Advanced Tools from Modern Cryptography

Lecture 3

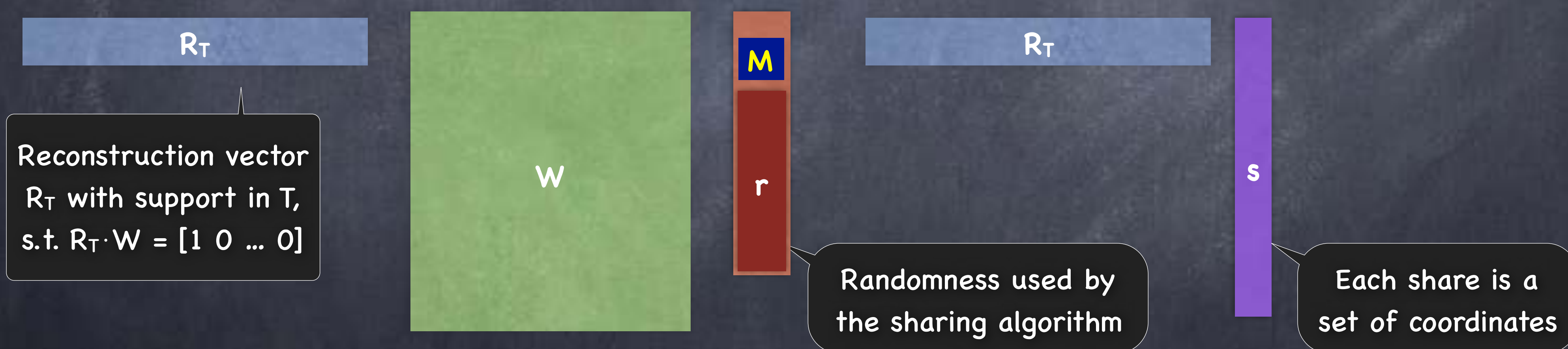
Secret-Sharing (ctd.)

Secret-Sharing

- Last time
  - (n,t) secret-sharing
    - (n,n) via additive secret-sharing
    - Shamir secret-sharing for general (n,t)
    - Shamir secret-sharing is a linear secret-sharing scheme

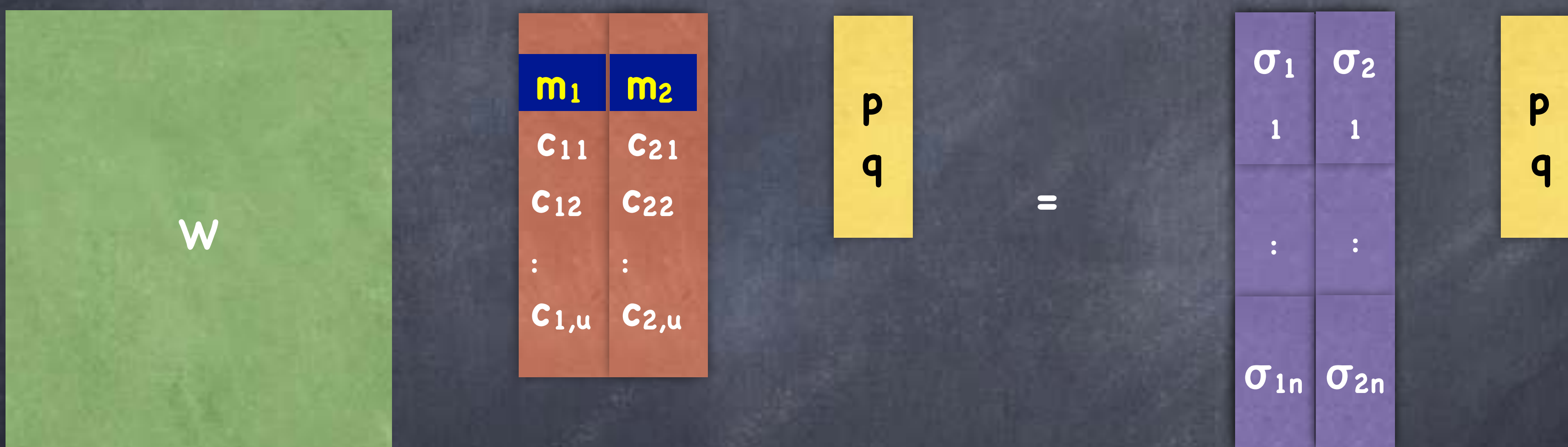
Linear Secret-Sharing

- Linear Secret-Sharing over a field: message and shares are field elements
- Reconstruction by a set $T \subseteq [n]$: solve $W_T \begin{bmatrix} M \\ r \end{bmatrix} = s_T$ for M



Linear Secret-Sharing: Computing on Shares

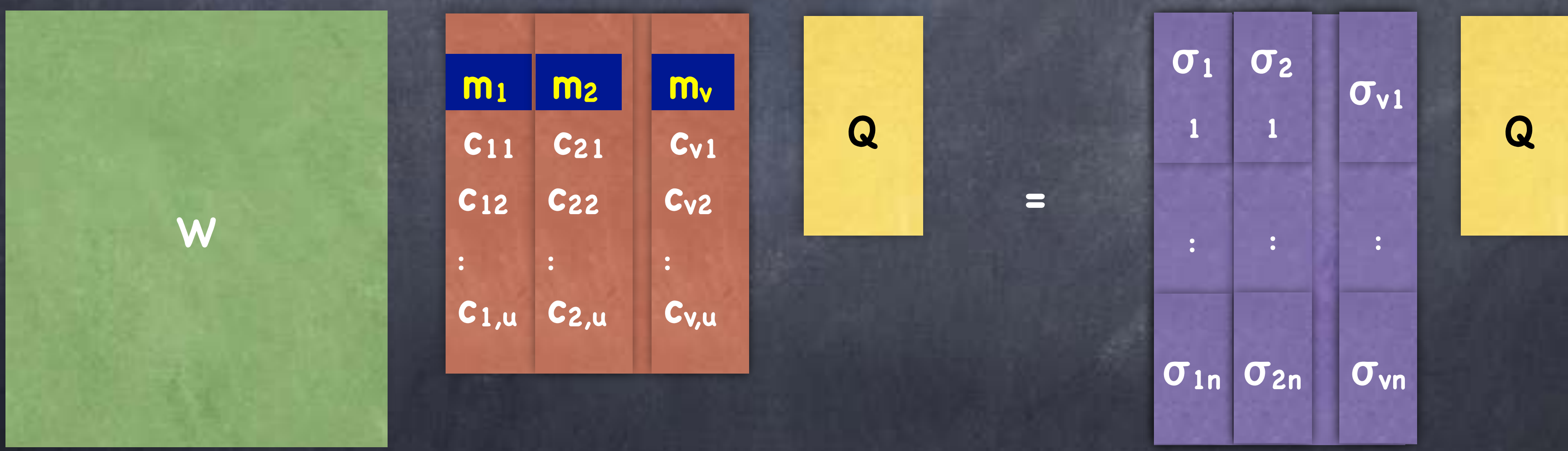
- Suppose two secrets m_1 and m_2 shared using the same secret-sharing scheme



- Then for any $p,q \in F$, shares of $p \cdot m_1 + q \cdot m_2$ can be computed locally by each party i as $\sigma_i = p \cdot \sigma_{1i} + q \cdot \sigma_{2i}$

Linear Secret-Sharing: Computing on Shares

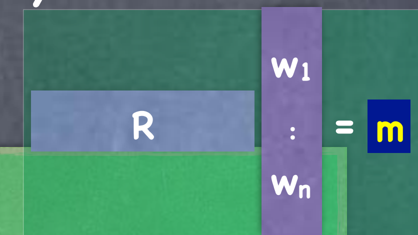
- More generally, can compute shares of any linear transformation



Each row computed locally

Switching Schemes

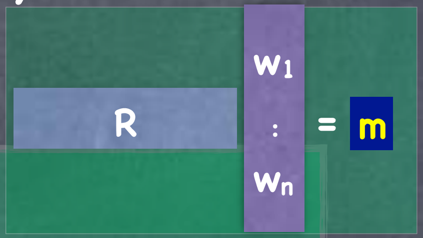
- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"



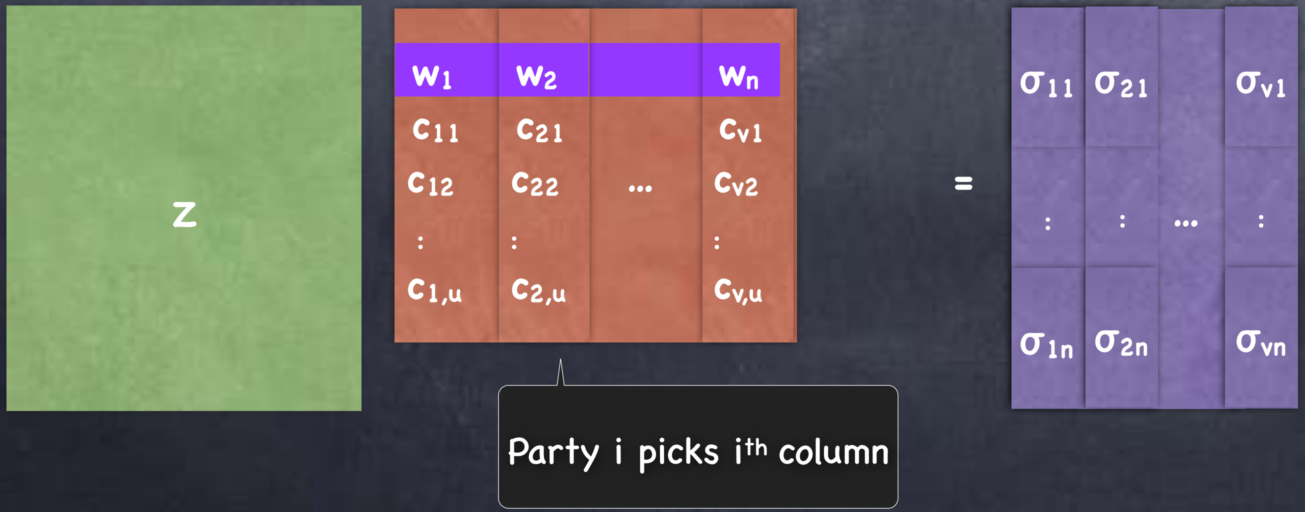
- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each w_i using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share(w_i)
- Locally each party j reconstructs using scheme W: $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$

Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"

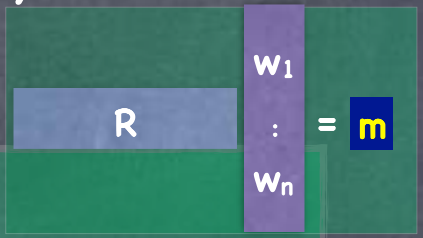


- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each w_i using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share(w_i)
- Locally each party j reconstructs using scheme W: $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$

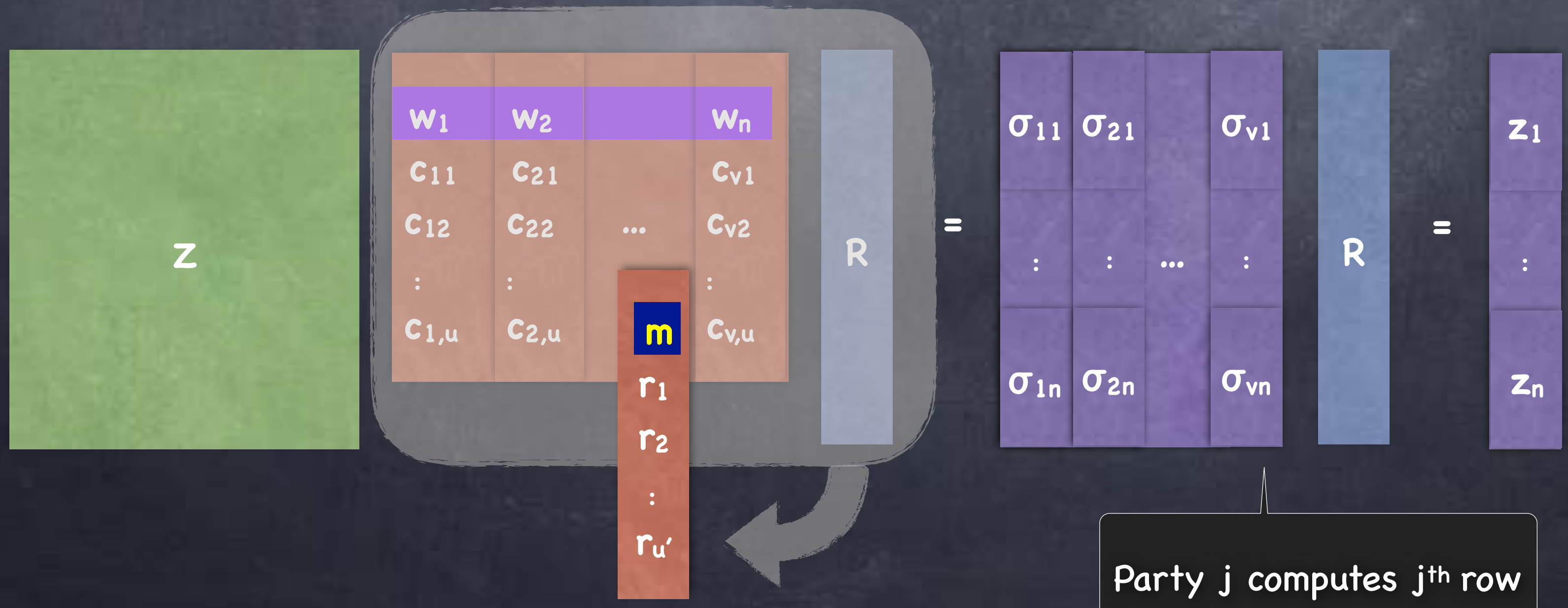


Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"



- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each w_i using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share(w_i)
- Locally each party j reconstructs using scheme W:
  $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$



Switching Schemes

- Can move from any linear secret-sharing scheme W to any other linear secret-sharing scheme Z "securely"

- Given shares $(w_1, \ldots, w_n) \leftarrow$ W.Share(m)
- Share each w_i using scheme Z: $(\sigma_{i1},\ldots,\sigma_{in}) \leftarrow$ Z.Share(w_i)
- Locally each party j reconstructs using scheme W:
  $z_j \leftarrow$ W.Recon $(\sigma_{1j},\ldots,\sigma_{nj})$

- Note that if a set of parties $T \subseteq [n]$ is allowed to learn the secret by either W or Z, then T learns m from either the shares it started with or the ones it ended up with

- Claim: If $T \subseteq [n]$ is not allowed to learn the secret by both W and Z, then T learns nothing about m from this process
  - Exercise

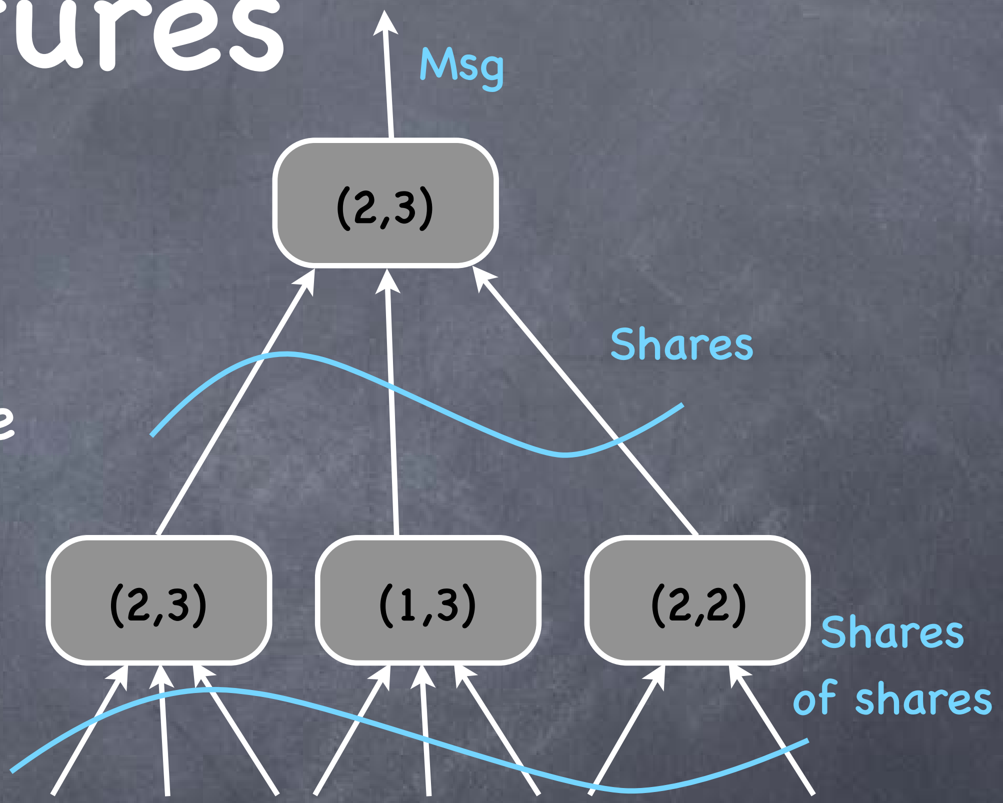
More General Access Structures

- Idea: For arbitrary monotonic access structure $\mathcal{A}$, there is a "basis" $\mathcal{B}$ of minimal sets in $\mathcal{A}$. For each S in $\mathcal{B}$ generate an (|S|,|S|) sharing, and distribute them to the members of S.
  - Works, but very "inefficient"

    $|\mathcal{B}| = \binom{n}{t}$
    - How big is $\mathcal{B}$? (Say when $\mathcal{A}$ is a threshold access structure)
    - Total share complexity = $\Sigma_{S \in \mathcal{B}} |S|$ field elements. (Compare with Shamir's scheme: n field elements in all.)

      $t \cdot \binom{n}{t}$
  - More efficient schemes known for large classes of access structures

More General Access Structures



- A simple generalization of threshold access structures
  - A threshold tree to specify the access structure
  - Can realize by recursively threshold secret-sharing the shares
- Note: linear secret-sharing
- Fact: Access structures that admit linear secret-sharing are those which can be specified using "monotone span programs"

Efficiency

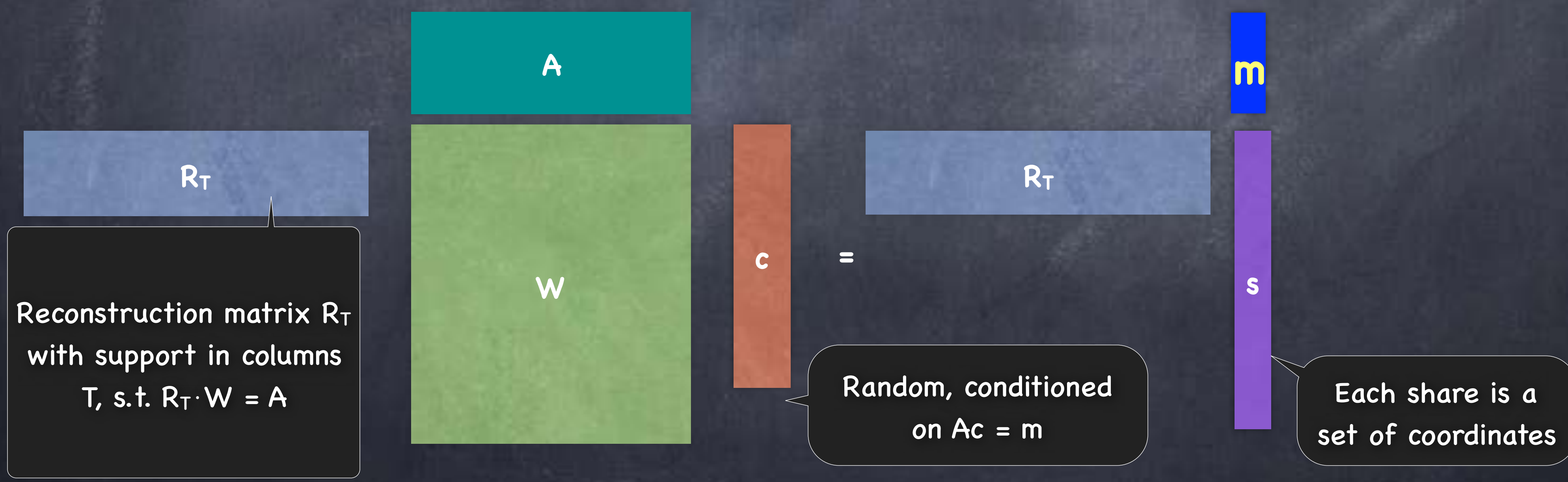
- Main measure: size of the shares (say, total of all shares)
  - Shamir's: each share is as as big as the secret (a single field element)
  - Naïve scheme for arbitrary monotonic access structure: if a party is in N sets in $\mathcal{B}$, N basic shares
    - N can be exponential in n (as $\mathcal{B}$ can have exponentially many sets)
  - Share size must be at least as big as the secret: "last share" in a minimal authorized set should contain all the information about the secret
    - Ideal: if all shares are only this big (e.g. Shamir's scheme)
    - Not all access structures have ideal schemes
  - Non-linear schemes can be more efficient than linear schemes

A More General Formulation

- Access structure consists of a monotonically "increasing" family $\mathcal{A}$ (allowed to learn), and a monotonically "decreasing" family $\mathcal{F}$ (forbidden from learning), with $\mathcal{A} \cap \mathcal{F} = \emptyset$
  - $T \in \mathcal{A} \Rightarrow \forall S \supseteq T, S \in \mathcal{A}$. $T \in \mathcal{F} \Rightarrow \forall S \subseteq T, S \in \mathcal{F}$.
  - For $T \notin \mathcal{A} \cup \mathcal{F}$, no requirements of secrecy or learning the message
- E.g., Ramp secret-sharing scheme: $\mathcal{A} = \{ S \subseteq [n] \mid |S| \geq t \}$ and $\mathcal{F} = \{ S \subseteq [n] \mid |S| \leq s \}$, where $s < t$
  - When $s = t-1$, a threshold secret-sharing scheme

Packed Secret-Sharing

- Shamir's scheme can be generalized to a ramp scheme, such that longer secrets can be shared with the same share size
  - $m_j = f(z_j)$ and $s_i = f(a_i)$ where $\{z_1,\ldots,z_k\} \cap \{a_1,\ldots,a_n\} = \emptyset$ and f has degree t-1 (t being the reconstruction threshold)
  - Access structure: $\mathcal{A} = \{ S : |S| \geq t \}$ and $\mathcal{F} = \{ S : |S| \leq t\text{-}k \}$



- $T \in \mathcal{A}$ if A spanned by W_T, and $T \in \mathcal{F}$ if every row of A independent of W_T