CS 594 – Advanced Crypto (Spring 2026)

Alex Block

Lecture 8

February 11, 2026

Linear Secret Sharing (Continued)

Blakley Secret Sharing

- Around the same time as Shamir, Blakley designed another (t, n) secret-sharing scheme

Blakley Secret Sharing

- Around the same time as Shamir, Blakley designed another (t, n) secret-sharing scheme
- This scheme is based on geometric principles: in a t-dimensional vector space, any t non-parallel hyperplanes of dimension $(t-1)$ intersect at exactly 1 point

Blakley Secret Sharing

- Around the same time as Shamir, Blakley designed another (t, n) secret-sharing scheme

- This scheme is based on geometric principles: in a t-dimensional vector space, any t non-parallel hyperplanes of dimension $(t-1)$ intersect at exactly 1 point
  - Example: $t = 2$, any 2 non-parallel 1-dimension hyperplanes (i.e., lines) intersect at a single point.

Blakley Secret Sharing

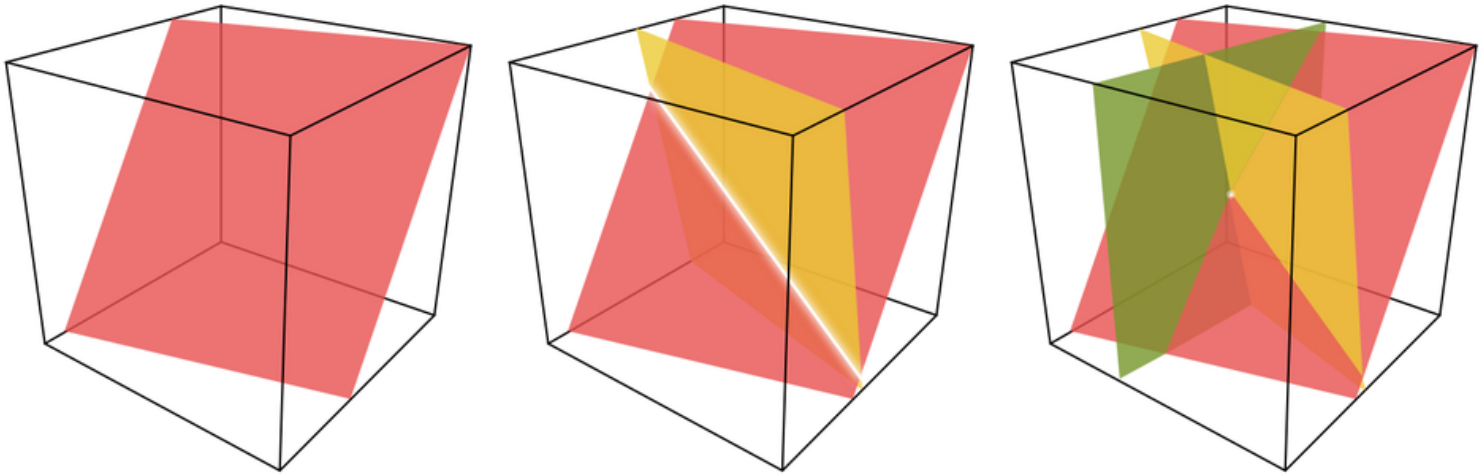
- Around the same time as Shamir, Blakley designed another (t, n) secret-sharing scheme

- This scheme is based on geometric principles: in a t-dimensional vector space, any t non-parallel hyperplanes of dimension $(t-1)$ intersect at exactly 1 point
  - Example: $t = 2$, any 2 non-parallel 1-dimension hyperplanes (i.e., lines) intersect at a single point.
  - Another example: $t = 3$, any 3 non-parallel 2-dimension hyperplanes (i.e., planes) intersect at a single point.

Blakley Secret Sharing

- Around the same time as Shamir, Blakley designed another (t, n) secret-sharing scheme

- This scheme is based on geometric principles: in a t-dimensional vector space, any t non-parallel hyperplanes of dimension $(t-1)$ intersect at exactly 1 point
  - Example: $t = 2$, any 2 non-parallel 1-dimension hyperplanes (i.e., lines) intersect at a single point.
  - Another example: $t = 3$, any 3 non-parallel 2-dimension hyperplanes (i.e., planes) intersect at a single point.



Blakley Secret Sharing

Blakley Secret Sharing

- Fix n, t.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Share(m) :

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Share(m) :
  - Sample random point/vector $\mathbf{x} \in \mathbb{F}^t$ such that $\mathbf{x}_1 = m$.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Share(m) :
  - Sample random point/vector $\mathbf{x} \in \mathbb{F}^t$ such that $\mathbf{x}_1 = m$.
  - For $i \in [n]$:

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- $\mathsf{Share}(m)$:
  - Sample random point/vector $\mathbf{x} \in \mathbb{F}^t$ such that $\mathbf{x}_1 = m$.
  - For $i \in [n]$:
    - For $j \in [t-1]$, sample $a_{i,j} \xleftarrow{\$} \mathbb{F}$.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- $\mathsf{Share}(m)$:
  - Sample random point/vector $\mathbf{x} \in \mathbb{F}^t$ such that $\mathbf{x}_1 = m$.
  - For $i \in [n]$:
    - For $j \in [t-1]$, sample $a_{i,j} \xleftarrow{\$} \mathbb{F}$.
    - Compute $c_i = \langle \mathbf{x}, (-a_{i,1}, \ldots, -a_{i,t-1}, 1) \rangle$.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- $\mathsf{Share}(m)$:
  - Sample random point/vector $\mathbf{x} \in \mathbb{F}^t$ such that $\mathbf{x}_1 = m$.
  - For $i \in [n]$:
    - For $j \in [t-1]$, sample $a_{i,j} \xleftarrow{\$} \mathbb{F}$.
    - Compute $c_i = \langle \mathbf{x}, (-a_{i,1}, \ldots, -a_{i,t-1}, 1) \rangle$.
    - Set $\mathbf{s}_i = (a_{i,1}, \ldots, a_{i,t-1}, c_i)$.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Share(m) :
  - Sample random point/vector $\mathbf{x} \in \mathbb{F}^t$ such that $\mathbf{x}_1 = m$.
  - For $i \in [n]$:
    - For $j \in [t-1]$, sample $a_{i,j} \xleftarrow{\$} \mathbb{F}$.
    - Compute $c_i = \langle \mathbf{x}, (-a_{i,1}, \ldots, -a_{i,t-1}, 1) \rangle$.
    - Set $\mathbf{s}_i = (a_{i,1}, \ldots, a_{i,t-1}, c_i)$.
  - Output $(\mathbf{s}_1, \ldots, \mathbf{s}_n)$.

$z \equiv a_{i,1} x_1 + \ldots + a_{i,t-1} x_{t-1} + c_i \pmod{\mathbb{F}}$

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Reconstruct$(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_m})$:

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Reconstruct$(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_m})$:
  - For each $k \in [m]$, parse $\mathbf{s}_{i_k} = (a_{i_k,1}, \ldots, a_{i_k,t-1}, c_{i_k})$.

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- $\mathsf{Reconstruct}(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_m})$:
  - For each $k \in [m]$, parse $\mathbf{s}_{i_k} = (a_{i_k,1}, \ldots, a_{i_k,t-1}, c_{i_k})$.
  - Initialize the linear system

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$.
- Reconstruct$(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_m})$:
  - For each $k \in [m]$, parse $\mathbf{s}_{i_k} = (a_{i_k,1}, \ldots, a_{i_k,t-1}, c_{i_k})$.
  - Initialize the linear system

$$\begin{bmatrix} a_{i_1,1} & \cdots & a_{i_1,t-1} & -1 \\ a_{i_2,1} & \cdots & a_{i_2,t-1} & -1 \\ \vdots & & \vdots & \vdots \\ a_{i_m,1} & \cdots & a_{i_m,t-1} & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} = \begin{bmatrix} -c_{i_1} \\ -c_{i_2} \\ \vdots \\ -c_{i_m} \end{bmatrix}$$

Blakley Secret Sharing

- Fix n, t.
- Set $\mathcal{M} = \mathbb{F}$, $\mathcal{S} = \mathbb{F}^t$. $\neq [n] \times \mathbb{F}^t$
- Reconstruct$(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_m})$:
  - For each $k \in [m]$, parse $\mathbf{s}_{i_k} = (a_{i_k,1}, \ldots, a_{i_k,t-1}, c_{i_k})$.
  - Initialize the linear system

$$\begin{bmatrix} a_{i_1,1} & \cdots & a_{i_1,t-1} & -1 \\ a_{i_2,1} & \cdots & a_{i_2,t-1} & -1 \\ \vdots & & \vdots & \vdots \\ a_{i_m,1} & \cdots & a_{i_m,t-1} & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} = \begin{bmatrix} -c_{i_1} \\ -c_{i_2} \\ \vdots \\ -c_{i_m} \end{bmatrix}$$

  - Solve the linear system and output $m = x_1$.

Blakley Secret Sharing: Correctness

Blakley Secret Sharing: Correctness

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| \geqslant t$, we have

$$\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in T})] = 1.$$

Blakley Secret Sharing: Correctness

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| \geqslant t$, we have

$$\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in T})] = 1.$$

"Proof:"

Blakley Secret Sharing: Correctness

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| \geqslant t$, we have

$$\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in T})] = 1.$$

"Proof:"

- From the definition of $\mathsf{Share}(m)$, reconstruction is done by solving the following system of linear equations, where $m \geqslant t$:

Blakley Secret Sharing: Correctness

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| \geqslant t$, we have

$$\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in T})] = 1.$$

"Proof:"

- From the definition of $\mathsf{Share}(m)$, reconstruction is done by solving the following system of linear equations, where $m \geqslant t$:

$$\begin{bmatrix} a_{i_1,1} & \cdots & a_{i_1,t-1} & -1 \\ a_{i_2,1} & \cdots & a_{i_2,t-1} & -1 \\ \vdots & & \vdots & \vdots \\ a_{i_m,1} & \cdots & a_{i_m,t-1} & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} = \begin{bmatrix} -c_{i_1} \\ -c_{i_2} \\ \vdots \\ -c_{i_m} \end{bmatrix}$$

Blakley Secret Sharing: Correctness

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| \geqslant t$, we have

$$\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in T})] = 1.$$

"Proof:"

- From the definition of $\mathsf{Share}(m)$, reconstruction is done by solving the following system of linear equations, where $m \geqslant t$:

$$\begin{bmatrix} a_{i_1,1} & \cdots & a_{i_1,t-1} & -1 \\ a_{i_2,1} & \cdots & a_{i_2,t-1} & -1 \\ \vdots & & \vdots & \vdots \\ a_{i_m,1} & \cdots & a_{i_m,t-1} & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} = \begin{bmatrix} -c_{i_1} \\ -c_{i_2} \\ \vdots \\ -c_{i_m} \end{bmatrix}$$

Issue!

What if the matrix is not full rank?

Blakley Secret Sharing: Correctness

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| \geqslant t$, we have

$$\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in T})] = 1.$$

"Proof:"

- From the definition of $\mathsf{Share}(m)$, reconstruction is done by solving the following system of linear equations, where $m \geqslant t$:

$$\begin{bmatrix} a_{i_1,1} & \cdots & a_{i_1,t-1} & -1 \\ a_{i_2,1} & \cdots & a_{i_2,t-1} & -1 \\ \vdots & & \vdots & \vdots \\ a_{i_m,1} & \cdots & a_{i_m,t-1} & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} = \begin{bmatrix} -c_{i_1} \\ -c_{i_2} \\ \vdots \\ -c_{i_m} \end{bmatrix}$$

Issue!

What if the matrix is not full rank?

- "Easy fix" ensure that the system is full rank when running Share

Blakley Secret Sharing: Security

Blakley Secret Sharing: Security

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| < t$, the following distributions are identical:

Blakley Secret Sharing: Security

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| < t$, the following distributions are identical:

$D_0(m, T)$:
return $(\mathsf{Share}(m)_i)_{i \in T}$

Blakley Secret Sharing: Security

- Need to argue: $\forall m \in \mathbb{F}$ and any $T \subset [n]$ such that $|T| < t$, the following distributions are identical:

$D_0(m, T)$:
return $(\mathsf{Share}(m)_i)_{i \in T}$

$D_1(m, T)$:
for $i \in T$
$\mathbf{s}_i \xleftarrow{\$} \mathbb{F}^t$
return $(\mathbf{s}_i)_{i \in T}$

$s_i := (a_{i,1}, \ldots, a_{i,t-1}, c_i)$

$|T| = t-1$

$\$ \quad \$ \quad \$$

$\Sigma \$$

non-parallel

Intersection of $(t-1)$ hyperplanes of dim $= (t-1)$? $\longrightarrow$ a line in dim $= t$

Blakley Secret Sharing: Efficiency

Blakley Secret Sharing: Efficiency

- Share(m)

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$; $O(t)$ work

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

$(m, x_2, \ldots, x_t)$

$c_i = \langle x, (-a_{i,1}, \ldots, -a_{i,t-1}, -1) \rangle$ $O(t)$

$\$\$$

$O(t)$ random samples per party

$O(t)$ $\mathbb{F}$-ops per party

$\rightarrow O(nt)$

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

- Reconstruct($\mathbf{s}_1, \ldots, \mathbf{s}_t$)

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

- Reconstruct($\mathbf{s}_1, \ldots, \mathbf{s}_t$)
  - Solving a linear system of t equations in t unknowns

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

- Reconstruct($\mathbf{s}_1, \ldots, \mathbf{s}_t$)
  - Solving a linear system of t equations in t unknowns
  - Worst case: $O(t^3)$, or a fancier algorithm which does $O(t^3/\log_q(t))$, where $q = |\mathbb{F}|$.

$$A X^T = C^T$$

$$\rightarrow X^T = A^{-1} C^T$$

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

- Reconstruct($\mathbf{s}_1, \ldots, \mathbf{s}_t$)
  - Solving a linear system of t equations in t unknowns
  - Worst case: $O(t^3)$, or a fancier algorithm which does $O(t^3/\log_q(t))$, where $q = |\mathbb{F}|$.
    - https://www.sciencedirect.com/science/article/pii/S0196885807000711

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

- Reconstruct($\mathbf{s}_1, \ldots, \mathbf{s}_t$)
  - Solving a linear system of t equations in t unknowns
  - Worst case: $O(t^3)$, or a fancier algorithm which does $O(t^3/\log_q(t))$, where $q = |\mathbb{F}|$.
    - https://www.sciencedirect.com/science/article/pii/S0196885807000711

- Share Size

Blakley Secret Sharing: Efficiency

- Share(m)
  - Sample $t-1$ random points in $\mathbb{F}$;
  - For each party $i \in [n]$, sample $t-1$ additional random points in $\mathbb{F}$, and compute an inner product of size t.

- Reconstruct($\mathbf{s}_1, \ldots, \mathbf{s}_t$)
  - Solving a linear system of t equations in t unknowns
  - Worst case: $O(t^3)$, or a fancier algorithm which does $O(t^3/\log_q(t))$, where $q = |\mathbb{F}|$.
    - https://www.sciencedirect.com/science/article/pii/S0196885807000711

- Share Size
  - Each party stores t elements of $\mathbb{F}$

Linear Secret Sharing from Linear Codes

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

- k is the *message length*; and

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

- k is the *message length*; and
- n is the *block/codeword length*.

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

- k is the *message length*; and
- n is the *block/codeword length*.

Equivalent Formulation of Linear Codes

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

- k is the *message length*; and
- n is the *block/codeword length*.

Equivalent Formulation of Linear Codes

Let $\mathbf{G} \in \mathbb{F}^{k \times n}$.

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

- k is the *message length*; and
- n is the *block/codeword length*.

Equivalent Formulation of Linear Codes

Let $\mathbf{G} \in \mathbb{F}^{k \times n}$. Then, $C = \{\mathbf{x} \cdot \mathbf{G} \colon \mathbf{x} \in \mathbb{F}^k\}$ is a $[n, k]_q$-linear code.

Linear Secret Sharing from Linear Codes

- McEliece and Sarwate, and later Massey, showed how to build Linear Secret Sharing from any *Linear Code*.

Definition 1 (Linear Codes)

A set $C \subseteq \mathbb{F}_q^n$ is a $[n, k]_q$-*linear code* a k-dimensional subspace of $\mathbb{F}_q^n$.

- k is the *message length*; and
- n is the *block/codeword length*.

Equivalent Formulation of Linear Codes

Let $\mathbf{G} \in \mathbb{F}^{k \times n}$. Then, $C = \{\mathbf{x} \cdot \mathbf{G} : \mathbf{x} \in \mathbb{F}^k\}$ is a $[n, k]_q$-linear code. We call $\mathbf{G}$ the *generator matrix* of the code C.

↑ encoding algorithm / encoder of C

Linear Codes: Distance and Dual Codes

Linear Codes: Distance and Dual Codes

Hamming wt : # non-zeros of x

Definition 2 (Distance of a Code)

Let C be a $[n, k]_q$ code. We say that C is a $[n, k, d]_q$ code if for all $\mathbf{x} \in C \setminus \{0^n\}$, we have $\mathbf{wt}(\mathbf{x}) \geqslant d$. We say that d is the *distance* of the code C.

$d = \max_{d'} \{ wt(x) \geq d' \ \forall x \in C \setminus \{0^n\} \}$

Linear Codes: Distance and Dual Codes

Definition 2 (Distance of a Code)

Let C be a $[n, k]_q$ code. We say that C is a $[n, k, d]_q$ code if for all $\mathbf{x} \in C \setminus \{0^n\}$, we have $\mathbf{wt}(\mathbf{x}) \geqslant d$. We say that d is the *distance* of the code C.

Definition 3 (Dual Code)

Let C be a $[n, k]_q$ code. Then, the *dual code of* C, denoted as $C^\perp$, is the unique $[n, n-k]_q$ code such that $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ for all $\mathbf{x} \in C$ and $\mathbf{y} \in C^\perp$.

Linear Codes: Distance and Dual Codes

Definition 2 (Distance of a Code)

Let C be a $[n, k]_q$ code. We say that C is a $[n, k, d]_q$ code if for all $\mathbf{x} \in C \setminus \{0^n\}$, we have $\mathbf{wt}(\mathbf{x}) \geqslant d$. We say that d is the *distance* of the code C.

Definition 3 (Dual Code)

Let C be a $[n, k]_q$ code. Then, the *dual code of* C, denoted as $C^\perp$, is the unique $[n, n-k]_q$ code such that $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ for all $\mathbf{x} \in C$ and $\mathbf{y} \in C^\perp$.

- $C^\perp$ is also known as the *parity check code*.

Linear Codes: Distance and Dual Codes

Definition 2 (Distance of a Code)

Let C be a $[n, k]_q$ code. We say that C is a $[n, k, d]_q$ code if for all $\mathbf{x} \in C \setminus \{0^n\}$, we have $\mathbf{wt}(\mathbf{x}) \geqslant d$. We say that d is the *distance* of the code C.

Definition 3 (Dual Code)

Let C be a $[n, k]_q$ code. Then, the *dual code of* C, denoted as $C^\perp$, is the unique $[n, n-k]_q$ code such that $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ for all $\mathbf{x} \in C$ and $\mathbf{y} \in C^\perp$.

- $C^\perp$ is also known as the *parity check code*.
- The generator matrix $\mathbf{H} \in \mathbb{F}^{(n-k)\times n}$ is called the *parity check matrix*

$G \cdot H^T = 0$, $H \cdot G^T = 0$

Linear Codes: Distance and Dual Codes

Definition 2 (Distance of a Code)

Let C be a $[n, k]_q$ code. We say that C is a $[n, k, d]_q$ code if for all $\mathbf{x} \in C \setminus \{0^n\}$, we have $\mathbf{wt}(\mathbf{x}) \geqslant d$. We say that d is the *distance* of the code C.

Definition 3 (Dual Code)

Let C be a $[n, k]_q$ code. Then, the *dual code of* C, denoted as $C^\perp$, is the unique $[n, n-k]_q$ code such that $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ for all $\mathbf{x} \in C$ and $\mathbf{y} \in C^\perp$.

- $C^\perp$ is also known as the *parity check code*.
- The generator matrix $\mathbf{H} \in \mathbb{F}^{(n-k)\times n}$ is called the *parity check matrix*
- We let $d^\perp$ denote the distance of $C^\perp$ and write that $C^\perp$ is a $[n, n-k, d^\perp]_q$ code.

Useful Facts About Linear Codes

Useful Facts About Linear Codes

Theorem 1 (Singleton Bound)

For any $[n, k, d]_q$ code, we have $k + d \leqslant n + 1$.

non-linear:

$A(n, d, q) \leq q^{n+1-d}$

$\frac{}{\#\text{ of vectors of length } n \text{ s.t. } \mathrm{HM}(x, x') \geq d}$

Useful Facts About Linear Codes

Theorem 1 (Singleton Bound)

For any $[n, k, d]_q$ code, we have $k + d \leqslant n + 1$.

- We say that a $[n, k, d]_q$ code is *Maximum Distance Separable (MDS)* if $k + d = n + 1$.

Useful Facts About Linear Codes

Theorem 1 (Singleton Bound)

For any $[n, k, d]_q$ code, we have $k + d \leqslant n + 1$.

- We say that a $[n, k, d]_q$ code is *Maximum Distance Separable (MDS)* if $k + d = n + 1$.
  - Example: Reed-Solomon Codes (we'll come back to this)

Useful Facts About Linear Codes

Theorem 1 (Singleton Bound)

For any $[n, k, d]_q$ code, we have $k + d \leqslant n + 1$.

- We say that a $[n, k, d]_q$ code is *Maximum Distance Separable (MDS)* if $k + d = n + 1$.
  - Example: Reed-Solomon Codes (we'll come back to this)

- Note that if C is a MDS code, then $C^{\perp}$ is also a MDS code.

Useful Facts About Linear Codes

Useful Facts About Linear Codes

Lemma 1

Let C be a $[n, k]_q$ code and let $\mathbf{H}$ be the parity check matrix. Then C has distance d if and only if $\mathbf{H}$ satisfies the following properties:

1. *There exists a set of d linearly dependent columns; and*
2. *All sets of at most $d-1$ columns are linearly independent.*

Useful Facts About Linear Codes

Lemma 1

Let C be a $[n, k]_q$ code and let $\mathbf{H}$ be the parity check matrix. Then C has distance d if and only if $\mathbf{H}$ satisfies the following properties:

1. *There exists a set of d linearly dependent columns; and*
2. *All sets of at most $d-1$ columns are linearly independent.*

- Note the above also holds for dual distance $d^{\perp}$ with respect to the generator matrix $\mathbf{G}$.

Useful Facts About Linear Codes

Lemma 1

Let C be a $[n, k]_q$ code and let $\mathbf{H}$ be the parity check matrix. Then C has distance d if and only if $\mathbf{H}$ satisfies the following properties:

1. *There exists a set of d linearly dependent columns; and*
2. *All sets of at most $d-1$ columns are linearly independent.*

- Note the above also holds for dual distance $d^{\perp}$ with respect to the generator matrix $\mathbf{G}$.
- Additionally, if C is a MDS code, (1) above is replaced with "all sets of at least d columns are linearly dependent."

Linear Secret Sharing from MDS Codes

columns

Let C be a MDS $[n, k]_q$ code with generator matrix $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$.

Linear Secret Sharing from MDS Codes

Let C be a MDS $[n, k]_q$ code with generator matrix $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$.

$\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$
$\mathcal{M} = \mathbb{F}; \mathcal{S} = \mathbb{F}$

Linear Secret Sharing from MDS Codes

Let C be a MDS $[n,k]_q$ code with generator matrix $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$.

$\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$
$\mathcal{M} = \mathbb{F}; \mathcal{S} = \mathbb{F}$

$\underline{\mathsf{Share}(m)}$:

$\mathbf{u} = (m, \mathbf{v})$, where $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-1}$

return $\mathbf{s} = \mathbf{uG} \in \mathbb{F}^n$

$s_i = \langle u, g_i \rangle \in \mathbb{F}$

Linear Secret Sharing from MDS Codes

Let C be a MDS $[n,k]_q$ code with generator matrix $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$.

$\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$

$\mathcal{M} = \mathbb{F}; \mathcal{S} = \mathbb{F}$

$\underline{\mathsf{Share}(m)}$:

$\mathbf{u} = (m, \mathbf{v})$, where $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-1}$

return $\mathbf{s} = \mathbf{uG} \in \mathbb{F}^n$

$\underline{\mathsf{Reconstruct}(s_{i_1}, \ldots, s_{i_m}):}$

Parse $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$

Solve for $\mathbf{x}$:

$\mathbf{e}_1 = \sum_{j=1}^{m} x_j \cdot \mathbf{g}_{i_j}$

return $m = \sum_{j=1}^{m} x_j \cdot s_{i_j}$

$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{F}^k$

$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = (x_1 \cdots x_m)(g_{i_1}, \ldots, g_{i_m})$

Linear Secret Sharing from MDS Codes

Linear Secret Sharing from MDS Codes

Theorem 2

Let $\mathbf{G}$ *be the generator matrix of a* $[n, k-1]_q$ *MDS code. Then, the above scheme is a* (n, k) *linear secret sharing scheme.*

Linear Secret Sharing from MDS Codes

Theorem 2

Let $\mathbf{G}$ *be the generator matrix of a* $[n, k-1]_q$ *MDS code. Then, the above scheme is a* (n, k) *linear secret sharing scheme.*

Proof. Need to show:

Linear Secret Sharing from MDS Codes

Theorem 2

Let $\mathbf{G}$ *be the generator matrix of a* $[n, k-1]_q$ *MDS code. Then, the above scheme is a* (n, k) *linear secret sharing scheme.*

Proof. Need to show:

- Correctness

Linear Secret Sharing from MDS Codes

Theorem 2

Let $\mathbf{G}$ *be the generator matrix of a* $[n, k-1]_q$ *MDS code. Then, the above scheme is a* (n, k) *linear secret sharing scheme.*

Proof. Need to show:

- Correctness
- Security

Linear Secret Sharing from MDS Codes

Theorem 2

Let $\mathbf{G}$ *be the generator matrix of a* $[n, k-1]_q$ *MDS code. Then, the above scheme is a* (n, k) *linear secret sharing scheme.*

Proof. Need to show:

- Correctness
- Security
- Linearity

Linear SS from MDS Codes: Linearity

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$
    - $\mathbf{x} = (m, \mathbf{v})$ and $\mathbf{x}' = (m', \mathbf{v}')$ for $\mathbf{v}, \mathbf{v}' \xleftarrow{\$} \mathbb{F}^{k-2}$.

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$
    - $\mathbf{x} = (m, \mathbf{v})$ and $\mathbf{x}' = (m', \mathbf{v}')$ for $\mathbf{v}, \mathbf{v}' \xleftarrow{\$} \mathbb{F}^{k-2}$.
    - Then, for any $\alpha, \beta \in \mathbb{F}$, we have

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$
    - $\mathbf{x} = (m, \mathbf{v})$ and $\mathbf{x}' = (m', \mathbf{v}')$ for $\mathbf{v}, \mathbf{v}' \xleftarrow{\$} \mathbb{F}^{k-2}$.
    - Then, for any $\alpha, \beta \in \mathbb{F}$, we have

$$\alpha\mathbf{y} + \beta\mathbf{y}'$$

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$
    - $\mathbf{x} = (m, \mathbf{v})$ and $\mathbf{x}' = (m', \mathbf{v}')$ for $\mathbf{v}, \mathbf{v}' \xleftarrow{\$} \mathbb{F}^{k-2}$.
    - Then, for any $\alpha, \beta \in \mathbb{F}$, we have

$$\begin{aligned} &\alpha\mathbf{y} + \beta\mathbf{y}' \\ =&\alpha\mathbf{x}\mathbf{G} + \beta\mathbf{x}'\mathbf{G} \end{aligned}$$

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$
    - $\mathbf{x} = (m, \mathbf{v})$ and $\mathbf{x}' = (m', \mathbf{v}')$ for $\mathbf{v}, \mathbf{v}' \xleftarrow{\$} \mathbb{F}^{k-2}$.
    - Then, for any $\alpha, \beta \in \mathbb{F}$, we have

$$\begin{aligned} &\alpha\mathbf{y} + \beta\mathbf{y}' \\ =&\alpha\mathbf{x}\mathbf{G} + \beta\mathbf{x}'\mathbf{G} \\ =&(\alpha\mathbf{x} + \beta\mathbf{x}')\mathbf{G} \end{aligned}$$

Linear SS from MDS Codes: Linearity

- Linearity follows directly by definition since $\mathbf{G}$ is a linear map.
  - Let $m \neq m' \in \mathbb{F}$.
  - Let $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$ and $\mathbf{y}' = \mathbf{x}' \cdot \mathbf{G}$
    - $\mathbf{x} = (m, \mathbf{v})$ and $\mathbf{x}' = (m', \mathbf{v}')$ for $\mathbf{v}, \mathbf{v}' \xleftarrow{\$} \mathbb{F}^{k-2}$.
    - Then, for any $\alpha, \beta \in \mathbb{F}$, we have

$$\begin{aligned} & \alpha\mathbf{y} + \beta\mathbf{y}' \\ =& \alpha\mathbf{x}\mathbf{G} + \beta\mathbf{x}'\mathbf{G} \\ =& (\alpha\mathbf{x} + \beta\mathbf{x}')\mathbf{G} \\ =& \mathbf{z}\mathbf{G}, \end{aligned}$$

    - where $\mathbf{z} = (\alpha m + \beta m', \alpha\mathbf{v} + \beta\mathbf{v}')$.

Linear SS from MDS Codes: Correctness

Linear SS from MDS Codes: Correctness

- For correctness, we need to argue that any k parties can reconstruct the secret m for sharing $\mathbf{s} \leftarrow \mathbf{xG}$, where $\mathbf{x} = (m, \mathbf{v})$ for $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-2}$

Linear SS from MDS Codes: Correctness

- For correctness, we need to argue that any k parties can reconstruct the secret m for sharing $\mathbf{s} \leftarrow \mathbf{xG}$, where $\mathbf{x} = (m, \mathbf{v})$ for $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-2}$

- Since $\mathbf{G}$ is a $[n, k-1, d]_q$ MDS code, the dual code with parity matrix $\mathbf{H}$ is also a $[n, n-(k-1), d^{\perp}]_q$ MDS code.

Linear SS from MDS Codes: Correctness

- For correctness, we need to argue that any k parties can reconstruct the secret m for sharing $\mathbf{s} \leftarrow \mathbf{xG}$, where $\mathbf{x} = (m, \mathbf{v})$ for $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-2}$
- Since $\mathbf{G}$ is a $[n, k-1, d]_q$ MDS code, the dual code with parity matrix $\mathbf{H}$ is also a $[n, n-(k-1), d^{\perp}]_q$ MDS code.
- Since $\mathbf{H}$ is a MDS code, it holds that $d^{\perp} + n - (k-1) = n + 1$, which implies that $d^{\perp} = k$.

Linear SS from MDS Codes: Correctness

- For correctness, we need to argue that any k parties can reconstruct the secret m for sharing $\mathbf{s} \leftarrow \mathbf{xG}$, where $\mathbf{x} = (m, \mathbf{v})$ for $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-2}$
- Since $\mathbf{G}$ is a $[n, k-1, d]_q$ MDS code, the dual code with parity matrix $\mathbf{H}$ is also a $[n, n-(k-1), d^{\perp}]_q$ MDS code.
- Since $\mathbf{H}$ is a MDS code, it holds that $d^{\perp} + n - (k-1) = n + 1$, which implies that $d^{\perp} = k$.
- This implies that every set of k columns of $\mathbf{G}$ is linearly dependent.

$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow e_1 = x_1 g_1 + x_2 g_2 + \cdots + x_k g_k$

can always solve

Linear SS from MDS Codes: Correctness

- For correctness, we need to argue that any k parties can reconstruct the secret m for sharing $\mathbf{s} \leftarrow \mathbf{xG}$, where $\mathbf{x} = (m, \mathbf{v})$ for $\mathbf{v} \xleftarrow{\$} \mathbb{F}^{k-2}$
- Since $\mathbf{G}$ is a $[n, k-1, d]_q$ MDS code, the dual code with parity matrix $\mathbf{H}$ is also a $[n, n-(k-1), d^{\perp}]_q$ MDS code.
- Since $\mathbf{H}$ is a MDS code, it holds that $d^{\perp} + n - (k-1) = n + 1$, which implies that $d^{\perp} = k$.
- This implies that every set of k columns of $\mathbf{G}$ is linearly dependent.
- So reconstruction with at least k parties will always succeed.

Linear SS from MDS Codes: Security

Linear SS from MDS Codes: Security

- For security, we need to argue that any set of $\leqslant k-1$ parties cannot learn anything about the secret.

Linear SS from MDS Codes: Security

- For security, we need to argue that any set of $\leqslant k-1$ parties cannot learn anything about the secret.

- More formally, any set of at most $k-1$ shares are uniformly random.

Linear SS from MDS Codes: Security

- For security, we need to argue that any set of $\leqslant k-1$ parties cannot learn anything about the secret.
- More formally, any set of at most $k-1$ shares are uniformly random.
- Follows since any set of $\leqslant k-1$ columns of $\mathbf{G}$ are linearly independent, and by construction of the secret sharing scheme.

$T \subseteq [n]$, $|T| \leq k-1$ $\quad T = \{1, \ldots, k-1\}$

$= k-1$

$u = (m, \$, \ldots, \$)$

$(\text{Share}(m)_i)_{i \in T} = (s_1, \ldots, s_{k-1})$

$s_i = (u_i \; g_i) = m \cdot g_{i,1} + \$$

$= \$$

Shamir Secret Sharing: Alternate View

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!
- Uses *Reed-Solomon codes*.

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!

- Uses *Reed-Solomon codes*.

Definition 4 (Reed-Solomon Code)

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!

- Uses *Reed-Solomon codes*.

Definition 4 (Reed-Solomon Code)

Let $\mathbb{F}_q$ be a finite field and let $k \leqslant n < q$.

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!

- Uses *Reed-Solomon codes*.

Definition 4 (Reed-Solomon Code)

Let $\mathbb{F}_q$ be a finite field and let $k \leqslant n < q$. Let $L \subset \mathbb{F}$ such that $|L| = n$.

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!

- Uses *Reed-Solomon codes*.

Definition 4 (Reed-Solomon Code)

Let $\mathbb{F}_q$ be a finite field and let $k \leqslant n < q$. Let $L \subset \mathbb{F}$ such that $|L| = n$. Then the *Reed-Solomon code*, denoted as $\mathsf{RS}[\mathbb{F}, L, k]$ is defined as

Shamir Secret Sharing: Alternate View

- Shamir is just a special case of the previous scheme!

- Uses *Reed-Solomon codes*.

Definition 4 (Reed-Solomon Code)

Let $\mathbb{F}_q$ be a finite field and let $k \leqslant n < q$. Let $L \subset \mathbb{F}$ such that $|L| = n$. Then the *Reed-Solomon code*, denoted as $\mathsf{RS}[\mathbb{F}, L, k]$ is defined as

$$\mathsf{RS}[\mathbb{F}, L, k] := \{(f(\ell))_{\ell \in L} \colon f(X) \in \mathbb{F}[X], \mathsf{deg}(f) \leqslant k\}$$

Shamir Secret Sharing: Reed-Solomon Codes

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.
  1. To share secret $m \in \mathbb{F}$, sample $\mathbf{c} \xleftarrow{\$} \mathsf{RS}[\mathbb{F}, L, t]$ such that $m = \mathsf{Interpolate}(\{(\ell_i, \mathbf{c}_i)\}_{i \in [n]})(0)$

ℓ_i is ith element of L

$(f(\ell_1), \ldots, f(\ell_n)) \in \mathbb{F}^n$

$1, \ldots, n$

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.
  1. To share secret $m \in \mathbb{F}$, sample $\mathbf{c} \xleftarrow{\$} \mathsf{RS}[\mathbb{F}, L, t]$ such that $m = \mathsf{Interpolate}(\{(\ell_i, \mathbf{c}_i)\}_{i \in [n]})$.
  2. Let $\mathbf{G} \in \mathbb{F}^{t \times n}$ be the generator matrix of $\mathsf{RS}[\mathbb{F}, L, t]$.

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.
  1. To share secret $m \in \mathbb{F}$, sample $\mathbf{c} \xleftarrow{\$} \mathsf{RS}[\mathbb{F}, L, t]$ such that $m = \mathsf{Interpolate}(\{(\ell_i, \mathbf{c}_i)\}_{i \in [n]})$.
  2. Let $\mathbf{G} \in \mathbb{F}^{t \times n}$ be the generator matrix of $\mathsf{RS}[\mathbb{F}, L, t]$. Then, to share $m \in \mathbb{F}$, share $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$, where $\mathbf{x} = (m, x_1, \ldots, x_{k-1})$ for $x_i \xleftarrow{\$} \mathbb{F}$.

$$f(X) = m + \sum_{i=1}^{t-1} x_i X^i$$

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.
  1. To share secret $m \in \mathbb{F}$, sample $\mathbf{c} \xleftarrow{\$} \mathsf{RS}[\mathbb{F}, L, t]$ such that $m = \mathsf{Interpolate}(\{(\ell_i, \mathbf{c}_i)\}_{i \in [n]})$.
  2. Let $\mathbf{G} \in \mathbb{F}^{t \times n}$ be the generator matrix of $\mathsf{RS}[\mathbb{F}, L, t]$. Then, to share $m \in \mathbb{F}$, share $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$, where $\mathbf{x} = (m, x_1, \ldots, x_{k-1})$ for $x_i \xleftarrow{\$} \mathbb{F}$.

What is $\mathbf{G}$?

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.
  1. To share secret $m \in \mathbb{F}$, sample $\mathbf{c} \xleftarrow{\$} \mathsf{RS}[\mathbb{F}, L, t]$ such that $m = \mathsf{Interpolate}(\{(\ell_i, \mathbf{c}_i)\}_{i \in [n]})$.
  2. Let $\mathbf{G} \in \mathbb{F}^{t \times n}$ be the generator matrix of $\mathsf{RS}[\mathbb{F}, L, t]$. Then, to share $m \in \mathbb{F}$, share $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$, where $\mathbf{x} = (m, x_1, \ldots, x_{k-1})$ for $x_i \xleftarrow{\$} \mathbb{F}$.

What is $\mathbf{G}$?

The Vandermonde Matrix! (with respect to L)

Shamir Secret Sharing: Reed-Solomon Codes

- Shamir SS can be viewed in two equivalent ways in terms of RS codes.
  1. To share secret $m \in \mathbb{F}$, sample $\mathbf{c} \xleftarrow{\$} \mathsf{RS}[\mathbb{F}, L, t]$ such that $m = \mathsf{Interpolate}(\{(\ell_i, \mathbf{c}_i)\}_{i \in [n]})$.
  2. Let $\mathbf{G} \in \mathbb{F}^{t \times n}$ be the generator matrix of $\mathsf{RS}[\mathbb{F}, L, t]$. Then, to share $m \in \mathbb{F}$, share $\mathbf{y} = \mathbf{x} \cdot \mathbf{G}$, where $\mathbf{x} = (m, x_1, \ldots, x_{k-1})$ for $x_i \xleftarrow{\$} \mathbb{F}$.

$L = [n]$

What is $\mathbf{G}$?

The Vandermonde Matrix! (with respect to L)

$$\begin{matrix} 1 \\ \ell_0 \\ \ell_0^2 \\ \vdots \\ \ell_0^{t-1} \end{matrix} \begin{bmatrix} 1 & \cdots & 1 \\ \ell_1 & \cdots & \ell_{n-1} \\ \ell_1^2 & \cdots & \ell_{n-1}^2 \\ \vdots & \ddots & \vdots \\ \ell_1^{t-1} & \cdots & \ell_{n-1}^{t-1} \end{bmatrix} \begin{matrix} n \\ n^2 \\ \\ n^{t-1} \end{matrix}$$

$\mathbb{F}^{t \times n} \ni$ ℓ_i are all distinct

$\omega, \omega^2, \omega^3, \ldots, \omega^{n-1}$

- If I have t valid shares

$S_{i_1}, \ldots, S_{i_t}$, then

$$\sum_{k=1}^{t} \lambda_{i_k} \cdot S_{i_k} = 0$$

Access Structure Secret Sharing from Linear Codes

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Definition 5 (Access Structures)

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Definition 5 (Access Structures)

Let n be an integer (the number of parties).

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Definition 5 (Access Structures)

Let n be an integer (the number of parties). Let $\Gamma \subseteq 2^{[n]}$ be a collection (a set of subsets of $[n]$).

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Definition 5 (Access Structures)

Let n be an integer (the number of parties). Let $\Gamma \subseteq 2^{[n]}$ be a collection (a set of subsets of $[n]$). We say that Γ is *monotone* if $B \in \Gamma$ and $B \subseteq C$ implies that $C \in \Gamma$.

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Definition 5 (Access Structures)

Let n be an integer (the number of parties). Let $\Gamma \subseteq 2^{[n]}$ be a collection (a set of subsets of $[n]$). We say that Γ is *monotone* if $B \in \Gamma$ and $B \subseteq C$ implies that $C \in \Gamma$.

An *access structure* is a monotone collection $\Gamma \subseteq 2^{[n]}$ of non-empty subsets of $[n]$. $\{1,2,3,4\}$

$\{1,2\}$ $\{3,4\}$

$\{1,2,3\}$, $\{1,2,4\}$

$\{1,2,3,4\}$...

Access Structure Secret Sharing from Linear Codes

- More generally, Massey showed how to build secret sharing for *access structures* from linear codes.

Definition 5 (Access Structures)

Let n be an integer (the number of parties). Let $\Gamma \subseteq 2^{[n]}$ be a collection (a set of subsets of $[n]$). We say that Γ is *monotone* if $B \in \Gamma$ and $B \subseteq C$ implies that $C \in \Gamma$.

An *access structure* is a monotone collection $\Gamma \subseteq 2^{[n]}$ of non-empty subsets of $[n]$. If $S \in \Gamma$, we say that S is *authorized*; otherwise ($S \notin \Gamma$) it is *unauthorized*.

Access Structure Secret Sharing

Definition 6 (Secret Sharing for Access Structures)

Access Structure Secret Sharing

Definition 6 (Secret Sharing for Access Structures)

Let n be an integer and let $\mathcal{M}, \mathcal{S}$ be a set of secrets and shares, respectively.

Access Structure Secret Sharing

Definition 6 (Secret Sharing for Access Structures)

Let n be an integer and let $\mathcal{M}, \mathcal{S}$ be a set of secrets and shares, respectively. We say that a secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$ realizes an access structure Γ if the following hold.

Access Structure Secret Sharing

Definition 6 (Secret Sharing for Access Structures)

Let n be an integer and let $\mathcal{M}, \mathcal{S}$ be a set of secrets and shares, respectively. We say that a secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$ realizes an access structure Γ if the following hold.

- (Correctness) For all $m \in \mathcal{M}$ and all $S \in \Gamma$, we have $\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in S})] = 1$.

Access Structure Secret Sharing

Definition 6 (Secret Sharing for Access Structures)

Let n be an integer and let $\mathcal{M}, \mathcal{S}$ be a set of secrets and shares, respectively. We say that a secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$ realizes an access structure Γ if the following hold.

- (Correctness) For all $m \in \mathcal{M}$ and all $S \in \Gamma$, we have $\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in S})] = 1$.
- (Security) For all $m \neq m' \in \mathcal{M}$ and any $T \notin \Gamma$, the following two distributions are identical:

Access Structure Secret Sharing

Definition 6 (Secret Sharing for Access Structures)

Let n be an integer and let $\mathcal{M}, \mathcal{S}$ be a set of secrets and shares, respectively. We say that a secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Reconstruct})$ realizes an access structure Γ if the following hold.

- (Correctness) For all $m \in \mathcal{M}$ and all $S \in \Gamma$, we have $\Pr[m = \mathsf{Reconstruct}((\mathsf{Share}(m)_i)_{i \in S})] = 1$.
- (Security) For all $m \neq m' \in \mathcal{M}$ and any $T \notin \Gamma$, the following two distributions are identical:

$$(\mathsf{Share}(m)_i)_{i \in T} \equiv \left(\mathsf{Share}(m')_i\right)_{i \in T}.$$

Next Time: Verifiable Secret Sharing