# MSAI: Masking Sensitive Area of Image on IoT Cameras

Jinqiang Liu[1], Yining Liu[1], Lei Cui[2], Shui Yu[3]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, China
[2] School of Information Technology, Deakin University, Australia
[3] School of Computer Science, University of Technology, Australia
jqliu000@gmail.com, ynliu@guet.edu.cn, cuil@deakin.edu.au, Shui.Yu@uts.edu.au

## Abstract

In smart cities, images captured by Internet of Things (IoT) cameras are transmitted to data center via intermediate nodes. The shared images can reveal much of sensitive information of users, which has caused increasing privacy concerns. Various encryption-based techniques have been developed for privacy preserving. However, they are not suitable for IoT cameras due to the high computation cost. In this paper, we propose a lightweight image privacy protection scheme considering personalized protection requirements of users. We first introduce an object detection algorithm to detect and pick up sensitive areas in images according to the specific requirements of users. Then, we propose a membrane-based method to mask the sensitive areas before uploading images to the data center. In particular, the masking operation does not require much computing resources on the used cloud platform, and the masking size of membrane can be dynamically adjusted. Our experiments on real-world datasets demonstrate the effectiveness and feasibility of the proposed scheme.

**Keywords:** IoT camera, Image privacy, Cloud platform, Masking operation, Membrane

## 1 Introduction

Internet of Things (IoT) cameras have been widely deployed in urban industries and traffic management systems, which accelerates the development of Smart City [1-3]. However, images collected by IoT cameras contain rich sensitive information of users, while the intermediate nodes or edge devices for transmitting images are normally untrusted [4-7]. In recent years, image privacy has caused increasing concerns.

The state-of-the-art image privacy protection methods mainly employ encryption algorithms. Early studies mainly employ AES and DES to encrypt the data. However, these algorithms are not suitable to encrypt image, because the amount of information contained in images are extremely [8]. To address this challenge, chaotic theory-based and permutation-based encryption algorithms have been investigated [9-12]. While the major disadvantage of these algorithms is inefficient and inflexible. Therefore, many researchers use visual-based encryption methods to protect image privacy. These methods mainly include masking, blurring, and pixilation [13].

Although many investigations have been conducted for protecting image privacy, the aforementioned algorithms cannot be directly used in real IoT-based applications. First, the video captured by IoT cameras is around 25 to 30 frames per second, and the size of the video captured in whole day is about 24G. If a vast majority of real-time images are encrypted frequently, the computation overhead is too high for IoT camera with limited resources. In addition, image privacy requirements are not uniform considering various policies in different regions. Taking the widely applied face Id as an example, some countries define the entire face as privacy area, while some others only regard eyes as the sensitive information. Traditional methods are not flexible as they cannot dynamically adapt to these customized protection requirements. As a result, novel image privacy protection scheme that are lightweight and personalized need to be developed in IoT camera-based applications.

In this paper, we propose a lightweight and flexible image privacy protect scheme in IoT environment. In order to meet the requirements of customized privacy protection, the proposed method can provide users with different area masking options such as face or license plate. In particular, we deployed an object detection algorithm YOLO v3 to detect and pick up the sensitive areas in images. In addition, we employ a masking strategy to protect the selected sensitive areas, which can reduce the computational requirements significantly. We conduct extensive experiments and the compared methods with existing image encryption algorithms. Experimental results demonstrate the effectiveness and efficiency of MSAI (Masking Sensitive Area of Image on IoT Camera). Specifically, the contributions of this

*Corresponding Author: Yining Liu; E-mail: ynliu@guet.edu.cn
DOI: 10.53106/160792642021122207009

paper are as follows.

(1) We propose an automatic method to mask sensitive areas in images according to personalized requirements of users. The masking operation is only executed to the sensitive area on frames containing sensitive information, which greatly reduces the computation cost of IoT camera.

(2) We employ a membrane-based masking algorithm to protect image privacy. The proposed method can change the size of masking dynamically according to the specific sensitive area defined by different users.

(3) Extensive experiments are conducted and the results demonstrate that our scheme consistently outperforms state-of-the-art methods in efficiency and flexibility.

The rest of this paper is organized as follows: Section 2 introduces the related works, preliminaries is introduced in Section 3, Section 4 introduces the system overview, Section 5 introduces the scheme our proposed. Section 6 presents the experimental results. Finally, the summary is summarized in Section 7.

## 2 Related Work

The intuitive idea to protect the image is encrypt image using traditional cryptographic tools, such as AES and DES. In 2013, Prerna Mahajan et al. [9] analyzed that these encryption methods are more suitable for text files with high-security level. However, it is not suitable for encrypting images. Since the amount of image data is very extremely, and traditional image encryption is inefficient in images [14].

Many researchers have proposed novel image encryption algorithms, for example, chaotic maps use the permutation, diffusion and other operations to disrupt the correlation of pixels in the image [15-16]. Chaotic maps are divided into one-dimensional chaotic maps and high-dimensional chaotic maps [17-18]. Advanced in unpredictability, ergodicity and sensitivity to initial conditions and parameters, chaotic maps are widely used in modern image encryption algorithms [19]. S. Behnia et al. [20] proved that traditional one-dimensional chaotic maps are insecure, many researchers have designed two-dimensional even high-dimensional chaotic map algorithms. For example, in 2018, K. Muhammad et al. [21] proposed a probabilistic monitoring framework for the IoT system, which uses a two-dimensional-Sine system to encrypt keyframes in surveillance video. In 2019, M. Alawida et al. [22] proposed an image encryption algorithm based on the perturbation of a hybrid chaotic system. Tang et al. [23] proposed a novel image encryption algorithm by jointly exploiting random overlapping block partition, double spiral scans, Henon chaotic map, and Lu chaotic map. In 2020, Zhang et al. [24] proposed a High-speed image encryption scheme based on multiple XORs. Although these chaotic maps algorithms really achieve the security requirements, they are not suitable for IoT cameras due to the complex. In addition, most of the chaotic map algorithms encrypt the entire image, although the security level of the image is improved, but lacks flexibility, which is not necessary for the actual IoT camera environment.

Compared with the high-security feature of chaotic maps, the image encryption algorithm based on permutation has attracted the researchers' attention, which is usually divided into several categories: pixel replacement [25], position replacement [26-27], and block replacement [28]. The pixel replacement algorithm and position replacement algorithm need more encryption time than the block replacement. In the block replacement algorithm, the smaller the block, the better the encryption effect. Although these image encryption algorithms based on permutation have faster encryption time, there is not security. These algorithms usually encrypt the entire image and lack flexibility [29].

Considering the security and flexibility of the image encryption algorithm, Lv et al. [30] proposed a variable membrane encryption system, in which the cloud platform generates a matrix, then the IoT camera masks the sensitive area using this matrix. Though Lv et al.'s scheme owns high flexibility and efficiency, cloud platform maybe colludes with malicious users to attack the system. In order to more accurately detect and pick the sensitive area, the neural network is a useful tool to ensure more flexible.

## 3 Preliminaries

In this section, Integer Vector Homomorphic Encryption (VHE) algorithm [31], the masking membrane algorithm and High-speed chaotic maps algorithm will be introduced. In this paper, the symbol we used is shown in Table 1.

**Table 1.** Symbol table

| symbols | significance |
|---|---|
| $S$ | Matrix generated by Entity A |
| $W$ | A large integer |
| $SA_r$ | The row of the sensitive area |
| $SA_c$ | The column of the sensitive area |
| $\tau$ | Pseudo-random sequence $\tau$ based on the Image size |
| $O$ | Original image |
| $k$ | Key of Image masking operation |
| $M$ | Masking membrane |
| $D$ | Cloud platform initializes a matrix |
| $O'$ | The sensitive area masking image |
| $O'_{SA}$ | The sensitive area of $O'$ |
| $A$ | Chaotic masking image |
| $A'$ | Masking recovery image |

### 3.1 Integer Vector Homomorphic Encryption Algorithm

VHE algorithm is an efficient encryption algorithm, this algorithm is widely used in cloud environments. The detail of the VHE algorithm are as follows.

#### 3.1.1 Key Generation

Entity A generates a matrix $S$ of size $m \times n$ and sends $S$ to entity B, where the elements of $S$ are integer.

#### 3.1.2 Encryption

**Step 1.** Entity A generates a large integer $w >> |S|$, where $|S| := \max_i \left\{ |S|_{ij} \right\}$, and sends $W$ to entity B.

**Step 2.** Entity A computes the the $w\beta$ with vector $\beta = (b_1, b_2, \ldots, b_m)^{\mathrm{T}}$, where $b_1, b_2, \ldots, b_m$ are integer.

**Step 3.** Entity A computes the vector $\alpha = (a_1, a_2, \ldots, a_n)^{\mathrm{T}}$ as ciphertext satisfying $S\alpha = w\beta + e$, where $a_1, a_2, \ldots, a_n$ are integer $e$ is a noisy vector as an error term with elements smaller than $\frac{w}{2}$, then sends $\alpha$ to entity B.

#### 3.1.3 Decryption

Entity B computes the $\lceil \frac{S\alpha}{w} \rfloor$ to recover the plaintext, where $\lceil \frac{S\alpha}{w} \rfloor$ is around $\frac{S\alpha}{w}$ to the nearest integer.

### 3.2 The Masking Membrane Algorithm

The masking membrane algorithm is a lightweight image encryption algorithm, which uses VHE algorithm to encrypt a matrix called the masking membrane. Only one multiplication operation is performed to ensure the lightweight when VHE algorithm is executed. The advantage of the masking membrane algorithm is lightweight and high flexibility. The detail of this algorithm are as follows.

#### 3.2.1 Masking

**Step 1.** IoT camera executes YOLO v3 [32] algorithm to find the location of the sensitive area and obtains the size $SA_r \times SA_c$ of the sensitive area in the image, where $SA_r$ is the row of the sensitive area, $SA_c$ is the column of the sensitive area.

**Step 2.** IoT camera sends $SA_r \times SA_c$ to the cloud platform, and requests a masking membrane with the size of $SA_r \times SA_c$.

**Step 3.** Cloud platform initializes a matrix $D$ of size $SA_r \times SA_c$, and the initial values of all elements are set to 1.

**Step 4.** Cloud platform generates $k$ as key, where $k$ is taken from finite field $GF(251)$, and sends $k$ to IoT camera.

**Step 5.** Cloud platform executes VHE algorithm for each element of matrix $D$ to obtain the masking membrane $M$.

**Step 6.** The IoT camera multiplies the elements in the sensitive area of the original image with the corresponding elements in the masking membrane $M$, and do nothing in other areas, and then sends the image of the sensitive area masked to the data center.

#### 3.2.2 Masking Recovery

**Step 1.** Data center finds the location of the sensitive area.

**Step 2.** Data center executes VHE algorithm to recover the masked sensitive area in image based on $k$, and do nothing in other areas.

### 3.3 High-speed Chaotic Maps Algorithm

High-speed chaotic maps is a block-based chaotic maps algorithm. The principle of the algorithm is shown in the Figure 1. The High-speed chaotic maps algorithm divides the image into several blocks and rearranges these blocks according to a pseudo-random sequence. This algorithm is a compromise between security and efficiency. Although the security of this algorithm is not better than other complex chaotic maps algorithms, the computational burden of this algorithm is small, so it is suitable for image privacy protection in the IoT camera environment.

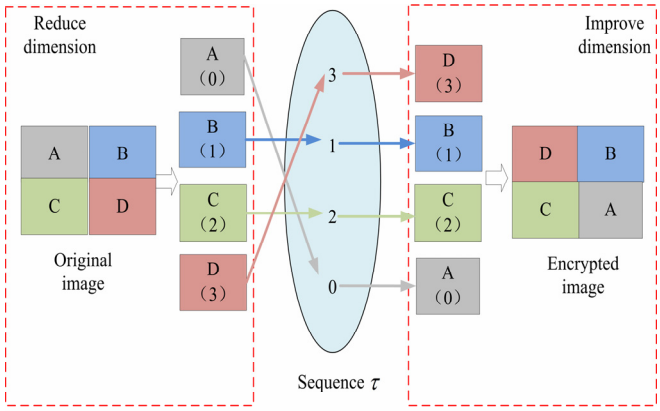



**Figure 1.** High-speed chaos mapping algorithm

#### 3.3.1 Encryption

**Step 1.** Data center generates a pseudo-random sequence $\tau$ based on the image size and sends this sequence $\tau$ to IoT camera, where $\tau_i \in \{0, \ldots, b-1\}$, $b$

is the number of image block and the elements value of $\tau_i$ are not repeated.

**Step 2.** The IoT camera divides the image into $b$ blocks, and re-arranges the image blocks using the sequence $\tau$ sent by data center.

**Step 3.** IoT camera sends the chaotic image to data center.

#### 3.3.2 Decryption

The data center rearranges the chaotic image according to the sequence $\tau$ to restore the original image.

## 4 The Proposed MSAI Scheme

### 4.1 System Overview

#### 4.1.1 Customizable Requirements

MSAI must ensure:

(1) Accurate identification: Different countries and regions have different laws and regulations on sensitive information in images. IoT camera accurately identifies sensitive areas in the image, according to different image privacy regulations in various countries and regions.

(2) Customized privacy protection: Protect sensitive areas of different sizes at different locations in the image.

#### 4.1.2 Security Requirements

MSAI must ensure:

(1) Data Confidentiality: Scheme can prevent resist collusion and active attacks. If the key of the masking membrane is leaked after the cloud platform is attacked, the masked image cannot be restored.

(2) Privacy Confidentiality: Intermediate nodes is not able to infer privacy information about the sensitive area of image.

#### 4.1.3 Efficiency Requirements

MSAI must ensure:

Lightweight: Due to the limited resource of IoT camera [33] and the massive real-time images, the computation executed on IoT camera should be lightweight, otherwise, the reliability and sustainability are difficult to be guaranteed.

### 4.2 Scheme Proposed

In order to protect the image privacy efficiently in daily life, MSAI: Masking Sensitive Area of Image on IoT cameras is proposed, the system model is shown in Figure 2, which includes two parts:

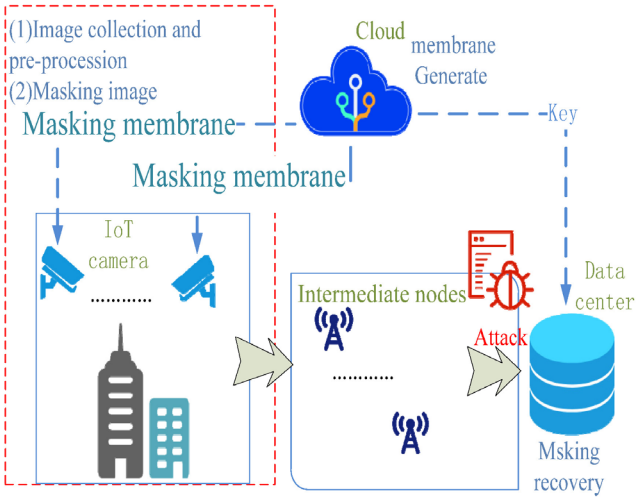



**Figure 2.** The MSAI scheme

(1) Image collection and pre-processing: IoT camera collects images and uses the object detection algorithm YOLO v3 to identify the coordinates and size of sensitive information areas in the image. The reason for choosing the YOLO v3 algorithm is that this algorithm is faster and more accurately than others object detection algorithm.

(2) Image masking operation: Cloud platform generates a masking membrane uses the size of the sensitive area obtained by the image pre-processing part, and sends the masking membrane to the IoT camera. The IoT camera masks the sensitive area, then the masked image is encrypted with High-speed chaotic maps.

In the following subsection, image collection and pre-processing and the image masking operation algorithm is presented in detail.

### 4.3 Image Collection and Pre-processing

The image collection and preprocessing part of MSAI is completed by the IoT camera. The details are as follows:

**Step 1.** Users select the privacy options provided by the scheme, such as covering the entire face or eyes in image, whether to cover the license plate, etc.

**Step 2.** The IoT camera collects images and uses the YOLO v3 algorithm to identify and obtain the sensitive area specified by the user.

### 4.4 Image Masking Operation

The image masking operation in MSAI consists of two parts: masking sensitive area of image and High-speed Chaotic maps algorithm, the details are as follows.

#### 4.4.1 Mask Sensitive Areas of the Image

**Step 1.** IoT camera executes YOLO v3 algorithm to find the sensitive area of the original image $O$. The size $SA_r \times SA_c$ of the sensitive area obtained by the IoT

camera is sent to cloud platform to request a masking membrane. The sensitive area $O_{SA}$ of original image is shown in Equation (1).

$$O_{SA}=\begin{bmatrix} o_{1,1} & o_{1,2} & \cdots & o_{1,SA_c} \\ o_{2,1} & o_{2,2} & \cdots & o_{2,SA_c} \\ \vdots & \ddots & \cdots & \vdots \\ o_{SA_r,1} & o_{SA_r,2} & \cdots & o_{SA_r,SA_c} \end{bmatrix} \quad (1)$$

**Step 2.** Cloud platform generates key $k$ and sends $k$ to IoT camera, where the $k$ is taken form finite field $GF(251)$.

$$M=\begin{bmatrix} \lambda_{1,1} & \lambda_{1,2} & \cdots & \lambda_{1,SA_c} \\ \lambda_{2,1} & \lambda_{2,2} & \cdots & \lambda_{2,SA_c} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{SA_r,1} & \lambda_{SA_r,2} & \cdots & \lambda_{SA_r,SA_c} \end{bmatrix} \quad (2)$$

**Step 3.** Cloud platform executes algorithm 1 to generate masking membrane and sends this masking membrane to IoT camera, and then cloud platform sends the key $k$ to the data center.

**Algorithm 1.** Masking membrane generation

Input: The size $SA_r \times SA_c$ of sensitive area, $k$, VHE encryption algorithm.

Output: Masking membrane $M$.

**Step 1.** Cloud platform initializes a matrix $D$ of size $SA_r \times SA_c$, and the initial values of all elements are set to 1.

**Step 2.** Cloud platform executes the VHE algorithm to encrypt each element of matrix $D$ to obtain the masking membrane $M$. The masking membrane is shown in Equation (2). In the matrix $M$, the value of $\lambda_{1,1},\lambda_{1,2}\ldots,\lambda_{m,n}$ is $k^{-1}\times(w+e_{1,1}),k^{-1}\times(w+e_{1,t}),\ldots,k^{-1}\times(w+e_{SA_r,SA_c})$. $w$ is taken from finite field $GF(251)$, $e$ is an error term. Then cloud platform sends the matrix $M$ to the IoT camera.

**Step 4.** The IoT camera finds the position of the sensitive area and masks this area, and do nothing in other areas. The sensitive area masking image $O'$ is obtained, the sensitive area $O'_{SA}$ has masked as shown as Equation (3).

$$O'_{SA}=\begin{bmatrix} o_{1,1}\times\lambda_{1,1} & o_{1,2}\times\lambda_{1,2} & \cdots & o_{1,SA_c}\times\lambda_{1,SA_c} \\ o_{2,1}\times\lambda_{2,1} & o_{2,2}\times\lambda_{2,2} & \cdots & o_{2,SA_c}\times\lambda_{2,SA_c} \\ \vdots & \vdots & \ddots & \vdots \\ o_{SA_r,1}\times\lambda_{SA_r,1} & o_{SA_r,2}\times\lambda_{SA_r,2} & \cdots & o_{SA_r,SA_c}\times\lambda_{SA_r,SA_c} \end{bmatrix} \quad (3)$$

### 4.4.2 Chaotic Sensitive Area Masking Image

**Step 1.** The data center generates a pseudo-random sequence $\tau$, based on the size of the original image $O$, where $\tau_i\in\{0,\ldots,b-1\}$, $b$ is the number of blocks divided in image. And then the data center sends the sequence $\tau$ to the IoT camera.

**Step 2.** The IoT camera executes the algorithm 2 based on the sequence $\tau$ and image $O'$ to obtain a chaotic image $A$ of the sensitive area masked.

**Algorithm 2.** High-speed Chaos Mapping algorithm

Input: Sequence $\tau$, image $O'$.

Output: Chaotic masking image $A$.

**Step 1.** IoT camera divides the image $O'$ into $b$ blocks.

**Step 2.** IoT camera uses the row-first method to convert a 2-D matrix into a 1-D cell array, the elements of the cell array are pixel blocks.

**Step 3.** The elements in the cell array are rearranged according to the element values in the sequence $\tau$.

**Step 4.** IoT camera converted 1-D cell array into a 2-D matrix to obtain a chaotic masking image $A$.

$$A'_{SA}=\begin{bmatrix} \left\lceil\frac{K\times O'_{1,1}}{w}\right\rfloor_{251} & \left\lceil\frac{K\times O'_{1,2}}{w}\right\rfloor_{251} & \cdots & \left\lceil\frac{K\times O'_{1,SA_c}}{w}\right\rfloor_{251} \\ \left\lceil\frac{K\times O'_{2,1}}{w}\right\rfloor_{251} & \left\lceil\frac{K\times O'_{2,2}}{w}\right\rfloor_{251} & \cdots & \left\lceil\frac{K\times O'_{2,SA_c}}{w}\right\rfloor_{251} \\ \vdots & \vdots & \ddots & \vdots \\ \left\lceil\frac{K\times O'_{SA_r,1}}{w}\right\rfloor_{251} & \left\lceil\frac{K\times O'_{SA_r,1}}{w}\right\rfloor_{251} & \cdots & \left\lceil\frac{K\times O'_{SA_r,SA_c}}{w}\right\rfloor_{251} \end{bmatrix} \quad (4)$$

### 4.4.3 Masking Recovery

According to the key $k$ sent by the cloud platform and the sequence $\tau$, the algorithm 3 is executed by data center to obtain the image $A'$ of masking recovery.

**Algorithm 3.** Masking recovery

Input: Image $A$, sequence $\tau$, $k$.

Output: Image $A'$ of masking recovery.

**Step 1.** Data center divides the image $A$ into $b$ blocks.

**Step 2**. Data center uses the row-first method to convert a 2-D matrix into a 1-D cell array, the elements of the cell array are pixel blocks.

**Step 3.** Data center rearranges the elements in the cell array according to the sequence $\tau$, and restores the image $O'$.

**Step 4.** Data center finds the location of the sensitive area in the image $O'_{SA}$.

**Step 5.** The masking recover operation will be performed by data center according to the key $k$. The

image $A'$ of masking recovery is obtained. The masking recovery perform in the sensitive area $A'_{SA}$ is shown in Equation (4), where $\lceil \frac{S\alpha}{w} \rfloor_{251}$ is around $\lceil \frac{S\alpha}{w} \rfloor_{251}$ to the nearest integer mod 251.

## 5 Experiment Results

### 5.1 Simulation Results

MSAI are implemented on Python and executed on Windows 10 with 2.3 GHz processor. In the experimental environment, the identified sensitive areas are license plate, fingerprint and face in images by YOLO v3 algorithm. The size of the block in the High-speed chaotic maps algorithm is $8\times8$. Image size and size of the sensitive area identified with YOLO v3 is shown in Table 2. We select three sets of images as examples of the algorithm, as shown in Figure 3.

**Table 2.** The of the original and the sensitive area

| image | The size of original image | The size of sensitive area |
|---|---|---|
| car | 256×256 | 30×17 |
| finger | 256×256 | 58×62 |
| Lena | 256×256 | 71×63 |


(a) Original image (b) Final masking image (c) Sensitive area masking image (d) Recover masking image

**Figure 3.**

#### 5.1.1 Histogram Analysis

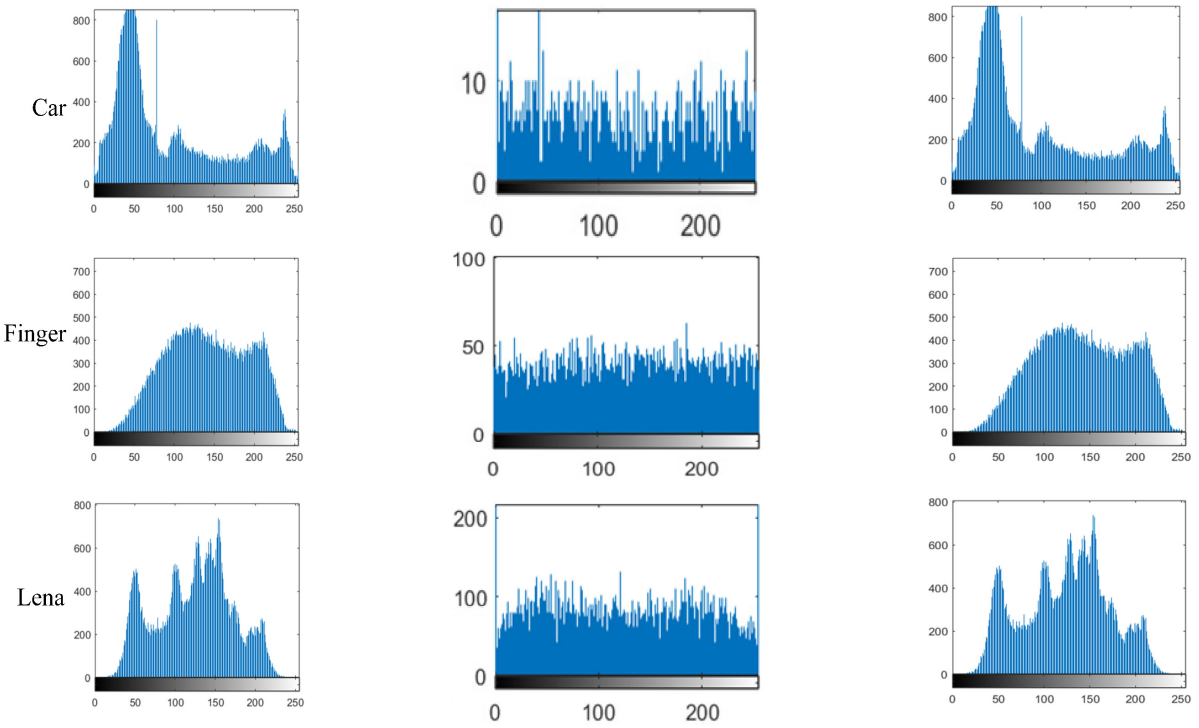
The digital histogram can intuitively evaluate the pixel tone distribution of the image. The x-axis and y-axis of the histogram represent the number of pixels of the corresponding intensity level. Figure 4(a) to Figure 4(c) shows the digital histogram of the original image, masking sensitive area and masking recover image about the three experimental images. As shown in Figure 4, the histogram of the original image is not uniform. The distribution of the masked sensitive area map is almost uniform, so the sensitive area is chaotic after a masking operation.

#### 5.1.2 NPCR and UACI

The number of pixels change rate (NPCR) represents the ratio of different gray values of different encrypted images at the same location. We use the two parameters number of changing pixel rate (NPCR) and unified averaged changed intensity (UACI) to verify the strength of our proposed image masking operation against differential attacks. NPCR represents the ratio of different gray values of different encrypted images at the same location. UACI represents the average change density between different encrypted images. NPCR and UACI can be described as Equations (5) and (6). Table 3 gives values of PNCR and UACI of experimental images.

$$NPCR(I_1,I_2)=\frac{1}{W\times H}\sum_{i=1}^{W}\sum_{j=1}^{H}D(i,j) \tag{5}$$

$$UACI=\frac{1}{W\times H}\sum_{i=1}^{W}\sum_{j=1}^{H}\frac{\left|I_1(i,j)-I_2(i,j)\right|}{255}$$
$$where\ D(i,j)=\begin{cases}0, & if\ I_1(i,j)=I_2(i,j)\\ 1, & if\ I_1(i,j)\neq I_2(i,j)\end{cases} \tag{6}$$




(a) Original image (b) Final masking image (c) Recover masking image

**Figure 4.**

**Table 3.** NPRC and UACI value between original image and masking recovery image

| image | NPCR | UACI |
|---|---|---|
| car | 0.9961 | 0.3346 |
| finger | 0.9981 | 0.3318 |
| Lena | 0.7871 | 0.3094 |

#### 5.1.3 MSE and PSNR

Mean square error (MSE) can be used to describe the difference in quality between the original image and the recover masking image. MSE can be described as Equation (7).

We use Peak Signal to Noise Ratio (PSNR) to verify the quality of the masking recover image, which is show in Equation (8). The larger the value of PSNR, the higher the quality of the masking recover image. After calculation, $PSNR$=+∞ in the above three examples. This result proves that the quality of the masking recover image is the same as the quality of the original image.

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{W} \sum_{j=1}^{H} \left( X(i,j) - Y(i,j) \right)^2 \quad (7)$$

$$PSNR = 10 \times \log_{10} \left( \frac{\left(2^n - 1\right)^2}{MSE} \right) \quad (8)$$

### 5.2 Security Analysis

We employed only existing, peer-reviewed cryptographic schemes and discussed the security of the scheme, we corroborate these arguments with a brief summary of the security analysis.

(1) Data confidentiality: MSAI can prevent resist collusion and active attacks. Collusion attack: Assume that the cloud platform collides with malicious users, the malicious user can get key $k$, but malicious user cannot get sequence S. the image that has been masked cannot be restored. Active attacks: The communication channel between the cloud platform and the IoT camera is attacked, malicious user cannot get the key $k$. The image that has been masked cannot be restored.

(2) Privacy confidentiality: Experiments show that the masked image is chaotic, so the confidentiality of the data can be guaranteed.

### 5.3 Efficiency Analysis

We select the three images shown in Figure 3(a) as an example to calculate average time for one hundred masking operations. The time compared with other encryption algorithms is shown in Figure 5. Through experimental verification, the efficiency of the masking operation is better than compared algorithms.

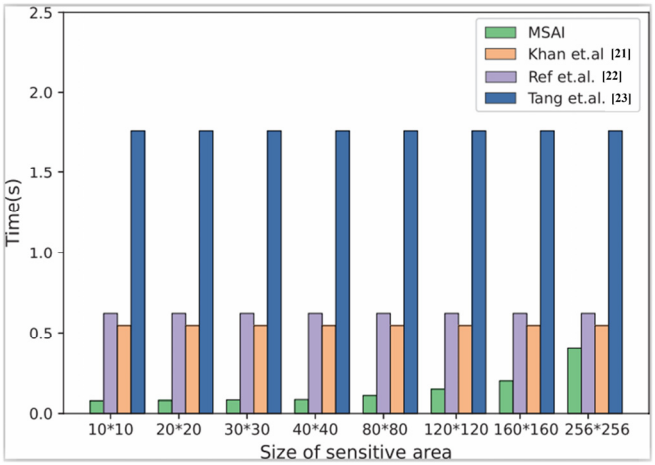



**Figure 5.** Time comparison between different algorithm

## 6 Summary

In this paper, we introduce the MSAI, a novel scheme for image privacy protection on IoT cameras. In particular, MSAI adopts an object detection algorithm to identify sensitive areas in images, and employs an improved masking algorithm to mask these areas according to the requirements of users. In order to make full use of computing resources in IoT environment, membranes generation operation that consumes a lot of computing resources is executed by cloud platform, while IoT cameras only needs to perform the masking operation. We have conducted extensive experiments in a real IoT environment. The experimental results demonstrate that MSAI has better performance in terms of efficiency, security and flexibility. In the future, we will further study how to select sensitive areas of images automated and provide more flexible and lightweight privacy protection services.

## Acknowledgments

This work was supported by Natural Science Foundation of China under Grants 62072133 and 61662016, Key projects of Guangxi Natural Science Foundation under Grant 2018GXNSFDA281040.

## References

[1] C. Perera, C. H. Liu, S. Jayawardena, The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey, *IEEE Transactions on Emerging Topics in Computing*, Vol. 3, No. 4, pp. 585-598, December, 2015.

[2] A Gaur, B Scotney, Gerard Parr, S. McClean, Smart city architecture and its applications Based on IoT, *Procedia Computer Science*, Vol. 52, pp. 1089-1094, 2015.

[3] J. Liu, J. Yu, S. Shen, Energy-efficient two-layer cooperative defense scheme to secure sensor-clouds, *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 2, pp. 408-420, February, 2018.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of Things for Smart Cities, *IEEE Internet of Things Journal*, Vol. 1, No 1, pp. 22-32, February, 2014.

[5] Q. Li, Z. Zheng, F. Wu, G. Chen, Generative Adversarial Networks-based Privacy-Preserving 3D Reconstruction, *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, Hang Zhou, China, 2020, pp. 1-10.

[6] J. Liu, X. Wang, S. Shen, G. Yue, S. Yu, M. Li, A Bayesian Q-Learning Game for Dependable Task Offloading Against DDoS Attacks in Sensor Edge Cloud, *IEEE Internet of Things Journal*, Vol. 8, No. 9, pp. 7546-7561, May, 2021.

[7] S. Shen, H. Li, R. Han, A. V. Vasilakos, Y. Wang, Q. Cao, Differential game-based strategies for preventing malware propagation in wireless sensor networks, *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 11, pp. 1962-1973, November, 2014.

[8] A. Kaur, G. Singh, A Random Selective Block Encryption Technique for Secure Image Cryptography Using Blowfish Algorithm, *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, 2018, pp. 1290-1293.

[9] P. Mahajan, A. Sachdeva, A study of encryption algorithms AES, DES and RSA for security, *Global journal of computer science and technology*, Vol. 13, No. 15, pp. 1-9, 2013.

[10] Z. Lin, S. Yu, J. Lü, S. Cai, G. Chen, Design and ARM-Embedded Implementation of a Chaotic Map-Based Real-Time Secure Video Communication System, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 25, No. 7, pp. 1203-1216, July, 2015.

[11] C. Li, B. Feng, S. Li, J. Kurths, G. Chen, Dynamic Analysis of Digital Chaotic Maps via State-Mapping Networks, *IEEE Transactions on Circuits and Systems*, Vol. 66, No. 6, pp. 2322-2335, June, 2019.

[12] Z. Wu, S. Shen, H. Zhou, H. Li, C. Lu, D. Zou, An effective approach for the protection of user commodity viewing privacy in e-commerce website, *Knowledge-Based Systems*, Vol. 220, Article No. 106952, May, 2021.

[13] K. Lander, V. Bruce, H. Hill, Evaluating the effectiveness of pixelation and blurring on masking the identity of familiar faces, *Applied Cognitive Psychology*, Vol. 15, No. 1, pp. 101-116, January/ February, 2001.

[14] X. Wang, J. Ma, X. Liu, Y. Miao, Search in My Way: Practical Outsourced Image Retrieval Framework Supporting Unshared Key, *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Paris, France, 2019, pp. 2485-2493.

[15] Z. Hua, B. Zhou, Y. Zhou, Sine Chaotification Model for Enhancing Chaos and Its Hardware Implementation, *IEEE Transactions on Industrial Electronics*, Vol. 66, No. 2, pp. 1273-1284, February, 2019.

[16] S. Y. D. Nezhad, N. Safdarian, S. A. H. Zadeh, New method for fingerprint images encryption using DNA sequence and chaotic tent map, *Optik*, Vol. 224, Article No. 165661,

December, 2020.

[17] J. Chen, L. Chen, Y. Zhou, Cryptanalysis of Image Ciphers with Permutation-Substitution Network and Chaos, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 6, pp. 2494-2508, June, 2021.

[18] A. Mansouri, X. Wang, A novel one-dimensional chaotic map generator and its application in a new index representation-based image encryption scheme, *Information Sciences*, Vol. 563, pp. 91-110, July, 2021.

[19] X. Gao, M. Cheng, S. Li, D. Zeng, Unveil the Time Delay Signature in Delayed Chaotic Communication System via CNN, *IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1-6.

[20] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, A. Akhavan, A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps, *Physics Letters A*, Vol. 366, No. 4-5, pp. 391-396, July, 2007.

[21] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, S. W. Baik, Secure surveillance framework for IoT systems using probabilistic image encryption, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 8, pp. 3679-3689, August, 2018.

[22] M. Alawida, A. Samsudin, J. S. Teh, R. S. Alkhawaldeh, A new hybrid digital chaotic system with applications in image encryption, *Signal Processing*, Vol. 160, pp. 45-58, July, 2019.

[23] Z. Tang, Y. Yang, S. Xu, C. Yu, X. Zhang, Image encryption with double spiral scans and chaotic maps, *Security and Communication Networks*, Vol. 2019, pp. 1-15, January, 2019.

[24] Y. Zhang, The fast image encryption algorithm based on lifting scheme and chaos, *Information Sciences*, Vol. 520, pp. 177-194, May, 2020.

[25] T. Sivakumar, R. Venkatesan, A novel Image encryption method with Z-Order curve and random number, *International Journal of Computer Applications*, Vol. 103, No. 12, pp. 17-25, October, 2014.

[26] C. Fu, B. Lin, Y. Miao, X. Liu, J. Chen, A novel chaos-based bit-level permutation scheme for digital image encryption, *Optics Communications*, Vol. 284, No. 23, pp. 5415-5423, November, 2011.

[27] S. V. Sathyanarayana, M. A. Kumar, K. N. Bhat, Symmetric key Image encryption scheme with key sequences derived from random sequence of cyclic elliptic curve points, *International Journal of Network Security*, Vol. 12, No. 3, pp. 137-150, May, 2011.

[28] S. Rakesh, A. K. Ajitkumar, B. C. Shadakshari, B. Annappa, Image encryption using block based uniform scrambling and chaotic logistic mapping, *International Journal on Cryptography and Information Security*, Vol. 2, No. 1, pp. 49-57, March, 2012.

[29] A. Jolfaei, X. Wu, V. Muthukkumarasamy, On the Security of Permutation-Only Image Encryption Schemes, *IEEE Transactions on Information Forensics and Security*, Vol. 11, No. 2, pp. 235-246, February, 2016.

[30] S. Lv, Y. Liu, J. Sun, IMES: An automatically scalable invisible membrane image encryption for privacy protection on IoT sensors, *cyberspace safety and security, 11th International Symposium, CSS 2019*, Guangzhou, China, 2019, pp. 265-273.

[31] H. Zhou, G. Wornell, Efficient homomorphic encryption on integer vectors and its applications, *Information Theory and Applications Workshop (ITA)*, San Diego, CA, USA, 2014, pp. 1-9.

[32] J. Redmon, A. Farhadi, YOLO v3: An Incermental Improvement, *arXive*, April, 2018.

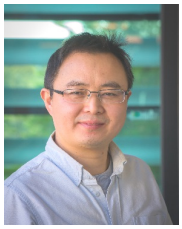
[33] M. Chehab, A. Mourad, LP-SBA-XACML: Lightweight Semantics Based Scheme Enabling Intelligent Behavior-Aware Privacy for IoT, *IEEE Transactions on Dependable and Secure Computing*, pp. 1-15, June, 2020. DOI:10.1109/TDSC.2020.2999866

## Biographies

**Jinqiang Liu** received the B.E. degree in software engineering from Tianjin Normal University, Tianjin, China, in 2019. He is currently a graduate of Guilin University of Electronic Technology, China. His research interests include image privacy, information security.

**Yining Liu** received B.S. degree from Information Engineering University, Zhengzhou, China, Ph.D. degree in mathematics from Hubei University, Wuhan. He is currently a professor with school of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include the information security protocol and data privacy.

**Lei Cui** has received his B.S. degree of Electrical and Power Engineering in 2010 and Ph.D. degree of Electronic Science and Technology in 2019 from Taiyuan University of Technology. His research interests focus on dealing with security and privacy issues in IoT, social networks, and machine learning.



**Shui Yu** obtained his Ph.D. from Deakin University, Australia. He currently is a Professor of School of Computer Science, University of Technology Sydney, Australia. His research interest includes Big Data, Security and Privacy,. He is a Senior Member of IEEE, and a Distinguished Lecturer of IEEE Communications Society.