Kybernetika

VOLUME *44* (2008), NUMBER 4

The Journal of the Czech Society for Cybernetics and Information Sciences

Published by:

Institute of Information Theory and Automation of the AS CR

Kybernetika is a bi-monthly international journal dedicated for rapid publication of high-quality, peer-reviewed research articles in fields covered by its title.

Kybernetika traditionally publishes research results in the fields of Control Sciences, Information Sciences, System Sciences, Statistical Decision Making, Applied Probability Theory, Random Processes, Fuzziness and Uncertainty Theories, Operations Research and Theoretical Computer Science, as well as in the topics closely related to the above fields.

The Journal has been monitored in the Science Citation Index since 1977 and it is abstracted/indexed in databases of Mathematical Reviews, Zentralblatt für Mathematik, Current Mathematical Publications, Current Contents ISI Engineering and Computing Technology.

K y b e r n e t i k a . Volume *44* (2008) ISSN 0023-5954, MK ČR E 4902.

Published bimonthly by the Institute of Information Theory and Automation of the Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. — Address of the Editor: P. O. Box 18, 182 08 Prague 8, e-mail: kybernetika@utia.cas.cz. — Printed by PV Press, Pod vrstevnicí 5, 140 00 Prague 4. — Orders and subscriptions should be placed with: MYRIS TRADE Ltd., P. O. Box 2, V Štíhlách 1311, 142 01 Prague 4, Czech Republic, e-mail: myris@myris.cz. — Sole agent for all "western" countries: Kubon & Sagner, P. O. Box 34 01 08, D-8 000 München 34, F.R.G.

Published in September 2008.

KYBERNETIKA — VOLUME 44 (2008), NUMBER 4, PAGES 501 – 510

A RELATED–KEY ATTACK ON ITERATED CHAOTIC CIPHERS

Yang Yang and Chenhui Jin

In this paper, we present a new type of attack on iterated chaotic ciphers using related keys. Based on the fact that a chaotic sequence is not sensitive to the less significant bits of initial conditions and parameters, a divide-and-conquer attack on iterated chaotic ciphers was presented by us before, which significantly reduces the computing complexity of attacks. However, if the information leaked is significant according to the distribution of the coincidence degrees, a measure for the information leakage of chaotic ciphers, or the size of the key is large, then it is difficult for the divide-and-conquer attack to reduce its computing complexity into a realizable level. The related-key attack we present in this paper simultaneously uses the information leaked from different chaotic sequences generated by related keys and combines the ideas of linear cryptanalysis and divide-and-conquer attack together, hence greatly enhances the efficiency of divide-and-conquer attack. As an example, we test the related-key attack on the *ZLL* chaotic cipher with a 64-bit key on a Pentium IV 2.5 GHz PC, which takes only 8 minutes and 45 seconds to recover all bits of the key successfully.

Keywords: chaotic cipher, related-key attack, *ZLL* chaotic cipher, divide-and-conquer attack, known plaintexts attack

AMS Subject Classification: 34C28, 94A60

1. INTRODUCTION

A chaotic sequence is a nonlinear deterministic sequence that is sensitive to the initial condition and the parameters eventually. A chaotic sequence is constructed usually by operations on real numbers and can be generated quickly. Two chaotic sequences with a slight difference on the initial conditions and the parameters will eventually become uncorrelated. Hence, chaotic sequences were frequently applied to construct stream ciphers, and the initial conditions and/or the parameters of a chaos-based encryption scheme are used as key in most cases.

The security of chaotic ciphers is a key issue for study. In this research area, a kind of multi-resolutionary cryptanalysis was proposed in [4], which can decrease the entropy of the key for the chaotic cipher proposed in [8] by 2 bits on average. But the amount of plain texts required and the computing complexity of the attack are very large. It is found in [3] and [2] that the less significant bits of the initial

conditions of chaotic ciphers have little effect on the initial signals of the output sequence, where the concept of coincidence degree was also introduced, which will be further explained below. In [3] the leaked information of a chaotic cipher is quantified by the distribution of coincidence degrees, where a divide-and-conquer attack on chaotic ciphers is presented, which can reduce the computing complexity of attacks greatly. But there are still two limitations to the above attack. One is that if the amount of information leaked is insignificant according to the distribution of the coincidence degrees, the reduced entropy of the key will not be large. The other is that it is difficult to reduce the computing complexity of a divide-and-conquer attack to a realizable level when the key size is large.

In this paper, we present a new type of attack on iterated chaotic ciphers using related keys. The attack uses simultaneously the information leaked from different chaotic sequences generated by related keys and combines the ideas of linear cryptanalysis and divide-and-conquer attacks together, hence greatly enhances the efficiency of the divide-and-conquer attack. It also overcomes the limitations mentioned above. The chaotic encryption scheme *ZLL*, which is based on a piecewise linear chaotic map (PWLCM for short), is a typical iterated chaotic cipher. As an example, we have tested the related-key attack to *ZLL* whose parameters and initial condition are used as keys, where the key size is set as 64 bits. The attack succeeds with a success rate of 0.94 for 50 known output sequences generated by related keys, and its runtime is 11 minutes and 39 seconds on average. With 80 known output sequences generated by related keys, it takes only 8 minutes and 45 seconds on average with a success rate near 1.0. The method is also applicable to chaotic ciphers with known parameters.

The paper is organized as follows. In Section 2, we review the *ZLL* cipher and analyze the distribution of the coincidence degrees. The related-key attack method on *ZLL* cipher is presented in Section 3. The success rate and the computing complexity are also discussed in this section. In Section 4, other types of related-key attacks are described. Section 5 gives the conclusions.

2. THE ZLL CHAOTIC CIPHER AND KEY INFORMATION LEAKED

The *ZLL* chaotic cipher consists of two transformations. One is a piecewise linear chaotic map (PWLCM for short) f, the other is a map T.

Let p be a natural number and $a_0, a_1, \ldots, a_{p+1}$ be real numbers with $0 = a_0 < a_1 < a_2 < \cdots < a_p < a_{p+1} = 1$. Let $a = (a_0, a_1, \ldots, a_{p+1})$ and define a chaotic map $f_a : [-1, 1) \to [-1, 1)$ by

$$f_a(x) = \begin{cases} -1 + 2(x - a_j)/(a_{j+1} - a_j), & \text{if } x \in [a_j, a_{j+1}),\ j = 0, 1, \ldots, p\ ; \\ f(-x), & \text{if } x < 0. \end{cases} \tag{1}$$

Let n be a natural number and $I_0, I_1, \ldots, I_{2^n-1}$ be intervals defined by $I_i =$

$[\frac{i}{2^{n-1}} - 1,\ \frac{i+1}{2^{n-1}} - 1)$ for $0 \le i < 2^n$. Define $T : [-1,\ 1) \to \{0,\ 1\}$ by

$$T(x) = \begin{cases} 0, & \text{if } x \in \bigcup_{k=0}^{2^{n-1}-1} I_{2k}; \\ 1, & \text{if } x \in \bigcup_{k=0}^{2^{n-1}-1} I_{2k+1}. \end{cases} \tag{2}$$

In [3], it was shown that equality $T(x) = \lfloor 2^{n-1}x \rfloor \mod 2$.

Let $x = \sum_{i=1}^{\infty} x_i/2^i$ be a nonnegative real number with $x_i \in \{0, 1\}$. The real number $\sum_{i=1}^{m} x_i/2^i$ is called the m-precision number of x. In this paper, we consider the m-dimensional binary vector $(x_1, x_2, \ldots, x_m)$ as the same as $\sum_{i=1}^{m} x_i/2^i$, and call $(x_1, x_2, \ldots, x_n)$ the most significant n bits of $x = \sum_{i=1}^{\infty} x_i/2^i$.

Let x_0 be a real number with precision m. The chaotic sequence $\{x_i\}_{i=1}^{\infty}$ of the *ZLL* cipher is generated by

$$x_i = [f_a(x_{i-1})]_m, \quad i \ge 1,$$

where $[f_a(x_{i-1})]_m$ is the m-precision number of $f_a(x_{i-1})$. The binary output sequence $\{s_i\}_{i=1}^{\infty}$ is constructed by $s_i = T(x_i)$ for $i \ge 1$. Suppose $\{m_i\}_{i=1}^{\infty}$ is the binary sequence of plaintext. Then, the binary sequence of ciphertext is $\{m_i \oplus s_i\}_{i=1}^{\infty}$, where $\oplus$ denotes addition modulo 2. Obviously, the output sequence $\{s_i\}_{i=1}^{\infty}$ can be discovered if the pair of plaintext and ciphertext are given, therefore an attack to the *ZLL* cipher with known plaintexts is equivalent to an attack with known output sequences.

The initial value x_0, the parameters n, p, and $a_0, a_1, \ldots, a_{p+1}$, may be used for the key of the *ZLL* chaotic cipher. Since n and p can be easily searched, we assume that n and p are known and the key is $k = (a_1, a_2, \ldots, a_p, x_0)$.

Definition 1. (Jin and Gao [3]) Let k be the key of a stream cipher and k' be a testing key. Let $\{s_i\}_{i=0}^{\infty}$ and $\{s'_i\}_{i=0}^{\infty}$ be output sequences generated by k and $k^{'}$, respectively. Then, the coincidence degree of $k'$is defined by

$$\max\{n : s'_i = s_i \text{ for any } i \text{ with } 1 \le i \le n\}.$$

For a well-designed stream cipher, the output sequence generated by $k^{'}$ should be balanced over $\{0, 1\}$ and independent of that generated by k. So the coincidence degree of $k^{'}$ should follow the geometric distribution, i. e. $p(\xi = t) = 2^{-t-1}$ for $t \ge 0$. Hence, $p(\xi \ge t) = 2^{-t}$ and the expectation of ξ is 1. However, most chaotic maps are designed by continuous functions or piecewise continuous functions, hence a slight difference in inputs of the chaotic map will result in a slight difference in the outputs. If we replace the less-significant bits of an input by 0's, the most significant bits of the output of the chaotic map will not be changed with a great probability. Thus the less-significant bits of initial values and parameters have little effect on the most significant bits of the output sequences.

Let $k = (a_1, \ldots, a_p, x_0)$ be the key of the *ZLL* cipher, and $k^{(m)} = (a_1^{(m)}, \ldots, a_p^{(m)}, x_0^{(m)})$ be constructed by replacing each element in k with its m-precision number.

The coincidence degree of $k^{(m)}$ is denoted by n_m. Note that the distribution of n_m is far different from the geometric distribution and the coincidence degree of $k^{(m)}$ is larger than that of the test keys constructed by random m-precision numbers. Hence, we can distinguish $k^{(m)}$ from other test keys constructed by random m-precision numbers.

It's difficult to give the distribution of n_m precisely. But we may reveal it by some testing methods. Let $n = 5, p = 1$, and the size of key be 64 bits. Then $k = (a, x_0)$, and the sizes of a and x_0 are both 32 bits. Table 1 displays the distribution of n_8 by an experiment with 10 thousand independent random keys. For example, the number of cases of $n_8 = 2$, out of 10 thousand trials, is 2634 as listed in Table 1.

Table 1. The distribution of n_8.

n_8	0	1	2	3	4	5	6	7	8	9	10	≥ 11
count	762	2246	2634	1929	1107	628	325	178	89	53	26	23

Moreover, we have $p(n_8 \geq 2) = 0.6992$, $p(n_{16} \geq 7) = 0.5398$, and $p(n_{24} \geq 12) = 0.4584$. With the test results, we carry out the divide-and-conquer attack on the *ZLL* in the following way.

We may attack $k^{(8)}$ firstly, and then $k^{(16)}$, $k^{(24)}$, $k^{(32)}$, step by step. For example, we may search for each possible value of $k^{(8)}$ first, and leave all test values with coincidence degrees ≥ 2 to be candidates of $k^{(8)}$. The candidates are not unique in most cases. Subsequently, we may search for all the possible values of $k^{(16)}$ in which the most significant 8 bits are used as one of the candidates of $k^{(8)}$, and leave all test keys with coincidence degrees ≥ 7 to be candidates of $k^{(16)}$. Similarly, $k^{(24)}$, $k^{(32)}$ are attacked.

The principle of the above divide-and-conquer attack is as follows.

Lemma 1. (Jin and Gao [3]) Let k be the key of a stream cipher. Then, the probability of a test key with coincidence degree $\geq t$ is 2^{-t}, and the probability that the set of test keys with coincidence degrees $\geq t$ containing $k^{(m)}$ is $p(n_m \geq t)$.

Let a and b be m-precision numbers and $k' = (a, b)$ be a test key of $k^{(m)}$. Then, $k' = (a, b)$ may be regarded as a candidate of $k^{(m)}$ if its coincidence degree $\geq t$ for a given t, by Lemma 1. In this way, the number of candidates is decreased by a factor of 2^{-t} on average. In other words, the entropy of the key is reduced by t bits, and the probability that the true key is not missed is $p(n_m \geq t)$.

In order to ensure the success rates of the divide-and-conquer attack, we should select t such that the probability $p(n_m \geq t)$ is large enough. If we can't find a larger t such that $p(n_m \geq t)$ is greater than an expected success rate p, it is hard to reduce the computing complexity of an attack to a realizable level. Moreover, our experiments show that the increasing speed of t with $p(n_m \geq t) \geq q$ for a fixed q is slower than that of m in general. So, once the key size becomes large, it is difficult to reduce the computing complexity to a realizable level though the entropy of the key could be decreased greatly.

3. RELATED–KEY ATTACK ON THE ZLL CHAOTIC CIPHER

In order to overcome the limitations of the divide-and-conquer attack, we design a related-key attack on chaotic ciphers, which combines the ideas of linear cryptanalysis and the divide-and-conquer attack. Consequently, the efficiency of the divide-and-conquer attack is enhanced greatly by using the output sequences generated from different related keys.

Let $\delta_1 = (\alpha_1, \beta_1), \delta_2 = (\alpha_2, \beta_2), \ldots, \delta_N = (\alpha_N, \beta_N)$, and let $d_i = \{d_{ij}\}_{j=1}^{\infty}$ be the output sequences of the *ZLL* chaotic cipher generated by the key $k \oplus \delta_i$, respectively, where $k = (a, x_0)$ and $k \oplus \delta_i = (a \oplus \alpha_i, x_0 \oplus \beta_i)$. Now, we try to recover the key $k = (a, x_0)$ under the condition that all δ_i and d_i are known. The attack is called a related-key attack since the keys $k \oplus \delta_1, k \oplus \delta_2, \ldots, k \oplus \delta_N$, generating the output sequences $d_1, \ldots, d_N$, are N related keys.

The above conditions for the related-key attack can be satisfied under some circumstances. For example, if a pair of users adopt the following session key protocol, the conditions of the related-key attack are satisfied automatically:

Let $k = (a, b)$ be the shared key of Alice and Bob, and $E_k(m)$ denote the ciphertext of message m encrypted by k. When Alice wants to send the message m to Bob secretly, she may choose α and β randomly with the lengths of α and β equal to that of a and b, respectively, and then send $E_{k\oplus\delta}(m)$ and $\delta = (\alpha, \beta)$ to a public channel.

The related-key attack on chaotic ciphers are designed as follows:

Let $p(n_m \geq d) = p$. From $k \oplus \delta_i = (a \oplus \alpha_i, x_0 \oplus \beta_i)$, we have

$$\begin{aligned}(k \oplus \delta_i)^{(m)} &= (a \oplus \alpha_i, x_0 \oplus \beta_i)^{(m)} = ((a \oplus \alpha_i)^{(m)}, (x_0 \oplus \beta_i)^{(m)}) \\ &= (a^{(m)} \oplus \alpha_i^{(m)}, x_0^{(m)} \oplus \beta_i^{(m)}) = k^{(m)} \oplus \delta_i^{(m)}.\end{aligned}$$

Hence, we can obtain $(k \oplus \delta_i)^{(m)}$ with known δ_i and assumed $k^{(m)}$. Since the initial d values of the two output sequences, generated by $k \oplus \delta_i$ and $(k \oplus \delta_i)^{(m)}$, are the same with probability $p = p(n_m \geq d)$, we can attack $k^{(m)}$ by searching for its every possible value by the following algorithm.

Algorithm 1. Let $k' = (a', b')$ be an assumed value of $k^{(m)} = (a^{(m)}, x_0^{(m)})$, and let $k' \oplus \delta_i^{(m)} = (a' \oplus \alpha_i^{(m)}, b' \oplus \beta_i^{(m)})$ for $1 \leq i \leq N$. We test whether the initial d values of the output sequence generated by $k' \oplus \delta_i^{(m)}$ are the same as those generated by $k \oplus \delta_i$. If the answer is yes, we declare that k' has passed the test and set $\xi_i(k') = 1$; otherwise, $\xi_i(k') = 0$. If we let $T_{k'} = \sum_{i=1}^{N} \xi_i(k')$, then after each possible value of $k^{(m)}$ has been tested, the k' with $T_{k'} > T_{k''}$ for all k'' where $k'' \neq k'$ is determined to be the correct $k^{(m)}$, and the $k_1, \ldots, k_n$ with $T_{k_1} \geq T_{k_2} \geq \cdots \geq T_{k_n} \geq T_{k''}$ for all k'' where $k'' \neq k_1, \ldots, k'' \neq k_n$ are determined to be the first, the second, ..., the nth candidate for $k^{(m)}$.

Lemma 2. Let $|K|$ denote the total number of exhausted keys in Algorithm 1 and let $p_{k'} = p(\xi_i(k') = 1)$, where k' is a test key. Suppose that the random variables

$\xi_i(k'), 1 \le i \le N, k' \in K$, are independent of each other. Then, the probability that the first candidate of $k^{(m)}$ in Algorithm 1 is correct is

$$\sum_{i=1}^{N} \Big[C_N^i p_{k^{(m)}}^i (1-p_{k^{(m)}})^{N-i} \prod_{k' \in \{0,1\}^m \setminus k^{(m)}} \sum_{j=0}^{i-1} C_N^j p_{k'}^j (1-p_{k'})^{N-j} \Big].$$

Proof. $T_{k'} = i$ means that the number of 1's in $\xi_1(k'), \ldots, \xi_N(k')$ is i and the number of 0's is $N-i$. Thus $p(T_{k'} = i) = C_N^i p_{k'}^i (1-p_{k'})^{N-i}$ since $\xi_1(k'), \ldots, \xi_N(k')$ are independent. The fact that the first candidate of $k^{(m)}$ in Algorithm 1 is correct is equivalent to $T_{k^{(m)}} > \max\limits_{k':k' \neq k^{(m)}} T_{k'}$. So the probability is

$$\begin{aligned}
&p\Big(T_{k^{(m)}} > \max_{k' \in \{0,1\}^m \setminus k^{(m)}} T_{k'}\Big) \\
&= \sum_{i=0}^{N} p(T_{k^{(m)}} = i \text{ and } T_{k'} < i \text{ for } \forall\, k' \in \{0,1\}^m \setminus k^{(m)}) \\
&= \sum_{i=0}^{N} p(T_{k^{(m)}} = i) \prod_{k' \in \{0,1\}^m \setminus k^{(m)}} p(T_{k'} < i) \\
&= \sum_{i=1}^{N} \Big[C_N^i p_{k^{(m)}}^i (1-p_{k^{(m)}})^{N-i} \prod_{k' \in \{0,1\}^m \setminus k^{(m)}} \sum_{j=0}^{i-1} C_N^j p_{k'}^j (1-p_{k'})^{N-j} \Big].
\end{aligned}$$

The second equation holds since all $T_{k'}, k' \in K$ are independent. □

A related-key attack on the *ZLL* cipher with the key size being 64 bits is described in Algorithm 2, in which we attack $k^{(8)}$, $k^{(16)}, k^{(24)}$, $k^{(32)}$ one by one.

Algorithm 2.

Step 1. Let N=80, d_8=2, d_{16}=7, d_{24}=12, $d_{32} = 20$, and $m_i = 8i$ for $1 \le i \le 4$. Set $i = 1$.

Step 2. Attack $k^{(m_i)} = (a^{(m_i)}, x_0^{(m_i)})$:
If $i = 1$, we use $k' = (a', b')$ as a test value of $k^{(8)}$ with a' and b' being selected from $\{0,1\}^8$ in turn. If $2 \le i \le 4$, we denote by $k'_{i-1} = (a_{i-1}, b_{i-1})$ the candidate of $k^{(m_{i-1})}$ and use $k' = (a', b')$ as a test value of $k^{(m_i)}$ with the most significant m_{i-1} bits of a' fixed as a_{i-1}, the most significant m_{i-1} bits of b' fixed as b_{i-1}, and the lower $m_i - m_{i-1}$ bits of $a'$and b' selected from $\{0,1\}^{m_i - m_{i-1}}$ in turn. If $i = 4$, go to Step 3. When $i \le 3$, for each j with $1 \le j \le N$, once the initial d_{m_i}values of the output sequence generated by $k' \oplus \delta_j^{(m)}$ are equal to those generated by $k \oplus \delta_j$, we set $\xi_j(k') = 1$; otherwise, $\xi_j(k') = 0$. If we let $T_{k'} = \sum_{j=1}^{N} \xi_j(k')$, then after each possible value k' of $k^{(m_i)}$ has been tested, we determine the k' with $T_{k'} > T_{k''}$ for all k'' where $k'' \neq k'$ to be the correct $k^{(m_i)}$. Increase i by 1 and return to Step 2.

Step 3. If the initial d_{m_4}values of the output sequence generated by $k' \oplus \delta_j^{(m)}$ are

equal to those generated by $k \oplus \delta_j$ for each j with $1 \le j \le N$, we output $k^{(m_4)}$ as a decision for the key of the *ZLL* cipher. Otherwise, declare Algorithm 2 has failed. The algorithm is terminated.

Theorem 1. The computing complexity of Algorithm 2 is 2^{25} times in computing chaotic maps on average.

P r o o f. The average computing complexity of Algorithm 2 is the sum of average computing complexity of attack on $k^{(8)}, k^{(16)}, k^{(24)}$ and $k^{(32)}$. Suppose the fundamental unit of computing complexity is the time of finishing the computing of $f_a(x)$ and $T(x)$. Then, the average computing complexity of attack on $k^{(8i)}$ is about $d_{8i} \times N \times 2^{16}/2$ for $i = 1, 2, 3$. So the average computing complexities of attack on $k^{(8)}, k^{(16)}, k^{(24)}$ are $1.25 \times 2^{22},\ 1.09 \times 2^{24},\ 1.88 \times 2^{24}$, respectively. Under the assumption that the coincidence degrees of test keys of $k^{(32)}$ should $\ge d_{24}$, we conclude that the probability is approximately $2^{-(j-1)(d_{32}-d_{24})-i}$. Then, a test key of $k^{(32)}$ is determined to be false just after $j-1$ output sequences and the i initial signals of the jth output sequence have been tested. Hence, the average computing complexity of attack on $k^{(32)}$ is

$$\sum_{j=1}^{80} \sum_{i=1}^{d_{32}-d_{24}} [(j-1)d_{32} + d_{24} + i] \times 2^{16} \times 2^{-(j-1)(d_{32}-d_{24})-i} = 1.76 \times 2^{19}.$$

So, the average computing complexity of Algorithm 2 is

$$1.25 \times 2^{22} + 1.09 \times 2^{24} + 1.88 \times 2^{24} + 1.76 \times 2^{19} \approx 1.67 \times 2^{25}. \qquad \square$$

Theorem 2. Let k be the key of the *ZLL* cipher and k' be the output of Algorithm 2. Denote $k^{(8)}$ by k_8, and $k^{(8i)}$ by $(k^{(8(i-1))}, k_{8i})$ for $2 \le i \le 4$. Similarly, Denote $k'^{(8)}$ by k'_8, and $k'^{(8i)}$ by $(k'^{(8(i-1))}, k'_{8i})$ for $2 \le i \le 4$. Then, we have

$$p(k = k') = p(k'_8 = k_8) \cdot p(k'_{16} = k_{16} | k'_8 = k_8) \cdot p(k'_{24} = k_{24} | (k'_8, k'_{16}) = k^{(16)}) \cdot p(k'_{32} = k_{32} | (k'_8, k'_{16}, k'_{24}) = k^{(24)}).$$

Let $k = (a, b)$ be a key of the *ZLL* cipher, and $k' = (a', b')$ be a test key of the *ZLL* cipher. Put

$$a = \sum_{i=1}^{n} a_i/2^i, \quad b = \sum_{i=1}^{n} b_i/2^i, \quad a' = \sum_{i=1}^{n} a'_i/2^i, \quad b' = \sum_{i=1}^{n} b'_i/2^i$$

and define

$$d(k, k') = \max\{m : a_i = a'_i \text{ and } b_i = b'_i \text{ for any } i \text{ with } 1 \le i \le m\}.$$

The larger the $d(k, k')$ is, the less the difference between k and k' is, hence the greater the coincidence degree of k' is. So, we may classify the test keys by $d(k, k')$, and obtain the probability distribution of the coincidence degrees of test keys in each

class by test methods. Consequently, we can compute the success rate of Algorithm 2 with the aid of Lemma 2 and Theorem 2.

To verify Algorithm 2, we have made 100 trials with independent and random keys for the *ZLL* cipher with a 64-bit key on a Pentium IV 2.5 GHz PC. For each trial, it took about 7 minutes and 15 seconds to recover the key of the *ZLL* cipher at a success rate of 0.85 on average. The success rate will be 0.56 if N=50, and will be 0.61 if N=60. Hence, we may increase N appropriately in order to increase the success rate.

We should attack the *ZLL* cipher with multiple candidates by modifying Algorithm 2 so as to enhance the success rate if N is small.

Algorithm 3.

Step 1. Let $N = 50$, $d_8 = 2$, $d_{16} = 7$, $d_{24} = 12$, $d_{32} = 20$ and $m_i = 8i$ for $1 \le i \le 4$. Denote by M_i the number of elements in the sets Ω_i for $1 \le i \le 3$ and set $M_1 = 4,\ M_2 = 3,\ M_3 = 4$. Set $t_i = 1$ for $1 \le i \le 3$. Set $i = 1$.

Step 2. Attack $k^{(m_i)} = (a^{(m_i)}, x_0^{(m_i)})$.
If $i = 1$, we use $k' = (a', b')$ as a test value of $k^{(8)}$ with a' and b' being selected from $\{0,1\}^8$ in turn. If $2 \le i \le 4$, denote by Ω_{i-1} the set of candidates for $k^{(m_{i-1})}$. Use $k'_{i-1} = (a_{i-1}, b_{i-1}) \in \Omega_{i-1}$ as the t_{i-1}th candidate for $k^{(m_{i-1})}$. Now, we use $k' = (a', b')$ as a test value of $k^{(m_i)}$ with the most significant m_{i-1} bits of a' fixed as a_{i-1}, the most significant m_{i-1} bits of b' fixed as b_{i-1}, and the lower $m_i - m_{i-1}$ bits of a' and b' being selected from $\{0,1\}^{m_i - m_{i-1}}$ in turn. If $i = 4$, go to Step 3. Otherwise, for any j with $1 \le j \le N$, if the initial d_{m_i} values of the output sequence generated by $k' \oplus \delta_j^{(m)}$ are equal to those generated by $k \oplus \delta_j$, we set $\xi_j(k') = 1$; otherwise, $\xi_j(k') = 0$. If we let $T_{k'} = \sum_{j=1}^{N} \xi_j(k')$, then after each possible value k' of $k^{(m_i)}$ has been tested, we determine $k'_1, \ldots, k'_{M_i}$ with $T_{k'_1} \ge T_{k'_2} \ge \cdots \ge T_{k'_{M_i}} \ge T_{k'}$ for all k', where $k' \ne k'_1, \ldots, k' \ne k'_{M_i}$, to be the first, second, ..., M_ith candidate for $k^{(m)}$ and set $\Omega_i = \{k'_1, \ldots, k'_{M_i}\}$. Increase i by 1 and return to Step 2.

Step 3. If the initial d_{m_4} values of the output sequence generated by $k' \oplus \delta_j^{(m)}$ are equal to those generated by $k \oplus \delta_j$ for each j with $1 \le j \le N$, we output $k^{(m_4)}$ as a decision for the key of the *ZLL* cipher and terminate the algorithm. Otherwise, increase t_3 by 1 and return to Step 2 when $t_3 \le M_3$. If $t_3 = M_3 + 1$, set $t_3 = 1$, $i = 2$ and increase t_2 by 1. Return to Step 2 when $t_2 \le M_2$. If $t_2 = M_2 + 1$, set $t_2 = 1$, $i = 1$ and increase t_1 by 1. Return to Step 2 when $t_1 \le M_1$. If $t_1 = M_1 + 1$, we declare that Algorithm 2 has failed and the algorithm is terminated.

For Algorithm 3, we note that the first candidate of k must be tested firstly. Hence, if the first candidates of $k^{(8)}$, $k^{(16)}$, $k^{(24)}$, $k^{(32)}$ are all correct, the computing complexity of the multiple candidates scheme must be the same as that of Algorithm 2. In our 100 trials for Algorithm 3 on a Pentium IV 2.5 GHz PC, it took 11 minutes and 39 seconds on average to recover the key of the *ZLL* cipher with a success rate at 0.94. Furthermore, it took 7 minutes and 48 seconds on average when the attack

succeeded. If N=80, the success rate of the Algorithm 3 is almost 1 and it only took 8 minutes and 45 seconds on average.

The performances of Algorithm 2 and Algorithm 3 may be changed greatly if the methods of dividing key blocks and $d_{m_1}, d_{m_2}, d_{m_3}, d_{m_4}$ are changed. If the mode of dividing key blocks or the choice of d_i is not proper, the success rate will decrease rapidly. The choice of $d_{m_1}, d_{m_2}, d_{m_3}, d_{m_4}$ should be based on the following principle: Under the condition that the attack has a satisfactory success rate, one should try to make d_i as small as possible so as to reduce the average computing complexity. From the results of experiments, we find that the success rate will reach the maximum when d_{m_i} equals the expectation of the coincidence degree of $k^{(m_i)}$.

After the N known output sequences have been tested in Algorithm 1, the count of $k^{(m)}$ is about $Np(n_m \geq d_m)$, and that of the wrong test key k' is about $Np_{k'}$, where $p_{k'}$ is the probability that the coincidence degree of k' is not less than d_m. Note that $Np(n_m \geq d_m)$ will be quite distinct from $Np_{k'}$ as N increases. So we can distinguish $k^{(m)}$ from its possible values at a high success rate. In other words, for a chaotic cipher with a large key size, the correct key can be derived with a small computing complexity and a high success rate by using the related-key attack. Moreover, as long as $p(n_m \geq d_m)$ is distinct from $p_{k'}$, one can realize a related-key attack effectively even if $p(n_m \geq d_m)$ is not big enough. Hence, the limitations of divide-and-conquer attacks to chaotic ciphers are overcome efficiently.

4. ATTACKS BY OTHER KINDS OF RELATED KEYS

The related-key attack proposed above is based on the following facts. The equation $(k \oplus \delta_i)^m = k^{(m)} \oplus \delta_i^{(m)}$ ensures that $(k \oplus \delta_i)^{(m)}$ could be obtained with known $k^{(m)}$ and $\delta_i^{(m)}$ for any i with $1 \leq i \leq N$. When the related keys of a chaotic cipher are in the forms of $k + \delta_1, k + \delta_2, \ldots, k + \delta_N$, where $+$ is addition modulo 2^n or addition modulo 2^n word-wise for $n \geq 1$, the chaotic map $f(k+\delta_i)$ also has the property that the less-significant bits of the input have little effect on the most significant bits of the output. This shows that the output sequence generated by $k^{(m)} + \delta_i$ and that generated by $k + \delta_i$ will be the same in the initial signals with a high probability. The results of experiments indicate that the related-key can be similarly applied in these cases.

Let k be a shared key by Alice and Bob, and δ be a random number being transferred publicly. Suppose Alice and Bob want to use $g(k, \delta)$ as their session key to encrypt their messages. For the same reason as above, if $g : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n$ has the property that the less-significant bits of the input have little effect on the most significant bits of the output, one can implement the related-key attacks on the chaotic cipher by using the related keys in the forms of $g(k, \delta_1), g(k, \delta_2), \ldots, g(k, \delta_N)$. Hence, in a session key protocol, the cryptographical property of function g is highly important for chaotic ciphers to resist related-key attacks. In other words, if the distribution of coincidence degrees of a chaotic cipher is not reasonable and may be detected efficiently, the chaotic cipher may be broken easily in some cases. The distribution of coincidence degrees of chaotic ciphers should be one of the measures of security for chaotic ciphers.

5. CONCLUSIONS

We have presented a related-key attack on chaotic ciphers in this paper. This method utilizes simultaneously the output sequences generated by related keys to attack on a chaotic cipher. Hence, the efficiency of the divide-and-conquer attack is enhanced greatly and the limitations of the divide-and-conquer attack [2, 3] on chaotic ciphers are overcome. As an example, we have realized the related-key attacks on the *ZLL* chaotic cipher with a 64-bit key.

The results of this paper indicate that it is much likely to obtain the total bits of the key for a chaotic cipher even with little information leaked by the distribution of coincidence degrees of a chaotic cipher. Hence, it is important for the designers of chaotic ciphers to ensure a reasonable distribution of the coincidence degrees. Although the related-key attacks presented in this paper aim at the iterated chaotic ciphers, which sets the initial conditions and parameters as keys, the ideals are also applicable to other types of chaotic ciphers.

(Received September 30, 2007.)

REFERENCES

[1] D. R. Frey: Chaotic digital encoding: An approach to secure communication. IEEE Trans. Circuits and Systems *40* (1993), 10, 660–666.
[2] Ch. Jin: The analysis of a block cipher algorithm based on chaos (in Chinese). China Engnrg. Sci. *3* (2001), 6, 1066–1070.
[3] Ch. Jin and H. Gao: Analysis of two stream ciphers based on chaos (in Chinese). Acta Electronic Sinica *34* (2004), 7, 1066–1070.
[4] S. Li, X. Mou, Z. Ji, and J. Zhang: Cryptanalysis of a class of chaotic stream ciphers (in Chinese). J. Electronics & Information Technology *25* (2003), 4, 473–479.
[5] M. Matsui: Linear cryptanalysis method for DES cipher. In: Advance in Cryptology — Eurocrypt'93 (Lecture Notes in Control Systems 765.) Springer-Verlag, Berlin 1994.
[6] H. Zhou and X.-T. Ling: Problems with the chaotic inverse systems encryption approach. IEEE Trans. Circuits and Systems-I *44* (1997), 3, 268–271.
[7] H. Zhou, J. Luo, and X. Ling: Generating nonlinear feedback stream ciphers via chaotic systems (in Chinese). Acta Electronic Sinica *25* (1997), 10, 57–60.
[8] H. Zhou, J. Yu, and X. Ling: Theoretical design of chaotic feed forward stream cipher (in Chinese). Acta Electronic Sinica *26* (1998), 1, 98–101.

Yang Yang and Chenhui Jin, Zhengzhou Information Science and Technology Institute, Zhengzhou 450004. China.
e-mails: yangyang_wawa@sina.com, jinchenghui@126.com