Partial image encryption using format-preserving encryption in image processing systems for Internet of things environment

Wonyoung Jang\textsuperscript{1} and Sun-Young Lee\textsuperscript{2}

Abstract

Concomitant with advances in technology, the number of systems and devices that utilize image data has increased. Nowadays, image processing devices incorporated into systems, such as the Internet of things, drones, and closed-circuit television, can collect images of people and automatically share them with networks. Consequently, the threat of invasion of privacy by image leakage has increased exponentially. However, traditional image-security methods, such as privacy masking and image encryption, have several disadvantages, including storage space wastage associated with data padding, inability to decode, inability to recognize images without decoding, and exposure of private information after decryption. This article proposes a method for partially encrypting private information in images using FF1 and FF3-1. The proposed method encrypts private information without increasing the data size, solving the problem of wasted storage space. Furthermore, using the proposed method, specific sections of encrypted images can be decrypted and recognized before decryption of the entire information, which addresses the problems besetting traditional privacy masking and image encryption methods. The results of histogram analysis, correlation analysis, number of pixels change rate, unified average change intensity, information entropy analysis, and NIST SP 800-22 verify the security and overall efficacy of the proposed method.

Keywords
Partial image encryption, format-preserving encryption, FF1, FF3-1, IoT

Date received: 7 November 2019; accepted: 3 March 2020

Handling Editor: Dush Nalin K Jayakody

Introduction

With continuing advancements in the development of image processing equipment, image data, such as images of people, are increasingly being collected and used in various fields. Image processing equipments are also increasingly being connected to network environments. Images are intuitive and sensitive, which means that without proper privacy protection, data leaks, such as privacy breaches, and confidential information leaks can occur.\textsuperscript{1–7} As such threats increase in severity and number, concerns about privacy are growing, spurring the requirement for image information to be stored and protected by encryption based on a security algorithm.\textsuperscript{8–20}

Privacy masking and image encryption are commonly applied as image data security measures. Privacy
masking involves methods such as blurring, mosaic masking, removal, and deformation. Blurring of a specific area in an image makes it difficult to recognize objects. Mosaic masking is a technique that blurs a specific area of a picture by weighting it in a tile form. Removal and deformation involve deleting data in a specific area or modifying data using arbitrary values. These privacy masking techniques all involve overwriting data in the original image. Although they are fast and efficient because of the low performance load involved, they make it impossible to recover the original data. This results in the problem of such methods only being applicable with images that do not need to be recovered, unless the original image is stored separately. There are two methods of encrypting image data: (1) full encryption, which encrypts the entire image using the likes of block-cipher-based and chaos-map-based encryption, and (2) partial encryption, which selectively encrypts only the sensitive parts of an image. Because of the nature of image file structures, it is difficult to encrypt only a portion of an image using image full encryption. Therefore, encrypted images cannot be recognized by users because all of the data must be encrypted—not only the data that would result in privacy violations if revealed. Private information in an image is defined as a visible face, part of the body, or vehicle registration number that could be used to identify an individual. When using a block cipher, which is typically used to encrypt text data for image encryption, the size of the encrypted image can be increased by padding. In particular, for an image, when the amount of data increases due to padding in every frame, storage space may be wasted. In this article, we propose a partial image encryption method using format-preserving encryption that can protect information requiring privacy and utilize images while overcoming these problems. Format-preserving encryption is an algorithm that is suitable for encryption of data that have a fixed length or format and characteristics that do not increase the length of the cryptogram. Therefore, it is possible to encrypt only the private information in an image, and because the data size does not increase, the encrypted data can be stored effectively without wasting storage space. The proposed method does not result in an increase in the amount of data due to the application of padding when encrypting only part of an image. In addition, the proposed method permits encrypted images to be recovered, addressing a problem associated with privacy masking. Furthermore, it allows encrypted images to be recognized by encrypting only private information. This article describes the proposed method and presents the results of comparisons of the method with existing video security methods. The remainder of this article is organized as follows. Section “Related work” gives an overview of image processing systems, traditional privacy masking methods, image encryption methods, and format-preserving encryption. Section “Contribution” introduces the advantages of the proposed method. Section “Proposed method” describes the proposed new method for encrypting parts of images using format-preserving encryption. Section “Experiment evaluation” compares the encryption results with traditional image security. Sections “Performance” and “Optimal block size” measure the encryption/decryption speed and derives the optimal encryption block size, respectively. Section “Security evaluation” presents the results of security analyses and other comparisons of the proposed method with existing image-security techniques. Conclusions are presented in section “Conclusion.”

Related work

Closed-circuit television camera system model

A closed-circuit television (CCTV) camera system takes a picture of an image and stores it as data in a manner that a user can see. In a CCTV system, a CCTV camera is connected to an image recorder. A digital video recorder (DVR) processes the codec algorithm and analog image that fit the image sent from various types of CCTVs into the type and form that fits the intended purpose. The video is stored on a hard disk in the DVR machine and is transmitted to a video storage server using Secure Sockets Layer (SSL)-encrypted communication. The video storage server encrypts and stores all the video sent from the DVR to protect private information. Control room operators and external users can see the images saved in the data storage server with appropriate user authentication. Figure 1 shows a schematic illustration of the CCTV camera system model.

Privacy masking techniques

Privacy masking techniques, such as blurring, mosaic masking, removal, and deformation, protect sensitive or personal information by changing some data in an image so that the original information cannot be seen. When these methods are used, it is difficult to recover the original image. Blurring and mosaic masking techniques blur the original image so that it cannot be properly recognized. Removal and deformation methods delete original image pixel values or overwrite original image pixel values with different values using arbitrary values. Privacy masking methods do not permit recovery of the original image because the original image data are modified. Another disadvantage of such methods is that the original image must be inferred, although the exact original is unknown.
There are two ways to protect an image from eavesdropping and leakage: full encryption and partial encryption. Full image encryption—encryption of the entire image—is used when it is necessary to perform encryption efficiently and quickly. Symmetric key block-cipher-based or chaotic-map-based image encryption methods, as well as convolution operation-based and hyper-chaos-based encryption, are currently being studied.\textsuperscript{22–30} Partial image encryption—also known as “selective image encryption”—is used to protect sensitive information or sensitive information within an image. Partial image encryption includes a method for selective encryption of a specific spatial area\textsuperscript{31–36} and another method used to encrypt part of an image based on unique attributes such as the frequency of the image.\textsuperscript{37,38}

Format-preserving encryption

Format-preserving encryption is an encryption approach in which the image data length is the same as that of the plain text and the form is that of the cipher.\textsuperscript{39} Based on the prefix cipher, cycle-walking cipher, generalized Feistel cipher, rank-then-encryption (RTE), and the tweak concept, practical approaches to format-preserving encryption have been proposed.\textsuperscript{40,41} Standard format-preserving encryption methods that have been proposed include FF1 (FFX mode), FF2 (VAES3), and FF3 (BPS).\textsuperscript{42–46} However, with the discovery of security vulnerabilities in FF2 and FF3, FF3-1, which is a modification of FF1 and FF3, has been established as the US National Institute of Standards and Technology (NIST) standard.\textsuperscript{47,48} Standard format-preserving encryption requires three pieces of information for encryption: plain text, a key, and a tweak. It does not matter if the tweak is public. Even if the key is the same as the plain text, it is safe from codebook attack because the cipher text changes every time the tweak changes. The method proposed in this article uses FF1 and FF3-1, standard format-preserving encryption methods. Figure 2 shows the Feistel structure of FF1 and FF3-1.

FF1 and FF3-1 divide plain text into components $A_0$ and $B_0$ in the best possible balance. For example, if you divide 10-byte plain text, the first five bytes are $A_0$ and the last five bytes are $B_0$. $B_0$ is encrypted using the advanced encryption standard (AES) encryption function $F_K$, along with the byte length $n$, tweak $T$, and round number of the message. The key uses a 128-bit AES key. The encrypted value is added to $A_0$, and then mod radix calculation is performed to make $C_0$. $C_0$ becomes $B_1$, $B_0$ becomes $A_1$, and a round ends. This process is repeated 10 times in total. Finally, when $A_{10}$ and $B_{10}$ are concatenated, the encryption process ends.

**Contribution**

The proposed method has the following three advantages:

- Specific spatial image partial encryption is possible;
- The amount of image data does not increase as a result of padding;
- Images can be used without exposing information requiring privacy.

Traditional image protection methods include privacy masking, image full encryption, and image partial
encryption. Privacy masking incurs a problem in that it cannot be decrypted. The image full encryption method encrypts all the pixels so that the image cannot be recognized at all until after decryption, and once decrypted, information requiring privacy cannot be protected. Traditional image partial encryption encrypts a large rectangular range that includes the area requiring protection. This incurs a problem in that even non-sensitive parts of the image are encrypted. In addition, the use of a block cipher causes problems because it produces an image file that does not conform to the structure determined by padding. The proposed method performs encryption by targeting a pixel range that specifically covers an area that the user wishes to encrypt. There are various methods for detecting objects to be encrypted in images, such as the use of deep learning, OpenCV, or software. The data can be recognized and used normally except for the encrypted area, such that images can still be used while protecting private information. For example, if the traditional image partial encryption was used to partially encrypt the license plate of a parked vehicle, and the person inside, as captured by CCTV, the entire image or the vehicle’s license plate and the person to be encrypted must be encrypted with a rectangle. Encrypting an unnecessarily large image can adversely affect image recognition. In contrast, the proposed method allows encryption of just the license plate and the person in the car. Therefore, it is possible to collect images while solving the problem of invasion of citizens’ privacy. In addition, privacy masking is not decrypted, but the proposed method has the advantage of allowing the decryption and use of an encrypted area as needed (such as in the case of a crime committed in the street). The format-preserving encryption used for encryption can encrypt even short-length data without padding. Therefore, there is no increase in the amount of data due to padding.

**Proposed method**

**Partial image encryption using format-preserving encryption**

This article proposes an image partial encryption method based on format-preserving encryption. It was implemented and tested using FF1 and FF3-1. The proposed method, which can be used to select and encrypt a portion of an image while preserving the data size, consists of three steps: (1) selection of the encryption area, (2) encryption of image pixel values using format-
preserving encryption, and (3) storage of encrypted image pixel values. First, to select the area to be encrypted in a video, the area that corresponds to private information that can identify an individual, such as a face, a part of a body, or a vehicle number, must be identified. The X and Y coordinate values of all pixels in the selected area are checked, and the pixel values are stored in the byte array. For example, a red-green-blue (RGB)-format image stores three bytes of pixel values per pixel, while a gray-format image stores one byte of pixel values per pixel. The proposed method can be applied to both RGB- and gray-format images. The pixel values stored in the byte array are encrypted using format-preserving encryption. When encrypting using format-preserving encryption, data can be divided by block size. In this article, the block size is set to 24 bytes for FF1 and 32 bytes for FF3-1. Details of the block size setting are described in section “Optimal block size.” The encryption process requires plain text, a key, and a tweak. The plain text is the pixel value of the selected area, and the key uses a 128-bit AES key. The tweak uses the X and Y coordinates of all of the pixel data in the block to be encrypted. Format-preserving encryption use a tweak in the encryption and decryption process, unlike ordinary block ciphers. If the pixel values are encrypted using the same tweak, one can visually infer the original image because the same pixel value becomes the same cipher text. In this case, the correlation between pixels may be vulnerable to known and selective plain text attacks. To solve this problem, the tweak must be a constantly changing value. The proposed method uses tweak as the value obtained by concatenating the X and Y coordinates of the pixel value to be encrypted. The final step in the process is encrypting all of the pixel values in the selected area. The decryption process is the reverse of the encryption process. Table 1 summarizes the fields and functions used in the encryption/decryption process of the proposed method. Figure 3 shows the entire process of encrypting pixel values in an RGB-format image using the proposed method. Algorithm 1 shows the process of dividing the pixel value into predetermined block units before the encryption process of the proposed method is applied. Algorithm 2 calculates tweak $T$ values used in $fpeEncryption$ and $fpeDecryption$. Algorithms 3 and 4 detail the $fpeEncryption$ and $fpeDecryption$ processes of the proposed method, respectively. Figure 4 presents an example of the encryption process used in the proposed method.

As shown in Figure 3, we select the X and Y coordinates and the pixel values of the privacy information in the image. Then, a $PixelArray$ is created by concatenating the selected pixel values in 24 bytes. $PixelArray$ values are computed with the $fpeEncryption$ and $fpeDecryption$ functions, respectively, according to encryption/decryption. As a result, the pixel data $EncPixel$ value encrypted in the generated $C$ are stored as the pixel values of the existing X and Y coordinates. As Figure 4 shows, only the area selected as the private information in the original image is encrypted. In addition, the private information can be encrypted and protected, and users can use encrypted images.

### Application scenario

The proposed method can be used in various applications such as CCTV camera systems and image drone systems. Figure 5 is an activity diagram of a scenario in which the proposed method is applied to a CCTV camera system.

As shown in the figure, a CCTV camera shoots the video, which is then transmitted to the storage server. The server encrypts the privacy information of the video received from the CCTV camera using the proposed method and stores the video in the server. Security officers and general users can view the videos with privacy information encrypted by connecting to the server or over the Internet. Figure 6 shows a CCTV camera system incorporated with improved security and applicability provided by the proposed method.

### Experiment evaluation

**Experimental results**

The proposed method was applied to images (6.1.01, 4.1.02, 5.2.08, house, 5.3.01, and 2.2.02) from the USC-
SIP database of various formats and sizes. Figure 7 shows the original image, while Figures 8–10 show the results of full image encryption using AES, FF1, and FF3-1, respectively. Figures 11 and 12 show the results of encrypting only the information requiring privacy in the image using the proposed method.

Increasing data size in videos

A video is data in which multiple frames are collected. When a captured video is encrypted in real time, the image data size may increase due to padding. Table 2 summarizes the increase in the data size over time when all the frames of a captured video are encrypted in real time. The data size increases as a result of the addition of 16 bytes padding data to every frame under various frame conditions. The units are Mb.

For CCTV, which involves the continuous recording of video, the size of the video may increase over time such that storage space is wasted. As the number of CCTV cameras capturing video increases, the wastage of server storage space worsens.

Comparison with traditional image-security methods

The proposed method was compared with the traditional image-security methods to assess the possibility of decryption, the degree to which file sizes increase, and the possibility of recognizing encrypted images. The following problems arise when the traditional image-security approach is applied to an image processing system. Privacy masking prevents decoding because the original data are lost by modifying pixel values. Full image encryption using a block encryption method, such as AES, does not permit encryption of a part of an image; the entire image must be encrypted. Therefore, the image cannot be used until decoding is performed, and private information is exposed when the decoding is performed. In addition, because of the characteristics of the block code, the size of the image is invariably increased by padding. Figure 13 shows the results of encryption using privacy masking, image full encryption, and the proposed method.

Figure 13(b) shows that private information can be inferred when mosaic masking is used and is not decrypted. Figure 13(b) shows mosaic image using the...
pixelate of Adobe Photoshop. Figure 13(c) shows the image encrypted for all pixel data using AES-128 CBC mode. The encrypted image is unrecognizable, and the data size is increased by padding. Figure 13(d) and (e) shows the result of encrypting the image by applying the proposed method. The image remains recognizable while the private information in the image is protected, and the data size of the image does not increase. If video is encrypted by applying the proposed method in a CCTV camera system, both the monitoring control room and the users connected to the Internet can use

---

**Algorithm 1.** PixelArray schedule

1: procedure PixelArray (PixelArray<sub>n</sub>, n)
2:  m = [n/8]
3:  for i in 1 to (m - 1) do
4:       q = (i - 1) × 8
5:       PixelArray<sub>i</sub> = SelectPixel[q + 1 · · · q + 8]
6:  end for
7:  PixelArray<sub>m</sub> = SelectPixel[8(m - 1) + 1 · · · n]
8:  Return PixelArray = PixelArray<sub>1</sub> || PixelArray<sub>2</sub> || · · · || PixelArray<sub>m</sub>
9: end procedure

**Algorithm 2.** Tweak schedule 1

1: procedure T (Pixel<sub>r</sub><sup>L</sup>, Pixel<sub>r</sub><sup>R</sup>)
2:  m = [n/8]
3:  for i in 1 to m do
4:       q = (i - 1) × 8
5:       T<sub>i</sub> = Pixel<sub>r</sub><sup>L</sup><sub>q+1</sub> || Pixel<sub>r</sub><sup>R</sup><sub>q+1</sub> || · · · || Pixel<sub>r</sub><sup>L</sup><sub>q+8</sub> || Pixel<sub>r</sub><sup>R</sup><sub>q+8</sub>
6:  end for
7:  Return T = T<sub>1</sub> || T<sub>2</sub> || · · · || T<sub>m</sub>
8: end procedure
Algorithm 3. Encrypt function

1: procedure fpeEncrypt $(K, T, PixelArray)$
2: $u = \lceil 1/2 \rceil; v = l - u$
3: $A = PixelArray[1 \ldots q]; B = PixelArray[u + 1 \ldots l]$
4: $b = \lceil \frac{x \cdot \log(radix)}{8} \rceil$
5: $d = 4\lceil \frac{b}{4} \rceil + 4$
6: $P = \lceil 2^{(\left\lfloor \frac{d}{16}\right\rfloor)} \rceil \left\lfloor \frac{d}{16} \right\rfloor + 1$
7: for $i$ in 0 to 9 do
8: $Q = T || 0^{[t-n-1]} || \left\lfloor \frac{d}{16} \right\rfloor \left\lceil \frac{d}{16} \right\lfloor \left\lceil \frac{d}{16} \right\lfloor$
9: $R = \text{PRF}(P || Q)$
10: $S$ be the first $d$ bytes of the following string of $[d/16]$ block:
11: $R || \text{AES}_{k}(R \oplus [1]_{16}) || \text{AES}_{k}(R \oplus [2]_{16})$
12: $y = \text{HTD}(S)$
13: if $i$ is even then
14: $r = u$
15: else
16: $r = v$
17: end if
18: $c = (\text{HTD}(A) + y) \mod 16$
19: $O = \text{DTH}(c)$
20: $A = B$
21: $B = O$
22: end for
23: Return $A || B$
24: end procedure

Algorithm 4. Decrypt function

1: procedure fpeDecrypt $(K, T, PixelArray)$
2: $u = \lceil 1/2 \rceil; v = l - u$
3: $A = PixelArray[1 \ldots q]; B = PixelArray[u + 1 \ldots l]$
4: $b = \lceil \frac{x \cdot \log(radix)}{8} \rceil$
5: $d = 4\lceil \frac{b}{4} \rceil + 4$
6: $P = \lceil 2^{(\left\lfloor \frac{d}{16}\right\rfloor)} \rceil \left\lfloor \frac{d}{16} \right\rfloor + 1$
7: for $i$ in 9 to 0 do
8: $Q = T || 0^{[t-n-1]} || \left\lfloor \frac{d}{16} \right\rfloor \left\lceil \frac{d}{16} \right\lfloor \left\lceil \frac{d}{16} \right\lfloor$
9: $R = \text{PRF}(P || Q)$
10: $S$ be the first $d$ bytes of the following string of $[d/16]$ block:
11: $R || \text{AES}_{k}(R \oplus [1]_{16}) || \text{AES}_{k}(R \oplus [2]_{16})$
12: $y = \text{HTD}(S)$
13: if $i$ is even then
14: $r = u$
15: else
16: $r = v$
17: end if
18: $c = (\text{HTD}(B) + y) \mod 16$
19: $O = \text{DTH}(c)$
20: $A = B$
21: $B = O$
22: end for
23: Return $A || B$
24: end procedure

Figure 5. Activity diagram of application scenario in CCTV camera system with the proposed method.

The images while private information remains protected, without decrypting the data. In general, no decryption is required, so no key exchange is required, and because no padding is required for encryption, storage space is not wasted. Table 3 compares the results of the proposed method and traditional image-security methods in three respects: whether image decryption is permitted, whether and how file size is increased, and whether encrypted images can be recognized.

The proposed method solves the problems associated with privacy masking (i.e. that encrypted images cannot be decrypted) and encryption using block ciphers (i.e. increased file size and wasted storage space). The proposed method improves security by encrypting only private information in the image and improves the availability of images to users by providing image information other than private information.

**Performance**

**Speed**

We verified the performance of the proposed scheme by measuring the encryption/decryption rates for Figures 8–12. The experimental results are listed in Table 4.

**Optimal block size**

To improve the encryption/decryption speed of the proposed scheme, we determined the optimal block size when encrypting with the FF1 and FF3-1 algorithms. FF1 and FF3-1, as used in the proposed method, do not have a fixed block length because they must be able to encrypt various lengths. As the time required to encrypt an image varies depending on the length of the block, the length of a block suitable for encryption/decryption should be set. Figure 14 is a graph showing the encryption times for FF1 and FF3-1 according to block length using Figure 7(c) and (d). As the length of the block increases, the encryption speed of FF1 also
Figure 6. CCTV camera system equipped with proposed method.

Figure 7. Original image: (a) 6.1.01, (b) 4.1.02, (c) 5.2.08, (d) house, (e) 5.3.01, and (f) 2.2.02.

Figure 8. Result of full encryption using AES: (a) 6.1.01, (b) 4.1.02, (c) 5.2.08, (d) house, (e) 5.3.01, and (f) 2.2.02.

Figure 9. Result of full encryption using FF1: (a) 6.1.01, (b) 4.1.02, (c) 5.2.08, (d) house, (e) 5.3.01, and (f) 2.2.02.
increases, but it is slightly slower at 25 bytes. FF1 was the fastest from a length of 25 + (32 × a) bytes (e.g., 25, 57, 89). FF3-1 gets faster as the block length increases. Because of the characteristics of the algorithm, the block length of FF3-1 was set to a maximum of 32 bytes and tested. As a result, it can be seen that 32 bytes is the most efficient block length. Therefore, in the encryption process of the proposed method, FF1 encrypts the plain text into 24-byte units and FF3-1 encrypts it into 32 bytes.

### Security evaluation

To evaluate the security of the proposed method, histogram, correlation coefficient, number of pixels change

Table 2. Data size increase due to padding in full encryption.

|         | I hour | I day | I week | I month |
|---------|--------|-------|--------|---------|
| 24 fps  | 0.0219 | 0.5273| 3.6914 | 110.7421|
| 25 fps  | 0.0228 | 0.5493| 3.8452 | 115.3564|
| 30 fps  | 0.0274 | 0.6591| 4.6142 | 138.4277|
| 50 fps  | 0.0457 | 1.0986| 7.6904 | 230.7128|
| 60 fps  | 0.0549 | 1.3183| 9.2285 | 276.8554|
| 100 fps | 0.0915 | 2.1972| 15.3808| 461.4245|
| 120 fps | 0.1098 | 2.6367| 18.4570| 553.710  |

Table 3. Comparison of privacy masking, image encryption, and the proposed method.

|                      | Privacy masking | AES | Our method (FF1, FF3-1) |
|----------------------|-----------------|-----|-------------------------|
| Decryption           | ×               | ○   | ○                       |
| Awareness of image   | ○               | ×   | ○                       |
| File size            | No change       | Increase | No change |

AES: advanced encryption standard.
rate (NPCR), unified average change intensity (UACI), and information entropy analyses, and NIST SP 800-22 tests were performed.

**Histogram analysis**

A histogram analysis graphically illustrates the weights of the pixel values in an image. The more uniform the histogram distribution value, the better the security. Figure 15 shows the histogram results for Figures 7–12. The histogram distributions of the original images are all unbalanced. However, for full encryption, AES, FF1, and FF3-1 all produce balanced histogram graphs.

**Correlation coefficient analysis**

Correlation coefficient analysis indicates the independence of two images through correlation calculation between pixels. If the correlation value is linear, the independence of the two images is not good. The more nonlinear the correlation values, the more independent the two images. In other words, the higher the correlation between the two images, the closer the absolute value of the weight is to 1; conversely, the higher the independence, the closer the absolute value of the weight is to 0. Formula (1) is used to calculate the correlation value, where $x$ and $y$ are the mean pixel values of the two images and $N$ is the total number of pixels.

$$C_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)D(y)}}$$

Worksheet

| Figure 8 | Encryption (ms) | Decryption (ms) |
|---|---|---|
| 8(a) | 3.243099 | 3.216401 |
| 8(b) | 2.449099 | 2.410699 |
| 8(c) | 2.697301 | 2.6238 |
| 8(d) | 6.769 | 5.9916 |
| 8(e) | 3.248401 | 3.354301 |
| 8(f) | 9.5483 | 9.546301 |

| Figure 9 | Encryption (ms) | Decryption (ms) |
|---|---|---|
| 9(a) | 172.5461 | 39.0605 |
| 9(b) | 527.3073 | 14.93308 |
| 9(c) | 689.3037 | 4.533195 |
| 9(d) | 2076.729 | 196.1894 |
| 9(e) | 2759.026 | 1554.101 |
| 9(f) | 8283.313 | 846.7711 |

| Figure 10 | Encryption (ms) | Decryption (ms) |
|---|---|---|
| 10(a) | 196.2087 | 43.6244 |
| 10(b) | 589.5579 | 15.54559 |
| 10(c) | 783.7156 | 5.241196 |
| 10(d) | 2346.49 | 317.8314 |
| 10(e) | 3131.515 | 1642.126 |
| 10(f) | 9394.513 | 949.9847 |

| Figure 11 | Encryption (ms) | Decryption (ms) |
|---|---|---|
| 11(a) | 39.21519 | 176.96918 |
| 11(b) | 14.821 | 533.02506 |
| 11(c) | 4.549602 | 706.27681 |
| 11(d) | 195.3813 | 2125.5758 |
| 11(e) | 1571.194 | 2828.251 |
| 11(f) | 844.1251 | 8472.7691 |

| Figure 12 | Encryption (ms) | Decryption (ms) |
|---|---|---|
| 12(a) | 44.39931 | 198.293 |
| 12(b) | 15.78791 | 594.2996 |
| 12(c) | 5.242199 | 795.5995 |
| 12(d) | 325.2316 | 2380.341 |
| 12(e) | 1663.161 | 3145.791 |
| 12(f) | 958.1904 | 9462.259 |

Figure 13. Image encryption results: (a) original image (4,147,254 bytes), (b) privacy masking (4,147,254 bytes), (c) AES (4,147,270 bytes), (d) proposed method (FF1, 4,147,254 bytes), and (e) proposed method (FF3-1, 4,147,254 bytes).

Figure 14. Encryption speed by block length for Figure 7(c) and (d).
\[
\text{cov}(x, y) = \frac{1}{N} \left( \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \right)
\]  

\[
E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i
\]  

\[
D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2
\]

Table 5 shows the correlation coefficient weight values for RGB and gray images with respect to the original image for AES and the proposed method. These results show that the proposed method yields a correlation coefficient weight value similar to those obtained by image encryption with AES.

**NPCR and UACI analysis**

The resistance to a chosen plain text attack (differential attack) was evaluated using NPCR and UACI for an image encrypted using the proposed method and block cipher encryption. NPCR is the pixel variation rate between the plain text and the encrypted image, and UACI is the average variation intensity of plain text and the encrypted image. This analysis method can reveal whether the change in the original image affects encryption. If the original image is greatly changed in comparison with the encrypted image, it can be said to have good resistance to a chosen plain text attack (differential attack). NPCR and UACI are calculated using formulas (5)–(7)\textsuperscript{54,55}

\[
\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{w \times h} \times 100\%
\]
In formula (6), \( C_1 \) denotes a plain image, \( C_2 \) denotes an encrypted image, and \( w \) and \( h \) are the horizontal and vertical sizes of the image, respectively. \( F \) is a value that represents the maximum support pixel value compatible with the image format of the cipher text, and the image of the 8-bit format is 255. Tables 6 and 7 present the NPCR and UACI values, respectively, obtained for images in RGB and gray formats with Figures 8–12.

These results show that the proposed method yields NPCR and UACI values similar to those obtained through image encryption using block cipher (AES). In other words, the proposed method is as safe as image encryption using a block cipher (AES).

**Information entropy analysis**

Information entropy analysis is used to measure the complexity of the encrypted data. The encrypted data must be complex enough to prevent information on the original data from being obtained. The optimal entropy value of the RGB and gray images used is 8. Table 8 lists the entropy values for the original images in RGB and gray formats and those achieved by encrypting the images using the proposed method and full encryption.

The results in Table 8 show that the proposed method and image encryption using a block cipher...
(AES) yield values close to the optimal entropy. Therefore, both the proposed method and the image encryption using a block cipher (AES) are sufficiently complex.

**NIST SP 800-22 tests**

NIST SP 800-22 tests were used to test the cipher randomness. The goal of those tests is to analyze the randomness of the encrypted image data. The analysis consisted of 15 tests. For the experimental results obtained in this study, tests were performed on the data for the encrypted region in the 1024 × 1024 size image of Figures 11 and 12, and all the tests were passed to verify the randomness. The test results are listed in Table 9.

**Security analysis comparison**

To evaluate the security level of images encrypted with AES and with the proposed method, histogram, correlation, NPCR, UACI, information entropy analysis, and NIST SP 800-22 were performed as described above.

| Table 8. Results of information entropy analysis. |
|-----------------------------------------------|
| Figure 7 | 7.2044467 | 6.2944978 | 7.201008 | 7.4857872 | 7.5237369 | 6.5785567 |
| Figure 8 | 7.9971571 | 7.9989656 | 7.9991992 | 7.9997215 | 7.999797 | 7.9999211 |
| Figure 9 | 7.9978015 | 7.9990454 | 7.9993489 | 7.9997852 | 7.9998212 | 7.9999405 |
| Figure 10 | 7.9972393 | 7.9990452 | 7.9993289 | 7.9997645 | 7.9998074 | 7.9999412 |
| Figure 11 | 7.3347614 | 6.4000599 | 7.218075 | 7.5697277 | 7.858972 | 6.8655011 |
| Figure 12 | 7.3287123 | 6.3991863 | 7.2180681 | 7.5699342 | 7.8589302 | 6.8651993 |

| Table 9. Results of NIST SP 800-22 tests. |
|------------------------------------------|
| Frequency test | 0.5641175 | 0.3561378 | 0.5163866 | 0.7413325 | All pass |
| Frequency test within a block | 0.7400311 | 0.6041753 | 0.5178028 | 0.5233779 | All pass |
| Runs test | 0.5730168 | 0.103916 | 0.7509988 | 0.6392064 | All pass |
| Test for the longest run of ones in a block | 0.4921633 | 0.6371338 | 0.6157612 | 0.0386303 | All pass |
| Binary matrix rank test | 0.2662403 | 0.7795677 | 0.8560301 | 0.1314836 | All pass |
| Discrete Fourier transform test | 0.1750349 | 0.3813249 | 0.9341601 | 0.7652656 | All pass |
| Non-overlapping template matching test | 0.9996462 | 1.000000 | 1 | 1 | All pass |
| Overlapping template matching test | 0.4720836 | 0.4837326 | 0.257433 | 0.369052 | All pass |
| Maurer’s “Universal statistical” test | 0.8827247 | 0.1445012 | 0.4011447 | 0.8781729 | All pass |
| Linear complexity test | 0.1087466 | 0.3251907 | 0.3641394 | 0.2987047 | All pass |
| Serial test | 0.7523306 | 0.131774 | 0.0125111 | 0.1323363 | All pass |
| Approximate entropy test | 0.8939406 | 0.1823651 | 0.0125726 | 0.4899735 | All pass |
| Cumulative sums test | 0.4193308 | 0.4878375 | 0.4555566 | 0.8067146 | All pass |
| Random excursions test | 0.4600447 | 0.0400129 | 0.0354376 | 0.2340035 | All pass |
| Random excursions variant test | 0.1367363 | 0.0518082 | 0.1243495 | 0.2208814 | All pass |

The distribution of the histogram and correlation coefficient of the proposed method is good, and the results of NPCR and UACI are very close to those obtained with encryption using AES. In addition, the value of the information entropy is as close to the optimal value as that of the AES-encoded image. Finally, all the NIST SP 800-22 tests were passed. Therefore, the proposed method can be said to be a safe means of using both FF1 and FF3-1.

**Conclusion**

In this article, we proposed a method for partially encrypting private information, such as faces and body parts, in images using the format-preserving encryption standards FF1 and FF3-1. Traditional image protection technology has problems such as increasing data by padding and storage space wastage over time. Furthermore, as the entire image is encrypted, the image cannot be recognized before decryption and, once decrypted, information requiring privacy is exposed. Traditional partial image encryption incurs a problem in that unnecessary portions are encrypted by encrypting a rectangular area that covers the information requiring privacy. The proposed method solves this
problem. The proposed method has no padding, so there is no increase in the data size, and the information requiring privacy can be specifically set and encrypted. In this study, we measured the encryption and decryption speed of the proposed method and determined the most suitable block unit for encryption to improve the encryption and decryption speed of the image part. The security of the proposed method was evaluated through analysis of histograms, correlation coefficients, NPCR, UACI, information entropy, and NIST SP 800-22. We verified through experiments that the proposed method is as secure as encryption using AES. The proposed method can be used in various environments that use images, such as Internet of things (IoT), drones, and CCTV, and can be useful in responding to image leakage threats, such as privacy invasion and confidential information leakage.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT; No. 2018R1A4A1025632), the Basic Science Research Program through the NRF that is funded by the Ministry of Education (No. NRF-2018R1D1A1B07047656), and the Soonchunhyang University.

ORCID iD
Wonyoung Jang https://orcid.org/0000-0002-2777-9830

References
1. Yu J, Zhang B, Kuang Z, et al. iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning. IEEE Trans Inf Foren Sec 2016; 12(5): 1005–1016.
2. Wang C, Zhang B, Ren K, et al. Privacy-assured outsourcing of image reconstruction service in cloud. IEEE Trans Emerg Top Comput 2013; 1(1): 166–177.
3. Nov O, Naaman M and Ye C. Motivational, structural and tenure factors that impact online community photo sharing. In: Third international AAAI conference on Weblogs and social media, San Jose, CA, 17–20 May 2009. Menlo Park, CA: AAAI.
4. Squicciarini AC, Xu H and Zhang X. CoPE: Enabling collaborative privacy management in online social networks. J Am Soc Inf Sci Technol 2011; 62(3): 521–534.
5. Wang N, Xu H and Grossklags J. Third-party apps on Facebook: privacy and the illusion of control. In: Proceedings of the 5th ACM symposium on computer human interaction for management of information technology, Cambridge, MA, 4–5 December 2011, p.4. New York: ACM.
6. Cho JH and Ahn CW. Auto detection system of personal information based on images and document analysis. J Inst Internet Broadcast Commun 2015; 15(5): 183–192.
7. Bang J, Kang S and Kim MS. The study of factors to affect on users’ self-disclosure in social networking services. J Korea Acad Ind Cooperation Soc 2016; 17(8): 69–76.
8. Kumari S. A research paper on cryptography encryption and compression techniques. Int J Eng Comput Sci 2017; 6(4): 20915–20919.
9. De Choudhury M, Sundaram H, Lin YR, et al. Connecting content to community in social media via image content, user tags and user communication. In: 2009 IEEE international conference on multimedia and expo, New York, 28 June–3 July 2009, pp.1238–1241. New York: IEEE.
10. Yeung CMA, Kagal L, Gibbins N, et al. Providing access control to online photo albums based on tags and linked data. In: AAAI spring symposium on social semantic web: where Web 2.0 meets Web 3.0, Palo Alto, CA, 23–25 March 2009, pp.9–14. Menlo Park, CA: AAAI.
11. Pesce JP, Casas DL, Rauber G, et al. Privacy attacks in social media using photo tagging networks: a case study with Facebook. In: Proceedings of the 1st workshop on privacy and security in online social media, Lyon, 17 April 2012, p.4. New York: ACM.
12. Christin D, López PS, Reinhardt A, et al. Share with strangers: privacy bubbles as user-centered privacy control for mobile content sharing applications. Inform Secur Tech Rep 2013; 17(3): 105–116.
13. Mannan M and Van Oorschot PC. Privacy-enhanced sharing of personal content on the web. In: Proceedings of the 17th international conference on World Wide Web, Beijing, China, 21–25 April 2008, pp.487–496. New York: ACM.
14. Vyas N, Squicciarini AC, Chang CC, et al. Towards automatic privacy management in Web 2.0 with semantic analysis on annotations. In: 2009 5th international conference on collaborative computing: networking, applications and worksharing, Washington, DC, November 2009, pp.1–10. New York: IEEE.
15. Squicciarini AC, Shehab M and Paci F. Collective privacy management in social networks. In: Proceedings of the 18th international conference on World Wide Web, Madrid, 20–24 April 2009, pp.521–530. New York: ACM.
16. Klemperer P, Liang Y, Mazurek M, et al. Tag, you can see it! Using tags for access control in photo sharing. In: Proceedings of the SIGCHI conference on human factors in computing systems, Austin, TX, 5–10 May 2012, pp.377–386. New York: ACM.
17. Ravichandran R, Benisch M, Kelley PG, et al. Capturing social networking privacy preferences. In: International symposium on privacy enhancing technologies symposium, Seattle, WA, 5–7 August 2009, pp.1–18. Berlin: Springer.
18. Bonneau J, Anderson J and Church L. Privacy suits: shared privacy for social networks. In: Proceedings of the
19. Bonneau J, Anderson J and Danezis G. Prying data out of a social network. In: 2009 international conference on advances in social network analysis and mining, Athens, 20–22 July 2009, pp.249–254. New York: IEEE.

20. Fang L and LeFevre K. Privacy wizards for social networking sites. In: Proceedings of the 19th international conference on World Wide Web, Raleigh, NC, 26–30 April 2010, pp.351–360. New York: ACM.

21. Park SC. Design and implementation of personal information identification and masking system based on image recognition. J Inst Internet Broadcast Commun 2017; 17(5): 1–8.

22. Kaur M and Kumar V. A comprehensive review on image encryption techniques. Arch Comput Method Eng 2018; 27: 15–43.

23. Younes MAB. A survey of the most current image encryption and decryption techniques. Int J Adv Res Comput Sci 2019; 10(1): 9.

24. OVSENIKL´, Turán J, Huszani´k T, et al. Image encryption algorithm with plaintext related chaining. Comput Inf 2019; 38(3): 647–678.

25. Brindha K, Sharma R and Saini S. Use of symmetric algorithm for image encryption. Int J Innov Res Comput Commun Eng 2014; 2(5): 4401–4407.

26. Gupta K, Gupta R, Agrawal R, et al. An ethical approach of block based image encryption using chaotic map. Int J Secur Appl 2015; 9(9): 105–122.

27. Zhang Y. A fast image encryption algorithm based on convolution operation. IETE J Res 2017; 63(1): 4–18.

28. Li Y, Wang C and Chen H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt Lasers Eng 2017; 90: 238–246.

29. Oravec J, Turan J, Ovsenı´k L, et al. Asymmetric image encryption approach with plaintext-related diffusion. Radioengineering 2018; 27(1): 281–288.

30. Lagmiri SN, Elalami N and Elalami J. Color and gray images encryption algorithm using chaotic systems of different dimensions. Int J Comput Sci Network Secur 2018; 18(1): 79–86.

31. He J, Liu B, Kong D, et al. PUPPIES: transformation-supported personalized privacy preserving partial image sharing. In: 2016 46th annual IEEE/IFIP international conference on dependable systems and networks (DSN), Toulouse, 28 June–1 July 2016, pp.359–370. New York: IEEE.

32. Parameshchahiri BD, Kiran Rashmi P, Supriya MC, et al. Controlled partial image encryption based on LSIC and chaotic map. In: Proceedings of the 3rd international conference on cryptography, security and privacy, Kuala Lumpur, Malaysia, 19–21 January 2019, pp.60–63. New York: ACM.

33. Steinbach M, Liu H, Stein R, et al. Hybrid image encryption. Electron Imaging 2018; 2018(7): 1–6.

34. Wong A and Bishop W. Backwards compatible, multi-level regions-of-interest (ROI) image encryption architecture with biometric authentication. In: Proceedings of the second international conference on signal processing and multimedia applications (SIGMAP), Barcelona, 28–31 July 2007, pp.324–329. New York: ACM.

35. Parameshchahiri BD, Karappa R, Sojaudah KS, et al. Partial image encryption algorithm using pixel position manipulation technique: the smart copyback system. In: 2014 4th international conference on artificial intelligence with applications in engineering and technology, Kota Kinabalu, Malaysia, 3–5 December 2014, pp.177–181. New York: IEEE.

36. Panduranga HT and Naveenkumar SK. Selective image encryption for medical and satellite images. Int J Eng Technol 2013; 5(1): 115–121.

37. Abdmouleh MK, Khalfallah A and Bouhlel MS. A novel selective encryption scheme for medical images transmission based on JPEG compression algorithm. Procedia Comput Sci 2017; 112: 369–376.

38. Parameshchahiri BD, Panduranga HT and Naveenkumar SK. Partial encryption of medical images by dual DNA addition using DNA encoding. In: 2017 international conference on recent innovations in signal processing and embedded systems (RISE), Bhopal, India, 27–29 October 2017, pp.310–314. New York: IEEE.

39. Brightwell M and Smith H. Using datatype-preserving encryption to enhance data warehouse security. In: 20th national information systems security conference proceedings (NISSC), Baltimore, MD, 7–10 October 1997, pp.141–149. Gaithersburg, MD: NIST.

40. Black J and Rogaway P. Ciphers with arbitrary finite domains. In: Cryptographers’ track at the RSA conference, San Jose, CA, 13–17 February 2002, pp.114–130. Berlin: Springer.

41. Bellare M, Ristenpart T, Rogaway P, et al. Format-preserving encryption. In: International workshop on selected areas in cryptography, Calgary, AB, Canada, 13–14 August 2009, pp.295–312. Berlin: Springer.

42. Bellare M, Rogaway P and Spies T. The FFX mode of operation for format-preserving encryption. Gaithersburg, MD: NIST, 2010.

43. Vance J. VAES3 scheme for FFX: an addendum to the FFX mode of operation for format preserving encryption. Gaithersburg, MD: NIST, 2011.

44. Brier E, Peyrin T and Stern J. BPS: a format-preserving encryption proposal. Рекомендации по стандартизации, 2010. http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/bps/bps-spec.pdf

45. Dworkin MJ and Perlner RA. Analysis of VAES3 (FF2), 2015, https://eprint.iacr.org/2015/306.pdf

46. Dworkin M. Recommendation for block cipher modes of operation: methods for format-preserving encryption. NIST Special Publication, 800-38G, 2016, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf

47. Durak FB and Vaudenay S. Breaking the FF3 format-preserving encryption standard over small domains. In: Annual international cryptology conference, Santa Barbara, CA, 20–24 August 2017, pp.679–707. Cham: Springer.

48. Dworkin M. Recommendation for block cipher modes of operation: methods for format-preserving encryption. NIST Special Publication, 800-38G, Revision 1, 2019, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38Gr1-draft.pdf
49. He K, Gkioxari G, Dollár P, et al. Mask R-CNN. In: 
Proceedings of the IEEE international conference on computer vision, Venice, 22–29 October 2017, pp.2961–2969. New York: IEEE.

50. Su H, Wei S, Yan M, et al. Object detection and instance segmentation in remote sensing imagery based on precise mask R-CNN. In: IGARSS 2019: 2019 IEEE international geoscience and remote sensing symposium, Yokohama, Japan, 28 July–2 August 2019, pp.1454–1457. New York: IEEE.

51. Zhou Y, Panetta K, Cherukuri R, et al. Selective object encryption for privacy protection. In: Mobile multimedia/image processing, security, and applications 2009, vol. 7351, Orlando, FL, 13–17 April 2009, p.73510F. Bellingham, WA: International Society for Optics and Photonics.

52. Chung JY and Hong YS. Parallel image encryption schemes for security of ultra high resolution images. In: Proceedings of the Korean information science society conference, 2012, pp.274–276. Seoul, Korea: Korean Institute of Information Scientists and Engineers.

53. Pareek NK, Patidar V and Sud KK. Diffusion-substitution based gray image encryption scheme. Digital Signal Process 2013; 23(3): 894–901.

54. Chen G, Mao Y and Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Soliton Fract 2004; 21(3): 749–761.

55. Mao Y, Chen G and Lian S. A novel fast image encryption scheme based on 3D chaotic baker maps. Int J Bifurc Chaos 2004; 14(10): 3613–3624.

56. Bassham LE III, Rukhin AL, Soto J, et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Sp 800-22 rev. 1a, 2010, https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final