A Study on Security Analysis of Image Encryption Algorithms

趙, 亮
九州大学システム情報科学府

https://doi.org/10.15017/25136

出版情報：九州大学, 2012, 博士（工学）, 課程博士
バージョン：
権利関係：

A Study on Security Analysis of Image Encryption Algorithms

Liang Zhao

July, 2012

Department of Informatics,

Graduate School of Information Science and

Electrical Engineering, Kyushu University

Contents

List of Figures

List of Tables

Preface

With the development of the computer and network technology, digital media information (specially the multimedia information) is used in many fields more widely, e.g., the industry, medical treatment and academic research. As a result, the corresponding security problem of the digital media information becomes increasingly significant. Moreover, to reduce the volume of the information for the transmission, the digital media information, in general, does the compression as one procedure. Based on this background, there are three kinds of methods for the secure communications of the digital media information. The traditional way is that the digital media information is first compressed to reduce the redundancy, and then encrypted to mask its meaning. However, in some application scenarios, the sender of the digital media information hopes encrypting the original data firstly and the network provider may compress the encrypted information without knowing any knowledge of the original data and the corresponding key. Therefore, the second method which does the encryption followed by the compression has been attracted as the considerable research interest recently. Specially, for the second method, the research of the corresponding encryption algorithms is one of the main research topics in the digital media security. The third method is that the compression of the digital media information and the corresponding encryption can be achieved synchronously. This research always considers the revision of the traditional compression algorithm such as the Huffman coding and the arithmetic coding (AC). The purpose of it focuses on the reduction of the time and computation of each operation (i.e., the compression and encryption), and this kind of research makes the system flexible for the advanced digital media processing.

According to the above analysis, it can be found that the second method and the third method are two interesting topics in the field of the digital media security. In fact, there have been many encryption algorithms belonging to these two methods which are proposed for protecting the secrecy of the digital media information. Correspondingly, the security analysis of the proposed algorithms also becomes important before they are employed in practice. In this thesis, we investigate the security of some encryption algorithms for the digital media information. In particular, as the digital image is one of the main carriers of the digital media information which causes that there have been many proposed image encryption algorithms, our security analysis primarily focuses on the digital image related encryption algorithms. In our analysis, we try to identify some properties in the different encryption algorithms which can be exploited to break the corresponding ciphers.

We first present a scrambling analysis of image spatial scrambling encryption algorithms. As the image scrambling algorithms, e.g., the Arnold cat map and Fibonacci transformation, are widely used to encrypt the digital image, the scrambling degree of the corresponding ciphertext image should be measured. Therefore, an evaluation method based on the bit-plane has been proposed. Specially, the bit-plane theory is the core of this evaluation method. In the evaluation step, the spatial distribution entropy and centroid difference for the bit-plane are used to measure the scrambling degree of each bit-plane. As the relationship between the original image and most significant bit-plane to least significant bit-plane reduces gradually, we set a level decreasing-based weight for each bit-plane when the final scrambling degree is computed. The experiment results show that this evaluation method can be considered as a candidate for finding the scrambling degree of the image scrambling algorithm.

Then, we address a security analysis on an image encryption algorithm of pixel bits and present the comparison with the previous state of the art. We provide the chosen-plaintext attack (CPA) for breaking an image scrambling encryption algorithm of pixel bits which was presented by Ye and published on *Pattern Recognition Letters* in 2010. The proposed attack reveals the equivalent vectors which can substitute for the user-

supplied keys. Compared with the former analysis work observed by Li and Lo which was published on *Signal Processing* in 2011, our attack has the lower complexity which implies that our attack is more efficient than Li and Lo's attack. Moreover, a suggested improvement against our attack is presented in details. In fact, we introduce the self-correlation which comes from the idea of the self-adaptive encryption, proposed by Chen et al. and published on *Journal of Software* in 2005, into the original algorithm to enhance the security. The final simulation results show the performance of this improvement which may be better than the original algorithm.

Finally, we provide a security analysis on the randomized arithmetic codes based on the Markov model and the further analysis on the corresponding improvement scheme. Randomized arithmetic code is seen as a kind of symmetric-key algorithm which can achieve the encryption and the compression for the digital media information synchronously. In 2011, Duan et al. proposed one kind of randomized arithmetic code based on the Markov model (ACMM). This research result is published on *Communications in Nonlinear Science and Numerical Simulation*. In order to achieve the security analysis, we first put forward a formal definition of ACMM, and then explore the corresponding security. In fact, our analysis shows that ACMM is insecure under the ciphertext-only attack (COA) even if a new pseudorandom bit sequence is used for the encryption of each plaintext message. Moreover, an enhanced algorithm which combines ACMM with the randomized arithmetic coding (RAC) is presented. Specially, RAC is introduced by Grangetto et al. and published on *IEEE Transactions on Multimedia* in 2006. However, the security analysis also shows that the combination scheme ACMM+RAC is insecure under the COA. Finally, we present the simulation results to confirm the proposed attacks for ACMM and ACMM+RAC.

Fukuoka Japan, July 2012

Liang Zhao

Acknowledgments

I would like to take the opportunity to thank all the people who supported and accompanied me during my Ph.D studies.

First and foremost, I would like to thank my supervisor Professor Kouichi Sakurai for giving me a very interesting research problem. I would like to present my thanks to Professor Sakurai for having an open door for my questions, and for professor's comments on my works. Each time when I finished a manuscript, Professor Sakurai would give me some detailed comments from both editorial and technical side. I would also very appreciate that Professor Sakurai gave me the freedom to do the research on this field which I liked. When I encountered the technical confusion on my research, Professor Sakurai always supported me which can give me the confidence. If there is the new idea on my research, Professor Sakurai would send to me at the first time.

I am grateful to Professor Takashi Nishide. Specially, when I asked the questions about my research, Professor Takashi Nishide was always full of patience for answering them. Moreover, Professor Nishide always shared the research expertise and new research topics with me. This made me have a great progress in my research field. I would like to thank you, Professor Nishide. I think that I learned a lot from Professor Nishide not only from the research content but also from the research behavior. I think these research behavior can help me in the future.

I would also like to thank Professor Kyung-Hyune Rhee and Professor Avishek Adhikari who were the co-authors on my works. As we had the same research topics, I really enjoyed the co-works with two professors. Every time when I discussed with professors about my research, I can achieved enough information. Specially, if I discussed with

professors about the revision of the manuscript, professors always give me the useful comments.

Moreover, my Ph.D colleagues would obtain my thanks. Thanks for the fun at our trips. We entered our laboratory together, and tried to discuss the research together even if the research topics are not the same. I felt luckily not only for the senseless discussions but also for the friendship among us.

I would specially like to appreciate to my father and mother for their continuous supports and encouragements. During my study, my father and mother experienced every moment of my happiness and sadness.

Lastly, I would be grateful for the governmental scholarship from China Scholarship Council. As the support from the governmental scholarship, I can remove my concerns on living, and only concentrate on my research. I also thank the Ph.D courses scholarship from Kyushu University which gave me the fund support for my travel to attend a number of academic events, including workshop and conference.

Chapter 1

Introduction

In this chapter, we will begin to present our work by giving the motivation for our research. Moreover, we provide the contributions in this thesis, and present its overall structure.

1.1 Motivation

In 1956, the first *computer network* was constructed. After that, various communication networks enter our life and work, e.g., Public Switched Telephone Networks (PSTNs), Public Switched Data Networks (PSDNs), Integrated Service Networks (ISNs) and mobile communication systems [1]. This change brings an important communication vehicle for us, i.e., digital media. In fact, the development of the digital media does always follow the progress of the computer and network technique. Today, we can find the applications of it in many fields, e.g., medical system, government, industry and academic research. This implies that the digital media has become the indispensable tool for our life and work.

Based on the above reason, providing *confidentiality* for the digital media is of significance, specially for the insecure communication channels. Therefore, researchers try to propose some kinds of approaches to protect the digital media information from leaking to unauthorized users. The *encryption algorithm* is seen as the main *protection* method. For the encryption algorithm, as the compression which can reduce the size

of input is the necessary step for the communications of the digital media information, there are three kinds of detailed methods provided for the *secure communications* of the digital media. The traditional approach is that the digital media information is compressed firstly, and encrypted secondly. This method is called the first-compression-then-encryption approach. The second approach is the reverse process which does the encryption at first and then finishes the *compression*. This method can be called the first-encryption-then-compression approach. Recently, there is the third approach which can do the encryption and compression in one step. Usually, the third approach focuses on the compression based encryption algorithm.

Since there are the encryption algorithms for protecting the digital media information, security analysis for these encryption algorithms has received considerable attention. The relationship between the security analysis and the encryption algorithm seems to be like the relationship between the lance and the shield. Generally speaking, the purpose of the security analysis is to evaluate or break the encryption algorithms. For a proposed encryption algorithm on the digital media, it needs a variety of security analyses. This is of great importance if the proposed encryption algorithm intends to be used in practice. Moreover, the security analysis is helpful for designing the more secure encryption algorithms.

1.2 Contributions

According to the above description, it can be found that the study on the security analysis of the encryption algorithm is an interesting and hot topic. Specially, as the still image is one of the main *digital media vehicles*, in this thesis, our works focus on the security analysis of some image encryption techniques and algorithms which belong to the second method and the third method, and present the further consideration. Generally speaking, our works include the following three primary contributions:

- A *scrambling analysis* of image spacial scrambling encryption algorithms is presented. The image scrambling encryption (e.g., the Generalized Arnold cat map)

is one of the main protection methods for the image content. Therefore, intuitively, the *scrambling degree* of the corresponding cipher image should be measured for satisfying the basic security condition. We propose an evaluation method based on the *bit-plane* for evaluating the scrambling degree of the scrambled image. The evaluation method utilizes the spatial distribution entropy and centroid difference for achieving the scrambling degree. The final experiment results demonstrate that this evaluation method can find the scrambling degree efficiently for the image scrambling schemes.

- A security analysis of an image encryption algorithm of pixel bits is proposed. Specially, the chosen-plaintext attack for an image scrambling encryption scheme of pixel bits [85] is presented. Compared with the former analysis work observed by Li and Lo [45], our analysis has the lower complexity. This demonstrates that our attack is more efficient than Li and Lo's attack. Furthermore, a suggested improvement against our attack is provided in details. The performance analysis shows that this suggested improvement may be better than the original algorithm.

- A security analysis of the *Markov model* based randomized arithmetic codes is addressed. For our analysis, at first, we put forward the formal definition of the *randomized arithmetic code* based on the Markov model (i.e., ACMM) [31]. Then, we explore the security of ACMM under the chosen-plaintext attack and ciphertext-only attack. Moreover, an enhanced encryption algorithm which combines ACMM with randomized arithmetic coding (RAC) [42] is presented, and the security of this combination scheme is also analyzed under the ciphertext-only attack. However, the security analysis shows that ACMM+RAC is insecure under the ciphertext-only attack.

The results provided in the thesis have previously been presented in [97, 98, 99, 100, 101]. Specially, the scrambling analysis of the second result is mainly related to the analysis on the spatial scrambling algorithm. The final scrambling degree is used to reflect the scrambling effect of the image scrambling algorithm for the image. The security analysis

of the third result focuses on breaking the image encryption algorithm which is proposed by Ye [85] and finding out the equivalent keys of this algorithm. Note that the image encryption algorithm proposed by Ye [85] can achieve the scrambling of the pixel positions and the encryption of pixel values simultaneously.

1.3 Organization of the Thesis

The remainder of this thesis which includes three parts is organized as follows:

- **Literature Review**: In Chapter 2, we first present some basic knowledge about the concept and applications of the digital media and the research progress of the security protection of the digital media. In particular, we briefly review the image encryption category and list some examples about the image encryption algorithms. Secondly, we provide the category of the security and the security considerations which include the target of adversary, category of attack scenarios and measurement of attack. Furthermore, some examples on the security analysis of the image encryption are also presented.

- **Our Results**: In Chapter 3, we propose a scrambling evaluation method based on the bit-plane for evaluating the scrambling degree of the scrambled image which is encrypted by the spatial scrambling algorithm. In Chapter 4, we present the details of our attack on an image encryption algorithm of pixel bits and give the comparison with the previously published attack. In Chapter 5, we provide a security analysis about a randomized arithmetic coding based on Markov model. In fact, we put forward a formal definition of this randomized arithmetic coding firstly. Then, we present our analysis on it. Moreover, we also address a security analysis on the improvement of this randomized arithmetic coding.

- **Conclusions**: In Chapter 6, we present the summary about each result in this thesis, and provide some possible directions of the future work.

Chapter 2

Preliminaries

In this chapter, on one hand the basic information about the digital media and the corresponding research progress about the security of the digital media are introduced. Specially, the image encryption classifications and some corresponding encryption algorithms are listed. On the other hand, the categories of the security and some security considerations (e.g., target of adversary, categories of attack scenarios and measurement of attack) are also presented. Moreover, some examples about the security analysis of the image encryption are provided.

2.1 Security Protection for Digital Media

2.1.1 Concept and Application of Digital Media

When people enter into the 21^{st} century, the need for the information is becoming increasingly large. Specially, with the development of the *computer technique* and *network technique*, people can obtain the necessary information from many kind of channels and methods. This implies that the information vehicle is gradually not dependent on the material which is like the paper, but related to the *digital media* which can be used to load the digital information.

Digital media denotes the information vehicle which can be used for the record, disseminating, acquisition, etc. in the form of the binary system. Rafiq et al. [2] have

listed some examples about the digital media, e.g., the compact disc, digital video and e-book. According to the description from Rafiq et al. [2], the digital media is seen as one kind of media with the electronic data. Specially, the media data is presented in the digital form for the *post-processing*.

It can be found that the digital media is being developed for associating with many technical problems of our life and work. This implies that it has been widely used in our world. In general, there may be two main categories for the digital media. Firstly, the digital media is assorted based on the time. This division implies the fact that the media content can be the same or be changed with the passage of time. If the media content is not changed it can be so-called *still media*. Secondly, the digital media is assorted according to the number of the information vehicles. If there is only one kind of information vehicle, this media is called as *single media*. Otherwise, this digital media belongs to the *multimedia*. Of course there may be some other categories, e.g., the classification based on the origin of data.

Nowadays, the digital media usually refers to the multimedia which has the complex representation and has been one of the necessary information receivers for people's life and work. The term 'multimedia' in fact was first created by Bob Goldstein in 1966. After that, this word was imparted by many meanings during some past years. About its concept, Williams et al. [3] provided a brief working definition about the multimedia, i.e., *multimedia=variety+integration*. Based on this point, we can consider that the *multimedia* is the combination of a variety of information forms (i.e., single media) which is used for the information share and communications.

The above description indicates that some different digital media types are integrated together to acquire the multimedia. If compared with the traditional forms of the printed and hand-produced material, in fact, the term multimedia has three primary merits which are described as follows:

- The cost of the information vehicle is usually low.
- This kind of information is easy to be stored and the corresponding period of

validity is long.

- The kind of information is easy to be spread.

Based on these merits of the multimedia, the communications of the multimedia information is generally being the significant *medium* for the intercourse of the digital information. Undoubtedly, the above merits are not only related to the multimedia but also suitable for the wider concept, i.e., digital media.

We turn back to the original topic. Generally speaking, the digital media includes the text, *still image*, video, audio, etc. which are the main *vehicles* of the digital information. For the term multimedia, it is seen as the interactivity content forms of these mediums. Therefore, according to the various formats of the digital media, the application of the digital media can be in many fields such as the education, finance, entertainment, medical treatment, military and cultural activities. The details about the application of the digital media can refer to the review [3].

2.1.2 General Description on Encryption of Digital Media

As the wide use of the digital media (e.g., multimedia) in many form, a crucial issue about the security of the digital media is produced. Although the digital media has many innate merits compared with the traditional media, it is still one kind of expression form for the information. Therefore, the importance of the security of the digital media is the same as the importance of that of the traditional media. There are the following three challenges about the protection of the digital media:

(1) Nowadays, as the used network is the *shared channel*, especially for *Internet*, it is easy for the adversary to eavesdrop on or falsify the unencrypted digital media information under the transmission.

(2) For the significant digital media information, e.g., the reports about the governmental conferences, the information about the business conferences and the medical image, the secrecy for the storage is necessary.

(3) For the digital media consumable, e.g., the MP3 music, the high-quality image and the digital TV, it is crucial to prevent the unauthorized user accessing them.

To settle the above security issues, the encryption can be considered as one of the effective and direct solutions. This is based on the fact that the user can obtain the digital media information until he/she has the corresponding secret keys. Therefore, it can be found that this is useful for ensuring the benefit of the manufacturer and the right of the consumer.

For the practical situation, as the digital media information usually has the huge volume, there is one necessary procedure called compression which is used to reduce the volume of the transmitted data in the network. Based on this precondition, there are three kinds of methods for the secure communications of the digital media, specially for the multimedia. The first and traditional method is that the data of the digital media is compressed firstly to reduce the redundancy, and then encrypted to keep its secrecy. For this method, the traditional compression and encryption algorithms can be utilized. However, according to the research background in the works [4, 5], in some application scenarios, the sender of the digital media intends to encrypt the original data at first for masking the meaning and the *network provider* may compress the encrypted information without knowing any knowledge on the original data of the digital media and the corresponding secret key. According to this scenario, the second method which does the encryption at first and then achieves the compression has been attracted as the considerable research interest recently. Specially, for this method, some researchers on the information security and the signal processing focus on finding and proposing the encryption algorithms which make use of the *characteristics* of the digital media, e.g., [55, 88, 61, 85, 39]. Moreover, except the above two methods, there is another candidate which does the encryption and compression simultaneously. In fact, this research is main about the revision of the traditional compression algorithm such as the *Huffman coding* and the arithmetic coding for implementing the confidentiality protection. There are some examples in the works [36, 44, 76, 78, 31]. However, some researchers also found the method that the compression can be inserted into the encryption algorithm. The

typical example is from the works [6, 7]. Generally speaking, the third method focuses on the reduction of the time and computation of each operation (i.e., the compression and encryption), which may be useful for the processing (or post-processing) about the digital media.

In particular, as a large amount of information can be achieved from the *vision* and the still image is one of the important digital media for transmitting the visual information, in the thesis, the main research is about the image encryption. Moreover, according to the above analysis, it can be found that the second and third method do attract the more researchers recently. Therefore, in the following section, the current research progress about the protection of the image content with the encryption is introduced, which specially focuses on the second and third methods.

2.2 Research Progress of Protection for Image Content

For the still image, it has some characteristics which are different from the text [8] or the audio, such as the large volume, the high fidelity and the strong relativity. Therefore, based on the *characteristics* of the still image, there are three essential conditions for the image encryption in the second method and the third method.

- (*For the second method*): As the image data has the large volume, for improving the encryption speed, the corresponding encryption algorithm should be efficient.
- (*For the second method*): As there is the strong relativity among the image pixels, the corresponding encryption algorithm should break the relationship among the image pixels. Specially, it is better to achieve the *diffusion* in the encrypted image for promoting the security.
- (*For the third method*): If the compression is produced in the encryption procedure, the compression ratio should be achieved to a certain level.

According to the above conditions, there have been some proposals about the image encryption. Researchers of the works [55, 88, 61] presented some image encryption al-

gorithms which are based on the higher-dimensional Fibonacci transformation, SCAN language and T-matrix, respectively. These image encryption algorithms are based on the fast scrambling of the image pixels. In 2010, Ye [85] proposed an image scrambling encryption of pixel bit. This method can implement the permutation of image pixels and the encryption of the values of the image pixels simultaneously. The similar idea is also used in the work [39], which focuses on the disposal of digital medical images. Podesser et al. [59] introduced a kind of selective image encryption algorithm which considers the *gray-scale image* as eight bit-plane and only encrypts the four or five most *significant bit-plane* (MSB). This kind of algorithm can reduce the encrypted image data for enhancing the efficiency. Chen et al. [29] also provided a SCAN-CA-based image encryption algorithm. This algorithm is based on the *permutation* of the image pixels, which uses the *SCAN language*, and replacement of the pixel values, which uses the recursive *cellular automata* (CA). Actually, this algorithm can be seen as the synchronous *stream cipher*. Wang et al. [56] proposed an approach of optical image encryption with binary *Fourier transform* computer-generated hologram (CGH) and pixel-scrambling. For this kind of encryption algorithm, the orders of the pixel scrambling and the encrypted image need to be used as the keys to decrypt the original image.

Moreover, if considering the compression and encryption simultaneously, some related works were also presented. For example, the compression-encryption algorithms which are based on the SCAN technique were presented in the works [24, 54]. Puech et al. [9] provided a reversible method which transfers the image quickly and safely. Specially, this method can achieve the *lossless compression* at the same process. Duan et al. [31] proposed a kind of randomized arithmetic coding based on the Markov model. This encryption algorithm makes use of the permutation of probability of each symbol in the *Markov tree* to encrypt the plaintext. The similar works about the revision of arithmetic coding can be found in the works [36, 44, 76]. In 2005, Wu et al. [78] also presented a combination algorithm of the compression and encryption, which uses the *multiple Huffman tables* to finish these two operations. Moreover, a secret key encryption algorithm was developed for both image data encryption and compression in

the work [79]. Specially, in order to achieve the lossless data compression effect, the *quadtree data structure* is used to represent the image. In order to finish the encryption, the scanning sequences of image data are provided.

After introducing the knowledge about the image encryption algorithm and the corresponding progress, we should consider the corresponding research from the contrary side, i.e., *security analysis*. Therefore, in the next section, some basic information about the security analysis for the image encryption is presented, which includes the categories of the security, target of adversary, categories of attack scenarios and measurement of attack.

2.3 Categories of Security

There can be some categories for the *security* of a *cryptosystem* which also includes the image encryption algorithms in the literature. In this section, two different concepts of security are presented, i.e., the *unconditional security* and *computational security*.

2.3.1 Unconditional Security

For a cryptosystem, if it is unconditionally secure, it can not be broken even under the condition of infinite computational resources [13]. In fact, Shannon [10] had presented a formal definition with the so-called name "*perfect secrecy*" in the paper "Communication Theory of Secrecy Systems" which was published on *Bell Systems Technical Journal* in 1949. According to the description from Shannon, the perfect secrecy implies that an adversary can not achieve any useful knowledge on the *plaintext* according to the observation of the corresponding *ciphertext*. The definition of the perfect secrecy [11, 10, 13] is presented as follows:

Definition 2.1 (Perfect secrecy)**.** *A cryptosystem has the perfect secrecy if*

$$\Pr[P = p|C = c] = \Pr[P = p]$$

for all $p \in \mathcal{P}$ and $c \in \mathcal{C}$. That is, the a posteriori probability that the plaintext is p, given

that the ciphertext c is observed, is identical to the a priori probability that the plaintext is p.

However, generally speaking, a cryptosystem which is unconditionally secure in a *attack scenario* may be easy to be broken in another attack scenario. For example, a cipher which may be unconditionally secure against the ciphertext-only attack (COA) (see 2.5 about the ciphertext-only attack) has been not proved that this cipher is also unconditional secure in some other attack scenarios (e.g., chosen-plaintext attack (CPA) (see 2.5)). Therefore, the definition of the unconditionally security is not equal to the definition of the perfect secrecy [13].

According to Shannon's opinion, the perfect secrecy can only be achieved if the length of the secret key equals or exceeds the length of the plaintext [10]. From this point, the perfect secrecy is impractical. However, the perfect secrecy can be realized by the cipher *One-time pad* which was depicted by Gilbert Vernam at first even if there was no mathematical proof. Moreover, Shannon presented the proof that the One-time pad [43] can provide the perfect secrecy. Specially, in One-time pad, the key k is chosen uniformly at random and there is no two plaintexts which are encrypted by using the same key k. Of course this produces the key management issues which limit the utilization of the cipher One-time pad in some commercial applications. Nevertheless, the cipher One-time pad has been employed in the military and diplomatic contexts in which the unconditional security may be very crucial [11]. The description about the One-time pad is as follows [13, 11]:

Definition 2.2 (One-time pad)**.** *Let $n>1$ be the length of the plaintext, and take $\mathcal{P}=\mathcal{C}=\mathcal{K}=\mathbb{Z}_2^n$. For $p=(p_1,\ldots,p_n)\in\mathcal{P}$, $c=(c_1,\ldots,c_n)\in\mathcal{C}$ and $k=(k_1,\ldots,k_n)\in\mathcal{K}$, the encryption is defined as*

$$E_k(p) = (p_1 + k_1, ..., p_n + k_n) \bmod 2.$$

The decryption is identical to the encryption. then

$$D_k(c) = (c_1 + k_1, ..., c_n + k_n) \bmod 2.$$

2.3.2 Computational Security

As our description, the above unconditional security considers the impractical situation. In fact, we should assume a practical environment that the adversary does not have infinite computational power to break the cryptosystem. Under this assumption, what is the security about the cryptosystem? Therefore, this requires us to present the practical definition on the security, i.e., the computational security.

Before we give the definition of the *computational security*, the concept about the exhaustive search [43] is first shown. For a cryptosystem, when the adversary is unable to make use of any weakness from it which would make his/her task easier, the exhaustive search can be always used. In general, it can also be called brute-force attack. The definition of the *exhaustive search* [13] is as follow:

Definition 2.3 (Exhaustive search)**.** *Exhaustive search is called an approach that tries all possible candidates in the search space until the right one is discovered. On average half of the candidates have to be tested.*

For the practical application, a cryptosystem is allowed to have the capability to resist against the adversary who has the bounded computational resources. This is rather reasonable compared with the unconditional security. The corresponding definition of computational security [13] is presented as follow:

Definition 2.4 (Computational security)**.** *A cryptosystem provides n bits of security if the best attack for breaking it requires the computational effort which equals to an exhaustive search over 2^n values.*

The above definition tells us a condition that if a cryptosystem provides n bits of security in which 2^n operations are too large to be computed currently or in the near future, it can be computationally secure. Specially, the parameter n is largely decided by the size of the *secret key*. The support for the above description is based on one possibility, i.e., the adversary may have to search and try all the 2^n secret keys until the right key is discovered. Therefore, the key size is generally used to generate the upper bound of the parameter n and sometimes the lower bound [13].

However, there is a problem that the practical cryptosystem is hard to be proved to be computationally secure. Therefore, the researchers usually focus on the study of a cryptosystem secure against some typical attacks. This is seen as the lower bound for a cryptosystem but does not imply that the cryptosystem can be secure against some other attacks.

The above information has introduced two definitions about the security. Generally speaking, they have given out the target for the designer of the cryptosystem. However, under this background, another question is produced. i.e., what is the target of an adversary? Moreover, How to define the success of an adversary? About these questions, we will present a brief overview on the goals of different attacks and the measurement on the successful attack in the following sections.

2.4 Targets of Adversary

Obviously, under any situation the highest purpose of an *adversary* is to find a method to reveal the user-supplied secret key. However, this does not imply that the adversary can obtain the information about the secret key each time. Generally speaking, the adversary may achieve much less information. With this in mind, we present the different goals for the possible attacks in the descending order which are as follows [13, 14, 12]:

- *Total Break*: The adversary reveals the user-supplied secret key k which is used for the encryption.

- *Global Deduction*: The adversary finds an algorithm A which equals to $E_k(\cdot)$ or $D_k(\cdot)$ without knowing the actual user-supplied secret key k.

- *Local Deduction*: The adversary can recover the plaintext/ciphertext of a previously unseen ciphertext/plaintext.

- *Distinguishing Algorithm*: The adversary can distinguish the output of a cipher with the randomly chosen encryption key from the output of a random permutation.

For an adversary, if he/she can reveal the user-supplied secret key k (i.e., Total break), he/she is able to achieve the other four purposes. This implies an inclusion relation among the goals of the attacks, i.e., achieving the higher goal can also achieve all the lower goals.

Beside the introduction of different goals of the attacks, we should also know the details about abilities of the adversary for the attack. In fact, an adversary who can access to the useful data (e.g., plaintext/ciphertext) is more powerful than an adversary who can just do the *eavesdropping*. Moreover, for the different types of accessed data, the power of the adversary is also different. In the next section, we will present the information.

2.5 Categories of Attack Scenarios

The categories of the capability of an adversary is based on $Kerckhoffs'\ Principle$ or $Kerckhoffs'\ Assumption$ [66], which is from six requirements for a usable field cipher of Kerckhoffs [67]. This principle or assumption tells us that when cryptanalyzing a cryptosystem, a general assumption is that cryptanalyst can acquire the information on the design and working of the studied cryptosystem. This implies that for any adversary, he/she can know everything on the cryptosystem except the user-supplied secret keys k for the encryption and decryption. Usually, this is a basic standard for the encryption algorithm in nowadays' secure communications. Therefore, based on the above principle, attacks are classified according to the types of the data [14, 12, 62] which are as follows, and the difficulty level is from the hardest one to the easiest one:

- Ciphertext-Only Attack (COA): The adversary only possesses a string of ciphertext C generated by the encryption. He/she tries to reveal some information about the plaintext P.

- Known-Plaintext Attack (KPA): The adversary can possess a string of plaintext P, and the corresponding ciphertext C.

- Chosen-Plaintext Attack (CPA): The adversary has obtained temporary access to the encryption machinery $E_k(\cdot)$. Hence he/she can request the encryption of the chosen plaintext string P, and intercept the corresponding ciphertext string C.

- Adaptively Chosen Plaintext Attack (ACPA): The adversary can access to the encryption machinery $E_k(\cdot)$ temporarily. He/she can choose the plaintext P after observing the previous chosen plaintext/ciphertext pairs.

- Chosen-Ciphertext Attack (CCA): The adversary has obtained temporary access to the decryption machinery $D_k(\cdot)$. Hence he/she can request the decryption of the chosen ciphertext string C, and intercept the recovered plaintext string P.

- Adaptively Chosen-Ciphertext Attack (ACCA): This is the adaptive version of the CCA. The adversary can access to the decryption machinery $D_k(\cdot)$ temporarily. He/she can choose the ciphertext C after observing the previous chosen ciphertext/plaintext pairs.

Obviously, the attack we consider is largely dependent on the number of the data and the type of the required data. This implies that the attacks are different and have a difficult degree. Generally speaking, if the cipher is vulnerable to the COA, it is considered weak, and if the cipher is secure against a very strong attack such as ACCA, it can be the convincing analysis on the security of this cipher. The KPA and the CPA are still seen as the realistic scenarios when we consider the attacks in the real world applications. Moreover, if the cipher is secure against the higher-level attack (i.e., ACCA), it can be also secure against the other attacks whose levels are lower than this attack.

The above introduction presents the abilities of the adversary which have been classified six types. After we know about them, we should focus on fixing the question that how to evaluate the success of the attack. The next section can give the answer.

2.6 Measurement of Attack

In order to evaluate the different types of the attacks, the importance is that which resources can be used to determine the complexity of the attack, and how many *resources* are consumed. According to [14], we list the following three resources that are usually used to measure the complexity of the attack.

- *Time Complexity*: The time required to mount the attack. This is usually considered as the first requirement or the unique thing for a attack.

- *Storage Complexity*: The amount of memory required during the attack. This complexity is also significant for an attack. If the amount of the memory is very large, the corresponding attack is also impractical.

- *Data Complexity*: The amount of data for a successful attack. If the attack needs much time to generate the data which is far from the normal usage patterns, the effect of the attack will be limited.

In general, when a cryptosystem is analyzed by itself, the above resources and data types are the most crucial indexes for evaluating the proposed attack [14]. For an adversary, he/she should try to find and realize an attack of which complexities are lower than the bound estimated from the designer of the cryptosystem. However, this does not imply that the successful attack can make the cryptosystem vulnerable or useless in practice as the proposed attack may still be computationally infeasible. A realistic meaning is that the successful attack which does not break the cryptosystem can reveal the previously unknown weakness of the cryptosystem which may be helpful for developing the further attack.

This section presents the complexity of the attack which is used to evaluate the success of the attack. In the next section, we will give a brief overview about the security analysis of the image encryption algorithms.

2.7 Examples on Security Analysis of Image Encryption

According to the above basic introduction on the attack, when we want to analyze a cryptosystem, various attacks can be proposed and used for it. If considering the target, the adversary can present the corresponding attacks for the targeted cryptosystem based on his/her intention. For the image encryptions, there exist some corresponding security analyses on them [25, 75, 48, 19, 62, 30, 65, 45, 91, 89, 92]. We present some examples which are as follows:

(1) In 2002, Chang et al. [25] analyzed the security of the binary image encryption-compression algorithm proposed by Chung et al. [16]. According to Chang's analysis, this encryption-compression algorithm is vulnerable against his proposed attack, specially under the condition that the same key is used. This encryption algorithm can be broken with some plaintext image and ciphertext image pairs.

(2) In 2005, Wang et al. [75] made use of the chosen-plaintext attack to break a three-dimensional (3D) Cat map based symmetric image encryption algorithm [26], which uses the 3D Cat map to scramble the positions of image pixels. Specially, this attack is composed of two mutually independent processes, i.e., the analysis of the spatial permutation process and the analysis of the diffusion process. Based on this attack, the equivalent initial condition of diffusion process and a valid equivalent 3D Cat matrix can be revealed.

(3) In 2008, Li et al. [48] discussed the security analysis on the general model of permutation-only multimedia ciphers. In that paper, when the plaintext is of size $M \times N$ and with L different levels of values, a quantitative cryptanalysis presents that all permutation-only multimedia ciphers are practically insecure against KPA/CPA under the assumption of a uniform distribution of each element in the plaintext. Moreover, the corresponding computational complexity of these attacks is $\mathrm{O}(n \times (MN)^2)$. Specially, the n is the number of the used known/chosen plaintexts.

(4) In 2011, Zhou et al. [91] first presented the weakness of an encryption algorithm which uses multiple Huffman tables [78]. Then, the effective chosen-plaintext attack and known-plaintext attack are introduced. The theoretic analysis and simulation results show that the secret key could be recovered with about 10 blocks of known plaintexts and ciphertexts. Moreover, the ciphertext-only attack was also presented for analyzing this encryption algorithm.

2.8 Conclusions

In this chapter, we briefly reviewed about the basic information of the research progress on the security of the digital media, some definitions and considerations about the security analysis. Firstly, the concept and applications about the digital media were introduced. Specially, we emphasized that the confidentiality is of importance for the digital media and the still image is one of the main communication vehicles in the digital media. Therefore, we presented the research progress on the protection of the still image. Secondly, the categories of the security, target of adversary, categories of security and measurement of attack were described. According to these considerations, some examples about the security analysis of the image encryption algorithms were introduced. In the following chapters, we will provide our results about the security analysis of image encryption algorithms.

Chapter 3

Scrambling Analysis of Image Spatial Scrambling Encryption

In this chapter, we discuss about the scrambling of the spacial scrambling image encryption algorithms. This discussion is related to the evaluation of the scrambling degree of the scrambled image. For presenting the analysis on the scrambling degree, an evaluation method is proposed which is based on the spatial distribution entropy and centroid difference.

3.1 Introduction

3.1.1 Research Background

Image spatial scrambling, which is suitable for practical applications on information protection [38, 39, 55, 88, 61], is one kind of the most prevailing protection methods for image data. It can permute the order (or position) of the plaintext image for achieving the encryption effect. Generally speaking, it breaks the correlation among the pixels. For the image spatial scrambling algorithm, *Generalized Arnold cat map* [61], *Fibonacci transformation* [93], *Baker map* and *sub-affine transformation*, etc. are widely used. Specially, Generalized Arnold cat map is seen as a typical scrambling map [94]. Moreover, the image spatial scrambling algorithms are also used for the data hiding and digital watermarking recently [86, 50, 52]. e.g., Ye et al. [86] proposed the

scrambling method based on the chaotic cellular automata, which is used to scramble the digital image as a pretreatment for the watermarking process. Moreover, Zhu et al. [50] introduced one kind of novel image scrambling algorithm for the digital watermarking. Furthermore, In Lin's work [52], the pixel scrambling method is adopted by the information hiding. As this technique has the wide applications in the protection of the image data and digital watermarking, the corresponding scrambling performance is of great significance. Therefore, it is necessary to evaluate the corresponding scrambling performance.

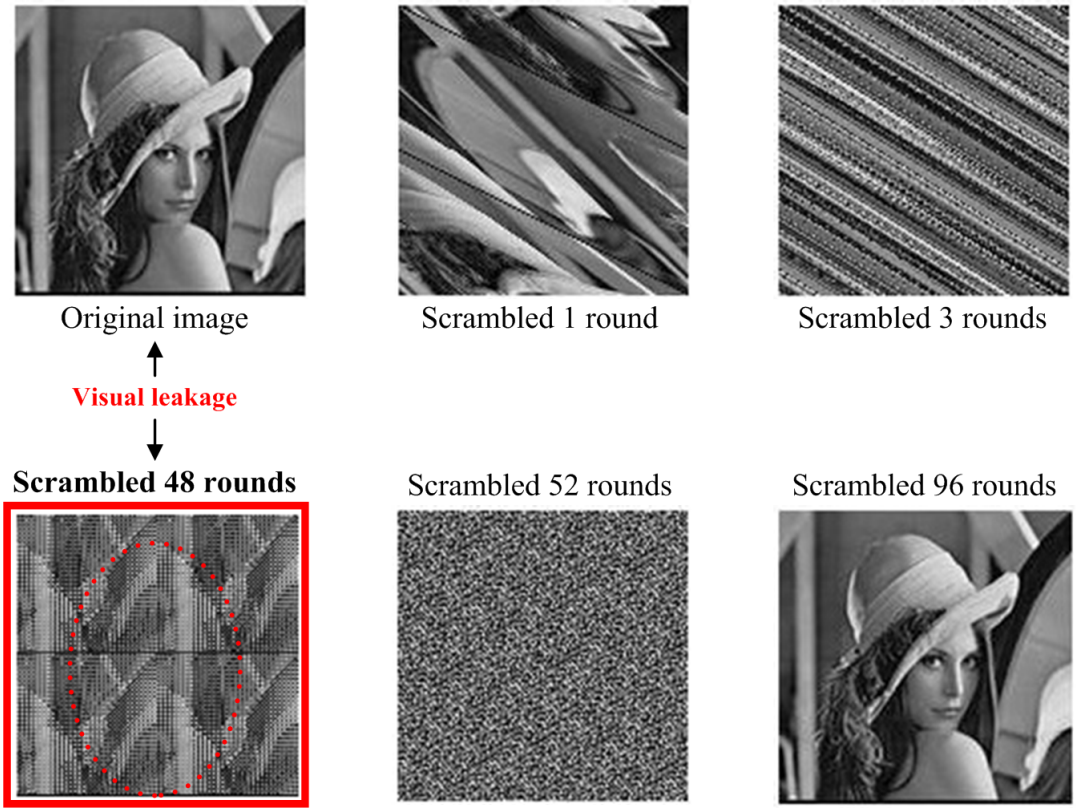

Figure. 3.1: Scrambled images of gray-scale image 'Lena' of size 128×128: Case of Generalized Arnold cat map.

3.1.2 Previous Work

For evaluating these spatial scrambling algorithms, some image scrambling evaluation methods are proposed [49, 87]. Specifically, Yu et al. [87] analyzed the structure of the scrambled image compared with the plaintext image, and proposed a method which makes use of the correlation of *adjacent pixels* between the scrambled image and the plaintext image to evaluate the image scrambling. Moreover, Li [49] presented a measure for the image *scrambling degree*, which takes advantage of the gray level difference and information entropy. According to the author, this new proposal evaluates the scrambling degree from both the local discreteness and the global uniformity which consider three aspects of the image, i.e., the randomness in statistical distribution, the discreteness and the uniformity of the discreteness [49].

3.1.3 Challenge Issues

For the scrambling degree evaluation of the image spatial scrambling, how to accurately detect the scrambling degrees of different scrambled images which are scrambled by a image spatial scrambling algorithm and how to analyze the 'weakness' (see Fig. 3.1) about them in practice are of significance. Of course some image scrambling evaluation methods have been proposed for giving the scrambling degree. However, the following four challenges about the evaluation, if possible, still should be considered:

- When the plaintext image is scrambled, not only the positions of pixels are permuted, but also the relationship among the adjacent pixels are completely disordered. This implies that final scrambling evaluation should consider both of the *values* and *positions* of pixels.

- The scrambling degree from the evaluation method can reflect the relationship between the scrambled image and the used spatial scrambling algorithm effectively, such as the relationship between the iteration rounds of the spatial scrambling algorithm and the corresponding scrambled image.

- If there is the weakness (e.g., *visual leakage*) on the spatial scrambling algorithm which can be reflected by the corresponding scrambled image, the scrambling degree from the evaluation method can also reflect this weakness obviously.

- As the pixel value of the gray-scale image (or color-scale image) has a large value range (e.g., {0, 1,. . ., 255}), the scrambling evaluation based on the original pixel value (e.g., the gray-scale pixel) may be not easy to computed. This implies that a simple basis from the image can facilitate the scrambling evaluation. Moreover, as the scrambled image has large volumes of data, it is better that the evaluation algorithm can achieve the approximate scrambling degree by using the less image data. Specially, this approximate scrambling degree is similar to the scrambling degree achieved by using all the image data.

3.1.4 Our Contribution

According to the analysis and the summarization of Subsection 3.1.3, our priority focus is to present an effective evaluation method which can measure the scrambling degree of the scrambled image, and explore the existing weakness in the image spatial scrambling algorithm.

In this chapter, an scrambling evaluation method based on the bit-plane is proposed. In our analysis, the gray-scale image are considered as the test image. The bit-plane theory is seen as the core of the proposed scrambling evaluation method. In the evaluation process, the spatial distribution entropy and centroid difference for bit-planes are used to measure the scrambling degree of the bit-plane. After that, the value of the scrambling degree of the whole image is obtained according to weighted sum of scrambling degree of bit-planes (as the steps in Section 5.3.3). Note that for a general gray-scale image such as 'Lena', as the correlation among the original gray-scale image and most significant bit-plane to least significant bit-plane reduces gradually, we can set a level-decreasing based weight for each bit-plane. In particular, as the last four least significant bit-planes have less relationship with the original image, instead of using the whole original image

data, we can use the first four most significant bit-planes for the evaluation. This can reduce 50% of the computational cost. The experimental results show (see Fig. 3.8, 3.9, 3.10 and 3.11) that the scrambling degree of a scrambled image for the four significant bit-planes selection is nearly the same as that for the eight bit-planes. Specially, according to the experimental analysis (compared with Fig. 3.8, 3.9 and Figs. 3.10, 3.11), it can be found that the dividing size 32×32 used in the evaluation may produce a better scrambling degree than the size 16×16 used in the evaluation.

3.1.5 Organization of the Chapter

The rest of this chapter is organized as follows. In Section 5.2, the corresponding knowledge of the bit-plane and the principle of this evaluation method are introduced. Section 5.3 introduces an evaluation method based on the spatial distribution entropy and centroid difference of bit-planes. Simulation experiments and analysis about this evaluation method are provided in Section 5.4. Future works and conclusions are drawn in the last section.

3.2 Preliminaries on Bit-Plane of Still Image

3.2.1 Bit-Plane of Gray-Scale Image

Assume that one pixel is located at the position (x, y) of a *gray-scale image*. Let us denote the corresponding value of it by $P(x, y)$ which is the *brightness intensity* of that pixel. As the computer only displays the discrete number, according to its representation precision, the brightness intensity of a pixel is divided into 256 parts and the intensity can take any value in the integer set $\{0, 1, \ldots, 255\}$ for the gray-scale image. As the computer only deals with the binary number, every pixel value is represented by an 8-bit binary stream, such as 127='01111111', i.e., $127=0\times2^7+1\times2^6+1\times2^5+\cdots+1\times2^0$. The example presents the fact that for a general gray-scale image, which has some large texture areas, it can be represented as eight *bit-planes* from the most significant bit-plane (MSB-P) to the least significant bit-plane (LSB-P). This is shown in Fig. 3.2 ('Lena' of

size 128×128).

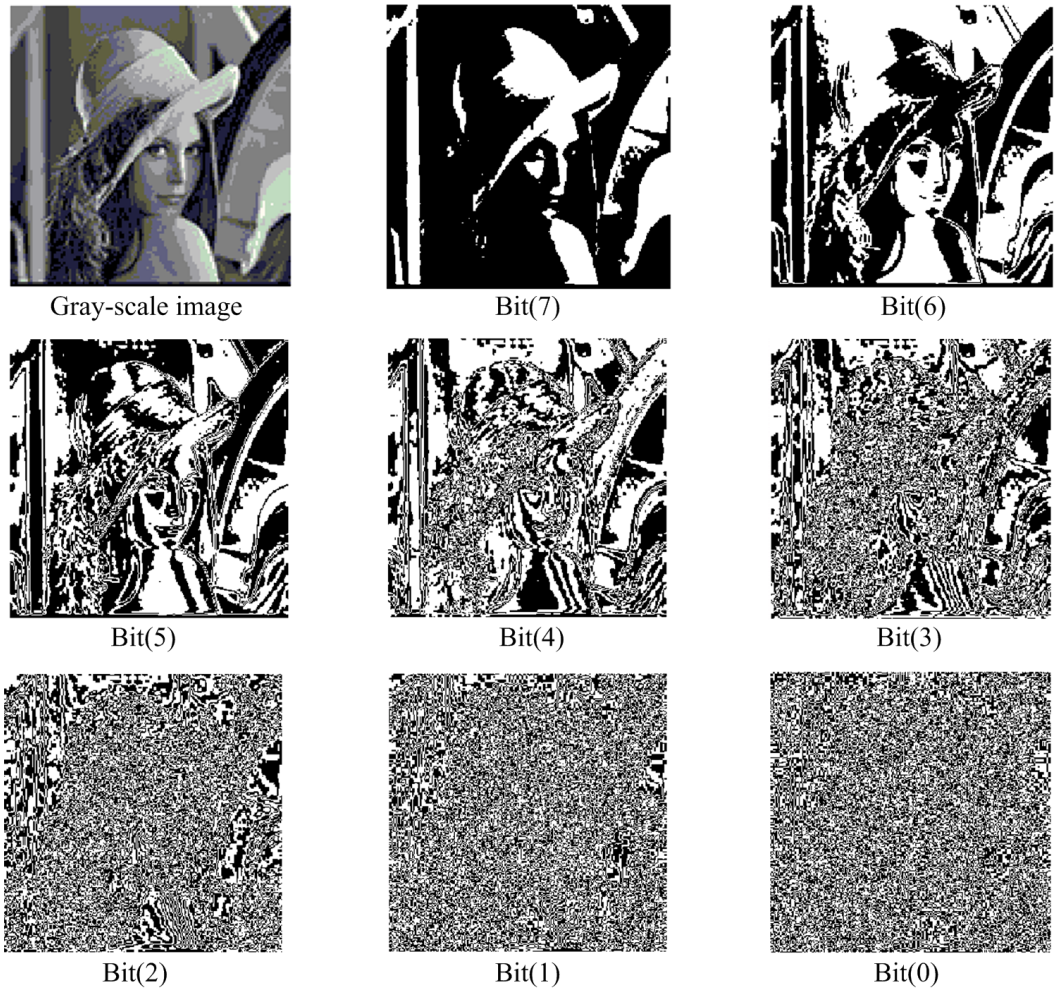

Figure. 3.2: Eight bit-planes of the gray-scale 'Lena': from MSB-P to LSB-P.

From Fig. 3.2, it can be seen that there is a different contribution to such a gray-scale image for each bit-plane. The impact can increase when the bit-plane is from LSB-P to MSB-P. The higher the bit-plane is, the stronger the correlation between the bit-plane and the original gray scale image is. Especially, for this kind of general gray-scale image, we can see that the significant bit-planes portray the outline of an image which reflect the information of the original image. However, the less significant bit-planes look like the *pseudorandomness*.

Another important fact is that for the general gray-scale image which has some large

texture areas, the relationship between two adjacent bit-planes does also increase for the higher bit-planes. In order to test the correlation value between any two bit-planes, Theorem 3.1 is used, and the result is shown in Fig. 3.3.

Theorem 3.1. *Let, for an $M \times N$ gray-scale image, bit_i and $bit_i(x,y)$ denote respectively the ith bit-plane and the pixel value at position (x,y) in the ith bit-plane, $i = 0, 1, \ldots, 7$. Further let X and Y denote respectively the random variables corresponding to bit_i and bit_j, where $i \neq j \in \{0, 1, \ldots, 7\}$. The correlation coefficient $r(\mathrm{X}, \mathrm{Y})$ can be expressed as:*

$$|r(X, Y)| = \left| \frac{p(X=1, Y=1) - p(X=1)p(Y=1)}{\sqrt{p(X=1)p(X=0)p(Y=1)p(Y=0)}} \right|. \tag{3.1}$$

where $p(\cdot)$ denotes the probability.

Proof. Note that X and Y can be described as follows:

$$X = \begin{cases} 1, & bit_i(x,y) = 1 \\ 0, & bit_i(x,y) = 0 \end{cases}, \qquad Y = \begin{cases} 1, & bit_j(x,y) = 1 \\ 0, & bit_j(x,y) = 0 \end{cases},$$

Let $E(X)$, $E(Y)$ and $E(XY)$ denote the expectations of X, Y and the joint distribution of X and Y, respectively. Further let, $D(X)$ and $D(Y)$ are the variances for X and Y, respectively. Note that

$$\begin{aligned}
&E(X) = \tfrac{1}{M\times N} \sum_{k=0}^{M\times N} X_k = p(X=1); \quad E(Y) = \tfrac{1}{M\times N} \sum_{k=0}^{M\times N} Y_k = p(Y=1); \\
&E(XY) = \tfrac{1}{M\times N} \sum_{k=0}^{M\times N} X_k Y_k = p(X=1, Y=1); \\
&D(X) = \tfrac{1}{M\times N} \sum_{k=0}^{M\times N} [X_k - E(X)]^2 \\
&\quad = p(X=0)[p(X=1)]^2 + p(X=1)[p(X=0)]^2 = p(X=1)p(X=0); \\
&D(Y) = \tfrac{1}{M\times N} \sum_{k=0}^{M\times N} [Y_k - E(Y)]^2 \\
&\quad = p(Y=0)[p(Y=1)]^2 + p(Y=1)[p(Y=0)]^2 = p(Y=1)p(Y=0); \\
&\text{Consequently, } |r(X, Y)| = \left| \tfrac{E(XY)-E(X)E(Y)}{\sqrt{D(X)D(Y)}} \right| = \left| \tfrac{p(X=1,Y=1)-p(X=1)p(Y=1)}{\sqrt{p(X=1)p(X=0)p(Y=1)p(Y=0)}} \right|.
\end{aligned}$$

□

Note that using Eq. (3.1), the *correlation coefficient* of any two bit-planes can be calculated. Fig. 3.3 demonstrates that there is a relationship between any two bit-planes (for

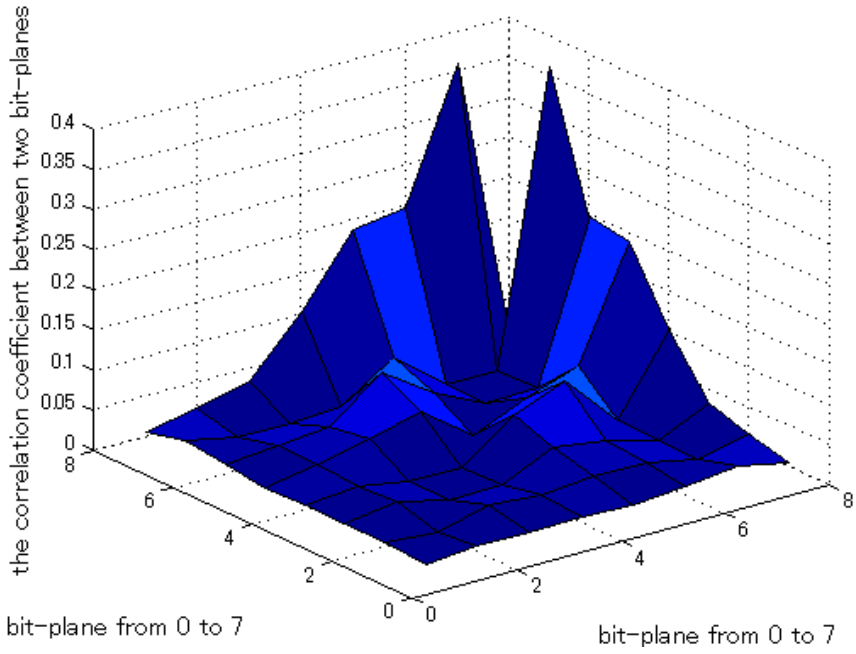

Figure. 3.3: Correlation coefficients between any two bit-planes of the gray-scale 'Lena'.

the convenience, the autocorrelation coefficient is set to 0). Particularly, for a general gray-scale image (e.g., 'Lena'), the correlation coefficient between the 7^{th} bit-plane and the 8^{th} bit-plane is much higher than others. Moreover, we also find that the correlation between the 8^{th} bit-plane and the 6^{th} or the 5^{th} bit-plane is also high. However, for the other pairs of bit-planes, the correlation coefficients are comparatively small and part of them are nearly equal to 0.

Moreover, in order to obtain the relationship between such a gray-scale image and the corresponding bit-planes, the following test is introduced (see Fig. 3.4).

Fig. 3.4 shows that the contribution of each bit-plane for the original image is different. This implies that if the bit-plane is the MSB-P, it has a strong relationship with the original image, and if the bit-plane is the LSB-P, the correlation is extremely small.

According to the above analysis on the bit-plane and the relationship between the original image and bit-plane, Definition 3.1 is used to present a correlation strength (CS) between the original gray-scale image and each bit-plane. It can be seen as a quantified relationship between them.

Definition 3.1. *For an $M \times N$ gray-scale image P, the correlation strength between P and bit(i), $i = 0, 1, \ldots, 7$ is expressed as: $CS(i)=Ibit(i)/255$, $Ibit(i) \in \{2^7, 2^6, 2^5, \ldots, 2^0\}$, where $CS(i)$ is the correlation strength, $Ibit(i)$ is the impact of each bit-plane (bit(i)) to the original gray-scale image. It satisfies that $\sum_{i=0}^{i=7} Ibit(i)=255$.*

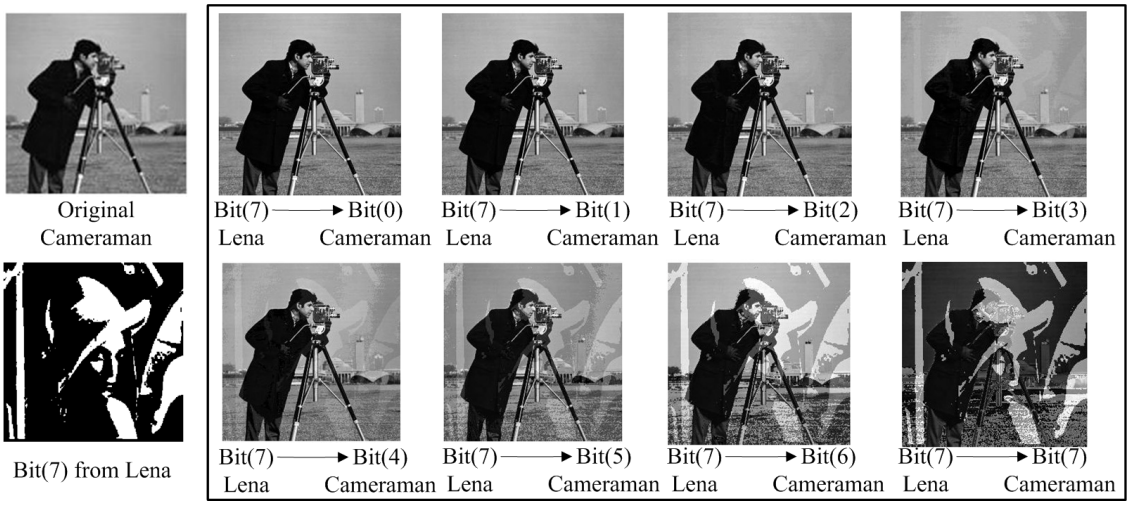

Figure. 3.4: Effect of the bit-plane for gray-scale image.

In particular, the conclusion of Definition 3.1 is based on such a precondition that the contribution of each bit-plane to the original gray-scale image ($Ibit(i)$) is largely dependent on the plane coefficient 2^i, such as $Ibit(7)=1\times2^7+0\times2^6+0\times2^5+\cdots+0\times2^0=128$. This precondition does accord with the test result in Fig. 3.4.

3.2.2 Why Use Bit-plane for Scrambling Evaluation?

According to the analysis and deduction from Subsection 3.2.1, it is evident that the following two important points may be used for demonstrating the reason why the bit-plane can be applied in the scrambling evaluation:

- Each bit-plane has an effect on the original gray-scale image. When the gray-scale image is divided into eight bit-planes, if the operation acts on the eight bit-planes, it can be seen as the processing for the original gray-scale image. Especially, according to Fig. 3.2 and 3.3, the observation and analysis can verify that for some general

gray-scale images, the 8^{th} bit-plane and 7^{th} bit-plane are largely related to the original image, while the last four bit-planes, especially the last two bit-planes, have less relationship with the original image (i.e., look like pseudorandomness). In fact, this property is of great significance for the evaluation which can be shown by Fig. 3.5.

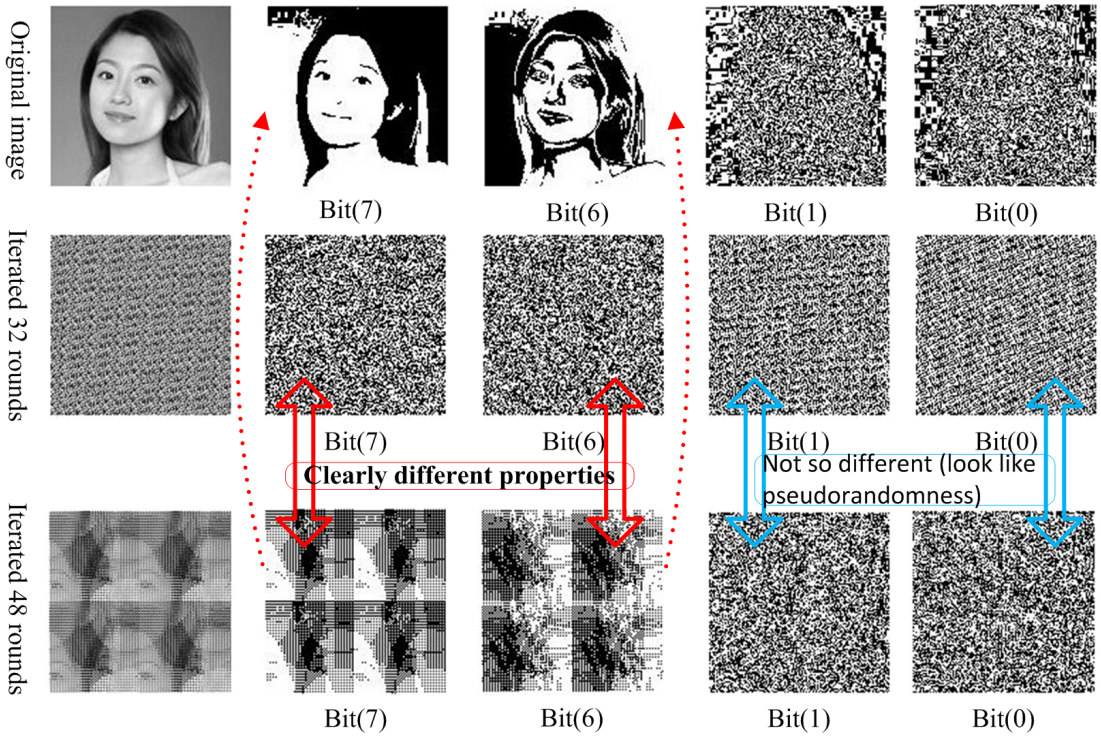

Figure. 3.5: Comparison between bit-planes from two different scrambled images: Original plaintext image is named as 'Internetgirl' of size 128×128.

Moreover, from Fig. 3.3, it can be found that the 6^{th} and 5^{th} bit-planes also have great relationship with the 8^{th} bit-plane. Therefore, for some scrambled gray-scale images, the evaluation method can be constructed according to these four significant bit-planes, which are 50% data of an image, instead of all the eight bit-planes.

- For each bit-plane, there is only 0 or 1 in each position of pixel. It can be seen as a binary image which is simply and easily analyzed by a scrambling evaluation.

However, for a gray-scale image, as the value range of pixel is the integer set {0, 1,. . . , 255}, the computation is not easy which does not facilitate for the scrambling evaluation.

The above two points of view show that the *bit-plane division* can be considered as the technique for the image scrambling evaluation, and help to generate the scrambling degree accurately. Based on this consideration, the scrambling evaluation can be reduced from the gray-scale image to the bit-plane. For each bit-plane, the corresponding evaluation should consider the relationship among pixels deeply. Therefore, the next section introduces two techniques, i.e., the *spatial distribution entropy* and *centroid difference* of each bit-plane, for this purpose.

3.3 Details of Scrambling Evaluation

3.3.1 Spatial Distribution Entropy of Still Image

In 2005, Sun et al. [69] proposed the spatial distribution entropy. For the image U, on the assumption that there are N kinds of pixel values, namely, B_1, B_2, B_3,. . . , B_N, let S_q=$\{(x, y)|(x, y)\in U, P(x, y) = B_q, q\in[1, N]\}$. Then, the centroid of S_q is calculated, and some ring cycles are ensured according to the centroid and radius which are produced by the *segmentation* with an equal distance or unequal distance. Finally, for the pixel value B_q, the spatial distribution entropy can be expressed as:

$$E_q^s = -\sum_{j=1}^{k} p_{qj} \log_2(p_{qj}), \tag{3.2}$$

where s denotes that this entropy is the spatial distribution entropy, k is the number of *ring cycles*, here,p_{qj}=$|\mathrm{R}_{qj}|/|\mathrm{R}_i|$ is the probability density of B_q in ring cycle j, R_i is the number of B_q in S_q, and R_{qj} is the number of B_q in the ring cycle j. The details about spatial distribution entropy are available in the work [69].

For a binary image, as there are only two kinds of pixel values (i.e., 0 and 1), the uncertainty of the pixel value is not quite important in our work. On the contrary, we are interested in the distribution of 0 and 1 in the bit-plane image. Based on this fact,

the spatial distribution entropy is used for evaluating the *scrambling distribution* in our proposal. However, since the pixel distribution of a scrambled image can be regarded as the uniform distribution, and for the convenience of the calculation and post-processing, we can make use of the *average partitioning* which cuts the bit-plane into some *rectangles* (or square) with the same size ($m \times n$) instead of ring cycles for dividing the bit-plane. If the column (or row) cannot be divided evenly by any other integer except 1, the image can add one or several row(s) (or column(s)) with the pixels of the last row (or column).

After the average partitioning is done, the spatial distribution entropy of each small block is calculated by Eq. (3.2). It should be noticed that k is the number of block, p_{qj} is the *probability density* of q ($q \in \{0, 1\}$) in every block. Finally, we can obtain two spatial distribution entropies, i.e., ${E_0}^s$ and ${E_1}^s$ for each bit-plane. To measure the scrambling degree of each bit-plane, we take advantage of the *first moment* of the spatial distribution entropy (Eq. (3.3)) as one part of the scrambling degree, and find that the larger the first moment is, the better the effect of a scrambled bit-plane is.

$$\mu_g = \frac{1}{2} \sum_{q=0}^{1} E_q^s, \qquad g \in \{0, 1, 2, 3, 4, 5, 6, 7\}. \tag{3.3}$$

where μ_g is the corresponding first moment of the g^{th} bit-plane.

3.3.2 Centroid Difference of Bit-Plane

Centroid is a mathematics tool which is used in engineering application field. It can be seen as the average location of a system of a particles distribution, i.e., the center of the quality for an object. In general, the centroid of a finite set of points $M_1(1,1)$, $M_2(1,2)$, $M_3(1,3)$, ..., $M_k(x,y)$ in R^2 is: $C_X = \sum_{i=1}^{k} M_i X_i / \sum_{i=1}^{k} M_i$, $C_Y = \sum_{i=1}^{k} M_i Y_i / \sum_{i=1}^{k} M_i$, where $(C_X,\ C_Y)$ is the corresponding centroid. M_i is the quality in $M_i(x,\ y)$, and $(X_i,\ Y_i)$ is the location. If the quality of this finite set is uniform and the geometry is regular, the centroid is the *geometric center*.

For the bit-plane (generally speaking, the bit-plane is regular), we can assume that each pixel can take the value 0 or 1. We refer the value of the pixel as the 'quality' of the pixel. After the image is scrambled, the location of '1' pixel and '0' pixel of every

bit-plane can be changed and the distribution of them is disordered. According to this knowledge, every bit-plane of an original image can be seen as having 'quality' with many '1' pixels, and the centroid is not located in the geometric center. For a scrambled image, the distribution of '1' pixels in every bit-plane is relatively uniform, which implies that if the centroid of each bit-plane is computed, it should be near to the geometric center in theory. In order to achieve the accurate coordinate of each bit-plane, in our proposal, the centroids of blocks using average partitioning (the same as the partitioning of the spatial distribution entropy) is calculated firstly. Then, they are applied to obtain the final centroid of each bit-plane.

For each bit block, the location of a centroid can be found according to the following formulas: $C_X^{rg}=\sum_{i=1}^{h}X_i/\sum_{i=1}^{h}n_i$, $C_Y^{rg}=\sum_{i=1}^{h}Y_i/\sum_{i=1}^{h}n_i$, where (C_X^{rg}, C_Y^{rg}) is the location of the centroid in each block, n_i=1, h is the number of '1' pixels in the block, r implies that this is the r^{th} block, $g\in\{0, 1, 2, 3, 4, 5, 6, 7\}$ denotes that which bit-plane the centroid belongs to.

For the computer, as the location (x, y) of a pixel in each bit-plane is a *discrete integer* and also for making sure that the coordinate of a calculated centroid is not a decimal, the final centroid should be the nearest integral location, i.e., $(\mathrm{C}_X^{rg})'=round(\mathrm{C}_X^{rg})$, $(\mathrm{C}_Y^{rg})'=round(\mathrm{C}_Y^{rg})$, where $round(\cdot)$ is a function to get the nearest integer.

In order to achieve the final centroid of each bit-plane, the centroids of blocks are used in Eq. (3.4). Especially, all of centroids of blocks have 'quality' which are equal to the amount of '1' pixels in one block.

$$C_X^g = \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i(C_X^{rg})')_r \Big/ \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i)_r; C_Y^g = \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i(C_Y^{rg})')_r \Big/ \sum_{r=1}^{a}(\sum_{i=1}^{h} n_i)_r. \tag{3.4}$$

where a is the number of blocks in a bit-plane. (C_X^g, C_Y^g) is the centroid of the g^{th} bit-plane.

Based on the above preliminaries, the *centroid difference* of a bit-plane can be obtained with Eq. (3.5), which is another part for computing the scrambling degree.

$$diffva^g = \sqrt{(C_X^g - X_c^g)^2 + (C_Y^g - Y_c^g)^2}, \tag{3.5}$$

where $(X^g{}_c,\ Y^g{}_c)$ is the *geometric center*. $diffva^g$ is the centroid difference of the g^{th} bit-plane. Generally speaking, the smaller the value of $diffva^g$ is, the better the effect of a scrambled bit-plane is.

3.3.3 Steps of Scrambling Evaluation

According to the analysis of the property of the bit-plane, it seems that the traditional scrambling evaluation based on the gray-scale pixel is not suitable for evaluating the scrambling of bit-plane. In fact, we should pay attention to the location distribution of pixels which can represent the scrambling degree of the bit-plane. The spatial distribution entropy and centroid difference can achieve this purpose. They can be used to reflect the distribution condition of '0' pixel and '1' pixel in the bit-plane. Specially, since the value of the pixel is only 0 or 1, the computation is not large, which implies that it can be used for practice applications. The steps of the proposed scramble evaluation method are as follows:

- Step 1: Divide the scrambled gray-scale image into eight bit-planes, and take each bit-plane into the evaluation. Specially, taking performance into account, for the general image, only the first four bit-planes are sufficient for our proposed evaluation method. The evaluated bit-planes belong to the set {Bit(8), Bit(7), Bit(6), Bit5}.

- Step 2: Calculate the spatial distribution entropy and centroid difference of each bit-plane according to the methods of Sections 5.3.1 and 5.3.2. For each bit-plane, μ_g and $diffva^g$ can be obtained respectively using Eqs. (3.3) and (3.5).

- Step 3: Evaluate the bit-plane with the scrambling degree. As the spatial distribution entropy is nearly in the direct proportion to the scrambling degree, and the centroid difference is the opposite, the value of the *scrambling degree* of each bit-plane is determined by following Eq. (3.6) which also considers the normalization.

$$scraval^g = \frac{\mu_g \cdot \sqrt{(X_c^g)^2 + (Y_c^g)^2}}{diffva^g \cdot \log 2(2a)}; \qquad g \in \{0, 1, 2, 3, 4, 5, 6, 7\}, \tag{3.6}$$

where $scraval^g$ is the scrambling degree of one bit-plane. a is the number of blocks, $({X^g}_c,\ {Y^g}_c)$ is corresponding geometric center.

- Step 4: Achieve the final value of the scrambling degree for the gray-scale image. As there may be a different impact from each bit-plane to the original gray-scale image, the *scrambling degree* (*scraderee*) of a gray-scale image should be the weighted sum of all the eight (or four) bit-planes, i.e., $scraderee{=}\sum_{g=0/4}^{7} w(g){\cdot}scraval^g$, $g{\in}\{0, 1, 2, 3, 4, 5, 6, 7\}$.

- Step 5: Divide the value of the scrambling degree *scraderee* by the size of the gray-scale image to get the final result: $Fscraderee{\doteq}scraderee/(M{\times}N)$. The purpose of this step is to remove the impact of the size of the gray-scale image for the final scrambling degree.

The above steps from Step 1 to Step 5 are the process of the proposed scrambling evaluation method. From these steps, it can be found that the proposal tries to consider the scrambling evaluation from a new side, i.e., the analysis of the bit-plane which has the pseudorandom distribution. Note that the weight ($w(g)$) in Step 4 is significant for achieving the final evaluation degree. Therefore, two kinds of weights for eight and four bit-planes are used in our proposal:

- As the most scrambling algorithms carry out the scrambling based on the gray-scale pixel, there is a different effect from each bit-plane. If eight bit-planes are used in this evaluation, the *weight* ($w(g)$) of each bit-plane (bit(i)) is the corresponding CS(i) which is defined in Section 3.2.1.

- If only four bit-planes are applied in the proposed evaluation method, this implies that the first four bit-planes have stronger relationship than the last four bit-planes to the original gray-scale image. The corresponding weights of the first four bit-planes are self-adaptive. This implies that the weight is decided by the correlation

coefficient $|r(X,Y)|$ defined in Section 3.2.1. The details are described in Eq. (3.7).

$$\begin{cases} w(7)+w(6)+w(5)+w(4)=1 \\ w(6)=|r(6,7)|\times w(7) \\ w(5)=|r(5,7)|\times w(7) \\ w(4)=|r(4,7)|\times w(7) \end{cases}, \tag{3.7}$$

where $w(g)$, $g\in\{7,6,5,4\}$ is the weight in Step 4, which is also the correlation measurement between the bit-plane and original gray-scale image.

3.4 Experiments and Analyses

3.4.1 Scrambling Strategy

Some digital image scrambling algorithms are referred in Section 3.1.2. Specially, *Arnold cat map* [61] and *generalized Gray code* [95], which are simple and the widely used, are utilized to test the proposed scrambling evaluation method. As the pixels of the image are the discrete number, the generalized versions of two transformations are introduced.

In particular, the binary-scale generalized Gray Code here is used to scramble the coordinates of pixels. The corresponding matrix of the generalized Gray code comes from the work [95], which is an 8×8 matrix. For the consistency, the tests in this chapter use the same control parameters for each image when scrambled by the above two maps. i.e., for the generalized Arnold cat map, the corresponding transformation matrix is the same as the matrix A of Eq. (3.5) in the work [61]. Fig. 3.6 lists the scrambling results of the above two scrambling transformations with the different number of rounds of the iterations (128×128 standard gray-scale 'Baboon').

From the observation of Fig. 3.6, some important information can be explored. It can be found that 96 and 8 are the periods of the *generalized Arnold cat map* and generalized Gray code (size: 128×128), respectively. When an image is scrambled by iterated some rounds, the visual effect of the scrambled image may not be 'good'. Moreover, for the transformation with a large period, when it is iterated half rounds of the period,

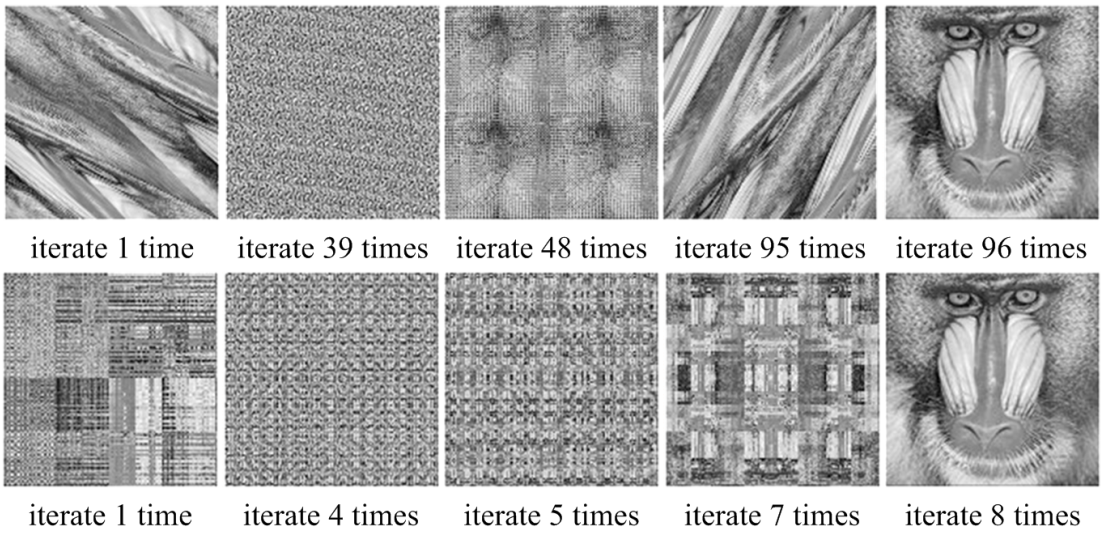

Figure. 3.6: Scrambled gray scale 'Baboon' using two kinds of transformations in different times of iterations: results of first row is from generalized Arnold cat map; results of second row is from generalized Gray code.

there is a "half-period phenomenon" in the corresponding ciphertext image, which may leak the information of the original image (i.e., the visual leakage). However, for the transformation with a short *period*, it seems that this phenomenon does not happen.

3.4.2 Scrambling Measurement

In order to test the effectiveness of the proposed scrambling evaluation method, two standard gray-scale images ('Baboon' and 'Boat') and two general images (we call them 'Internetgirl' and 'Landscape') of size 128×128 are chosen which are scrambled by the above generalized transformations (see Fig. 3.7). In order to make a comparison about the size of the block dividing each bit-plane, 16×16 and 32×32 are set as this size in our tests. The weights for eight bit-planes and four bit-planes, in these tests, are produced by the methods in Section 5.3.3.

Fig. 3.8, 3.9, 3.10 and 3.11 are used to show the scrambling degree values of two standard images and two general images scrambled by the above two transformations within one period and four periods, respectively (Fig. 3.8 and 3.9: the size of the block

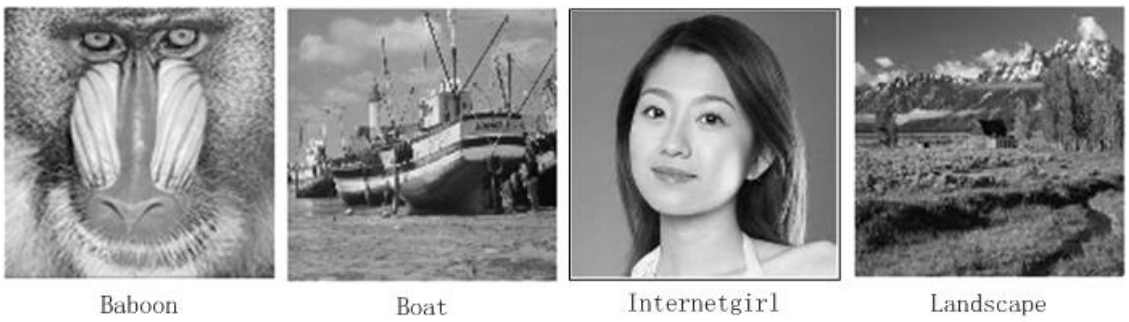

Figure. 3.7: Original gray-scale images ('Baboon', 'Boat', 'Internetgirl', 'Landscape').

is 16×16; Fig. 3.10 and 3.11: the size of the block is 32×32). From these figures, the following facts come out: it is easily found that the proposed scrambling evaluation method basically reflects the effect of the scrambled images, which can be also used to reveal the performance of the corresponding scrambling method. Specially, considering the transformation with a long period, the iteration round of the obvious "half-period phenomenon" can be appeared according to this proposed method. However, the generalized Gray code, which is the map with the short period, is not suitable for this rule. This is also shown by the proposed method. Based on the above facts, it is testified that compared with the results of evaluation methods in the previous research [49, 87], our proposal can reflect the scrambling effect precisely.

It can also be found that for these general gray-scale images, the result from the four bit-planes selection is similar to that of the eight bit-planes selection. The reason is that the last four bit-planes have a less impact than the first four bit-planes on the original gray-scale image. Therefore, if a gray-scale image (e.g., Fig. 3.7) is scrambled by a scrambling algorithm, we can make use of the first four bit-planes instead of all the eight bit-planes to evaluate the scrambling degree. Specially, according to the comparison between Fig. 3.9 (and Fig. 3.8) and Fig. 3.11 (and Fig. 3.10), if the size of the block is 32×32, the evaluation results are intact and better than those of the size 16×16. In Fig. 3.9, some evaluation results are set to 0. This is based on the fact that when the size 16×16 is used, one or more block(s) in some bit-planes may be full of 0 or 1. However, this may not reflect the practical situation that the scrambled image has a

'bad' scrambling effect instead of no scrambling effect. Therefore, from this point, the size 32×32 is seen to be more suitable for the proposed evaluation method.

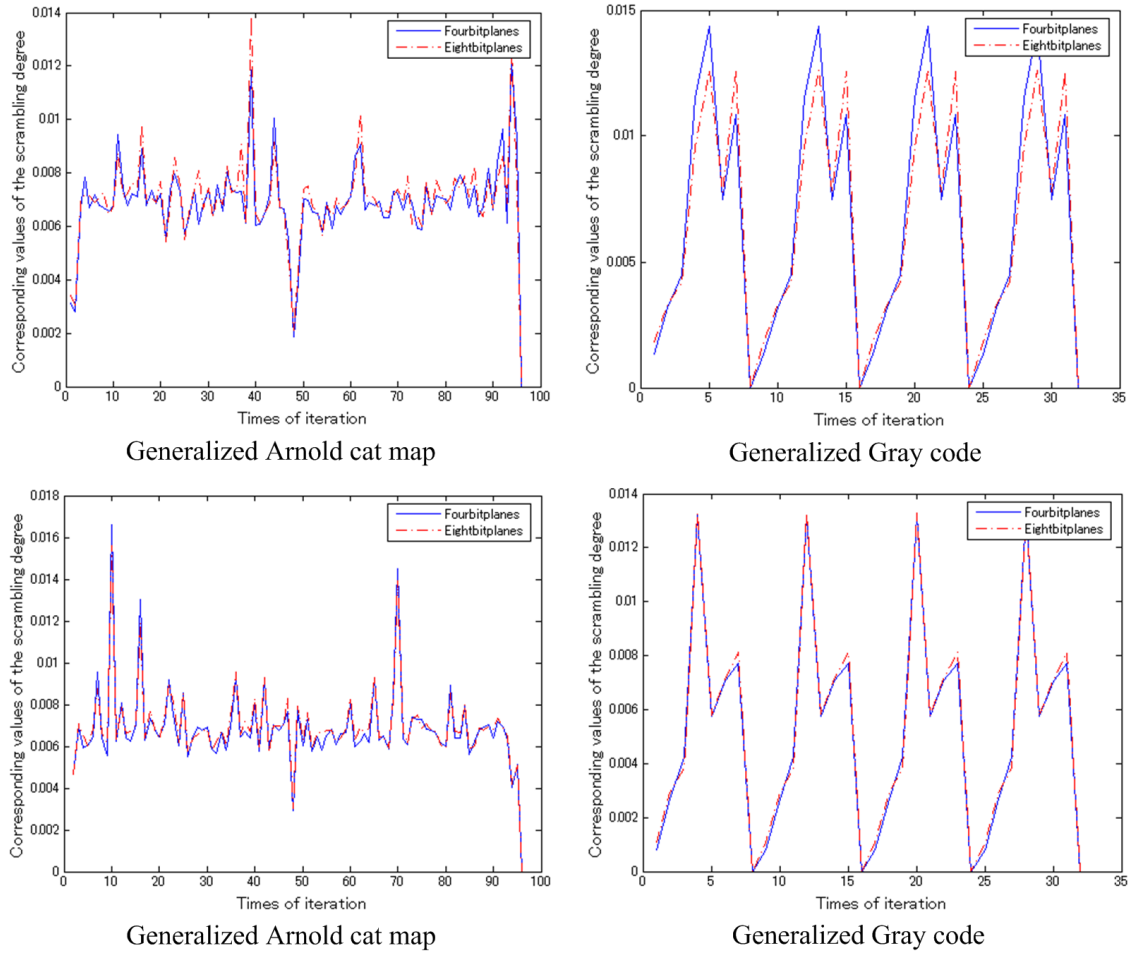

Figure. 3.8: Scrambling degree values for the used two transformations (Size of block is 16×16): first row is corresponding results of 'Baboon'; second row is corresponding results of 'Boat'. (blue: four bit-planes selection; red: eight bit-planes selection)

Moreover, according to Fig. 3.8, 3.9, 3.10 and 3.11, another merit of the proposed method can be achieved, i.e., the highest scrambling degree and the lowest scrambling degree can be calculated and shown obviously. This merit can be used to analyze the optimal and 'weak' iteration round for the encryption and watermarking. Specially, if the security of a combination cryptosystem/watermarking algorithm is considered, the proposed evaluation method is suitable for analyzing the first step of this cryptosys-

tem/watermarking algorithm, i.e., the scrambling step. In fact, according to the scrambling results shown in Fig. 3.8, 3.9, 3.10 and 3.11, these highest and lowest scrambling degrees, which can be also concluded from Fig. 3.6, are in accordance with the visual observation of human.

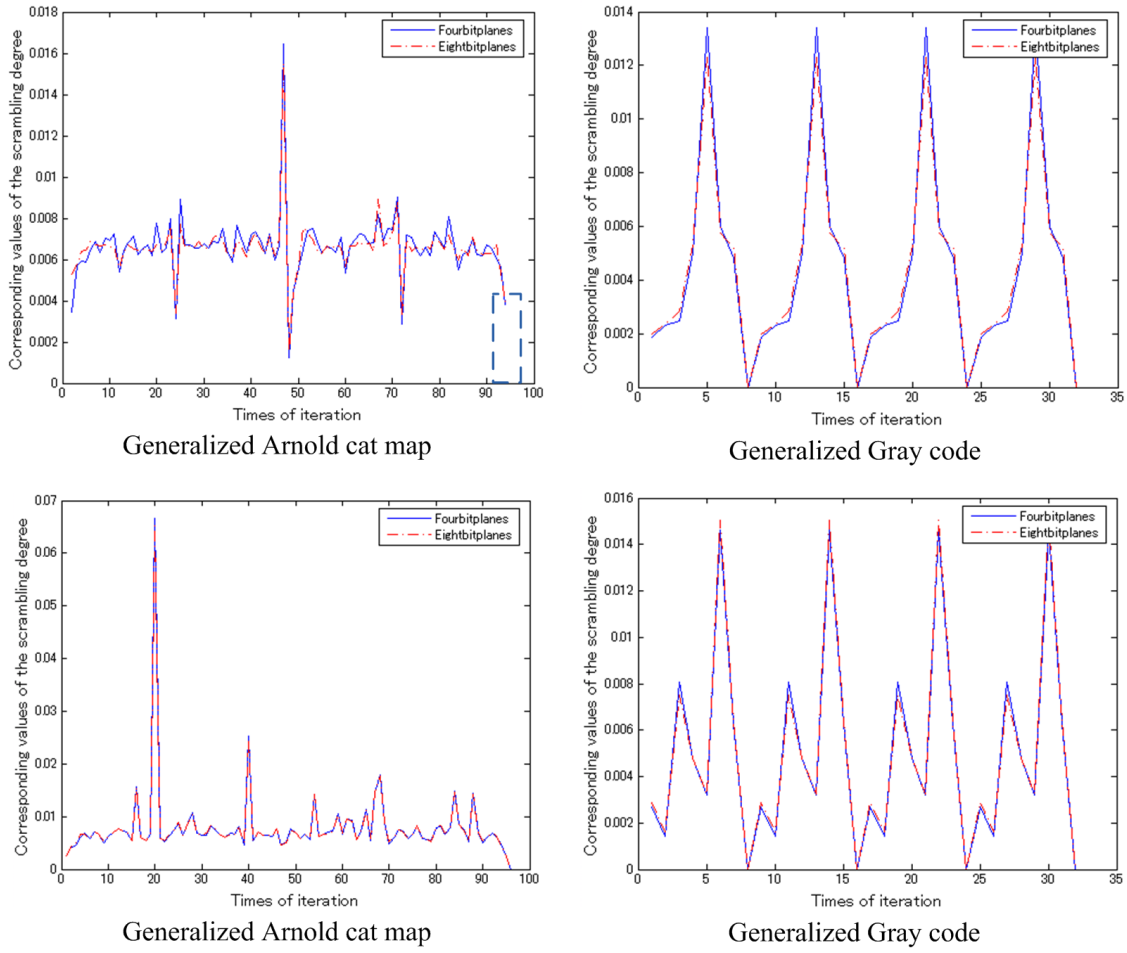

Figure. 3.9: Scrambling degree values for the used two transformations (Size of block is 16×16): first row is corresponding results of 'Internetgirl'; second row is corresponding results of 'Landscape'. (blue: four bit-planes selection; red: eight bit-planes selection)

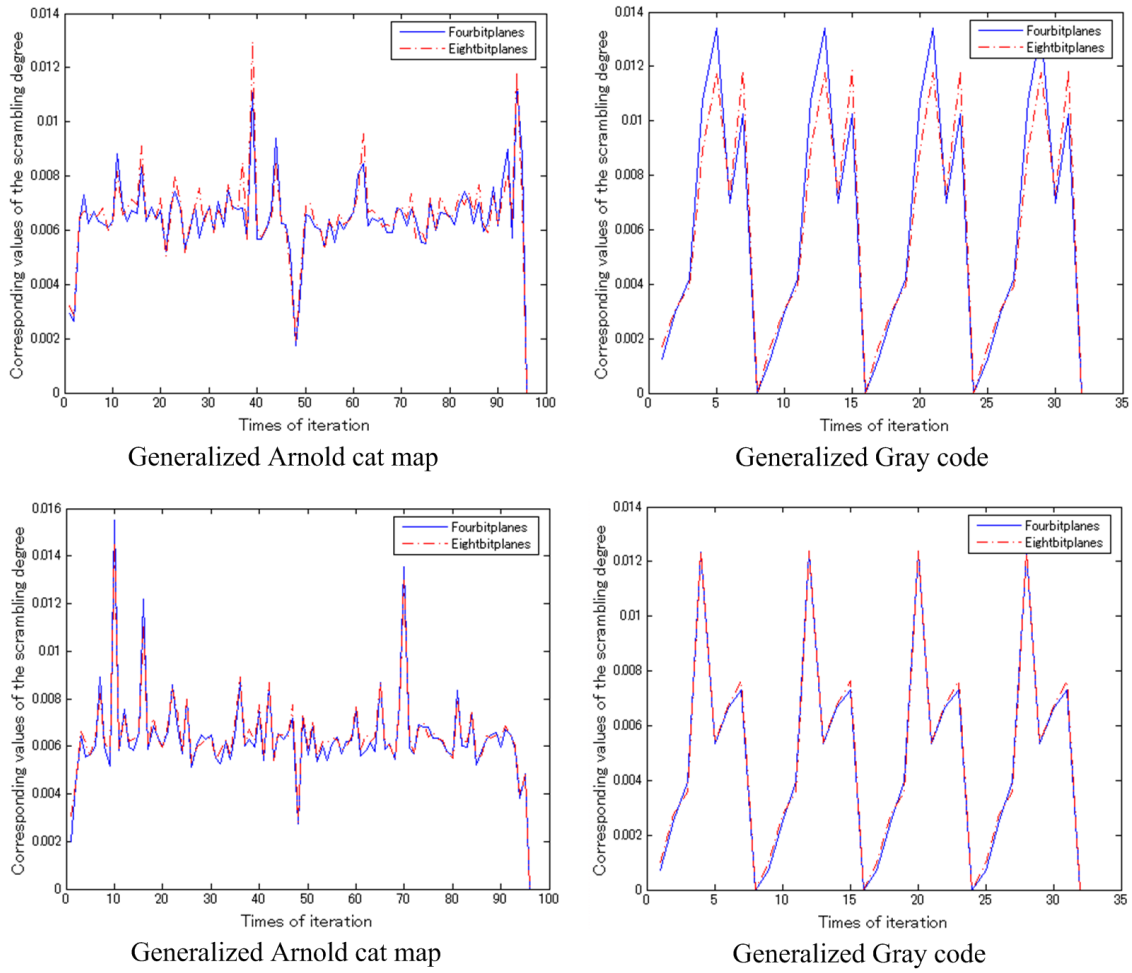

Figure. 3.10: Scrambling degree values for the used two transformations (Size of block is 32×32): first row is corresponding results of 'Baboon'; second row is corresponding results of 'Boat'. (blue: four bit-planes selection; red: eight bit-planes selection)

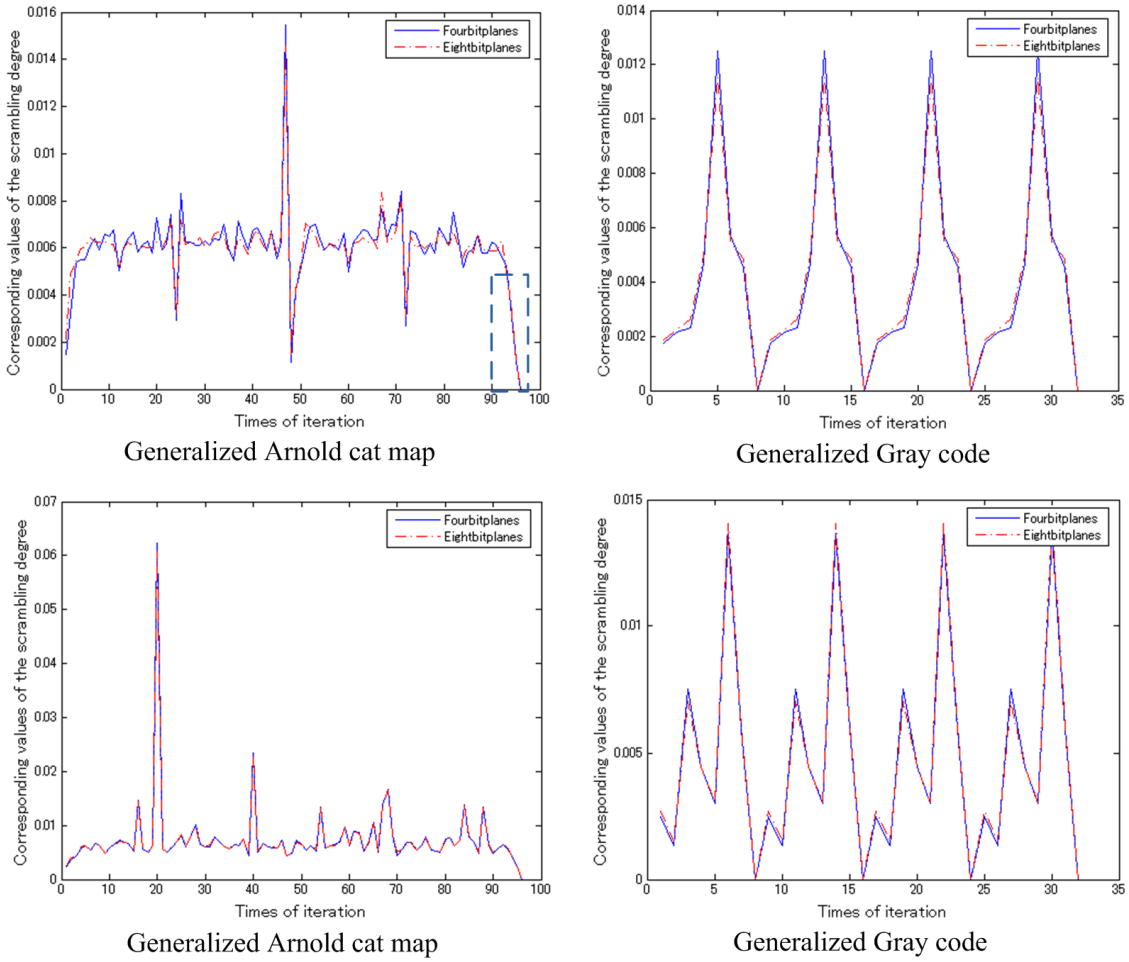

Figure. 3.11: Scrambling degree values for the used two transformations (Size of block is 32×32): first row is corresponding results of 'Internetgirl'; second row is corresponding results of 'Landscape'. (blue: four bit-planes selection; red: eight bit-planes selection)

3.5 Conclusions and Further Discussion

In this chapter, a scrambling evaluation method and the corresponding steps was proposed to measure and scrambling and reveal the weakness of the scrambling method. Based on our analysis, the bit-plane dividing for the gray-scale image can be used as the first step of the proposed evaluation method. The spatial distribution entropy and centroid difference were provided for measuring the scrambling degree of each bit-plane. The final result of the scrambling evaluation is the weighted sum of the eight bit-planes or the first four bit-planes. The experimental results demonstrate that for some gray-scale images, not only the eight bit-planes, but also the first four bit-planes can be utilized in the proposed evaluation method for reflecting the merit and weakness of the corresponding scrambling algorithm effectively. However, for some special images such as the *fingerprint* image, the proposal based on the first four bit-planes may not work well. This is based on the fact that all the eight bit-planes of those kinds of images (e.g., fingerprint image) have the relationship with the original gray-scale image. In fact, Engel et al. [18] has confirmed about this point of view. Therefore, when the scrambled images of the fingerprint images are evaluated, all the eight bit-planes of the scrambled images need to be considered. This implies that the corresponding computation is hard to be reduced.

Moreover, if the scrambling algorithm is used to encrypt the *RGB* color image, the corresponding scrambling degree can also be evaluated. The RGB color image is usually regarded as the integration of three primary *color-components*, i.e., Red, Green and Blue. As there are 256 possible values or intensities based on 8 bits for each color-component, the proposal for the gray-scale image can be used to evaluate each color-component of the RGB color image. Specially, as the three color-components have a different correlation with the original color image, the final scrambling degree of the color image should be the *linear combination* of the three color-components. It can be defined as follow:

$$Fscraderee_{\mathsf{color}} = 0.301 \times Fscraderee_{\mathsf{R}} + 0.586 \times Fscraderee_{\mathsf{G}} + 0.113 \times Fscraderee_{\mathsf{B}},$$

where 0.301, 0.586 and 0.113 are the corresponding weights [41], $Fscraderee_{\mathsf{color}}$ is the

scrambling degree of the *color image*, $Fscraderee_{\mathsf{R}}$, $Fscraderee_{\mathsf{G}}$ and $Fscraderee_{\mathsf{B}}$ are the scrambling degree of Red, Green and Blue components, respectively. It can be found that after the scrambling degree of these three components are achieved, the scrambling degree of the corresponding color image can be also computed.

Chapter 4

Security Analysis of Image Encryption Algorithm of Pixel Bits

In this chapter, we introduce an image encryption algorithm of pixel bits, and present a corresponding security analysis on this encryption algorithm. The complexity of the proposed attack is compared with the previous work for demonstrating the efficiency of the proposed attack. Moreover, the suggestion on the improvement is provided.

4.1 Introduction

4.1.1 Research Background

Security protection techniques of digital image, e.g., the image encryption [55, 88, 61, 39, 16, 26, 37, 71, 34], the image authentication [58, 72, 20] and image hash [57, 80, 70], are of significance for protecting the content of image in the communications/storage environment. Specially, the image encryption is a general method for this protection, which has been demonstrated in Chapter 2. In Chapter 3, we have discussed about the scrambling evaluation for the image spatial scrambling algorithm. However, recently there are some image encryption methods which can scramble the pixel positions and encrypt the pixel values, e.g., [26, 37, 71]. In 2004, Chen et al. [26] proposed a generalized three-dimensional (3D) Arnold cat map, and used it in a symmetrical image encryption algorithm based on *chaos*. Guan et al. [37] introduced a fast image encryption design

according to the character of *hyper-chaos*. Tong et al. [71] designed a new compound two-dimensional chaotic function, and a novel image encryption scheme is introduced, which is based on this new compound chaos. For this kind of image encryption, of course we may discuss about the spatial scrambling of pixels. However, as there is the encryption of pixels' value, the importance is that we can recover the original plaintext image/part of original plaintext image or find out some information about the secret key according to the ciphertext image. This is what we have defined in Section 2.4 of Chapter 2. In fact, some image encryption algorithms are not secure enough and have been attacked, e.g., [25, 75, 30, 82, 64, 65]. The typical attacks, e.g., chosen-plaintext attack (CPA) and know-plaintext attack (KPA), are usually effective to analyze the image encryption methods (e.g., [30, 82, 63]). The more discussions about security of image that readers can obtain are based on the recent surveys [47, 33, 74]. Moreover, several useful rules about how to evaluate the security of encryption algorithms based on chaos are presented in the work [21]. A quantitative cryptanalysis on the performance of permutation-only multimedia ciphers against plaintext-based attacks is performed in the work [48].

4.1.2 Our Contribution

Recently, Ye [85] proposed one kind of image encryption algorithm for shuffling all the bits of pixels by using one-dimensional chaos system. The proposed encryption method mainly possesses two characteristics, i.e., firstly, the scrambling of pixels is not considered from gray level {0, 1,. . . , 255} but from bit-plane level {0, 1}. Secondly, the encryption process that only the row and column are shuffled is used to encrypt the value of the pixel as well as the position of the pixel simultaneously.

In 2011, Li and Lo [45] presented one kind of known-plaintext attack (KPA) and chosen-plaintext attack (CPA) to the original image encryption algorithm, i.e., Ye's scheme. The plain image which has some noise points is recovered if more than $\lceil log_2(8MN) \rceil$ known-plaintext images of size $M \times N$ are used. Whereas, in the CPA, the requisite number of chosen plaintext images for recovering the exact plaintext image is at least $3+\lceil log_2(MN) \rceil$. The ideas of both attacks are nearly the same, which involve construct-

ing a multi-branch tree for recovering the permutation matrix W. This chapter studies the security issues of the original image encryption algorithm [85] in details. According to our analyses, a particular type of CPA/CCA is used to analyze the original encryption algorithm. For the CPA (assume that the secret key is fixed), two kinds of plaintext images are used to recover the equivalent vectors TM and TN which are seen as encryption keys. As a result, after these vectors are revealed, the original plaintext image can be acquired easily, which results the breakage of the algorithm. Specially, the number of chosen images is not only decided by the sizes of M and N, but also determined by the ratio between M and N. This is different from the attack method of Li and Lo [45]. Therefore, in our proposal, a ciphertext image of size 128×128 and one of size 256×256 can be recovered with the same number of chosen images. For the CCA, the same method is suitable for achieving the decryption vectors $TM^{'}$ and $TN^{'}$, which are the inverse vectors of TM and TN, respectively. To present the merit of our attack, the computational complexity is compared between our attack and the attack proposed by Li and Lo [45]. The analysis illustrates that our attack is more efficient than the attack proposed by Li and Lo. As the original image encryption algorithm has some merits towards applications (e.g., the parallel encryption for the position and value of the pixel), we present the suggestion on the improvement of the original encryption algorithm which is based on the idea of the "self-correlation" encryption. This suggested encryption makes use of the property of the plaintext image to encrypt itself. Later, we perform some simulation experiments to evaluate this improvement. The simulation results demonstrate that our suggested improvement has the good performance to accommodate practical applications.

4.1.3 Organization of the Chapter

The outline of this chapter is as follow. In Section 4.2, the brief introduction of the original scrambling encryption algorithm is presented. Section 4.3 indicates the drawbacks of the original encryption algorithm and demonstrates how to implement the CPA/CCA. In Section 4.4, some simulation examples are used to illustrate our attack in

details. The comparison between our attack and Li and Lo's attack is drawn in Section 4.6. Section 4.7 introduces the suggestion on the improvement which is based on the 'self-correlation' method, and Section 4.8 presents the corresponding experiments and analyses results about the suggested improvement. The concluding remarks are drawn in the last section.

4.2 Description of the Original Algorithm Under Study

The details of original image scrambling algorithm [85] can be described as follows, which are also shown in Fig. 4.1:

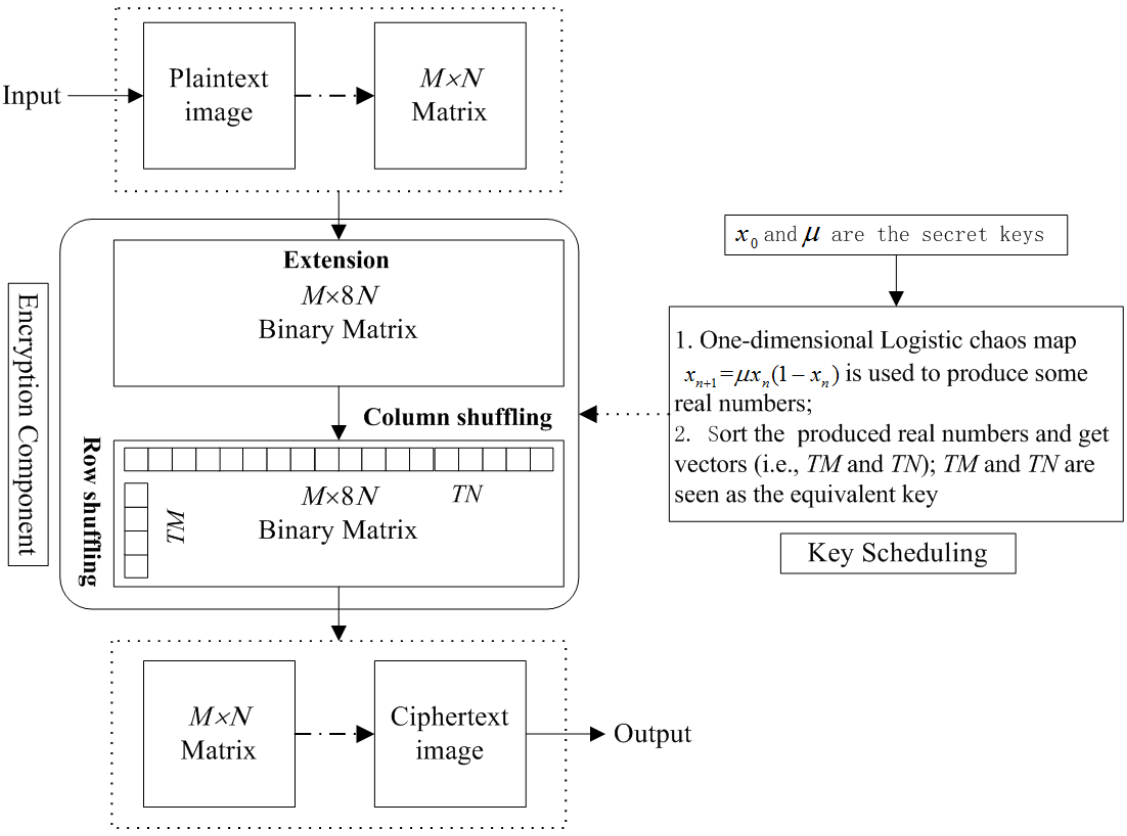

Figure. 4.1: Details of encryption process.

- Step 1: An original gray-scale image of size $M \times N$ can be considered as an $M \times N$ matrix, say P, with entries, denoting the pixel values, chosen from the set $\{0, 1, 2, \ldots,$

255}.

- Step 2: Every pixel value in the decimal *matrix P* should be transformed to an 8-bit sequence. Then, these bit sequences are connected and integrated into a $M\times 8N$ binary matrix P^t. Especially, the gray pixels can be transformed into bit pixels according to the following Eq.(4.1) [85]:

$$p^t(i,j) = \begin{cases} 1 & if\ (P(i,j)/2^t) \bmod 2 = 1 \\ 0 & others \end{cases} \tag{4.1}$$

in which $P(i,j)$ denotes the *pixel value* in the i^{th} row and j^{th} column of the original image, and $p^t(i,j)$ denotes binary number, $i\in\{1, 2, \ldots, M\}$, $j\in\{1, 2, \ldots, N\}$ and $t\in\{0, 1, 2, 3, 4, 5, 6, 7\}$.

- Step 3: The one-dimensional *Logistic chaos map* $x_{n+1}=\mu x_n(1-x_n)$ is used for producing the scrambling vectors *TM* and *TN*, where μ is the *system parameter*, x_n is the iteration value. Specifically, at first, two iteration values sequences $\{x_{m+1}, x_{m+2}, x_{m+3},\ldots, x_{m+M}\}$ and $\{x_{n+1}, x_{n+2}, x_{n+3},\ldots, x_{n+N}\}$ are generated by the Logistic chaos map. After that, the sequences are sorted for obtaining the positions of the values, and these transform positions are marked down by $TM=(t_1, t_2,\ldots, t_{M)}$ and $TN=(t'_1, t'_2,\ldots, t'_N)$, respectively.

- Step 4: The binary matrix P^t is encrypted by the scrambling vectors *TM* and *TN*, i.e., the row and column of P^t are shuffled according to the elements of TM and TN, separately. As the scrambling vectors TM and TN can be described by two elementary transformation matrices TM^{TR} and TN^{TR}, respectively, the cipher matrix D can be achieved by $D=[TM^{TR}\times P^t\times TN^{TR}]_{M,8N}$, i.e., the elementary transformation of P^t is equivalent to the fact that P^t is multiplied left by TM^{TR} and TN^{TR}.

- Step 5: The decimal matrix $C=[P(x, y)]_{M,N}$ is acquired when the matrix D is recovered by Eq.(4.2):

$$C(i,j) = \sum_{t=0}^{7} 2^t \times P^t(i,j), \tag{4.2}$$

- Step 6: This scrambling encryption process is completed, and the decimal cipher matrix C is the corresponding *ciphertext image*.

For the decryption process, the ciphertext image is transformed back to the plaintext image according to the *inverse vectors* (i.e., TM' and TN') of scrambling vectors TM and TN. The whole process is almost the same as the encryption except the used scrambling vectors. Therefore, this image encryption algorithm can be seen as a reversible encryption algorithm. Fig. 4.2 is a simulation result about the original encryption algorithm (gray-scale image "Baboon" of size 128×128, the secret keys and parameters are x_0=0.2009, μ=3.98, m=20, n=51). The corresponding *histogram* about the ciphertext image is also presented. For obtaining more details, the readers can refer to the work [85].

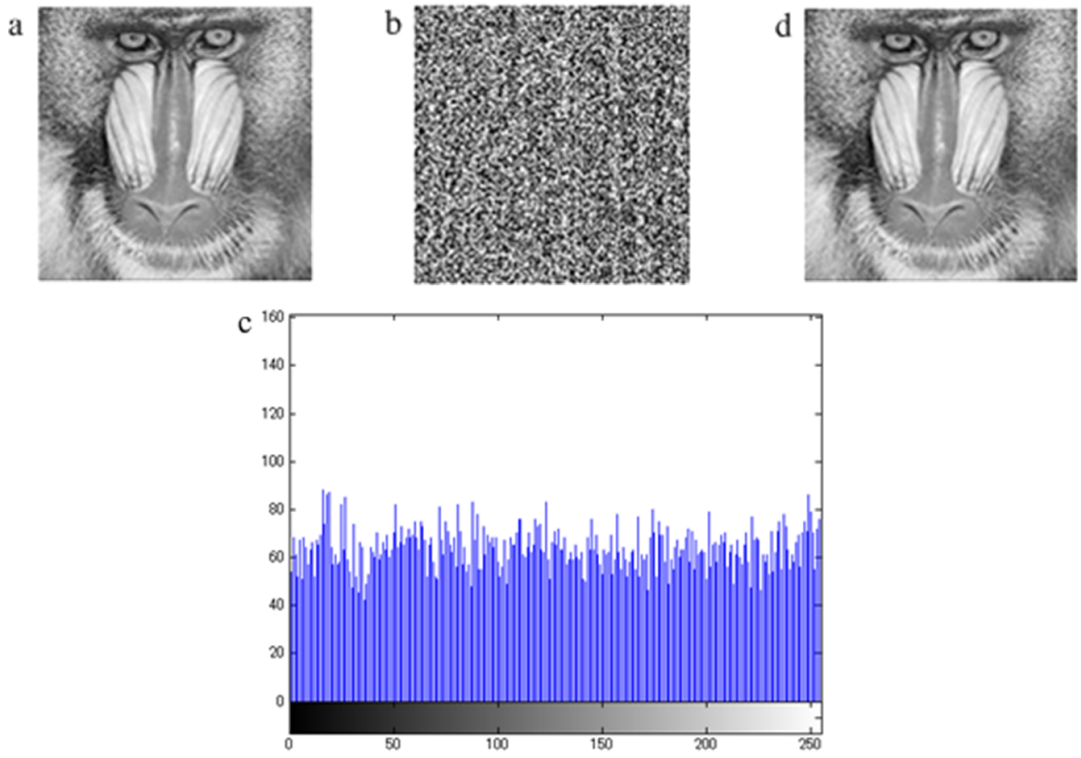

Figure. 4.2: Encryption and decryption effect on original scheme.

Although the author has provided some security analyses about the original encryption algorithm (e.g., the key sensitivity analysis and correlation analysis of pixels), it is still necessary for the researchers to explore the corresponding security. According to the

introduction of the attack scenarios in the Subsection 2.5, in this research, we assume that the secret key is fixed and the adversary can choose the plaintext images/ciphertext images and obtain the corresponding results which implies that the CPA/CCA are used for the security analysis. Then, the *equivalent key* are revealed. Specially, the definition of CPA and CCA are nearly the same as those in Subsection 2.5. Note that when the plaintext images/ciphertext images with the specific structure, which consider the special properties of the encryption, are constructed, the attack performance can be optimized.

4.3 Drawbacks in the Original Algorithm and Corresponding Attack

4.3.1 Drawbacks of the Original Algorithm

The original encryption algorithm is the bit-plane based encryption. It carries out the scrambling of the locations of pixels and the encryption of values of pixels simultaneously. However, three potential drawbacks are presented in the original algorithm, and consequently have a significant effect on the security of the ciphertext image:

- According to the description of the original algorithm, for every bit in the binary matrix P^t, it only implements the row and column exchange respectively (Fig. 4.3), which implies that the location $\{(x,\ y)|x{=}i,\ i{\in}\{1,\ \ldots,\ M\};\ y{=}k,\ k{\in}\{1,\ \ldots,\ 8N\}\}$ of every original-bit in each column maps to the location $\{(x^*,\ y)|x^*{=}j,\ j{\in}\{1,\ \ldots,\ M\};\ y{=}k,\ k{\in}\{1,\ \ldots,\ 8N\}\}$ firstly, and then to the location $\{(x^*,\ y^*)|x^*{=}j,\ j{\in}\{1,\ \ldots,\ M\};\ y^*{=}e,\ e{\in}\{1,\ \ldots,\ 8N\}\}$. Therefore, the transformation range of a bit is confined into a narrow space which consists of one row and one column. Specially, for each gray-scale pixel, the original scrambling algorithm is not perfect as the row transformation (i.e., the transformation using TM) only finishes the column scrambling of the pixel location, and the column transformation (i.e., the transformation using TN) can be seen as the gray encryption in one row.

- The second obvious problem is that the position exchange of every bit only depends

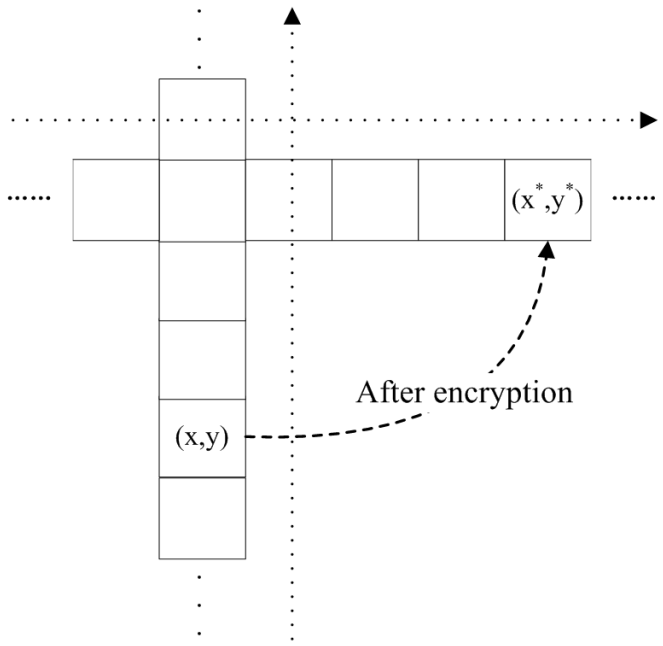

Figure. 4.3: Transformation of a bit pixel in one binary matrix.

on the generated keys, but is not related to other information of the plaintext image (e.g., the pixel value). From the steps of the original algorithm, the process of the bit pixel encryption is only decided by the *equivalent keys* (i.e., the scrambling vectors TM and TN): C=$\mathrm{E}_{Ye}(I, TM, TN)$, which demonstrates that if TM and TN have been found, the rule for scrambling the image ($\mathrm{E}_{Ye}(\cdot)$) is also broken, and the *plaintext image* I is easily obtained.

- In the original algorithm, for decreasing the computational complexity, only two equivalent keys (i.e., TM and TN) are used to encrypt each row and column of the corresponding binary matrix P^t of size M×$8N$. According to the encryption process, firstly, the P^t is encrypted by $TM: \ TM \times P^t(i,:)_M \rightarrow (P^t)'(i,:)_M$. For each bit in one row i of $(P^t)'(i,:)_M$, the exchange rule is the same. Then $(P^t)'$ is encrypted by $TN: \ (P^t)'(:,j)_{8N} \times TN \rightarrow \mathcal{P}(:,j)_{8N}$, in which $\mathcal{P}$ is the cipher binary matrix. For each bit in one column j of $\mathcal{P}(:,j)_{8N}$, the exchange rule is also the same. On one hand, this method can enhance the speed assuredly as only two vectors are used in the whole process. On the other hand, this can also let down

the security of the scrambling encryption as when the rule of one row (or column) is cracked, the exchange method of the whole columns (or rows) is found.

4.3.2 Attack Against the Original Algorithm Under Study

The analyses of Section 4.3.1 show that the original algorithm has distinct drawbacks which can be utilized by the adversary. Specially, from the second drawback, it is found that if the scrambling vectors (i.e., TM and TN), which can be seen as the equivalent keys, have been revealed, the ciphertext image C can be decrypted apparently. This implies that the used one-dimensional chaos system can be isolated. The first and third drawbacks present the fact that since the x-coordinate and y-coordinate of each pixel bit are encrypted independently, and the same encryption rule is carried out, there may exist some special images for searching these equivalent keys in the original image encryption scheme. Therefore, our purpose is to find some particular and simple plaintext images for revealing TM and TN. Usually, this is called the chosen-plaintext attack (CPA). As there are two scrambling vectors (i.e., TM and TN), for each one, there should be at least one special plaintext image for acquiring it. Fortunately, such plaintext images do exist, and can finish the attack efficiently (e.g., Fig. 4.4). The process of the CPA is as follows:

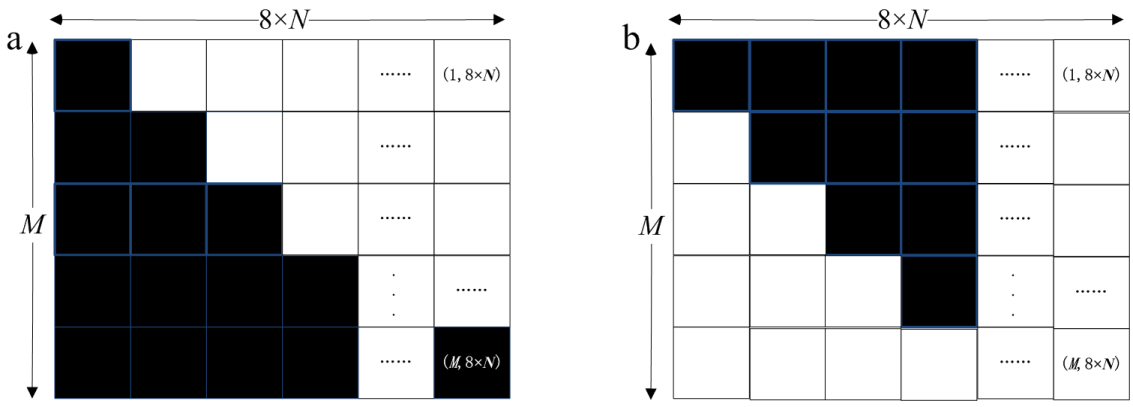

Figure. 4.4: Examples of plaintext image: (a) One for revealing vector TM, (b) One for revealing vector TN.

Algorithm 1 For revealing the row scrambling vector TM

Step1
for $i = 1$ to M **do**
  for $j = 1$ to N **do**
    $RCM(i, j) = 255$;
  end for
end for
Step2
for $i = 1$ to M **do**
  for $j = 1$ to N **do**
    for $g = 8(j-1)+1$ to $8j$ **do**
      $RCM'(i, g) = 1 \Leftarrow [RCM(i, j) \leftarrow Eq.(4.1)]$;
    end for
  end for
end for
Step3
for $p = 1$ to M **do**
  $\{e(k)|e(k) \subseteq \{1, 2, 3, \ldots, 8N\}\}$ $\leftarrow$Choose any k many y-coordinate(s) in $\{1, 2, 3, \ldots, 8N\}$;
  for $u = 1$ to k **do**
    $RCM'(p, e(u)) \leftarrow 0$;
  end for
end for
Step4
for $i = 1$ to M **do**
  for $j = 1$ to N **do**
    for $g = 8(j-1)+1$ to $8j$ **do**
      $t \Leftarrow [RCM'(i, g) \leftarrow Eq.(4.2)]$;
    end for
    $RCM(i, j) \leftarrow t$;
  end for
end for

- On the assumption that the used encryption machinery has been acquired provisionally. According to the size of the ciphertext image C (i.e., $M \times N$), we construct the chosen images of the same size (i.e., RCM, RCN_{s+1} (s$\in${0,1,2,3,4,5,6,7,..., m}, $s \geq 7$)) for the CPA. The details of the principle are as follows:

  - Case 1: $M \leq 8N$:

    The constitution method for revealing the row scrambling vector TM is described as "Algorithm 1", and the constitution method for revealing the column scrambling vector TM is described as "Algorithm 2".

    When $N < M \leq 8N$, the need of different plaintext images for revealing the vector TN is decided by the size of M, which demonstrates that the number

of RCN_{s+1} (i.e., NUM_{RCN}), in fact, only need to satisfy $1 \leq NUM_{RCN} \leq 8$. This is based on the fact that the maximal number of "0"s in RCN_{s+1} is dependent on the size of M in this case.

– Case 2: $M > 8N$:

Two kinds of methods can be utilized to construct the used plaintext images. The first method is the same as that of the condition of $M \leq 8N$. However, before this conformation, the image should be transposed at first, i.e., $M' = 8N$ and $(8N)' = M$ are used to set the plaintext images RCM' and RCN'_t ($t \in \{0,1,\ldots,k\}$), respectively. This is shown in Fig. 4.5. The second method can be seen as the inversion of the method under the condition $M \leq 8N$, i.e., the algorithm, which aims to construct the plaintext images for revealing TN and TM under the condition $M \leq 8N$, is used to produce the plaintext images for revealing TM and TN under the condition $M > 8N$, respectively.

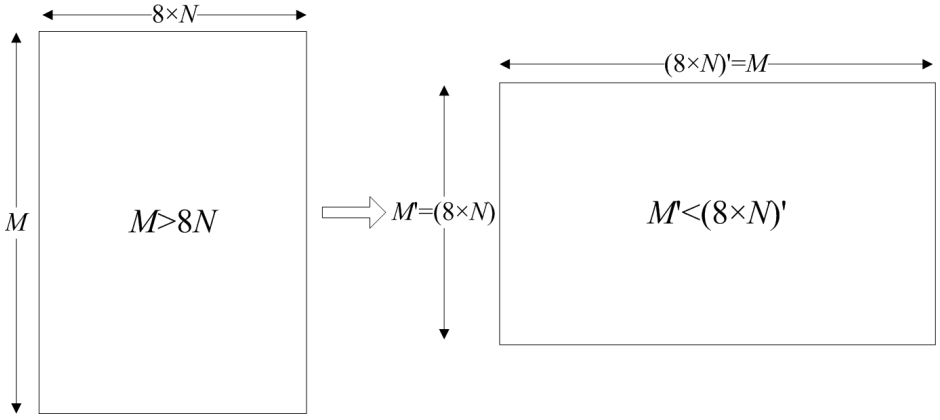

Figure. 4.5: Size exchanging before the CPA when $M > 8N$.

- Attack on TM:

The chosen image RCM is used in this temporary encryption machinery ($M \leq 8N$):

$$RCMM = E_{Ye}(RCM, TM, TN) = EQTWO([TM^{TR} \times EQONE(RCM) \times TN^{TR}]_{M,8N})_{M,N}, \quad (4.3)$$

where $EQONE(\cdot)$ implies Eq. (4.1), and $EQTWO(\cdot)$ indicates Eq. (4.2) in the above expression.

The following Eq. (4.4) is utilized for the ciphertext image $RCMM$ to get $RCMM'$ of size $M\times 8N$:

$$RCMM' \Leftarrow [RCMM \leftarrow Eq.(4.1)], \tag{4.4}$$

As the number of "0"(s) in each row is not changed during the encryption process, the number of "0"(s) in each row of the ciphertext image can be counted:

$$TM(x) = \sum_{y=0}^{8N-1} ZL(RCMM'(y)), \tag{4.5}$$

where $ZL(\cdot)$ is a function for counting the number of "0", $x\in\{1, 2, \ldots, M\}$.

The result is the equivalent key TM. For the decryption, the inversion of TM is the real key. This attack process can be simply described in Fig. 4.6 which reveals the used TM accurately.

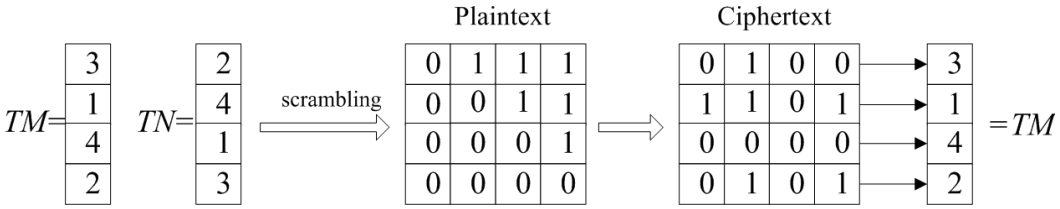

Figure. 4.6: Example of attack on TM.

- Attack on TN:

  The attack method on TN is similar to the way on TM. However, there are some differences for revealing TN:

  (1) If the number of RCN_{s+1} is more than 8, the attack process should be repeated at least 8 times for obtaining TN.

  (2) We should ensure that whether this column has "0" pixel(s): if this column has, we should count the number of "0" pixel(s). Otherwise, this column can

be ignored. For the column which has "0" pixel(s), the number of "0" pixel(s) (i.e., r) is counted in this column, and the corresponding value in the vector TN is $(r+s\times N)$.

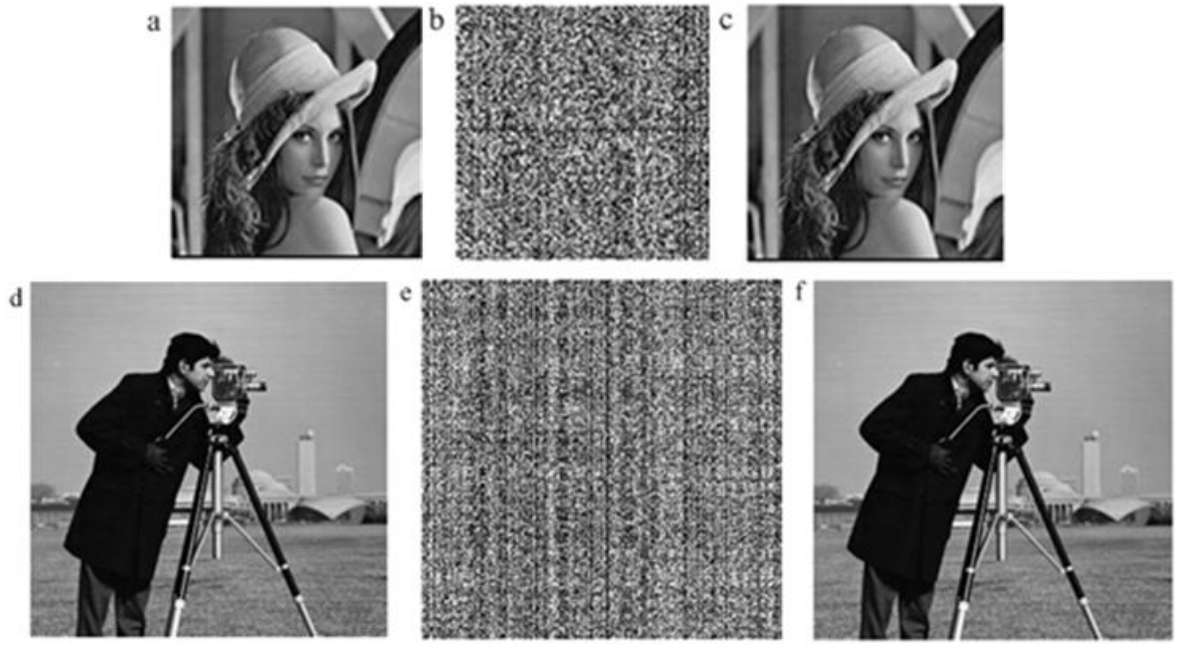

Figure. 4.7: Test images for proposed attacks: (a-c) plaintext image, ciphertext image and decrypted image of "Lena";(d-f) plaintext image, ciphertext image and decrypted image of "Cameraman".

When $M>8N$, the attack process on TM and TN is an exchange as $M\leq 8N$: RCM is used to reveal TN, and RCN_{s+1} is used to reveal TM. As this decryption process is the same as the encryption procedure except the keys for the *decryption*, which are the inversion vectors of the vectors TM and TN, the CCA can be also utilized to reveal the equivalent keys TM' and TN' for the decryption, not considering the used one-dimensional chaos system. The crack procedure and the used chosen-ciphertext images (RCM' and RCN'_{s+1}) are identical with the CPA. i.e., the chosen-plaintext images RCM and RCN_{s+1} are considered as chosen-ciphertext images RCM' and RCN'_{s+1} in the process of decryption analysis for acquiring the vectors TM' and TN'. If the decryption

vectors TM' and TN' are found, they can be used to recover the needful cipher image directly. The detailed steps of the CCA are similar to the above CPA (1-3).

4.4 Simulation on Proposed Attack

In this section, some experiments are provided for illustrating the proposed CPA/CCA. For demonstrating our attacks sufficiently, the gray-scale images "Lena" and "Cameraman" of size 128×128 and 256×256 (see Fig. 4.7(a) and (d)) are used in the original algorithm. The whole simulations are performed under the Matlab program running on AMD Turion(tm) Neo X2 Dual Core Processor L625 1.60GHz with 2 GB RAM. The secrets keys and corresponding parameters are chosen from the original example in [85]: x_0=0.2009, μ=3.98, m=20, n=51.

For the CPA, the adversary can choose 9 images (i.e., RCM and RCN_{s+1} (s∈{0, 1, 2, 3, 4, 5, 6, 7,})) as the used plaintext images which are utilized for revealing the equivalent keys TM and TN, respectively. Specially, the RCM is used to obtain the vector TM, and RCN_{s+1} are utilized to obtain the vector TN. The attack results is illustrated in Fig. 4.8:

In the attack process, it must be noted that the used plaintext images (i.e., Fig. 4.8a(d) and b(e)) are used to reveal TM and TN, respectively. Specially, in RCM, the values of gray pixels belong to {254, 252, 248, 240, 224, 192, 128, 0}, the white pixels only signify 255. In RCN_{s+1}, the values of gray pixels belong to {127, 63, 31, 15, 7, 3, 1, 0}, and the white pixels are the same as that in RCM. As examples for the chosen images in Fig. 4.8a(d) and b(e), if the size (n×n)=(19×19), the RCM and RCN_1 can

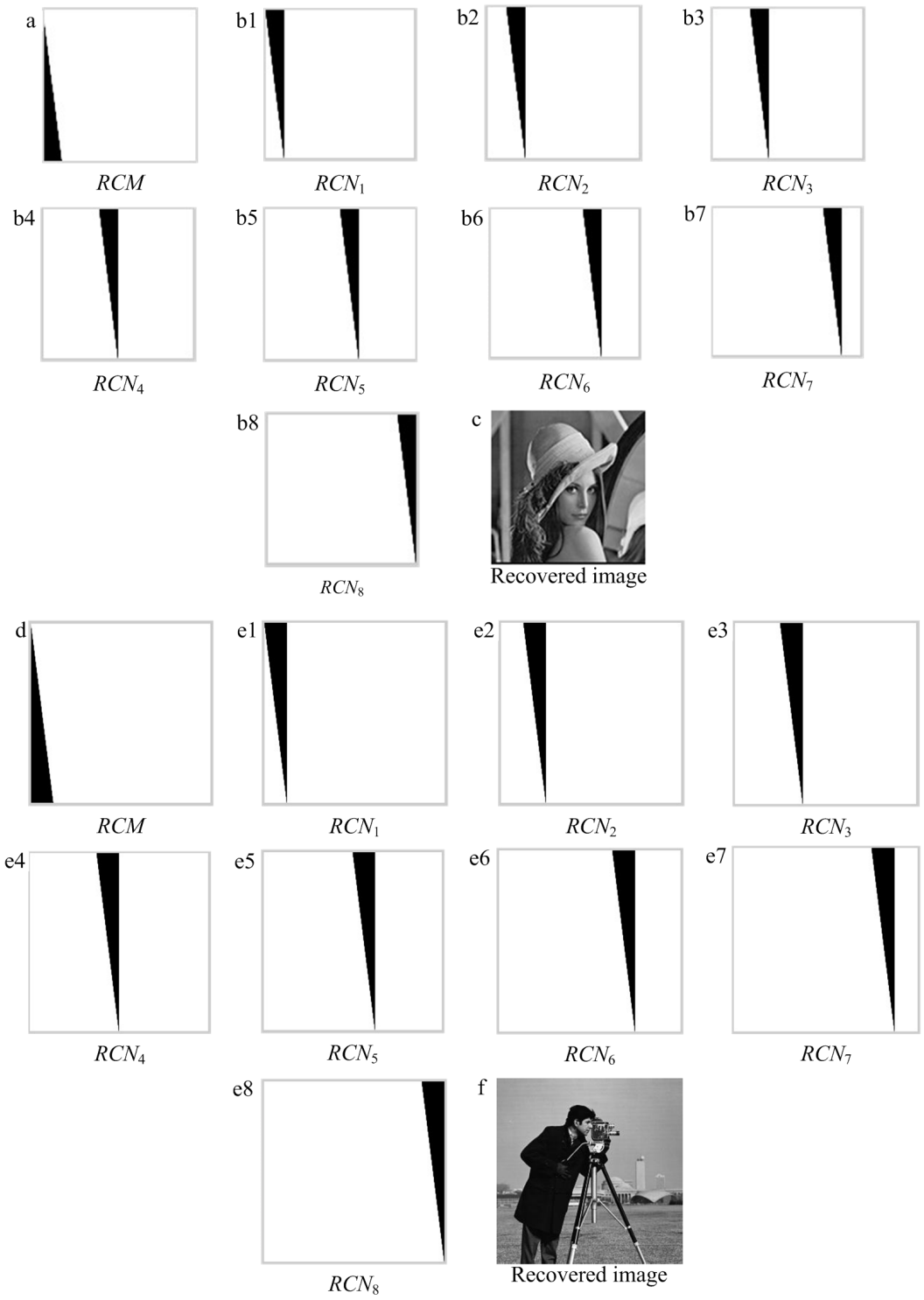

Figure. 4.8: Chosen-plaintext attack to ciphertext images b and e in Fig. 4.7: (a, d) plaintext images for revealing *TM*; (b, e) plaintext images for revealing *TN*; (c, f) recovered images of "Lena" and "Cameraman".

be expressed as:

$$RCM=\begin{bmatrix} 254 & 255 & 255 & \cdots & 255 & 255 \\ 252 & 255 & 255 & \cdots & 255 & 255 \\ 248 & 255 & 255 & \cdots & 255 & 255 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 252 & \cdots & 255 & 255 \\ 0 & 0 & 248 & \cdots & 255 & 255 \end{bmatrix}; \quad RCN_1 = \begin{bmatrix} 0 & 0 & 248 & \cdots & 255 & 255 \\ 1 & 0 & 248 & \cdots & 255 & 255 \\ 3 & 0 & 248 & \cdots & 255 & 255 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 255 & 255 & 249 & \cdots & 255 & 255 \\ 255 & 255 & 251 & \cdots & 255 & 255 \end{bmatrix}$$

Then, these matrices can be transformed into the binary matrices of size 19×152:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 1 \end{bmatrix}; \quad \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ 1 & 1 & \cdots & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 1 & 1 & 1 \\ 1 & 1 & \cdots & 1 & 1 & 0 & \cdots & 1 & 1 & 1 \end{bmatrix}$$

For this RCM, the number of "0"s in each row equals to the corresponding row number e.g., the 19^{th} row has 19 "0"s. If such a RCM is encrypted by the original algorithm, the number of "0"(s) in each row is the corresponding value in TM, e.g., in the 6th row of the scrambled matrix, if the number of "0"s is 18, the vector value is 18 in the $TM(6, 1)$. According to Eq. (4.5), the number of "0"s in each row can be computed, which also implies that the TM is revealed. For this RCN_1, the analysis process is similar to that of RCM. If there are 16 "0"s in the 8th column of the scrambled matrix, the vector value in $TM(1,8)$ is 16.

Specially, if the whole attack is used for the ciphertext image with the same height and width (M=N), only 9 chosen-plaintext images are required to reveal TM and TN.

$$(TM, TN)$$
$$= \sum ZL(EQONE((E_{Ye}(RCM, TM, TN),\ E_{Ye}((RCN_1, RCN_2, ...RCN_8), TM, TN))))$$

After that, the decryption vector TM' and TN' are the inversion of TM and TN:

$$(TM', TN') = INV(TM, TN), \tag{4.6}$$

in which $INV(\cdot)$ is a function of the inversion operation. Then, the original plaintext image can be recovered according to TM' and TN' successfully.

For the CCA, the process is similar to the CPA. The chosen-ciphertext images are the same as the chosen-plaintext images, which implies that RCM and RCN_{s+1} can be seen as the used ciphertext images (i.e., RCM' and RCN'_{s+1}). Specially, the revealed vector are the decryption keys which can be used to decrypt the needful ciphertext image immediately.

4.5 Remarks on Attack

Particularly, there are still three remarks on our attacks:

- On the generation of the vectors TM and TN used in the original algorithm [85]: The Logistic map is used to produce the random sequence in the original algorithm. However, according to the analysis proposed by Li et al. [46], it was found that the Logistic map can not be considered as a good *pseudorandom number generator*. In our opinion, whether the chaos system is the Logistic map or not is not important for the whole encryption process. Since the transformation vector TM and TN which are produced by a chaotic system can be revealed by our attacks, no matter which chaos system is used, these vectors can still be obtained.

- The iteration encryption of the original algorithm [85]: The original algorithm can be iterated several rounds for ensuring security. However, the iteration encryption of this algorithm is the same as the *multiplication* of many matrices:

$$\left(\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\times\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\times\cdots\times\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}\right)\times\begin{bmatrix}127&46&2\\6&254&78\\89&48&24\end{bmatrix}$$
$$\times\left(\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\times\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}\times\cdots\times\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\right) \tag{4.7}$$

In the above equation, the multiplication of matrices in the left side and in the right side can be seen as:

$$\left(\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\times\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\times\cdots\times\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}\right)=TM^{TR}$$

$$\left(\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\times\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}\times\cdots\times\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\right)=TN^{TR}$$

Therefore, the objective of the attack is to find out the final vector which corresponds to the product of these matrices. Since our attack sets up a relationship between the final ciphertext image and the plaintext image, it can effectively obtain the final product of many matrices on the left (or right) side of the original image. This makes the *iteration encryption* ineffective.

- The used plaintext images/ciphertext images RCM and RCN_{s+1} are not unique: e.g., if the matrix A as defined bellow is the used plaintext image, its complement image A^c can also play the role for our attack, where A and A^c are defined bellow:

$$A=\begin{bmatrix}254&255&255\\252&255&255\\248&255&255\end{bmatrix}\Rightarrow A^c=\begin{bmatrix}1&0&0\\3&0&0\\7&0&0\end{bmatrix}. \tag{4.8}$$

4.6 Quantified Comparison

Li and Lo [45] presented the known-plaintext attack (KPA) and the chosen-plaintext attack (CPA) for breaking the original image encryption scheme. For both attacks, Li and Lo, made use of the constructed binary tree to reveal the permutation matrix W, which can be seen as an equivalent key to decrypt the cipher image encrypted by the same secret keys. Especially, for the known-plaintext attack, the number of known-plaintext images (i.e., Kn) should satisfy $Kn{>}\lceil\log_2(8MN\text{-}1)\rceil$, and for the chosen-plaintext attack, the

Table. 4.1: Number of plaintext images/ciphertext images in our attack and Li and Lo's attack.

	$M=N$	$M<N$	$M>N$ $M>8N$	$M\leq 8N$
Our Attack	9	$\lceil 8N/M \rceil+1$	$\lceil M/8N \rceil+1$	≤ 9
Li and Lo's attack	SO	SO	SO	SO

SO: $\geq(\lceil \log_2(8MN\text{-}1) \rceil+1)/\geq\lceil \log_2 8MN \rceil$

number of chosen-plaintext images (i.e., Cn) should satisfy: $Cn\geq 3+\lceil \log_2(MN) \rceil$, where M and N are the size of an image. The spatial complexity and computational complexity of Li and Lo's attack [45] (i.e., the known-plaintext attack and chosen-plaintext attack) are $O(32{\cdot}MN)$ and $O(16{\cdot}n_0{\cdot}MN)$, respectively, where n_0 denotes Kn or Cn. About our attack, the corresponding spatial complexity and computational complexity are $O(8{\cdot}MN)$ and $O(8{\cdot}n_1{\cdot}MN)$ for obtaining TM and TN, respectively, where n_1 is the number of chosen-plain images/chosen-cipher images.

According to the above analysis, it can be found that both the spatial complexity and computational complexity of out attack and Li and Lo's attack can be roughly expressed as $O(MN)$ and $O(num{\cdot}MN)$, where num is the number of the used plaintext images/ciphertext images. This implies that the number of the used images largely determines which attack is more efficient. Table 4.1 lists the number of the used plaintext images/ciphertext images for our attack and Li and Lo's attack.

From the Table 4.1, it can be concluded that when $M=N$ and $M\leq 8N$, in general, our attack is better than the attack by Li and Lo, as the number of chosen-plain images/chosen-cipher images in our attack is not more than 9. For $M<N$ and $M>8N$, some concrete tests about the number of the used plain images/cipher images are listed as follows:

- Case 1: $M<N$:

  Some sizes of N are randomly selected in Table 4.2, which come from [83]. $M\in\{1,$

Table. 4.2: General sizes of N.

	Internet Ads		PDA and phone screens		Computer screens			Television screens	
N	728	468	320	480	1024	1600	1920	576	720

2, 3, ..., N-1}, which is the height of an image. These general sizes of N are used in Internet Ads, PDA and Phone Screens, Computer Screens and Television Screens. Fig. 4.9 and 4.10 are the corresponding numbers of the used plaintext images/ciphertext images for the different N.

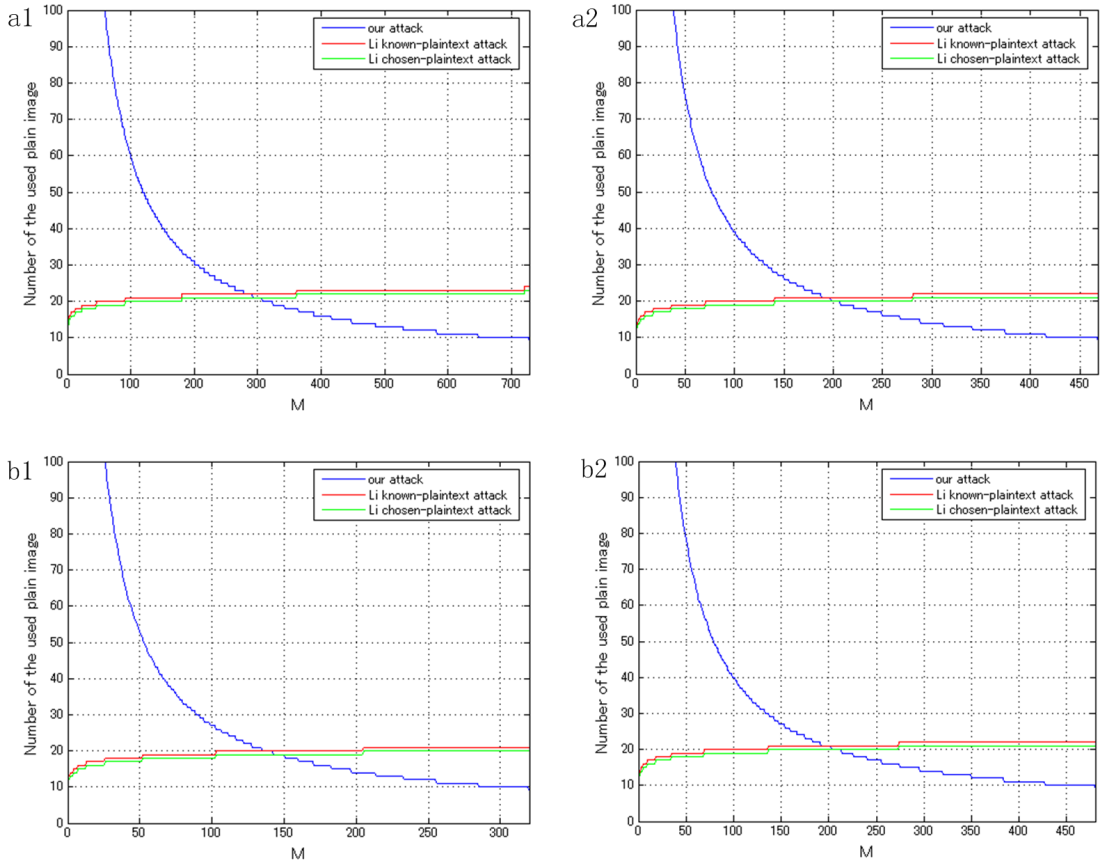

Figure. 4.9: Numbers of used plaintext images/ciphertext images for different N: (a1) N=728, (a2) N=468, (b1) N=320, (b2) N=480.

From Fig. 4.9 and 4.10, it can be found that when the size of M is larger than some fixed value, the number of the plaintext images/ciphertext images in our attack are gradually fewer than that of Li and Lo's attack. Note that, when the

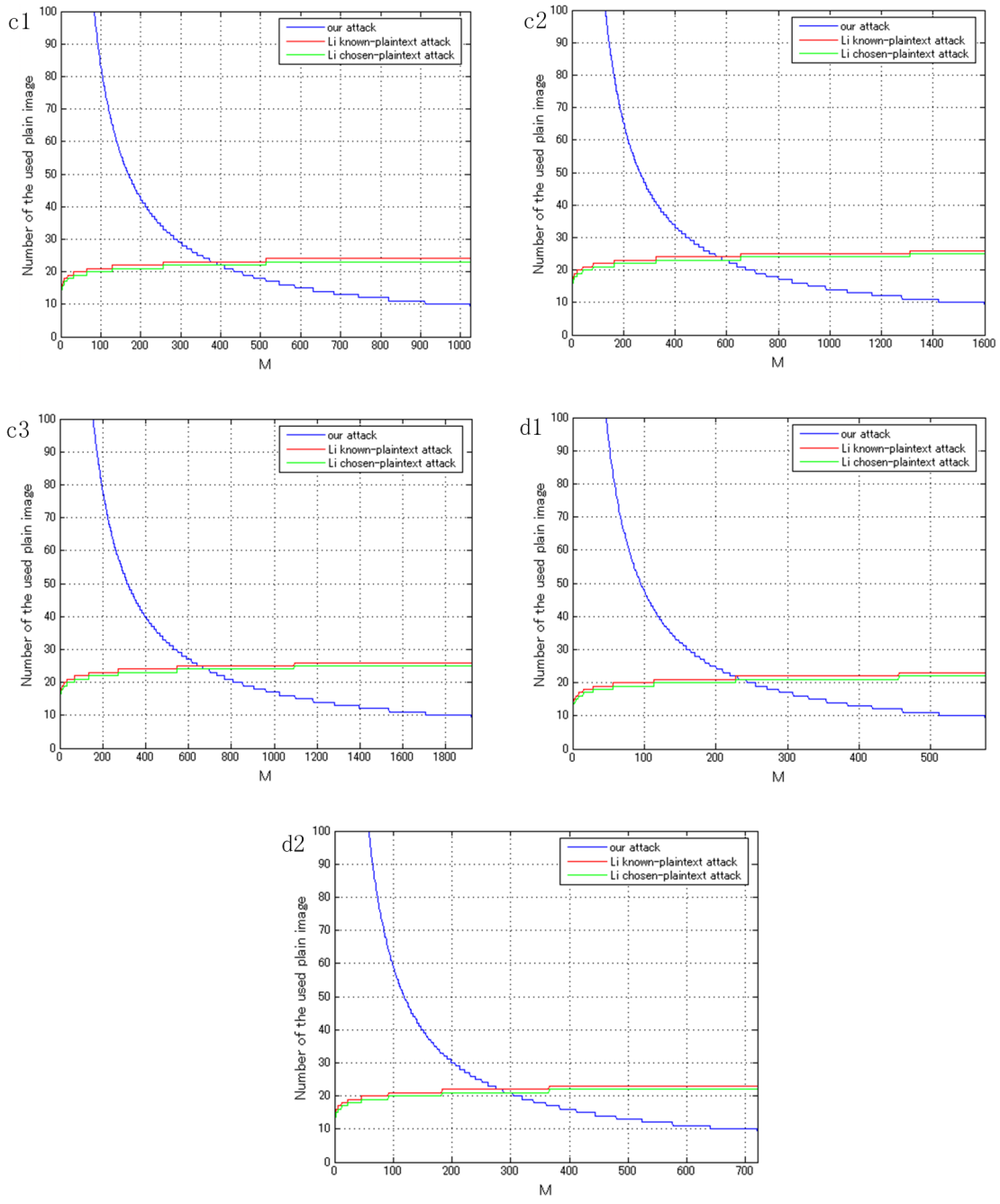

Figure. 4.10: Numbers of used plaintext images/ciphertext images for different N: (c1) N=1024, (c2) N=1600, (c3) N=1920, (d1) N=576, (d2) N=720.

Table. 4.3: Common aspect ratios.

Aspect ratio (N/M)	1:1	5:4	4:3	8:5	16:9
Decimal	1.00:1	1.25:1	1.33:1	1.60:1	1.78:1

Table. 4.4: Aspect ratios of Fig. 4.9 and 4.10.

	Internet Ads		PDA and phone screens		Computer screens		
	N=728	N=468	N=320	N=480	N=1024	N=1600	N=1920
KPC	$\frac{728}{291}$	$\frac{468}{197}$	$\frac{320}{142}$	$\frac{480}{202}$	$\frac{1024}{390}$	$\frac{1600}{581}$	$\frac{1920}{668}$
	≈2.50:1	≈2.38:1	≈2.25:1	≈2.38:1	≈2.63:1	≈2.75:1	≈2.88:1
CPC	$\frac{728}{306}$	$\frac{468}{207}$	$\frac{320}{150}$	$\frac{480}{213}$	$\frac{1024}{490}$	$\frac{1600}{609}$	$\frac{1920}{699}$
	≈2.38:1	≈2.26:1	≈2.13:1	≈2.25:1	≈2.50:1	≈2.63:1	≈2.75:1
	Television screens						
	N=576	N=720					
KPC	$\frac{576}{230}$	$\frac{720}{287}$					
	≈2.50:1	≈2.51:1					
CPC	$\frac{576}{242}$	$\frac{720}{303}$					
	≈2.38:1	≈2.38:1					

size of M is close to N, the difference between the number of the used plaintext images/ciphertext images in our attack and that in Li and Lo's attack is increasing largely. Table 4.4 shows the aspect ratios (i.e., N/M) of Fig. 4.9 and 4.10 when the number of the plaintext images/ciphertext images in our attack is just larger than that of Li and Lo's attack. Specially, KPC denotes the *aspect ratio* (i.e., N/M) when the number of the plaintext images/ciphertext images in our attack is just larger than that of Li and Lo's known-plaintext attack, and CPC denotes the aspect ratio (i.e., N/M) when the number of the plain images/cipher images in our attack is just larger than that of Li and Lo's chosen-plaintext attack.

Table 4.4 demonstrates that Li and Lo's attack have a lower computational complexity than ours only when the aspect ratio is higher than the values in this table.

However, according to the aspect ratios in Table 4.3, which come from the link [83], we can find that for the general image, the aspect ratio is always lower than 2. This illustrates that our attack is more efficient than Li and Lo's attack when the general images are considered.

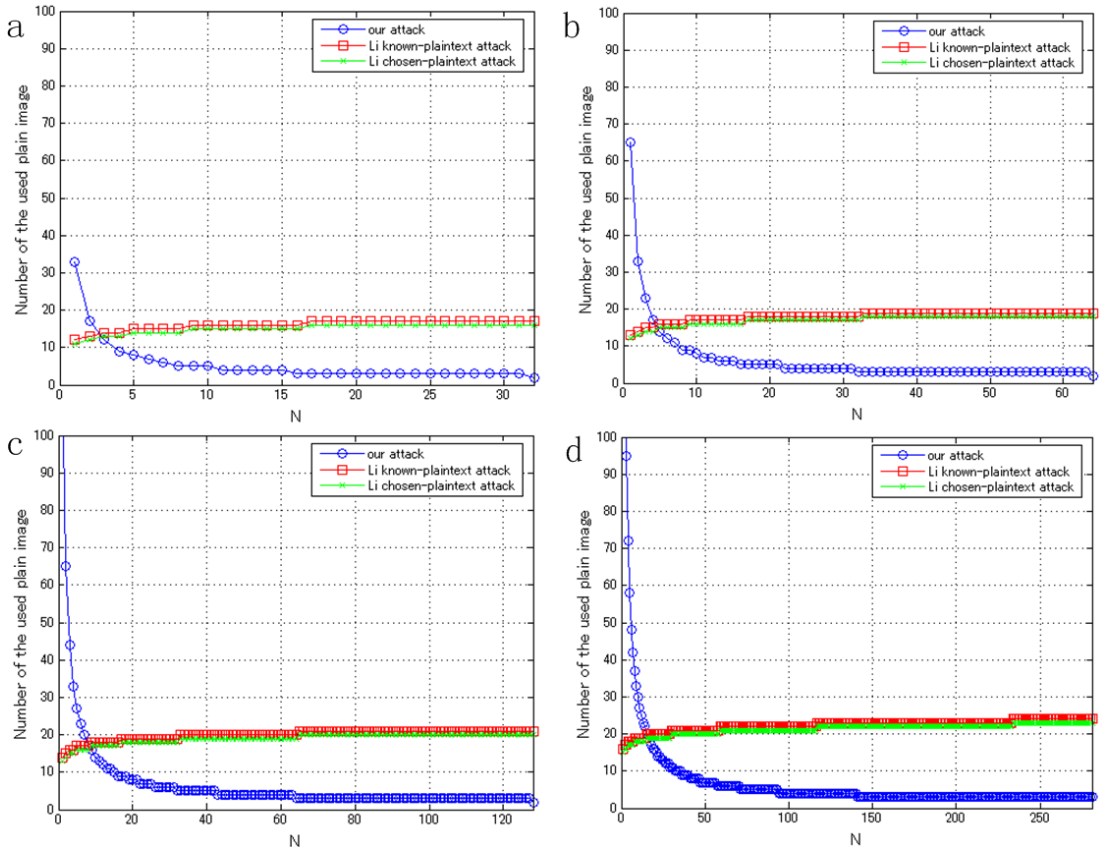

Figure. 4.11: Numbers of used plaintext images/ciphertext images for different M: (a) M=256, (b) M=512, (c) M=1024, (d) M=2250.

- Case 2: $M>8N$:

  As the most digital images satisfy $M<N$, it is hard for us to find some general sizes of M, which satisfy $M>8N$. In our analysis, some standard sizes from the image database [84] are set as the sizes of M: $M\in\{256, 512, 1024, 2250\}$. N is the width of an image. Fig. 4.11 shows the corresponding numbers of the used plaintext images/ciphertext images for the different values of M.

Table. 4.5: Aspect ratios of Fig. 4.11.

	M=256	M=512	M=1024	M=2250
KPC	2/256=1:128	4/512=1:128	8/1024=1:128	15/2250=1:150
CPC	2/256=1:128	4/512=1:128	8/1024=1:128	16/2250=1:140.625

From Fig. 4.11, it can also be found that when the size of N is larger than some fixed value, the number of the plaintext images/ciphertext images in our attack is gradually fewer than that of Li and Lo's attack. Moreover, when the size of N is close to $M/8$, the difference between the number of the used plaintext images/ciphertext images in our attack and that in Li and Lo's attack is also increasing largely. For our attack, the number of the used plaintext images/ciphertext images is close to 2 when N approaches to $M/8$. Table 4.5 presents the aspect ratios of Fig. 4.11 when the number of the plaintext images/ciphertext images in our attack is just larger than that of Li and Lo's attack.

Table 4.5 shows that if the computational complexity of Li and Lo's attack is lower than that of our attack, it must satisfy the aspect ratios in Table 4.5. However, these aspect ratios are too small for the general images. Table 4.6 lists the probability P_N which is defined as P_N=S_N/L_N, where S_N is the number of the size N which makes the number of the plaintext images/ciphertext images in our attack be fewer than that of Li and Lo's attack. L_N is the largest number of the size N, which satisfies L_N=$\lfloor M/8 \rfloor$-1.

According to Table 4.6, it demonstrates that for the most general images, our attack needs fewer plain images/cipher images than that of Li and Lo's attack. Therefore, our attack have a lower computational complexity, which makes our attack more suitable for the practical situation.

To sum up, the above analyses illustrate that our attack has a lower computational complexity for the general images which satisfy the common aspect ratios. From the attack precision, according to the example in [45], the known-plaintext attack

Table. 4.6: Probability P_N.

	M=256	M=512	M=1024	M=2250
KPC	30/32=0.9375	60/64=0.9375	120/128=0.9375	266/281≈0.9466
CPC	30/32=0.9375	60/64=0.9375	120/128=0.9375	265/281≈0.9431

can only recover the plain image with some noise if ($\lceil \log_2(8MN\text{-}1) \rceil$+1) known-plain images are used in Li and Lo's known-plaintext attack (KPA). To obtain a plain image with the better quality, more known-plain images have to be used in Li and Lo's known-plaintext attack (KPA). Therefore, the number of the known-plain image in Li and Lo's known-plaintext attack (KPA) should satisfy $\lceil \log_2(8MN\text{-}1) \rceil + add$, where $add \geq 2$. Based on this reason, the precision of our attack is better than Li and Lo's known-plaintext attack (KPA) if ($\lceil \log_2(8MN\text{-}1) \rceil$+1) known-plain images are used in Li and Lo's known-plaintext attack.

4.7 Suggestion on Improvement of Original Algorithm

According to the analysis of Sections 4.3 and 4.4, the primary flaw of the original algorithm is that the equivalent vectors TM and TN (or TM' and TN'), generated by the secret keys, are not related to the original plaintext image. As a result, the adversary can obtain TM and TN (or TM' and TN') independently. Moreover, the encryption of the pixel value, only implemented in every row, can be considered as the secondary flaw. This causes the drawback that the encryption range of pixel value is limited. However, the original algorithm can be still considered as a novel image encryption idea which has some inherent merits, e.g., the parallel encryption of pixel position and pixel value. In this section, we discuss about the improvement on the original algorithm, which retains the good properties of the original algorithm [85]. Especially, in the suggested improved algorithm, the *self-correlation* encryption is introduced to solve the main drawback. This kind of encryption comes from the idea of the original *self-adaptive encryption* proposed by Chen et al. [28]. In the work [28], the image was firstly divided into two equal parts.

Then, the arranging order of pixel values in one part is used to scramble the other part of the image. After that, the scrambled part is, in turn, used to shuffle the other one which had not been scrambled in this round. In Fig. 4.12, the right part of the original image encrypts the left part firstly, and the left part of this image is used to encrypt the right part secondly. In addition, the image can also be cut into upper part and lower part for finishing the same encryption. In our suggested improvement, we take advantage of the self-correlation encryption to produce the encryption vectors (i.e., $RM(z)$, $z \in \{1, 2, 3, 4\}$). The self-correlation encryption can ensure that the generated $RM(z)$ are not only based on secret keys, but also related to the plaintext image. More precisely, the self-correlation encryption implies that the generation of the equivalent key (i.e., $RM(z)$ in the improvement) depends on both the plaintext image and secret keys controlled by the user. Furthermore, in each round, the equivalent key is different.

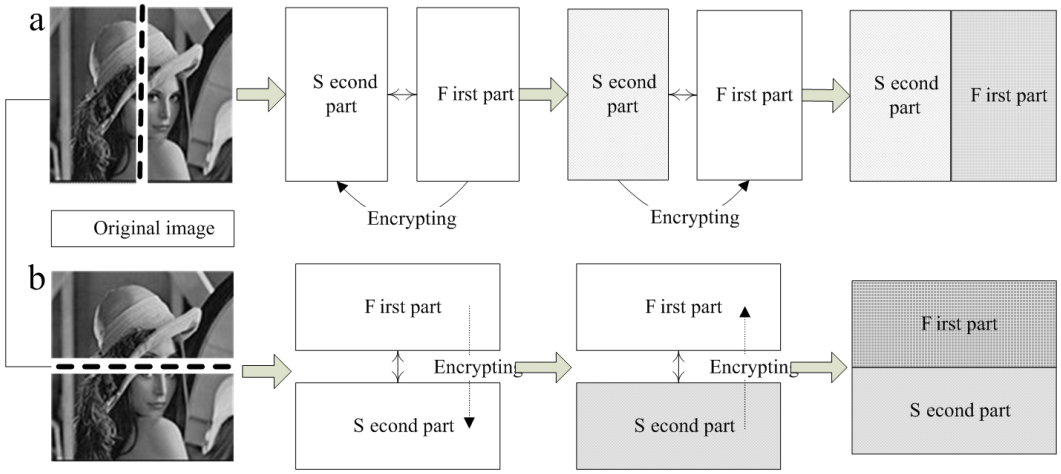

Figure. 4.12: One round of self-adaptive encryption: (a) partitioning according to vertical-direction, (b) partitioning according to horizontal direction.

To permute the pixel bit sufficiently in light of the row and column to solve the second issue, the suggested improved algorithm employs two stages: firstly, permute the pixel location and pixel value of the row simultaneously; secondly, permute the pixel location and pixel value of the column simultaneously. In other words, the suggested improvement carries out two-time encryption operation, which *transposing operation* to

connect them.

4.7.1 Partitioning Method of Plaintext Image

For a plaintext image, the partitioning of the proposed self-correlation encryption can be implemented in any size and any direction. Specially, the plaintext image can also be cut into any parts, (e.g., Fig. 4.13). This implies that there is no limitation on the partitioning method for an image with any shape (e.g., square or rectangle). In our proposal, for the convenience, if the height (or width) of the image (i.e., M or N) is even number, the original image is cut into two parts of the same height (or width), i.e., $M/2$ or $N/2$, otherwise, the original image is divided into two parts of the different height (or width), e.g., if the M is odd number and the image is cut according to the horizontal-direction, the heights for two parts are (M-1)/2 and (M-1)/2+1, respectively. Moreover, as the transposing operation can transform the pixels of the rows into the pixels of the columns, we only need to adopt the vertical-direction partitioning in the suggested improved algorithm.

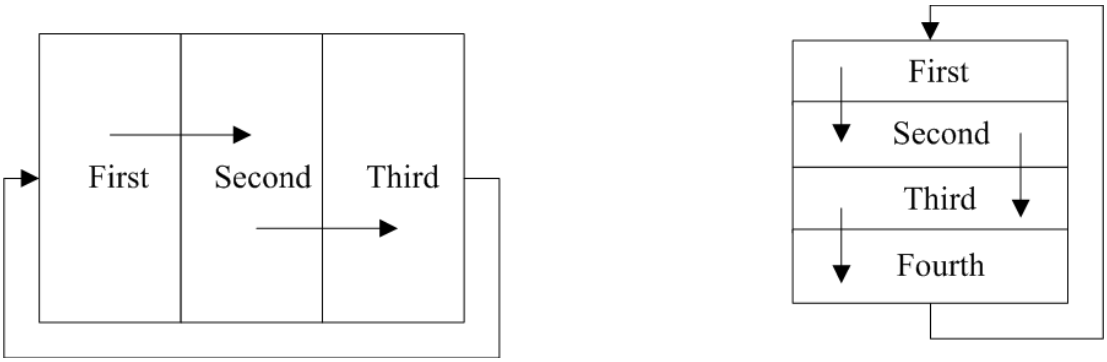

Figure. 4.13: Partitioning method of self-correlation encryption.

4.7.2 Suggestion on Key Scheduling

It is well known that the *key scheduling* is extremely significant. However, in the works [45, 46], it can be found that the Logistic chaotic map is not considered to have the ability to yield the random sequence with the good performance. This demonstrates

that if the encryption vectors TM and TN are produced by the Logistic chaotic map, they may not be pseudorandom which may affect the security of the cipher. Therefore, in the suggested key generation, the coupled map lattices (CML) consisting of the skew tent map is used to construct the *spatiotemporal chaotic system* for producing the necessary *pseudorandom sequences*. The spatiotemporal chaotic system is seen as a natural choice for the cryptographic applications since it possesses a large number of positive *Lyapunov exponents*. As the drawback of the *short period* from discretized chaotic maps in the digitalized implementation is overcome, it is more suitable for generating a single or multiple pseudorandom sequences compared with other chaotic systems [81]. The more merits on the spatiotemporal chaotic system can be found in [81]. The mathematical formula of this system is given by: x^i_{t+1}=(1-ε)$g(x^i_t)$+$\varepsilon g(x^{i-1}_t)$, where ${x_t}^i$ is the state value in the i^{th} site at time t. $\varepsilon\in(0,\ 1)$ is the coupling coefficient (ε=0.99 in our improvement), i=$\{1,\ 2\}$, $g(\cdot)$ function is the *skew tent map* presented by Eq. (4.9), in which q^i is the system parameter. The *periodic boundary condition* is used in CML for any t, such as ${x_t}^0$= ${x_t}^2$.

$$x^i_{t+1} = \begin{cases} x^i_t/q^i, & x^i_t \in (0, q^i) \\ (1-x^i_t)/(1-q^i), & x^i_t \in [q^i, 1) \end{cases} . \tag{4.9}$$

As the two-time encryption operations in one round are carried out, there are six transformation vectors for the encryption algorithm. They can be called ($RM(1)$, $RM(2)$, $CN(1)$); ($RM(3)$, $RM(4)$, $CN(2)$). The details of the generation rule for each sequence vector are as follows:

For $RM(z)$ ($z\in\{1,\ 2,\ 3,\ 4\}$):

- Produce a chaos sequence ${x_h}^u(u\in\{1,\ 2\})$ according to the above spatiotemporal chaotic system. Especially, One (${x_h}^1$) is used for producing ($RM(1)$, $RM(2)$), and the other one (${x_h}^2$) can be used to generate ($RM(3)$, $RM(4)$).
- Collect M values $\{x_{g+1},\ x_{g+2},\ldots,\ x_{g+M}\}$ from the g^{th} number of the sequence ${x_h}^u$.
- Transform each value in the sequence $\{x_{g+1},\ x_{g+2},\ldots,\ x_{g+M}\}$ from (0, 1) to $\{0,\ 1,\ 2,\ \ldots,\ 255\}$ according to: $\ddot{e}_{\dot{g}}$=$round(x_{\dot{g}}\times 255)$, where $round(\alpha)$ is used to obtain

an integer which is near to α. $\ddot{e}_{\dot{g}}$ is the corresponding value of $x_{\dot{g}}$ in {0, 1, 2, ..., 255}.

- Select nu column(s) in the corresponding part of an image (i.e., I) randomly ($nu \geq 1$, nu is the number of the selected column. In our experiments, $nu=1$ for the convenience). The coordinate of column (i.e., l) can be decided by Eq. (4.10):

$$l = \begin{cases} \begin{cases} N - floor(\mathrm{mod}((x_h^u \times 10^{14}),\ (N - floor(N/2)))) \\ l \in (floor(N/2) + 1,\ N) \end{cases} \\ \begin{cases} floor(\mathrm{mod}((x_h^u \times 10^{14}),\ floor(N/2))) + 1 \\ l \in (1,\ floor(N/2)) \end{cases} \end{cases} , \tag{4.10}$$

  where $floor(\alpha)$ is a function for obtaining the largest integer which is less than α. mod$(\cdot)$ is a modular arithmetic. For the x_h^u, it satisfies $h>1$, $u=1$ for ($RM(1)$, $RM(2)$) and $u=2$ for ($RM(3)$, $RM(4)$).

- Pick up the values of the column(s) in the corresponding part of I, These values are combined with $\ddot{e}_{\dot{g}}$ using the *bitwise exclusive-or operation* (x-or operation):

$$YM(z) = \ddot{e}_{\dot{g}} \oplus I(1) \oplus I(2) \cdots \oplus I(l) \ , \tag{4.11}$$

  where $YM(z)$ is the final result of the x-or operation(s).

- Sort for $YM(z)$. Find and mark down the transformed positions $RM(z)$. $RM(z)$ is the vector for the row position permutation.

For $CN(v)$ ($v \in \{1, 2\}$):

The $CN(v)$ is used for the column permutation in bit-plane of which the size is $8N$. This permutation can have an effect on the pixel value encryption. In the suggested improvement, the production method of $CN(v)$ is the same as the Section 3.2 of the work [85] which preserves the characteristic of the original algorithm. In particular, the used chaos sequence ${x_z}^u$ ($u \subseteq \{1, 2\}$) is also obtained by the above spatiotemporal chaotic system.

Note that we should make use of the property of the spatiotemporal chaotic system sufficiently for producing some pseudorandom sequences synchronously. in our suggested improvement, we generate the vectors according to this turn: ($RM(1)$, $RM(2)$), ($CN(1)$, $CN(2)$) and ($RM(3)$, $RM(4)$). Based on this mechanism, the efficiency of the key scheduling can be enhanced.

However, the spatiotemporal chaotic system is not the unique choice for the improved scheme. Some other chaotic systems which have good performance can also be used in the key generation.

4.7.3 Encryption Steps

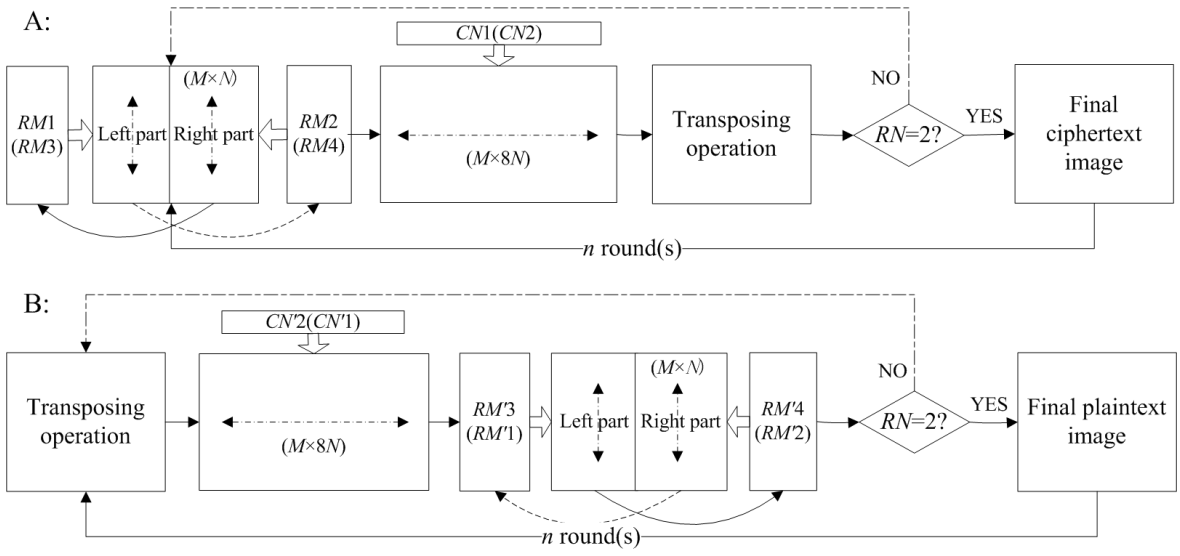

Figure. 4.14: Encryption/decryption demonstration (RN is time of encryption in one round): (A) encryption procedure; (B) decryption procedure.

The suggested improvement is based on the original algorithm, which retains the merits of the original algorithm. Therefore, the suggested improvement can also implement the location scrambling and gray value encryption simultaneously. The main encryption operation in the improved image encryption scheme is also the permutation for the row and column. The primary difference is that in the improved algorithm, there exists the x-or operation for producing the vector $YM(z)$ used to achieve the $RM(z)$. The steps

are as follows:

- Step 1: An image is first presented as a decimal matrix $P(i,\ j)_{M,N}$ of size $M\times N$ with value from $\mathbb{Z}_{256}$.

- Step 2: The used spatiotemporal chaotic system is iterated 200+nir rounds initially for making it into the chaos state, and the last values (${x_{200+nir}}^1$ and ${x_{200+nir}}^2$) are taken out to decide the encryption turn (ET):

$$ET = \begin{cases} The\ 1^{st}\ encryption: \\ \begin{cases} \mathrm{mod}(floor(x^1_{200+nir}\times 10^3),\ 2)=0 & LMA\ is\ encrypted\ firstly \\ \mathrm{mod}(floor(x^1_{200+nir}\times 10^3),\ 2)\neq 0 & RMA\ is\ encrypted\ firstly \end{cases} \\ The\ 2^{nd}\ encryption: \\ \begin{cases} \mathrm{mod}(floor(x^2_{200+nir}\times 10^3),\ 2)=0 & LMA\ is\ encrypted\ firstly \\ \mathrm{mod}(floor(x^2_{200+nir}\times 10^3),\ 2)\neq 0 & RMA\ is\ encrypted\ firstly \end{cases} \end{cases}, \tag{4.12}$$

  where LMA is the left part of an image, RMA is the right part of an image, and nir is an integer which satisfies $nir\geq 1$. In our experiment, nir=1 for the convenience. Note that nir can be changed for the new encryption, and it is decided by the user.

- Step 3: This matrix $P(i,\ j)_{M,N}$ is divided into two parts equally: one is the left part ($LMA_{M,floor(N/2)}$), and the other one is the right part ($RMA_{M,N-floor(N/2)}$).

- Step 4: According to Step 2, if ET=0, $RMA_{M,N-floor(N/2)}$ is used to produce $RM(1)$ according to the corresponding method in the Section 4.7.2. Then, $RM(1)$ is utilized to scramble $LMA_{M,floor(N/2)}$: ${LMA'}_{M,floor(N/2)}$=[$RM(1)^{TR}\times LMA_{M,floor(N/2)}$], where $RM(1)^{TR}$ is the corresponding transformation matrix of $RM(1)$ with the size of $M\times M$. Note that for the above step, if $ET\neq 0$, the process is the same. However, $LMA_{M,floor(N/2)}$ is used to generate $RM(1)$ according to the method in about section. $RMA_{M,N-floor(N/2)}$ is encrypted by the $RM(1)$ for achieving the ${RMA'}_{M,N-floor(N/2)}$.

- Step 5: Based on the Step 4 (i.e., if ET=0), the acquired $LMA'_{M,floor(N/2)}$ is used to generate $RM(2)$ according to the same way in Section 4.7.2. After that, $RM(2)$ is utilized to encrypt $RMA_{M,N-floor(N/2)}$: $RMA'_{M,N-floor(N/2)}$= $[RM(2)^{TR} \times RMA_{M,N-floor(N/2)}]$, where $RM(2)^{TR}$ is the corresponding transformation matrix of $RM(2)$ of size $M \times M$. Moreover, for the condition $ET \neq 0$, the process is also the same. Nevertheless, the $RMA'_{M,N-floor(N/2)}$ is used to produce $RM(2)$, and $LMA_{M,floor(N/2)}$ is scrambled by the $RM(2)$ for achieving the $LMA'_{M,floor(N/2)}$.

- Step 6: The $LMA'_{M,floor(N/2)}$ and $RMA'_{M,N-floor(N/2)}$ are integrated into one matrix P^t, and the pixel values in the matrix $P^t(i, j)_{M,N}$ are transformed into the bits sequences ($C(i, j)$=($b_{(7)}b_{(6)}b_{(5)}...b_{(0)}$)) for forming a $M \times 8N$ binary matrix C according to Eq. (4.1).

- Step 7: The vector sequence $CN(1)$ is generated following the method in Section 4.7.2. Then, the matrix $C_{M,8N}$ is encrypted by $CN(1)$: $C^t{}_{M,8N}$=$[C_{M,8N} \times CN(1)^{TR}]$, where $CN(1)^{TR}$ is the corresponding transformation matrix of $CN(1)$ of size $8N \times 8N$.

- Step 8: The ciphertext matrix $C^t{}_{M,8N}$ is transformed back to a decimal matrix K according to Eq. (4.2). Then, the matrix K implements one-time transposing operation to get transport matrix K^t.

- Step 9: For the matrix K^t, the steps from Step 3 to Step 8 are repeated to be carried out again (i.e., $RM(3)$, $RM(4)$ and $CN(2)$ are used in Step 4, Step 5 and Step 7, respectively), and the final ciphertext image $D_{M,N}$ is achieved after the transposing operation.

The details about this algorithm can be obtained from Fig. 4.14.

This encryption process can be performed several rounds to achieve a good encryption effect. During each round (round>1), the encryption turn (ET) is different, and it is decided by the last two values of the spatiotemporal chaotic system in the previous round. For the decipher procedure (see Fig. 4.14), it is a complete reverse operation order for

obtaining the plaintext image. The computationally complexity of the decryption is the same as that of the encryption.

4.8 Performance and Security Analysis on Suggested Improvement

4.8.1 Resistance to Proposed CPA and KPA

CPA and KPA are considered as the main threat for the original algorithm from our analysis and Li's analysis [45]. An effective approach to resist these attacks is by making the encryption keys related to both the secret keys and the corresponding plaintext. In our improvement, the vector $RM(z)$ is dependent not only on the initial conditions (i.e., ${x_0}^1$, ${x_0}^2$) and system parameters (i.e., q^1 and q^2) of the proposed spatiotemporal chaotic system, but also on some pixel values of the plain image [$I(1)$, $I(2)$,..., $I(l)$]. However, the encryption vector TM or TN of the original encryption algorithm depends solely on the secret keys (i.e., μ, x_0) from the Logistic chaotic map. The details are shown in Fig. 4.14, which demonstrates that the whole encryption process can construct a relationship among encryption vectors, secret keys and the pending plaintext image. Moreover, the encryption turn ET is dependent on the values of the iteration of the spatiotemporal chaotic system (i.e., x_t^1 and x_t^2). These two values are related to the value of nir which can be changed for the new encryption. Therefore, the suggested improvement may avoid the KPA and CPA. Especially, the effect of this resistance is related to the number of the selected column(s) in an image I (nu is the number of the selected column(s)). Furthermore, as the suggested improvement has twice encryptions in one round, this mechanism dose also increase the difficulty of the analysis to the attackers.

4.8.2 Key Space and Sensitivity Analysis

The key space and key sensitivity of the proposed improved scheme are analyzed as follows:

- Key space: In the improvement, the secret keys includes system parameters $q^i \in (0, 1)$ and initial condition ${x_0}^i \in (0, 1)$ of the spatiotemporal chaotic system. Therefore, the secret keys include 4 parts (not including nu) which is no less than that of the original algorithm. Assume that the precision of a floating-point number is 10^{-14}, the size of the key space can be considered as $(10^{14} \times 10^{14}) \times (10^{14} \times 10^{14}) = 10^{56}$. This space is large and safe enough for ordinary applications.

- Key sensitivity: Key sensitivity is extremely crucial to any cryptosystem. This implies that even if there is a slight change in the cipher keys, the final ciphertext image is fully different. As the spatiotemporal chaotic system is used for producing the vector sequences ($RM(z)$, $CN(v)$), the sensitivity of this spatiotemporal chaotic system can assure that the produced vector sequences are totally different for any two different secret keys (q^i, ${x_0}^i$). To test the secret key sensitivity of the suggested improvement, a typical test method has been performed in the following steps:

  - Step 1: One 128×128 digital image (e.g., the image 'Lena') is encrypted by the secret keys in one round: x_0^1=0.41236, x_0^2=0.63297, q^1=0.34567, q^2=0.82194.
  - Step 2: Only the secret key ${x_0}^1$ is changed to 0.32984000000001 (i.e., others are the same as above except x_0^1), and the secret keys are used to encrypt the same image,
  - Step 3: The two ciphertext images which are encrypted by the two slightly different keys are compared and investigated.

  The result is plotted in Fig. 4.15. From this figure, we can find that most of the cipher pixels in these two cipher images are different, which shows the key sensitivity of the suggested improvement.

  Then, the one-bit-difference secret key ${x_0}^1$ is also utilized in the decryption processing, i.e., ${x_0}^1$=0.41236 is for the encryption and ${x_0}^1$=0.41236000000001 is for the decryption. The test result is shown in Fig. 4.15, which illuminates that the

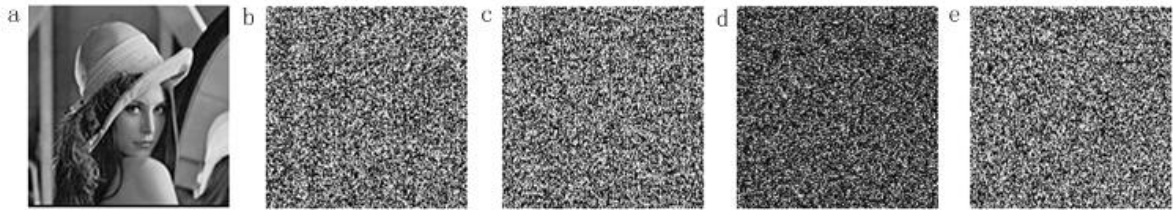

Figure. 4.15: Test for key sensitivity: (a) plaintext image, (b) ciphertext image using key ${x_0}^1$=0.41236, (c) ciphertext image using key ${x_0}^1$=0.41236000000001, (d) difference image of above two ciphertext images, (e) decrypted image using a slight change key ${x_0}^1$=0.41236000000001.

Table. 4.7: Key sensitivity at one round

Test image	Size	${x_0}^1$=0.41236000000001	${x_0}^2$=0.63297000000001
Lena	128×128	0.9946	0.9962
Cameraman	256×256	0.9962	0.9958
Test image	Size	q^1=0.34567000000001	q^2=0.82194000000001
Lena	128×128	0.9957	0.9962
Cameraman	256×256	0.9959	0.9960

tiny change in decryption key can make the decrypted image totally unrecognition, and the information about the original image can not be obtained. Table 4.7 is the corresponding test results of the key sensitivity for two gray-scale images of different sizes (128×128 and 256×256).

4.8.3 Statistical Analysis

The statistical analysis about the ciphertext is extremely crucial for any cryptosystem. If a cipher is an ideal cryptosystem, it should be robust against any statistical attack. Therefore, the following two statistical tests are used to analyze the suggested improvement:

- Histogram analysis: The *histogram* can reflect the distribution of pixels of an image effectively. The histograms of the plaintext image, corresponding ciphertext image

after one round of the encryption and decrypted image are plotted in Fig. 4.16. The test image is the gray-scale image of 'Lena'. It can be found that the histogram of the ciphertext image is significantly different from those of the plaintext image and decrypted image. It is changed greatly by the suggested improvement.

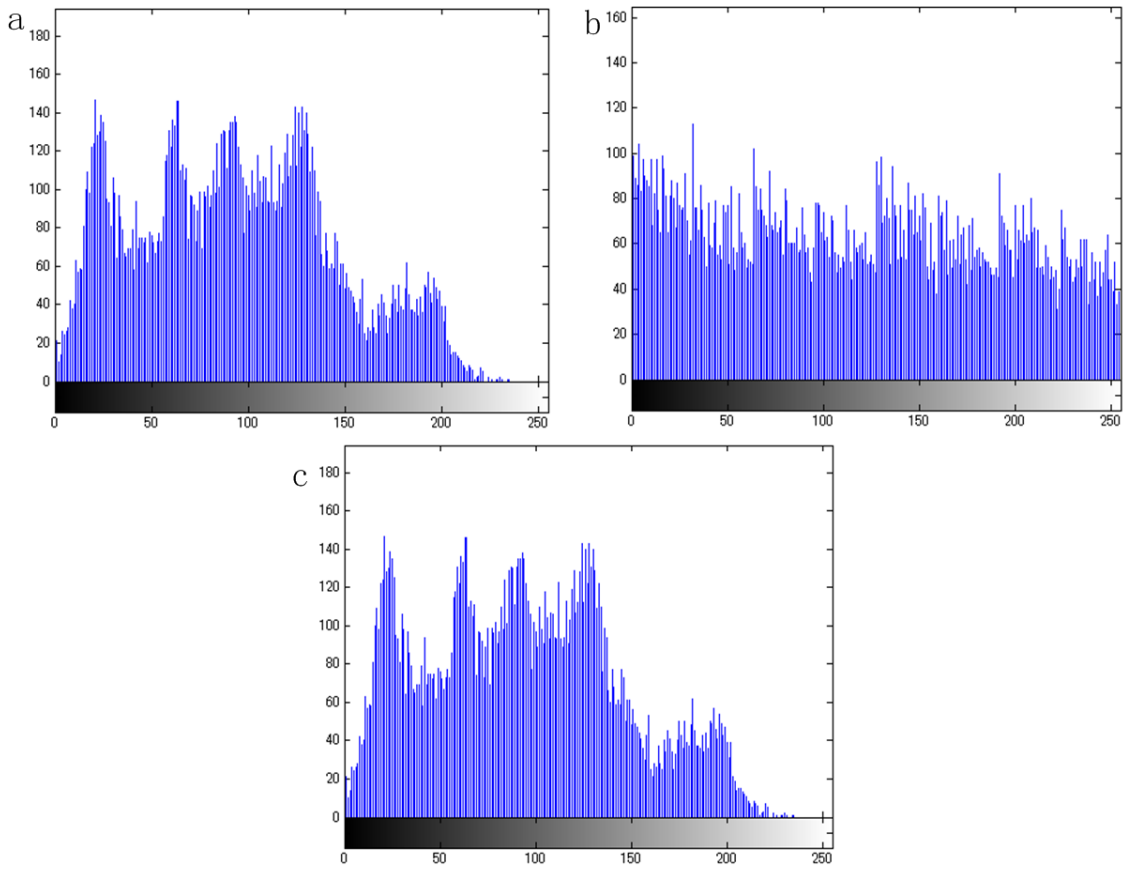

Figure. 4.16: Histogram of (a) plaintext image, (b) ciphertext image, (c) decrypted image about 'Lena'.

- Correlation analysis: For a ciphertext image, the confusion and diffusion properties can be verified by investigating the correlations of *adjacent pixels*. Therefore, to test the correlation between two horizontally ((x, y), $(x+1, y)$), two vertically ((x, y), $(x, y+1)$) and two diagonally ((x, y), $(x+1, y+1)$) adjacent pixels in the plaintext image and the corresponding ciphertext image, the following simulations are carried out: first of all, 2048 pairs of two adjacent pixels from the plaintext image

Table. 4.8: Correlation coefficients of two adjacent pixels in plaintext image and corresponding ciphertext image of 'Lena' (128×128) and 'Cameraman' (256×256)

	Plaintext image		Improved scheme	
	Lena	Cameraman	Lena	Cameraman
Horizontal	0.9480	0.9584	0.0248	0.0199
Vertical	0.8851	0.9350	-0.0094	0.0431
Diagonal	0.8546	0.9054	-0.0183	-0.0034

are randomly selected. Then, the *correlation coefficient* of each pair is calculated by the following probability formulas:

$$\begin{cases} E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i; \quad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \\ r_{xy} = \frac{E(x-E(x))(y-E(y))}{\sqrt{D(x)}\sqrt{D(y)}} \end{cases}, \tag{4.13}$$

where $E(x)$ and $D(x)$ are the expectation and variance, respectively. The test result on the correlation distribution of two horizontally adjacent pixels in the plaintext image and those in the ciphertext image are shown in Fig. 4.17. We can find that there is a strong correlation between adjacent pixels of each direction for the plaintext image. However, for the ciphertext image, this relationship is negligible, which implies that the suggested improvement has immediately decreased the correlation in the ciphertext image. Specially, the correlation coefficients are calculated and listed in Table 4.8 for demonstrating the effect of the encryption.

4.8.4 Information Entropy Analysis

About the *entropy* $H(m)$ of a message source m, it can be calculated by the formula: $H(m)=\sum_{i=0}^{2^N-1} P(m_i)\log_2(1/P(m_i))$, where $P(m_i)$ is the probability of symbol m_i, and especially, for a gray-scale image, $m_i \in \{0, 1, 2, \ldots, 255\}$. For any source emitting 2^8 symbols with equal probability, when the entropy is expressed in bits, the idea result is 8 corresponding to a truly random source. Nevertheless, as a practical symbol source seldom produces the random messages, the entropy in fact is less than 8. In particular, for

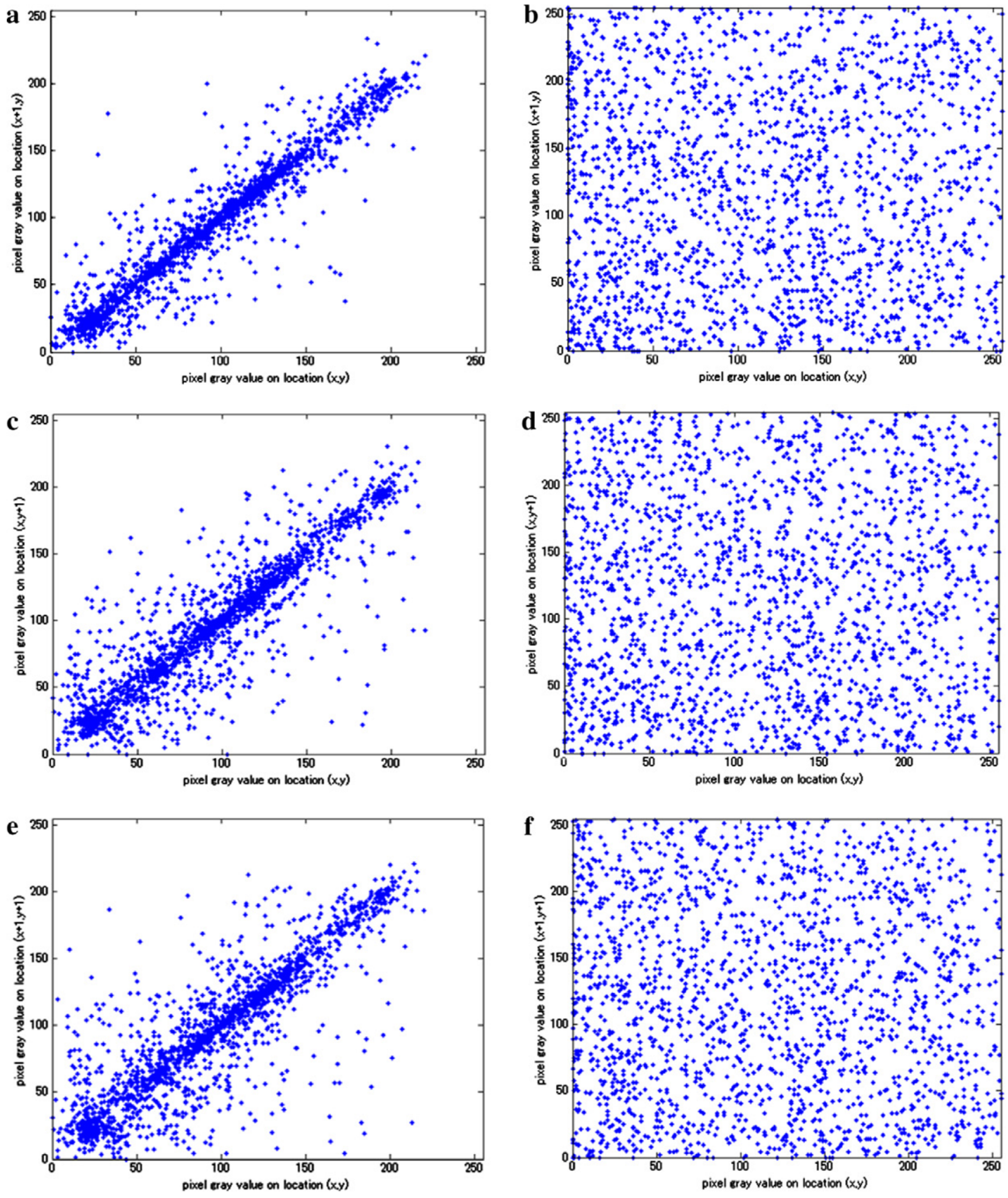

Figure. 4.17: Correlation of two adjacent pixels in plaintext image ('Lena') and corresponding ciphertext image: (a and b) Horizontal-direction, (c and d) Vertical-direction, (e and f) Diagonal-direction.

Table. 4.9: Results of information entropy

Test image	Size	Improved algorithm	Original algorithm
Lena	128×128	7.9577	7.9494
Cameraman	256×256	7.9719	7.9630

the encrypted messages, the ideal entropy should also be 8. If the output of a cipher emits symbols with the entropy less than 8, there exists certain degree of the predictability, which can be considered as a security problem. Therefore, for the suggested improvement, the entropies of the ciphertext images about 'Lena' and 'Cameraman' are measured by the above formula, and the final values are listed in Table 4.9.

The entropy values obtained from the suggested improvement are extremely close to the ideal value 8, which demonstrates that the corresponding ciphertext images approach to a pseudorandom source.

4.8.5 Comparison with the Original Algorithm

The suggested improvement is based on the original algorithm [85]. Some merits of the original algorithm are reserved. To overcome the defects of the original algorithm, we introduce the idea of the self-correlation to establish the connection between plaintext and encryption vectors. Consequently, the security of the suggested improvement may be higher than the original algorithm. In addition, the encryption effect of the improved scheme is also better than the original one.

- Comparison of the encryption effect: As one round of the improved scheme includes two-time of the encryption, the original scheme need to finish two rounds of the encryption for this comparison. The encryption results about the original algorithm and improved one are shown in Fig. 4.18. From this figure, it is found that the suggested improvement has a better encryption effect. This is mainly due to the fact that in each round, the original algorithm do the row permutation for scrambling the pixel location, and the column permutation is used for the encryption of the

pixel value. Therefore, the encryption of pixels is confined, e.g., the results in Table 4.10.

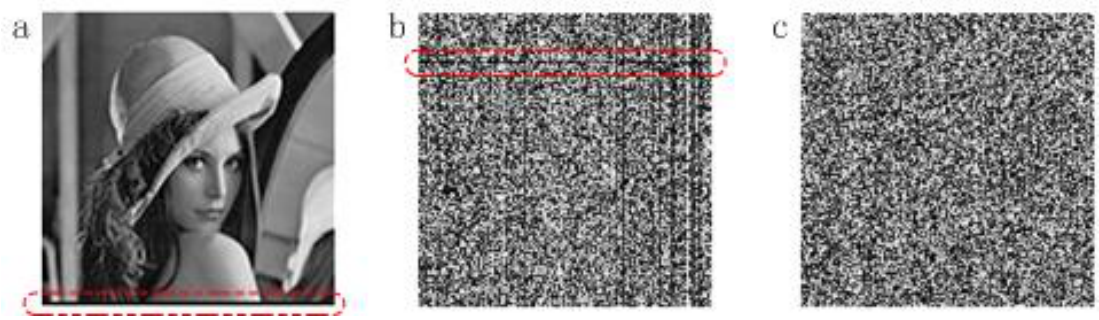

Figure. 4.18: Comparison between original encryption algorithm and improved one: (a) plain image, (b) original encryption algorithm, (c) improved scheme.

In Table 4.10, assume that there are three matrices in which the sizes of N are 4, 5 and 6, and the bits sequence in one row of the three matrices are '1010', '10101' and '101010', respectively. The corresponding numbers of the possible permutation for these three bits sequences are equal to $6<(2^4=16)$, $10<(2^5=32)$ and $20<(2^6=64)$. Moreover, if the bits sequence is '1000', the number of the possible permutation is equal to $4<6<(2^4=16)$. The difference between the permutation space and 2^N is defined as $Di(N)$. It can be found that $Di(N)$ can increase following the size N, e.g., $Di(4)=10$, $Di(5)=22$ and $Di(5)=44$. These analyses illustrate the potential security problem of the original algorithm. However, for the improved one, as the transposing operation is used, the scrambling of the pixel location and encryption of the pixel value can be achieved for the pixels in both the row and column. This can increase the security of the corresponding cipher effectively.

- Gray difference degree: the *gray difference* can be considered as an index for measuring the performance of the suggested improvement according to the original work [85]. It can be calculated according to the following formulas:

$$\begin{cases} GN = \frac{\sum[G(x,y)-G(x',y')]^2}{4};\ AN(GN(x,y)) = \frac{\sum_{x=2}^{M-1}\sum_{y=2}^{N-1}GN(x,y)}{(M-2)\times(N-2)} \\ GVD = \frac{AN'(GN(x,y))-AN(GN(x,y))}{AN'(GN(x,y))+AN(GN(x,y))} \end{cases}, \tag{4.14}$$

Table. 4.10: Permutation space of original algorithm for one row

Bit sequence	Permutation space				
1010	0101	0011	1100		
	0110	1001	***1010***		
10101	01011	00111	11001	01101	10011
	10101	01110	10110	11010	11100
101010	010110	011010	010101	011001	011100
	001110	100110	001101	100101	101100
	110010	***101010***	110001	101001	110100
	111000	100011	010011	001011	000111

Note: The significance of bold values implies that the permutation space includes the original bit sequence.

where $G(x, y)$ is the gray value of the pixel at (x, y). AN and AN' are the average neighborhoodgray difference of an image before and after the corresponding encryption, separately. Fig. 4.19 is the gray difference degree produced by the suggested improvement [1].

From this figure, it can be found that the value of the gray difference degree is extremely near to 1. The range of the values is limited in (0.9946, 0.9948], which presents the good performance of the suggested improvement. Moreover, the gray difference degree of the suggested improvement is also not lower than that of the original one.

4.8.6 Other Analysis

As mentioned above, the suggested improvement is based on the original algorithm. For the basic implementation, there are two kinds of operations for the whole process, i.e., x-or and shifting operation. Following we test and analyze the spent time of the

[1]The result is from 1 round of the encryption to 100 rounds of the encryption

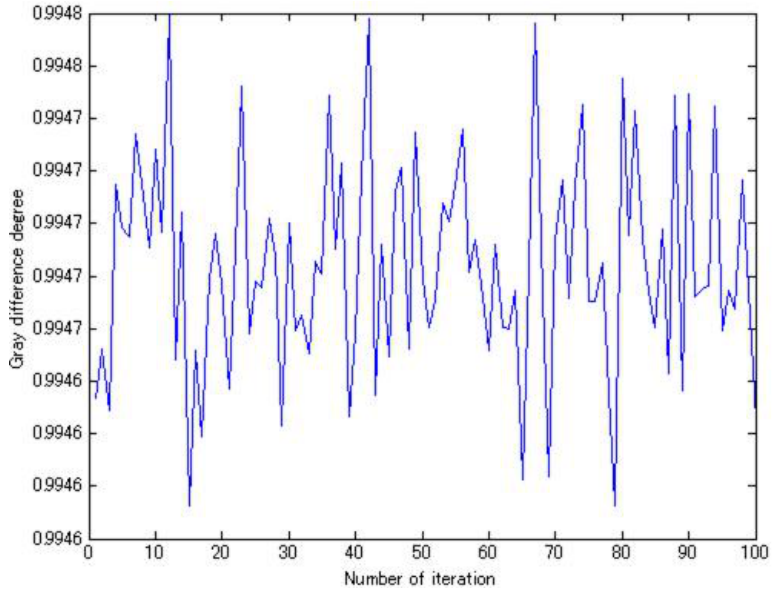

Figure. 4.19: Gray difference degree of improvement.

encryption of the suggested improvement which is based these two operations. Table 4.11 shows the corresponding spent time for encrypting some gray-scale images [84] with one round. The timing test for each image was executed five times, and the average of the times is presented. According to the simulation results in this table, it is found that the cost of the whole encryption can be accepted for the security. Note that all the simulation experiments were done by Matlab2009 running on Core 2 Duo CPU 1.40 GHz with 2.00 GB RAM.

4.9 Conclusions

In this chapter, the chosen-plaintext attack (CPA) and chosen-ciphertext attack (CCA) were presented to break a recently proposed image encryption algorithm based on chaos. If the size of the ciphertext image is $M \times M$, i.e., $M=N$ (e.g., 128×128 or 256×256), only 9 plaintext images/ciphertext images are used to reveal the vector TM and TN (or TM' and TN') according to our attack. Furthermore, for any M and N, a quantified comparison was made between our attack and Li and Lo's attack [45] which was also proposed to attack the original algorithm [85]. Specially, for the general images, our

Table. 4.11: Spent time for encrypting some gray-scale images with suggested improvement (unit: second).

Test images	Lena (128×128)	Baboon (128×128)	House (128×128)
Average time	0.2634	0.2658	0.2706
Peak time	0.270576	0.267742	0.277140
Minimum time	0.261198	0.264103	0.259949
Test images	Cameraman (256×256)	Barbara (256×256)	Elaine (256×256)
Average time	1.3839	1.3912	1.3798
Peak time	1.388215	1.414676	1.398585
Minimum time	1.379967	1.373380	1.365981

attack has a lower computational complexity than Li and Lo's attack. To overcome the drawbacks of the original algorithm, the suggestion on the improvement, which uses the idea of the self-correlation, was proposed. The simulation tests show that the suggested improvement may possess a higher security than the original one.

Algorithm 2 For revealing the column scrambling vector TN

for $L = 1$ to $s + 1$ ($s \in \{0, 1, \ldots, m\}$, $m \in \mathbb{Z}$) **do**
  Step1
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      $RCN_L(i, j) = 255$;
    end for
  end for
  Step2
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      for $g = 8(j-1)+1$ to $8j$ **do**
        $RCN'_L(i, g) = 1 \Leftarrow [RCN_L(i, j) \leftarrow Eq.(4.1)]$;
      end for
    end for
  end for
  Step3
  $Q \leftarrow M$
  if $(8N - L \times M) < M$ **then**
    $D \leftarrow (8N - L \times M)$;
  else
    $D \leftarrow M$
  end if
  for $k = (L-1) \times Q + 1$ to $(L-1) \times Q + D$ **do**
    $\{e(\omega)|e(\omega) \subseteq \{1, 2, 3, \ldots, M\}\} \leftarrow$ Choose any ω many x-coordinate(s) in $\{1, 2, 3, \ldots, M\}$ // $\omega \in \{1, 2, \ldots, D\}$;
    for $u = 1$ to ω **do**
      $RCN'_L(e(u), k) \leftarrow 0$;
    end for
  end for
  Step4
  for $i = 1$ to M **do**
    for $j = 1$ to N **do**
      for $g = 8(j-1)+1$ to $8j$ **do**
        $t \Leftarrow [RCN'_L(i, g) \leftarrow Eq.(4.2)]$;
      end for
      $RCN_L(i, j) \leftarrow t$;
    end for
  end for
end for

Chapter 5

Security Analysis of Randomized Arithmetic Codes

In this chapter, we discuss about the security of one kind of arithmetic coding based on the Markov model (ACMM). This discussion demonstrates that the ACMM is insecure. The security of the combination scheme of ACMM and Randomized Arithmetic Coding (RAC) is also analyzed.

5.1 Introduction

5.1.1 Research Background

As the content which we have provided in Chapter 2, with the development of the information processing technology, affording the compression and security is of significance as the increased use of the multimedia files in many applications such as the digital cameras and internet [36]. Specially, the compression removes the *redundancy* of the data by analyzing the statistics of the input. The size of the output is shortened compared with the size of the input. The typical compression algorithms are the *Huffman Coding* and *Arithmetic Coding* (AC) [78, 40, 44]. The security makes the adversary difficult to obtain the plaintext information without the permission. Therefore, there are three kinds of protection methods. In Chapters 3 and 4, we discussed about the analysis on the second protection method. In this Chapter, we focus on the analysis of the third protection

method, i.e., the encryption and compression are performed in one single step for acting on the multimedia data [36, 51, 78, 40, 44, 76, 22]. Compared with the traditional first-compression-then-encryption approach and first-encryption-then-compression approach, the major merit of joint compression-encryption hybrid method is that the goal of compression and encryption can be achieved simultaneously. This can simplify the design of the system for reducing the time and computation. Moreover, it makes the system flexible for the advanced *multimedia processing* [90].

5.1.2 Previous Work

For the third protection method, the compression based encryption is seen as one of the main choices. In general, the algorithm which incorporates the security into the compression can improve the efficiency for the practical applications. Two compression algorithms (i.e., the Huffman Coding and the Arithmetic Coding (AC)) are always considered to be modified for the encryption. However, according to the viewpoint of Witten et al. [77], the AC has the better compression ratio than the Huffman coding. Therefore, the AC is widely used in the recent multimedia compression standards (e.g., JPEG2000, H.264/AVC), which brings the AC based encryption into focus on a large scale.

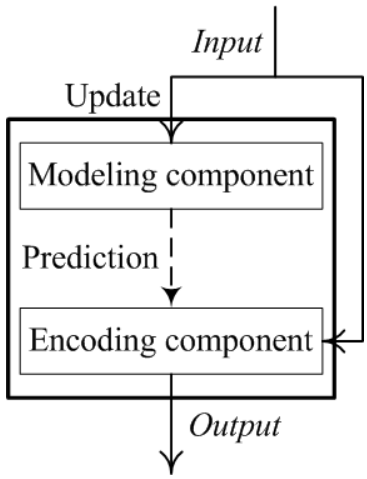

Figure. 5.1: Tow parts of compression: modeling component and encoding component.

Moreover, according to Witten et al.'s analysis [77], the compression is separated into

Table. 5.1: Some existing works on modified AC

Previous works	Abbreviation
Adaptive model based AC encryption [77]	AAC
Combination of model and coder based encryption [53]	CMCE
Randomized AC [36]	RAC
AC with key-based interval splitting [44]	AC-KIS
Markov model based AC encryption [31]	ACMM

Table. 5.2: Attacks on existing works of modified AC

Previous works	Corresponding attack
AAC	Chosen-plaintext attack [23]; Adaptive brute-force attack [51]
CMCE	Bergen/Hogan analysis [23] based attack [73]
RAC	Ciphertext-only attack [42]
AC-KIS	Adaptive chosen-plaintext attack [90]; Adaptive chosen-plaintext attack [68] Ciphertext-only attack [68]
ACMM	Our work

the *modeling component* and the *encoding component* (see Fig. 5.1), where the modeling component is used to provide, in any given context, a probability distribution for the forthcoming symbol. Hence, there are two main kinds of methods for incorporating the security into the AC, i.e., the *model based encryption* and *coder based encryption* [51]. In Table 5.1, some detailed works about the modified AC are presented. Moreover, the corresponding attacks on these works are listed in Table 5.2.

5.1.3 Our Contributions

In 2011, Duan et al. [31] proposed a randomized AC based on the first-order Markov model (ACMM). This algorithm can be seen as a model based randomized AC. The randomness is achieved by choosing the probability of the binary symbol from the Markov model. Specially, the used Markov model is expressed as the tree structure which has two orders, i.e., order-0 model and order-1 model. The order-0 model is the single symbol (i.e., 0 or 1) model without context which has two values corresponding to $\Pr[s_i{=}0]$ and $\Pr[s_i{=}1]$, while the order-1 model for the binary symbol has four values of conditional probabilities of the form $\Pr[s_i{=}0|s_{i-1}{=}0]$, $\Pr[s_i{=}1|s_{i-1}{=}0]$, $\Pr[s_i{=}0|s_{i-1}{=}1]$ and $\Pr[s_i{=}1|s_{i-1}{=}1]$ (see Fig. 5.3). In the current work, we formally define the ACMM scheme and its different security notions at first. Based on these security notions, we address some security issues about the ACMM as follows:

(1) We assume that the *same* pseudorandom bit sequence is generated for encrypting different plaintext messages, and the lower bound of the encoding interval is considered as the ciphertext message. With this setup, we establish that the ACMM is insecure under the chosen-plaintext attack (CPA) by revealing the used pseudorandom bit sequence.

(2) We reduce the assumption which implies that the *different* pseudorandom bit sequences are used to encrypt different plaintext messages. Under this condition, we show that the ACMM does not have indistinguishable encryptions under the ciphertext-only attack (COA). i.e., the ACMM does not have indistinguishable encryptions in the presence of an *eavesdropper* (see Katti et al.'s work [42] about this concept).

(3) We combine the ACMM with the RAC for obtaining a combined scheme which may have the higher security. However, the proposed analysis shows that even if these two steps of the AC (i.e., the modeling and encoding component) are encrypted simultaneously by the ACMM and RAC respectively, this combined scheme ACMM+RAC is still insecure under the COA.

Moreover, for confirming the proposed analyses, the simulation experiments are implemented. The corresponding simulation results (see Tables 5.5 and 5.6) accord with our analyses which demonstrate that both the ACMM and ACMM+RAC are insecure.

5.1.4 Organization of the Chapter

The chapter is organized as follows. In Section 5.2, we review the description of the ACMM and the corresponding drawbacks. In Section 5.3, we present the formal definition on ACMM and the related security notions. According to these definitions, we investigate the insecurity of ACMM under the CPA and COA. Section 5.4 presents the formal definition of ACMM+RAC. Moreover, the insecurity of ACMM+RAC under COA is discussed. Section 5.5 shows the simulation results of the proposed analyses on ACMM and ACMM+RAC which are used to confirm our analyses. The concluding remarks are drawn in the last section.

5.2 Randomized Markov Model Based Arithmetic Code and Its Drawbacks

5.2.1 Scheme Review

The ACMM [31] is a modified AC which makes use of the first-order Markov model to predict the encoding probability for each symbol s_i of the binary plaintext message $S{=}s_1s_2...s_N$ of length N. It introduces the *pseudorandom bit sequence* $q{=}q_1q_2...q_N$, generated by a *pseudorandom bit generator* (PRBG), for the encryption ($r{=}q$ which denotes the pseudorandom bit sequence in the original work [31].). This encryption is the permutation of the probability which is produced by the Markov model and used to encode the binary plaintext message S. Specially, the process for encrypting each symbol s_i is as follows (see Fig. 5.2),

(1) Generate the pseudorandom bit sequence q according to the PRBG and draw a pseudorandom bit q_i for the current symbol s_i, where $i{\in}[0,\ N{-}1]$.

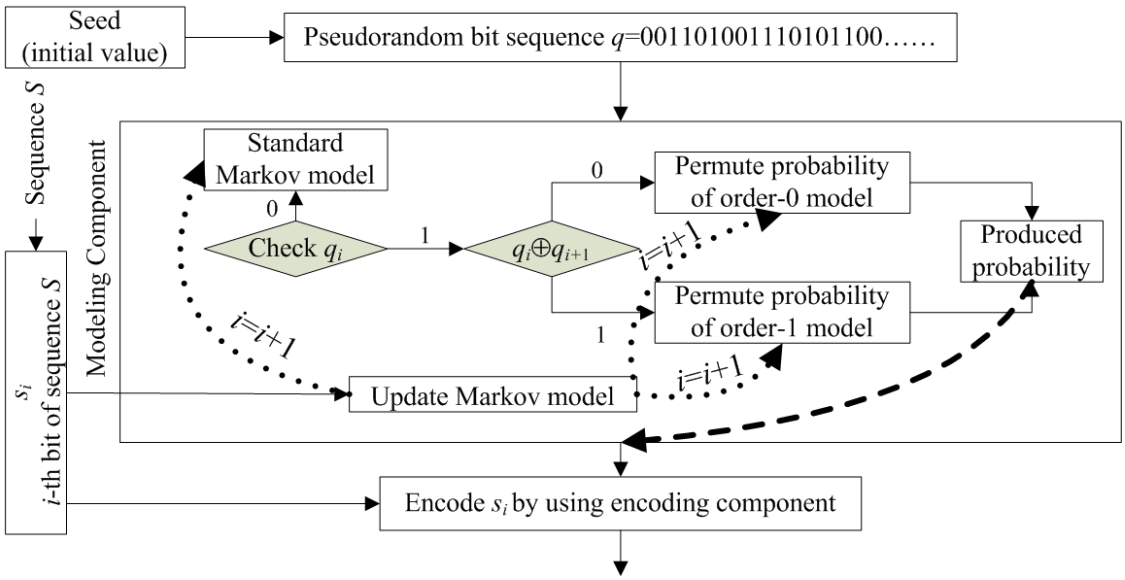

Figure. 5.2: Encryption procedure of each symbol s_i.

(2) Encode s_i with q_i. If q_i=0, s_i is encoded by the standard Markov model which is not encrypted [31]. While, if q_i=1, s_i is encoded by the scrambled Markov model as follows,

- Produce the value k_i=$q_i \oplus q_{i+1}$. If i=N–1, k_{N-1}=$q_{N-1} \oplus q_0$.
- If k_i=0, the probabilities of the order-0 (i.e., Pr[s_i=0] and Pr[s_i=1]) is permuted.
- If k_i=1, the probability of the order-1 (i.e., Pr[s_i=0|s_{i-1}=0/1] and Pr[s_i=1|s_{i-1} =0/1]) is permuted.

(3) Update the probability model according to the value of the symbol s_i.

Furthermore, the ACMM does not do the encryption on the encoding component. Then, the standard arithmetic coder (i.e., SAC) [53] can be used to encode the binary plaintext message S. This implies that it is different from the RAC [36] which does the encryption on the encoding process with the pseudorandom bit sequence q. Specially, according to the work [31], the initial model of the *Markov model* can be as Fig. 5.3, in which all the symbol counts (SC) are equal to 1. The detailed description can refer the

work [31].

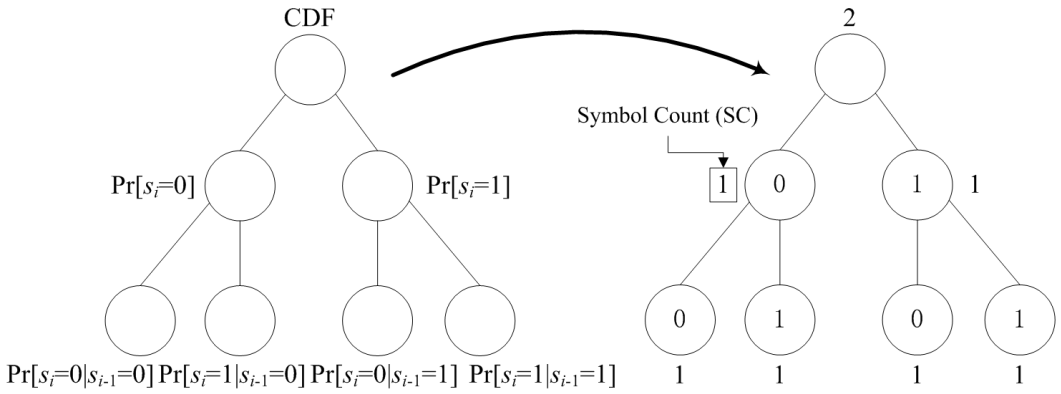

Figure. 5.3: Probability model and corresponding initial model.

5.2.2 Drawbacks

The encryption mechanism of the ACMM is the *randomized* Markov model controlled by the pseudorandom bit sequence q. According to the value of each pseudorandom bit q_i, the probability that arises from the Markov model can be decided to encode the i-th symbol s_i. However, two potential drawbacks exist in the ACMM, and have a significant impact on the security of the ciphertext C.

(1) The *initial model* in Fig. 5.3 is used to encode the 1-st symbol s_1 of the plaintext message S. However, probabilities from the order-0 model (i.e., Pr[s_1=0] and Pr[s_1=1]) correspond to the same SC (e.g., 1). This implies that no matter what pseudorandom bit q_1 is, the probability for encoding the 1-st symbol s_1 is 1/2, and the interval [0, 1) is divided into [0, 0.5) and [0.5, 1) for 0 and 1, respectively.

(2) Let $I(S)$ represent the interval of the plaintext message S. According to the value of the pseudorandom bit q_i (or $q_i \oplus q_{i+1}$), probabilities of the order-0 model or the order-1 model are permutated. However, the positions of the symbols 0 and 1 in the interval $I(S)$ are not scrambled. This implies that for an interval $[x, y)$, $I(S'0) := [x, z)$ and $I(S'1) := [z, y)$ for any pseudorandom bit q_i, where S' is a binary string of any length, x, y and z are real numbers between 0 and 1 or integers between

0 and a sufficiently large integer (e.g., 65535), and z is decided by the current pseudorandom bit q_i (see Fig. 5.4).

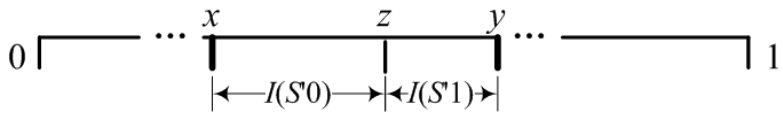

Figure. 5.4: Tow components of compression: modeling and encoding component.

5.3 Insecurity of ACMM

In this section, two different analyses are presented for the ACMM [31]. Note that the pseudorandom bit sequences play an important role in the security of the ACMM. Depending on the use of the pseudorandom bit sequences, we use two attack scenarios: one is the CPA, in which the same pseudorandom bit sequence is used to encrypt different plaintext messages (defined as **Case 1**), and the other one is the COA, in which different pseudorandom bit sequences are used to encrypt different plaintext messages (defined as **Case 2**). Specially, **Case 1** can be seen as the simplified version of **Case 2**. This implies that pseudorandom bit sequences in **Case 1** can be considered as the secret sequence. Our analysis begins with this simplified version. Before stating the proposed attacks, we first present some general assumptions for the convenience of the following analysis:

- The initial interval of encoding is the most commonly used interval, i.e., the interval [0, 1).
- For **Case 1**, the expression of the ciphertext C is seen as a real number (e.g., C=0.628). Specially, for a pair of plaintext messages (S_0 and S_1) of same length N, the expression precision of the ciphertexts ($C(S_0)$ and $C(S_1)$) is the same. For **Case 2**, the real number expression of the ciphertext C is defined as RC (e.g., RC=0.628). Moreover, the same rule as **Case 1** is used for RC.

The formal definition of ACMM and the related security notions are presented in the next subsection following the works [42, 43].

5.3.1 Formal Definition on ACMM and Corresponding Security Notions

5.3.1.1 Formal Definition of ACMM under Case 1

The ACMM can be viewed as a *symmetric-key encryption* algorithm which is a triple of probabilistic polynomial-time algorithm $\prod$=(Enc, Dec, Gen)[35, 43]. In this definition, Enc is the *encryption function*, Dec is the *decryption function* and Gen is the *probabilistic key-generation function*. Specially, under **Case 1**, every plaintext message S is encrypted by a fixed (same) pseudorandom bit sequence q obtained through the pseudorandom function F.

Definition 5.1. *Let n be the security parameter and N=$\ell(n)$ be a polynomial in n. Then $\prod$=(Enc, Dec, Gen) can be defined as follows:*

- *Gen (This step is executed once): On input 1^n, choose $k \leftarrow \{0, 1\}^n$ uniformly and randomly. Then, output k as the key. Moreover, choose $h \leftarrow \{0, 1\}^n$ uniformly and randomly for the pseudorandom function F.*

- *Enc: On input $k \in \{0, 1\}^n$, $h \in \{0, 1\}^n$ and a plaintext message $S \in \{0, 1\}^N$, output the set of probabilities $\bigcup_{i=1}^{N} Pr^a[s_i] :=$ ACMM$(F_k(h), S)$. Let $F_k(h)=q$. Then, generate the ciphertext $C :=$ SAC$(\bigcup_{i=1}^{N} Pr^a[s_i], S)$, where F is a pseudorandom function which outputs a pseudorandom bit sequence of length N. ACMM(q, S) implies performing the ACMM algorithm on the plaintext message S using the pseudorandom bit sequence q, and SAC(p, S) implies the standard arithmetic coding is used for the plaintext S with the probability set $Pr^a[S] := \{Pr^a[s_i] | i \in \{1, 2, \ldots, N\}\}$. $Pr^a[s_i]$ is the permuted probability (see notations in Subsection 5.3.2).*

- *Dec: On input k, h and the ciphertext C, the plaintext message S is decrypted by producing $F_k(h)$ and decoded using standard AC.*

In the subsequent sections we are going to use the notions of the negligible function and the *pseudorandom function* from the work [43]. The definitions of these two functions are as follows:

Definition 5.2 (Negligible Function)**.** *A function f is* negligible *if for every polynomial $p(\cdot)$ there exists an N such that for all integers $n>N$ it holds that $f(n)<1/p(n)$. Specially, this kind of function is denoted as* negl*.*

Definition 5.3 (Pseudorandom Function)**.** *Let F: $\{0, 1\}^*\times\{0, 1\}^*\rightarrow\{0, 1\}^*$ be an efficient, length-preserving, keyed function. We can say that F is a pseudorandom function if for all probabilistic polynomial-time distinguishers $\mathcal{D}$, there exists a negligible function negl such that:*

$$\left|\Pr[\mathcal{D}^{F_k(\cdot)}(1^n)=1]-\Pr[\mathcal{D}^{f(\cdot)}(1^n)=1]\right|\leq \mathsf{negl}(n),$$

where $k\leftarrow\{0, 1\}^n$ is chosen uniformly and randomly, and f is chosen uniformly and randomly from the set of functions mapping n-bit strings to n-bit strings.

5.3.1.2 Definition of Security under Case 1

Let n be a security parameter and $N=\ell(n)$ be a polynomial in n. We first define the *experiment* $\mathsf{Privk}^{\mathrm{cpa}}_{\mathcal{A},\prod}(n)$ for the ACMM (i.e., $\prod=(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Gen})$) as follow:

Experiment $\mathsf{Privk}^{\mathrm{cpa}}_{\mathcal{A},\prod}(n)$:

(1) A key k is produced by using $\mathsf{Gen}(1^n)$.

(2) The adversary $\mathcal{A}$ is given input 1^n and access to the encryption oracle, and output a pair of messages S_0 and S_1 of the same length N, where S_0 and $S_1\in\{0, 1\}^N$. Note that the adversary $\mathcal{A}$ can choose any pair of (S_0, S_1).

(3) The fixed $F_k(h)=q\in\{0, 1\}^N$ is used and a random bit $b\leftarrow\{0, 1\}$ is chosen. The ciphertext $C:=\mathsf{SAC}(\bigcup_{i=1}^{N}\Pr^a[s_i^b], S_b)$, $\bigcup_{i=1}^{N}\Pr^a[s_i^b]:=\mathsf{ACMM}(q, S_b)$ is generated by the challenger and given to the adversary $\mathcal{A}$. This C is called the challenge ciphertext.

(4) The adversary $\mathcal{A}$ is given access to the encryption oracle and request for the encryption of a polynomial number of chosen plaintexts.

(5) The adversary $\mathcal{A}$ outputs a bit b'. If $b'=b$, the output of this experiment is 1, otherwise, the output is 0. In case $\mathsf{Privk}^{\mathrm{cpa}}_{\mathcal{A},\prod}(n)=1$, we say that $\mathcal{A}$ succeeded.

Based on this experiment, the definition of the indistinguishability of ACMM encryption under the CPA is presented below.

Definition 5.4. *The fixed-length symmetric-key encryption algorithm $\prod$=(Enc, Dec, Gen) is CPA-secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function negl(·) which satisfies $|Pr[\mathsf{Privk}^{cpa}_{\mathcal{A},\prod}(n)=1]\text{-}1/2| \leq negl(n)$, where the probability is taken over the random coins used by the adversary $\mathcal{A}$, as well as the random coins used in the experiment.*

The above security definition (Definition 5.4) actually implies that the adversary $\mathcal{A}$ can not tell which message was used to achieve the ciphertext C except making a random guess even if $\mathcal{A}$ is given access to the encryption oracle. However, in the Subsection 5.3.2, we shall show that for ACMM, when the adversary $\mathcal{A}$ can have access to the encryption oracle $\mathsf{Enc}(\cdot)$, the used pseudorandom bit sequence can be revealed according to the ciphertexts from some suitable plaintexts. Therefore, for this condition, the adversary $\mathcal{A}$ always succeeds in the above game with probability 1. The details of this attack is presented in Subsection 5.3.2.

5.3.1.3 Formal Definition of ACMM and Its Security Notions under Case 2

For ACMM, the difference between **Case 1** and **Case 2** is the used pseudorandom bit sequences for the encryption. Note that in $\prod$=(Enc, Dec, Gen) (i.e., Under **Case 1**), the fixed pseudorandom bit sequence q is used to encrypt each plaintext message S. Under **Case 2**, for each encryption of a message S, a new pseudorandom bit sequence is used. Therefore, the formal definition of the ACMM under **Case 2**, i.e., $\prod'$=(Enc′, Dec′, Gen′), is similar to $\prod$=(Enc, Dec, Gen) and can be presented as follow:

Definition 5.5. *Let n be the security parameter and $N=\ell(n)$ be a polynomial in n. Then $\prod'$=(Enc′, Dec′, Gen′) can be defined as follows:*

- Gen' *(This step is executed once): On input* 1^n*, choose* $k \leftarrow \{0, 1\}^n$ *uniformly and randomly. Then, output* k *as the key.*

- Enc'*: On input* $k \in \{0, 1\}^n$ *and a plaintext message* $S \in \{0, 1\}^N$*, choose* $h \in \{0, 1\}^n$ *uniformly and randomly, and output the set of probabilities* $\bigcup_{i=1}^{N} Pr^a[s_i] := \mathsf{ACMM}(F_k(h), S)$*. Let* $F_k(h) = q$*. Then, generate the ciphertext* $VC := \{h, \mathsf{SAC}(\bigcup_{i=1}^{N} Pr^a[s_i], S)\}$*, where* F *is a pseudorandom function which outputs a pseudorandom bit sequence of length* N*.* $C := \mathsf{SAC}(\bigcup_{i=1}^{N} Pr^a[s_i], S)$*.* $\mathsf{ACMM}(q, S)$ *implies performing the* ACMM *algorithm on the plaintext message* S *using the pseudorandom bit sequence* q*, and* $\mathsf{SAC}(p, S)$ *implies the standard arithmetic coding is used for the plaintext* S *with the probability set* $Pr^a[S] := \{Pr^a[s_i] | i \in \{1, 2, \ldots, N\}\}$*.* $Pr^a[s_i]$ *is the permuted probability (see notations in Subsection 5.3.2).*

- Dec'*: On input* k *and the ciphertext* VC*, the plaintext message* S *is decrypted by producing* $F_k(h)$ *and decoded using standard AC.*

For **Case 2**, the definition of the COA indistinguishability experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$ is also similar to the experiment $\mathsf{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)$. However, in the $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$, the adversary $\mathcal{A}$ is not allowed to access the encryption oracle, and for each encryption of a message S, a new pseudorandom bit sequence is used. Therefore, we define the experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$ for the ACMM as follow:

Experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$:

(1) A key k is produced by using $\mathsf{Gen}'(1^n)$.

(2) The adversary $\mathcal{A}$ is given input 1^n and output a pair of messages S_0 and S_1 of the same length N, where S_0 and $S_1 \in \{0, 1\}^N$. Note that the adversary $\mathcal{A}$ can choose any pair of (S_0, S_1).

(3) The random bit $b \leftarrow \{0, 1\}$ and random $h \leftarrow \{0, 1\}^n$ are chosen. The ciphertext $VC := \{h, \mathsf{SAC}(\bigcup_{i=1}^{N} \mathrm{Pr}^a[s_i^b], S_b)\}$, $\bigcup_{i=1}^{N} \mathrm{Pr}^a[s_i^b] := \mathsf{ACMM}(q, S_b)$ is generated by the challenger and given to the adversary $\mathcal{A}$. This VC is called the challenge ciphertext.

(4) The adversary $\mathcal{A}$ outputs a bit b'. If $b'=b$, the output of this experiment is 1, otherwise, the output is 0. In case $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod}(n)=1$, we say that $\mathcal{A}$ succeeded.

Moreover, based on the above experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)$, the definition of the security under the COA is presented as follow:

Definition 5.6. *The fixed-length symmetric-key encryption algorithm $\prod'$=(Enc′, Dec′, Gen′) has indistinguishable encryptions under the COA if for any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function negl(·) which satisfies $|Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\prod'}(n)=1]-1/2|\leq negl(n)$, where the probability is taken over the random coins used by the adversary $\mathcal{A}$, as well as the random coins used in the experiment.*

According to Section 3.5 in the work [43], it can be known that $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)$ is a special case of $\mathsf{Privk}^{\mathrm{cpa}}_{\mathcal{A},\prod}(n)$ [43]. Specially, in this $\prod$, the $h\in\{0,1\}^n$ in the pseudorandom function $F_k(\cdot)$ is chosen uniformly and randomly for each encryption. Therefore, if an encryption algorithm does not have indistinguishable encryptions under the COA, it implies that this algorithm is also insecure under the CPA. In Subsection 5.3.3, the details of the COA for ACMM are presented based on the above definitions, which show that ACMM does not have indistinguishable encryptions under the COA.

5.3.1.4 Remark

For the above definition of ACMM under **Case 2** (i.e., Definition 5.5), the pseudorandom generator G [43] can be used to substitute the pseudorandom function F to generate the pseudorandom bit sequence. Under this situation, the scheme $\prod'$=(Enc′, Dec′, Gen′) can also be defined as follows:

- Gen′: On input 1^n, and for each encryption of a binary plaintext message S, choose $k\leftarrow\{0,1\}^n$ uniformly and randomly. Then, output k as the key corresponding S.
- Enc′: On input $k\in\{0,1\}^n$ and a plaintext message $S\in\{0,1\}^N$, output the set of probabilities $\bigcup_{i=1}^{N}\Pr^a[s_i]:=\mathsf{ACMM}(G(k), S)$. Let $G(k)=q$. Then, generate the ciphertext $C:=\mathsf{SAC}(\bigcup_{i=1}^{N}\Pr^a[s_i], S)$, where G is a pseudorandom generator which

outputs a pseudorandom bit sequence of length N. $\mathsf{ACMM}(q, S)$ implies performing the ACMM algorithm on the plaintext message S using the pseudorandom bit sequence q, and $\mathsf{SAC}(p, S)$ implies the standard arithmetic coding is used for the plaintext S with the probability set $\Pr^a[S]:=\{\Pr^a[s_i]|i\in\{1, 2,\ldots, N\}\}$. $\Pr^a[s_i]$ is the permuted probability.

- Dec': On input $k\in\{0, 1\}^n$ and the ciphertext C, the plaintext message S is decrypted by producing $G(k)$ and decoded using standard AC.

For the experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)$ under the COA, it is also suitable for the above *modified definition* of the scheme $\prod'=(\mathsf{Enc}', \mathsf{Dec}', \mathsf{Gen}')$. There is no difference for the analysis of ACMM no matter whether the pseudorandom function or the pseudorandom generator is used. In the following analysis of ACMM, the pseudorandom function is used to generate the pseudorandom bit sequences.

5.3.2 Security Analysis of ACMM under *CPA*

In this subsection, we show that the ACMM is not $\mathsf{CPA\text{-}secure}$ by revealing the used pseudorandom bit sequence q that is used to encrypt different plaintexts. Before presenting the details of this attack, we state the following assumption:

- If the current interval $I(S)$ of the plaintext message S is $[C(S), C(S)+E(S))$, where $E(S)$ is the product of probabilities of the whole symbols in S (i.e., $E(S)=\prod_{i=1}^{N}\Pr^a[s_i]$), the endpoint $C(S)$ is stored as the ciphertext C.

5.3.2.1 Notations and Properties

We first present some notations and results that are important for our proposed attack.

- $n^i(0)$, $n^i(1)$: the current SC of order-0 model after updating the i-th symbol in the Markov model.

- ${n_0}^i(0)$, ${n_0}^i(1)$, ${n_1}^i(0)$, ${n_1}^i(1)$: the current SC of order-1 model after updating the i-th symbol in the Markov model.
- $\Pr[s_i{=}0]$, $\Pr[s_i{=}1]$: the *order-0 probabilities* of the Markov model for encoding the i-th symbol s_i.
- $\Pr[s_i{=}0|s_{i-1}{=}0]$, $\Pr[s_i{=}1|s_{i-1}{=}0]$, $\Pr[s_i{=}0|s_{i-1}{=}1]$, $\Pr[s_i{=}1|s_{i-1}{=}1]$: the *order-1 probabilities* of the Markov model for encoding the i-th symbol s_i.
- $\Pr^a[s_i{=}0]$, $\Pr^a[s_i{=}1]$: the permuted probabilities for encoding the symbol s_i. They belong to the set $\{\Pr[s_i{=}0], \Pr[s_i{=}1], \Pr[s_i{=}0|s_{i-1}{=}0], \Pr[s_i{=}1|s_{i-1}{=}0], \Pr[s_i{=}0|s_{i-1}{=}1], \Pr[s_i{=}1|s_{i-1}{=}1]\}$.

Definition 5.7. *For the initial encoding interval as [0, 1), when the i-th symbol s_i needs to be encoded, the probability in the Markov model can be expressed as*:

$$\begin{cases} \begin{cases} \Pr[s_i=0] = \frac{n^{i-1}(0)}{n^{i-1}(0)+n^{i-1}(1)} \\ \Pr[s_i=1] = \frac{n^{i-1}(1)}{n^{i-1}(0)+n^{i-1}(1)} \end{cases} \\ \begin{cases} \Pr[s_i=0|s_{i-1}=0] = \frac{n_0^{i-1}(0)}{n_0^{i-1}(1)+n_0^{i-1}(0)} \\ \Pr[s_i=1|s_{i-1}=0] = \frac{n_0^{i-1}(1)}{n_0^{i-1}(1)+n_0^{i-1}(0)} \end{cases} \\ \begin{cases} \Pr[s_i=0|s_{i-1}=1] = \frac{n_1^{i-1}(0)}{n_1^{i-1}(1)+n_1^{i-1}(0)} \\ \Pr[s_i=1|s_{i-1}=1] = \frac{n_1^{i-1}(1)}{n_1^{i-1}(1)+n_1^{i-1}(0)} \end{cases} \end{cases}, \tag{5.1}$$

where $\Pr[s_i{=}0]{+}\Pr[s_i{=}1]{=}1$, $\Pr[s_i{=}0|s_{i-1}{=}0]{+}\Pr[s_i{=}1|s_{i-1}{=}0]{=}1$ *and* $\Pr[s_i{=}0|s_{i-1}{=}1]{+}\Pr[s_i{=}1|s_{i-1}{=}1]{=}1$.

Lemma 5.1. *Let us consider the binary plaintext message $S{=}s_1s_2...s_{N-1}s_N{=}00...01$ of length N, where $N{\geq}2$. Then, during the encryption of $s_N{=}1$, $Pr[s_N{=}d_1]{\neq}Pr[s_N{=}d_2|s_{N-1}{=}0]$, where d_1, $d_2{\in}\{0,1\}$.*

Proof. See appendix A for the proof. □

Lemma 5.2. *Let the two binary plaintext messages $S_1{=}s_1s_2...s_m$ and $S_2{=}{s'}_1{s'}_2...{s'}_m{s'}_{m+1}$ be encrypted by using the same pseudorandom bit sequence q, where $s_i{=}{s'}_i$, $i{\in}\{1,...,m\}$. Then, $C(S_1){\leq}C(S_2)$. The equality is achieved when ${s'}_{m+1}{=}0$.*

Proof. This result follows from the fact that $I(S_2) \subseteq I(S_1)$. Specially, if $s'_{m+1}=0$, according to the above assumption on the representation of the ciphertext corresponding to the plaintext interval, we have, $C(S_1)=C(S_2)$. □

Based on *Lemma* 5.2 and the definition of arithmetic function in the work [32] (i.e., Eq. (3) in the work [32]), the ACMM can be presented as a formula.

Theorem 5.1. *Let us consider two plaintext messages* $S_1=s_1...s_m$ *and* $S_2=S_1 s_{m+1}$, *where* $m \geq 1$. *Then, the encoder of the* ACMM *can be described through Eq. (5.2):*

$$C(S_2) = C(S_1) + s_{m+1} \times \Pr^a[s_{m+1} = 0] \times E(S_1), \tag{5.2}$$

where $E(S_1)$ *is given by Eq. (5.3):*

$$E(S_1) = \prod_{i=1}^{m} (\Pr^a[s_i = 1])^{s_i} \times (\Pr^a[s_i = 0])^{1-s_i}, \tag{5.3}$$

In particular, when $m=1$, $Pr[s_0=0]=Pr[s_0=1]=0.5$. $C(S_1)=s_0 \times 0.5$.

Proof. According to *Lemma* 5.2, $C(S_1) \leq C(S_2)$, and if $s_{m+1}=0$, $C(S_1)=C(S_2)$. Hence, we have $C(S_1 1)$–$C(S_1 0)=E(S_1 0)$. This implies that $C(S_2)=C(S_1)+s_{m+1} \times E(S_1 0)$. Specially, as $E(S_1 0)$ is the product of the whole probabilities corresponding to the symbols of $S_1 0$, $E(S_1 0)=E(S_1) \times \Pr^a[s_{m+1}=0]=\Pr^a[s_1=t_1] \times \Pr^a[s_2=t_2]... \times \Pr^a[s_N=t_m] \times \Pr^a[s_{m+1}=0]$, where $t_1, t_2,...,t_m \in \{0, 1\}$. This implies that the above equations can be achieved. When $m=1$, as $\Pr[s_0= 0] = \Pr[s_0= 1] = 0.5$, $C(S_1)=s_0 \times 0.5$. □

5.3.2.2 Method of Attack

The security of ACMM depends on the encryption of the Markov model by the pseudorandom bit sequence q. Therefore, the proposed attack should try to reveal the probability for encoding each symbol at first, and then, unveil the corresponding pseudorandom bit q_i.

$\Pr^a[s_i=0]$ Recovery: Let the chosen plaintext messages, each of length N, be S_1=100...0, S_2=010...0, ..., S_N=000...1. According to the Theorem 5.1, the corresponding ciphertext

is given by $C(S_i)$=0+s_i×$\Pr^a[s_i=0]$×$E(S_j)$=s_i×$\Pr^a[s_i=0]$×$\prod_{k=0}^{i-1}\Pr^a[s_k=0]$. Therefore, after S_1, S_2,..., S_N are encoded by the ACMM, the corresponding ciphertexts $C(S_1)$, $C(S_2)$,..., $C(S_N)$ are expressed as follows:

$$C(S_1) = 0.5,\ C(S_2) = 0.5 \times \Pr^a[s_2 = 0],\ C(S_3) = 0.5 \times \Pr^a[s_2 = 0] \times \Pr^a[s_3 = 0], ...,\ C(S_N) = 0.5 \times \Pr^a[s_2 = 0] \cdots \times \Pr^a[s_N = 0].$$

Based on the relationship among $C(S_1)$, $C(S_2)$,..., $C(S_N)$, we can compute $\Pr^a[s_2=0]$, $\Pr^a[s_3=0]$,..., $\Pr^a[s_N=0]$ by using Eq. (5.4).

$$\Pr^a[s_i = 0] = \frac{C(S_i)}{C(S_{i-1})}, \quad i \in \{2, 3, ..., N\}. \tag{5.4}$$

Pseudorandom Bit Recovery: According to the *Lemma* 5.1, it can be found that $\forall i\in\{1, 2,...,N\}$, the revealed $\Pr^a[s_i=0]$ belongs to either the set $\{\Pr[s_i=0], \Pr[s_i=1]\}$ or the set $\{\Pr[s_i=0|s_{i-1}=0], \Pr[s_i=1|s_{i-1}=0]\}$. To estimate the order of $\Pr^a[s_i=0]$ which is used to recover the pseudorandom bit q_i, a Detector $f_{det}(\cdot, \cdot, \cdot)$ is defined as follow:

Definition 5.8. *For an adversary $\mathcal{A}$ who obtains the set of probabilities $\{Pr^a[s_i=0]|i\in\{1, 2, ..., N\}\}$, a Detector is a function $f_{det}(\cdot,\cdot,\cdot)$ which is used to reveal the pseudorandom bit sequence q. This function can be presented as $(q, \{CM_i|i\in\{1,2,...,N\}) = f_{det}(S_{azp}, IM, \{Pr^a[s_i=0]|i\in\{1,2,...N\}\})$, where IM is the initial model as in Fig. 5.3, S_{azp}= 00...0 is the all zero plaintext, and CM_i is the updated Markov model for encoding the i-th symbol.*

According to the definition of the Detector, if the adversary $\mathcal{A}$ obtains the set of probabilities $\{\Pr^a[s_i=0]|i\in\{1, 2,...,N\}\}$, he/she can reveal the pseudorandom bit sequence q through this Detector. In fact, the Detector can be seen as the constructor of the updated Markov model for each symbol s_i. When the pseudorandom bit q_i is revealed, the corresponding Markov model can also be updated.

Fig. 5.5 is an example on the work principle of the Detector. Suppose that the first two symbols are 00. The Markov model is updated and output after the 2-nd symbol 0 is encoded. Then, the revealed $\Pr^a[s_2=0]$ is compared with the probabilities $\Pr[s_2=0]$, $\Pr[s_2=1]$, $\Pr[s_2=0|s_1=0]$, and $\Pr[s_2=1|s_1=0]$ in this model to decide the used

pseudorandom bit q_2 (see Fig. 5.5). The construction of every model CM_i reveals each pseudorandom bit q_i. Table 5.3 lists the revealed pseudorandom bit q_i and possible q_{i+1} according to the $\Pr^a[s_i{=}0]$ and the probabilities $\Pr[s_i{=}0]$, $\Pr[s_i{=}1]$, $\Pr[s_i{=}0|s_{i-1}{=}0]$, and $\Pr[s_i{=}1|s_{i-1}{=}0]$ in the Markov model.

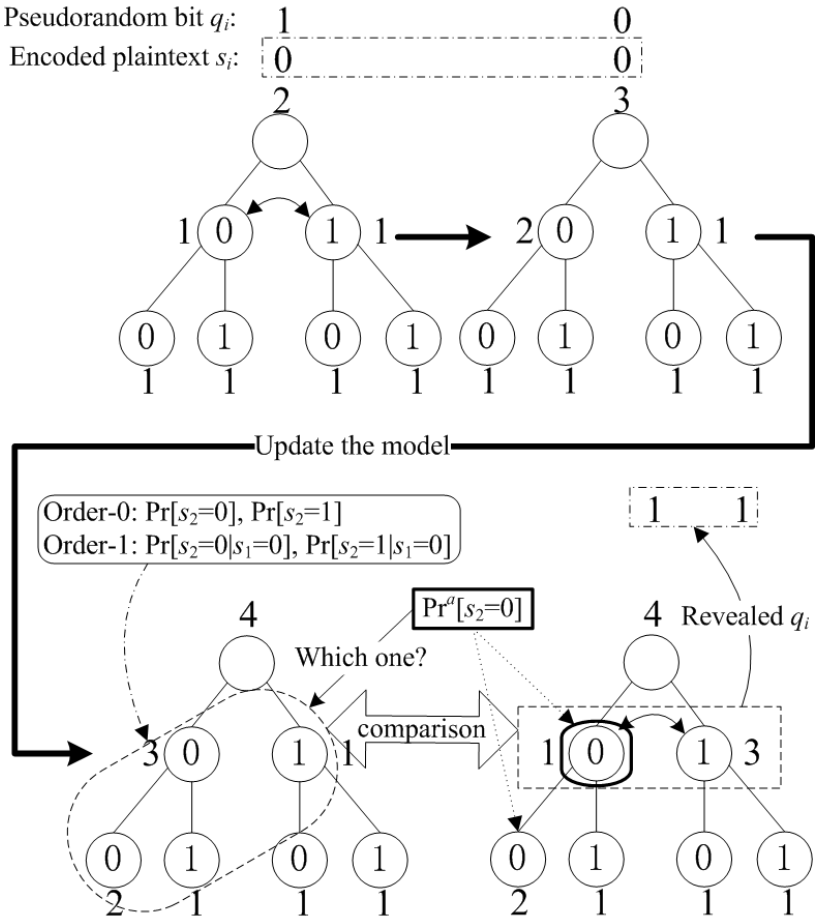

Figure. 5.5: Example of pseudorandom bit recovery (S=0010...0), assume that q_1 and q_2 have been revealed.

Specially, in the Markov model, if $\Pr[s_i{=}0]{=}\Pr[s_i{=}1]$ after the (i–1)-th symbol 0 is encoded (suppose that $\Pr^a[s_i{=}0]{=}\Pr[s_i{=}0]$), the pseudorandom bit q_i can be 0 or 1. However, if q_i=0 and there has existed sequence '00' in front of the (i–1)-th symbol 0, the coder should use the order-1 probability to encode the (i-1)-th symbol 0. Hence, according to this rule, we can ensure that q_i=1 when $\Pr^a[s_i{=}0]{=}\Pr[s_i{=}0]$. Moreover, if

Table. 5.3: Four states of pseudorandom bit recovery

	$\Pr^a[s_i{=}0]$	q_i	q_{i+1}
$\Pr[s_i{=}0]$	$\Pr^a[s_i{=}0]{=}\Pr[s_i{=}0]$	0	0/1
$\Pr[s_i{=}1]$	$\Pr^a[s_i{=}0]{=}\Pr[s_i{=}1]$	1	1
$\Pr[s_i{=}0\|s_{i-1}{=}0]$	$\Pr^a[s_i{=}0]{=}\Pr[s_i{=}0\|s_{i-1}{=}0]$	0	0/1
$\Pr[s_i{=}1\|s_{i-1}{=}0]$	$\Pr^a[s_i{=}0]{=}\Pr[s_i{=}1\|s_{i-1}{=}0]$	1	0

$\Pr[s_i{=}0|s_{i-1}{=}0]{=}\Pr[s_i{=}1|s_{i-1}{=}0]$ after the (i–1)-th symbol 0 is encoded (suppose that $\Pr^a[s_i{=}0]{=}\Pr[s_i{=}0|s_{i-1}{=}0]$), the pseudorandom bit q_i can also be 0 or 1. Under this situation, we can check that whether there is an 'escape symbol' which is used in the *standard Markov model* for telling the decoder to use order-1 probability to decode. If there exists such an 'escape symbol', the encoder use the standard Markov model (i.e., $q_i{=}0$), otherwise, $q_i{=}1$. Algorithm 3 is used to describe the revealing process for the whole pseudorandom bit sequence q.

5.3.2.3 Attack Complexity

Based on the method of the proposed CPA as the above description, we state the following propositions on the data complexity and time complexity.

Proposition 5.1. *For a binary sequence $S{=}s_1s_2\ldots s_N$ of length N, if the length of the corresponding pseudorandom bit sequence q is also N, the data complexity of the CPA is $N{+}1$ chosen plaintext messages.*

Proof. This proof is immediate from the description of the attack processes in Subsection 5.3.2. □

Proposition 5.2. *If the encryption for one binary symbol s_i is considered as one computation, the time complexity of this proposed attack is $\mathcal{O}(N^2)$.*

Proof. Two steps of the proposed attack should be considered. Specially, the computation load of the $\Pr^a[s_i{=}0]$ recovery is N^2 (more accurately, the computation load is

Algorithm 3 Recovery of the Pseudorandom Bit Sequence q

Input: S_{azp}=$s_1 s_2 ... s_N$=00...0, IM, {0.5, $\Pr^a[s_2=0]$, $\Pr^a[s_3=0]$,..., $\Pr^a[s_N=0]$}
Output:q,{CM_1,CM_2,...,CM_N}

1: $z \leftarrow S_{azp}$;
2: ES$\leftarrow$ACMM(S_{azp}); /* Returning escape symbol of q */
3: **for** $g = 1$ to N **do**
4: **if** g=1 **then**
5: (q_g, CM_g)$\leftarrow f_{det}(z(g)$, IM, 0.5);
6: **else**
7: (q_g, CM_g)$\leftarrow f_{det}(z(g)$, CM_{g-1}, $\Pr^a[s_g=0]$); /* $q_g \in$ {0, 1, 2}. q_g=2 implies that q_g can be 0 or 1. */
8: **end if**
9: **if** q_g=2 **then**
10: **if** $\Pr^a[s_g=0]$=Pr[s_g=0]=Pr[s_g=1] and $g \geq 2$ **then**
11: **if** $g < N$ **then**
12: (q_g, q_{g+1})=(1, 1);
13: **else if** g=N **then**
14: (q_g, q_0)=(1, 1);
15: **end if**
16: **else if** $\Pr^a[s_g=0]$=Pr[s_g=0|s_{g-1}=0]=Pr[s_g=1|s_{g-1}=0] **then**
17: escape symbol$\leftarrow$ES(s_g);
18: **if** $g < N$ **then**
19: f_{check}(escape symbol)=1?(q_g=0):((q_g, q_{g+1})= (1, 0)); /* Checking escape symbol for recovering q_g */
20: **else if** g=N **then**
21: f_{check}(escape symbol)=1?(q_g=0):((q_g, q_0)=(1, 0)); /* Checking escape symbol for recovering q_g */
22: **end if**
23: **end if**
24: **end if**
25: **end for**
26: **return** q, {CM_1, CM_2,..., CM_N};

$N \times (N\text{-}1)$). For the pseudorandom bit recovery, the computation load is N. If there is an extra encryption for obtaining the escape symbol, the corresponding computation load is $2 \times N$. Therefore, the total time complexity is $\mathcal{O}(N^2+N)$ (or $\mathcal{O}(N^2+2 \times N)$) which can be simplified to $\mathcal{O}(N^2)$. □

5.3.3 Security Analysis of ACMM under *COA*

In Subsection 5.3.2, the CPA is proposed for revealing the fixed pseudorandom bit sequence q used in ACMM. In this section, the security of the ACMM is explored if the different pseudorandom bit sequences are used to encrypt the different binary plaintext

messages. The analysis proves that the ACMM does not have indistinguishable encryptions under the COA.

For the following analyses, two plaintext messages of the same length N are defined as $0x_1$ and $1x_2$, where x_1 and $x_2 \in \{0, 1\}^{N-1}$. Moreover, based on the initial model as Fig. 5.3, the interval [0, 1) is divided into the subinterval as J_1=[0, 0.5) and J_2=[0.5, 1).

Lemma 5.3. *Let $S_0 = 0x_1$ and $S_1 = 1x_2$. Generate a key $k \leftarrow \{0,1\}^n$ uniformly and randomly, choose $h \leftarrow \{0, 1\}^n$ uniformly at random and select a random bit $b \leftarrow \{0, 1\}$. Let $q = F_k(h)$ be the pseudorandom bit sequence of length N. Generate the ciphertext $VC := \{h,$ $\mathsf{SAC}(\bigcup_{i=1}^{N} Pr^a[s_i^b],\ S_b)\}$, $\bigcup_{i=1}^{N} \mathsf{Pr}^a[s_i^b] := \mathsf{ACMM}(q,\ S_b)$. Specially, $C := \mathsf{SAC}(\bigcup_{i=1}^{N} Pr^a[s_i^b],\ S_b)$. Then, generate the real number RC according to C. If $RC \in J_1$, $Pr[S_b = S_0 | RC \in J_1] = 1$, and if $RC \in J_2$, $Pr[S_b = S_1 | RC \in J_2] = 1$. Therefore, $Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\prod'}(n) = 1 | RC \in J_1] = 1$ and $Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\prod'}(n) = 1 | RC \in J_2] = 1$.*

Proof. See appendix B for the proof. □

According to *Lemma* 5.3, the following theorem can be achieved.

Theorem 5.2. *The $\prod'$ does not have indistinguishable encryptions under the COA.*

Proof. A distinguisher $\mathcal{D}$ is constructed for the adversary $\mathcal{A}$ who uses it in the experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)$. The challenge ciphertext is denoted as $VC := \{h,\ \mathsf{SAC}(\bigcup_{i=1}^{N} \mathrm{Pr}^a[s_i^b],\ S_b)\}$, $\bigcup_{i=1}^{N} \mathrm{Pr}^a[s_i^b] := \mathsf{ACMM}(q,\ S_b)$. Specially, $C := \mathsf{SAC}(\bigcup_{i=1}^{N} \mathrm{Pr}^a[s_i^b],\ S_b)$. Generate RC according to C. Define the intervals as CI_0=[0, 1/2) and CI_1=[1/2, 1), respectively. The distinguisher $\mathcal{D}$ can be described as:

Distinguisher $\mathcal{D}$:

- If the value of RC, which is from the challenge ciphertext VC, is in CI_0, S_0 is used to generate VC. The adversary $\mathcal{A}$ outputs b'=0.
- If the value of RC, which is from the challenge ciphertext VC, is in CI_1, S_1 is used to generate VC. The adversary $\mathcal{A}$ outputs b'=1.

It can be denoted that if $|\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1]-1/2| \not\leqslant \mathsf{negl}(n)$, the ACMM does not have indistinguishable encryptions under the COA. In the following, this result can be shown.

$$\begin{aligned}&\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1] = \Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1|RC\in J_1]\\ &\times \Pr[RC\in J_1] + \Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1|RC\in J_2] \quad ,\\ &\times \Pr[RC\in J_2]\end{aligned}$$

From *Lemma* 5.3, i.e., $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1|RC\in J_1]=1$ and $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1|RC\in J_2]=1$. The above equation is simplified to

$$\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1] = \Pr[RC\in J_1] + \Pr[RC\in J_2].$$

As $\Pr[RC\in J_1]+\Pr[RC\in J_2]=1$, we can obtain that $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)=1]=1=1/2+1/2 >1/2$, which demonstrates that the ACMM does not have indistinguishable encryptions under the COA. □

5.3.4 Extension Consideration

For the ACMM, if the initial model is not as the Fig. 5.3 (e.g., in the initial model, $n^0(0)=3$, $n^0(1)=2$, ${n_0}^0(0)=2$, ${n_0}^0(1)=1$, ${n_1}^0(0)=2$, ${n_1}^0(1)=1$), the ACMM is also insecure under the proposed attack. This is due to the fact that the proposed attack is not based on the initial model as Fig. 5.3.

Specially, for **Case 1**, according to the process of the attack, the ciphertexts $C(S_1)$, $C(S_2)$,..., $C(S_N)$ can be obtained in the first step (i.e., $\Pr^a[s_i=0]$ Recovery) which are as follows:

$$\begin{aligned}&C(S_1) = \Pr^a[s_1=0],\ C(S_2) = \Pr^a[s_1=0]\times\Pr^a[s_2=0],\\ &C(S_3) = \Pr^a[s_1=0]\times\Pr^a[s_2=0]\times\Pr^a[s_3=0],\ldots,\\ &C(S_N) = \Pr^a[s_1=0]\times\Pr^a[s_2=0]\cdots\times\Pr^a[s_N=0].\end{aligned}$$

where $\Pr^a[s_1=0]$ is the permuted probability from the initial model. It can be found that all the permuted probabilities $\{\Pr^a[s_1=0], \Pr^a[s_2=0], \ldots, \Pr^a[s_N=0]\}$ are revealed by Eq. (5.4), and can be used in the second step (i.e., Pseudorandom Bit Recovery) for revealing the pseudorandom bit sequence q. Specially, if $n^0(0)\neq n^0(1)$, the pseudorandom bit q_1 can be directly revealed, which is different from the condition that $n^0(0)=n^0(1)$.

For **Case 2**, if the initial model is known to the adversary $\mathcal{A}$, the interval [0, 1) can be divided into the subintervals as J_1=[0, $\Pr^a[s_1{=}0]$) and J_2=[$\Pr^a[s_1{=}0]$, 1) according to the SC of the order-0 model. Then, the same analysis process, which is presented in the above section, is used. As $\Pr[RC{\in}J_1]{+}\Pr[RC{\in}J_2]{=}1$, it can be found that the $\prod'$ also does not have indistinguishable encryptions under the COA.

5.4 Insecurity of ACMM+RAC

Note that ACMM is a particular type of AC where the encryption is done in the modeling component whereas RAC is again a particular type of AC in which the encryption is done in the encoding component, both by using the pseudorandom bit sequence. The Section 5.3 demonstrates that ACMM is insecure while Katti et al. [42] demonstrate that RAC is also insecure. Hence, intuitively, it seems that combination of these two encryptions, i.e., encryption at both modeling as well as at encoding component synchronously by using *different* pseudorandom bit sequences may enhance the security of the combined scheme (see Fig. 5.6). We refer this combined scheme as ACMM+RAC.

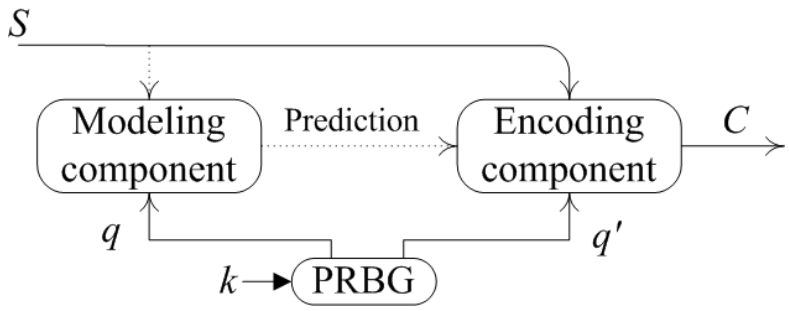

Figure. 5.6: Combined encryption scheme from modeling and encoding component.

Let us demonstrate the combined scheme through an example. Let the plaintext message be 11 and the corresponding pseudorandom bit sequences be $q = 11$ (for ACMM) and $q' = 01$ (for RAC), the final interval for 11 is given by $I(11) = [1/2, 2/3)$ (see Fig. 5.7).

Though, intuitively, it seems that ACMM+RAC gives security, the following analysis

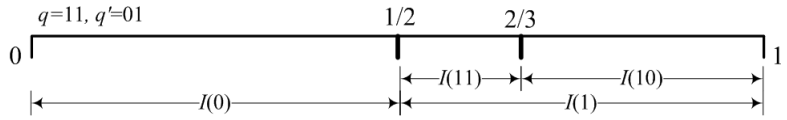

Figure. 5.7: Encryption example according to q=11 and q'=01.

can shows that the ACMM+RAC also does not have indistinguishable encryptions under the COA. Let us first provide the formal definition of the ACMM+RAC.

5.4.1 Formal Definition on ACMM+RAC

We are now going to define formally the ACMM+RAC scheme denoted by $\widetilde{\prod}$=($\widetilde{\mathsf{Enc}}$, $\widetilde{\mathsf{Dec}}$, $\widetilde{\mathsf{Gen}}$).

Definition 5.9. *Let n be the security parameter and N=$\ell(n)$ be a polynomial in n. Then $\widetilde{\prod}$= ($\widetilde{Enc}$, $\widetilde{Dec}$, $\widetilde{Gen}$) can be defined as follows*:

- $\widetilde{Gen}$: *On input 1^n and for each encryption of a binary plaintext message S of length N, choose $k\leftarrow\{0, 1\}^n$ uniformly and randomly. Then, output k as the key.*

- $\widetilde{Enc}$: *On input $k\in\{0, 1\}^n$ and the plaintext message $S\in\{0, 1\}^N$, choose $h\in\{0, 1\}^n$ and $h'\in\{0, 1\}^n$ uniformly at random but insure $h\neq h'$. Compute q=$F_k(h)$, q'=$F_k(h')$ and output the set of probabilities $\bigcup_{i=1}^{N} Pr^a[s_i]$:=$ACMM(q, S)$. Then, generate the ciphertext VC:=$\{h, h', RAC(q', \bigcup_{i=1}^{N} Pr^a[s_i], S)\}$, where C:=$RAC(q', \bigcup_{i=1}^{N} Pr^a[s_i], S)$, the notation ACMM has already been defined in Definition 5.1, and $RAC(q', p, S)$ implies the randomized arithmetic coding which is used to encrypt the plaintext message S with the pseudorandom bit sequence q' and probabilities set $Pr^a[S]$:=$\{Pr^a{}_i[s_i]|i\in\{1, 2,\ldots, N\}\}$.*

- $\widetilde{Dec}$: *On input k and ciphertext VC, the plaintext message S is decrypted by producing $F_k(h)$ and $F_k(h')$ and decoded through randomized AC (i.e., RAC).*

In the above Definition 5.9, h and h' are used to generate two pseudorandom bit sequences, i.e., q and q'. In fact, only one $h\in\{0, 1\}^n$ is needed to be chosen uniformly and

randomly, and h' can be expressed as: h'=$h \circ fv$, where fv is a fixed number. Then, these two pseudorandom bit sequences q and q', which are generated by $F_k(\cdot)$, are different. VC is redefined as $\{h, \mathsf{RAC}(q', \bigcup_{i=1}^{N}\Pr^a[s_i], S)\}$.

To analyze the security of $\widetilde{\prod}$, the experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)$ can also be used. However, some modifications are required which are listed below:

- For the encryption of a binary plaintext message S of length N, a fixed key k is generated by running the $\widetilde{\mathsf{Gen}}(1^n)$, and the h and h' are chosen uniformly at random. Using the h, h' and the key k for the plaintext message S, two pseudorandom bit sequences q and q' are generated by $F_k(\cdot)$. The ciphertext VC:=$\{h, h', \mathsf{RAC}(q', \bigcup_{i=1}^{N}\Pr^a[s_i], S)\}$, $\bigcup_{i=1}^{N}\Pr^a[s_i]$:=$\mathsf{ACMM}(q, S)$ is generated by the challenger and given to the adversary $\mathcal{A}$. Specially, C:=$\mathsf{RAC}(q', \bigcup_{i=1}^{N}\Pr^a[s_i], S)$. This VC is called the challenge ciphertext.

Then, this experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\prod'}(n)$ is redefined as $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\prod}}(n)$ for the following analysis. The security definition in Subsection 5.3.1 can be used for the analysis of the ACMM+RAC. Moreover, for the convenience of the analysis, the assumptions in Section 5.3 are still fit for the ACMM+RAC, and we also suppose that any real number in the final interval $I(S)$, which corresponds to C, is seen as the ciphertext RC.

5.4.2 Security Analysis of ACMM+RAC under *COA*

In this section, we state that the ACMM+RAC does not satisfy the security requirements as mentioned above. For the following analysis, two plaintext messages $10x_3$ and $11x_4$ of the same length N are considered, where x_3, $x_4 \in \{0, 1\}^{N-2}$. The interval [0, 1) is divided into the subintervals as [0, 1/6)$\bigcup$[1/6, 1/4)$\bigcup$[1/4, 1/3)$\bigcup$[1/3, 1/2)$\bigcup$[1/2, 2/3)$\bigcup$[2/3, 3/4)$\bigcup$[3/4, 5/6)$\bigcup$[5/6, 1) according to the observation of Table 5.4 which is the interval distribution of the encryption of s_1s_2=10 and $s'_1s'_2$=11. Specially, s_1s_2=10 and $s'_1s'_2$=11 belong to the messages $10y_1$ and $11y_2$ of length N ($N \geq 3$), respectively.

Lemma 5.4. *Let S_0= $10x_3$ and S_1= $11x_4$. Produce the random key k, and choose h and h' uniformly at random. Let q=$F_k(h)$ and q'=$F_k(h')$. Select a random bit $b \leftarrow \{0, 1\}$.*

Table. 5.4: Interval partition of s_1s_2=10 and $s'_1s'_2$=11 with different q and q'

q'	q	$I(10)$	$I(11)$	q'	q	$I(11)$	$I(10)$
10	000	[0, 1/6)	[1/6, 1/2)	11	000	[0, 1/3)	[1/3, 1/2)
	001	[0, 1/6)	[1/6, 1/2)		001	[0, 1/3)	[1/3, 1/2)
	010	[0, 1/4)	[1/4, 1/2)		010	[0, 1/4)	[1/4, 1/2)
	011	[0, 1/3)	[1/3, 1/2)		011	[0, 1/6)	[1/6, 1/2)
	100	[0, 1/6)	[1/6, 1/2)		100	[0, 1/3)	[1/3, 1/2)
	101	[0, 1/6)	[1/6, 1/2)		101	[0, 1/3)	[1/3, 1/2)
	110	[0, 1/4)	[1/4, 1/2)		110	[0, 1/4)	[1/4, 1/2)
	111	[0, 1/3)	[1/3, 1/2)		111	[0, 1/6)	[1/6, 1/2)
00	000	[1/2, 2/3)	[2/3, 1)	01	000	[1/2, 5/6)	[5/6, 1)
	001	[1/2, 2/3)	[2/3, 1)		001	[1/2, 5/6)	[5/6, 1)
	010	[1/2, 3/4)	[3/4, 1)		010	[1/2, 3/4)	[3/4, 1)
	011	[1/2, 5/6)	[5/6, 1)		011	[1/2, 2/3)	[2/3, 1)
	100	[1/2, 2/3)	[2/3, 1)		100	[1/2, 5/6)	[5/6, 1)
	101	[1/2, 2/3)	[2/3, 1)		101	[1/2, 5/6)	[5/6, 1)
	110	[1/2, 3/4)	[3/4, 1)		110	[1/2, 3/4)	[3/4, 1)
	111	[1/2, 5/6)	[5/6, 1)		111	[1/2, 2/3)	[2/3, 1)

Then, generate the ciphertext VC:={h, h', $\mathsf{RAC}(q', \bigcup_{i=1}^{N} Pr^a[s_i^b], S_b)$}, $\bigcup_{i=1}^{N} Pr^a[s_i^b]$:=$\mathsf{AC\_MM}(q, S_b)$. *Specially,* C:=$\mathsf{RAC}(q', \bigcup_{i=1}^{N} Pr^a[s_i^b], S_b)$ *and* RC *corresponds to* C. *If* RC *is in the interval* J_3, *where* J_3∈{*[0, 1/6), [1/3, 1/2), [1/2, 2/3), [5/6, 1)*}, $Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\widetilde{\prod}}(n)= 1|RC\in J_3]$=*19/35,* b'= 0 *is chosen as the output of the adversary* $\mathcal{A}$. *If* RC *is in* J_4, *where* J_4∈{*[1/6, 1/4), [1/4, 1/3), [2/3, 3/4), [3/4, 5/6)*}, $Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\widetilde{\prod}}(n)=1|RC\in J_4]$=*8/13,* b'=*1 is chosen as the output of the adversary* $\mathcal{A}$.

Proof. See appendix C for the proof. □

According to *Lemma* 5.4, the following theorem can be achieved.

Theorem 5.3. *The* $\widetilde{\prod}$ *does not have indistinguishable encryptions under the COA.*

Proof. A distinguisher $\mathcal{D}'$ is constructed for the adversary $\mathcal{A}$ which is used in the experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$. Generate the challenge ciphertext VC:={h, h', $\mathsf{RAC}(q', \bigcup_{i=1}^{N}\mathrm{Pr}^a[s_i^b], S_b)$}, $\bigcup_{i=1}^{N}\mathrm{Pr}^a[s_i^b]$:=$\mathsf{ACMM}(q, S_b)$. Specially, C:=$\mathsf{RAC}(q', \bigcup_{i=1}^{N}\mathrm{Pr}^a[s_i^b], S_b)$. Define the intervals CI_2=[0, 1/6)$\bigcup$[1/3, 1/2)$\bigcup$[1/2, 2/3)$\bigcup$[5/6, 1) and CI_3=[1/6, 1/3)$\bigcup$[2/3, 5/6). The distinguisher $\mathcal{D}'$ is described as follow:

Distinguisher $\mathcal{D}'$:

- If the value of RC, which is from the challenge ciphertext C, is in CI_2, S_0 is used to generate VC. The adversary $\mathcal{A}$ outputs b'=0.
- If the value of RC, which is from the challenge ciphertext C, is in CI_3, S_1 is used to generate VC. The adversary $\mathcal{A}$ outputs b'=1.

Then, for the $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$, $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)=1]$ can be expressed as

$$\Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\widetilde{\Pi}}(n)=1]=\sum\nolimits_{i=1}^{8}\Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\widetilde{\Pi}}(n)=1|RC\in[x_i,\,y_i)]\times\Pr[RC\in[x_i,\,y_i)],$$

where $[x_1,\,y_1)$=[0, 1/6), $[x_2,\,y_2)$=[1/6, 1/4), $[x_3,\,y_3)$=[1/4, 1/3), $[x_4,\,y_4)$=[1/3, 1/2), $[x_5,\,y_5)$=[1/2, 2/3), $[x_6,\,y_6)$=[2/3, 3/4), $[x_7,\,y_7)$=[3/4, 5/6), $[x_8,\,y_8)$=[5/6, 1). Then, according to *Lemma* 5.4,

$$\begin{aligned}
&\Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\widetilde{\Pi}}(n)=1]=\tfrac{19}{35}\times(\Pr[RC\in[0,\,\tfrac{1}{6})]\\
&+\Pr[RC\in[\tfrac{1}{3},\,\tfrac{1}{2})]+\Pr[RC\in[\tfrac{1}{2},\,\tfrac{2}{3})]+\Pr[RC\in[\tfrac{5}{6},\,1)])\\
&+\tfrac{8}{13}\times(\Pr[RC\in[\tfrac{1}{6},\,\tfrac{1}{4})]+\Pr[RC\in[\tfrac{2}{3},\,\tfrac{3}{4})]\\
&+\Pr[RC\in[\tfrac{3}{4},\,\tfrac{5}{6})]).
\end{aligned}$$

According to Eq. (A.2) (see Appendix C), the computation of $\Pr[C\in J_3]$ and $\Pr[C\in J_4]$ can come from the condition S_b=S_0 and S_b=S_1, respectively. Then, based on Table 5.4, we can obtain that

$$\begin{aligned}
&\Pr[RC\in[0,\,\tfrac{1}{6})]=\Pr[RC\in[\tfrac{1}{3},\,\tfrac{1}{2})]=\Pr[RC\in[\tfrac{1}{2},\,\tfrac{2}{3})]=\Pr[RC\in[\tfrac{5}{6},\,1)]=\tfrac{35}{192},\\
&\Pr[RC\in[\tfrac{1}{6},\,\tfrac{1}{4})]=\Pr[RC\in[\tfrac{1}{4},\,\tfrac{1}{3})]=\Pr[RC\in[\tfrac{2}{3},\,\tfrac{3}{4})]=\Pr[RC\in[\tfrac{3}{4},\,\tfrac{5}{6})]=\tfrac{13}{192}.
\end{aligned}$$

Then, $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)=1]$ is achieved by

$$\Pr[\mathsf{Privk}^{coa}_{\mathcal{A},\widetilde{\Pi}}(n)=1] = \frac{19}{35}\times\frac{140}{192}+\frac{8}{13}\times\frac{52}{192}=\frac{108}{192}.$$

As $108/192=1/2+12/192>1/2$, it demonstrates that the $\widetilde{\prod}$ does not have indistinguishable encryptions under the COA. □

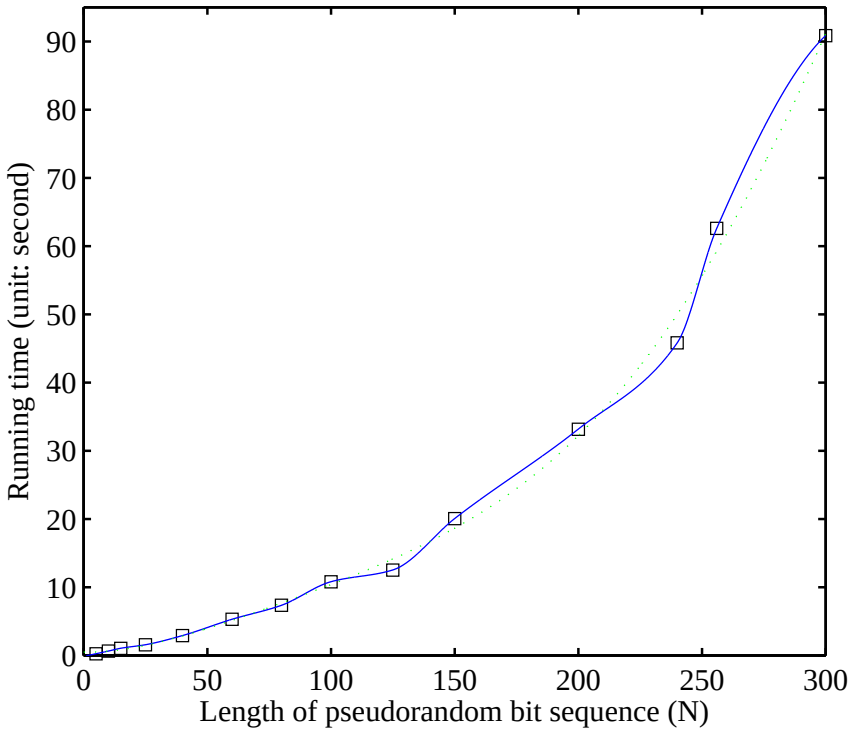

Figure. 5.8: Running time of proposed CPA.

5.5 Simulation of Proposed Attacks

This section presents the corresponding experimental results about our analyses on both the ACMM and ACMM+RAC.

5.5.1 Simulation Results of *CPA* on ACMM

To show the success of the CPA on ACMM, the following two simulations are presented. Firstly, according to the attack process, the used pseudorandom bit sequences of different length N are revealed and listed in Table 5.5. Specially, $Value(q)$ and $Value(q^{rv})$ are the decimal number of the original and revealed pseudorandom bit sequences, respectively. i.e., $Value(q)$=$q_1\times2^0+q_2\times2^1+q_3\times2^2...+q_N\times2^{N-1}$. From this table, it can be found that the revealed pseudorandom bit sequences are nearly the same as the original pseudorandom bit sequences. In some results, $Value(q)$ and $Value(q^{rv})$ are 1 apart. (e.g., when N=18, $Value(q)$=22531, $Value(q^{rv})$=22530). This is based on the fact that the first pseudorandom bit q_1 is not revealed. For the first pseudorandom bit q_1, the unique method to reveal it is related to the formula k_N=$q_N\oplus q_1$ when q_N=1. If q_N=0, q_1 can not be revealed according to the proposal.

Secondly, we test the running time of the proposed CPA for the different values of the lengths N. Fig. 5.8 is the variation trend of the running time for the different values of the lengths N. The test length of the pseudorandom bit sequence q is in the set {5, 10, 15, 25, 40, 60, 80, 100, 125, 150, 200, 240, 256, 300}. For each length N, 10 times experiments are performed for obtaining the average value. Note that all the experiments were done by Matlab2009 running on Core 2 Duo CPU 1.40GHz with 2.00GB RAM.

5.5.2 Simulation Results of *CPA* on ACMM and ACMM+RAC

For the COA, we implement the experiments by computing the probability that the adversary $\mathcal{A}$ answers correctly. This probability corresponds to a real number in [0, 1]. Specially, in this experiment, the adversary $\mathcal{A}$ chooses two kinds of plaintext message pairs for analyzing the ACMM and ACMM+RAC, respectively, i.e., (S_0, S_1)=($0x_1$, $1x_2$) and (S_0, S_1)=($10x_3$, $11x_4$). x_1, $x_2\in\{0, 1\}^{N-1}$ and x_3, $x_4\in\{0, 1\}^{N-2}$. For each plaintext message pair, the lengths are the same. In each experiment, the new pseudorandom bit sequence (q or (q and q')) is used to encrypt the plaintext message S_b. The length of S_0 and S_1 is from 20 to N' ($N'\in${1019, 2019, 3019, 4019, 5019,

Table. 5.5: Comparison between $Value(q)$ and $Value(q^{rv})$

N	$Value(q)$	$Value(q^{rv})$
5	30	30
8	213	213
10	689	689
14	2415	2414
15	17752	17752
18	22531	22530
20	568743	568743
22	1624471	1624470
25	14030286	14030286
26	65092697	65092697
27	62841898	62841898
28	1114909	1114908
29	422480402	422480402
32	4083488449	4083488449
33	940768214	940768214
34	2576196589	2576196588
35	17458990810	17458990810
36	44876744589	44876744589
37	76969885356	76969885356
38	150507139771	150507139771
39	496095156996	496095156996
40	744640262774	744640262774
44	5852185964842	5852185964842
46	36910857133194	36910857133194
47	79163154273721	79163154273721
48	249782728386734	249782728386734
51	31408458593527	31408458593526
53	3334307707163655	3334307707163654

6019, 7019, 8019, 9019, 10019}). This implies that the adversary $\mathcal{A}$ did 1000 experiments $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$ (or $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$), 2000 experiments $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$ (or $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$), 3000 experiments $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$ (or $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$),..., 10000 experiments $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$ (or $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$), respectively. Table 5.6 lists the corresponding results about this simulation.

In Table 5.6, Pr[·] is Pr[$\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$=1] (or Pr[$\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$=1]). $N(b'=b)$ denotes the number of times that the adversary $\mathcal{A}$ answers correctly within a fixed experiment. $N(\mathrm{b}'\neq\mathrm{b})$ denotes the number of times that the adversary fails to answer correctly within the same experiment. From the values of this table, it can be found that for the ACMM, if $(S_0,\ S_1)=(0x_1,\ 1x_2)$, the values of the Pr[$\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)$=1] match the deduction value of Theorem 5.2 in Subsection 5.3.3. i.e., the ACMM does not have indistinguishable encryptions under the COA. Moreover, for the ACMM+RAC, when $(S_0,\ S_1)=(10x_2,\ 11x_3)$, the values of the Pr[$\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)$=1] are near to the deduction value of Theorem 5.3 (i.e, 108/192). This implies that the ACMM+RAC also does not have indistinguishable encryptions under the COA.

5.6 Conclusions

In 2011, Duan et al. [31] presented an improved AC called ACMM. It was stated that ACMM can resist the attacks from AC aspect. In this chapter, we discussed the security of ACMM and presented the corresponding analysis. For achieving this purpose, we first put forward the formal definition of ACMM. Along with the definition, we presented various security notions related to ACMM. According to these definition and security notions, the security discussion was done under two kinds of conditions. For the first condition that the same pseudorandom bit sequence is used to encrypt the different messages, a chosen-plaintext attack was proposed to reveal the used pseudorandom bit sequence for the encryption. For the second condition that the different pseudorandom bit sequences are used to encrypt the different messages, we also showed that the ACMM does not have indistinguishable encryptions under the COA.

Table. 5.6: Simulation results on the $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\Pi'}(n)=1]$ and $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)=1]$

Experiment times	ACMM:$(S_0, S_1)=(0x_1, 1x_2)$		
	$N(b'=b)$	$N(b'\neq b)$	$\Pr[\cdot]$
1000	1000	0	1
2000	2000	0	1
3000	3000	0	1
4000	4000	0	1
5000	5000	0	1
6000	6000	0	1
7000	7000	0	1
8000	8000	0	1
9000	9000	0	1
10000	10000	0	1
Experiment times	ACMM+RAC:$(S_0, S_1)=(10x_2, 11x_3)$		
	$N(b'=b)$	$N(b'\neq b)$	$\Pr[\cdot]$
1000	546	454	0.5460
2000	1115	885	0.5575
3000	1675	1325	0.5583
4000	2235	1765	0.5587
5000	2762	2238	0.5524
6000	3323	2677	0.5538
7000	3875	3125	0.5536
8000	4428	3572	0.5535
9000	4987	4013	0.5541
10000	5526	4474	0.5526

As far as we know, to improve the security of AC, Grangetto et al. [36] proposed another variant of AC, defined as randomized arithmetic coding (RAC). Unfortunately, Katti et al. [42] demonstrated that RAC is also insecure. Hence, intuitively, it seems that combination of these two encryptions (i.e., encryption at both modeling and encoding component synchronously by using different pseudorandom bit sequences) may improve the security of each algorithm. Specially, this combined scheme can be referred as ACMM+RAC. However, we proved that ACMM+RAC does not have indistinguishable encryptions under the COA.

Moreover, Katti et al. [42] provided a new AC based scheme using AES [60] in the counter mode. Nevertheless, according to the opinion from Katti et al. [42], this use may be seen as the compression followed by the encryption, and it may increase the overhead. Therefore, it can be found that for this kind of research it should be done carefully.

Chapter 6

Conclusions and Future Research

In this chapter, we summarize the analysis results presented in the thesis, and present some possible directions for the future research.

6.1 Summary

In this thesis, we discussed some typical security analysis issues on the digital media encryption algorithms. For this discussion, the general description on the security analysis of the digital media encryption was presented in the Chapter 2. As the still image is one of the main digital media vehicles for transmitting and storing the information, our security analysis focuses on some typical image encryption algorithms. The used analysis methods have effect on these encryption algorithms. Moreover, some further consideration on these analyzed encryption algorithms were given. The main security analysis results are as follows.

(1) A scrambling analysis of image spacial scrambling algorithms was addressed. As the spacial scrambling algorithms, e.g., the Arnold cat map and Fibonacci transformation, are widely used to encrypt the still image, the scrambling degree of the corresponding ciphertext image needs to be measured by some specific method. In this thesis, an bit-plane based evaluation method was provided, which makes use of the bit-plane theory for the analysis. In particular, the scrambling degree of

each bit-plane is measured by the spatial distribution entropy and centroid difference for the bit-plane. Since the relationship between the original image and the most significant bit-plane to the least significant bit-plane reduces gradually, the level-decreasing based weight is set for computing the final scrambling degree of the still image. The experimental analysis shows that this evaluation method can find the scrambling degree efficiently for the image spacial scrambling encryption.

(2) A security analysis of an image encryption algorithm of pixel bits proposed by Ye [85] was presented, and the comparison with the previous work was done. In the work [85], Ye proposed an image encryption scheme of pixel bits which can achieve the permutation of pixels in the space domain and the encryption of the pixel values simultaneously. According to the analysis of the work [85], this encryption algorithm has good performance. However, we found that the chosen-plaintext attack (CPA) can be used to break this algorithm when the user-supplied secret key is fixed for the encryption. When some special plaintext-ciphertext pairs are used in the original encryption algorithm, the encryption vectors are completely revealed. Compared with the previous analysis work achieved by Li and Lo [45], our attack has the lower complexity. Moreover, in this thesis, an suggested improvement against the proposed attack was presented in details. This suggested improvement introduces the idea of the self-correlation into the encryption procedure. The analysis and simulation results demonstrate that the performance of the suggested improvement may be better than the original one.

(3) The security on the randomized arithmetic codes based on the Markov model (i.e., ACMM) was discussed. Duan et al. [31] presented a Markov model based randomized arithmetic coding which can achieve the encryption and compression for the multimedia information simultaneously. This kind of algorithm can be seen as a revised arithmetic code. The corresponding performance is presented by Duan et al. in the original work [31]. In this thesis, we put forward a formal definition of ACMM firstly, and then, explored the security of ACMM by using the definition. The anal-

ysis illuminates that ACMM is insecure under the chosen-plaintext attack when the same paseudorandom bit sequence is used in the encryption. Moreover, even if a new pseudorandom bit sequence is used for the encryption of each message, the corresponding security analysis demonstrates that the ACMM is insecure under the ciphertext-only attack (COA). Moreover, an combination algorithm ACMM+RAC was presented in our work, where RAC denotes the randomized arithmetic coding [36]. The security of ACMM+RAC was also discussed under the COA, and we showed that this combination is insecure.

6.2 Possible Directions for Future Research

We present some possible directions for the future research on digital media encryption algorithms, specially on image encryption algorithms, and these directions may be interesting to be investigated in the future.

(1) In Chapter 2, we have mentioned that the second method (i.e., first-encryption-then-compression approach) is one kind of method for the secure communication (and storage) of still image. Therefore, the study on the design of the secure encryption algorithm for the still image is still a challenge. In particular, the characteristics of the still image should be considered in the design of the encryption algorithm, which implies that the encryption efficiency may be improved.

(2) As our work in Chapter 5, we have analyzed the security of one kind of revised arithmetic code which can do the encryption simultaneously (i.e., ACMM). Moreover, since the arithmetic coding has two steps (i.e., modeling component and encoding component), we also analyzed the security of a combination algorithm which makes use of the ACMM [31] and RAC [36]. The results show that these algorithms are insecure. Therefore, it seems that inserting the encryption into the arithmetic coding is hard to construct the randomized scheme. However, we may consider the encryption-compression algorithm from other side, i.e., the compression inserted into the encryption algorithm. In fact, there have been some related

work about it, e.g., [6, 7]. Therefore, we can try to investigate the research on the encryption-compression algorithm from this side. Moreover, the corresponding security analysis should also be explored in the future.

(3) Security analysis is a very fast moving field. Therefore, it is possible to improve our analysis results presented in Chapter 4 and Chapter 5. For example, recently, Hermassi et al. [96] have presented a security analysis of Ye's encryption algorithm [85], which can be seen as the improvement about our second result in Chapter 4. According to their analysis [96], it seems that *in real world application*, the information quantity used in their attack is less than that in our attack.

Appendix

Appendix A: Proof of Lemma 5.1

Proof. For the sake of the simplicity, we only prove the case for d_1=0 and d_2=0. The other cases follow the similar proof and obtain the same deduction. To prove this *Lemma*, it is sufficient to show that Pr[s_N=0]–Pr[s_N=0|s_{N-1}=0]≠0. According to Eq. (5.1), Pr[s_N=0]–Pr[s_N=0|s_{N-1}=0] can be changed to

$$\frac{n_0^{N-1}(0)}{n_0^{N-1}(0)+n_0^{N-1}(1)}-\frac{n^{N-1}(0)}{n^{N-1}(0)+n^{N-1}(1)}. \tag{A.1}$$

When the N–1 many 0s have been encoded (i.e., the Markov model has been updated for encoding the N-th symbol), $n^{N-1}(0)$, $n^{N-1}(1)$, ${n_0}^{N-1}(0)$ and ${n_0}^{N-1}(1)$ should satisfy:

$$\begin{cases} n_0^{N-1}(1)+n_0^{N-1}(0)+1=n^{N-1}(0)+n^{N-1}(1)=N+1 \\ n_0^{N-1}(1)\geq 1,\ n_0^{N-1}(0)\geq 1,\ n^{N-1}(0)\geq 1,\ n^{N-1}(1)\geq 1 \end{cases}$$

Therefore, Eq. (A.1) is equivalent to the following transformation:

$$\frac{n_0^{N-1}(0)\times(N+1)-n^{N-1}(0)\times N}{(n_0^{N-1}(0)+n_0^{N-1}(1))\times(n_0^{N-1}(0)+n_0^{N-1}(1)+1)},$$

where ${n_0}^{N-1}(1)+{n_0}^{N-1}(0))\times({\rm n_0}^{N-1}(1)+{\rm n_0}^{N-1}(0)+1)\neq 0$. To estimate the value of ${n_0}^{N-1}(0)\times(N+1)$–$n^{N-1}(0)\times N$, the apagoge is used. Suppose that ${n_0}^{N-1}(0)\times(N+1)$–$n^{N-1}(0)\times N$=0, then,

$$\frac{n_0^{N-1}(0)}{n^{N-1}(0)}=\frac{N}{N+1}.$$

However, as gcd(N, N+1)=1, ${n_0}^{N-1}(0)<N$ and $n^{N-1}(0)<(N+1)$,

$$\frac{n_0^{N-1}(0)}{n^{N-1}(0)}\neq\frac{N}{N+1}.$$

Hence,

$$\frac{n_0^{m-1}(0)}{n_0^{m-1}(0)+n_0^{m-1}(1)} \neq \frac{n^{m-1}(0)}{n^{m-1}(0)+n^{m-1}(1)}.$$

This implies that $\Pr[s_N{=}0]$–$\Pr[s_N{=}0|s_{N-1}{=}0]{\neq}0$. □

Appendix B: Proof of Lemma 5.3

Proof. Suppose that the plaintext message is S_0, the ciphertext is $VC_0{:=}\{h, \mathsf{SAC}(\bigcup_{i=1}^{N} \Pr^a[s_i^0], S_0)\}$, $\bigcup_{i=1}^{N}\Pr^a[s_i^0]{:=}\mathsf{ACMM}(q, S_0)$. Specially, $C_0{:=}\mathsf{SAC}(\bigcup_{i=1}^{N}\Pr^a[s_i^0], S_0)$ and RC_0 is the real number corresponding to C_0. If only the encryption of the first symbol s_1 is considered, according to the encryption steps of the ACMM, $\Pr^a[s_0{=}0]{=}\Pr^a[s_0{=}1]{=}0.5$ for encrypting the s_1. Moreover, as the encoding component is the standard AC, for the $s_1{=}0$, the corresponding interval must be [0, 0.5). Then, based on the fact that $I(0x_1){\subseteq}I(0)$, RC_0 must be in J_1. Similarly, if the plaintext message is S_1, it can show that the RC_1 corresponding to the ciphertext VC_1 must be in J_2.

Therefore, if the real number RC is in J_1, the ciphertext VC must correspond to the plaintext message S_0, otherwise, it must be the encryption of S_1. This implies that for such plaintext messages S_0 and S_1, the adversary $\mathcal{A}$ will succeed in the proposed experiment. □

Appendix C: Proof of Lemma 5.4

Proof. For the experiment $\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\prod}}(n)$, the success of the adversary $\mathcal{A}$ is dependent on the value of b, i.e., $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\prod}}(n){=}1]$ is decided by the condition $S_{b'}{=}S_b$. For the intervals J_3 and J_4, both the encryptions of S_0 and S_1 can be within them. Then, for these intervals, the probabilities under the condition $S_b{=}S_0$ and $S_b{=}S_1$ should be compared. e.g., for the interval J_3, if the probability $\Pr[RC{\in}J_3|S_b{=}S_0]{=}\Pr[RC{\in}J_3|S_b{=}S_1]$, $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\prod}}(n){=}1|RC{\in}J_3]{=}1/2$. Otherwise, the adversary $\mathcal{A}$ should output the $S_{b'}$ which has the bigger probability for producing the ciphertext within the interval J_3. This implies that if $\Pr[RC{\in}J_3|S_b{=}S_0]{>}\Pr[RC{\in}J_3|S_b{=}S_1]$, the adversary $\mathcal{A}$ outputs $b'{=}0$. To

obtain the probabilities of $RC \in J_3$ and $RC \in J_4$ under the condition $S_b=S_0$ and $S_b=S_1$, the adversary $\mathcal{A}$ draws the interval distribution table of first two binary symbols 10 and 11 (see Table 5.4) for the analysis, where q is $F_k(h)$ and q' is $F_k(h')$. This analysis is based on the fact that as $I(10x_3) \subseteq I(10)$ and $I(11x_4) \subseteq I(11)$, $RC(S_0) \in I(10)$ and $RC(S_1) \in I(11)$. Then, these probabilities can be produced by computing the following formula:

$$\begin{aligned} &\Pr[RC \in [x,\, y) | S_b = S_w] \\ &= \sum_{d'=1}^{e'} \tfrac{1}{4} \times \sum_{d=1}^{e} \tfrac{|[x,y)|}{|I_d(s_0 s_1)|} \times \tfrac{\#\{I_d^{F_k(h_1)}(s_0 s_1) = I_d^{F_k(h_2)}(s_0 s_1) :\, F_k(h_1) \neq F_k(h_2)\}}{8} \end{aligned} \quad , \tag{A.2}$$

where $|\cdot|$ denotes the length of the interval, $I(s_0 s_1)$ corresponds to the plaintext $S_b=S_w$, $w \in \{0,\, 1\}$, $e' \in \{1,\, 2\}$, $e \in \{1,\, 2,\, 3\}$, $\#\{I^{F_k(h_1)}(s_0 s_1) = I^{F_k(h_2)}(s_0 s_1) :\ F_k(h_1) \neq F_k(h_2)\}$ is the number of the same interval (e.g., for $F_k(h_1)=000$ and $F_k(h_2)=001$, the intervals of $I(10)$ or $I(11)$ are the same. Then, $\#\{10\}=\#\{11\}=2$). Specially, $[x,\, y) \in \{J_3,\, J_4\}$.

To achieve the $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)=1]$, each sub-interval should be considered separately. In this proof, two examples are given in details. For $J_3=[0,\, 1/6)$, if $s_0 s_1=10$, when $F_k(h) \in \{000,\, 001,\, 010,\, 011,\, 100,\, 101,\, 110,\, 111\}$ and $F(k')=10$, $J_3 \subseteq I(10)$. According to Eq. (A.2),

$$\Pr[RC \in [0,\, \tfrac{1}{6}) | S_b = S_0] = \tfrac{1}{4} \times (\tfrac{1}{2} + \tfrac{2}{3} \times \tfrac{1}{4} + \tfrac{1}{2} \times \tfrac{1}{4}) = \tfrac{19}{96} \ ,$$

Moreover, if $s_0 s_1=11$, when $F_k(h) \in \{000,\, 001,\, 010,\, 011,\, 100,\, 101,\, 110,\, 111\}$ and $F_k(h')=11$, $J_3 \subseteq I(11)$. Then, for S_1,

$$\Pr[RC \in [0,\, \tfrac{1}{6}) | S_b = S_1] = \tfrac{1}{4} \times (\tfrac{1}{4} + \tfrac{1}{2} \times \tfrac{1}{2} + \tfrac{2}{3} \times \tfrac{1}{4}) = \tfrac{1}{6} \ ,$$

As $\Pr[RC \in [0,\, 1/6) | S_b=S_0] > \Pr[RC \in [0,\, 1/6) | S_b=S_1]$, $b'=0$ is chosen as the output of the adversary $\mathcal{A}$. The $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n)=1 | RC \in [0,\, 1/6)]$ should be computed as follow,

$$\begin{aligned} &\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 | RC \in [0,\, \tfrac{1}{6})] \\ &= \tfrac{\Pr[RC \in [0,\, \frac{1}{6}) | S_b=S_0] \times \Pr[S_b=S_0]}{\Pr[RC \in [0,\, \frac{1}{6})]} = \tfrac{19}{35} \end{aligned} \ ,$$

where $\Pr[RC \in [0,\, 1/6)] = \Pr[RC \in [0,\, 1/6) | S_b=S_0] \times \Pr[S_b=S_0] + \Pr[RC \in [0,\, 1/6) | S_b=S_1] \times \Pr[S_b=S_1]$. For $J_4=[1/6,\, 1/4)$, if $s_0 s_1=10$, when $F_k(h) \in \{010,\, 011,\, 110,\, 111\}$, $F_k(h')=$

10, and when $F_k(h)\in\{011, 111\}$, $F_k(h')=11$, it is within $I(10)$. Then,

$$\Pr[RC \in [\tfrac{1}{6}, \tfrac{1}{4})|S_b = S_0] = \tfrac{1}{4} \times (\tfrac{1}{3} \times \tfrac{1}{4} + \tfrac{1}{4} \times \tfrac{1}{4}) + \tfrac{1}{4} \times (\tfrac{1}{4} \times \tfrac{1}{4}) = \tfrac{5}{96} \ ,$$

If $s_0s_1=11$, when $F_k(h)\in\{000, 001, 100, 101\}$, $F_k(h')=10$, and when $F_k(h)\in\{000, 001, 010, 100, 101, 110\}$, $F_k(h')=11$, $J_4\in I(11)$. Then,

$$\Pr[RC \in [\tfrac{1}{6}, \tfrac{1}{4})|S_b = S_1] = \tfrac{1}{4} \times (\tfrac{1}{2} \times \tfrac{1}{4} + \tfrac{1}{3} \times \tfrac{1}{4}) + \tfrac{1}{4} \times (\tfrac{1}{2} \times \tfrac{1}{4}) = \tfrac{1}{12} \ ,$$

As $\Pr[RC\in[0, 1/6)|S_b=S_1]>\Pr[RC\in[0, 1/6)|S_b=S_0]$, $b'=1$ is chosen as the output of the adversary $\mathcal{A}$. The $\Pr[\mathsf{Privk}^{\mathrm{coa}}_{\mathcal{A},\widetilde{\prod}}(n)=1|RC\in[1/6, 1/4)]$ should be computed as follow,

$$\begin{aligned}&\Pr[\mathsf{Privk}^{coa}_{A,\widetilde{\prod}}(n) = 1|RC \in [\tfrac{1}{6}, \tfrac{1}{4})]\\ &= \tfrac{\Pr[RC\in[\frac{1}{6}, \frac{1}{4})|S_b=S_1]\times\Pr[S_b=S_1]}{\Pr[RC\in[\frac{1}{6}, \frac{1}{4})]} = \tfrac{8}{13} \ ,\end{aligned}$$

where $\Pr[RC\in[1/6, 1/4)]=\Pr[RC\in[1/6, 1/4)|S_b=S_0]\times \Pr[S_b=S_0]+\Pr[RC\in[1/6, 1/4)|S_b=S_1]\times\Pr[S_b=S_1]$.

The same method can be used to analyze the other sub-intervals, i.e., $\{[1/3, 1/2), [1/2, 2/3), [5/6, 1), [1/4, 1/3), [2/3, 3/4), [3/4, 5/6)\}$. Then, the conclusion is achieved

$$\begin{cases} \Pr[\mathsf{Privk}^{coa}_{A,\widetilde{\prod}}(n) = 1|RC \in J_3] = 19/35, \ b' = 0 \\ \Pr[\mathsf{Privk}^{coa}_{A,\widetilde{\prod}}(n) = 1|RC \in J_4] = 8/13, \ b' = 1 \end{cases} .$$

□

Bibliography

[1] J.Q. Lu. Cryptanalysis of block ciphers. *Ph.D. Thesis*, Department of Mathematics, University of London, Available: `https://www.iacr.org/phds/index.php?p=detail&entry=282`, 2008.

[2] M. Rafiq, K. Ameen. Use of digital media and demand for digitized contents in higher education sector of Pakistan. *The International Information & Library Review*, Elsevier press, `doi:10.1016/j.iilr.2012.04.007`, 2012. (In press)

[3] N. Williams, G.S Blair Distributed multimedia applications: A review. *Computer Communications*, 17(2):119–132, 1994.

[4] D. Schonberg, SC Draper, C. Yeo, K. Ramchandran. Toward compression of encrypted images and video sequences. *IEEE Transactions on Information Forensics and Security*, 3(4):749–762, 2008.

[5] X. Zhang. Lossy Compression and Iterative Reconstruction for Encrypted Image. *IEEE Transactions on Information Forensics and Security*, 6(1):53–58, 2011.

[6] K.W. Wong, C.H. Yuen. Embedding compression in chaos-based cryptography. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 55(11):1193–1197, 2008.

[7] J. Chen, J. Zhou, K.W. Wong. A modified chaos-based joint compression and encryption scheme. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 99:1–5, 2011.

[8] C.C. Chang, M.S. Hwang, T.S. Chen. A new encryption algorithm for image cryptosystems. *Journal of Systems and Software*, 58(2):83–91, 2001.

[9] W. Puech, J.M. Rodrigues, J.E. Develay-Morice. A new fast reversible method for image safe transfer. *Journal of Real-Time Image Processing*, 2(1):55–65, 2007.

[10] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.

[11] D.R. Stinson. Cryptography theory and practice. *CRC Press*, Second edition, 2002.

[12] A. Bogdanov. Analysis and design of block cipher constructions. *Ph.D. Thesis*, Horst Görtz Institute for IT Security, Ruhr University Bochum, Available: `https://www.iacr.org/phds/index.php?p=detail&entry=480`, 2009.

[13] J. Borghoff. Cryptanalysis of lightweight ciphers. *Ph.D. Thesis*, Department of Mathematics, Technical University of Denmark, Available: `https://www.iacr.org/phds/index.php?p=detail&entry=748`, 2010.

[14] L.R. Knudsen, M.J.B. Robshaw. The Block cipher companion. *Springer Press*, 2011.

[15] A. Webster, S. Tavares. On the design of S-boxes. *Proceedings of Advances in Cryptology (CRYPTO'85)*, Lecture notes in computer science: volume 218, pages 523–534, Springer-Verlag, Heidelberg, August, 1985.

[16] K.L. Chung, L.C. Chang. Large encrypting binary images with higher security. *Pattern Recognition Letters*, 19(5):461–468, 1998.

[17] C.C. Chang, T.X. Yu. Cryptanalysis of an encryption scheme for binary images. *Pattern Recognition Letters*, 23(14):1847–1852, 2002.

[18] D. Engel, E. Pschernig, A. Uhl. An analysis of lightweight encryption schemes for fingerprint images *IEEE Transactions on Information Forensics and Security*, 3 (2):173–182, 2008.

[19] S. Li, C. Li, G. Chen, K.T. Lo. Cryptanalysis of the RCES/RSES image encryption scheme. *Journal of Systems and Software*, 81(7):1130–1143, 2008.

[20] F. Ahmed, M.Y. Siyal, and V. Uddin Abbas. A secure and robust hash-based scheme for image authentication. *Signal Processing*, 90(5):1456–1470, 2010.

[21] G. Alvarez and S. Li. Some basic cryptographic requirements for chaos-based cryptosystems. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, 16(8):2129, 2006.

[22] H.A. Bergen and J.M. Hogan. Data security in a fixed-model arithmetic coding compression algorithm. *Computers & Security*, 11(5):445–461, 1992.

[23] H.A. Bergen and J.M. Hogan. A chosen plaintext attack on an adaptive arithmetic coding compression algorithm. *Computers & Security*, 12(2):157–167, 1993.

[24] N.G. Bourbakis. Image data compression-encryption using g-scan patterns. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. IEEE International Conference on Computational Cybernetics and Simulation*, volume 2, pages 1117–1120. IEEE Society, Octorber, 1997.

[25] C.C. Chang and T.X. Yu. Cryptanalysis of an encryption scheme for binary images. *Pattern Recognition Letters*, 23(14):1847–1852, 2002.

[26] G. Chen, Y. Mao, and C.K. Chui. A symmetric image encryption scheme based on 3d chaotic cat maps. *Chaos, Solitons & Fractals*, 21(3):749–761, 2004.

[27] K. Wang, L. Zou, A. Song, Z. He. On the security of 3D Cat map based symmetric image encryption scheme. *Physics Letters A*, 343(6):432–439, 2005.

[28] G. Chen, X.Y. Zhao, and J.L. Li. Self-adaptive algorithm on image encryption. *Journal of Software*, 16(11):1975–1982, 2005.

[29] R.J. Chen and S.J. Horng. Novel SCAN-CA-based image security system using SCAN and 2-d *von neumann* cellular automata. *Signal Processing: Image Communication*, 25(6):413–426, 2010.

[30] C. Çokal and E. Solak. Cryptanalysis of a chaos-based image encryption algorithm. *Physics Letters A*, 373(15):1357–1360, 2009.

[31] L.L. Duan, X.F. Liao, and T. Xiang. A secure arithmetic coding based on markov model. *Communications in Nonlinear Science and Numerical Simulation*, 16(6): 2554–2562, 2011.

[32] F.G. Zhao, E.X. Jiang, and X.F. Ni. On the specific expression of bit-level arithmetic coding. *Numerical Mathematics, A Journal of Chinese Universities*, 7(2): 211–220, 1998.

[33] B. Furht, D. Socek, and A.M. Eskicioglu. Multimedia security handbook, volume 4 of Internet and Communications Series, chapter "Fundamentals of multimedia encryption techniques". *CRC press*, 93–132 (Chapter 3), 2004.

[34] T. Gao and Z. Chen. A new image encryption algorithm based on hyper-chaos. *Physics Letters A*, 372(4):394–400, 2008.

[35] V. Gligor and P. Donescu. Integrity-aware PCBC encryption schemes. In *Proceedings of the 7th International Workshop on Security Protocols, (IWSP'99)*,Lecture notes in computer science: volume 1796, pages 153–168. Springer-Verlag, Heidelberg, April, 1999.

[36] M. Grangetto, E. Magli, and G. Olmo. Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Transactions on Multimedia*, 8(5):905–917, 2006.

[37] Z.H. Guan, F. Huang, and W. Guan. Chaos-based image encryption algorithm. *Physics Letters A*, 346(1-3):153–157, 2005.

[38] V.D. Viile, W. Philips, V.D. Walle, I. Lemahieu. Image scrambling without bandwidth expansion. *IEEE Transactions on Circuits System Video Technology* 14 892–897, 2004.

[39] J. Hu and F. Han. A pixel-based scrambling scheme for digital medical images protection. *Journal of Network and Computer Applications*, 32(4):788–794, 2009.

[40] G. Jakimoski and K.P. Subbalakshmi. Cryptanalysis of some multimedia encryption schemes. *IEEE Transactions on Multimedia*, 10(3):330–338, 2008.

[41] H.L. Jiao and G Chen. A color image fractal compression coding method. *Journal of Software*, 14(4):864–868, 2003.

[42] R.S. Katti, S.K. Srinivasan, and A. Vosoughi. On the security of randomized arithmetic codes against ciphertext-only attacks. *IEEE Transactions on Information Forensics and Security*, 6(1):19–27, 2011.

[43] J. Katz and Y. Lindell. Introduction to modern cryptography. *Chapman & Hall*, 2008.

[44] H. Kim, J. Wen, and J.D. Villasenor. Secure arithmetic coding. *IEEE Transactions on Signal Processing*, 55(5):2263–2272, 2007.

[45] C. Li and K.T. Lo. Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing*, 91(4):949–954, 2011.

[46] C. Li, S. Li, G. Alvarez, G. Chen, and K.T. Lo. Cryptanalysis of two chaotic encryption schemes based on circular bit shift and xor operations. *Physics Letters A*, 369(1-2):23–30, 2007.

[47] S. Li, G. Chen, and X. Zheng. Multimedia Security Handbook, volume 4 of Internet and Communications Series, chapter "Chaos-based encryption for digital images and videos". *CRC press*, 133–167 (Chapter 4), 2004.

[48] S. Li, C. Li, G. Chen, N.G. Bourbakis, and K.T. Lo. A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing: Image Communication*, 23(3):212–223, 2008.

[49] X. Li. A new measure of image scrambling degree based on grey level difference and information entropy. In *Proceedings of International Conference on Computational Intelligence and Security (CIS'08)*, volume 1, pages 350–354. IEEE Society, December, 2008.

[50] L.H. Zhu, W.Z. Li, L.J. Liao, and H. Li. A novel image scrambling algorithm for digital watermarking based on chaotic sequences. *International Journal of Computer Science and Network Security*, 6(8B):125–130, 2006.

[51] J. Lim, C. Boyd, and E. Dawson. Cryptanalysis of adaptive arithmetic coding encryption schemes. In *Proceedings of the 2nd Australasian Conference on Information Security and Privacy, (ACISP'97)*, Lecture notes in computer science: volume 1270, pages 216–227, Springer-Verlag, Heidelberg, July, 1997.

[52] K.T. Lin. Information hiding based on binary encoding methods and pixel scrambling techniques. *Applied Optics*, 49(2):220–228, 2010.

[53] X. Liu, P. Farrell, and C. Boyd. Resisting the bergen-hogan attack on adaptive arithmetic coding. *Proceedings of the 6th IMA International Conference on Crytography and Coding, (IMACC'97)*,Lecture notes in computer science volume 1355 , pages 199–208, Springer-Verlag, Heidelberg, December, 1997.

[54] S.S. Maniccam and N.G. Bourbakis. Lossless image compression and encryption using scan. *Pattern Recognition*, 34(6):1229–1245, 2001.

[55] S.S. Maniccam and N.G. Bourbakis. Image and video encryption using scan patterns. *Pattern Recognition*, 37(4):725–737, 2004.

[56] Y.Y. Wang, Y.R. Wang, Y. Wang, H.J. Li, W.J. Sun. Optical image encryption based on binary Fourier transform computer-generated hologram and pixel scrambling technology *Optics and Lasers in Engineering*, 45(7):761–765, 2007.

[57] V. Monga, A. Banerjee, and B.L. Evans. A clustering based approach to perceptual image hashing. *IEEE Transactions on Information Forensics and Security*, 1(1): 68–79, 2006.

[58] A.H. Paquet, R.K. Ward, and I. Pitas. Wavelet packets-based digital watermarking for image verification and authentication. *Signal Processing*, 83(10):2117–2132, 2003.

[59] M. Podesser, H.P. Schmidt, and A. Uhl. Selective bitplane encryption for secure transmission of image data in mobile environments. In *Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02)*, pages 4–6, Octorber, 2002.

[60] N.F. Pub. 197: Advanced encryption standard (AES). *Federal Information Processing Standards Publication*, 197:441–0311, 2001.

[61] D. Qi, J. Zou, and X. Han. A new class of scrambling transformation and its application in the image information covering. *Science in China Series E: Technological Sciences*, 43(3):304–312, 2000.

[62] R. Rhouma, and S. Belghith. Cryptanalysis of a spatiotemporal chaotic cryptosystem. *Chaos, Solitons & Fractals*, 41(4):1718–1722, 2009.

[63] E. Solak. On the security of a class of discrete-time chaotic cryptosystems. *Physics Letters A*, 320(5-6):389–395, 2004.

[64] E. Solak. Cryptanalysis of image encryption with compound chaotic sequence. In *Proceedings of the 6th International Multi-Conference on Systems, Signals and Devices,(SSD'09)*, pages 1–5. IEEE Society, 2009.

[65] E. Solak, R. Rhouma, and S. Belghith. Cryptanalysis of a multi-chaotic systems based image cryptosystem. *Optics Communications*, 283(2):232–236, 2010.

[66] D.R. Stinson. Cryptography: theory and practice. *CRC Press*, 2006.

[67] D. Kahn. The Codebreakers: The Story of Secret Writing. *Macmillan Press*, 1967.

[68] H.M. Sun, K.H. Wang, and W.C. Ting. On the security of the secure arithmetic code. *IEEE Transactions on Information Forensics and Security*, 4(4):781–789, 2009.

[69] J.D. Sun, Z.G. Ding, and L.H. Zhou. Image retrieval based on image entropy and spatial distribution entropy. *Journal Infrared Millimeter and Waves*, 24(2): 135–139, 2005.

[70] A. Swaminathan, Y. Mao, and M. Wu. Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, 1(2):215–230, 2006.

[71] X. Tong and M. Cui. Image encryption scheme based on 3d baker with dynamical compound chaotic sequence cipher generator. *Signal Processing*, 89(4):480–491, 2009.

[72] H.M. Tsai and L.W. Chang. Secure reversible visible image watermarking with authentication. *Signal Processing: Image Communication*, 25(1):10–17, 2010.

[73] T. Uehara and R. Safavi-Naini. Attack on liu/farrell/boyd arithmetic coding encryption scheme. In *IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security, (CMS'99)*, IFIP Conference Proceedings volume 152, pages 273–290, Kluwer, September, 1999.

[74] A. Uhl and A. Pommer. Image and video encryption: from digital rights management to secured personal communication. *Springer press*, 15, 2005.

[75] K. Wang, L. Zou, A. Song, Z. He, et al. On the security of 3D cat map based symmetric image encryption scheme. *Physics Letters A*, 343(6):432–439, 2005.

[76] J. Wen, H. Kim, and J.D. Villasenor. Binary arithmetic coding with key-based interval splitting. *IEEE Signal Processing Letters*, 13(2):69–72, 2006.

[77] I.H. Witten and J.G. Cleary. On the privacy afforded by adaptive text compression. *Computers & Security*, 7(4):397–408, 1988.

[78] C.P. Wu and C.C.J. Kuo. Design of integrated multimedia compression and encryption systems. *IEEE Transactions on Multimedia*, 7(5):828–839, 2005.

[79] H.K.C. Chang, J.L. Liu. A linear quadtree compression scheme for image encryption. *Signal Processing: Image Communication*, 10(4):279–290, 1997.

[80] D. Wu, X. Zhou, and X. Niu. A novel image hash algorithm resistant to print–scan. *Signal Processing*, 89(12):2415–2424, 2009.

[81] T. Xiang, K.W. Wong, X.F. Liao, et al. Selective image encryption using a spatiotemporal chaotic system. *Chaos (Woodbury, NY)*, 17(2):023115, 2007.

[82] D. Xiao, X.F. Liao, and P.C. Wei. Analysis and improvement of a chaos-based image encryption algorithm. *Chaos, Solitons & Fractals*, 40(5):2191–2199, 2009.

[83] FileFormat.Info. Available: `http://www.fileformat.info/tip/web/imagesize.htm`, 2009.

[84] The USC-SIPI image database. Available: `http://sipi.usc.edu/database/`.

[85] G. Ye. Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, 31(5):347–354, 2010.

[86] R. Ye and H. Li. A novel image scrambling and watermarking scheme based on cellular automata. In *Proceedings of The International Symposium on Electronic Commerce and Security (ISECS'08)*, pages 938–941, IEEE Society, 2008.

[87] X.Y. Yu, J. Zhang, H.E. Ren, S. Li, and X.D. Zhang. A new measurement method of iimage encryption. In *Journal of Physics: Conference Series*, 48:408, IOP Publishing, 2006.

[88] M.R. Zhang, G.C. Shao, and K.C. Yi. T-matrix and its applications in image processing. *Electronics Letters*, 40(25):1583–1584, 2004.

[89] J. Zhou, Z. Liang, Y. Chen, and O.C. Au. Security analysis of multimedia encryption schemes based on multiple huffman table. *IEEE Signal Processing Letters*, 14 (3):201–204, 2007.

[90] J. Zhou, O.C. Au, and P.H.W. Wong. Adaptive chosen-ciphertext attack on secure arithmetic coding. *IEEE Transactions on Signal Processing*, 57(5):1825–1838, 2009.

[91] Q. Zhou, K.W. Wong, X.F. Liao, and Y. Hu. On the security of multiple huffman table based encryption. *Journal of Visual Communication and Image Representation*, 22(1):85–92, 2011.

[92] R.S. Katti, A. Vosoughi. On the Security of Key-Based Interval Splitting Arithmetic Coding With Respect to Message Indistinguishability. *IEEE Transactions on Information Forensics and Security*, 7(3):895–903, 2012.

[93] J. Zou, R.K. Ward, and D. Qi. A new digital image scrambling method based on fibonacci numbers. In *Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS'04)*, volume 3, pages 965–968, IEEE Society, May, 2004.

[94] F. Chen, K.W. Wong, X.F. Liao, and T. Xiang. Period Distribution of Generalized Discrete Arnold Cat Map for $N=p^e$ *IEEE Transactions on Information Theory*, 58(1):445–452, 2012.

[95] J.C. Zou, G.F. Li, and D.X. Qi. Generalized gray code and its application in the scrambling technology of digital images. *Applied Mathematics (A), A Journal of Chinese Universities*, 17(3):363–370, 2002.

[96] H. Hermassi, R. Rhouma, and S. Belghith. Security analysis of image cryptosystems only or partially based on a chaotic permutation. *Journal of Systems and Software*, 2012 (In Press).

[97] L. Zhao, A. Adhikari, and K. Sakurai. A New Scrambling Evaluation Scheme Based on Spatial Distribution Entropy and Centroid Difference of Bit-Plane. *Proceedings of the 9th International Workshop on Digital Watermarking (IWDW'10)*, Lecture notes in computer science: volume 6526, pages 29–44, Springer-Verlag, Heidelberg, October, 2010.

[98] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai. Cryptanalysis on an Image Scrambling Encryption Scheme Based on Pixel Bit. *Proceedings of the 9th International Workshop on Digital Watermarking (IWDW'10)*, Lecture notes in computer science volume 6526, pages 45–59, Springer-Verlag, Heidelberg, October, 2010.

[99] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai. Security Improvement of a Pixel Bit Based Image Scrambling Encryption Scheme Through the Self-correlation Method. *Proceedings of the 6th China International Conference on Information Security and Cryptology (INSCRYPT'10)*, (short paper):88–102, Science Press of China, October, 2010.

[100] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai. On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption. *Communications in Nonlinear Science and Numerical Simulations*, 17 (8):3303–3327, 2012.

[101] L. Zhao, T. Nishide, A. Adhikari, K.H. Rhee, and K. Sakurai. Cryptanalysis of Randomized Arithmetic Codes Based on Markov Model. *Proceedings of the 7th China International Conference on Information Security and Cryptology (INSCRYPT'11)*, Springer-Verlag, 2011 (In press).

Published Papers

Journal Papers

(1) L. Zhao, X.F. Liao, D. Xiao, T. Xiang, Q. Zhou, S.K. Duan. True random number generation from mobile telephone photo based on chaotic cryptography. *Chaos, Solitons & Fractals*, 42(3):1692–1699, 2009.

(2) L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai. On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption. *Communications in Nonlinear Science and Numerical Simulations*, 17 (8):3303–3327, 2012.

International Conference Papers with Review

(1) L. Zhao, D. Xiao, K. Sakurai. Image Encryption Design Based on Multi-dimensional Matrix Map and Partitioning Substitution and Diffusion-Integration Substitution Network Structure. *Proceedings of the 1st International Conference on Information Science and Applications (ICISA'10), (Track 5. Security and Privacy)*, Article number 5480269:pages 1–8, IEEE Society, April 2010.

(2) L. Zhao, A. Adhikari, and K. Sakurai. A New Scrambling Evaluation Scheme Based on Spatial Distribution Entropy and Centroid Difference of Bit-Plane. *Proceedings of the 9th International Workshop on Digital Watermarking (IWDW'10)*, Lecture notes in computer science: volume 6526, pages 29–44, Springer-Verlag, Heidelberg, October, 2010.

(3) L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai. Cryptanalysis on an Image Scrambling Encryption Scheme Based on Pixel Bit. *Proceedings of the 9th International Workshop on Digital Watermarking (IWDW'10)*, Lecture notes in computer science volume 6526, pages 45–59, Springer-Verlag, Heidelberg, October, 2010.

(4) L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai. Security Improvement of a Pixel Bit Based Image Scrambling Encryption Scheme Through the Self-correlation Method. *Proceedings of the 6th China International Conference on Information Security and Cryptology (INSCRYPT'10)*, (short paper):88–102, Science Press of China, October, 2010.

(5) L. Zhao, T. Nishide, A. Adhikari, K.H. Rhee, and K. Sakurai. Cryptanalysis of Randomized Arithmetic Codes Based on Markov Model. *Proceedings of the 7th China International Conference on Information Security and Cryptology (INSCRYPT'11)*, Lecture notes in computer science, Springer-Verlag, 2011 (In press).

(6) L. Zhao, T. Nishide, K. Sakurai. Differential Fault Analysis of Full LBlock. *Proceedings of the 3rd International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'12)*, Lecture notes in computer science: volume 7275 , pages 135–150, Springer-Verlag, Heidelberg, May, 2012.

Japanese Domestic Conference Papers without Review

(1) L. Zhao, D. Xiao, K. Sakurai. Image Encryption Design Based on Multi-dimensional Matrix Map and bS-D-wS Structure. *Proceedings of the 27th Symposium of Cryptography and Information Security (SCIS'10)*, CD-ROM 4F2-4, Kagawa, January, 2010.

(2) L. Zhao, K. Sakurai. Effective Digital Image Scrambling Evaluation Based on Bit-plane Selection. *Proceedings of the 27th Symposium of Cryptography and Information Security (SCIS'10)*, CD-ROM 4F2-5, Kagawa, January, 2010.

(3) L. Zhao, K. Sakurai. An Effective Attack Against a Chaos-based Image Scrambling Encryption. *IEICE Technical Report (ISEC)*, volume 109(445), pages 269–274, Nagano, March, 2010.

(4) L. Zhao, K. Sakurai. Image Encryption System Based on Self-correlation Permutation. *Proceedings of the 28th Symposium of Cryptography and Information Security (SCIS'11)*, CD-ROM 3E4-2, Kokura, January, 2011.

(5) L. Zhao, T. Nishide, A. Adhikari, K.H. Rhee, K. Sakurai. On the Insecurity of Randomized Arithmetic Codes Based on Markov Model. *IEICE Technical Report (ISEC)*, volume 111(285), pages 181–188, Osaka, November, 2011.

(6) L. Zhao, T. Nishide, K. Sakurai. Differential Fault Analysis on LBlock with Non-uniform Differential Distribution. *Proceedings of the 29th Symposium of Cryptography and Information Security (SCIS'12)*, CD-ROM 2C1-1E, Kanazawa, January (February), 2012.

Index