Monotone Erasure Codes

Vivien Bammert

Annalisa Cimatti (University of Bern)
Orestis Alpos (Common Prefix)
Giuliano Losa (Stellar Development Foundation)
Christian Cachin (University of Bern)

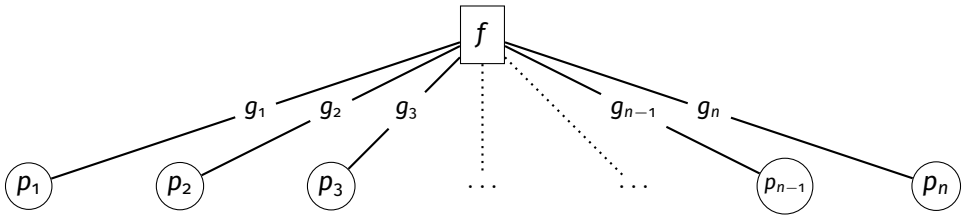
University of Bern
Institute of Computer Science
Cryptology and Data Security Group

May 9, 2026

Problem

Goal: Store a file f among the nodes in a set $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$.

Each node p_i holds a piece of information g_i, called a **fragment** of the file f.



Certain groups of nodes in $\mathcal{P}$ should be able to reconstruct f.

Erasure Codes

An **erasure code** splits data into fragments so that the original data can still be recovered even if some fragments are lost.

Erasure Codes

An **erasure code** splits data into fragments so that the original data can still be recovered even if some fragments are lost.

- *Maximum-distance separable (MDS)* codes (e.g., Reed-Solomon codes [RS60]) divide data into k pieces and encode them into $n > k$ fragments of equal size, where any k fragments allow reconstruction.

subsets of $\mathcal{P}$ of size at least k can reconstruct f

Threshold access structures

Access Structures

An **access structure** $\mathcal{A}$ on a set of nodes $\mathcal{P}$ is a collection of subsets of $\mathcal{P}$ such that no set is contained in another. Each set $A \in \mathcal{A}$ is called an **access set**.

Example: $\mathcal{A} = \{ \{a, b\}, \{b, c, d\}, \{b, c, e, f\} \}$

$\longrightarrow$ An access structure corresponds to a **monotone Boolean formula** (MBF).

Example: $(a \wedge b) \vee (b \wedge c \wedge d) \vee (b \wedge c \wedge e \wedge f)$

Access Structures

An **access structure** $\mathcal{A}$ on a set of nodes $\mathcal{P}$ is a collection of subsets of $\mathcal{P}$ such that no set is contained in another. Each set $A \in \mathcal{A}$ is called an **access set**.

- Secret sharing
- Voting power in blockchain systems
- Consensus protocols (e.g., XRP Ledger or Stellar network)

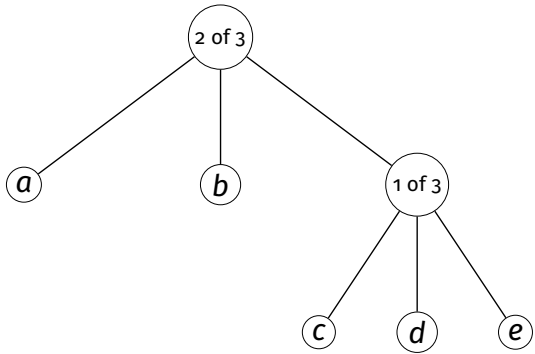
$\longrightarrow$ Access structures represent the underlying trust assumptions

$\longrightarrow$ Access sets specify which subset of nodes have enough power to perform certain actions

Representation of Access Structures

Let $\mathcal{A}$ be an access structure on $\mathcal{P}$

- An **access tree** of $\mathcal{A}$ is a rooted, labeled tree in which
  - internal vertex $\leftrightarrow$ threshold operator Θ_t^r (t of r)
  - leaf vertex $\leftrightarrow$ node in $\mathcal{P}$.



Contribution of this Work: Erasure Codes for any Access Structure

How can erasure codes be extended when the underlying assumption is captured by a general access structure?

$\longrightarrow$ Monotone erasure codes

$\longrightarrow$ Construction of *optimal* monotone erasure codes

$\longrightarrow$ Integrate monotone erasure codes into distributed protocols, e.g., AVID: Asynchronous Verifiable Information Dispersal [CT05]

Monotone Erasure Codes

Let $f \in \mathcal{F}$ be a file and let $\mathcal{G} = \mathcal{G}_1 \times \ldots \times \mathcal{G}_n$, where $\mathcal{G}_i$ are (possibly different) sets.

A **monotone erasure code** $\mathcal{C}$ for $\mathcal{A}$ on $\mathcal{P}$ is a scheme consisting of two functions:

$$f \in \mathcal{F} \xrightarrow{\text{Encode}} g = (g_1, g_2, \ldots, g_n) \in \mathcal{G}$$

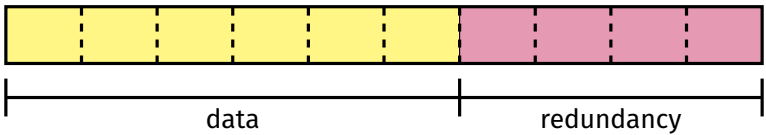
$g_1 \downarrow p_1$, $g_2 \downarrow p_2$, $g_n \downarrow p_n$ with $g_i \in \mathcal{G}_i$ the **fragment** of p_i

$$u \in \mathcal{G} \xrightarrow{\text{Decode}} \begin{cases} f' \in \mathcal{F} \\ \perp \end{cases}$$

Completeness of $\mathcal{C}$:
The nodes in each access set $A \in \mathcal{A}$ have enough information to reconstruct f, i.e., $f' = f$.

Overhead

The **overhead** β of $\mathcal{C}$ measures the amount of redundancy added to the original data during the encoding phase for tolerating loss of fragments.



$$\beta = \frac{\text{redundancy}}{\text{data}}$$

Efficiency: $\mathcal{C}$ should minimize the overhead

Linear Monotone Erasure Codes

An $[m, k]$-**linear monotone erasure code** $\mathcal{C}$ for $\mathcal{A}$ is a monotone erasure code for $\mathcal{A}$ that encodes files $f \in \mathbb{F}_q^k$ and produces fragments with combined size m with a linear operation.

ENCODE: $G \in \mathbb{F}_q^{k \times m}$ full-rank matrix with columns $(G_i)_{i \in [m]}$ and a labeling function $\phi : \{1, \ldots, m\} \to \mathcal{P}$

$$G = \left(\begin{array}{c|c|c|c} G_1 & G_2 & \ldots & G_m \\ \downarrow & \downarrow & & \downarrow \\ \phi(1) & \phi(2) & & \phi(m) \end{array} \right)$$

Linear Monotone Erasure Codes

An $[m, k]$-**linear monotone erasure code** $\mathcal{C}$ for $\mathcal{A}$ is a monotone erasure code for $\mathcal{A}$ that encodes files $f \in \mathbb{F}_q^k$ and produces fragments with combined size m with a linear operation.

ENCODE: $G \in \mathbb{F}_q^{k \times m}$ full-rank matrix with columns $(G_i)_{i \in [m]}$ and a labeling function $\phi : \{1, \ldots, m\} \to \mathcal{P}$

$$G = \left(\begin{array}{c|c|c|c} G_1 & G_2 & \ldots & G_m \\ \downarrow & \downarrow & & \downarrow \\ \phi(1) & \phi(2) & & \phi(m) \end{array} \right)$$

$$G_{p_i} = \left(\begin{array}{c|c|c|c} G_{j_1} & G_{j_2} & \ldots & G_{j_{m_i}} \end{array} \right)$$

with $\phi(j_1) = \ldots = \phi(j_{m_i}) = p_i$

$$g_i = \begin{cases} f \cdot G_{p_i} & m_i > 0 \\ \perp & m_i = 0 \end{cases} \quad \text{(fragment of } p_i\text{)}$$

Linear Threshold Erasure Codes

Let $\mathcal{A}$ be a threshold access structure on $\mathcal{P}$:

$$\mathcal{A} = \{A \in \mathcal{A} \mid |A| = w\} \text{ for some } w \in \mathbb{N}$$

An $[m, k]$-**linear threshold erasure code** $\mathcal{C}$ for $\mathcal{A}$ is an $[m, k]$-linear monotone erasure code where

$$m = n \text{ and } k = w$$

Linear Threshold Erasure Codes

Let $\mathcal{A}$ be a threshold access structure on $\mathcal{P}$:

$$\mathcal{A} = \{A \in \mathcal{A} \mid |A| = w\} \text{ for some } w \in \mathbb{N}$$

An $[m, k]$-**linear threshold erasure code** $\mathcal{C}$ for $\mathcal{A}$ is an $[m, k]$-linear monotone erasure code where

$$m = n \text{ and } k = w$$

$$G = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \vdots & & & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_m^{k-1} \end{pmatrix} \in \mathbb{F}_q^{k \times m}.$$

$$\phi(1) \downarrow \quad \phi(2) \downarrow \quad \phi(n) \downarrow$$

$$p_1 \quad p_2 \quad p_n$$

where G is a Vandermonde matrix with pairwise distinct $\alpha_1, \dots, \alpha_m \in \mathbb{F}_q^m$.

Linear Threshold Erasure Codes

Let $\mathcal{A}$ be a threshold access structure on $\mathcal{P}$:

$$\mathcal{A} = \{A \in \mathcal{A} \mid |A| = w\} \text{ for some } w \in \mathbb{N}$$

An $[m, k]$-**linear threshold erasure code** $\mathcal{C}$ for $\mathcal{A}$ is an $[m, k]$-linear monotone erasure code where

$$m = n \text{ and } k = w$$

$$G = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \vdots & & & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_m^{k-1} \end{pmatrix} \in \mathbb{F}_q^{k \times m}.$$

$$\phi(1) \downarrow \quad \phi(2) \downarrow \quad \phi(n) \downarrow$$

$$p_1 \quad p_2 \quad p_n$$

where G is a Vandermonde matrix with pairwise distinct $\alpha_1, \dots, \alpha_m \in \mathbb{F}_q^m$.

Linear Threshold Erasure Codes $\leftrightarrow$ MDS codes

Optimal Linear Monotone Erasure Codes

Given an access structure $\mathcal{A}$ on a set of nodes $\mathcal{P}$, the goal is to find

- Parameters m, k
  - m: Combined size of all fragments
  - k: File size
- Labeling function ϕ
- Encoding matrix G

defining an $[m, k]$-linear monotone erasure code $\mathcal{C}$ that is complete and has *minimal* storage overhead

$$\beta = \frac{m - k}{k}$$

Finding the Optimal Parameters

For a given access strucure $\mathcal{A}$

Goal: Find parameters m, k of a linear monotone erasure code $\mathcal{C}$ such that

$$\frac{m}{k} = \frac{1}{k} \sum_{i=1}^{n} m_i$$

is minimized and it holds

$$\sum_{p_i \in A} m_i \geq k \quad \forall A \in \mathcal{A}$$

$$y_i = \frac{m_i}{k} \longrightarrow$$

Linear Programming Problem (LPP):

Find $y = (y_1, \ldots, y_n) \in \mathbb{Q}^n_{\geq 0}$ such that

$$\min \sum_{i=1}^{n} y_i$$

$$\text{subject to} \sum_{p_i \in A} y_i \geq 1 \quad \forall A \in \mathcal{A}$$

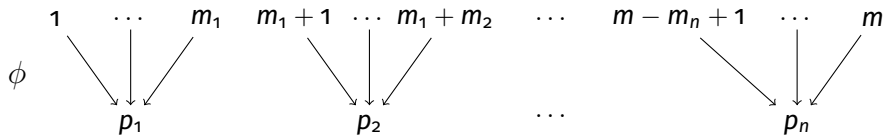
Scale to integer solution: choose $k =$ lcm of denominators of all y_i.

Building the Code from a Threshold Code

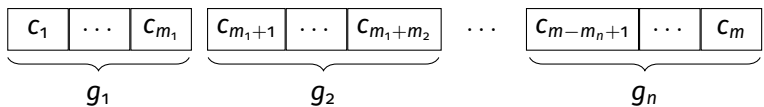
With optimal m, k and m_i for $i \in [n]$, we build an $[m, k]$-linear monotone erasure code:

Base code. $[m, k]$-linear threshold erasure code $\mathcal{C}'$ (MDS-code) over $\mathbb{F}_q$, where $q \geq m$

Labeling Function.



Encoding.

$\text{MDSEncode}(f) \to f \cdot G =$ 

Decoding. Corresponds to MDSDecode

Building the Code from a Threshold Code

Solving LPP is efficient if $\mathcal{A}$ has a compact description in the number of nodes

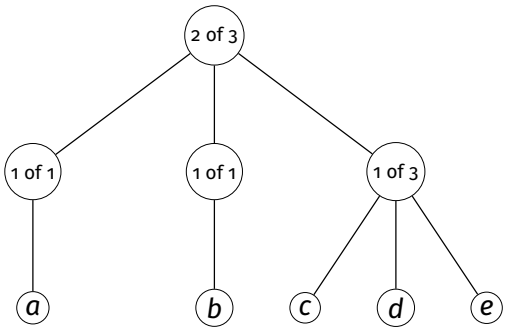
$\longrightarrow$ But there may be exponentially many (in n) access sets in $\mathcal{A}$!

Goal: Find the optimal parameters of the base code without solving the corresponding LP problem for a given access structure $\mathcal{A}$.

Partitioned Threshold Access Structures

A **partitioned threshold access structure** $\mathcal{A}$ on $\mathcal{P}$ is given by an access tree T where every node in $\mathcal{P}$ appears at most once in T.

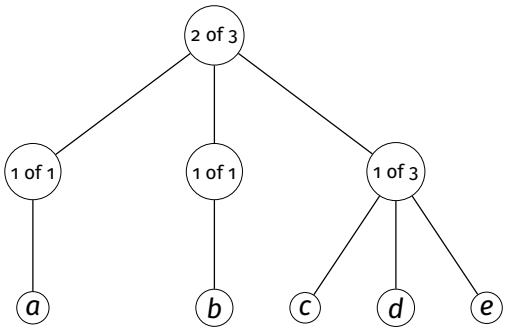
- Consensus protocol in the Stellar blockchain (*https://stellar.org*): Each organization has its own nodes, i.e., the node set $\mathcal{P}$ is partitioned into organizations $B_1, \ldots, B_\delta$.



Partitioned Threshold Access Structures

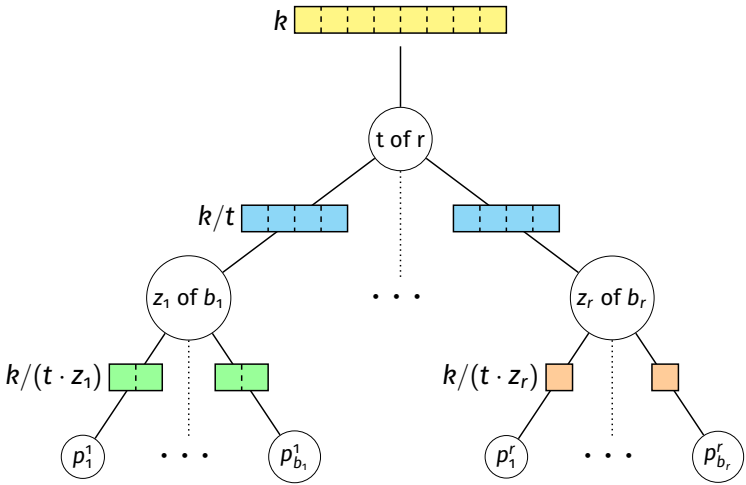
A **partitioned threshold access structure** $\mathcal{A}$ on $\mathcal{P}$ is given by an access tree T where every node in $\mathcal{P}$ appears at most once in T.

- Consensus protocol in the Stellar blockchain (*https://stellar.org*): Each organization has its own nodes, i.e., the node set $\mathcal{P}$ is partitioned into organizations $B_1, \ldots, B_\delta$.



A partitioned access structure is called **L-level partitioned threshold access structure** if all leaves have depth L in the tree.

Uniform Fragment Assignment

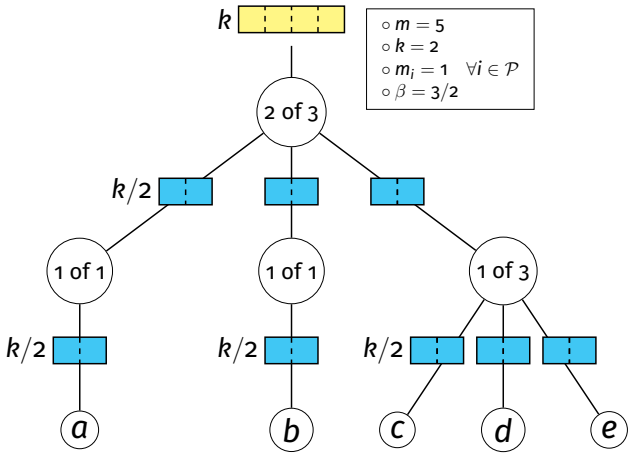


Given a 2-level partitioned access structure:

Base code $\mathcal{C}'$ has parameters:

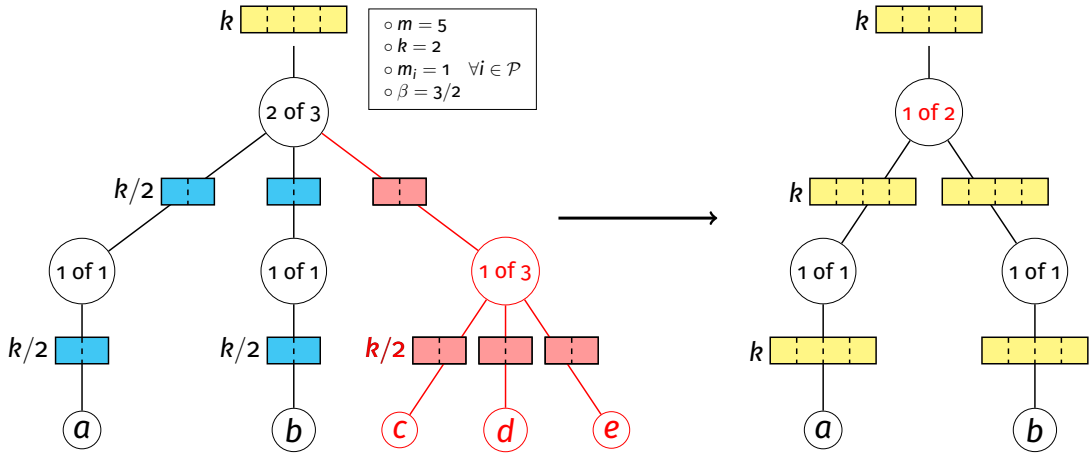
- $m = \sum_{i=1}^{r} b_i \cdot k/(t \cdot z_i)$
- $k = lcm(t \cdot z_1, \ldots, t \cdot z_r)$

Example - Tree Reduction

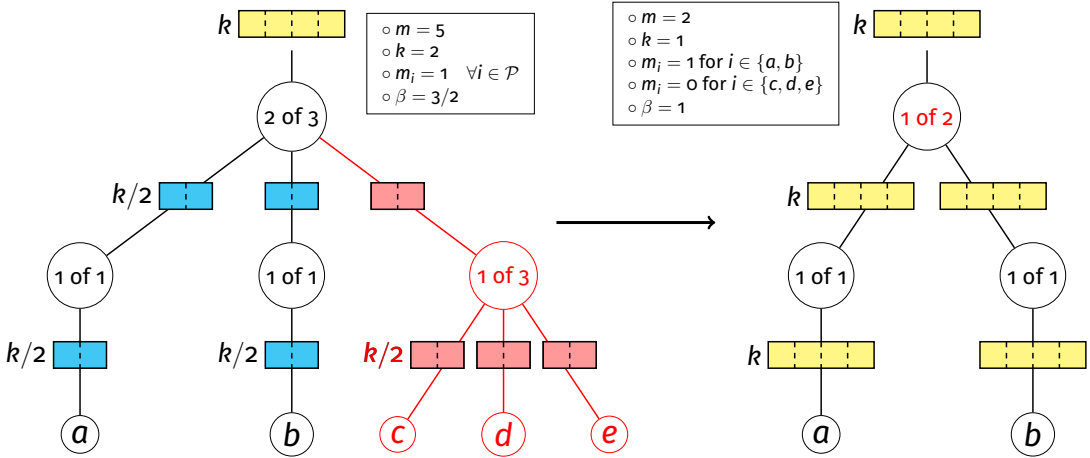


- $m = 5$
- $k = 2$
- $m_i = 1 \quad \forall i \in \mathcal{P}$
- $\beta = 3/2$

Example - Tree Reduction



Example - Tree Reduction



Algorithm

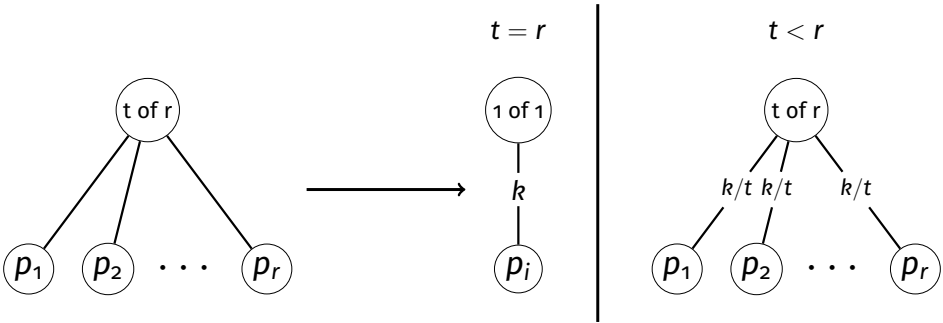
Let T be an access tree of an L-level partitioned threshold access structure

Observation: Overhead is minimized if all subtrees of T have an optimal fragment assignment

$\rightarrow$ Reduce T by pruning all subtrees with negative overhead contribution recursively in a depth-wise manner

$\rightarrow$ Applying uniform assignment to the reduced tree yields the optimal parameters of the base code

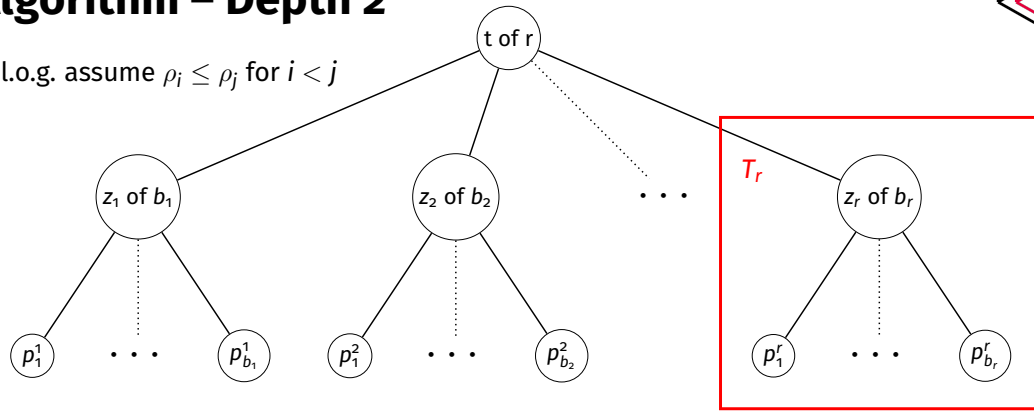
Algorithm – Depth 1



for each T_i compute $\rho_i = \frac{r}{t}$ (cost of T_i)

Algorithm – Depth 2

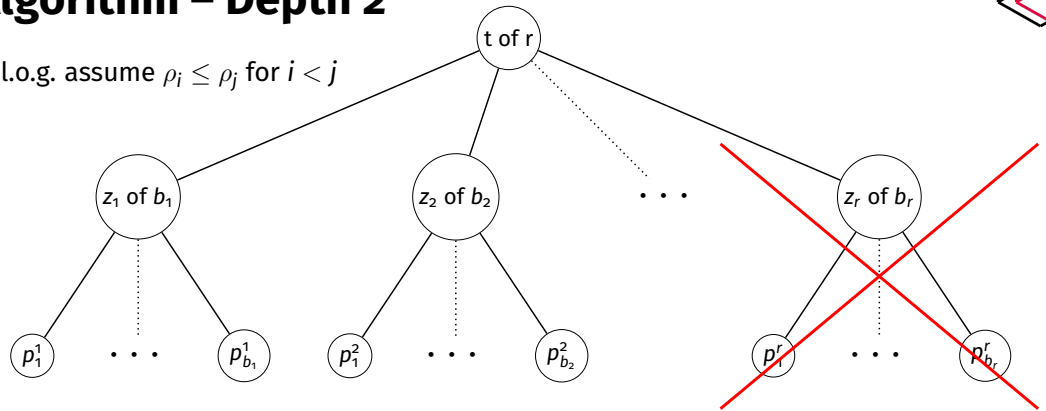
W.l.o.g. assume $\rho_i \leq \rho_j$ for $i < j$



Remove T_r if $t > 1$ and $\rho_r \geq \underbrace{\left(\frac{t}{t-1} - 1\right) \sum_{i=1}^{r-1} \rho_i}_{\text{additional cost}}$

Algorithm – Depth 2

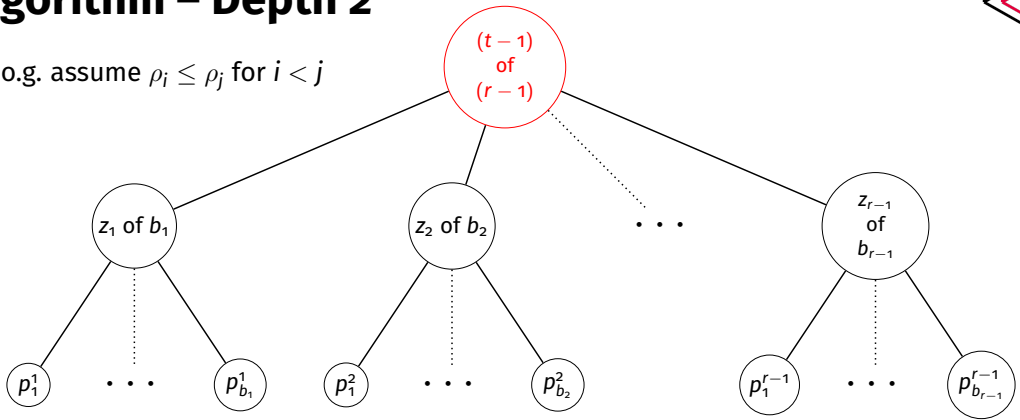
W.l.o.g. assume $\rho_i \leq \rho_j$ for $i < j$



Remove T_r if $t > 1$ and $\rho_r \geq \underbrace{\left(\frac{t}{t-1} - 1\right) \sum_{i=1}^{r-1} \rho_i}_{\text{additional cost}}$

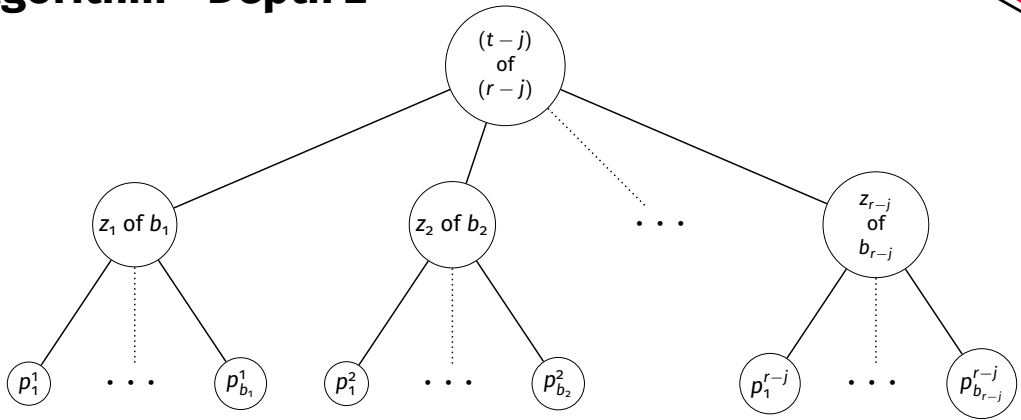
Algorithm – Depth 2

W.l.o.g. assume $\rho_i \leq \rho_j$ for $i < j$



Remove T_{r-1} if $t - 1 > 1$ and $\rho_{r-1} \geq \underbrace{\left(\frac{t-1}{t-2} - 1\right) \sum_{i=1}^{r-2} \rho_i}_{\text{additional cost}}$

Algorithm - Depth 2



Cost of T': $\rho_{T'} = \frac{1}{t-j} \sum_{i=1}^{r-j} \rho_i$ where $j \in \mathbb{N}$ is the number of pruned subtrees

Algorithm for any Depth

- Apply the procedure to all depth-3 subtrees
- Replace each subtree with a reduced tree and compute its cost
- Repeat recursively until T is fully reduced (if possible)

Applications

Goal: Allow a node to distribute data among n nodes such that it can always be recovered correctly despite Byzantine failures.

AVID [CT05]

- Up to t Byzantine nodes
- Threshold Byzantine quorum systems

Classical erasure code

GAVID

- Subsets of nodes, specified by the *fail-prone system*, may be Byzantine
- General Byzantine quorum systems [HM00; MR97]

Monotone erasure code

Future Work

- *Efficient* construction of optimal monotone erasure codes for *arbitrary* monotone access structures
- Constructing “flexible” monotone erasure codes that adapt to changing access structures
- Investigate how other methods from coding theory (e.g., rateless codes or low-density parity-check codes) can be employed in the setting of monotone erasure codes

Thank you for your attention

University of Bern
Institute of Computer Science
Cryptology and Data Security Group

May 9, 2026

References I

[CSB25] Dave Cohen et al. *XRP Ledger Documentation: Consensus Structure*. Available online, https://xrpl.org/docs/concepts/consensus-protocol/consensus-structure. 2025.

[CT05] Christian Cachin and Stefano Tessaro. "Asynchronous Verifiable Information Dispersal". In: *24th IEEE Symposium on Reliable Distributed Systems (SRDS 2005),26-28 October 2005, Orlando, FL, USA*. IEEE Computer Society, 2005, pp. 191–202. DOI: 10.1109/RELDIS.2005.9.

[HM00] Martin Hirt and Ueli M. Maurer. "Player Simulation and General Adversary Structures in Perfect Multiparty Computation". In: *J. Cryptol.* 13.1 (2000), pp. 31–60. DOI: 10.1007/S001459910003.

References II

[Lok+19] Marta Lokhava et al. “Fast and secure global payments with Stellar”. In: *Proceedings of the 27th ACM Symposium on Operating Systems Principles, SOSP 2019, Huntsville, ON, Canada, October 27-30, 2019*. Ed. by Tim Brecht and Carey Williamson. ACM, 2019, pp. 80–96. DOI: 10.1145/3341301.3359636. URL: https://doi.org/10.1145/3341301.3359636.

[MR97] Dahlia Malkhi and Michael K. Reiter. “Byzantine Quorum Systems”. In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*. Ed. by Frank Thomson Leighton and Peter W. Shor. ACM, 1997, pp. 569–578. DOI: 10.1145/258533.258650. URL: https://doi.org/10.1145/258533.258650.

[RS60] I. S. Reed and G. Solomon. “Polynomial Codes Over Certain Finite Fields”. In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304. DOI: 10.1137/0108018.

Weighted Access Structure

Proof-of-stake blockchains use a **weighted** access structure $\mathcal{A}$: each node $p_i \in \mathcal{P}$ has a weight w_i.

Access sets: $A \subseteq \mathcal{P}$ such that $\sum_{p_i \in A} w_i > W$, for some $W \in \mathbb{R}$.

Then:

- $\mathcal{A}$ is monotone, so we can apply our algorithm to find an optimal monotone erasure code for $\mathcal{A}$
- $\mathcal{A}$ is not partitioned, so the algorithm is inefficient

Open problem: Find an efficient way to build an optimal linear monotone erasure code for $\mathcal{A}$