UNIVERSITY OF TWENTE.

How to reconstruct (anonymously) a secret CA

Luca Mariot, Federico Mazzone, Luca Manzoni, Alberto Leporati

l.mariot@utwente.nl

Automata 2026

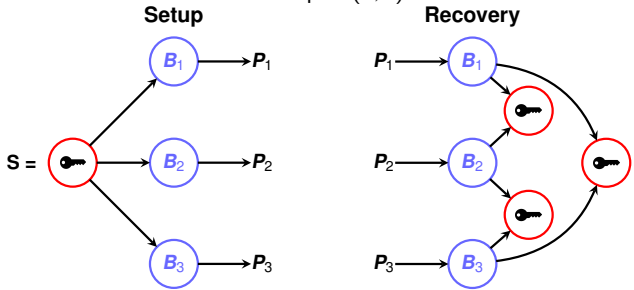
Ghent, July 6, 2026

Latin Squares & Secret Sharing Schemes

Threshold Secret Sharing Schemes

(k, n) **Threshold Secret Sharing Scheme**: a **dealer** shares a **secret** S among n **players** so that at least k players out of n are required to recover S [S79]

Example: $(2,3)$–scheme



Remark: $(2, n)$–scheme $\Leftrightarrow$ set of n-MOLS [S04]

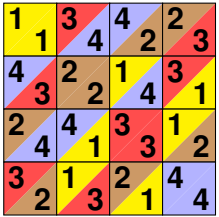
Mutually Orthogonal Latin Squares (MOLS)

Definition

A *Latin square* is a $n \times n$ matrix where all rows and columns are permutations of $[n] = \{1, \cdots, n\}$. Two Latin squares are *orthogonal* if their superposition yields all the pairs $(x, y) \in [n] \times [n]$.

1	3	4	2
4	2	1	3
2	4	3	1
3	1	2	4

1	4	2	3
3	2	4	1
4	1	3	2
2	3	1	4

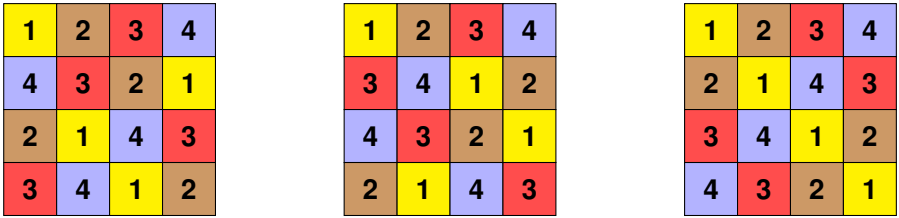


- n-**MOLS**: set of n pairwise orthogonal Latin squares [K15]

$(2,n)$-Schemes through n-MOLS - Setup

Setup Phase

1. The dealer D selects n-MOLS of order N and makes them public

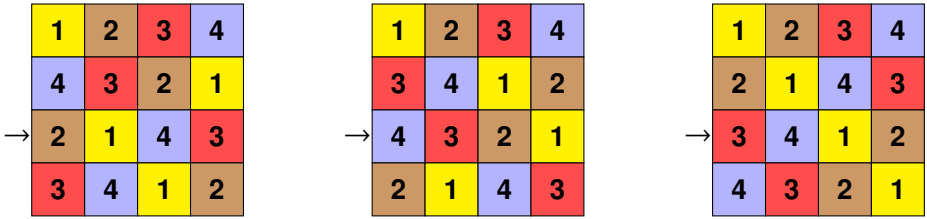


Example: $(2,3)$-scheme, $n = 3$, $N = 4$

$(2,n)$-Schemes through n-MOLS - Setup

Setup Phase

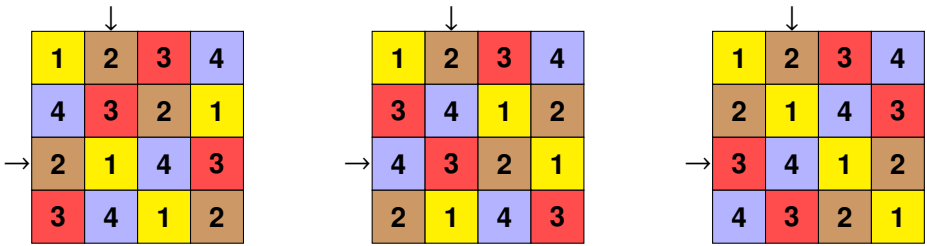
2. D encodes the secret as a row $S \in \{1, \cdots, N\}$



Example: $(2,3)$-scheme, $S = 3$

Setup Phase

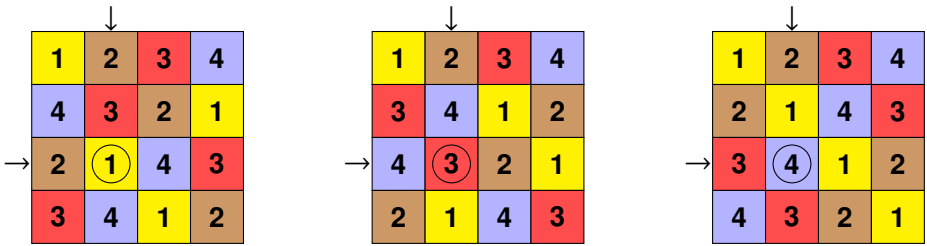
3. D randomly selects a column $j \in \{1, \cdots, N\}$



Example: $S = 3$, $j \leftarrow 2$

Setup Phase

4. The value of $L_i(S, j)$ for $i \in [N]$ is the share given to player P_i

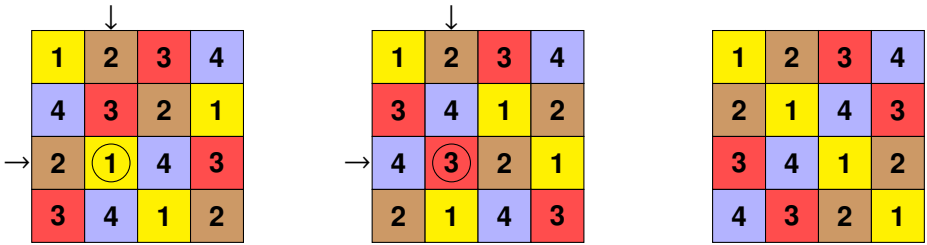


Example: $(2,3)$-scheme, $S = 3$, $j \leftarrow 2$, $B_1 = 1$, $B_2 = 3$, $B_3 = 4$

$(2,n)$-Schemes through n-MOLS - Recovery

Recovery Phase

1. Since L_i, L_k are orthogonal, (B_i, B_k) uniquely identify (S, j)

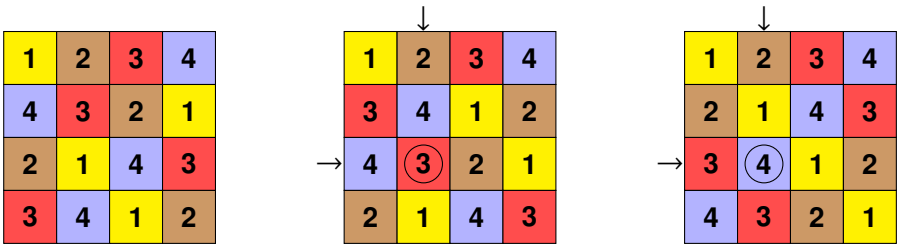


Example: $(2,3)$-scheme, $B_1 = 1$, $B_2 = 3 \Rightarrow (3,2)$

$(2,n)$-Schemes through n-MOLS - Recovery

Recovery Phase

1. Since L_i, L_k are orthogonal, (B_i, B_k) uniquely identify (S, j)

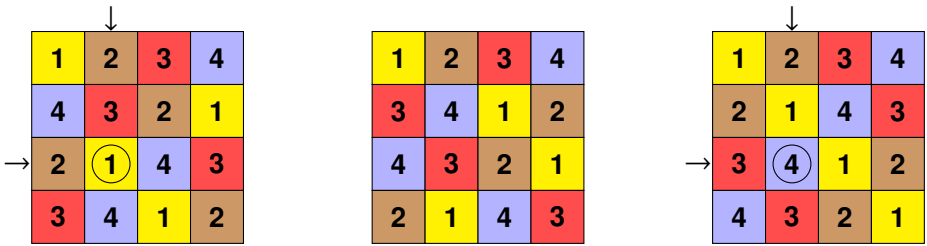


Example: $(2,3)$-scheme, $B_2 = 3$, $B_3 = 4 \Rightarrow (3,2)$

$(2,n)$-Schemes through n-MOLS - Recovery

Recovery Phase

1. Since L_i, L_k are orthogonal, (B_i, B_k) uniquely identify (S, j)

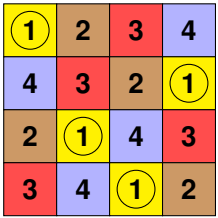


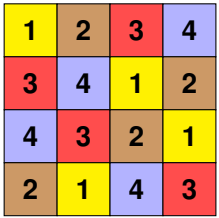
Example: $(2,3)$-scheme, $B_1 = 1$, $B_3 = 4 \Rightarrow (3,2)$

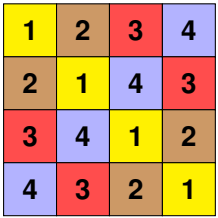
$(2,n)$-Schemes through n-MOLS - Security

Security

1. Knowledge of a single B_i leaves S completely undetermined





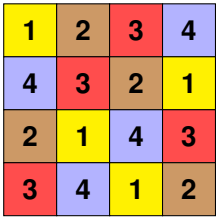


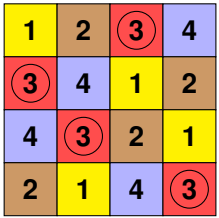
Example: $(2,3)$-scheme, $B_1 = 1, \Rightarrow S =$???

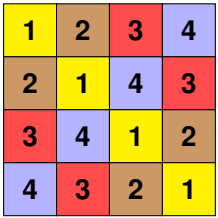
$(2,n)$-Schemes through n-MOLS - Security

Security

1. Knowledge of a single B_i leaves S completely undetermined





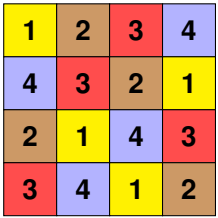


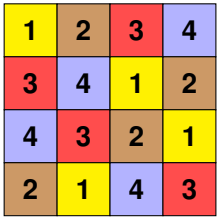
Example: $(2,3)$-scheme, $B_2 = 3$, $\Rightarrow S =$???

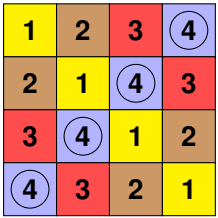
$(2,n)$-Schemes through n-MOLS - Security

Security

1. Knowledge of a single B_i leaves S completely undetermined



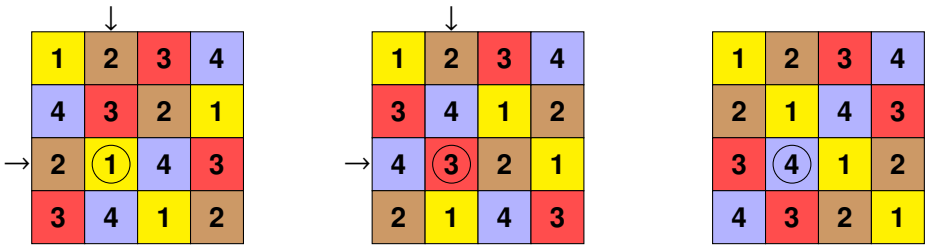




Example: $(2,3)$-scheme, $B_3 = 4$, $\Rightarrow S =$???

$(2,n)$-Schemes through n-MOLS - Anonymity?

- **Remark:** if P_i and P_j find S, they can determine the shares of other players!
- **So:** the scheme is not *anonymous*, i.e. recovery requires shares and a fixed ordering of players/Latin squares.



Example: $(2,3)$-scheme, $B_1 = 1$, $B_2 = 3 \Rightarrow B_3 = 4$

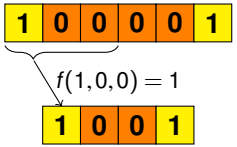
Anonymous SSS: recovery of the secret is a function *only* of the shares values [B97]

Cellular Automata & Latin Squares

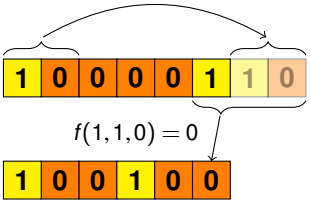
Cellular Automata

- One-dimensional Cellular Automaton (CA): a discrete parallel computation model composed of a finite array of n cells

Example: $n = 6$, $d = 3$, $f(s_i, s_{i+1}, s_{i+2}) = s_i \oplus s_{i+1} \oplus s_{i+2}$ (rule 150)



No Boundary CA – NBCA

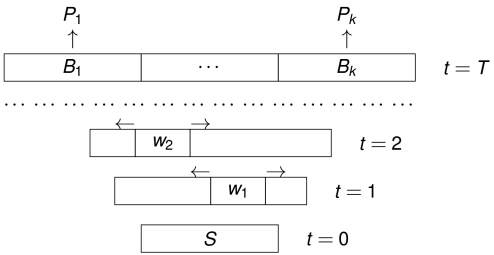


Periodic Boundary CA – PBCA

- Each cell updates its state $s \in \{0,1\}$ by applying a local rule $f : \{0,1\}^d \to \{0,1\}$ to itself and the $d-1$ cells on its right [M19, M26]

Sequential Threshold CA-Based SSS

- **Sequential threshold:** shares are *adjacent* blocks of CA preimages
- Any two adjacent shares "collapse" on a copy of the secret via CA forward evolution [M14]
- Does not allow two non-adjacent players to reconstruct the secret



Latin Squares through Bipermutive CA (1/2)

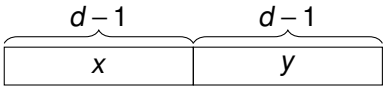
- Bipermutive CA: local rule f is defined as

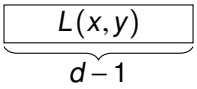
$$f(x_1, \cdots, x_d) = x_1 + \varphi(x_2, \cdots, x_{d-1}) + x_d$$

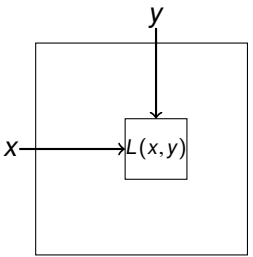
- $\varphi : \mathbb{F}_q^{d-2} \to \mathbb{F}_q$: generating function of f [L13]

Lemma ([MFL16])

A (no-boundary) CA $F : \mathbb{F}_q^{2(d-1)} \to \mathbb{F}_q^d$ with bipermutive rule $f : \mathbb{F}_q^d \to \mathbb{F}_q$ generates a Latin square of order $N = q^{d-1}$

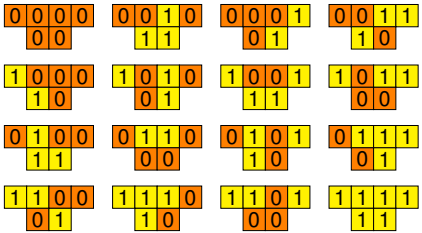






Latin Squares through Bipermutive CA (2/2)

- Example: CA $F : \mathbb{F}_2^4 \to \mathbb{F}_2^2$, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$ (Rule 150)
- Encoding: $00 \mapsto 1, 10 \mapsto 2, 01 \mapsto 3, 11 \mapsto 4$



(a) Rule 150 on 4 bits

1	4	3	2
2	3	4	1
4	1	2	3
3	2	1	4

(b) Latin square L_{150}

Linear CA

- Local rule: *linear combination* of the neighborhood cells

$$f(x_1, \cdots, x_d) = a_1 x_1 + \cdots + a_d x_d \ , \ a_i \in \mathbb{F}_q$$

- Associated polynomial:

$$f \mapsto p_f(X) = a_1 + a_2 X + \cdots + a_d X^{d-1}$$

- $(n-d+1) \times n$ transition matrix:

$$M_F = \begin{pmatrix} a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_1 & \cdots & a_d \end{pmatrix}, \quad x \mapsto M_F x^\top$$

- **Remark:** a linear rule is bipermutive iff $a_1, a_d \neq 0$

MOLS from Linear Bipermutive CA (LBCA)

Theorem ([M20])

A set of t linear bipermutive CA $F_1, \dots F_t : \mathbb{F}_q^{2(d-1)} \to \mathbb{F}_q^{d-1}$ generates a family of t-MOLS of order $N = q^{d-1}$ if and only if their associated polynomials are pairwise coprime

1	4	3	2
2	3	4	1
4	1	2	3
3	2	1	4

(a) Rule 150

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

(b) Rule 90

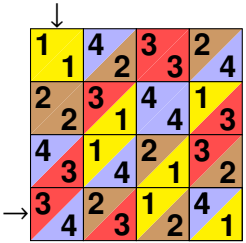
1,1	4,2	3,3	2,4
2,2	3,1	4,4	1,3
4,3	1,4	2,1	3,2
3,4	2,3	1,2	4,1

(c) Superposition

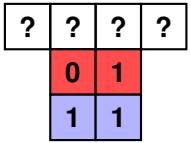
Figure: $P_{150}(X) = 1 + X + X^2$, $P_{90}(X) = 1 + X^2$ (coprime)

Recovery in $(2, n)$-schemes with CA-based Latin squares

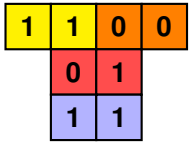
- **Input**: A pair $w, z \in \{0,1\}^{n-1}$ of output configurations
- **Output**: The unique preimage x generating w, z under the action of the two CA



(a) rule 90-150



(b) Input



(c) Output

- Use the **coupled de Bruijn graph** to compute the preimage [M18]

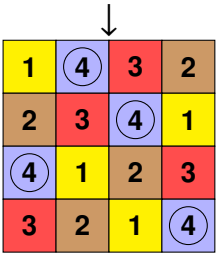
Anonymous $(2, n)$-Scheme with CA

Main Idea: Swap the roles of the Latin squares and configurations: the specific Latin square is the secret, its preimages are the shares

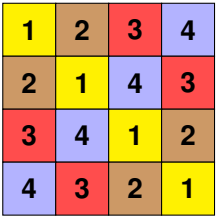
Anonymous CA-based Scheme - Setup

Setup Phase

1. The dealer D selects the secret as one of the MOLS, $S \in \{90, 150\}$, and a random output configuration $R \leftarrow \mathbb{F}_2^{d-1}$



(a) Rule 150



(b) Rule 90

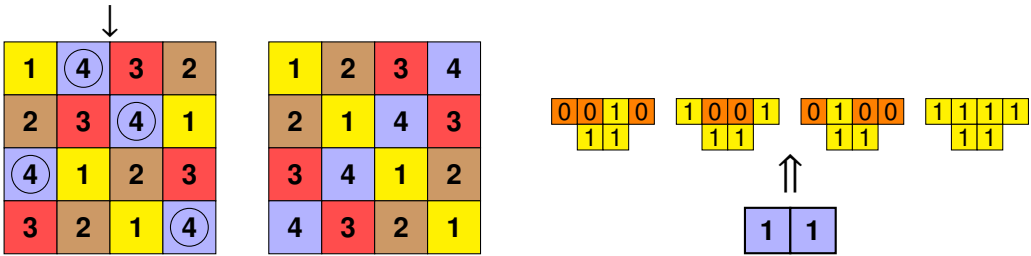
(c) Preimage computation

Example: $S = 150$, $R = 11$ (mapped to 4)

Anonymous CA-based Scheme - Setup

Setup Phase

2. D computes the *preimage set* $F_S^{-1}(R) = \{x \in \mathbb{F}_2^{2(d-1)} : F_S(x) = R\}$.



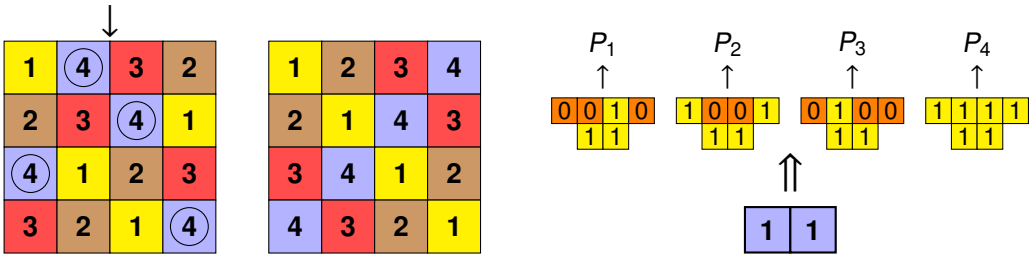
(a) Rule 150 (b) Rule 90 (c) Preimage computation

Example: $S = 150$, $R = 11$ (mapped to 4)

Anonymous CA-based Scheme - Setup

Setup Phase

3. Each player P_i receives a preimage $x \in F_S^{-1}(R)$ as a share



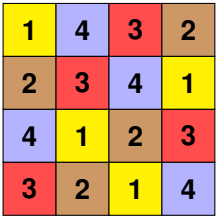
(a) Rule 150 (b) Rule 90 (c) Preimage computation

Example: $S = 150$, $R = 11$ (4). $P_1 \leftarrow 0010$ (2), $P_2 \leftarrow 1001$ (7), $P_3 \leftarrow 0100$ (9), $P_4 \leftarrow 1111$ (16)

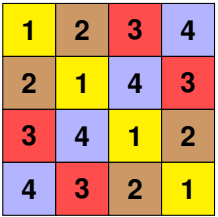
Anonymous CA-based Scheme - Recovery

Recovery Phase

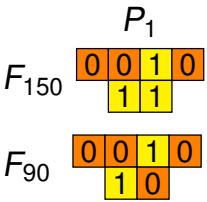
1. Player P_i and P_j compute the output configurations starting from their preimage, for each CA rule

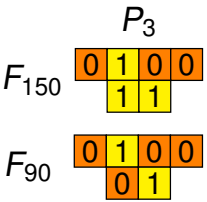


(a) Rule 150



(b) Rule 90





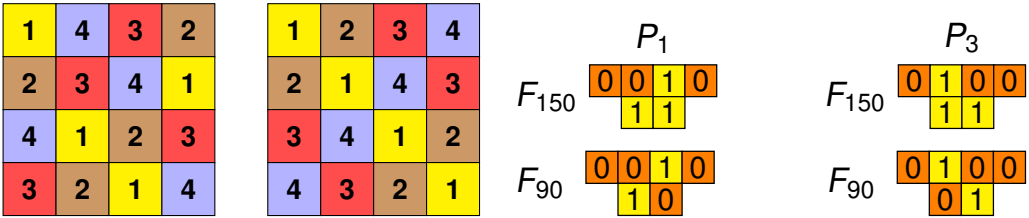
(c) Preimage computation

Example: $S = 150$, $R = 11$ (4). $P_1 \leftarrow 0010$ (2), $P_3 \leftarrow 0100$ (9)

Anonymous CA-based Scheme - Recovery

Recovery Phase

2. Player P_i and P_j compute the preimage sets $\mathcal{A}_i$, $\mathcal{A}_j$ of the output configurations they obtained



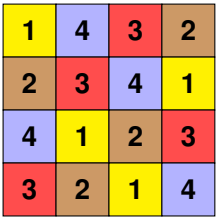
(a) Rule 150 (b) Rule 90 (c) Preimage computation

Example: $\mathcal{A}_1 = \{\{2,7,9,16\},\{2,5,12,15\}\}$, $\mathcal{A}_3 = \{\{2,7,9,16\},\{3,8,9,14\}\}$.

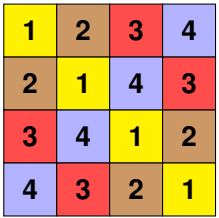
Anonymous CA-based Scheme - Recovery

Recovery Phase

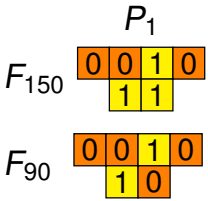
3. Player P_i and P_j compute the (private) intersection of their preimage sets $\mathcal{A}_i \cap \mathcal{A}_j$, which uniquely identifies the secret rule used

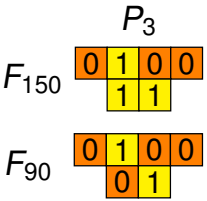


(a) Rule 150



(b) Rule 90





(c) Preimage computation

$$\mathcal{A}_1 \cap \mathcal{A}_3 = \{\{2,7,9,16\},\{2,5,12,15\}\} \cap \{\{2,7,9,16\},\{3,8,9,14\}\} = \{\{2,7,9,16\}\}$$

$$\Rightarrow S = 150$$

Anonymity - Parallel Class Decomposition [K15]

Theorem

Each pair of preimages uniquely identifies a single family of preimages in the parallel class decomposition of the CA

1	4	3	2
2	3	4	1
4	1	2	3
3	2	1	4

(a) Rule 150

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

(b) Rule 90

y	Π_{90}	Π_{150}
00 (1)	$\{1,6,11,16\}$	$\{1,8,10,15\}$
10 (2)	$\{2,5,12,15\}$	$\{4,5,11,14\}$
01 (3)	$\{3,8,9,14\}$	$\{3,6,12,13\}$
11 (4)	$\{4,7,10,13\}$	$\{2,7,9,16\}$

(c) Π-class decomposition

$$\mathcal{A}_1 \cap \mathcal{A}_3 = \{\{2,7,9,16\},\{2,5,12,15\}\} \cap \{\{2,7,9,16\},\{3,8,9,14\}\} = \{\{2,7,9,16\}\}$$

$$\Rightarrow S = 150$$

Conclusions

Conclusions

Summing up:

- The new scheme uses CA rules as *secrets*, and preimages as *shares*
- Anonymity stands on the parallel class decomposition of the MOLS family generated by the CA [K15]
- Max number of players: 2^{d-1}, max number of secrets: max number of mutually orthogonal CA of diameter d [M20]

Open questions:

- **Complexity:** each player has to compute $n \cdot 2^{d-1}$ preimages
- **Recovery:** how to compute efficiently a *private* intersection of the preimage sets?

References

[B97] C. Blundo, D.R. Stinson: Anonymous secret sharing schemes. Discret. Appl. Math. 77(1):13–28 (1997)

[K15] A.D. Keedwell, J. Dénes: Latin squares and their applications. Elsevier (2015)

[L13] A. Leporati and L. Mariot: 1-Resiliency of bipermutive cellular automata rules. In: Proceedings of AUTOMATA 2013, pp. 110–123 (2013)

[M26] L. Manzoni, L. Mariot, G. Menara: Combinatorial designs and cellular automata: A survey. Discret. Appl. Math. 379:656–674 (2026)

[M20] L. Mariot, M. Gadouleau, M., E. Formenti, A. Leporati: Mutually orthogonal Latin squares based on cellular automata. Des. Codes Cryptogr. 88(2):391–411 (2020)

[M19] L. Mariot, S. Picek, A. Leporati, and D. Jakobovic. Cellular automata based S-boxes. Cryptography and Communications 11(1):41–62 (2019)

[M18] L. Mariot, A. Leporati: Inversion of Mutually Orthogonal Cellular Automata. In: Proc. of ACRI 2018, pp. 364–376 (2018)

[M16] L. Mariot, E. Formenti, A. Leporati: Constructing Orthogonal Latin Squares from Linear Cellular Automata. In: Exploratory papers of AUTOMATA 2016 (2016)

[M14] L. Mariot, A. Leporati: Sharing Secrets by Computing Preimages of Bipermutive Cellular Automata. In: Proc. of ACRI 2014, pp. 417–426 (2014)

[S79] A. Shamir: How to share a secret. Commun. ACM 22(11):612–613 (1979)

[S04] D.R. Stinson: Combinatorial designs - constructions and analysis. Springer (2004)