Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Supplementary materials for

Ming LI, Wenwen ZHOU, Mengdie WANG, Yushu ZHANG, Yong XIANG, 2025. SPJEU: a self-sufficient plaintext-related JPEG image encryption scheme based on a unified key. *Front Inform Technol Electron Eng*, 26(6):847-861. https://doi.org/10.1631/FITEE.2400721

1 Three image encryption schemes

An illustration of three different types of image encryption schemes is shown in Fig. S1. In Fig. S1a, the key generation of the traditional image encryption scheme is not associated with the plaintext image to be encrypted, and the security level is low. In Fig. S1b, the key generation of the plaintext-related image encryption scheme is associated with the plaintext image to enhance the security, but the cost of transmitting and managing too much additional data related to plaintext is high. In Fig. S1c, the key generation of the proposed SPEU is associated with the plaintext image, and no additional data are required for transmission and management.

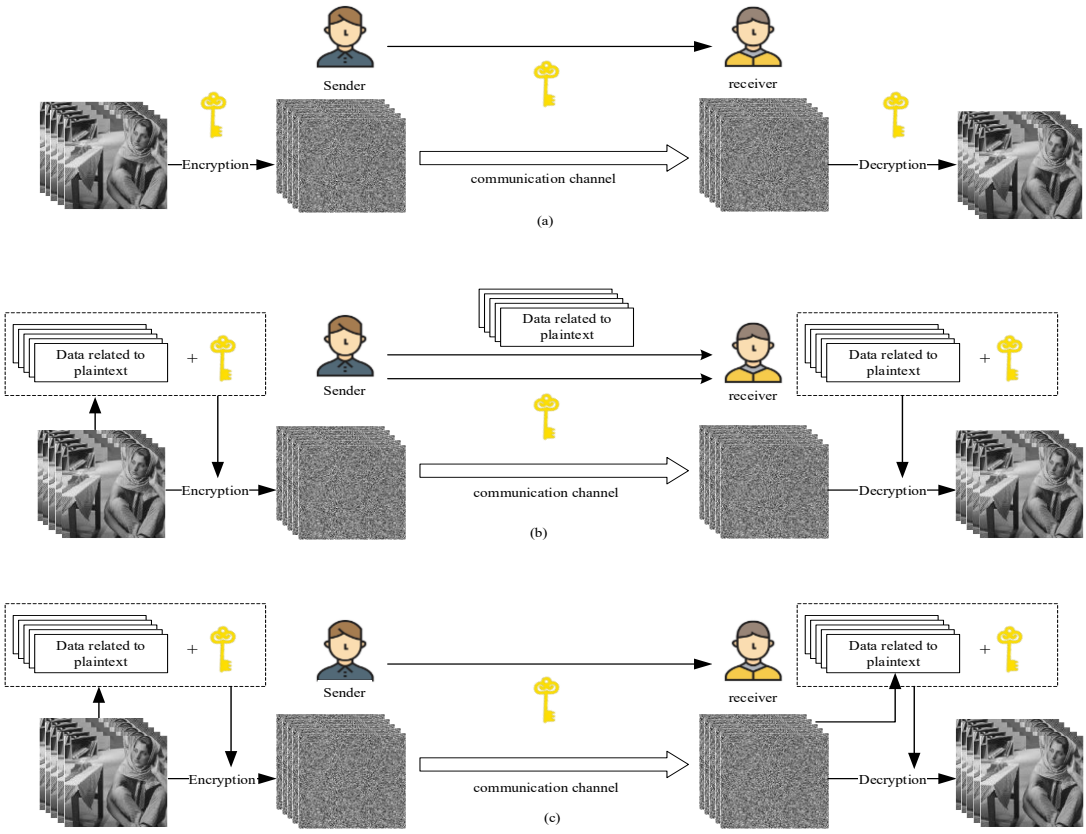

Fig. S1 Comparison of the three encryption mechanisms: (a) traditional image encryption mechanism; (b) image encryption mechanism associated with plaintext; (c) SPEU

2 Generation of key streams for DC and AC coefficients

The generation of the key streams required for encrypting the quantized DC coefficients and AC coefficients are both related to the plaintext. Specifically, the generation of the keystream for encrypting the quantized DC coefficients is related to the selected partial quantized DC coefficients. The initial key is generated by randomly and securely selecting some DC coefficients from all quantized DC coefficients through a pseudo-random number generator. The keystream required to encrypt the quantized AC coefficients is associated with all quantized DC coefficients. The specific process is as follows:

1. The generation of the key stream used for encrypting quantized DC coefficients

Step 1: Assuming that the size of the original image img is $M \times N$, the sub-blocks are sized 8×8, and the number of the sub-blocks $S = (M \times N)/64$. Using the unified keys key_1, key_2 as the initial value of the pseudo-random number generator, generate two pseudo-random sequences k_1 and k_2 of length S. Note that the pseudo-random sequences can be generated by any pseudo-random number generator, such as [34, 35]. Then k_1 and k_2 are processed by (S1) and (S2) to obtain H_1 and H_2.

$$H_1 = (\lfloor k_1 \times 10^{13} \rfloor) \, mod \, S + 1, \tag{S1}$$

$$H_2 = (\lfloor k_2 \times 10^{13} \rfloor) \bmod (\max + |\min| + 1). \tag{S2}$$

Step 2: Select the first 16 non-overlapped elements from the H_1 sequence to form sequence W. The obtained sequence W is random and unique, and the range of each element is [1, S]. Suppose the elements in W are the position indices of the sub-blocks of size 8×8. According to W, 16 sub-blocks of the image can be selected randomly, and the DC coefficients of the 16 sub-blocks are extracted to form a sequence B. Since there are some coefficients less than 0 in the DC coefficient, to facilitate the operation, B plus |min| is computed, as shown in (S3). The initial key is then generated by processing (S4) and (S5) on B_1.

$$B_1 = B + |\min|, \tag{S3}$$

$$L = \max + |\min| + 1, \tag{S4}$$

$$\begin{cases} \text{initial}_1 = (\text{sum}(\text{B}_1(1:8))) \bmod L. \\ \text{initial}_2 = (\text{sum}(\text{B}_1(9:16))) \bmod L. \end{cases} \tag{S5}$$

Step 3: The key stream required for the encryption of quantized DC coefficients is generated by a pseudo-random number generator and the initial values x_0 and y_0 are obtained by (S6), and two pseudo-random sequences z_1 and z_2 of length (S-16) are generated.

$$\begin{cases} x_0 = \text{key}_3 + \text{initial}_{1.} \\ y_0 = \text{key}_4 + \text{initial}_{2.} \end{cases} \tag{S6}$$

2. The generation of the key stream used for encrypting quantized AC coefficients

Step 1: The quantized DC coefficients in all sub-blocks are extracted to form the sequence D. Then D is processed by (S7).

$$\text{initial}_3 = \text{sum}(B)/(M \times N). \tag{S7}$$

Step 2: The initial values x_1, y_1 of the pseudo-random number generator are obtained by (S8). Two random sequences z_3, z_4 are generated by a pseudo-random number generator.

$$\begin{cases} x_1 = \text{key}_5 + \text{initial}_3. \\ y_1 = \text{key}_6 + \text{initial}_3. \end{cases} \tag{S8}$$

3 Example of DC coefficient encryption

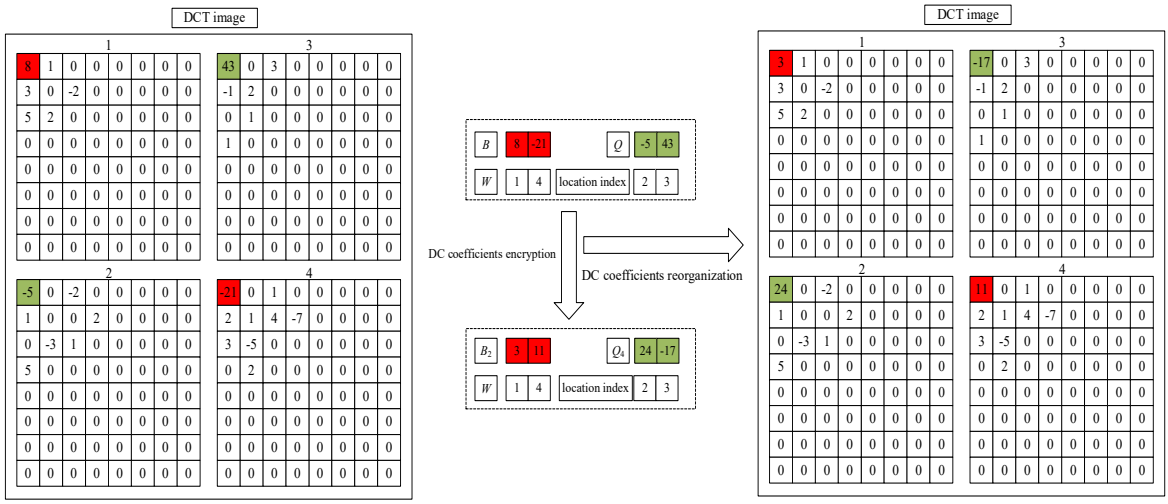

Fig. S2 Example of DC coefficient encryption

4 Proof of homomorphism

Proposition 1: Modulo addition (L)-based encryption is additively homomorphic.

Proof: Let b_1 and b_2 denote two different DC coefficients in B_1, respectively, and h_1 and h_2 denote random numbers used to encrypt b_1 and b_2 in key stream H_2. If $\odot_M$ is an arithmetic addition and $\odot_C$ is a modulo addition, that is:

$$\begin{aligned} E_n(b_1 \odot_M b_2,\ h_1 + h_2) &= E_n(b_1 + b_2, h_1 + h_2) \\ &= (b_1 + b_2 + \ h_1 + h_2)\text{mod}(L) \\ &= ((b_1 + h_1)\text{mod}(L) + (b_2 + h_2)\text{mod}(L))\text{mod}(L) \\ &= (E_n(b_1, h_1) + E_n(b_2, h_2))\text{mod}(L) \\ &= E_n(b_1, h_1) \odot_C E(b_2, h_2), \end{aligned}$$

then, the modulo-L encryption algorithm has the property of additive homomorphism.

Decrypt $E(b_1,h_1)\odot_C E(b_2,h_2)$ with h_1+h_2:

$$\begin{aligned} D_n(E_n(b_1, h_1) \odot_C E_n(b_2, h_2), h_1 + h_2) &= D(E_n(b_1 \odot_M \ b_2, h_1 + h_2), h_1 + h_2) \\ &= (b_1 \odot_M \ b_2)(mod(L)) \\ &= ((b_1 + b_2)\big(mod(L)\big). \end{aligned}$$

B_1 adopts the homomorphic encryption algorithm based on modulo L (L=max+|min|+1), the encrypted DC coefficient B_2 and the key stream used for encryption is H_2. Then B_2-|min|, so that the range of the encrypted DC coefficient is between [0, max+|min|]. B_2 is the encryption result of the DC coefficients B.

In the encryption of B, the 1st-7th and 9th-15th numbers in H_2 are generated randomly, but the 8th and the 16th are calculated to satisfy $(h_1 + h_2 + \cdots + h_8) \bmod L = 0$ and $(h_9 + h_{10} + \cdots + h_{16}) \bmod L = 0$ respectively, so that the plaintext-related information can be computed from the encrypted image easily and securely. For example:

$$\begin{aligned} &(E_n(b_1, h_1) + E_n(b_2, h_2) + \cdots + E_n(b_8, h_8))\mathrm{mod}(L) \\ &= \big((b_1 + h_1)mod(L) + (b_2 + h_2)mod(L) + \cdots (b_8 + h_8)\mathrm{mod}(L)\big)\mathrm{mod}(L) \\ &= (b_1 + b_2 + \cdots + b_8 + h_1 + h_2 \ldots h_8)\mathrm{mod}(L) \\ &= (b_1 + b_2 + \cdots + b_8)\mathrm{mod}(L). \end{aligned}$$

5 Example of AC coefficient encryption

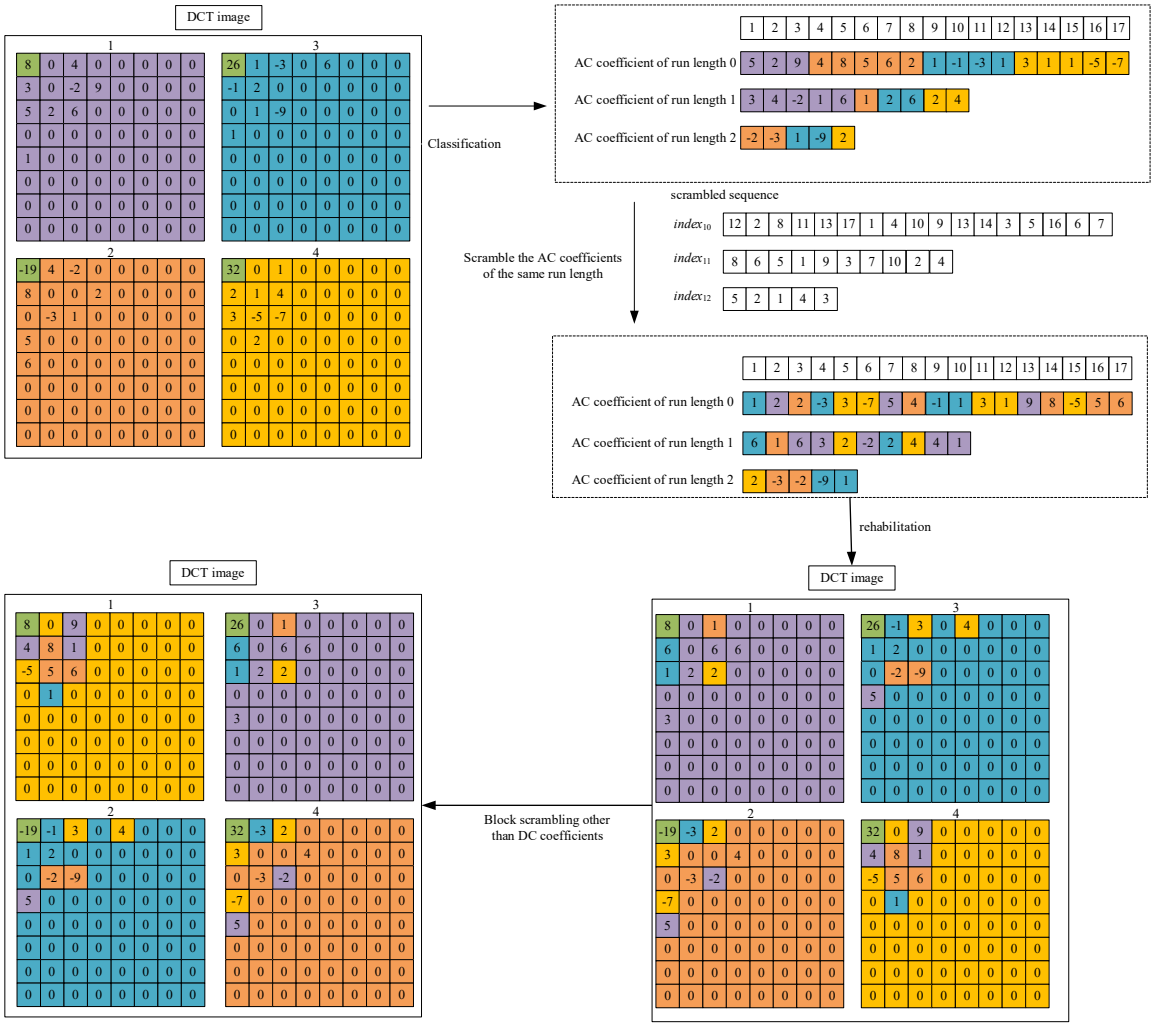

Fig. S3 Example of AC coefficient encryption

6 Example of AC coefficient run length distribution

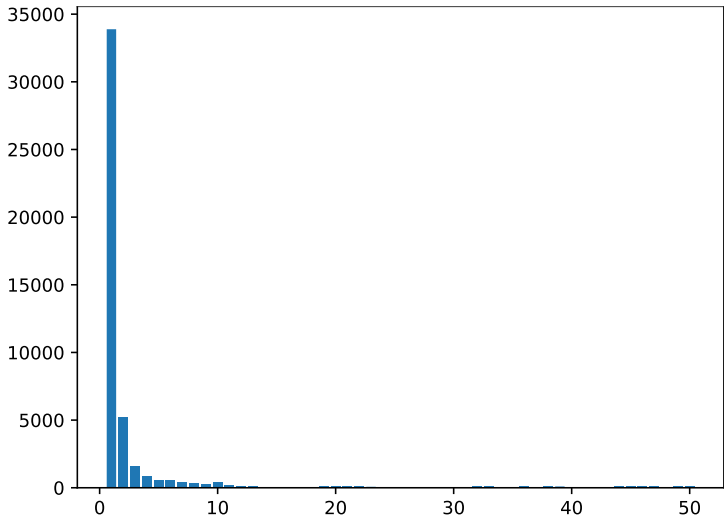

Fig. S4 AC coefficient run length distribution of grayscale image Barbara (256×256)

7 Flowchart of JPEG decryption

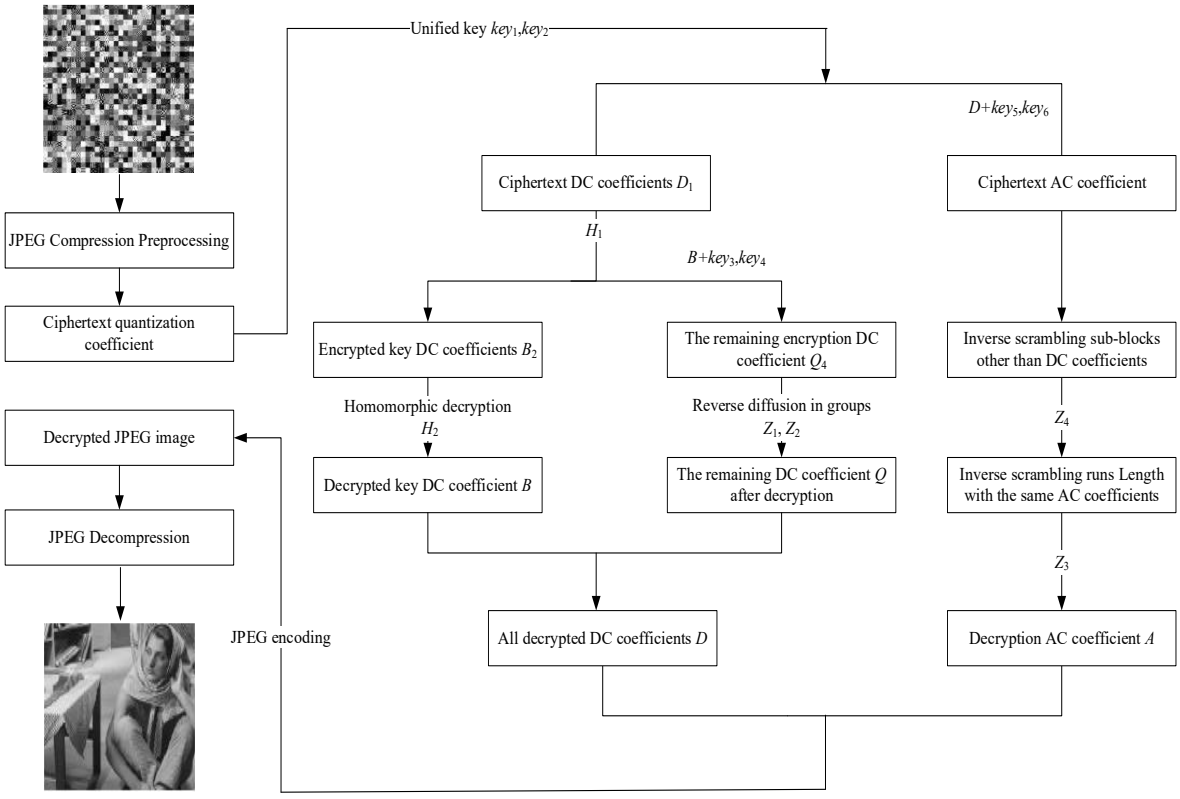

Fig. S5 Flowchart of JPEG decryption

8 Security proof of SPJEU

We used ciphertext-only attacks to analyze the security of SPJEU from two perspectives: the key generation mechanism and the JPEG image encryption algorithm. The proof goes as follows:

Suppose that the attacker obtains the ciphertext image without knowing the unified key, the size of the ciphertext image is $(M \times N)$, and the number of plaintext information contained in the ciphertext is $2a$.

8.1 Security analysis of the key generation mechanism in SPJEU

The encryption key of the SPJEU model consists of the unified key and related plaintext information, and the key space of the encryption key should be large enough to resist brute force cracking.

8.1.1 Security analysis of unified key

Suppose that the attacker decrypts the ciphertext in the form of brute-force keys, and there are six unified keys, namely key_1, key_2, key_3, key_4, key_5, and key_6. The precision of each key is 10^{16}, and the key space is $(10^{16})^6=10^{96}\approx 2^{318}$, which is much larger than 2^{100} (Alvarez and Li, 2006).

8.1.2 Security analysis of relevant plaintext information

To obtain information related to the plaintext from the ciphertext image, the attacker first needs to obtain the location of its plaintext information. The attacker needs a total of $C_{M\times N}^{a} \times C_{(M\times N-a)}^{a}$ attempts to successfully obtain the relevant plaintext location information, which means that the key space of the relevant plaintext information is $C_{M\times N}^{a} \times C_{(M\times N-a)}^{a}$. The length of the key space should be greater than 2^{100} to resist the brute-force attack of the attacker (Alvarez and Li, 2006). To ensure the security of the SPJEU model, the probability of the attacker successfully obtaining the location information should be less than $\frac{1}{2^{100}}$ (Alvarez and Li, 2006).

The probability that the attacker succeeds in obtaining the location information is

$$P_w = \frac{1}{C_{M\times N}^{a} \times C_{(M\times N-a)}^{a}},\ P_w = \frac{1}{C_{M\times N}^{a} \times C_{(M\times N-a)}^{a}} < \frac{1}{2^{100}}.$$

Taking the size of a 256×256 JPEG image as an example, in the SPJEU image encryption algorithm, we chose the information related to the plaintext as the quantized DC coefficient. The unified key selects 8 quantized DC coefficients each time, a total of 2 times, so a total of 16 relevant plaintext information is selected. A JPEG image of size 256×256 contains a total of 1024 quantized DC coefficients. That is, the probability of obtaining the location information is

$$P_w = \frac{1}{C_{1024}^{8} \times C_{1016}^{8}} \approx \frac{1}{8.5106\times 10^{38}} << \frac{1}{2^{100}} \approx \frac{1}{1.2677\times 10^{30}}.$$

Therefore, it is difficult to extract the information related to the plaintext from the ciphertext image without knowing the unified key, and the larger the image and the less information related to the plaintext is selected, the closer the probability of P_w is to 0, and the higher the safety factor.

8.1.3 The security analysis of the encryption key

Assuming that the attack successfully obtains the location information after $C_{M\times N}^{a} \times C_{(M\times N-a)}^{a}$ attempts, the plaintext information also needs to be combined with other unified keys to form an encryption key to crack the ciphertext.

There are four keys combined with the plaintext information: key_3, key_4, key_5, and key_6. The four keys have a precision of 10^{16}, and it takes brute-force cracking $(10^{16})^4 = 10^{64}$ times to obtain the key. The probability is $P_k = \frac{1}{10^{64}} \approx \frac{1}{2^{205}} < \frac{1}{2^{100}}$. Then the probability that the attacker gets the key is $P = P_w \times P_k = \frac{1}{C_{M\times N}^{a} \times C_{(M\times N-a)}^{a}} \times \frac{1}{2^{205}} << \frac{1}{2^{100}}$.

So, obviously, it is infeasible for the attacker to crack the ciphertext image by brute force.

8.2 Security analysis of SPJEU JPEG image encryption algorithm

Assuming that the attacker directly makes a ciphertext-only attack on the ciphertext image and performs brute-force cracking on each step of the algorithm, it takes *C* times to successfully crack the ciphertext image. Table S1 presents the number of brute-force attacks required at each step of the algorithm. Assuming that a 256×256 ciphertext image needs to be brute-force cracked, a total of exhaustive C=16!×252!×8391!×1024! times is required. This number is very large, and it is obviously not feasible to attack the ciphertext image by brute force.

Table S1 Number of times required for brute force cracking

Encryption steps	The number of times required for brute-force cracking
Encryption of *B* (DC coefficients that are used for computing)	$16!$ =2.0923×10^{13}
Encryption of *Q* (the remaining DC coefficients)	$(m+n)!$
Scrambling of the AC coefficients with the same run length	$\prod_{i=0}^{62} Y_i\,!$
Scrambling of all sub-blocks except DC coefficients	$S!$
Total number of times	$C = 16! \times (m+n)! \times \prod_{i=0}^{62} Y_i \times S!$

Reference

Alvarez G, Li S, 2006. Some basic cryptograpvhic requirements for chaos-based cryptosystems. *Int J Bifurc Chaos*, 16(08):2129-2151. https://doi.org/10.1142/S0218127406015970