

# Sharing Secrets by Computing Preimages of Bipermutive Cellular Automata

ACRI 2014 - September 22-25 - Krakow

Luca Mariot, Alberto Leporati

Dipartimento di Informatica, Sistemistica e Comunicazione  
Università degli Studi Milano - Bicocca  
[l.mariot@campus.unimib.it](mailto:l.mariot@campus.unimib.it), [alberto.leporati@unimib.it](mailto:alberto.leporati@unimib.it)

September 25, 2014

# Outline

Cellular Automata and Secret Sharing Schemes

Building Preimages of Bipermutive CAs

A New  $(k, k)$  Scheme Based on Bipermutive CAs

An Extension to the Basic Scheme

Conclusions and Future Developments

# Outline

## Cellular Automata and Secret Sharing Schemes

Building Preimages of Bipermutive CAs

A New  $(k, k)$  Scheme Based on Bipermutive CAs

An Extension to the Basic Scheme

Conclusions and Future Developments

# One-Dimensional Cellular Automata

## Definition

A **finite boolean one-dimensional cellular automaton** (CA) is a triple  $\langle n, r, f \rangle$  where  $n \in \mathbb{N}$  is the number of cells,  $r \in \mathbb{N}$  is the radius and  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  is a boolean function specifying the CA local rule.

- ▶ During a single time step, a cell  $i$  updates its boolean state  $c_i$  in parallel by computing  $f(c_{i-r}, \dots, c_i, \dots, c_{i+r})$
- ▶ **No Boundary CA:** only the central cells  $i \in \{r+1, \dots, n-r\}$  update their states; the array shrinks by  $2r$  cells at each time step

## Secret Sharing Schemes: Basic Definitions

- ▶ A **secret sharing scheme** is a procedure which enables a **dealer** to share a **secret**  $S$  among a set  $\mathcal{P}$  of **players**, in such a way that only some **authorized subsets** can recover  $S$
- ▶ An **access structure**  $\Gamma \subseteq 2^{\mathcal{P}}$  specifies the authorized subsets
- ▶ In  $(k, n)$  **threshold schemes**, the access structure  $\Gamma$  contains all those subsets of at least  $k$  players
- ▶ Shamir's scheme [Shamir79], which is based on polynomial interpolation, is an example of  $(k, n)$  threshold scheme
- ▶ The CA-based scheme proposed in [Rey05] features a **sequential**  $(k, n)$  threshold scheme

## Perfect and Ideal Secret Sharing Schemes

- ▶ Let us assume that a probability distribution  $Pr(S)$  is defined on the space of the secrets, and that  $\delta_U$  represents a shares distribution to an unauthorized subset  $U \notin \Gamma$
- ▶ A secret sharing scheme is **perfect** if for all unauthorized subsets  $U \notin \Gamma$  and for all shares distributions  $\delta_U$  it results that

$$Pr(S|\delta_U) = Pr(S)$$

- ▶ A secret sharing scheme is called **ideal** if the size of each share equals the size of the secret

# Outline

Cellular Automata and Secret Sharing Schemes

**Building Preimages of Bipermutive CAs**

A New  $(k, k)$  Scheme Based on Bipermutive CAs

An Extension to the Basic Scheme

Conclusions and Future Developments

## Permutive and Bipermutive Rules

Rule  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  is called:

- ▶ **leftmost permutive** if there exists  $g_L : \mathbb{F}_2^{2r} \rightarrow \mathbb{F}_2$  such that:

$$f(x_1, x_2, \dots, x_{2r+1}) = x_1 \oplus g_L(x_2, \dots, x_{2r+1})$$

- ▶ **rightmost permutive** if there exists  $g_R : \mathbb{F}_2^{2r} \rightarrow \mathbb{F}_2$  such that:

$$f(x_1, \dots, x_{2r}, x_{2r+1}) = g_R(x_1, \dots, x_{2r}) \oplus x_{2r+1}$$

- ▶ **bipermutive** if there exists  $g : \mathbb{F}_2^{2r-1} \rightarrow \mathbb{F}_2$  such that:

$$f(x_1, x_2, \dots, x_{2r}, x_{2r+1}) = x_1 \oplus g(x_2, \dots, x_{2r}) \oplus x_{2r+1}$$



## Building Preimages of (Bi)Permutive CAs [Gutowitz93] (1/6)

Given a **rightmost permutive** rule  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  and a configuration  $c \in \mathbb{F}_2^m$ , a preimage  $p \in \mathbb{F}_2^{m+2r}$  of  $c$  can be computed as follows:

1. Set the **leftmost**  $2r$  cells  $p_1, \dots, p_{2r}$  of the preimage  $p$  to random values

$$p = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & ? & ? & ? & ? & ? & ? \\ \hline \end{array}$$
$$c = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

Figure : Example of preimage construction under rule 30 (R-permutive)

## Building Preimages of (Bi)Permutive CAs [Gutowitz93] (2/6)

Given a **rightmost permutive** rule  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  and a configuration  $c \in \mathbb{F}_2^m$ , a preimage  $p \in \mathbb{F}_2^{m+2r}$  of  $c$  can be computed as follows:

- By right permutivity,  $c_1 = g_R(p_1, \dots, p_{2r}) \oplus p_{2r+1}$ . Hence,  $p_{2r+1}$  can be computed as  $p_{2r+1} = g_R(p_1, \dots, p_{2r}) \oplus c_1$

$$\begin{array}{r}
 p = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & ? & ? & ? & ? & ? & ? \\ \hline \end{array} \\
 c = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}
 \end{array}$$

Figure : Example of preimage construction under rule 30 (R-permutive)

## Building Preimages of (Bi)Permutive CAs [Gutowitz93] (3/6)

Given a **rightmost permutive** rule  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  and a configuration  $c \in \mathbb{F}_2^m$ , a preimage  $p \in \mathbb{F}_2^{m+2r}$  of  $c$  can be computed as follows:

- Shift the  $2r$ -bit window one place to the right and compute

$$p_{2r+2} = g_R(p_2, \dots, p_{2r+1}) \oplus c_2$$

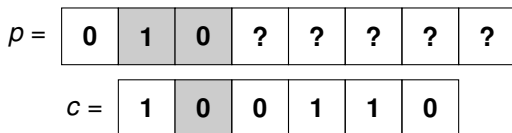


Figure : Example of preimage construction under rule 30 (R-permutive)

## Building Preimages of (Bi)Permutive CAs [Gutowitz93] (4/6)

Given a **rightmost permutive** rule  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  and a configuration  $c \in \mathbb{F}_2^m$ , a preimage  $p \in \mathbb{F}_2^{m+2r}$  of  $c$  can be computed as follows:

- Continue to apply Step 3 until the rightmost bit in the preimage has been computed

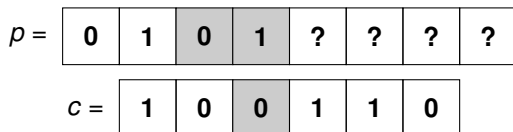


Figure : Example of preimage construction under rule 30 (R-permutive)

## Building Preimages of (Bi)Permutive CAs [Gutowitz93] (5/6)

Given a **rightmost permutive** rule  $f : \mathbb{F}_2^{2r+1} \rightarrow \mathbb{F}_2$  and a configuration  $c \in \mathbb{F}_2^m$ , a preimage  $p \in \mathbb{F}_2^{m+2r}$  of  $c$  can be computed as follows:

- Continue to apply Step 3 until the rightmost bit in the preimage has been computed

$$p = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$
$$c = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

Figure : Example of preimage construction under rule 30 (R-permutive)

## Building Preimages of (Bi)Permutive CAs [Gutowitz93] (6/6)

- ▶ For leftmost permutive rules, a symmetrical result holds by starting from the right and completing leftwards
- ▶ Each image in a rightmost (leftmost) permutive CA has thus  $2^{2r}$  preimages
- ▶ If  $f$  is **bipermutive**, the initial block can be placed at any position [Oliveira04]. This possibility does not increase the number of preimages

$$p = \begin{array}{|c|c|c|c|c|c|c|c|} \hline ? & ? & ? & ? & 0 & 1 & ? & ? \\ \hline \end{array}$$

$$c = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

(a) Initialization

$$p = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

$$c = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

(b) Complete preimage

Figure : Example with bipermutive rule 150

# Outline

Cellular Automata and Secret Sharing Schemes

Building Preimages of Bipermutive CAs

**A New  $(k, k)$  Scheme Based on Bipermutive CAs**

An Extension to the Basic Scheme

Conclusions and Future Developments

## Observations on Preimage Computation

- ▶ By iterating the procedure of preimage computation, at each step the size of the preimage grows by  $2r$  cells
- ▶ In particular, starting from a CA configuration  $c$  of length  $m$ , after  $t$  steps the resulting preimage will have length  $L(t) = 2rt + m$
- ▶ Hence, given  $k \in \mathbb{N}$ , the number of iterations  $t$  necessary to get a preimage of length  $k \cdot m$  is:

$$t = \frac{m(k-1)}{2r}$$

- ▶ Since  $t$  is integer, it means that  $2r$  must divide  $m(k-1)$
- ▶ **Additional security requirement:**  $2r \mid m$



## Basic $(k, k)$ Secret Sharing Scheme - Setup Phase (1/5)

Assuming that there are  $k$  players  $P_1, P_2, \dots, P_k$ :

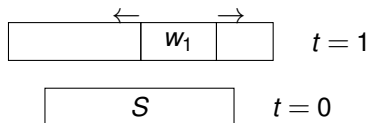
1. The *dealer*  $D$  sets the secret  $S$  as an  $m$ -bit configuration of a CA, and randomly selects a bipermutive rule of radius  $r$ , where  $r$  is such that  $2r \mid m$

S      $t = 0$

## Basic $(k, k)$ Secret Sharing Scheme - Setup Phase (2/5)

Assuming that there are  $k$  players  $P_1, P_2, \dots, P_k$ :

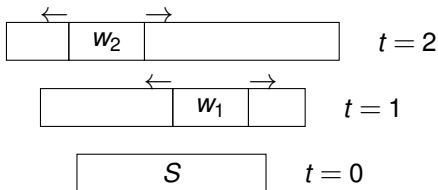
2.  $D$  evolves the CA backwards for  $T = m(k-1)/2r$  iterations, randomly choosing at each step the value and the position of the initial  $2r$ -bit block



## Basic $(k, k)$ Secret Sharing Scheme - Setup Phase (3/5)

Assuming that there are  $k$  players  $P_1, P_2, \dots, P_k$ :

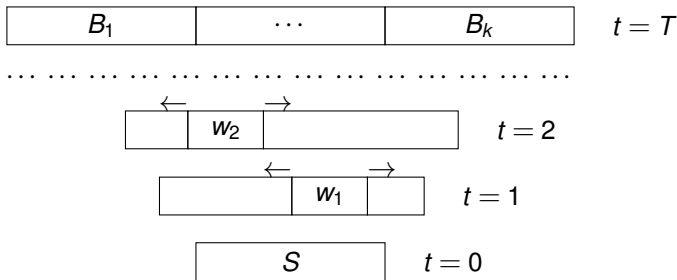
- $D$  evolves the CA backwards for  $T = m(k-1)/2r$  iterations, randomly choosing at each step the value and the position of the initial  $2r$ -bit block



## Basic $(k, k)$ Secret Sharing Scheme - Setup Phase (4/5)

Assuming that there are  $k$  players  $P_1, P_2, \dots, P_k$ :

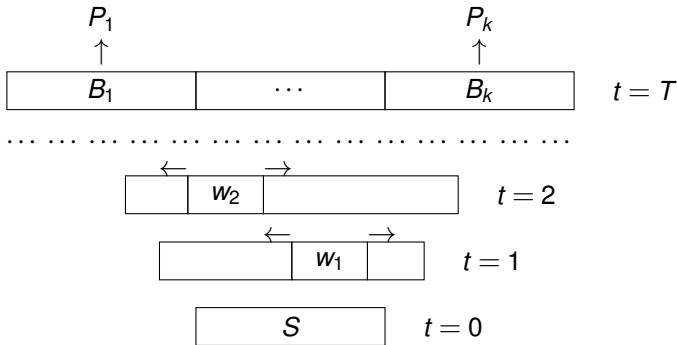
- After  $T = m(k - 1)/2r$  iterations, the dealer splits the resulting preimage in  $k$  blocks of  $m$  bits



## Basic $(k, k)$ Secret Sharing Scheme - Setup Phase (5/5)

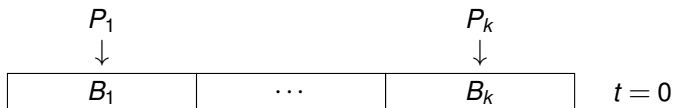
Assuming that there are  $k$  players  $P_1, P_2, \dots, P_k$ :

4. Finally,  $D$  securely sends one block to each player and publishes the bipermutive rule used to evolve the CA backwards



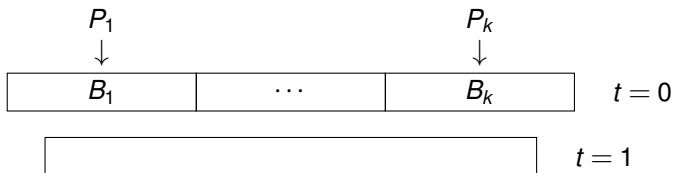
## Basic $(k, k)$ Secret Sharing Scheme - Recovery Phase (1/4)

1. Using a pre-established protocol, the  $k$  players pool their shares in the correct order to get the complete preimage of the CA



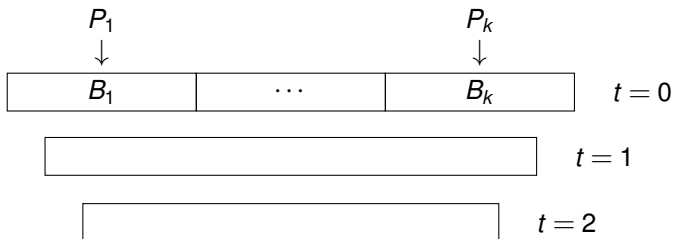
## Basic $(k, k)$ Secret Sharing Scheme - Recovery Phase (2/4)

- The players evolve the CA forward, using the local rule published by the dealer



## Basic $(k, k)$ Secret Sharing Scheme - Recovery Phase (3/4)

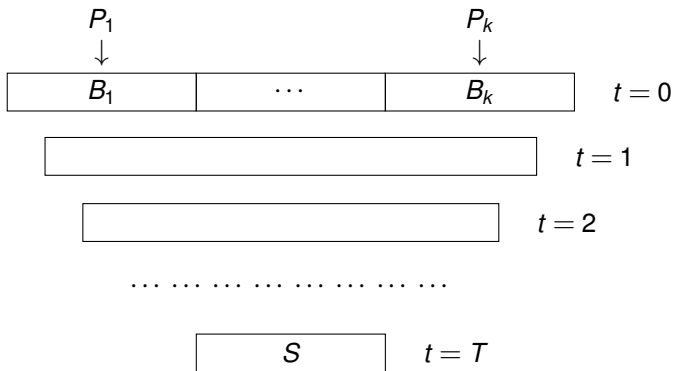
- The players evolve the CA forward, using the local rule published by the dealer





## Basic $(k, k)$ Secret Sharing Scheme - Recovery Phase (4/4)

3. The configuration obtained after  $T = m(k - 1)/2r$  iterations is the secret  $S$ . Notice that the players can compute  $T$  by themselves



## Security Properties of the Basic Scheme

### Lemma

Let  $B_l$ , with  $1 \leq l \leq k$ , be the only unknown share among  $B_1, \dots, B_k$ . Then, under the condition that  $2r \mid m$ , there exists a permutation  $\Pi : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^m$  between  $B_l$  and the secret  $S$ .

From the previous Lemma, the following result holds:

### Theorem

Suppose that the secret  $S$  and the  $2r$ -bit blocks in the setup phase are chosen *uniformly* at random. Then, the basic  $(k, k)$  scheme is perfect

Moreover, the basic scheme is also ideal, since each share is a block of  $m$  bits, as the secret

# Outline

Cellular Automata and Secret Sharing Schemes

Building Preimages of Bipermutive CAs

A New  $(k, k)$  Scheme Based on Bipermutive CAs

**An Extension to the Basic Scheme**

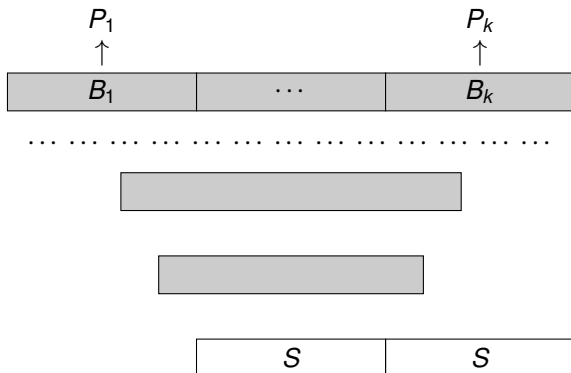
Conclusions and Future Developments

## Considerations on the Basic scheme

- ▶ The basic scheme can be used to implement any access structure  $\Gamma \subseteq 2^P$ : simply re-run the setup phase for each authorized subset  $A \in \Gamma$
- ▶ However, as the number of participants grows, the scheme turns out to be impractical, since each player must hold a different share for each authorized subset he belongs to
- ▶ Necessity to find an extended scheme which allows the players to reuse the same shares
- ▶ Suppose that a set of  $k$  shares has been distributed to  $k$  players using the basic setup phase. The scheme can be extended using **secret juxtaposition**

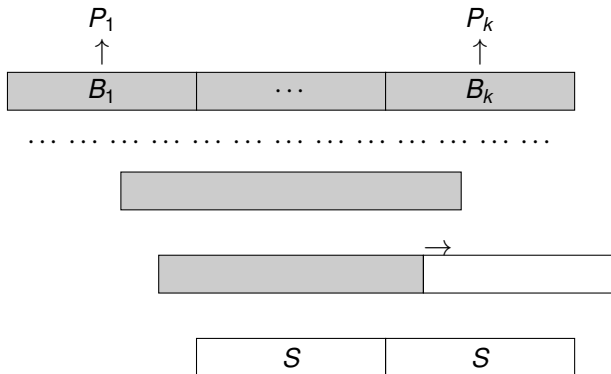
## Secret Juxtaposition (1/4)

1. Append a copy of the secret  $S$  to the right of the final CA image



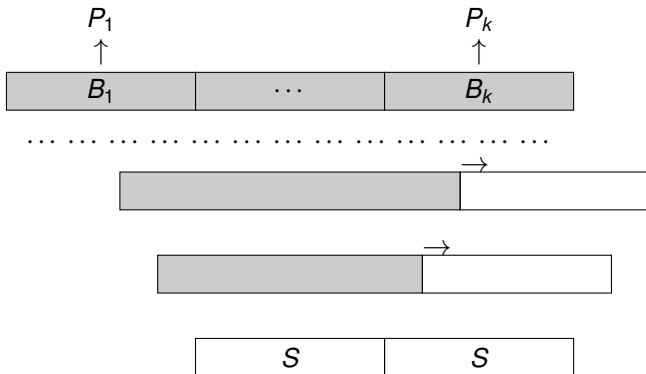
## Secret Juxtaposition (2/4)

- Update the preimages by completing them rightwards (note that it is not necessary to pick extra random bits)



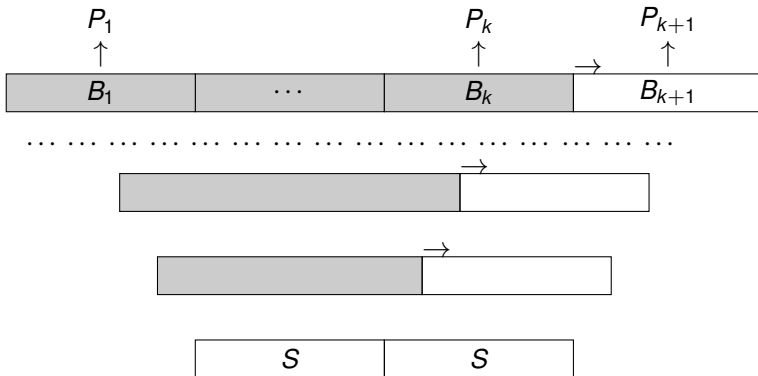
## Secret Juxtaposition (3/4)

- Update the preimages by completing them rightwards (note that it is not necessary to pick extra random bits)



## Secret Juxtaposition (4/4)

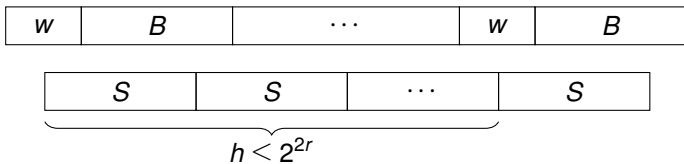
3. The last preimage contains an additional block for the new player.  
 The sets  $\{P_1, \dots, P_k\}$  and  $\{P_2, \dots, P_{k+1}\}$  can recover  $S$





## Access Structure of the Extended Scheme

- ▶ The extended scheme implements a  $(k, n)$ -**sequential threshold** access structure: at least  $k$  **consecutive** shares are necessary to recover the secret
- ▶ In particular, if we continue to append copies of the secret, the final shares will eventually repeat. Thus, the access structure becomes **cyclic**



**Figure :** After at most  $h \leq 2^{2r}$  juxtaposed copies of  $S$ , by completing rightwards the  $2r$ -bit block  $w$  will repeat at the end of the preimage.

# Outline

Cellular Automata and Secret Sharing Schemes

Building Preimages of Bipermutive CAs

A New  $(k, k)$  Scheme Based on Bipermutive CAs

An Extension to the Basic Scheme

Conclusions and Future Developments

## Conclusions

- ▶ We showed how the surjectivity of bipermutve CAs can be employed to design a basic secret sharing scheme where all the players are required in order to recover the secret  $S$
- ▶ This basic scheme can be proved to be both perfect and ideal
- ▶ The secret juxtaposition method allows to extend the basic scheme with a cyclic access structure

## Future Developments

- ▶ Find a general method to compute after how many juxtapositions of the secret the shares begin to repeat themselves. This is equivalent to the following open problem:

### Open Problem (PCAP - Periods of CA Preimages)

*Given a bipermutive CA and a spatially periodic configuration  $c \in A^{\mathbb{Z}}$  with period  $m$ , find the periods of its preimages*

- ▶ Recent investigation indicates that PCAP can be completely solved in the case of **additive** bipermutive CAs
- ▶ Other improvements: investigate possible applications of the scheme to **secure multiparty computation protocols**, and extend the scheme to  $d$ -dimensional CAs with  $d > 1$

# Thanks for your attention!

## References



del Rey, Á.M., Mateus, J.P., Sánchez, G.R.: A secret sharing scheme based on cellular automata. *Appl. Math. Comput.* 170(2), 1356–1364 (2005)



Gutowitz, H.: Cryptography with dynamical systems. In: Goles, E., Boccaro, N. (eds.) *Cellular Automata and Cooperative Phenomena*. pp. 237–274. Kluwer Academic Press (1993)



Oliveira, G., Coelho, A., Monteiro, L.: Cellular automata cryptographic model based on bi-directional toggle rules. *Int. J. Mod. Phys. C* 15(8), 1061–1068 (2004)



Shamir, A.: How to share a secret. *Commun. ACM* 22(11), 612–613 (1979)