Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6660709, 12 pages
https://doi.org/10.1155/2021/6660709

Research Article

A High-Quality Authenticatable Visual Secret Sharing Scheme Using SGX

Denghui Zhang and Zhaoquan Gu

Cyberspace Institute of Advanced Technology, Guangzhou University, China

Correspondence should be addressed to Zhaoquan Gu; zqgu@gzhu.edu.cn

Received 12 November 2020; Revised 14 December 2020; Accepted 26 February 2021; Published 17 March 2021

Academic Editor: Zhuojun Duan

Visual cryptography scheme (VCS) is a secret-sharing scheme which encrypts images as shares and can decrypt shares without digital devices. Although a participant can reveal the secret image by merely stacking a sufficient number of shares, the visual quality of recovered images is reduced, and malicious adversaries can cheat participants by giving faked shares. The paper presents a novel VCS called T-VCS (trusted VCS) which consists of two main components: a high-quality VCS and an enhanced verification scheme of shares based on the emerging Intel Software Guard eXtensions (SGX). While providing high-quality recovery, T-VCS keeps the size of the shares the same as the original secret image. We use SGX to act as a trusted third party (TTP) to verify the validity of the shares in an attested enclave without degrading the image quality. The experimental results show that T-VCS can achieve a balance among contrast, share size, and verification efficiency.

1. Introduction

With the development of the Internet of things (IoT), wearable and mobile devices are forming more and more social and big data networks. It is now common to take and transfer personal and sensitive data on an untrusted communication channel [1]. Unfortunately, both diverse social datasets and big data technologies raise stringent privacy concerns. Malicious users can perceive, collect, analyze, and upload large amounts of data; the privacy issues related to the collection of data have been widely concerned and become a research hotspot [2–4]. There is an urgent need to ensure the security of data that include images.

In 1994, Naor and Shamir [5] introduced a VCS which combines the notions of perfect ciphers and secret sharing in cryptography with those of raster graphics. In the Naor and Shamir's k-out-of-n threshold VCS, they split up a secret binary image into n shares (known as sheets or pieces) and distribute them to each participant (known as shareholder). The decryption is impossible unless k or more participants superimpose any k transparencies together. Participants can print out shares onto transparencies and superimpose them to reconstruct the original image. The merit of VCS lies in the fact that the Human Visual System (HVS) can recover the shared secret directly. Thus, the decryption process is computation-free. This feature makes VCS particularly suitable for human-computer interaction scenarios with limited computing or networks, such as ATM [6, 7] and electronic voting [8].

Despite the fact that VCS eliminates complex computation of the traditional cryptography, there remain two significant drawbacks. One is the pixel expansion and contrast degression of the recovered secret, while human eyes can only identify patterns of secret image when the contrast is good enough. Figure 1 illustrates the Naor and Shamir VCS. As shown in the first two columns, if the secret pixel p is white, superposition of the two shares always outputs a gray region where half of the pixels are white and half are black, no matter which column of subpixel pairs are chosen. As shown in the last two columns, if p is black, it yields the original black pixel. There is a contrast loss in this scheme since the original white pixel only yields a gray region. The width of the decoded image is twice that of the original secret image because p is expanded to two subpixels in each share.

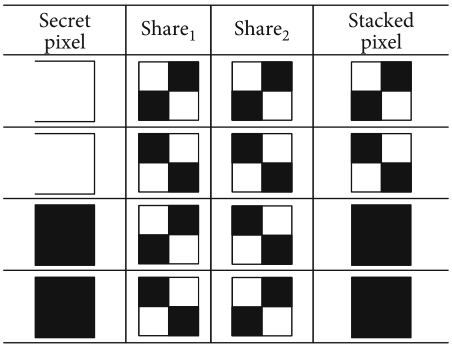


Figure 1: Construction of black and white pixels in a 2-out-of-2 VCS.

Pixel expansion of the shares implies that the size of secret image cannot be too big because a big transparency is inconvenient to align for recovery.

The other one is cheating problems in VCS. If there is a cheater who gives a faked share, sharevictims will fail to decrypt the secret image or believe that the decoded fake image is a genuine secret image. Figure 2 shows an example of cheating participants in the 2-out-of-2 VCS. If both participants are honest, they can recover the true secret image. However, if a malicious participant has information about the Share_1 that the participant A holds, he can construct a faked share FakeShare and cheat A to believe that the secret image is false, which Share_1 + FakeShare reveals.

Sharing secrets with high-quality recovery is interesting, and consequently, many improved VCSs have been proposed. Some schemes present methods where high-quality secret recovery is possible [9]. These schemes, however, rely on complicated computation or archive at the expense of expanding pixels. Other schemes avoid to expand the pixels on secret images at the expanse of reducing image contrast [10].

Researchers have experimented with the idea of cheating VCS and proposed many cheating immune visual cryptography schemes (CIVCS). These schemes often expand the pixel or decline in contrast [11]. Furthermore, analysis results imply that these CIVCS are not enough to secure against cheater colluding [12]. One alternative is to make use of an online trusted authority to verify the validity of the stacked shares. These TTPs often run in untrusted remote environments, while software-based security is often insufficient due to vulnerabilities in applications or operating systems [13, 14].

In this paper, we propose a novel visual secret sharing scheme called T-VCS to address all the drawbacks listed above. This method retains the advantage of traditional visual cryptography without any cryptography computation. T-VCS eliminates pixel expansion by encrypting the pixels of secret images block by block instead of a single pixel. It first constructs a basis matrices' query table which corresponds precisely to the encrypted pixel block and then generates multiple share images by the odd-even quantization watermarking algorithm. The stacked shares have the same pixel permutation with the original image. So, we can recover a higher-quality image.

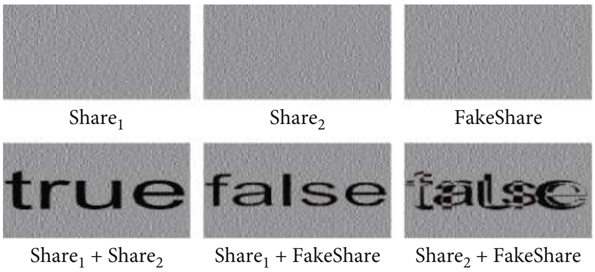

Share_1
Share_2
FakeShare
true
false
false
Share_1 + Share_2
Share_1 + FakeShare
Share_2 + FakeShare

Figure 2: Cheating 2-out-of-2 VCS.

We make use of Intel SGX to ensure the validity of validation results. Intel SGX is a processor instruction set extension that allows the creation of a CPU-based TEE (Trusted Execution Environment), that is, a secure enclave.

We carry out verification in an attested enclave where participants can ensure even OS and higher priority software do not tamper with verification code. T-VCS seals the query table and watermark file into a disk to make watermarks undetectable.

The main contributions of this paper are as follows:

(1) We propose a novel VCS that can both recover secret images with high quality and eliminate pixel expansion

(2) We develop a self-attesting framework to verify the validity of shares based on TEE. This method neither requires the user to maintain additional verification data nor reduces image quality

(3) We elaborately evaluate the performance of applying T-VCS to encode grayscale and color images and verifying sheets in TEE

The remainder of this paper is organized as follows. Section 2 introduces existing VCS and CIVCS and highlight their known limitations. Section 3 explains the proposed T-VCS and shows its effectiveness. After that, Section 4 gives experimental results and comparisons. Finally, Section 5 presents conclusions and future works.

2. Related Work

Born in 1994, VCS has been an emerging research field in the field of information security [15]. To address drawbacks of VCS, many improvements and extensions follow.

2.1. Visual Cryptography Scheme. To keep the size of the sharing images the same as the original secret image, Chen et al. [16] maps a block in a secret image onto a corresponding equal-sized block in the shared image. This scheme, however, may lose some information when the number of pixels describing the information only occupies a tiny part of a secret image. In the scheme of Yang et al. [17], we can distinguish the black and white due to the frequency of white pixels in a white area which is higher than that in a black area. However, this scheme is based on a probabilistic method and the value of contrast is not consistent with the recovered image.

Table 1: Feature comparisons among our proposal and previous schemes.

Schemes	Size invariant	Quality improvement	Cheating immune
Naor and Shamir [20]	×	√	×
Hu and Tzeng [11], Liu et al. [12]	×	×	√
Chen et al. [21] T-VCS	√	√	√

Hou and Tu [18] takes advantage of image contrast reduction and halftone technology to avoid expanding pixels in secret images. This method degrades the contrast of the resulting image by 50%. Superimposing shares is equivalent to an OR-logical operation on the corresponding rows. New visual cryptography models which utilize the polarisation of the light and XOR operation can keep image size [19], but this method does not work with printed transparencies since it is the idea behind VC.

To obtain a better contrast than the previous one, Naor and Shamir propose an alternative model for reconstruction [20]. This model improves the contrast from $1/2$ to $1 - 1/c$, where c is both the number of sheets and the number of subpixels to map each pixel. This scheme expanded each pixel in the original image into c subpixels. The set of operations they call the cover semigroup inspires the design of T-VCS, while T-VCS can archive the same contrast without pixel expansion.

Most of the previous research work on black and white VCS, while it is an essential area of research to apply visual cryptography techniques to color images. This method allows the use of natural color images to secure information. The scheme of Hou [22] is one of the first color decomposition techniques to generate visual cryptograms for color images. Every color within the image can be decomposed into one of three primary colors. In contrast to color decomposition, Yang [17] proposes an additive color mixing scheme based on probabilities. One of the problems with these schemes is that the overall contrast is reduced when revealing secret images.

2.2. Cheating Immune Visual Cryptography Schemes. VCS assumes shareholders to be semihonest, and the image shown on the stacking of shares is a real secret image. Researchers, however, have present methods for cheating the basic VCS schemes [21] and consequently proposed many CIVCS by generating extra verification shares or expanding the pixel to embed extra authentication information. Hu and Tzeng [11] propose a generic method to convert a VCS to another VCS that has the property of cheating prevention. The overhead of the conversion is near-optimal in both contrast degression and pixel expansion. Liu et al. [12] first analyze the drawbacks of some known cheating immune visual cryptography schemes, then proposed a new CIVCS, which avoids all the previous drawbacks. However, this scheme has to randomly select t pixels from the original secret image to act as authentication pixels in each participant, which inevitably increases the burden of share management.

An alternative to prevent cheating in VCS is to use a TTP to validate shares. Yang and Lai proposed [23] used a TTP to perform the verification between the participants. Chen et al. [21] find that if an attacker knows the exact distribution of black and white pixels of each share, then they can attack and cheat the TPP. Table 1 presents the comparison of existing methods and our scheme.

2.3. Intel SGX. A practice challenge of TTP is that they often run in an untrusted cloud environment [24], so the participant is not sure if a validation result is valid [25]. TEEs such as Intel SGX and ARM TrustZone [26] enable execution of programs in secure enclaves.

As an important research progress in the field of trusted computing, Intel SGX offers an efficient solution for anonymous authentication and verification. Besides shielding systems [27–29], SGX has been used in a number of applications including a map-reduce framework [30], machine learning and big data models [31, 32], and SQL querying [33]. The security of these client-server applications is based on the establishment of trust in a remotely executing program. The procedure requires an enclave to generate a hash of the code running inside it, which is signed by an Intel-provided service enclave running on the same platform. Participants are then able to verify the report through the Intel Attestation Service (IAS) provided by Intel. More details about SGX can be found in [34, 35].

3. T-VCS

In this section, we propose a novel VCS to address the contrast degression and pixel expansion problems at the same time. The contrast determines the clarity of the recovered secret by HVS. Having shares that are close to the original secret's size is easier to manage and transmit.

There are three modes of images: black and white, gray, and color. We treat a color image as the composition of three primary color images. Then, the halftone technology can transform a single-color image with gray levels into a binary image. So the research of black and white image is fundamental even in the study of color images.

We firstly deal with the black and white image, where a white pixel is denoted by the number 0 and a black pixel by 1. To construct shares of an image for participants, we need to prepare two groups, C_0 and C_1, which consist of bit matrices. A row in matrix C_0 and C_1 corresponds to m subpixels of a pixel. For a white (or black) pixel in the image, we randomly choose a matrix from C_0 or C_1 and assign the row of groups to the corresponding position of share S_i, $i < n$. Each pixel of the original image will be encoded into n subpixels, each of which consists of m subpixels. Instead of constructing C_0 and C_1 directly, we can construct two basis matrices S^0 and S^1 and let C_0 and C_1 be the set of all matrices obtained by permuting columns of S^0 and S^1, respectively. So, we can write the basis matrices and collections of the 2-out-of-2 VCS as

Equation (2). By merging the sheets of participant A and participant B, that is, putting the ith sheet of B on top of the i sheet of A, we can reconstruct the secret images.

$$S^0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad S^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{1}$$

$$C^0 = \left\{ \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \mid \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \right\}, \quad C^1 = \left\{ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mid \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\}. \tag{2}$$

3.1. Generate Pixel Blocks. The drawback of the traditional VCS is that the shares are $n:m$ times the size of the secret image. So, we adopt block-wise operation and generate shares block by block for the nonexpansion scheme. A block in a share image corresponds to an equal-sized block in the secret image, while in the traditional expansionary VCS, a pixel is mapped onto a subpixel.

The cover semigroup operation inspires the design of T-VCS; Table 2 demonstrates the cover color rule of the monochromatic construction, where the top opaque wins. If we superimpose three pixels, that is, transparency, black, and white, respectively, we will get a white pixel at the top.

Figure 3(a) demonstrates the revealing procedure of the pixel block in which $c = 4$ and the revealed pixels are [B; W; W; W]. The stacked pixels in Figure 3(a) represent neither the original single black nor white pixel, but the pixel block [B; W; W; W]. We number sheets from bottom to top, starting with zero. The participant B hold sheets numbered even, that is, {0, 2, 4, 6}th sheet. When stacking the four sheets, B only obtains an all-white image. The participant A holds sheets numbered odd, that is, {1, 3, 5, 7}, respectively. When stacking the four sheets, A only sees an all-black image.

Figure 4 illustrates sheets. There are two opaque colors (black and white) and a completely transparent one in the new model. It is obvious that the proposed share images do not leak any secret information from the share images. It should be noted that the white color in Figure 4 indicates transparency. To display white pixels on the white background, we replace the white sheets held by B with gray. The reconstruction is done by merging the sheets of A and B. When stacking them all together, we can obtain a clear secret image, as shown in Figure 5(d).

A holds c odd sheets, while B holds c even sheets. By rolling sheets held by B down a row (for the whole sheets, it is equivalent to roll down two rows), the position of the ith sheet will be $(i + 1)\%c$th. We are able to obtain the pixel block [B; B; W; W] as shown in Figure 3(b). Naor and Shamir have proved the contrast is $1 - 1/c$ in the construction method [20]. By such a method, T-VCS can generate the basis matrices for kinds of pixel blocks.

Table 2: Color rule in shares and stacked images.

Color	White (W)	Black (B)	Transparent (T)
W	W	B	B
B	W	B	B
T	W	B	T

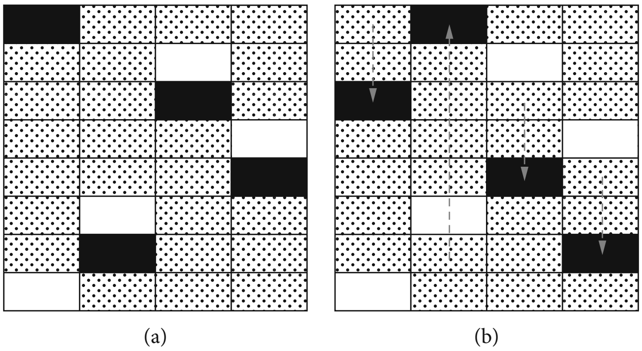
T-VCS gives a certain way to construct the basis matrices. There are $4! = 24$ permutations for the case of 4 black pixels (the number of white pixels is 0). For the case of 3 black pixels (the number of white pixels is 1) or 1 black pixel (the number of white pixels is 3), taking 1 black pixel as an example, the black and white pixel must locate in the top (7) and bottom (2) layers; otherwise, there will be a white pixel in the bottom (2) layer, and this column will be covered by black pixels, resulting in the number of revealed black pixel pixels which is greater than 1. The number of permutations for the other three columns of white pixels are $3! = 6$.

For the case of 2 black pixels (the number of white pixels is 2), taking the 2 black pixels ([B; B; W; W]) as an example, the black pixels in columns 1 and 2 can only be selected from layers 5 and 7. Respectively, the white pixels in columns 1 and 2 can be selected in layers 2 and 4, to ensure that the superimposed pixels in columns 1 and 2 are black. However, if the white and black pixels in columns 1 and 2 are located in layers 1 and 3, after rolling twice, the superimposed white pixels will appear in columns 1 and 2, which is inconsistent with the precondition. So, there are two ways to arrange the white pixels in columns 1 and 2. The white pixels in columns 3 and 4 are similar, and there also are two ways to arrange them. So, the total arrangement is $2 + 2 = 4$. There are still other ways to construct the basis matrices besides the rotation method. So, what is given here is not the total number of permutations. The relationship between rows to roll and generated pixel blocks as shown in Table 3.

Each pixel block has multiple permutations. The scheme not only ensures the security of image encryption but also provides conditions for the subsequent zero watermark verification.

It should be noted that the roll period of the pixel block is 4 for the case $c = 4$, so when rolling down 4 rows, repeated pixel block will be generated. So, this scheme cannot generate all-black pixel block or all-white pixel block (in Table 3, the pixel block [W; W; W; W] cannot be generated). We replace the basis matrices of all-white blocks with those of 3 black and 1 white situation.

Our new scheme contains several important changes from previous work. The first difference is the order in which the transparencies are stacked. There is a requirement for order to correctly recover secret images. Therefore, we need to record the order of each share. The second change is that each participant has c sheets, rather than a single transparency. Each pixel in the original image is mapped into c subpixels. In order to facilitate the user to manage the sheets, we can use color images to store sheets of grayscale images and use the TIFF multilayer image format to store sheets of color images.



(a) (b)

Figure 3: Stacking of gray-level visual cryptography without pixel expansion (the dot block represents transparency).

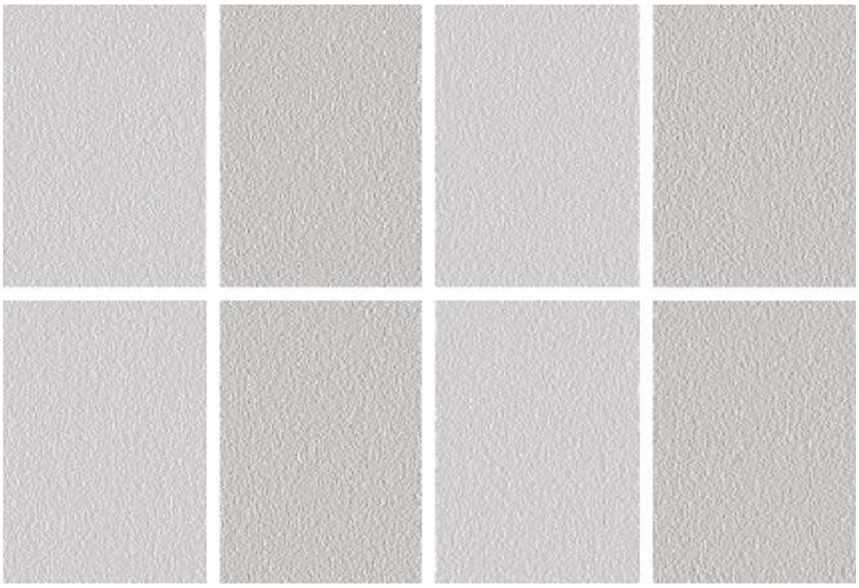
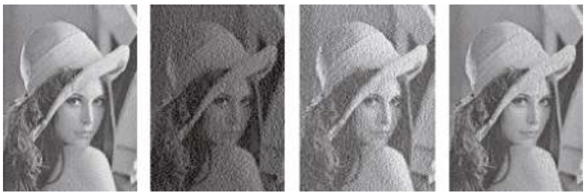


Figure 4: Encrypted sheets ($c = 4$; contrast $= 1 - 1 = c$).



Mode	Halftone	Hou [18]	Random T-VCS	T-VCS
PSNR	100.00%	51.57%	51.62%	68.06%
	(a)	(b)	(c)	(d)

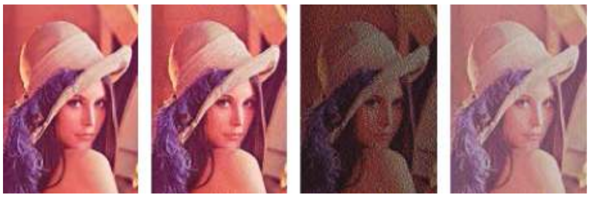
Figure 5: Contrast comparison between Hou's VCS and T-VCS.

Table 3: Generate pixel blocks by rolling.

Rows to roll	Pixel block	Number of permutations
0	[B, B, B, B]	24
1	[B, B, B, W]	6
2	[W, B, B, W]	4
3	[W, W, W, B]	6
4	[W, W, W, W]	24

3.2. Generate the Permutation of a Pixel Block. In the halftone image, there are not only different pixel blocks but also different arrangements of the same pixel block. By permuting columns in Table 3 until getting the same arrangement as the original color blocks, T-VCS precomputes C_0 and C_1 for kinds of pixel blocks and permutations.

An objective way to test alteration between the original image and the recovered image is to use PSNR (Peak



Mode	Original	Halftone	CMYK	T-VCS
PSNR	100.00%	43.29%	27.83%	32.95%
	(a)	(b)	(c)	(d)

Figure 6: Contrast comparison between color VCSs.

Signal-to-Noise Ratio) given in Equation (3) to measure the difference, where $x(i, j)$ and $y(i, j)$ are corresponding pixels in the ith row and jth column in the secret image x and the recovered image y. When the PSNR value is greater than 30, it means that the transparency of the watermark [36] is better. The larger the PSNR value, the better the clarity.

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{\mathrm{MAX}_I^2 \times m \times n}{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left(x(i, j) - y(i, j)\right)^2}\right). \tag{3}$$

Figure 5 shows a contrast comparison between kinds of halftone VCS. We select the halftone image of Lena (Figure 5(a)) as the reference image. As shown in Figure 5(b), since Hou's scheme takes advantage of image contrast reduction, the recovered image is darker, and its PSNR is the lowest. By limiting selectable matrices to the same space as the original block of pixels, we archive to increase the PSNR from 51.62% to 68.06%. It can be inferred that T-VCS has a higher quality of secret recovery than directly applying the block-wise operation, which generates pixel blocks with a random permutation. Most of VCSs have the property of perfect black. The reconstructed image in T-VCS also is perfect black since stacked blocks associated with black pixels of the secret image are all black.

3.3. Color T-VCS. In this section, we propose a nonexpansion VCS for color images based on the above gray-level scheme [37]. We first divide a color image into three color channels: cyan (C), magenta (M), and yellow (Y), since most color printers use C, M, and Y inks to display color. This scheme represents the gray levels of each color channel of the secret image by vectors of 8 bits; that is, the secret image is divided into 8-bit levels and each bit level forms a binary image. For each bit level j and each color channel h, we choose a block in the secret image and encrypt it by the query table. For a color image with a bit depth of 8, it is equivalent to repeating the black and white image encryption operation 24 times.

The method has the characteristic of gradual restoration. The more the superimposed share image, the higher the clarity of the restored image. So participants do not need to generate all the shares for all the bit levels. We can recover a clear enough image by selecting the highest number of bits, since the information about a higher bit level is not as important as that of a lower bit level for HVS.

Figure 6 shows the experimental results of the color T-VCS. We calculate the PSNR of color images by summing and averaging each channel's PSNR where the maximum pixel value MAX is 255. Figure 6(a) is the original color image. Due to the fact that digital halftoning is a lossy process in itself, it is impossible to reconstruct the original secret image fully, so the PSNR reduces to 43.29% in Figure 6(b). The PSNR of image recovered from the T-VCS is 32.95% in Figure 6(d), while the PSNR of image recovered from random T-VCS is 27.83% shown in Figure 6(c). It can be inferred that a careful arrangement of pixels in encryption improves the quality of color images.

3.4. Verify Sheets by SGX. Despite visual cryptography's secure nature, it would be terrible if participants cannot verify distributed shares. Not only can we use VCS to encrypt images, but we can also verify sheets. But the disadvantage of VCS is that once malicious attackers tamper with authentication information, human vision cannot detect it. We introduce TEE to prevent malicious systems from tampering with the verification data on untrusted remote environments.

The verification process is shown in Figure 7, which briefly includes the distribution phase and verification phase. During the distribution phase, T-VCS first constructs a basis matrices' query table according to the method above and then generates sheet images according to the watermark image [38].

(A.1) Generate basis matrices. Based on the value of each pixel in the watermark image, the basis matrices are selected randomly from odd parts if the pixel pw in watermark image (W) is white or even parts if pw is black, respectively. We can generate the watermark image dynamically in the enclave or select from existing images. We then seal the query table to the disk for subsequent verification. The watermark image is the only sensitive data in the T-VCS. Sealing enables encrypting and authenticating the enclave's data such that no process other than the exact enclave can decrypt or modify it. The security of the watermark can also be guaranteed since no secret information will leak out in the enclave.

(A.2) Generate sheets and serialize them to disk through the *OCALL* instruction of the Intel protected file

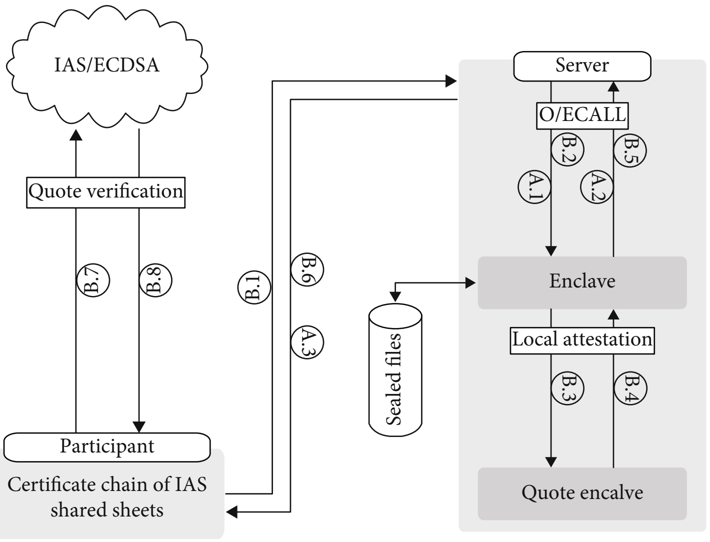


Figure 7: Sheet attestation workflow.

system. Isolated execution of an enclave process restricts access to a subset of memory such that only that particular enclave can access it. Entry points defined in trusted code are called *ECALL*. Valid interaction with an enclave is only possible via explicit *ECALL*. Similarly, entry points defined in untrusted code are called *OCALL*, which calls out of the enclave.

(A.3) Distribute sheet images to each participant.

During the verification phase, T-VCS acts as a TPP to provide an integrity validation service for participants by hardware attestation. The phase includes the following steps:

(B.1) The participant *P* sends a held sheet and received sheet images to the T-VCS server for verification.

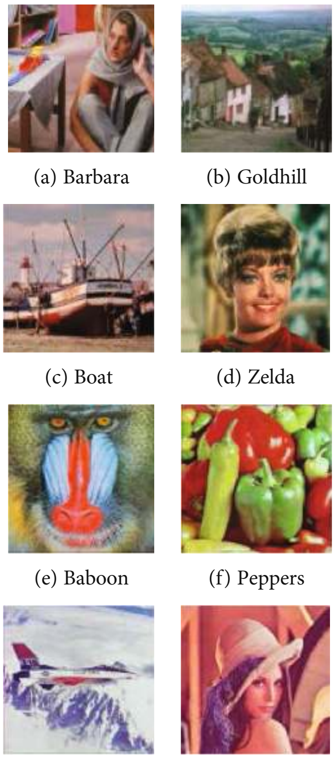
(B.2) The server transfers the two images which may contain multiple channels to the verification enclave (*E*). *E* first reads the pixel value p_s of the watermark image which is involved with received sheets and then determines whether p_s is consistent with parts from which the basis matrices are selected. This procedure is repeated until traversing all the pixel values. If all of the pixels match, that is, $p_{wi} == p_{si}$, $\forall \{i\} \in \{W\}$, the verification is passed, otherwise failed.

(B.3) *E* requests the quote enclave (*Q*) to sign the MRr (MRENCLAVE) and verification result and generate the quote report (*R*). *R* consists of and attached report data R_a and enclave data R_e. *Q* first performs local attestation with *E*. After passing attestation, *Q* signs the hash value of the authentication result (*a*), the requested sheets S_{ri} and write *R* into the report data field, that is, $R = \text{hash}(a \| S_{ri} \cdots \| Srn)$; , where *n* is the number of sheets. We patch the *tkey_exchange* library in SGX SDK to generate a custom report data field. It is crucial to include a hash of verification information into the report to avoid masquerading attacks, as this binds validation results to this enclave. R_e contains the enclave measurement/identity (MRENCLAVE), and the signing identity (MRSIGNER) verifies that the enclave contained the expected code/data pages at launch. Although the length of report data is limited in SGX, the quote report can still ensure the authenticity of sheet images and verification results by the hash value *R*.

(B.4) The signature information of *R* can only be verified by the IAS, so the participant needs to send the quote report to IAS for authentication. Remote attestation is digital signatures produced by the SGX over the code of enclaves. *P* (challenger) can verify it using the manufacturer's public key [39] to ensure that an enclave has been deployed correctly and is running on a trustworthy Intel SGX hardware platform.

(B.5-8) After receiving the verification report returned by IAS, *P* first verifies the report by signature chains and then compares the MR_r with local stored MR_l, the image hash of *Sri* in the report with the local cache S_{li}. After verification is passed, the participant can confirm the validation of sheet images.

In the validation procedure, participants and the T-VCS server do not need to establish a trusted channel, which makes the whole process much more secure and user-friendly. Through the Intel remote attestation service, participants can be sure that the specified enclave completes the image sharing and verification operation. Ensuring the integrity of operation is necessary because a malicious OS could drop messages and tamper with data and validation process. We call this CIVCS self-attesting because the enclave attests itself before verifying integrity of sheets. The only situation



(a) Barbara (b) Goldhill

(c) Boat (d) Zelda

(e) Baboon (f) Peppers

(g) Airplane (h) Lena

Figure 8: Color test images.

Table 4: Performance comparison of gray images between Hou and Tu's algorithm and T-VCS.

Images	PSNR (Hou and Tu [18])	PSNR (random T-VCS)	PSNR (T-VCS)
Barbara	51.50%	51.86%	67.86%
Goldhill	51.35%	51.74%	67.17%
Boat	51.63%	51.49%	70.31%
Zelda	52.02%	54.69%	61.70%
Baboon	51.93%	52.06%	63.54%
Peppers	52.12%	51.85%	64.08%
Airplane	53.19%	52.95%	60.45%
Lena	53.37%	53.14%	59.64%

the system ends without being suspended is it runs on a trusted, nondisruptive environment.

4. Evaluation

In this section, we select eight commonly used color images which are shown in Figures 8(a)–8(h), to demonstrate the performance of T-VCS. The size of all the secrets is 512 × 512. All the data shown below are the average of test results for 100 runs on test images.

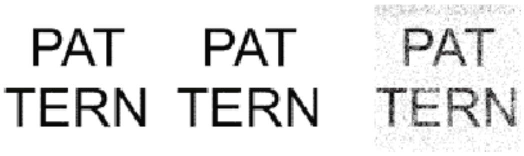
4.1. VCS Evaluation. To illustrate the encryption and recovery results of the T-VCS on grayscale images, we first convert Figures 8(a)–8(h) into grayscale images and then use halftone technology to binarize images. We use the halftone technology to simulate gray levels by altering the density of the printed dots. In the bright parts of an image, the density is sparse, while in the darker parts of the image, the density is dense. There are many halftoning techniques available, where error diffusion produces superior results and is adopted in this paper.

Table 5: Performance comparison of color images among Hou, Liu et al., and T-VCS.

Images	PSNR (Hou [22])	PSNR (Liu et al. [40])	PSNR (T-VCS)
Barbara	27.78%	27.78%	43.49%
Goldhill	27.82%	27.82%	43.20%
Boat	27.90%	27.91%	41.89%
Zelda	27.83%	27.82%	42.90%
Baboon	27.81%	27.81%	44.76%
Peppers	27.86%	27.85%	44.23%
Airplane	27.44%	27.43%	40.92%
Lena	27.83%	27.84%	42.48%

Table 4 shows a performance comparison of gray images between Hou and Tu's algorithm and T-VCS. Hou and Tu's experimental results [18] are shown in the second column of Table 4. They adjust all of the gray values in the grayscale image to more than 127 by linearly interpolating. After halftone conversion, the number of black subpixels in each block is between 2 and 4. They can obtain the same arrangement by transforming the distribution of black and white subpixels. We, however, find that Hou and Tu's method cannot convert all blocks to the requirements of 2 to 4 black blocks. For the Barbara image, there are 41 abnormal blocks with 3 white and 1 black pixels, which accounts for 0.062% of all the blocks. The PSNR for images recovered from this method is between 51.35% and 53.37%, which is similar to the results of random T-VCS, while the values of PSNR for images recovered from T-VCS are all greater than 59%. Although both schemes can keep the image size, it is obvious that our method can recover images with higher quality.

The PSNRs of recovered color image are shown in Table 5. Using this color decomposition, Hou [22] decomposes every color within the image into three primary colors. This proposal is similar to traditional visual cryptography for the pixel expansion that occurs. The loss of contrast will accumulate because color images use subchannels or bit-by-bit encryption, generally. So, PSNRs of color images are lower than those of grayscale images. As shown in the second column of Table 5, the PSNRs are all around 27%, which is similar to Liu et al.'s scheme [40]). Based on the black and white schemes, Liu et al. propose a color k-out-of-n VCS to divide a natural color image into 24 binary images. The values of PSNR for this scheme are shown in the third column of Table 5. Although our method also uses bit-wise encryption for color images, images recovering from our scheme have higher quality, whose PSNRs are all above 40%.



Mode	Original	Recovered	Salt pepper attack
PSNR	100.00%	62.09%	59.39%
NC	100.00%	97.76%	95.73%
	(a)	(b)	(c)

Figure 9: Extracted watermarks.

Table 6: Robustness illustration of T-VCS.

Images	PSNR	PSNR (salt-pepper)	NC (salt-pepper)
Barbara	43.49%	39.27	95.55%
Goldhill	43.20%	39.19	96.12%
Boat	41.89%	38.71	95.78%
Zelda	42.90%	38.99	91.23%
Baboon	44.76%	39.50	94.70%
Peppers	44.23%	39.21	95.18%
Airplane	40.92%	38.37	93.04%
Lena	42.48%	38.57	95.41%

4.2. Robustness Evaluation. There are usually two metrics with respect to evaluate watermarking algorithms: imperceptibility and robustness. (i) Imperceptibility means that the presence of the watermark should not distort the perceived quality of the host image. The PSNR is typically used to measure imperceptibility. (ii) Robustness is a measure of the immunity of the watermark against attempts to remove or degrade it. We measured the similarity between the original watermark and the watermark extracted from the attacked image using the NC (normal correlation factor) given in the following equation:

$$\mathrm{NC}(w, \widehat{w}) = \frac{\sum_{i=1}^{N} w_i \widehat{w_i}}{\sqrt{\sum_{i=1}^{N} w_i^2 \sqrt{\sum_{i=1}^{N} \widehat{w_i}^2}}}, \tag{4}$$

where N is the number of pixels in the watermark and w and $\widehat{w}$ are the original and extracted watermarks, respectively. In general, an NC of about 0.75 or above is considered acceptable [41].

We first evaluate the performance of the zero watermarking algorithm in T-VCS using a 512 × 512 Lena image as the original cover host image, and a 256 × 256 black-white image with the expression PATTERN as the watermark image. A pseudorandom number generator generates the secret data used in our experiments. Figure 9 shows PSNR and NC values among the original watermark, the watermarks extracted from T-VCS, and the extracted watermark after being subjected to the salt-pepper attack independently. The value of NC for the extracted watermark is higher than 0.95. We can clearly see the expression value from the watermark. We also verify the performance of watermark extraction on the above eight images. As shown in Table 6, the values of PSNR are above 40%. After applying the attack of salt-pepper, the values of PSNR drop by only 1%, which is within acceptable limits. The values of NC are all above 91%, indicating the algorithm has strong robustness and can resist salt-and-pepper noise.

Experimental results show that the proposed scheme does not suffer from salt-and-pepper noise. Although we can use the Discrete Wavelet Transform- (DWT-)/Discrete Cosine Transform- (DCT-) based digital image watermarking algorithms [42] to wavelet transform the image to obtain a more robust watermark, it will further reduce the image resolution. For two sheets, operations such as wavelet transform and inverse transform will destroy the corresponding relationship of pixels, resulting in the restoration of recovered image. Therefore, we apply no transform to preprocess sheets but directly use the way of quantization to odd/even to embed the watermark into lossless sheets.

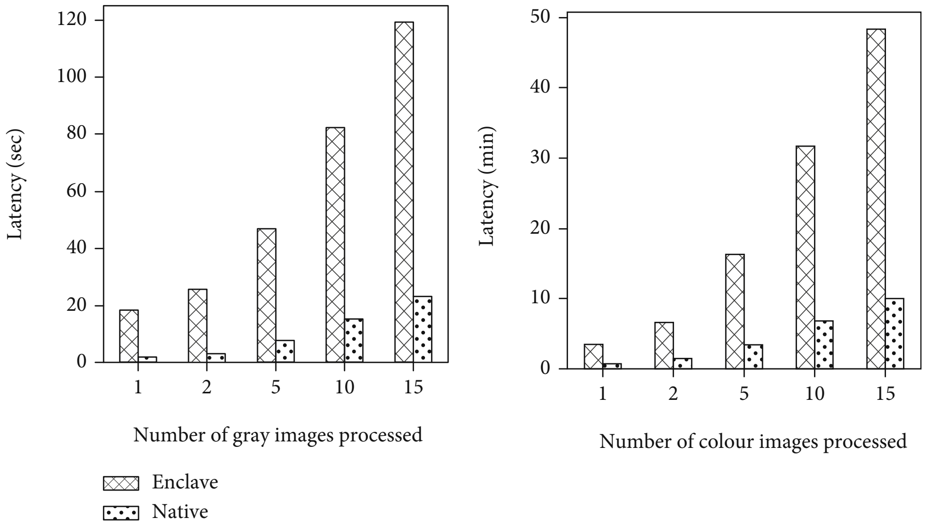
4.3. Performance Evaluation of Running T-VCS in TEE. In this section, we mainly evaluate the performance of online verification sharing when running T-VCS in TEE. The performance of sheet distribution is not our main concern because we can do it online or offline.

There are usually two concerns when using SGX to prevent cheating of VCS: one is the code refactoring [43]. It is not an easy task to port an application to run within an SGX container because Intel has envisioned SGX as a protection technology for only small parts of the application code and data. The native application code often has to be modified to meet the implicit prerequisite. While a fully featured library, OS [44] can rapidly deploy unmodified applications.

To quickly verify the proposed TEE solution in this paper, we first develop a T-VCS prototype system on a native OS by Python and then make use of Graphene-SGX and *ptrace* interposition to run the system in SGX. Another reason behind this is that easy data exploration and visualization are often more important than writing the most optimized solution [33]. We selected the Graphene-SGX as the library OS. It should be noted that although the Graphene-SGX supports running Python, and all dependent files must be specified manually, so it is impractical to run complex Python programs like T-VCS which has 516 lines of Python code. We first make use of *ptrace* interposition to trace system calls invoked by the T-VCS and then extract dependencies into a manifest. The patched Graphene-SGX ensures the integrity of T-VCS in runtime by verifying these manifest files.

All benchmarks are measured on an HP Z240 SFF Workstation with Intel Xeon Intel i7-7700 3.6 GHz processor (with Skylake microarchitecture, 4 cores, and SGX version 1) and 16 GB RAM. We install Intel's SGX Linux Driver and SDK 2.0 on the Ubuntu 18.04.3 LTS.

As shown in Figure 10, the processing time to run T-VCS in SGX is higher than that in native Linux. One of the reasons is that the enclave creation time which an application has to pay to run on SGX is relatively higher. The time is determined by the latency of the hardware and the driver. It is primarily a function of the size of the enclave. As shown in Figure 10(a), with the number of processed images



(a) Number of gray images processed

(b) Number of color images processed

Figure 10: Performance overhead to run T-VCS in SGX and native Linux.

Table 7: Performance overhead of remote attestation in T-VCS.

Time (sec)	Sigrl	Report	Load enclave	Net	Other
2.5004	1.0274	1.42	0.0105	0.0003	0.0394

increasing, the performance overhead decreases from 11 : 3x (for one image) to 5 : 1x (for 15 images). The overload to load enclave and library OS is gradually weakened.

The performance overhead for processing color images is shown in Figure 10(b), which is also about 5x of the native program. The reason why the order of performance overhead rises to the minute is that the color image has three channels, each of which has a bit length of 8. So, it is equivalent to processing 24 black and white images for a color image. However, the SGX and library OS technologies are not friendly to parallel processing. Furthermore, to use the remote attestation feature of SGX, the hyperthreading function, which is vulnerable to side-channel attack, must be disabled in the BIOS, which further limits the parallelism of SGX. So, in the experimental results, the performance overhead of Figure 10(b) is approximately 24x of Figure 10(a).

The overload to verify sheets are as shown in Table 7. It should be noted that in order to achieve a balance between the SGX programming model and the legacy code, the test program only generates a signed verification report. The verification process is still completed in the library OS. The SGX SDK already provides a basic framework for remote attestation. We patched the *tkey_exchange* library to add the hash of the sheets to be verified, which ensures this verification is involved with the sheets. It can be seen from Table 7 that the network overload (Sigrl+Report) with the IAS takes up most of the overload (97.88%). The network overload *Net* between the T-VCS and participants is only 0.0003 seconds, which is because the T-VCS and participant programs are on the same host. Compared with the sharing procedure, the overload of the verification code is negligible.

It should be noted that this prototype system does not use optimization techniques and its efficiency needs to be improved. With the development of SGX supported server hardware [45], we can enhance parallel capabilities of T-VCS and launch up a local authentication server to neutralize overload.

5. Conclusions

In this paper, we proposed a novel VCS which improves the pixel expansion and contrast properties compared with many of the known results in the literature. By encrypting an image by pixel blocks, we eliminate pixel expansion. We archive higher contrast by elaborately processing the correspondence between a secret image and its sheets. The multiple permutation ways of the same pixel block in T-VCS provides artifice for subsequent zero watermarking verification. The SGX technology used in this paper simplifies the verification model and provides authenticatable verification results even if the software or OS is compromised. The proposed scheme is efficient and straightforward and can be applied to various images, as shown in the experimental results.

Unfortunately, part of the information about the original share images disappears in the recovered secret image in T-VCS. It is hard to eliminate such a phenomenon, but it is possible to find a method to weaken it. Furthermore, in traditional VCSs, each participant only holds a sheet, while the proposed scheme needs each participant to hold multiple sheets, which is inconvenient for management. Reducing the number of sheets will also become our future work.

Data Availability

The data used to support the findings of this study were supplied by Denghui Zhang under license and so cannot be made freely available. Requests for access to these data should be made to Denghui Zhang (zhang.denghui@foxmail.com).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (2019YFB1706003 and 2018YFB1004003), the National Natural Science Foundation of China under Grant 61902082, and the Guangdong Key Research and Development Program of China (2019B010136003).

References

[1] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[2] X. Wang, L. T. Yang, Y. Wang, L. Ren, and M. J. Deen, "ADTT: a highly efficient distributed tensor-train decomposition method for IIoT big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1573–1582, 2021.

[3] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[4] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

[5] M. Naor and A. Shamir, "Visual cryptography," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 1–12, Springer, 1994.

[6] A. G. Forte, J. A. Garay, T. Jim, and Y. Vahlis, "EyeDecrypt —private inter- actions in plain sight," in *Security and Cryptography for Networks, Lecture Notes in Computer Science*, M. Abdalla and R. Prisco, Eds., pp. 255–276, Springer International Publishing, 2014.

[7] S. J. Andrabi, M. K. Reiter, and C. Sturton, "Usability of augmented reality for revealing secret messages to users but not their devices," in *Eleventh Symposium on Usable Privacy and Security (SOUPS 2015)*, pp. 89–102, USENIX As- sociation, Ottawa, 2015.

[8] D. Chaum, "Secret-ballot receipts: true voter-verifiable elections," *IEEE Security Privacy*, vol. 2, no. 1, pp. 38–47, 2004.

[9] F. Liu, W. Q. Yan, P. Li, and C. Wu, "ESSVCS: an enriched secret sharing visual cryptography," in *Transactions on Data Hiding and Multimedia Security IX: Special Issue on Visual Cryptography, Lecture Notes in Computer Science*, Y. Q. Shi, F. Liu, and W. Yan, Eds., pp. 1–24, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[10] F. Liu, C. Wu, and X. Lin, "Step construction of visual cryptography schemes," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 27–38, 2010.

[11] C.-M. Hu and W.-G. Tzeng, "Cheating prevention in visual cryptography," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 36–45, 2006.

[12] F. Liu, C. Wu, and X. Lin, "Cheating immune visual cryptography scheme," *IET Information Security*, vol. 5, no. 1, pp. 51–59, 2011.

[13] R. Amankwah, P. K. Kudjo, and S. Y. Antwi, "Evaluation of software vulnerability detection methods and tools: a review," *International Journal of Computers and Applications*, vol. 169, no. 8, pp. 22–27, 2017.

[14] W. Qiang, Z. Dong, and H. Jin, "Se-Lambda: securing privacy-sensitive server- less applications using SGX enclave," in *Security and Privacy in Communication Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, R. Beyah, B. Chang, Y. Li, and S. Zhu, Eds., pp. 451–470, Springer International Publishing, Cham, 2018.

[15] A. Ross and A. Othman, "Visual cryptography for biometric privacy," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 70–81, 2011.

[16] Y.-F. Chen, Y.-K. Chan, C.-C. Huang, M.-H. Tsai, and Y.-P. Chu, "A multiple-level visual secret-sharing scheme without image size expansion," *Information Sciences*, vol. 177, no. 21, pp. 4696–4710, 2007.

[17] C.-N. Yang, "New visual secret sharing schemes using probabilistic method," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 481–494, 2004.

[18] Y. Hou and C. Tu, "Visual cryptography techniques for color images without pixel expansion," *Journal of Information, Technology and Society*, vol. 1, pp. 95–110, 2004.

[19] P. Tuyls, H. D. Hollmann, J. H. Van Lint, and L. Tolhuizen, "XOR-based visual cryptography schemes," *Designs, Codes and Cryptography*, vol. 37, no. 1, pp. 169–186, 2005.

[20] M. Naor and A. Shamir, "Visual cryptography II: improving the contrast via the cover base," in *Security Protocols, Lecture Notes in Computer Science*, M. Lomas, Ed., pp. 197–202, Springer Berlin Heidelberg, 1997.

[21] Y.-C. Chen, D.-S. Tsai, and G. Horng, "Visual secret sharing with cheating prevention revisited," *Digital Signal Processing*, vol. 23, no. 5, pp. 1496–1504, 2013.

[22] Y.-C. Hou, "Visual cryptography for color images," *Pattern Recognition*, vol. 36, no. 7, pp. 1619–1629, 2003.

[23] C.-N. Yang and C.-S. Laih, "Some new types of visual secret sharing schemes," *National Computer Symposium*, vol. 3, pp. 260–268, 1999.

[24] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[25] K. A. Küçük, A. Paverd, A. Martin, N. Asokan, A. Simpson, and R. Ankele, "Exploring the use of Intel SGX for secure many-party applications," *Proceedings of the 1st Workshop on System Software for Trusted Execution, SysTEX '16*, , pp. 5:1–5:6, ACM, New York, NY, USA, 2016.

[26] B. Yang, K. Yang, Y. Qin, Z. Zhang, and D. Feng, "DAA-TZ: an efficient DAA scheme for mobile devices using ARM TrustZone," *IACR Cryptology ePrint Archive*, pp. 209–227, 2015.

[27] S. Arnautov, B. Trach, F. Gregor et al., "SCONE: secure Linux containers with Intel SGX," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 689–703, USENIX Association, Savannah, GA, 2016.

[28] A. Baumann, M. Peinado, and G. Hunt, "Shielding applications from an untrusted cloud with haven," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pp. 267–283, USENIX Association, Broom- field, CO, 2014.

[29] C. C. Tsai, D. E. Porter, and M. Vij, "Graphene-SGX: a practical library OS for unmodified applications on SGX," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pp. 645–658, Santa Clara, CA, 2017.

[30] F. Schuster, M. Costa, C. Fournet et al., "VC3: trustworthy data analytics in the cloud using SGX," in *2015 IEEE symposium on security and privacy*, pp. 38–54, IEEE, San Jose, CA, 2015.

[31] F. Tramer and D. Boneh, "Slalom: fast, verifiable and private execution of neural networks in trusted hardware," in *International Conference on Learning Representations*, New Orleans, LA, USA, 2019.

[32] Z. Gu, W. Hu, C. Zhang, H. Lu, L. Yin, and L. Wang, "Gradient shielding: towards understanding vulnerability of deep neural networks," *IEEE Transactions on Network Science and Engineering*, 2020.

[33] F. Shaon, M. Kantarcioglu, Z. Lin, and L. Khan, "SGX-BigMatrix: a practical encrypted data analytic framework with trusted processors," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17*, pp. 1211–1228, New York, NY, USA, 2017.

[34] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for CPU based attestation and sealing," *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, , ACM, New York, NY, USA, 2013.

[35] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, *Using Innovative Instructions to Create Trustworthy Software Solutions, HASP@ ISCA 11*, 2013.

[36] Z. Gu, T. Shen, Y. Wang, and F. C. M. Lau, "Efficient rendezvous for heterogeneous interference in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 91–105, 2020.

[37] X. Wang, L. T. Yang, L. Song, H. Wang, L. Ren, and M. J. Deen, "A tensor- based multiattributes visual feature recognition method for industrial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2231–2241, 2021.

[38] Z. Gu, Y. Wang, T. Shen, and F. C. M. Lau, "On heterogeneous sensing capability for distributed rendezvous in cognitive radio networks," *IEEE Transactions on Mobile Computing*, 2020.

[39] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[40] F. Liu, C. K. Wu, and X. J. Lin, "Colour visual cryptography schemes," *IET Information Security*, vol. 4, no. 2, pp. 151–165, 2008.

[41] A. Al-Haj, "Combined DWT-DCT digital image watermarking," *Journal of Computer Science*, vol. 3, no. 9, pp. 740–746, 2007.

[42] K. Deb, M. S. Al-Seraj, M. M. Hoque, and M. I. H. Sarkar, "Combined DWT- DCT based digital image watermarking technique for copyright protection," in *2012 7th International Conference on Electrical and Computer Engineering*, pp. 458–461, Dhaka, Bangladesh, 2012.

[43] J. Lind, C. Priebe, D. Muthukumaran et al., "Glamdring: automatic application partitioning for Intel SGX," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pp. 285–298, Santa Clara, CA, 2017.

[44] D. E. Porter, S. Boyd-Wickizer, J. Howell, R. Olinsky, and G. C. Hunt, "Rethinking the library OS from the top down," in *ACM SIGARCH Computer Architecture News*, vol. 39, pp. 291–304, ACM,, 2011.

[45] F. McKeen, I. Alexandrovich, I. Anati et al., "Intel software guard extensions (intel sgx) support for dynamic memory management inside an enclave," *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*, , pp. 1–9, ACM, 2016.