scheme based on $\widetilde{G}$ with respect to the second approach a set of shares belonging to players $A \subset \mathcal{P}$ determines the secret s if and only if the first column in $\widetilde{G}$ is a linear combination of the columns with indices in A.
Furthermore, the secret-sharing is perfect.

Two approaches for constructing SSS from codes

The second approach uses an $[n+1, k+1, d]$ linear code $\widetilde{\mathcal{C}}$ with generator matrix $\widetilde{G}$ $(\mathbb{F}^{(k+1)\times(n+1)})$. 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.

Theorem (Massey 93)

Let $\widetilde{G}$ be a generator matrix of an $[n+1, k+1, d]$ linear code. In a secret sharing scheme based on $\widetilde{G}$ with respect to the second approach a set of shares belonging to players $A \subset \mathcal{P}$ determines the secret s if and only if the first column in $\widetilde{G}$ is a linear combination of the columns with indices in A.
Furthermore, the secret-sharing is perfect.

MDS codes provide cheating detection to SSS

Theorem (McEliece and Sarwate 81)

Consider an $[n+1, k+1, d]$ MDS code $\mathcal{C}$ and select at random any codewords $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n)$ with $\mathbf{c}_0 = s$. The dealer gives c_i as a share to participant P_i, $1 \leq i \leq n$.

If $k+1+2k_a$ or more participants pool together their shares, and at most k_a of these values are incorrect, then the secret s can be recovered correctly and the lying participants can be identified.

If $k+2k_a$ or less participants pool together their shares, and precisely k_a of these values are incorrect, then the secret s can not be recovered correctly. In fact, each value of s is equally likely.

MDS codes provide cheating detection to SSS

Theorem (McEliece and Sarwate 81)

Consider an $[n+1, k+1, d]$ MDS code $\mathcal{C}$ and select at random any codewords $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n)$ with $\mathbf{c}_0 = s$. The dealer gives c_i as a share to participant P_i, $1 \leq i \leq n$.
If $k + 1 + 2k_a$ or more participants pool together their shares, and at most k_a of these values are incorrect, then the secret s can be recovered correctly and the lying participants can be identified.
If $k + 2k_a$ or less participants pool together their shares, and precisely k_a of these values are incorrect, then the secret s can not be recovered correctly. In fact, each value of s is equally likely.

MDS codes provide cheating detection to SSS

Theorem (McEliece and Sarwate 81)

Consider an $[n+1, k+1, d]$ MDS code $\mathcal{C}$ and select at random any codewords $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n)$ with $\mathbf{c}_0 = s$. The dealer gives c_i as a share to participant P_i, $1 \leq i \leq n$.
If $k+1+2k_a$ or more participants pool together their shares, and at most k_a of these values are incorrect, then the secret s can be recovered correctly and the lying participants can be identified.
If $k+2k_a$ or less participants pool together their shares, and precisely k_a of these values are incorrect, then the secret s can not be recovered correctly. In fact, each value of s is equally likely.

A class of Error-Correcting Codes

For any vector $\mathbf{x} = (\mathbf{x^0}, \mathbf{x^1}, \ldots, \mathbf{x^n})$ the set $\mathcal{P}$ defines a partition.

Define $\mathcal{P}$-support of vector $\mathbf{x}$: $\sup_{\mathcal{P}}(\mathbf{v}) = \{i : \mathbf{v^i} \neq \mathbf{0}\}$.

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

For two vectors define the set $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \{i : \mathbf{x^i} \neq \mathbf{y^i}\}$.

1. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{x}) = \varnothing$,
2. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \delta_{\mathcal{P}}(\mathbf{y}, \mathbf{x})$ (symmetry)
3. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{z}) \subseteq \delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) \cup \delta_{\mathcal{P}}(\mathbf{y}, \mathbf{z})$,

Obviously $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \sup_{\mathcal{P}}(\mathbf{x} - \mathbf{z})$

A class of Error-Correcting Codes

For any vector $\mathbf{x} = (\mathbf{x^0}, \mathbf{x^1}, \ldots, \mathbf{x^n})$ the set $\mathcal{P}$ defines a partition.
Define $\mathcal{P}$-support of vector $\mathbf{x}$: $\sup_{\mathcal{P}}(\mathbf{v}) = \{i : \ \mathbf{v^i} \neq \mathbf{0}\}$.

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

For two vectors define the set $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \{i : \mathbf{x^i} \neq \mathbf{y^i}\}$.

1. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{x}) = \varnothing$,
2. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \delta_{\mathcal{P}}(\mathbf{y}, \mathbf{x})$ (symmetry)
3. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{z}) \subseteq \delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) \cup \delta_{\mathcal{P}}(\mathbf{y}, \mathbf{z})$,

Obviously $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \sup_{\mathcal{P}}(\mathbf{x} - \mathbf{z})$

A class of Error-Correcting Codes

For any vector $\mathbf{x} = (\mathbf{x^0}, \mathbf{x^1}, \ldots, \mathbf{x^n})$ the set $\mathcal{P}$ defines a partition.
Define $\mathcal{P}$-support of vector $\mathbf{x}$: $\sup_{\mathcal{P}}(\mathbf{v}) = \{i : \ \mathbf{v^i} \neq \mathbf{0}\}$.

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

For two vectors define the set $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \{i : \mathbf{x^i} \neq \mathbf{y^i}\}$.

1. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{x}) = \varnothing$,
2. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \delta_{\mathcal{P}}(\mathbf{y}, \mathbf{x})$ (symmetry)
3. $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{z}) \subseteq \delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) \cup \delta_{\mathcal{P}}(\mathbf{y}, \mathbf{z})$,

Obviously $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \sup_{\mathcal{P}}(\mathbf{x} - \mathbf{z})$

A class of Error-Correcting Codes

The Idea:

- To work in a new metric: **Numbers** $\rightarrow$ **Sets**
- Replace monotone properties defined by numbers into sets.

For any $\mathbf{x}, \mathbf{y} \in \mathbb{F}^N$

$$d(\mathbf{x}, \mathbf{y}) = |\{i : \mathbf{x}_i \neq \mathbf{y}_i\}| \quad \rightarrow \quad \delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) = \{i : \mathbf{x}^{\mathbf{i}} \neq \mathbf{y}^{\mathbf{i}}\}$$
$$wt(\mathbf{x}) = |\{i : \mathbf{x}_i \neq 0\}| \quad \rightarrow \quad sup_{\mathcal{P}}(\mathbf{x}) = \{i : \mathbf{x}^{\mathbf{i}} \neq 0\}$$

We could use $\delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y})$ instead of the Hamming distance and explore the properties of the so defined space.

A class of Error-Correcting Codes

Define Δ-*neighborhood* of pseudo-radii in Δ centered around $\mathbf{x} \in \mathbb{F}^N$: $B_\Delta(\mathbf{x}) = \{\mathbf{y} \in \mathbb{F}^N : \delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) \in \Delta\}$.

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?

For a code $\mathcal{C}$ we define the set of

- possible (allowed) distances: $\Gamma(\mathcal{C})$
  $= \{A : \text{ there exist } \mathbf{a}, \mathbf{b} \text{ in } \mathcal{C},\ \mathbf{a} \neq \mathbf{b} \text{ such that } \delta_{\mathcal{P}}(\mathbf{a}, \mathbf{b}) \subseteq A\}$
- forbidden distances: $\Delta(\mathcal{C}) = \Gamma(\mathcal{C})^c$.

We will call the so-defined codes error-set correcting codes.

A class of Error-Correcting Codes

Define Δ-*neighborhood* of pseudo-radii in Δ centered around $\mathbf{x} \in \mathbb{F}^N$: $B_\Delta(\mathbf{x}) = \{\mathbf{y} \in \mathbb{F}^N : \delta_{\mathcal{P}}(\mathbf{x}, \mathbf{y}) \in \Delta\}$.

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?

For a code $\mathcal{C}$ we define the set of

- possible (allowed) distances: $\Gamma(\mathcal{C})$
  $= \{A : \text{ there exist } \mathbf{a}, \mathbf{b} \text{ in } \mathcal{C},\ \mathbf{a} \neq \mathbf{b} \text{ such that } \delta_{\mathcal{P}}(\mathbf{a}, \mathbf{b}) \subseteq A\}$
- forbidden distances: $\Delta(\mathcal{C}) = \Gamma(\mathcal{C})^c$.

We will call the so-defined codes error-set correcting codes.

A class of Error-Correcting Codes

Theorem (NN03)

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

Example

Consider the special case with threshold access structure:

$$\Delta = \{A : |A| \leq e\}.$$

$B_\Delta(\mathbf{x}) = B_e(\mathbf{x})$ - the usual Hamming sphere. Now
$\Delta \uplus \Delta = \{A : |A| \leq 2e\} = \Delta(\mathcal{C})$ and $\Gamma(\mathcal{C}) = \{A : |A| \geq 2e+1\}$.
Hence the minimum distance of $\mathcal{C}$ is $d_{min} = 2e+1$.

SSS as an Example of a Particular Class of Codes

Corollary (NN03)

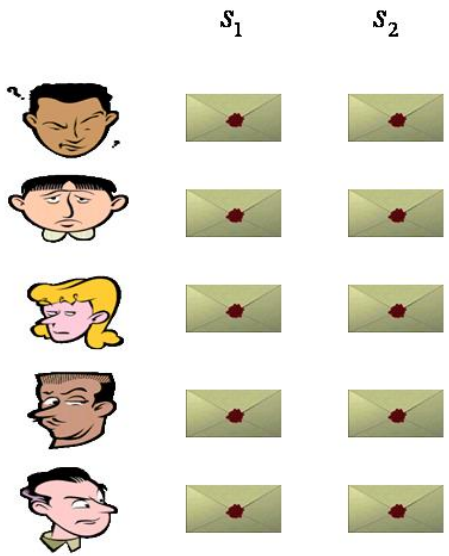
Let $\mathcal{M}$ be an MSP program computing Γ, and $\mathcal{M}^{\perp}$ be an MSP computing the dual access structure $\Gamma^{\perp}$. Let code $\mathcal{C}^{\perp}$ have the parity check matrix $H^{\perp} = (\varepsilon \mid (M^{\perp})^T)$ and let code $\mathcal{C}$ have the parity check matrix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

Theorem (NN03)

Let $\mathcal{M} = (\mathbb{F}, M, \varepsilon, \psi)$ be an MSP computing an access structure Γ. Let $\widetilde{\mathcal{C}}$ be an error-set correcting code, with a set of forbidden distances $\Delta(\widetilde{\mathcal{C}})$ and with a generator matrix $\widetilde{G}$ of the form $\widetilde{G} = (\varepsilon \mid M^T)$. Then the $\mathcal{P}$-minimal codewords for $\widetilde{\mathcal{C}}$ are the vectors of the form $(1, c)$ and $sup_{\mathcal{P}}(c) \in \Gamma^{\perp}$.

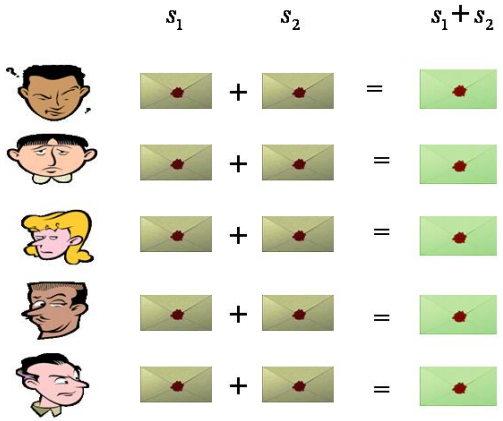
Operations on the Secrets

Multi-Party Computation protocols [Yao 82], enable a set of players to securely evaluate an arbitrary function on their private inputs.



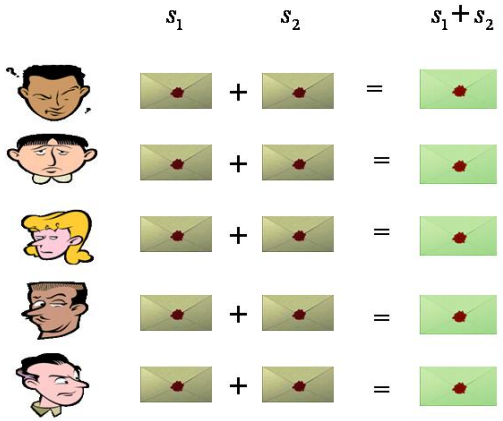
"Securely" means that the computation must guarantee the correctness of the result while retaining the privacy of the players' inputs.

Addition of Secrets



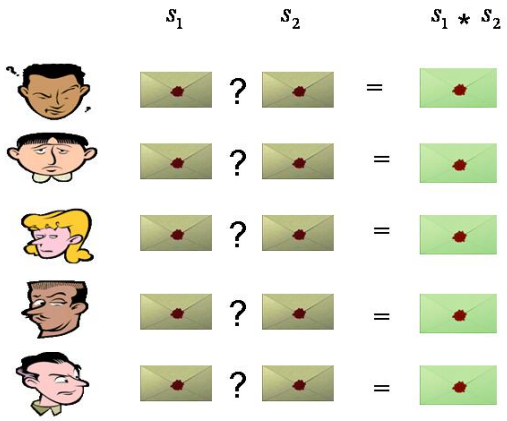
- Each participant in a group $\mathcal{P}$ holds shares of the secrets.
- They wish to compute shares of their sum **using only the local shares**: $\exists \vec{\lambda} \qquad s + s' = \sum_{i \in \mathcal{P}} \lambda_i (s_i + s'_i)$.
- Follows from the properties of the Linear SSS.

Addition of Secrets



- Each participant in a group $\mathcal{P}$ holds shares of the secrets.
- They wish to compute shares of their sum **using only the local shares**: $\exists \vec{\lambda} \qquad s + s' = \sum_{i \in \mathcal{P}} \lambda_i (s_i + s'_i)$.
- Follows from the properties of the Linear SSS.

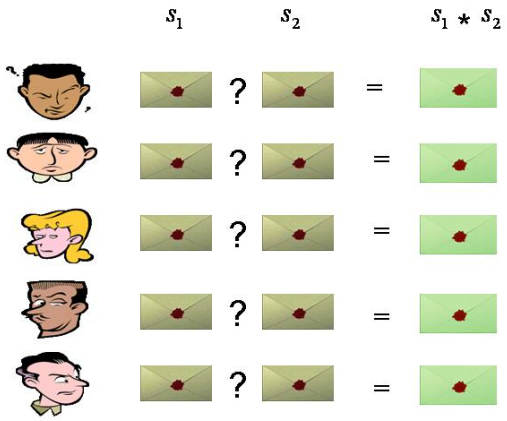
Multiplication of Secrets



- Each participant in a group $\mathcal{P}$ holds shares of the secrets.
- They wish to compute shares of their product **using only the local shares**: $\exists \vec{\lambda} \qquad s \cdot s' = \sum_{i \in \mathcal{P}} \lambda_i s_i s'_i$.
- Shamir's scheme is multiplicative for $k < \frac{n}{2}$

$$P(\alpha_i) \cdot Q(\alpha_i) = (P \cdot Q)(\alpha_i), \quad \deg(P \cdot Q) = 2t < n.$$

Multiplication of Secrets



- Each participant in a group $\mathcal{P}$ holds shares of the secrets.
- They wish to compute shares of their product **using only the local shares**: $\exists \vec{\lambda} \qquad s \cdot s' = \sum_{i \in \mathcal{P}} \lambda_i s_i s'_i$.
- Shamir's scheme is multiplicative for $k < \frac{n}{2}$

$$P(\alpha_i) \cdot Q(\alpha_i) = (P \cdot Q)(\alpha_i), \quad \deg(P \cdot Q) = 2t < n.$$

Revisiting Massey construction

- Recall: An $[n+1, k+1, d]$ linear code $\mathcal{C}$ with generator matrix G leads to an SSS from a codeword $\mathbf{y}$ subject to $\mathbf{y}_0 = s$ - the secret, and $\mathbf{y}_j$ is player P_j share $(1 \leq j \leq n)$. The constructed secret sharing is perfect.
- But the access structure computed by the constructed SSS depends on the choice of G.
- Meaning for an $[n+1, k+1, d]$ code take two generator matrices the obtained SSSs will compute different access structures!

Revisiting Massey construction

- Recall: An $[n+1, k+1, d]$ linear code $\mathcal{C}$ with generator matrix G leads to an SSS from a codeword $\mathbf{y}$ subject to $\mathbf{y}_0 = s$ - the secret, and $\mathbf{y}_j$ is player P_j share $(1 \leq j \leq n)$. The constructed secret sharing is perfect.
- But the access structure computed by the constructed SSS depends on the choice of G.
- Meaning for an $[n+1, k+1, d]$ code take two generator matrices the obtained SSSs will compute different access structures!

Revisiting Massey construction

- Nevertheless something still can be claimed:
  - **Privacy:** $d^{\perp} - 2$ participants learn nothing about the secret.
  - **Reconstruction:** $n - d + 2$ can recover the secret.
- Remember the Singleton bound implies $d^{\perp} - 2 \leq n - d + 2$ with equality if and only if MDS code.
- *Question:* can t participants recover the secret, if $d^{\perp} - 2 < t < n - d + 2$.
- *Answer:* sometimes.
- Remember the access structure is complete, but we can consider it as a ramp scheme.

Revisiting Massey construction

- Nevertheless something still can be claimed:
  - **Privacy:** $d^{\perp} - 2$ participants learn nothing about the secret.
  - **Reconstruction:** $n - d + 2$ can recover the secret.
- Remember the Singleton bound implies $d^{\perp} - 2 \leq n - d + 2$ with equality if and only if MDS code.
- *Question:* can t participants recover the secret, if $d^{\perp} - 2 < t < n - d + 2$.
- *Answer:* sometimes.
- Remember the access structure is complete, but we can consider it as a ramp scheme.

Revisiting Massey construction

- Nevertheless something still can be claimed:
  - **Privacy:** $d^{\perp} - 2$ participants learn nothing about the secret.
  - **Reconstruction:** $n - d + 2$ can recover the secret.
- Remember the Singleton bound implies $d^{\perp} - 2 \leq n - d + 2$ with equality if and only if MDS code.
- *Question:* can t participants recover the secret, if $d^{\perp} - 2 < t < n - d + 2$.
- *Answer:* sometimes.
- Remember the access structure is complete, but we can consider it as a ramp scheme.

Revisiting Massey construction

- Nevertheless something still can be claimed:
  - **Privacy:** $d^{\perp} - 2$ participants learn nothing about the secret.
  - **Reconstruction:** $n - d + 2$ can recover the secret.
- Remember the Singleton bound implies $d^{\perp} - 2 \leq n - d + 2$ with equality if and only if MDS code.
- *Question:* can t participants recover the secret, if $d^{\perp} - 2 < t < n - d + 2$.
- *Answer:* sometimes.
- Remember the access structure is complete, but we can consider it as a ramp scheme.

Revisiting Error Correction Codes

Definition (Pellikaan 92)

Let U, V and C be linear codes of length n. We call (U, V) a t-**error-locating** pair of C if the following hold:
$U * V \subseteq C^{\perp}, \qquad k(U) > t, \qquad d(V^{\perp}) > t.$
We call (U, V) be a t-**error-correcting** pair of C if it is error-locating and additionally satisfies $d(C) + d(U) > n$.

Codes which posses error-correcting pair have efficient decoding algorithm (a generalization of Berlecamp-Welch decoding algorithm for Reed-Solomon codes).

Unfortunately only few classes of codes are known to have such pairs.

Revisiting Error Correction Codes

Definition (Pellikaan 92)

Let U, V and C be linear codes of length n. We call (U, V) a t-**error-locating** pair of C if the following hold:
$U * V \subseteq C^{\perp}, \qquad k(U) > t, \qquad d(V^{\perp}) > t.$
We call (U, V) be a t-**error-correcting** pair of C if it is error-locating and additionally satisfies $d(C) + d(U) > n$.

Codes which posses error-correcting pair have efficient decoding algorithm (a generalization of Berlecamp-Welch decoding algorithm for Reed-Solomon codes).

Unfortunately only few classes of codes are known to have such pairs.

Revisiting Error Correction Codes

Definition (Pellikaan 92)

Let U, V and C be linear codes of length n. We call (U, V) a t-**error-locating** pair of C if the following hold:
$U * V \subseteq C^{\perp}, \qquad k(U) > t, \qquad d(V^{\perp}) > t.$
We call (U, V) be a t-**error-correcting** pair of C if it is error-locating and additionally satisfies $d(C) + d(U) > n$.

Codes which posses error-correcting pair have efficient decoding algorithm (a generalization of Berlecamp-Welch decoding algorithm for Reed-Solomon codes).

Unfortunately only few classes of codes are known to have such pairs.

Multiplicative SSS

Definition (Cramer et al. 00)

An MSP $\mathcal{M}$ is multiplicative MSP if there exists a vector $\boldsymbol{\lambda}$, such that for any two secrets s_1 and s_2 and for any random vectors $\mathbf{c^1}$ and $\mathbf{c^2}$ it holds: $s_1 s_2 = \langle \boldsymbol{\lambda}, M(s_1, \mathbf{c^1}) * M(s_2, \mathbf{c^2}) \rangle$.

Definition (Cramer et al. 03)

An MSP $\mathcal{M}$ is multiplicative if there exists a block-diagonal matrix D such that $M^T D M = \varepsilon \varepsilon^T$, where block-diagonal means that the non-zero entries of D are collected in blocks $D^{(j)}$ such that for every player P_i the rows and columns in $D^{(i)}$ are labeled by him.

Intuitively, SSS is multiplicative if each player P_i can, from his shares of secrets s_1 and s_2, compute shares of the product $s_1 s_2$ in such a way that together all players can reconstruct this product.

Multiplicative SSS and multiplicative Codes

Recall that for an MSP $\mathcal{M}$ there exists a code $\mathcal{C}$ with parity check matrix of the form $H = (\varepsilon \mid M^T)$.
Let $\mathcal{M}$ be a multiplicative MSP, i.e. let D be a block-diagonal matrix satisfying $M^T D M = \varepsilon\varepsilon^T$. Set $\overline{D} = \begin{pmatrix} -1 & \mathbf{0} \\ \mathbf{0} & D \end{pmatrix}$.

- A code $\mathcal{C}$ is called multiplicative [NN03] if the parity check matrix H satisfies equation $H\overline{D}H^T = 0$.
- Code $\mathcal{C}$ is called weakly self-dual if $\mathcal{C} \subsetneq \mathcal{C}^\perp$,
- Code $\mathcal{C}$ is called self-dual if $\mathcal{C} = \mathcal{C}^\perp$.
- For a weakly self-dual code $\mathcal{C}$ there exists a non-invertible matrix W such that $WH = G$.
- Therefore the self-dual codes are subset of the multiplicative codes, but a weakly self-dual code may not be multiplicative.

Multiplicative SSS and multiplicative Codes

Recall that for an MSP $\mathcal{M}$ there exists a code $\mathcal{C}$ with parity check matrix of the form $H = (\varepsilon \mid M^T)$.
Let $\mathcal{M}$ be a multiplicative MSP, i.e. let D be a block-diagonal matrix satisfying $M^T D M = \varepsilon\varepsilon^T$. Set $\overline{D} = \begin{pmatrix} -1 & \mathbf{0} \\ \mathbf{0} & D \end{pmatrix}$.

- A code $\mathcal{C}$ is called <u>multiplicative</u> [NN03] if the parity check matrix H satisfies equation $H\overline{D}H^T = 0$.
- Code $\mathcal{C}$ is called <u>weakly self-dual</u> if $\mathcal{C} \subsetneq \mathcal{C}^\perp$,
- Code $\mathcal{C}$ is called <u>self-dual</u> if $\mathcal{C} = \mathcal{C}^\perp$.
- For a weakly self-dual code $\mathcal{C}$ there exists a non-invertible matrix W such that $WH = G$.
- Therefore the self-dual codes are subset of the multiplicative codes, but a weakly self-dual code may not be multiplicative.

Multiplicative SSS and multiplicative Codes

Recall that for an MSP $\mathcal{M}$ there exists a code $\mathcal{C}$ with parity check matrix of the form $H = (\varepsilon \mid M^T)$.
Let $\mathcal{M}$ be a multiplicative MSP, i.e. let D be a block-diagonal matrix satisfying $M^T D M = \varepsilon\varepsilon^T$. Set $\overline{D} = \begin{pmatrix} -1 & \mathbf{0} \\ \mathbf{0} & D \end{pmatrix}$.

- A code $\mathcal{C}$ is called <u>multiplicative</u> [NN03] if the parity check matrix H satisfies equation $H\overline{D}H^T = 0$.
- Code $\mathcal{C}$ is called <u>weakly self-dual</u> if $\mathcal{C} \subsetneq \mathcal{C}^\perp$,
- Code $\mathcal{C}$ is called <u>self-dual</u> if $\mathcal{C} = \mathcal{C}^\perp$.
- For a weakly self-dual code $\mathcal{C}$ there exists a non-invertible matrix W such that $WH = G$.
- Therefore the self-dual codes are subset of the multiplicative codes, but a weakly self-dual code may not be multiplicative.

Multiplicative SSS and multiplicative Codes

Recall that for an MSP $\mathcal{M}$ there exists a code $\mathcal{C}$ with parity check matrix of the form $H = (\varepsilon \mid M^T)$.
Let $\mathcal{M}$ be a multiplicative MSP, i.e. let D be a block-diagonal matrix satisfying $M^T D M = \varepsilon\varepsilon^T$. Set $\overline{D} = \begin{pmatrix} -1 & \mathbf{0} \\ \mathbf{0} & D \end{pmatrix}$.

- A code $\mathcal{C}$ is called <u>multiplicative</u> [NN03] if the parity check matrix H satisfies equation $H\overline{D}H^T = 0$.
- Code $\mathcal{C}$ is called <u>weakly self-dual</u> if $\mathcal{C} \subsetneq \mathcal{C}^\perp$,
- Code $\mathcal{C}$ is called <u>self-dual</u> if $\mathcal{C} = \mathcal{C}^\perp$.
- For a weakly self-dual code $\mathcal{C}$ there exists a non-invertible matrix W such that $WH = G$.
- Therefore the self-dual codes are subset of the multiplicative codes, but a weakly self-dual code may not be multiplicative.

Multiplicative SSS and multiplicative Codes

- Chen et al. 07 define $\mathcal{C}$ to be "self-dual" if and only if there exist diagonal matrix W such that $W\mathcal{C} \subseteq \mathcal{C}^{\perp}$.
- Of course if $W = E$ (unit matrix) both definitions coincide, but there exists a counterexample for a code which is "self-dual" but not self-dual.
- Notice that we can rewrite definition for "self-dual" code into $GWG^T = 0$ where G is the generator matrix of the code $\mathcal{C}$.
- But then it follows that a "self-dual" code is the code which is dual to a multiplicative one.

Multiplicative SSS and multiplicative Codes

- Chen et al. 07 define $\mathcal{C}$ to be "self-dual" if and only if there exist diagonal matrix W such that $W\mathcal{C} \subseteq \mathcal{C}^{\perp}$.
- Of course if $W = E$ (unit matrix) both definitions coincide, but there exists a counterexample for a code which is "self-dual" but not self-dual.
- Notice that we can rewrite definition for "self-dual" code into $GWG^T = 0$ where G is the generator matrix of the code $\mathcal{C}$.
- But then it follows that a "self-dual" code is the code which is dual to a multiplicative one.

Multiplicative SSS and Codes with error-correcting pair

Cramer et al. 05 established an interesting connection between the problem of the strong multiplication in ideal linear SSSs and the existence of codes with error-correcting pair and hence with efficient decoding algorithms.

They have shown that all strongly multiplicative SSSs are in certain sense related to codes with error-correcting pairs and as consequence allow for efficient reconstruction of a shared secret in the presence of malicious faults.

Let's stress that SSS we want to recover the secret (i.e. the first coordinate) and not the whole codeword.

Multiplicative SSS and Codes with error-correcting pair

Cramer et al. 05 established an interesting connection between the problem of the strong multiplication in ideal linear SSSs and the existence of codes with error-correcting pair and hence with efficient decoding algorithms.

They have shown that all strongly multiplicative SSSs are in certain sense related to codes with error-correcting pairs and as consequence allow for efficient reconstruction of a shared secret in the presence of malicious faults.

Let's stress that SSS we want to recover the secret (i.e. the first coordinate) and not the whole codeword.

Multiplicative SSS and Codes with error-correcting pair

Cramer et al. 05 established an interesting connection between the problem of the strong multiplication in ideal linear SSSs and the existence of codes with error-correcting pair and hence with efficient decoding algorithms.

They have shown that all strongly multiplicative SSSs are in certain sense related to codes with error-correcting pairs and as consequence allow for efficient reconstruction of a shared secret in the presence of malicious faults.

Let's stress that SSS we want to recover the secret (i.e. the first coordinate) and not the whole codeword.

Multiplicative SSS and Codes with error-correcting pair

Observe that any SSS which has t-privacy and $(n-t)$-reconstruction is multiplicative!

Theorem (Chen et al. 07)

If $\mathcal{C}$ is "self-dual" code of length $n+1$ with minimum distance d, then the SSS based on $\mathcal{C}$ offers t-privacy and $n-t$-reconstruction with $t = d-2$ and hence it is multiplicative.

Let $\mathcal{C}$ be a code of length $n+1$ with minimum distance d, define $t(C) = \min(d, d^{\perp}) - 2$. Then the SSS based on $\mathcal{C}$ offers $t(C)$-privacy and $n - t(C)$-reconstruction and again is multiplicative.

Algebraic-geometric codes as a base for multiplicative MSP

- Algebraic-geometric code $\mathcal{C}$ is defined as $n+1$-tuple $\{(f(P_0), f(P_1), \ldots, f(P_n)) : f \in L(D)\}$, where $P_0, P_1, \ldots, P_n$ are points in affine/projective space and f runs through a specified set of functions.
- The similarity of AG codes to Reed-Solomon code (defined as $(f(0), f(1), f(2), ..., f(n))$ for a polynomial f) implies that AG codes have error-correcting pairs and as consequence efficient decoding algorithm.
- It can be directly seen that AG codes are multiplicative (in the same way as Reed-Solomon codes are).
- Hence (as shown by Chen et al. 07) algebraic-geometric codes generate multiplicative SSS!

Algebraic-geometric codes as a base for multiplicative MSP

- Algebraic-geometric code $\mathcal{C}$ is defined as $n+1$-tuple $\{(f(P_0), f(P_1), \ldots, f(P_n)) : f \in L(D)\}$, where $P_0, P_1, \ldots, P_n$ are points in affine/projective space and f runs through a specified set of functions.
- The similarity of AG codes to Reed-Solomon code (defined as $(f(0), f(1), f(2), ..., f(n))$ for a polynomial f) implies that AG codes have error-correcting pairs and as consequence efficient decoding algorithm.
- It can be directly seen that AG codes are multiplicative (in the same way as Reed-Solomon codes are).
- Hence (as shown by Chen et al. 07) algebraic-geometric codes generate multiplicative SSS!

Algebraic-geometric codes as a base for multiplicative MSP

- Algebraic-geometric code $\mathcal{C}$ is defined as $n+1$-tuple $\{(f(P_0), f(P_1), \ldots, f(P_n)) : f \in L(D)\}$, where $P_0, P_1, \ldots, P_n$ are points in affine/projective space and f runs through a specified set of functions.
- The similarity of AG codes to Reed-Solomon code (defined as $(f(0), f(1), f(2), ..., f(n))$ for a polynomial f) implies that AG codes have error-correcting pairs and as consequence efficient decoding algorithm.
- It can be directly seen that AG codes are multiplicative (in the same way as Reed-Solomon codes are).
- Hence (as shown by Chen et al. 07) algebraic-geometric codes generate multiplicative SSS!

Conclusions

- We have shown several interconnections between error correcting codes and secret sharing.
- McEliece and Sarwate, Brickel and Massey have established several relations between MDS (Reed-Solomon) codes and linear SSS (e.g. Shamir's)
- Their approach were generalized and led to definition of error-set correcting codes, which are a particular class of codes that correspond to general access structure SSS.
- It was shown that ideal multiplicative SSSs correspond to codes which posses error-correcting pairs and as a consequence they have efficient decoding/reconstruction algorithm.
- Several new constructions of multiplicative SSSs based on algebraic-geometric codes and Massey construction were proposed last 3 years, but the general case is still an open

Conclusions

- We have shown several interconnections between error correcting codes and secret sharing.
- McEliece and Sarwate, Brickel and Massey have established several relations between MDS (Reed-Solomon) codes and linear SSS (e.g. Shamir's)
- Their approach were generalized and led to definition of error-set correcting codes, which are a particular class of codes that correspond to general access structure SSS.
- It was shown that ideal multiplicative SSSs correspond to codes which posses error-correcting pairs and as a consequence they have efficient decoding/reconstruction algorithm.
- Several new constructions of multiplicative SSSs based on algebraic-geometric codes and Massey construction were proposed last 3 years, but the general case is still an open

Conclusions

- We have shown several interconnections between error correcting codes and secret sharing.
- McEliece and Sarwate, Brickel and Massey have established several relations between MDS (Reed-Solomon) codes and linear SSS (e.g. Shamir's)
- Their approach were generalized and led to definition of error-set correcting codes, which are a particular class of codes that correspond to general access structure SSS.
- It was shown that ideal multiplicative SSSs correspond to codes which posses error-correcting pairs and as a consequence they have efficient decoding/reconstruction algorithm.
- Several new constructions of multiplicative SSSs based on algebraic-geometric codes and Massey construction were proposed last 3 years, but the general case is still an open

Conclusions

- We have shown several interconnections between error correcting codes and secret sharing.
- McEliece and Sarwate, Brickel and Massey have established several relations between MDS (Reed-Solomon) codes and linear SSS (e.g. Shamir's)
- Their approach were generalized and led to definition of error-set correcting codes, which are a particular class of codes that correspond to general access structure SSS.
- It was shown that ideal multiplicative SSSs correspond to codes which posses error-correcting pairs and as a consequence they have efficient decoding/reconstruction algorithm.
- Several new constructions of multiplicative SSSs based on algebraic-geometric codes and Massey construction were proposed last 3 years, but the general case is still an open

Conclusions

- We have shown several interconnections between error correcting codes and secret sharing.
- McEliece and Sarwate, Brickel and Massey have established several relations between MDS (Reed-Solomon) codes and linear SSS (e.g. Shamir's)
- Their approach were generalized and led to definition of error-set correcting codes, which are a particular class of codes that correspond to general access structure SSS.
- It was shown that ideal multiplicative SSSs correspond to codes which posses error-correcting pairs and as a consequence they have efficient decoding/reconstruction algorithm.
- Several new constructions of multiplicative SSSs based on algebraic-geometric codes and Massey construction were proposed last 3 years, but the general case is still an open