# Joint Heterogeneous PUF-Based Security-Enhanced IoT Authentication

Seungwook Yoon, *Student Member, IEEE*, Seungnam Han, *Graduate Student Member, IEEE*,
and Euiseok Hwang, *Senior Member, IEEE*

***Abstract*—This study proposes a novel authentication scheme that provides enhanced security for Internet of Things (IoT) applications by integrating a radio frequency (RF) physical unclonable function (PUF) with a device PUF. While traditional PUF-based authentication methods are lightweight, they are vulnerable to replay attacks because challenge-response pairs (CRPs) are exposed to adversaries over wireless channels. To prevent CRP exposure, we propose a joint scheme that integrates the physical layer features of wireless channels with those of the device PUF. Our authentication process consists of two stages: an enrollment stage and an authentication stage. During the enrollment stage, the physical features of the static random-access memory (SRAM) in the IoT devices are shared with the server, which generates a hashing model using the amplitudes of channel state information (CSI) as the RF-PUF and transfers it to the IoT device. In the authentication stage, the server and IoT devices exchange pilot signals to estimate the shared CSI. Both parties then generate the challenge information using the hashing model, which depends on the CSI. The challenge information is mapped to the response of the device PUF as the cryptographic key. If both parties are legitimate, the proposed model is updated through CSI amplitudes. We evaluated the proposed authentication protocol using a testbed based on Raspberry Pi and acquired a CSI data set and SRAM-PUF using universal software radio peripheral and Arduino, respectively. Numerical results demonstrate that our method effectively defends against diverse attacks, even in critical CRP exposure scenarios.**

***Index Terms*—Channel state information (CSI), Internet of Things (IoT), IoT authentication, physical unclonable function (PUF), radio frequency (RF) fingerprint.**

## I. Introduction

With the digitization of industries, the Internet of Things (IoT) has played a crucial role in controlling and monitoring intelligent systems. Cyber–physical systems, such as smart homes and smart factories have installed several IoT devices totaling over 13 billion globally in 2022 [1]. Many IoT networks are connected with wireless communication, and several components of the systems are controlled through wireless devices [2]. Unfortunately, wireless networks in cyber–physical systems are easily exposed to cyberattacks owing to factors, such as extensive system scale, device heterogeneity, and resource constraints [2], [3]. In particular, resource constraints, such as low-cost CPU and battery capacity make it difficult for robust security technologies to use IoT networks [4]. Therefore, lightweight security technology for IoT networks has become an important research issue.

Diverse technologies have been developed to ensure the security of IoT devices. An authentication scheme using cryptographic keys is a major technology for IoT security because it can block malicious users [5], [6]. In general, conventional keys are stored in the nonvolatile memory (NVM) of an IoT device. However, the cryptographic key in NVM can be deciphered from its electrical nature, considering that the adversary can physically access the IoT device in an open place [7], [8]. In recent years, the physical unclonable function (PUF) has gained widespread interest as a lightweight cryptographic key generator [9]. PUF is a physical fingerprint that depends on the manufacturing process and hence can be used for authentication and identification, considering each device has unique physical features. Moreover, the adversary cannot clone the cryptographic key based on a device PUF since it is not stored in the NVM [7].

The PUF-based authentication scheme uses the challenge-response pair (CRP), which has a one-to-one mapping relationship. A verifier, such as an authentication server and sends challenge information, e.g., memory addresses, to the IoT device. Then, the IoT device generates response information for the cryptographic key. However, PUF-based authentication has some security vulnerabilities. First, the adversary can eavesdrop on CRPs for a replay attack when legitimate users authenticate from the authentication server. As a result, legitimate users must change their cryptographic key in case of such attacks. However, a PUF-based authentication system uses physical characteristics and cannot quickly update its CRPs. Second, some PUFs, such as Arbiter-PUF [10] and ring oscillator (RO)-PUF [11], are vulnerable to modeling attacks that use machine learning (ML) because of the linear relationship between challenges and responses. Therefore, we propose a PUF-based authentication scheme to prevent replay and ML attacks.

Fig. 1 shows an IoT wireless network wherein the server, as a local verifier, authenticates IoT devices. In conventional

Manuscript received 9 February 2023; revised 18 April 2023; accepted 15 May 2023. Date of publication 25 May 2023; date of current version 9 October 2023. This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2021-2021-0-01835, and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2021R1A2C1009803. *(Corresponding author: Euiseok Hwang.)*

Seungwook Yoon is with the School of Mechatronics, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea (e-mail: ysw1207@gist.ac.kr).

Seungnam Han and Euiseok Hwang are with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea (e-mail: snhan0911@gm.gist.ac.kr; euiseokh@gist.ac.kr).

Digital Object Identifier 10.1109/JIOT.2023.3279847

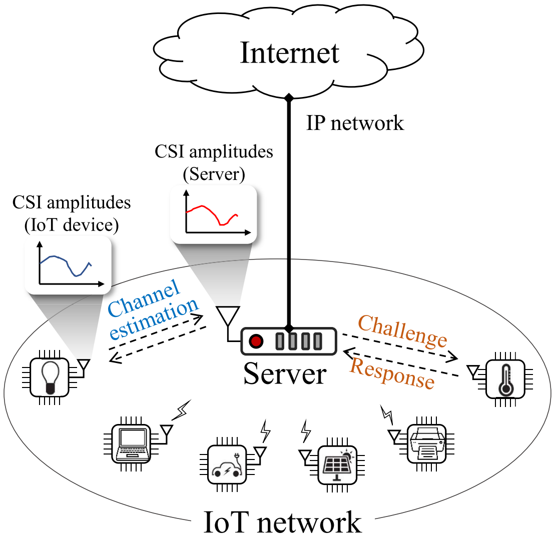


Fig. 1. System structure of the IoT network.

authentication systems, the CRPs of IoT devices are shared through wireless networks. Herein, the channel estimation uses the pilot signal to optimize the link performance of the wireless communication before communicating information [12]. Both parties can obtain the channel state information (CSI), which is a radio frequency (RF) fingerprint that is affected by the physical environment, using channel estimation.

Therefore, this study proposes a novel IoT authentication scheme that integrates the device PUF and RF-PUF to enhance IoT network security without exposing CRPs. Note that a server and legitimate device with a shared wireless communication channel can estimate similar CSI amplitudes using the pilot signal. Two parties generate the challenge information of a device PUF using RF-PUF, which are amplitudes of CSI. We employed the feature extraction-based hashing model to transform the CSI amplitudes into the challenge information. The CSI amplitudes are changed based on time and place, considering that they are affected by factors, such as multipath fading and scattering. Therefore, the challenge information of the device PUF can be hidden from the adversary, given that estimating the CSI amplitudes of legitimate devices from the location of the adversary is difficult. After generating the challenge information, the two parties map the challenge information to the device PUF response and use their responses as the cryptographic key. The proposed system can block replay and ML attacks by malicious users because the adversary cannot precisely estimate the challenge information of the CRP table.

### A. Our Contributions

We propose a joint heterogeneous PUF-based IoT authentication scheme that combines the device PUF and RF-PUF. Our proposed approach has three functions: 1) mutual authentication between both parties; 2) robust PUF-based key against replay and ML attacks; and 3) feasibility for IoT devices. Therefore, we conducted a key evaluation based on an actual data set to assess the applicability of the joint heterogeneous PUF key. Furthermore, the performance of the proposed approach in blocking adversaries was analyzed under diverse scenarios. The main contributions of our study are as follows.

1) We designed a joint heterogeneous PUF-based IoT authentication scheme for IoT networks. A joint heterogeneous PUF is a new PUF structure, wherein the challenge information of a device PUF is controlled using an RF-PUF.
2) The proposed PUF-based key was generated using the static random-access memory (SRAM)-PUF and CSI amplitudes, and the performance of the PUF-based key was evaluated. Our proposed scheme showed reliable performance in terms of uniformity, reliability, and uniqueness.
3) The proposed method can resist replay attacks since CRPs are not shared using the RF-PUF in wireless networks. The weak PUF, which is vulnerable to replay attacks owing to a small number of CRPs, can be used through the proposed method.
4) The proposed authentication scheme can avoid modeling attacks because the adversary cannot acquire the data to train ML models through eavesdropping.
5) We designed a mutual authentication protocol by updating the unreusable (temporary) identity of the IoT device based on CSI amplitudes.
6) The proposed scheme was evaluated using actual CSI amplitudes and SRAM-PUFs using universal software radio peripheral (USRP) and Arduino. Furthermore, the authentication protocol was tested using Raspberry Pi, which mimics the IoT device.

### B. Organization of This Article

The remainder of this article is organized as follows. Section II presents works related to IoT authentication. Section III explains the preliminary studies to realize the proposed method. Section IV introduces the joint heterogeneous PUF-based IoT authentication scheme. Section V describes the experimental environment and the threat model. Section VI describes the performance evaluation of the proposed method. Section VII presents an overall assessment of our proposed method. Finally, Section VIII presents the study conclusions.

## II. Related Work

Many researchers have studied PUF-based IoT authentication for enhancing IoT security. According to the number of CRPs, PUF can be classified as weak PUF and strong PUF. Strong PUFs have many CRPs and hence have been widely used in authentication protocols [13]. However, strong PUFs are vulnerable to modeling attacks owing to the linear relationship between the challenge and the response. Therefore, configurable PUFs have been studied to resist ML-based modeling attacks. Gope et al. [14] investigated the scalable protocol-level approach for PUF-based authentication to avoid ML attacks. A one-time PUF, which can change the PUF configuration, is used in this method. Similarly, Zhang et al. [15]

proposed a configurable tristate PUF to mitigate the modeling attack for IoT security. The configurable tristate PUF flexibly selects the working states, including the bistable ring (BR)-PUF, RO-PUF, and Arbiter-PUF. Zerrouki et al. [16] proposed mutual authentication based on Arbiter-PUF. The reliability of the PUF affected by noise is enhanced using error correction coding, and ML attacks are blocked since the response of the PUF is encrypted. In other studies, fake PUFs were used for poisoning attacks on ML models of adversaries [17]. Although several authentication schemes can prevent resistible ML attacks, it is not a fundamental solution considering that CRPs are exposed through wireless channels. Therefore, this study proposes a new authentication structure, which hides CRPs using the RF fingerprint.

In recent years, RF fingerprints, which include hardware-based and channel signature-based fingerprints, have been used for IoT security [18], [19], [20]. In general, channel signature-based fingerprints, which include the received signal strength (RSS) and CSI, have been employed for indoor localization because of their sensitivity to the physical environment [21], [22]. Because the diversity of keys is limited in a static environment, it is challenging to employ channel signature-based fingerprints for generating security keys. Moreover, in a static environment, an adversary can easily obtain the secret key of legitimate users through a replay attack. In [23], hardware-based fingerprints were used to identify IoT devices. Four modulation features were employed, including in-phase and quadrature-phase (I/Q) offset, modulation offset, carrier frequency offset (CFO), and differential constellation trace figure. The weights of the RF features were adaptively controlled to classify IoT devices according to the signal-to-noise ratio (SNR). Similarly, device identification was studied using the RF fingerprint in a low SNR environment in [24]. The dynamic shrinkage learning network was designed to classify some devices by training the I/Q samples of RF signals. Sankhe et al. [25] also proposed optimized radio classification through convolutional neural networks (ORACLE), which IoT devices using the I/Q and DC offsets of WiFi channels. To enhance the security against spoofing attacks, ORACLE switched the order of keys between pairwise impairments using a pseudo-noise binary sequence. However, the authors did not consider the performance degradation of the convolutional neural network owing to the contamination of the RF fingerprints. Therefore, Bari et al. [26] proposed a preprocessing scheme using CFO and frequency offset variation to improve the identification scheme based on ORACLE. As described above, hardware-based RF fingerprint-based deep learning schemes have been employed for IoT security. However, these schemes have two disadvantages: 1) the hardware-based features may be copied by a malicious attack (e.g., eavesdropping) and 2) deep learning models should be retrained when a new IoT device is added to the wireless network.

The proposed method employs SRAM-PUF, which has the advantages of low cost and randomness, as a device PUF for IoT authentication. SRAM-PUF is robust against modeling attacks considering that PUF responses do not correlate within each cell [27]. However, it has some weaknesses, including a small number of CRPs and low entropy. In [28], a reordered SRAM-PUF was applied to mitigate the problem of a small number of CRPs. In [29], IoT authentication schemes were proposed as wearable devices. To overcome the problem of low entropy, SRAM-PUF was integrated with multiple entropy sources, including electrocardiogram and heart rate variability. However, this method can only be applied to wearable devices owing to the data being biometric. Gope and Sikdar [30] employed SRAM-PUF to securely update the firmware of the smart meter in a smart grid. This method combines a secure state of firmware and SRAM-PUF to block unauthorized installations. This approach needs more resources to enhance security than the conventional PUF, but it is sufficient to apply to IoT devices.

## III. Preliminaries

In this study, the challenge information of a device PUF was generated using the RF fingerprint. We employed SRAM-PUF (as device PUF) and CSI amplitudes (as RF-PUF) to apply the proposed scheme. This section elaborates on their characteristics.

### A. SRAM-PUF

SRAM has been employed as a physical feature for diverse applications, such as true random number generator [31] or authentication scheme [28], [29], owing to the randomness of the power-up states of SRAM cells. An SRAM cell comprises six transistors, including cross-coupled inverters and access transistors [32]. The binary output of the power-up state depends on the order of the driven inverters. The power-up state of the SRAM cell is affected by noise and other environmental factors. However, stable cells that can generate the same value (0-value or 1-value) are either 0-skewed or 1-skewed cells, respectively. Otherwise, the unstable cell is called a neutral-skewed cell. To classify each cell type, the probabilities of SRAM cells, wherein the power-up state with 1-value, are evaluated as follows:

$$p_i = \frac{1}{N_{\text{SRAM}}} \sum_{j=1}^{N_{\text{SRAM}}} x_{i,j} \text{ and } \forall i, x_{i,j} \in \{0, 1\} \tag{1}$$

where $x_{i,j}$ is the $j$th reading value of the $i$th cell address, and $N_{\text{SRAM}}$ is the number of readings for evaluating the cell type. $p_i$ is the empirical probability of 1-values of the $i$th cell address, which is evaluated by $N_{\text{SRAM}}$.

Fig. 2 shows $p_i$ of 200 cells measured using the Arduino testbed. As $N_{\text{SRAM}}$ decreased, the variability of the probability increased. Because each cell is affected by the environment, e.g., temperature and noise, it was repeatedly tested to determine the cell type. As shown in Fig. 2, the $p_i$ of each cell converged when $N_{\text{SRAM}}$ was over 300. Here, we assumed that the $i$th cell is stable when $p_i$ is 0 or 1 at $N_{\text{SRAM}}$. Most PUF-based authentication studies employed stable cells to generate PUF-based keys. In addition, some studies focused on mitigating the unreliability of SRAM-PUFs that are affected by unstable cells [33].

Additionally, we evaluated the ratio of cell types, as shown in Fig. 3. The ratio of stable cells, including 1-skewed and

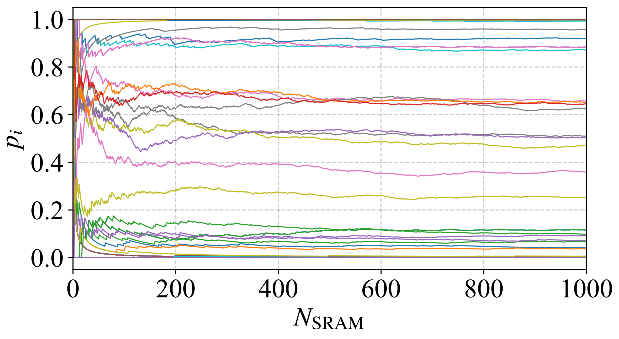
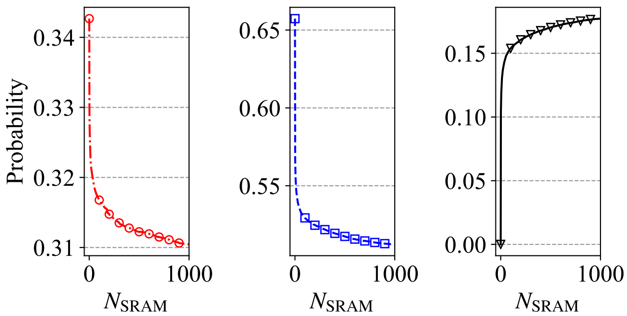


Fig. 2. Empirical probabilities of 1-values by $N_{\text{SRAM}}$ for 200 cells.



(a) (b) (c)

Fig. 3. Ratio of cell types by $N_{\text{SRAM}}$. (a) 0-skewed cell. (b) 1-skewed cell. (c) Neutral-skewed cell.

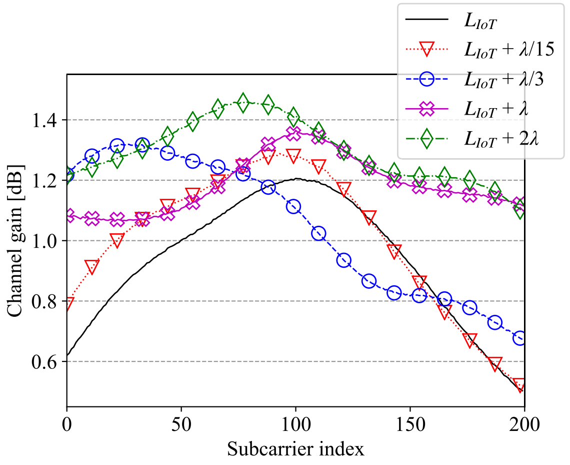


Fig. 4. CSI amplitudes based on the change in the location of the IoT device in the LTE environment. The center frequency is 1 GHz.

0-skewed cells, gradually decreased as $N_{\text{SRAM}}$ increased. Moreover, the ratio of the 0- and 1-skew cells was approximately 30% and 50%, respectively. The asymmetric ratio of cell types can lower the uniformity performance and weaken the security of the SRAM key [28].

### *B. Channel State Information*

Many researchers have used RSS as an RF fingerprint considering that it has a low hardware requirement. However, RSS has low stability [34] and is unsuitable for IoT authentication schemes. Therefore, we employed CSI for our proposed scheme. CSI amplitudes, which are estimated from the signal received, always exhibit different values since they reflect the physical environment in real time. In wireless communication, a transmitted signal may be distorted by several factors, including multipath fading and shadowing [35]. Owing to these characteristics, CSI amplitudes have been employed as a dependable metric for feature extraction [36].

As shown in Fig. 4, CSI amplitudes appeared in the LTE framework, where $L_{\text{IoT}}$ is the initial location of the IoT device and $\lambda$ is the wavelength of the LTE communication. It was measured at 30 cm with a center frequency of 1 GHz. As shown in Fig. 4, the change in the CSI amplitude was observed based on the location of the IoT device. The changes in the amplitude patterns increased as the IoT device moved farther away. When the location moved by $\lambda/15$, the amplitude patterns moved slightly upward in specific subcarriers. However, the pattern of CSI amplitudes showed different waveforms at distances over $\lambda/3$. If an adversary cannot get close to the legitimate user within $\lambda/3$, estimating the amplitudes of the legitimate user becomes difficult. Furthermore, the diversity of the CSI amplitudes can be enhanced as the maximum number of paths increases [37]. In our simulation environment, the CSI amplitudes were acquired in the indoor channel, and the CSI amplitudes' waveform appeared as a form of a polynomial with a degree less than five. In a complex environment like outdoors, when CSI amplitudes are obtained, their waveforms can exhibit increased complexity.

## IV. Joint Heterogeneous PUF-Based IoT Authentication Scheme

Our goal was to authenticate IoT devices and the authentication server using the joint heterogeneous PUF. The proposed PUF combines a device PUF and RF-PUF to enhance the security of the IoT network. As shown in Fig. 5, our proposed method comprises the enrollment and authentication stages, which are discussed in this section.

### *A. Enrollment Phase*

In the enrollment phase, we describe: 1) the CRP enrollment method for SRAM-PUF and 2) the hashing model using CSI amplitudes.

*1) CRP Enrollment Method for SRAM-PUF:* To employ SRAM-PUF for IoT authentication, the authentication server must have the SRAM device information, which includes the device ID and power-up states of all cells. Fig. 6(a) shows the enrollment method for SRAM-PUF. First, the vendor observes the power-up states of all cells at $N_{\text{SRAM}}$ times using the SRAM controller and calculates the probabilities, $\boldsymbol{P} = \{p_1, p_2, \ldots, p_{N_{\text{addr}}}\}$, using (1), where $N_{\text{addr}}$ is the number of SRAM cells. After calculating the probabilities of SRAM power-up states by the vendor, it communicates the SRAM device ID and $\boldsymbol{P}$ to the authentication server through secure communication. Next, the server stores the device ID and $\boldsymbol{P}$ of each device in its database and uses them when authenticating the device.

*2) Hashing Model Using CSI Amplitudes:* Our authentication scheme generates the challenge information for a device PUF using CSI amplitudes. To apply our system model, the

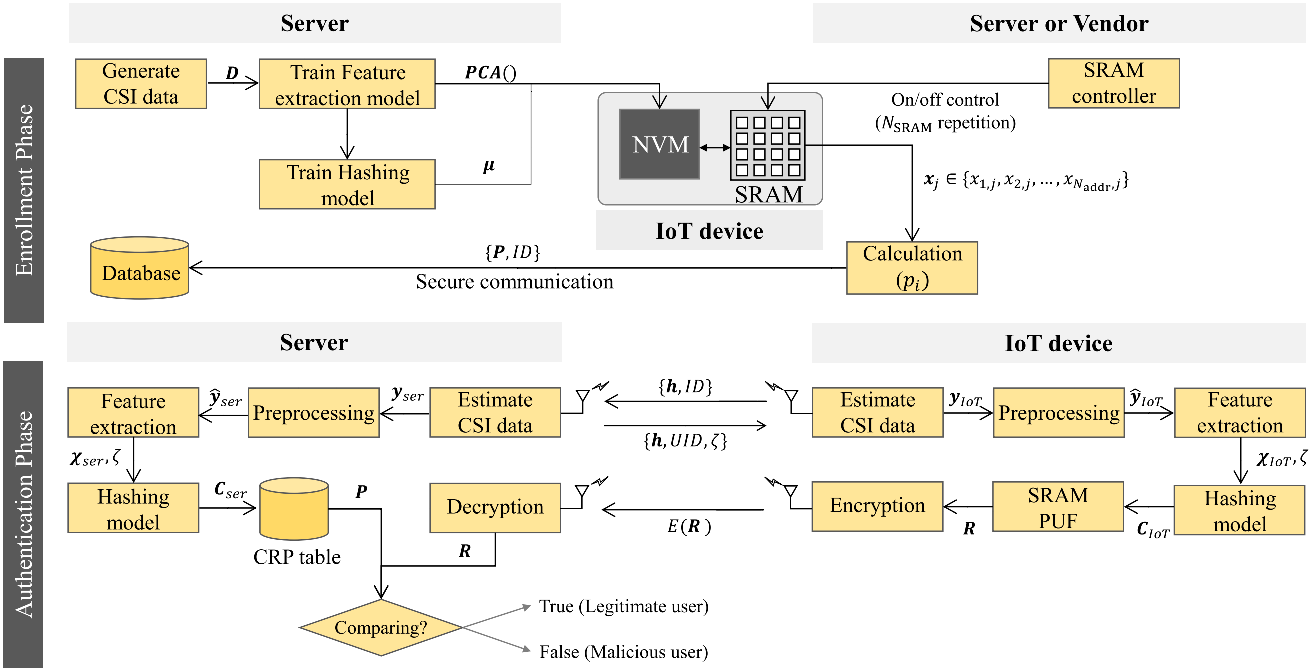


Fig. 5. System structure for the proposed method.

same challenge information must be generated using CSI amplitudes, estimated by the IoT device and server. However, there are some issues with creating the challenge information using CSI amplitudes. First, the patterns of CSI amplitudes are analog values with a wide range of offsets and are affected by contamination, such as RF noise. Therefore, the CSI amplitude errors between the server and the legitimate device can be generated owing to uncertainty of the CSI amplitudes. Second, the number of subcarriers $N_{\text{sub}}$ is 200 in the LTE framework; therefore, the authentication processes may burden the IoT device because their computation complexity increases in proportion to $N_{\text{sub}}$.

Therefore, we use the downsampling method based on summation to reduce the uncertainty and dimension of the CSI amplitudes. It can mitigate the noise effects by summing the data samples in a window. The downsampling is expressed as follows:

$$\tilde{y}_n = \sum_{m=1}^{w_D} y_{m+w_D(n-1)}, \, n \in \left\{1, 2, \ldots, \frac{N_{\text{sub}}}{w_D}\right\} \tag{2}$$

where $y$ is the CSI amplitude, which has $N_{\text{sub}}$ subcarriers, $w_D$ is the sampling rate for the downsampling, and $n$ is the index of the downsampled CSI amplitudes. $m$ is the temporal index for calculating the downsampling scheme. We then normalize CSI amplitudes in the range between 0 and 1 to eliminate the offset of the CSI amplitudes, expressed as follows:

$$\hat{y}_n = \frac{\tilde{y}_n - \min_n \tilde{\boldsymbol{y}}}{\max_n \tilde{\boldsymbol{y}} - \min_n \tilde{\boldsymbol{y}}} \tag{3}$$

where $\hat{y}_n$ is the normalized CSI amplitude. Although the offset of the CSI amplitude is eliminated by normalization, CSI amplitudes have diverse patterns. Therefore, our proposed scheme needs to collect numerous CSI data sets to train the hashing model, but it is challenging to acquire numerous patterns since the CSI pattern is changed by the physical environment [34].

We propose a fake CSI data set to solve the lack of training data. The CSI amplitudes of each subcarrier were similar to those of the adjacent subcarrier, as shown in Fig. 4. Therefore, the profile of the CSI amplitudes revealed a polynomial of order five or less in the indoor environment. These characteristics were used to generate the training CSI data set using Algorithm 1. In this algorithm, $K$ is the number of fake CSI data sets generated, and $\gamma_{\min}$ and $\gamma_{\max}$ are the minimum and maximum distance between inflection points. $v$ is the degree of the polynomial, which is set using uniformly random distribution. $\alpha_t$ and $\beta_t$ are the subcarrier index and normalized amplitudes of the inflection point, respectively. $N_{\text{sub}}$ is the number of all subcarriers, which varies depending on the network environment.

To train the hashing model, a preprocessed fake CSI data set $\hat{\boldsymbol{D}}$ was employed, as shown in Fig. 6(b). We used feature extraction and clustering-based hashing models to generate the challenge information for a device PUF. We employed the principal component analysis (PCA) model as the feature extraction model [38]. It can improve clustering accuracy by mitigating the noise effects and reduce clustering complexity through dimension reduction.

Furthermore, we employed $K$-means clustering as the hashing model to map the CSI amplitudes to integer values. Many clusters are needed to increase the number of CRPs; however, it is challenging to set many clusters owing to the computation resource of IoT devices. As the number of clusters increases, the computation complexity and sensitivity of the clustering model increase. The challenge information of IoT devices or servers can be mapped to different values

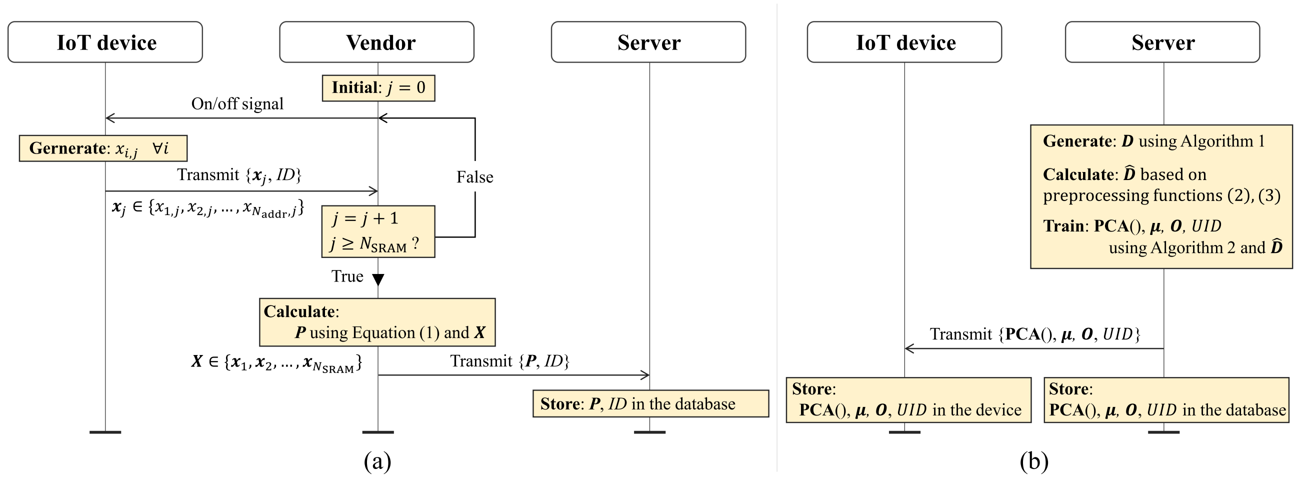

IoT device
Vendor
Server
Initial: $j = 0$
On/off signal
Gernerate: $x_{i,j}$ $\forall i$
Transmit $\{\boldsymbol{x}_j, ID\}$
False
$\boldsymbol{x}_j \in \{x_{1,j}, x_{2,j}, \ldots, x_{N_{\text{addr}},j}\}$
$j = j + 1$
$j \geq N_{\text{SRAM}}$ ?
True
Calculate:
$\boldsymbol{P}$ using Equation (1) and $\boldsymbol{X}$
$\boldsymbol{X} \in \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_{\text{SRAM}}}\}$
Transmit $\{\boldsymbol{P}, ID\}$
Store: $\boldsymbol{P}$, $ID$ in the database
(a)
IoT device
Server
Generate: $\boldsymbol{D}$ using Algorithm 1
Calculate: $\widehat{\boldsymbol{D}}$ based on preprocessing functions (2), (3)
Train: PCA(), $\boldsymbol{\mu}$, $\boldsymbol{O}$, $UID$ using Algorithm 2 and $\widehat{\boldsymbol{D}}$
Transmit $\{$PCA(), $\boldsymbol{\mu}$, $\boldsymbol{O}$, $UID\}$
Store: PCA(), $\boldsymbol{\mu}$, $\boldsymbol{O}$, $UID$ in the device
Store: PCA(), $\boldsymbol{\mu}$, $\boldsymbol{O}$, $UID$ in the database
(b)

Fig. 6. Proposed enrollment phase: (a) SRAM-PUF and (b) RF-PUF.

**Algorithm 1:** Fake CSI Data Generation Algorithm for the Hashing Model

```
input: $K$, $\gamma_{\min}$, $\gamma_{\max}$
output: $\boldsymbol{D}$
$\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_K\}$;
$v \leftarrow$ randint(2, 5);
$\alpha_t \in \{\alpha_0, \alpha_1, \ldots, \alpha_{v+1}\}$;
$\beta_t \in \{\beta_0, \beta_1, \ldots, \beta_{v+1}\}$;
for $k = 1 : K$ do
    for $t = 0 : v + 1$ do
        $\beta_t \leftarrow$ rand;
        if $t == 0$ then
            $\alpha_t \leftarrow 0$;
        else
            $\alpha_t \leftarrow$ randint($\alpha_{t-1} + \gamma_{\min}, \alpha_{t-1} + \gamma_{\max}$);
    $\check{\boldsymbol{d}}$ = interpolation($\boldsymbol{\alpha}, \boldsymbol{\beta}$, type : quadratic);
    $\boldsymbol{d}_k \leftarrow \check{\boldsymbol{d}}[1 : N_{\text{sub}}]$
```

**Algorithm 2:** Sliding Window-Based Clustering Model

```
Input: $\widehat{\boldsymbol{D}}, K, N_{\text{sub}}, w_D, S, \omega, \theta, \phi, L_{ID}$
Output: PCA(), $\boldsymbol{\mu}$, $\boldsymbol{O}$, $UID$
$T = \text{Floor}(\frac{N_{\text{sub}}/w_D - \omega}{S}) + 1$;
$\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$;
$\boldsymbol{Z} \leftarrow$ zeros[$TK, \omega$];
// Sliding window
for $t = 1 : T$ do
    $\boldsymbol{Z}[(t-1)K + 1{:}tK, :]$
    $= \widehat{\boldsymbol{D}}[:, (t-1)S + 1 : (t-1)S + \omega]$;
// Training PCA and the clustering model
Train $\boldsymbol{\chi}$ = PCA($\boldsymbol{Z}$, dimension : $\theta$);
Train $\boldsymbol{\mu}$ = Kmeans($\boldsymbol{\chi}$, the number of clusters : $\phi$);
// $\boldsymbol{\mu}$ : Center points of clustering model
// Generating a temporary identity
$UID \leftarrow$ randint(0, 1, $L_{\text{ID}}$);
// Permutating labels of clusters
for $t = 1 : T$ do
    $\boldsymbol{o}_t \leftarrow$ permutations([1, 2, ..., $\phi$])
```

owing to the sensitivity of the clustering model and the uncertainty of CSI amplitudes. Conversely, if the number of clusters decreases, the amount of challenge information generated decreases significantly, thereby weakening the security. Therefore, we designed a hashing model to increase the challenge information using a sliding window scheme. The training scheme of the feature extraction and clustering models is summarized in Algorithm 2.

In this algorithm, $S$ is the shift length of the sliding window method, $T$ is the number of sliding windows, and $\omega$ is the window length. $\theta$ is the number of dimensions for feature extraction, and $\phi$ is the number of clusters. Here, the trained feature extraction model and center points of the clustering model are stored in the IoT device and the server. UID is an unreusable (temporary) identity, with which the IoT device can verify the server. $L_{\text{ID}}$ is the ID length. Furthermore, $\boldsymbol{O}$ are permuted indices to increase the diversity of keys. When a new IoT device is installed in the wireless network, the server provides inherent permuted indices, which are generated randomly, to new IoT devices. Therefore, the server does not require much computation for model training to add IoT devices to the network. Fig. 7 shows the results of the feature

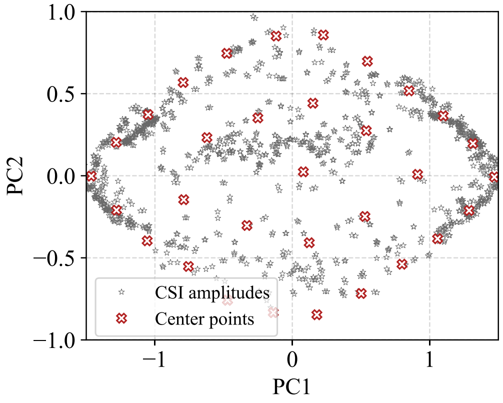


Fig. 7. Feature extraction results of CSI amplitudes ($K = 20\,000$, $\theta = 2$, and $\phi = 2^5$).

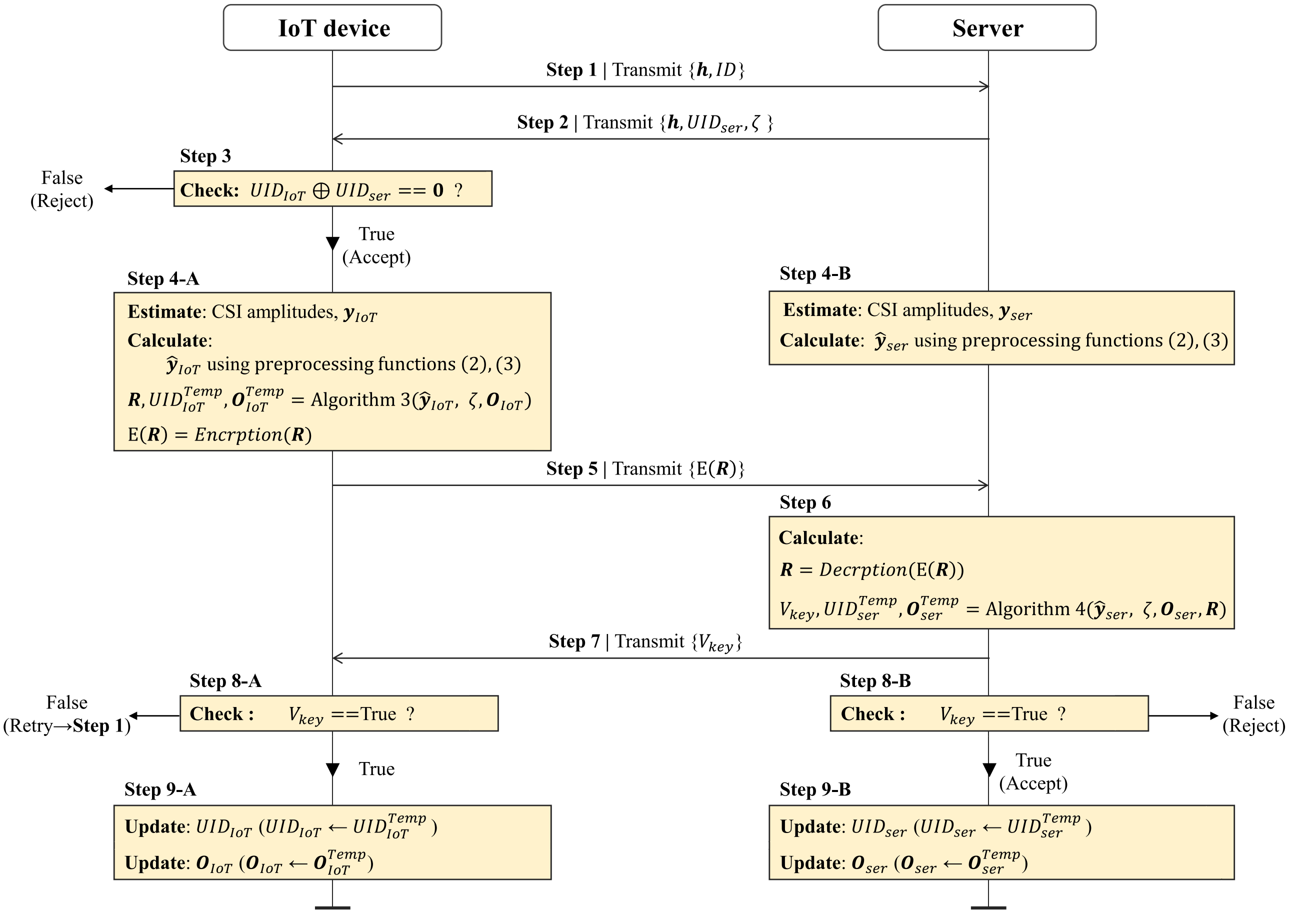

IoT device
Server
Step 1 | Transmit $\{\boldsymbol{h}, ID\}$
Step 2 | Transmit $\{\boldsymbol{h}, UID_{ser}, \zeta\}$
Step 3
Check: $UID_{IoT} \oplus UID_{ser} == \mathbf{0}$ ?
False (Reject)
True (Accept)
Step 4-A
Estimate: CSI amplitudes, $\boldsymbol{y}_{IoT}$
Calculate:
$\hat{\boldsymbol{y}}_{IoT}$ using preprocessing functions (2), (3)
$\boldsymbol{R}, UID_{IoT}^{Temp}, \boldsymbol{O}_{IoT}^{Temp}$ = Algorithm 3($\hat{\boldsymbol{y}}_{IoT}, \zeta, \boldsymbol{O}_{IoT}$)
E($\boldsymbol{R}$) = $Encrption(\boldsymbol{R})$
Step 4-B
Estimate: CSI amplitudes, $\boldsymbol{y}_{ser}$
Calculate: $\hat{\boldsymbol{y}}_{ser}$ using preprocessing functions (2), (3)
Step 5 | Transmit {E($\boldsymbol{R}$)}
Step 6
Calculate:
$\boldsymbol{R} = Decrption$(E($\boldsymbol{R}$))
$V_{key}, UID_{ser}^{Temp}, \boldsymbol{O}_{ser}^{Temp}$ = Algorithm 4($\hat{\boldsymbol{y}}_{ser}, \zeta, \boldsymbol{O}_{ser}, \boldsymbol{R}$)
Step 7 | Transmit $\{V_{key}\}$
Step 8-A
Check : $V_{key}$ ==True ?
False (Retry→Step 1)
True
Step 8-B
Check : $V_{key}$ ==True ?
False (Reject)
True (Accept)
Step 9-A
Update: $UID_{IoT}$ ($UID_{IoT} \leftarrow UID_{IoT}^{Temp}$)
Update: $\boldsymbol{O}_{IoT}$ ($\boldsymbol{O}_{IoT} \leftarrow \boldsymbol{O}_{IoT}^{Temp}$)
Step 9-B
Update: $UID_{ser}$ ($UID_{ser} \leftarrow UID_{ser}^{Temp}$)
Update: $\boldsymbol{O}_{ser}$ ($\boldsymbol{O}_{ser} \leftarrow \boldsymbol{O}_{ser}^{Temp}$)

Fig. 8. Proposed authentication protocol.

extraction. This result was tested at 150 locations and observed five times at each location.

### B. Authentication Phase

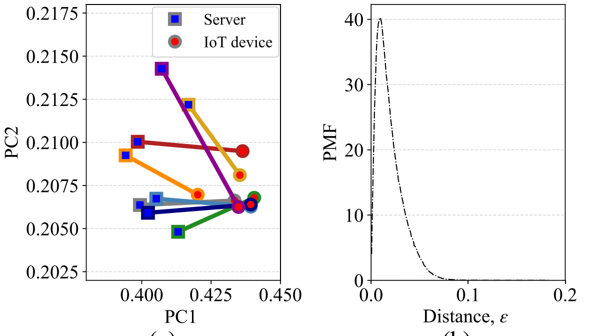
Our authentication method comprises a joint heterogeneous PUF-based key generation for the IoT device and a key verification scheme for the server. Fig. 8 shows the overall authentication process. First, the IoT device sends its identity *ID* with the pilot signal to estimate the CSI. The server receives them from the IoT device and then communicates the unreusable identity $\text{UID}_{\text{ser}}$, the random value $\zeta$, and the pilot signal. Thereafter, the IoT device verifies the $\text{UID}_{\text{ser}}$ of the server using its $\text{UID}_{\text{IoT}}$ to check whether the server is legitimate. The IoT device operates the PUF-based key generation scheme if the server is legitimate.

*1) Joint Heterogeneous PUF-Based Key Generation for the IoT Device:* The profile of CSI amplitudes was employed to generate the challenge information. The IoT device estimates the CSI amplitudes using the pilot signal $\boldsymbol{h}$ received from the server. The downsampling and normalization schemes are then used for the CSI amplitudes $\boldsymbol{y}$. Next, the challenge information is generated using the feature extraction and $K$-means clustering. A detailed description is provided in Algorithm 3.

**Algorithm 3:** Key Generation Scheme

```
input:  PCA(), $\hat{\boldsymbol{y}}, \boldsymbol{\mu}, \boldsymbol{O}, S, \omega, \phi, L_{\text{bit}}, L_{ID}, N_{\text{addr}}, \zeta, T$
output: $\boldsymbol{R}$, $UID$, $\boldsymbol{O}$
$L_{\text{b}} = \frac{L_{bit}}{T}$;
for $t = 1 : T$ do
    $\boldsymbol{d}$ = zeros[$\phi$];
    // PCA based K-means clustering
    $\boldsymbol{\chi} = \textbf{PCA}(\hat{\boldsymbol{y}}[(t-1)S+1 : (t-1)S+\omega])$;
    for $q = 1 : \phi$ do
        $d[q] = \sqrt{(\boldsymbol{\mu}[q,:] - \boldsymbol{\chi})^2}$;
    // Set random number generator
    $H[t] = \boldsymbol{o}_t[\text{argmin}_q \boldsymbol{d}] + \zeta$;
    seed($H[t]$); $\zeta = \zeta + \phi \times \boldsymbol{o}_t[\text{argmin}_q \boldsymbol{d}]$;
    // Generate the challenge information
    $\boldsymbol{C} \leftarrow$ randint(0, $N_{\text{addr}}, L_{\text{b}}$);
    // Map the Response of SRAM-PUF
    if $H[t]$ == odd number then
        $\boldsymbol{R}[(t-1)L_{\text{b}}+1 : tL_{\text{b}}] \leftarrow$ SRAM($\boldsymbol{C}$);
    else
        $\boldsymbol{R}[(t-1)L_{\text{b}}+1 : tL_{\text{b}}] \leftarrow \mathbf{1} -$ SRAM($\boldsymbol{C}$);
// Update the UID and labels of clusters
seed(concatenate($\boldsymbol{H}$));
$UID \leftarrow$ randint(0, 1, $L_{\text{ID}}$);
for $t = 1 : T$ do
    $\boldsymbol{o}_t \leftarrow$ permutations([1, 2, ..., $\phi$])
```

In this algorithm, $N_{\text{addr}}$ is the number of cells in the SRAM module and $L_{\text{bit}}$ is the key length. The CSI amplitudes are changed to the label of clusters through the PCA-based $K$-means clustering. Thereafter, a random seed $H[t]$ can be obtained using the permutation matrix $\boldsymbol{o}_t$ and a random value $\zeta$. If there are few CSI amplitude changes in a static



(a) (b)

Fig. 9. (a) PCA results of CSI amplitudes in the same location and (b) distance between the server and the IoT device.

environment, the adversary can mimic the key by accessing a legitimate device. Thus, we consider the random value $\zeta$ in the proposed method to adjust keys even in a static environment. Next, the random seed is fixed and then the challenge information is randomly generated. Moreover, we consider the uniformity of the generated keys. As shown in Fig. 3, the ratios of the 0-skewed and 1-skewed cells had approximately 30% and 50%, respectively. The number of 1-values was larger than that of 0-values when the key was generated randomly. Thus, we flipped the key according to the type of seed number (*odd* or *even*). Furthermore, the IoT device can encrypt the generated SRAM response using diverse authentication protocols, which can also be achieved by hiding the response information of a device PUF. After generating the PUF-based key, the temporary identity UID and the permutation matrix $\boldsymbol{O}$ are updated using a random seed $\boldsymbol{H}$. Whenever the server authenticates the legitimate device, UID is generated using a random seed value. Hence, the adversary cannot estimate UID because the random seed is not shared in the wireless network.

*2) Joint Heterogeneous PUF-Based Key Verification Scheme for the Server:* The server should generate the challenge information for the device PUF using PCA-based $K$-means clustering to verify the IoT device key. However, the challenge information of the server may be different from that of the IoT device owing to the uncertainty of CSI amplitudes. Fig. 9 shows the uncertainty of the CSI amplitudes. As shown in Fig. 9(a), the CSI amplitudes are different even at the same location, which affects the authentication results. Therefore, we analyzed the PCA distances between the server and IoT device in Fig. 9(b) and considered the authenticated boundary $\epsilon$ to mitigate the randomness of the CSI amplitudes. The detailed algorithm is described in Algorithm 4.

In this algorithm, the key generation scheme of the server is similar to that of the IoT device. To find the challenge information of the IoT device, the server derives all possible random seeds $\boldsymbol{U}$ within the $\epsilon$ range based on the PCA values of the server. Herein, the number of cases is $\upsilon$, and $\boldsymbol{H}$ is defined as the random seed set estimated to the IoT device's random seed set. If the PCA distance between the IoT device and the server is lower than $\epsilon$, the server can obtain the $\boldsymbol{H}$ of the IoT device. To decide $\boldsymbol{H}$, addresses of SRAM cells are generated randomly as the key length by applying all possible random seeds. Herein, CRP() is the mapping function that transforms the addresses (Challenge) to the probabilities $\boldsymbol{P}$ (Response) received from the vendor. The probability set $\boldsymbol{P_C}$ for the authentication is generated through the mapping function. Thereafter, the server verifies the PUF-based key $\boldsymbol{R}$ of the IoT device using the probability $\boldsymbol{P_C}$, as follows:

**Algorithm 4:** Key Verification for the Server

```
input:  $\hat{\boldsymbol{y}}, \boldsymbol{\mu}, \boldsymbol{O}, S, \omega, \phi, \epsilon, \tau, T, L_{\mathrm{bit}}, L_{\mathrm{ID}}, N_{\mathrm{addr}}, N_{\mathrm{sub}}, \zeta,$
        $\boldsymbol{PCA}(), \boldsymbol{R}$
output: $V_{key}$, $UID$, $\boldsymbol{O}$
$L_{\mathrm{b}} = \frac{L_{\mathrm{bit}}}{T}$; $\rho = 1$;  // Indicator for verifying
 the IoT device
for $t = 1 : T$ do
    $\upsilon = 0$ ;  // The number of possible all
     cases
    $U = []$ ;     // Seed set of possible all
     cases
    $\boldsymbol{d} = \mathrm{zeros}[\phi]$;
    // PCA based K-means clustering
    $\boldsymbol{\chi} = \mathbf{PCA}(\hat{\boldsymbol{y}}[(t-1)S+1:(t-1)S+\omega])$;
    for $q = 1 : \phi$ do
        $d[q] = \sqrt{(\boldsymbol{\mu}[q,:]-\boldsymbol{\chi})^2}$;
    // Derive possible all cases
    for $q = 1 : \phi$ do
        if $d[q] - \min_q \boldsymbol{d} \le \epsilon$ then
            $\upsilon = \upsilon + 1$; $U[\upsilon] \leftarrow \boldsymbol{o}_t[q] + \zeta$;
    $\boldsymbol{\varrho} = \mathrm{zeros}[\upsilon]$;
    for $u = 1 : \upsilon$ do
        // Set random number generator
        seed($U[u]$); $\boldsymbol{C} \leftarrow \mathrm{randint}(0, N_{\mathrm{addr}}, L_{\mathrm{b}})$;
        if $U[u]$ == odd number then
            $\boldsymbol{P_C}[(t-1)L_{\mathrm{b}}+1 : tL_{\mathrm{b}}] \leftarrow \mathrm{CRP}(\boldsymbol{C})$;
        else
            $\boldsymbol{P_C}[(t-1)L_{\mathrm{b}}+1 : tL_{\mathrm{b}}] \leftarrow \mathbf{1} - \mathrm{CRP}(\boldsymbol{C})$;
        $\boldsymbol{\varrho}[u] \leftarrow$ Calculating Equations (4) and (5);
    $H[t] \leftarrow U[\mathrm{argmax}_u \boldsymbol{\varrho}]$; $\zeta = \zeta + \phi \times (H[t] - \zeta)$;
    $\rho = \rho \times \max_u \boldsymbol{\varrho}$;
// Verify the PUF key of IoT device
if $\rho > \tau$ then
    $V_{key} \leftarrow$ True;
else
    $V_{key} \leftarrow$ False;
// Update the UID and the labels of
    clusters
seed(concatenate($\boldsymbol{H}$)); $UID \leftarrow \mathrm{randint}(0, 1, L_{\mathrm{ID}})$;
for $t = 1 : T$ do
    $\boldsymbol{o}_t \leftarrow$ permutations($[1, 2, \ldots, \phi]$)
```

$$\tilde{p}_c = \begin{cases} p_c + \frac{1}{2N_{\mathrm{SRAM}}}, & \text{if } p_c < 0.5 \\ p_c - \frac{1}{2N_{\mathrm{SRAM}}}, & \text{otherwise} \end{cases} \tag{4}$$

$$\rho = \prod_{c=1}^{L_{\mathrm{b}}} |\tilde{p}_c R_c + (1-\tilde{p}_c)(1-R_c)|, R_c \in \{0, 1\} \tag{5}$$

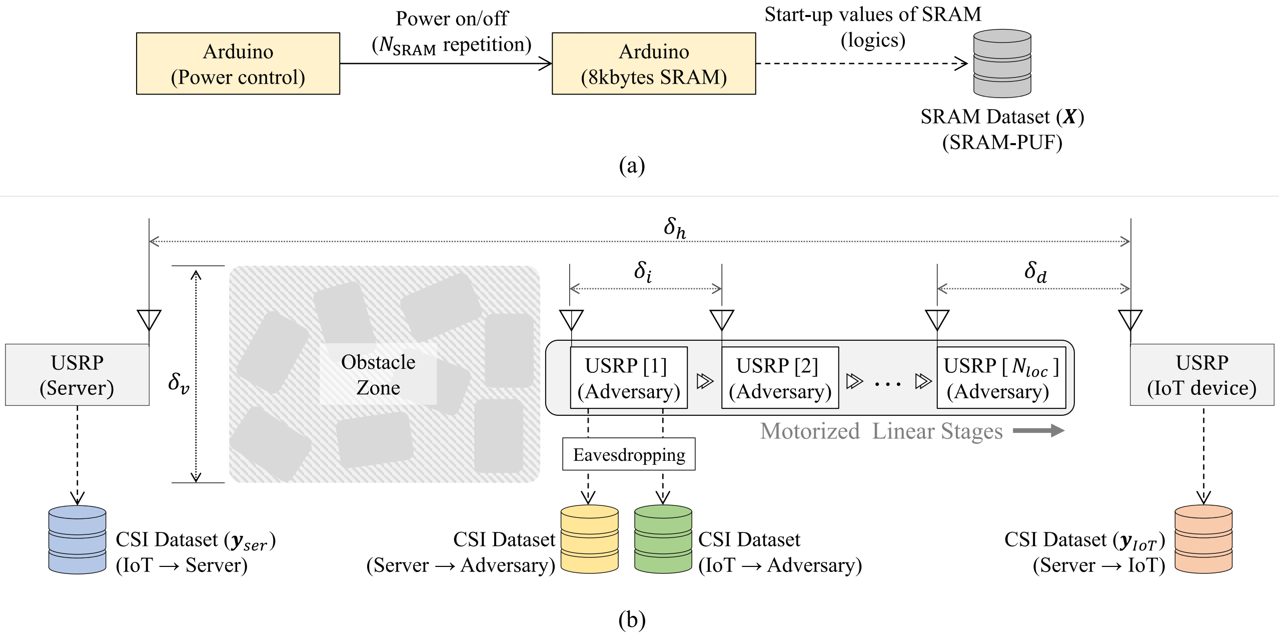


Fig. 10. Experimental setup for collecting data set. (a) CRP data set of the SRAM-PUF. (b) Wireless channel gains as CSI at different locations.

where $R_c$ is the response of the IoT devices and $c$ is the address of the SRAM cells selected for authentication. Our authentication method was designed such that the joint probability $\rho$ of the selected cells becomes approximately 1 when authenticating the legitimate device, as shown in (5). However, if the selected cell is a stable cell ($\tilde{p}_c = 0$ or 1), the joint probability can become 0. Therefore, (4) adds values less than $1/N_{\text{SRAM}}$ to prevent $\rho$ from becoming 0.

$\boldsymbol{H}$ is considered as the random seed that maximizes $\rho$. Moreover, $\rho$ is used as the verifier for the authentication because it denotes the similarity of CSI between the IoT device and the server. If the joint probability $\rho$ is higher than the threshold $\tau$, the server determines a legitimate IoT device. In addition, this method can precisely estimate the challenge information of legitimate IoT devices by controlling $\epsilon$. However, the false acceptance rate (FAR) of the adversary can increase as $\epsilon$ increases. After the key verification, the server sends the authentication results to the IoT device. If the authentication is accepted, the server and the IoT device update UID and the permutation matrix $\boldsymbol{O}$ using $\boldsymbol{H}$ for the next authentication.

## V. Experimental Environment

### A. Experimental Environment

To evaluate the proposed method, we collected the SRAM and CSI data sets using a testbed. Each process in the proposed method in the testbed was handled separately, as shown in Figs. 10 and 11.

*1) Experimental Setup for SRAM-PUF:* To collect the SRAM-PUF data set, the power-up states of the 8-kB SRAM of each Arduino Mega were observed 3000 times at room temperature, between 18 °C and 23 °C. We used a microcontroller for the power control to automatically observe the power-up states, as shown in Fig. 10(a) [39]. Since the Arduino used some cells of SRAM for the program execution, we selected $N_{\text{addr}}$ cells of the SRAM-PUF (8 kB) for the proposed scheme. $N_{\text{SRAM}}$ and $N_{\text{addr}}$ of Arduino Mega were set at 200 and 60 000, respectively, to calculate the probabilities $\boldsymbol{P}$.

*2) Experimental Setup for CSI Amplitudes:* To observe the CSI amplitudes, three USRPs were employed, as shown in Fig. 10(b). The center frequency of the LTE framework was set at 915 MHz, and the number of subcarriers $N_{\text{sub}}$ was 200 [40]. The CSI amplitude data set was collected 1000 times at the same location. The locations of a server and an IoT device were fixed, whereas the adversary moved in a motorized linear stage. $N_{\text{Loc}}$ was set at 50, while $\delta_i$ and $\delta_h$ were set at 2 cm (0.06$\lambda$) and 480 cm (14.64$\lambda$), respectively. The minimum distance $\delta_d$ between the adversary and the IoT device was set to 24 cm (0.73$\lambda$), and $\delta_v$ was set to 90 cm. In the obstacle zone, there were eight boxes, which induced multipath fading. Eight obstacles, which were plastic cube boxes of 33 cm × 33.3 cm × 35.7 cm, were placed between the server and the IoT device to mimic the actual environment. The data set of CSI amplitudes was shared in [41].

*3) Experimental Setup for Data Processing:* We employed two computation environments, which mimicked the server and the IoT device, as shown in Fig. 11. Two computation environments were deployed to evaluate the computation time: 1) IoT device—Quad-core Cortex-A72 1.5 GHz (Raspberry Pi) and 2) server—Intel Core i7-8550U 2.0 GHz. To generate fake CSI data, $\gamma_{\min}$ and $\gamma_{\max}$ was set to $N_{\text{sub}}/v$ and $(N_{\text{sub}}+20)/v$ in Algorithm 1, respectively. Also, the number of fake CSI data sets $K$ was set to 20 000. To develop the hashing model, $\epsilon$ was set at 0.20, and $\omega$, $S$ for the sliding window were 10 and 10, respectively. Furthermore, $w_D$ was set at 5 for the downsampling scheme. The number of clusters $\phi$ was set at $2^5$. $\zeta$ was randomly selected as an integer value between 0 and $2^{16}$ whenever the IoT device attempted to authenticate the server.

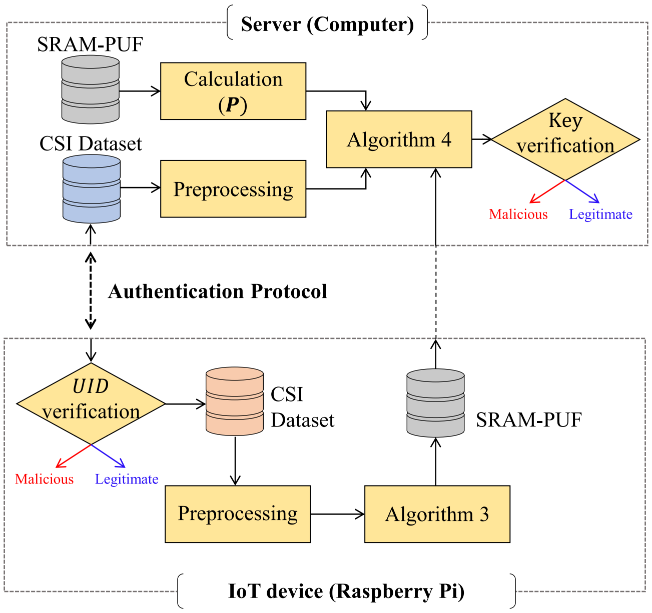


Fig. 11. Data processing structure for each process.

To precisely evaluate the performance of the joint heterogeneous PUF-based key, we assumed a more fragile situation wherein the encryption model was not considered.

### *B. Threat Model*

To evaluate the proposed scheme, a threat model was assumed in which the adversary could eavesdrop and manipulate the communicated data in the wireless network [42], [43]. The adversary authenticates to the server (or IoT device) instead of legitimate users. Therefore, the adversary can attempt to model the PUF construction using communicated data and impersonate the IoT device or server based on the proposed authentication protocol. If the adversary does not obtain sufficient data to estimate the PUF model, it can use the random key to obtain the control right of the legitimate user. Moreover, it can move closer to the IoT device and estimate the CSI amplitudes similar to that of the IoT device, considering the adversary is aware of the proposed authentication protocol. Then, the adversary can obtain the hashing model through an invasive attack in the NVM of an IoT device. Finally, it can generate the PUF-based key using the hashing model extracted from the legitimate device.

## VI. Numerical Evaluation

### *A. Evaluation of Joint Heterogeneous PUF-Based Key*

We evaluated the joint heterogeneous PUF-based key and verified the performance of the proposed PUF-based key based on four evaluation indicators: 1) CRPs space; 2) uniformity; 3) reliability; and 4) uniqueness.

*1) CRPs Space:* First, we evaluated the number of CRPs. As the number of CRPs increased, the security against modeling and replay attacks enhanced. Our proposed scheme generated the challenge information using CSI amplitudes and the random value $\zeta$. The response information was decided through the random seed generated by the challenge information. Therefore, the number of CRPs can be calculated based on the number of clusters $\phi$, number of sliding windows $T$, and the size of $\zeta$. In our experimental setup, the number of CRPs was approximately 6.9 $\times 10^{10}$ ($\phi^T \times 2^{16}$). If the size of $\zeta$ and the number of clusters increase, the CRPs space can be increased; however, the computation and communication costs increase. Therefore, these parameters should be controlled while considering the environment of the wireless network.

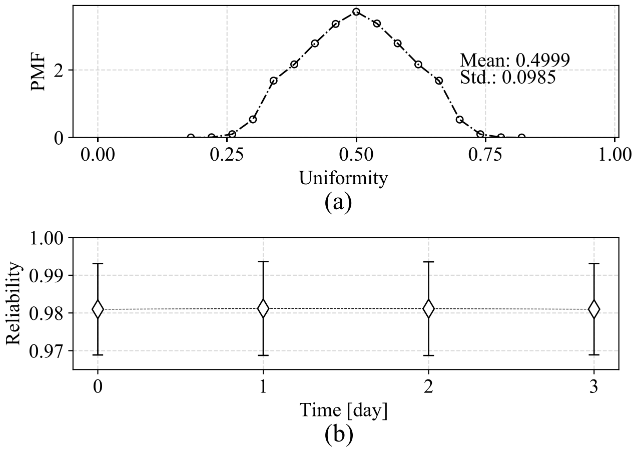


Fig. 12. Key evaluation of joint heterogeneous PUF: (a) uniformity and (b) reliability.

*2) Uniformity:* Uniformity means the ratios of zero and one in the PUF-based key. When the ratios of zero and one are the same, 0.5 is the ideal value. Uniformity is defined as follows:

$$\text{Uniformity} = \frac{1}{L_{\text{bit}}} \sum_{i=1}^{L_{\text{bit}}} r_i, r_i \in \{0, 1\} \tag{6}$$

where $r_i$ is $i$th bit of PUF-based key, which has the length of $L_{\text{bit}}$. The uniformity of joint heterogeneous PUF was calculated as shown in Fig. 12(a). The average and standard deviation of the uniformity was 0.4999 and 0.0985, respectively.

*3) Reliability:* Because the responses of PUF may be affected by the change in the physical environment, it is crucial to verify the reliability of the PUF-based key. In particular, the CRP table should be reliable to ensure the server can determine whether the security key is valid by only using the CRP table. Therefore, the reliability of the PUF-based key is expressed as follows:

$$\text{Reliability} = 1 - \frac{1}{L_{\text{bit}}} \sum_{i=1}^{L_{\text{bit}}} (\text{round}(p_i) - r_i) \tag{7}$$

where $p_i$ is $i$th cell's probability in the CRP table. In the proposed scheme, all SRAM-PUF cells are used to generate the PUF-based key. The reliability of the PUF-based key can deteriorate because of an unstable cell. As shown in Fig. 12(b), we evaluated the reliability of our proposed key 2500 times over four days in an indoor environment. The average result was 0.9811, and no significant difference was observed in reliability performance in four consecutive days.

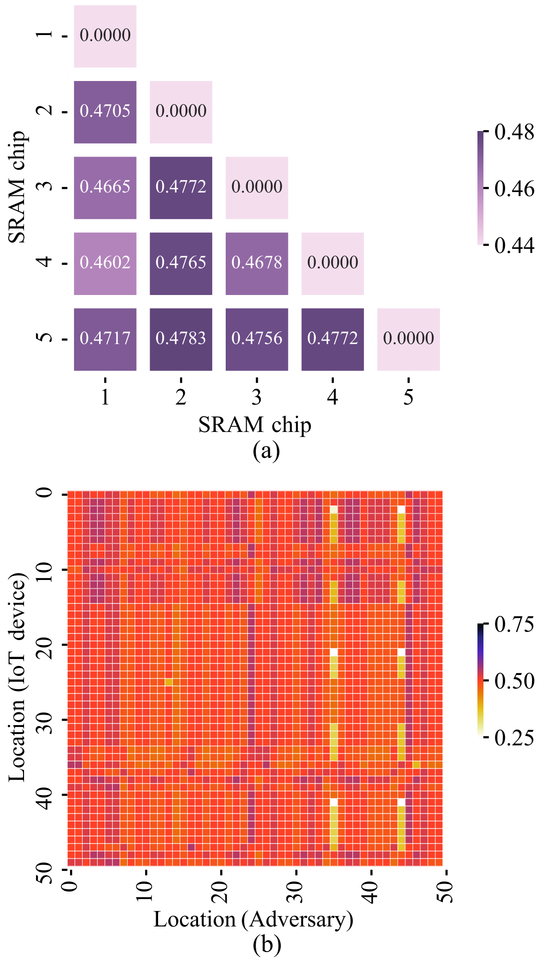


Fig. 13. Uniqueness evaluations for the proposed scheme: (a) interchips (SRAM) and (b) interlocations.

According to [28], environmental change (such as temperature and magnetic field) may cause additional degradation (approximately 0.01%–0.05%) of the reliability performance.

*4) Uniqueness:* Uniqueness evaluates the difference between the responses of several devices when the challenge information is the same. Uniqueness refers to an evaluation indicator in which an IoT device has a unique PUF design and is expressed as follows:

$$\text{Uniqueness} = \frac{1}{L_{\text{bit}}}\mathbf{HD}(\boldsymbol{R}_a, \boldsymbol{R}_b) \tag{8}$$

where $a$ and $b$ are indices of IoT devices, and **HD** is the hamming distance. As shown in Fig. 13(a), the uniqueness between interchips was evaluated using five SRAM chips. The average and standard deviation of uniqueness were 0.4721 and 0.006, respectively. Additionally, we evaluated the uniqueness of CSI amplitudes between interlocations in Fig. 13(b). In the proposed scheme, an adversary can move to the location of the legitimate user and estimate the CSI amplitudes. Therefore, an adversary can attempt to generate the PUF-based key using CSI amplitudes. To analyze the uniqueness between interlocations, the PUF-based keys were generated in 50 locations as shown in Fig. 10(b). The average and standard deviation of uniqueness were 0.497 and 0.03, respectively. The CSI amplitudes of the IoT device are not largely affected by the moving adversary. Therefore, the adversary can explore CSI amplitudes by relocating its location. When the adversary was close to the IoT device (Adversary's location = 36 or 45), the uniqueness was 0.36, as in Fig. 10(b). Thus, $\zeta$, which can update key values, is necessary for the proposed scheme in a static environment.

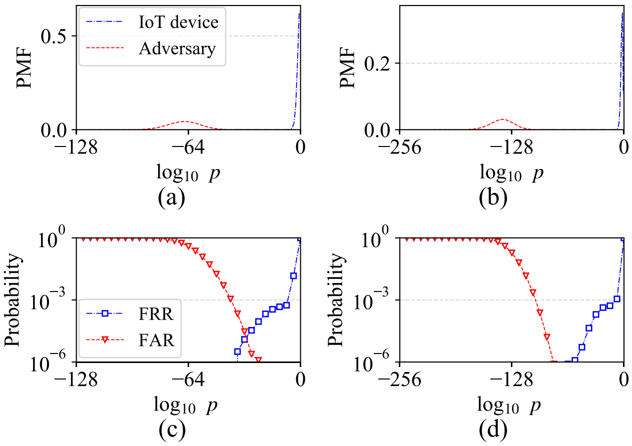


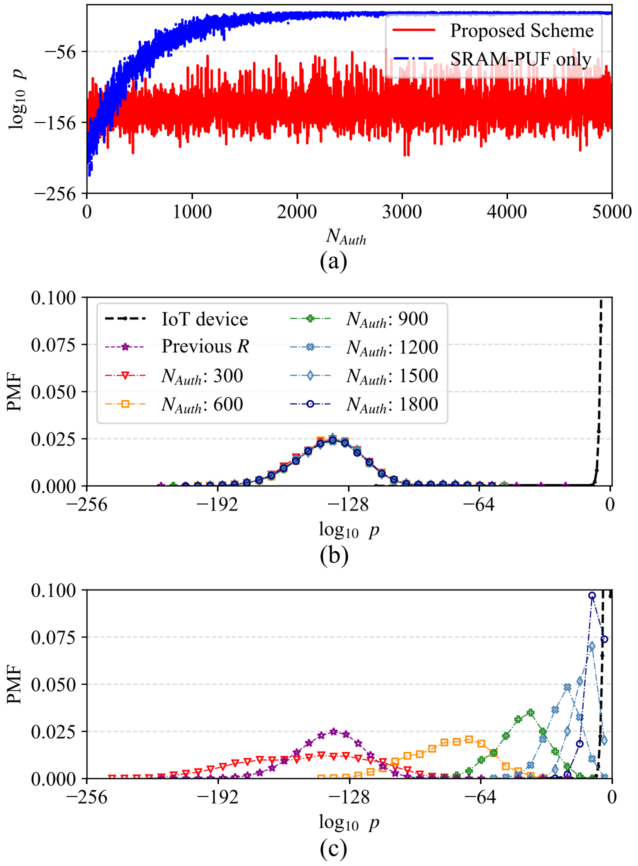
Fig. 14. Distribution of the joint probabilities: (a) $L_{\text{bit}}$: 64, (b) $L_{\text{bit}}$: 128, and security performances for brute-force attack, (c) $L_{\text{bit}}$: 64, and (d) $L_{\text{bit}}$: 128.

## *B. Attack Models*

We considered various attack models to evaluate the security of joint heterogeneous PUF-based authentication.

*1) Brute-Force Attack:* For a brute-force attack, it was assumed that an adversary could not estimate CSI amplitudes and PUF's construction. Then, an adversary can only employ random keys to impersonate the IoT device or server. As shown in Fig. 14, the security against the brute-force attack was evaluated. Fig. 14(a) and (b) show the distributions of joint probability $\rho$ calculated using (5), while Fig. 14(c) and (d) show the performance of the false rejection rate (FRR) and FAR. In general, security performance increases as the length of the PUF-based key increases. When $L_{\text{bit}}$ and $\tau$ were set at 128 and −74, the proposed scheme could block brute-force attacks 5 M times.

*2) Replay Attack:* The replay attack estimates the CRP table by sniffing legitimate packets. In conventional SRAM-PUF-based authentication, an adversary can acquire the challenge information (cell addresses) and response information (keys) in the wireless network. Conversely, the adversary in our proposed scheme can eavesdrop only $\zeta$ and the response information $\boldsymbol{R}$. It is challenging to map $\zeta$ to $\boldsymbol{R}$ considering $\zeta$ is part of the challenge information. As shown in Fig. 15, the results of the replay attack were evaluated, and $N_{\text{Auth}}$ indicates the amount of communicated data that the adversary eavesdropped on. Fig. 15(a) and (c) show that an adversary can generate the CRP table by eavesdropping if $N_{\text{Auth}}$ is larger than 2000 for the conventional SRAM-PUF-based authentication. As shown in Fig. 15(b), the CRP table of the proposed scheme cannot be estimated because CSI amplitudes are not shared through the wireless network.



(a)

(b)

(c)

Fig. 15. Security evaluation against the replay attack when the adversary eavesdrops on the communicated data for the authentication. (a) Joint probability $\rho$. (b) $\rho$ distribution of the proposed scheme. (c) $\rho$ distribution of the SRAM-PUF.

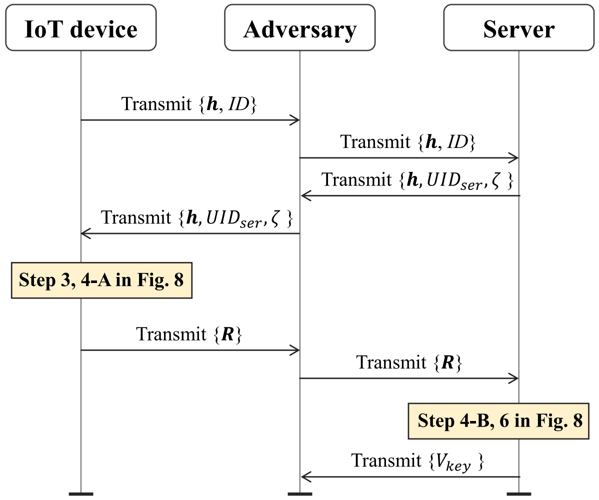

IoT device
Adversary
Server
Transmit $\{\boldsymbol{h}, ID\}$
Transmit $\{\boldsymbol{h}, ID\}$
Transmit $\{\boldsymbol{h}, UID_{ser}, \zeta\}$
Transmit $\{\boldsymbol{h}, UID_{ser}, \zeta\}$
Step 3, 4-A in Fig. 8
Transmit $\{\boldsymbol{R}\}$
Transmit $\{\boldsymbol{R}\}$
Step 4-B, 6 in Fig. 8
Transmit $\{V_{key}\}$

Fig. 16. MITM attack for the proposed scheme.

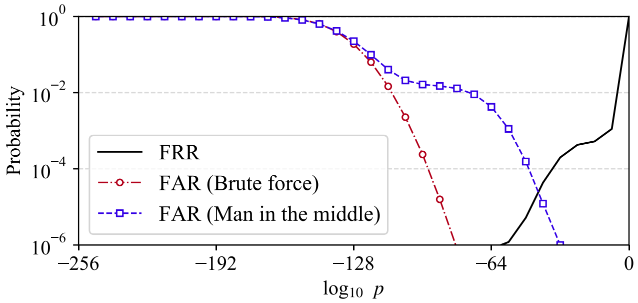


Fig. 17. Security performance for the MITM attack.

*3) Modeling Attack:* A modeling attack, which includes ML attacks, aims to model the PUF constructions for authentication. To attack the proposed scheme, the adversary should estimate two physical models: 1) device-PUF and 2) RF-PUF. However, it is difficult to model the SRAM-PUF through eavesdropping given that the challenge (cell addresses) and response (key values) information in the SRAM-PUF have low correlations. Additionally, the adversary cannot easily model the CSI amplitudes because they can be affected by the physical environment (e.g., multipath fading and scattering). The CSI amplitudes of the legitimate users can be changed even by the movement of the adversary. Therefore, the proposed scheme is robust against the modeling attack.

*4) Stolen-Verifier Attack:* Stolen-verifier attack impersonates the legitimate user by stealing the verifier from the storage device, e.g., NVM. To impersonate a legitimate user, an adversary should have **PCA**(), $\boldsymbol{\mu}$, $\boldsymbol{O}$, UID, and CRP for Algorithm 3. Even if an adversary steals them from a legal server, they cannot generate the PUF-based key for authentication considering $\boldsymbol{O}$ and UID are updated. As described in Algorithms 3 and 4, $\boldsymbol{O}$ and UID are changed by the random seed based on the CSI amplitudes whenever a legitimate user is authenticated with the server. Therefore, $(N_{\text{Auth}}-1)$th PUF-based key is different from $N_{\text{Auth}}$th PUF-based key. Hence, an adversary should repeatedly extract the verifier from legitimate device and server for attacking this authentication scheme; however, it is a challenging task.

*5) Man-in-the-Middle Attack:* A man-in-the-middle (MITM) attack intercepts data communicated for authentication and manipulates the messages between two legitimate users. As shown in Fig. 16, an adversary can attempt to be authenticated as a legitimate user. An adversary can impersonate the server using $\text{UID}_{\text{ser}}$ and $\zeta$ to acquire the PUF-based key of a legitimate user. However, an adversary cannot be authenticated in the proposed scheme considering the CSI amplitudes between the adversary and server differ from those between the adversary and the IoT device. Fig. 17 shows the security performance against an MITM attack. The security performance of the MITM attack is less than that of the brute-force attack since the adversary can obtain the CSI amplitudes by moving close to the IoT device up to $\delta_d$. The cross point of FRR and FAR appeared at $2.4 \times 10^{-5}$.

*6) Mutual Authentication:* The server and IoT device can be mutually authenticated in our proposed scheme. As shown in Fig. 8, an IoT device received $\text{UID}_{\text{ser}}$ from a server, and then $\text{UID}_{\text{IoT}}$ was compared with $\text{UID}_{\text{ser}}$. UID was updated using the CSI amplitudes whenever the IoT device succeeded in authentication. If $\text{UID}_{\text{ser}}$ and $\text{UID}_{\text{IoT}}$ are the same, an IoT device can verify whether a server is legal. Additionally, a server can decide whether the IoT device is a legitimate device using PUF-based key $\boldsymbol{R}$.

TABLE I
TIME CONSUMPTION FOR EACH PROCESS ($\phi = 2^5$ AND $T = 4$)

| Process | @ IoT device | @ Server |
|---|---|---|
| Model training | | |
| Feature extraction | - | 0.1656 s |
| Hashing model | - | 15.53 s |
| Key generation | | |
| Resampling | 0.101 ms | 0.036 ms |
| Normalization | 0.107 ms | 0.025 ms |
| Key generation | 2.116 ms | - |
| Key verification | - | 1.031 ms |

TABLE II
COMMUNICATION COST IN THE PROPOSED SCHEME

| Data type | @ IoT device | | @ Server |
|---|---|---|---|
| ID | 32 bits | → | - |
| $\{\zeta, UID_{ser}\}$ | - | ← | {16+32} bits |
| $R$ | 128 bits | → | - |
| $V_{key}$ | - | ← | 8 bits |
| Total | 160 bits | ⇌ | 56 bits |

### C. Evaluation of Time Consumption

We evaluated the time consumption for joint heterogeneous PUF. Table I shows the time consumption for each process based on the IoT device (Raspberry Pi) and server (a computer). In recent years, many researchers have studied RF-PUF-based security systems based on ML [24], [25]. These schemes are time consuming in terms of training the hashing model. However, our proposed scheme needs approximately 16 s to train considering that simple $K$-means clustering was used. Moreover, the server does not need much computation time for new IoT devices in the wireless network given that it generates $\boldsymbol{O}$ and UID for new IoT devices.

To evaluate the computation time for the IoT device, the CRP table was employed instead of the SRAM chip because we cannot obtain an IoT device that controls the built-in SRAM chip. The total time for key generation was 1.72 ms, and it could be controlled according to $\phi$ and $T$ in Algorithm 3. As $\phi$ and $T$ increase, the security of the PUF-based key increases; however, the computation time may increase. Therefore, these parameters can be handled considering the specification of the IoT device.

### D. Evaluation of Communication Cost

The results of evaluating the communication cost for joint heterogeneous PUF are shown in Table II. The communication cost of the pilot signal was not considered. For authentication, an IoT device sends ID, $\boldsymbol{R}$ to the server. The size of $\boldsymbol{R}$ can be changed depending on the length of keys $L_{\text{bit}}$. Additionally, the server sends $\zeta$, UID$_{\text{ser}}$, $V_{key}$ to the IoT device. In our experimental setup, $\zeta$ was set at 16 bits, and the length of UID was set at 32. In general, the authentication based on SRAM-PUF communicates the challenge information (cell addresses) to the IoT device. For example, the challenge information needs 2048 bits to generate the responses of 128 bits when the address information has 16 bits. However, our proposed scheme does not need to send the challenge information considering that it is generated using the CSI amplitudes. Our proposed scheme can reduce the energy for communication effectively, compared with the conventional PUF-based authentication.

TABLE III
STORAGE COST IN THE PROPOSED SCHEME

| Device | Data (Variable) | Size [kB] |
|---|---|---|
| IoT device | $\mathbf{PCA}() + \boldsymbol{\mu} + \boldsymbol{O} + UID$ | 1.24 |
| Server | $\mathbf{PCA}() + \boldsymbol{\mu}$<br>$+(\boldsymbol{O} + UID + \boldsymbol{P})N_D$ | $1.11 + 58.72N_D$ |

### E. Evaluation of Storage Space

To execute our proposed scheme, the IoT device needed a feature extraction model and hashing model, which has approximately 1.2 kB, as shown in Table III. The PCA model accounted for most of the data size, and the model size can be changed by the window length $\omega$ and the number of dimensions $\theta$. The storage capacity of IoT devices varies depending on the type and functionality, ranging from hundreds of kilobytes to hundreds of megabytes. For example, the flash memory of a microcontroller has typically 128 kB [44], and the data size required for our proposed scheme was approximately 0.9% of the flash memory. Thus, our proposed scheme can be applied to the IoT devices.

The server stores the permutation matrix $\boldsymbol{O}$, CRP table, and UID whenever a new IoT device is registered in the wireless network. Compared to SRAM-PUF-based authentication [28], the server should additionally store $\boldsymbol{O}$ and UID, as many as the number of IoT devices $N_D$. Moreover, the data size of $\boldsymbol{P}$ is larger than the CRP table (1–8 kB) of general SRAM-PUF because the probabilities $\boldsymbol{P}$ are not binary values. However, Our proposed scheme, which hides the challenge information, can notably enhance the security of the IoT device, similar to the strong PUF. The strong PUF generally needs more storage space, over 100 kB [45], but our proposed approach can further reduce the storage space compared with the strong PUF.

## VII. DISCUSSION

In this study, we investigated a mutual authentication scheme between a single IoT device and an authentication server. We evaluated the PUF-based key and the security performance against diverse attacks. Our PUF-based key considered four factors: 1) CRP space; 2) uniformity; 3) reliability; and 4) uniqueness. The uniformity and reliability were mainly affected by the device-PUF, and uniqueness degraded owing to characteristics of the RF-PUF when the adversary was close to a legitimate device. However, our proposed scheme was able to secure a large amount of CRP space compared to conventional SRAM-PUF having $N_{\text{addr}}$ CRPs.

Moreover, our proposed scheme is robust against replay and ML attacks since the CRP of the device-PUF is not exposed during wireless communication. The CRP table of the conventional SRAM-PUF was estimated when the adversary could eavesdrop on the CRP table over 2000 times during wireless communication. On the other hand, our proposed scheme could protect the CRP table of the device-PUF because the adversary could only obtain the response signal. Although an adversary can coincidentally obtain the same CSI amplitudes, the adversary cannot continuously observe the precise CRP tables because the CSI amplitudes are changed over time.

Furthermore, our approach does not need additional model training when new IoT devices are added to the wireless network. Therefore, the server can easily enroll many IoT devices. Also, the energy consumption for communication is reduced because the challenge information is not shared through wireless communication.

In future work, we plan to expand our proposed scheme in two directions, broadening the scope and potential applications. First, we intend to develop a lightweight, integrated model that combines feature extraction and hashing models, using deep learning for an effective approach. It must be able to optimize sensitivity considering the uncertainty of the RF-PUF and the FAR of the adversary. Some researchers have been conducting studies aimed at enhancing PUF-based authentication performance by incorporating deep learning [46]. Second, diverse peer-to-peer applications, such as energy trading schemes in the smart grid, have been recently developed based on decentralized systems, and PUF-based authentication schemes have been studied with decentralized systems [47], [48]. Therefore, our proposed method will be enlarged for the peer-to-peer authentication scheme, which can update the PUF-based key based on the RF-PUF.

## VIII. Conclusion

This study proposed the application of a joint heterogeneous PUF-based authentication scheme to IoT authentication to enhance security in IoT networks. To hide the CRP in the wireless communication, the joint heterogeneous PUF combines the device PUF and RF-PUF. The RF-PUF of the server and IoT devices can have similar values considering both parties share the wireless channel. The challenge information of the device PUF is generated using the RF-PUF and is not shared through wireless communication. Moreover, the mutual authentication protocol and the feature extraction-based hashing model were designed considering the uncertainty of the RF-PUF.

To achieve this, we acquired an actual data set using USRP and Arduino, and the characteristics of the joint heterogeneous PUF were evaluated, including uniformity, reliability, and uniqueness. For the numerical evaluation of security performance, a testbed was designed using Raspberry Pi, and both FAR and FRR achieved 0% when the adversary could not estimate the CSI. Furthermore, our proposed scheme blocked the eavesdropping replay attack. Even with the MITM attacks from close physical proximity, FRR and FAR can be as low as $2.4 \times 10^{-5}$. Consequently, we validated the proposed method on a real-world testbed and found that it could serve as a robust security solution for mitigating the CRP exposure problem, a known vulnerability of PUF.

## References

[1] "80+ amazing IoT statistics (2023–2030)." Exploding Topics. Mar. 2023. [Online]. Available: https://explodingtopics.com/blog/iot-stats

[2] B. A. Salau, A. Rawal, and D. B. Rawat, "Recent advances in artificial intelligence for wireless Internet of Things and cyber–physical systems: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12916–12930, Aug. 2022.

[3] K. Sha, W. Wei, T. A. Yang, Z. Wang, and W. Shi, "On security challenges and open issues in Internet of Things," *Future Gener. Comput. Syst.*, vol. 83, pp. 326–337, Jun. 2018.

[4] S.-R. Oh and Y.-G. Kim, "Security requirements analysis for the IoT," in *Proc. Int. Conf. Platform Technol. Serv. (PlatCon)*, 2017, pp. 1–6.

[5] M. B. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.

[6] Y. Bai and Z. Yan, "A novel key generation scheme using quaternary PUF responses and wiretap polar coding," *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2142–2145, Jul. 2021.

[7] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107593.

[8] M. N. I. Khan, S. Bhasin, B. Liu, A. Yuan, A. Chattopadhyay, and S. Ghosh, "Comprehensive study of side-channel attack on emerging non-volatile memories," *J. Low Power Electron. Appl.*, vol. 11, no. 4, p. 38, 2021.

[9] A. R. Korenda, F. Afghah, B. Cambou, and C. Philabaum, "A proof of concept SRAM-based physically unclonable function (PUF) key generation mechanism for IoT devices," in *Proc. 16th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, 2019, pp. 1–8.

[10] N. Shah, D. Chatterjee, B. Sapui, D. Mukhopadhyay, and A. Basu, "Introducing recurrence in strong PUFs for enhanced machine learning attack resistance," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 11, no. 2, pp. 319–332, Jun. 2021.

[11] S. Kumar and M. Niamat, "Machine learning based modeling attacks on a configurable PUF," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, 2018, pp. 169–173.

[12] K. Hassan, M. Masarra, M. Zwingelstein, and I. Dayoub, "Channel estimation techniques for millimeter-wave communication systems: Achievements and challenges," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1336–1363, 2020.

[13] S. Chen, B. Li, Z. Chen, Y. Zhang, C. Wang, and C. Tao, "Novel strong-PUF-based authentication protocols leveraging Shamir's secret sharing," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14408–14425, Aug. 2022.

[14] P. Gope, B. Sikdar, and O. Millwood, "A scalable protocol level approach to prevent machine learning attacks on physically unclonable function based authentication mechanisms for Internet of Medical Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1971–1980, Mar. 2022.

[15] J. Zhang, C. Shen, Z. Guo, Q. Wu, and W. Chang, "CT PUF: Configurable tristate PUF against machine learning attacks for IoT security," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14452–14462, Aug. 2022.

[16] F. Zerrouki, S. Ouchani, and H. Bouarfa, "PUF-based mutual authentication and session key establishment protocol for IoT devices," *J. Ambient Intell. Humanized Comput.*, to be published.

[17] C. Gu, C.-H. Chang, W. Liu, S. Yu, Y. Wang, and M. O'Neill, "A modeling attack resistant deception technique for securing lightweight-PUF-based authentication," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1183–1196, Jun. 2021.

[18] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multisampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6786–6799, Aug. 2019.

[19] Z. Zhang, X. Guo, and Y. Lin, "Trust management method of D2D communication based on RF fingerprint identification," *IEEE Access*, vol. 6, pp. 66082–66087, 2018.

[20] X. Zhou, A. Hu, G. Li, L. Peng, Y. Xing, and J. Yu, "Design of a robust RF fingerprint generation and classification scheme for practical device identification," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2019, pp. 196–204.

[21] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "CRISLoc: Reconstructable CSI fingerprinting for indoor smartphone localization," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3422–3437, Mar. 2021.

[22] Q. Li, X. Liao, M. Liu, and S. Valaee, "Indoor localization based on CSI fingerprint by siamese convolution neural network," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12168–12173, Nov. 2021.

[23] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, Feb. 2019.

[24] W. Wu, S. Hu, D. Lin, and Z. Liu, "DSLN: Securing Internet of Things through RF fingerprint recognition in low-SNR settings," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3838–3849, Mar. 2022.

[25] K. Sankhe et al., "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, Mar. 2020.

[26] M. F. Bari, B. Chatterjee, K. Sivanesan, L. L. Yang, and S. Sen, "High accuracy RF-PUF for EM security through physical feature assistance using public Wi-Fi dataset," in *Proc. IEEE MTT-S Int. Microw. Symp. Dig.*, 2021, pp. 108–111.

[27] Y. Su, Y. Gao, M. Chesser, O. Kavehei, A. Sample, and D. C. Ranasinghe, "SecuCode: Intrinsic PUF entangled secure wireless code dissemination for computational RFID devices," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 4, pp. 1699–1717, Jul./Aug. 2021.

[28] F. Farha, H. Ning, K. Ali, L. Chen, and C. Nugent, "SRAM-PUF-based entities authentication scheme for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5904–5913, Apr. 2021.

[29] S. K. Cherupally, S. Yin, D. Kadetotad, C. Bae, S. J. Kim, and J.-S. Seo, "A smart hardware security engine combining entropy sources of ECG, HRV, and SRAM PUF for authentication and secret key generation," *IEEE J. Solid-State Circuits*, vol. 55, no. 10, pp. 2680–2690, Oct. 2020.

[30] P. Gope and B. Sikdar, "A reconfigurable and secure firmware updating framework for advanced metering infrastructure," in *Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids (SmartGridComm)*, 2022, pp. 453–459.

[31] P.-S. Yeh, C.-A. Yang, Y.-H. Chang, Y.-D. Chih, C.-J. Lin, and Y.-C. King, "Self-convergent trimming SRAM true random number generation with in-cell storage," *IEEE J. Solid-State Circuits*, vol. 54, no. 9, pp. 2614–2621, Sep. 2019.

[32] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, Sep. 2008.

[33] M.-S. Kim et al., "Error reduction of SRAM-based physically unclonable function for chip authentication," *Int. J. Inf. Security*, to be published.

[34] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surveys* , vol. 46, no. 2, pp. 1–32, 2013.

[35] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2015, pp. 1666–1671.

[36] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using WiFi channel state information," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 98–104, Oct. 2017.

[37] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications With MATLAB*. Hoboken, NJ, USA: Wiley, 2010.

[38] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[39] P. P. Aung, K. Mashiko, N. B. Ismail, and O. C. Yee, "Evaluation of SRAM PUF characteristics and generation of stable bits for IoT security," in *Proc. Int. Conf. Rel. Inf. Commun. Technol.*, 2019, pp. 441–450.

[40] S. Han, Y. Lee, J. Choi, and E. Hwang, "Lightweight physical layer aided key agreement and authentication for the Internet of Things," *Electronics*, vol. 10, no. 14, p. 1730, 2021.

[41] S. Yoon, S. Han, and E. Hwang. "HeteroPUF." 2023. [Online]. Available: https://github.com/GIST-IIS/HeteroPUF

[42] M. F. Ayub, M. A. Saleem, I. Altaf, K. Mahmood, and S. Kumari, "Fuzzy extraction and PUF based three party authentication protocol using USB as mass storage device," *J. Inf. Security Appl.*, vol. 55, Dec. 2020, Art. no. 102585.

[43] U. Chatterjee et al., "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 3, pp. 424–437, May/Jun. 2019.

[44] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, "Secure firmware updates for constrained IoT devices using open standards: A reality check," *IEEE Access*, vol. 7, pp. 71907–71920, 2019.

[45] A. Ali-Pour, D. Hely, V. Beroulle, and G. Di Natale, "Strong PUF enrollment with machine learning: A methodical approach," *Electronics*, vol. 11, no. 4, p. 653, 2022.

[46] O. Millwood, J. Miskelly, B. Yang, P. Gope, E. Kavun, and C. Lin, "PUF-phenotype: A robust and noise-resilient approach to aid intragroup-based authentication with DRAM-PUFs using machine learning," 2022, *arXiv:2207.04692*.

[47] Y. Zheng, W. Liu, C. Gu, and C.-H. Chang, "PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications," *IEEE Trans. Dependable Secure Comput.*, early access, Jul. 25, 2022, doi: 10.1109/TDSC.2022.3193570.

[48] A. Zahoor et al., "An access control scheme in IoT-enabled smart-grid systems using blockchain and PUF," *Internet Things*, to be published.

**Seungwook Yoon** (Student Member, IEEE) received the B.S. degree from the Department of Electric Engineering, Kwangwoon University, Seoul, South Korea, in 2014. He is currently pursuing the integrated M.S. and Ph.D. degrees with the School of Mechatronics, Gwangju Institute of Science and Technology, Gwangju, South Korea.

His research interests include energy informatics, vehicle grid integration, and physical unclonable function-based authentication.

**Seungnam Han** (Graduate Student Member, IEEE) received the B.E. degree in mechanical engineering from Chungnam National University, Daejeon, South Korea, in 2019, and the M.S. degree in electrical engineering and computer science from the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2021, where he is currently pursuing the Ph.D. degree.

His research interests include signal processing, wireless security, physical layer, and software-defined radio.

**Euiseok Hwang** (Senior Member, IEEE) received the bachelor's and master's degrees from Seoul National University, Seoul, South Korea, in 1998 and 2000, respectively, and the master's and Doctoral degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2010 and 2011, respectively.

He is an Associate Professor with the School of Electrical Engineering and Computer Science (EECS), Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. He worked for the Digital Media Research Center, Daewoo Electronics Company, Ltd., Gwangju, from 2000 to 2006 and for Channel Architecture Group of Data Controller Division, LSI (currently Broadcom), San Jose, CA, USA, from 2011 to 2014. Since 2015, he has been an Assistant Professor/an Associate Professor with the School of EECS/AI Graduate School/School of Mechatronics, GIST. From 2021 to 2022, he worked as a Visiting Scholar with the Department of Computer Science and Engineering, University of Michigan Ann Arbor, Ann Arbor, MI, USA. His research interests include statistical signal processing, machine learning, and channel coding for data storage and communication systems, and their emerging information and communication technology applications, such as Internet of Things and smart grids.