A Realistic Lightweight PUF-based Mutual Authentication in RFID Environments

E Haodudin Nurkifli

Abstract—Radio frequency identification has been widely used in several fields based on its low-cost property. However, the wireless-based communication part in the radio frequency identification still suffers from several kinds of attacks such as impersonation attack, cloning attack, DoS attack, and tracking attack. To solve these problems in the radio frequency identification environment, this manuscript proposes a new authentication protocol that just uses limited resources. Moreover, both the informal and the formal analyses are used to ensure that the proposed scheme achieves the security features, withstands kinds of attacks, and has low computational complexity.

Index Terms—Radio Frequency Identification (RFID), Authentication protocol, informal and formal analyses, computational complexity.

I. INTRODUCTION

Radio Frequency Identification (RFID) has been widely applied in several fields, such as, transportation system, supply chain, livestock management, patient medical caring system, e-payment system, and e-passport system. There are three main components in the RFID environment, namely Tag, Reader, and the Server.

The Tag transmits information to the reader where it is processed, then the reader forwards it to the server. The communication between the Tag and the reader is done via the public channel. However, the reader and server communicate via a private channel since both of them are in one unit. The attacker can fully control public channels in which he/she can eavesdrop, intercept, edit, and delete the message. Hence, to avoid these attacks, the authentication method, a security feature to ensure that both parties recognize the identity mutually, is needed in the RFID environment. Nevertheless, the limited resources in RFID make most of the existing cryptographic algorithms, such as public-key cryptosystems cannot be used in RFID. Thus, a new authentication method is needed in the RFID since the ones with these cryptographic algorithms cannot be directly used in the RFID environment.

Recently, the physical unclonable function (PUF) is used to ensure authentication in the RFID environment. A PUF is a unique property of a circuit that maps a set of Challenge C to a set of the Response R in the chip manufacturing process [1][2][3]. Several authentication protocols based on the PUF in the RFID environment are proposed by [4]–[9]. Unfortunately, these protocols fail to achieve security features such as anonymity, untraceability, solving desynchronization problem, perfect forward, and backward secrecy. Additionally, these protocols fail to withstand various kinds of attacks such as impersonation attack, DoS attack, and tracking attack. Additionally, the previous protocol [4]–[9] used ideal PUF hence it did not denote the realistic condition, in the real condition output of PUF has noisy. Subsequently [10] and [11] proposed protocol security in the RFID environment using realistic PUF. However, [12] and [13] point out that the protocols proposed by [10] and [11] fail to achieve scalability, untraceability. Subsequently, two popular PUF-based authentication protocols in the RFID environment are proposed by [12] and [13], their researches discuss ideal PUF and Non-ideal PUF. However, after we evaluate them deeply, both of them also fail to withstand impersonation attack, namely if the attacker stole the Tag by activating the PUF and Fuzzy Extractor with programming, the attacker can obtain response and fixed key and then the attacker is able to generate a new set of {response, fixed key, challenge, session key}. As result, the attacker impersonates the Tag as a legitimate user (see Section II for more detail of discussion the weaknesses of [12] and [13]).

Besides, both [12] and [13] cannot achieve perfect forward and backward secrecy for after the attacker obtains response, the attacker can obtain past and future secret such as new {response, challenge, session key}. On the other hand, their scheme cannot achieve untraceability as well because the attacker can compute a new challenge. The attacker searches for a new challenge in collected messages, if found, he learns that the owner of the two messages is the same (see Section II for more detail of discussion the weaknesses of [12] and [13])..

In this paper, we propose a new mutual authentication protocol for focusing on non-ideal PUF-based which is used for authentication. The proposed protocol can achieve mutual authentication, anonymity, untraceability, perfect forward, and backward secrecy. Further, the proposed protocol withstands various kinds of attacks such as impersonation attack, DoS

Manuscript submitted June 03, 2021; This work was support by the National Science Council of Taiwan under Contract No. MOST 109-2221-E-006-168-; No. MOST 108-2221-E-006-107.

The author is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan, Universitas Singaperbangsa Karawang, Karawang, Indonesia (e-mail: dudink95@gmail.com, dudi.nurkifli@staff.unsika.ac.id).
Attack, tracking attack, and cloning attack.

The rest of this paper is organized as follows. To support our point of view, Section II presents a brief review of two state-of-the-art protocols proposed by Gope et al. and Feng Zhu et al. and points out their problems. Section III presents our proposal in detail. To ensure that our scheme achieves security features and withstands any kind of attacks, we present informal analysis in Section IV followed by formal analysis using BAN Logic, and Scyther Tool Section V, VI, respectively. Finally, we conclude our article in Section VII.

A. Related Works and Motivation

Several schemes in the RFID environment have been proposed, such as the schemes by [14]–[19] using Elliptic curve cryptography with exponential module operation in the algorithm, however, it is not suitable to be applied in a tiny Tag in an RFID environment.

The other research by [20]–[23] proposed the protocols using a symmetric-key cryptosystem. Unfortunately, their schemes cannot guarantee to withstand the cloning attack. [4]–[9] offer authentication PUF-based which is resilient to the cloning attack. However, these PUF-based authentication schemes [4]–[9] have weaknesses, e.g., [4] tree-based authentication scheme using PUF. [5] pointed out that the scheme proposed by [4] fails to withstand the DoS and impersonation attack.

[5] proposed a new scheme based on PUF to enhance security features. However, [7] presented that the scheme introduced by [4] is vulnerable against the cold boot attack, and then [7] showed the evidence that the attacker is easy to tamper and obtain all data after impersonating it as well as tracing previous and future communication. In other research, [6] came up with a scheme to enhance security features [5]. However, [7] raised awareness that the scheme previously suggested by [5] is vulnerable against cold boot attack. [8] also applied authentication PUF-based in an RFID environment, but the scheme does not achieve forward secrecy and is weak against the DoS attack. However, since cold-boot attacks will cause a defect in the data in Tag memory due to the Tag sensitivity, the attacker cannot obtain data in Tag memory. At the other time, [24], [25] proposed PUF-based authentication in RFID, yet, both of them cannot withstand the DoS attack.

[9] presented a PUF-based authentication in the RFID system. However, [12] P. Gope et al. showed that the scheme proposed by [9] cannot achieve forward secrecy. P. Gope [12] then came up with mutual PUF-based authentication. Feng Zhu et al. [13] presented a lightweight RFID mutual authentication protocol with PUF. However, after careful investigation, we found that both [12]-[13] have the problem to achieve perfect forward and backward secrecy in which the attacker can impersonate the Tag. Additionally, their scheme does not achieve untraceability.

This article resolves the problems in the previous schemes specified by [4]-[13]. We present the review and point out the weakness of two papers by [12] and [13] to support our scheme's perspective in Sections 3 and 4. The proposed scheme contributions are as follow:

1. Evaluating two popular authentication protocols in

RFID environment PUF-based and denote several weaknesses of them
2. Designing the new protocol in realistic conditions using non-ideal PUF. Our scheme achieves security features including mutual authentication, anonymity, untraceability, perfect forward secrecy, and backward secrecy, resolves the desynchronization problem, scalability.
3. Our proposal withstands various attacks such as impersonation attack, DoS attack, cloning attack, and Traceability attack.
4. A formal analysis using BAN Logic was carried out to ensure that our authentication protocol achieves secure mutual authentication.
5. The other analysis formally using the Scyther tool ensures that our protocol withstands various kinds of attacks.
6. A comparison of security features and computational complexity was conducted to ensure our proposal is not only secure but also has a low computational complexity

B. Preliminaries

This subsection briefly presents the preliminary background of the PUF and the capabilities of the attacker.

a. Physically Unclonable Function (PUF)

A PUF is a unique property of a circuit that maps a set of Challenge C to a set of the Response R in the chip manufacturing process [26]. Formally, inputting C into PUF to produce R where \( R = \text{PUF(C)} \) is hard to clone [27]. We divide PUF into two types, namely ideal PUF and non-ideal PUF.

An ideal PUF will produce a similar response with similar challenges inputted into PUF even if the non-stable temperature factor occurs. In the last few years, researchers have developed an ideal PUF which ensures 0% Bit-Error Rate [28]–[31], however, the ideal PUF is not realistic due to the real condition output of PUF has noisy. Meanwhile, a non-ideal PUF may produce a different response with a similar challenge inputted into PUF due to temperature factors. The fuzzy extractor is used to obtain stable PUF output [32], [33]; Therefore, in this paper, we use the non-ideal PUF, where \( R'_i = \text{PUF}(C) \).

b. Fuzzy Extractor.

The fuzzy extractor has two functions, namely \( FE.Gen() \) and \( FE.Rec() \). \( FE.Gen() \) is the probabilistic function to generate the fixed key \( K_u \) and helper data \( hd \) from the input noisy response \( R'_i \), where \( (K_u, hd) = FE.Gen(R'_i) \). \( FE.Rec() \) is the deterministic function to reconstruct \( K_u \) from helper data \( hd \) and noisy input \( R'_i \), where \( (K_u, hd) = FE.Gen(R'_i) \rightarrow K_u = FE.Rec(hd, R'_i) \). The successful \( FE \) is based on the similarity of original data and noisy data. Therefore, in this paper, we use the fuzzy extractor to obtain a fixed key, namely \( K_u \) from noisy response, where \( (K_u, hd) = FE.Gen(R'_i) \) [32], [33].
The Adversary Model
The attacker’s capabilities are summarized as follows: The proposed scheme could still remain secure under the attacks by attackers with the following capabilities. Our work adopts the research conducted by [34].

1. C1: the attacker has full control of the communication channel
2. C2: with a stolen RFID Tag the attacker can activate the PUF, and Fuzzy Extractor by programming

The steps of the attacker:
Step 1: The attacker intercepts the message based on C1
Step 2: Steal RFID Tag and obtains data from Tag’s memory
Step 3: Activate PUF to compute R′ = PUF(C) based on C2
Step 4: Activate Fuzzy Extractor to compute K_u = FE.Rec(hd, R′) based on C2
Step 5: If R′ = PUF(C) and K_u = FE.Rec(hd, R′) can pass the verification, the attacker obtains correct K_u.

II. STATE OF THE ART PROTOCOLS AND PROBLEMS
This section presents a brief review of two state-of-the-art protocols proposed by [12], [13], and points out their problems.

A. Gope et al.’s Protocol
This subsection presents a brief review of the authentication protocol proposed by [12], including notations, setup phase, mutual authentication, and the main weakness. Table 1 is the notations used throughout the authentication protocol proposed by [12].

| Notation | Definition |
|----------|------------|
| TID_i   | Temporary identity of the tag for i-th round |
| CRP(C_i,R'_i) | Challenge-Response Pairs for the i-th round |
| K_i    | Session Key for the i-th round |
| PUF_T  | Physically Unclonable Function of the tag |
| h(·)   | One-way hash function |
| ⊕      | Exclusive-OR operation |
| ||     | Concatenation operation |
| FE     | Fuzzy Extractor for generating fixed-key from noisy-response |

a. The setup phase
Step 1: The backend server generates Challenge C_i and sets the emergency challenge C_em = {c_1, c_2, ... c_n} and then the backend server sends C_i, C_em to the Tag.
Step 2: Upon receiving C_i, C_em, the Tag produces the response R'_i, R'_em using PUF_T where R'_i = PUF_T(C_i) and R'_em = PUF_T(C_em). The Tag sends R'_i, R'_em to the backend server.
Step 3: The server generates the temporary identities TID_i for the i-th round, computes (K_i, h_d_i) = FE.Gen(R'_i) and (K_em, h_d_em) = FE.Gen(R'_em). The server generates a set of the un-liked pseudo identities PID = (pid_1, pid_2, ..., pid_n). The server copies and sends TID_i, PID, (C_i, h_d_i) (C_em, h_d_em) to the Tag and the backend server stores TID_i, PID as well along with (C_i, C_em) (K_i, K_em).
Step 4: The Tag stores TID_i, PID, (C_i, h_d_i) (C_em, h_d_em) . The setup phase takes place via a secure channel.

b. The mutual authentication between tag and reader-server unit
Step 1: The Tag selects TID_i, generates COUNT, and sends TID_i,COUNT to the backend server
Step 2: Upon receiving TID_i,COUNT, the server looks for TID_i in its database, if the server does not locate TID_i, the communication will be terminated, otherwise, the server reads challenge-fixed key pairs CRP (C_i, K_i) generates a nonce N, computes N_s = K_i ⊕ N_s, and computes the verifier of the server Res_s = h(COUNT + 1 || N_s) || N_s). The server sends C_i, N_s, Res_s to the Tag.
Step 3: Upon receiving C_i,N_s,Res_s, the Tag generates R'_i = PUF_T(C_i), selects h_d_i, computes K_i = FE.Rec(h_d_i, hd_i), verifies Res_s, computes N_s = K_i ⊕ N_s, c_i = h(COUNT + 2 || N_s) || N_s), R'_i = PUF_T(C_i), R'_i = K_i || R'_i, Res_t = h(N_s || N_s) || N_s). The Tag sends R'_i, Res_t to the backend server.
Step 4: Upon receiving R'_i, Res_s, the server verifies Res_t. If the Res_t is not equal to Res_s the server will terminate the communication, otherwise computes R'_i = K_i || R'_i, generates (K_i, h_d_i) = FE.Gen(R'_i), computes C_i = h(COUNT + 2 || N_s) || N_s), h_d_i = h(COUNT + 3 || N_s) || N_s), updates TID_i = h(TID_i || N_s), b_s = h(d_i || N_s) || N_s), and stores TID_i+1 (C_i, K_i). The server sends h_d_i, V_s to the Tag.
Step 5: Upon receiving h_d_i, V_s, the Tag verifies V_s. If the Res_t is not equal to Res_s the server will terminate the communication, otherwise computes h_d_i = h(COUNT + 3 || N_s) || N_s), updates TID_i+1 = h(TID_i || N_s), finally, the Tag stores TID_i+1 (C_i, h_d_i+1).

c. The main weakness in [12].
Step 1: The attacker intercepts every message, namely M_1: {TID_i,COUNT}, M_2: {C_i,N_s,Res_s}, M_3: {R'_i,Res_t}, and M_4: {h_d_i+1, V_s} based on C1.
Step 2: Steal RFID Tag and obtains data from the Tag’s memory
Step 3: Activate PUF to compute R'_i = PUF_T(C_i), based on C2
Step 4: Activate Fuzzy Extractor to compute K_i = FE.Rec(h_d_i, ), based on C2
Step 4: If the attacker passes verification Res_s by K_i, that is, the attacker obtains correct fixed key
Step 5: The attacker can compute secrets include current response $R'_i$, $K_i$, new challenge $C_{i+1}$, new response $R'_{i+1} = PUF^T(C_{i+1})$, new help data $hd_{i+1}$ based on $C_1, C_2$.

d. The scheme [12] does not achieve untraceability

Since the attacker can compute correct R, and the attacker can compute new challenge $C_{i+1}$. The scheme [12] does not achieve untraceability, the proof is as follow:

Step 1: Compute $C_{i+1} = h(COUNT + 2||N_i||K_i)$

Step 2: Search $C_{i+1}$ in his collected message

Step 4: If the attacker finds similar $C_{i+1}$ in his/her collected message, he/she will know that the owner of the two messages is the same.

B. Feng Zhu et al.’s Protocol

This subsection presents a brief review of the authentication protocol in [13], including Notations, Setup phase, Mutual authentication, and the main weakness. Table 2 is the notations used throughout the authentication protocol proposed by [13].

A. The setup phase

| Notation   | Definition                                                                 |
|------------|-----------------------------------------------------------------------------|
| TID$i$    | Temporary identity of the Tag for i-th round                               |
| CRP(C, R$i$) | Challenge-Response Pairs for the i-th round                                |
| K$i$     | Session Key for the i-th round                                              |
| PUF$T$   | Physically Unclonable Function of the Tag                                  |
| h()      | One-way hash function                                                      |
| ⨁         | Exclusive-OR operation                                                     |
| ||        | Concatenation operation                                                    |
| FE        | Fuzzy Extractor for generating fixed-key from noisy-response, where Fixed key $K_i$ and help data $hd$ are generated from $R'_i$ where $(K_i, hd) = FE.Gen(R'_i)$ Fixed key $K_i$ is reconstructed from $R'_i$, $hd$ where $(K_i) = FE.Rec(R'_i, hd)$ |

Step 1: The server generates Challenge $C_i$ and sends $C_i$ to the tag.

Step 2: The tag generates $R'_i = PUF^T C_i$ and sends $R'_i$ to the server.

Step 3: The server generates $SID^T_i$, computes $(k_i, hd_i)FE.Gen(R'_i)$, sends $SID^T_i, hd_i$ to the Tag, and stores: $\{SID^T_i, C_i, k_i\}$

Step 4: The tag stores $SID^T_i, hd_i$. The setup phase via secure channel.

b. The mutual authentication between the tag and reader-server unit

Step 1: The server generates Hello, and sends $M_1:\{Hello\}$ to the tag

Step 2: The tag reads $SID^T_i, hd_i$, and sends $M_2:SID^T_i$ to the server.

Step 3: upon receiving $M_2:SID^T_i$, the server finds $SID^T_i$ in database, reads $C_i, k_i$, generates $N$, computes $N' = k_i \oplus N$,

$Auth_1 = h(N||k_i)$. The server sends $M_3:\{C_i, N', Auth_1\}$ to the tag

Step 4: upon receiving $M_3:\{C_i, N', Auth_1\}$, the tag computes $R'_i = PUF^T C_i, k_i = FE.Rec(R'_i, hd'_i)$, $N = N' \oplus k_i$, verifies $Auth_1$, generates $M$, computes $C_{i+1} = h(N||M||C_i)$. $R'_{i+1} = PUF^T(C_{i+1})$. $R'' = R'_{i+1} \oplus N$. $M' = M \oplus N$. $Auth_{i+2} = h(N||M + 1||R'_{i+1})$. The tag sends $M_4:R'', M', Auth_{i+2}$ to the server.

Step 5: upon receiving $M_4:R'', M', Auth_{i+2}$, the server computes $(k_{i+1}, hd_{i+1})FE.Gen(R'_{i+1})$. $C_{i+1} = h(N||M||C_i)$. $SID^T_{i+1} = h(N||M + 1||SID^T_i)$. $hd'_i = h(M||k_i) \oplus hd_i$. $Auth_{i+3} = h(N||M + 2||hd_{i+1})$, stores $SID^T_{i+1}, C_{i+1}, k_{i+1}$. The server sends $M_5:\{Auth_{i+3}\}$ to the tag.

Step 6: upon receiving $M_5:\{Auth_{i+3}\}$, the tag computes $hd_{i+1} = h(M||k_i) \oplus hd'_i$ verifies $Auth_{i+3}$, computes $SID^T_{i+1} = h(N||M + 2||SID^T_i)$ and stores $SID^T_{i+1}, hd_{i+1}$

c. The weakness of Feng Zhu et al.’s Scheme

This subsection presents the weakness of a scheme by [13] based on the stolen card, and the attacker can activate PUF by programming, [13] fails to achieve perfect forward and backward secrecy.

Step 1: The attacker intercepts every message namely $M_1:\{Hello\}, M_2:SID^T_i, M_3:\{C_i, N', Auth_1\}$, $M_4:R'', M', Auth_{i+2}$ and based on $C_1$.

Step 2: Steal RFID tag and obtains data from the Tag’s memory

Step 3: Activate PUF to compute $R'_i = PUF^T C_i$ based on $C_2$

Step 4: Activate Fuzzy Extractor to compute $k_i = FE.Rec(R'_i, hd'_i)$ based on $C_2$

Step 5: If the attacker passes verification $Auth_1$ by $k_i$, that is, the attacker obtains correct fixed key $k_i$

Step 6: The attacker can compute all secrets include current response $R'_i$, $k_i$ and then the attacker can compute new challenge $C_{i+1}$, new response $R'_{i+1}$, new help data $hd_{i+1}$ based on $C_1, C_2$.

d. The scheme [13] does not achieve untraceability

Based on the attacker’s ability to compute correct $k_i$ and a new challenge, [13] does not achieve untraceability, the proof is as the following:

Step 1: Can compute $C_{i+1} = h(N||M||C_i)$

Step 2: Search $C_{i+1}$ in his collected message

Step 4: if the attacker finds a similar $C_{i+1}$ in his/her collected message, he/she will know the owner of two messages is the same.

III. Proposed Scheme

This section presents our proposed scheme starting from System structure, Assumptions, Notation, Setup phase between
Tag and backend server, and mutual authentication between Tag and the reader-server unit. For convenience, we use the notation in Table 3 throughout our scheme.

A. System structure

This subsection presents a brief description of the system structure.

In Fig 1, the system structure consists of three participants, namely, Tag, Reader, and server. The Tag transmits information to the reader where it is processed, then the reader forwards it to the server. The communication between the Tag and the reader is done via the public channel. However, the reader and server communicate via a private channel since both of them are in one unit.

B. Assumptions

We have several assumptions as follows:

a. The RFID is equipped with PUF.

b. The RFID is able to resist side-channel attacks [35].

c. Cold Boot Attack will cause data defect in Memory of Tag [35].

C. Notations of Cryptography Function

This subsection presents the notations of the cryptography function in table III throughout this paper. Additionally, based on the assumption aforementioned, the devices are equipped with PUF; hence our scheme uses this facility.

D. Setup Phase between Tag and the Server

The setup phase between Tag and the server has 4 steps as follows:

Step 1: The tag sends ID_T, Req to the server.

Step 2: The server generates C_T, K_T, PID_T, computes PID_Syn = {pid_1, pid_2, ..., pid_n}, K_syn = {k_1, k_2, ..., k_n}, C_syn = {c_1, c_2, ..., c_n} and sends them to the tag.

Step 3: The tag computes R_T = PU_F(C_T), R_T' = PU_F(C_syn). The tag sends R_T, R_T' to the server. The server stores K_T, PID_T, PID_Syn, K_Syn.

Step 4: Finally, the server stores C_T, R_T', ID_T, PID_T, K_T, C_syn, R_T'(PID_Syn, K_Syn) and the tag stores K_T, PID_T, PID_Syn, K_Syn. Setup phase via a secure channel.

On normal conditions the tag using PID_T, K_T is replaced by new values given by the server. When desynchronization occurs, PID_T, K_T are replaced by (pid_1, k_1),...(pid_n, k_n).

TABLE III
NOTATIONS OF CRYPTOGRAPHY FUNCTIONS

| Notation   | Definition                                      |
|------------|------------------------------------------------|
| ID_T       | Identity of the Tag                             |
| K_T        | Shared key between server and Tag               |
| PU_FT      | Physically uncloneable function of Tag          |
| PID_T      | One-time Pseudonym of Tag to communicate with the server |
| C_T        | Challenge for the i-th iteration               |
| R_T        | The noisy Response of the respective PUF for C_T |
| ||         | Concatenation operation                         |
| o          | Exclusive-OR operation                          |
| PID_Syn    | Pseudo identity of synchronization is used if a loss of synchronization occurs where PID_Syn = {pid_1, pid_2, ..., pid_n} |
| K_Syn      | Key of synchronization is used if a loss of synchronization occurs where K_syn = {k_1, k_2, ..., k_n} |
| FE         | Fuzzy Extractor for generating fixed-key from response and help data, where Fixed key K_T and help data h_d are generated from R_T where (K_T, h_d) = FE.Gen(R_T) Fixed key K_T is reconstructed from R_T, h_d where (K_T) = FE.Rec(R_T, h_d) |

E. Mutual Authentication Phase between Tag and the reader-server unit

Mutual authentication phase between Tag and The reader-server unit has steps as follows:

Step 1: The tag selects PID_T, generates N_a, and then sends M_1: {PID_T, N_a} to the reader-server unit.

Step 2: Upon receiving M_1: {PID_T, N_a}, the reader-server unit verifies PID_T in the database if the reader-server unit does not find PID_T in the database, the communication will be terminated, otherwise, the reader-server unit reads the CRP (C_T, R_T), K_T in server’s database, computes C_T = C_T o h(K_T || N_a). The reader-server unit decides to compute PID_T = PID_T ^ h(D_T || C_T || PID_T), the reader-server unit copies and stores PID_T. The reader-server unit computes authentication code V_T = h(D_T || N_a || C_T || PID_T) to the tag.

Step 3: Upon receiving M_3: {C_T, PID_T, V_T} the tag verifies V_T, computes C_T = C_T o h(K_T || N_a), generates R_T = PU_F(C_T), computes (k_T, h_d_T) = FE.Gen(R_T), h_d_T = h_d_T o h(K_T || N_a), new challenge where C_T = C_T * h(K_T || N_a).
\[ h(C_T | k_T) \] computes \[ R_T^{\text{new}'} = \text{PUF}_T(C_T^{\text{new}}), \]
\[ N_1 = h(k_T | K_{TS}) \oplus R_T^{\text{new}'}, \]
\[ V_A = h(1D_T | C_T^{\text{new}} | R_T^{\text{new}'}) N_1 | h(d_T) \]
and then the tag sends \( M_3: \{ V_A, N_1, h(d_T) \} \) to the server. The server updates \( K_{TS}^{\text{new}} = h(K_{TS} | k_T) \), the tag computes \( P_{ID_{TS}}^{\text{new}} = P_{ID_{TS}} \oplus h(1D_T | R_T') \) and stores \( K_{TS}^{\text{new}}, P_{ID_{TS}}^{\text{new}} \).

**Step 4:** Upon receiving \( M_4: \{ V_A, N_1, h(d_T) \} \), the reader-server unit computes \( h.d_T = h(d_T) \oplus h(K_{TS} | N_a) \), \( k_T = \text{FE}. \text{Rec}(R_T', h.d_T) \), \( R_T^{\text{new}'} = h(k_T | K_{TS}) \oplus N_1 \), \( C_T^{\text{new}} = h(C_T | k_T) \), then verifies \( V_A \). The reader-server unit computes \( K_{TS}^{\text{new}} = h(K_{TS} | k_T) \), Finally, the reader-server stores \( K_{TS}^{\text{new}}, P_{ID_{TS}}^{\text{new}}, (C_T^{\text{new}}, R_T^{\text{new}'}). \)

**IV. ANALYSIS INFORMAL AND COMPARISON**

This section presents the informal analysis using intuitive reasoning to show that our scheme achieves security features. This section also compares capabilities to withstand kinds of attacks with the previous authentication protocol by [10], [11], [12], and [13].

**A. Informal Analysis**

This subsection presents the security feature analysis. The details are as follows:

**a. Achieve Mutual Authentication**

Both parties recognize the identity of the other party by the possession of a shared key \( K_{TS} \). The tag checks \( N_a \), and the reader-server unit checks \( N_1 \) to ensure the freshness. Therefore, the scheme achieves mutual authentication.

**b. Achieve Strong Anonymity**

The tag uses a one-time Pseudonym \( P_{ID_{TS}} \) to communicate with the reader-server unit. The adversary cannot obtain the original identity. Therefore, the scheme provides anonymity.
c. Achieve Perfect Forward Secrecy

In our scheme, the message updates in every session include shared key \( K_{TS}^{new} \), challenge \( C_T^{new} \), and response \( R_T^{new} \). Even if the attacker obtains the current shared key \( K_{TS} \), the attacker cannot obtain past secrets since those are deleted from memory. Therefore, the scheme achieves perfect forward.

d. Achieve Perfect Backward Secrecy

Our scheme protected challenge \( C_T \) by a shared key \( K_{TS} \) where \( C_T = C_T \oplus h(K_{TS})[N_a] \): Even if the attacker can obtain shared key \( K_{TS} \) and \( R_T \), the attacker cannot obtain fixed key \( k_f \) simultaneously because of \( K_f \) generated by the FE.Gen, where FE.Gen is the probabilistic function that produces different output if the attacker inserts the same response \( R_T \), and then the attacker cannot compute new challenge \( C_T^{new} \) and new response \( R_T^{new} \). Therefore, our proposed scheme achieves perfect backward secrecy.

e. Achieve Untraceability

Because of the attacker cannot compute new challenge, since the attacker cannot know the owner of the challenge based on the collected message in the public channel. Therefore, the scheme achieves untraceability.

f. Resolve the Loss of Synchronization

When desynchronization occurs, the tag replaces \( PID_{TS}, K_{TS} \), by \( pid_1, k_1 \). Run the same step with the authentication phase. Therefore, the scheme can resolve the loss of synchronization.

B. Attack analysis

This subsection presents that our scheme withstands various kinds of attacks including impersonation attack, Tracking attack, and DoS attack. The details are as follows:

a. Withstand against Impersonation Attack

The attacker can’t reveal the current fixed key, new challenge

| Schemes                  | SF1 | SF2 | SF3 | SF4 | SF5 |
|-------------------------|-----|-----|-----|-----|-----|
| Asyu et al.[10]         | No  | No  | No  | Yes | No  |
| Huth et al.[11]         | No  | No  | No  | Yes | No  |
| Gope et al. [12]        | No  | No  | Yes | Yes | Yes |
| Feng Zhu et al. [13]    | No  | No  | Yes | Yes | Yes |
| Our scheme              | Yes | Yes | Yes | Yes | Yes |

The details of notations are SF: Security Features, SF1: Perfect Forward Secrecy, SF2: Perfect Backward Secrecy, SF3: Untraceability, SF4: Anonymity SF5: Resolve the Desynchronization problem.

and new response even if the attacker stoles and activate PUF; hence, the attacker can’t impersonate the Tag. Therefore, the scheme withstands Impersonation attack.

b. Withstand Tracking Attack

Because the scheme achieves untraceability, hence, our propose withstands a tracking attack.

c. Withstand against the DoS Attack

Based on the security features analysis, our scheme solves the loss of synchronization; hence automatically, our proposed scheme withstands against the DoS attack as well.

### Table III

| Protocol       | Computation           | Overall                      |
|----------------|-----------------------|------------------------------|
|                | Tag                   | Reader-Server unit           |

| Schemes                  | T\( \_H \) + 2TPUF + TRNG | T\( \_H \) + 2TPUF + 2TRNG |
|-------------------------|---------------------------|-----------------------------|
| Asyu et al.[10]         | 3\( T_H \) + 2TPUF + TRNG + T_{SKE} + T_{FE.Gen} | 3\( T_H \) + 2TPUF + 2TRNG + T_{SKE} + T_{SKD} |
| Huth et al.[11]         | 3\( T_H \) + 2TPUF + TRNG + T_{SKE} + T_{CBKA} + T_{FE.Gen} | 6\( T_H \) + 2TPUF + 2TRNG + T_{SKE} + T_{SKD} + T_{FE.Gen} + T_{FE.Rec} |
| Gope et al. [12]        | 5\( T_H \) + 2TPUF + TRNG + T_{FE.Rec} | 5\( T_H \) + 2TPUF + TRNG + T_{FE.Gen} + T_{FE.Rec} |
| Feng Zhu et al. [13]    | 4\( T_H \) + 2TPUF + TRNG + T_{FE.Rec} | 5\( T_H \) + 2TPUF + TRNG + T_{FE.Gen} + T_{FE.Rec} + T_{FE.Gen} |
| Our scheme              | 4\( T_H \) + 2TPUF + TRNG + T_{FE.Gen} | 7\( T_H \) + 2TPUF + 2TRNG + T_{FE.Gen} + T_{FE.Rec} |

The details of notations are \( T_H \): Time of the one-way hash function, \( T_{PUF} \): Time of PUF process, and \( TRNG \): Time of Random Number Generation, \( T_{SKE} \): Time of Symmetric Key Encryption, \( T_{SKD} \): Time of Symmetric Key Decryption, \( T_{CBKA} \): Time of Channel-Based of Key Agreement Operations, \( T_{FE.Gen} \): Time of Fuzzy Extractor Generation, \( T_{FE.Rec} \): Time of Fuzzy Extractor Reconstruction.
C. Comparison

The section presents a comparison between our proposed scheme with the previous protocol authentication by [10], [11], [12], and [13] in terms of security features and the capability to resist attack. The comparison is below:

Based on table 4, [10], and [11] only achieve SF4 and does not achieve SF1-SF3, SF5. [12] and [13] do not achieve SF1-SF3, however, both of them achieve SF4 and SF5. The proposed scheme achieves all security features covering Perfect Forward Secrecy and Perfect Backward Secrecy, untraceability, Anonymity, and resolve the Desynchronization problem, represented as SF1-SF5.

Based on table 5, [10], and [11] cannot resist W1-W3 but they are resilient to W4. [12] and [13] are vulnerable against WA1 and WA2 yet withstand W3 and W4. Only our proposed scheme can withstand all kinds of attacks, namely WA1-WA4.

Table 6. presents the computational complexity of the schemes respectively. [10] has a result in overall computational complexity, namely $6T_H + 2T_{PUF} + 2T_{RNG} + T_{SKR} + T_{SKD} + T_{FE:Gen} + T_{FE:Rec}$. [11] obtains total computational complexity of $6T_H + 2T_{PUF} + 2T_{RNG} + T_{SKR} + T_{SKD} + 2T_{CRS} + T_{FE:Gen} + T_{FE:Rec}$. [12] gets total computational complexity of $9T_H + 2T_{PUF} + 2T_{RNG} + T_{FE:Rec} + T_{FE:Gen}$. [13] gets total computational complexity of $9T_H + 2T_{PUF} + 2T_{RNG} + T_{FE:Rec} + T_{FE:Gen}$. Our scheme gets total complexity of $7T_H + 2T_{PUF} + 2T_{RNG} + T_{FE:Gen} + T_{FE:Rec}$. The final result shows that our scheme is lowest than [10], [11], [12] and [13]. Therefore, our proposal is secure and more efficient, suitable, and comfortable as well as practical to be applied in the RFID environment.

V. ANALYSIS FORMALLY USING BAN LOGIC

This subsection presents an analysis formally using BAN logic as evidence that our scheme achieves secure mutual authentication.

A. Brief explanation of BAN Logic

The BAN Logic has three objects [36], namely, participants, encryption keys, and logical formulas. In this paper participants are Tag and Reader-server unit, encryption keys are $K_{TS}$ and $k_T$, logical formulas are based on Table VII and VIII.

B. Proof of the scheme’s formal analysis using BAN Logic

This subsection presents the formal analysis of Mutual Authentication between Tag (T) and Reader-Server Unit (RS),
starting from the idealized protocol, the assumption and goal, and protocol analysis.

1) Create the idealize protocol
We create the ideal protocol and omit the first process.

a. Message
Step2: RS → T: \( C_7^*, PID_{TS}^I, V_0 \)
Step3: T → RS: \( N_1, R_{T_{new}}^I, hd_T^I \)

b. Logical Formula
Step2: RS → T: \( (C_7^*, PID_{TS}^I)_KTS \)
Step3: T → RS: \( \{N_1, R_{T_{new}}^I, hd_T^I\}_kTKTS \)

2) The assumption or goals
This subsection presents the assumptions and goals including the belief of shared keys, the belief of freshness, and trust

a. Belief of shared keys
\[ T \equiv RS \]
The Tag believes that Tag and RS using shared \( K_{TS} \)
\[ RS \equiv RS_KTS \rightarrow T \]
The RS believes that RS and Tag using shared \( K_{TS} \)
\[ T \equiv T_KTS \rightarrow RS \]
The Tag believes that Tag and RS using shared \( k_T \)
\[ RS \equiv RS_KTKTS \rightarrow T \]
The RS believes that RS and the Tag using shared \( k_T \)

b. Belief of freshness
\[ T \equiv \#(N_a) \]
The Tag believes that \( N_a \) is fresh
\[ RS \equiv \#(N_a) \]
The RS believes that \( N_a \) is fresh
\[ \equiv (N_1) \]
The RS believes that \( N_1 \) is fresh
\[ T \equiv \#(N_1) \]
The Tag believes that \( N_1 \) is fresh

b. Trust
\[ T \equiv RS \equiv RS_KTS \rightarrow T \]
the Tag believes that RS believes using shared key \( K_{TS} \)
\[ T \equiv RS \equiv RS_KTKTS \rightarrow T \]
the Tag believes that RS believes using shared secret \( k_T \)

3) The protocol analysis
This subsection presents the protocol analysis in mutual authentication between Tag (T) and Reader server unit (RS), starting from step 2 then step 3.

Step2: RS → T: \( (C_7^*, PID_{TS}^I)_KTS \)

a. Message meaning
\[ T \equiv RS \rightarrow T \quad \langle (C_7^*, PID_{TS}^I)_KTS \rangle, \quad RS_KTS \rightarrow T \]
\[ T \equiv \#(N_a) \]
Because T believes that RS and T share a key \( K_{TS} \), T sees that \( (C_7^*, PID_{TS}^I)_KTS \) is encrypted by \( K_{TS} \), T believes that RS once said \( (C_7^*, PID_{TS}^I) \)

b. Nonce verification
\[ T \equiv RS \rightarrow T \quad \langle (C_7^*, PID_{TS}^I) \rangle, \quad T \equiv \#(N_a) \]
Because T believes that RS once said \( (C_7^*, PID_{TS}^I) \) and T believes that \( N_a \) is fresh, T believes that the \( (N_a, (C_7^*, PID_{TS}^I)) \) is fresh

c. Belief
\[ T \equiv RS \rightarrow \langle (C_7^*, PID_{TS}^I) \rangle, \quad \#(N_a) \]
Because T believes that RS once said \( (C_7^*, PID_{TS}^I) \) and T believes that \( N_a \) is fresh, T believes that RS believes \( (N_a, (C_7^*, PID_{TS}^I)) \)

Step3: T → RS: \( \{N_1, R_{T_{new}}^I, hd_T^I\}_kTKTS \)

a. Message meaning
\[ RS \equiv T \rightarrow RS \]
\[ RS \equiv RS_KTKTS \rightarrow T \]
Because RS believes that RS and T using shared key \( k_T, K_{TS} \), RS sees that \( \{N_1, R_{T_{new}}^I, hd_T^I\}_kTKTS \) are encrypted by \( k_T, K_{TS} \), RS believes that T once said \( (N_1, R_{T_{new}}^I, hd_T^I) \)

b. Nonce verification
\[ RS \equiv T \equiv \langle (N_1, R_{T_{new}}^I, hd_T^I) \rangle, \quad RS \equiv \#(N_1) \]
Because RS believes that T once said \( (N_1, R_{T_{new}}^I) \) and RS believes that \( N_1 \) is fresh, RS believes that the \( (N_1, (N_1, R_{T_{new}}^I, hd_T^I)) \) is fresh

c. Belief
\[ RS \equiv T \equiv \langle (N_1, R_{T_{new}}^I, hd_T^I) \rangle, \quad RS \equiv \#(N_1) \]
Because RS believes that T once said \( (N_1, (N_1, R_{T_{new}}^I, hd_T^I)) \) and T believes that \( N_1 \) is fresh, the RS believes that T believes \( (N_1, (N_1, R_{T_{new}}^I, hd_T^I)) \)

Based on formal analysis using the BAN logic, our proposed scheme achieved mutual authentication with evidence that upon the receiving \( (C_7^*, PID_{TS}^I) \), the Tag obtains \( C_T \) by computing \( C_T = C_7^* \oplus h(K_{TS} || N_a) \) generates \( R_T^* \) with \( R_T^* = PUF_T(C_T) \). \( (k_T, hd_T^I = FE.TG(R_T^*) \) verifies \( V_0 \) to ensure the message from the reader-server unit. On the other hand, the reader-server unit also authenticates the Tag with evidence; upon receiving \( \{R_{T_{new}}^I\}_{k_TKTS^I}, \{hd_T^I\}_{KTS-N_a} \) the reader-server unit computes
\[ h_d = h_d^* \oplus h(K_{TS}||N_a) \] reconstruction \( k_T \) from \( R_T^* \), and \( h_d \). The reader computes \( R_T^{new} = h(k_T||K_{TS}) \oplus N_a \), then verifies \( V_A \) to ensure the message from the \( T \). Therefore, all parties recognize the identity of the other party by the possession of the shared key \( K_{TS}, k_T \) and check freshness based on \( N_a, N_1 \).

VI. ANALYSIS FORMALLY USING SCYTHER TOOL

In this article, we also use the Scyther Tool for validating our proposed protocol. The Scyther Tool is implemented in Python programming, which is used to verify the protocol security [37]. This tool also follows Delov-Yao (DY) adversary model [38]. There are two steps to evaluate a protocol using the Scyther Tool. First, we modeled our proposed protocol using Protocol Description Language (SPDL), and the second one ran the Scyther tool to validate the protocol’s security claim.

Figure 3 denotes the result of the modeling of our protocol. There are two entities in the modeling, namely a tag and reader-server. Later, the verification result for our proposed protocol is shown in figure 4. Based on figure 4, the Scyther Tool denotes that it cannot find any attacks, which proved our security claim.
VII. CONCLUSION

This article proposed a new lightweight realistic PUF-based and mutual authentication scheme to improve the security features and resolve the RFID environment problem. Based on the informal analysis, our proposed scheme fulfills security features such as Mutual Authentication, Perfect Forward, and Backward Secrecy, Untraceability, Anonymity, resolve Desynchronization (SF1–SF5), and withstand various kinds of attacks (WA1–WA4). Additionally, formal analysis using BAN Logic ensures our scheme achieves mutual authentication, and analysis formally using Scyther tool denotes that our scheme withstands various kinds of attacks. On the other hand, based on the computational complexity comparison, our protocol obtains lower computational cost compared to the scheme proposed by by [10], [11], [12], and [13]. Therefore, our scheme is much suitable to be applied in the RFID environment.

ACKNOWLEDGMENT

The author would like to thank the editor and all the anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] T. Meagher et al., “A PUF taxonomy A PUF taxonomy,” Appl. Phys. Rev., vol. 6, p. 011303, 2019.
[2] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, “Physical Unclonable Functions and Applications: A Tutorial,” Proc. IEEE, vol. 102, no. 8, pp. 1126–1141, 2014.
[3] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Kandielwal, “Design and Implementation of PUF-Based ‘Unclonable’ RFID ICs for Anti-Counterfeiting and Security Applications,” IEEE Int. Conf. RFID, pp. 58–64, 2008.
[4] J. Bringer, “Improved Privacy of the Tree-Based Hash Protocols Using Physically Unclonable Function,” Secur. Cryptogr. Networks, vol. 5229, pp. 77–91, 2008.
[5] A. Sadeghi, I. Visconti, and C. Wachsmann, “PUF-enhanced RFID security and privacy,” Proc. Secur. Compon. Syst. Identify, Col., no. January 2010, pp. 102–106, 2016.
[6] M. Akg and C. U. M, “PUF Based Scalable Private RFID Authentication,” 2011 Sixth Int. Conf. Availability, Reliab. Secur., pp. 473–478, 2011.
[7] K. Suleyman and K. S. Mehmet, “A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions,” RFID. Secur. Privacy, RFIDSec 2011. Lect. Notes Comput. Sci., vol. 7055, pp. 78–93, 2012.
[8] K. Suleyman, C. Serkan, Y. Muhammet, and L. Albert, “PUF-enhanced offline RFID security and privacy Kardas,” J. Netw. Comput. Appl., vol. 35, pp. 2059–2067, 2012.
[9] M. Akg and M. Ufuk, “Providing destructive privacy and scalability in RFID systems using PUF’s,” Ad Hoc Networks, vol. 32, pp. 32–42, 2015.
[10] A. Aysu, E. Gulcan, and D. M. B, “End-To-End Design of a PUF-Based Privacy,” 17th Int. Conf. Cryptogr. Hardw. Embed. Syst., vol. 2, pp. 556–576. 2015.
[11] C. H. B, A. Aysu, J. Guajardo, P. Duplys, and G. Tim, “Secure and Private , yet Lightweight , Authentication for the IoT via PUF and CBKA,” Inf. Secur. Cryptol., vol. 1, pp. 28–48, 2017.
[12] P. Gope, J. Lee, and T. Q. S. Quek, “Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions,” IEEE Trans. Inf. FORENSICS Secur., vol. 13, no. 11, pp. 2831–2843, 2018.
[13] F. Zhu, P. Li, and H. Xu, “A Lightweight RFID Mutual Authentication Protocol with PUF,” Sensors (Basel), no. January, pp. 1–22, 2019.
[14] D. Hein, J. Wolkerstorfer, and N. Felber, “ECC Is Ready for RFID – A Proof in Silicon,” (ed)Selected Areas Cryptogr. SAC 2008. Lect. Notes Comput. Sci., vol. 5381, pp. 401–413, 2009.
[15] M. Hutter, M. Feldhofer, and T. Plos, “An ECDSA Processor for RFID Authentication,” RadioFreq. Identif. Secur. Priv. Issues. RFIDSec 2010. Lect. Notes Comput. Sci., vol. 6370, pp. 189–202. 2010.
[16] A. Gildas, B. A. Muhammed, C. Xavier, and Y. O. B. Siddika, “Privacy-Friendly Authentication in RFID Systems: On Sublinear Protocols Based on Symmetric-Key Cryptography,” IEEE Trans. Mob. Comput., no. October, pp. 2037–2049, 2013.
[17] S. Kim, S. K. Kim, M. Ohkubo, F. Hoshino, and G. Morohashi, “Privacy Enhanced Active RFID Tag,” Int. Work. Exploit. Context Hist. Smart Env., pp. 43–52, 2005.
[18] C. Y. Ng, W. Susilo, Y. Mu, and R. Safavi-naini, “New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing,” Comput. Secur. – ESORICS 2009. ESORICS 2009. Lect. Notes Comput. Sci., vol. 5789, pp. 321–336. 2009.
[19] C. Lee and H. Chien, “An Elliptic Curve Cryptography-Based RFID Authentication Securing E-Health System,” Int. J. Distrib. Sens. Networks, vol. 2015, p. 7, 2015.
[20] C. C. Tan, B. Sheng, and Q. Li, “Secure and Serverless RFID Authentication and Search Protocols,” IEEE Trans. Wirel. Commun., vol. 7, no. 4, pp. 1400–1407, 2008.
[21] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim, “Mutual Authentication Protocol for Low-cost RFID,” Work. RFID Light. Cryptogr., no. February, pp. 17–24, 2015.

[22] Z. Luo, T. Chan, and J. S. Li, “A Lightweight Mutual Authentication Protocol for RFID Networks,” IEEE Int. Conf. E-bus. Eng., pp. 620–625, 2005.

[23] P. Gope and T. Hwang, “A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system,” Comput. Secur., vol. 55, pp. 271–280, 2015.

[24] H. Xu, “A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function,” Sensors (Basel.), pp. 1–20, 2018.

[25] Y. Bendavid and N. Bagheri, “IoT Device Security : Challenging “ A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function,” Sensors (Basel.), vol. 18, no. 12, 2018.

[26] G. E. Suh and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” 2007 44th ACM/IEEE Des. Autom. Conf. San Diego, pp. 9–14, 2007.

[27] C. Bohm and M. Hofer, Physical Unclonable Functions in Theory and Practice. New York: NY, USA: Springer, 2012.

[28] D. Jeon, J. H. Baek, D. K. Kim, and B. Choi, “Toward Zero Bit-Error-Rate Physical Unclonable Function : Mismatch-Based vs . Physical-Based Approaches in Standard CMOS Technology,” 2015 Euromicro Conf. Digit. Syst. Des., pp. 407–414, 2015.

[29] K. Chuang et al., “A Physically Unclonable Function Using Soft Oxide Breakdown Featuring 0 % Native,” IEEE J. Solid-State Circuits, vol. 54, no. 10, pp. 2765–2776, 2019.

[30] P. P. Crystals, X. Lu, S. Member, L. Hong, and S. Member, “CMOS Optical PUFs Using Noise-Immune Incorporating Passive Variations,” IEEE J. Solid-State Circuits, vol. 53, no. 9, pp. 2709–2721, 2018.

[31] W. Wang, Y. Yona, S. N. Diggavi, and P. Gupta, “Design and Analysis of Stability-Guaranteed PUFs,” IEEE Trans. Inf. FORENSICS Secur., vol. 13, no. 4, pp. 978–992, 2018.

[32] P. Tuyls and L. Batina, “RFID-Tags for Anti-counterfeiting,” Top. Cryptol. CT-RSA (LNCS 3860), Heidelberg, Ger. Springer, pp. 115-131, 2006.

[33] D. Dolev, “On the Security of Public Key Protocols,” IEEE Trans. Inf. THEORY, no. M, pp. 198–208, 1983.

E Haodudin Nurkifli received S.T. degree from The Faculty of Engineering, Universitas Ahmad Dahlan (UAD), Indonesia, in 2008, and M.Cs. degree in Computer Science from Universitas Gadjah Mada (UGM), Indonesia, in 2012. He is currently pursuing a Ph.D. degree in computer science and information engineering with National Cheng Kung University (NCKU), Tainan, Taiwan. Works as a research assistant in NCKU and a lecturer in Universitas Singaperbangsa Karawang, Indonesia. His research primarily focuses on cryptography, authentication, security in the IoT, IoT, Smart Grid, WSN environment, hardware security PUF-based.