A Secret Key Cryptosystem by Iterating a Chaotic Map

Toshiki Habutsu
Yoshifumi Nishio
Iwao Sasase
Shinsaku Mori
Department of Electrical Engineering, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223 JAPAN
Tel. +81-45-563-1141 Ext. 3319
Fax. +81-45-563-3421

Abstract

Chaos is introduced to cryptology. As an example of the applications, a secret key cryptosystem by iterating a one dimensional chaotic map is proposed. This system is based on the characteristics of chaos, which are sensitivity of parameters, sensitivity of initial points, and randomness of sequences obtained by iterating a chaotic map. A ciphertext is obtained by the iteration of a inverse chaotic map from an initial point, which denotes a plaintext. If the times of the iteration is large enough, the randomness of the encryption and the decryption function is so large that attackers cannot break this cryptosystem by statistic characteristics. In addition to the security of the statistical point, even if the cryptosystem is composed by a tent map, which is one of the simplest chaotic maps, setting a finite computation size avoids a ciphertext only attack. The most attractive point of the cryptosystem is that the cryptosystem is composed by only iterating a simple calculations though the information rate of the cryptosystem is about 0.5.

1 Introduction

Random oscillation of the solutions in deterministic systems described as differential or difference equations, is called chaos [1]. Recently many types of chaos-generating systems have been proposed and analyzed in various fields. Especially, chaotic behavior of solutions in some types of one-dimensional difference equations

$$X_{n+1} = F(X_n) \qquad X_n \in [0,1] \tag{1}$$

is investigated in detail [2]. One-dimensional discrete maps F generating chaotic solutions are called chaotic maps.

Chaotic solutions have the following features.

1. Sensitivity of parameters. – If a parameter (the shape of F) varies slightly, two sequences obtained from repeated calculations on a chaotic map from an initial point, eventually become quite different.

2. Sensitivity of initial points. – If an initial point X_0 varies slightly, two sequences obtained from repeated calculations on a chaotic map with a parameter, eventually become quite different.

D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp. 127-140, 1991.

3. Randomness. – Solutions starting from almost all X_0 in $[0,1]$ wander in $[0,1]$ at random and their distribution is uniform.

Therefore, if one doesn't know both the exact parameter and the exact initial point, he cannot expect the motion of the chaotic solution.

In this paper, we propose a secret key cryptosystem by iterating a one-dimensional map F generating chaos. This system is based on repeated calculations on a chaotic map as $X_n = F^n(X_0)$. We use the parameter α of the map for a secret key, and a point p in an interval $[0,1]$ for plaintext. Encryption function is n-times composite of F^{-1} and decryption function is n-times composite of F. Therefore, encryption and decryption are achieved by only repeating a very simple calculation.

Generally, because F is m to one map, one plaintext has m^n ciphertexts and any one of m^n ciphertexts can be deciphered only using the secret key. Therefore, senders can select the ciphertexts by any arbitrary random generator. We determine the parameter sizes to prevent statistic attacks. If the times of composite is large enough, it is expected that ciphertext variations act at random and are independent of key variations, because of the characteristics of chaotic maps.

We also discuss about a ciphertext only attack. In the following section, we explain our cryptosystem by iterating a tent map. Although tent map has linearity, we can prevent the ciphertext only attack from breaking our cryptosystem by setting finite computation size.

2 A Secret Key Cryptosystem by Iterating a Tent Map

In this section, we explain our cryptosystem. As an example of chaotic maps, we use tent map which is one of the most popular and the simplest chaotic maps.

2.1 Preliminaries

Tent map is a one-dimensional and piecewise linear map. Figures 1(a) and 1(b) show a tent map and its inverse map. These maps transform an interval $[0,1]$ onto itself and contain only one parameter α, which represents the location of the top of the tent. These maps are described as follows.

$$F : \begin{cases} X_{k+1} = \dfrac{X_k}{\alpha} & (0 \leq X_k \leq \alpha) \\ X_{k+1} = \dfrac{X_k - 1}{\alpha - 1} & (\alpha < X_k \leq 1). \end{cases} \tag{2}$$

$$F^{-1} : \begin{cases} X_{k-1} = \alpha X_k \\ \quad\text{or} \\ X_{k-1} = (\alpha - 1)X_k + 1. \end{cases} \tag{3}$$

Sequences calculated from arbitrary initial point with iterating F act chaotically because the function F is expansionary everywhere in the interval $[0,1]$. Such the sequences obtained by iterating a tent map distribute in uniform $U(0,1)$ [3].

F is two to one map and F^{-1} is one to two map. Therefore, F^n is 2^n to one map and F^{-n} is one to 2^n map. Since $X = F(F^{-1}(X))$ is always satisfied, $X = F^n(F^{-n}(X))$ is also satisfied.

2.2 Cryptosystem

(1) Secret Key

A parameter α denotes a secret key. If a sender and a receiver have a secret key, they are able to calculate the function F accurately.

(2) Encryption

i) — Set an initial point as a plaintext p, where $0 < p < 1$.

ii) — Calculate n-times composite of the inverse map $F^{-n}(p)$ by calculating F^{-1} repeatedly.

$$C = F^{-1}(F^{-1}(\cdots F^{-1}(p)\cdots)) = F^{-n}(p). \tag{4}$$

On each calculation, select one of two equations of F^{-1} in eq. (3) in any arbitrary way. This means that one plaintext has 2^n ciphertexts and one of 2^n ciphertexts is sent to the receiver. Finally, send the value C to the receiver.

(3) Decryption

Calculate n-times composite of the map $F^n(C)$ by calculating F repeatedly and recover the plaintext p.

$$p = F(F(\cdots F(C)\cdots)) = F^n(C) = F^n(F^{-n}(p)). \tag{5}$$

Note that only α is required for this computation. The information about which of two equations is used for each encryption process (F^{-1}), is not necessary for the decryption process. Any one of 2^n ciphertexts, even when the coin-flipping is used in the encryption process, is deciphered without fail.

Figure 2 visualizes an encryption and a decryption. Firstly, a sender sets an initial point p as a plaintext. On the first step of the encryption, he chooses right or left. If he chooses right, p is mapped to X^{-1} in the figure. The sender repeats this n times. The receiver only has to do is to trace inversely. The plaintext p which is exactly equal to α, is not a singular point. It is easy to confirm that the plaintext is enciphered similar to another plain texts: simply choose right or left side as the other plaintexts.

The encryption and the decryption are achieved by repeating a simple calculation. They require n times multiplications. On the each calculation, it is necessary to set a computation size. There are two reasons to set it. The first reason is that memory size of computer is finite. The second reason is about security of our cryptosystem. Because tent map is piecewise linear, our cryptosystem also has linearity. If ciphertext is described with the whole size digits, there exists a ciphertext only attack to our cryptosystem because of its linearity. We discuss about this problem in the following section.

3 Discussions

In this section, we discuss about the security and performances of our cryptosystem. Firstly, we determine the size of the parameters to prevent statistical attack and step-by-step attack. Secondly, we discuss about the size of ciphertexts to prevent failing decryption. Thirdly, we discuss about the ciphertext only attack. And finally, we discuss about the other chaotic maps to increase the security.

3.1 Requirements of the Parameters

3.1.1 Secret Key and Plaintext Size

Figures 3(a) and 3(b) show the distribution of the ciphertexts for different parameters. When α is close to 0, the distribution of ciphertexts is narrow as in figure 3(a) and eavesdroppers have larger probability of the achievement of attacking the key. Similarly, α must not be near 1. However when α is around 0.5 as in figure 3(b), the distribution of the ciphertexts is uniform enough. Therefore, we assume that α should be between 0.4 and 0.6.

The key space size and the plaintext size are required 64 bits against step-by-step attack. If they are described with 20 digits, both of the key space size and the plaintext size are about 64 bits.

3.1.2 The Times of Mapping : n

If a ciphertext is deciphered with two keys which are slightly different, the sequences are separating as n is getting larger, and eventually they become independent. Therefore, we determine n so as to satisfy the following two conditions.

i) By selecting some keys and computing plaintexts by deciphering a ciphertext, the distribution of the plaintexts for respective keys is uniform distribution $U(0,1)$.

ii) Changing the keys chosen in i) slightly makes the distribution independence from the distribution in i).

If these two conditions are satisfied, attackers cannot expect the plaintext from the ciphertext, as far as they do not know the accurate key.

Figure 4 shows the distribution of plaintexts obtained from a ciphertext with 1000 keys, where $n = 75$. It is shown that the distribution is consistent with uniform distribution $U(0,1)$. Therefore, condition i) is satisfied.

In order to test the condition ii), we use χ^2 test. The concept of the methods is as follows. Further details about the test of independence are in [4].

i) Divide the interval $[0,1]$ into l class intervals.

ii) Compute the N pairs of $F_\alpha{}^n(C)$ and $F_{\alpha+\Delta\alpha}{}^n(C)$, and make $l \times l$ contingency table (frequency $= k_{ij}$).

iii) Compute

$$\chi^2 = N(\sum_{i=1}^{l}\sum_{j=1}^{l} \frac{k_{ij}^2}{\sum_{j=1}^{l} k_{ij} \cdot \sum_{i=1}^{l} k_{ij}} - 1). \tag{6}$$

If this value is smaller than the upper 5% point of χ^2 of which the number of the degrees of freedom is $(l-1) \times (l-1)$, the independence is not rejected using the level of significance 0.05.

Figure 5 shows times of mapping n versus χ^2, where $l = 11$, $N = 1000$ and $\Delta\alpha = 10^{-20}$. Because the upper 5% point of χ^2_{100} is 124.3, the independence is not rejected when $n \geq 73$.

Leaving a safety margin, we determine that the times of mapping n is 75.

3.2 Ciphertext Size

Ciphertext size is equal to calculation size. If we have a computer with infinite memory, it is clear that the decryption process has no error. However, digital computer's memory is finite, so calculation error always exist. For this reason, we determine the size S not to occur any calculation error.

Firstly, we discuss about error in encryption process. Encryption function is contractional and its coefficient is about 0.5. At worst, error is 0.5×10^{-S} on each step of encryption and it is accumulated. Consequently, the error in encryption process is at worst

$$E_e = 0.5 \times 10^{-S} \times \sum_{k=0}^{n-2} (\frac{1}{2})^k = 10^{-S}(1 - (\frac{1}{2})^{n-1}). \tag{7}$$

Secondly, we discuss about error in decryption process. Decryption function is expansionary and its coefficient is about 2. Consequently, the error in decryption process is at worst

$$E_d = 0.5 \times 10^{-S} \times \sum_{k=0}^{n-1} 2^k. \tag{8}$$

Totally, computation error is at worst

$$E = 2^n \times E_e + E_d = 3 \times 2^{n-1} \times 10^{-S}. \tag{9}$$

If this error is smaller than 0.5×10^{-20}, plaintext is always recovered. Consequently, calculation size should be

$$S > n\log_{10}2 + \log_{10}3 + 20 = 43.05. \tag{10}$$

Figure 6 shows the rate of the correct decryption versus the significant digits obtained by a computer experiment. Since the times of composite of inverse map is 75, the size of ciphertext space is 20 digits +75 bits (= 42.58 digits). Actually, some more digits are required because computation error is accumulated by each step. As a result, if 44 digits is taken for the computation size, the decryption process is always correct.

We briefly discuss about the information rate of the cryptosystem. The information rate R is

$$R = \frac{\text{plaintext size}}{\text{ciphertext size}} = \frac{20}{44} \sim 0.5. \tag{11}$$

If you use FEAL or DES, for example, with a 32 bits message and a 32 bits random number, this system is similar to our cryptosystem, because its information rate is 0.5 and one message has 2^{32} ciphertexts, which all can be deciphered with the same decryption key. However, our cryptosystem is only composed of an easy function, which is an interesting point of our cryptosystem.

3.3 Ciphertext Only Attack

Because tent map is piecewise linear, n-times composite of tent map is also piecewise linear. Therefore, our cryptosystem has also linearity. If computation size is infinite, our cryptosystem is attacked because of its linearity. First we show the ciphertext only attack, and then we show why this attack does not succeed to break our cryptosystem.

From the encryption function eq. (3), almost all X_k are divided into the following two states, and thus almost all ciphertexts are divided into these states.

State 1 : a multiple of α

State 2 : 1+ a multiple of α.

[Proof]

i) First, we think the case when X_k is in state 1. If the sender chooses the left side of the tent map at this step, X_{k-1}, which is the next X_k, is in the state 1 because

$$X_{k-1} = \alpha(\alpha A_1), \tag{12}$$

where $X_k = \alpha A_1$. If the sender chooses the right side of the tent map at this step, X_{k-1} is in state 2 because

$$X_{k-1} = (\alpha - 1)\alpha A_2 + 1 = \alpha(\alpha - 1)A_2 + 1, \tag{13}$$

where $X_k = \alpha A_2$.

ii) Second, we think the case when X_k is in state 2. Whichever the sender chooses, X_{k-1} is in state 1 because

$$X_{k-1} = \alpha(\alpha A_3 + 1), \tag{14}$$

where $X_k = \alpha A_3 + 1$, and

$$\begin{aligned} X_{k-1} &= (\alpha - 1)(\alpha A_4 + 1) + 1 \\ &= \alpha(\alpha A_4 - A_4 + 1), \end{aligned} \tag{15}$$

where $X_k = \alpha A_4 + 1$.

iii) Finally, we think the first step of the encryption. Whatever plaintext p is, just after the sender chooses the left side of the tent map, X_k is in state 1. After this state, X_k is in state 1 or state 2 as we mentioned above. The only one case which X_k is never in these two states is that the sender chooses the right side of the tent map during all the encryption steps. If the sender chooses the side randomly, the provability of this is 2^{-75}. Consequently, almost all ciphertexts are divided into these states.

This fact enables attackers the following attack. If an attacker can eavesdrop two ciphertexts C_0 and C_1, he can obtain the key α after at most four times tests like

$$\gcd(|\,(C_0 - b_0)\,|\,10^{20\times(n+1)}, |\,(C_1 - b_1)\,|\,10^{20\times(n+1)}) = \text{a multiple of } \alpha \times 10^{20}, \qquad (16)$$

where b_0 and b_1 are 1 or 0. This is the ciphertext only attack.

Next we show why this attack does not succeed to break our cryptosystem if we set computation size. If a sender calculated a ciphertext whose size was infinite, it is described by

$$\text{key size} \times n + \text{plaintext size} = 1520 \text{ digits} \qquad (17)$$

because key size and plaintext size are both 20 digits, and $n = 75$. If he sent the ciphertext of this size to receiver and an attacker could eavesdrop it, the attacker can obtain the key α because the linearity of our cryptosystem still exists. However, ciphertext can be described by only 44 digits. This means that the attacker lacks the information to succeed the attack. In other words, although our cryptosystem is described by linear functions, setting computation size saves our cryptosystem from the attack.

3.4 Other Chaotic Maps

As we mentioned above, the ciphertext only attack is avoided by the setting computation size but the tent map cryptosystem still has linearity. There will exist other types of attacks such as chosen plaintext attack, known plaintext attack, and so on. We expect that these attack will be based on the characteristics of the linearity. We recommend other chaotic maps to avoid these attacks. For example, a certain of non-linear one-dimensional chaotic map meets this condition. Further research is necessary for this aspect.

4 Conclusions

We have proposed a new secret key cryptosystem by iterating a chaotic map. In the case that we use a tent map as a chaotic map, we determine the parameter sizes to prevent statistic attacks by χ^2 test, whose result is that the times of mapping should be larger than 73 if the key size and the plaintext size are both 20digits. We verify that correct decryption is achieved if the computation size is larger than 44 digits. We also verify that the computation size prevent the ciphertext only attack from breaking our cryptosystem. In the proposed system, a plaintext has 2^n ciphertexts and one of 2^n ciphertexts is sent to the receiver. Even if the ciphertext is chosen by any arbitrary way, the receive can obtain the plaintext only using the secret key.

Acknowledgement

The authors wish to thank Dr. Tsutomu Matsumoto at Yokohama National University for his valuable suggestions and all the participants of EUROCRYPT 91 for their useful discussions.

References

[1] J. M. T. Thompson and H. B. Stewart: "Nonlinear Dynamics and Chaos", John Wiley and Sons, Chichester, 1986.

[2] P. Collet and J. P. Eckmann: "Iterated Maps on the Interval as Dynamical Systems", Birkhäuser, Boston, 1980.

[3] S. Oishi and H. Inoue: "Pseudo-Random Number Generators and Chaos", Trans. IECE Japan, E65, 9, pp.534-541 (Sept. 1982)

[4] G. K. Bhattacharyya and R. A. Johnson: "Statistical Concepts and Methods", John Wiley and Sons, Tronto, 1977.

[5] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori: "A Secret Key Cryptosystem Using a Chaotic Map", Trans. IEICE Japan, E73,7, pp.1041-1044 (July 1990)

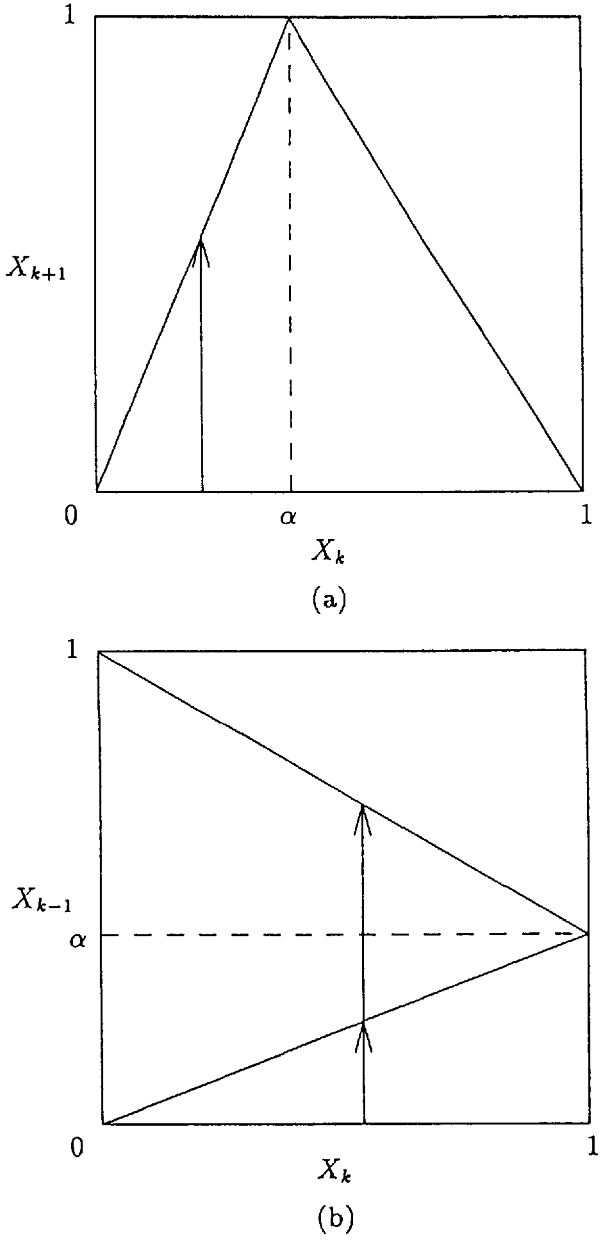


Fig.1 (a) Tent map.
(b) Inverse tent map.

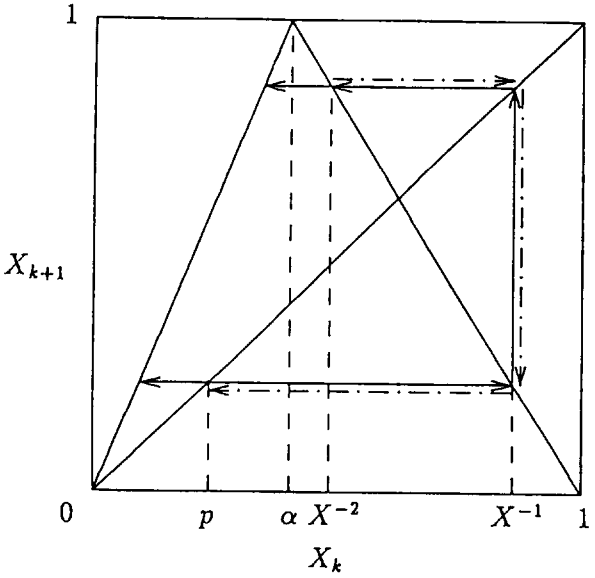
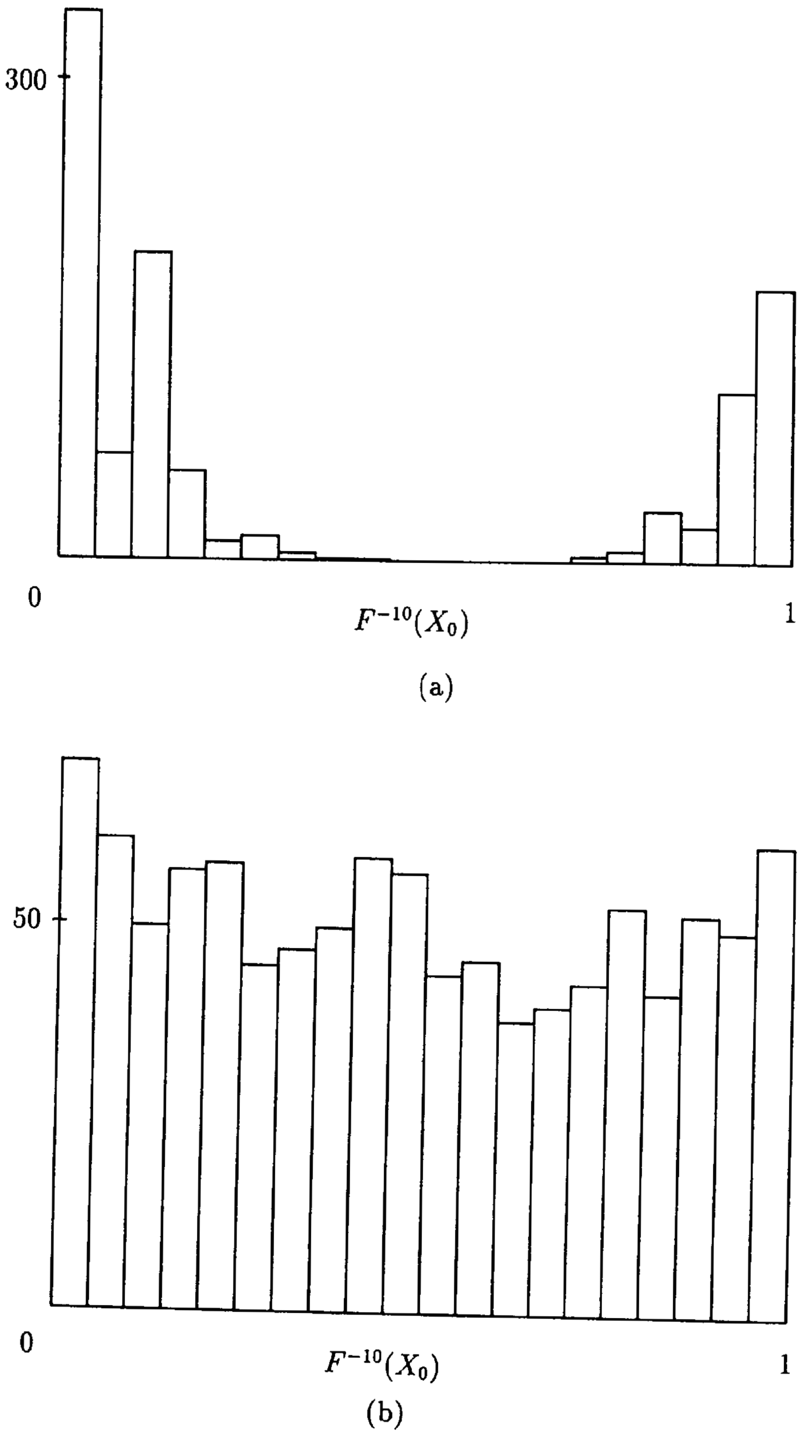


Fig.2 Encryption and Decryption.

⟶ : Encryption

—·→ : Decryption



(a)

(b)

Fig. 3 The histogram of $F^{-10}(X_0)$
in 20 intervals $[i/20,(i+1)/20)$, $i = 0,\cdots,19$.
(a) $\alpha = 0.11$ $X_0 = 0.2356$

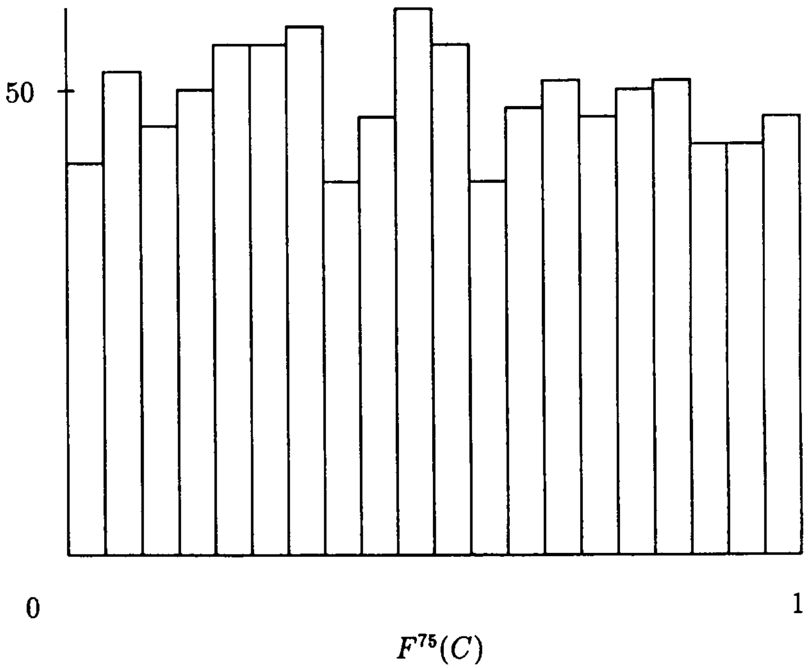


Fig. 4 The histogram of plaintexts obtained from the same ciphertext for 1000 keys ($C = 0.3987$)
:20 intervals $[i/20, (i+1)/20)$, $i = 0, \cdots, 19$.

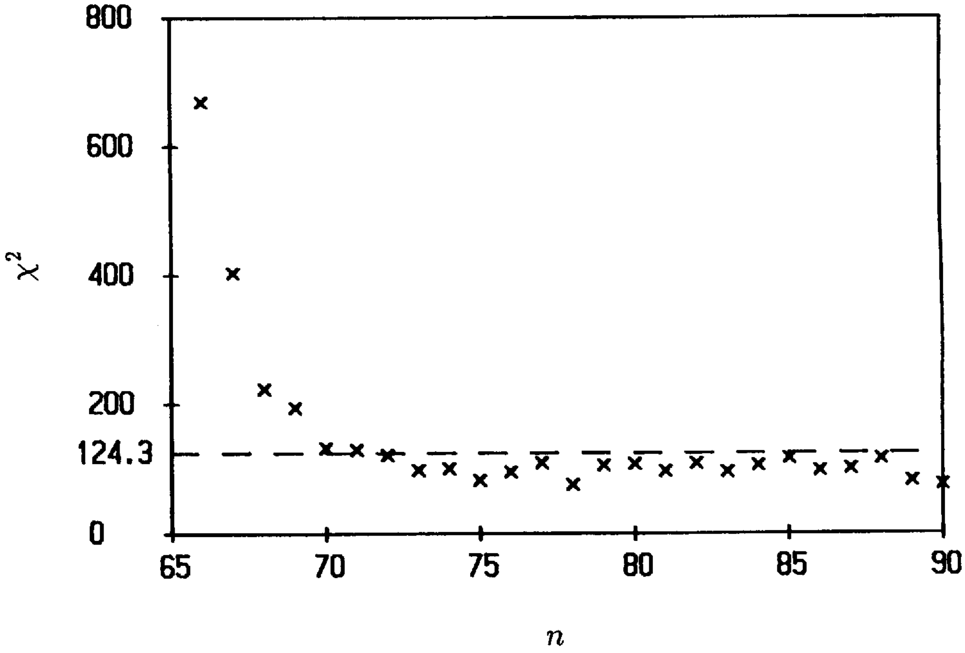


Fig. 5 The results of χ^2 test.

$(\chi^2_{100}(0.05) = 124.3)$

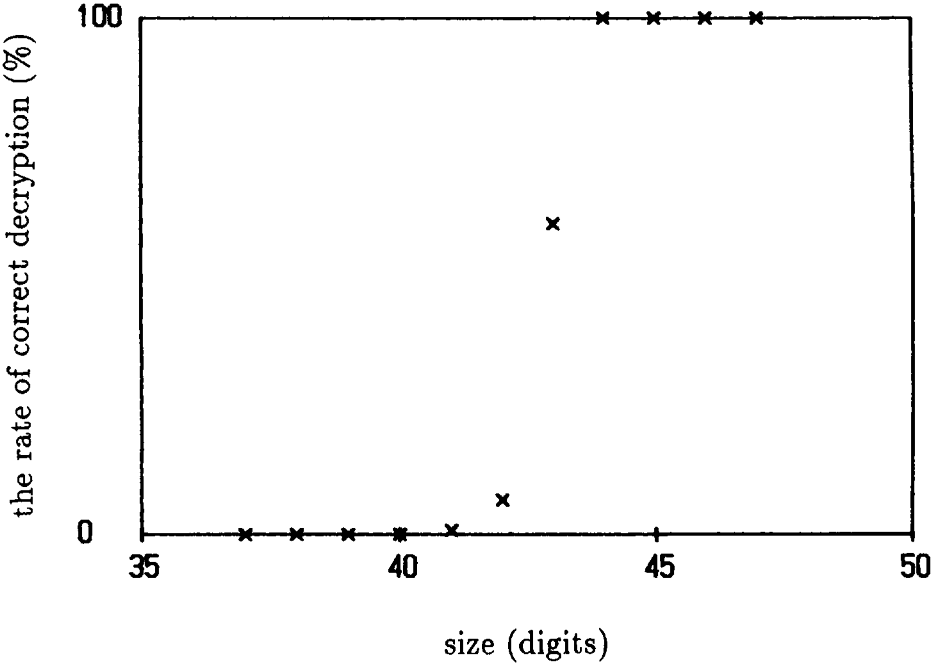


Fig. 6 The rate of correct decryption.
(Computer simulation : 1000 samples)