Chosen Text Attacks Against an Image Encryption Based on the Kronecker Xor Product, the Hill Cipher and the Sigmoid Logistic Map

George Teşeleanu[1,2]

[1] Advanced Technologies Institute
10 Dinu Vintilă, Bucharest, Romania
tgeorge@dcti.ro
[2] Simion Stoilow Institute of Mathematics of the Romanian Academy
21 Calea Grivitei, Bucharest, Romania

Abstract. In 2023, Mfungo *et al.* presented an image encryption scheme that relies on a series of diffusion techniques and uses a chaotic map to generate three secret keys. Note that two out of three keys are dynamically generated based on the size of the original image, while the remaining key is static. The authors claim that their proposal offers 149 bits of security. Unfortunately, we found a chosen plaintext attack that requires 2 oracle queries and has a worse case complexity of $\mathcal{O}(2^{32})$. If the attacker has access to 1 encryption oracle query and 1 decryption oracle query, we can lower the complexity to $\mathcal{O}(2^{18.58})$. Encrypting an image with Mfungo *et al.*'s scheme has a worst case complexity of $\mathcal{O}(2^{33})$. Therefore, both our attacks are faster than encrypting an image. Our attacks are feasible because the dynamic keys remain unchanged for different plaintext images of the same size, and the static key remains the same for all images.

1 Introduction

The security risks associated with digital images, particularly theft and unauthorized distribution, have been amplified by the widespread use of social media. Consequently, researchers have devoted significant attention to this issue and have developed various techniques to encrypt images. Chaotic maps have emerged as a popular choice due to their high sensitivity to initial conditions and previous states, which makes predicting their behavior difficult. As a result, several novel cryptographic algorithms based on chaos have been developed. However, many image encryption schemes based on chaotic maps suffer from critical security vulnerabilities due to inadequate security analysis and a lack of design guidelines. In fact, numerous compromised schemes exist, which are listed non-exhaustively in Table 1. For further information, please refer to [9,26,28,48].

In [24], the authors propose a new image encryption scheme that combines the Kronecker xor product, Hill cipher and sigmoid logistic map. More specifically,

Scheme	[44]	[23]	[39]	[12]	[13]	[33]	[3]	[10]	[27]	[11]	[6]
Broken by	[19]	[38]	[2]	[42]	[1]	[41]	[10]	[16]	[15]	[46]	[36]

Scheme	[30]	[20]	[31]	[32]	[43]	[45]	[14]	[29]	[25]	[5]
Broken by	[37]	[22]	[40]	[47]	[4]	[21]	[7]	[17]	[18]	[34]

Table 1. Broken chaos based image encryption algorithms.

their algorithm starts by shifting the values in each row of all 4×4 image blocks using the AES shift row operation. Then, the algorithm performs a bitwise xor between the top value of each odd or even column and all other values in the corresponding even or odd column, excluding the top value. Next, the Hill Cipher encrypts each 4×4 block of the result. The resulting image is then xor-ed with a key generated using the sigmoid logistic map. To further obscure the image's pixels, the result is transformed using the Kronecker xor product. Finally, another key generated using the sigmoid logistic map is xor-ed with the output to obtain the encrypted image. Since the sigmoid logistic map is simply used as a pseudorandom number generator (PRNG) and the scheme's weakness is independent of the employed generator, we omit its description and simply consider the two keys as being randomly generated.

The focus of this paper is to conduct a security analysis of the Mfungo *et al.* scheme [24]. We describe two chosen text attacks, which would allow an attacker to decrypt all images of a specific size. To mount such attacks, the attacker needs access to ciphertexts of 2 chosen plaintexts[3], or the ciphertext and plaintext of a plaintext and an adaptive ciphertext[4]. Once the attacker obtains either set of information, he can easily extract the secret keys. According to the authors, the largest image size that they experimented with was 256×256 pixels due to the large computational resources required by their proposal: $\mathcal{O}(2^{33})$ for encryption. Our chosen text attacks have a complexity of $\mathcal{O}(2^{32})$ and $O(2^{18.58})$, respectively. Thus, both attacks require less resources than encryption to recover the secret keys. However, if the attacker has already computed the Hill key, then only 1 chosen plaintext is required, and the complexity of the recovery process becomes $\mathcal{O}(1)$. In summary, the attacks proposed in this paper reduce the scheme's security from 149 bits to 32 bits and 18.58 bits, respectively. Once the Hill key is recovered, the security is reduced to 0 bits. Regarding the case of chosen ciphertext attacks, the repetition code embedded in the Mfungo *et al.* encryption scheme prevented us from devising an efficient attack.

Previous work. The chosen plaintext attack against Mfungo *et al.*'s image encryption scheme was initially presented in [35]. This version of the paper also addresses the security of the scheme in the context of adaptive mixed chosen plaintext/chosen ciphertext attacks.

[3] chosen plaintext attack

[4] adaptive mixed chosen plaintext/chosen ciphertext attack

Structure of the paper. We provide the necessary preliminaries in Section 2. An alternative description of Mfungo *et al.*'s scheme is outlined in Section 3. In Sections 4 and 5 we show how an attacker can recover the secret keys in a chosen plaintext scenario and an adaptive mixed chosen plaintext/chosen ciphertext scenario, respectively. We conclude in Section 6.

2 Preliminaries

Notations. In this paper, the subset $\{1, \ldots, s-1\} \in \mathbb{N}$ is denoted by $[1, s)$. The action of selecting a random element x from a sample space X is represented by $x \xleftarrow{\$} X$, while $x \leftarrow y$ indicates the assignment of value y to variable x. By H and W we denote an image's height and width.

2.1 Mfungo *et al.* Image Encryption Scheme

In this section we present Mfungo *et al.*'s encryption (Algorithm 2) and decryption (Algorithm 3) algorithms as described in [24]. Note that W and H must be divisible by 4.

The first step of the encryption process consists in breaking the image in 4×4 blocks and then circular shifting row i of each block to the left by i positions. The exact function is provided in Algorithm 1 as $shift\_rows$. Note that the function takes as input one of the following matrices

$$shift \leftarrow \begin{bmatrix} 0\,1\,2\,3 \\ 1\,2\,3\,0 \\ 2\,3\,0\,1 \\ 3\,0\,1\,2 \end{bmatrix} \quad \text{or} \quad inv\_shift \leftarrow \begin{bmatrix} 0\,1\,2\,3 \\ 3\,0\,1\,2 \\ 2\,3\,0\,1 \\ 1\,2\,3\,0 \end{bmatrix},$$

one for encryption and the other one for decryption. Then the top values of the resulting matrix are preserved, while all values in even columns[5] are xor-ed with the top value of the previous odd column. In the case of odd columns, the values are xor-ed with the top value of the next column, except their top value. The corresponding function is $xor\_between\_pairwise\_columns$ from Algorithm 1. Using a secret 4×4 matrix h, each row of each 4×4 block is multiplied with h. Hill encryption is presented in Algorithm 1, $Hill$. The resulting image is then xor-ed with $k^{(1)}$. Another diffusion layer is then added, *i.e.* the rows are moved down with 3 positions (see Algorithm 1, $shift\_entire\_rows$). The Kronecker xor transformation is then applied. More precisely, the authors apply the Kronecker product between the image and itself, with the following modifications: the product between two elements from two distinct positions is replaced by xor, while the ones from the same position remain unaltered. The pseudo-code is given in the $Kronecker\_xor\_transformation$ function from Algorithm 1. Finally, we perform a final xor with the second key $k^{(2)}$.

[5] except their top values

Algorithm 1: Helper functions.

Function $shift\_rows(P, shift)$
  for $i \in [0, W)$ *and at each step increment* i *with* 4 **do**
    for $j \in [0, H)$ **do**
      for $k \in [0, 4)$ **do**
        $index \leftarrow i + shift_{k,j \bmod 4}$
        $Q_{i+k,j} \leftarrow P_{index,j}$
  return Q
Function $xor\_between\_pairwise\_columns(P)$
  for $i \in [0, W)$ **do** $R_{i,0} \leftarrow P_{i,0}$
  for $i \in [0, W)$ *and at each step increment* i *with* 2 **do**
    for $j \in [1, H)$ **do**
      $R_{i,j} \leftarrow P_{i,j} \oplus P_{i+1,0}$
      $R_{i+1,j} \leftarrow P_{i+1,j} \oplus P_{i,0}$
  return R
Function $Hill(P, h)$
  for $i \in [0, W)$ *and at each step increment* i *with* 4 **do**
    for $j \in [0, H)$ **do**

$$S_{i,j} \leftarrow P_{i,j}h_{0,0} + P_{i+1,j}h_{0,1} + P_{i+2,j}h_{0,2} + P_{i+3,j}h_{0,3} \bmod 256$$
$$S_{i+1,j} \leftarrow P_{i,j}h_{1,0} + P_{i+1,j}h_{1,1} + P_{i+2,j}h_{1,2} + P_{i+3,j}h_{1,3} \bmod 256$$
$$S_{i+2,j} \leftarrow P_{i,j}h_{2,0} + P_{i+1,j}h_{2,1} + P_{i+2,j}h_{2,2} + P_{i+3,j}h_{2,3} \bmod 256$$
$$S_{i+3,j} \leftarrow P_{i,j}h_{3,0} + P_{i+1,j}h_{3,1} + P_{i+2,j}h_{3,2} + P_{i+3,j}h_{3,3} \bmod 256$$

  return S
Function $shift\_entire\_rows(P, n)$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do**
    $T_{i,j} \leftarrow P_{i,j+n \bmod H}$
  return T
Function $Kronecker\_xor\_transformation(P)$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do**
    for $k \in [0, W)$ *and* $\ell \in [0, H)$ **do**
      if $i = k$ *and* $j = \ell$ **then** $U_{i \cdot W+k, j \cdot H+\ell} \leftarrow P_{i,j}$
      else $U_{i \cdot W+k, j \cdot H+\ell} \leftarrow P_{i,j} \oplus P_{k,\ell}$
  return U
Function $compress\_Kronecker\_xor\_transformation(P)$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do**
    $D_{i,j} \leftarrow \emptyset$
    for $k \in [0, W)$ *and* $\ell \in [0, H)$ **do**
      if $i = k$ *and* $j = \ell$ **then** $E \leftarrow P_{i \cdot W+k, j \cdot H+\ell}$
      else $E \leftarrow P_{i \cdot W+k, j \cdot H+\ell} \oplus P_{i \cdot W+i, j \cdot H+j}$
      $D_{i,j}[E] \leftarrow D_{i,j}[E] + 1$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do**
    $T_{i,j} \leftarrow max\_value(D_{i,j})$
  return T

Algorithm 2: Encryption algorithm.

Input: A plaintext P, two secret keys $k^{(1)}$ and $k^{(2)}$, and a secret matrix h
Output: A ciphertext C

$Q \leftarrow shift\_rows(P, shift)$
$R \leftarrow xor\_between\_pairwise\_columns(Q)$
$S \leftarrow Hill(R, h)$
for $i \in [0, W)$ *and* $j \in [0, H)$ **do** $S_{i,j} \leftarrow S_{i,j} \oplus k^{(1)}_{i,j}$
$T \leftarrow shift\_entire\_rows(S, -3)$
$U \leftarrow Kronecker\_xor\_transformation(T)$
for $i \in [0, W^2)$ *and* $j \in [0, H^2)$ **do** $C_{i,j} \leftarrow U_{i,j} \oplus k^{(2)}_{i,j}$
return C

Algorithm 3: Decryption algorithm.

Input: A ciphertext C, two secret keys $k^{(1)}$ and $k^{(2)}$, and a secret matrix h
Output: A plaintext P

for $i \in [0, W^2)$ *and* $j \in [0, H^2)$ **do** $U_{i,j} \leftarrow C_{i,j} \oplus k^{(2)}_{i,j}$
$T \leftarrow compress\_Kronecker\_xor\_transformation(U)$
$S \leftarrow shift\_entire\_rows(T, 3)$
for $i \in [0, W)$ *and* $j \in [0, H)$ **do** $S_{i,j} \leftarrow S_{i,j} \oplus k^{(1)}_{i,j}$
$R \leftarrow Hill(S, h^{-1})$
$Q \leftarrow xor\_between\_pairwise\_columns(R)$
$P \leftarrow shift\_rows(Q, inv\_shift)$
return P

To decrypt we simply perform all the inverse operations in reverse order. Note that when reversing the Kronecker xor transformation, we recover the matrices from all $W \times H$ block and take a majority vote for each byte. This is done in order to provide protection against data loss and noise alteration. Basically, the compression of the Kronecker xor transformation is used as a repetition code. To describe the compression function, we use several dictionaries $D_{i,j}$ initialized as empty. Each time we access the dictionary with an new "key", a "key-value" entry is created and the corresponding "value" is initialized to 0. Additionally, we use a function *max_value* that, when given a dictionary, returns the "key" with the largest "value".

3 A New Look at Mfungo *et al.*'s Scheme

In this section we present an alternative description of Mfungo *et al.*'s scheme. More precisely, we show how to combine $k^{(1)}$ and $k^{(2)}$ into a single key $k^{(3)}$. The alternative encryption and decryption algorithms are provided in Algorithms 4 and 5.

We further show how we derived the equivalent description of lines 4-7, Algorithm 2. After the $shift\_entire\_row$ operation we obtain

$$T_{i,j} \leftarrow S_{i,j+n \bmod H} \oplus k^{(1)}_{i,j+n \bmod H}.$$

Applying the Kronecker transformation we get

$$U_{i\cdot W+k,j\cdot H+\ell} \leftarrow T_{i,j} = S_{i,j+n \bmod H} \oplus k^{(1)}_{i,j+n \bmod H}$$

when $i = k$ and $j = \ell$ and

$$\begin{aligned} U_{i\cdot W+k,j\cdot H+\ell} &\leftarrow T_{i,j} \oplus T_{k,\ell} \\ &= S_{i,j+n \bmod H} \oplus k^{(1)}_{i,j+n \bmod H} \oplus S_{k,\ell+n \bmod H} \oplus k^{(1)}_{k,\ell+n \bmod H} \\ &= (S_{i,j+n \bmod H} \oplus S_{k,\ell+n \bmod H}) \oplus (k^{(1)}_{i,j+n \bmod H} \oplus k^{(1)}_{k,\ell+n \bmod H}), \end{aligned}$$

otherwise. Finally, we get

$$\begin{aligned} C_{i\cdot W+k,j\cdot H+\ell} &\leftarrow U_{i\cdot W+k,j\cdot H+\ell} \oplus k^{(2)}_{i\cdot W+k,j\cdot H+\ell} \\ &= S_{i,j+n \bmod H} \oplus (k^{(1)}_{i,j+n \bmod H} \oplus k^{(2)}_{i\cdot W+k,j\cdot H+\ell}) \end{aligned}$$

when $i = k$ and $j = \ell$ and

$$\begin{aligned} C_{i\cdot W+k,j\cdot H+\ell} &\leftarrow U_{i\cdot W+k,j\cdot H+\ell} \oplus k^{(2)}_{i\cdot W+k,j\cdot H+\ell} \\ &= (S_{i,j+n \bmod H} \oplus S_{k,\ell+n \bmod H}) \\ &\oplus (k^{(1)}_{i,j+n \bmod H} \oplus k^{(1)}_{k,\ell+n \bmod H} \oplus k^{(2)}_{i\cdot W+k,j\cdot H+\ell}), \end{aligned}$$

otherwise. Note that if we compose $Kr = Kronecker\_xor\_transformation$ with $ser = shift\_entire\_rows$ we get

$$Kr(ser(S,n)) = \begin{cases} S_{i,j+n \bmod H}, & \text{if } i = k \text{ and } j = \ell \\ S_{i,j+n \bmod H} \oplus S_{k,\ell+n \bmod H}, & \text{otherwise} \end{cases}$$

Therefore, if we define $k^{(3)}$ as follows

$$k^{(3)}_{i\cdot W+k,j\cdot H+\ell} = \begin{cases} k^{(1)}_{i,j-3 \bmod H} \oplus k^{(2)}_{i\cdot W+k,j\cdot H+\ell}, & \text{if } i = k \text{ and } j = \ell \\ k^{(1)}_{i,j-3 \bmod H} \oplus k^{(1)}_{k,\ell-3 \bmod H} \oplus k^{(2)}_{i\cdot W+k,j\cdot H+\ell}, & \text{otherwise} \end{cases}$$

we get the equivalent description of lines 4-7, Algorithm 2 provided in lines 4-6, Algorithm 4.

Algorithm 4: Equivalent encryption algorithm.

Input: A plaintext P, a secret key $k^{(3)}$, and a secret matrix h
Output: A ciphertext C

1 $Q \leftarrow shift\_rows(P, shift)$
2 $R \leftarrow xor\_between\_pairwise\_columns(Q)$
3 $S \leftarrow Hill(R, h)$
4 $T \leftarrow shift\_entire\_rows(S, -3)$
5 $U \leftarrow Kronecker\_xor\_transformation(T)$
6 **for** $i \in [0, W^2)$ *and* $j \in [0, H^2)$ **do** $C_{i,j} \leftarrow U_{i,j} \oplus k^{(3)}_{i,j}$
7 **return** C

Algorithm 5: Equivalent decryption algorithm.

Input: A ciphertext C, a secret key $k^{(3)}$, and a secret matrix h
Output: A plaintext P

1 **for** $i \in [0, W^2)$ *and* $j \in [0, H^2)$ **do**
2 $\quad U_{i,j} \leftarrow C_{i,j} \oplus k^{(3)}_{i,j}$
3 $T \leftarrow compress\_Kronecker\_xor\_transformation(U)$
4 $S \leftarrow shift\_entire\_rows(T, 3)$
5 $R \leftarrow Hill(S, h^{-1})$
6 $Q \leftarrow xor\_between\_pairwise\_columns(R)$
7 $P \leftarrow shift\_rows(Q, inv\_shift)$
8 **return** P

4 Chosen Plaintext Attack

A chosen plaintext attack (CPA) is a scenario in which the attacker A briefly gains access to the encryption machine $\mathcal{O}_{enc}$ and is permitted to query it with various inputs. In this way, A generates specific plaintexts that can facilitate his attack and uses $\mathcal{O}_{enc}$ to obtain the corresponding ciphertexts. We prove in this section that Mfungo *et al.*'s image encryption scheme is vulnerable to such attacks.

In the first step of our attack we aim to retrieve $k^{(3)}$. This can be easily done if we encrypt an image I_0 with all its pixels set to 0. By setting all the pixels to 0, after passing the image through lines 1-5, Algorithm 4 we end up with the same image I_0. Therefore, we retrieve the key from $k^{(3)}_{i,j} = C_{i,j}$.

Now we aim to find the secret matrix h. Hence, we create an image I_h such that

$$P_{[0,4),[0,4)} = \begin{bmatrix} P_{0,0} & P_{1,0} & P_{2,0} & P_{3,0} \\ P_{0,1} & P_{1,1} & P_{2,1} & P_{3,1} \\ P_{0,2} & P_{1,2} & P_{2,2} & P_{3,2} \\ P_{0,3} & P_{1,3} & P_{2,3} & P_{3,3} \end{bmatrix} \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

and the remaining pixels are set to 0. Since we are only interested in the first 4×4 block, we will only study its evolution. Thus, after the $shift\_row$ and

Algorithm 6: CPA attack.

Function *compress_Kronecker_xor_transformation_short*(P)
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do**
    if $i = 0$ *and* $j = 0$ **then** $T_{i,j} \leftarrow P_{i,j}$
    else $T_{i,j} \leftarrow P_{i,j} \oplus P_{0,0}$
  return T
Function *main*()
  %recover $k^{(3)}$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do** $P_{i,j} \leftarrow 0$
  Send the plaintext P to the encryption oracle $\mathcal{O}_{enc}$.
  Receive the ciphertext C from the encryption oracle $\mathcal{O}_{enc}$.
  $k^{(3)} \leftarrow C$
  %recover h
  $P_{0,0}, P_{1,0}, P_{2,0}, P_{3,0} \leftarrow 1, 0, 0, 0$
  $P_{0,1}, P_{1,1}, P_{2,1}, P_{3,1} \leftarrow 0, 0, 0, 0$
  $P_{0,2}, P_{1,2}, P_{2,2}, P_{3,2} \leftarrow 1, 0, 0, 1$
  $P_{0,3}, P_{1,3}, P_{2,3}, P_{3,3} \leftarrow 1, 0, 1, 0$
  Send the plaintext P to the encryption oracle $\mathcal{O}_{enc}$.
  Receive the ciphertext C from the encryption oracle $\mathcal{O}_{enc}$.
  for $i \in [0, W^2)$ *and* $[0, H^2)$ **do** $U_{i,j} \leftarrow C_{i,j} \oplus k^{(3)}_{i,j}$
  $T \leftarrow$ *compress_Kronecker_xor_transformation_short*(U)
  $S \leftarrow$ *shift_entire_rows*$(T, 3)$
  $h \leftarrow S_{[0,4),[0,4)}$
  return $k^{(3)}, h$

xor_between_pairwise_columns operations we obtain

$$Q_{[0,4),[0,4)} \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad R_{[0,4),[0,4)} \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore, we obtain that

$$S_{[0,4),[0,4)} \leftarrow \begin{bmatrix} h_{0,0} & h_{1,0} & h_{2,0} & h_{3,0} \\ h_{1,0} & h_{1,1} & h_{2,1} & h_{3,1} \\ h_{2,0} & h_{1,2} & h_{2,2} & h_{3,2} \\ h_{3,0} & h_{1,33} & h_{2,3} & h_{3,3} \end{bmatrix}$$

is exactly the matrix h. Since we already know $k^{(3)}$ and the remaining operations are easily reversible, it results that we can retrieve h from the ciphertext corresponding to I_h. The formal description of our CPA attack is provided in Algorithm 6. Since, we consider the ideal case when oracle answers are relayed unaltered, we can simply recover the image from the first $W \times H$ block, and thus lower our attacks' complexity.

The complexity of Algorithm 6 is $\mathcal{O}(H^2W^2 + 2HW)$ and we need 2 oracle queries. Note that Mfugo *et al.*'s encryption scheme has a complexity of

$\mathcal{O}(2H^2W^2 + 8HW)$ and according to the authors the maximum image size that they experimented on is $H = W = 256$. Thus, in this case, our attack has a complexity of $\mathcal{O}(2^{32})$, while Mfugo *et al.*'s scheme has one of $\mathcal{O}(2^{33})$. Remark that if we already recovered h in a previous iteration, we only need to run lines 8-11, Algorithm 6. Thus, the complexity becomes $\mathcal{O}(1)$ and we need 1 oracle query.

5 Adaptive Mixed Chosen Plaintext/Chosen Ciphertext Attack

A mixed chosen plaintext/chosen ciphertext attack (CP/CCA) [8] is similar to a CPA attack. The main difference is that, in addition to having access to the encryption oracle $\mathcal{O}_{enc}$, A also gains access to the decryption oracle $\mathcal{O}_{dec}$ and is allowed to query it with various inputs. Therefore, A can generate plaintexts and ciphertexts that can aid him in attacking the encryption scheme, and uses $\mathcal{O}_{enc}$ and $\mathcal{O}_{dec}$ to obtain the corresponding ciphertexts and plaintexts. In the case of adaptive attacks, A chooses each text one at a time, based on the responses to his previous queries. We further provide such an attach for Mfungo *et al.*'s image encryption scheme.

The first step of our attack is identical to the first step of the CPA. More precisely, we encrypt an all 0 image I_0, and thus obtain the key $k^{(3)}_{i,j} = C_{i,j}$ directly from $\mathcal{O}_{enc}$.

When constructing the adaptive ciphertext we want to obtain after line 4, Algorithm 5 the image $I_{h^{-1}}$ composed of

$$\bar{P}_{[0,4),[0,4)} = \begin{bmatrix} \bar{P}_{0,0} & \bar{P}_{1,0} & \bar{P}_{2,0} & \bar{P}_{3,0} \\ \bar{P}_{0,1} & \bar{P}_{1,1} & \bar{P}_{2,1} & \bar{P}_{3,1} \\ \bar{P}_{0,2} & \bar{P}_{1,2} & \bar{P}_{2,2} & \bar{P}_{3,2} \\ \bar{P}_{0,3} & \bar{P}_{1,3} & \bar{P}_{2,3} & \bar{P}_{3,3} \end{bmatrix} \leftarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the remaining pixels set to zero. Therefore, after line 5, Algorithm 5 we obtain h^{-1} in the first 4×4 block. Since the *xor_between_pairwise_columns* and *shift_rows* operations are unkeyed, we can easily recover h^{-1} from the plaintext outputted by $\mathcal{O}_{dec}$.

To see how we can achieve this, we first look at the effect of lines 4-5, Algorithm 4 on the image $I_{h^{-1}}$. Let $0^{a,b}$ be a block of width a and height b. After the *shift_entire_rows* function we obtain

$$T \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 & 0^{W-4,1} \\ 0 & 0 & 0 & 0 & 0^{W-4,1} \\ 0 & 0 & 0 & 0 & 0^{W-4,1} \\ 1 & 0 & 0 & 0 & 0^{W-4,1} \\ 0 & 1 & 0 & 0 & 0^{W-4,1} \\ 0 & 0 & 1 & 0 & 0^{W-4,1} \\ 0 & 0 & 0 & 1 & 0^{W-4,1} \\ 0^{1,H-7} & 0^{1,H-7} & 0^{1,H-7} & 0^{1,H-7} & 0^{W-4,H-7} \end{bmatrix}.$$

Let $T^{a,b}$ be the image composed of ab copies of T arranged as an $a \times b$ matrix. After the $Kronecker\_xor\_transformation$ operation we obtain

$$U \leftarrow \begin{bmatrix} T & T & T & T & T^{W-4,1} \\ T & T & T & T & T^{W-4,1} \\ T & T & T & T & T^{W-4,1} \\ T^0 & T & T & T & T^{W-4,1} \\ T & T^1 & T & T & T^{W-4,1} \\ T & T & T^2 & T & T^{W-4,1} \\ T & T & T & T^3 & T^{W-4,1} \\ T^{1,H-7} & T^{1,H-7} & T^{1,H-7} & T^{1,H-7} & T^{W-4,H-7} \end{bmatrix}, \tag{1}$$

where T^0 to T^3 are some known blocks.

Note that the $compress\_Kronecker\_xor\_transformation$ function takes a majority vote between all the $W \times H$ blocks it recovers. Therefore, when constructing our image we can consider $U \leftarrow T^{W,H}$ instead of Equation (1) to minimise our computations. Note that in this case we have only 4 blocks that are not correct before the majority vote, compared to $HW - 4$ correct ones, where $H, W \geq 4$.

We can easily see that after the key addition step each T matrix changes only 4 key bits in an $W \times H$ key block. Therefore, the total number of changed key bits is $4HW$. The formal description of our adaptive CP/CCA attack is provided in Algorithm 7. Note that we use the C++ language operators & as reference to a variable.

The complexity of Algorithm 7 is $\mathcal{O}(10HW)$ and we need 1 encryption oracle query and 1 decryption oracle query. Hence, in this case, our attack has a complexity of $\mathcal{O}(2^{19.32})$, which is significantly lower than the CPA complexity $\mathcal{O}(2^{32})$. As in the CPA case, once h^{-1} is recovered the complexity for further attacks becomes $\mathcal{O}(1)$ and we need 1 encryption oracle query.

Optimization. Given the nature of the Kronecker compression function, we can further improve the attack's complexity by introducing more suitable errors in the U matrix that we construct. For simplicity, we further assume that $W \geq 12$.

Since W is divisible by 4, we can write $W = 2w$. Therefore, we can construct the matrix as follows

$$U \leftarrow \begin{bmatrix} 0^{w-2,1} & T & T^{w,1} \\ 0^{w-2,H-1} & 0 & T^{w,H-1} \end{bmatrix}.$$

From Equation (1) we can see that in this case we have $Hw + 1$ correct blocks compared to $Hw - 1$ incorrect ones. Note that since $w - 2 \geq 4$ we prevent overwriting the T^0 to T^3 matrices from Equation (1) with T, thereby avoiding an increase in the number of incorrect blocks.

The only change needed to optimize our CP/CCA attack (Algorithm 7) is to replace the $key\_transformation$ function with the $key\_transformation\_opt$ function. We present this function in Algorithm 8. In this case, the complexity becomes $\mathcal{O}(6HW)$ instead of $\mathcal{O}(10HW)$. Therefore, when $H = W = 256$ we obtain a complexity of $\mathcal{O}(2^{18.58})$, compared to $\mathcal{O}(2^{32})$ in the case of the CPA.

Algorithm 7: Adaptive CP/CCA attack.

Function $key\_transformation(\&k)$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do**
    $k_{iW,jH+3} \leftarrow k_{iW,jH+3} \oplus 1$
    $k_{iW+1,jH+4} \leftarrow k_{iW+1,jH+4} \oplus 1$
    $k_{iW+2,jH+5} \leftarrow k_{iW+2,jH+5} \oplus 1$
    $k_{iW+3,jH+6} \leftarrow k_{iW+3,jH+6} \oplus 1$
Function $main()$
  %recover $k^{(3)}$
  for $i \in [0, W)$ *and* $j \in [0, H)$ **do** $P_{i,j} \leftarrow 0$
  Send the plaintext P to the encryption oracle $\mathcal{O}_{enc}$.
  Receive the ciphertext C from the encryption oracle $\mathcal{O}_{enc}$.
  $k^{(3)} \leftarrow C$
  %recover h^{-1}
  $key\_transformation(k^{(3)})$
  Send the ciphertext $k^{(3)}$ to the decryption oracle $\mathcal{O}_{dec}$.
  Receive the plaintext P from the decryption oracle $\mathcal{O}_{dec}$.
  $Q \leftarrow shift\_rows(P, shift)$
  $R \leftarrow xor\_between\_pairwise\_columns(Q)$
  $h^{-1} \leftarrow R_{[0,4),[0,4)}$
  %undo the changes to $k^{(3)}$
  $key\_transformation(k^{(3)})$
  return $k^{(3)}, h^{-1}$

Algorithm 8: Optimized helper function CP/CCA attack.

Function $key\_transformation\_opt(\&k)$
  $w \leftarrow W/2$
  $k_{(w-1)W,3} \leftarrow k_{(w-1)W,3} \oplus 1$
  $k_{(w-1)W+1,4} \leftarrow k_{(w-1)W+1,4} \oplus 1$
  $k_{(w-1)W+2,5} \leftarrow k_{(w-1)W+2,5} \oplus 1$
  $k_{(w-1)W+3,6} \leftarrow k_{(w-1)W+3,6} \oplus 1$
  for $i \in [w, W)$ *and* $j \in [0, H)$ **do**
    $k_{iW,jH+3} \leftarrow k_{iW,jH+3} \oplus 1$
    $k_{iW+1,jH+4} \leftarrow k_{iW+1,jH+4} \oplus 1$
    $k_{iW+2,jH+5} \leftarrow k_{iW+2,jH+5} \oplus 1$
    $k_{iW+3,jH+6} \leftarrow k_{iW+3,jH+6} \oplus 1$

6 Conclusions

In [24], a novel proposal for encrypting images was introduced. The authors combined a variety of diffusion techniques to build their encryption algorithm and claimed that the resulting scheme has a security strength of 149 bits. However, our security analysis showed that its actual strength is only $\mathcal{O}(2^{32})$ in the CPA scenario, and $\mathcal{O}(2^{18.58})$ in the CP/CCA scenario. Both attacks require 2 oracle

queries: 2 encryption queries for the CPA, and 1 encryption and 1 decryption queries for the CP/CCA. Consequently, the proposed cryptosystem fails to meet the necessary security strength needed to protect confidential information.

References

1. Alanazi, A.S., Munir, N., Khan, M., Asif, M., Hussain, I.: Cryptanalysis of Novel Image Encryption Scheme Based on Multiple Chaotic Substitution Boxes. IEEE Access **9**, 93795–93802 (2021)
2. Arroyo, D., Diaz, J., Rodriguez, F.: Cryptanalysis of a One Round Chaos-Based Substitution Permutation Network. Signal Processing **93**(5), 1358–1364 (2013)
3. Chen, J.x., Zhu, Z.l., Fu, C., Zhang, L.b., Zhang, Y.: An Efficient Image Encryption Scheme Using Lookup Table-Based Confusion and Diffusion. Nonlinear Dynamics **81**(3), 1151–1166 (2015)
4. Chen, J., Chen, L., Zhou, Y.: Cryptanalysis of a DNA-Based Image Encryption Scheme. Information Sciences **520**, 130–141 (2020)
5. Essaid, M., Akharraz, I., Saaidi, A., Mouhib, A.: A New Approach of Image Encryption Based on Dynamic Substitution and Diffusion Operations. In: SysCoBIoTS 2019. pp. 1–6. IEEE (2019)
6. Essaid, M., Akharraz, I., Saaidi, A., Mouhib, A.: Image Encryption Scheme Based on a New Secure Variant of Hill Cipher and 1D Chaotic Maps. Journal of Information Security and Applications **47**, 173–187 (2019)
7. Fan, H., Zhang, C., Lu, H., Li, M., Liu, Y.: Cryptanalysis of a New Chaotic Image Encryption Technique Based on Multiple Discrete Dynamical Maps. Entropy **23**(12), 1581 (2021)
8. Fischer, M.J.: Cpsc 467: Cryptography and security lecture 5. https://zoo.cs.yale.edu/classes/cs467/2020f/lectures/ln05.pdf (2020)
9. Hosny, K.M.: Multimedia Security Using Chaotic Maps: Principles and Methodologies, vol. 884. Springer (2020)
10. Hu, G., Xiao, D., Wang, Y., Li, X.: Cryptanalysis of a Chaotic Image Cipher using Latin Square-Based Confusion and Diffusion. Nonlinear Dynamics **88**(2), 1305–1316 (2017)
11. Hua, Z., Zhou, Y.: Design of Image Cipher Using Block-Based Scrambling and Image Filtering. Information sciences **396**, 97–113 (2017)
12. Huang, X., Sun, T., Li, Y., Liang, J.: A Color Image Encryption Algorithm Based on a Fractional-Order Hyperchaotic System. Entropy **17**(1), 28–38 (2014)
13. Khan, M.: A Novel Image Encryption Scheme Based on Multiple Chaotic S-Boxes. Nonlinear Dynamics **82**(1), 527–533 (2015)
14. Khan, M., Masood, F.: A Novel Chaotic Image Encryption Technique Based on Multiple Discrete Dynamical Maps. Multimedia Tools and Applications **78**(18), 26203–26222 (2019)
15. Li, M., Lu, D., Wen, W., Ren, H., Zhang, Y.: Cryptanalyzing a Color Image Encryption Scheme Based on Hybrid Hyper-Chaotic System and Cellular Automata. IEEE access **6**, 47102–47111 (2018)
16. Li, M., Lu, D., Xiang, Y., Zhang, Y., Ren, H.: Cryptanalysis and Improvement in a Chaotic Image Cipher Using Two-Round Permutation and Diffusion. Nonlinear Dynamics **96**(1), 31–47 (2019)
17. Li, M., Wang, P., Liu, Y., Fan, H.: Cryptanalysis of a Novel Bit-Level Color Image Encryption Using Improved 1D Chaotic Map. IEEE Access **7**, 145798–145806 (2019)

18. Li, M., Wang, P., Yue, Y., Liu, Y.: Cryptanalysis of a Secure Image Encryption Scheme Based on a Novel 2D Sine–Cosine Cross-Chaotic Map. Journal of Real-Time Image Processing **18**(6), 2135–2149 (2021)
19. Li, S., Zheng, X.: Cryptanalysis of a Chaotic Image Encryption Method. In: ISCAS 2002. vol. 2, pp. 708–711. IEEE (2002)
20. Liu, L., Hao, S., Lin, J., Wang, Z., Hu, X., Miao, S.: Image Block Encryption Algorithm Based on Chaotic Maps. IET Signal Processing **12**(1), 22–30 (2018)
21. Liu, Y., Qin, Z., Liao, X., Wu, J.: Cryptanalysis and Enhancement of an Image Encryption Scheme Based on a 1-D Coupled Sine Map. Nonlinear Dynamics **100**(3), 2917–2931 (2020)
22. Ma, Y., Li, C., Ou, B.: Cryptanalysis of an Image Block Encryption Algorithm Based on Chaotic Maps. Journal of Information Security and Applications **54**, 102566 (2020)
23. Matoba, O., Javidi, B.: Secure Holographic Memory by Double-Random Polarization Encryption. Applied Optics **43**(14), 2915–2919 (2004)
24. Mfungo, D.E., Fu, X., Wang, X., Xian, Y.: Enhancing Image Encryption with the Kronecker Xor Product, the Hill Cipher, and the Sigmoid Logistic Map. Applied Sciences **13**(6) (2023)
25. Mondal, B., Behera, P.K., Gangopadhyay, S.: A Secure Image Encryption Scheme Based on a Novel 2D Sine–Cosine Cross-Chaotic (SC3) Map. Journal of Real-Time Image Processing **18**(1), 1–18 (2021)
26. Muthu, J.S., Murali, P.: Review of Chaos Detection Techniques Performed on Chaotic Maps and Systems in Image Encryption. SN Computer Science **2**(5), 1–24 (2021)
27. Niyat, A.Y., Moattar, M.H., Torshiz, M.N.: Color Image Encryption Based on Hybrid Hyper-Chaotic System and Cellular Automata. Optics and Lasers in Engineering **90**, 225–237 (2017)
28. Özkaynak, F.: Brief Review on Application of Nonlinear Dynamics in Image Encryption. Nonlinear Dynamics **92**(2), 305–313 (2018)
29. Pak, C., An, K., Jang, P., Kim, J., Kim, S.: A Novel Bit-Level Color Image Encryption Using Improved 1D Chaotic Map. Multimedia Tools and Applications **78**(9), 12027–12042 (2019)
30. Pak, C., Huang, L.: A New Color Image Encryption Using Combination of the 1D Chaotic Map. Signal Processing **138**, 129–137 (2017)
31. Shafique, A., Shahid, J.: Novel Image Encryption Cryptosystem Based on Binary Bit Planes Extraction and Multiple Chaotic Maps. The European Physical Journal Plus **133**(8), 1–16 (2018)
32. Sheela, S., Suresh, K., Tandur, D.: Image Encryption Based on Modified Henon Map Using Hybrid Chaotic Shift Transform. Multimedia Tools and Applications **77**(19), 25223–25251 (2018)
33. Song, C., Qiao, Y.: A Novel Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. Entropy **17**(10), 6954–6968 (2015)
34. Teşeleanu, G.: Security Analysis of a Color Image Encryption Scheme Based on Dynamic Substitution and Diffusion Operations. In: ICISSP 2023. pp. 410–417. SCITEPRESS (2023)
35. Teseleanu, G.: Security Analysis of an Image Encryption Based on the Kronecker Xor Product, the Hill Cipher and the Sigmoid Logistic Map. In: ICISSP 2024. pp. 467–473. SCITEPRESS (2024)
36. Teseleanu, G.: Security Analysis of an Image Encryption Scheme Based on a New Secure Variant of Hill Cipher and 1D Chaotic Maps. In: ICISSP 2024. pp. 745–749. SCITEPRESS (2024)

37. Wang, H., Xiao, D., Chen, X., Huang, H.: Cryptanalysis and Enhancements of Image Encryption Using Combination of the 1D Chaotic Map. Signal processing **144**, 444–452 (2018)
38. Wang, L., Wu, Q., Situ, G.: Chosen-Plaintext Attack on the Double Random Polarization Encryption. Optics Express **27**(22), 32158–32167 (2019)
39. Wang, X., Teng, L., Qin, X.: A Novel Colour Image Encryption Algorithm Based on Chaos. Signal Processing **92**(4), 1101–1108 (2012)
40. Wen, H., Yu, S.: Cryptanalysis of an Image Encryption Cryptosystem Based on Binary Bit Planes Extraction and Multiple Chaotic Maps. The European Physical Journal Plus **134**(7), 1–16 (2019)
41. Wen, H., Yu, S., Lü, J.: Breaking an Image Encryption Algorithm Based on DNA Encoding and Spatiotemporal Chaos. Entropy **21**(3), 246 (2019)
42. Wen, H., Zhang, C., Huang, L., Ke, J., Xiong, D.: Security Analysis of a Color Image Encryption Algorithm Using a Fractional-Order Chaos. Entropy **23**(2), 258 (2021)
43. Wu, J., Liao, X., Yang, B.: Image Encryption Using 2D Hénon-Sine Map and DNA Approach. Signal processing **153**, 11–23 (2018)
44. Yen, J.C., Guo, J.I.: A New Chaotic Key-Based Design for Image Encryption and Decryption. In: ISCAS 2000. vol. 4, pp. 49–52. IEEE (2000)
45. Yosefnezhad Irani, B., Ayubi, P., Amani Jabalkandi, F., Yousefi Valandar, M., Jafari Barani, M.: Digital Image Scrambling Based on a New One-Dimensional Coupled Sine Map. Nonlinear Dynamics **97**(4), 2693–2721 (2019)
46. Yu, F., Gong, X., Li, H., Wang, S.: Differential Cryptanalysis of Image Cipher Using Block-Based Scrambling and Image Filtering. Information Sciences **554**, 145–156 (2021)
47. Zhou, K., Xu, M., Luo, J., Fan, H., Li, M.: Cryptanalyzing an Image Encryption Based on a Modified Henon Map Using Hybrid Chaotic Shift Transform. Digital Signal Processing **93**, 115–127 (2019)
48. Zolfaghari, B., Koshiba, T.: Chaotic Image Encryption: State-of-the-Art, Ecosystem, and Future Roadmap. Applied System Innovation **5**(3), 57 (2022)