Chapter

Hardware Implementation of an Improved Hybrid Cryptosystem for Numerical Image Encryption and Authenticity

Amal Hafsa, Jihene Malek and Mohsen Machhout

Abstract

Cryptography is the science that concerns protecting information by transforming its comprehensible form into an incomprehensible one. The conception of a robust cryptosystem is a challenge. In this paper, an improved hybrid cryptosystem for numerical image protection is presented. First, the initial secret key is generated by a secure hash function (keccak). Secondly, the plain image is encrypted through the advanced encryption standard (AES) with CTR mode. Finally, a Rivest-Shamir-Adleman (RSA) algorithm is used to secure the symmetric key transmitted over the insecure channel and owner signature. Our cryptosystem is implemented in hardware and evaluated by different tools mainly identified from the image cryptography community using numerous kinds of standard images. The experimental and analytical findings prove that our framework security gives a trade-off between robustness and performance, which can be used in several domains like medicine, military, and community privacy.

Keywords: hybrid scheme, encryption/decryption, signature/verification

1. Introduction

The digital image is a technology that contains secret information. In today's highly networked world, web servers, application servers, and database backend all connect through public or shared digital networks. In this environment, image is vulnerable to potentially more destructive attacks such as replay or human-based attacks, brute-force, statistical attack, etc. Cryptography is the ideal solution to protect numerical images in storing and moving. It provides a deep defense enough against last record. Evidently, security surrounds us daily at a personal level. Cryptography solutions should guarantee the confidentiality, integrity, and authentication of the secret data. The confidentiality is enabled by encryption that transforms secret data from readable forms to unintelligible forms. Encryption provides a good defense against the new generation of attacks. Even if systems are compromised and the information is taken, encryption can keep it unusable. Integrity is that any change in the transmitted data can be detected at the destination. However, authenticity is that the receiver can verify the identity of the sender. Among encryption schemes, hybrid scheme is considered the ideal idea to protect numerical images that permits both confidentiality and authenticity. Symmetric scheme is effective for large volume data encryption because it is generally hundreds to thousands of execution times faster than asymmetric scheme, but it suffers from secret key distribution. On the other hand, the asymmetric scheme is more secure than the symmetric scheme since it uses different pair keys for encryption than for decryption. However, the hybrid scheme is an innovative idea that combines the symmetric approach for large volume data encryption and the asymmetric approach for secret key exchanging and authentication.

In this paper, we make the following contributions:

- A strong hybrid framework using Keccak, RSA, and AES-256 is suggested, in which all security services are guaranteed.

- Undertake in-depth experimental measurements for several kinds of numerical images with different types, contents, and sizes to evaluate the robustness of the cryptosystem put forward.

- Undertake in-depth evaluation study of the performance of the execution and compare the findings with other works.

The rest of the paper presents the following sections: A survey of existing works is given in Section 2. In Section 3, a preliminary study of Keccak, AES, RSA, and counter encryption mode (CTR) are respectively described. The proposed methodology to create the overall algorithm is detailed in Section 4. Section 5 presents the evaluation and the security analysis of the technique put forward. Finally, the last section gives conclusions and related works.

2. Related works

In this section, we recall diverse related works that studied the encryption algorithms designed for securing digital images.

An image cryptosystem that combines chaos sequences and a modified AES algorithm is put forward in [1]. Firstly, the key is generated by Arnold's chaos sequence. Secondly, the plain image is ciphered by the modified AES and by implementing the round keys produced by the chaos system. In [2], the authors proposed a dynamic AES for numerical image encryption based on the logistic chaotic map and the advanced encryption standard (AES) with Galois mode. Utilizing the logistic mapping, the key is generated and mixed at different stages with the plain data. In [3], the authors put forward a secure framework using a chaos system-based S-box. However, a chaos system is employed to generate three sequences of random numbers, then, an S-box is performed. The image is ciphered by the XOR operation and the sub-byte function.

The main issue of these works is that they suffer from the secure transmission of the symmetric key. In [4], a secure method for color image protection is suggested. It is based on the elliptic curve and AES. Random numbers are generated by the elliptic curve, hence these numbers are employed for generating three maskers to cipher the three components-red, blue, and green-of the image. The drawback of this paper is that all operations are performed sequentially, which can degrade the system. In [5], the authors present a novel idea to protect images against attacks. The initial level is performed by applying the conformal mapping to the private data. Then the image result is encrypted and decrypted using the (RSA) algorithm. Thirty they use less significant bit (LSB) as the hiding method to hide the message inside the cover image. Finally, they propose to compress the image using GZIP. However, the use of an asymmetric algorithm to encrypt big data can degrade the system in terms of execution time. In [6], the authors propose a quantum logistic image cryptosystem that combines both RSA and SHA-3 algorithms. Firstly, the RSA is employed to randomly generate key pairs with private keys and public keys. A fixed matrix is then generated for the confusion. Secondly, the preprocessed image is calculated by the hash function SHA-3 to obtain the clear message that is then stored safely. Using the RSA algorithm, the encrypted message can be performed corresponding to the original message. After mixing both the original and the encrypted messages, the initial conditions of the quantum logistic map are computed using a novel mathematical technique. The main drawback of this paper is the use of an asymmetric algorithm that is a hard algorithm to encrypt the big data.

Our motivation for this work is to guarantee all services of cryptography with low computational time.

3. Preliminary study

The proposed security framework uses the SHA-3 for the initial key generation, the AES algorithm for the whole image encryption–decryption, and the RSA algorithm for the key exchange and authentication. More information on the suggested algorithms is given in this section.

3.1 Keccak

Hash functions get a finite arbitrary length of data as an input argument and produce an output data digest of a fixed length of bits. The keccak hash function is selected by the NIST as the contest winner SHA-3 [7]. The keccak enables the integrity of information such that a tiny alteration in the input data, with one bit, will cause a greatly significant change in the output. As a sequence, each data has its own data digest. It is considered a sponge function because it is based on construction sponge as shown in **Figure 1**. It can produce pseudo-random output with the desired length from an arbitrary length entry.

The keccak algorithm is the permutation f, which is repeatedly applied to a fixed data length ($b = r + c$ bits), where c and r are, respectively, the capacity and the throughput binary. Higher values of c correspond to a higher level of security then those higher values of r improve the speed. The sponge function consists of three phases: the first is the padding phase, where the entry is completed by a padding rule that produces an output length multiple of r, the bit rate. Second, the message is divided into blocks with each of r bits in length such that each block is XOR with the bit rate r. The result will then be concatenated with the capacity c to constitute a state that is an entry of the absorbing phase. In this phase, the state is absorbed by a function f, then, a calculation is repeated for a number determined of rounds. Finally, the output of the absorbent phase is extracted from its state (phase of compression) and constitutes the digest with the desired length. More details are given in [8].

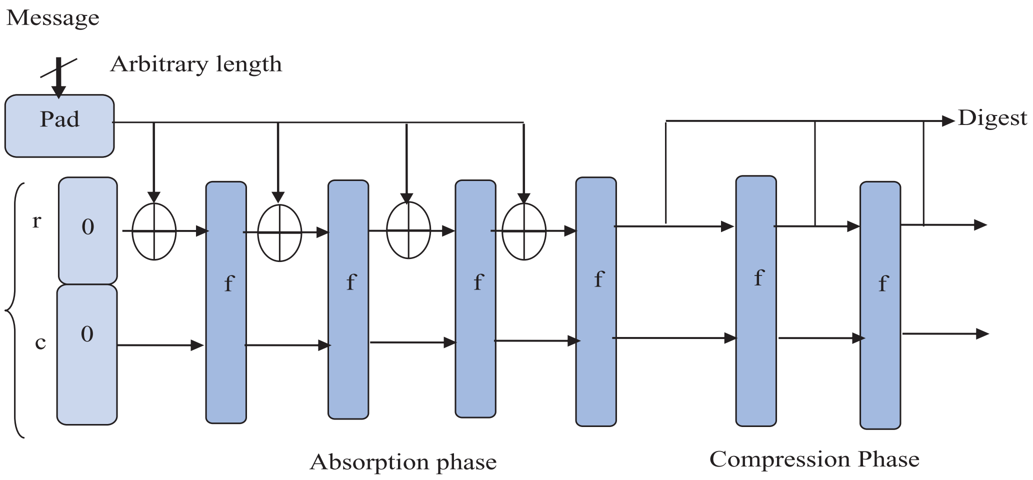


Figure 1.
The construction mechanism of the sponge.

3.2 AES

AES is a symmetric encryption standard considered by the FIPS as the rapid and the more efficient among symmetric algorithms [9]. AES is a block cipher algorithm; data is processed in 128-bit blocks for plaintext and ciphertext. The secret key is 128 bits long, hence the version name is AES 128 (there are two other variants whose keys are 192 and 256 bits). The key with 256 bits of length is the most secure to perform high security. The AES is composed of four main operations to perform both confusion and diffusion for the Shannon Principe, which are the AddRound key, the Subbytes, the Shiftrows, and the MixColumns. The key expansion is used to generate keys from an initial secret key. More information is given in [9]. The flow design of the AES is shown in the **Figure 2**.

3.3 CTR encryption mode

In this mode, the key flow is obtained by encrypting the successive values of a counter. This mode combines many advantages because it allows stream encryption and is pre-computable. In addition, it allows random access to data, is parallelizable, and only uses the encryption function. The counter used can be a pseudo-random sequence that will be easy to find from the seed (initialization vector). Encryption architecture is detailed in **Figure 3**.

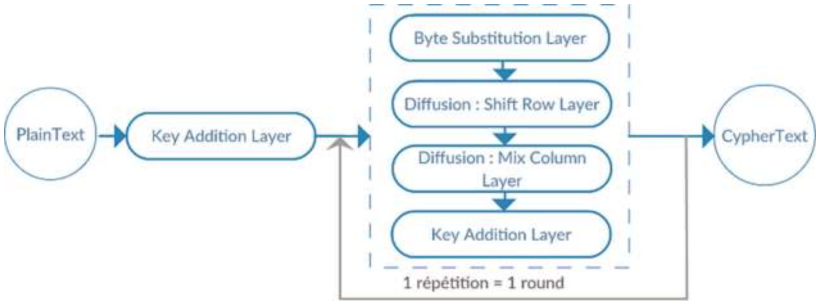


Figure 2.
Flow design of the AES.

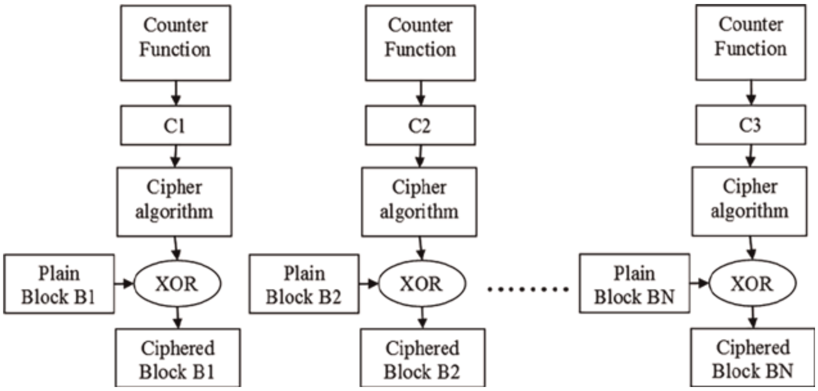


Figure 3.
General architecture of CTR encryption mode.

3.4 RSA algorithm

The RSA cryptosystem provides a digital signature scheme (sign + verify), based on the math of the modular exponentiations and discrete logarithms and the computational difficulty of the RSA problem (and its related integer factorization problem). The RSA sign/verify algorithm works as described below.

1. Key Generation Procedure
    1. Select two distinct big random prime numbers p and q.
    2. Calculate n by multiplying p and q.
    3. Calculate $\varphi(n) = (p-1)\ (q-1)$.
    4. Choose e, $1 < e < \varphi(n)$, which is relatively prime to $\varphi(n)$.
    5. Calculate d to perform $d \equiv e-1 \bmod \varphi(n)$; d is considered as a private key exponent.
    6. The public key is (n, e) and the private key is (n, d). Keep all the values p, q, and $\varphi(n)$ private.
2. RSA Sign
    1. Signing a data A with the private key exponent d
    2. Compute the digest of the message: h = hash(A)
    3. Encrypt h to calculate the signature
3. RSA Verify Signature
    1. Verifying a signature 's' for the message A with the public key exponent e

2. Compute the hash: h = hash(A)

3. Decrypt the signature: $h'=se(\mathrm{mod}n)$

4. Compare h with h' to find whether the signature is valid or not

5. If the signature is correct, then the following will be correct:

$$h' = se(\mathrm{mod}\, n) = (\mathrm{hd})\, e(\mathrm{mod}\, n) = h$$

4. Proposed cryptosystem

Hybrid cryptography framework presents the perfect idea for numerical image protection. It employs both symmetric and asymmetric algorithms. The first one is employed for large data encryption and the second one is to share the secret key and permits authentication by signature generation, fully related to the data and the sender. The keccak hash function is used to generate the initial secret key provided as the digest of the original image.

The flow design of the hybrid cryptosystem is given in **Figures 4** and 5.

4.1 Encryption process

The encryption process requires a symmetric algorithm to encrypt the plain image. For this, we have chosen to use the AES-256 algorithm with CTR mode for the encryption process because it is the rapid mode. However, AES requires an initial key Ki for key expansion, so we have proposed to generate the keccak hash function. On the other hand, the image is decomposed into blocks of 32 bytes. As a consequence,

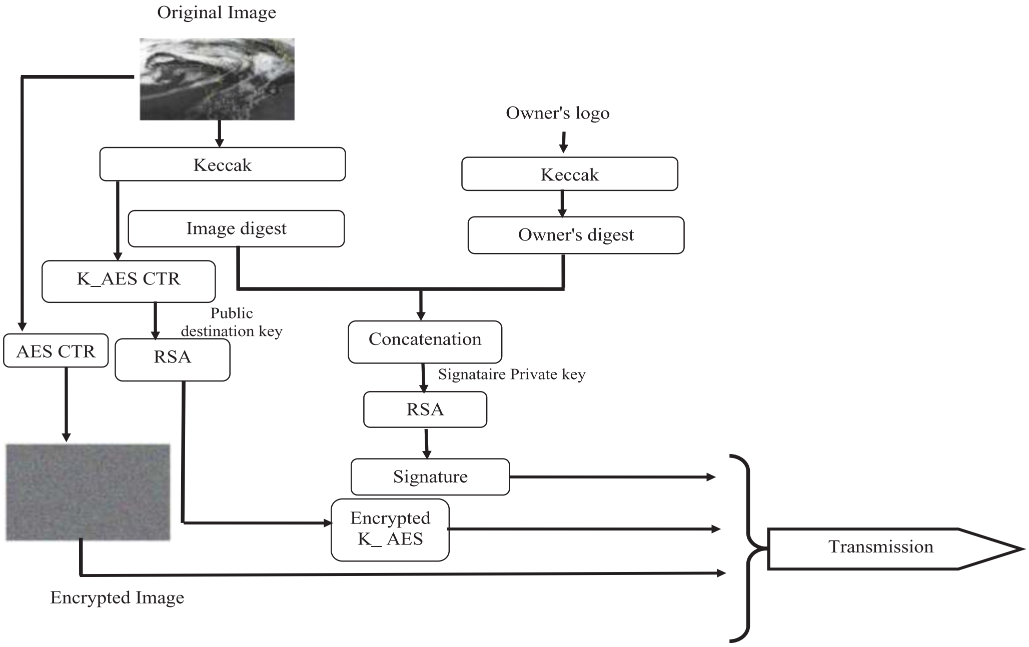


Figure 4.
Encryption and generation of the digital signature.

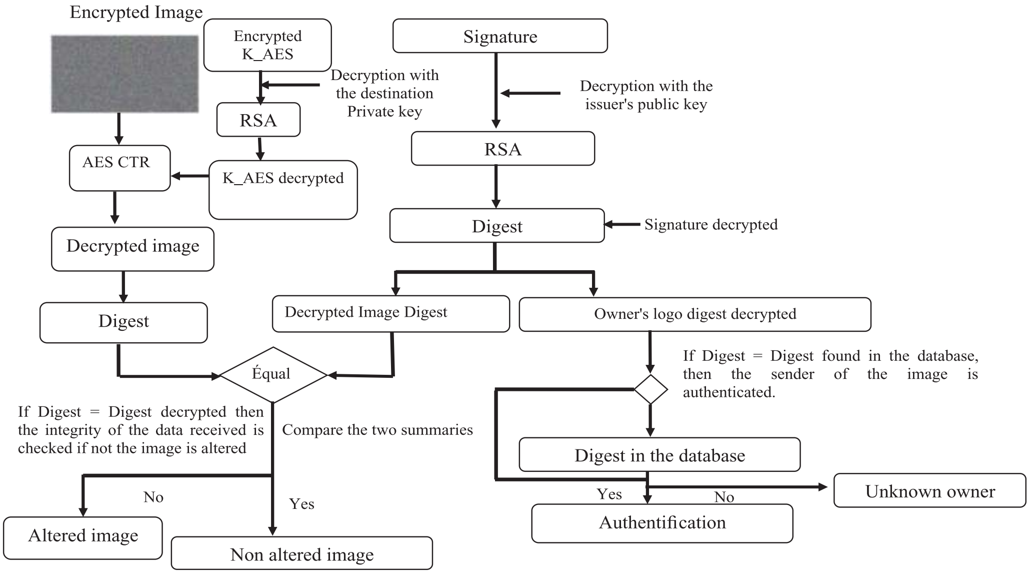

Encrypted Image
Encrypted K_AES
Decryption with the destination Private key
RSA
K_AES decrypted
AES CTR
Decrypted image
Digest
Signature
Decryption with the issuer's public key
RSA
Digest
Signature decrypted
Decrypted Image Digest
Owner's logo digest decrypted
Équal
If Digest = Digest decrypted then the integrity of the data received is checked if not the image is altered
Compare the two summaries
No
Yes
Altered image
Non altered image
If Digest = Digest found in the database, then the sender of the image is authenticated.
Digest in the database
Yes
No
Unknown owner
Authentification

Figure 5.
Decryption and verification of the digital signature.

the used encryption mode is the CTR. Thus, a counter function is employed for generating 256-bit count values, after that, it is encrypted by the AES-256 mechanism. On the other part, to secure the transmission of the initial key Ki, which is ciphered using the RSA algorithm.

4.2 Image authentication and the Key exchange

For image authentication, a 256-bit signature is generated fully related to the owner logo using keccak algorithm. The signature is combined with the 256-bit initial key and encrypted together by the RSA for more security. However, the initial key is encrypted using the public key of the receiver using the Eq. (1). At the reception, the private key is only used for decrypting the initial key, using Eq. (2),

$$K_f = K_i^e \; mod \; (n) \tag{1}$$

where K_i is the plain key, Kf is the correspondent encrypted key, and (n, e) is the public key of destination.

$$K_i = K_f^d \; mod \; (n) \tag{2}$$

where K_i is the decrypted key, K_f is the encrypted key, and (n, d) is the private key of the destination.

4.3 Decryption process

After the encryption process, ciphered image is sent to the destination over an insecure channel. At the reception, a decryption process should be done to restrain the plain image and detect the owner’s logo. This phase is the reverse of the encryption process. The algorithm starts by decrypting the encrypted key by the RSA system to obtain the plain key. The first part, 256-bit, is the initial key Ki used for image decryption, and the remaining part, 256-bit, is the owner's digital signature used for image authentication. However, using Ki, the received image is ciphered by the decryption system. This treatment enables finding the original image. On the other hand, using the digital signature with the help of a database, we detect the image owner’s logo.

5. Hardware implementation of the proposed hybrid cryptosystem

The suggested hardware architecture is performed with reduced resources that the embedded system makes available. Al cryptographic functions are required to design the hybrid secure framework operated in 32 bits datapath. In our design, six essential blocks are performed with the Control Unit, AES block, the RSA block, keccak block, and I/O buffer for 32-bit data bus. The control unit controls all algorithms and the information exchanged from external devices. Buffer in and buffer out are necessary for communicating the data from and to the on-chip bus. The AES block is used for data encryption and decryption. The keccak block is designed for hashing the message and finally, the RSA block is performed for the signature generation and verification. The proposed hardware architecture is given in **Figure 6**.

The suggested hybrid cryptographic framework is implemented on the DE2–115 board featuring Cyclone IV.E FPGA. **Table 1** gives the utilization of resources when synthesized by the Quartus II tools. The system necessitates 80% of logic elements, 79% of combinational functions, 27% of logic registers, and 7% of memory and consumes 226.15 mW. Concluding the obtained results, the proposed cryptosystem hardware design occupies a small hardware area and consumes reduced power. Thus, the proposed cryptosystem meets the constraints of onboard systems.

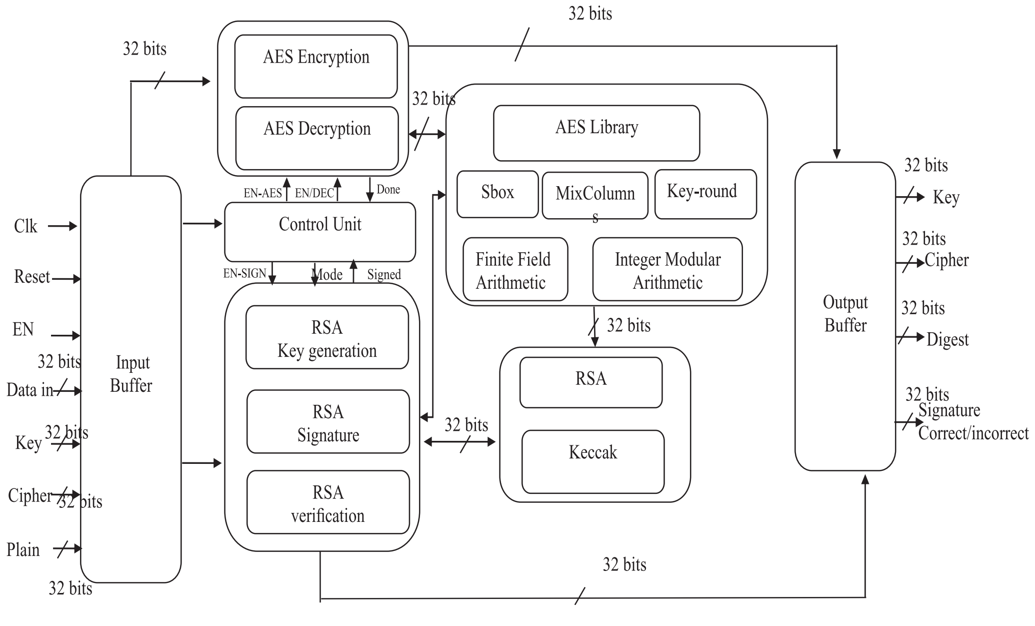


Figure 6.
Hardware architecture of the proposed cryptosystem.

Hardware performances	Resources utilizations
Total logic elements	91.584 (80%)
Total combinational functions	90.439 (79%)
logic registers	30.90 (27%)
Memories	273.408/3981312 (7%)
Total power consumption (mW)	226.15

Table 1.
Hardware resources results.

6. Evaluation and security analysis

To prove the robustness of the proposed security framework, we evaluate it using various metrics or tools for different standard images. This section contains the statistical analysis, the key analysis, Know Plain Text and Chosen Plain Text Attack, and the speed.

6.1 Statistical analysis

Statistical analysis indicates the factor of coincidence between plain and cipher images to test the ability of the algorithm to defend against attacks. Statistical analysis can include the image histogram, the entropy, the normalized correlation, the correlation coefficient (ρ), the peak signal to noise ratio ($PSNR$), and structural similarity index measure ($SSIM$) tools.

6.1.1 Histogram analysis

The histogram represents the data distribution in function of the pixel's values. A perfect secure algorithm must create a cipher image with uniform and totally different histograms compared to the original images. We analyze the histograms of original and encrypted samples and the contents are very different as shown in **Figure 7**. As a consequence, it is so hard to understand the encrypted image's appearance.

6.1.2 PSNR, SSIM, and entropy analysis

After encryption, the entropy factor, the PSNR, and SSIM are calculated as reported in **Table 1**. Practically, a perfect cryptosystem performs random information equal to "8" [10]. As given in **Table 2**, we constate that the entropy value of the cipher sample is close to the ideal value "8." This means that our algorithm put forward can resist statistical attacks. When analyzing the PSNR results. We note that the value is lower than 5 dB (<5 dB). Following the reference [11], the suggested cryptosystem gives a bad quality between both plain and cipher samples. Thus, it is hard to predict the original image from the cipher one. When turning to the SSIM factors computed between the original and the cipher images, we constate that values are close to 0. Therefore, we cannot extract the content of the clear sample from the cipher one. A comparative study of other existing work for entropy and PSNR evaluation tools is

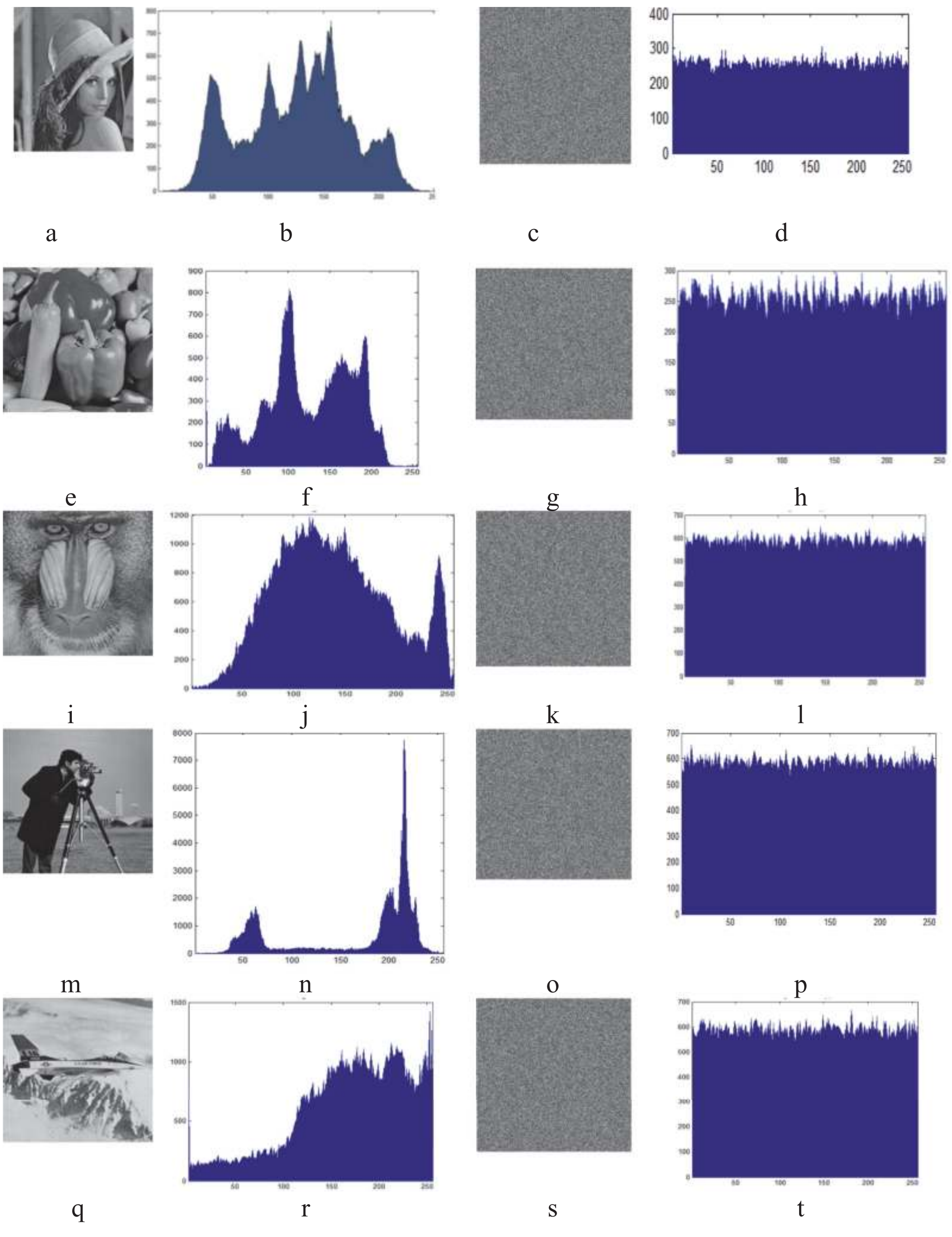


Figure 7.
Histogram of the plain samples and their corresponding of the cipher samples.

reported in **Table 3**. This comparison lets us conclude that our cryptographic model outperforms the other works and can resist statistical attacks.

6.1.3 Correlation coefficient analysis

The standard plain sample is characterized by a correlation close to 1. Although in a cipher sample the adjacent pixels should be uncorrelated [15]. Let x and y two gray

Standard image	E	PSNR	SSIM
Lena	7.99975	4.521	0.0101
Peppers	7.99972	4.765	0.0089
Baboon	7.99972	4.580	0.0102
Walkbridge	7.99975	4.769	0.0086
Cameraman	7.99972	4.395	0.0089
Jetplane	7.99936	4.378	0.0105

Table 2.
E, PSNR, SSIM, and NC values of cipher samples.

Algorithm	PSNR	E
Ref. [12]	_	7.99920
Ref. [3]	_	7.99122
Ref. [13]	10.0454	7.75970
Ref. [6]	–	7.9993
Ref. [14]	0.9756	7.5631
Proposed algorithm	4.521	7.99975

Table 3.
Comparative study of PSNR and E values for Lena image.

scale parameters of two adjacent pixels in the sample, and the correlation of the adjacent pixels is evaluated by the following Eqs. (3)–(6).

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x_i))^2 \tag{4}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))\left(y_i - E(y)\right) \tag{5}$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{6}$$

Table 4 demonstrates that the cipher sample's correlation coefficients are close to 0. Therefore, we cannot extract the content of the plain sample from the cipher one. **Figure 8** illustrates the distributions of 2000 pairs of randomly selected adjacent pixels of the clear and ciphered Lena image.

Findings show that the correlation coefficient of the clear samples is close to 1, while the cipher images are close to zeros. Similarly, the distribution of adjacent pixels is inconsistent; i.e., there is no correlation between them. This proves that the cryptosystem eliminates the correlation of adjacent pixels in the clear sample, and it makes a ciphered sample with no correlation. A comparative evaluation with some existing

Image	Status	Horizontal	Vertical	Diagonal
Lena	Original	0.80482	0.81942	0.82776
	Cipher	0.000359	−0.0001339	−0.007713
Peppers	Original	0.84679	0.97335	0.88347
	Cipher	0.000638	−0.06850	−0.05433
Baboon	Original	0.83045	0.87679	0.87929
	Cipher	0.000757	−0.000602	−0.0010326
Walkbridge	Original	0.87108	0.87934	0.84997
	Cipher	0.00093	0.00380	−0.00776
Cameraman	Original	0.87790	0.89079	0.90087
	Cipher	0.000216	−0.00774	−0.00580
Jetplane	Original	0.80839	0.81643	0.82083
	Cipher	−0.005928	0.00381	−0.0015028

Table 4.
ρ values of clear image and its correspondent cipher image.

work for the correlation coefficient is tabulated in **Table 5**. The comparison proves that our system put forward gives the perfect results. Thus, the system can resist the statistical attacks.

6.2 Key analysis

To validate the robustness of the suggested algorithm, the key space, the key sensitivity, and the randomness analysis are tested in this section.

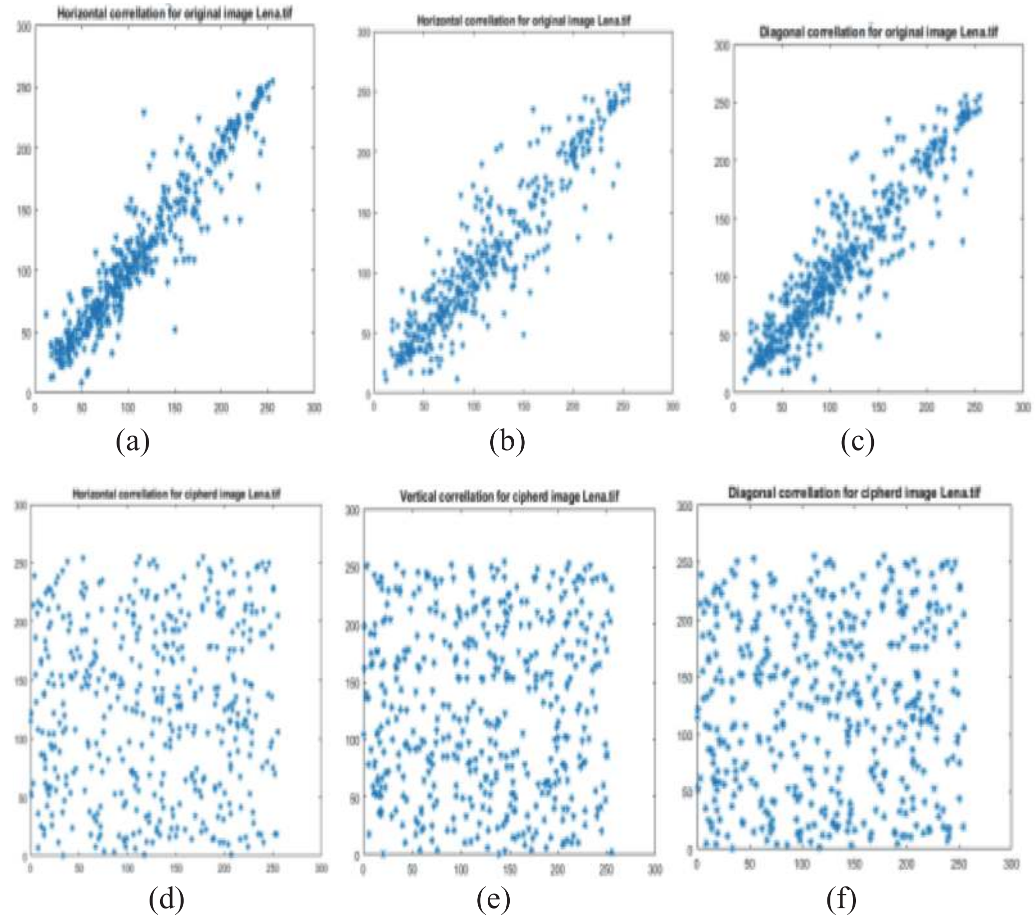
6.2.1 Key space

According to [18–20], the key space of a robust cryptosystem must be large to be protected from the brute-force hacker. In our algorithm, for an initial key Ki, there are 2^{128} dissimilar keys, which are very large. Therefore, the key brute-force attacks are computationally infeasible.

6.2.2 Key sensitivity

To assure a high level of protection, the cryptographic algorithm must be sensitive to the sample input and the initial secret key Ki. A simple alteration (one bit) in Ki, or in the sample, will cause a greatly significant modification in the output generated keys for encryption and output sample. The parameter Key sensitivity can be performed by employing the number of changing pixel rate (*NPCR*) and unified averaged changed intensity (*UACI*) randomness tests to test the force of the algorithm to defend against differential attacks [20], which are described as follows:

$$NPCR = \frac{1}{S}\sum D(i,j) \times 100\% \qquad (7)$$



(a) (b) (c)

(d) (e) (f)

Figure 8.
Distribution of 2000 pairs of randomly chosen adjacent pixels for Lena image: (a)–(c): horizontal, vertical, and diagonal distribution of the original image; (d)–(f): horizontal, vertical, and diagonal distribution of the cipher image.

Work	Horizontal	Vertical	Diagonal
Ref. [12]	0.0265	0.0792	0.0625
Ref. [3]	−0.0001	0.0089	0.0091
Ref. [15]	0.0036	0.0035	0.0064
Ref. [16]	0.0040	0.0015	0.030
Ref. [17]	0.0002	−0.0133	−0.0791
Proposed system	0.0003	−0.0001	−0.00771

Table 5.
Comparative study of correlation coefficient for Lena image.

$$UACI = \frac{1}{S}\sum\frac{|d|}{G} \times 100\% \qquad (8)$$

where S is the size of the image, and $D(i, j)$ is a logical value affected by the following cases:

$$D(i,j) = \begin{cases} 0 \textit{ if } I_1(i,j) = I_2(i,j) \\ 1 \textit{ if } I_1(i,j) \neq I_2(i,j) \end{cases} \tag{9}$$

d is the variance between two pixels on the sample with the same coordinates.

$$d = p_1(i,j) - p_2(i,j) \tag{10}$$

Thus, a sensitivity test applied to the initial key is evaluated by two initial keys, Ki1 and Ki2, where the key Ki2 is dissimilar by only one bit from the initial key Ki1 to cipher the same input. Then, we try deciphering the obtained samples with a wrong key. Here, the two keys, Ki1 and Ki2, are permuted in the decryption step; i.e., each image is decrypted by a wrong key, which is different by one bit from the correct key. This test is carried out with many Ki keys, which are randomly selected to properly evaluate the algorithm. Simulation results for the Lena image are shown in **Figure 9** and **Table 6**.

When analyzing results, we can conclude that our encryption system is very sensitive to small modifications in the entered sample. This proves the efficacity of the keccak hash function, which puts an image's initial key specific for encryption.

A comparative study is given in **Table 7** to evaluate the system compared to other related work and the results prove that the system is robust.

6.3 Know plain text and chosen plain text attack

This type of hacker is used to crack some of the cryptographic algorithms. Usually, an adversary employs black or white samples to extract the possible patterns in the algorithm. The white and dark samples are encrypted by the proposed method.

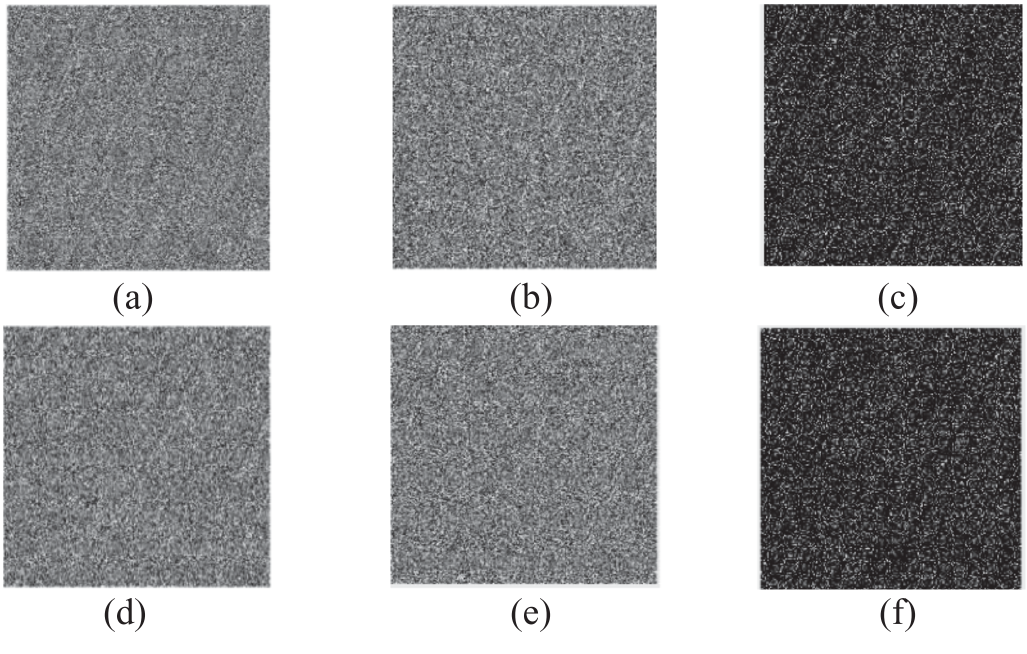


Figure 9.
Key sensitivity test applied on initial key, (a) ciphered Lena by Ki_1, (b) ciphered image by key Ki_2, (c) variance between (a) and (b), (d) deciphered (a) by Ki_2, (e) deciphered (b) by Ki_1, and (f) variance between (d) and (e).

Image	*NPCR* (%)	*UACI* (%)
Lena	99.7129	33.2936
Peppers	99.7698	33.2384
Baboon	99.75895	33.9487
Walkbridge	99.6789	33.45796
Cameraman	99.69875	33.60235
Jetplane	99.76589	33.38546

Table 6.
NPCR *and* UACI *tests applied on the initial key.*

Methods	*NPCR* (%)	*UACI* (%)
Ref. [21]	99.63230	33.12500
Ref. [22]	99.61000	33.53000
Ref. [23]	100.0000	33.43150
Ref. [24]	99.58948	33.46458
Ref. [25]	99.62530	33.48070
Proposed algorithm	99.7129	33.2936

Table 7.
Comparative evaluation of NPCR and UACI parameters for Lena image.

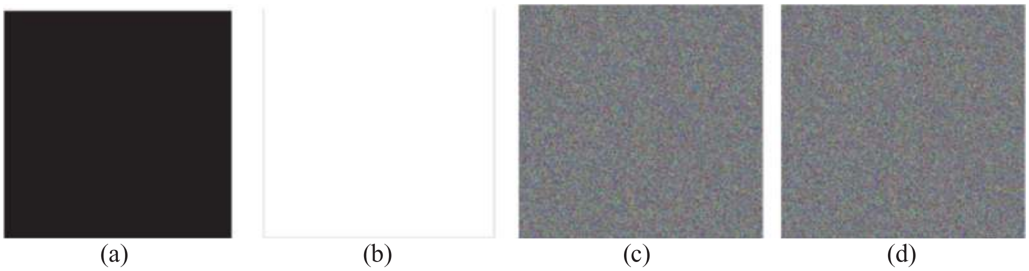


Figure 10.
(a) Black image, (b) white image, (c) ciphered black image, (d) ciphered white image.

Figure 10 gives the encrypted samples and no pattern is apparent. The value of the entropy for samples is selfsame as in other images and correlation coefficients are perfect. **Table 8** tabulates the correlation between adjacent pixels and the entropy values of both samples. Our proposed hybrid scheme is very resistant to attacks.

6.4 Encryption algorithm speed

In real-time application, the run time is considered as a main constraint. Eqs. (11) and (12) are used to calculate the speed (S) and the number of cycles per byte obtained by a specific algorithm running on the processor.

Image	Correlation coefficients			
	Entropy	Horizontal	Vertical	Diagonal
Black	0	—	—	—
Cipher Black	7.99975	−0.0026	0.00079	0.0002
White	0	—	—	—
Cipher White	7.99975	0.00026	−0.00201	0.00056

Table 8.
Entropy and correlation values.

Work	Used tools	*S (MB/S)*	*CpB*
Ref. [4]	Elliptic curve + AES CBC	0.002	1,400,000
Ref. [26]	Logistic-Tent and Tent-Sine	0.19	10,520
Ref. [27]	Chaos system + DWT	0.019	147,368
Ref. [28]	SHA2 + AES + RSA	0.38	8947
Proposed Algorithm	KECCAK +AES-CTR+ RSA	0.45	7.555

Table 9.
Performance results and comparison with the state of the art.

$$S = \frac{data\ size}{Time} MB/s \tag{11}$$

$$number\ of\ cycles/byte = \frac{Frequency}{S} \tag{12}$$

A comparison with related works is given in this state (**Table 9**). The comparison proves that the suggested scheme has the best findings in terms of speed.

7. Conclusion and future work

In this paper, we have put forward a strong hybrid cryptographic framework for image encryption, decryption, and authentication. The algorithm has combined a hash function and symmetric and asymmetric algorithms. The keccak hash function has been used for the initial secret key generation related to the plain image and owner's signature. The RSA encryption system has been used for the secret key exchanging and authentication. For image encryption, we have utilized the AES-256 bits with CTR mode. The main advantages of this mode are the rapidity of treatment and the no error propagation. The evaluation and analysis results prove that our proposed algorithm allows high performance and security. It can resist most known cryptanalysis attacks. For future work, we aim to design a mechanism for dynamically changing the S-box values of the AES.

Author details

Amal Hafsa[1*], Jihene Malek[1,2] and Mohsen Machhout[1]

1 Electronics and Micro-Electronics Laboratory, University of Monastir, Monastir, Tunisia

2 Department of Electronics, Sousse University, Higher Institute of Applied Sciences and Technology, Sousse, Tunisia

*Address all correspondence to: hafsaamal12@gmail.com

References

[1] Arab A, Rostami MJ, Ghavami B. An image encryption method based on chaos system and AES algorithm. The Journal of Supercomputing. 2019;**75**: 6663-6682. DOI: 10.1007/s11227-019-02878-7

[2] Shariat M, Mohammad Z, Rostami J, Eftekhari M. Proposing a novel dynamic AES for image encryption using a chaotic map key management approach. Optik. 2021;**246**:167779. DOI: 10.1016/j.ijleo.2021

[3] Unal C. Secure image encryption algorithm design using a novel chaos-based S-Box. Chaos, Solitons Fractals. 2017;**95**:92-101

[4] Toughi S. An image encryption scheme based on elliptic curve pseudo random and advanced encryption system. SIG Process. 2017;**141**:217-227

[5] Alawsi WA, Obayes HK, Hussain SM. A novel image encryption approach for IoT applications. Webology. 2022;**19**. DOI: 10.14704/WEB/V19I1/WEB19107

[6] Ye G, Jiao K, Huang X. Quantum logistic image encryption algorithm based on SHA-3 and RSA. Nonlinear Dynamics. 2021;**104**:2807-2827. DOI: 10.1007/s11071-021-06422-2

[7] Sideris A, Sanida T, Dasygenis M. High throughput pipelined implementation of the SHA-3 cryptoprocessor. In: 2020 32$^{\text{nd}}$ International Conference on Microelectronics (ICM); Aqaba, Jordan. 2020. pp. 1-4. DOI: 10.1109/ICM50269.2020.9331803

[8] Christof Paar, Jan Pelzl, SHA-3 and the Hash Function Keccak, An Extension Chapter for Understanding Cryptography—A Textbook for Students and Practitioners Springer, (2012). Available from: www.crypto-textbook.comm

[9] FIPS PUB 197: Advanced Encryption Standard (AES). Computer Security Standard, Cryptography. 2001

[10] Dridi M et al. Cryptography of medical images based on a combination between chaotic and neural network. Journal of Image Processing IET. 2016; **11**(5):324-332

[11] Melo A et al. Chapter 11: PriorityQoE: Atool for Improving the QoE in Video Streaming. Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools. 2013. DOI: 10.4018/978-1-4666-2833-5

[12] Xiuli C et al. A color image cryptosystem based on dynamic DNA encryption and chaos. Journal of Signal Processing. 2019;**155**:44-62. DOI: 10.1016/j.sigpro.2018.09.029

[13] Suri S et al. An AES–CHAOS-Based Hybrid Approach to Encrypt Multiple Images, International Conference in Recent Developments in Intelligent Computing, Communication and Devices. Springer; 2017. pp. 37-43. DOI: 10.1007/978-981-10-3779-5_6

[14] Shi M, Guo S, Song X, Zhou Y, Wang E. Visual secure image encryption scheme based on compressed sensing and regional energy. Entropy (Basel). 2021;**13**. DOI: 10.3390/e23050570

[15] Zhang W, Wong K -w, Yu H, Zhu Z-l. An image encryption scheme using reverse 2-dimentional chaotic map and dependent diffusion. Communications in Nonlinear Science and Numerical Simulation. 2013;**18**:2066-2080. DOI: 10.1016/j.cnsns.2012.12.012

[16] Kaur M, Singh D. Multiobjective evolutionary optimization techniques based hyperchaotic map and their applications in image encryption. Multidimensional Systems and Signal Processing. 2021;**32**:281-301. DOI: 10.1007/s11045-020-00739-8

[17] Gafsi M, Hajjaji MA, Malek J, Mtibaa A. Efficient encryption system for numerical image safe transmission. Journal of Electrical and Computer Engineering. 2020;**8937676**:12. DOI: 10.1155/2020/8937676

[18] Hafsa A, Sghaier A, Malek J, Machhout M. Image encryption method based on improved ECC and modified AES algorithm. Multimedia Tools and Applications. 2021;**80**:19769-19801. DOI: 10.1007/s11042-021-10700-x

[19] Hafsa A, Gafsi M, Malek J, Machhout M. FPGA implementation of improved security approach for medical image encryption and decryption. Scientific Programming. 2021;**2021**: Article ID 6610655, 20 p. DOI: 10.1155/2021/6610655

[20] Hafsa A, Fradi M, Sghaier A, et al. Real-time video security system using chaos-improved advanced encryption standard (IAES). Multimedia Tools and Applications. 2022;**81**:2275-2298. DOI: 10.1007/s11042-021-11668-4

[21] Ramzi G et al. A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2. Nonlinear Dynamics. 2016;**83**(3): 1123-1136

[22] Jiahui W et al. Color image encryption based on chaotic systems and elliptic curve ElGamal scheme. Journal of Signal Processing. 2017;**141**:109-124

[23] Khalaf A. Fast image encryption based on random image key. International Journal of Computer Applications. 2016;**3**:0975-8887

[24] Akram B et al. Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. Journal of Optics and Lasers in Engineering. 2017;**88**:37-50

[25] Rim Z et al. Image encryption based on new beta chaotic maps. Journal of Optics and Lasers in Engineering Elsevier. 2017;**96**:39-49

[26] Ye H-S et al. Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion. Journal of Signal Processing. 2020;**175**. DOI: 10.1016/j.sigpro.2020.107652

[27] Ahmed N. Timing filter for counter mode encryption. In: 2013 2nd National Conference on Information Assurance (NCIA). 2013. DOI: 10.1109/ncia.2013.6725333

[28] Gafsi M, Malek J, Ajili S, Hajjaji MA, Mtibaa A. High securing cryptography system for digital image transmission. SETIT 2018 SIST. 2020;**146**:311-322. DOI: 10.1007/978-3-030-21005-2_30