Discrete Physics, Cellular Automata and Cryptography

Stephane Marconi and Bastien Chopard

Computer Science Department, University of Geneva, Switzerland
stephane.marconi@cui.unige.ch,
bastien.chopard@cui.unige.ch

Abstract. This paper aims at showing that Physics is very close to the substitution-diffusion paradigm of symmetric ciphers. Based on this analogy, we present a new Cellular Automata algorithm, termed `Crystal`, implementing fast, parallel, scalable and secure encryption systems. Our approach provides a design principle to ensure an invertible dynamics for arbitrary neighborhood. Thus, several variants of our CA can be devised so as to offer customized encryption-decryption algorithms. Considering larger data blocks improve both security and speed (throughput larger than 10Gbps on dedicated hardware).

1 Introduction

As introduced by Shannon [7], symmetric block ciphers are usually based on r rounds of diffusion and confusion operations applied to a plain text message M. This transformation is usually considered in a purely mathematical framework, with no reference to any physical process despite the fact that the term diffusion actually refers to a well known physical phenomena.

It seems that the contribution of physics to **classical** cryptography (quantum cryptography thus excluded) has been only to provide some vocabulary but no design principles and the few physical devices that have been proposed to encode a message are usually rather exotic and their security hard to prove [6].

Here we claim that the analogy between classical physics and symmetric block cipher is strong, natural and useful. This claim is made very clear when considering discrete physical models such lattice gases automata (LGA) used to model fluids [1].

These models consist of a discrete space time abstraction of the real world. N point-particles move on a regular lattice in D spatial dimensions. The possible velocities of each particle are restricted by the lattice topology: the propagation P moves, in one time step Δt, a particle from one site to one of its neighbor. Thus, if z is the lattice coordination number, particles may have z possible velocities. A collision C occurs between particles entering the same site from different directions. The result of such a collision is to create new particles in some directions and to remove some particles in others. Particle motion and collision are repeated alternatively for any chosen amount of time. Mathematically, the dynamics of our discrete fluid can be described by

$$M(t + \Delta t) = PCM(t)$$

S. El Yacoubi, B. Chopard, and S. Bandini (Eds.): ACRI 2006, LNCS 4173, pp. 617–626, 2006.

where $M(t)$ is the configuration of the particles over the full lattice at iteration t. This dynamics is structurally identical to the diffusion-confusion paradigm of cryptography. Diffusion is produced by an operator P and operator C implements a substitution box.

The other relevant ingredient from Physics is the second principle of thermodynamics which states that all configurations evolve to a final state which seems to contain no more memory of the initial situation. As such, this process is a good encryption mechanism. Deciphering, fortunately, is possible since the microscopic laws of physics are fully symmetrical with respect to past and future. Theoretically then, there is a way to come back. It is however highly impractical with real physical systems: one would have to reach every single particle of the system and to reverse its microscopic velocity with arbitrary precision.

On the contrary, with LGA systems, this time reversal is possible since the calculation is Boolean and performed without truncation error. We can thus reverse the arrow of time by simply inverting the direction of motion of each particles: $M(t) = RM(t)$, where R is the so-called time-reversal operator. Therefore, a deciphering mechanism is already embedded in a system obeying $CRC = R$ and $PRP = R$. It is then identical to the ciphering steps because

$$\begin{aligned}(CP)^r R(PC)^r &= (CP)^{r-1} CPRPC(PC)^{r-1} \\ &= (CP)^{r-1} R(PC)^{r-1} = R \end{aligned} \tag{1}$$

It is well known that time-reversibility in a physical system is highly sensitive to any small perturbation. Thus, the keying mechanism for the cipher may be viewed as errors that are deliberately introduced to prevent an attacker to reverse time.

Due to their properties of producing a complex behavior, cellular automata (CA) have been considered by several authors as a way to build cryptographic devices [5,9,4]. Several of the proposed CA are designed to produce a sequence of bits out of a secrete key and, as such, provide a stream cipher in which a sender and a receiver can both produce the same complex sequence of bits starting from an initial state given by the key.

However, when symmetric block ciphers are devised, it is necessary that encryption can be inverted in order to be able to decipher an encoded message. Therefore, a central question arises about how to build invertible CA's.

The standard definition of CA uses the so-called "gather-update" paradigm [1] (first get the neighbor values and then update the cell). It is well known that finding the inverse of a CA rule when the gather-update paradigm is used is a difficult task [3]. A procedure to produce a reversible CA rule (the rule is its own inverse) is the so-called technique of Fredkin [8]: a reversible cellular automata can be constructed by using the following rule:

$$s(r, t+1) = f(s(N(r), t)) \oplus s(r, t-1)$$

where f is arbitrary and N designate the neighborhood of cell r. This rule is said to be of second order since it requires state t and $t-1$ to compute the evolution.

Another approach to produce invertible CA uses the so-called block-permutation CA [3]. The central idea is to partition the CA cells into adjacent blocs of size $w \times w$, with respect to origin (ox, oy), and to define a function F applying the block to itself. By changing the partition offset (ox, oy), one obtains a family of different transformations of the cell space. Several of these transformation can be composed so as to produce a CA rule. Within this paradigm of block-partition CA, an invertible CA can be designed by taking the function F invertible. This approach however is restricted to regular, Cartesian grids and is non-local.

Finally, a last paradigm to implement a CA rule is the collision-propagation paradigm of LGA discussed above to model discrete physical systems. In this approach, it has been noted that the dynamics is reversible (i.e. is its own inverse) when the collision operator implements a reversible physical processes.

Our approach exploits this last paradigm to build a general reversible CA in a possibly irregular topology, of arbitrary dimension, through the introduction of three inter-related operators P, R and C. The main advantage of this formulation is that it offers an effective way to build both a hardware and a software device, with high scalability. In addition, it reconciliates the well admitted Shannon generic model of symmetric cryptography (confusion and diffusion) with the promising domain of complex dynamical systems (e.g CA) that are often considered as exotic and non-reliable cryptographic methods.

2 Description of the Algorithm

We first discuss a simple instance of the algorithm and then we formalize a general approach. Let us consider a 2D square periodic lattice with z bits per site and containing a N-bit message. With N bits distributed over the z directions, the lattice size must be $\sqrt{N/z} \times \sqrt{N/z}$.

When $z = 8$, each lattice site has eight neighbors, four along the main lattice directions, as well as four along the diagonals. This so-called D2Q8 topology defines the action of P. Note that the z links are two-way between the interconnected neighbors; they are labeled by a direction index $j = 0, ..., (z-1)$ so that opposite directions j and j' are such that $j' = j + (z/2) \mod z$. By definition, the reverse operator R swaps the content of direction j and j'. By construction, we thus have $R^2 = 1$, i.e. $R^{-1} = R$.

The collision C is implemented as lookup table. In order to ensure $CRC = R$, the following randomized algorithm is used (here $z = 8$):

```
for all a=0 to 255, such that C(a) is still undefined
  do b=rand(0,255)
  until C(R(b)) is undefined
  C(a)=b;  C(R(b))=R(a);
endfor
```

The cipher key K is a N-bit string. It can be easily constructed from a N'-bit string, with $N' \leq N$, using any acceptable padding procedure.

With these ingredients, we propose the following block cipher algorithm

```
algorithm Crystal(M,K)  // M is the message, K the key
  reverse(M), reverse(K)
  propagation(M), propagation(K)
  repeat r times
    M=M+K
    collision(M),  collision(K)
    propagation(M), propagation(K)
  end repeat
  M=M+K
  return M, K
end algorithm
```

Note that operators R, C and P act locally but, by extension we also use the same symbols to denote the synchronous action of R, C and P at all sites.

It can be shown that `Crystal` both encodes and decodes the blocks. Indeed the above algorithm can be expressed in a matrix formulation

$$\begin{pmatrix} M' \\ K' \end{pmatrix} = \begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \left[PC \begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \right]^n PR \begin{pmatrix} M \\ K \end{pmatrix}$$

in which we assume a modulo 2 algebra so that

$$\begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \begin{pmatrix} M \\ K \end{pmatrix} = \begin{pmatrix} M \oplus K \\ K \end{pmatrix}$$

In addition, we define the product of the operator PC by a matrix as

$$PC \begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} = \begin{pmatrix} PC & PC \\ 0 & PC \end{pmatrix}$$

In order for our scheme to be reversible, we need

$$\begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \left[PC \begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \right]^n PR \begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \left[PC \begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix} \right]^n PR = \begin{pmatrix} 1\ 0 \\ 0\ 1 \end{pmatrix}$$

This is achieved provided that

$$PRP = R^{-1} \qquad CR^{-1}C = R$$

The proof follows by applying the same procedure as used in eq. 1 and by the fact that, in a modulo 2 algebra

$$\begin{pmatrix} 1\ 1 \\ 0\ 1 \end{pmatrix}^2 = \begin{pmatrix} 1\ 0 \\ 0\ 1 \end{pmatrix}$$

Note that in simple and regular topologies we have $R = R^{-1}$. However, the above formulation shows that any topology of interconnected cells for which

$PRP = R^{-1}$ and $CR^{-1}C = R$ hold can be used to implement the `Crystal` algorithm.

Such a topology can be constructed in a very general way, with possibly a different number of neighbors for each cell. The key condition is to distinguish the input and output links of the cell and to impose a suitable symmetry relation between them. This is detailed below.

Let R be a one-to-one mapping from the inputs to the output, as shown for instance in fig. 1 (a). So, within a cell there must be the same number of input and output ports. The collision operator C is also a one-to-one transformation of the the input data into the output data. This mapping is constructed so that $CR^{-1}C = R$. The propagation operator P transfers these output data to the input ports of the corresponding neighboring cells, as illustrated in fig. 1 (b). In order to build a reversible CA rule the following must be true: for each link connecting output j of cell r to input ℓ of cell r', there is a second link connecting output ℓ' of cell r' to input j' of cell r. If j and ℓ are such that $j = R(j')$ and $\ell' = R(\ell)$ (see fig. 1) then, by construction $PRP = R^{-1}$.

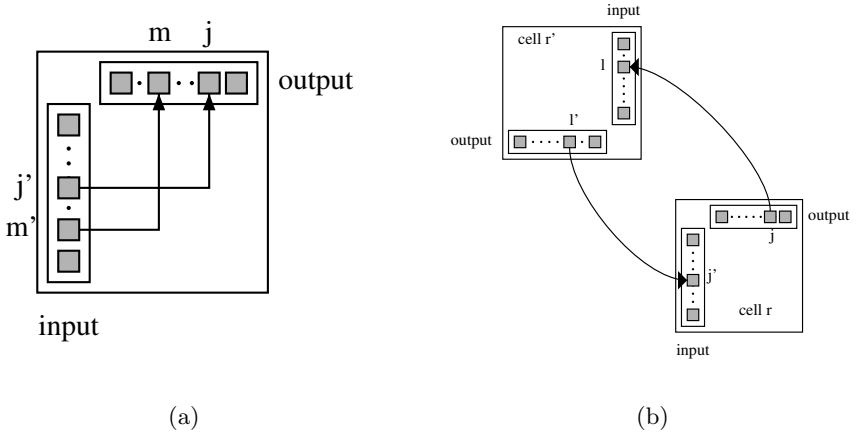

Fig. 1. (a) Illustration of the reverse operator R. (b) Illustration of the propagation operator P.

Therefore, any irregular interconnection topology obeying this pairwise symmetry can be devised to obtained a *reversible* dynamics. More generally, an *invertible* dynamics can be obtained by having two collision operators C and C' such that $C'R^{-1}C = R$. Thus, we can also think of our algorithm as a way to connect different processors, each running locally an invertible encryption process C and whose decryption is C'.

Within this relatively large framework, we can easily imagine several keying mechanisms, such as a secret topology, a secret collision or the more classical choice of secret bit string K.

3 Throughput and Security

We derive some properties of our cryptographic system in the case of the D2Q8 topology. These properties allow us to quantify both the security and the performances of the `Crystal` algorithm.

(1) A required property of a cipher is a high sensitivity to a little modification in the initial message M. We observe in fig. 2 that, after a number r of rounds equals to the lattice diameter $d = (1/2)\sqrt{N/z}$, a single bit error causes an avalanche of the full lattice size. The average Hamming distance between two messages initially differing only by one bit is $(1/2)N$ as expected for two random messages. Based on the speed at which information travels in the lattice, the

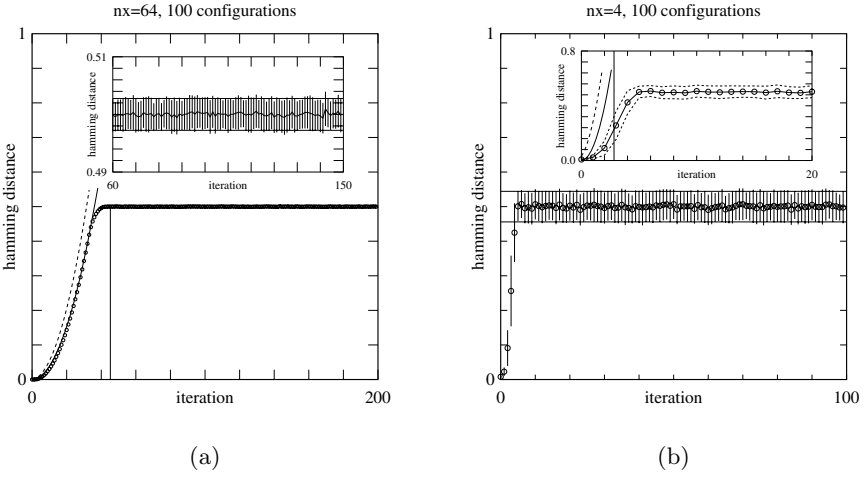

Fig. 2. Evolution of the Hamming distance between two messages initially differing only by one bit. In (a) we have $N = 64 \times 64 \times 8 = 32768$ bits and in (b) $N = 4 \times 4 = 128$ bits. Comparison with the ideal curve (eq. 2 is given with the doted parabola. The solid line parabola is the theoretical estimate of eq. 3). Finally, the vertical line show the iteration $r = (\sqrt{2}/2)\sqrt{N/z}$ at which the plateau should be reached.

Hamming distance can at best evolve as (see [2])

$$H(r) = \frac{1}{2}z(2r+1)^2 \tag{2}$$

In numerical experiments, the speed at which the 1/2 plateau is reached is less than predicted by eq. 2 because after a collision, only about $z/2$ bits differ from the reference configuration. From fig. 3, we can assume that the error propagates roughly as a disk. Its diameter grows on average by one lattice site at each iteration. Thus, during the first $r = \sqrt{N/z}/2$ rounds, H behaves as

$$H(r) = \frac{z}{2}\pi r^2 \tag{3}$$

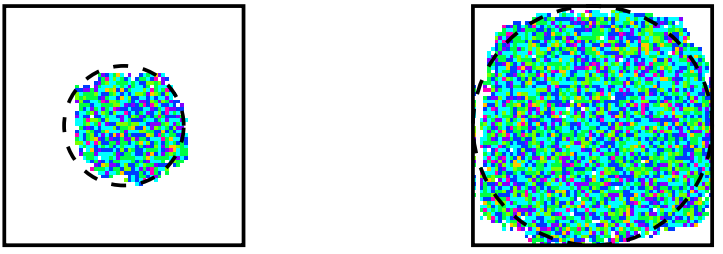

Fig. 3. Snapshot of the error propagation region, after 16 and 32 iterations, in a system of size 64×64. The non-blank regions indicates where the two configurations differ. The darker the gray, the more are the bits that differ. The dashed-line disks have radius 16 and 32, respectively; thus, the error propagates at speed one for this topology.

Therefore the minimal number r of rounds needed to mix the information all over the system must be $r = \alpha d$ where α is some constant larger than 1.

(2) Once the number of round is determined, we may compute the throughput of the algorithm. As any CA model, the dynamics of our system can be fully parallelized so that propagation and collision take a constant time for any N. Then, the time T needed to encrypt is proportional to the number of rounds but independent of the block size

$$T \propto r = \alpha d \propto \sqrt{N/z} \tag{4}$$

and therefore the encryption throughput W is

$$W = \frac{N}{T} \propto \sqrt{N} \tag{5}$$

Thus, when large data blocks are encrypted, the throughput increases although the number of round increases. The reason is that the number of rounds grows slower than the amount of data. Implementation studies on FPGA indicates that $W > 10Gb/s$ can be achieved with reasonable resources.

(3) It is commonly accepted that increasing the number of round r increases security. Therefore, with a full parallel implementation and large data blocks, both security and throughput are improved when `Crystal` is used.

Security can be assessed quantitatively by a differential cryptanalysis approach. The goal is to obtain information on the key K by considering how two plain text messages M_1 and M_2 get encrypted into M_1' and M_2'.

With $M_i^{(m)}$ and $K^{(m)}$ denoting the state of the messages and the key after m rounds, the algorithm `Crystal` gives

$$M_i^{(m)} = PC\left(M_i^{(m-1)} \oplus K^{(m-1)}\right) \tag{6}$$

for $i = 1, 2$. By XORing the above relation for $i = 1$ and $i = 2$ and applying inverse propagation, we obtain

$$P^{-1}\left(M_1^{(m)} \oplus M_2^{(m)}\right) = C\left(M_1^{(m-1)} \oplus K^{(m-1)}\right) \oplus C\left(M_2^{(m-1)} \oplus K^{(m-1)}\right) \quad (7)$$

It is now convenient to define $\mathcal{F}^{-1}$ as

$$a_1 \oplus a_2 \in \mathcal{F}^{-1}(b) \qquad \text{iff} \qquad b = C(a_1) \oplus C(a_2) \quad (8)$$

For a given collision operator C, $\mathcal{F}^{-1}$ can be computed easily by an exhaustive search [2]. With definition 8, we can rewrite eq. 7 as

$$\begin{aligned}\mathcal{F}^{-1}P^{-1}\left(M_1^{(m)} \oplus M_2^{(m)}\right) &= M_1^{(m-1)} \oplus K^{(m-1)} \oplus M_2^{(m-1)} \oplus K^{(m-1)} \\ &= M_1^{(m-1)} \oplus M_2^{(m-1)} \quad (9)\end{aligned}$$

By repeating this relation, one obtains

$$M_1^{(1)} \oplus M_2^{(1)} = \left(\mathcal{F}^{-1}P^{-1}\right)^{r-1}\left(M_1^{(r)} \oplus M_2^{(r)}\right) \quad (10)$$

where r is the number of rounds. In [2] we show that if $M_1^{(1)} \oplus M_2^{(1)}$ is known to the attacker, it is rather easy to obtain the secret key K with an extra 2^z operations.

Below we compute how much computational effort is required to obtain $M_1^{(1)} \oplus M_2^{(1)}$ from $M_1^{(r)} \oplus M_2^{(r)}$ which, by hypothesis, is known since attackers are supposed to have access to any pairs (M, M') they want.

Since we assume that $r > d$, where d is the lattice diameter, $M_1^{(r)}$ and $M_2^{(r)}$ differ over all N/z lattice sites. In order to perform the backward scheme indicated in eq. 9, one has to find all possible pre-images of $P^{-1}\left(M_1^{(m)} \oplus M_2^{(m)}\right)$ by $\mathcal{F}^{-1}$. Empirically we observe that the number of pre-image of a given b is larger than $2^z/4$. Of course this depends on the choice of C, but this seems to be a minimal value for a C constructed with our randomized procedure.

Therefore, for each lattice site, at least 2^{z-2} values are possible for $M_1^{(r-1)} \oplus M_2^{(r-1)}$. This requires to select $(N/z)2^{z-2}$ candidates for $M_1^{(r-1)} \oplus M_2^{(r-1)}$. The same argument can be repeated $r - d$ times. After that, we can quickly exclude some possibilities. Indeed, at this point, we know that the error has not been able to propagate up to the outer boundary of the lattice. For these lattice sites, $M_1^{(d-1)} \oplus M_2^{(d-1)}$ must be zero. Thus the number of sites for which the exploration continues is $(\sqrt{N/z} - 2)^2$. If we undo one more step, even more possibilities can be excluded and the pre-images of "only" $(\sqrt{N/z} - 4)^2$ sites must be investigated.

Following this idea for the $d - 1$ steps, one has to explore $3^2 \times 5^2 \times ... \times (\sqrt{N/z} - 1)^2$ possible configurations[1], each with $2^z/4 = 2^{(z-2)}$ possible values.

[1] For a D2Q8 topology.

An inferior bound for this number is (see [2])

$$\left(3^2 \times 5^2 \times ... \times (\sqrt{N/z} - 1)^2\right) 2^{z-2} > (d/2)^{2d} 2^{z-2} = \frac{1}{4} \left(\frac{N}{z}\right)^d 2^{z-2}$$

Thus, in total (undoing the rounds beyond and below the diameter) implies to investigate

$$\mathcal{N} > (N/z)^{r-d} 2^{(z-2)(r-d)} (N/z)^d 2^{z-4} = (N/z)^r 2^{(z-2)(r-d)+(z-4)} \tag{11}$$

candidates for $M_1^{(1)} \oplus M_2^{(1)}$.

Let us define the security measure S as the logarithm of our estimate of $\mathcal{N}$

$$S = \log_2 \mathcal{N} \tag{12}$$

A security of $S = 128$ is usually considered as safe. Eq. 11 can be shown as a graph. In figure 4 (a), we show how r must change with respect to N, for a given security level S. In figure 4 (b) we show how security S increases with N when we take the number of round r as twice the diameter d.

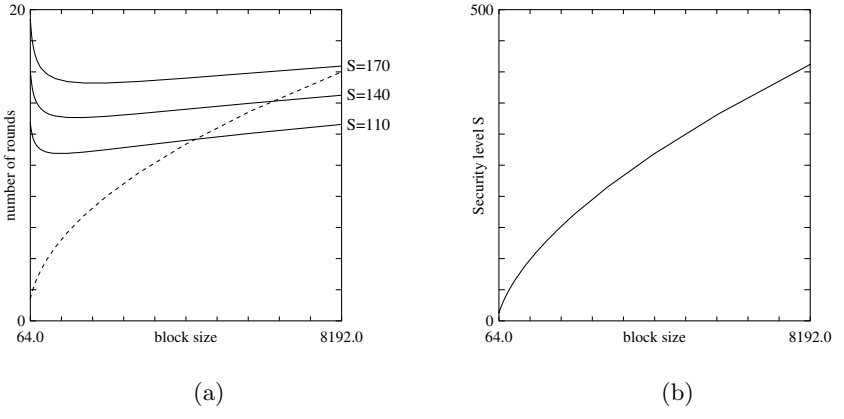

Fig. 4. (a) Number of rounds r as a function of block size N, to keep a given security level S. Note that r must be larger than the diameter d. The limit $r = d$ is shown by the dashed curve. (b) Security S as a function of block size N, for $r = 2d$.

4 Conclusion

A first specificity of `Crystal` with respect to standard block ciphers is that it is made of many fully identical components (the sites). Hence, it is local, scalable, fully parallel and fits naturally on silicon.

Second, `Crystal` can be tailored in many different variants, so as to provide each user with a unique encryption-decryption method, whose details can be kept

secret in addition to the key. The simplest way to customize `Crystal` is to choose a personal substitution box C. Indeed a large number of C's can be generated with the same level of security. Other ways to customize the algorithm is to have a main substitution box C and a second one C' active only at some secrete cells. Finally the shape of the encryption domain can be a secrete information.

In conclusion, we have described a new cipher which is cost effective to develop and implement, simple to analyze and which efficiently addresses the increasing needs for high throughput, high security and high level of versatility.

References

1. B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.
2. B. Chopard and S. Marconi. Discrete physics: a new way to look at cryptography. Technical report, University of Geneva, 2005. http://arXiv.org/abs/nlin.CG/0504059.
3. Jerome Durand-Lose. Representing reversible cellular automata with reversible block cellular automata. *Discrete Mathematics and Theoretical Computer Sciences Proceedings AA (DM-CCG)*, pages 145–154, 2001.
4. E. Franti, S. Goschin, M. Dascalu, and N. Catrina. Criptocel: Design of cellular automata based cipher schemes. In *Communications, circuits and systems*, volume 2, pages 1103–1107. ICCCAS, IEEE, 2006.
5. Howard Gutowitz. Cryptography with dynamical systems. In E.Goles and N.Boccara, editors, *Cellular Automata and Cooperative phenomena*. Kluwer Academic Press, 1993.
6. Pour la Science: dossier hors srie, editor. *L'Art du Secret*, 2002.
7. Claude Shannon. Communication theory of secrecy systems. *Bell Syst. Tech. Journal*, 28:656–715, 1949.
8. T. Toffoli and N. Margolus. *Cellular automota machines: a new environment for modelling*. MIT Press, 1987.
9. Stephen Wolfram. Cryptography with cellular automata. In *Advances in Cryptology: Crypto85*, volume 218 of *Lectures Notes in Computer Science*, pages 429–432. Springer Verlag, 1986.