Received June 19, 2021, accepted June 28, 2021, date of publication July 7, 2021, date of current version July 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095348

Two-Factor Device DNA-Based Fuzzy Vault for Industrial IoT Device Security

EUNGI HONG[1], (Student Member, IEEE), SANGJAE LEE[2], MI-KYUNG OH[2], AND SEUNG-HYUN SEO[1], (Member, IEEE)

[1]Division of Electrical Engineering, Hanyang University at ERICA, Ansan, Gyeonggi 15588, Republic of Korea
[2]Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea

Corresponding author: Seung-Hyun Seo (seosh77@hanyang.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant through the Korean Government by the Ministry of Science and ICT (MSIT) (Development on Autonomous Trust Enhancement Technology of Internet of Things (IoT) Device and Study on Adaptive IoT Security Open Architecture based on Global Standardization [TrusThingz Project]) under Grant 2018-0-00230, and in part by the MSIT, Korea, through the Information Technology Research Center (ITRC) supervised by the Institute for Information and Communications Technology Promotion (IITP) under Program IITP-2021-2018-0-01417.

ABSTRACT The benefit of a smart manufacturing Industrial Internet of Things (IIoT) platform is that it can provide real-time monitoring, accurate analysis, and reporting for equipment by collecting data throughout the whole manufacturing facility. However, the increased internet connectivity of manufacturing machines or devices leads to various security vulnerabilities. In order to securely operate smart manufacturing IIoT systems in unmanned environments, it is necessary to establish a cryptographic key for protecting exchanged data between IIoT devices and stored data in the devices by using cryptographic algorithms. Especially, since the IIoT system is in an unmanned environment, the following two challenges must be solved: 1) The IIoT device must recover its own secret key without user interaction. 2) The IIoT device must prevent secret key recovery when anomaly situations such as unauthorized physical access occur. In this paper, we present a novel method to protect an IIoT device's secret key in unmanned smart manufacturing environments, called Two-Factor Device DNA-based Fuzzy Vault scheme. To satisfy the two challenges, our proposed method generates a specific two-factor device DNA through the combination of the IIoT device's intrinsic factor and its surrounding environments and then creates a vault set to conceal the secret key based on the two-factor device DNA. We also implement a prototype for ensuring the feasibility of our method by utilizing an EPUF and IEEE 802.15.4g receiver in a Raspberry Pi and a laptop, respectively, and then measure their performance. We then conduct experiments in an unmanned environment at the Smart Manufacturing Learning Center at Hanyang University by considering various normal and abnormal situations. Our experiment results show that the proposed method quickly extracts the secret key stored in the device in normal cases, but fails at key extraction in abnormal cases.

INDEX TERMS Received signal strength (RSS), device DNA, physical unclonable function (PUF), fuzzy vault.

I. INTRODUCTION

The rapid growth of Internet of Things (IoT) has caused it to spread to the industrial sector, creating the Industrial Internet of Things (IIoT), which connects machines, advanced analytical technologies, and workers. In particular, by utilizing IIoT in smart manufacturing systems, each part of the manufacturing process can be efficiently and automatically analyzed, and the maintenance cost can be significantly reduced.

However, IIoT inherits the security issues already present in IoT as well as the lack of security built in to the Industrial Control System (ICS) devices themselves. Moreover, increased device connectivity and more aggregation of data in IIoT can lead to increased security vulnerabilities. This is the reason why the Telecommunication Standardization Sector of ITU (International Telecommunication Union), ITU-T [1] requires that smart manufacturing systems protect production

The associate editor coordinating the review of this manuscript and approving it for publication was Weizhi Meng.

TABLE 1. Comparison of the recent related studies for IIoT security.

Ref #	Cryptographic primitive	Security service	Target application area
[2]	Keyword searchable encryption	Data confidentiality, secure data sharing	IIoT data Management in cloud server
[3]	Certificateless keyword searchable encryption		
[4]	Public key based searchable encryption		
[5]	Tensor based homomorphic encryption		
[6]	Symmetric key encryption, Hash algorithm	Device Authentication	IIoT healthcare
[7]	Symmetrica key encryption, Hash algoritm	Device Authentication	M2M communication for IIoT
[8]	Fuzzy extractor, hash algorithm symmetric key encryption	Privacy preservation of data	Cloud-based IIoT deployment
[9]	Identity-based Signcryption	Data confidentiality and authentication	
[10]	Secret sharing scheme	Data confidentiality and secure key sharing	M2M communication for IIoT

facilities, communication networks and data from misuse and unauthorized access. In this paper, we are focusing on the third, data protection. Cryptographic techniques such as encryption algorithms are used to achieve confidentiality and integrity of data and to ensure that only authorized entities share data. In order to protect data from abuse and theft, the data exchanged by the IIoT devices should be encrypted using secret keys previously known to the legitimate devices. However, if the secret key is stored unprotected on the device, the key may leak due to unauthorized remote or physical access, as well as malware infection. So, the most critical thing in using cryptography for securing data is to protect cryptographic keys stored in IIoT devices. Specifically, the smart manufacturing system operating in an unmanned environment must take measures to protect the IIoT device's secret key in preparation for the case when unauthorized physical access occurs.

A. MOTIVATION AND CHALLENGES

So far, only a few research studies into IIoT security [2]–[10] such as searchable encryption, which allows searching and accessing data while maintaining data confidentiality in IIoT cloud infrastructure, homomophic encryption and signcryption for IToT data confidentiality, and symmetric key based authentication schemes for resource constrained IIoT devices have been conducted as shown in Table 1. Nonetheless, these studies on securing IIoT did not consider how to protect the cryptographic key stored in the IIoT device. Also, a number of key management systems for existing industrial environments [13] have been proposed, but these works did not consider the diverse deployment challenges in smart manufacturing systems of IIoT environments or cover securely protecting cryptographic keys.

To protect the secret key used for authentication and encryption between IIoT devices operating in an unmanned smart manufacturing environment, the following challenges must be solved: *The IIoT device must be able to recover its own secret key without user interaction in unmanned environments.*

To achieve this challenge, the IIoT device must use its intrinsic factor, that is an element which can be created by the device itself for device authentication, like biometric information for user authentication. As one major type of intrinsic factor, physical unclonable functions (PUFs) generate device-specific characteristics unique to a semiconductor or device hardware, which are derived from the manufacturing process [16]. Using these characteristics, the PUFs can work as a device DNA, because each device is identified based on its physical characteristics, like human DNA [17]. So, PUFs have been commonly used in IoT devices to generate and maintain secret credentials for these reasons. However, if only the intrinsic factor is used to construct IIoT device security, the cryptographic keys or sensitive data stored in the devices cannot be protected from leak or theft in unmanned environments.

Therefore, we have to also consider the following challenge: *The IIoT device must prevent secret key recovery when anomaly situations such as unauthorized access occur.*

To achieve this second challenge, it is necessary to build the security of IIoT devices in unmanned environments using an environment factor together with an intrinsic factor. Most recently, Choi *et al.* [11] firstly proposed a two-factor fuzzy commitment technique by using both PUFs as an intrinsic factor and the surrounding area image data as an environment factor to protect the secret key of an IoT device in unattended environments, such as a surveillance situation. Yet, their scheme cannot be directly applied to the smart manufacturing system in unmanned IIoT platforms. The fuzzy commitment technique is prone to order invariance, so it is sometimes impossible to recover the secret key when there is a slight change in the direction of the device which shifts the image. Also, since the surrounding area image data is vulnerable to changes in illumination, the keys may not be recovered even though it is a legitimate device in the same location. Moreover, a camera sensor is required to capture the image of the surrounding environment, so it is difficult to directly apply to all IIoT devices.

In April 2019, Amen *et al.* [15] proposed a two-factor authentication scheme using PUF and the characteristics of the wireless signal from the IoT devices. The purpose of their scheme is to provide a strong authentication of IoT devices by using two-factors such as the device's own characteristics (e.g. PUF) and the positioning information of the device (e.g. wireless signal) against various attacks including spoofing attacks and physical attacks. Their method is to perform the first mutual authentication between IoT Device and Server with Challenge-Response Pairs created in PUF, and the second authentication is carried out through the positioning mechanism using RSS and LQI (Link Quality Indicator). Through this way, the IoT device is able to verify its identity and the fact of its proper location. However, their method only provides authentication service by using PUF and RSS, which can be used to verify identity and location. It does not provide a secret key protection of the IIoT device.

B. CONTRIBUTIONS

In this paper, in order to enhance the security of IIoT devices in smart manufacturing systems, we first propose the concept of two-factor device DNA, which is generated through the interaction of the device with the surrounding environment. The two-factor device DNA is a unique combination of environment and device and varies depending on location. Secondly, we design the two-factor device DNA generator, which utilizes the Extended PUF (EPUF) as the intrinsic factor and the RSS (Received Signal Strength) of the signal emitted by other IIoT devices in the same network as the environment factor. Unlike existing PUFs which use PUF-dedicated ASIC, EPUF has the advantage that it can be implemented without extra hardware because it can generate device DNA by using SRAM, Flash, or an RC(resistor-capacitor) circuit embedded in IIoT devices. In this paper, we utilize EPUF using an RC circuit, called RC PUF [42], to generate a two-factor device DNA.

Our two-factor device DNA generator inputs the RSS data acquired only at the location of the device as the challenge of the EPUF, and generates the two-factor device DNA using the response that only the EPUF outputs. Since the strength of the wireless signal is affected by the effects of diffraction, attenuation, etc. on the surrounding environment, the RSS may be used as the environment factor that informs the position of the device. IIoT devices, being connected to a wireless network, can easily obtain the RSS if an AP (Access Point) is installed in the smart manufacturing system. Therefore, in smart manufacturing systems, RSS works better as an environment factor than the surrounding area image (used in [11]) does. However, RSS data acquired by IIoT devices is difficult to obtain constantly due to unpredictable noise and movement of surrounding manufacturing equipment. So, we perform noise reduction on RSS data by utilizing an LPF. This LPF allows the IIoT device to obtain RSS data with a constant value by removing normal noise, but it is not able to remove abnormal noise, such as the kind generated by an attacker physically approaching the device. Therefore, since the noise generated in this kind of abnormal situation cannot be removed with the parameters of the LPF set as in the normal situation, the IIoT device cannot acquire the proper RSS data, thus preventing secret key recovery.

Finally, we propose a two-factor device DNA based fuzzy vault scheme that provides the following security benefits to meet the two challenges of having a unique intrinsic factor and being able to prevent secret key recovery when anomalies occur.

- Since the RSS data acquired by an IIoT device at a specific location is unique, it is impossible to obtain valid RSS data when the device is randomly moved to another location. So, it fails to recover the key. This prevents key recovery because the move is an anomaly.
- Even though an illegitimate device is placed in the same location as the IIoT device, it cannot generate the valid two-factor device DNA of that IIoT device because the illegitimate device does not have a valid intrinsic factor. So, key recovery fails.
- A secret key of the IIoT device is wrapped and protected in the form of a fuzzy vault which has information-theoretical security to hide the secret key in a data set. The vault is stored in the device, not the secret key itself, so the leak or theft of the vault cannot result in key recovery or have a fatal effect on the smart manufacturing system.

Moreover, in order to show the feasibility of our two-factor device DNA based fuzzy vault scheme in unmanned smart manufacturing platforms, we implemented our scheme into an IIoT device prototype based on the Raspberry Pi with the RC PUF. Then, we conducted various experiments considering normal and abnormal cases in an actual unmanned smart manufacturing system environment in Hanyang University ERICA campus [12], where the IIoT devices and machines are operating, to check the key recovery rate for the correct key. We evaluated the system performance of our scheme and also analyzed the security of our fuzzy vault and showed it to meet the security levels 112, 128, and 192 recommended by NIST's Recommendation for Key Management [14].

The rest of this paper is organized as follows: we introduce some preliminary information and related works in Section II. We present our system model in Section III and then propose two-factor device DNA generation and our fuzzy vault scheme in Section IV. Section V shows the security of our fuzzy vault and Section VI shows various experimental results and evaluates the performance of our two-factor fuzzy vault scheme. Section VII is the conclusion.

II. RELATED WORKS

In this section, we introduce the concept of a fuzzy vault scheme and analyze the state-of-the-art research related to our work.

A. FUZZY VAULT SCHEMES

Juels and Sudan [19] proposed a fuzzy vault scheme, which securely protects a cryptographic key using the user's private data set. They used biometrics with noise as the user's private data set. While fuzzy commitment is affected by the order-invariant of the data, a fuzzy vault is not and instead provides sufficient difficulty even on non-uniform distribution noise data. A fuzzy vault scheme consists of two phases: a locking phase and an unlocking phase. During the locking phase, a polynomial and a locking private data set are used to embed the secret key into the vault. Later, the secret key can be reconstructed from the vault in the unlocking phase using an unlocking data set if it is similar to or mostly overlaps the locking data set. The security of a fuzzy vault scheme is based on the polynomial reconstruction problem.

The fuzzy vault has been commonly studied for biometric cryptosystems for either authentication or securing the cryptographic key using biometrics such as fingerprints, irises, faces and palms [26]–[30]. In addition, the user's behavior-based key sharing techniques [31] based on two on-body devices using Photoplethysmogram (PPG) and electrocardiogram (EKG) signals have been proposed by utilizing the fuzzy vault. Recently, Revadigar *et al.* [32] proposed a group key generation scheme using a fuzzy vault based on the user's unique movement gathered by an accelerometer and then sharing the group key with other smart devices worn by the user. However, existing works [26]–[30] are not suitable for securing data and cryptographic keys in IIoT devices which operate in unmanned environments, because they need biometric data and user interaction. You *et al.* [34] proposed a fuzzy vault scheme using authentication based on location information gathered from a smart phone's GPS data. However, using GPS data is prone to errors, especially indoors which makes it problematic for use in a fuzzy vault scheme. This means that the GPS data is not appropriate for use in smart manufacturing systems as it does not accurately locate the IIoT devices indoors.

B. RECEIVED SIGNAL STRENGTH

RSS (Received Signal Strength) is the reception strength when receiving the radio frequency (RF) signal that the receiver uses for wireless communication. Any device that receives an RF signal can easily measure and verify it, and the strength or weakness of the signal is determined by the effects of diffraction, interference, and attenuation depending on the surrounding environment.

Research on localization by constructing a Wi-Fi fingerprint or secret key generation using similar RSS has been actively pursued. Kwon *et al.* [22] proposed to prevent unauthorized device access to autonomous vehicles with a Wi-Fi fingerprint that specifies the location using Wi-Fi signals [21]. However, their scheme [22] has the disadvantage of using various factors and complicated processes to generate Wi-Fi fingerprints. Zi *et al.* [23] proposed a secret key generation technique using the signal strength which varies depending on Wi-Fi distance. The same secret key can be generated by measuring similar wavelength changes of two devices with the same movement. Key generation techniques using the RSS were also proposed in [20]–[25]. However, any device can obtain the signal and generate the same secret key in [20]–[25], so these works are not proper for smart manufacturing system environments where only authorized IIoT devices are allowed.

C. PHYSICAL UNCLONABLE FUNCTION

PUF can generate device-specific values using the device's physical property differences [35]–[37]. Even if the device to which the PUF are applied is manufactured by the same manufacturing process, the value generated by the PUF cannot be duplicated because the same large mapping cannot be substantially repeated. Because of this property, the value generated in the PUF inside the device can be used to uniquely identify individual devices such as device DNA. PUFs are generally classified as two classes: weak PUF and strong PUF [38]–[40].

- Weak PUFs have limited challenge and response space. This is generally only used to derive a secret key.
- Strong PUFs have a number of Challenge-Response Pairs (CRPs) that are not physically replicable, and it is impossible to collect all CRPs within a reasonable time. Also, it is difficult to predict the response according to the challenge.

Strong PUF's response space, unlike weak PUFs, creates a response without being protected. Therefore, access restriction to PUF-response is unnecessary, and attacks on PUF-response space are mostly not easy with strong PUF. Adi [17] proposed a dynamic identification technique that utilizes a DNA-like identity generated by TRG (True Random Generator) with PUF inside the device. The DNA-like identity is created once and deleted after being used, so it has stronger replication resistance. However, it is difficult to guarantee the security against physical attacks such as attempting to use the device identity after stealing the device. Therefore, we propose a technique for generating two-factor device DNA by using RSS data acquired from the IIoT's surrounding environment as a challenge to a strong PUF. Two-factor device DNA can be generated only when a legitimate device holding the strong PUF is located at the specific legitimate location, so it can provide the security of encrypted data by using the two-factor device DNA against physical attacks. We utilize a resistor-capacitor (RC) PUF [42] as a strong PUF. The RC PUF adopts analog circuits such as resistors and capacitors and an analog-to-digital converter (ADC) as a PUF primitive, so it can provide affordable PUF functionality for resource-constrained IoT devices without using on-chip/off-chip PUF-dedicated hardware [42].

III. SYSTEM MODEL

A. ASSUMPTIONS

As shown in Figure 1, for the purposes of our design we are using the IIoT platform in the unattended smart manufacturing system, which is a very controlled environment. Figure 1 shows the IIoT platform, which consists of gateways and IIoT manufacturing machines and sensors (called IIoT devices for short) operating in unattended environments. In these unattended environments, there are multiple devices. The gateways check the status of IIoT devices and gather data from them and analyze it. An edge server manages the manufacturing process and allows users to monitor and control all aspects of automated production. The target IIoT devices for our scheme are sensors used in smart manufacturing facilities. It is difficult to continuously supply power to or change batteries for these IIoT devices. So, we need a wireless communication technology with wide communication coverage

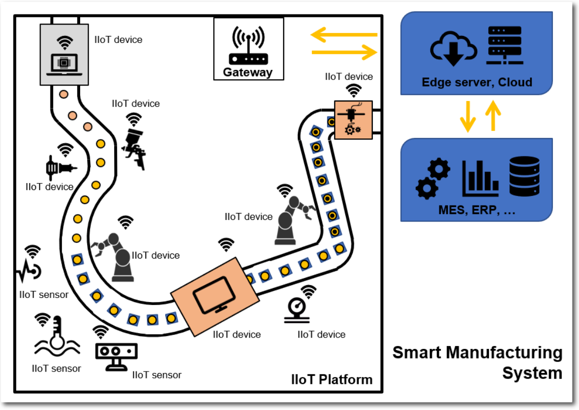


FIGURE 1. **IIoT platform in smart manufacturing system.**

and low power characteristics, LPWAN (Low Power Wide Area Network), such as IEEE 802.15.4g in a smart factory [41], which allows long-range communications at a low bit rate among sensors that use batteries. All IIoT devices in IIoT platforms include EPUF and can transmit/receive using IEEE 802.15.4g as the Rx (Receive) node and the Tx (Transmit) node.

B. THREAT MODEL

As a threat model for our design, we consider passive eavesdropping attackers and active attackers. We assume that a passive attacker is aware of the two-factor device DNA and fuzzy vault mechanism and can eavesdrop on all the packets and wireless signals exchanged between the legitimate IIoT devices. The passive attacker attempts to obtain meaningful information for recovering a secret key by analyzing each captured signal or packet data.

An active attacker is assumed to possess IIoT devices similar to legitimate IIoT devices in a smart manufacturing system. The attacker can attempt to gain physical access to the smart manufacturing system in order to steal the fuzzy vault (the hidden information) and data stored in IIoT machines, but we do not assume that the attacker will steal the IIoT machine itself. The active attacker also attempts to gather the signal data to recover a secret key from the fuzzy vault by installing its own mimic IIoT device at the location of the legitimate IIoT device in the smart manufacturing system. In addition, in order to manipulate the IIoT machines, the active attacker can sabotage Rx or Tx sensors to cause a malfunction, or change the location of Rx or Tx nodes, causing the received signal to contain errors. We assume that the legitimate IIoT devices working in the smart manufacturing system are benign without any embedded malware, are non-compromised, and it is impossible for the attacker to alter the RSS recorded by the legitimate IIoT devices. It is assumed that the attacker does not have unlimited computational power to use in the attack, and legitimate devices are not able to be accessed remotely. We do not consider DDoS and jamming attacks.

C. REQUIREMENTS

In order to construct a secure two-factor device DNA based fuzzy vault scheme for IIoT platforms against attacks mentioned in the Threat Model, we should consider following security requirements. We assume that a legitimate IIoT device applied the two-factor device DNA based fuzzy vault scheme to protect a secret key for performing cryptographic operations such as encryption/decryption.

1) On an IIoT platform in an unattended environment, if the legitimate device is in a valid position, it must be able to reconstruct the secret key.
2) If the IIoT device is not in a valid position on the IIoT platform, it should not be able to reconstruct the secret key because no valid environment factor can be obtained.
3) IIoT devices should not be able to reconstruct the secret key unless they are in an unattended environment, such as when an unauthorized person attempts to physically access the IIoT platform to cause devices to malfunction, such as by changing the location of Tx or manipulating Rx. In addition, IIoT devices should be able to determine that the current state is unattended or attended.
4) Any IIoT device that cannot obtain a valid intrinsic factor of a legitimate device should not be able to reconstruct the secret key of the legitimate IIoT device even if it has obtained a valid environment factor.
5) Even if the fuzzy vault stored inside a legitimate device is leaked, the secret key of the device should not be exposed.

We will show that our two-factor device DNA based fuzzy vault scheme satisfies Requirements 1),2),3) and 4) through the experimental results in Chapter 6, and prove that our scheme satisfies Requirement 5) through the security analysis in Chapter 5.

IV. PROTOCOL DESIGN

In this section, we present the details of our design for generating two-factor device DNA and Fuzzy Vault for protecting secret keys in IIoT devices.

A. TWO-FACTOR DEVICE DNA GENERATION

In order to construct the two-factor device DNA generator, we employ the EPUF value as the intrinsic factor and the RSS of the signal as the environment factor.

Figure 2 shows an overview of our two-factor device DNA generation. All nodes (Rx/Tx) in Figure 2 are assumed to be IIoT devices. An Rx (receive) node generates two-factor Device DNA, and a Tx (transmit) node transmits signals which provide environment factors for the Rx node to generate two-factor Device DNA. The Tx node emits a signal that is used by the Rx to determine the environment factor, which in this case is the RSS. Once the Rx node receives the signal from the Tx, it can measure the RSS and generate the two-factor device DNA as follows:

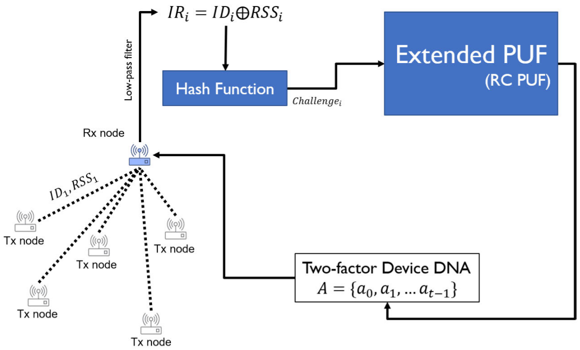


FIGURE 2. Two-factor Device DNA generator.

1) PROCESSING THE RSS DATA

RSS data measured by the Rx node in the unattended IIoT platform contains noise due to the wireless signal itself and is affected by the movement of the internal manufacturing equipment. So, it is necessary to process the RSS data to remove the noise, in order to use it as an environment factor.

While the manufacturing equipment is in operation (e.g. the robot arm repeatedly moves to manufacture the product), the RSS data measured by Rx is affected, so the received signal strength oscillates within a predictable range. Therefore, we utilize a LPF which allows through signals with a frequency lower than a designated number, while attenuating signals with frequencies over a certain cutoff in order to minimize the impact of manufacturing equipment actions in normal working conditions. The LPF derives the estimated value RSS'_i from the recently measured RSS_i using $RSS'_{(i-1)}$.

$$RSS'_i = (1-\alpha)RSS'_{(i-1)} + \alpha RSS_i,$$

where α is a smoothing factor and $0 < \alpha < 1$. The larger the amount of noise to be removed, the lower the α is, but the abnormal change in RSS data cannot be observed well. RSS data measured by the Rx node have different effects on noise and movement of internal equipment depending on the position of the Tx node that transmits the signal. For this reason, if α is set to the same value in order to filter RSS data transmitted by each Tx, it may not be possible to effectively remove noise from the RSS transmitted by each Tx. Therefore, when the Rx node measures RSS data in the setup stage, it is necessary to evaluate the RSS data for each Tx and select an appropriate weight as the α.

2) GENERATING OUTPUT VALUE OF EXTENDED PUF

RSS data from IEEE 802.15.4g signals is within a typical range (approximately -10 dB to -100 dB), so if the RSS value itself is used as the challenge value, the possibility of entering the same challenge value to calculate the PUF response for each device increases. Therefore, to widen the range of challenge values for each device, we combine the ID value and the RSS value of the device and convert them to a uniform distributed value by employing a cryptographic hash function that generates the challenge.

- The *Rx* node performs an *XOR* operation on the ID of the *Tx* node and the RSS value of the signal transmitted by the *Tx* node and then generates $challenge_i$ by using a cryptographic hash function $H(.)$. $challenge_i = H(ID \oplus RSS'_i)$, $(0 \leq i < t)$.
- The *Rx* node executes the embedded EPUF by inputting the $challenge_i$ respectively and generates two-factor device DNA, $DNA = \{a_0, a_1, a_2, \ldots, a_{t-1}\}$.

B. FUZZY VAULT USING TWO-FACTOR DEVICE DNA

Each IIoT device generates its own secret key *sk* and performs the vault locking process to hide its secret key *sk* by using two-factor device DNA. The generated secret key *sk* is transmitted to the gateway using a secure channel and stored in the gateway of the IIoT platform containing the device. We assume that the gateway securely stores the *sk*s of each IIoT device using a separate hardware security module [44]

1) VAULT LOCKING PROCESS

The vault locking process as shown in Figure 3 is performed once for each device. If a device is taken over or damaged, all devices proceed to refresh the stored vaults.

- **Polynomial Construction:**
  1) Each IIoT device such as Rx or Tx generates CRC(Cyclic Redundancy Check) codes based on its own secret key *sk* by using CRC encoder. The CRC code used for the fuzzy vault can identify whether the reconstructed polynomial for recovering the secret key is valid or not in the vault unlocking process [18]. So we can apply the CRC code to check whether the key recovery is successful or not. We use CRC-32 to generate a 32-bit CRC code based on *sk*. CRC-32 has a 2^{-32} probability of not identifying an invalid reconstructed polynomial. The following 32-bit primitive polynomial was used in our implementation: $g_{crc}(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$
  2) The IIoT device concatenates *sk* and the *CRC* codes generated based on *sk*, and then $sk|CRC$ is divided into $n+1$ consecutive blocks p_i, $sk|CRC = \{p_0, p_1, p_2, \ldots, p_n\}$. Each block is used as the coefficient of the *n*th order polynomial as follows:

$$f(x) = p_n x^n + p_{n-1} x^{n-1} + \ldots + p_1 x + p_0$$

- **Construction of a message set:** The IIoT device utilizes the two-factor device DNA generator to construct a message set $A = \{a_0, a_1, a_2, \ldots, a_{t-1}\}$, where t is the number of Tx nodes and $t > n$.
- **Vault Construction:** In order to obtain a set of real points $RP_i = \{a_i, y_i = f(a_i)\}, (0 \leq i < t)$, the IIoT device calculates the projection of set A on polynomial $f(x)$. Then, it generates a large set of random chaff points CP, which excludes all elements in set A and has no

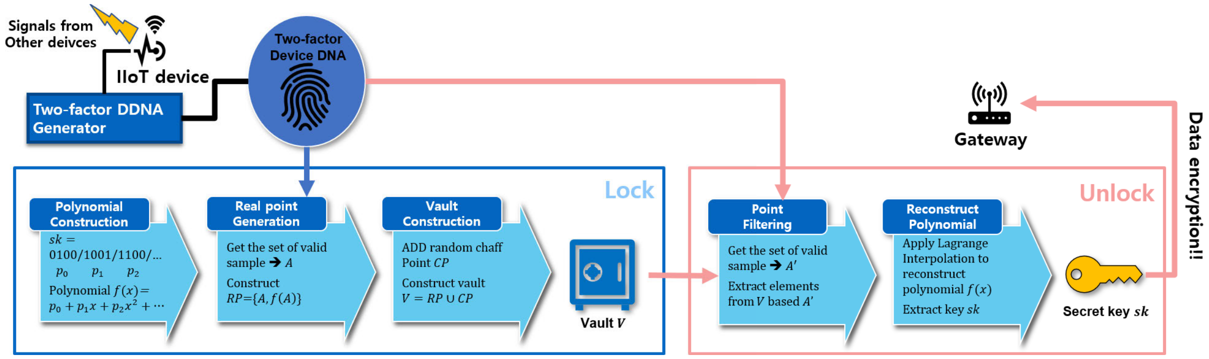

Signals from Other deivces
IIoT device
Two-factor DDNA Generator
Two-factor Device DNA
Lock
Polynomial Construction
$sk =$ 0100/1001/1100/...
p_0 p_1 p_2
Polynomial $f(x) = p_0 + p_1x + p_2x^2 + \cdots$
Real point Generation
Get the set of valid sample ➔ A
Construct $RP=\{A, f(A)\}$
Vault Construction
ADD random chaff Point CP
Construct vault $V = RP \cup CP$
Vault V
Unlock
Point Filtering
Get the set of valid sample ➔ A'
Extract elements from V based A'
Reconstruct Polynomial
Apply Lagrange Interpolation to reconstruct polynomial $f(x)$
Extract key sk
Secret key sk
Gateway
Data encryption!!

FIGURE 3. Two-factor Device DNA-based fuzzy vault for IIoT platform in smart manufacturing system.

intersection with the polynomial $f(x)$.

$$CP_j = \left\{b_j, y_{b_j} = f_R(b_j)\right\}, (b_j \notin A)$$

The larger the j range is, the more secure the device is. Finally, the vault $V = RP \cup CP$ is a large set of points in which the secret points of A are hidden by random chaff points. By mixing the chaff points CP with the real points RP, the CP and RP cannot be distinguished from the *vault* V using statistical analysis [19]. Each IIoT device stores the vault V.

2) VAULT UNLOCKING PROCESS

The legitimate IIoT devices placed in a valid position must perform the Vault Unlocking process to extract a secret key sk from the vault V. The vault unlocking process as shown in Figure 3 consists of the following steps.

- **Construction of a message set:** The Rx device performs the two-factor device DNA generation process by obtaining the signal where it is placed and executing the embedded EPUF, and generates two-factor Device DNA $A' = \left\{a'_0, a'_1, a'_2, \ldots, a'_{t-1}\right\}$. Set A' is constructed by RSS after the signal noise has been filtered out.
  The parameter u is the number of measurements of RSS needed to filter the signal noise. u is selected appropriately to ensure that A' overlaps sufficiently with set A and does not contain any chaff points. Both set A' and A must include at least $(n + 1)$ elements to reconstruct polynomial p, if the order of the polynomial is n.
- **Reconstruction of polynomial** The Rx device selects a set of real points RP' belonging to A' from the vault V stored in the Rx device. The polynomial is reconstructed by using Lagrange interpolation to RP'. The coefficients of the reconstructed polynomial p' are $\left\{p'_0, p'_1 . p'_2, \ldots, p'_n\right\}$.
  Rx combines these coefficients to form $sk'|CRC'$ and generates sk'-based CRC code, called $CRC^{''}$. If the generated $CRC^{''}$ code value matches the CRC', it means that the recovered secret key sk' in the vault unlocking process is the same as the original secret key sk. Through this step, Rx can check whether the original secret key sk is successfully recovered or not. If $CRC^{''}$ code value does not match the CRC', it means that the key recovery has failed. So, Rx can recognize the failure of key recovery and report it to the gateway. We do not cover anomaly detection in this paper, but many studies have been already published on anomaly detection. Rx can perform a separate anomaly detection process to identify the abnormal situation by customizing these techniques [20].

V. SECURITY OF FUZZY VAULT

In this section, we discuss the security of our two-factor device DNA based fuzzy vault scheme. This is used in a situation where an eavesdropper obtains the fuzzy vault and attempts to recover the key from the vault. The security of the fuzzy vault depends on the number of chaff points in the target vault set [19]. When the number of chaff points goes up, a set of spurious polynomials emerges, looking like the correct polynomial p needed for secret key recovery. Without additional information, the attacker cannot distinguish between the correct polynomial and all of the spurious ones. So, the valid polynomial p is hidden in a vault set using information-theoretic security, with security which increases proportionally to the number of spurious polynomials.

The minimum number of points required to reconstruct a polynomial of order k is $k + 1$. Take for example a vault size of $v = 1600$ and polynomial order of 13. An attacker must attempt a brute force attack by constructing polynomials using all the combinations of 14 points in the vault with a size of 1600 to unlock. So, the total number of possible trials to find a valid polynomial p to reconstruct the key are 7.8×10^{33} and it is equivalent to brute force attacking a secret key of 112 bits in length.

Figure 4 presents the security offered by vaults of different sizes v for various polynomial orders n. In order for the Two-factor Device DNA based fuzzy vault to meet Requirement (5) in Section 3.3, the security level of the fuzzy vault must be at least 112 bits [14].

As shown in Figure 4, we can see that fuzzy vault with a vault with (v, n) where $v = 900$ and $n = 15$ or $v = 1600$ and $n = 13$ provides 112-bit security. If a much higher level

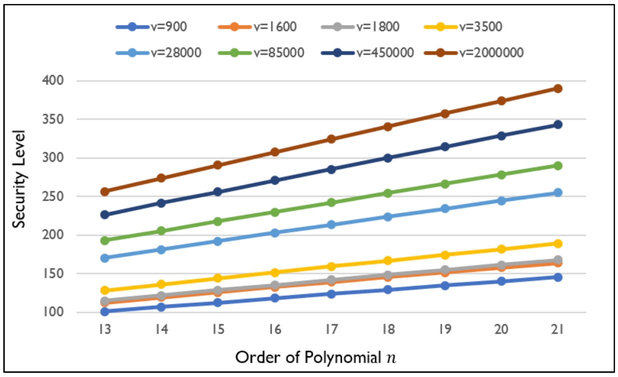


FIGURE 4. Security of the vault.

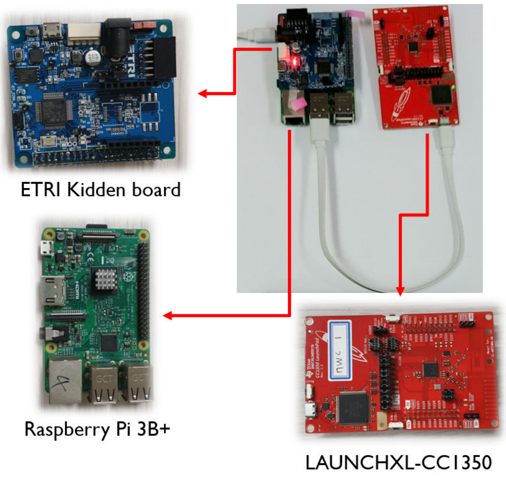


FIGURE 5. Prototype IIoT device with two-factor device DNA fuzzy vault.

```
Algorithm 1 Generate Two-Factor Device DNA
Public Parameter: $GF(q)$
Input: Number of Real point $t$.
Output: Two-factor DDNA $A = \{a_i\}_{i=0}^{t-1}$.
begin
$A \leftarrow \phi$
for $i = 0$ to $t - 1$ do
    $ID_i \longleftarrow$ Identify $Tx_i$ Device ID;
    $RSS'_i = LPF(RSS_i, \alpha_i)$;
    $challenge_i = H(ID_i \oplus RSS'_i)$;
    $a_i = Extended_PUF(challenge_i)$;
    Set $A \leftarrow A \bigcup a_i$;
end for
Output $A$;
end
```

```
Algorithm 2 Vault Lock - Generate the Vault
Public Parameter: $GF(q)$
Input: Number of Real point $t$; Secret key $sk$; Order of polynomial $n$.
Output: Vault $V = \{x_i, y_i\}_{i=0}^{q-1}$ such that $x_i \in GF(q)$.
begin
$V, RP, A \leftarrow \phi$;
Coefficient set $p \longleftarrow$ Partition $sk$ by $n + 1$;
$A = Generate\_Two\text{-}factor\_Device\_DNA(t)$;
for $i = 0$ to $t - 1$ do
    $(x_i, y_i) \leftarrow (a_i, f(a_i))$;
    $V \leftarrow V \bigcup (x_i, y_i)$;
end for
for $i = 1$ to $q - 1$ do
    $x_i \leftarrow b_i \in_r GF(q) - A$;
    $y_i = f_r(b_i) \longleftarrow f_r(x)$ is the polynomial having the random coefficient;
    $V \leftarrow V \bigcup (x_i, y_i)$;
end for
Output $V$;
end
```

of security is needed, large v and n are required. The more detailed theoretical study of the security provided by fuzzy vault scheme is referred in [19].

VI. PERFORMANCE EVALUATION

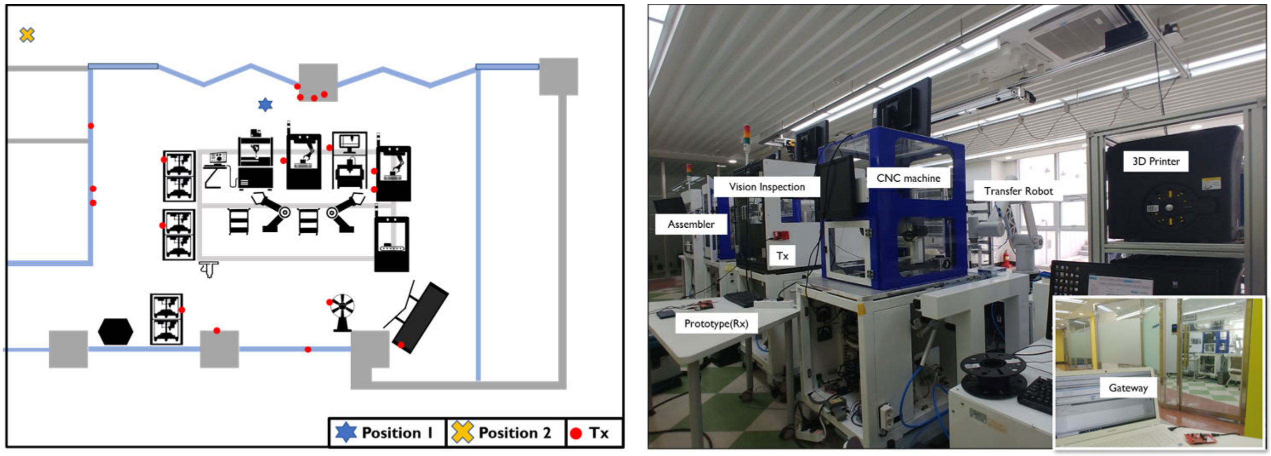
A. PROTOTYPE FOR IIoT DEVICES

We built the prototype for IIoT device(Rx/Tx nodes) systems to demonstrate the feasibility of our fuzzy vault based on two-factor device DNA. The configuration of our prototype IIoT device is shown in Figure 5. Our prototype consists of a Raspberry Pi, a LAUNCHXL-CC1350, and Kidden board with attached RC PUF [42] with functionalities to obtain PUF as an intrinsic factor.

B. EXPERIMENTAL SetUp

Through serial communication using the GPIO, the Kidden board is connected to the Raspberry Pi to generate the PUF output. The LAUNCHXL-CC1350 is connected to the Raspberry Pi via serial communication using the USB. LAUNCHXL-CC1350 is a development board capable of IEEE 802.15.4g communications, which continuously receives IEEE 802.15.4g signals from other devices in the same network and passes the received data to the Raspberry Pi. First, the Raspberry Pi generates a challenge based on the received IEEE 802.15.4g signal data. Since IEEE 802.15.4g is less affected by diffraction, refraction, and interference due to the characteristics of using a low frequency band, the IEEE 802.15.4g signal can penetrate obstacles, so that RSS data obtained can be more stable than with a high frequency signal. However, the RSS data may not always have the same value when extracted, because the intermittent noise may happen. So, in order to filter out this noise, we consider employing data pre-processing functions such as Mode function.

Next, once the Kidden board with attached RC-PUF receives the challenge from the Raspberry Pi, it generates a response based on the environment factor, RSS, and transmits it on to the Raspberry Pi. In order to evaluate the feasibility of the performance of our protocol, we implemented the two-factor device DNA generator and fuzzy vault algorithm



(a)

(b)

FIGURE 6. Experiment environment setup.

using C language. Algorithm 1 shows the pseudocode of the two-factor Device DNA generator. Algorithm 2 and 3 show the pseudocode of fuzzy vault Lock and Unlock algorithms, respectively.

Algorithm 3 Vault Unlock - Regenerate the Secret Key

Public Parameter: $GF(q)$

Input: Number of Real point t; Vault V.

Output: Secret key sk.

```
begin
$RP', A' \leftarrow \phi$;
$A' = Generate\_Two\text{-}factor\_Device\_DNA(t)$;
for $i = 0$ to $t - 1$ do
    $(x_i, y_i) \xleftarrow{(a',0)} V$;
    $RP' \leftarrow RP' \bigcup (x_i, y_i)$;
end for
Set $p' \longleftarrow$ Reconstruction polynomial using Lagrange Interpolation
$sk \longleftarrow$ Regenerate sk by combining $p'$
Output $sk$;
end
```

As shown in Figure 6, the experiment was carried out in the Smart Manufacturing Learning Center at Hanyang University ERICA campus of 14.85 × 14.95 m^2 with 18 LAUNCHXL-CC1350s to serve as Tx. Hanyang University's Smart Manufacturing Learning Center is a test bed for smart factories that provides a test manufacturing and production environment for producing prototypes in an unmanned environment [12]. We placed the laptop outside the space to act as a gateway, which verified the validity of the encrypted data sent by our prototype. We installed the prototype in position 1 in Figure 6 to act as Rx. In order to confirm that our proposed two-factor device DNA based fuzzy vault scheme satisfies the requirements (1 through 4) mentioned in Section 3.3, we performed the following case-by-case experiments.

- **Case 1:** A legitimate IIoT device generates Two-factor Device DNA at the location where the vault was created and attempts to reconstruct the secret key using it.
- **Case 2:** A legitimate IIoT device generates Two-factor Device DNA at a location different from where the vault was created and attempts to reconstruct the secret key using it.
- **Case 3:** If the legitimate Tx and Rx devices are in the location where the vault is created, but the attacker with the purpose of manipulating the IIoT device approaches these Tx or Rx nodes, to sabotage the Rx or change the location of the Tx. The attacker attempts to generate Two-factor device DNA and reconstruct the secret key using it.
- **Case 4:** Any illegitimate device that has taken a valid vault of a legitimate IIoT device and attempts to reconstruct the secret key of the legitimate IIoT device at the location where the vault was created.

Case 1 is the normal case, while Cases 2-4 are abnormal cases. So, key reconstruction using the two-factor device DNA-based fuzzy vault scheme should only be successful in Case 1 and not in the others. In the experimental setup phase, our prototype generated the Two-factor Device DNA at position 1 and created the vault by performing the vault locking process. We applied the smoothing factor α for the LPF of our prototype by selecting the optimal value for each Tx before performing the vault locking process in this experiment. In order to select this optimal value, it is necessary to evaluate the RSS data of each Tx that the prototype device acquires according to the specifics of each experimental environment. The amount of noise included in the raw RSS data obtained by the prototype differs depending on the positions of the Txs transmitting the signal.

As shown in the Figure 7, the graphs included in the raw data on the left of Figure 7 are raw RSS data obtained by Tx 3, Tx 4, Tx 7 and Tx 14 respectively in our experimental

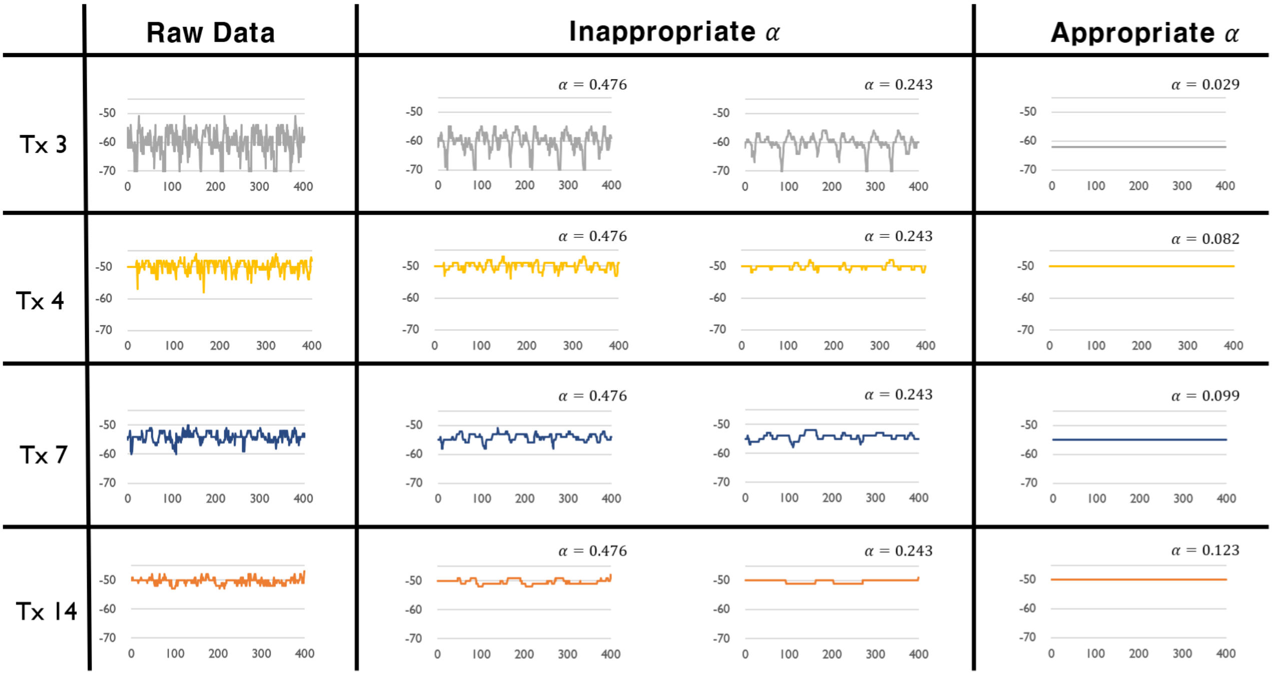


FIGURE 7. Filtered RSS Data using appropriate alpha for Tx 3, 4, 7 & 14.

TABLE 2. α for noise reduction by amount of noise.

Noise (dBm)	±1	±2	±3	±4	±5	±6
α	0.476	0.243	0.163	0.123	0.099	0.082
Noise (dBm)	±7	±8	±9	±10	±11	±12
α	0.070	0.062	0.055	0.049	0.045	0.041
Noise (dBm)	±13	±14	±15	±16	±17	±18
α	0.038	0.035	0.033	0.031	0.029	0.027

TABLE 3. α selection for each Tx.

Tx#	$Tx1$	$Tx2$	$Tx3$	$Tx4$	$Tx5$	$Tx6$
α	0.062	0.476	0.029	0.082	0.099	0.243
Tx#	$Tx7$	$Tx8$	$Tx9$	$Tx10$	$Tx11$	$Tx12$
α	0.099	0.163	0.476	0.123	0.163	0.476
Tx#	$Tx13$	$Tx14$	$Tx15$	$Tx16$	$Tx17$	$Tx18$
α	0.082	0.123	0.243	0.243	0.099	0.082

environment. The graphs in the middle of Figure 7 are filtered RSS data obtained when smoothing factor α is selected that is not appropriate for the noise size included in raw RSS data. The graphs on the right of Figure 7 are filtered RSS data obtained when smoothing factor α is selected that is appropriate for the noise size included in raw RSS data. It can be seen that the minimum value necessary for acquiring constant RSS data differs depending on the amount of noise included in the RSS data. If the prototype applies the same large α value to filter the RSS data from each Tx, it is impossible for them to be sensitive to changes that occur in abnormal cases. Therefore, α must be set individually for each Tx to fit the environmental situation. Before selecting the α for each Tx, we derived an effective α value according to the size of the noise through an experiment. Then, as shown in Table 1, we selected α individually by identifying the maximum noise of RSS data per Tx that the prototype obtained in the normal situation before performing the Vault Locking process. We stated that the α value is determined according to the maximum noise included in the RSS sent by each IIoT device regardless of the environment of the smart manufacturing facility. Therefore, every test setup must first install txs in their smart manufacturing environment, measure the noise level, and then refer to Table 2 to select the alpha value for noise removal. For our experiment, we set α for each Tx with a maximum noise level as shown in Table 3 according to our experimental conditions.

Next, we had the prototype generate the two-factor device DNA based on the RSS data obtained in each experimental case and attempted to reconstruct the secret key by proceeding with the vault unlocking process. Then we calculated the key reconstruction rate (KRR) for each case. If the number of successful key reconstructions is $N_{success}$ and the total number of the vault unlock attempts for the experiment is N_{total}, then the key reconstruction rate (KRR) is the following formula.

$$KRR = \frac{N_{success}}{N_{total}} * 100(\%)$$

C. EXPERIMENTAL RESULTS

1) CASE 1

To show that the key recovery is successfully performed when a legitimate device with two factors is normally in an unattended environment, we did the following experiment. We placed our prototype with an embedded legitimate PUF as an intrinsic factor in position 1 inside our

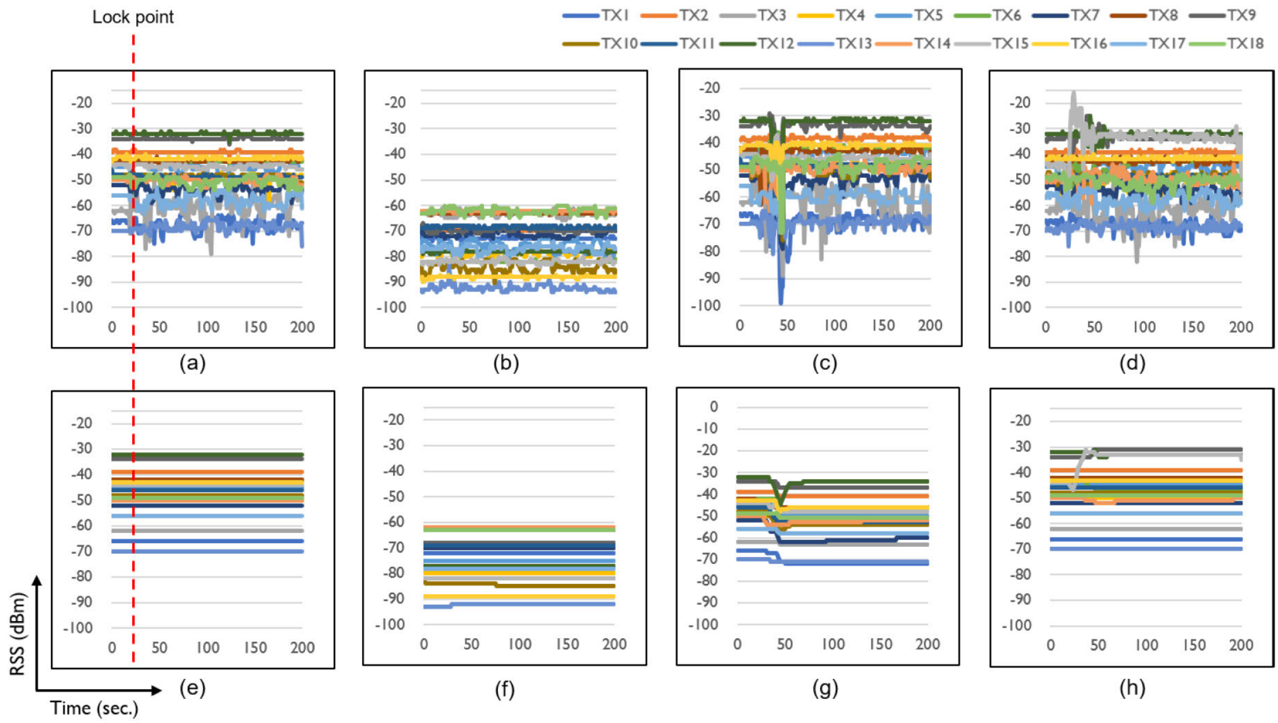


FIGURE 8. (a) RSS data on Position 1. (b) RSS data on Position 2. (c) RSS data on Position 1 when attacker accesses Rx. (d) RSS data on Position 1 when attacker changes the position of a Tx. (e) Filtered RSS data based on (a). (f) Filtered RSS data based on (b). (g) Filtered RSS data based on (c). (h) Filtered RSS data based on (d).

smart manufacturing learning center to proceed with the vault unlocking process. Figure 8-(a) shows the raw RSS obtained by our prototype at that location and Figure 8-(e) shows RSS data after applying the LPF to the raw RSS data. The prototype performed the unlocking process continuously using the filtered RSS data as shown in Figure 8-(e), and the key recovery success rate was 100%. So we can see that our prototype always works in the legitimate place where the vault was created whenever the prototype can get two-factors normally.

2) CASE 2

To demonstrate that key recovery is not possible when an unattended legitimate device does not have an environment factor, we performed the following experiment. We placed our prototype in position 2 inside the smart manufacturing learning center to proceed with the vault unlocking process. Figure 8-(b) shows the raw RSS obtained by our prototype at that location and Figure 8-(f) shows RSS data after applying the LPF to the raw RSS data. The prototype performed the unlocking process continuously using RSS data as shown in Figure 8-(f), and the key recovery success rate was 0%. This is because the RSS data obtained by the prototype in Position 2 is different from the RSS data acquired in Position 1, the location where the vault is created. Therefore, our prototype cannot successfully perform key recovery outside of the legitimate location even if it has the proper intrinsic factor. The position change of the prototype can be detected successfully by using our anomaly detection phase.

3) CASE 3

To show that the legitimate device is in the legitimate position where the vault locking process was undertaken, but the attacker cannot recover the key due to physically approaching the Rx or Tx location, we performed the following experiments, cases 3-1 and 3-2. We started by placing the prototype in position 1 where the vault was created. Then we had one of our experimenters act as the attacker. In Case 3-1, the attacker physically approached position 1 to begin the vault unlocking process in person. And in Case 3-2, the attacker changed the physical location of the Tx. We proceeded with the vault unlocking process by placing the prototype that created the vault at position 1 inside the experiment environment. Figure 8-(c) shows the raw RSS obtained by our prototype at that location and Figure 8-(g) shows RSS data after applying the LPF to the raw RSS data in Case 3-1. As shown in Figure 8-(g), the raw RSS data experienced a period of extreme fluctuation in interference when the attacker approached the Rx, which subsided when the attacker left the area. After applying the filter to the RSS, we can still see the impact the attacker had on the signal, as it did not return to normal as shown in Figure 8-(e). We can filter with the alpha we set under normal conditions like Case 1, but it is incapable of filtering out the extreme noise caused by the attacker's presence.

Therefore, when the prototype attempted the vault unlocking process, the key recovery success rate was 100% until the attacker approached, but the key recovery success rate was 0% after the attacker attempted physical access to the device. This confirms that the attacker cannot gain access to the Rx. Figure 8-(d) shows the raw RSS obtained by our prototype at position 1 and Figure 8-(h) shows RSS data after applying the LPF to the raw RSS data in Case 3-2. The grey line in Figure 8-(h) represents the RSS data from Tx15, and the sharp spike and extreme fluctuation in the raw RSS data is when the attacker approached and changed the position of Tx15. We can also see the impact of the attacker's approach in the raw RSS data from nearby Tx12 in the dark green line.

After applying the filter to the RSS, we can still see the impact the change in the Tx position by the attacker had on the signal, as it did not return to normal as shown in Figure 8-(h). Again, we see that filtering with the alpha we set under normal conditions does not filter out the extreme noise caused by the attacker's actions. So, the key recovery success rate was 0%.

4) CASE 4

Since RSS is location-based data, RSS data obtained at the same location is the same. So even if illegitimate prototype 2 is placed in position 1 of Figure 6, it can get the RSS that prototype 1 obtained when creating the vault. However, since prototype 2 does not have the RC PUF of prototype 1, it is not possible to produce a valid two-factor device DNA. Table 4 compares the two-factor device DNA generated by prototype 1 and prototype 2 at position 1 where the vault was created for use in the vault locking process. Both prototype 1 and 2 have the same challenge based on the RSS data of each Tx, but the two-factor device DNA generated by applying these challenge values to the RC PUF is totally different. Therefore, even though the illegitimate device can obtain a valid environment factor, it cannot produce valid two-factor device DNA because its own intrinsic factor is different, and the KRR is always 0%.

D. SYSTEM PERFORMANCE

In this section, we evaluated the performance of our Two-factor Device DNA based fuzzy vault. In order to show the feasibility of our fuzzy vault for IIoT platforms we implemented prototypes using Raspberry Pi 3B+ and a laptop that can act as edge devices such as sensors and a gateway in the IIoT platform. The Raspberry Pi 3B+ has a 1.4 GHz ARM Coretex-A53 MP4 as a processor, and the laptop has a 2.3 GHz Dual Core Intel Pentium Gold Processor 4415U processor.

Our two-factor device DNA based fuzzy vault uses Lagrange Interpolation for secret key reconstruction. The Lagrange Interpolation requires at least $n + 1$ points, $(x, f(x))$s, to reconstruct the nth order polynomial $f(x)$. So, since the computational complexity of Lagrange Interpolation is $O(n^2)$ [43], increasing the polynomial order for the high level of security will cause a large overhead in the unlocking process.

TABLE 4. **Comparing Two-factor Device DNA Generated from prototype 1 & prototype 2.**

ID	Challenge	Two-factor Device DNA	
		prototype 1	prototype 2
Tx1	C7	328A	D754
Tx2	C9	E098	DFC0
Tx3	D6	68B8	6FAB
Tx4	81	54F4	4A80
Tx5	B5	A20E	9F7B
Tx6	C4	B2DE	571D
Tx7	8A	152C1	3AAD
Tx8	9C	CAB3	6BC
Tx9	6F	17EAE	9A9E
Tx10	9B	6D9F	70AF
Tx11	A4	1CEE2	13BE2
Tx12	AE	1A9F	C5EF
Tx13	D1	1FD8	1DC48
Tx14	B3	D2B1	D6F0
Tx15	8C	C930	DEF8
Tx16	74	5B7C	6DB7
Tx17	CB	4A48	4AE4
Tx18	69	A174	8423

For this reason, when constructing a two-factor device DNA based fuzzy vault, we adjusted the security level by increasing the chaff point (increasing v) using the 13th and 15th order polynomials instead of increasing the order of the polynomials. We implemented our fuzzy vault to meet the 112, 128 and 192-bit security levels recommended in [14]. We used C programming to work efficiently on low-performance, low-capacity IIoT devices like sensors.

Table 5 shows the performance comparison of Two-factor Device DNA Fuzzy vault by security level applied to the laptop and the Raspberry Pi 3B+.

As shown in table 5, when comparing our fuzzy vault with the 13th order polynomial that provides each security level, implemented on the Raspberry Pi 3B+, the clock cycle was 0.24×10^9, 0.32×10^9, and 3.20×10^9 for the locking process and 1.48×10^9, 1.68×10^9, and 2.04×10^9 for the unlocking process, respectively. On the other hand, the number of clock cycles for running our fuzzy vault with the 15th order polynomial that provides each security level (112, 128, and 192) was 0.21×10^9, 0.27×10^9, and 1.42×10^9 for the locking process and 6.04×10^9, 6.02×10^9, and 6.30×10^9 for the unlocking process, respectively. A laptop needs about 25% fewer clock cycles than the Raspberry Pi 3B+ to perform our fuzzy vault.

Since the clock cycle needed to perform the locking process depends on the size of the vault to be created, a program running in the 15th order polynomial shows fewer clock cycles than a program running in the 13th order polynomial. On the other hand, because the unlocking process performs Lagrange Interpolation, the fuzzy vault with the lower order of the polynomial is the more efficient. Therefore, the program that runs the 13th polynomial showed fewer clock cycles than the 15th order polynomial.

Our two-factor device DNA based fuzzy vault has a locking process once in the set up phase, and the unlocking

TABLE 5. Performance comparison by size of the vault *v* and order of the polynomial.

Security Level	v	n	Device	Clock Cycle (10^9)		Time (s)		Power Consume (W)		Vault Capacity (KB)
				lock	unlock	lock	unlock	lock	unlock	
112	1600	13	Laptop	0.06	0.43	0.03	0.19	-	-	67
			Rasp-Pi 3B+	0.24	1.48	0.17	1.06	3.88	4.32	
	900	15	Laptop	0.05	1.91	0.02	0.83	-	-	39
			Rasp-Pi 3B+	0.21	6.04	0.15	4.31	4.06	4.64	
128	3500	13	Laptop	0.08	0.43	0.03	0.19	-	-	161
			Rasp-Pi 3B+	0.32	1.68	0.23	1.20	3.88	4.33	
	1800	15	Laptop	0.06	1.92	0.03	0.83	-	-	87
			Rasp-Pi 3B+	0.27	6.02	0.19	4.30	4.19	4.64	
192	85000	13	Laptop	0.79	0.59	0.34	0.25	-	-	5632
			Rasp-Pi 3B+	3.20	2.04	2.28	1.46	3.98	4.35	
	28000	15	Laptop	0.34	1.98	0.15	0.86	-	-	1843
			Rasp-Pi 3B+	1.42	6.30	1.02	4.50	4.21	4.66	

process continues each time a secret key needs to be reconstructed. Therefore, we recommend the 13th polynomial for the two-factor device DNA based fuzzy vault to provide 112, 128 and 192-bit security levels to configure the unlocking process efficiently.

We also compared the power consumption of the locking and unlocking processes on the Raspberry Pi 3B+ to evaluate whether our Fuzzy vault can be effectively applied to low power devices. Power consumption was $3.88 \sim 4.54$W in the locking process and $4.32 \sim 4.76$W in the unlocking process. That means about 10% to 30% more power consumption than the 3.6W standby power of the prototype implemented with the Raspberry Pi 3B+. This low power consumption indicates that our Fuzzy vault can be easily applied to low power IIoT devices.

As shown in the Table 5, the size of the vault created by the locking process in our program varies greatly from 39KB to 5632KB depending on the security level and the order of polynomial of the fuzzy vault. To ensure a high security level while keeping the vault size small, the order of the polynomial must be increased, which reduces the performance of the unlocking process. This is a trade-off, because the higher order of the polynomial decreases the performance of the unlocking process and the larger size of the vault increases memory costs. Therefore, in order to select the appropriate vault size and polynomial order according to the security level to be guaranteed, the capacity and characteristics of various IIoT devices constituting the IIoT platform should be considered.

VII. DISCUSSION AND LIMITATION

Our scheme operates in unmanned smart manufacturing environments, which should not have any people in them under normal conditions. It is assumed that authorized people will schedule their time to enter the factory to do things such as check the status of the manufacturing process or to manage the factory, thus letting the system know they will be there at a specific time. So, our scheme will report a security breach when any unexpected perturbation such as unplanned intrusions of any kind, even by genuine people, happens. So, it seems that our scheme is not scalable. However, if we only focused on scalability from a technical point of view, we would weaken our security. Our scheme would not report a security breach every time an unscheduled person entered the system. Due to the peculiarities of operating in unattended environments, unmanned environments have more security considerations, issues, vulnerabilities, and threats than manned environments do. To protect the system in unattended environments, any person who enters without a pre-scheduled appointment must be treated like a potential security threat. So, even if the person is genuine but has not scheduled their entrance into the factory, the system must flag them as an attacker since they did not get approval and we do not actually know their true intent. They may appear to be genuine, but without a scheduled appointment, it is safer to flag all unscheduled interactions as attacks.

Therefore, we think that this scalability issue should be covered by each smart manufacturing facility's security policy. They will determine the proper security rules that fit their requirements and situation. Each one will need to be tailored for that factory. For example, they can configure their security policy of the manufacturing facilities by setting a time for valid personnel to enter and exit, stopping reconfiguration of the secret key at that time, and using the key generated before that time, etc. This will vary for every facility. In order to have scalability and to maintain security, it is best to use our scheme in an unmanned smart manufacturing environment along with a proper security policy.

VIII. CONCLUSION

In this paper, we have first proposed a novel concept of the two-factor device DNA, that is generated through the interaction of the device with the surrounding environments, by utilizing an EPUF embedded in the device and RSS data gathered from the place where the device is located. Next, we designed the two-factor device DNA based fuzzy vault scheme for securing secret keys in IIoT devices in unmanned smart manufacturing environments. Our scheme prevents the attackers who attempt to obtain the device's internal data or keys from exposing the secret key stored in the IIoT device. In order to demonstrate the feasibility of our fuzzy vault scheme, we also implemented our fuzzy vault scheme into the IIoT device prototype with an EPUF and a IEEE 802.15.4g receiver to obtain the combination of an intrinsic factor (i.e, a unique internal noisy source of PUF) and a surrounding factor (i.e, RSS data). Finally, we conducted experiments in an unmanned environment at the Smart Manufacturing Learning Center at Hanyang University and then evaluated the key recovery rate for each experiment case in normal and abnormal situations for each security level 112, 128 and 192 recommend by NIST. The experimental results show that our fuzzy vault scheme can always recover a valid secret key in normal cases, but fails to recover the key in abnormal situations, such as when an attacker approaches or the IIoT device location is changed. In the case of security level 112, our prototype using the Raspberry Pi took 0.17 seconds to lock the key, using 3.88W and 1.06 seconds to unlock, using 4.32W when v is 1600 and n is 13. In the case of security level 128, it took 0.23 seconds to lock and 1.2 seconds to unlock when v is 3500 and n is 13. Our fuzzy vault scheme is shown to work well even on a low power IIoT device like a Raspberry Pi.

REFERENCES

[1] *Overview of Smart Manufacturing in the Context of the Industrial Internet of Things*, document Rec. ITU-T, Y.4003, Jun. 2018.

[2] R. Zhou, X. Zhang, X. Du, X. Wang, G. Yang, and M. Guizani, "File-centric multi-key aggregate keyword searchable encryption for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3648–3658, Aug. 2018.

[3] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3618–3627, Aug. 2018.

[4] P. Xu, S. He, W. Wang, W. Susilo, and H. Jin, "Lightweight searchable public-key encryption for cloud-assisted wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3712–3723, Aug. 2018.

[5] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3590–3598, Aug. 2018.

[6] F. A. Turjman and S. Alturjman, "Context-sensitive access in industrial Internet of Things (IIoT) healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2736–2744, Jun. 2018.

[7] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, and J. Bastos, "A lightweight authentication mechanism for M2M communications in industrial IoT environment," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 288–296, Feb. 2019.

[8] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[9] A. Karati, S. H. Islam, G. Biswas, M. Z. A. Bhuiyan, P. Vijayakumar, and M. Karuppiah, "Provably secure identity-based signcryption scheme for crowdsourced industrial Internet of Things environments," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2904–2914, Aug. 2018.

[10] J. Shen, T. Zhou, X. Liu, and Y.-C. Chang, "A novel latin-square-based secret sharing for M2M communications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3659–3668, Aug. 2018.

[11] D. Choi, S.-H. Seo, Y.-S. Oh, and Y. Kang, "Two-factor fuzzy commitment for unmanned IoT devices security," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 335–348, Feb. 2019.

[12] *SMLC Introduction*. Accessed: Mar. 23, 2020. [Online]. Available: https://smlc.hanyang.ac.kr/front/smlc-introduce/space

[13] N. Tuptuk and S. Hailes, "Security of smart manufacturing systems," *J. Manuf. Syst.*, vol. 47, pp. 93–106, Apr. 2018.

[14] E. Barker and Q. Dang, *Recommendation for Key Management, Part 1: General*, NIST Special Publication 800-57 Part 1, Revision 4, Jan. 2016.

[15] M. N. Aman, M. H. Basheer, and B. Sikdar, "Two-factor authentication for IoT with location information," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3335–3351, Apr. 2019.

[16] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, Jun. 2007, pp. 9–14.

[17] W. Adi, "Clone-resistant DNA-like secured dynamic identity," in *Proc. Bio-Inspired, Learn. Intell. Syst. Secur.*, Aug. 2008, pp. 148–153.

[18] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," in *Proc. Int. Conf. Audio Video-Based Biometric Person Authentication* (Lecture Notes in Computer Science), vol. 3546. Berlin, Germany: Springer, 2005, pp. 310–319.

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

[20] J. Tang, P. Fan, and X. Tang, "A RSSI-based cooperative anomaly detection scheme for wireless sensor networks," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2007, pp. 2783–2786.

[21] W. K. Zegeye, S. B. Amsalu, Y. Astatke, and F. Moazzami, "WiFi RSS fingerprinting indoor localization for mobile devices," in *Proc. IEEE 7th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2016, pp. 1–6.

[22] H. Kwon, S. Lee, and B. Chung, "Wi-Fi fingerprint-based approach to securing the connected vehicle against wireless attack," in *Proc. 8th Int. Conf. Netw. Commun.*, Dec. 2016, pp. 1–7.

[23] Z. Li, Q. Pei, I. Markwood, Y. Liu, and H. Zhu, "Secret key establishment via RSS trajectory matching between wearable devices," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 802–817, Mar. 2018.

[24] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 927–935.

[25] M. F. Awan, K. Kansanen, S. Perez-Simbor, C. Garcia-Pardo, S. Castello-Palacios, and N. Cardona, "RSS-based secret key generation in wireless in-body networks," in *Proc. 13th Int. Symp. Med. Inf. Commun. Technol. (ISMICT)*, May 2019, pp. 1–6.

[26] I. Nakouri, M. Hamdi, and T.-H. Kim, "Chaotic construction of cryptographic keys based on biometric data," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2016.

[27] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcardbased fingerprint authentication," in *Proc. ACM SIGMM Workshop Biometrics Methods Appl. (WBMA)*, Nov. 2003, pp. 45–52.

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

[29] X. Wu, N. Qi, K. Wang, and D. Zhang, "A novel cryptosystem based on iris key generation," in *Proc. 4th Int. Conf. Natural Comput.*, Oct. 2008, pp. 53–56.

[30] X. Wu, K. Wang, and D. Zhang, "A cryptosystem based on palmprint feature," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[31] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "PSKA: Usable and secure key agreement scheme for body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 1, pp. 60–68, Jan. 2010.

[32] G. Revadigar, C. Javali, W. Xu, A. V. Vasilakos, W. Hu, and S. Jha, "Accelerometer and fuzzy vault-based secure group key generation and sharing protocol for smart wearables," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2467–2482, Oct. 2017.

[33] Y. Wu, Y. Sun, L. Zhan, and Y. Ji, "Low mismatch key agreement based on wavelet-transform trend and fuzzy vault in body area network," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 6, Jun. 2013, Art. no. 912873.

[34] L. You, Y. Chen, B. Yan, and M. Zhan, "A novel location-based encryption model using fuzzy vault scheme," *Soft Comput.*, vol. 22, no. 10, pp. 3383–3393, Apr. 2017.

[35] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Controlled physical random functions," in *Proc. 18th Annu. Comput. Secur. Appl. Conf.*, Dec. 2002, pp. 15–22.

[36] G. E. Suh, C. W. O'Donnell, and S. Devadas, "AEGIS: A single-chip secure processor," *IEEE Des. Test. Comput.*, vol. 24, no. 6, pp. 570–580, Nov. 2007.

[37] P. Tuyls, "RFID-tags: Privacy and security issues," *Philips Res.*, 2006. [Online]. Available: https://www.key4biz.it/files/000048/00004831.pdf

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

[39] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[40] U. Ruhrmair and D. E. Holcomb, "PUFs at a glance," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Apr. 2014, pp. 1–6.

[41] H. Harada, K. Mizutani, J. Fujiwara, K. Mochizuki, K. Obata, and R. Okumura, "IEEE 802.15. 4g based Wi-SUN communication systems," *IEICE Trans. Commun.*, vol. E100-B, no. 7, pp. 1032–1043, Jul. 2017.

[42] S. Lee, M. K. Oh, Y. Kang, and D. Choi, "Design of resistor-capacitor physically unclonable function for resource-constrained IoT devices," *Sensors*, vol. 20, no. 2, pp. 326–337, Aug. 2019.

[43] R. Bevilaqua, D. Bini, M. Capovani, and O. Menchi, "Appunti di calcolo numerico," Ph.D. dissertation, Dipartimento di Informatica dell', Università di Pisa, Pisa, Italy, 2008, ch. 5, p. 89. [Online]. Available: http://pages.di.unipi.it/bevilacq/Dispensa11-12.pdf

[44] Hardware security module. [Online]. Available: https://en.wikipedia.org/wiki/Hardware_security_module

EUNGI HONG (Student Member, IEEE) received the B.S. degree from the Department of Electronic System Engineering, Hanyang University at ERICA, South Korea, in 2018, and the M.S. degree in electronic engineering from Hanyang University, in 2020.

His research interests include the IoT security, security of embedded systems, and post-quantum cryptography.

SANGJAE LEE received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in information and communication engineering from Chungbuk National University, South Korea, in 1999, 2001, and 2013, respectively.

He has been a Principal Researcher with the Electronics and Telecommunications Research Institute, Daejeon, since 2000. He participated in developing the technologies for home gateway, home server, IEEE1394, VoIP, network traffic controller, and wireless PAN MAC and UWB SoC. His current research interests include wireless MAC and SoC design for wireless PAN, and the IoT security technologies.

MI-KYUNG OH received the B.S. degree from the Department of Electrical Engineering, Chung-Ang University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2002 and 2006, respectively.

From September 2002 to February 2004, she was a Visiting Researcher with the Department of Electrical and Computer Engineering, University of Minnesota, USA. She is currently a Principal Researcher with the Electronics and Telecommunications Research Institute, South Korea. Her research interests include wireless communication systems, SoC design, and the IoT security technologies.

SEUNG-HYUN SEO (Member, IEEE) 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 Postdoctoral 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 Associate Professor with Korea University at Sejong, Sejong City, South Korea, for two years. In 2017, she joined Hanyang University, Ansan, where she is currently a Professor. Her current research interests include cryptography, the IoT security, mobile security, secure cloud computing, and blockchain security.