IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 5, 2022, accepted January 15, 2022, date of publication January 20, 2022, date of current version January 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3145005

Secure Image Encryption Based on Compressed Sensing and Scrambling for Internet-of-Multimedia Things

JAEPHIL CHOI, (Graduate Student Member, IEEE),
AND NAM YUL YU, (Senior Member, IEEE)
School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea

Corresponding author: Nam Yul Yu (nyyu@gist.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under NRF-2017R1A2B4004405.

ABSTRACT In this paper, we propose a secure image encryption system based on compressed sensing (CS) with a scrambling mechanism. For efficient encryption, we use a sparse measurement matrix, where the nonzero elements are generated by a linear feedback shift register (LFSR) based keystream generator. Then, several pairs of data scramblers, also based on LFSR, are attached behind the CS-encryption for diffusion. While guaranteeing theoretically reliable CS-decryption, numerical results indicate that the proposed cryptosystem has more reliable CS-decryption performance than other CS-based cryptosystems. Examining the histogram, entropy and correlation of the ciphertexts, experimental results demonstrate that the proposed cryptosystem has strong statistical security. Moreover, it turns out that the proposed cryptosystem has higher plaintext sensitivity than other CS-based cryptosystems, thanks to the bit-level diffusion from the scramblers.

INDEX TERMS Compressed sensing, diffusion, image encryption, plaintext sensitivity, scramblers, statistical security.

I. INTRODUCTION

Compressed sensing (CS) is a signal processing technique which allows to reconstruct a sparse signal from the incomplete measurements sampled at a rate lower than the Nyquist rate [1]. We say that a signal $\mathbf{x} \in \mathbb{R}^N$ is K*-sparse* with respect to an orthonormal basis $\mathbf{\Psi} \in \mathbb{R}^{N\times N}$ if $\boldsymbol{\alpha} = \mathbf{\Psi x}$ has at most K nonzero elements, where $K \ll N$. In CS, the sparse signal $\mathbf{x}$ is measured by $\mathbf{y} = \mathbf{\Phi x} \in \mathbb{R}^M$, where $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ is a measurement matrix with $M \ll N$. The reconstruction of the original signal $\mathbf{x}$ can be achieved by solving an l_1-minimization problem [2]. Thanks to the efficient data acquisition, there have been various CS applications, such as communications [3], [4], bio signal processing [5], image processing [6], RFID identification [7], etc. In particular, CS can be an attractive choice for data acquisition in a resource-limited transmitter of the Internet of Things (IoT) [8]–[11].

The CS techniques can be applied in a symmetric-key cryptosystem for information security. In CS-based cryptosystems, a plaintext $\mathbf{x}$ is encrypted to a ciphertext $\mathbf{y} = \mathbf{\Phi x}$ using a secret matrix $\mathbf{\Phi}$. A sender and a legitimate recipient share a secret key to generate the elements of the secret matrix $\mathbf{\Phi}$. The legitimate recipient who knows $\mathbf{\Phi}$ can reconstruct the original plaintext through CS reconstruction algorithms. Rachlin and Baron [12] showed that the CS-based cryptosystem cannot be perfectly secure but might be computationally secure. Orsdemir *et al.* [13] claimed that estimating the random Gaussian secret matrix of the CS-based cryptosystem is computationally infeasible. In [14], Bianchi *et al.* proved that CS-based cryptosystems can resist known-plaintext attacks (KPA) by encrypting plaintexts in a *one-time-sensing (OTS)* manner, where the secret matrix is renewed at every encryption. Furthermore, they showed that the *Gaussian-OTS (G-OTS)* cryptosystem, which uses a random Gaussian secret matrix in the OTS manner, can be perfectly secure if the plaintext has constant energy. In [15], it is quantitatively shown that the *Bernoulli-OTS (B-OTS)* cryptosystem, which uses a random Bernoulli matrix to encrypt a plaintext in the OTS manner, can resist against KPA. Moreover, the indistinguishability [16] was discussed for the G-OTS and the B-OTS cryptosystems [17], respectively.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran Tariq.

For image encryption, Zhou *et al.* proposed CS-based cryptosystems [18], [19] based on chaotic systems. In [20], a multi-class CS-encryption is proposed, which gives different decryption qualities to different class recipients. Zhang *et al.* [21] designed a multi-image encryption, which uses the random convolution [22] for CS-encryption. In [23], Zhu *et al.* proposed a CS-based cryptosystem employing a circular shift for diffusion. Li *et al.* [24] proposed an efficient image encryption for IoT monitoring applications by reducing time consumption in encryption and reconstruction. Zhu *et al.* [25] proposed a CS-based cryptosystem, where permutation and diffusion are executed simultaneously for encryption. There are many other applications of CS for information security [26]–[29].

For processing large size data, the secret matrix of a CS-based cryptosystem may require large data storage and high computational complexity. To resolve this issue, the parallel CS-based cryptosystem was proposed in [30], where the columns of an image are encrypted in parallel with the counter mode operation. Also, a CS-based cryptosystem [31] was proposed to reduce data storage for the secret matrix by employing the semi-tensor product. In [32], Cho *et al.* proposed the *sparse-OTS (S-OTS)* cryptosystem in which a plaintext is encrypted in the OTS manner by a sparse secret matrix with only a few nonzero elements. By using the sparse matrix, the S-OTS cryptosystem can save storage space and computational cost. Therefore, it can be an attractive choice to use the S-OTS cryptosystem in a resource-limited environment, e.g., sensors of IoT.

To evaluate the security of image encryption, we can analyze the encrypted images with statistical measures, such as histogram, entropy, correlation and plaintext sensitivity [21], [24]. Although the S-OTS cryptosystem has efficiency and security, we realize in this paper that it has weak security in terms of the statistical measures. To enhance the statistical security of the S-OTS cryptosystem, a scrambler can be introduced for diffusion mechanism. Scramblers have been widely used in communications [33], [34] and information security [35]. Based on a linear feedback shift register (LFSR), a scrambler randomizes its input bitstream such that the corresponding output bitstream seems to be independently distributed at random. Moreover, a few changes of an input bitstream to a scrambler can lead to significant changes in the output by the recursive structure. Exploiting this property, a scrambler structure was proposed to give the avalanche effect for a secure key distribution to the receivers of wireless local area networks [36].

In this paper, a secure CS-based cryptosystem with strong statistical security is proposed for image encryption. The proposed cryptosystem uses a sparse matrix of the S-OTS cryptosystem and several scrambler chains for diffusion. It is noteworthy that the proposed cryptosystem can theoretically guarantee reliable CS-decryption inherited from the S-OTS cryptosystem, which will be numerically demonstrated. Each scrambler chain consists of a pair of scramblers aligned in series, where the output of the first scrambler is fed as an input to the second scrambler in reverse order. While this scrambler chain is similar to the scrambler structure in [36], we use it at a transmitter to give the avalanche effect for CS-based encryption. To prevent a malicious attacker from restoring the scrambled contents, a keystream generator of a stream cipher is also employed for simultaneous scrambling and encryption. Due to the hardware-friendly structure, the proposed cryptosystem can be implemented easily in real world applications.

For statistical security, we conduct numerical experiments to analyze the histogram, entropy, correlation and plaintext sensitivity for the proposed cryptosystem. We demonstrate that the proposed cryptosystem can achieve flat histogram, high entropy and low correlation of adjacent pixels in ciphertexts, thanks to the diffusion by the proposed scrambler structure. Especially, since the diffusion mechanism of the proposed cryptosystem is executed in ciphertext bit-level, the quantitative evaluation measures of the plaintext sensitivity are much better than those of the other CS-based cryptosystems [23]–[25], [30] that execute the diffusion mechanisms in ciphertext element-level, where each ciphertext element consists of multiple bits. In summary, the proposed CS-based cryptosystem has a hardware-friendly structure and reliable CS-decryption performance. Through numerical results, we demonstrate that the proposed cryptosystem can be statistically secure achieving flat histogram, high entropy, low correlation and high plaintext sensitivity, thanks to the proposed scrambler structure.

The rest of the paper is organized as follows. Some background knowledge for understanding this paper is given in Section II. In Section III, we briefly introduce other CS-based cryptosystems [23]–[25], [30] for comparison. The details of the proposed CS-based cryptosystem, including encryption, scrambling and decryption processes, are described in Section IV. In Section V, experimental results about the statistical security measures of the proposed CS-based cryptosystem are presented and compared to those of the other CS-based cryptosystems [23]–[25], [30]. Finally, concluding remarks will be given in Section VI.

II. BACKGROUND

A. CS-BASED CRYPTOSYSTEMS

Let $\mathbf{x} = \mathbf{\Psi}^T \boldsymbol{\alpha} \in \mathbb{R}^N$ be a plaintext, which is K*-sparse* with respect to an orthonormal basis $\mathbf{\Psi} \in \mathbb{R}^{N\times N}$. Then, the corresponding ciphertext of a CS-based cryptosystem is $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ is a secret matrix with $M \ll N$. A legitimate recipient who knows the secret matrix can decrypt the ciphertext by CS reconstruction techniques, such as an algorithm for l_1-minimization [37] and a greedy algorithm [38].

B. S-OTS CRYPTOSYSTEM

The secret matrix $\mathbf{\Phi}$ of the S-OTS cryptosystem is a sparse matrix with only a few nonzero bipolar entries, which is represented by

$$\mathbf{\Phi} = \frac{1}{\sqrt{Mr}}\mathbf{SP}, \tag{1}$$

where $\mathbf{S} \in \{-1, 0, 1\}^{M\times N}$ is a sparse matrix and $\mathbf{P} \in \mathbb{R}^{N\times N}$ is a column permutation matrix. The row-wise sparsity r is defined by $r = \frac{q}{N}$, where q is the number of the nonzero elements of each row of $\mathbf{S}$. The nonzero elements of $\mathbf{S}$ are generated by the self-shrinking generator (SSG) [39], which is an LFSR-based keystream generator. The first qM bits of an SSG keystream $\mathbf{k} = (k_1, k_2, \cdots)$ are used to generate $\mathbf{S}$. The i-th row of $\mathbf{S}$ has a column index set Λ_i of the nonzero elements defined as

$$\Lambda_i = \{((i-1) \bmod \eta)\cdot q + l \mid l = 1, \cdots, q\}, \tag{2}$$

where $\eta = \frac{N}{q}$. Then, the element in the i-th row and the j-th column of $\mathbf{S}$ is

$$s_{i,j} = \begin{cases} b_{\lfloor \frac{i-1}{\eta} \rfloor \cdot N + j}, & \text{if } j \in \Lambda_i, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $b_i = (-1)^{k_i}$ with $k_i \in \{0, 1\}$ for $i = 1, 2, \cdots$.

The S-OTS cryptosystem can encrypt a plaintext quickly, since the matrix-vector multiplication of the encryption process can be implemented row-wise in parallel. The CS-decryption can also be processed efficiently, exploiting the few nonzero elements of $\mathbf{\Phi}$. In addition, the authors of [32] showed that the S-OTS cryptosystem can be computationally secure through the security analyses against ciphertext only attacks (COA) and chosen plaintext attacks (CPA). Moreover, the S-OTS cryptosystem can theoretically guarantee a stable and reliable CS-decryption for a legitimate recipient.

C. STATISTICAL SECURITY MEASURES

The histogram of an image shows the number of the occurrences of all pixel values. For secure image encryption, the histogram of an encrypted image should have a fairly uniform distribution, which means that the frequencies of the occurrences of all pixel values are almost equal. If the histogram of an encrypted image is non-uniform, it may be vulnerable to statistical attacks. To measure the randomness of an image quantitatively, we can use the entropy. An encrypted image with higher entropy means that its histogram is closer to the uniform distribution. The entropy E of an 8-bit gray-scale image is defined as

$$E = -\sum_{i=0}^{l-1} p(i)\log_2 p(i), \tag{4}$$

where $l = 256$ is the number of the gray levels of the image, i is a pixel value in $[0, 255]$, and $p(i)$ is the frequency of the occurrences of i. Ideally, $p(i) = \frac{1}{256}$ for all i, where E takes the maximum of 8, and the histogram shows the ideal uniform distribution.

In correlation analysis, we measure the correlation of adjacent pixels of an encrypted image for horizontal, vertical and diagonal directions, respectively. We can quantitatively evaluate the correlation of an encrypted image using the correlation coefficient (CC) [21], which is defined as

$$\mathrm{CC} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{5}$$

where x_i and y_i represent randomly selected adjacent pixel values of an encrypted image, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, and n is the number of the selected pixel pairs.

The plaintext sensitivity shows the degree of variation in a ciphertext when we modify its plaintext slightly. Let C_1 be the ciphertext of a plaintext M_1 for a cryptosystem by a key $\mathbf{K}$. Another plaintext M_2, which is different with M_1 by one-bit, can be encrypted to a ciphertext C_2 by the same key $\mathbf{K}$. Then, we can compare C_1 and C_2 by using the unified average changing intensity (UACI) [24] and the number of pixels change rate (NPCR) [24], respectively. The UACI, which represents the average of absolute difference of pixel values for two 8-bit gray-scale images, is defined as

$$\mathrm{UACI}\ (\%) = \frac{1}{WH}\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100, \tag{6}$$

where W and H are width and height of the images, respectively. In (6), $C_1(i,j)$ and $C_2(i,j)$ are the pixel values of C_1 and C_2 at the pixel position (i,j), respectively. The NPCR, which represents how many pixels are different for two 8-bit gray-scale images, is defined as

$$\mathrm{NPCR}\ (\%) = \frac{1}{WH}\sum_{i=1}^{H}\sum_{j=1}^{W} D(i,j) \times 100, \tag{7}$$

where $D(i,j) = 1$ if $C_1(i,j) \neq C_2(i,j)$, and $D(i,j) = 0$ if $C_1(i,j) = C_2(i,j)$. The expected values of the NPCR and the UACI are theoretically calculated by treating every pixel value of two images to be uniformly distributed, which results in 99.6094% and 33.4635%, respectively [40].

D. SCRAMBLERS

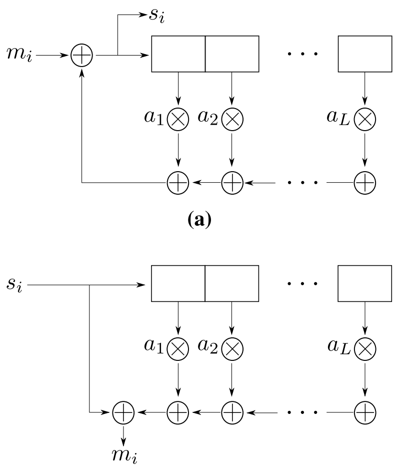
An input bitstream of length H, denoted by $\mathbf{m} = (m_1, m_2, \cdots, m_H)$, can be scrambled by an L-stage LFSR-based scrambler characterized by a polynomial $P(x) = 1 + a_1x^1 + a_2x^2 + \cdots + a_Lx^L$ with $a_i \in \{0, 1\}$ and $a_L = 1$. In Fig. 1(a), the output bitstream is $\mathbf{s} = (s_1, s_2, \cdots, s_H)$, where

$$s_i = m_i \oplus a_1s_{i-1} \oplus a_2s_{i-2} \oplus \cdots \oplus a_Ls_{i-L} \tag{8}$$

for $i = 1, 2, \cdots, H$ and $\oplus$ denotes the bit-wise XOR operation. In this paper, we assume that the initial state of the LFSR is all-zero state. We can restore the original input bitstream $\mathbf{m}$ by using the descrambler in Fig. 1(b), where the descrambling process can be described as

$$m_i = s_i \oplus a_1s_{i-1} \oplus a_2s_{i-2} \oplus \cdots \oplus a_Ls_{i-L}. \tag{9}$$

Note that the scrambling process is recursive, so the current input bit can influence the output bits thereafter. Therefore, we expect that if a bit flip occurs in m_t, a scrambler can make significant changes in s_i's for $i \geq t$.



(a)

(b)

FIGURE 1. LFSR based (a) scrambler, (b) descrambler.

E. GRAIN-128

The Grain-128 [41] is a stream cipher with a 128-bit key and a 96-bit initialization vector (IV). The stream cipher consists of a 128-stage LFSR, a 128-stage nonlinear feedback shift register (NFSR) and an output generating function, as illustrated in Fig. 2. Due to the simple structure, Grain-128 is suited for a resource-limited environment with easy implementation in hardware.

Let $(b_0, b_1, \cdots, b_{127})$ and $(f_0, f_1, \cdots, f_{127})$ be the states of the NFSR and LFSR at a clock, respectively. The feedback Boolean function of the NFSR generating g is defined as

$$g = b_0 \oplus b_{26} \oplus b_{56} \oplus b_{91} \oplus b_{96} \oplus b_3 b_{67} \oplus b_{11} b_{13} \oplus b_{17} b_{18} \oplus b_{27} b_{59} \oplus b_{40} b_{48} \oplus b_{61} b_{65} \oplus b_{68} b_{84}. \tag{10}$$

With $f = f_0$, the recursive relation of the LFSR is $f_0 \oplus f_7 \oplus f_{38} \oplus f_{70} \oplus f_{81} \oplus f_{96}$. The Boolean function generating h is

$$h = b_{12} f_8 \oplus f_{13} f_{20} \oplus b_{95} f_{42} \oplus f_{60} f_{79} \oplus b_{12} b_{95} f_{95}. \tag{11}$$

Finally, the output keystream bit z is defined as

$$z = b_2 \oplus b_{15} \oplus b_{36} \oplus b_{45} \oplus b_{64} \oplus b_{73} \oplus b_{89} \oplus h \oplus f_{93}. \tag{12}$$

Readers are referred to [41] for more details on Grain-128.

III. SOME KNOWN CS-BASED CRYPTOSYSTEMS

For investigating statistical security measures of our proposed CS-based cryptosystem, we introduce other CS-based cryptosystems for comparison in this section. Note that $\overline{\oplus}$ denotes the bit-wise XOR operation between a-bit integers. We define the binary representation of an a-bit integer d as $\vec{d} = (d_1, d_2, \cdots, d_a)$ with $d_i \in \{0, 1\}$ for $i = 1, 2, \cdots, a$, where $d = \sum_{i=1}^{a} (d_i \cdot 2^{i-1})$. Then, if p and q are a-bit integers, $r = p \,\overline{\oplus}\, q$ is an a-bit integer, where $\vec{r} = (r_1, r_2, \cdots, r_a)$ with $r_i = p_i \oplus q_i$ for $i = 1, 2, \cdots, a$.

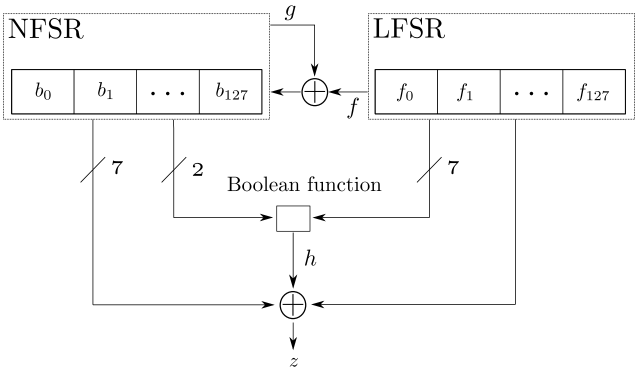


FIGURE 2. Structure of Grain-128.

A. CIRCULAR SHIFT-BASED CRYPTOSYSTEM

In the circular shift-based cryptosystem [23], a sparse representation matrix $\boldsymbol{\alpha} = \boldsymbol{\Psi}\mathbf{X} \in \mathbb{R}^{n\times n}$ is randomly permuted to $\boldsymbol{\alpha}'$, where $\boldsymbol{\Psi} \in \mathbb{R}^{n\times n}$ is an orthonormal basis and $\mathbf{X} \in \mathbb{R}^{n\times n}$ is an image. Then, $\boldsymbol{\alpha}'$ is encrypted to $\mathbf{Y} = \boldsymbol{\Phi}\boldsymbol{\alpha}' \in \mathbb{R}^{m\times n}$, where the secret matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ is generated by using the Chebyshev map [42]. Then, $\mathbf{Y}$ is quantized to $\mathbf{Y}_Q$ by an a-bit quantizer. Each row of $\mathbf{Y}_Q$ can be concatenated into $\mathbf{y}_Q = (y_{Q,1}, y_{Q,2}, \cdots, y_{Q,mn}) \in \mathbb{R}^{mn}$, where $y_{Q,i}$ is an a-bit integer for $i = 1, 2, \cdots, mn$. Each element of $\mathbf{y}_Q$ is modified by

$$e_i = \text{circshift}(y_{Q,i}, k_i^{(1)}) \tag{13}$$

for $i = 1, 2, \cdots, mn$, where circshift(u, v) is a bit-wise circular shift operation to an a-bit integer u, i.e., $w = \text{circshift}(u, v)$ is an a-bit integer, where $w_i = u_{\{(i+v-1) \bmod a\}+1}$ for $i = 1, 2, \cdots, a$. In (13), the keystream $\mathbf{k}^{(1)} = (k_1^{(1)}, k_2^{(1)}, \cdots, k_{mn}^{(1)})$ is generated by using the 4-D hyper-chaotic system [23], where $k_i^{(1)} < a$ is an integer for $i = 1, 2, \cdots, mn$. Finally, an element-level diffusion is applied to each element e_i to yield the ciphertext $\mathbf{c} = (c_1, c_2, \cdots, c_{mn}) \in \mathbb{R}^{mn}$ by

$$c_i = \{(e_i + c_{i-1}) \bmod 2^a\} \,\overline{\oplus}\, \{(k_i^{(3)} + e_i') \bmod 2^a\}, \tag{14}$$

where $e_i' = (\lfloor k_i^{(2)} \cdot c_{i-1} \cdot 10^{14} \rfloor) \bmod 2^a$, $c_0 = 0$, and $i = 1, 2, \cdots, mn$. The keystreams $\mathbf{k}^{(2)} = (k_1^{(2)}, k_2^{(2)}, \cdots, k_{mn}^{(2)})$ and $\mathbf{k}^{(3)} = (k_1^{(3)}, k_2^{(3)}, \cdots, k_{mn}^{(3)})$ of (14) are generated by using the 4-D hyper-chaotic system [23], where $k_i^{(2)}$ is a real number and $k_i^{(3)}$ is an a-bit integer. The details of the keystream generation are described in [23].

B. KRONECKER CS-BASED CRYPTOSYSTEM

The Kronecker CS-based cryptosystem [24] encrypts an image $\mathbf{X} \in \mathbb{R}^{n\times n}$ to $\mathbf{Y} = \boldsymbol{\Phi}\mathbf{X} \in \mathbb{R}^{m\times n}$. In this cryptosystem, the secret matrix $\boldsymbol{\Phi}$ can be constructed by

$$\boldsymbol{\Phi} = \mathbf{A} \otimes \mathbf{P} \in \mathbb{R}^{m\times n}, \tag{15}$$

where $\otimes$ denotes the Kronecker product [43]. In (15), $\mathbf{A} \in \mathbb{R}^{\frac{m}{p}\times\frac{n}{p}}$ is a random chaotic matrix with each element generated from a chaotic system called the Tent map [44].

Also, $\mathbf{P} = \mathbf{P}_\pi \mathbf{D} \in \mathbb{R}^{p \times p}$ is a weighted permutation matrix, where $\mathbf{P}_\pi \in \mathbb{R}^{p \times p}$ is a permutation matrix and $\mathbf{D} \in \mathbb{R}^{p \times p}$ is a diagonal matrix with diagonal elements generated from the Logistic-Tent chaotic system [45].

After CS-encryption, a-bit quantization is performed to $\mathbf{Y}$, which results in a quantized ciphertext $\mathbf{Y}_Q$. Each row of $\mathbf{Y}_Q$ can be concatenated into a vector $\mathbf{y}_Q = (y_{Q,1}, y_{Q,2}, \cdots, y_{Q,mn}) \in \mathbb{R}^{mn}$, where $y_{Q,i}$ is an a-bit integer for $i = 1, 2, \cdots, mn$. The first diffusion is applied to each element of $\mathbf{y}_Q$, which yields $\mathbf{e} = (e_1, e_2, \cdots, e_{mn}) \in \mathbb{R}^{mn}$ with

$$e_i = e_{i-1} \overline{\oplus}\, y_{Q,i} \overline{\oplus}\, k_i^{(1)}, \tag{16}$$

where $i = 1, 2, \cdots, mn$ and $e_0 = 0$. In (16), the keystream $\mathbf{k}^{(1)} = (k_1^{(1)}, k_2^{(1)}, \cdots, k_{mn}^{(1)})$ is generated by using the Logistic-Tent chaotic system [45], where $k_i^{(1)}$ is an a-bit integer. Note that in the element-level diffusion, the l-th bit position of $y_{Q,i}$ cannot influence the t-th bit position of $y_{Q,j}$, where $l \neq t$ for $i, j = 1, 2, \cdots, mn$. Therefore, bit-level variations cannot be diffused within the whole ciphertext, which leads to weak plaintext sensitivity. The second element-level diffusion yields the final ciphertext $\mathbf{c} = (c_1, c_2, \cdots, c_{mn}) \in \mathbb{R}^{mn}$ by

$$c_i = c_{i+1} \overline{\oplus}\, e_i \overline{\oplus}\, k_i^{(2)}, \tag{17}$$

where $i = mn, mn-1, \cdots, 1$ and $c_{mn+1} = 0$. In (17), the keystream $\mathbf{k}^{(2)} = (k_1^{(2)}, k_2^{(2)}, \cdots, k_{mn}^{(2)})$ is generated by using the Logistic-Tent chaotic system [45], where $k_i^{(2)}$ is an a-bit integer. The details of the keystream generation are described in [24].

C. DIVISION-BASED CRYPTOSYSTEM

In the division-based cryptosystem [25], an image $\mathbf{X} \in \mathbb{R}^{n \times n}$ is encrypted to $\mathbf{Y} = \mathbf{\Phi} \boldsymbol{\alpha} \in \mathbb{R}^{m \times n}$, where $\boldsymbol{\alpha} = \mathbf{\Psi} \mathbf{X} \in \mathbb{R}^{n \times n}$ is a sparse representation of $\mathbf{X}$ with an orthonormal basis $\mathbf{\Psi} \in \mathbb{R}^{n \times n}$ and the secret matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ is generated by using the Chebyshev map [42]. Then, $\mathbf{Y}$ is quantized to $\mathbf{Y}_Q$ by an a-bit quantizer. Each row of $\mathbf{Y}_Q$ can be concatenated into a vector $\mathbf{y}_Q = (y_{Q,1}, y_{Q,2}, \cdots, y_{Q,mn}) \in \mathbb{R}^{mn}$, where $y_{Q,i}$ is an a-bit integer for $i = 1, 2, \cdots, mn$. Let $\mathbf{y}_Q = (\mathbf{y}_Q^{(1)}, \mathbf{y}_Q^{(2)})$, where $\mathbf{y}_Q^{(w)} = (y_{Q,1}^{(w)}, y_{Q,2}^{(w)}, \cdots, y_{Q,\frac{mn}{2}}^{(w)}) \in \mathbb{R}^{\frac{mn}{2}}$ for $w = 1, 2$. Finally, elements of $\mathbf{y}_Q^{(1)}$ and $\mathbf{y}_Q^{(2)}$ are modified by an element-level diffusion to yield the ciphertext $\mathbf{c} = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) \in \mathbb{R}^{mn}$, where $\mathbf{c}^{(w)} = (c_1^{(w)}, c_2^{(w)}, \cdots, c_{\frac{mn}{2}}^{(w)}) \in \mathbb{R}^{\frac{mn}{2}}$, by

$$\begin{aligned} c_i^{(1)} &= \{(c_{\pi_i}^{(2)} + k_i^{(1)}) \bmod 2^a\} \\ &\quad \overline{\oplus}\, \{(y_{Q,\pi_i'}^{(1)} + k_i^{(2)}) \bmod 2^a\} \overline{\oplus}\, c_{i-1}^{(1)}, \\ c_i^{(2)} &= \{(y_{Q,\pi_i}^{(1)} + k_i^{(1)}) \bmod 2^a\} \\ &\quad \overline{\oplus}\, \{(y_{Q,\pi_i'}^{(2)} + k_i^{(2)}) \bmod 2^a\} \overline{\oplus}\, c_{i-1}^{(2)} \end{aligned} \tag{18}$$

for $i = 1, 2, \cdots, \frac{mn}{2}$ and $c_0^{(w)} = 0$. In (18), $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_{\frac{mn}{2}})$ and $\boldsymbol{\pi}' = (\pi_1', \pi_2', \cdots, \pi_{\frac{mn}{2}}')$ are random permutations of $(1, 2, \cdots, \frac{mn}{2})$, respectively, and $\mathbf{k}^{(w)} = (k_1^{(w)}, k_2^{(w)}, \cdots, k_{\frac{mn}{2}}^{(w)})$ is generated by using the 6-D chaotic system [25], where $k_i^{(w)}$ is an a-bit integer. The details of the keystream generation are described in [25].

D. PARALLEL CS-BASED CRYPTOSYSTEM

In the parallel CS-based cryptosystem [30], the columns of an image $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n] \in \mathbb{R}^{n \times n}$ are encrypted in parallel. Each column $\mathbf{X}_i$ is encrypted to $\mathbf{Y}_i = \mathbf{\Phi}_i \mathbf{X}_i \in \mathbb{R}^m$ for $i = 1, 2, \cdots, n$, where $\mathbf{\Phi}_i \in \mathbb{R}^{m \times n}$ is a secret matrix with each element generated from the Logistic-Tent chaotic system [45]. Then, $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n] \in \mathbb{R}^{m \times n}$ is quantized to $\mathbf{Y}_Q$ by an a-bit quantizer. Each row of $\mathbf{Y}_Q$ can be concatenated into a vector $\mathbf{y}_Q = (y_{Q,1}, y_{Q,2}, \cdots, y_{Q,mn}) \in \mathbb{R}^{mn}$, where $y_{Q,i}$ is an a-bit integer for $i = 1, 2, \cdots, mn$. The final ciphertext $\mathbf{c} = (c_1, c_2, \cdots, c_{mn}) \in \mathbb{R}^{mn}$ can be obtained after an element-level diffusion, which can be described as

$$c_i = c_{i-1} \overline{\oplus}\, y_{Q,i} \overline{\oplus}\, k_i, \tag{19}$$

where $i = 1, 2, \cdots, mn$ and $c_0 = 0$. In (19), the keystream $\mathbf{k} = (k_1, k_2, \cdots, k_{mn})$ is generated by using the Logistic-Tent chaotic system [45], where k_i is an a-bit integer. The details of the keystream generation are described in [30].

IV. PROPOSED CS-BASED CRYPTOSYSTEM

In this section, we describe the details of the proposed CS-based cryptosystem. For CS-encryption, we use the secret matrix of the S-OTS cryptosystem. Then, the quantized ciphertext of the S-OTS cryptosystem is fed into the proposed scrambler structure, which enhances the statistical security of the proposed cryptosystem. Fig. 3 illustrates the overall structure of the proposed CS-based cryptosystem. In the proposed CS-based cryptosystem, it is noteworthy that the LFSR-based keystream generation by SSG and Grain-128 can be easier to be implemented in real world applications than others, e.g., chaos-based keystream generation [46], [47].

A. CS-ENCRYPTION

The columns of an 8-bit gray-scale $n \times n$ image are stacked into a column vector $\mathbf{x} \in \mathbb{R}^N$, where $N = n^2$. We assume that each 8-bit gray-scale pixel of $\mathbf{x}$ takes a value in $[-128, 127]$ by shifting its original value by -128. With the shared key $\mathbf{K}_1$, one can generate a keystream for constructing the secret matrix $\mathbf{\Phi} = \frac{1}{\sqrt{Mr}} \mathbf{SP} \in \mathbb{R}^{M \times N}$, where $\mathbf{S}$ is a sparse matrix with only q nonzero bipolar elements in each row, and $\mathbf{P}$ is a column permutation matrix. The secret matrix $\mathbf{\Phi}$ encrypts the plaintext $\mathbf{x}$ to $\mathbf{y} = \mathbf{\Phi x} \in \mathbb{R}^M$. Then, $\mathbf{y}$ is quantized by an a-bit quantizer. The quantized CS-encrypted ciphertext $\mathbf{y}_Q \in \mathbb{R}^M$ is

$$\mathbf{y}_Q = \text{round}[\frac{(2^a - 1) \cdot (\mathbf{y} - y_{min} \cdot \mathbf{1})}{y_{max} - y_{min}}], \tag{20}$$

where $y_{max} = \frac{q \cdot 128}{\sqrt{Mr}}$, $y_{min} = \frac{-q \cdot 128}{\sqrt{Mr}}$, $\mathbf{1}$ is an all-one vector of length M, and round($\mathbf{v}$) replaces each element of a vector $\mathbf{v}$ with the nearest integer. As each element of $\mathbf{x}$ is between

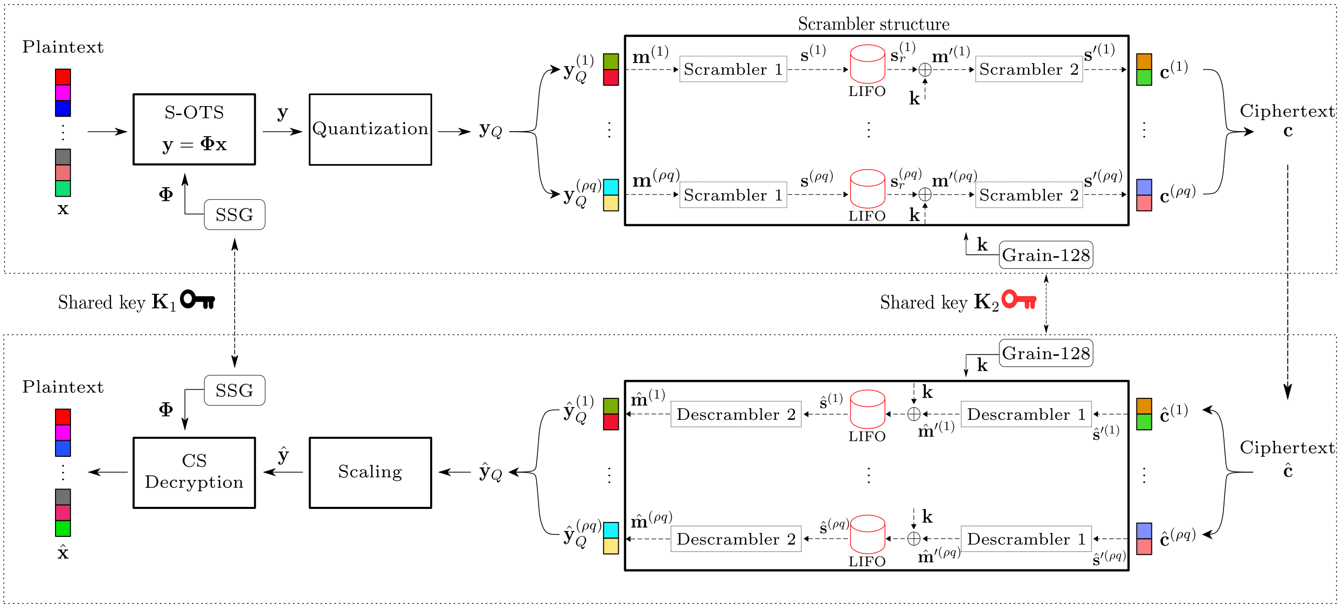


FIGURE 3. Proposed CS-based cryptosystem.

-128 and 127, it is readily checked that the range of the elements of $\mathbf{y}$ is from $\frac{-q\cdot 128}{\sqrt{Mr}}$ to $\frac{+q\cdot 128}{\sqrt{Mr}}$ with its resolution $\frac{1}{\sqrt{Mr}}$, from which the elements of $\mathbf{y}$ have $2\cdot q\cdot 128+1$ possible values. Thus, the number of quantization level should be sufficiently high to represent all possible values of each element of $\mathbf{y}$, i.e., $2^a \geq 2\cdot q\cdot 128+1$, which suggests that the number of quantization bits should be $a \geq \log_2(2\cdot q\cdot 128+1)$.

B. PROPOSED SCRAMBLER STRUCTURE

In (2), the i-th row of $\mathbf{S}$ has the index set $\Lambda_i = \{((i-1) \bmod \eta)\cdot q + l \mid l = 1,\cdots,q\}$ for the nonzero elements, where $\eta = \frac{N}{q}$. Accordingly, it is clear that $\Lambda_{i+\eta} = \Lambda_i$. Assume that $\frac{M}{\eta} = \rho q$ is an integer with the compression ratio $\rho = \frac{M}{N}$. In $\mathbf{y} = \frac{1}{\sqrt{Mr}}\mathbf{SPx}$, if a variation occurs at x_k for $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T$, it would be dispersed to $y_d, y_{d+\eta}, \cdots, y_{d+(\rho q-1)\cdot\eta}$, where the index k is permuted to $k' \in \Lambda_d = \Lambda_{d+\eta} = \cdots = \Lambda_{d+(\rho q-1)\cdot\eta}$ by $\mathbf{Px}$. Then, $\mathbf{y}_Q = (y_{Q,1}, y_{Q,2}, \cdots, y_{Q,M})^T$ can be divided to ρq segments, i.e., $\mathbf{y}_Q = (\mathbf{y}_Q^{(1)}, \mathbf{y}_Q^{(2)}, \cdots, \mathbf{y}_Q^{(\rho q)})^T$, where the w-th segment can be represented as

$$\mathbf{y}_Q^{(w)} = (y_{Q,(w-1)\eta+1}, y_{Q,(w-1)\eta+2}, \cdots, y_{Q,w\eta}) \qquad (21)$$

for $w = 1, 2, \cdots, \rho q$. Then, the variation of x_k will be embedded to each segment for the avalanche effect after scrambling.

In Fig. 3, $\mathbf{m}^{(1)}, \mathbf{m}^{(2)}, \cdots, \mathbf{m}^{(\rho q)}$, which are the corresponding bitstreams of $\mathbf{y}_Q^{(1)}, \mathbf{y}_Q^{(2)}, \cdots, \mathbf{y}_Q^{(\rho q)}$, respectively, are fed into the proposed scrambler structure in parallel. In the scrambler structure, each scrambler chain consists of a series of an L-stage scrambler pair connected by the last-in-first-out (LIFO) buffer of $a\eta$-bits. Through the LIFO buffer, the output bitstream $\mathbf{s}^{(w)} = (s_1^{(w)}, s_2^{(w)}, \cdots, s_H^{(w)})$ of Scrambler 1 with $H = a\eta$ is fed reversely into Scrambler 2, where we encrypt every LIFO output bitstream $\mathbf{s}_r^{(w)} = (s_H^{(w)}, s_{H-1}^{(w)}, \cdots, s_1^{(w)})$ by a common keystream to enhance the security. In other words, $\mathbf{m}'^{(w)} = (m_1'^{(w)}, m_2'^{(w)}, \cdots, m_H'^{(w)})$ is fed to Scrambler 2, where $m_i'^{(w)} = s_{H-i+1}^{(w)} \oplus k_i$ for $i = 1, 2, \cdots, H$ and a keystream $\mathbf{k} = (k_1, k_2, \cdots, k_H)$ with $k_i \in \{0, 1\}$ is generated from the Grain-128 using the shared key $\mathbf{K}_2$. Then, we can get $\mathbf{s}'^{(w)} = (s_1'^{(w)}, s_2'^{(w)}, \cdots, s_H'^{(w)})$ at the output of Scrambler 2, which is the corresponding bitstream of $\mathbf{c}^{(w)} = (c_1^{(w)}, c_2^{(w)}, \cdots, c_\eta^{(w)})$, where $c_i^{(w)}$ is an a-bit integer for $i = 1, 2, \cdots, \eta$. Finally, $\mathbf{c} = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \cdots, \mathbf{c}^{(\rho q)})$ is the ciphertext.

In the proposed scrambler structure, the reverse order feeding by the LIFO buffer can cause the avalanche effect for each scrambler chain. Suppose that a bitstream $\mathbf{m}^{(w)} = (m_1^{(w)}, m_2^{(w)}, \cdots, m_H^{(w)})$ enters Scrambler 1, which yields $\mathbf{s}^{(w)} = (s_1^{(w)}, s_2^{(w)}, \cdots, s_H^{(w)})$. If another bitstream $\mathbf{m}_{\text{err}}^{(w)} = (m_{\text{err},1}^{(w)}, m_{\text{err},2}^{(w)}, \cdots, m_{\text{err},H}^{(w)})$ enters Scrambler 1, where $m_{\text{err},i}^{(w)} = m_i^{(w)} \oplus 1$ if $i = t$, and $m_{\text{err},i}^{(w)} = m_i^{(w)}$ otherwise, it yields $\mathbf{s}_{\text{err}}^{(w)} = (s_{\text{err},1}^{(w)}, s_{\text{err},2}^{(w)}, \cdots, s_{\text{err},H}^{(w)})$. Then, we may assume that $g \approx \frac{H-t}{2}$ bits of $\mathbf{s}_{\text{err}}^{(w)}$ would be different with those of $\mathbf{s}^{(w)}$. This is because the recursion of Scrambler 1 causes almost a half of the last $(H-t)$ bits of $\mathbf{s}^{(w)}$ and $\mathbf{s}_{\text{err}}^{(w)}$ to be different. Thus, we expect that the one-bit difference at the t-th position can cause the avalanche effect by Scrambler 1 only if $t \ll H$. Meanwhile, if $\mathbf{s}^{(w)}$ and $\mathbf{s}_{\text{err}}^{(w)}$ are fed in reverse order through the LIFO buffer, there is a g-bit difference between the first $(H-t)$ bits of the corresponding encrypted LIFO outputs $\mathbf{m}'^{(w)}$ and $\mathbf{m}_{\text{err}}'^{(w)}$, which results in approximately $\frac{t}{2}$-bit difference between the last t bits of $\mathbf{s}'^{(w)}$ and $\mathbf{s}_{\text{err}}'^{(w)}$ at the

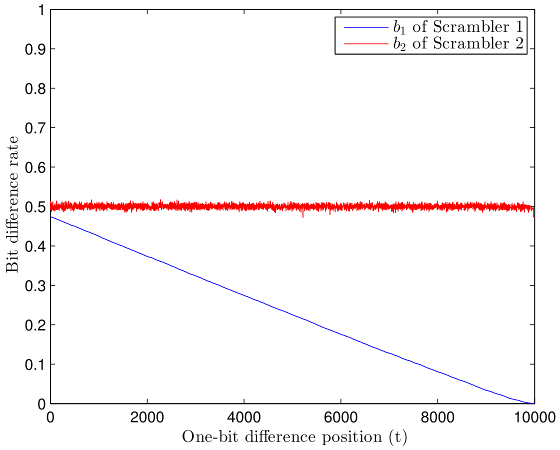


FIGURE 4. Bit difference rate at the outputs of Scrambler 1 and Scrambler 2, respectively, according to the one-bit difference position of input bitstreams to a scrambler chain.

output of Scrambler 2. Finally, two output bitstreams $\mathbf{s}'^{(w)}$ and $\mathbf{s}'^{(w)}_{\text{err}}$ would be almost 50% different regardless of the one-bit difference position t.

When $\mathbf{m}^{(w)}$ and $\mathbf{m}^{(w)}_{\text{err}}$ with $H = 10^4$ enter a scrambler chain, Fig. 4 sketches the bit difference rate at the outputs of Scrambler 1 and Scrambler 2, respectively. We assume that every scrambler is characterized by a polynomial $P(x) = 1 + x^{18} + x^{23}$. We denote d_1 as the number of different bits between $\mathbf{s}^{(w)}$ and $\mathbf{s}^{(w)}_{\text{err}}$ for Scrambler 1 and d_2 as the number of different bits between $\mathbf{s}'^{(w)}$ and $\mathbf{s}'^{(w)}_{\text{err}}$ for Scrambler 2. We can see that the bit difference rate $b_1 = \frac{d_1}{H}$ of Scrambler 1 decreases as t increases, which means that the avalanche effect cannot be guaranteed by a single scrambler. On the other hand, $b_2 = \frac{d_2}{H}$ of Scrambler 2 maintains almost 50% regardless of t, which numerically demonstrates the avalanche effect of the scrambler chain.

C. CS-BASED DECRYPTION

In decryption, the received ciphertext $\hat{\mathbf{c}}$ can be divided to $\hat{\mathbf{c}}^{(1)}, \hat{\mathbf{c}}^{(2)}, \cdots, \hat{\mathbf{c}}^{(\rho q)}$, which yield the corresponding bitstreams $\hat{\mathbf{s}}'^{(1)}, \hat{\mathbf{s}}'^{(2)}, \cdots, \hat{\mathbf{s}}'^{(\rho q)}$, respectively. Each bitstream $\hat{\mathbf{s}}'^{(w)} = (\hat{s}_1'^{(w)}, \hat{s}_2'^{(w)}, \cdots, \hat{s}_{a\eta}'^{(w)})$ is fed into Descrambler 1, where the output is decrypted by a Grain-128 keystream $\mathbf{k}$ generated by using the shared key $\mathbf{K}_2$. With the LIFO buffer, the decrypted output of Descrambler 1 is reversely fed into Descrambler 2 as an input bitstream $\hat{\mathbf{s}}^{(w)}$ to yield $\hat{\mathbf{m}}^{(w)}$, which finally returns $\hat{\mathbf{y}}_Q^{(w)}$. Then, $\hat{\mathbf{y}}_Q = (\hat{\mathbf{y}}_Q^{(1)}, \hat{\mathbf{y}}_Q^{(2)}, \cdots, \hat{\mathbf{y}}_Q^{(\rho q)})^T$ is scaled to the range of $\mathbf{y}$, i.e.,

$$\hat{\mathbf{y}} = \frac{\hat{\mathbf{y}}_Q \cdot (y_{max} - y_{min})}{(2^a - 1)} + y_{min} \cdot \mathbf{1}. \tag{22}$$

We can get the plaintext $\hat{\mathbf{x}}$ after CS-decryption through an l_1-minimization algorithm

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \text{ subject to } \hat{\mathbf{y}} = \boldsymbol{\Phi}\boldsymbol{\Psi}^T\boldsymbol{\alpha}, \tag{23}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{N\times N}$ is an orthonormal basis, such as 2D version of the Daubechies 4 (D4) wavelet basis with $\hat{\mathbf{x}} = \boldsymbol{\Psi}^T\hat{\boldsymbol{\alpha}}$. Finally, the decrypted plaintext $\hat{\mathbf{x}}$ is mapped to the range [0, 255] by adding 128 to each element.

V. EXPERIMENTAL RESULTS

In this section, we demonstrate the statistical security of the proposed CS-based cryptosystem through numerical experiments. The statistical security measures are compared to those of the circular shift-based [23], the Kronecker CS-based [24], the division-based [25], and the parallel CS-based [30] cryptosystems, which are referred to as *circular CS*, *Kronecker CS*, *division CS*, and *parallel CS*, respectively. To demonstrate the statistical security enhancement by the proposed scrambler structure, we encrypt plaintexts only with the S-OTS cryptosystem [32], which is called *S-OTS only*. In contrast, the proposed cryptosystem, called *S-OTS scrambled*, includes the proposed scrambler structure, where Scrambler 1 and Scrambler 2 are characterized by a polynomial $P(x) = 1+x^{18}+x^{23}$, respectively. We use 'Lena', 'Barbara', 'Boat', 'Plane', and 'Peppers' of 256 × 256 test images, which are shown in Fig. 5. In S-OTS encryption, the SSG operates with a 128-stage LFSR, the number of the nonzero elements of each row of the secret matrix is $q = 8$, the compression ratio is $\rho = 0.625$, and the quantization bit size is $a = 12$. For the *Kronecker CS*, we select the parameter $p = 8$ for the secret matrix. With the same quantization bit size and compression ratio, the other experimental settings of *circular*, *Kronecker*, *division*, and *parallel CS*, such as initial values and control parameters for chaotic systems and initial conditions, are selected randomly to meet the constraints presented in [23]–[25], [30], respectively. For all CS-based cryptosystems, each a-bit ciphertext $\mathbf{c}$ is mapped to $\mathbf{e} = \left\lfloor \frac{1}{2^{a-8}}\mathbf{c} \right\rfloor$ in [0, 255] for statistical analysis.

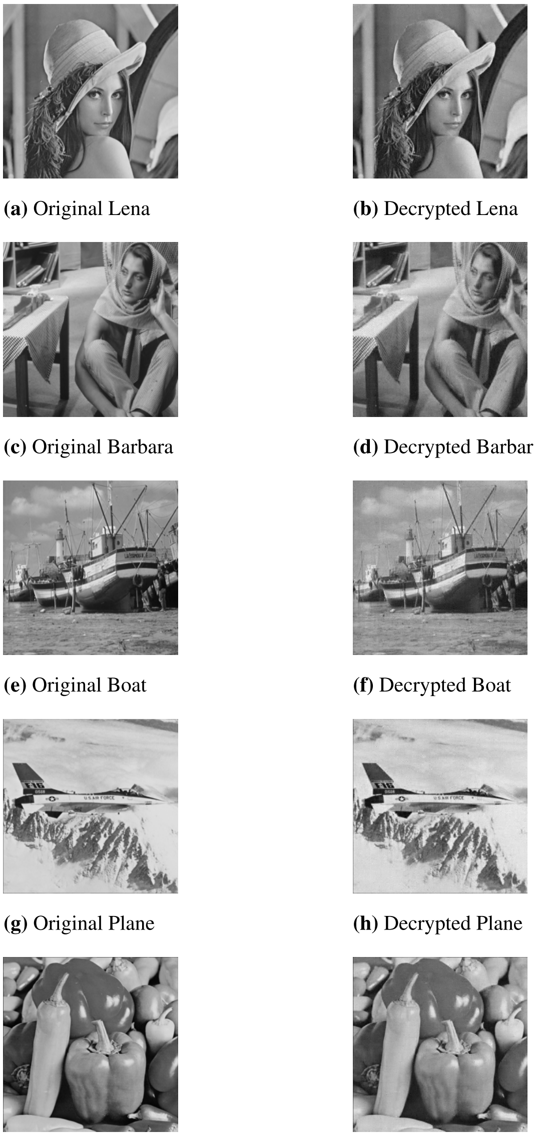
A. CS-DECRYPTION PERFORMANCE

In [32], reliable and stable CS-decryption of the S-OTS cryptosystem have been theoretically guaranteed. Moreover, the phase transition shown in [32] demonstrated that the CS-decryption performance of the S-OTS cryptosystem is similar to that of the B-OTS cryptosystem with random Bernoulli secret matrices. Clearly, the *S-OTS scrambled* inherits the theoretically guaranteed reliable performance of CS-decryption from the S-OTS cryptosystem.

The peak signal-to-reconstruction noise ratio (PSNR), defined by $\text{PSNR} = 10\log_{10}\left(\frac{N\cdot 255^2}{\|\mathbf{x}-\hat{\mathbf{x}}\|^2}\right)$, can be used to evaluate the CS-decryption performance, where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are original and decrypted plaintexts, respectively. For CS-decryption, we employ SPGL1 [48] for the basis pursuit (BP) with D4 wavelet basis. The decryption results of the proposed cryptosystem in noiseless condition are shown in Fig. 5, which demonstrates that the decrypted images are visually acceptable. Table 1 shows that the proposed cryptosystem numerically guarantees higher PSNR for decrypted images than *circular*, *Kronecker*, *division*, and *parallel CS*. Therefore, we conclude that the proposed cryptosystem can achieve more reliable CS-decryption

TABLE 1. PSNR (dB) of decrypted images in noiseless condition.

Image Scheme	'Lena'	'Barbara'	'Boat'	'Plane'	'Peppers'
S-OTS scrambled	34.7884	32.0400	31.9962	33.1949	35.0885
circular CS [23]	32.9776	28.7620	28.3125	29.0521	31.4574
Kronecker CS [24]	30.0425	27.2451	26.2842	27.5739	29.1625
division CS [25]	31.3410	28.2575	27.6412	28.9690	29.7942
parallel CS [30]	31.8795	28.2499	28.0019	28.6933	31.0631



(a) Original Lena **(b)** Decrypted Lena

(c) Original Barbara **(d)** Decrypted Barbara

(e) Original Boat **(f)** Decrypted Boat

(g) Original Plane **(h)** Decrypted Plane

(i) Original Peppers **(j)** Decrypted Peppers

FIGURE 5. Decryption results of the proposed cryptosystem (*S-OTS scrambled*) in noiseless condition, where $N = 65536$, $q = 8$, $\rho = 0.625$, and $a = 12$.

performance than the other CS-based cryptosystems, while guaranteeing theoretically reliable and stable CS-decryption performance.

B. HISTOGRAM

Fig. 6 shows the histograms of original and encrypted images of *S-OTS only* and *S-OTS scrambled*, respectively. A secure cryptosystem should conceal the histograms of original images and have ciphertexts with uniform histograms. We can see that the encrypted images of the *S-OTS only* show non-uniform histograms, which can be vulnerable to statistical attacks. The encrypted images of the *S-OTS scrambled* show uniform histograms, which means that the proposed scrambler structure can successfully flatten the histograms of the ciphertexts of the *S-OTS only*. We also observed that the encrypted images of 'Lena', 'Barbara', 'Boat', 'Plane', and 'Peppers' have uniform histograms for *circular*, *Kronecker*, *division*, and *parallel CS*. Therefore, our proposed CS-based cryptosystem is statistically secure with the uniform histograms of ciphertexts.

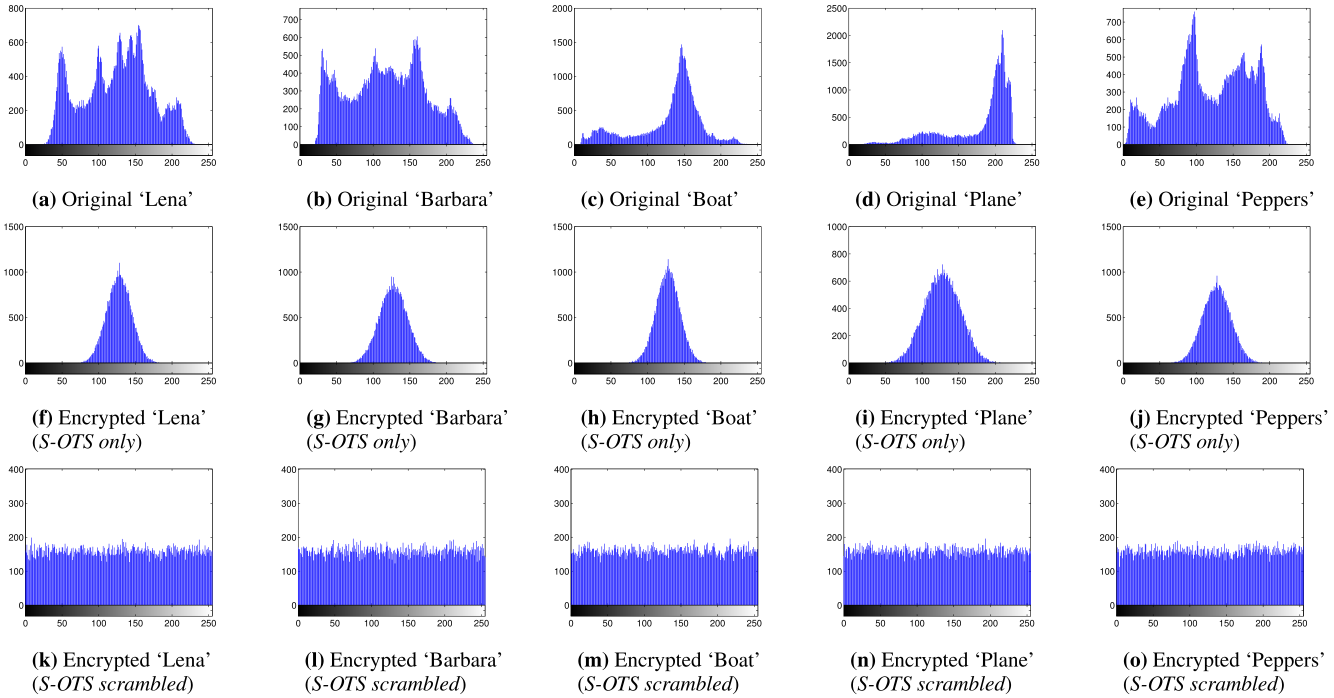
C. ENTROPY

In Table 2, the entropies of encrypted images of each CS-based cryptosystem are shown. We observe that the *S-OTS scrambled* can increase entropies of the encrypted images of the *S-OTS only* by means of scramblers, which demonstrates the statistical security enhancement by the proposed scrambler structure. Similar to *circular*, *Kronecker*, *division*, and *parallel CS*, the *S-OTS scrambled* has entropies close to the ideal value 8. Therefore, the *S-OTS scrambled* is sufficiently secure in terms of entropy.

D. CORRELATION

In correlation analysis, we randomly select $2,000$ adjacent pixel pairs from an encrypted image and calculate the correlation coefficients by using (5) in horizontal, vertical and diagonal directions, respectively. Table 3 shows that the encrypted images of the *S-OTS scrambled* present low correlation coefficients, similar to the other CS-based cryptosystems, which demonstrates the statistical security of the proposed cryptosystem. We observed similar results of low correlation coefficients for the other test images of 'Boat', 'Plane' and 'Peppers' for all the CS-based cryptosystems, respectively.

For analyzing the correlation, we also examine the correlation distribution, which is a scatter diagram plotting randomly selected $2,000$ adjacent pixel pairs of an encrypted image. Fig. 7 displays the correlation distributions of original and encrypted images of 'Lena'. We also observed that the other test images show similar correlation distributions for original and encrypted images. The original image shows a linear distribution in all directions, which means that the adjacent pixel pairs are highly correlated. The encrypted image of the *S-OTS only* has a concentrated distribution.



(a) Original ‘Lena’ **(b)** Original ‘Barbara’ **(c)** Original ‘Boat’ **(d)** Original ‘Plane’ **(e)** Original ‘Peppers’

(f) Encrypted ‘Lena’ (*S-OTS only*) **(g)** Encrypted ‘Barbara’ (*S-OTS only*) **(h)** Encrypted ‘Boat’ (*S-OTS only*) **(i)** Encrypted ‘Plane’ (*S-OTS only*) **(j)** Encrypted ‘Peppers’ (*S-OTS only*)

(k) Encrypted ‘Lena’ (*S-OTS scrambled*) **(l)** Encrypted ‘Barbara’ (*S-OTS scrambled*) **(m)** Encrypted ‘Boat’ (*S-OTS scrambled*) **(n)** Encrypted ‘Plane’ (*S-OTS scrambled*) **(o)** Encrypted ‘Peppers’ (*S-OTS scrambled*)

FIGURE 6. The histograms of original and encrypted images (*S-OTS only* and *S-OTS scrambled*).

TABLE 2. Entropies of encrypted images.

Image Scheme	‘Lena’	‘Barbara’	‘Boat’	‘Plane’	‘Peppers’
S-OTS only [32]	6.1137	6.2853	6.0533	6.6391	6.2931
S-OTS scrambled	7.9955	7.9955	7.9957	7.9958	7.9958
circular CS [23]	7.9961	7.9956	7.9955	7.9959	7.9953
Kronecker CS [24]	7.9955	7.9946	7.9955	7.9956	7.9954
division CS [25]	7.9950	7.9958	7.9956	7.9951	7.9953
parallel CS [30]	7.9954	7.9950	7.9951	7.9954	7.9954

TABLE 3. Correlation coefficients ($\times 10^{-4}$) of encrypted images. (horizontal, vertical, diagonal).

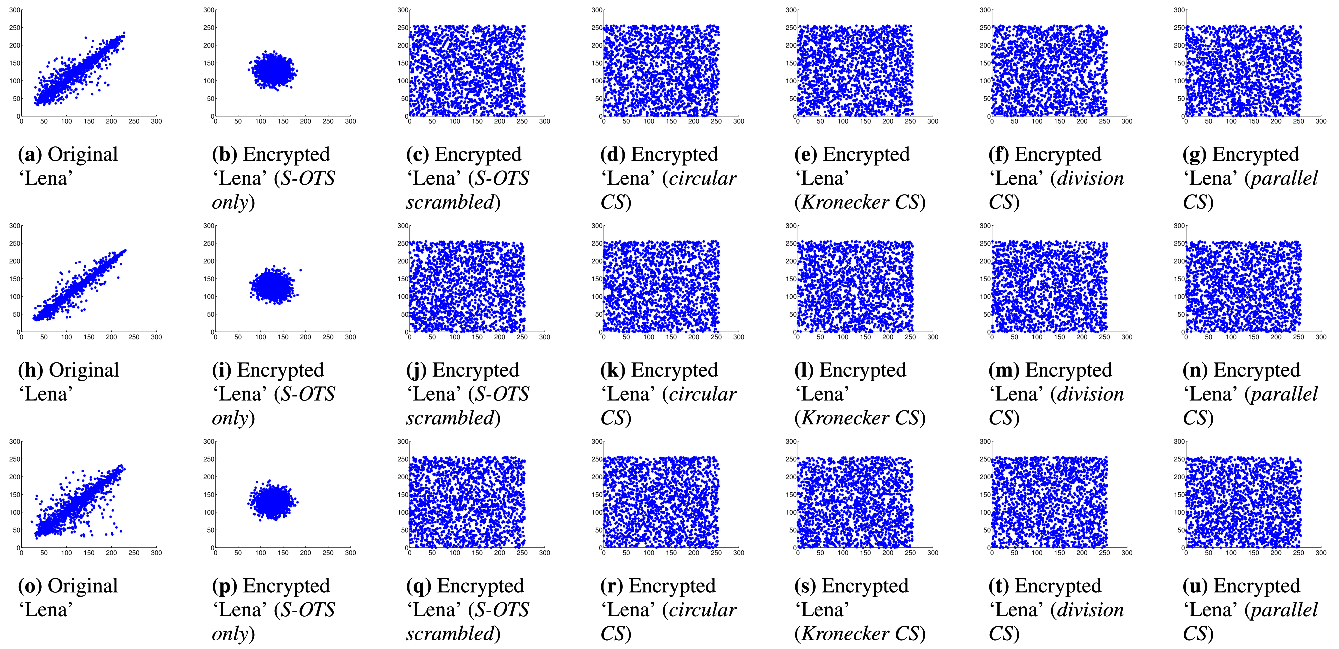
Image Scheme	‘Lena’	‘Barbara’
S-OTS only [32]	$(91, -307, 251)$	$(-130, -279, 215)$
S-OTS scrambled	$(27, 488, 185)$	$(-78, -156, 218)$
circular CS [23]	$(-271, 108, -376)$	$(273, 96, -416)$
Kronecker CS [24]	$(111, -41, -172)$	$(-259, -114, 4)$
division CS [25]	$(-135, 253, 3)$	$(140, 273, -123)$
parallel CS [30]	$(438, 73, -66)$	$(-56, -429, -382)$

On the other hand, the uniform distribution of the *S-OTS scrambled* suggests that adjacent pixel pairs of the encrypted image are uncorrelated to each other. Therefore, we conclude that the proposed scrambler structure can successfully reduce the correlation between adjacent pixel pairs.

E. PLAINTEXT SENSITIVITY

To analyze the plaintext sensitivity, we calculate the NPCR and the UACI for test images. For secure encryption, the NPCR and the UACI should be close to the theoretically expected values, 99.6094% and 33.4635%, respectively.

In numerical experiments, we randomly select a pixel position and flip the least significant bit of the pixel for test images. To evaluate the plaintext sensitivity, we examine the average NPCR and UACI over 100 trials of each cryptosystem. Table 4 shows that *circular*, *Kronecker*, *division* and *parallel CS* with ciphertext element-level diffusion mechanisms show lower average NPCR and UACI than the theoretically expected values, respectively, while the *S-OTS scrambled* has average NPCR and UACI close to the theoretically expected values, respectively. This demonstrates that our proposed CS-based cryptosystem has high plaintext sensitivity. Moreover, the variances of NPCR and UACI for ‘Lena’ are 0.0027 and 0.0272 for the *S-OTS scrambled*, 271.3460 and 30.7554 for the *circular CS*, 1120 and 24.3601 for the *Kronecker CS*, 262.2163 and 1.0785 for the *division CS*, 1179 and 22.3574 for the *parallel CS*, respectively. We observed similar results for the other test images of ‘Barbara’, ‘Boat’, ‘Plane’ and ‘Peppers’. Since *circular*, *Kronecker*, *division*, and *parallel CS* have large variances in NPCR and UACI, their diffusion performance for the plaintext sensitivity is not stable. Meanwhile,



(a) Original ‘Lena’
(b) Encrypted ‘Lena’ (*S-OTS only*)
(c) Encrypted ‘Lena’ (*S-OTS scrambled*)
(d) Encrypted ‘Lena’ (*circular CS*)
(e) Encrypted ‘Lena’ (*Kronecker CS*)
(f) Encrypted ‘Lena’ (*division CS*)
(g) Encrypted ‘Lena’ (*parallel CS*)
(h) Original ‘Lena’
(i) Encrypted ‘Lena’ (*S-OTS only*)
(j) Encrypted ‘Lena’ (*S-OTS scrambled*)
(k) Encrypted ‘Lena’ (*circular CS*)
(l) Encrypted ‘Lena’ (*Kronecker CS*)
(m) Encrypted ‘Lena’ (*division CS*)
(n) Encrypted ‘Lena’ (*parallel CS*)
(o) Original ‘Lena’
(p) Encrypted ‘Lena’ (*S-OTS only*)
(q) Encrypted ‘Lena’ (*S-OTS scrambled*)
(r) Encrypted ‘Lena’ (*circular CS*)
(s) Encrypted ‘Lena’ (*Kronecker CS*)
(t) Encrypted ‘Lena’ (*division CS*)
(u) Encrypted ‘Lena’ (*parallel CS*)

FIGURE 7. **Correlation distributions of original and encrypted ‘Lena’ in the horizontal (first row), vertical (second row), and diagonal (third row) directions.**

TABLE 4. **NPCR (%) and UACI (%) for plaintext sensitivity in CS-based cryptosystems.**

Scheme	Image / Measure	‘Lena’	‘Barbara’	‘Boat’	‘Plane’	‘Peppers’
S-OTS only [32]	NPCR ($\times 10^{-4}$)	19	17	17	16	17
	UACI ($\times 10^{-4}$)	0.0727	0.0670	0.0679	0.0641	0.0670
S-OTS scrambled	NPCR	99.6209	99.6224	99.6100	99.6121	99.6113
	UACI	33.4677	33.4666	33.4660	33.4718	33.4821
circular CS [23]	NPCR	62.6405	50.0881	46.9720	60.5797	63.9393
	UACI	21.0327	16.8183	15.8107	20.3693	21.4801
Kronecker CS [24]	NPCR	35.9286	29.2818	33.2155	28.4033	36.2758
	UACI	1.5565	0.7397	1.9612	1.8439	1.6057
division CS [25]	NPCR	73.9121	74.4104	76.5034	73.2109	77.4027
	UACI	17.1179	17.4513	17.6104	17.7933	17.5589
parallel CS [30]	NPCR	48.8664	48.4993	51.2022	41.3771	48.5385
	UACI	1.3182	1.7278	1.4300	0.9816	1.4016

extremely small variances of the *S-OTS scrambled* indicate that our diffusion mechanism can stably guarantee high plaintext sensitivity. As the *S-OTS only* rarely achieves high plaintext sensitivity, the stable diffusion performance of the *S-OTS scrambled* demonstrates security enhancement in terms of plaintext sensitivity by the proposed scrambler structure.

Fig. 8 shows the differential images of the CS-based cryptosystems for test image ‘Lena’, respectively. In this experiment, the original ‘Lena’ is encrypted to the ciphertext C_1, while the other ciphertext C_2 is obtained from a modified ‘Lena’ with one pixel modification. Then, bit-wise XOR operation between each pixel pair of C_1 and C_2 yields the differential image. The noise pattern of Fig. 8(a) demonstrates that a slight modification of a plaintext results in significant changes in the ciphertext for the *S-OTS scrambled*. Meanwhile, the other CS-based cryptosystems show partially black or almost black differential images due to the high similarity of C_1 and C_2, which implies the weak plaintext sensitivity of *circular*, *Kronecker*, *division*, and *parallel CS*, respectively. Therefore, we conclude that the proposed cryptosystem has high plaintext sensitivity due to the bit-level diffusion by the proposed scrambler structure.

F. KEY SPACE

It is important to have a large key space to resist a brute-force attack for a secure cryptosystem. In the proposed cryptosystem, we use a 128-bit key for the SSG to construct the secret matrix $\mathbf{\Phi}$ and a 128-bit key for Grain-128 for diffusion. This means that the key space of the proposed cryptosystem is $2^{128} \times 2^{128} = 2^{256}$, which implies that the proposed cryptosystem can be secure against a brute-force attack with a sufficiently large key space.

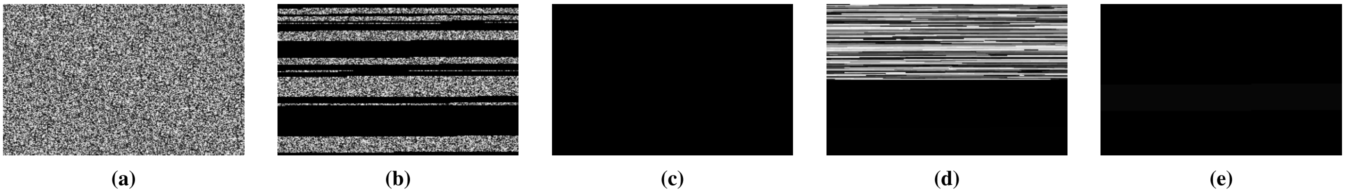


FIGURE 8. Differential images of ciphertexts C_1 and C_2 with size 160 × 256. The ciphertext C_1 is obtained by original 'Lena', while the other ciphertext C_2 is obtained from 'Lena' with one pixel modification. (a) *S-OTS scrambled* (b) *circular CS* (c) *Kronecker CS* (d) *division CS* (e) *parallel CS*.

TABLE 5. NPCR (%) and UACI (%) for key sensitivity in CS-based cryptosystems.

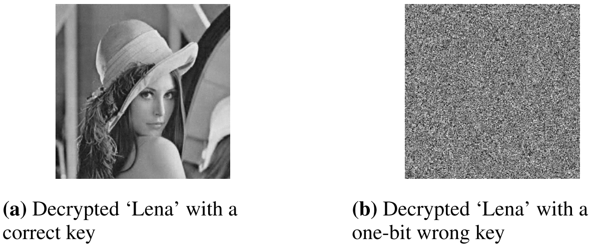
Scheme	Measure \ Image	'Lena'	'Barbara'	'Boat'	'Plane'	'Peppers'
S-OTS scrambled	NPCR	99.6102	99.6049	99.6014	99.6091	99.6124
	UACI	33.4638	33.4527	33.4595	33.4675	33.4640
circular CS [23]	NPCR	99.6094	99.6164	99.6070	99.6113	99.6062
	UACI	33.4651	33.4652	33.4691	33.4593	33.4630
Kronecker CS [24]	NPCR	99.6120	99.6038	99.6104	99.6078	99.6071
	UACI	33.4552	33.4631	33.4689	33.4551	33.4573
division CS [25]	NPCR	99.5990	99.5829	99.6111	99.5861	99.6052
	UACI	33.5379	33.4542	33.4888	33.3945	33.5644
parallel CS [30]	NPCR	99.6089	99.6126	99.6068	99.6209	99.6071
	UACI	33.4534	33.4480	33.4431	33.4796	33.5171

TABLE 6. PSNR (dB) of decrypted images of CS-based cryptosystems with a wrong key.

Scheme \ Image	'Lena'	'Barbara'	'Boat'	'Plane'	'Peppers'
S-OTS scrambled	9.2766	8.8251	9.4505	7.3328	8.6472
circular CS [23]	5.9674	6.0264	5.8058	3.6655	6.0358
Kronecker CS [24]	6.1381	6.1648	6.0751	6.2047	4.6905
division CS [25]	7.0838	7.1050	7.0187	5.1825	7.1631
parallel CS [30]	6.5898	6.4962	6.4569	5.5337	6.3787
benchmark	9.2749	8.9388	9.3759	8.0782	8.9232

G. KEY SENSITIVITY

For a secure cryptosystem, if we modify the key slightly, the ciphertext should be changed significantly. In numerical experiments for key sensitivity, we change either $\mathbf{K}_1$ or $\mathbf{K}_2$ by one-bit randomly for the *S-OTS scrambled*, while one of the chaotic parameters is changed by 10^{-15} randomly for *circular*, *Kronecker*, *division*, and *parallel CS*, respectively. Table 5 shows the average NPCR and UACI over 100 trials of the ciphertexts of all the test images for the CS-based cryptosystems, when we change their keys, respectively. We observe that one-bit modification of the key can result in significant changes in the ciphertext for the *S-OTS scrambled*, where the average NPCR and UACI are close to the theoretically expected values for all test images, respectively. Also, the numerical results of the *S-OTS scrambled* are similar to those of the other CS-based cryptosystems. In addition, we observed that the variances of NPCR and UACI for all the test images are extremely small for all the CS-based cryptosystems, which implies that the proposed cryptosystem has key sensitivity with stable performance.



(a) Decrypted 'Lena' with a correct key

(b) Decrypted 'Lena' with a one-bit wrong key

FIGURE 9. Decrypted 'Lena' images from the *S-OTS scrambled* with a one-bit wrong key.

TABLE 7. The number of operations of CS-based encryptions.

Scheme \ Operation	Addition	Multiplication	XOR
S-OTS scrambled	$\rho n^2(q-1)$	$\rho n^2 q$	$5a\rho n^2$
circular CS [23]	$\rho n^2(n-1)+2\rho n^2$	$\rho n^3+2\rho n^2$	$a\rho n^2$
Kronecker CS [24]	$\rho n^2(\frac{n}{p}-1)$	$\rho\frac{n^3}{p}$	$4a\rho n^2$
division CS [25]	$\rho n^2(n-1)+2\rho n^2$	ρn^3	$2a\rho n^2$
parallel CS [30]	$\rho n^2(n-1)$	ρn^3	$2a\rho n^2$

Fig. 9 shows the decryption results of the *S-OTS scrambled* with a one-bit wrong key. It demonstrates that one cannot visually recover the original 'Lena' after decryption with a one-bit wrong key. Table 6 shows the average PSNR over 100 trials of decrypted images for the CS-based cryptosystems, when we use a wrong key. For a cryptosystem with high key sensitivity, it is reasonable to assume that a decrypted image by a wrong key has the uniform distribution of the pixel values to provide no useful information about the original image. Under this assumption, Table 6 also presents the average PSNR of the decrypted images of uniformly distributed pixel values as a benchmark. Compared to the other CS-based cryptosystems, the PSNR of the decrypted images of the *S-OTS scrambled* is relatively high, but close to the benchmark PSNR, where the decrypted images with a one-bit wrong key are not recognizable, as shown in Fig. 9(b).

H. COMPUTATIONAL COMPLEXITY

For each CS-based cryptosystem, Table 7 shows the numbers of addition, multiplication, and XOR operations for encryption, respectively. Since the CS-decryption performance of the S-OTS cryptosystem is known to be irrelevant to q [32], we can select $q \ll n$ to reduce the computational complexity for the *S-OTS scrambled*, while maintaining reliable CS-decryption performance. Given identical compression ratio ρ and quantization bit size a for all the CS-based cryptosystems, the computational complexity including all operations of the *S-OTS scrambled* is $O(n^2)$. For the *Kronecker CS*, the minimum computational complexity is $O(n^2)$ for $\frac{n}{p} = 2$, since $\frac{n}{p} \geq 2$ is an integer from the secret matrix structure of the *Kronecker CS*. This implies that the *S-OTS scrambled* can have similar computational complexity to the *Kronecker CS*. Also, the computational complexity of the *S-OTS scrambled* is lower than the computational complexity $O(n^3)$ of *circular*, *division*, and *parallel CS*, respectively. Note that the multiplication operations with bipolar entries of the secret matrix of the *S-OTS scrambled* can be implemented as simple operations, i.e., sign change for the numbers multiplied with the negative entries. Meanwhile, the other CS-based cryptosystems require more complex multiplication with floating point numbers. Therefore, we conclude that the proposed cryptosystem has a benefit in terms of computational complexity with a proper selection of q.

VI. CONCLUSION

In this paper, we proposed a secure CS-based cryptosystem for image encryption with hardware-friendly structure, which can be easily implemented in real world applications. In addition to a sparse secret matrix for efficient CS-encryption, a scrambling mechanism has been proposed for enhancing the statistical security of the CS-encryption. We numerically confirmed that the proposed CS-based cryptosystem has more reliable CS-decryption performance than the other CS-based cryptosystems. For analyzing the statistical security, we investigated the histogram, entropy, correlation and plaintext sensitivity of the proposed cryptosystem, which demonstrates that the proposed cryptosystem is sufficiently secure in terms of the statistical measures. In particular, we confirmed that our bit-level diffusion by the proposed scrambler structure has superior performance in plaintext sensitivity than the element-level diffusion mechanisms of the other CS-based cryptosystems. In conclusion, the proposed cryptosystem can be statistically secure thanks to the proposed scrambler structure.

Despite the strong statistical security, the proposed CS-based cryptosystem may have some potential drawbacks. First, the bit-level diffusion by the proposed scrambler structure takes longer time than the element-level diffusion. Also, the proposed scrambler structure requires memory space for the LIFO buffers as well as the scramblers. For a future work, we will further study to reduce the time complexity and the memory requirement for the proposed cryptosystem. We found that the CS-based encryption of [49] employs the hash of a plaintext for keystream generation. Our future work may exploit the hash-based structure to improve our proposed CS-based cryptosystem further.

REFERENCES

[1] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[2] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[3] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, "Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive MIMO," *IEEE Trans. Signal Process.*, vol. 68, pp. 764–779, 2020.

[4] V. K. Amalladinne, J. F. Chamberland, and K. R. Narayanan, "A coded compressed sensing scheme for unsourced multiple access," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6509–6533, Jul. 2020.

[5] J. Chen, S. Sun, L.-B. Zhang, B. Yang, and W. Wang, "Compressed sensing framework for heart sound acquisition in Internet of Medical Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2000–2009, Mar. 2022.

[6] H. Peng, B. Yang, L. Li, and Y. Yang, "Secure and traceable image transmission scheme based on semitensor product compressed sensing in telemedicine system," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2432–2451, Mar. 2020.

[7] L. Wu, P. Sun, Z. Wang, Y. Yang, and Z. Wang, "Toward efficient compressed-sensing-based RFID identification: A sparsity-controlled approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7714–7724, Aug. 2020.

[8] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2177–2186, Nov. 2013.

[9] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "UAV-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1893–1905, Apr. 2019.

[10] F. Wu, K. Yang, and Z. Yang, "Compressed acquisition and denoising recovery of EMGdi signal in WSNs and IoT," *IEEE Trans. Ind. Inf.*, vol. 14, no. 5, pp. 2210–2219, May 2018.

[11] P. Sun, L. Wu, Z. Wang, Y. Feng, and Z. Wang, "SCRA: Structured compressive random access for efficient information collection in IoT," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2356–2367, Mar. 2020.

[12] Y. Rachlin and D. Baron, "The secrecy of compressed sensing measurements," in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2008, pp. 813–817.

[13] A. Orsdemir, H. O. Altun, G. Sharma, and M. F. Bocko, "On the security and robustness of encryption via compressed sensing," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2008, pp. 1–7.

[14] T. Bianchi, V. Bioglio, and E. Magli, "Analysis of one-time random projections for privacy preserving compressed sensing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 313–327, Feb. 2016.

[15] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "On known-plaintext attacks to a compressed sensing-based encryption: A quantitative analysis," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2182–2195, Oct. 2015.

[16] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed. London, U.K.: Chapman & Hall, 2015.

[17] N. Y. Yu, "Indistinguishability and energy sensitivity of Gaussian and Bernoulli compressed encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1722–1735, Jul. 2018.

[18] N. Zhou, A. Zhang, J. Wu, D. Pei, and Y. Yang, "Novel hybrid image compression–encryption algorithm based on compressive sensing," *Optik*, vol. 125, no. 18, pp. 5075–5080, Sep. 2014.

[19] N. Zhou, A. Zhang, F. Zheng, and L. Gong, "Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing," *Opt. Laser Technol.*, vol. 62, pp. 152–160, Oct. 2014.

[20] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-complexity multiclass encryption by compressed sensing," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2183–2195, May 2015.

[21] Y. Zhang, Q. He, Y. Xiang, L. Y. Zhang, B. Liu, J. Chen, and Y. Xie, "Low-cost and confidentiality-preserving data acquisition for Internet of Multimedia Things," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3442–3451, Oct. 2018.

[22] J. Romberg, "Compressive sensing by random convolution," *SIAM J. Imag. Sci.*, vol. 2, no. 4, pp. 1098–1128, Nov. 2009.

[23] S. Zhu and C. Zhu, "A new image compression-encryption scheme based on compressive sensing and cyclic shift," *Multimedia Tools Appl.*, vol. 78, pp. 20855–20875, Aug. 2019.

[24] L. Li, G. Wen, Z. Wang, and Y. Yang, "Efficient and secure image communication system based on compressed sensing for IoT monitoring applications," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 82–95, Jan. 2020.

[25] S. Zhu, C. Zhu, Y. Fu, W. Zhang, and X. Wu, "A secure image encryption scheme with compression-confusion-diffusion structure," *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 31957–31980, Nov. 2020.

[26] M. Yamaç, M. Ahishali, N. Passalis, J. Raitoharju, B. Sankur, and M. Gabbouj, "Multi-level reversible data anonymization via compressive sensing and data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1014–1028, 2021.

[27] Q. Xu, K. Sun, S. He, and C. Zhu, "An effective image encryption algorithm based on compressive sensing and 2D-SLIM," *Opt. Lasers Eng.*, vol. 134, Nov. 2020, Art. no. 106178.

[28] Y. Zhang, P. Wang, H. Huang, Y. Zhu, D. Xiao, and Y. Xiang, "Privacy-assured FogCS: Chaotic compressive sensing for secure industrial big image data processing in fog computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3401–3411, May 2021.

[29] X. Wang and Y. Su, "Image encryption based on compressed sensing and DNA encoding," *Signal Process., Image Commun.*, vol. 95, Jul. 2021, Art. no. 116246.

[30] G. Hu, D. Xiao, Y. Wang, and T. Xiang, "An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 116–127, Apr. 2017.

[31] W. Wen, Y. Hong, Y. Fang, M. Li, and M. Li, "A visually secure image encryption scheme based on semi-tensor product compressed sensing," *Signal Process.*, vol. 173, Aug. 2020, Art. no. 107580.

[32] W. Cho and N. Y. Yu, "Secure and efficient compressed sensing-based encryption with sparse matrices," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1999–2011, 2020.

[33] W. P. Siriwongpairat and K. J. R. Liu, *Ultra-Wideband Communications Systems: Multiband OFDM Approach*. New York, NY, USA: Wiley, 2007.

[34] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 9)*, 3GPP, document TS 36.211, Jun. 2021.

[35] S. A. Parah, J. A. Sheikh, A. M. Hafiz, and G. M. Bhat, "Data hiding in scrambled images: A new double layer security data hiding technique," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 70–82, Jan. 2014.

[36] M. Baldi, M. Bianchi, N. Maturo, and F. Chiaraluce, "A physical layer secured key distribution technique for IEEE 802.11 g wireless networks," *IEEE Wireless Commun. Lett.*, vol. 2, no. 2, pp. 183–186, Apr. 2013.

[37] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Feb. 2001.

[38] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Jan. 2007.

[39] W. Meier and O. Staffelbach, "The self-shrinking generator," in *Advances in Cryptology-EUROCRYPT* (Lecture Notes in Computer Science), vol. 950. Berlin, Germany: Springer, 1995, pp. 205–214.

[40] Y. Luo, L. Cao, S. Qiu, L. Hui, J. Harkin, and J. Liu, "A chaotic map-control-based and the plain image-related cryptosystem," *Nonlinear Dyn.*, vol. 83, no. 4, pp. 2293–2310, Mar. 2016.

[41] M. Hell, T. Johansson, A. Maximov, and W. Meier, "A stream cipher proposal: Grain-128," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 1614–1618.

[42] T. J. Rivlin, *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Hoboken, NJ, USA: Wiley, 1990.

[43] L. Hogben, *Handbook of Linear Algebra*. London, U.K.: Chapman & Hall, 2013.

[44] M. Frunzete, L. Yu, J. P. Barbot, and A. Vlad, "Compressive sensing matrix designed by tent map, for secure data transmission," in *Proc. Signal Process. Algorithms, Archit., Arrangements, Appl.*, Sep. 2011, pp. 1–6.

[45] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, no. 11, pp. 172–182, Apr. 2014.

[46] S. Li, X. Mou, and Y. Cai, "Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography," in *Progress in Cryptology-INDOCRYPT* (Lecture Notes in Computer Science), vol. 2247. Berlin, Germany: Springer, 2001, pp. 316–329.

[47] J. S. Teh, M. Alawida, and Y. C. Sii, "Implementation and practical problems of chaos-based cryptography revisited," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102421.

[48] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 32, no. 2, pp. 890–912, 2008.

[49] Y. Dou and M. Li, "An image encryption algorithm based on compressive sensing and m sequence," *IEEE Access*, vol. 8, pp. 220646–220657, 2020.

JAEPHIL CHOI (Graduate Student Member, IEEE) received the B.S. degree in electronics engineering from Kwangwoon University, Seoul, South Korea, in 2019. He is currently pursuing the M.S. degree with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea.

His research interests include compressed sensing and cryptography.

NAM YUL YU (Senior Member, IEEE) received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1995, the M.S. degree in electronics and electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2000, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2007.

From 2000 to 2003, he was with the Telecommunication Research and Development Center, Samsung Electronics, South Korea, where he worked on channel coding schemes for wireless communication systems. In 2007, he was a Senior Research Engineer at LG Electronics, South Korea, working on the standardization of the 3GPP-LTE. From 2008 to 2014, he was an Assistant/Associate Professor with the Department of Electrical Engineering, Lakehead University, Thunder Bay, ON, Canada. In 2014, he joined the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, where he is currently working as an Associate Professor with the School of Electrical Engineering and Computer Science. His research interests include communications and signal processing techniques for wireless communications.

Dr. Yu has served as an Associate Editor for Sequences in IEEE TRANSACTIONS ON INFORMATION THEORY, from 2009 to 2011.