

A cryptosystem based on cellular automata

Jesús Urías, Edgardo Ugalde, and Gelasio Salazar

IICO, Universidad Autónoma de San Luis Potosí, 78000 San Luis Potosí, SLP, México

(Received 6 August 1998; accepted for publication 14 September 1998)

Cryptosystems for binary information are based on two primitives: an *indexed* family of permutations of binary words and a generator of pseudorandom sequences of *indices*. A very efficient implementation of the primitives is constructed using the phenomenon of synchronization in cellular automata. © 1998 American Institute of Physics. [S1054-1500(98)01204-X]

Digital secure communications systems are based on permutations of binary words that are selected by following a pseudorandom sequence of choices. The phenomenon of synchronization in cellular automata is used to implement large families of permutations of binary words and to generate pseudorandom sequences of permutations. These two basic elements, the family of permutations and the generator, exist within a single bidimensional array of elementary binary processing units. Such compact arrays fit naturally in the present digital technology and in the emerging technology based on charge-coupled quantum dots.

I. INTRODUCTION

The central elements in secure communications systems are a process for the scrambling of plain information at the transmitting end and a key, that is not feasible to guess, necessary to rebuild the original information at the receiving end. The first cryptosystems based on dynamical systems exploit the mixing character of expansive mappings¹ to scramble up information. Recently, mechanisms for secure communications based on the phenomenon of synchronization in continuous chaotic systems^{2,3} have been investigated. A driver-replica pair of continuous systems is used to provide specific implementations of the following. (i) A method to incorporate the information, whether in the form of a continuous signal^{2,4,5} or a discrete symbolic sequence,^{6,7} into the driver's orbit and (ii) a set of synchronizing coordinates that are communicated to the response system to make it follow very closely the driver's orbit in phase space. In this scheme, the continuous chaotic system is used as a mixer to encrypt the plain information (either a wave form or a symbolic sequence). To disclose information in its original form the parameters of the driving system should be known by the receiver to implement the pertinent chaotic filters. A further approach is to use chaotic systems just to generate pseudorandom wave forms that are then used to scramble the information in standard mixers.⁸

A new mechanism based on synchronization in cellular automata (CA)⁹ is presented here. In contrast to the schemes mentioned above, the synchronizing coordinates in our CA cryptosystem are suited ad hoc and are not just copied from a free running CA. In this way, we are able to implement the necessary primitives to build cryptosystems in a single bidi-

mensional array of binary processing elements, the *unit cipher square* (UCS).

In Secs. II and III we report on the fundamentals and construction of the unit cipher square. We use synchronization to handle the combinatorics of binary sequences and combining it with the expansive nature of the CA mapping we construct a very efficient algorithm to generate pseudorandom sequences of keys. In Sec. IV the CA primitives are discussed from the point of view of cryptography and a simple block cryptosystem is constructed in Sec. V using the CA primitives.

The CA cryptosystem fits naturally in the present digital technology and in the emerging device technology that is being constructed on the basis of charge-coupled quantum dots.¹⁰

II. PRIVATE KEY CRYPTOSYSTEMS

Texts are binary sequences of fixed length N in the set $\{0,1\}^N$. A sequence that has an explicit meaning is called a plain text. The cryptosystem provides a mechanism to transform a plain text sequence \mathbf{m} to an apparently meaningless sequence $\mathbf{c} \in \{0,1\}^N$, called the ciphertext. The transformation $\mathbf{m} \mapsto \mathbf{c}$ is selected from a family of permutations $\Psi = \{\psi_{\mathbf{k}} : \{0,1\}^N \rightarrow \{0,1\}^N | \mathbf{k} \in K\}$ by choosing an index \mathbf{k} from the set of indices K . The cryptosystem also provides the set of inverse permutations $\Phi = \{\phi_{\mathbf{k}} : \{0,1\}^N \rightarrow \{0,1\}^N | \mathbf{k} \in K\}$ such that for every $\mathbf{k} \in K$ one has $\mathbf{m} = \phi_{\mathbf{k}}[\psi_{\mathbf{k}}(\mathbf{m})]$ for every \mathbf{m} .

When two parties want to communicate securely by means of a cryptosystem, they agree to use a key \mathbf{k} , selected from the set of indices K . The emitter encrypts message $\mathbf{m} \in \{0,1\}^N$ as the cipher text $\mathbf{c} = \psi_{\mathbf{k}}(\mathbf{m})$ that is then communicated through an open channel. If the cipher text \mathbf{c} is intercepted it provides no information about the plain text m and provides no clue about the key k . The type of cryptosystem we have described has the property of perfect secrecy, introduced by Shannon.¹¹

In common practice, the plain text to be encrypted is much longer than the length N that is accepted by the permutations in Ψ . In this case the sender proceeds to factor out the long plain text into a succession of blocks $\mathbf{m}^0, \mathbf{m}^1, \mathbf{m}^2, \dots$ each of length N . They are then encrypted sequentially by using a different key \mathbf{k}^i for each block \mathbf{m}^i . The cipher text takes the form of the sequence of blocks $\psi_{\mathbf{k}^0}(\mathbf{m}^0), \psi_{\mathbf{k}^1}(\mathbf{m}^1), \psi_{\mathbf{k}^2}(\mathbf{m}^2), \dots$. The succession of permutations is selected at

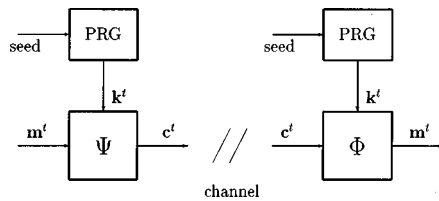


FIG. 1. The general scheme of a private key cipher block cryptosystem. The boxes marked Ψ and Φ represent indexed families of permutations. The key for encryption and decryption is the seed for the generators (PRG) of pseudorandom sequences of indices, (k^t) .

random so as to avoid a third party that intercepts the cipher text to infer any information about the plain text. A practical problem here is that the communicating parties should have to agree on a very long sequence of keys k^0, k^1, k^2, \dots that determine the permutations. The effective key is as long as the plain text.

The problem of very long effective keys is solved by using pseudorandom generators of keys. The encryption and decryption processes use the same deterministic generator that is initialized with a common seed. The communicating parties agree on a given seed through a secure channel. The complete process of a private key cryptosystem is illustrated in Fig. 1. The two primitive building blocks are (i) the families of permutations, Ψ and Φ , and (ii) the pseudorandom generator of keys (PRG). In Sec. III we use the phenomenon of synchronization in cellular automata to construct large families of permutations and a generator of pseudorandom sequences of keys.

III. THE UNIT CIPHER SQUARE

The phenomenon of synchronization in coupled pairs of linear elementary cellular automata is described in detail in Ref. 9. There we showed that a pair of coupled linear elementary CA with local rule $\mathcal{A}_{\mathcal{L}}(x_{-1}x_0x_1) = x_{-1} + x_1 \pmod{2}$ can synchronize if every pair of consecutive coupled coordinates are separated by a block of $2^k - 1$ uncoupled sites, with non-negative k . In the coupled system, the driver's configuration evolves autonomously according to the local

rule $x_i^{t+1} = \mathcal{A}_{\mathcal{L}}(x_{i-1}^t, x_i^t, x_{i+1}^t)$ while the response automaton configuration at a coupled coordinate i evolves according to $y_i^{t+1} = x_i^{t+1}$ and an uncoupled coordinate i follows the rule $y_i^{t+1} = \mathcal{A}_{\mathcal{L}}(y_{i-1}^t, y_i^t, y_{i+1}^t)$.

In Fig. 2 we show an example of evolution patterns of the driver and the response automata along with the discrepancy patterns. Figure 2 shows the evolution of a block of length $N=25$ with coupled coordinates x_0 and x_{24} . The ticks in Fig. 2 mark the coupled coordinates. At time $t=0$ the initial sequences \mathbf{x}^0 and \mathbf{y}^0 are given arbitrarily and for $t \geq 2^4$, the blocks between the two coupled coordinates synchronize, i.e., $x_1^t x_2^t \dots x_{24-1}^t = y_1^t y_2^t \dots y_{24-1}^t$ for $t \geq 2^4$. See the discrepancy pattern in Fig. 2.

The response configuration $\mathbf{y}^t = y_1^t y_2^t \dots y_{24-1}^t$ is guided by the time sequence of values y_0^t and y_{24}^t at the coupled coordinates that are being forced by the driver. The response automaton configurations at times $t \geq 2^4$ are fully determined by the forcing at the end coordinates y_0^t and y_{24}^t .

In general, the system consisting of the evolution rule $\mathcal{A}_{\mathcal{L}}$ acting on binary sequences of length $2^k - 1$ subjected to fixed boundary conditions reaches, at a time t not greater than 2^k , an orbit that depends only on the boundary conditions and is independent of the initial configuration. We use this property to introduce the *unit cipher square* (UCS) as the succession of configurations \mathbf{y}^t from $t=0$ to $t=2^k$, along with the fixed boundary time sequences $\mathbf{k} = y_0^0 y_1^0 \dots y_{2^k-1}^0$ and $\mathbf{c} = y_{2^k}^1 y_{2^k-1}^2 \dots y_{2^k}^{2^k-1}$. The notation \mathbf{k} and \mathbf{c} stands for encryption key and cipher text, respectively. The configuration at time $t=2^k$, $\mathbf{m} = y_1^{2^k} y_2^{2^k} \dots y_{2^k-1}^{2^k}$, at the bottom of the UCS is the plain text or message. The entries at the interior of the UCS and at \mathbf{m} satisfy the condition $y_{i+1}^{t+1} = y_i^t + y_{i+2}^t \pmod{2}$ that is imposed by the automaton local rule. The UCS is exemplified in Fig. 3 for $k=3$.

IV. THE CRYPTOSYSTEM PRIMITIVES

The unit cipher square introduced in Sec. III provides the two primitives for cryptosystems, the pseudorandom generator of keys and the indexed families of permutations Ψ and Φ .

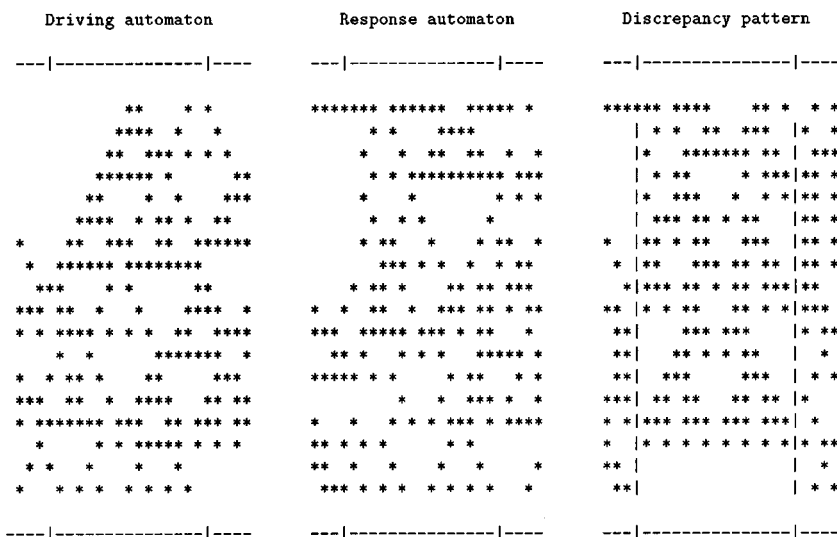


FIG. 2. Patterns for a pair of coupled cellular automata with a synchronizing block of length $2^4 - 1$.

	<i>0</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>1</i>	
0	<i>1</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>
0	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>1</i>
1	<i>0</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>1</i>
0	<i>1</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>
1	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>
1	<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>0</i>
1	<i>1</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>0</i>
	<i>0</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>

FIG. 3. A particular realization of the unit cipher square for $k=3$. The plain and cipher texts are entered in italics, the key block is entered in bold face.

After the discussion in Sec. III about the evolution of a synchronizing pair of CA⁹ we know that the sequence \mathbf{m} in the UCS is determined solely by \mathbf{c} and \mathbf{k} . For each \mathbf{k} the UCS defines a mapping that takes \mathbf{c} to \mathbf{m} . We denote it by $\mathbf{m} = \phi_{\mathbf{k}}(\mathbf{c})$.

On the other hand, giving \mathbf{k} and \mathbf{m} the cipher text \mathbf{c} is generated by applying the CA rule backwards in time and from left to right, $y_{i+1}^t = y_i^t + y_{i-1}^{t-1} \pmod{2}$, starting with \mathbf{m} at time $t=2^k$ up to \mathbf{y}^0 in the UCS. This procedure defines, for every \mathbf{k} , a mapping that takes \mathbf{m} to \mathbf{c} that we denote by $\mathbf{c} = \psi_{\mathbf{k}}(\mathbf{m})$. It is, by construction, the inverse of $\phi_{\mathbf{k}}$.

Due to synchronization, the procedures that perform the permutations in the families Ψ and Φ are independent of the actual configuration \mathbf{y}^0 in the UCS. This extra degree of freedom is used to construct the generator of pseudorandom sequences of keys. Indeed, when the automaton rule is applied backwards in time, the configuration \mathbf{y}^0 that is attained is determined by \mathbf{k} in conjunction with the time sequence $\mathbf{k}' = y_1^0 y_1^1 \dots y_1^{2^k-1}$ that can be fixed arbitrarily without affecting

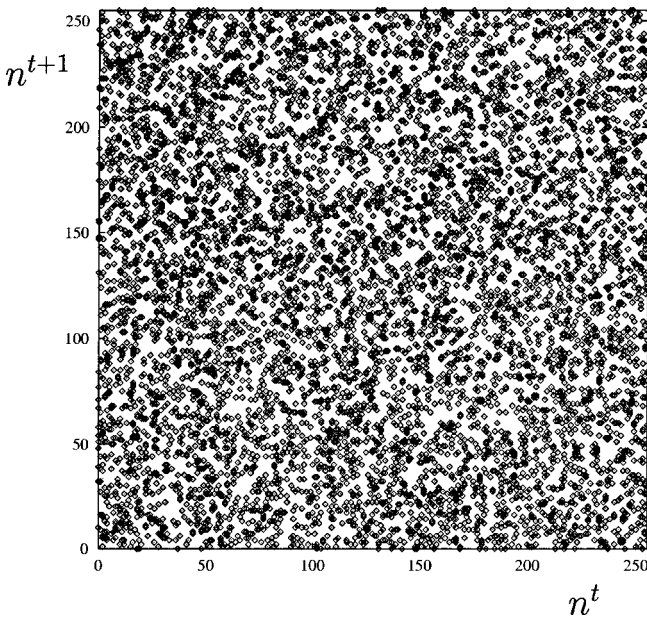


FIG. 4. Plot of pairs (n^t, n^{t+1}) , $t \geq 0$, for a pseudorandom sequence of 8000 keys generated by the function ρ in a CSU of size 63×63 .

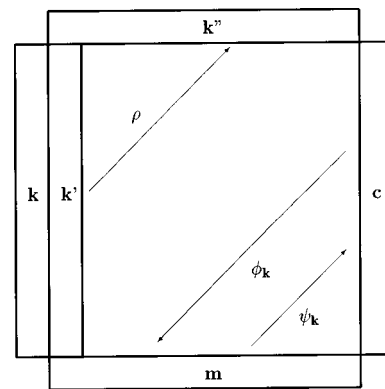


FIG. 5. Primitives defined by the unit cipher square (UCS). The functions ρ and $\psi_{\mathbf{k}}$ are computed by running the automaton backwards in time, from bottom to top in the UCS. The function $\phi_{\mathbf{k}}$ is computed by running it forward, from the top to the bottom.

$\psi_{\mathbf{k}}$ and $\phi_{\mathbf{k}}$. In this way the UCS defines a mapping that takes the pairs of sequences $(\mathbf{k}, \mathbf{k}')$ to the sequence $y_1^0 y_2^0 \dots y_{2^k}^0 = \mathbf{k}''$. The mapping is denoted by $\rho(\mathbf{k}, \mathbf{k}') = \mathbf{k}''$. Given a seed $(\mathbf{k}^0, \mathbf{k}^{-1})$ a sequence of keys $\mathbf{k}^0, \mathbf{k}^1, \mathbf{k}^2, \dots, \mathbf{k}^{t-1}, \mathbf{k}^t, \dots$ is generated iteratively as $\mathbf{k}^{t+1} = \rho(\mathbf{k}^t, \mathbf{k}^{t-1})$.

The performance of ρ as a generator of pseudorandom sequences of keys is exemplified for the case of blocks of length $63 = 2^6 - 1$ in Fig. 4. For every output \mathbf{k}^t , $0 \leq t < 8000$, of the generator we take its central subblock of length 8 and we interpret it as the binary expansion of an integer $0 \leq n^t < 256$. The points in the plot of Fig. 4 correspond to the pairs (n^t, n^{t+1}) . We think that the random aspect of the generated sequence of keys made evident in Fig. 4 is due to the ergodic properties of the automaton rule used in the UCS.^{12,13}

Summarizing, the UCS provides (i) the families Ψ and Φ of 2^{2^k-1} permutations of binary words indexed by all the binary blocks of length $2^k - 1$ and (ii) a pseudorandom generator of indices. These objects implemented in the UCS are illustrated in Fig. 5.

V. A BLOCK CA CRYPTOSYSTEM

To encrypt a binary message \mathbf{M} that is longer than the block size of the UCS we adopt the general block cipher scheme described in Sec. I and illustrated in Fig. 1. In this scheme \mathbf{M} is first factored out into a sequence of blocks of length $2^k - 1$ as $\mathbf{M} = \mathbf{m}^0 \mathbf{m}^1 \mathbf{m}^2 \dots$ and the blocks are then encrypted sequentially using the keys in the sequence $\mathbf{K} = \mathbf{k}^0 \mathbf{k}^1 \mathbf{k}^2 \dots \mathbf{k}^t \dots$ that is produced iteratively by the pseudorandom generator ρ . The schematics of the block cipher scheme using the UCS is shown in Fig. 6. The encrypting mechanism is shown in Fig. 6(E) and the decrypting mechanism in Fig. 6(D). The key to initialize the system is the pair of sequences $(\mathbf{k}^0, \mathbf{k}^{-1})$ that is used as a seed to initialize the pseudorandom generator, shown in Fig. 6 as the box marked ρ . The same seed has to be used for encryption and decryption.

The encryption process is shown in Fig. 6(E) at time step t , when the permutation $\psi_{\mathbf{k}^t}$ is being selected from the family of permutations Ψ by means of the key \mathbf{k}^t . The action of

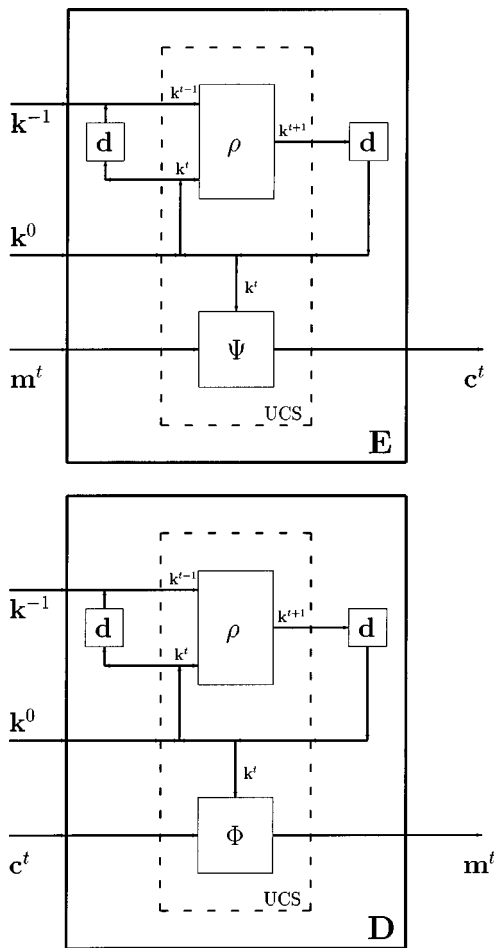


FIG. 6. A cryptosystem based on the unit cipher square. (E) the encryption unit. (D) The decryption unit.

permutating \mathbf{m}^t into $\psi_{k^t}(\mathbf{m}^t)$ is represented in Fig. 6(E) by the box marked Ψ . The input and output blocks at the box of permutations Ψ are the plain text \mathbf{m}^t and the cipher text $\mathbf{c}^t = \psi_{k^t}(\mathbf{m}^t)$.

The generator of pseudorandom sequences is the box marked ρ in the schematics of Fig. 6. For the iteration of ρ we need to store the two previous keys, \mathbf{k}^t and \mathbf{k}^{t-1} . This is indicated in the schematics of Fig. 6 by the delay boxes marked \mathbf{d} , that are external to the UCS. After the encryption process is started, $t > 0$, the two delayed keys are the inputs to the box ρ . The encryption process is initialized at $t = 0$ by giving two seed keys \mathbf{k}^{-1} and \mathbf{k}^0 as inputs for ρ . The seed $(\mathbf{k}^0, \mathbf{k}^{-1})$ plays the role of the private key for the cryptosystem. The two parties have to agree on $(\mathbf{k}^0, \mathbf{k}^{-1})$ prior to the initiation of the communication.

The structure of the decrypting mechanism in Fig. 6(D) is quite similar to that for the encrypting mechanism. However, while the permutation $\psi_{\mathbf{k}}$ in the encryption unit is computed by running the UCS from bottom to top, the permutation $\phi_{\mathbf{k}}$, that is the inverse to $\psi_{\mathbf{k}}$, is computed in the decryption unit by running the UCS from top to bottom, see

Fig. 5. Since the generators of pseudorandom sequences work identically in both units, a single UCS in the decryption unit has to run twice: downwards to decrypt and upwards to generate the next key.

VI. CONCLUDING REMARKS

There are two main advantages in the implementation of the primitives we are proposing. One is that the generator of pseudorandom sequences of keys and the indexed families of permutations are all realized within the same arithmetic device, the UCS. The second advantage resides in the local nature of the automaton rule that corresponds to short wiring when the cryptosystem is implemented as a very large scale integrated device, allowing very high speeds of block encryption. Furthermore, the bidimensional arrays for information processing we have presented fit naturally in the emerging technology of electronic devices that is based on quantum dots, since it fully relies on cellular automata.¹⁰

The statistical characterization of the sequence of keys that are generated by iterating the UCS is a combinatorial problem to be solved.

ACKNOWLEDGMENTS

J.U. and E.U. have benefited from discussions with Valentin Afraimovich during their stay at CPT-Marseilles, financed by the joint program CNRS-CONACyT. J.U. and E.U. are grateful to Ricardo Lima for his hospitality during their stay at CPT-Marseilles. This work received partial support from FAI-UASLP and CONACyT.

¹M. Bianco and D. Reed, U.S. Patent No. 5,048,086, 1991; H. A. Gutowitz, U.S. Patent No. 5,365,589, 1994; V. A. Protopopescu, R. T. Santoro, and J. T. Tolliver, U.S. Patent No. 5,479,513, 1995; O. E. Lefe, Intl. Patent No. WO 97/12330, 1997.

²L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," Phys. Rev. Lett. **64**, 821 (1990).

³L. Kocarev, Z. Tasev, T. Stojanovski, and U. Parlitz, "Synchronizing spatiotemporal chaos," Chaos **7**, 635 (1997).

⁴K. S. Halle, C. W. Wu, M. Ito, and L. O. Chua, "Spread spectrum communications through modulation of chaos," Int. J. Bifurcation Chaos Appl. Sci. Eng. **3**, 469 (1993).

⁵N. J. Corron and D. W. Hahs, "A new approach to communications using chaotic signals," IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **44**, 373 (1997).

⁶S. Hayes, C. Grebogi, and E. Ott, "Communicating with chaos," Phys. Rev. Lett. **70**, 3031 (1993).

⁷C. L. Koh and T. Ushio, "Digital communication method based on M -synchronized chaotic systems," IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **44**, 383 (1997).

⁸R. He and P. G. Vaidya, "Implementation of chaotic cryptography with chaotic synchronization," Phys. Rev. E **57**, 1532 (1998).

⁹J. Urías, G. Salazar, and E. Ugalde, "Synchronization of cellular automata pairs," Chaos (submitted).

¹⁰C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," Proc. IEEE **85**, 541 (1997).

¹¹C. E. Shannon, "Communication theory of secrecy systems," Bell Syst. Tech. J. **28**, 656 (1949).

¹²F. Blanchard, P. Kurka, and A. Maass, "Topological and measure-theoretic properties of one-dimensional cellular automata," Physica D **103**, 86 (1997).

¹³M. A. Shereshevsky, "Ergodic properties of certain surjective cellular automata," Monatsh. Math. **114**, 305 (1992).