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