2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Conventional cryptography and message-embedding

G. Millérioux[†], A. Hernandez[†] and J.M Amigó[‡]

†Universit´e Henri Poincar´e
Centre de Recherche en Automatique de Nancy (CRAN UMR CNRS 7039)
ESSTIN, 2 Rue Jean Lamour, Vandœuvre-Les-Nancy, France
‡Centro de Investigaci´on Operativa
Universidad Miguel Hern´andez
03202 Elche (Alicante), Spain
Email: gilles.millerioux@esstin.uhp-nancy.fr, jm.amigo@umh.es

Abstract—A lot of encryption methods involving chaotic dynamics have been proposed in the literature since the 90's. Most of them consists in "mixing" the information to be hidden with a chaotic sequence. The recovering of the original information usually calls for reproducing, at the receiver side, the same chaotic sequence. The synchronization mechanism of the two chaotic sequences is known as *chaos synchronization*. In this paper, a connection between chaotic and conventional encryption is carried out with special emphasis concerning one of the most popular scheme, namely the chaotic *message-embedding*.

1. Introduction

Nowadays, communications are electronically processed and information are conveyed along public networks. One of the objectives of cryptography is to preserve the information secrecy from all except the ones the information is intended for, that is privacy and confidentiality. Since the early 1960s, cryptography has no longer been restricted to military or government concerns. Indeed, the advances in digital communications technology has provided a way of designing new efficient encryption schemes. History of modern cryptography found its origin in the works of Feistel at IBM during the years 1970s. One of the key date is the year 1977 when the Data Encryption Standard (DES) has been adopted. Another key date is the year 1978 which has been marked by the discovering of the other well-known encryption scheme named RSA.
Since 1993, a lot of methods involving chaotic systems in order to "hide" an information have been proposed, because these systems can exhibit complex behaviors. The chaotic behaviors can be distinguished by their extreme sensitivity to initial conditions. Thus, the signals resulting from chaotic systems are broadband, long-term unpredictable and present random-like statistical properties although they are generated by deterministic systems. That is why, there is likely a connection between the random-look behaviors exhibited by chaotic systems and the required properties like confusion and diffusion of cryptosystems. A lot of chaos-based methods have been proposed so far. An overview of these different methods can be found in [1]. Nevertheless, very few works (see however [2][3]) have really established the connection between the standard encryption algorithms and those based on the generation of chaotic sequences.
This paper contributes to give a deeper insight by comparing the structures involved in the chaotic and the conventional cryptography schemes.

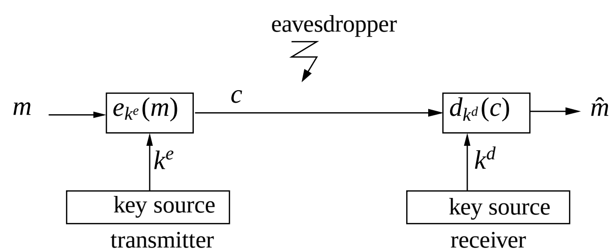

Figure 1: General encryption mechanism

A general encryption mechanism is illustrated on Fig. 1. On the *transmitter* part, a plaintext $m \in \mathcal{M}$ (also called information or message) is encrypted according to an encryption function e which depends on the key $k^e \in \mathcal{K}$. The resulting ciphertext $c \in \mathcal{C}$ is conveyed through a channel to the *receiver*. At the receiver side, the ciphertext c is decrypted according to the decryption function d which depends on the key $k^d \in \mathcal{K}$. The function e (*resp.* d) must be a bijection from $\mathcal{M}$ to $\mathcal{C}$ (*resp.* $\mathcal{C}$ to $\mathcal{M}$). The encryption scheme corresponding to the pair (e, d) must be designed such that it's a hard task for an eavesdropper to retrieve the plaintext m. Thus, there must exist a unique pair (k^e, k^d) such that $d_{k^d}(c) = m$ where $c = e_{k^e}(m)$. Let us pointing out that the design of a cryptographic scheme must takes into account that the sets $\mathcal{M}$, $\mathcal{C}$, $\mathcal{K}$ and the pair (e, d) are known. Only the pair (k^e, k^d) can be assumed to be secret. As a matter of fact, in some special situations, only k^d must be kept secret.

2. Chaotic encryption

There are basically two approaches when using chaotic dynamical systems for "secure" communications purposes (even if the terminology “secure” is sometimes abusively adopted). The first one amounts to numerically computing a great number of iterations of a discrete chaotic system, in using e.g. the message as initial data (see [4] and the references therein). The second one amounts to hiding a message in a chaotic dynamics. Only a part of the state vector (the “output”), which is of weak dimension and ideally unidimensional, is conveyed through the public channel. A synchronization mechanism enables to retrieve the message at the receiver part. The receiver often consists of an observer (also called state reconstructor). This second approach is the one we are interested below through three popular encryption schemes. In this note, we deal with discrete-time systems (maps) and the underscript k is associated to all time-varying quantities.

2.1. Additive masking

This scheme has been suggested for the first time in [5] or [6]. The information m_k to be coded is simply added to the output y_k of the *transmitter* (Fig. 2). The output y_k is a part of the internal state x_k. Unfortunately, there exist some inevitably cases where the information is not exactly retrieved, that is $\hat{m}_k \neq m_k$. Indeed, m_k acts as a perturbation and prevents the *receiver* from being exactly synchronized, that is $\hat{x}_k \neq x_k$ and so $\hat{y}_k \neq y_k$.

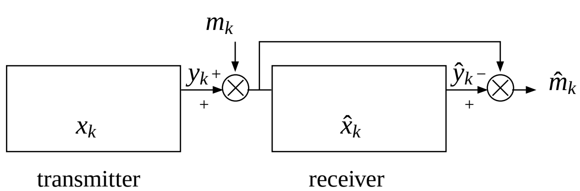

Figure 2: Additive masking

2.2. Two-channel transmission

For a two-channel transmission (Fig. 3), a first channel is used to convey the output $y_k = h_\theta(x_k)$ of a chaotic system described by a dynamics f. h and f are parametrized by θ. Since the chaotic signal is information-free, a perfect synchronization is achieved by, for instance, an observer at the receiver end which ensures $\hat{x}_k = x_k$. Besides, a function e, parametrized by a time-varying quantity, say the state vector x_k of the chaotic system, encrypts the information m_k and produces the ciphertext $u_k = e_{x_k}(m_k)$. Then, the encrypted signal u_k is transmitted via a second channel. At the receiver end, the information m_k can be correctly recovered by the decryption function d. The equality $\hat{m}_k = d_{\hat{x}_k}(u_k) = m_k$ holds provided that $\hat{x}_k = x_k$, which is actually always fulfilled as motivated just above. This technique has been proposed for example in [7][8]. The advantage lies in that, at each discrete time k, m_k can be recovered without any transients. On the other hand, a two-channel transmission may be redhibitory for throughput purposes.

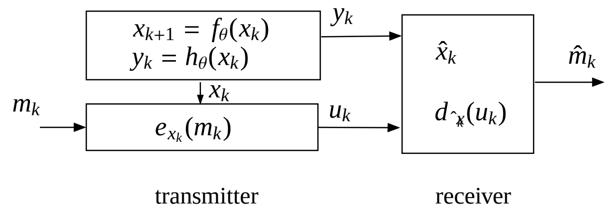

Figure 3: Two-channel transmission

2.3. Message-embedding

The message-embedding technique (Fig. 4) uses, at the transmitter side, the same units as the ones involved in the two-channel transmission but they are combined in a single setup. Indeed, the ciphertext $u_k = e_{x_k}(m_k)$ is not directly conveyed through a channel but is reinjected (embedded) into the chaotic dynamics. Only the output $y_k = h_\theta(x_k)$ of the system, which implicitly depends on u_k and so on m_k, is transmitted. The *receiver* system must be designed such that u_k and x_k can be recovered, given the only available data y_k. Once u_k is recovered, the plaintext m_k is correctly extracted by applying the decryption function d provided that $\hat{x}_k$ is exactly synchronized with x_k. Recently, in [9] and [10], two powerful mechanisms of synchronization, based on unknown input observers, have been proposed to achieve the task. The fact that only a single channel is needed and that the synchronization is guaranteed without restriction on the rate of variation of m_k makes such a scheme very attractive.

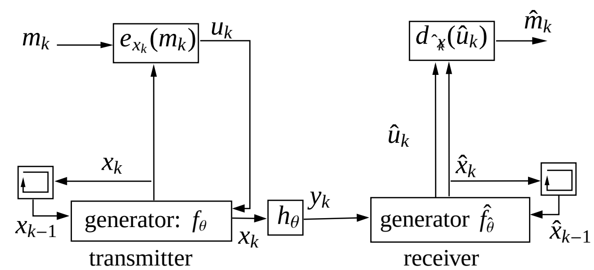

Figure 4: Message-embedding

3. Conventional symmetric cryptography

For details concerning conventional cryptography, the reader can refer to the book of Menezes [11] from which some basics are recalled. Symmetric-key cipher are characterized by an encryption scheme (e_{k^e}, d_{k^d}), whose determination of the key k^d can be easily done from the knowledge of k^e. Usually, both keys are identical, that is $k^d = k^e$. Consequently, not only k^d must be kept secret but the key k^e as well. There exist two distinct symmetric-key encryption schemes : block ciphers and stream ciphers.

A block cipher is an encryption scheme which breaks up the plaintext messages into strings (called blocks) of a fixed length over an alphabet and encrypts one block at a time. Block ciphers usually involve *substitution* ciphers, *transposition* ciphers or *product* ciphers by using composition of these functions.

Stream ciphers involve an encryption which can change for each symbols. There exists two common classes of stream ciphers, one is called synchronous stream cipher (SSC) and the other self-synchronous stream cipher (SSSC). They are respectively illustrated on the Figures 5(a) and 5(b).

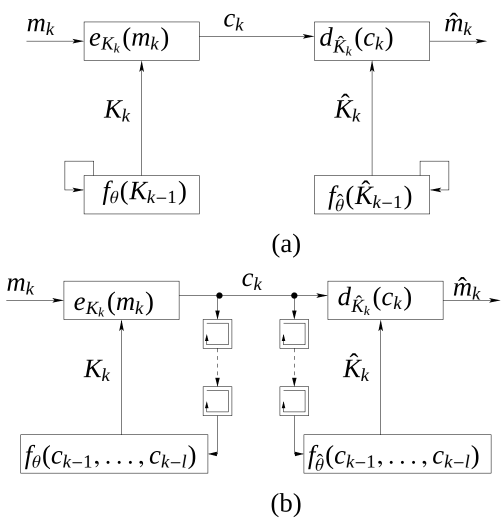

Figure 5: Stream cipher: (a) synchronous, (b) self-synchronous

3.1. Transmitter and encryption

The SSC obeys, at the transmitter side:

$$\begin{cases} K_k = f_\theta(K_{k-1}) \\ c_k = e_{K_k}(m_k) \end{cases} \tag{1}$$

For this encryption scheme, the *plaintext* is divided into blocks of same length, called symbols and denoted by m_k. The encryption function e can change for each symbol because e depends on a time-varying key K_k which is called *keystream*. The keystream K_k is generated by a function f_θ, parameterized by θ acting as the static key.
The SSSC obeys, at the transmitter side:

$$\begin{cases} K_k = f_\theta(c_{k-1}, \ldots, c_{k-l}) \\ c_k = e_{K_k}(m_k) \end{cases} \tag{2}$$

f_θ is also a function parameterized by θ, and generates the keystream K_k. Unlike the SSC, K_k does not depend on an internal dynamics but only on a fixed number l of past values of c_k. However, as previously, c_k is generated by the encryption function e with time-varying key K_k.

3.2. Receiver and reconstruction of the plaintext

The reconstruction of the plaintext requires the synchronization of the sequences K_k and $\hat{K}_k$ at both the transmitter and the receiver ends. The decryption is described, in the SSC case, by:

$$\begin{cases} \hat{K}_k = f_{\hat{\theta}}(\hat{K}_{k-1}) \\ \hat{m}_k = d_{\hat{K}_k}(c_k) \end{cases} \tag{3}$$

and, in the SSSC case, by:

$$\begin{cases} \hat{K}_k = f_{\hat{\theta}}(c_{k-1}, \ldots, c_{k-l}) \\ \hat{m}_k = d_{\hat{K}_k}(c_k) \end{cases} \tag{4}$$

In both cases, the decryption function d is such that $\hat{m}_k = m_k$ if $\hat{K}_k = K_k$. For the SSC, the sequences K_k and $\hat{K}_k$ resulting from autonomous recurrences, the key generators f_θ at both sides have to be initialized at the same value ($\hat{K}_0 = K_0$). K_0 acts as the static key, that is $\theta = K_0$. At the contrary, for the SSSC, the sequences synchronize automatically.

4. A comparative study

A major and obvious difference between chaotic encryption and conventional cryptography lies in the fact that a chaotic generator is assumed to produce an aperiodic sequence ranging in a dense set while symmetric conventional cryptography involves pseudo-random generators which produce discrete sequences. Nevertheless, when implemented in a machine with finite accuracy, the sequences $\{x_k\}$ and $\{y_k\}$ are not really chaotic but "pseudo-chaotic". Indeed, the cardinality of the set where they take values being finite, the sequences will obviously get trapped into a loop, called *cycle*, of finite period. We can expect this period to be not too short and the degree of "randomness" of the sequence to be high but that requires some deep cautions to guarantee those properties. Some important studies related to this issue can be found in [12][13]. Here, we rather focus on the structure of the proposed setups for the comparative study.

Additive masking : A natural connection can be made between the additive masking and the SSC. Indeed, the transmitter of the respective schemes has exactly the same structure. The sequences x_k (*resp.* K_k) are independent from the plaintext m_k and the ciphertext u_k (*resp.* c_k). For a SSC, a same initialization is required at both ends to guarantee the synchronization. For the additive masking, assuming that the generator is really chaotic, due to the sensitivity property with respect to initial conditions, synchronization is inevitably lost on a very short horizon time. In the literature, to handle such a problem, a controlled synchronization usually based on observers is often suggested at the transmitter part. Nevertheless, as previously mentioned, the added information to be masked acts as a perturbation and prevents the control to guarantee an exact synchronization. That renders such a scheme not very appealing compared with a conventional SSC.

Message-embedding : The structure combines the specificities of both the SSC and the SSSC. Indeed, as K_k in the SSC, the keystream x_k is produced by a recursion and is a dynamical quantity. Furthermore, as in the SSSC, the ciphertext is reinjected into the dynamics. On the other hand, the message-embedding is distinguished by the fact that the ciphertext is not directly conveyed through the channel but transmitted implicitly via the output y_k of the system. That induces a drastic difference as for the way of recovering the plaintext. For SSC or SSSC, the receiver is a copy of the transmitter ($f_\theta = f_{\hat{\theta}}$). For the message-embedding, the receiver must compute u_k from the knowledge of y_k while the transmitter produces y_k for a given u_k. Thus, the receiver performs the inverse operation. That's why distinct notation f and $\hat{f}$ has been adopted in Fig.4. Besides, the inversion is carried out although the respective internal vectors x_k and $\hat{x}_k$ are not initialized at the same value since the synchronization between both ends is controlled. As a result, unlike K_0 for SSC, x_0 cannot play the role of the static key for the message-embedded technique. The static key is vector of the parameters involved in the dynamics f (and sometimes h). Note that the system inversion issue has been first addressed in [14].

The message-embedded scheme seems to bring together many advantages. Some of them are inherited from the SSC and SSSC schemes *i)* and *ii)* and others are specific *iii)* and *iv)*.

i) The reinjection of the ciphertext into the dynamics induces a spread of the plaintext. In other words, unlike for SSC, a ciphertext does not depend only on the plaintext but also on the past values and contributes to the diffusion.

ii) It is robust against loss of synchronization. Indeed, the synchronization is controlled and can be guaranteed with a prescribed finite transient time, which limits the propagation error similar to the self-synchronizing scheme. On the other hand, several techniques such as inserting markers in the ciphertext are required for SSC to restore the synchronization if it is lost.

iii) Since the synchronization of the running key sequences is controlled, the same initialization at both sides is no longer needed. It follows that, to a same plaintext, may correspond different ciphertexts according to the initial value of the keystream, which contributes to an increase of the confusion.

iv) The scheme seems to be more robust against known plaintext attack. Indeed, it is recalled that this technique usually consists in choosing a segment of the plaintext m_k and in analyzing the corresponding ciphertext c_k. And yet, in a message-embedded technique, the ciphertext is not directly transmitted through the channel, only the output y_k is available, rendering a known plaintext attack harder.

Conclusion: Based on the above structural analysis, message-embedding seems to be a promising technique. But claiming that it could be an alternative to SSC or SSSC would deserve more thorough cryptanalytic works.

References

[1] T. Yang. "A survey of chaotic secure communication systems". *Int. J. of Computational Cognition*, 2004. (available at `http://www.YangSky.com/yangijcc.htm`).

[2] F. Dachselt, K. Kelber, J. Vandewalle, and W. Schwarz. "Chaotic versus classical stream ciphers – a comparative study". In *Proc. of Int. Symp. on Circuits and Systems ISCAS'98*, volume IV, pages 518–521, Monterey, June 1998.

[3] L. Kocarev. "Chaos-based cryptography :a brief overview". *IEEE Circuits and Systems Magazine*, 1(3):6–21, 2001.

[4] R. Schmitz. "Use of chaotic dynamical systems in cryptography". *Journal of the Frank. Inst.*, 338:429–441, 2001.

[5] Cuomo K. M., Oppenheim A. V., and Strogatz S. H. "Synchronization of lorenz-based chaotic circuits with applications to communications". *IEEE Trans. Circuits. Syst. II: Anal. Digit. Sign. Process*, 40(10):626–633, 1993.

[6] Wu C. W. and Chua L. O. "A simple way to synchronize chaotic systems with applications to secure communications systems". *International Journal of Bifurcation and Chaos*, 3(6):1619–1627, 1993.

[7] G. Millerioux and C. Mira. "Coding scheme based on chaos synchronization from noninvertible maps". *International Journal of Bifurcation and Chaos*, 8(10):2019–2029, 1998.

[8] Jiang Z-P. "A note on chaotic secure communication systems". *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 49(1):92–96, January 2002.

[9] G. Millerioux and J. Daafouz. "An observer-based approach for input independent global chaos synchronization of discrete-time switched systems". *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, pages 1270–1279, October 2003.

[10] G. Millerioux and J. Daafouz. "Unknown input observers for message-embedded chaos synchronization of discrete-time systems". *International Journal of Bifurcation and Chaos*, 14(4):1357–1368, April 2004.

[11] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, October 1996.

[12] J. Szczepanski, J.M. Amig´o, T. Michalek, and L. Kocarev. "Crytographically secure substitutions based on the approximation of mixing maps". *IEEE Trans. Circuits and Systems I : Regular Papers*, 52(2):443–453, February 2005.

[13] J.M. Amig´o, J. Szczepanski, and L. Kocarev. "A chaos-based approach to the design of crytographically secure substitutions". *Phys. Lett. A*, 343:55–60, February 2005.

[14] Feldmann U., Hasler M., and Schwarz W. "Communication by chaotic signals :the inverse system approach". *Int. J. of Circuit Theory Appl.*, 24:551–579, 1996.