Sharing Secrets by Computing Preimages of Bipermutive Cellular Automata

Luca Mariot and Alberto Leporati

Dipartimento di Informatica, Sistemistica e Comunicazione,
Università degli Studi Milano, Bicocca,
Viale Sarca 336/14, 20124 Milano, Italy
l.mariot@campus.unimib.it, alberto.leporati@unimib.it

Abstract. A secret sharing scheme based on one-dimensional bipermutive cellular automata is discussed in this paper. The underlying idea is to represent the secret as a configuration of a bipermutive CA and to iteratively apply a preimage computation algorithm until a sufficiently long configuration to be splitted among the participants is obtained. The scheme is proved to be both perfect and ideal, and a simple extension is shown to induce a sequential access structure which eventually becomes cyclic, where the upper bound on the length of the cycles depends on the radius of the adopted local rule.

Keywords: Cellular automata, cryptography, secret sharing schemes, bipermutivity, preimage computation, cyclic access structure.

1 Introduction

Secret sharing schemes (SSS) were originally introduced by Shamir [8] and Blakley [1] as a method to securely share a *secret* among a set $\mathcal{P}$ of n participants, in such a way that only the members belonging to some *authorized subsets* of $\mathcal{P}$, specified through an *access structure*, can recover the secret by pooling their shares.

During the last few years some SSS based on *cellular automata* (CA) have been proposed in the literature, the first of which can be traced back to del Rey, Mateus and Sánchez [2]. Specifically, the scheme described in [2] exploits the reversibility of *linear memory cellular automata* (LMCA). The secret is represented as one of the k initial conditions in a k-th order LMCA which is then evolved for n iterations. Each player then receives one of the n resulting CA configurations as a share. The access structure generated by this scheme can be defined as a (k,n) *sequential threshold*, since at least k *consecutive* shares are required in order to evolve backwards the LMCA and recover the secret, meaning that there are in total $n-k+1$ minimal authorized subsets. Most of the later CA-based SSS [6,4,3] use the same LMCA principle of del Rey, Mateus and Sánchez's scheme, and thus feature similar access structures.

In this paper, we propose a secret sharing scheme designed upon a different CA primitive, namely *bipermutive* cellular automata (BCA), which is less complex (since BCA are memoryless) and which generates a more flexible access structure than LMCA-based schemes. We initially show a basic version of our scheme where all participants are required to combine their shares to recover the secret, which is set by the dealer as

J. Wąs, G.C. Sirakoulis, and S. Bandini (Eds.): ACRI 2014, LNCS 8751, pp. 417–426, 2014.

an m-bit configuration of a one-dimensional BCA. The automaton is then evolved backwards by iteratively applying a preimage computation algorithm until a configuration of length $k \cdot m$ is obtained, which is finally splitted among the k players. To recover the secret, the players just have to juxtapose their shares and evolve the CA forwards.

We prove that the scheme is *perfect*, meaning that an attacker knowing fewer than k shares cannot determine anything about the secret in an information-theoretic sense, and *ideal*, since the size of the shares equals the size of the secret. We finally introduce an extension to the scheme, called *secret juxtaposition*, which induces a (k,n) *sequential* threshold access structure that eventually becomes *cyclic*, thus yielding n minimal authorized subsets where n is bounded by 2^{2r}, being r the radius of the local rule.

The rest of the paper is structured as follows. Section 2 covers the basic notions and terminology about cellular automata and secret sharing schemes used throughout the paper. Section 3 shows the algorithm PREIMAGE-CONSTRUCTION, used to compute the preimage of a CA configuration under a bipermutive rule. Section 4 describes the basic version of our SSS, where all the k shares are required to recover the secret. Section 5 analyses the security properties of the basic scheme, proving that it is both perfect and ideal. In Section 6, the extended scheme is introduced and shown to generate an eventually cyclic access structure. Finally, Section 7 recaps the key features of the proposed scheme and its advantages over del Rey, Mateus and Sánchez's scheme, and sketches some possible developments for future research.

2 Preliminary Definitions

2.1 Cellular Automata

In this work we focus on the model of *one-dimensional finite boolean cellular automata*, which we define as a triple $\langle C, r, f \rangle$ where C is a finite one-dimensional array of *cells*, $r \in \mathbb{N}$ is the *radius* and $f : \mathbb{F}_2^{2r+1} \to \mathbb{F}_2$ is a boolean function specifying the *local rule*. We denote by $|C| = m$ the size of the array. During a single time step, each of the central cells $i \in \{r+1, \cdots, m-r\}$ in C updates its binary *state* by computing in parallel the local rule f on the neighbourhood formed by itself, the r cells at its left and the r cells at its right. We do not deal with any *boundary condition*, since the leftmost and rightmost r cells are not updated. Consequently, the *global rule* of a CA can be considered as a vectorial boolean function $F : \mathbb{F}_2^m \to \mathbb{F}_2^{m-2r}$, and thus the size of the cell array shrinks by $2r$ cells from one time step to the next. Clearly, this means that the global rule can be applied only a limited number of times, as long as $m \geq 2r+1$.

Given the radius r, there exist $2^{2^{2r+1}}$ local rules. Each rule f can be compactly indexed by its corresponding *Wolfram code*, which is the decimal representation of the truth table of f. A local rule $f : \mathbb{F}_2^{2r+1} \to \mathbb{F}_2$ is *leftmost permutive* if there exists a *generating function* $g_L : \mathbb{F}_2^{2r} \to \mathbb{F}_2$ such that $f(x) = x_1 \oplus g_L(x_2, \cdots, x_{2r+1})$ for all $x = (x_1, \cdots, x_{2r+1}) \in \mathbb{F}_2^{2r+1}$. Similarly, f is called *rightmost permutive* if there is $g_R : \mathbb{F}_2^{2r} \to \mathbb{F}_2$ such that $f(x) = g_R(x_1, \cdots, x_{2r}) \oplus x_{2r+1}$. Rule f is called *bipermutive* if it is both leftmost and rightmost permutive. In this case, g_L is itself rightmost permutive with a certain generating function $g : \mathbb{F}_2^{2r-1} \to \mathbb{F}_2$ (equivalently, g_R is leftmost permutive with the same g). Hence, f can be written as $f(x) = x_1 \oplus g(x_2, \cdots, x_{2r}) \oplus x_{2r+1}$.

2.2 Secret Sharing Schemes

Generally speaking, a *secret sharing scheme* is a procedure which enables a *dealer D* to share a *secret S* (for instance, a cryptographic key) among a set $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ of participants or *players*. Each player P_i receives a *share* B_i from the dealer, and a subset $A \subseteq \mathcal{P}$ is *authorized* if its members can reconstruct S by combining their shares. Authorized subsets are specified through an *access structure* $\Gamma \subseteq 2^{\mathcal{P}}$. Usually, Γ is required to be *monotone*, that is, if $A_1 \in \Gamma$ and $A_1 \subseteq A_2$ then $A_2 \in \Gamma$. An authorized subset $M \in \Gamma$ is called *minimal* if $N \notin \Gamma$ for all $N \subset M$. A monotone access structure Γ can thus be defined as the union-closure of the *basis* Γ_0, which is the family of all minimal authorized subsets. In (k,n) *threshold schemes*, such as Shamir's scheme [8], the basis is defined as $\Gamma_0 = \{A \subseteq \mathcal{P} : |A| = k\}$, meaning that all subsets of k or more players can recover the secret. The CA-based secret sharing scheme proposed by del Rey, Mateus and Sánchez [2] can be defined as a *sequential* (k,n) threshold scheme, since at least k *consecutive* shares are required to recover the secret. The minimal authorized subsets are thus of the form $A = \{P_i, P_{i+1}, \cdots, P_{i+k-1}\}$, with $1 \leq i \leq n-k+1$.

The security model adopted for the study of secret sharing schemes considers the information an attacker can obtain about the secret S by having the shares of a generic unauthorized subset. In particular, schemes which do not leak any information on S by knowing the shares of any unauthorized subset $U \notin \Gamma$ are called *perfect*. To formalise this notion in a probabilistic framework, we follow the approach laid out by Stinson [9].

Let $\mathcal{S}$ be the space of secrets and $\mathcal{B}$ the space of possible shares. We define a *distribution rule* as a function $\varphi : \mathcal{P} \to \mathcal{B}$ which assigns to each player in $\mathcal{P}$ a share from $\mathcal{B}$. Given a secret $S \in \mathcal{S}$, the set $\mathcal{F}_S$ denotes the family of all distribution rules induced by S. The dealer selects both the secret and the corresponding distribution rule according to two probability distributions, which we respectively denote by $Pr(S)$ and $Pr(\varphi)$. These probability distributions and the set $\mathcal{F} = \bigcup_{S \in \mathcal{S}} \mathcal{F}_S$ are assumed to be public, hence known to an attacker. Considering a generic subset of players $G \subseteq \mathcal{P}$, a *shares distribution* δ_G is a possible assignment of shares to the members of G. Given a distribution rule φ, the corresponding shares distribution δ_G is thus the image of the restriction $\varphi|_G$. By $\mathcal{B}_G(S) = \{\varphi|_G : \varphi \in \mathcal{F}_S\}$ we denote the set of all possible shares distributions to G induced by the secret S. The probability distribution on all possible values of δ_G is obtained as follows:

$$Pr(\delta_G) = \sum_{S \in \mathcal{S}} \left(Pr(S) \cdot \sum_{\varphi \in \mathcal{B}_G(S)} Pr(\varphi) \right) \quad . \tag{1}$$

We can now give the formal definition of *perfect* secret sharing scheme.

Definition 1. *A set of distribution rules* $\mathcal{F} = \bigcup_{S \in \mathcal{S}} \mathcal{F}_S$ *is a* perfect secret sharing scheme *having access structure* $\Gamma \subseteq 2^{\mathcal{P}}$ *if for all unauthorized subsets* $U \notin \Gamma$ *and for all shares distributions* δ_U *it results that* $Pr(S|\delta_U) = Pr(S)$.

Assuming that a suitable notion of *size* is defined on both the secrets and the shares (for example, the number of bits used to encode them), a perfect secret sharing scheme is called *ideal* if for all $S \in \mathcal{S}$ the sizes of the shares generated by any distribution rule $\varphi \in \mathcal{F}_S$ equal the size of S.

3 Building Preimages of Bipermutive CA

In this section we describe a procedure to reconstruct a preimage of a CA configuration under a bipermutive rule, which is the basic primitive used in our secret sharing scheme.

Let us suppose that $f : \mathbb{F}_2^{2r+1} \to \mathbb{F}_2$ is bipermutive. Denoting by g the generating function of f on the central $2r-1$ variables and by y the value of $f(x_1, \cdots, x_{2r+1})$, the following equalities hold:

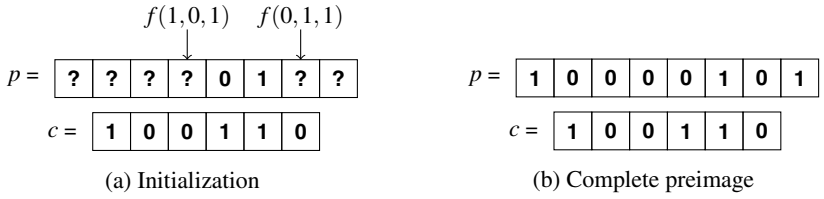
$$\begin{aligned} y &= x_1 \oplus g(x_2, \cdots, x_{2r}) \oplus x_{2r+1} \\ x_{2r+1} &= x_1 \oplus g(x_2, \cdots, x_{2r}) \oplus y \\ x_{2r+1} &= f(x_1, x_2, \cdots, x_{2r}, y) \ . \end{aligned}$$

Hence, the value of the rightmost input bit can be recovered by computing f on the vector $(x_1, x_2, \cdots, x_{2r}, y)$. Clearly, a symmetrical result holds when knowing y and the rightmost $2r$ bits of the input, which give the value of x_1. This property of bipermutive rules allows one to determine a preimage $p \in \mathbb{F}_2^{m+2r}$ of a configuration $c \in \mathbb{F}_2^m$ using the following procedure:

PREIMAGE-CONSTRUCTION

1. Set in a random position of p a block of $2r$ random bits $(p_i, p_{i+1}, \cdots, p_{i+2r-1})$.
2. Determine the value of p_{i+2r} by computing $f(p_i, \cdots, p_{i+2r-1}, c_i)$.
3. Shift the window of $2r$ bits one place to the *right*, and compute the value of p_{i+2r+1} by evaluating $f(p_{i+1}, \cdots, p_{i+2r}, c_{i+1})$. Continue to apply this step until the rightmost bit p_{m+2r} has been computed.
4. Determine the value of p_{i-1} by computing $f(c_{i-1}, p_i, \cdots, p_{i+2r-1})$.
5. Shift the window of $2r$ bits one place to the *left*, and compute the value of p_{i-2} by evaluating $f(c_{i-2}, p_{i-1}, \cdots, p_{i+2r-2})$. Continue to apply this step until the leftmost bit p_1 has been computed.

Thus, the preimage is uniquely determined by the value of configuration c and by the initial $2r$-bit random block. This implies that every CA configuration has exactly 2^{2r} preimages under a bipermutive rule; the fact that the initial block can be placed in any position does not influence the total number of preimages. Figure 1 reports an example of preimage computation using the elementary (i.e. with radius $r = 1$) rule 150.



(a) Initialization (b) Complete preimage

Fig. 1. Example of preimage computation for $c = (1,0,0,1,1,0) \in \mathbb{F}_2^6$ using the elementary bipermutive rule 150, defined as $f(x_{i-1}, x_i, x_{i+1}) = x_{i-1} \oplus x_i \oplus x_{i+1}$

Gutowitz [5] originally proposed to use the algorithm PREIMAGE-CONSTRUCTION to implement a CA-based symmetric cryptosystem, using local rules which are either leftmost or rightmost permutive, so that the initial block can be placed only to the leftmost and rightmost position of the preimage, respectively. The secret key is the permutive rule employed to build the preimages of the CA. Oliveira, Coelho and Monteiro [7] refined Gutowitz's cryptosystem by using bipermutive rules, since in this case the differences in the ciphertexts obtained starting from slightly different plaintexts propagate in both directions, and thus make differential cryptanalysis more difficult.

4 A (k,k) Secret Sharing Scheme

We now describe the basic version of our secret sharing scheme, in which all participants must pool their shares in order to recover the secret by using the properties of bipermutive rules explained in Section 3.

First, we observe that by iterating the procedure PREIMAGE-CONSTRUCTION, at each time step the size of the preimage grows by $2r$ bits. Hence, starting from a configuration c having length m, after t iterations we obtain a preimage of length $L(t) = 2rt + m$. Thus, given $k \in \mathbb{N}$, the number of iterations necessary to obtain a preimage of size $k \cdot m$ is $t = m(k-1)/2r$. Since t has to be an integer number, $2r$ must divide $m(k-1)$. However, in order to prove the security properties of our scheme in the next section, we assume the additional constraint that $2r$ divides m. Denoting by $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ the set of k players, we also assume that an order on $\mathcal{P}$ has already been mutually established by the dealer and the players, and that each player knows its index $i \in \{1, \cdots, k\}$.

Our secret sharing scheme can be described as follows.

Setup Phase

1. The *dealer* D sets the secret S to be shared (having length $|S| = m$ bits) as a configuration of a cellular automaton, and it chooses at random a bipermutive local rule of radius r, where r is such that $2r|m$.
2. D applies PREIMAGE-CONSTRUCTION for $T = m(k-1)/2r$ iterations, randomly choosing at each step the value and the position of the initial $2r$-bit block to start the construction of the preimage.
3. After T iterations, D obtains a preimage having length $k \cdot m$ which contains a sequence of $2r$ random adjacent bits starting at a random position. The preimage depends in general on $2rT = m(k-1)$ random bits. The dealer splits the preimage in k blocks of m bits, and securely sends one block to each player according to the order defined on $\mathcal{P}$ (hence block B_1 goes to P_1, B_2 to P_2, etc. up to P_k). Finally, D publishes the bipermutive rule used to evolve the CA backwards.

Recovery Phase

1. All the k players pool their shares in the correct order to get the complete preimage of the CA.
2. After having determined the preimage, the players evolve the CA forward for $T = m(k-1)/2r$ iterations, using the local rule published by the dealer. Notice that the players can compute by themselves T, since they know m, k and r.
3. The configuration obtained after T iterations is the original secret S.

Figure 2 depicts the setup phase.

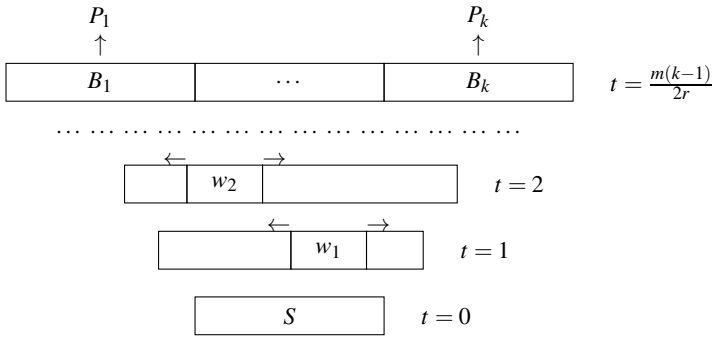


Fig. 2. Setup phase of the (k,k) secret sharing scheme. The randomly placed blocks w_i represent the initial $2r$ random adjacent bits used to reconstruct each preimage.

5 Security Properties of the Basic Scheme

The access structure of the basic scheme shown in Section 4 is trivially composed by one set, which is the set of all players $\mathcal{P}$. In order to investigate the security properties of the scheme, we thus have to analyse the information an attacker can gain about the secret by knowing a subset of $k-1$ or fewer shares. We begin with the following preliminary results.

Lemma 1. *Let $F : \mathbb{F}_2^{m+2r} \to \mathbb{F}_2^m$ be the global rule of a CA defined by a bipermutive local rule $f : \mathbb{F}_2^{2r+1} \to \mathbb{F}_2$. Then, by fixing a block $\tilde{x} \in \mathbb{F}_2^{2r}$ of $2r$ adjacent coordinates in $x \in \mathbb{F}_2^{m+2r}$, the resulting restriction $F|_{\tilde{x}} : \mathbb{F}_2^m \to \mathbb{F}_2^m$ is a permutation on $\mathbb{F}_2^m$.*

Proof. Let us denote by $x \in \mathbb{F}_2^{m+2r}$ the CA configuration and by $y \in \mathbb{F}_2^m$ the image of F. Assuming that the first component of the block $\tilde{x} \in \mathbb{F}_2^{2r}$ is placed at position i of x, with $1 \leq i \leq m-2r+1$, we have to prove that the function $F|_{\tilde{x}}$ which maps the remaining vector $\hat{x} = (x_1, \cdots, x_{i-1}, x_{i+2r}, \cdots, x_m) \in \mathbb{F}_2^m$ to y is bijective. Given $y \in \mathbb{F}_2^m$ and a block of $2r$ consecutive bits in the preimage, the remaining ones are uniquely determined by the application of the algorithm PREIMAGE-CONSTRUCTION. *Consequently, each output $y \in \mathbb{F}_2^m$ has a unique preimage under $F|_{\tilde{x}}$, and thus $F|_{\tilde{x}}$ is bijective.* □

In the next Lemma we denote by $x(t)$ the configuration obtained by evolving the CA forward for t steps, with $x(0)$ being the first preimage resulting from the juxtaposition of the k shares in the correct order. We also use the notation $(x_u, \cdots, x_v)(t)$ to represent the subvector of the configuration $x(t)$ included between the indices $u < v$.

Lemma 2. *Let B_l, with $1 \leq l \leq k$, be the only unknown share among $B_1, \cdots, B_k$. Then, there exists a permutation $\Pi : \mathbb{F}_2^m \to \mathbb{F}_2^m$ between B_l and the secret S.*

Proof. We first consider the case where $l = 1$, in which the unknown block of m bits is $B_1 = (x_1, \cdots, x_m)(0)$. By evolving the CA forward, at each time step t the block $(x_1, \cdots, x_m)(t)$ remains unknown. Indeed, since the application of the global rule shrinks the configuration by $2r$ bits, after t iterations only the rightmost $m(k-1) - 2rt$ bits of $x(t)$ are determined. In particular, the block $(x_{m+1}, \cdots, x_{m+1+2r})(t)$ of $2r$ bits is known.

Hence, for all $t \in \{1, \cdots, T\}$ *where* $T = m(k-1)/2r$*, by Lemma 1 there is a permutation* $\pi_1(t) : \mathbb{F}_2^m \to \mathbb{F}_2^m$ *which maps the block* $(x_1, \cdots, x_m)(t-1)$ *in the block* $(x_1, \cdots, x_m)(t)$*. By observing that* $(x_1, \cdots, x_m)(T) = S$*, the permutation* Π *between* B_1 *and* S *can thus be defined as* $\Pi = \pi_1(T) \circ \pi_1(T-1) \circ \cdots \circ \pi_1(1)$*.*

If the missing share is B_k*, a symmetric reasoning stands by considering for all* t *the block* $(x_{m(k-1)+1}, \cdots, x_{mk})(t)$ *containing the rightmost* m *unknown bits.*

We now prove the generic case where $1 < l < k$*. This means that the unknown* m*-bit block is* $(x_{(l-1)m+1}, \cdots, x_{lm})(0)$*. Similarly to the case* $l = 1$*, by evolving the CA forward the block* $(x_{(l-1)m+1-2rt}, \cdots, x_{lm-2rt})(t)$ *is undetermined, and by Lemma 1 there is a permutation* $\pi_l(t)$ *between this block and* $(x_{(l-1)m+1-2r(t-1)}, \cdots, x_{lm-2r(t-1)})(t-1)$*. In particular, the* $2r$*-bit block which fixes* $\pi_l(t)$ *is the one placed to the left of the cell* $x_{(l-1)m+1-2r(t-1)}$*. Clearly,* t *does not range in the set* $\{1, \cdots, T\}$*. In fact, at a certain step* $\hat{t} < T$ *the index of the first cell in the unknown* m*-bit block will be less than* $2r$*. Recall that in Section 4 we required that* $2r|m$*, i.e.* $m = 2rb$ *for some* $b \in \mathbb{N}$*. Thus, at time* $\hat{t} = 1 + (l-1)b$ *the vector* $(x_1, \cdots, x_m)(\hat{t})$ *is undetermined. But we know from the case* $l = 1$ *that there is a series of permutations* $\pi_1(t)$ *for* $\hat{t} < t \leq T$ *which maps this block to the secret* S*. Hence, the permutation* Π *between* B_l *and* S *can be defined as* $\Pi = \pi_1(T) \circ \cdots \circ \pi_1(\hat{t}+1) \circ \pi_l(\hat{t}) \circ \cdots \circ \pi_l(1)$*.* □

In what follows, we denote by $\mathcal{S} = \mathbb{F}_2^m$ the space of secrets, which coincides with the space of shares $\mathcal{B}$. Let us assume that the uniform probability distribution is defined on $\mathcal{S}$, that is $Pr(S) = 1/2^m$ for all $S \in \mathcal{S}$. Given a secret S, the distribution rule $\varphi \in \mathcal{F}_S$ assigning to each of the k players a share from $\mathcal{B}$ is determined by the $2r$-bit blocks used to build the CA preimages, thus by a total of $m(k-1)$ bits. Assuming that the dealer chooses uniformly at random these bits, for all $\varphi \in \mathcal{F}_S$ it follows that $Pr(\varphi) = 1/2^{m(k-1)}$.

Let us suppose that $U \subseteq \mathcal{P}$ is a subset of participants such that $|U| = |\mathcal{P}| - 1 = k - 1$. The probability that a particular share distribution δ_U occurs given a secret S can be computed as

$$Pr(\delta_U | S) = \sum_{\varphi \in B_U(S)} Pr(\varphi) \ . \tag{2}$$

Given $S \in \mathcal{S}$, by Lemma 2 we know that for all distributions of $k-1$ shares δ_U there is only a single additional share B_l that, joined to those of U, uniquely determines the secret S, since there is a permutation Π between B_l and S. Consequently, $|B_U(S)| = 1$ and Equation (2) becomes

$$Pr(\delta_U | S) = \sum_{\varphi \in B_U(S)} Pr(\varphi) = \frac{1}{2^{m(k-1)}} \ . \tag{3}$$

Computing the probability of a particular share distribution δ_U over all possible secrets $S \in \mathcal{S}$ is now straightforward:

$$Pr(\delta_U) = \sum_{S \in \mathcal{S}} \left(P(S) \cdot \sum_{\varphi \in B_U(S)} Pr(\varphi) \right). \tag{4}$$

Since $Pr(S) = 1/2^m$ for all $S \in \mathcal{S}$ and $|\mathcal{S}| = 2^m$, Equation (4) can be rewritten as

$$Pr(\delta_U) = 2^m \cdot \frac{1}{2^m} \cdot \frac{1}{2^{m(k-1)}} = \frac{1}{2^{m(k-1)}} \ . \tag{5}$$

We have thus concluded that, for all $S \in \mathcal{S}$,

$$Pr(\delta_U) = Pr(\delta_U | S) \ , \tag{6}$$

that is, the occurrence of the share distribution δ_U to the subset U is independent from the occurrence of the secret S. Clearly, we can make the same reasoning for all subsets $U \subseteq \mathcal{P}$ such that $|U| < |\mathcal{P}| = k$; the only quantity which changes is the cardinality of the set $B_U(S)$, but this is irrelevant in order to get Equation (6).

We can now prove the following result.

Theorem 1. *The (k,k) secret sharing scheme described in Section 4 is perfect.*

Proof. Let U be a generic unauthourized subset of players having cardinality $|U| < k$, and let δ_U be a share distribution to the members of U. Using Bayes' theorem, we have

$$Pr(S|\delta_U) = \frac{Pr(\delta_U|S) \cdot Pr(S)}{Pr(\delta_U)} \ . \tag{7}$$

By Equation (6)*, we know that $Pr(\delta_U) = Pr(\delta_U|S)$. Hence, $Pr(S|\delta_U) = Pr(S)$.* □

Thus, by knowing $k-1$ or fewer shares the attacker gains no information about the secret. Finally, it is also easy to see that the scheme is ideal, since the CA is iterated the number of times necessary to get a preimage having length $k \cdot m$. Hence, the size of each of the k shares equals the size of the secret.

6 An Extension to the Basic Scheme

The secret sharing scheme described in Section 4 can be trivially employed to implement any access structure $\Gamma \subseteq \mathcal{P}$: for each authorized subset $A \in \Gamma$, it simply suffices to re-run the setup phase and create a new (independent) set of shares to be distributed to the members of A. However, as the size of $\mathcal{P}$ grows, this method quickly becomes impractical, since each player must hold a different share for every authorized subset he belongs to. We now introduce an extension to the basic scheme called *secret juxtaposition*, which allows one to reuse the same shares thus creating more authorized subsets with a single setup phase.

Let us assume that a set of k shares has been created by following the basic setup procedure, and distributed to a set of k participants. In order to add an additional player without having to recompute all the shares, we can append another copy of the secret S to the right of the existing one (respectively, to the left). Then, we run the algorithm PREIMAGE-CONSTRUCTION for each preimage towards the right (respectively, towards the left) to compute the additional share of $m = |S|$ bits. Note that in this case it is not necessary to pick random bits, since in each preimage more than $2r$ bits are already determined.

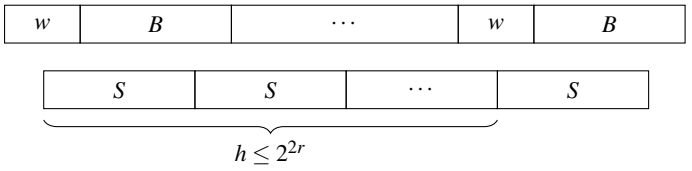


Fig. 3. After at most $h \leq 2^{2r}$ juxtaposed copies of S, the algorithm PREIMAGE-CONSTRUCTION repeats the $2r$-bit block w at the end of the preimage. At this point, the subsequent m-bit block in the preimage will be a copy of B, since the part of the image which is relevant for PREIMAGE-CONSTRUCTION is always S.

Thus, by juxtaposing q copies of the secret S and by evolving the CA backwards for $T = m(k-1)/2r$ steps, we get $q+k-1$ shares of m bits, and each of the q subsets of k consecutive shares can recover the secret, since the corresponding space-time cone collapses on a copy of S. The resulting access structure is however more flexible than the sequential threshold induced by del Rey, Mateus and Sánchez's scheme [2]. In fact, in our scheme by continuing to append copies of the secret the shares obtained will eventually repeat. This happens because after at most 2^{2r} juxtapositions of S, the last $2r$-bit block in the first preimage will be already occurred at another index corresponding to the end of a copy of S and the beginning of the next one (see Figure 3).

As a consequence, after a certain number of juxtapositions the access structure of our scheme becomes *cyclic*, meaning that the sets of k consecutive shares which can recover the secret are the ones that can be formed by considering the CA preimage at time T as a ring. The minimal authorized subsets $M \in \Gamma_0$ of a generic (k,n) cyclic access structure are defined for all $i \in \{1,\cdots,n\}$ as $M = \{P_{j(i)}, P_{j(i+1)}, \cdots, P_{j(i+k-1)}\}$, where $j(z) = 1 + [(z-1) \mod n]$ for all $z \in \{i,\cdots,i+k-1\}$. It is easy to see that in a (k,n) cyclic access structure there are n minimal authorized subsets, in contrast to the $n-k+1$ yielded by a sequential threshold scheme. Thus, assuming that the first preimage of the CA repeats itself after n juxtapositions of the secret S, with $n \leq 2^{2r}$, the extended scheme can be used to implement a cyclic access structure for a set of n players, by evolving the CA backwards for T steps starting from a configuration composed by n copies of S. We note that Theorem 1 still holds for the extended scheme, under the assumption that the attacker knows fewer than k consecutive shares.

7 Conclusions

We presented a new secret sharing scheme which employs bipermutive cellular automata as a primitive, differently from the LMCA-based approach schemes usually proposed in the literature. The main idea of our scheme is to set the secret S as a configuration of a bipermutive CA and to evolve it backwards using the algorithm PREIMAGE-CONSTRUCTION until a preimage of length $k \cdot m$ is obtained, which is subsequently splitted among the k players. We proved that this basic version of the scheme in which all the k players have to pool their shares to recover the secret is perfect, meaning that an attacker who knows fewer than k shares gains no information about S. Moreover, we showed that the scheme is ideal, since the shares have the same size of the secret.

We finally introduced an extension to the basic scheme which allows one to generate more authorized subsets. The extension simply consists in juxtaposing q copies of the secret S and then evolve the CA backwards until a preimage of $m(q+k-1)$ bits is reached. In this way, there are q sets of k consecutive shares which can determine the secret. The resulting access structure eventually becomes cyclic, since after at most 2^{2r} juxtapositions of the secret the final shares will begin to repeat themselves.

The main advantages of our scheme compared to del Rey, Mateus and Sánchez's LMCA-based model [2] can be synthesised as follows. First, bipermutive CAs have a simpler structure than k-th order linear memory CA, since the next states of the cells depend only on the current configuration. Hence, our scheme is possibly amenable to more efficient and cost-effective hardware implementations. Moreover, the cyclic access structure induced by our scheme is more flexible, since it eventually generates n minimal authorized subsets rather than the $n-k+1$ produced by the scheme of del Rey, Mateus and Sánchez [2].

There are several possibilities for further research and improvements on this subject. From a practical point of view, it would be useful to investigate if there exists a general method to determine *exactly* after how many juxtapositions of the secret the shares begin to repeat themselves, without having to simulate the CA backwards. This is equivalent to the following problem: given a bipermutive rule f and a CA configuration having spatial period m, find the periods of its preimages under the application of f. Another interesting direction of research would be to generalise the secret sharing scheme to the case of d-dimensional cellular automata, with $d \geq 2$, and to consider the resulting access structures: clearly, the number of authorized subsets would be greater than in the one-dimensional case, since each share would be adjacent to more shares.

Acknowledgements. The authors wish to thank the anonymous referees for their comments to improve the paper. This work was partially supported by Consorzio Milano Ricerche and International Analytics Ltd.

References

1. Blakley, G.: Safeguarding cryptographic keys. In: Proceedings of the 1979 AFIPS National Computer Conference, pp. 313–317. AFIPS Press, Monval (1979)
2. 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)
3. Eslami, Z., Ahmadabadi, J.Z.: A verifiable multi-secret sharing scheme based on cellular automata. Inf. Sci. 180(15), 2889–2894 (2010)
4. Eslami, Z., Razzaghi, S.H., Ahmadabadi, J.Z.: Secret image sharing based on cellular automata and steganography. Pattern Recogn. 43(1), 397–404 (2010)
5. Gutowitz, H.: Cryptography with dynamical systems. In: Goles, E., Boccara, N. (eds.) Cellular Automata and Cooperative Phenomena, pp. 237–274. Kluwer Academic Press (1993)
6. Marañón, G.Á., Encinas, L.H., del Rey, Á.M.: A multisecret sharing scheme for color images based on cellular automata. Inf. Sci. 178(22), 4382–4395 (2008)
7. Oliveira, G., Coelho, A., Monteiro, L.: Cellular automata cryptographic model based on bidirectional toggle rules. Int. J. Mod. Phys. C 15(8), 1061–1068 (2004)
8. Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (1979)
9. Stinson, D.R.: Cryptography: theory and practice, 3rd edn. CRC Press, Boca Raton (2006)