Cryptanalysis of a new substitution–diffusion based image cipher

Rhouma Rhouma\textsuperscript{a,}*, Ercan Solak\textsuperscript{b}, Safya Belghith\textsuperscript{a}

\textsuperscript{a}Syscom Laboratory, Ecole Nationale d’Ingénieurs de Tunis, 37, Le Belvédère, 1002 Tunis, Tunisia
\textsuperscript{b}Computer Science and Engineering, Isik University, Turkey

\textbf{ABSTRACT}
This paper introduces two different types of attacks on a recently proposed cryptosystem based on chaotic standard and logistic maps. In the two attacks, only a pair of (plaintext/ciphertext) was needed to totally break the cryptosystem.

\begin{itemize}
  \item Article history:
    \begin{itemize}
      \item Received 16 March 2009
      \item Received in revised form 8 July 2009
      \item Accepted 11 July 2009
      \item Available online 16 July 2009
    \end{itemize}
  \item Keywords:
    \begin{itemize}
      \item Chaos based cryptography
      \item Diffusion
      \item Substitution
      \item Chosen plaintext attack
      \item Known plaintext attack
    \end{itemize}
\end{itemize}

1. Introduction

During the last 20 years, there has been an increased interest in chaos based cryptography. There have been many proposals of chaotic cryptosystems based on nonlinear systems like the logistic map, the PWLCM, the standard map, the OCML spatiotemporal maps, etc. Despite this explosion of proposals [1–5], many of these chaotic cryptosystems suffer from serious security problems that make them vulnerable to classical attacks like the chosen plaintext attack or the known plaintext attack or other types of attacks [6–11]. Recently, a new chaos based cryptosystem was proposed by Patidar et al. [1]. This cryptosystem is very fast because it uses simple XORing and mixing operations. The time complexity of these simple operations is very low. Security analysis made by the authors indicate that the cryptosystem satisfies the confusion and diffusion requirements that any good cryptosystem should have. However, by analyzing the algebraic description of the proposed cryptosystem, we have found a drawback in its structure. Indeed, the encryption steps involve linear transformations of the plain image, which allows the attacker to design an equivalent model of the cryptosystem. The equivalent description uses linearity to combine the secret keys that are used in different steps of the original description. This paper proposes two different types of attacks to break the cryptosystem described in Ref. [1]. Section 2 gives a brief description of the cryptosystem under study. Section 3 proposes an equivalent model of the cryptosystem. Section 4 proposes two attacks to break totally the cryptosystem and Section 5 concludes the paper.

2. Brief description of the cryptosystem

Let the RGB components of the plain image \( P \) be denoted as \( P^R, P^G \) and \( P^B \). Each color component is a matrix of size \( H \times W \) with 8-bit color values as entries. Similarly, let \( C^R, C^G \) and \( C^B \) denote the color components of the ciphered image \( C \).
The cryptosystem keys are the real numbers \(x_0, y_0, K\) and the integer \(N\) that satisfy \(x_0, y_0 \in (0, 2\pi), K > 18, N > 100\).

Encryption consists of the composition of four transformations on the plain image. The first transformation is the mixing of the plain image with a key image generated from four XORing keys. Then the resulting color components are independently diffused horizontally. The third step is the vertical diffusion of all the color components together. Lastly, the output of the third step is XORed with a chaotically generated key image.

We now give the mathematical expressions for the transformations applied in each step.

First, the keys \(x_0, y_0, K, N\) are used to generate two key images \(X\) and \(CKS\) that are, in turn, used in the first and the fourth steps of the encryption procedure. In order to calculate \(X\), an intermediate 4-entry vector \(X_{\text{key}}\) is derived as

\[
X_{\text{key}}(1) = \left\lfloor \frac{x_0}{2\pi} \times 256 \right\rfloor, \quad X_{\text{key}}(2) = \left\lfloor \frac{y_0}{2\pi} \times 256 \right\rfloor, \\
X_{\text{key}}(3) = [K \mod 256], \quad X_{\text{key}}(4) = N \mod 256.
\]

(1)

The pixels of the three color components of the key image \(X\) are calculated using the following algorithm:

**Algorithm 1. Generation of the key image \(X\).**

**Input:** \(X_{\text{key}}\)

**Output:** \(X\)

\[
k \leftarrow 1
\]

\[
\text{for } i \leftarrow 1 \text{ to } H \text{ do}
\]

\[
\text{for } j \leftarrow 1 \text{ to } W \text{ do}
\]

\[
\text{foreach } U \in \{R, G, B\} \text{ do}
\]

\[
X_{ij}^U \leftarrow X_{\text{key}}(k)
\]

\[
k \leftarrow (k \mod 4) + 1
\]

end

end

end

Red and Green components of \(CKS\) are generated by iterating \(N + HW\) times the chaotic standard map with \(x_0, y_0\) as the initial conditions and \(K\) as the system parameter. Blue component is generated by iterating \(N + HW\) times the logistic map.

As far as our particular cryptanalysis method is concerned, it is irrelevant how \(X\) and \(CKS\) are related to the secret keys. For our present analysis, we can treat \(X\) and \(CKS\) as randomly generated keys of the system. In the following, we give a brief description of the cryptosystem steps for the encryption of the plain image \(P\) to generate the ciphered image \(C\):

(1) **MIX1:** Mix the plain image components with the key color components \(X^R, X^G, X^B\) as

\[
F^R = P^R \oplus X^R, \quad F^G = P^G \oplus X^G, \quad F^B = P^B \oplus X^B.
\]

(2) **Horizontal diffusion (HD):** For each color component of the resulting image \(F\) from Step 1, mix the properties of horizontally adjacent pixels as

\[
E_{i,1}^U = F_{i,1}^U,
\]

\[
E_{ij}^U = E_{i,j-1}^U \oplus F_{ij}^U, \quad i = 1, \ldots, H, \quad j = 2, \ldots, W,
\]

\[
E_{i,1}^U = E_{i-1,1}^U \oplus F_{i,1}^U, \quad i = 2, \ldots, H,
\]

where \(U\) is one of the color symbols \(R, G, B\). Note that the operations in (3) are all linear. Let \(HD\) denote this linear horizontal diffusion operator. Thus, we have

\[
E = HD(F).
\]

(4)

(3) **Vertical diffusion (VD):** Mix the properties of vertically adjacent pixels of the resulting image from Step 2. The vertical diffusion is expressed as

\[
D_{i,H-1}^R = E_{i,H-1}^R,
\]

\[
D_{i,H-1}^G = E_{i,H-1}^G,
\]

\[
D_{i,H-1}^B = E_{i,H-1}^B,
\]

\[
D_{ij}^R = E_{ij}^R \oplus D_{i,j-1}^R \oplus D_{i+1,j}^R, \quad i = H - 1, \ldots, 2, \quad j = W, \ldots, 1,
\]

\[
D_{ij}^G = E_{ij}^G \oplus D_{i,j-1}^G \oplus D_{i+1,j}^G, \quad i = H - 1, \ldots, 2, \quad j = W, \ldots, 1,
\]

\[
D_{ij}^B = E_{ij}^B \oplus D_{i,j-1}^B \oplus D_{i+1,j}^B.
\]
\[
D^\delta_{ij} = E^\delta_{ij} \oplus D^\delta_{i,j+1} \oplus D^\delta_{i+1,j}, \quad i = H - 1, \ldots, 2, \quad j = W, \ldots, 1,
\]
\[
D^\delta_{ij} = E^\delta_{ij} \oplus D^\delta_{i,j+1} \oplus D^\delta_{i+1,j}, \quad j = W - 1, \ldots, 1,
\]
\[
D^\delta_{ij} = E^\delta_{ij} \oplus D^\delta_{i,j+1} \oplus D^\delta_{i+1,j}, \quad j = W - 1, \ldots, 1,
\]
\[
D^\delta_{ij} = E^\delta_{ij} \oplus D^\delta_{i,j+1} \oplus D^\delta_{i+1,j}. \quad j = W - 1, \ldots, 1.
\]

Again, the operations in (5) are all linear. Let \( VD \) denote this linear vertical diffusion operator. Thus, we have
\[
D = VD(E).
\]

(4) \( MIX2 \): Mix the resulting image component pixels with the \( CKS \) image. The resulting image is the ciphered image \( C \). Namely, we have
\[
C^U = D^U \oplus CKS^U, \quad U = R, G, B.
\]

Using (2), (4), (6) and (7), we can write the encryption as
\[
C = CKS \oplus VD(HD(P \oplus X)).
\]

The flowchart of the cryptosystem is given in Fig. 1. For further information about the original cryptosystem, the reader is referred to Ref. [1].

3. Equivalent description of the cryptosystem

The cryptosystem can be described in a different but equivalent way. Both encryption steps in Section 2 and in this section lead to the same results. However, the new description makes the cryptosystem vulnerable to attacks by dividing the cryptosystem procedure in two majors successive steps: (1) the diffusion process and (2) the masking process. The new steps can be described as the following:

1. \( \text{Horizontal diffusion (HD)} \): For each color component of the plain image \( P \), mix the properties of horizontally adjacent pixels as done in the second step of the original description. Obtain the horizontally diffused image \( H \) as:
\[
H = HD(P).
\]

2. \( \text{Vertical diffusion (VD)} \): Mix the properties of vertically adjacent pixels of \( H \) and obtain the modified image \( V \) as
\[
V = VD(H).
\]

3. Diffuse the key image \( X \) horizontally then vertically and obtain \( X_{HV} \) as:
\[
X_{HV} = VD(HD(X)).
\]

4. Mix the resulting image \( X_{HV} \) from Step 3 and the \( CKS \) and obtain a new key image \( Y \). Hence, we have
\[
Y = X_{HV} \oplus CKS.
\]

4. Mix the new key image \( Y \) from Step 4 with the diffused image \( V \) from Step 2. Obtain the ciphered image \( C \) as:
\[
C = Y \oplus V.
\]

![Fig. 1. Block diagram of the original cryptosystem.](image-url)
Thus, we can generate the ciphered images by two equivalent methods: the method proposed by the authors of Ref. [1] or the second method which merges the two key images $X$ and $CKS$ to obtain a new key image $Y$ before mixing it with the horizontally–vertically diffused plain image $V$. The block diagram of the equivalent cryptosystem is given in Fig. 2. Because all the operations in the original encryption procedure are all linear, it was possible to permute their orders with appropriate modifications. Indeed, starting with (8), we can arrive at the equivalent description as

$$
C = CKS \oplus VD(HD(P \oplus X)),
$$

$$
= CKS \oplus VD(HD(P) \oplus HD(X)),
$$

$$
= CKS \oplus VD(H) \oplus VD(HD(X)),
$$

$$
= CKS \oplus V \oplus X_{HV},
$$

$$
= Y \oplus V.
$$

(14)

4. Proposed attacks

The goal of the attacks described in the following sections is to recover the plain image $P$ from its ciphered image $C$ without knowing the cryptosystem keys $x_0, y_0, K$ and $N$. In order to bypass the need to know the original keys, an attacker only needs to know the equivalent keystream image key $Y$. Indeed, once the attacker knows $Y$, he can use (14) to reveal the plain-text image as

$$
P = HD^{-1}(VD^{-1}(C \oplus Y)).
$$

(15)

4.1. Chosen plaintext attack

Suppose that the attacker has temporary access to the encryption machinery. Then he can choose special images and generate their corresponding ciphered images to recover the equivalent keystream image key $Y$. Suppose the attacker chooses a zero image as an input to the encryption machinery. Using (14) with $P = 0$, we have:

$$
C = Y \oplus VD(HD(0)),
$$

$$
= Y.
$$

Thus, the attacker obtains the secret $Y$ as the ciphered image for his chosen plain image.

![Fig. 2. Equivalent block diagram of the original cryptosystem.](image)

![Fig. 3. Chosen plaintext attack: (a) equivalent key image $Y$, (b) chosen plain image $P = \text{zeros}(200, 200, 3)$, (c) ciphered image $C$ which is equal to the equivalent keystream image key $Y$.](image)
As an illustration of the chosen plaintext attack, we simulated the cryptosystem with the original key parameters $x_0 = 3.98235562892545, y_0 = 1.3453656538912, K = 108.54365761256745, N = 110$. The equivalent key image $Y$ for this choice of parameters is given in Fig. 3(a). The figure also shows the chosen image and the corresponding ciphered image $C$.

4.2. Known plaintext attack

Suppose that the attacker knows one couple of (plain image $P$, ciphered image $C$). Using (14), we have

$$Y = VD(HD(P)) \oplus C.$$  

Note that the attacker can perform the last calculation since the diffusion operations $VD$ and $HD$ do not involve any secret keys. The attacker follows the following steps to reveal the equivalent keystream image key $Y$:

1. Horizontally diffuse the known plain image $P$ and obtain an image $P_h$.
2. Vertically diffuse $P_h$ and obtain $P_{hv}$.
3. Find the equivalent keystream image key $Y$ by applying the following: $Y = P_{hv} \oplus C$.

As an illustration, suppose the attacker knows that the plain Lena image in Fig. 4(a) is encrypted into the ciphered image given in Fig. 4(b). The equivalent key image $Y$ is given in Fig. 4(c). The diffused images $P_h$ and $P_{hv}$ are given in Fig. 4(d) and (e), respectively. The recovered key image $Y$ is shown in Fig. 4(f).

5. Conclusion

Only one couple of (plaintext/ciphertext) was needed to break the cryptosystem in Ref. [1] although that it has good cryptographic properties like diffusion and confusion. It also uses simple mixing and diffusing operations that make it fast and a good candidate to be implemented in the real world security mechanisms. But it needs a major improvement to make it robust against our proposed attacks.

Acknowledgement

Ercan Solak was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Project No. 106E143.
References