Nonlinear Dyn (2015) 79:2531–2539
DOI 10.1007/s11071-014-1830-2

ORIGINAL PAPER

Security analysis and improvement of a block cipher with dynamic S-boxes based on tent map

Dragan Lambić

Received: 23 July 2014 / Accepted: 26 November 2014 / Published online: 4 December 2014

Abstract In this paper, a thorough analysis of a block cipher with dynamic S-boxes based on tent map is made, which reveals the existence of some serious design and security problems. The large number of weak keys is discovered, and the problem with diffusion property is indicated. Two chosen plaintext attacks that exploit the weaknesses in the design of block cipher are presented. Improved version of the analyzed cryptosystem is proposed, which can eliminate the perceived shortcomings.

Keywords Chaos · Block cipher design · Cryptanalysis · Cryptography

1 Introduction

A strong block cipher should be resistant to various attacks, such as linear and differential cryptanalysis. Cryptographic system must possess basic characteristics of cryptography such as confusion and diffusion [1], which overlaps with the properties of chaos such as mixing, random-like behavior, ergodic behavior and sensitivity to initial conditions. In recent years, the relationship between the science of chaos and cryptography has led to the development of a number of chaotic cryptosystems. Chaos in these systems is generally used for the generation of S-boxes, but there are also systems in which it has been used in the shuffling stage [2].

D. Lambić (✉)
Faculty of Education, University of Novi Sad,
Podgorička 4, Sombor, Serbia
e-mail: draganposao@yahoo.com

In [3], a block cipher with dynamic S-boxes based on tent map is proposed which, unfortunately, has certain drawbacks which can have negative effects on the safety and functioning of the cryptosystem. It is considered desirable for a cipher to have no weak keys [4]. Weak keys usually represent a very small part of the keyspace, so there should be a small likelihood of the random generated weak key. Inadequate selection of chaotic map used in [3] has led to a situation where a significant part of the key space consists of weak keys which may, depending on the implementation of the system, lead to a weak security or cessation of functioning of cryptographic system. Also, analyzed cryptosystem does not meet the requirement of good diffusion property because it has low sensitivity to the change of plaintext.

To break a cipher, attacker must find a weakness in the cryptosystem which can be exploited with a complexity lower than a brute-force attack [5]. Weaknesses in the design of analyzed cryptosystem allow two attacks, whose effectiveness depends on the parameters used in implementation.

The rest of the paper is organized as follows. In Sect. 2, the cryptosystem under examination is described. Design problems causing the large number of weak keys and problem with diffusion property are discussed in Sect. 3. In Sect. 4, examples of attacks against the cryptosystem are presented. In Sect. 5, improved version of the analyzed cryptosystem which can eliminate the perceived shortcomings is proposed. Finally, in Sect. 6, conclusions are drawn.

2 Description of the encryption scheme

In paper [3], a block encryption scheme based on dynamic S-boxes is proposed. This cipher is supposed to encrypt plaintext blocks of arbitrary length of l bytes. In this paper, we analyze a concrete example of the system, which was also described in [3], that encrypts blocks of $l = 32$ bytes by using 32 8×8 S-boxes. The 32 S-boxes are used to encrypt only q plaintext blocks, in [3] authors proposed $q = 7$. S-boxes are obtained based on the key, consisting of two double float numbers b_0, x_0 and arbitrary permutation K of integer sequence $\{0, 1, \dots, 255\}$. The key space is estimated at $2^{48} \cdot 2^{48} \cdot 256! \simeq 2^{1780}$.

2.1 Preliminary definitions

In [3], the chaotic tent map is defined by the following equation:

$$x_{i+1} = \begin{cases} x_i/b, & 0 < x_i \leq b \\ (1-x_i)/(1-b), & b < x_i < 1 \end{cases}$$

where $b \in (0, 1)$ and $b \neq 0.5$ is a constant, x_i ($i = 0, 1, \dots$) is the state value of the chaotic map.

2.2 The method of generating $n \times n$ S-boxes

In [3], the authors first defined function OPS(i, j, K) where i is the number of iterations of the tent map, j is the number of subintervals and K is one of the arbitrary permutations of the integer sequence $\{0, 1, 2, \dots, 2^n - 1\}$. The function OPS$(i, j, K)$ includes the following operations:

1. Divide the interval $[0, 1]$ into j subintervals: $[0, \frac{1}{j}), [\frac{1}{j}, \frac{2}{j}), \dots, [\frac{j-1}{j}, 1)$.
2. Iterate the tent map for i times.
3. Suppose the current state value of the tent map is in the mth subinterval $[\frac{m-1}{j}, \frac{m}{j})$. Denote the values of elements of K with $K[i] = k_i$ for $0 < i \leq j$. Exchange the mth integer k_m with the jth integer k_j in the sequence K, i.e., $K[m] = k_j$ and $K[j] = k_m$.
4. Change the control parameter b of the tent map according to equation:

$$b_{i+1} = 0.9b_i + 0.1 \cdot k_m/(2^n - 1)$$

In [3], two methods for generation of S-boxes are presented. That is why we will assume that the authors have proposed two versions of their cipher. In version 1., authors set $i = 32$. The 8×8 S-boxes are generated by executing the function OPS(32, j,K) from $j = 2^8$ to 2.

In version 2., 32 identical integer sequences $K_1, K_2, \dots, K_{32}$ are defined, which are one of the arbitrary permutations of the integer sequence $\{0, 1, 2, \dots, 255\}$. The 8×8 S-boxes are generated by executing the functions OPS$(1, j, K_1)$, OPS$(1, j, K_2), \dots,$ OPS$(1, j, K_{32})$ from $j = 2^8$ to 2. The following pseudocode fragment explains how to implement the above operations.

```
for ($j = 2^n$; $j > 1$; $j--$)
{
for ($i = 1$; $i < 32$; $i++$) OPS$(1, j, K_i)$
}
```

2.3 Encryption algorithm

Details of studied algorithm are described below.

1. Divide the plaintext message P into blocks B_j of length l bytes ($l = 32$ in [3]). Pad zeros to P if its length is not a multiple of l bytes.
2. Iterate tent map from Sect. 2.1 for N_0 times ($N_0 = 30$ in [3]), where N_0 is a constant. Set $K_1 = K_2 = \dots = K_{32} = K$
3. Generate 32 8×8 S-boxes according to the one of the methods described in Sect. 2.2.
4. For the convenience of description, define K_s, p_s as the sth S-box and the sth byte in block B_j, respectively. Substitute p_s according to K_s. Then, permute the substituted block B_j by left cyclic shift 4 bits. Finally, we get the cipher block C_j by doing the substitution and shift operations for $l-1$ rounds.
5. Based on the ciphertext, number D is calculated. After encryption of q blocks, tent map is iterated D times, and the process continues from the third step.

3 Design problems

3.1 Weak keys

One of the main problem in chaos-based cryptography is problem with the selection of the chaotic system [6]. In some chaos-based cryptosystems, the control parameters of the underlying chaotic systems are determined by the secret key. If the link between the secret key and the control parameters is not established carefully, then it is possible that the underlying chaotic system evolves in an non-chaotic way, which further erodes the confusion and diffusion properties required by the resulting cryptosystem.

The cryptosystem described in [3] uses chaotic tent map presented in Sect. 2.1. Problem with this chaotic map is that the key space for parameters x_i, b_i contains a large number of keys leading to non-chaotic behavior. In the case $x_i = b_i$, we obtain $x_{i+1} = x_i/b_i = 1$ thus the state of chaotic map goes out from a set of permitted values. If the implementation of cryptographic system is precisely defined to allow only permitted values, this pair of values of x_i, b_i will cause interruption of the functioning of the system. Otherwise, for $x_{i+1} = 1$ all subsequent states of chaotic map will be equal to 0, which causes that all subsequent S-boxes will be equal to each other. The author of this paper failed to find the precise implementation of the cipher described in [3], so the both cases can be considered possible.

Key space for both parameters x_i, b_i is estimated at 2^{48}, so we can say that there are 2^{48} couples x_i, b_i such that $x_i = b_i$ of the total number of couples of parameters of chaotic map $2^{48} \cdot 2^{48} = 2^{96}$. Also, a much larger number of keys can lead to this situation after a certain number of iterations of the chaotic tent map.

For encryption of $q \cdot 256$ bits of plaintext, $256 \cdot 32 = 2^{13}$ iterations of chaotic maps are required. For a plaintext of L bits, we need $T = \frac{32 \cdot L}{q}$ iterations of tent map for encryption. This means that for the key, we must choose x_0, b_0 so that $x_t \neq b_t$ for all $0 \leq t \leq T$. The probability to choose such pair x_0, b_0 is $(1 - \frac{1}{2^{48}})^T$. If, for example, $T = 2^{50}$ (for $q = 7$, corresponding plaintext has 2^{40} blocks) probability of choosing a good key is less than 2 %. Bearing in mind that this cipher can be used for image encryption [3], there is a real possibility that the cipher encounters such a large amount of data. If we take into account the permutation K, which is part of the key, number of weak keys is at least 2^{1732} and their number, depending on the amount of data that needs to be encrypted, (i.e., the number of iterations T of the chaotic map) can go very close to 2^{1780}. Increasing the number of blocks of plaintext q that are encrypted with the same 32 S-boxes can reduce the probability that x_i comes out of the permitted interval.

Table 1 Influence of plaintext change, when changed bytes in states have different first and second half from original bytes

p_{27} **S**	p_{28} **S**	p_{29} **S**	p_{30} **S**	p_{31} C
$c_{27}{}^{(1)}$ **S**	$c_{28}{}^{(1)}$ **S**	$c_{29}{}^{(1)}$ **S**	$c_{30}{}^{(1)}$ **S**	$c_{31}{}^{(1)}$ C
$c_{27}{}^{(2)}$ **S**	$c_{28}{}^{(2)}$ **S**	$c_{29}{}^{(2)}$ **S**	$c_{30}{}^{(2)}$ C	$c_{31}{}^{(2)}$ C
$c_{27}{}^{(3)}$ **S**	$c_{28}{}^{(3)}$ **S**	$c_{29}{}^{(3)}$ C	$c_{30}{}^{(3)}$ C	$c_{31}{}^{(3)}$ C
$c_{27}{}^{(4)}$ **S**	$c_{28}{}^{(4)}$ C	$c_{29}{}^{(4)}$ C	$c_{30}{}^{(4)}$ C	$c_{31}{}^{(4)}$ C
$c_{27}{}^{(5)}$ C	$c_{28}{}^{(5)}$ C	$c_{29}{}^{(5)}$ C	$c_{30}{}^{(5)}$ C	$c_{31}{}^{(5)}$ C

Table 2 Influence of plaintext change, when changed $c_{31}{}^{(1)}$ byte have same left half as original byte

p_{27} **S**	p_{28} **S**	p_{29} **S**	p_{30} **S**	p_{31} C
$c_{27}{}^{(1)}$ **S**	$c_{28}{}^{(1)}$ **S**	$c_{29}{}^{(1)}$ **S**	$c_{30}{}^{(1)}$ **S**	$c_{31}{}^{(1)}$ C
$c_{27}{}^{(2)}$ **S**	$c_{28}{}^{(2)}$ **S**	$c_{29}{}^{(2)}$ **S**	$c_{30}{}^{(2)}$ **S**	$c_{31}{}^{(2)}$ C
$c_{27}{}^{(3)}$ **S**	$c_{28}{}^{(3)}$ **S**	$c_{29}{}^{(3)}$ **S**	$c_{30}{}^{(3)}$ C	$c_{31}{}^{(3)}$ C
$c_{27}{}^{(4)}$ **S**	$c_{28}{}^{(4)}$ **S**	$c_{29}{}^{(4)}$ C	$c_{30}{}^{(4)}$ C	$c_{31}{}^{(4)}$ C
$c_{27}{}^{(5)}$ **S**	$c_{28}{}^{(5)}$ C	$c_{29}{}^{(5)}$ C	$c_{30}{}^{(5)}$ C	$c_{31}{}^{(5)}$ C

3.2 Low sensitivity to the change of plaintext

In some encryption, architectures plaintexts with slightest differences are associated with very similar ciphertexts, which is a clear violation of the diffusion property [6]. In Shannon's original definition, diffusion refers to dissipating the statistical structure of plaintext over bulk of ciphertext [1]. In particular, for a randomly chosen input, if one input bit is changed, then the probability that some output bit will change should be one half, and this is termed the strict avalanche criterion [7].

In [3], property of diffusion is achieved by performing 32 rounds of substitution and left cyclic shift by 4 bits on a plaintext $P = p_0 p_1, \ldots, p_{31}$. In this way, by changing single byte of plaintext, for example p_{31}, we can affect all 32 bytes of ciphertext $C = c_0 c_1, \ldots, c_{31}$ only under the condition that in each round by substituting the changed input bytes, we obtain output bytes that have different first and second half (4 bits) from original (unchanged) output.

Denote by $c_0{}^{(a)} c_1{}^{(a)}, \ldots, c_{31}{}^{(a)}$ state of ciphertext after a rounds. Then, change of the last byte p_{31} in plaintext block can influence just one last byte of the state $c_{31}{}^{(1)}$ in the first round, in the second round last two bytes $c_{30}{}^{(2)} c_{31}{}^{(2)}$, in ath round last a bytes of the state $c_{32-a}{}^{(a)}, \ldots, c_{30}{}^{(a)} c_{31}{}^{(a)}$ and after the final round all bytes of ciphertext C (Table 1). In Tables 1 and 2,

S denotes same bytes as original state and C bytes that are different from original state.

Suppose that on the basis of the change of the last byte of plaintext in the round a as a result of substitution, we get byte $c_{32-a}{}^{(a)}$ which has the same left 4 bits (for example $c_{32-a}{}^{(a)} = 00000000$ instead $c_{32-a}{}^{(a)} = 00001111$) as in the original state. Then, due to shift to the left by only 4 bits, the number of bytes in the next round, on which the change had the impact, is reduced by one from the left side (Table 2).

If this situation is repeated d times, the change has no effect on the first d bytes of ciphertext. A similar situation happens if as a result of substitution, we get byte $c_{32-a}{}^{(a)}$ which has the same right 4 bits. Then, the change has no impact on the bytes at the end of ciphertext. As a consequence of this occurrence, altered ciphertext has a number of connected unchanged bytes (including the ones at the beginning and at the end).

The analyzed cryptosystem uses 8×8 S-boxes which are generated by pseudo-random method. Each S-box has 256 input and output values, so there is likelihood of $\frac{30}{255}$ that for the modified input byte, we get output byte with the same first or second half of bits. Cryptosystem has 32 rounds so the likelihood of obtaining the output bytes with a completely different first and second halves in all rounds is $(\frac{225}{255})^{32} \simeq 1,8\,\%$. On the basis of low sensitivity to the change of plaintext attacker can find certain regularities related to the functioning of cryptographic system.

4 Cryptanalysis of the encryption scheme

4.1 Weak key attack

Section 3.1 described that there is a high probability that after 2^{50} iterations, values of chaotic tent map could get out of the allowed set. If this situation occurs, all subsequent values of chaotic tent map will be equal to 0, which causes all subsequent S-boxes to be equal to permutation K cyclically shifted by one position to the left.

The proposed attack is based on the assumption that the attacker can choose certain number of the plaintext blocks that will be encrypted by using secret key. First, the attacker must determine when the chaotic tent map will get out of the allowed set of values. This is easiest to determine, in the case when all the chosen blocks P are equal, by checking whether the last $q + 1$ ciphertext blocks are also equal with each other. The attacker should, in a worse case, chose $q \cdot 2^{37}$ equal blocks P. However, it is logical to assume that at the time of the attack cipher is already functioning for some time. It means that the chaotic tent map was iterated a certain number of times, which reduces the required number of chosen blocks for the attack.

Once the attacker finds that the cipher uses the same S-boxes for encryption of the plaintext, it is necessary to choose another 256 different plaintext blocks which will be encrypted. Each of the selected blocks has to have all 32 bytes equal, so that block P_0 has all bytes equal to 0, P_1 all bytes equal to 1 and so on. Since all S-boxes used for encryption are equal with each other, based on the plaintext $P_0, P_1, \ldots, P_{255}$, ciphertexts $C_0, C_1, \ldots, C_{255}$ which also have all equal bytes are obtained. As each of the 32 rounds of the cipher uses substitution with only one S-Box K_1 (for example) and then the cyclic shift by 4 bits, it can be considered that the result of each round is substitution with S-Box K_1' which is equal to the S-box K_1 with permuted halves of the output values. In this subsection, each byte of the block P_i will be denoted with p_i, because all the bytes of the observed blocks are equal. By permuting halves of the c_i, we get bytes c_i' so we can say that $c_i' = K_1'^{32}(p_i)$ for all $0 \leq i < 256$ where $K_1'^{32}$ represent a composition of the S-box K_1' 32 times with itself.

The attacker knows all 256 values of S-box $K_1'^{32}$ and can obtain the values of the S-box K_1'. The attacker compares the bytes of plaintext and ciphertext and seeks for $p_i = c_j$. In this way, the attacker finds all cycles $p_i = K_1'^{32 \cdot r}(p_i)$ and can group couples p_i, c_i by cycles of lengths $r_1, r_2, \ldots, r_o$ that satisfy $r_1 + r_2 + \cdots + r_o = 256$. Now, the attacker has all the elements of the S-box $K_1'^{32}$ grouped by cycles of lengths r_i and all elements of the S-box K_1' grouped by cycles of lengths $r_i \cdot 32$. By dividing numbers 32,64,…,$r_i \cdot 32$ per module r_i, the attacker can easily obtain all elements of S-box K_1'. By permuting halves of the output values of S-box K_1', S-box K_1 is obtained. By right cyclic shift of the elements of the S-box K_1 by one position, permutation K (part of the key) is recovered.

This attack gives complete break of the analyzed cipher. Attack requires from 256 blocks of chosen plaintext up to over $q \cdot 2^{37}$ blocks in the worse cases. The complexity of the attack is equal to the number of required chosen plaintext blocks and should not exceed 2^{96}. If after 2^{96} blocks, chaotic tent map does not get out of the set of allowed values, then the key x_0, b_0 is element of the cycle, and described attack is not applicable.

4.2 Attack based on poor diffusion property

The previous subsection describes the attack when part of the key x_0, b_0 belongs to one of the trajectories that lead to one of the 2^{48} weak keys. If x_0, b_0 does not belong to any of the trajectories, then they belong to a cycle, and values of chaotic map in this case would never get out of the set of allowed values. The next attack is applicable when (x_0, b_0) are not weak keys.

Due to poor diffusion property, attacker can gain some information about relationships that exist between the elements of S-boxes used in a cryptosystem and ciphertext. This cryptanalytic attack aims to find the elements of S-boxes, based on which permutation K can be reconstructed.

4.2.1 Reconstruction of permutation K

If all the elements of the S-box $K_i (0 < i \leq 32)$ are known, then permutation K (part of the key) can be reconstructed. The attacker not need to know the exact values of (x_0, b_0) used as a part of the key, as key space for these two parameters is only 2^{96}, which is insignificant compared to the entire key space.

If we consider the first version of the cipher from Sect. 2.2, then the key K can be reconstructed on the basis of elements of S-box K_1 as follows. Suppose that the attacker known parameters (x_0, b_0). In [3], elements of S-boxes are obtained by method described in Sect. 2.2 placing mth element of permutation K, which corresponds to the current state of tent map, on last unassigned position in S-box. The last element of the permutation K is moved to position m. Based on this, attacker can get the mth element of K on the basis of last element of the K_1 and parameters (x_0, b_0). Attacker at this point still do not know the last element of the permutation K, because generation of S-box K_1 could lead to the multiple shifts of the same element of permutation K. The parameters (x, b) are updated by the procedure from Sect. 2.2, and the process of reconstruction of key K is repeated with the penultimate element of S-box. This procedure is repeated until first element of the S-box is reached, taking into account the elements that are more than once shifted during the process of generation.

And when it comes to version 2 of the cryptographic system, the process of reconstruction of the key is very similar to the previously described. Instead of all the elements of one S-box,it is necessary that the attacker knows the last 8 elements from each of the 32 S-boxes. This number of known elements must be increased by the number of repeated elements.

4.2.2 Identification of the elements of S-boxes

The proposed attack is based on the assumption that the attacker can choose the plaintext that will be encrypted by using secret key. Based on the chosen plaintext and the corresponding ciphertext, attacker can gain some information about relationships that exist between the elements of S-boxes and ciphertext. The aim of the attack was to identify any S-box K_i if it is version 1 of the cipher, or identification of a certain number of the last elements of each S-box if it is a version 2, that would be faster than the search of all possible permutations of K.

Although the analysis showed greater number of patterns and connections between the elements of S-box and ciphertext, in this paper, we focus on the one most obvious regularity.

If all the bytes of the plaintext block except the last byte p_{31} are constant, then for 256 different values of p_{31}, we get 256 different ciphertexts. Obtained ciphertexts are compared ($\frac{256 \cdot 255}{2}$ pairs) and number of connected quadruples of bits that pairs of ciphertexts have equal at the beginning $s_{i,j}$ and at the end $e_{i,j}$ is monitored, where i, j represent different values of p_{31}. The analysis of the obtained data, based on multiple encryption implemented with various open texts and different keys, has shown the following regularity. If $m = min\,(s_{0,1}, s_{0,2}, \ldots, s_{254,255}, e_{0,1}, e_{0,2}, \ldots, e_{254,255})$, then pairs of the ciphertexts which have m equal connected quadruples of bits at the beginning (end), obtained based on last bytes $p'_{31} = i$ and $p''_{31} = j$, point to the fact that $K_{31}(i)$ and $K_{31}(j)$ must have different first (second) 4 bits.

This regularity is obtained by observing the influence of the number m on the relationship of output values of the observed S-box (K_{31}). Denote by N the number of pairs (i, j) having m equal connected quadruples of bits at the beginning or at the end of ciphertext. Number N can be represented as $N = N_1 + N_2 + N_r$, where N_1 is number of pairs of ciphertext for which it has been observed that the output values of K_{31} for input values

i and j have first 4 bits equal, N_2 number of pairs of ciphertext for which second four output bits are equal, and N_r pairs for which both halves of output byte are different. For pairs of input values (i, j) having m equal connected quadruples of bits at the beginning (end) of ciphertext was observed that the value N_1 (N_2) is in many cases equal to 0, which means that for these values of (i, j), we can not obtain output values of K_{31} with the equal first (second) half of the byte. Previous regularity is based on poor diffusion property of observed cryptographic system, because the diffusion of all bytes of ciphertext can be achieved only if in each round, based on the changes of the plaintext, we always have the output values with different first and second half of the byte.

Based on the previous, when attacker is trying to identify all the elements of the S-box K_{31}, on basis of $K_{31}(P0')$, it is not required to browse all of the remaining 255 possibilities for $K_{31}(P0'')$, but the attacker can focus search on 240 values with first (second) 4 bits different from $K_{31}(P0')$.

If the attacker does not have the opportunity to choose a large number of plaintext blocks, the success of this attack depends largely on the parameter q. As after every q blocks of plaintext, S-boxes which are used for encryption are changed, we can conclude that we have only q pairs of (P, C) available for the attack with choosen P. The attacker possesses only $q*(q-1)$ pairs of ciphertexts. Discrete chaotic tent map, used in analyzed cipher, has a finite number of values (2^{96}) and every cycle of this map also has a finite number of elements. Finite number of elements of the cycle causes that after the chaotic map goes through all elements of the cycle, the same elements again occur, i.e., after a certain number of iterations of the tent map the same S-boxes, which are used for encryption, are used again.

The efficiency of this attack should largely depend on parameter q ;however, due to repetition of S-boxes used for encryption, limitations caused by changing the S-boxes for every q blocks can be overcome if the attacker has the opportunity to choose a large number of blocks of plaintext P.

Attacker, with a maximum of 2^{96} chosen plaintext blocks (equal with each other), can determine the length of the cycle, after which the same S-boxes occur. Cycle length o can be much smaller (even $o = 1$ if it is a fixed point), so the same S-boxes can occur very quickly. Attacker only needs to wait to get the ciphertext C, which coincides with the ciphertext of the first chosen block. Once the attacker knows the length of the cycle to which value of chaotic tent map belong, he can have $u \cdot q$ blocks encrypted with the same S-boxes instead of q, but only on condition that he can chose another $u \cdot o$ plaintext blocks.

Table 3 Number of connected quadruples of bits that pairs of ciphertexts have equal at the beginning (end)

p_{31}	249	250	251	252	253	254	255
249	–	5(7)	1(5)	7(7)	5(5)	5(5)	5(7)
250	5(7)	–	1(5)	5(9)	5(5)	11(5)	5(9)
251	1(5)	1(5)	–	1(5)	1(5)	1(5)	1(5)
252	7(7)	5(9)	1(5)	–	5(5)	5(5)	5(13)
253	5(5)	5(5)	1(5)	5(5)	–	5(5)	6(5)
254	5(5)	11(5)	1(5)	5(5)	5(5)	–	5(5)
255	5(7)	5(9)	1(5)	5(13)	6(5)	5(5)	–

For the proposed attack that focuses on only one byte of plaintext, number of pairs P, C where P is chosen is not required to be greater than 256. This amount of information the attacker can have if $q = 256$ or if q is smaller, but then attacker must choose more than $\frac{256}{q} \cdot o$ plaintext blocks.

4.2.3 Example of attack for $q = 7$

Observe the case when $q = 7$ as described in [3]. The secret keys are $x_0 = 0,123456789$, $b_0 = 0,987654321$ and $K = \{0, 1, \ldots, 255\}$. The attacker then has $q = 7$ available pairs P, C such that he can choose P. The attacker assigns all bytes a value 0, except the last byte p_{31} to which assigns the largest seven values (from 255 to 249). On the basis of these seven values of p_{31}, the attacker gets seven different ciphertexts, i.e., $\frac{7 \cdot 6}{2} = 21$ ciphertext pairs for analysis.

The data in Table 3 show that the ciphertext obtained on the basis of $p_{31} = 251$ has only the first 4 bits same with other ciphertext which is noticeably less than the other pairs. Then, it is $m = 1 = min(s)$, $N = 6$. Based on the above data, the attacker can conclude that $K_{31}(251)$ has the first 4 bits different from the $K_{31}(249)$, $K_{31}(250)$, $K_{31}(252)$, $K_{31}(253)$, $K_{31}(254)$ and $K_{31}(255)$.

The attacker first checks all 256 possibilities for the value of the $K_{31}(251)$. For the remaining 6 output values of S-box, attacker knows that they do not have the first 4 bits the same as $K_{31}(251)$, so instead of $\frac{255!}{249!}$, there are $\frac{240!}{234!}$ possibilities for these elements. For the remaining 249 elements, attacker has no data, so the search is the same as the brute-force attack with complexity of the 249!. This attack is about $2^{0.53}$ faster than a brute-force attack. If the attacker has the option to choose another $37 \cdot o(\frac{256}{7} \simeq 36, 57)$ plaintext blocks, in order to establish the length of the cycle of the values x_0, b_0, then the attacker can obtain the remaining 249 pairs P, C that are needed for this attack. In this case, this attack is faster than a brute-force attack for around 2^{79} times. Checks required to detect the length of the cycle are independent from the rest of the attack, and even the maximal complexity of the checking of about $\frac{256}{q} \cdot 2^{96}$ is negligible compared to the complexity of the attack of about 2^{1701}.

4.2.4 Example of attack for $q = 256$

Observe the case when $q = 256$. The values of the x_0, b_0 are the same as in the previous example, while in permutation K, the first two elements are transposed (instead of 0, 1 it is 1, 0). The attacker now has 256 pairs of P, C available. The procedure from the previous example is repeated, but now all 256 values are assigned to the last byte, and the corresponding ciphertext is obtained. Based on the analysis of all pairs of ciphertext, attacker obtains m=3=min(e) (at the end of the ciphertext), $N = 16{,}384$. Obtained data show that all output values of the S-box K_{31} are divided into two groups containing eight 16-tuples, so that elements from one group have different second 4 bits from elements from other group.

The attacker knows which 128 input values of K_{31} are in the first group and which 128 input values are in the second group, and it is known that there are no output values from different groups with last four bits equal. The attacker, having such large quantity of information, can perform following attack in order to reconstruct elements of S-box K_{31}. The attacker first chooses eight of sixteen possible values for last 4 output bits of elements from first group. After that, 128 appropriate output values are arranged on 128 input values from that group. At the end, remaining 128 values are arranged. This attack has complexity of $\frac{16!}{8!} \cdot 128! \cdot 128! = 2^{1432}$ which is compared to 2^{1684} cheks needed to recover all elements of K_{31} with the brute-force attack, faster by about 2^{252} times.

5 Improvement of the encryption scheme

In this section, improvements of the analyzed encryption scheme, which should remove the noticed shortcomings, are presented. Another important issue is the length of the key which is in [3] $48 + 48 + 256 \cdot 8 = 2{,}144$ bits, sufficient to store the key (K, x_0, b_0). The key length is large compared to some other block cipher, eg. AES (usually between 128 and 512 bits) or chaotic cipher presented in [8] (216 bits), and also large compared to the estimated safety of 1,780 bits which the authors claim that this cryptographic system provides. A difference of $2{,}144 - 1{,}780 = 364$ bits between key length and security of the system is significantly higher compared to for example AES, which due to biclique attack [9], provides the security of approximately 2^{254} with key of the 256 bits. Due to the above, corrected cryptosystem will use the keys of variable length, and its security should correspond to the key length.

The key length L, of the improved cipher, should be greater or equal to 256 bits and to be multiple of 4. The cause of the large number of weak keys is selection of chaotic map. For the aforementioned reasons, it is necessary to completely change the method for the generation of S-boxes. The proposed method for the generation of S-boxes will be described for $n \times n$ S-boxes where $n = 8$, but it may be applicable for any other value of $n > 1$.

For the generation of 32 S-boxes, simple algorithm based on chaotic map and composition method [10] is used. The set of 16 fixed bijective starting $n \times n$ S-boxes $f_0, f_1, \ldots, f_{15}$ is used. These starting S-boxes are not part of a secret key, can be public and can be generated by any known method (by the methods mentioned in [11–13] for example). Any chaotic map $x_{t+1} = f(x_t)$ can be used to generate a chaotic sequence of $m > 1$ indexes $i_1, i_2, \ldots, i_m \in \{0, 1, \ldots, 15\}$, where $i_t = \text{floor}(x_t \cdot 16)$ for $1 \leq t \leq m$. For example, one of the chaotic maps mentioned in [11] can be used. The composition $h = fg$ of two permutations f and g of the same set A is the permutation mapping each $y \in A$ into $h(y) = f(g(y))$. This S-box generation method returns the $n \times n$ S-box $S_t = \prod_{t=1}^{m} f_{i_t}$.

In order to make the process of generation of S-boxes dependent on the key, first $\frac{L}{4}$ indexes (i_t) are obtained based on a quadriples of bits from the key, respectively. Thereafter, based on the chaotic map, for every q blocks, 32 indexes of starting s-boxes are obtained. For the first q blocks in the process of encryption, we use S-boxes $S_{\frac{L}{4}+1}, \ldots, S_{\frac{L}{4}+32}$. For the initial state of the chaotic map x_0, we can use the first 48 bits of the key, while the other parameters of chaotic maps can be made public.

Besides the above changes in [3], it is necessary that the number of rounds of the system is increased from 32 to at least 64 (it could be more, but would reduce the speed of encryption) in order to avoid problems with the diffusion. Previous changes should eliminate shortcomings of the analyzed cipher, which are described in this paper.

By changing the method for generation of S-boxes, we eliminated possibility of the occurrence of weak keys. This change enabled use of shorter key, in a way that the key length corresponds to the system's security. By increasing the number of rounds, poor diffusion is remedied which prevents the attack based on poor diffusion property described in this paper.

The method of generation of S-boxes from [10] is approximately 6 times faster than the method from [3]. Also, by increasing the number of rounds, encryption process is twice as slow. This leads to a situation in which encryption process is about 2 times faster than the process of generation of S-boxes, which is significantly smaller than the difference of 24 times in [3].

Increasing the number of rounds affect the speed of the system, however, bearing in mind that the method for generation of S-boxes is faster than previously used, we can say that the whole system does not work significantly slower. If we consider the case when $q = 7$, the time needed for the system [3] to encrypt q blocks is $7 \cdot 1 \cdot t + 24 \cdot t = 31 \cdot t$, while the improved system needed $7 \cdot 2 \cdot t + 4 \cdot t = 18 \cdot t$ time units. For $q < 20$, improved system is faster than original one and is significantly safer.

6 Conclusion

In this paper, security analysis of a block cipher with dynamic S-boxes based on tent map [3] is presented. Design problems are identified which significantly affect the number of weak keys and diffusion property of the analyzed cipher.

Two chosen plaintext attacks on this cryptographic system, based on the perceived shortcomings, are presented. First chosen plaintext attack uses weak keys and leads to a complete break of the analyzed cipher.

The second attack uses the weak diffusion property. This cryptanalytic attack aims to find the elements of S-boxes on the basis of certain number of chosen-plaintexts, based on which permutation K can be reconstructed. This attack can be faster than a brute-force attack up to 2^{252} times. Described attack requires more chosen plaintext blocks when q is smaller, but then the system itself has a greater chance of not functioning properly, because with a smaller q, the number of weak keys increases due to a larger number of iterations of the chaotic map required.

It can be concluded that the observed cryptographic system has weaknesses which threaten its security and its performance. As a solution to the identified problems, corrections of the cipher are proposed, in order to eliminate the weaknesses. By changing the method for generation of S-boxes and increasing the number of rounds, the identified shortcomings are removed, and the security of the system is significantly increased without a significant impact on the speed of encryption.

Acknowledgments I am grateful to Professor M. Živković for useful correspondence.

References

1. Shannon, C.E.: Communication theory of secrecy system. Bell. Syst. Tech. J. **28**, 656–715 (1949)
2. Hermassi, H., Rhouma, R., Belghith, S.: Security analysis of image cryptosystems only or partially based on a chaotic permutation. J. Syst. Softw. **85**, 21332144 (2012)
3. Wang, Y., Wong, K.W., Liao, X., Xiang, T.: A block cipher with dynamic S-boxes based on tent map. Commun. Nonlinear Sci. Numer. Simul. **14**, 3089–3099 (2009)
4. Wang, X., Zhang, W., Guo, W., Zhang, J.: Secure chaotic system with application to chaotic ciphers. Inf. Sci. **221**, 555570 (2013)
5. Schneier, B.: Applied Cryptography. Wiley, New York (1996)
6. Alvarez, G., Amigo, J.M., Arroyo, D., Li, S.: Lessons learnt from the cryptanalysis of chaos-based ciphers. In: Kocarev, Lj, Lian, S. (eds.) Chaos-Based Cryptography: Theory, Algorithms and Applications, pp. 257–295. Springer-Verlag GmbH, Berlin (2011)
7. Webster, A., Tavares, S.: On the design of S -boxes. In: Advances in cryptology: Proceeding of CRYPTO'85. Lecture notes in computer science, pp. 523–534 (1986)
8. Ozkaynak, F., Yavuz, S.: Analysis and improvement of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Nonlinear Dyn. **78**, 1311–1320 (2014)
9. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In: ASIACRYPT 2011, LNCS, vol. 7073, pp. 344–371. Springer, Berlin (2011)

10. Lambić, D.: A novel method of S-box design based on chaotic map and composition method. Chaos Solitons Fractals **58**, 16–21 (2014)
11. Lambić, D., Živković, M.: Comparison of random S-box generation methods. Publications de l'Institut Mathematique **93**, 109–115 (2013)
12. Wang, Y., Wong, K.W., Li, C., Li, Y.: A novel method to design S-box based on chaotic map and genetic algorithm. Phys. Lett. A **14**, 827–833 (2012)
13. Khan, M., Shah, T., Mahmood, H., Gondal, M.A.: An efficient method for the construction of block cipher with multi-chaotic systems. Nonlinear Dyn. **71**, 489–492 (2013)