Stochastic Cryptanalysis of Crypton

Published in Fast Software Encryption - FSE 2000 - LNCS 1978

Marine Minier and Henri Gilbert

France Télécom R&D
38-40, rue du Général Leclerc
92794 Issy les Moulineaux Cedex 9 - France
Tel: +33 1 45 29 44 44

Abstract. Crypton is a 12-round blockcipher proposed as an AES candidate by C.H. Lim in 1998. In this paper, we show how to exploit some statistical deficiencies of the Crypton round function to mount stochastic attacks on round-reduced versions of Crypton. Though more efficient than the best differential and linear attacks, our attacks do not endanger the practical security offered by Crypton.

1 Introduction

Crypton [Li98] is a 12-round blockcipher which was submitted by C.H. Lim as one of the 15 candidates at the first Advanced Encryption Standard conference in August 1998. Crypton offers several interesting features. The encryption and decryption processes are strictly identical up to the key schedule (a quite remarkable property given the substitution/permutation structure of the cipher). Crypton is highly parallelizable and flexible, and thus well suited for efficient implementation on nearly any hardware or software platform. Moreover, Crypton provides some provable resistance against linear and differential cryptanalysis.

The main cryptanalytic results obtained on Crypton so far are the analysis of the best differential and linear attacks by the algorithm designer [Li98], a transposition of the square attack to the 6-round Crypton by C. D'Halluin et al. [Hal99], the discovery of some weak keys by Vaudenay [Ba99], and statistical observations contained in an annex of [Ba99].

C.H. Lim introduced in 1999 [Li99] a modified Crypton (denoted by Crypton v1.0) with a new keyschedule and new S-boxes designed as to lower the number of high probability differential and linear characteristics. Though most of this paper is dedicated to the analysis of the initial version of Crypton, the impact of the Crypton v1.0 S-box modifications is also discussed in the last Section.

We present here two attacks of round reduced versions of Crypton (up to 8 rounds) which are based on iterative statistical properties of the round function. A short outline of preliminary (unquantified) versions of these attacks has been already published in a paper presented at the second AES conference [Ba99]. Based on extra analysis and computer experiments, we provide here more precise assessments of the statistical biases and the performance of these attacks.

Both attacks can be broadly described as stochastic attacks. The block values (or a subset of the difference values if the attack uses differential statistics) are

B. Schneier (Ed.): FSE 2000, LNCS 1978, pp. 121–133, 2001.

Published in *Fast Software Encryption – FSE 2000*, LNCS 1978

partitioned into a small number of classes, and any k-rounds partial encryption is modeled as a stochastic process (i.e. a sequence of random variables providing the class values at the boundaries of the various rounds). Each round is characterized by one matrix of key dependent or key-independent transition probabilities between input and output classes. Under the heuristic assumption that the above process is nearly markovian, the behaviour of such a k-rounds partial encryption is well approximated by the product of the k one-round transition probabilities matrices [1].

We do not claim that the idea of that kind of generalization of more traditional "characteristics-based" attacks is new : Murphy et al.' likehood estimation [Mu], Vaudenay's χ^2 cryptanalysis[Va95], Lai and Massey's modeling of markovian ciphers [LM91], Harpes and Massey's partition cryptanalysis [HM97]provide frameworks which describe similar generalizations. However, there are not yet numerous examples of ciphers where such approaches bring some real added value. We believe Crypton is a good example of an algorithm for which this is the case.

A stochastic attack is feasible if there exists a partition of the blocks (or of the considered subset of difference values) such that for each key value, the transition probabilities among classes differ substantially from the transition probabilities a random permutation of the block or difference values would provide. The key cryptanalytic issue consists in finding such a suitable partition. In the case of Crypton, the partitions of the block values and of difference values we are using are based upon some invariance properties of the linear part of the round function which involve only four "active" bytes of the inputs or outputs to the non linear part.

The rest of this paper is organized as follows : Section 2 briefly summarizes the Crypton cipher. Section 3 presents the iterative statistical properties which form the starting point for our attacks. Section 4 presents a stochastic attack based upon a partition of difference values into 257 classes. Section 5 presents a stochastic attack based upon a partition of blocks into 16 classes. Section

[1] Stochastic attacks represent a generalization of "Characteristics-based attacks", such as linear attacks, differential attacks, or truncated differential attacks. As a matter of fact, characteristics based attacks are based upon a partition of block or difference values at each round into only two classes. For instance, in linear attacks, blocks are partitioned according to the binary value of a fixed linear combination of the key bits. In differential attacks one considers the unbalanced partition of the differences between one single difference value on one hand and the complementary set of all other difference values on the other hand. In truncated differential attacks, one considers a partition of difference values according to the characteristic function of a set of difference values satisfying certain constraints, etc. In characteristics based attacks of blockciphers, one single transition probability, namely the probability of the considered characteristic, entirely determines the (2x2) transition probabilities matrix associated with a partial encryption. The stochastic attacks considered in this paper are not "characteristics-based" because they involve partitions in strictly more than two classes.

Published in *Fast Software Encryption - FSE 2000* - LNCS 1978

6 investigates the (quite positive) impact of these modifications introduced in Crypton v1.0 and Section 7 concludes the paper.

2 An Outline of Crypton

Crypton encrypts 128-bit blocks under the control a key of length up to 256 bits. The nominal value of the r number of rounds is 12. The algorithm consists of the encryption function itself and a keyschedule that derives $(r+1)$ 128-bit subkeys from the key and an encryption function. Since the attacks presented here do not rely at all upon the properties of the key schedule, we do not describe it here.

The encryption function consists of r rounds surrounded by an input transformation (defined by an XOR between the plaintext block and the first subkey) and an output transformation (defined as a fixed modulo 2 linear mapping).

Let us represent a 128-bit block A by :

$$A = \begin{pmatrix} a_{0,3}\ a_{0,2}\ a_{0,1}\ a_{0,0} \\ a_{1,3}\ a_{1,2}\ a_{1,1}\ a_{1,0} \\ a_{2,3}\ a_{2,2}\ a_{2,1}\ a_{2,0} \\ a_{3,3}\ a_{3,2}\ a_{3,1}\ a_{3,0} \end{pmatrix} \begin{matrix} A[0] \\ A[1] \\ A[2] \\ A[3] \end{matrix}$$

where each $a_{i,j}$ is a byte.

One round consists of a byte substitution γ, followed by a bit permutation π, a bytes transposition τ and a subkey addition σ. γ (γ_o for odd round, γ_e for even round) uses two S-boxes S_0 and S_1. We only describe of γ_o ; to obtain γ_e one just needs to exchange S_0 and S_1.

$$\begin{pmatrix} b_{0,3}\ b_{0,2}\ b_{0,1}\ b_{0,0} \\ b_{1,3}\ b_{1,2}\ b_{1,1}\ b_{1,0} \\ b_{2,3}\ b_{2,2}\ b_{2,1}\ b_{2,0} \\ b_{3,3}\ b_{3,2}\ b_{3,1}\ b_{3,0} \end{pmatrix} \xleftarrow{\gamma_o} \begin{pmatrix} S_1(a_{0,3})\ S_0(a_{0,2})\ S_1(a_{0,1})\ S_0(a_{0,0}) \\ S_0(a_{1,3})\ S_1(a_{1,2})\ S_0(a_{1,1})\ S_1(a_{1,0}) \\ S_1(a_{2,3})\ S_0(a_{2,2})\ S_1(a_{2,1})\ S_0(a_{2,0}) \\ S_0(a_{3,3})\ S_1(a_{3,2})\ S_0(a_{3,1})\ S_1(a_{3,0}) \end{pmatrix}$$

π (π_o for odd rounds, π_e for even rounds) is a bit permutation. We only describe effects of π_o for odd rounds, effects of π_e are similar. π_o is given by

$$\begin{aligned} T &= A[0] \oplus A[1] \oplus A[2] \oplus A[3], \\ B[0] &\leftarrow (A[0] \wedge MI_0) \oplus (A[1] \wedge MI_1) \oplus (A[2] \wedge MI_2) \oplus (A[3] \wedge MI_3) \oplus T, \\ B[1] &\leftarrow (A[0] \wedge MI_1) \oplus (A[1] \wedge MI_2) \oplus (A[2] \wedge MI_3) \oplus (A[3] \wedge MI_0) \oplus T, \\ B[2] &\leftarrow (A[0] \wedge MI_2) \oplus (A[1] \wedge MI_3) \oplus (A[2] \wedge MI_0) \oplus (A[3] \wedge MI_1) \oplus T, \\ B[3] &\leftarrow (A[0] \wedge MI_3) \oplus (A[1] \wedge MI_0) \oplus (A[2] \wedge MI_1) \oplus (A[3] \wedge MI_2) \oplus T, \end{aligned}$$

where $MI_0 = c0300c03, MI_1 = 03c0300c, MI_2 = 0c03c030, MI_3 = 300c03c0$ (in hexadecimal). We can notice that if we omit the addition with T, π results in permutations of 4 2-bit words in each of the 16 2-bit words columns of the A matrix.

The bytes transposition τ just consists in exchanging all $a_{i,j}$ and $a_{j,i}$ pairs of bytes in the A matrix, and the key addition σ_K just consists of an exclusive-or between A and a 128-bit subkey K.

Published in Fast Software Encryption - FSE 2000 - LNCS 1978

3 Statistical Properties of the Round Function

In this Section we introduce two iterative properties of the $\tau \circ \pi$ linear part of the Crypton round function which involve only four bytes of the inputs or outputs to the non linear part - and thus represent limitations in the diffusion achieved by $\tau \circ \pi$. These two properties, which are outlined in [Ba99] are to some extent dual of each other, and can be summarized as follows : (1) some $\tau \circ \pi$ input values equal to zero except on at most four bytes are transformed into output values equal to zero except for at most four other bytes ; (2) for certain sets of 4 of the 16 $\tau \circ \pi$ input bytes and certain associated sets of 4 of the 16 $\tau \circ \pi$ output bytes, there exist a (linear) four bytes to 4 bits function Φ such that the images by Φ of the four input bytes and the four output bytes are equal. Property (1), as applied to difference values in the encryption of pairs of plaintexts, can be seen as an iterative truncated differential whereas property (2) can be to a certain extent compared to a set of iterative linear characteristics.

We introduce some additional notation to split each $a_{i,j}$ byte of an A block into four 2-bit words : $a_{i,j} = (a_{3,i,j}, a_{2,i,j}, a_{1,i,j}, a_{0,i,j})$ where i is the line index and j the column index. $i \in [0,3]$ and $j \in [0,3]$. $a_{k,i,j}$ will be the 2-bits word of line i, column j and position k. We define the "square" associated with the (k,i,j) triplet as the $((a_{k,i,j}, a_{k,i+2,j}, a_{k,i,j+2}, a_{k,i+2,j+2}))$ quartet of 2-bit words. Note that indexes are implicitly taken modulo 4. Under some conditions, squares are preserved (up to a modification of the associated (k,i,j) indexes) by the γ_o, γ_e, π_o, π_e and τ functions.

3.1 Property (1)

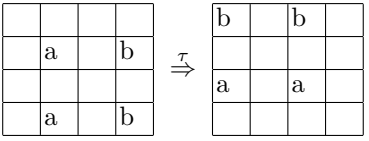
One can see that if we omit T, π_o and π_e just permute the $a_{k,i,j}$ 2-bit words (more precisely, they only permute the j indexes), and thus they transform any square associated to a (k,i,j) triplet into the same square associated with another (k,i',j') triplet. This stays valid for the real π_o and π_e functions under the two additional conditions (i) $a_{k,i,j} = a_{k,i+2,j}$ and (ii) $a_{k,i,j+2} = a_{k,i+2,j+2}$. Squares are also preserved by the τ function (up to a modification of the i and j indexes). Moreover, under conditions (i) and (ii), "twin squares" associated with two (k,i,j) and $(k+2,i,j)$ triplets of indexes are transformed into two other "twin squares" associated with two (k,i',j') and $(k+2,i',j')$ triplets of indexes.

In order to cryptanalytically exploit property (1), we can consider special difference values equal to zero except on two "twin squares" (k,i,j) and $(k+2,i,j)$ and such that the (i) and (ii) conditions are satisfied for both squares. For instance, squares of difference values of the following form stay entirely invariant under the π_o mapping :

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0\ 0 0 x_1 x_2 0 0 y_1 y_2 & 0\ 0 0 x'_1 x'_2 0 0 y'_1 y'_2 \\ 0 & 0 & 0 & 0 \\ 0\ 0 0 x_1 x_2 0 0 y_1 y_2 & 0\ 0 0 x'_1 x'_2 0 0 y'_1 y'_2 \end{pmatrix} = \begin{pmatrix} 0\ 0\ 0\ 0 \\ 0\ a\ 0\ b \\ 0\ 0\ 0\ 0 \\ 0\ a\ 0\ b \end{pmatrix}$$

Published in Fast Software Encryption - FSE 2000 - LNCS 1978

Moreover, for the above example, τ just modifies bytes positions :



In the sequel we will keep using a representation of twin squares at the byte level (as in the above representation of τ) to more compactly depict the effects of π_o, π_e and τ.

3.2 Property(2)

Let us consider any two twin squares with a (k, i, j) triplet and a $(k+2, i, j)$ triplet of indexes. As seen before, if there was no T term in the π definition, these two twin squares would be just displaced by the $\tau \circ \pi$ linear function (without any change in the quartet values) to two twin squares associated with (k, i', j') and $(k+2, i', j')$ triplets of indexes. Now if we take the T term into account, we can see that the $\phi_{k,i',j'}$ 2-bit XOR of the four 2-bit words of the (k, i', j') output square is still equal to the $\phi_{k,i,j}$ XOR of the four 2-bit words of the (k, i, j) input square (just because the additional terms introduced by T twowise compensate). The same property obviously holds for the squares associated with the $(k+2, i, j)$ input triplet and the $(k+2, i', j')$ output triplet : $\phi_{k+2,i,j} = \phi_{k+2,i',j'}$. Thus, if we denote by $\Phi_{k,i,j}$ the 4-bit word $\phi_{k,i,j} \oplus 4.\phi_{k+2,i,j}$, we can summarize the obtained invariance property by the equality $\Phi_{k,i,j} = \Phi_{k,i',j'}$. In other words, the 4-bit linear combination $\Phi_{k,i,j}$ of the four bytes involved in two twin squares is kept invariant by the linear part of Crypton (up to a change of considered four bytes positions).

In Section 5, we will mount an attack based upon partitioning block values in 16 classes according to the value of such a 4-bit Φ values.

In summary, we have identified in properties (1) and (2) some correlations between the input and the output of the $\tau \circ \pi$ part of the round function which involve only 4 "active" input and output bytes. It remains to study how much correlation is left on entire rounds if one also takes the non linear part of the Crypton round function into account.

4 Stochastic Attack Using Differential Properties

4.1 Computing Transition Probabilities

The cryptanalysis is based upon property (1) and uses differences of the form described in part 3.1. That's why, in order to describe elements which stay invariant by π and τ, we introduce two sets of possible difference values at the byte level :

Published in Fast Software Encryption, FSE 2004, LNCS 1978

$D_1 = \{0,1,2,3,16,17,18,19,32,33,34,35,48,49,50,51\}$
$D_2 = \{0,4,8,12,64,68,72,76,128,132,136,140,192,196,200,204\}$

We include zero in both sets although in practice this difference value can't appear in an attack. D_1 corresponds to all possible choices of the 00xx00xx byte and D_2 to all possible choices of the xx00xx00 byte. Those sets permit us to create "twin square" invariant under the linear part. We denote by (δ_1, δ_2) differences of the form

$$\Delta_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \delta_1 & 0 & \delta_2 \\ 0 & 0 & 0 & 0 \\ 0 & \delta_1 & 0 & \delta_2 \end{pmatrix}$$

where Δ_1 is represented by a 4×4 matrix of bytes with $\delta_1 \in D_1$ and $\delta_2 \in D_1$ or of the form

$$\Delta_2 = \begin{pmatrix} 0 & \delta_1 & 0 & \delta_2 \\ 0 & 0 & 0 & 0 \\ 0 & \delta_1 & 0 & \delta_2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

with $\delta_1 \in D_2$ and $\delta_2 \in D_2$. The above Δ_1 and Δ_2 difference matrices are just examples of all possible difference values that can be taken. As a matter of fact, (δ_1, δ_2)can be any "twin square" difference satisfying conditions (i) and (ii) of Section 2.
We can say by the invariance properties seen above that there exist δ'_1 and δ'_2 in D_1 such that

$$\tau(\pi_i(\Delta_1)) = \begin{pmatrix} \delta'_2 & 0 & \delta'_2 & 0 \\ 0 & 0 & 0 & 0 \\ \delta'_1 & 0 & \delta'_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

So we can deduce that with a certain probability p, the "twin squares" represented by the (δ'_1, δ'_2) pair could result, after being passed through S-boxes, into a (δ_3, δ_4) pair such that

$$\gamma_i(\tau(\pi_i(\Delta_1))) = \begin{pmatrix} \delta_3 & 0 & \delta_4 & 0 \\ 0 & 0 & 0 & 0 \\ \delta_3 & 0 & \delta_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

An S-box output with this form permits us to use another time the invariance property of $\gamma_i(\tau(\pi_i(\Delta_1)))$ into π_i and τ on an other "twin square". We obtain, with a certain probability p', the invariance of Δ_1 on one entire round. With the same construction, we can establish the invariance of Δ_2 on one round with a certain probability p".
For all (δ_1, δ_2) pairs of D_1 and all (δ_3, δ_4) pairs of D_1, we can compute the probability of getting a (δ_3, δ_4) output difference from a (δ_1, δ_2) input difference after

Published in *Fast Software Encryption - FSE 2005* - LNCS 1978

one round. All probabilities are key-independent and only depend on differential properties of S-boxes.

$$\begin{aligned} Pr[\Delta' = (\delta_3, \delta_4) | \Delta = (\delta_1, \delta_2)] &= Pr[\delta_1 \rightarrow \delta_3].Pr[\delta_2 \rightarrow \delta_3] \\ &\quad .Pr[\delta_1 \rightarrow \delta_4].Pr[\delta_2 \rightarrow \delta_4] \\ &= \frac{d_{\delta_1\delta_3}}{256} \cdot \frac{d_{\delta_2\delta_3}}{256} \cdot \frac{d_{\delta_1\delta_4}}{256} \cdot \frac{d_{\delta_2\delta_4}}{256} \end{aligned}$$

where $d_{\delta_i\delta_j}$ is the number of bytes such that : $\delta_j = S(x) \oplus S(x \oplus \delta_i)$, S represents the appropriate S-box of Crypton. We obtain a 256×256 matrix M of transition probabilities over one round by computing p for 256 couples of possible input values (δ_1, δ_2) and for 256 couples of possible output values (δ_3, δ_4) with $\delta_1, \delta_2, \delta_3, \delta_4 \in D_1$. Columns of M represent all probabilities of transition between input (δ_1, δ_2) and output (δ_3, δ_4).We use the same method to compute probabilities of transition associated with D_2.

Now let us consider differentials over two rounds, i.e. transition between an input value (δ_1, δ_2) and an output value (δ_3, δ_4). A lower bound on the probabilities of such differentials is provided by summing up the probabilities of all intermediate values which belong to D_1 :

$$p_{i,j} = \sum_{i=0}^{255} \sum_{j=0}^{255} p_{i,k} p_{k,j}$$

where $p_{i,j}$ represent coefficients of M. With this relation, we consider all the possible intermediate values. There exists a stochastic dependence between the difference values at the various rounds. We make the heuristic assumption that the sequence of difference values over several rounds satisfies the "Markov property", i.e. the distribution of probabilities of the differences at the output of any round only depend on the distribution of probabilities of the differences at the input of the same round. So, we can compute M^n, which represents key-independent transition probabilities between input and output differences on an n-rounds scheme. For example, to compute transition probabilities for 6 rounds, we compute M^6. One cryptanalytically meaningful measure of the unbalanceness of the obtain matrix M consists of computing the sums of the probabilities in each column. The performance of the attack hereafter depends upon the value obtained for the "best column", i.e. the best pair of input difference values (δ_1, δ_2).

4.2 Attack Procedure

We present here an attack on a complete eight-round Crypton (i.e. taking into account the first addition of subkey K_0 and the final transformation). Under the heuristic assumptions summarized above, we can compute probabilities of best couples of difference on several rounds. In particular we obtain the following figures for 6 inner rounds of Crypton : if the input difference is the (18,

Published in: Fast Software Encryption - FSE 2000 - LNCS 1978

18) pair of Δ_1 values. The probability that the output be a (δ_1, δ_2) pair with $(\delta_1,\delta_2) \in D_1 \times D_1$ is $2^{-120.72}$. In the same way, if input is the (128, 128) pair of D_2 values, probability that the output be a (δ_1, δ_2) pair of D_2 values is $2^{-112.62}$.

This result on six inner rounds can be used to attack an eight inner rounds attack. Due to the late occurrence of the σ key addition in the first inner round, we can control differences at the output of the first occurrence of γ, and thus the first round can be treated just as a keyless "initial permutation". A 1R-attack permits to also gain the eighth round. So, we can exploit the a priori known 6-rounds properties in a chosen plaintext attack to obtain some bits of information on the last round key K_8.

In order to efficiently generate pairs of chosen plaintexts which difference is equal to the $\Delta_2 = (128, 128)$ value at the output of the first occurrence of γ, we group plaintexts in structures. Two 128-bits elements belong to the same structure if and only their images by γ are equal except for some of the 16 bits of Δ_2. Each structure has 2^{16} elements. We know that, if two X and X' plaintexts belong to the same structure, then with probability $2^{-112.62}$, the correponding inputs to the the eighth round are of the form Y and $Y \oplus \Delta_2$ (where Δ_2 is of D_2 values). We only consider those (X, X') pairs such that the C$\oplus$C' ciphertext difference is null everywhere except at most on the four non zero bytes of the Δ_2 Y differences. For those pairs which pass that filtering condition, we go up the last round starting from C and C' and checking whether the resulting $Y \oplus Y'$ has the right form. Some couples are "false" alarms, i.e. pass the filtering condition but don't belong to our set (this happens with a probability equal to 2^{-96}). Once selected "good" couples, we test the possible values of four bytes of key in going up at the end of eighth round. The candidate value which appears most often is the right one. We obtain four bytes of information on K_8 [2].

We can go up through the final transformation Φ_e because it does not change output values of eighth round. Taking into account the first key-addition σ_0 just increases the number of "false" alarms. The number of plaintexts to cipher is $N = 2^{112.62}$. But we must take a security for "false" alarms. We claim that taking $N = 2^{114.62}$ is enough. So we can obtain 32 bits of information of K_8 by ciphering $2^{114.62}$ couples X and X$\oplus\Delta$ in a complete eight-rounds version of Crypton. Complexity of this attack is $2^{114.62}$ encryptions and 2^{96} additional computations.

This attack is faster than an exhaustive search and than all differential attacks. As a matter of fact, it can be shown that the probability of the best characteristic for an eigth-round attack is 2^{-120}.

[2] The same procedure can be repeated with other square locations (at the expanse of a slight increase of the N number of chosen plantexts) to entirely derive K_8. Once the last subkey has been entirely derived, the same procedure can be repeated (with the same plaintexts) to derive the entire expanded key.

Published in Fast Software Encryption, FSE 2000, LNCS 1978

5 Stochastic Cryptanalysis of Crypton Using a Partition of Blocks in 16 Classes

5.1 Computation of Transition Probabilities

The cryptanalysis presented in this Section is based on Property (2) of Section 2. Let us consider inside an intermediate block

$$A = \begin{pmatrix} * & * & * & * \\ * & X & * & Y \\ * & * & * & * \\ * & Z & * & T \end{pmatrix}$$

encountered in the Crypton encryption process, a (X, Y, Z, W) quartet of bytes associated with a given (i, j) pair of indices. We can partition the blocks space into 16 classes according to the 4-bit value $\Phi_0[X, Y, Z, W] = \Phi_{0,i,j}$ (or alternatively into 16 other classes according to the 4-bit value $\Phi_1[X, Y, Z, W] = \Phi_{1,i,j}$).

Property (2) states that the linear part of Crypton leaves Φ_0 and Φ_1 values unchanged (provided that the final Φ_0 or Φ_1 value is computed from four appropriately selected bytes associated with a (i', j') pair of indices deduced from (i, j)).

It is easy to see that the σ key addition transformation just results in XORing the $\Phi_0[X, Y, Z, W]$ (resp $\Phi_1[X, Y, Z, W]$) value associated with a (X, Y, Z, W) quartet of bytes with a $\Phi_0[K_X, K_Y, K_Z, K_W]$ (resp $\Phi_1[K_X, K_Y, K_Z, K_W]$) 4-bit constant which depends on four subkey bytes.

We now investigate the effect of the S_0 and S_1 S-boxes of the γ non linear part of the Crypton round function on the $\Phi_0[X, Y, Z, W]$ (resp $\Phi_1[X, Y, Z, W]$) class values. For that purpose, for each of the S_0 and S_1 S-boxes, we compute the values

$$\begin{gathered} \#\{\mathrm{X,Y,Z,W}\in[0,255]^4 \ /\Phi_0\ [\mathrm{X,Y,Z,T}]=\mathrm{a} \\ \text{and } \Phi_0\ [S_\epsilon(\mathrm{X}),S_\epsilon(\mathrm{Y}),S_\epsilon(\mathrm{Z}),S_\epsilon(\mathrm{T})]=\mathrm{b}\}\text{-}(256)^3 \\ \text{and} \\ \#\{\mathrm{X,Y,Z,W}\in[0,255]^4 \ /\Phi_1\ [\mathrm{X,Y,Z,T}]=\mathrm{a} \\ \text{and } \Phi_1\ [S_\epsilon(\mathrm{X}),S_\epsilon(\mathrm{Y}),S_\epsilon(\mathrm{Z}),S_\epsilon(\mathrm{T})]=\mathrm{b}\}\text{-}(256)^3 \end{gathered}$$

where ϵ takes values 0 or 1, and a and b are in [0,15].These values represent biases with respect to the average value $(256)^3$. We obtain four 16×16 matrices (one for Φ_0 and S_1 (see appendix A), one for Φ_0 and S_0, one for Φ_1 and S_1, one for Φ_1 and S_0). The columns of such matrices are indexed by a and their lines by b, and the value associated with column a and line b represents (up to a multiplicative factor of $(256)^4$) the bias of the transition probability from the class associated with the Φ input value a to the class associated with the Φ output value b.

Now it clearly results from the above properties of γ, $\tau \circ \pi$, and σ that the way one entire round of Crypton affects the Φ_0 or Φ_1 values can be represented by a 16×16 matrix of transition probabilities (or equivalently of biases) obtained by multiplying one of the four above matrices by the 16×16 key-dependent

Published in Fast Software Encryption - FSE 2000 - LNCS 1978

permutation matrix which are associated with the a column and the b line is equal to 1 if $b = a \oplus \Phi_0[K_X, K_Y, K_Z, K_W]$.

Under the heuristic assumption that the behaviour of the cipher is nearly markovian, we can multiply those matrices to represent transitions of Crypton on n rounds. Of course the obtained matrix is key dependent (it depends upon 4 linear combinations of the key bits per round). However the orders of magnitude of the biases encountered in the product matrix are the same for most values of the keybits (they are larger than average for a few special values, e.g. when all 4-bit key words are equal to zero).

5.2 Attack Procedure

We present here a chosen plaintext of 8 inner rounds of Crypton (i.e. without the first key-addition and without the final transformation) which can be extended to an attack of the full 8-rounds version of Crypton (including the initial and the final transformation), with very similar performance. This basic attack is a 1-R attack based upon the above described computations of 6-rounds transition matrices (given by the product of 6 one-round transition matrices).

We select a (i, j) pair of first-round indexes and bit ϵ (0 or 1) and encrypt N chosen plaintext blocks such at the input to the first occurrence of the σ transformation (at the end of the first round), the $\Phi = \Phi(b, i, j)$ is equal to a constant 4-bit value α. We can expect the resulting Φ value associated with the input to the last round to be distributed according to unbalanced probabilities (or equivalently biases) given by the α column of the 6-rounds transition matrix.

We are using the χ^2 test in the same way as described in [HG97] to test four key bytes derived from the last round subkeys (i.e. those linear combinations of the last round subkey bits enabling to recover the four bytes involved in the Φ value of the input to the last round). For each of the 2^{32} key assumptions, we can partially decrypt the last round and compute (in at most 2^{32} operations, provided that ciphertext blocks have been first partitioned according to the value of a suitable 4-bytes word) the distribution of the Φ values at the input to the last round and the χ^2 indicator associated with the obtained distribution of 16 empiric frequencies. The obtained indicator is expected to be substantially higher for the right assumption on the four key bytes.

The N number of plaintexts required by the attack is inversely proportional to the sum of the squares of the a priori expected biases. Thus the required number of plaintexts can be deduced from the biases in the above introduced 6-round matrices. The best result are obtained with Φ_1 computations (instead of Φ_2 computations. For average values on the 6-uples of 4-bit key words, the obtained N value is close to 2^{112}. For the best 6-uple values (e.g. the null 6-uple), about 2^{104} suffice to recover the 32 last round key bits.

So in summary we obtain 32 bits of information about the last round keybit using 2^{112} chosen plaintexts. Therefore we can recover the entire last round subkey (and then, once having decrypted the last round, derive the other subkeys, using the same method) with say 2^{116} chosen plaintexts.

Published in Fast Software Encryption - FSE 2000 - LNCS 1978

6 Results Concerning Crypton v1.0

Crypton v1.0 was introduced by Chae Hoon Lim in [Li99] to modify the initial key schedule and improve the S-boxes.

Instead of using two S-boxes like the initial version, Crypton v1.0 uses four S-boxes S_0, S_1, S_2, S_3 which verify $S_0^{-1} = S_2$ and $S_1^{-1} = S_3$. The γ_o and γ_e transformations are redefined as follows :

$$B = \gamma_o(A) \Leftrightarrow b_{i,j} = S_{i+j \bmod 4}(a_{i,j})$$
$$B = \gamma_e(A) \Leftrightarrow b_{i,j} = S_{i+j+2 \bmod 4}(a_{i,j})$$

We still have : $\gamma_o^{-1} = \gamma_e$ and $\gamma_e^{-1} = \gamma_o$ and thus the encryption and decryption processes are still identical.

The new S-boxes S_0, S_1, S_2, S_3 are all designed from an 8x8 involutive S-box S, chosen for its good diffusion properties. The purpose of the replacement of the Crypton S-boxes was to lower the number of the most probable low weight differential and linear characteristics, and thus to speed up the diffusion achieved by Crypton.

According to our computer experiments, these S-box modifications very significantly improve the resistance of Crypton against the stochastic attacks presented in Sections 4 and 5.

Let us first consider stochastic cryptanalysis using differential properties on 6 rounds of Crypton v1.0. We found that if input difference is the (49, 49) pair of Δ_1 values, then the probability that the output be a (δ_1, δ_2) pair with $(\delta_1, \delta_2) \in D_1 \times D_1$ is $2^{-151.23}$, taking into account all intermediate values with an alternation of elements of Δ_1 and Δ_2. The $2^{-151,23}$ value represents the best probability associated with 6 rounds of Crypton v1.0. It is significantly lower than the $2^{-112.62}$ best probability associated with 6 rounds of Crypton.

For stochastic cryptanalysis using a partition of blocks in 16 classes, the number of plaintexts required for a 1-R attack is at least 2^{135}. This figure was derived from Φ_1, and corresponds to the most favorable values of the 6-uple of 4-bit key words involved in the computations. It is significantly higher than the $N = 2^{104}$ minimal number of required plaintexts obtained for the initial Crypton, and higher than the number of distinct plaintexts (2^{128}).

Thus in summary Crypton v1.0 resists the stochastic attacks of Sections 4 and 5 much better than the initial version of Crypton. This seems to be a direct consequence of the design criteria of the new Crypton S-boxes. Because of the decrease of the number of low weight highest probability differential and linear characteristics at each round, the number of "high probability paths" that together form the transition probabilities considered in our stochastic attacks is decreased and the performance of the attacks is significantly affected. Changes in S-boxes proposed in Crypton v1.0 were very discerning.

Published in *Fast Software Encryption - FSE 2000* - LNCS 1978

7 Conclusion

We have described two stochastic attacks on the eight-round version of the Crypton block cipher which are faster than an exhaustive search and more efficient than the best attacks discovered so far.

The first of these two attacks is close to a truncated differential attack, but we believe that probability matrices computations are more precise than truncated differentials probabilities computations would be. It seems to us that an analysis based on truncated-differential probabilities would have led to an overestimate of the performance of the attack.

Our attacks do not threaten the security of Crypton in a full version, but nevertheless put the highlight on a (slight) diffusion weakness in the linear part of the Crypton round function. Even if finding the most relevant cryptographic criteria on the linear part of substitution-permutation blockciphers is still to a large extent an open issue, diffusion criteria related to the number of "active" S-boxes (e.g. MDS properties) offer some clues. Our attacks exploit the existence of low weigh (and iterative) relations between the input and the output of the linear part of Crypton, which lead to statistics involving only 4 active S-boxes per round.

Finally, the comparison between the results obtained on the initial Crypton and Crypton v1.0 confirms that the S-box modifications introduced in Crypton v1.0 significantly improve its resistance against some classes of attacks.

References

Ba99. O. Baudron, H. Gilbert, L. Granboulan, H. Handschuh, A. Joux, P. Nguyen, F. Noilhan, D. Pointcheval, T. Pornin, G. Poupard, J. Stern, S. Vaudenay, "Report on the AES Candidates", *The Second Advanced Encryption Standard Candidate Conference,* N.I.S.T., 1999.

Hal99. C. D'Halluin, G. Bijnens, V. Rijmen, B. Preneel, "Attack on Six Rounds of Crypton". In *Fast Software Encryption - FSE'99* , p. 46, Springer Verlag, Rome, Italy, March 1999.

HG97. H. Handschuh, H. Gilbert, "χ^2 Cryptanalysis of SEAL Encryption Algorithm". In *Fast Software Encryption - FSE'97*, pp. 1-12, Springer Verlag, Haifa, Israel, 1997.

Li98. C.H. Lim, "Crypton : A New 128-bit Block Cipher", *The First Advanced Encryption Standard Candidate Conference,* N.I.S.T., 1998.

Li99. C.H. Lim, "A Revisited Version of Crypton : Crypton V1.0". In *Fast Software Encryption - FSE'99* , p. 31, Springer Verlag, Rome, Italy, March 1999.

Ma93. M. Matsui, "Linear Cryptanalysis Method for DES Cipher". In *Advances in Cryptology - Eurocrypt'93*, pp. 386-396, Springer Verlag, Lofthus, Norway, 1993.

LM91. X. Lai, J.L.Massey and S. Murphy, "Markov Ciphers and Differential Cryptanalysis". In *Advances in Cryptology - Eurocrypt'91*, p. 17, Springer Verlag, Brighton, UK, 1991.

HM97. C. Harpes and J.L.Massey, "Partitioning Cryptanalysis". In *Fast Software Encryption - FSE'97* , p. 13, Springer Verlag, Haifa, Israel, January 1997.

Mu. S. Murphy, F. Piper, M. Walker, P. Wild, "Linear Cryptanalysis for Block Cipher Keys". Unpublished.

Va95. S. Vaudenay, "La sécurité des Primitives Cryptographiques". Doctoral Dissertation, 1995.

Published in: *Fast Software Encryption, FSE 2000*, LNCS 1978

A Matrix of Transition Biases for the Φ Value

	0	1	2	3	4	5	6	7
0	204800	-36864	124928	-28672	-53248	77824	-18432	69632
1	-28672	-83968	-4096	-94208	-94208	63488	-118784	36864
2	-55296	53248	-28672	45056	112640	-94208	53248	-86016
3	-69632	102400	-94208	75776	28672	-45056	53248	-51200
4	-114688	53248	-112640	45056	-36864	-94208	6144	-86016
5	-20480	67584	-12288	110592	143360	-47104	135168	-53248
6	34816	-36864	77824	-28672	-92160	77824	-102400	69632
7	86016	-86016	77824	-92160	-45056	28672	-36864	67584
8	28672	-135168	14336	-126976	-57344	12288	-71680	4096
9	102400	-2048	94208	-12288	-61440	145408	-53248	118784
10	-55296	36864	-61440	28672	112640	-77824	86016	-69632
11	-86016	102400	-77824	75776	45056	-45056	36864	-51200
12	-77824	118784	-59392	110592	106496	4096	116736	12288
13	-53248	-14336	-77824	28672	12288	-129024	36864	-135168
14	34816	-53248	45056	-45056	-92160	94208	-69632	86016
15	69632	-86016	94208	-92160	-28672	28672	-53248	67584

	8	9	10	11	12	13	14	15
0	4096	-77824	-30720	-86016	-114688	36864	-116736	45056
1	143360	-47104	151552	-36864	-20480	100352	-28672	61440
2	-75776	94208	-69632	102400	51200	-53248	12288	-61440
3	-45056	61440	-53248	22528	86016	-86016	94208	-79872
4	4096	94208	-14336	102400	106496	-53248	161792	-61440
5	-126976	30720	-102400	53248	4096	-83968	-20480	-77824
6	63488	-77824	118784	-86016	-38912	36864	-61440	45056
7	61440	-45056	36864	-38912	-102400	69632	-77824	96256
8	90112	-12288	88064	-20480	-36864	135168	-55296	143360
9	45056	-129024	20480	-118784	-86016	18432	-61440	-20480
10	-75776	77824	-102400	86016	51200	-36864	45056	-45056
11	-61440	61440	-36864	22528	102400	-86016	77824	-79872
12	-73728	-4096	-75776	4096	20480	-118784	43008	-126976
13	-61440	112640	-69632	135168	102400	-2048	110592	4096
14	63488	-94208	86016	-102400	-38912	53248	-28672	61440
15	45056	-45056	53248	-38912	-86016	69632	-94208	96256

Table 1. Matrix of distribution for S_1 and Φ_0