Two-Factor Fuzzy Commitment for Unmanned IoT Devices Security

Dooho Choi, *Member, IEEE*, Seung-Hyun Seo, *Member, IEEE*, Yoon-Seok Oh, *Student Member, IEEE*, and Yousung Kang, *Member, IEEE*

***Abstract*—To create an environment for IoT devices, securely, it is necessary to establish a cryptographic key for those devices. Conventionally, this key has been stored on the actual device, but this leaves the key vulnerable to physical attacks in the IoT environment. To solve this problem, several research studies have been conducted on how best to conceal the cryptographic key. Recently, these studies have most often focused on generating the key dynamically from noisy data using a fuzzy extractor or providing secure storage using a fuzzy commitment. Thus, far, all of these studies use only one type of noisy source data, such as biometric data or physical unclonable function (PUF). However, since most IoT devices are operated in unmanned environments, where biometric data is unavailable, the method using biometric data cannot be utilized for unmanned IoT devices. Although the method using PUF is applied to these unmanned devices, these are still vulnerable against physical attacks including unintended move or theft. In this paper, we present a novel way to use the fuzzy commitment on such devices, called two-factor fuzzy commitment scheme. The proposed method utilizes two noisy factors from the inside and outside of the IoT device. Therefore, although an attacker acquiring the IoT device can access the internal noisy source, the attacker cannot extract the right key from that information only. We also give a prototype implementation for ensuring the feasibility of our two-factor fuzzy commitment concept by utilizing the image data and PUF data for two noisy factors.**

***Index Terms*—Error correcting codes, fuzzy commitment, noisy source data, physical unclonable function (PUF).**

I. Introduction

DUE TO an explosive increase in the use of IoT devices, the possibility of malicious attacks has also increased drastically. To resist such attacks, cryptographic functions are being used more often in IoT devices. Vital to these cryptographic functions are the generation and use of cryptographic keys. Therefore, protecting these keys has become an essential aspect of the overall security of IoT devices.

Manuscript received February 14, 2018; revised April 29, 2018; accepted May 5, 2018. Date of publication May 17, 2018; date of current version February 25, 2019. This work was supported in part by the Institute for Information and Communications Technology Promotion through the Study on Secure Key Hiding Technology for IoT Devices (KeyHAS Project) under Grant 2016-0-00399 and in part by the National Research Foundation of Korea through the Korea Government under Grant 2018R1A2B6006903. *(Corresponding author: Seung-Hyun Seo.)*

D. Choi and Y. Kang are with the Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea (e-mail: dhchoi@etri.re.kr; youskang@etri.re.kr).

S.-H. Seo and Y.-S. Oh are with the Division of Electrical Engineering, Hanyang University, ERICA Campus, Ansan 15588, South Korea (e-mail: seosh77@hanyang.ac.kr; ashbringer@hanyang.ac.kr).

Digital Object Identifier 10.1109/JIOT.2018.2837751

There are two competing methods for protecting cryptographic keys. One involves hiding the key in a secure part of the device, such as hardware security module (HSM) or trusted platform module (TPM) [1]–[4]. The other method requires the generation of the key dynamically, without storing it on the device [5]–[8]. Originally, this second method consisted of a fuzzy commitment scheme which combined error correcting codes and cryptography [5]. This provided a general method, called the fuzzy extractor [6]–[8], which generated the cryptographic keys using noisy data collected via biometrics. More recently Juels has proposed a fuzzy vault scheme which can generate a cryptographic key from biometrics, typically, fingerprints. After these works, the fuzzy extractor concept was applied to several biometrics data, such as fingerprint [8], [9], iris [10]–[12], face [13], and palmprint [14]. Furthermore, there was an attempt to generate the cryptographic key using physiological signals [15]. To use the biometric data as the noisy source data of the fuzzy extractor concept is very adequate in the case of the smart phone, which has almost seamless interaction between human and device. It is, however, a very limited approach to cover various IoT applications, especially, unmanned environments, such as surveillance/monitoring services.

The other main noisy source data of the fuzzy extractor is the output value of physical unclonable function (PUF). PUF is a physical functionality, which can produce some value hard to predict and duplicate. PUF functionality can be implemented with various physical characteristics (see [16] for the silicon-based PUF survey). PUF basically can provide an anti-counterfeit property for a given device at the hardware level, and also be used to generate a cryptographic key, dynamically. But since most PUF functionalities have some noise, it is necessary to ensure the robustness of PUFs output in order to use the cryptographic key, and so several papers employed and implemented the fuzzy extractor or a similar concept to secure the cryptographic key from PUF [17]–[22].

One major threat of unmanned IoT application environments, for example, surveillance/monitoring with IP cameras and/or small sensors, is that an attacker steals the device, physically, and tries to acquire critical information in the device (e.g., collected data, access credentials to the IoT network, or a group key for communicating with each other, etc.). In order to respond to this kind of threat, secure hardware components, such as HSM/TPM or PUF-based fuzzy extractors can be applied. However, the use of a secure hardware component has a little difficulty for the limited resource constraint

of the IoT device, so low cost PUF-based fuzzy extractors can be good candidates against this threat. In [29], the authors proposed DRAM-based PUF functionality for a commodity device, and CMOS image sensor-based PUF proposed for sensor-level authentication at [30]. Recently, an LWE-based fuzzy extractor using SRAM-based PUF was proposed in [31] for the use of IoT devices. Although the low cost PUF-based approach can be applied in the IoT device, one additional requirement, which is physical or logical tamper resistance, should be guaranteed, because it is possible to obtain the critical information in the device if the attacker who acquires the device can access the low cost PUF functionality. However, utilizing the tamper resistant technique raises the cost of the device significantly.

A. Our Contributions

To counter these drawbacks, we propose a novel concept, which is called two-factor fuzzy commitment. The two-factor fuzzy commitment scheme uses two noisy sources from the inside and outside of the IoT device, unlike the conventional fuzzy extractor concept. Essentially, if we apply this mechanism to the IoT device, we can lock the device in its legitimate operating environment, because our proposed mechanism cannot extract the right key in a place which is not legitimate, even if the attacker can access the internal noisy data of the device. We will show that the security of the two-factor fuzzy commitment with respect to concealing and binding properties fully depends on the randomness of the internal noisy source.

To ensure that the two-factor fuzzy commitment scheme can be applied well in the scenario providing a secure IoT device for surveillance purposes, we implemented our mechanism into an IoT surveillance camera prototype based on the Raspberry Pi with a camera sensor and PUF chip. We used the PUF as the internal noisy source and the image from the camera sensor as the external noisy source. In order to use that image data as external noisy source data, we binarize that image data by using an image binarization technique, such as Otsu [40]. And we use the binary Bose Chaudhuri Hocquenghem (BCH) codes [41], which form a large class of powerful random error-correcting cyclic codes, to correct errors of the internal/external noisy sources. Moreover, we also will give the experiment results under various environments, where the surveillance camera is operating (e.g., indoor/outdoor, brightness/darkness, object interruption, different places, and similar places) to check the key recovery rate (KRR) for the correct key.

B. Organization of This Paper

The remainder of this paper consists as follows. In Section II, we introduce some necessary preliminary information. In Section III, we propose our two-factor fuzzy commitment concept and discuss the security aspects. In Section IV, we present a scenario to apply our scheme to the IoT surveillance camera, and give the experiment results to prove the feasibility of the two-factor fuzzy commitment in Section V. The conclusions are described in Section VI.

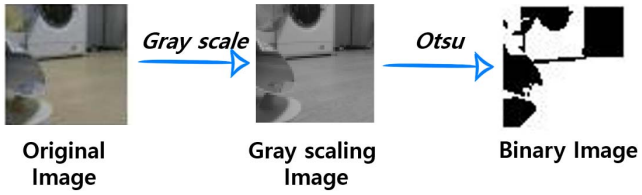


Fig. 1. Image conversion process using Otsu method.

II. Preliminaries

A. Physical Unclonable Function

A PUF is a functionality that provides unique device identity based on physical characteristics, such as physical variations occurring naturally in the semiconductor fabrication. PUF is defined in the following two classes [16], [32].

1) *Weak PUF:* A weak PUF is typically only used to derive a secret key. It is assumed that the challenge may be limited, and that the response space is not revealed.
2) *Strong PUF:* A strong PUF assumes that the cloning is physically impossible and it is impossible to collect the entire set of challenge/response pairs within a reasonable time. And also, it is difficult to predict the response to any challenge.

B. Image Binarization

As an image binarization technique is the process of converting a pixel image to a binary image, it converts the image of up to 256 gray levels to a black and white image. The simplest way to use image binarization is to choose a threshold value, and classify all pixels with values above this threshold as white, and all other pixels as black. Generally, finding one threshold compatible to the entire image is very difficult or in many cases even impossible. Thus, adaptive image binarization is needed, where an optimal threshold is chosen for each image area. So far, several researchers have proposed various thresholding techniques for binarization [40], [42]–[46]. Among them, the Otsu method is the most successful global thresholding method proposed by Nobuyuki Otsu. It is a technique to automatically perform histogram shape-based image thresholding for the reduction of a gray-level image to a binary image. The algorithm assumes that the image for thresholding contains two classes of pixels (e.g., foreground and background) and then calculates the optimum threshold separating those two classes so that their combined spread (intraclass variance) is minimal. The goal of setting the optimal threshold is to minimize the variance of the values in the image, and as a result, we can expect to get an accurate binary image. Fig. 1 shows the process of applying the Otsu method to an image.

C. Binary BCH Codes

The BCH codes [41] are multiple error correcting codes and a generalization of the Hamming codes. These are the possible BCH codes for $m \geq 3$ and $t < 2^{m-1}$.

1) *Block Length:* $n = 2^m - 1$.
2) *Parity Check Bits:* $n - k \leq mt$.
3) *Minimum Distance:* $d \geq 2t + 1$.

The BCH codes exist for length n and with at most mt check bits, it can correct any set of t independent errors within the block of n bits. m and t are arbitrary positive integers. Since the BCH code is a cyclic code generated on the basis of the generator polynomial, generation parameters can be recovered by using the characteristic of the cyclic code. The codewords are formed by taking the remainder after dividing a polynomial representing information bits by a generator polynomial. The generator polynomial is selected to give the code its characteristics. All codewords are multiples of the generator polynomial. Let α be a primitive element of the finite field $GF(2^m)$, then the t-error-correcting BCH code may be described as the set of all polynomials such that $\{a(x)\}$ over $GF(2)$ of degree $n-1$ or less, such that

$$a\left(\alpha^i\right) = 0, \qquad i = 1, 3, 5, \ldots, 2t-1 \tag{1}$$

where $a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$ and $a_i = 0, 1 (i = 0, 1, 2, \ldots, n-1)$. It is known in coding theory that these polynomials consist of all the multiples of a single polynomial $g(x)$, known as the generator polynomial of the code. $g(x)$ also satisfies the equations

$$g\left(\alpha^i\right) \equiv 0, \qquad i = 1, 3, 5, \ldots, 2t-1. \tag{2}$$

When the coefficient of $a(x)$ is viewed as a vector, one may equivalently define the code as the set of all n-tuples orthogonal to the parity check matrix

$$M = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^3 & \cdots & \alpha^{2t-1} \\ \alpha^2 & \alpha^6 & \cdots & \alpha^{2(2t-1)} \\ \alpha^3 & \alpha^9 & \cdots & \alpha^{3(2t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{n-1} & \alpha^{3(n-1)} & \cdots & \alpha^{(n-1)(2t-1)} \end{bmatrix} \tag{3}$$

that is, the set of all a's for which

$$aM = 0 \tag{4}$$

where $a = [a_0, a_1, a_2, a_3, \ldots, a_{n-1}]$.

III. Two-Factor Fuzzy Commitment Scheme

In this section, we propose a novel concept of dynamic key generation mechanism called two-factor fuzzy commitment, which is a variant of fuzzy commitment and uses two noisy sources (i.e., two factors) from inside and outside of the device. In this concept, two noisy inputs are involved in generating a cryptographic key dynamically. One is from the internal noisy source, such as the PUF component within the device and the other is from the external noisy source. The latter is collected through the sensor component of the device under an external environment, which is a fixed place to start its operation.

A. Proposed Scheme

Two-factor fuzzy commitment consists of the following two processes.

1) Enrollment(n_E, n_I) = h, where n_E and n_I are an external and internal noisy input, respectively, and h is an output which is the helper data to be used in order to generate the key, k (see Fig. 2).
   a) Select r at random and encode it with a proper error correcting code.
   b) Compute $w = n_E \oplus n_I$.
   c) Output $h = \text{Encoder}(r) \oplus w$.
   d) Use a hash function H for generating a cryptographic key.
2) Reproduction(h, n'_E, n'_I) = k, where n'_E and n'_I are an external and internal noisy input, respectively, and h is the helper data (see Fig. 3).
   a) Reproduce the r by computing $r = \text{Decoder}(h \oplus n'_E \oplus n'_I)$.
   b) Generate the key k from the hash function H.

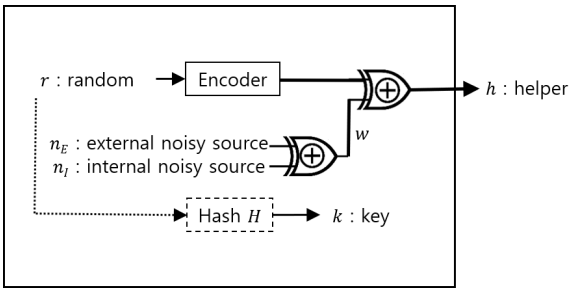


Fig. 2. Conceptual diagram of two-factor fuzzy commitment enrollment.

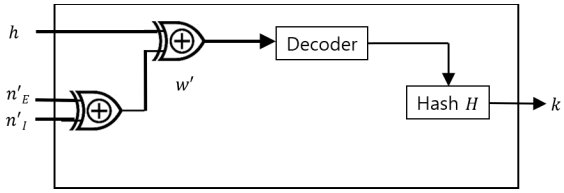


Fig. 3. Conceptual diagram of two-factor fuzzy commitment reproduction.

An essential strength of this concept is that it is difficult to reveal the key K without using two factors at the same time. Because of this strength, the following security effects can be obtained.

1) The device cannot generate its own cryptographic key if it is not located in the place which performed the enrollment process of the two-factor fuzzy commitment. Even if an attacker steals the device, he/she cannot access or obtain any critical information protected by the key from the two-factor fuzzy commitment.
2) Only the legitimate device itself can generate the key K, because the internal noisy input is made with the unique characteristics of the device. So even if an attacker sets up an external noisy input equally at the location, where the device is placed, it is impossible to get a key because the attacker cannot create the unique internal noisy input for that device.

B. Meaning of Two-Factor

In general, multifactor user authentication is a method of enhancing the security of authentication by applying various authentication factors, and the main authentication factors are: 1) knowledge factors, such as password and PIN number;

2) possession factors, such as security token and OTP module; and 3) inherent(or intrinsic) factors, such as fingerprint, iris, and voice.

In the case of a device itself that has no human interaction, we can consider the use of the following "factors" for deriving device characteristics.

1) *Knowledge Factor:* It is similar to the knowledge factors of user authentication, such as a password stored in the storage of the device.
2) *Possession Factor:* It is not clear how to define this possession factor in the case of the device itself. However, we can consider other devices capable of wired/wireless communication with the device. These devices can also help to prove the identity for the device.
3) *Intrinsic Factor:* The characteristics of the device that can be matched with human biometric information can be seen as function such as PUF.
4) *Environmental Factor:* It may be possible to identify the characteristics of this device through the characteristics of the external environmental factors, where the device is operating. It seems to be like using it as an element for authentication through a person's location information.

Therefore, our two-factor fuzzy commitment can be thought of as a key generation algorithm through two factors from the intrinsic and environment characteristics.

C. Security

For the security of the fuzzy commitment scheme, two properties are required, i.e., concealing and binding. Juels proved the following theorem for the concealment of the fuzzy commitment scheme in [5].

Theorem 1: Assume that r is random in $\{0, 1\}^k$ and w is uniformly at random from $\{0, 1\}^n$. An attacker is able to determine r from the fuzzy commitment scheme in time T with probability $p(T)$. Then it is possible for the attacker to invert $H(r)$ on a random input $r \in_R \{0, 1\}^k$ in time T with probability $p(T)$.

Proposition 1: Let X and Y be random variables. Suppose that X has a uniform distribution and X and Y are independent. Then random variable $X \oplus Y$ has also a uniform distribution.

In the case of the two-factor fuzzy commitment, since $w = n_E \oplus n_I$, the concealment property can be guaranteed by the above theorem and the proposition, if it is assumed that PUFs output is uniformly random. For the binding property of the fuzzy commitment scheme, [5] used a definition of strong binding. A fuzzy commitment scheme is strongly binding if for a given helper data h, it is infeasible for any polynomial time to find a collision w_1, w_2 $(w_1 \neq w_2)$ such that w_1 and w_2 produce the same key k related to the h but are not close in a sense of underlying error-correcting code, i.e., Decoder$(w_1 \oplus h) \neq$ Decoder$(w_2 \oplus h)$. The fuzzy commitment scheme is strongly binding if the hash function H is collision resistant [5].

Fuzzy extractor based on code-offset, i.e., fuzzy commitment, has been known to have information leakage between helper data and noisy input (secret) when the noisy input, e.g., PUF response, does not have full entropy [18], [33], [34], and Maes *et al.* [18], [34] proposed an efficient debiasing techniques based on classic von Neumann debiasing [35] to prevent this information leakage due to PUF bias.

Therefore, we can apply the debiasing step of [18] and [34] on $w = n_E \oplus n_I$ before xoring with Encoder(r) in the enrollment of our two-factor fuzzy commitment. In this case, debiasing data should be stored with the helper data and it is used in the reproduction process. For practical application of our two-factor fuzzy commitment, it is highly recommended to add these techniques to get rid of the possibility of low entropy of internal/external noisy sources.

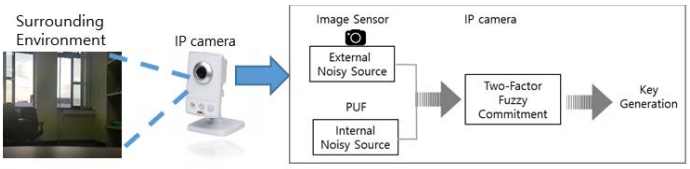


Fig. 4. Use scenario for surveillance IoT device.

IV. Securing IoT Surveillance Camera

In this section, we demonstrate a usecase to apply our two-factor fuzzy commitment into an IoT surveillance camera application. First, we suppose that the surveillance device possesses a (low-cost) PUF functionality for the internal noisy source. And an outside-image from the surveillance camera when it is initialized to start the operation in a fixed environment will be used to induce an external noisy input of the two-factor fuzzy commitment.

A. Use Scenario

As shown in Fig. 4, the two-factor fuzzy commitment can be applied to the surveillance service. First, an IoT surveillance camera is installed at the fixed place intended to be monitored and is ready to start the surveillance operations. At this moment, second, the IoT surveillance camera performs the following processes.

1) External noisy data is extracted from the initial outside image from the camera when the camera starts the operation. Image processing methods, which are introduced in the preliminaries section, are implemented to extract the external noisy data from the image in Section V.
2) Internal noisy data is collected from a PUF inside of the camera. VIA-PUF chip [36] is used as a PUF functionality for our implementation.
3) The enrollment process of the two-factor fuzzy commitment is performed by using this internal/external noisy data, and the helper data (and the debiasing data, if the two-factor fuzzy commitment includes a debiasing process) is stored in the device.

Finally, the critical information is protected by the cryptographic key derived from the reproduction process.

As we mentioned in Section IV-A. Proposed Scheme of Section III, the two-factor fuzzy commitment scheme requires someone to obtain two-factors (internal/external noisy input

factors) at the same time for generating or deriving the cryptographic key. Because of this requirement, we can promise the following security advantages in this application scenario.

1) The cryptographic key is not properly derived if the surveillance camera is moved to a different place by an illegitimate someone. So, the stored data that was encrypted by the key cannot be revealed. Therefore, even if an attacker steals the device from the original location and tries to reveal the key, he/she cannot extract any critical information inside the device.
2) Only a legitimate surveillance camera can generate the key because the camera can obtain the unique internal noisy input from the embedded PUF. Since the PUF cannot be replicated, only the camera uses this internal noisy input. So, even though an attacker gets the same image that was used in the enrollment process as the external noisy input, it is impossible for the attacker to get the key, because he cannot obtain the PUF from the camera.

In order to prove the feasibility of this application scenario which uses the image as the external noisy source, we need to consider several requirements as follows.

1) It is necessary to reproduce the proper key under indoor and outdoor environments since the surveillance camera can be placed in not only the indoor environment but also the outdoor environment.
2) The correct key should be recovered regardless of the brightness of the place if it is possible.
3) When the surveillance camera acquires image data for key recovery in the same place, where it performed the enrollment process, the key should be recovered correctly, even if some object is added in the small region.
4) KRR should be negligible for the images which are different from the original image of the enrollment process of the two-factor fuzzy commitment.
5) It is possible to consider an attacker who has an image similar to the original image of the enrollment process since the attacker can collect images from the same location, where the camera is operating. And hence, KRR should be very low even in this case.

In Section V, we introduce our prototype for the implementation and give the experiment results that are able to discuss the above requirements.

B. Extracting External Noisy Data From Images

We take an outside image, where the IoT surveillance camera is set up. The image is used as an external noisy source. We assume that images A and A' are taken in the same direction and in the same place. Due to the differences between the environmental lights at the moment of shooting, moving objects, and the noise of the camera lens itself, the values obtained by binarizing the images A and A' are represented by different values. The difference in the binary values becomes the main noise of our external noisy data. In order to binarize the image, we utilize an image processing technique such as the Otsu method. Also, we use an average filtering and a median filtering with the Otsu method to mitigate some errors. In order to make the outside image suitable for use as an external noisy source, the image must reflect the specific characteristics of the location, where the IoT surveillance camera is installed. Otherwise, there is a risk that the key can be recovered even if an attacker tries to recover the key using a similar image from another location. For example, if the IoT surveillance camera takes a picture of a common item such a door or window, those can be easily copied unless more information unique to the location is included in the image.

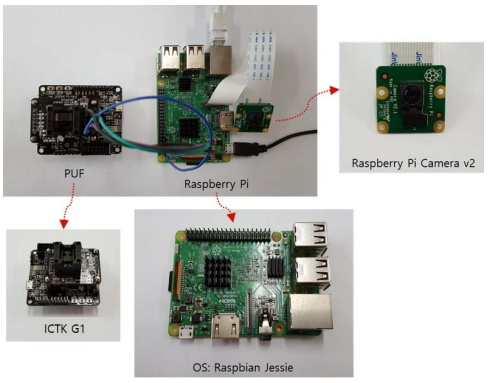


Fig. 5. Prototype IoT surveillance camera with two-factor fuzzy commitment.

C. Getting Internal Noisy Data From PUF

In general, since external noisy data can be usually considered to have a lot of noise, we tried to find and utilize a PUF function with very little noise for internal noisy data to validate the feasibility of our two-factor fuzzy commitment. Compared to other PUFs, VIA-PUF creates little noise data [36]. This is the main reason that we use VIA-PUF [36] as the internal noisy source of our two-factor fuzzy extractor implementation. VIA hole is a vertical electrical connection between two metal layers through a silicon wafer. Certain size of VIA hole, which is not recommended by the design rules of fabrication, can give an "open or short" state randomly. VIA-PUF is generated by this ambiguity of VIA-holes [37], [38]. According to the experiment of [37], VIA-PUF value was unchanged during one year under the supply voltage variations with noises and in the temperature range between 25 °C and 70 °C. Furthermore, at [39], it was reported that there is no bit change after the stress test at 125 °C for 96 h (for further detailed results, see [39]).

For our experiment environment, the IoT camera (Raspberry Pi with camera module) directly uses the PUF output from the ICTK G1 VIA-PUF chip of [38] as the internal noisy data for our two-factor fuzzy commitment.

V. Experiments

A. Prototype for Secure IoT Surveillance Camera

We introduce the main prototype and algorithms of our two-factor fuzzy commitment scheme in this section. Fig. 5 shows our prototype IoT surveillance camera. Our prototype consists of Raspberry Pi 3, Pi camera module v2 and ICTK's

Algorithm 1: Two-Factor Fuzzy Commitment Enrollment for Feasibility Testing

n : A codeword length of BCH code.
k : An input bit-length of BCH encoder
t : A correction threshold of BCH code such that $n - k \leq mt$
L : The total length of random value

Input : *Image* such that S-size and t
Output: helper data $H[i]$ and random seed(key) $R[i]$, $i = 0, \ldots, \lceil L/n \rceil$

Choose k-bit $R[i]$ at random, $i = 0, \ldots, \lceil L/n \rceil$
Encode $R[i]$ by using BCH encoder : $C[i] = BCHencoder(n, k, t)(R[i])$ for $i = 0, \ldots, \lceil L/n \rceil$
Get PUF value $n_I[i]$ with size n from ICTK G1 chip for $i = 0, \ldots, \lceil L/n \rceil$
Process the *Image* to generate a binary vector: $(n_E[i], i = 0 \ldots, \lceil L/n \rceil) = binarization(Image)$
Compute $W[i] = n_E[i] \oplus n_I[i]$, $i = 0 \ldots, \lceil L/n \rceil$
Compute a helper data : $H[i] = C[i] \oplus W[i]$
Return $H[i]$ and $R[i]$

Algorithm 2: Algorithm to Check the KRR of Two-Factor Fuzzy Commitment Reproduction

n : A codeword length of BCH code
k : An input bit-length of BCH encoder
t : A correction threshold of BCH code such that $n - k \leq mt$
L : The total length of random value

Input : $Image'$, $H[i]$, $R[i]$, t
Output: 1 or 0

Get PUF value $n'_I[i]$, $i = 0, \ldots, \lceil L/n \rceil$
Load the $Image'$, and process to make a binary vector: $(n'_E[i], i = 0, \ldots, \lceil L/n \rceil) = binarization(Image')$
Compute $W'[i] = n'_E[i] \oplus n'_I[i]$
Decode by using BCH Decoder : $R'[i] = BCHdecoder(n, k, t)(H[i] \oplus W'[i])$
Return 1, if $R'[i] == R[i]$ for all $i = 0, \ldots, \lceil L/n \rceil$ and 0, otherwise

VIA-PUF chip [36] for implementing functionalities, such as obtaining external and internal noisy sources. ICTK G1 [38] which includes VIA-PUF functionality is a chip for providing the internal noisy source of our prototype. Through the serial communication using the GPIO, the ICTK G1 is connected to the Raspberry Pi to receive the PUF value. Next, the Pi camera module v2 is used to capture an environmental image to be used as an external noisy source. For our feasibility test, we resize the image from the Pi camera module with 55×55-size.

In order to test the feasibility of our use scenario, we implemented the proposed two-factor fuzzy commitment scheme using C language and Octave. The communication package and image package of Octave was utilized to implement error correcting codes and image processing related functions. Algorithm 1 shows the pseudocode that runs the enrollment process, and Algorithm 2 shows the pseudocode that checks, where the reproduced keys (or seeds) $R'[i]$ are equal to the original $R[i]$ from Algorithm 1.

Our IoT surveillance camera prototype performs the enrollment algorithm of the two-factor fuzzy commitment by using Algorithm 1. And then the prototype performs the two-factor fuzzy commitment reproduction algorithm of Algorithm 2 to check, where the recovery of key $R[i]$ is successful. Algorithm 2 returns 1 only if the error is corrected.

To correct the error from the value $W'[i]$ which is XORing of the internal and external noisy data, we utilize the BCH decoding operation at the two-factor fuzzy commitment reproduction algorithm. Since the limit of error correction depends on the (n, k, t)-BCH parameter setting, it is important to choose a proper parameter setting. The parameter n is the codeword length of the (n, k, t)-BCH error correction code. For our experiments, we set n to 511 $(= 2^9 - 1)$ after checking the candidate parameters of the BCH codes. The parameter t is the correction threshold of the (n, k, t)-BCH error correction code. We will show various experiment results according to proper BCH parameter t in the next section. Since $n = 511$ and the size of Image is $55 \times 55 (= 3025)$, we get six $n_E[i]$s with size 511 after the binarization process (see Algorithm 1). Therefore, in our experiment for checking the KRR, we use all these $n_E[i]$ for $i = 0, \ldots, 5$.

B. Experiment Results

In this section, we evaluated the feasibility of the application scenario (i.e., securing IoT surveillance camera) by implementing our two-factor fuzzy commitment scheme on the prototype IoT surveillance camera. In order to evaluate KRRs and find the parameters that are effective for error correcting, we conducted various experiments by considering the following normal situation cases.

1) Outdoor experiment case without external influences, such as weather or people.
2) Indoor experiment case without external influences.
3) Outdoor experiment case including normal burst error (where the camera lens is slightly moved due to external weather effects, such as wind, rain, etc.).
4) Indoor experiment case including normal burst error (where objects are temporarily placed in front of the lens of the prototype).
5) Outdoor experiment case with differences between day and night.
6) Indoor experiment case with differences between bright and dark.

In addition, in order to show that key recovery is difficult when using images other than those taken during the enrollment phase, we conducted experiments on false image cases and analyzed the KRR. The number of image sets taken in each experiment is 100. If the number of images that successfully performed key recovery is $\text{num}_{\text{success}}$ and the total number of images used in the reproduction phase for the experiment is $\text{num}_{\text{total}}$, then the KRR is calculated as the following

Fig. 6. Images used in normal situation experiments. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4. (e) Case 5. (f) Case 6.

formula:

$$KRR = \frac{\text{num}_{\text{success}}}{\text{num}_{\text{total}}} * 100(\%).$$

Fig. 6 shows images taken during the enrollment and reproduction phases of the prototype in normal situations for each case.

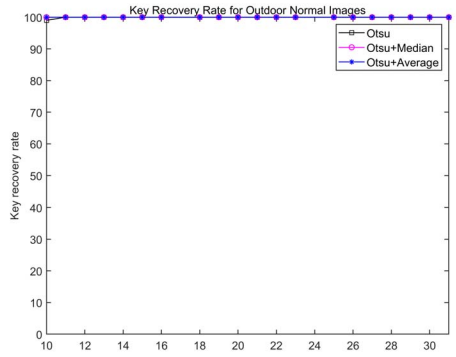
Normal Case 1 (Outdoor Experiment Results Without External Influences): We installed the prototype outside the building and conducted an experiment to evaluate the KRR of our two-factor fuzzy commitment scheme in a normal outdoor case. We set up an experimental environment with no external environmental factors, such as weather. The left image in Fig. 6(a) is the image registered in our prototype as the external source data in the enrollment phase, and the right image in Fig. 6(a) is the image used as the external source to recover the key in the reproduction phase. As shown in Fig. 7(a), when t is 10, regardless of which of the image processing techniques is used in the reproduction phase, the KRR is 100%.

Normal Case 2 (Indoor Experiment Results Without External Influences): For the indoor normal case experiment, we installed the prototype in an office without any external influences. We conducted experiments by setting up the experimental environment without obstacles, such as objects placed in front of the prototype or people passing by in front of it. The left image in Fig. 6(b) is the image registered in our prototype as the external source data in the enrollment phase, and the right image in Fig. 6(b) is the image used as the external noisy data to recover the key in the reproduction phase. As shown in Fig. 7(b), we can see that the KRR is 100% in most image processing methods at $t = 10$.

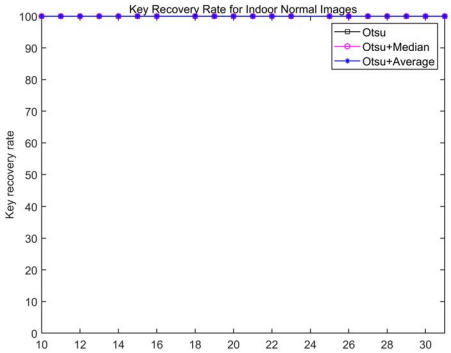
Normal Case 3 (Weather-Affected Outdoor Experiment Results): We installed the prototype outside the building and conducted an experiment to evaluate the KRR of our two-factor fuzzy commitment scheme in cases, where the camera lens of the prototype was moved slightly due to outdoor weather effects such as wind. The left image in Fig. 6(c) is the image registered in our prototype as the external noisy data in the enrollment phase, and the right image in Fig. 6(c) is the image used as the external noisy data to recover the key in the reproduction phase. Due to the wind, the camera lens was very mildly shaken, so the image taken at the reproduction phase was slightly different from the registered image. We marked the differences with red boxes.

As shown in the graph of Fig. 8(a), until we increased t to 18, the key was not recovered. At $t = 19$, the KRR increases to 96% and becomes 99% at $t = 23$, if the BCH decoding operation is performed after applying both the Otsu method and the average filtering to the right image in Fig. 6(c).

Normal Case 4 (Indoor Experiment Results Including Normal Burst Error): We installed the prototype inside a room and conducted an experiment to evaluate the KRR of our two-factor fuzzy commitment scheme in cases, where the prototype is still located in the installation location but some objects have been added to small regions of the image. The left image in Fig. 6(d) is the image registered in our prototype as the external noisy data in the enrollment phase, and the right image in Fig. 6(d) is the image used as the external noisy data to recover the key in the reproduction phase. We put a pink object in front

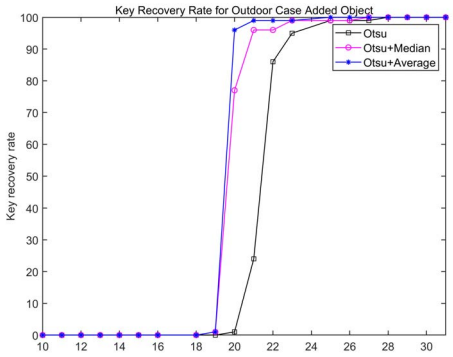


(a)

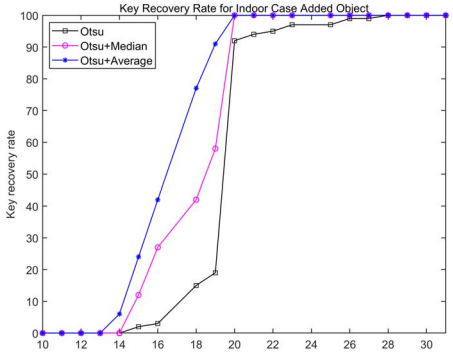


(b)

Fig. 7. Experiment results of cases 1 and 2. KRR for the normal (a) outdoor case and (b) indoor case.



(a)



(b)

Fig. 8. Experiment results of cases 3 and 4. (a) KRR for outdoor case with added weather influence. (b) KRR for indoor case with added object influence.

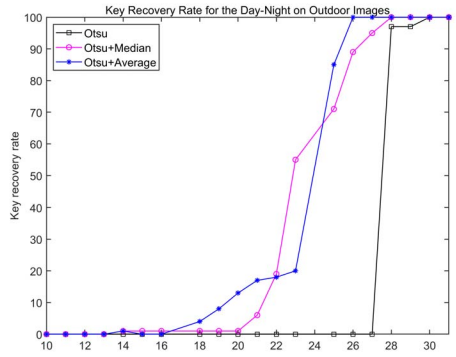
of the lens, where the prototype IoT surveillance camera was located for the experiment. Due to the object's presence, the burst error occurs at the position, where the object is placed in the right image in Fig. 6(d). We marked it with a red box.

As shown in the graph of Fig. 8(b), in cases where the BCH decoding operation is performed after applying both the Otsu method and the average filtering to the right image in Fig. 6(d), the KRR increases to 90% at $t = 18$. From the moment t becomes 20, the KRR becomes 100% when applying the Otsu method and average filtering or median filtering to the right image in Fig. 6(d).

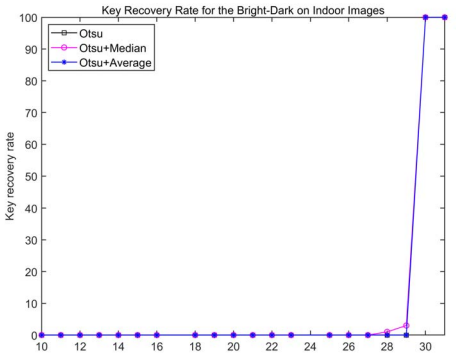
Normal Case 5 (Outdoor Experiment Case With Differences Between Day and Night): For this experiment, we installed the prototype outside the building and set the image taken during the day as external noisy data for use in the enrollment phase. The left image in Fig. 6(e) is the image registered in the enrollment phase. Then, we performed the reproduction phase to recover the key by using the image taken during the night in the same location. The right image in Fig. 6(e) is the image used as the external noisy data in the reproduction phase. In this experiment's case, it is difficult to recover the key normally due to the contrast of the image changes. So, if we set the number of error-correcting bits to small numbers between 10 and 18, it cannot recover the key as shown in Fig. 9(a). From the moment t becomes 19, the key began to be recovered when the Otsu method with average filtering was applied to the right image in Fig. 6(e), but the KRR was as low as 10%. At $t = 25$, the KRR has started to become more than 90% and then becomes 100% at $t = 26$.

Normal Case 6 (Indoor Experiment Case With Differences Between Bright and Dark): For this experiment, we installed the prototype in a room and took the image in the room with the light turned on [see the left image in Fig. 6(f)]. The image was used for external noisy data in the enrollment phase. Then, we performed the reproduction phase to recover the key by using the image taken in the same location with the light turned off [see the right image in Fig. 6(f)]. This image was used as the external noisy data in the reproduction phase. As in case 5, it is also difficult to recover the key normally due to the contrast of the image changes. So, until t becomes 29, it cannot recover the key as shown in Fig. 9(b) simply by applying the Otsu method. In this case, in order to recover the key, it is necessary to use a special image processing technique that can detect the features in a resilient manner even when the brightness of the image is completely changed. For example, if the Canny edge detection method [47] is applied to the right image in Fig. 6(f), we can confirm that key recovery is 90% from $t = 22$ or more. [See Fig. 15(a)] Since studying image processing techniques for this situation is beyond the scope of our current experiments, we do not cover the experiments using various image processing methods in this paper.

As mentioned in Section IV, we can assume an attacker has images very similar to the original image of the enrollment process since the attacker can collect images from the same



(a)



(b)

Fig. 9. Experiment results of cases 5 and 6. (a) KRR for the day-night outdoor case. (b) KRR for the bright-dark indoor case.

location, where the camera is operating, and the attacker may try to steal the surveillance camera to extract information.

In order to perform this type of experiment, we took images as if we were the attacker. Fig. 10 shows all the cases of our false images obtained from the left and right, top and bottom, front and back sides of our prototype surveillance camera by using another device. From these experiments, we can know the maximum boundary of the t parameter in BCH code to preserve the negligible KRR in our prototype implementation for false image cases.

Abnormal Case 1 (False Image Experiment Results): For this experiment, first, we installed the prototype in the room and performed the enrollment phase.

Second, we took the images slightly to the left, right, top, bottom, front, and back sides of our prototype installed in the room by using the other device. Finally, we checked the KRR for all the cases. Fig. 11(a) and (b), respectively, show the KRR of the false images taken from the left and right side of the prototype of the surveillance camera according to the change of the BCH parameter t. The KRR remains zero only if $t \leq 29$ and $t \leq 27$, respectively. For images taken from the top and bottom of the prototype, similarly, the KRR remains zero only if $t \leq 29$ [see Fig. 11(c) and (d)].

In the case for false images taken from the front and back sides of the prototype [see Fig. 11(e) and (f)], negligible KRR

TABLE I
MEMORY OVERHEAD OF OUR PROTOTYPE

	Otsu	Otsu +median	Otsu +average	C/C++ Otsu
Memory	43,147 KB			22,894 KB
Program Size	64 KB			132 KB
Image Size	28 KB			

value is maintained for $t \leq 29$. Therefore, $t = 27$ is the maximum boundary of the BCH parameter of our implementation prototype and our experiment environment.

If t is less than the maximum value, we can prevent key extraction attacks from false images. That is, whenever someone tries to extract the key using a false image, even if they take a very similar image, they cannot extract the key. Therefore, our experiment result shows that it can be secure against such an attack.

Abnormal Case 2 (Intentional/Unintentional Movement Experiment Results): In our use scenario, since we assume that the IoT camera is fixed at the place intended to be monitored, it is difficult for the position of the camera to be moved naturally. However, if the IoT camera is moved intentionally or unintentionally by someone, it should be reported to the administrator. Moreover, the key inside the camera that allows it to operate properly should not be extracted because the device cannot obtain the proper image to function correctly. For this experiment, we installed the prototype in the room and performed the enrollment phase. Then, we took the images by moving the prototype from the original position to the left side, right side and back side. The images are shown in Fig. 12. Finally, we checked the KRR for all the cases. Fig. 13(a) and (b), respectively, show the KRR of the images taken from the prototype moved to the left and right/back according to the change of the BCH parameter t. The KRR remains zero only if $t \leq 27$ and $t \leq 30$, respectively. We could see that the KRR of this abnormal case 2 is similar to the KRR of abnormal case 1. From the experiments, we can know that the maximum boundary of the t parameter in BCH code is 27 to preserve the negligible KRR in our prototype implementation for the intentional movement cases. Also, we can check again the following observation: $t = 27$ is the maximum boundary of the BCH parameter of our implementation prototype and our experiment environment. Therefore, if t is less than the maximum value, we can prevent key extraction from both false images and intentional/unintentional movement trials.

C. Performance Evaluation

The two-factor fuzzy commitment program of our prototype utilized the Otsu method to binarize the image. It also used an average filtering and a median filtering with the Otsu method to mitigate some errors. Thus, the overhead of the image processing method can be considered as the main factor affecting performance evaluation. Moreover, we used Octave [48], an interpreted programming language, to develop our program because Octave provides plenty of useful packages including error correcting code and image processing method. However, in general, the interpreted programming language takes much

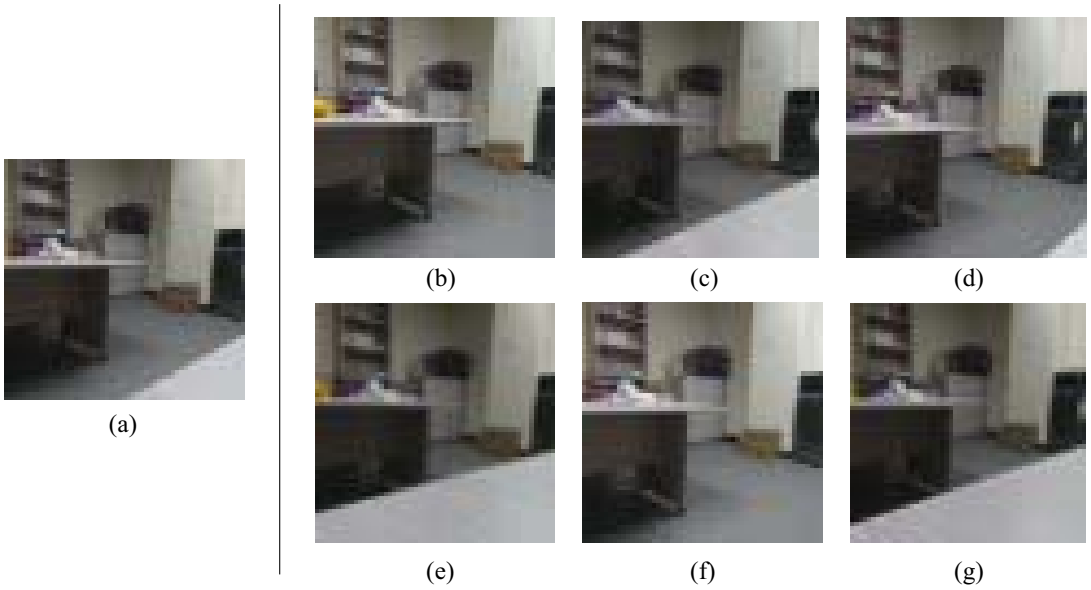


Fig. 10. Images used in false image experiments. (a) Image of enrollment phase. (b) Left. (c) Right. (d) Top. (e) Bottom. (f) Front. (g) Back.

TABLE II
SUMMARY OF THE EXPERIMENT RESULTS FOR EACH CASE

	case		KRR	$Otsu$		$Otsu + median$		$Otsu + average$	
				t	key size	t	key size	t	key size
normal cases	outdoor		100%	≥ 11	412 bits	≥ 10	421 bits	≥ 10	421 bits
	indoor		100%	≥ 10	421 bits	≥ 10	421 bits	≥ 10	421 bits
	outdoor (weather-affected)		99%	≥ 25	304 bits	≥ 23	313 bits	≥ 21	331 bits
	indoor (burst error)		100%	≥ 28	277 bits	≥ 20	340 bits	≥ 20	340 bits
	outdoor (day/night)		95%	≥ 28	277 bits	≥ 27	286 bits	≥ 26	295 bits
	indoor (bright/dark)		100%	≥ 30	259 bits	≥ 30	259 bits	≥ 30	259 bits
abnormal cases	false images	left	0%	≤ 29	268 bits	≤ 29	268 bits	≤ 29	268 bits
		right	0%	≤ 27	286 bits	≤ 27	286 bits	≤ 29	268 bits
		top/bottom	0%	≤ 29	268 bits	≤ 29	268 bits	≤ 29	268 bits
		front/back	0%	≤ 29	268 bits	≤ 29	268 bits	≤ 29	268 bits
	intentional/unintentional movement	left	0%	≤ 30	259 bits	≤ 27	286 bits	≤ 27	286 bits
		right/back	0%	≤ 30	259 bits	≤ 30	259 bits	≤ 30	259 bits

more execution time than a compiled language like C/C++. So, in order to eliminate the cause of performance degradation of our prototype due to inherent characteristics of the interpreted language, we also developed our program using C. To handle the image processing, we utilized OpenCV library [49]. Then, we evaluated the performance of the prototype by comparing the execution time of the implementation in C and the implementation in Octave, respectively.

In order to evaluate the CPU overhead according to each image processing method, we measured the CPU execution time of the reproduction phase, which is executed whenever the key is extracted from the IoT camera prototype. We measured the execution time of reproduction according to the number of error-correcting bits. As shown in Fig. 14, the average execution time of the program implemented in Octave was 1.117 s when only Otsu was applied. When the Otsu with median filtering was applied, the average execution time was 1.2025 s. In the case when the Otsu with average filtering was applied, it took 1.1375 s on average. Although there was no significant difference with each method, we found that the execution time was slightly longer when the filtering was applied. When we ran a program implemented in C/C ++ to measure the reproduction time, the average execution time was about 0.1385 s. The result of C/C++ implementation is about ten times faster than the program implemented by Octave.

To evaluate the memory or storage overhead, we measured the memory usage, the size of the program code, and the size of the image used for an external noisy source we used while the program was running. Table I shows the memory usage of our program with different image processing methods, such as Otsu-only, Otsu-median, and Otsu-average. The memory taken up by the two-factor fuzzy commitment program, running on Raspberry Pi, during the execution of each step is independent of the image processing technique. Regardless of whether we apply Otsu-only, Otsu-median, or Otsu-average as an image processing technique, our program written in the Octave programming language uses about 43 147 KB of memory, which is 4.9% of the total memory (880 552 KB) of the Raspberry

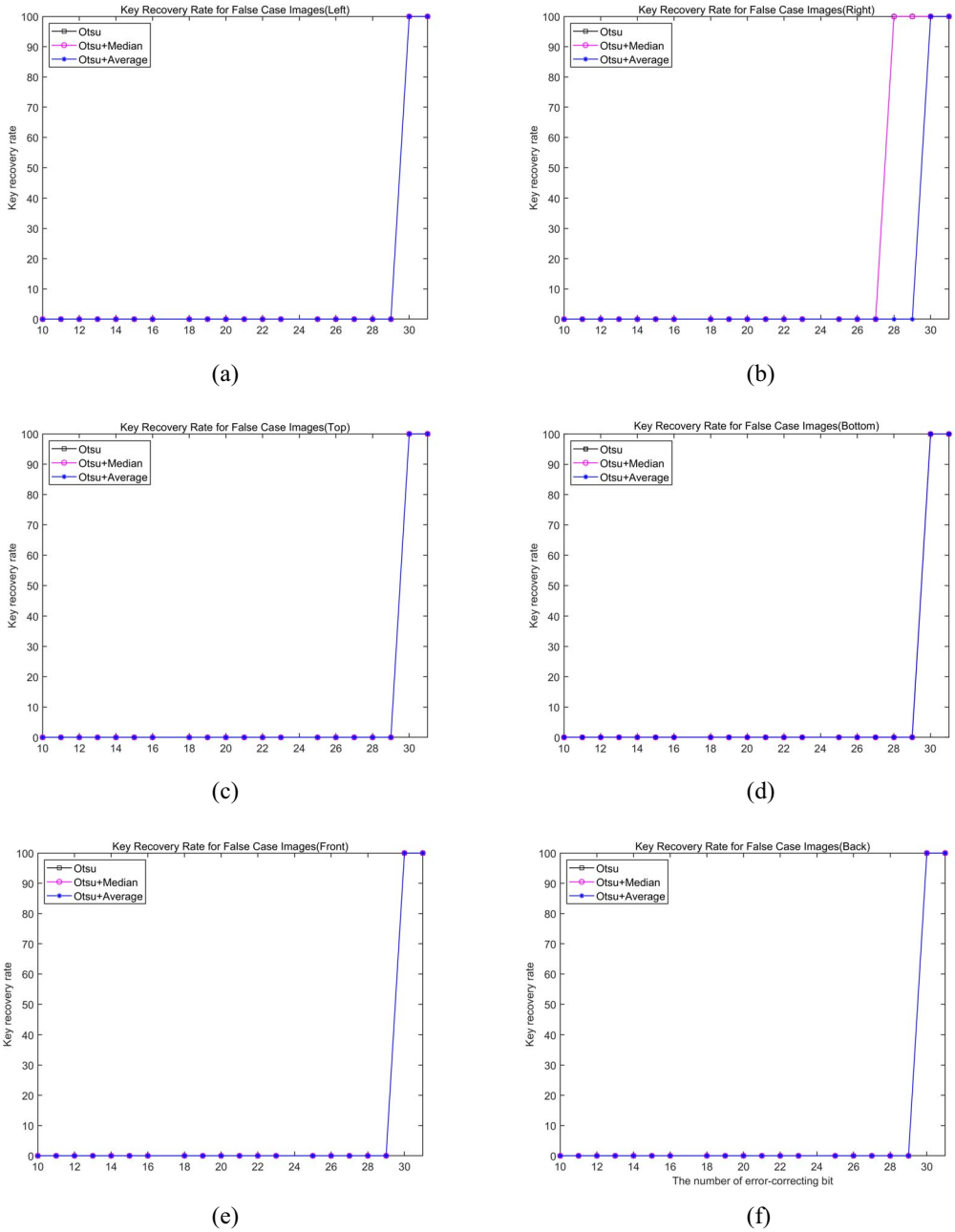


Fig. 11. Experiment results for false image cases. KRR for false cases (a) left, (b) right, (c) top, (d) bottom, (e) front, and (f) back.

Fig. 12. Images used in intentional movement experiments. (a) Original and left. (b) Right and back.

Pi when the program is running. On the other hand, our program, developed using the C/C++ programming language, uses about 22 894 KB, which is 2.6% of the total Raspberry Pi memory. The size of the image used as an external noisy source is 28 KB. The program code size implemented by Octave and by C/C++ are 64 KB and 132 KB, respectively.

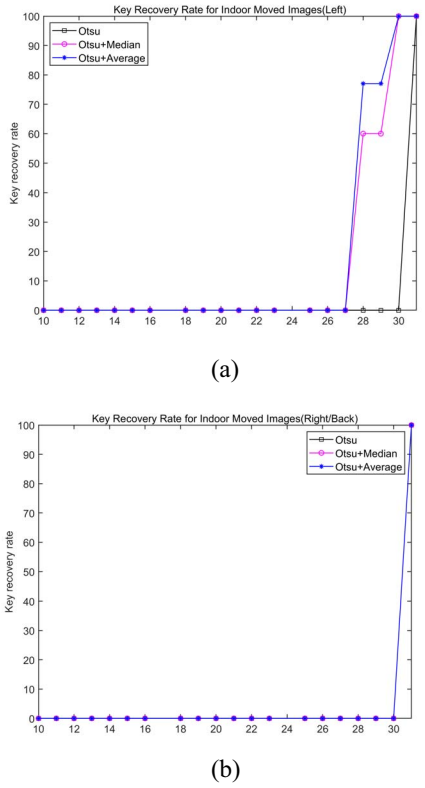


Fig. 13. Experiment results for intentional movement cases. (a) Left. (b) Right/Back.

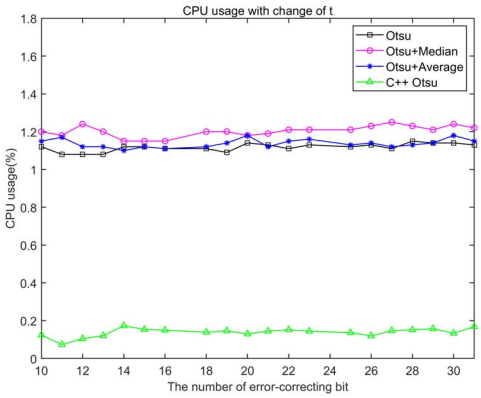


Fig. 14. CPU overhead of our prototype.

The program code size itself is smaller than the octave implementation, but we can see that the memory overhead used during program execution is cut in half when the program is implemented in C. It is possible that through implementation optimization on our prototype programs, we can expect to reduce CPU overhead even further.

D. Discussion

Table II shows the summary of the experiment results in the previous section. For indoor/outdoor cases without external influences (see the first two lines of Table II), the KRR maintains 100% at $t \geq 10$, and so, we can generate the key with 421 bits. And t parameter should be greater than

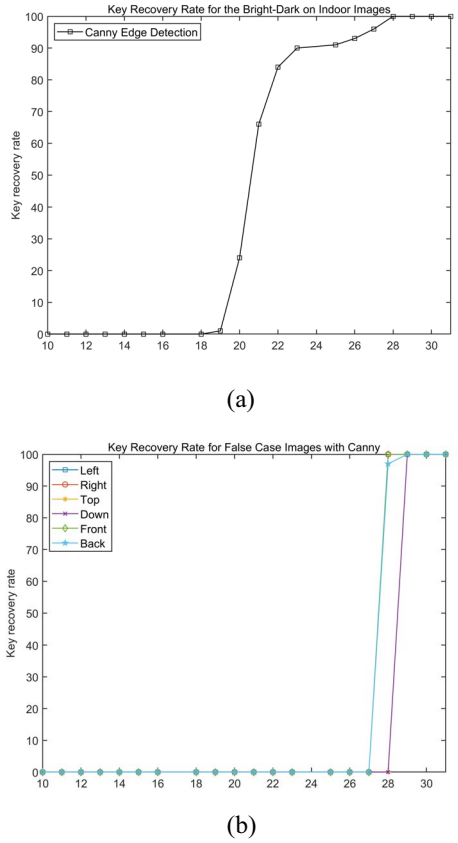


Fig. 15. Experiment results using the Canny edge detection method. (a) KRR for the bright-dark indoor images. (b) KRR for false cases.

indoor/outdoor cases without external influences to reproduce the correct key even if there are some temporary burst errors in the indoor/outdoor cases (see the third and fourth lines of Table II). Among the three image processing methods (Otsu, Otsu-median, and Otsu-average), when Otsu-average is applied, the key can be recovered most effectively at $t = 21$ for weather-affected outdoor case and $t = 20$ for indoor case with burst error. As shown in Table II, the KRR reaches 100% when Otsu-average is applied at $t \geq 26$ and $t \geq 30$, respectively, for the outdoor and indoor cases with a difference in brightness. In this case, the correct key recovery for $t = 23$ is not guaranteed by using our binarization techniques, such as Otsu, Otsu-median, and Otsu-average. The KRR, however, is able to remain over 90% from after $t \geq 22$, if we apply another image processing technique, such as the Canny edge detection method [see Fig. 15(a)]. On the other hand, in the cases of abnormal cases, the negligible KRR value is maintained for $t \leq 27$. Even if we applied the Canny image processing technique to the false image cases, it can be seen that the same result as shown in Table II is obtained for the false cases [see Fig. 15(b)]. Therefore, we can conclude that $t = 27$ is the maximum boundary of the BCH parameter and therefore, the minimum 286-bit key can be enrolled and reproduced in our experiment environment.

To demonstrate the feasibility of our two-factor fuzzy commitment, we used an environmental image and PUF as the

external noisy source and the internal noisy source, respectively. It is appropriate that the image taken by the image sensor of the IoT device can be used as an external noise source, because our use scenario for the two-factor fuzzy commitment is an IoT surveillance service with a camera module. However, in the case of resource constraint IoT devices without a camera module, another type of external source needs to be used. Various types of sensors, such as temperature, humidity [50], motion detection [51], optical [52], infrared (IR) sensors [53], etc. have been used in IoT devices. Among them, the IR sensor is used to sense certain characteristics of its surroundings by either emitting or detecting IR radiation, and the motion detector sensor is used to detect the physical movement in a given area. So, the IR and motion detection sensors would be considered as the sensors that can recognize the surrounding environment. However, more detailed analysis is needed to see if the data sensed by these sensors can be an external noisy source. Therefore, in order to apply the two-factor fuzzy commitment to various IoT devices, analyzing various types of sensing data which may be used as noisy source would be an important further study.

VI. Conclusion

In this paper, we have proposed a novel concept of the two-factor fuzzy commitment scheme that uses both an intrinsic factor of an IoT device and an environmental factor outside of the IoT device. Our two-factor fuzzy commitment scheme is expected to be useful as a countermeasure against attackers who physically steal the IoT device and then try to access information in it. In order to demonstrate the feasibility of our two-factor fuzzy commitment, we also presented the prototype IoT surveillance camera. For our prototype implementation, we utilized the image data as an external noisy source (i.e., the environmental factor) and PUF data as an internal noisy source (i.e., the intrinsic factor). Finally, we conducted experiments by considering various situations (e.g., indoor/outdoor, brightness/darkness, object moving, and false images) and evaluated the KRR for each experiment case.

References

[1] S. Barjikar, *Trusted Platform Module (TPM) Based Security on Notebook PCS-White Paper*, Mobile Platforms Group, Intel Corporation, Santa Clara, CA, USA, 2002, pp. 1–20.

[2] K. Dietrich and J. Winter, "Implementation aspects of mobile and embedded trusted computing," *Trusted Computing*. Heidelberg, Germany: Springer, 2009, pp. 29–44.

[3] *Building a Secure System Using TrustZone Technology*, ARM Ltd., Cambridge, U.K., 2009.

[4] R. Vanderhoof, "Applying the NFC secure element in mobile identity apps," in *Proc. RSA*, 2012, pp. 1–14.

[5] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. ACM Conf. Comput. Commun. Security*, 1999, pp. 28–36.

[6] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. EUROCRYPT*, Interlaken, Switzerland, 2004, pp. 79–100.

[7] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, Mar. 2008.

[8] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs Codes Cryptography*, vol. 38, no. 2, pp. 237–257, Feb. 2006.

[9] I. Nakouri, M. Hamdi, and T.-H. Kim, "Chaotic construction of cryptographic keys based on biometric data," in *Proc. High Perform. Comput. Simulat.*, Innsbruck, Austria, Jul. 2016, pp. 509–516.

[10] G. Itkis, V. Chandar, B. W. Fuller, J. P. Campbell, and R. K. Cunningham, "Iris biometric security challenges and possible solutions: For your eyes only? Using the iris as a key," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 42–53, Sep. 2015.

[11] R. A. Marino, F. H. Álvarez, and L. H. Encinas, "A crypto-biometric scheme based on iris-templates with fuzzy extractors," *Inf. Sci.*, vol. 195, pp. 91–102, Jul. 2012.

[12] F. H. Alvarez, L. H. Encinas, and C. S. Avila, "Biometric fuzzy extractor scheme for iris templates," in *Proc. World Congr. Comput. Sci. Comput. Eng. Appl. Comput.*, 2009, pp. 563–569.

[13] Y. Sutcu, Q. Li, and N. Memon, "Design and analysis of fuzzy extractors for faces," in *Proc. SPIE*, vol. 7306. Orlando, FL, USA, 2009, Art. no. 73061X.

[14] L. Leng and A. B. J. Teoh, "Alignment-free row-co-occurrence cancelable palmprint fuzzy vault," *Pattern Recognit.*, vol. 48, no. 7, pp. 2290–2303, Jul. 2015.

[15] D. K. Altop, A. Levi, and V. Tuzcu, "Towards using physiological signals as cryptographic keys in body area networks," in *Proc. Pervasive Comput. Technol. Healthcare*, Istanbul, Turkey, 2015, pp. 92–99.

[16] J. R. Wallrabenstein, "Practical and secure IoT device authentication using physical unclonable functions," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud*, Vienna, Austria, 2016, pp. 99–106.

[17] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M.-D. M. Yu, "Efficient fuzzy extraction of PUF-induced secrets: Theory and applications," in *Proc. CHES*, 2016, pp. 412–431.

[18] R. Maes, V. van der Leest, E. van der Sluis, and F. Willems, "Secure key generation from biased PUFs," in *Proc. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, 2015, pp. 517–534.

[19] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura, "Cryptographie key generation from PUF data using efficient fuzzy extractors," in *Proc. Adv. Commun. Technol.*, 2014, pp. 23–26.

[20] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Secure lightweight entity authentication with strong PUFs: Mission impossible?" in *Proc. CHES*, 2014, pp. 451–475.

[21] R. Maes, A. V. Herrewege, and I. Verbauwhede, "PUFKY: A fully functional PUF-based cryptographic key generator," in *Proc. CHES*, 2012, pp. 302–319.

[22] C. Bosch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, and P. Tuyls, "Efficient helper data key extractor on FPGAs," in *Proc. CHES*, 2008, pp. 181–197.

[23] X. Li, J. Liu, Q. Yao, and J. Ma, "Efficient and consistent key extraction based on received signal strength for vehicular ad hoc networks," *IEEE Access*, vol. 5, pp. 5281–5291, 2017.

[24] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Efficient key generation by exploiting randomness from channel responses of individual OFDM subcarriers," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2578–2588, Jun. 2016.

[25] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Secure key generation from OFDM subcarriers' channel responses," in *Proc. IEEE Globecom Workshops*, Austin, TX, USA, 2014, pp. 1302–1307.

[26] C. Javali, G. Revadigar, D. Pletea, and S. Jha, "Demo abstract: Location fingerprint evidence and authorisation using WiFi channel characteristics," in *Proc. IEEE PerCom Workshops*, Sydney, NSW, Australia, 2016, pp. 1–3.

[27] J. Huang and T. Jiang, "Secret key generation exploiting ultra-wideband indoor wireless channel characteristics," *Security Commun. Netw.*, vol. 8, no. 13, pp. 2329–2337, Sep. 2015.

[28] J. Huang and T. Jiang, "Dynamic secret key generation exploiting ultra-wideband wireless channel characteristics," in *Proc. IEEE Wireless Commun. Netw. Conf.*, New Orleans, LA, USA, 2015, pp. 1701–1706.

[29] W. Xiong *et al.*, "Run-time accessible DRAM PUFs in commodity devices," in *Cryptographic Hardware and Embedded Systems* (LNCS 9813). Heidelberg, Germany: Springer, 2016, pp. 432–453.

[30] Y. Cao, L. Zhang, S. S. Zalivaka, C.-H. Chang, and S. Chen, "CMOS image sensor based physical unclonable function for coherent sensor-level authentication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 11, pp. 2629–2640, Nov. 2015.

[31] C. Huth, D. Becker, J. G. Merchan, P. Duplys, and T. Güneysu, "Securing systems with indispensable entropy: LWE-based lossless computational fuzzy extractor for the Internet of Things," *IEEE Access*, vol. 5, pp. 11909–11926, 2017.

[32] U. Rührmair *et al.*, "Modeling attacks on physical unclonable functions," in *Proc. 17th ACM CCS*, Chicago, IL, USA, 2010, pp. 237–249.

[33] P. Koeberl, J. Li, A. Rajan, and W. Wu, "Entropy loss in PUF-based key generation schemes: The repetition code pitfall," in *Proc. IEEE Int. Symp. Hardw. Orient. Security Trust (HOST)*, Arlington, VA, USA, 2014, pp. 44–49.

[34] R. Maes, V. van der Leest, E. van der Sluis, and F. Willems, "Secure key generation from biased PUFs: Extended version," *J. Cryptograph. Eng.*, vol. 6, no. 2, pp. 121–137, Jun. 2016.
[35] J. von Neumann, *Various Techniques Used in Connection With Random Digits* (Applied Math Series 12). New York, NY, USA: Nat. Bureau Stand., 1951.
[36] ICTK. (2014). *VIA-PUF*. [Online]. Available: http://www.ictk.com/servicenproduct/puf
[37] T. W. Kim, B. D. Choi, and D. K. Kim, "Zero bit error rate ID generation circuit using via formation probability in 0.18 μm CMOS process," *Electron. Lett.*, vol. 50, no. 12, pp. 876–877, Jun. 2014.
[38] D. Bak, "The innovative IoT security solution, VIA PUF," in *Proc. Connect Security World*, 2017, pp. 1–21.
[39] D. Jeon, J. H. Baek, D. K. Kim, and B. D. Choi, "Towards zero bit-error-rate physical unclonable function: Mismatch-based vs. physical-based approaches in standard CMOS technology," in *Proc. Euromicro Conf. Digit. Syst. Design*, Funchal, Portugal, 2015, pp. 407–414.
[40] N. Chaki, S. H. Shaikh, and K. Saeed, "A comprehensive survey on image binarization techniques," in *Exploring Image Binarization Techniques, Studies in Computational Intelligence*, vol. 560. New Delhi, India: Springer, May 2014, pp. 5–15.
[41] R. Chien, "Cyclic decoding procedures for Bose–Chaudhuri–Hocquenghem codes," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 4, pp. 357–363, Oct. 1964.
[42] B. Gatos, I. Pratikakis, and S. J. Perantonis, "Adaptive degraded document image binarization," *Pattern Recognit.*, vol. 39, no. 3, pp. 317–327, 2006.
[43] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "Performance evaluation methodology for historical document image binarization," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 595–609, Feb. 2013.
[44] M. Valizadeh and E. Kabir, "Binarization of degraded document image based on feature space partitioning and classification," *Int. J. Document Anal. Recognit.*, vol. 15, no. 1, pp. 57–69, Mar. 2012.
[45] R. Hedjam, R. F. Moghaddam, and M. Cheriet, "A spatially adaptive statistical method for the binarization of historical manuscripts and degraded document images," *Pattern Recognit.*, vol. 44, no. 9, pp. 2184–2196, Sep. 2011.
[46] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
[47] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[48] J. W. Eaton, D. Bateman, and S. Hauberg, *GNU Octave: Free Your Numbers*, 3rd ed. Boston, MA, USA: Free Softw. Foundation, Feb. 2011. [Online]. Available: https://www.sci.unich.it/acciaro/octave.pdf
[49] Doxygen. (Feb. 2018). *OpenCV Modules (3.4.1)*. [Onlines]. Available: https://docs.opencv.org/3.4.1/
[50] G. Parameswaran and K. Sivaprasath, "Arduino based smart drip irrigation system using Internet of Things," *Int. J. Eng. Sci. Comput.*, vol. 6, no. 5, pp. 5518–5521, May 2016.
[51] A. N. Ansari, M. Sedky, N. Sharma, and A. Tyagi, "An Internet of Things approach for motion detection using Raspberry Pi," in *Proc. IEEE Int. Conf. Intell. Comput. Internet Things (ICIT)*, Harbin, China, Jan. 2015, pp. 131–134.
[52] V. Sundareswaran *et al.*, "Electro-optical infrared sensor technologies for the Internet of Things," in *Internet of Things and Data Analytics Handbook*. Hoboken, NJ, USA: Wiley, Dec. 2017, pp. 167–185.
[53] A. Singh, P. Aggarwl, and R. Arora, "IoT based waste collection system using infrared sensors," in *Proc. Rel. Infocom Technol. Optim. (ICRITO)*, 2016, pp. 505–509.

Dooho Choi (M'18) received the B.S. degree in mathematics from Sungkyunkwan University, Seoul, South Korea, in 1994, and the M.S. and Ph.D. degrees in mathematics from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1996 and 2002, respectively.

He has been a Principal Researcher with Electronics and Telecommunications Research Institute, Daejeon, since 2002. He was a Visiting Research Fellow with Queens University Belfast, Belfast, U.K., from 2016 to 2017. His current research interests include side channel analysis and its countermeasure design, quantum crypto analysis, and security technologies of IoT.

Seung-Hyun Seo (M'13) received the B.S., M.S., and Ph.D. degrees from Ewha Womans University, Seoul, South Korea, in 2000, 2002, and 2006, respectively.

She was a Post-Doctoral Researcher of Computer Science with Purdue University, West Lafayette, IN, USA, for two and half years, a Senior Researcher with the Korea Internet and Security Agency for two years, and a Researcher for three years with Financial Security Agency, Seoul. She was an Assistant Professor with Korea University Sejong Campus, Sejong City, South Korea for two years. In 2017, She joined with Hanyang University, Ansan, where she is an Associate Professor. Her current research interests include cryptography, IoT security, mobile security, secure cloud computing, and malicious code analysis.

Yoon-Seok Oh (S'18) received the B.S. degree from the Department of Mathematics, Korea University Sejong Campus, Sejong City, South Korea, in 2017. He is currently pursuing the master's degree with the Department of Electrical Engineering, Hanyang University, Ansan, South Korea.

He was with Electronics and Telecommunications Research Institute, Daejeon, South Korea, as a Student Intern Researcher in 2016. His current research interests include drone security and security/privacy in wireless networks and Internet of Things.

Yousung Kang (M'08) received the B.Eng. and M.Eng. degrees in electronics engineering from Chonnam National University, Gwangju, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in electrical and electronic engineering from KAIST, Daejeon, South Korea, in 2015.

He has been a Principal Researcher with Electronics and Telecommunications Research Institute, Daejeon, since 1999. Since 2004, he has been with the IT International Standard Expert of Telecommunications Technology Association, Seoul, South Korea. He was a Visiting Researcher with Queens University Belfast, Belfast, U.K., from 2011 to 2012. His current research interests include cryptographic protocol, side channel analysis, and key-hiding technology.