Chaos-based Pseudo-Random Number Generators and Chip Implementation

Zhong Li [1], Ping Li [1], Yaobin Mao [2] and W.A. Halang [1]

[1] *Faculty of Electrical and Computer Engineering*
FernUniversität in Hagen, 58084 Hagen, Germany
E-mail: zhong.li@fernuni-hagen.de

[2] *Department of Automation*
Nanjing University of Science and Technology
Nanjing 210094, P. R. China

Abstract: Cryptography as an ancient subject is endowed with new vigor by chaos theory. Cryptography protects the security of today's ubiquitous Internet communication, which as an open network is vulnerable to attack. In this paper, chaos-based cryptography is surveyed with focus on designing chaotic pseudo-random number generators (CPRNGs) for stream cipher and their chip implementation. The properties of the proposed CPRNG are analyzed.

Keywords: Chaos-based pseudo-random number generator (CPRNG), cryptography, chaos, chip implementation.

1. Introduction

Cryptography, defined as the science and study of secret writing, concerns the ways in which communications and data can be encoded to prevent disclosure of their contents through eavesdropping or message interception, using codes, ciphers, or other methods, so that only certain people can see the real message. The science of cryptography is very old, and can be traced to Ancient Egypt. From Julius Caesar to Mary, Queen of Scots to Abraham Lincoln's Civil War ciphers, cryptography has been a part of the history. At that time, cryptography was concerned only by those associated with the military, the diplomatic service and government in general, and was used as a tool to protect national secrets and strategies.

Nowadays, Internet has become an indispensable part of our daily life. However, over the Internet various communications, such as E-mails, or the use of WWW browsers, are not secure for sending and receiving information. Therefore, varieties of cryptographic methods have been proposed to secure Internet communication. For instance, the Data Encryption Standard (DES) was adopted as a U.S Federal Information Processing Standard for encrypting unclassified information. Others include IDEA (International Data Encryption Algorithm), and RSA (developed by Rivest, Shamir and Adleman), these encryption algorithms are based on number theory. However, none of them is absolutely secure.

Cryptography can be strong or weak. Cryptographic strength is measured in the time and resources, which would require to recover the plaintext. A strong cryptography makes ciphertext difficult to be deciphered without possession of the appropriate decoding tool. In other words, given all of today's computing power and available time - even a billion computers doing a billion checks in a second, it is still impossible to decipher the result of strong cryptography before the end of the universe. Straightforward, one would think that even that strong cryptography would hold up rather well against even an extremely determined cryptanalysis. Nevertheless, no one can prove that the strongest encryption obtainable today will hold up under tomorrow's computing power. Therefore, some emerging theories, such as chaos theory, can be adopted to strengthen the existing cryptography.

The reason of applying chaos theory in cryptography lies in its intrinsic essential properties, such as the sensitivity to initial conditions (or control parameters) and ergodicity, which meet Shannon requirements of confusion and diffusion for cryptography. Shannon wrote in his seminal paper (Shannon 1949): In a good mixing transformation … functions are complicated, involving all variables in a sensitive way. A small variation of any one (variable) changes (the outputs) considerably. An important difference between chaos and cryptography lies on the fact that systems used in chaos are defined on real numbers, while cryptography deals with systems defined on finite number of integers. Nevertheless, it is believed that the two disciplines can benefit from each other (Baptista 1998).

Chaotic cryptosystems can be analog or digital. The analog ones are based on chaotic synchronization technique, which was proposed in (Pecora 1990), to design analog circuits for secure communications via noisy channels. But this can not be extended to design modern cryptographic algorithms implemented with digital techniques (Frey 1993). The digital chaotic ciphers can be categorized into stream ciphers and block ciphers. Stream ciphers employ chaotic systems to generate pseudo-random keystream to mask plaintext, while block ciphers use the plaintext and/or the secret keys multiple times to obtain ciphertext. In addition, some other chaotic encryption schemes have also been proposed and tested (Kotulski 2000,

Schneier 1996, Szczepanski 2000, Wong 2001, Zhou 1997). In this paper, we focus on designing chaotic pseudo-random number generators (CPRNG) with chip implementation, because it turns out that pseudo-random number generators (PRNG) play a central role in the construction of encryption schemes. The security of many cryptographic systems depends on the generation of unpredictable quantities, such as the keystream in the one-time pad, the secret key in the DES encryption algorithms, the primes *p, q* in the RSA encryption and digital signature schemes, etc. CPRNGs have particularly attractive properties which guarantee the uniqueness of the generated sequences for any chosen seed and the independence of the generated numbers along the obtained trajectory (the sequence).

2. CHAOTIC PSEUDO-RANDOM NUMBER GENERATORS

In this section, we discuss how to construct a CPRNG and analyze its properties. To ensure the required statistical properties of generated sequences the systems are not only chaotic but also ergodic or even mixing. Traditionally, statistical testing was used to assess or estimate the quality of the proposed CPRNGs. For instance, the American norm FIPS 140-2 is one of the standard benchmarks (NIST 2001).

2.1 Generating Chaotic Pseudo-Random Bit Sequence.

Given a dynamic system, (X,ϕ), with a normalized invariant measure μ. Divide the state space X in some appropriate way into two disjoint parts, X_0 and X_1, such that $\mu(X_0)=\mu(X_1)=1/2$ and take an initial value $x_0 \in X$ as a seed.

To obtain a pseudo-random bit sequence we start to observe the evolution of the system governed by ϕ initiated from x_0, i.e., the sequence $x_n=\phi^n(x_0)$. Then, the *n*-th bit b_n of the sequence is determined by the coin-tossing formula:

$$b_n=\begin{cases}0 & if\ x_n \in X_0\\ 1 & if\ x_n \in X_1\end{cases}.$$

Thus, an infinite bit sequence, $B(x_0)=\{b_1,b_2,\ldots,b_n,\ldots\}$, is obtained.

Meanwhile, we get a map: $B: X \to \prod_{i=1}^{\infty}\{0,1\}$, such that

$$B(x_0)=\{b_i(x_0)\}_{i=1,2,\ldots}=\{b_1(x_0),b_2(x_0),\ldots\},$$

where $\prod_{i=1}^{\infty}\{0,1\}$ is a Cartesian product. Owing to the intrinsic properties of chaos, like the extreme sensitivity to initial conditions, ergodicity and mixing, the CPRNG possesses the fundamental properties: unique dependence of the sequence on the seed, equiprobable occurrence of "0" and "1", asymptotic statistical independence of bits, and so on.

2.2 Properties of the CPRBG.

2.2.1 Sensitivity to initial conditions

Theorem 1. For each $x_0 \in X$ the following holds true:

$$\mu\left(B^{-1}(b_i(x_0))\right)=0.$$

Theorem 1 says that if we take two different seeds in the generator, then with probability one, we obtain two different sequences of bits. In practice, due to chaos, and with some appropriate partitions, two different seeds lead to completely different sequences (Kotulski 2000, Szczepanski 2000).

2.2.2 Ergodicity

Ergodicity implies that the space X can not be divided into invariant nontrivial (w.r.t the measure μ) disjoint parts. Therefore, if a trajectory starts from any point $x_0 \in X$, it never settles in a small region, and even though knowing the final state of the system we can never identify the region (smaller than X) where the trajectory started. By ergodicity we obtain that the expected number of "0" bits in the generated sequence is equal to the expected number of "1" bits.

We say that a dynamic system (X,ϕ) is ergodic if and only if it has only trivial invariant sets, i.e., if and only if either $\mu(B)=0$ or $\mu(X/B)=0$, wherever B is a measurable, invariant under ϕ, subset of the space X (the invariance of B means that $\phi(B)\subset B$).

To be more precise, applying the Birkhoff-Khinchin Ergodic Theorem to the system yields:

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\chi x_0\left(\phi^i(x)\right)=\int_X \chi x_0 d\mu=\mu(X_0)$$

where χx_0 is the indicator function of the set X_0 and $x \in X$. Since by our assumption $\mu(X_0)=1/2$ we obtain that in the pseudo-random sequence determined by the seed x the average number of "0" tends to $1/2$.

2.2.3 Mixing

The mixing property means that any measurable set $A \subset X$ will be $\mu-$uniformly distributed over the whole state space X under iterations. We give the following theorem without proof, which states that

the bits generated by CPRBG are asymptotically independent.

Theorem 2. For $n = 1,2,\ldots$, the bits B_n, B_{n+k} (considered as random variables) generated by a given mixing dynamical system (X, ϕ) are asymptotically independent as k increases (Kotulski 2000, Szczepanski 2000).

2.3 Chaotic maps.

Many existing chaotic maps can be adopted to generate pseudo-random bit sequence, such as the Logistic map, $X_{n+1} = 4X_n(1 - X_n)$, with its analytic solution, $X_n = \sin^2(2^n \arcsin\sqrt{X_0})$, and the Baker's map, $X_{n+1} = \begin{cases} 2X_n & 0 \le X_n < 1/2 \\ 2(1 - X_n) & 1/2 \le X_n < 1 \end{cases}$, with its analytical solution, $X_n = \frac{1}{\pi}\arccos(\cos 2^n \pi X_0)$.

These can be extended to more general forms, as $X_n = \sin^2\left(k^n \arcsin\sqrt{X_0}\right)$, for $k = 2,4,\ldots$ and $X_i \in [0,1]$, and $X_n = \sin\left(k^n \arcsin X_0\right)$, for $k = 1,3,\ldots$ and $X_i \in [-1,1]$, as well as $X_n = \frac{1}{\pi}\arccos\left(\cos k^n \pi X_0\right)$, for $k > 2$, which corresponds to the Baker's map. In addition, all the known solutions can be represented as the following general form:

$$X_n = \Psi\left(\theta T \kappa^n\right),$$

where $\Psi(t)$ represents a periodic function (trigonometric, elliptic, hypoelliptic, Weirstrass, etc.), κ is an integer number, T is the period of $\Psi(t)$, $X_0 = \Psi(\theta T)$ is the initial condition of the chaotic system (θ is a real parameter defining this condition).

The Lyapunov exponent of such a system can be calculated as $\lambda = \ln \kappa$. These exactly solvable chaotic systems enable us to increase the accuracy and the speed of calculations.

2.4 Generating Chaotic Pseudo-random Numbers

It is known that each integer can be represented as a binary:

$$z = \sum_{k=0}^{M-1} b_k 2^k = (b_0, b_1, \ldots, b_{M-1}).$$

In order to obtain a highly random number with uniform distribution, we can randomly select each bit to correspond to the binary representation of an integer. That is to say, we use above obtained CPRBS to get the sequence $Z_n = (b_0, b_1, \ldots b_{M-1})_n$. Since b_k is taken from set $\{0,1\}$ with probability $\frac{1}{2}$, and an integer is obtained after M such independent events, the integer can take any value between 0 and $2^M - 1$ with equivalent probability $\frac{1}{2^M}$.

3. Implementation of the CPRNG on Chips

A chip implementation of the CPRNG is described in Fig.1 with an adoption of Logistic map. To generate an integer in between 0 and $2^M - 1$, M sets of CPRBSGs are built in. For each set, a chaotic sequence is first generated then fed into a comparator to get a binary sequence by one bit quantization.

However, in practice, the chaotic pseudo-random numbers are generated by computers in which computerization is in finite precision, which results in the appearance of period in CPRNS. In this case, the digitized nonlinear map is no longer chaotic and instead it shows a kind of stabilization, which leads to a random cycle length in CPRNS. Numerical simulations have shown that the cycle length is related to the computational precision and depends on the initial conditions. To solve this problem, a feedback mechanism is introduced here to increase the cycle length of the digitized chaotic map.

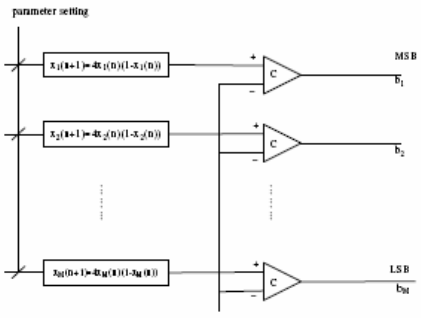

Fig. 1 Block scheme of the CPRNG

3.1 Chaotic Nonlinear Feedback Shift Registers

A feedback shift register is shown in Fig. 2, which consists of two parts: an *n*-bit register to right shift bits and a feedback Boolean function $f(a_1, a_2, \ldots a_n)$ to feed a binary value back to input. $f(a_1, a_2, \ldots a_n)$ can be either linear or nonlinear. A linear feedback Boolean function is often employed for simplifying the design and analysis. However, it is demonstrated that employing nonlinear feedback Boolean function, which is derived from a 1-D chaotic map, can greatly increase the cycle length.

The block diagram of the pseudo-random bit sequence generator is illustrated in Fig. 3, where the

digitized chaotic map takes the form of $x(k+1) = c(x(k)+1) \bmod 2^n$, followed by a comparator as a feedback Boolean function. Two

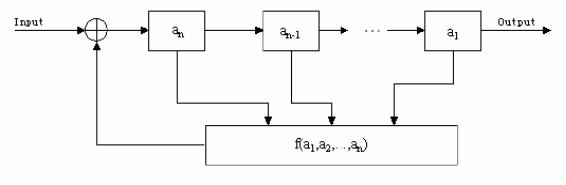

Fig.2 Feedback shift register

parameters, c and x_0, which are of *n*-bit length, determine the behavior of the bit sequence. So far, many digitization methods have been proposed. Here, the adopted digitization method is described as follows. At the *k*-th step, the integer input to the digitized chaotic map is $x(k) = x_{n-1}^k x_{n-2}^k \ldots x_0^k$, where x_i^k is the *i*-th bit of the integer. After one iteration, we get the output, $x(k+1) = x_{n-1}^{k+1} x_{n-2}^{k+1} \ldots x_0^{k+1}$, which is divided into even part, $x_e(k+1) = x_{n-2}^{k+1} x_{n-4}^{k+1} \ldots x_0^{k+1}$, and odd part, $x_o(k+1) = x_{n-1}^{k+1} x_{n-3}^{k+1} \ldots x_1^{k+1}$. The ensemble output of the feedback Boolean function is:

$$b = \begin{cases} 0 & if \ x_o(k+1) \le x_e(k+1) \\ 1 & if \quad x_o(k+1) > x_e(k+1) \end{cases}.$$

The bipolarized value b is further operated with plain-bit to get a feedback bit b^*. The b^* is appended at the end of $x(k)$ and the new $x(k)$ is left-shifted by one bit to get a new interger $x^*(k+1) = x_{n-2}^k \ldots x_0^k b^*$, which is to be used in the next round of iteration.

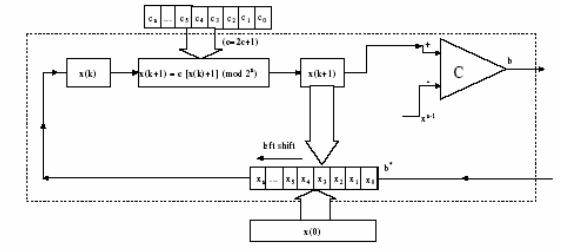

Fig.3 Structure of the chaotic pseudo-random bit sequence generator

3.2 Chaotic Pseudo-random Number Generator

The pseudo-random bit sequence generated by CNFSR is highly uncorrelated and of long circle length, therefore, a CPRNG consisting of sets of them is easily realized with good performance.

For simplicity, we construct a CPRNG with 8 sets of CNFSRs, as shown in Fig.4. The coupling of the CPRNSs complicates the ensemble behavior of the CPRNG and diffuses and confuses each CPRBS.

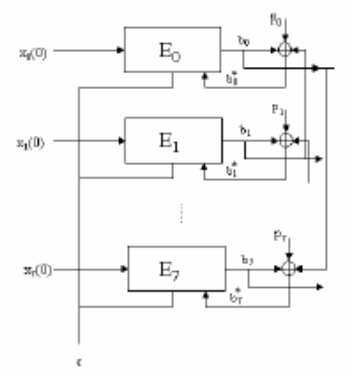

Fig.4 CPRNG with coupled CNFSR sets

3.3 Chip Implementation

Using the above proposed chaotic pseudo-random number generator, we can encrypt digitized texts, speeches or images byte by byte. Here, an image encryption is taken as an example for illustration.

The proposed encryption scheme is suitable for hardware implementation due to without float-point operation. To integrate the algorithm into a chip, only some registers, fix-point multipliers, comparators and some other logic circuits are needed, as shown in Fig.5.

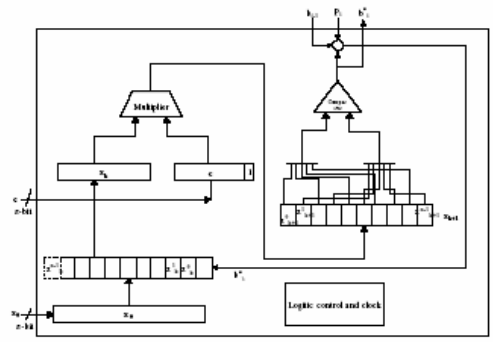

Fig.5 Chip design of a set of NFSR

Eight sets of such modules constitute an encryption chip to encrypt data byte-wise. The interface of the chip is shown in Fig.6. The interface circuit is very simple and the chip design is also not complex, thus it can be easily integrated into handhold or mobile devices.

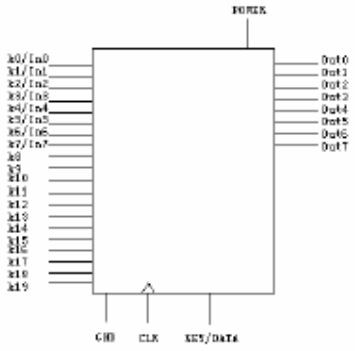

Fig. 6 Interface of chaotic encryption chip

3.4 Experimental Results

Fig.7 shows the histograms of encrypted-images, which are uniform. It makes statistical attacks difficult.

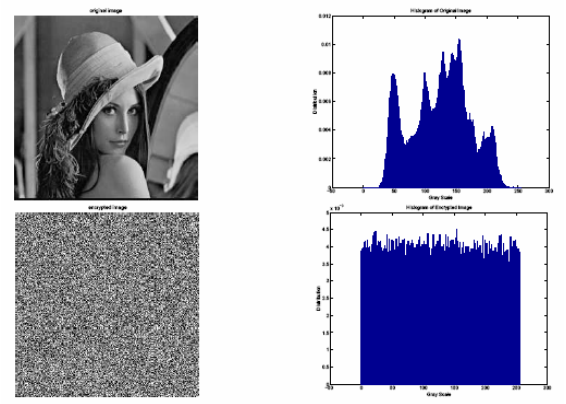

Fig.7 Histograms of plain-image and encrypted image XOR-ed with sequence of CPRN

A secure encryption should resist known plain-text or chosen plain-text attack, each of which can be used by opponents to dope out cipher keys so that the cryptosystem is broken. If we have two pieces of plain-texts, P and P^*, such that $\|P - P^*\| << \varepsilon$, and if their corresponding cipher-texts $C = E(P)$ and $C^* = E(P^*)$ are significantly different, i.e., $\|C - C^*\| >> M$, we say that the encryption scheme E is sensitive to plain-text. If an encryption scheme is sensitive to plain-text, it can resist known plain-text or chosen plain-text attack. Since the proposed encryption scheme has used cipher-bits feedback, the encrypted output is highly correlated to its corresponding plain-text. Thus, it can resist known plain-text or chosen plain-text attack. Two images are employed here for illustration, which have only one byte difference on the top-left regions. The experimental results are shown in Fig. 8, where we can find that almost all pixels have been changed after one pixel on top-left corner was modified. Actually, the ensemble pixel difference of the two encrypted images is about 92.58%.

4. Concluding Remarks

In this paper, chaos-based cryptography is surveyed with focus on designing chaotic pseudo-random bits generators for stream cipher. A chaotic stream encryption scheme is proposed with chip implementation. In particular, it uses the digitized chaotic map instead of a continuous one, which simplifies chip design and makes it easy for hardware implementation. Experimental results have illustrated the effect of the proposed scheme. In the future, the detailed chip design and other tests will be further carried out.

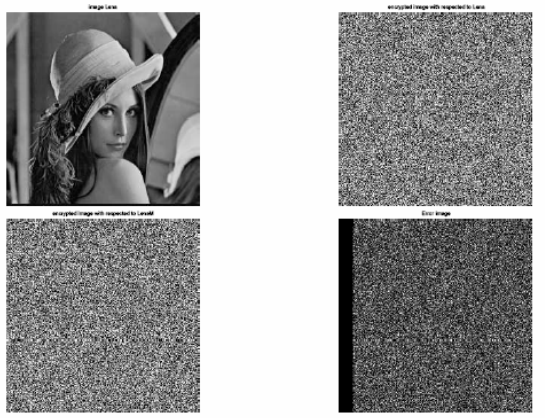

Fig.8 Sensitivity to a slight change in plain-text

REFERENCES

Baptista, M.S., "Cryptography with chaos," *Physics Letters A*, Vol.240, 1998, pp.50—54.

Frey, D.R., "Chaotic digital encoding: an approach to secure communication," *IEEE Trans. Circuits and Systems – II*, Vol.40, No.10, 1993, pp.660—666.

Kotulski, Z. and Szczepanski, J., "On constructive approach to chaotic pseudorandom number generators," *RCMCIS'2000*, pp.191—203.

Pecora, L.M. and Carroll, T.L., "Synchronization in chaotic systems," *Physical Review Letters*, Vol.64, No.8, 1990, pp.821—824.

Schneier, B., *Applied Cryptography – Protocols, algorithms, and source code in C*, John Wiley &Sons Inc., New York, 2nd edition, 1996.

Shannon, C.E., "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, Vol. 28, 1949, pp.656—715.

Szczepanski, J. and Kotulski, Z., "Chaotic pseudorandom numbers generators based on chaotic dynamical systems," Open Sys. And Information Dyn. Vol.7, 2000, pp.1—10.

Wong, W.K., Lee, L.P. and Wong, K.W., "A modified chaotic cryptographic method," *Computer Physics Communications*, Vol.138, 2001, pp.1932—1934.

Zhou, H. and Ling, X.T., "Problems with the chaotic inverse system encryption approach," *IEEE Trans. Circuits and Systems – I*, Vol.44, No.3, 1997, pp.268—271.

NIST, National Institute of Standards and Technology (2001), Federal Information Processing Standards Publication FIPS PUB 140-2: Security requirements for Cryptographic Modules, May 25.