Chaos-Based Cryptography: End of the Road?

Daniel-Ioan Curiac, Daniel Iercan
"Politehnica" University
Timisoara, Romania
daniel.curiac@aut.upt.ro

Octavian Dranga
James Cook University,
Townsville, Australia
octavian.dranga@jcu.edu.au

Florin Dragan, Ovidiu Banias
"Politehnica" University
Timisoara, Romania
florin.dragan@aut.upt.ro

Abstract

Chaos-based cryptography emerged in the early 1990s as an innovative application of nonlinear dynamics in the chaotic regime. Even if in theory chaotic dynamics was thought to evolve into a new revolution in cryptography, in real-life an efficient and reliable chaos-based cryptosystem didn't emerge. The main but not the only reason is the dynamic degradation of digital chaotic systems, a subject that became very popular in the last few years. This paper presents a new theoretical background related to this issue that proves the inefficiency of chaos-based encryption algorithms. Even more, in one of the two relevant case studies presented, another myth is demolished: the analog encryption base on synchronized chaos.

1. Introduction

In the last two decades the use of chaos in cryptography has been a growing interest. Although conceptually the solutions proposed by the professional community seem correct, serious problems can arise when implementing chaotic cryptographic systems. What are the causes for these problems? Answering this question is not an easy task at all. First let us start by having a look at the evolution of chaotic cryptosystems.

The use of chaos in cryptography has emerged as a potential solution to many problems due to the following fundamentals of chaotic systems: (a) chaotic systems are determinist dynamical systems; and (b) chaotic systems exhibit sensitive dependence on initial conditions.

The suggested chaotic cryptosystems can be divided in two main categories:

(a) Cryptosystems that use the idea of synchronized chaos, initially developed by L.M. Pecora and T.L. Carroll [1]. From the implementation point of view there are two types of synchronized chaotic systems:

(1) synchronized chaotic systems which are implemented on digital equipment, and

(2) synchronized chaotic systems that are implemented on analog equipment.

(b) Cryptosystems that do not use synchronized chaos. Examples of such cryptosystems are the Baptista-cryptosystem [2] and the Alvarez-cryptosystem [3].

Nevertheless employing chaotic dynamics in order to develop secure cryptosystems proved to be a difficult task. All the cryptosystems mentioned above has been shown to have some weaknesses, e.g., attacks on Baptista-type systems were presented in [4], [5], [6] and [7]. However, the most important problem is that trying to generate a chaotic signal on a finite precision machine leads to dynamical degradation of chaotic properties. The problem of dynamical degradation has already been addressed in [8], [9] and [10], but as we will show in the next section, the degradation is more dramatic than what has been presented in the literature so far. Intuitively, when a chaotic signal is generated on a finite-precision machine, the state space does not provide an infinity of possible states anymore, thus a state trajectory will pass through the same point more than once, which makes the trajectory to lose its chaotic properties.

The main source for all the problems encountered when implementing a chaotic cryptosystem is represented by the lack of serious investigation of discrete-time chaotic systems when implemented on digital equipments. We are going to show that chaotic systems cannot be implemented on digital computers without losing the property of being chaotic. As a result, the implementation of a chaotic system is pseudo-chaotic at most and not chaotic. Since all analog equipments have a certain tolerance, we are going to show that the problem of chaos degradation is true in this case also.

0-7695-2989-5/07 $25.00 © 2007 IEEE
DOI 10.1109/SECURWARE.2007.25

In the following paragraphs we will prove why there can not be chaos when finite precision is considered. After that, we will present what are the side effects of implementing chaotic cryptosystems on finite precision equipments, together with the related limitations of chaotic cryptosystems. In the end we will discuss two case studies one for each chaotic cryptosystem category.

2. Non-existence of discrete chaos in finite-length implementations

In this section we will prove that implementing a chaotic system on a finite precision machine will make the system entirely lose its chaotic behavior. Thus the chaotic systems implemented on finite precision machine are not chaotic anymore, but pseudo-chaotic systems.

A fundamental theorem in the chaos theory is the following:

The no-intersection theorem: Two distinct chaotic state space trajectories cannot intersect each other (in a finite period of time). Nor can a single chaotic trajectory cross itself at a later time [11].

The no-intersection theorem is valid only for autonomous chaotic system; this does not limit our results to only autonomous systems, since one can easily extrapolate our results to non-autonomous systems.

Theorem of non-existence of discrete chaos in finite-length implementations: A chaotic state space trajectory cannot be generated on a finite-precision machine.

Proof: Given a chaotic trajectory T, one of the properties of T is that it is bounded, e.g., there exists a bounded subspace S so that $T \subset S$, and *as the no-intersection theorem* says T will not pass through the same point twice. Thus to prove the theorem it is enough to show that if S is bounded and finite precision is considered, then the cardinality of S is finite. Next we will prove using mathematical induction that a bounded n-dimensional subspace S has a finite cardinality if finite precision is considered.

The basis: We consider that S is a one-dimensional bounded subspace, i.e. S is a bounded interval. If infinite precision is considered then S has an infinite cardinality, but if finite precision is considered then S has a finite cardinality.

The inductive step: We consider that a bounded n-dimensional subspace S has finite cardinality if finite precision is considered. Given an $(n+1)$-dimensional subspace $S`$, we can decompose $S`$ as the product of an n-dimensional and a one-dimensional subspace. Since both the n-dimensional space and the one-dimensional space have a finite cardinality, if finite precision is considered, then $S`$, which is the product of these two finite sets, also has finite cardinality if finite precision is considered.

As shown above, an n-dimension bounded subspace has an infinite number of points as long as the precision is infinite, but if a finite precision is considered then the number of points in S is also finite. Thus if we consider a chaotic trajectory T that was generated on a finite precision machine, it is not possible for this trajectory to be bounded and not to cross itself at the same time, since the number of points in a bounded subspace is finite in case of finite precision, therefore T contradicts *the no-intersection theorem*. Thus although T might exhibit a chaotic behavior in the beginning, it is not chaotic, but pseudo-chaotic, provided it was generated on a finite precision machine.

As a consequence of this theorem, the finite precision can cause two nearby pseudo-chaotic trajectories to intersect when they get very close to each other. Similarly, in the case of a singular trajectory, it will become periodic due to the finite precision. These scenarios are shown in Figure 1**,** for four state trajectories T, $T`$, $T``$, $T```$, which are assumed to be chaotic when infinite precision is considered. However, as shown in the figure, if the four trajectories are generated on a finite-precision machine, trajectory T will become periodic and trajectories $T`$, $T``$, $T```$ will intersect T in different points and, as a result, will also become periodic.

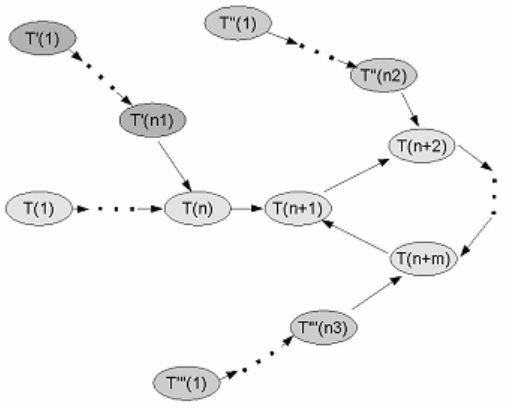

Figure 1. State trajectory intersection due to finite precision.

Similarly, Figure 2 presents the possible scenario for a pseudo-chaotic trajectory T, which due to the fact that was generated in a finite precision environment, falls into an attractor. The degradation of the chaotic dynamics is due to the fact that at some point the trajectory T will be very close to an attractor and since

finite precision is considered it is possible for the trajectory to fall in that attractor and never leave it.

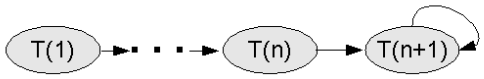

Figure 2. Falling into attractor.

3. Chaotic cryptography and the theorem of non-existence of discrete chaos in finite-length implementations

The chaotic systems can be divided in two main categories: (a) continuous-time chaotic systems, and (b) discrete-time chaotic systems.

When a continuous-time chaotic system is implemented on digital equipment, it will be discretized both spatially and temporally. The time discretization can be: (a) implicit, i.e. the chaotic system is realized in a direct form, under fixed-point or floating-point precision, and (b) explicit, i.e. the equation system is re-defined in digital form. The spatial discretization is always implicit.

When a discrete chaotic system is implemented on digital computer, although there is no need for a temporal discretization, the spatial discretization will still be implicit.

As a conclusion, no matter whether the cryptosystem is discrete or continuous when it will be implemented on a digital computer it will be implicitly spatially discretized. Thus, as we have proved in the previous section, a chaotic system loses its property of being chaotic when it is digitally implemented because the space is discretized (e.g., its cardinality is reduced to a finite number of points from an infinite one).

There have been proposed different solutions to improve the digital pseudo-chaos, but none of these methods can solve the problem. These methods are:
- using higher finite precision [12], [13];
- cascading multiple chaotic systems [14];
- using perturbation-based algorithm [15], [16], [17].

Using higher finite precision does not solve the problem at all, it just makes some pseudo-chaotic trajectory to be longer, but they will still become periodic, and the worst case (falling in attractor) can still happen. Higher precision only increases the granularity of the bounded sub-space, so that there will be more possible states, but the number of states will remain finite.

Cascading multiple chaotic systems consists of interconnecting two pseudo-chaotic systems. This method again only increases the length of the cycle of some (most of) pseudo-chaotic orbit, but does not make the orbit chaotic. The two cascaded systems will be implemented on finite precision machines. This method just increases the bounds of the bounded sub-space, but the problems still remain.

The perturbation-based algorithm consists of using a second system to perturb the pseudo-chaotic orbits. This solution is also far from being perfect. First the pseudo-random generator will be implemented on the same finite-precision machine, which can make this system to run into the same problem as the chaotic one. Second the perturbation of a trajectory will do nothing more but to switch to a different trajectory which in the end will lead to the same problems, plus there is no guarantee that at some time the perturbation will not make the trajectory to jump exactly on a closed loop. Third the key for the new cryptosystem will have to contain the parameters which describe the used pseudo-random generator, which will make the key to become even bigger.

With all this limitations it is obvious that pseudo-chaotic cryptosystems have to be used carefully, only for short messages and only when the information is valid a short period of time. We proposed the use of an off-line keys validation, which will ensure that a message of a given length will be encrypted.

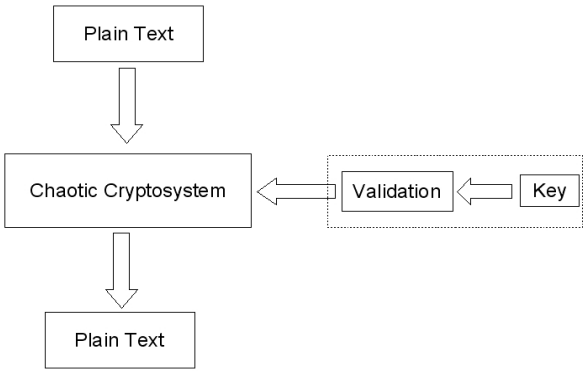

Figure 3. Off-line key validation.

In Figure 3 it can be see that we have introduced the “Validation” block between the “Key” and “Chaotic Cryptosystem” blocks. This block will check the keys so that one can be sure that a message of a given length will be encrypted no matter what its content will be, using the specified crypto-system. Obviously this solution will reduce the key space and will weaken the cryptosystem against brute force attack, however, if the encrypted message has a short life time (a few hours) the cryptosystem could still be useful.

In the last part of the paper we will present two case studies regarding the useless of chaos in

cryptosystems, taking into consideration both categories of cryptosystems.

4. Case study

4.1. Baptista-type cryptosystems

The Baptista-type cryptosystem is based on partitioning the visiting interval of chaotic orbits of the logistic map and associating each partition to a character. The encryption of a character consists of iterating the chaotic system until the partition associated with the character is reached and counting the number of iterations. The encoding of the character will be represented by the number of iterations. The decoding of a character consists of iterating the same chaotic system as in the case of encryption, starting from the same initial condition for the number of times indicated by the encoding. The partition that was reached in the last iteration will give us the decoded character.

First let us have a look at the original Baptista-type cryptosystem [2], [10]: *BC = (F, K, A_e, A_d)*, where *F* is a chaotic map $F : X \to X$, *K* is a secret key, A_e is an encryption algorithm and A_d is a decryption algorithm. The chaotic map *F* is represented by the logistic map $F(x) = bx(1-x)$, where $b \in R^{+*}$ is a parameter. It is well known that the logistic map exhibits chaotic behavior for $b \in [3.57, 4]$. The key $K = (f_S, x_0, \beta)$, with f_S a bijective function:

$f_S : X_\varepsilon = \{X_1, X_2, \ldots, X_S\} \to A = \{\alpha_1, \alpha_2, \ldots, \alpha_S\}$,

where $S \in N^*$

In order to define the subintervals X_i, $i = \overline{1, S}$, a new interval $X' \subseteq X$ is defined as $X' = [x_{\min}, x_{\max})$, then X' is divided in *S* subintervals $X_i = [x_{\min} + (i-1)\varepsilon, x_{\min} + i\varepsilon)$, where $\varepsilon = \frac{x_{\max} - x_{\min}}{S}$ and $i = \overline{1, S}$. *A* is a set of characters, x_0 is the initial condition for the chaotic map and β is the value of the parameter *b*. Another function f'_S is defined as: $f'_S : X \to A \cup \{\beta\}$, where $\beta \notin A$ and

$$f'_S(x) = \begin{cases} f_S(X_i), & x \in X_i \\ \beta, & x \notin X' \end{cases}$$

The encryption algorithm A_e for a message $M = m_1 m_2 \ldots m_i \ldots$, where $m_i \in A$ consists of the following steps:

- take each character $m_i \in T$;
- compute the initial condition $x_0^{(i-1)} = F^{C_1 + C_2 + \ldots + C_{i-1}}(x_0)$, $x_0^{(0)} = x_0$;
- iterate the chaotic map *F* starting from initial condition $x_0^{(i-1)}$ until a state x will be reached so that $f'_S(x) = m_i$; the number of iterations C_i represents the encoding of m_i.

The decryption algorithm A_d for a message $M = m_1 m_2 \ldots m_i \ldots$, where $m_i \in A$ is represented by the following steps:

- take each encoding C_i of a character m_i;
- iterate the chaotic map F for C_i times starting from initial condition $x_0^{(i-1)}$; the initial condition is computed as in the case of A_e, the value $x_0^{(i)} = F^{C_1 + C_2 + \ldots + C_i}(x_0)$ obtained after iterating the map F will be used to find the decrypted character $m_i = f'_S(x_0^{(i)})$.

The result of encoding a character C_i should satisfy the restriction $C_i \in [250, 65532]$.

The Baptista-type cryptosystem is far from being perfect. It is a relatively slow cryptosystem due to the large number of iterations needed to encrypt a character and because the size of the encrypted message is greater then the size of the original text. Another problem is that the distribution of the encrypted text is not uniform [2]. There are also some security issues concerning the Baptista cryptosystem. [3], [4], and [10] lists a few more defects of this cryptosystem:

- problems regarding the logistic map used for encryption:
  - non-uniform visiting probability of each ε-interval: this problem is due to the fact that the logistic map has a non-uniform invariant density function [2]
  - limits on the control parameter *b*: to ensure that the generated pseudo-chaotic orbit is pseudo-chaotic enough, the parameter *b* has to be close enough to 4, which limits the range of values that *b* can take.

- problems regarding the secret key: including f_S in the secret key will make it difficult to be remembered by a user (in [10] it is suggested that an algorithm should be used to generate this function)
- dynamic degradation of digital chaotic systems
- an obvious problem arises when $C_i > 65532$.

As it can be seen, the problem of dynamic degradation of digital chaotic systems has been mentioned in [10], but *the problem is even more dramatic than presented* there. The reason is that, a chaotic system can not be simulated on a finite-precision machine. Thus implementing the Baptista-type cryptosystem on a finite-precision machine, as in the real case, will make the cryptosystem unusable, since it is very hard to find an initial state x_0 and a function f_S so that for any possible text message (in terms of length and number of similar characters) as an input to the system, the cryptosystem does not run intro troubles due to non-existence of discrete chaos in finite-length implementations. There is no guarantee that the trajectory used to encrypt a message will not end up on a closed loop or even in an attractor due to rounding problems, as shown in the previous section. Another issue could be that due to the same rounding problem the trajectory will not visit all small subintervals X_i'. The reason is that in the end all the trajectories become periodic, thus it is possible for a trajectory to become periodic before it visits all the cells. The longer the message to be encrypted, the more important this problem is.

4.2. Synchronized cryptosystems

The synchronization is realized by using two "very-similar" chaotic systems: subsystem *D* which is the drive subsystem and subsystem *R* which is the response subsystem. The drive subsystem is used to generate a chaotic signal to which the plain message is added, the resulting signal will be sent to the receiver. On the receiver side the signal is used as input for the response subsystem, and the output of this subsystem is subtracted from the received signal. The result will be the decrypted message. The main idea behind this method is that the behavior of chaotic systems *R* and *D* are exactly the same with an infinite precision. An overview on the synchronized chaotic cryptosystems can be found in [19]. In Figure 4 we present a synchronized chaotic cryptosystem which is based on the Lorenz chaotic system.

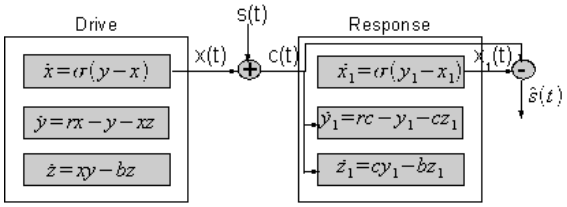

Figure 4. Synchronized chaotic cryptosystem using Lorenz chaotic system.

As it was presented in the Introduction, the synchronized cryptosystems were implemented both on analog devices and on digital computers. In case of digital computers the problem is that they have finite precision so the chaotic behavior disapeare, while in the analog case the precision is still finite, due to the fact that each analog device has a certain class of precision. Moreover, in practice, one analog device cannot be replicated exactly (with infinite precision). This means that *R* is not exactly a copy of *D*, thus the decryption is not done correctly.

As in the digital case, in the analog case the orbits will either become periodic or will fall into an attractor, due to the finite precision of the analog devices. Imagine what would happen if the chaotic system on which a cryptosystem is based will fall into attractor, the encrypted message will be computed as an exclusive OR between the plain text and a constant number.

5. Conclusion

In this paper we have shown that a chaotic cryptosystem becomes a pseudo-chaotic cryptosystem when it is implemented on a finite precision machine, even more we have shown that not even the analog implementation is good enough. We have proved that once a chaotic system is implemented on a digital computer, it will not be a chaotic system anymore. We have also presented what are the limitations when using a pseudo-chaotic cryptosystem.

6. References

[1] L. M. Pecora, and T. L. Carroll: "Synchronization in Chaotic Systems", Phys. Rev. Lett. 64, 1990, pp. 821–825.

[2] M. Baptista: "Cryptography with chaos", Phys. Lett. A 240 (1-2), 1998, pp. 50–54.

[3] E. Alvarez, A. Fernandez, P. Garcia, J. Jimenez, and A. Marcano: "New approach to chaotic encryption", Physics Letters A 263, 1999, pp. 373–375.

[4] G. Jakimoski, and L. Kocarev: "Analysis of some recently proposed chaos-based encryption algorithms", Physics Letters A 291 (6), 2001, pp. 381–384.

[5] G. Alvarez, F. Montoya, M. Romera, and G. Pasto: "Cryptanalysis of an ergodic chaotic cipher", Physics Letters A 311 (2-3), 2003, pp. 172–179.

[6] G. Alvarez, F. Montoya, M. Romera, and G. Pastor: "Keystream cryptanalysis of a chaotic cryptographic method", Computer Physics Communications 156 (2), 2003, pp. 205–207.

[7] G. Alvarez, F. Montoya, M. Romera, and G. Pastor: "Cryptanalysis of dynamic lookup table based chaotic cryptosystems", Physics Letters A 326 (3-4), 2004, pp. 211–218.

[8] S. Li: "Analyses and new designs of digital chaotic ciphers", Ph.D. thesis, School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, available online at http://www.hooklee.com/pub.html, 2003.

[9] S. Li, G. Chen, and X. Mou: "On the dynamical degradation of digital piecewise linear chaotic maps", accepted by Int. J. Bifurcation and Chaos, preprint available online at http://www.hooklee.com/pub.html, 2004.

[10] S. Li, G. Chen, K. W. Wong, X. Mou, and Y. Cai: "Baptista-type chaotic cryptosystems: Problems and countermeasures", 2004.

[11] R. C. Hilborn: "Chaos and nonlinear dynamics, Oxford Universit Press", 1994;

[12] D.D. Wheeler: "Problems with chaotic cryptosystems", Cryptologia XIII, 1989, pp. 243–250.

[13] D. D. Wheeler, and R. A. J. Matthews: "Supercomputer investigations of a chaotic encryption algorithm", Cryptologia XV, 1991, pp. 140–151.

[14] G. Heidari-Bateni, C. D. McGillem: "A chaotic direct-sequence spread-spectrum communication system", IEEE Trans. Communications 42, 1994, pp. 1524–1527.

[15] H. Zhou, and X. Ling: "Realizing finite precision chaotic systems via perturbation of m-sequences", Acta Eletronica Sinica 25, 1997, pp. 95–97.

[16] J. Cermak: "Digital generators of chaos", Physics Letters A 214, 1996, pp. 151–160.

[17] T. Sang, R. Wang, and Y. Yan: "Clockcontrolled chaotic keystream generators", Electronics Letters 34, 1998, pp. 1932–1934.

[18] S. Li, G. Chen, and X. Mou: "On the Dynamical Degradation of Digital Piecewise Linear Chaotic Maps", International Journal of Bifurcation and Chaos in vol. 15, no. 10, 2005. pp. 3119-3151.

[19] G. Alvarez, G. P. F. Monotoya, and M. Romera: "Chaotic cryptosystems", in: Proc. IEEE 33rd Annual Int. Carnahan Conf. Security Technology, IEEE, 1999, pp. 332–338.