
Article:
Gope, P. orcid.org/0000-0003-2786-0273 and Sikdar, B. (2019) Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE Internet of Things, 6 (1). pp. 580-589. ISSN 2327-4662

https://doi.org/10.1109/jiot.2018.2846299

Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices

Prosanta Gope and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Device authentication is an essential security feature for Internet of Things (IoT). Many IoT devices are deployed in the open and public places, which makes them vulnerable to physical and cloning attacks. Therefore, any authentication protocol designed for IoT devices should be robust even in cases when an IoT device is captured by an adversary. Moreover, many of the IoT devices have limited storage and computational capabilities. Hence, it is desirable that the security solutions for IoT devices should be computationally efficient. To address all these requirements, in this article, we present a lightweight and privacy-preserving two-factor authentication scheme for IoT devices, where physically uncloneable functions (PUFs) have been considered as one of the authentication factors. Security and performance analysis show that our proposed scheme is not only robust against several attacks, but also very efficient in terms of computational efficiently.**

***Index Terms*—Mutual authentication, Privacy-Preserving, Physically uncloneable functions, Fuzzy extractor, IoT device.**

I. Introduction

THE Internet of Things refers to the environment or framework which enables everyday objects in our world to have network connectivity and the ability to send and receive data. Usually, devices in IoT systems have limited power, storage, and processing capabilities. In addition, IoT devices are often deployed in the open and public places, which may cause them to be vulnerable to physical and cloning attacks. Therefore, it is important that any security solution designed for IoT devices should not only be efficient but also detect any violations of physical security of the IoT devices. In these scenarios, conventional password-based or secret-key-based authentication schemes, in which a shared secret is the only authentication factor, is not enough for addressing the security problems. In this context, an adversary who has physical access to an IoT device can launch various physical or side-channel attacks to acquire the device's secret, and thus compromise the device and the entire system. To overcome the above problem, we need a two-factor authentication scheme that can ensure a layered defense and at the same time, make it harder for unauthorized individuals to gain control of the IoT devices. The major benefit of two-factor authentication is to provide a more resilient way of authenticating IoT devices. From the attackers' perspective, multiple barriers have to be overcome in order to break the security of the IoT devices.

To provide two-factor authentication to IoT devices, in addition to a password or a shared secret key as the first authentication factor, this paper proposes the use of physically uncloneable functions [1-2] as the second authentication factor. PUFs have emerged as a promising cryptographic primitive and already gained popularity in the security domain, and their practicality has also been demonstrated in many recent works. PUFs are the result of the manufacturing process of Integrated Circuits (ICs) which introduces random physical variations into the micro-structure of an IC, making it unique. It is impossible to control these variations in the micro-structure of an IC during the manufacturing process. In addition, the outputs are derived from intrinsic characteristics of the PUF's physical elements, and are therefore difficult to predict and almost impossible to clone. In this regard, PUF uses their internal structure to provide a one-way function that cannot be duplicated. The fact that PUFs are hard to predict but easy to construct and evaluate makes them a good choice for use as a security primitive for IoT devices.

A. Related Work

Many two-factor authentication schemes have been proposed in the recent years. However, majority of these schemes [22-24] are user centric. In these schemes, passwords and smart cards/devices are used as two-factor security. Since smart cards are not tamper proof, these schemes are often vulnerable to several physical attacks. On the other hand, recently a few interesing PUF-based authentication schemes have been proposed for IoT systems [3-6]. However, most of them are based on computationally inefficient public key systems. More recently, some PUF-based authentication protocols using symmetric key cryptosystems have been proposed. Most of these works are mainly focused on reliably computing a PUF response to a challenge [7-8]. Similarly, some literature describe techniques for implementing authentication protocols on reconfigurable hardware for the purpose of intellectual property (IP) protection [9-10]. On the other hand, PUFs are also used for designing authentication protocols for wireless sensor networks (WSNs) and radio frequency identification (RFID) systems [11-13]. Recently Aman et al. proposed two PUF-based mutual authentication protocols for IoT systems [14]. However, their scheme cannot ensure the privacy of the IoT devices. In addition, noise and sensitivity to environmental factors are still important factors in PUF design, which may result in one or several of the output bits of the PUF being

P. Gope, is with Department of Computer Science, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077. (E-mail: prosanta@comp.nus.edu.sg/prosanta.nitdgp@gmail.com)

B. Sikdar is with Department of Electrical and Computer Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077. (Email: bsikdar@nus.edu.sg)

Corresponding author: B. Sikdar

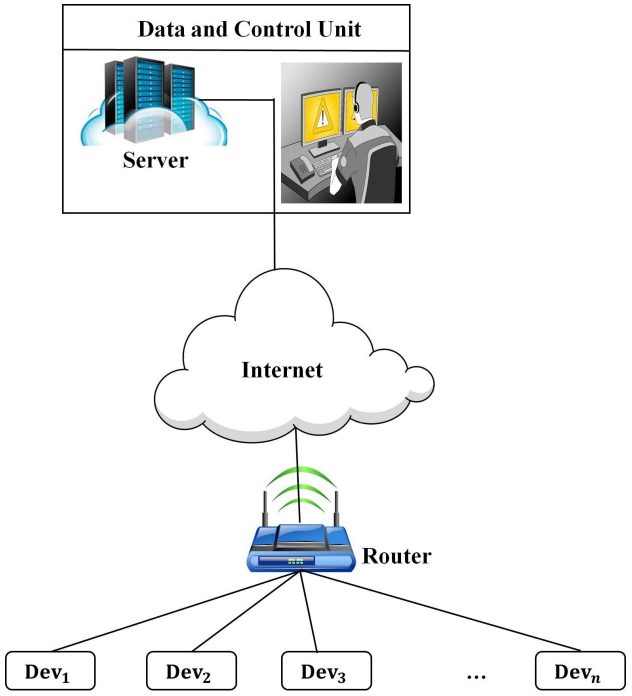

Figure 1. System Model.

incorrect for any challenge. However, the scheme presented in [14] does not support noisy PUF environment.

To address all the above issues, in this article we propose a lightweight and privacy-preserving two-factor authentication scheme for IoT devices. In our proposed scheme, PUFs have been considered as one of the authentication factors. Moreover, to address the issue of noise during the PUF's operation, the concept of *reverse fuzzy extractor* has been exploited. In a nutshell, this article makes the following three major contributions:

(i) A novel privacy-preserving two-factor authentication protocol for IoT devices.
(ii) Consideration of noise factor in the PUF design.
(iii) A computationally efficient security solution, which is feasible for resource constraint IoT devices.

The rest of the article is organized as follows. In Section II, we first provide a brief introduction to PUFs and *fuzzy extractors*. This section ends with the description of the system model of our proposed system. In Section III, we present our proposed privacy-preserving two-factor authentication protocol for IoT devices. Security of the proposed scheme is analyzed in Section IV. Performance analysis of the proposed protocol is then provided in Section V. In Section VI, we formaly analyzed the security odf our proposed scheme using BAN logic. Finally, conclude our article with concluding remarks in Section VII. The symbols and cryptographic functions used in the proposed scheme are defined in Table I.

Table I
SYMBOLS AND CRYPTOGRAPHIC FUNCTION

Symbol	Definition
AID	One-time alias identity
$CRP(C, R)$	Challenge-Response pair
sk	Session key between D_i and server
PUF_{D_i}	Physically uncloneable functions of D_i
$h(\cdot)$	One-way hash function
$\oplus$	Exclusive-OR operation
FE	Fuzzy extractor
$\|$	Concatenation operation

II. PRELIMINARIES AND SYSTEM MODEL

A. Fuzzy Extractor

A fuzzy extractor (d, λ) [15-18] is composed with two algorithms: FE.Gen and FE.Rec. FE.Gen is a probabilistic key generation algorithm, which takes a bit string R as input and outputs a key K and helper data hd, i.e., (K, hd) = FE.Gen(R). On the other hand, FE.Rec is a deterministic reconstruction algorithm that recovers the key K from the noisy input variable R' and the helper data hd i.e., K = FE.Rec (R', hd), if the Hamming distance between R' and R is at most d. A fuzzy extractor (FE) ensures security in the extraction of a strong cryptographic key if the min-entropy of the input R is at the minimum λ, and K is close to a uniformly random distribution in $\{0,1\}^k$. Since repeated exposure of the helper data may result in additional min-entropy loss [17-18], the helper data should not be exposed during the execution of the authentication protocol.

B. Physically Uncloneable Function

A PUF is characterized by a challenge-response pair (*CRP*). It is an IC which takes a string of bits as an input challenge and produces a arbitrary string of bits called the response. The response R of a PUF PUF_D to a challenge C can be represented as follows: $R = PUF_D(C)$. We say PUF_D is a $(d, n, l, \lambda, \epsilon)$-secure PUF if the following requirements hold:

1) For any two PUFs $PUF_{D1}(\cdot)$ and $PUF_{D2}(\cdot)$, and $C_1 \in \{0,1\}^k$, $\Pr[\text{HD}(PUF_{D1}(C_1), PUF_{D2}(C_2) > d] \geq 1 - \varepsilon$. Here, HD represents the Hamming distance.
2) For any PUF $PUF_D(\cdot)$ and for any input $C_1, \cdots, C_n \in \{0,1\}^k$, $\Pr[\hat{H}_\infty(PUF_D(C_i), PUF_D(C_j))_{1 \leq i,j \leq n, i \neq j} > \lambda] \geq 1 - \varepsilon$, which denotes that the min-entropy of the PUF output is always larger than λ with high probability, when the intra-distance, i.e., the distance between two PUF responses from the same PUF instance and using the same challenge is smaller than d, and the inter-distance, i.e., the distance between two PUF responses from different PUF instances using the same challenge is greater than d.

C. System Model

In this paper, we consider the same system model as that proposed in Aman et al.'s scheme [14]. The system model is

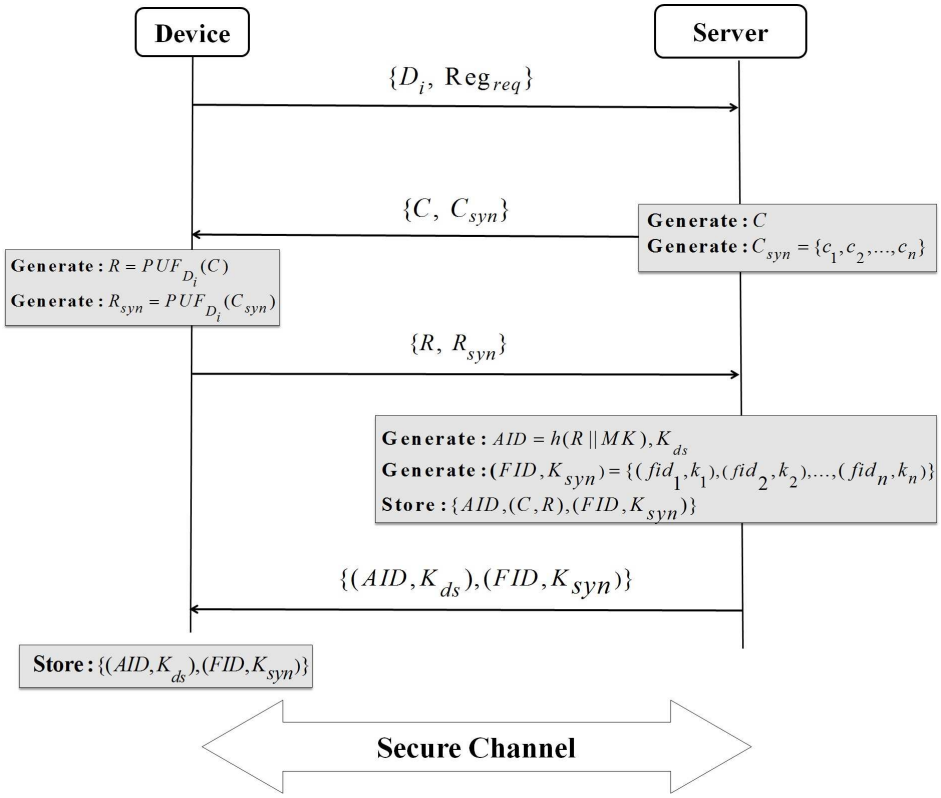

Figure 2. Setup Phase of the Proposed Scheme

composed of two major entities: a set of IoT devices and a server located in a data and control unit. Here, IoT devices can communicate and send their data to the server of a data and control unit by using the Internet. It is assumed that all the IoT devices are equipped with a PUF, where any attempt to tamper with the PUF will change the behavior of the device and render the PUF useless. In addition, here we also assume that IoT devices have limited resources while the server in the data center is trusted and has no such resource limitation. Our system model is depicted in Fig. 1.

III. Proposed Scheme

In this section, we present a practical anonymous authentication scheme, which consists of two phases: Setup, and Authentication.

A. Setup Phase

The operations of the setup phase are carried out over a secure channel. To start the setup phase, an IoT device D_i sends its identity along with a registration request to the server. Upon receiving the request, the server first randomly generates a challenge C for the next interaction with the device D_i. Then the server also generates a set of new challenges $C_{syn} = \{c_1, \cdots, c_n\}$ for resynchronization with device D_i and sends $\{C, C_{syn}\}$ to the device. After receiving the challenges $\{C, C_{syn}\}$, the device extracts the PUF outputs $R = PUF_{D_i}(C)$ and $R_{syn} = PUF_{D_i}(C_{syn})$, and sends $\{R, R_{syn}\}$ to the server. Hereafter, the server first generates a one-time alias identity $AID = h(R||MK)$, and a secret key K_{ds}, which will be used as the first authentication factor for proving the legitimacy of the IoT device D_i. Here, MK denotes the master key of the server. Next, the server also generates a set of unique fake identity and synchronization key pairs $(FID, K_{syn}) = \{(fid_1, k_1), \cdots, (fid_n, k_n)\}$ and sends $\{(AID, K_{ds}), (FID, K_{syn})\}$ to device D_i. Finally, for IoT device D_i, the server will store $\{(AID, K_{ds}), (C, R), (C_{syn}, R_{syn}), (FID, K_{syn})\}$ in its database and the device stores $\{(AID, K_{ds}), (FID, K_{syn})\}$. Details of this phase are depicted in Fig. 2.

B. Authentication Phase

Our authentication phase consists of the following steps:

Step 1 (Request for Interaction): When a IoT device D_i wants to interact with the server, then the device first selects the one-time alias identity AID. It then generates a random number N_d and computes $N_d^* = N_d \oplus K_{ds}$. Finally, the device composes a request message $\text{M}_1 : \{AID, N_d^*\}$ and sends it to the server for interaction.

Step 2 (Server Response): After receiving the authentication request message M_1, the server first locates one-time alias identity AID in its database and subsequently reads and loads $\{(C, R), K_{ds}\}$ into its memory. Hereafter, the server generates a nonce N_s and computes $N_s^* = K_{ds} \oplus N_s$, a key-hash response $V_0 = h(N_d||K_{ds}||N_s^*)$ and then composes a response message $\text{M}_2 : \{C, N_s^*, V_0\}$ and sends it to the device.

Step 3 (Server Authentication): Next, upon receiving response message M_2, the device extracts the PUF output $R^{'} = PUF_{D_i}(C)$ and subsequently computes then checks

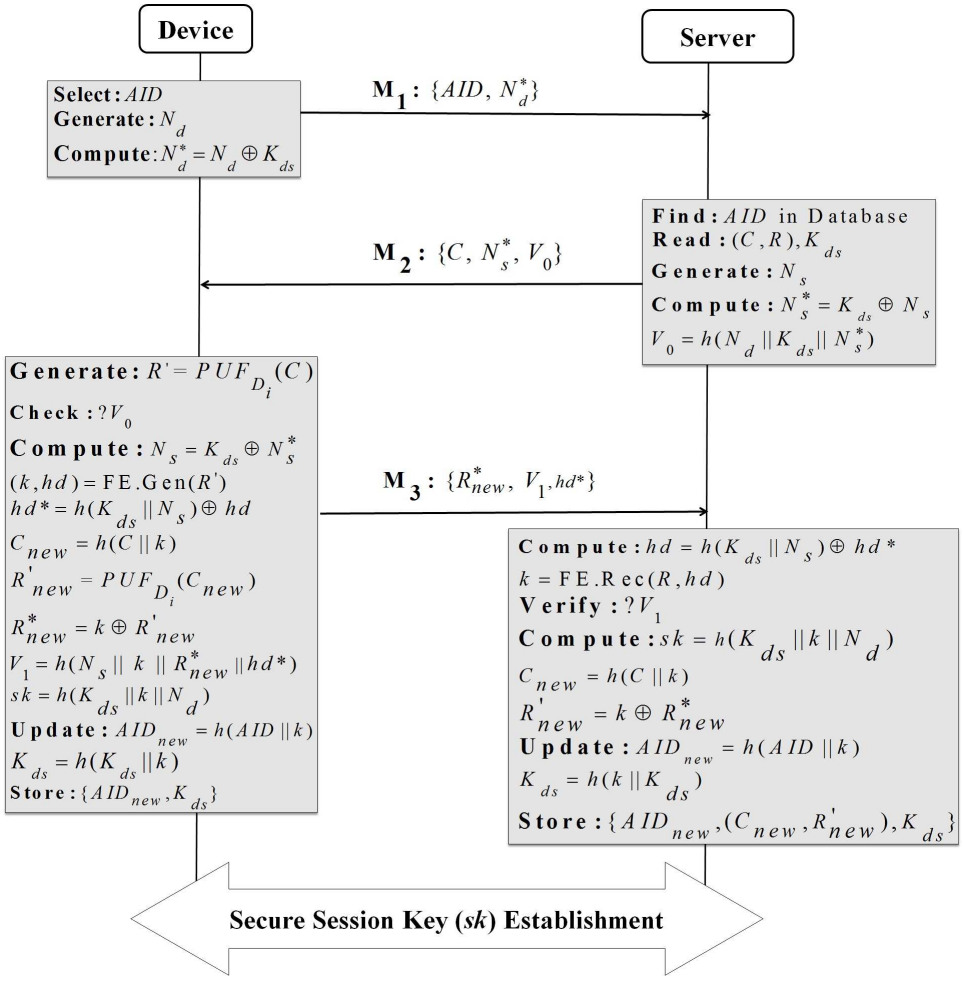

Figure 3. Proposed Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices

the key-hash response V_0. If it is not valid, the device terminates the execution of the protocol. Otherwise, the device authenticates the server and decodes $N_s = K_{ds} \oplus N_s^*$, obtains the key-element and heper data from the helper data generation algorithm FE.Gen i.e., $(k, hd) = \text{FE.Gen}(R^{'})$. After that the device calculates $hd^* = h(K_{ds}||N_s) \oplus hd$, $C_{new} = h(C_i||K_i)$, $R^{'}_{new} = PUF_{D_i}(C_{new})$, $R^*_{new} = k \oplus R^{'}_{new}$, $V_1 = h(N_s||k||R^*_{new}||hd^*)$, $AID_{new} = h(AID||k)$, $K_{ds} = h(K_{ds}||k)$, and the session key $sk = h(K_{ds}||k||N_d)$. Then, the device forms a message $\text{M}_3 : \{R^*_{new}, V_1, hd^*\}$ and sends it to the server.

Step 4 (Device Authentication): After receiving message M_3, the server first computes and decodes the helper data $hd = h(K_{ds}||N_s) \oplus hd^*$, and obtains the key-element $k = \text{FE.Rec}(R, hd)$ from the reconstruction algorithm FE.Rec. Hereafter the server verifies the key-hash response V_1. If the verification is successful, then the server authenticates the device and calculates the session key $sk = h(K_{ds}||k||N_d)$. After that, the server computes the new challenge $C_{new} = h(C||k)$, and decodes the new PUF output $R^{'}_{new} = k \oplus R^*_{new}$ and updates the alias identity $AID_{new} = h(AID||k)$, and the $K_{ds} = h(k||K_{ds})$. Finally, the server stores $\{(AID_{new}, K_{ds}), (C_{new}, R^{'}_{new})\}$ for the next interaction with the device.

Now, if the server cannot recognize the IoT device in Step 2, then the server asks the device to try again by using one of the unused pairs of $(\mathit{fid}_x, k_x) \in (FID, K_{syn})$. Once a pair is used up, it must be deleted from both the ends. In this case, the server will select one of the unused CRPs from (C_{syn}, R_{syn}) and a new alias identity will be provided to the device. Finally, the CRP for this resynchronization also needs to be deleted from (C_{syn}, R_{syn}). In this way, the proposed scheme can handle the desynchronization problem without compromising anonymity support. Details of this phase are depicted in Fig. 3.

IV. Security Model and Analysis

In this section, we first define our security and privacy model and subsequently, we use them to analyze the security of the proposed scheme.

A. Security Model

Consider a set of IoT devices $\mathcal{D} = \{D_1, D_2, \cdots, D_n\}$ that communicate with the trustworthy server S of the data and control unit. The server executes a setup algorithm $\text{Setup}(1^k)$ for enrolling into a trusted environement and a public parameter pp and secret key K_{ds} are generated for initialization.

Here, pp denotes all the available public parameters (crypto suites) of the environment (e.g., PUF output length, coding mode, pseudo-random function (PRF) algorithm name, etc.) In the authentication phase of the proposed scheme, these parties communicate through an insecure network and mutually authenticate each other. At the end of the authentication process, the parties output 1 (Accept) or 0 (Reject) as the outcome of the authentication process, respectively. We call the communication sequence between the two parties (the server, and the IoT device) is a unique session and a session identifier sid is used for distinguishing each session. We say that a session has a matching session if the messages exchanged between S and devices in $\mathcal{D}$ are honestly transferred until they authenticate each other.

We now consider the following security game (denoted by $\mathrm{Exp}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(\lambda)$) between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$ against a mutual authentication protocol Π:

$\mathrm{Exp}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(\lambda)$:

1) $(pp, K_{ds}) \xleftarrow{Rand} \mathrm{Setup}(1^{\lambda})$;
2) $(\mathrm{sid}^*, D_j) \xleftarrow{Rand} \mathcal{A}_1^{\mathrm{Launch},\mathrm{Send}S,\mathrm{Send}\mathcal{D},\mathrm{Result},\mathrm{Reveal}}(pp, S, \mathcal{D})$;
3) $b := \mathrm{Result}(\mathrm{sid}^*, D_j)$;
4) Output b.

At the end of the setup phase, $\mathcal{A}$ can issue the following oracle queries:

- $\mathrm{Launch}(1^{\lambda})$: A new session is started by S.
- $\mathrm{Send}S$: A random message m is sent to S.
- $\mathrm{Send}\mathcal{D}(D_j, m)$: An arbitrary message m is sent to device $D_j \in \mathcal{D}$.
- $\mathrm{Result}(\mathcal{P}, sid)$: Output whether session sid of $\mathcal{P}$ is accepted or not where $\mathcal{P} \in \{S, \mathcal{D}\}$.
- $\mathrm{Reveal}(D_j)$: Output all information contained in the memory of the device D_j.

The advantage of the adversary $\mathcal{A}$ against Π, denoted by $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(\lambda)$, is defined as the probability that $\mathrm{Exp}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(\lambda)$ outputs 1 when sid^* of $\mathcal{P}$ has no matching session.

Definition 1. *An authentication protocol* Π *is resilience to the man-in-the-middle attacks with key compromise if for any probabilistic polynomial time adversary* $\mathcal{A}$, $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible, i.e.,* $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(\lambda) \leq \epsilon$, *(for large enough* λ*).*

B. Privacy Model

Now we consider the indistinguishability-based privacy. In that case, the adversary selects two IoT devices and tries to distinguish the communication derived from the two devices. The privacy experiment between the challenger $\mathcal{C}$ and adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ is then described as follows:

$\mathrm{Exp}^{\mathrm{IND}*-\mathrm{b}}_{\Pi,\mathcal{A}}(\lambda)$:

- $(D_0^*, D_1^*, st_1) \xleftarrow{Rand} \mathcal{A}_1^{\mathrm{Launch},\mathrm{Send}S,\mathrm{Send}\mathcal{D},\mathrm{Result},\mathrm{Reveal}}(pp, S, \mathcal{D})$;
- $b \xleftarrow{U} \{0,1\}$, $\mathcal{D}^{'} := D \backslash \{D_0^*, D_1^*\}$;
- $\Pi_0 \xleftarrow{Rand} \mathrm{Execute}(S, D_0^*)$, $\Pi_1 \xleftarrow{Rand} \mathrm{Execute}(S, D_1^*)$, $st_2 \xleftarrow{Rand} \mathcal{A}_2^{\mathrm{Launch},\mathrm{Send}S,\mathrm{Send}\mathcal{D},\mathrm{Result},\mathrm{Reveal}}(S, \mathcal{D}^{'}, \mathcal{I}(D_b^*), \Pi_0, \Pi_1, st_1)$;
- $\Pi_0^{'} \xleftarrow{Rand} \mathrm{Execute}(S, D_0^*)$, $\Pi_1^{'} \xleftarrow{Rand} \mathrm{Execute}(S, D_1^*)$, $b^{'} \xleftarrow{Rand} \mathcal{A}_3^{\mathrm{Launch},\mathrm{Send}S,\mathrm{Send}\mathcal{D},\mathrm{Result},\mathrm{Reveal}}(S, \mathcal{D}, \Pi_0^{'}, \Pi_1^{'}, st_1)$;
- Output $b^{'}$;

After the execution of the setup phase, the adversary $\mathcal{A}_1$ issues the oracle queries and sends the queries with IoT device identities (D_0^*, D_1^*) to challenger $\mathcal{C}$. After that, $\mathcal{C}$ flips a random coin $b \xleftarrow{U} \{0,1\}$ and allows the adversary to communicate with D_b^* in an anonymous way. For the accomplishment of anonymous access, $\mathcal{A}_2$ calls the $\mathrm{Send}D$ query with intermediate algorithm $\mathcal{I}$ as the input to honestly transfer the communication message between $\mathcal{A}_2$ and D_b^*. After the challenge phase, $\mathcal{A}_3$ can continuously interact with all devices, including (D_0^*, D_1^*), as $\mathcal{A}_1$. Next, D_0^* and D_1^* call the Execute query to avoid trivial attacks (e.g. man-in-the-middle attacks) in the symmetric key based construction, and after that, they send their transcripts (Π_0, Π_1) and $(\Pi_0^{'}, \Pi_1^{'})$ to the adversary. The advantage of the adversary in guessing the correct tag bit can be defined as

$$\mathrm{Adv}^{\mathrm{IND}*}_{\Pi,\mathcal{A}}(\lambda) := |\Pr[\mathrm{Exp}^{\mathrm{IND}*-0}_{\Pi,\mathcal{A}}(\lambda) \rightarrow 1] - \Pr[\mathrm{Exp}^{\mathrm{IND}*-1}_{\Pi,\mathcal{A}}(\lambda) \rightarrow 1]|.$$

C. Security Analysis of the Proposed Authentication Protocol

Next we consider the above models for analyzing the security of the proposed authentication protocol.

Theorem 1:*Let* h *be a secure pseudorandom function,* FE *be a* (d, λ)*-fuzzy extractor, and consider a* $(d, n, l, \lambda, \epsilon)$*-secure physically uncloneable function. Then, the proposed mutual authentication protocol is secure against man-in-the-middle attacks with memory leakage.*

Proof. The goal of the adversary $\mathcal{A}$ is to violate the security experiment and convince the device and the server to accept the session without a corresponding matching session, while communication is modified by the adversary. Now we consider the following game transformations. Let $\mathcal{X}_i$ be the advantage that the adversary wins the game in Game *i*.

Game 0: It represents the original game between the challenger $\mathcal{C}$ and the adversary.

Game 1: $\mathcal{C}$ randomly guesses the identity of the device $D^* \xleftarrow{U} \{D_1, \cdots, D_n\}$. If the adversary does not impersonate D^*, then $\mathcal{C}$ aborts the game.

Game 2: Assume that l is the upper bound on the number of sessions that the adversary can establish in the game. For $1 \leq j \leq l$, we evaluate or change the related variables in the session between the server unit and D^* as per following games and its variations:

- **Game 2**$(j, 1)$**: In the *j*-th session,** $\mathcal{C}$ **evaluates the output of the PUF implemented in** D^***. If the output of the PUF does not have enough entropy or is correlated to the other outputs derived from the inputs to the PUF,** $\mathcal{C}$ **aborts the game.**
- **Game 2**$(j, 2)$**: The output from the fuzzy extractor** (k, hd) **is turned into a random variable.**
- **Game 2**$(j, 3)$**: In this game the output from the pseudorandom functions (PRF)** $h(k, \cdot)$ **and** $h(K_{ds}, \cdot)$ **is derived from a truly random function.**

- **Game 2$(j,4)$: In this game the output from the PRF $h(K_{syn},\cdot)$ is derived from a truly random function.**
- **Game 2$(j,5)$: In this game, we alter the XORed output $R^*_{new} = k \oplus R^{'}_{new}$, and $hd^* = h(K_{ds}||N_s) \oplus hd$ to arbitrarily chosen $R^*_{new}, hd^* \underset{\leftarrow}{\cup} \{0,1\}^{|R^*_{i+1},hd^*|}$.**

The main idea of the security proof is to modify the messages corresponding to the IoT device D^* to arbitrary strings. We proceed with the game transformation starting with the first call of the device D^*. After that, we gradually change the communication message from Game $2(j,1)$ to Game $2(j,5)$. We move to the next section, once these transformations are finished. Through these game transformations, we show that the advantage of the adversary against the authentication protocol can be limited to negligible values as shown in the results of Lemma 1 through 5. ■

Lemma 1: *If the numbers of IoT devices is n, then $\mathcal{X}_0 = n\mathcal{X}_1$.*

Proof. We say the adversary wins the game when she/can can convince the device or server to accepts the session while communication is modified by the adversary. Since we consider that there are n IoT devices, $\mathcal{C}$ correctly guess the related session with probability $1/n$. ■

Lemma 2: *If PUF_{D_i} is a (d,n,l,λ,ϵ)-secure PUF, then $\mathcal{X}_1 = \mathcal{X}_{2(j,1)}$ and $\mathcal{X}_{2(j,5)} = \mathcal{X}_{2(j,1)}$ for any $2 \leq j \leq l$.*

Proof. Given that the PUF is (d,n,l,λ,ϵ)-secure, its intra-distance is less than d, its inter-distance is larger than d, and the min-entropy of the PUF is lager than λ. In addition, the PUF also has the property that even if the input to the PUF is exposed, the output derived from the input maintains sufficient min-entropy property and the outputs are thus uncorrelated. Now, if an adversary issues the *reveal* query and obtains the stored information from the PUF's memory, then, since the games in $\mathcal{X}_1$, $\mathcal{X}_{2(j,1)}$ and $\mathcal{X}_{2(j-1,5)}$ are based on the above condition, the gap between them is bounded by ϵ. Therefore, we can write $\left|\mathcal{X}_1 - \mathcal{X}_{2(j,1)}\right| \leq \epsilon$ and $\left|\mathcal{X}_{2(j,5)} - \mathcal{X}_{2(j,1)}\right| \leq \epsilon$. This means there is no effect on proceeding with the game transformations. ■

Lemma 3:*If the* FE *algorithm is a (d,λ)- secure fuzzy extractor, then no attacker can distinguish the difference between the game $\mathcal{X}_{2(j,1)}$ and $\mathcal{X}_{2(j,2)}$, $\forall\ 0 \leq j \leq l$*

Proof. As mentioned in the proof of Lemma 2, the PUF used in the protocol ensures a min-entropy of λ. Then the operation of the (d,λ) fuzzy extractor ensures that the output of the fuzzy extractor is close to random and no adversary can distinguish the difference between Game $2(j,1)$ and Game $2(j,2)$. Therefore, the advantage of the adversary in distinguishing between these two games can be represented as $\left|\mathcal{X}_{2(j,2)} - \mathcal{X}_{2(j,1)}\right| \leq \epsilon$. ■

Lemma 4: *Let $\mathrm{Adv}^{\mathrm{PRF}}_{h(\cdot),\beta}(k)$ denote the advantage of β to break the security of the PRF $h(\cdot)$. Then, $\forall 1 \leq j \leq l$, we have $\left|\mathcal{X}_{2(j,2)} - \mathcal{X}_{2(j,3)}\right| \leq \mathrm{Adv}^{\mathrm{PRF}}_{h(\cdot),\beta}(k)$.*

Proof. Now, an algorithm β is constructed which breaks the security of the PRF $h(\cdot)$. β sets up all the security credentials and simulates our protocol except for the i-th session (the current session). β can access the real PRF $h(k,\cdot)$ or a truly random function. When the adversary invokes the i-th session, β sends the uniformly random challenge $\{N^*_s \underset{\leftarrow}{\cup} \{0,1\}^k\}$ as the output of the server. When $\mathcal{A}$ sends $N^{\#}_s$ to the device, β continues the computations as per the protocol specification and issues $N^{\#}_s$ to the oracle instead of the normal computation of $h(\cdot)$. After receiving V_1, β outputs $\{R^*_{new}, hd^*, V_1\}$ as the response of the device. When the adversary sends $\{R^{\#}_{new}, hd^{\#}, V^{\#}_1\}$, β issues $N^{\#}_s$ to the oracle and obtains V_1, which is used to authenticate the device.

If β accesses the real PRF, this simulation is equivalent to the Game $2(j,2)$. Otherwise, the oracle query issued by β is completely random, and its distribution is equivalent to that in Game $2(j,3)$. Therefore, we can write $|\mathcal{X}_{2(j,2)} - \mathcal{X}_{2(j,3)}| \leq \mathrm{Adv}^{\mathrm{PRF}}_{h(\cdot),\beta}$. ■

Lemma 5: $\forall\, 1 \leq j \leq l$, $|\mathcal{X}_{2(j,3)} - \mathcal{X}_{2(j,4)}| \leq \mathrm{Adv}^{\mathrm{PRF}}_{h(\cdot),\beta}(k)$.

Proof. The proof for lemma follows along the lines of the proof for Lemma 4. ■

Lemma 6: $\forall\, 1 \leq j \leq l$, *we have* $\mathcal{X}_{2(j,2)} = \mathcal{X}_{2(j,4)} = \mathcal{X}_{2(j,5)}$.

Proof. In the three games considered in this lemma, the fuzzy extractor FE and the PRF $h(\cdot)$ are changed to the truly random function. Therefore, k and $h(K_{ds}||N_s)$ are used as effective one-time pads to encode $R^{'}_{new}$ and hd_{new}, respectively. Therefore, no adversary can differentiate $R^*_{new} = k \oplus R^{'}_{new}$, and $hd^* = h(K_{ds}||N_s) \oplus hd$ from a randomly chosen string. ■

Theorem 2: *Let* FE *be a (d,λ) fuzzy extractor and consider a (d,n,l,λ,ϵ)-secure physically uncloneable function. Also, let h be a secure pseudorandom function. Then the proposed protocol satisfies indistinguishability-based privacy.*

Proof. The proof for this theorem is similar to that for Theorem 1, where we have shown that the proposed authentication protocol holds security against forgery attacks. According to the game transformation described in the proof of Theorem 1, if we continuously modify the communication messages for device D^*_0 and D^*_1, then the whole transcript will be identical to a random string. Thus, no information that identifies the challenger's coin will be leaked. Since all the identity related parameters stored in the memory such as $\{(AID, K_{ds}), (FID, K_{syn})\}$ are randomly generated and each pair can only be used once, these parameters do not provide any information about the real identity of the device. The probability that the challenger can identify D^*_0 and D^*_1 so the game transformation is finished within a polynomial time is $1/n^2$. Therefore, we can argue that the proposed scheme holds indistinguishability-based privacy. ■

D. Informal Security Analysis

We now provide an intuitive reasoning to demonstrate how the proposed protocol fulfills some of the security requirements such as mutual authentication, privacy of the IoT devices, etc.

1) *Mutual Authentication*: In the proposed scheme, only the legitimate device D_i with the correct *two factors* (i.e., secret key K_{ds}, and PUF PUF_{D_i}) can obtain $N_s = K_{ds} \oplus N^*_s$, $R^{'} = PUF_{D_i}(C)$, and $(k, hd) = \mathrm{FE.Gen}(R^{'})$ to generate a valid key-hash response $V_1 = h(N_s||k||R^*_{new}||hd^*)$. Thus, the server can authenticate the device by using the parameter V_1. On the other hand, only the server who knows the secret key K_{ds} can

compose a valid respond message M_2. Thus, the device can authenticate the server when it can successfully validate the key-hash output $V_0 = h(N_d||K_{ds}||N_s^*)$. Therefore, the proposed protocol is able to provide mutual authentication.

2) *Session Key Agreement*: In the proposed scheme, at the end of the mutual authentication phase, both the device and the server share the identical session key $sk = h(K_{ds}||k||N_d)$. Therefore, the proposed scheme is able to provide session key agreement.
3) *Privacy of the IoT Devices*: During the execution of the proposed authentication protocol, for each session, a device needs to use a valid one-time alias identity AID which cannot be used twice. Therefore, no one except the server can recognize the activity of the IoT device. Besides, in case of loss of synchronization, the device needs to use one of the unused fake identities fid_j from $FID = \{fid_1, \cdots, fid_n\}$. After that, the device needs to delete this identity from its memory. Therefore, changing the identities in each session ensures identity intractability. This approach of the proposed scheme is quite useful for achieving privacy against eavesdropper (PAE).
4) *Protection Against Physical Attacks*: Suppose an adversary wants to perform physical tampering on an IoT device in order to compromise it or influence its behavior. However, any such attempt to tamper with the device changes the behavior of the PUF embedded in it and renders the PUF useless. Consequently, during the execution of the proposed authentication protocol, the PUF will not be able to produce the desired output $R^{'} = PUF_{D_i}(C)$. Therefore, the server can comprehend such attempts at tampering. On the other hand, since PUFs are safe against cloning and a PUF cannot be recreated [19], the proposed scheme can be considered safe against cloning attack.
5) *Protection Against Impersonation Attacks*: The proposed protocol has the ability to prevent the impersonation attacks, which can be shown as follows. An attacker cannot impersonate an legitimate IoT device D_i, since he/she does know the shared key K_{ds} and also cannot obtain the PUF ouput $R^{'}$=$PUF_{D_i}(C)$. Without the knowledge of $R^{'}$ and K_{ds} the attacker cannot compute (k,hd) = FE.Gen($R^{'}$), V_1= $h(N_s||k||R_{new}^*||hd^*)$, and the session key sk =$h(K_{ds}||k||N_d)$ and thus cannot construct the valid response in message M_3. Similarly, an attacker cannot impersonate the server since he/she cannot obtain a valid *CRP* (C, R) and the shared key K_{ds}. Without a valid *CRP* and shared key K_{ds} the attacker cannot construct a valid response as in message M_2. Moreover, even if the attacker captures the IoT device he/she cannot obtain a valid CRP because any attempt to remove the PUF from the IoT device destroys.
6) *Protection Against Message Tampering Attacks*: The proposed protocol uses the *key-hash function* and the concept of *challenge-response* to verify the source, integrity, and freshness of the messages. The intended receiver can identify any alteration of a received message using the key-hash output. For instance, if an attacker attempts to change the contents of message M_2 of the protocol, i.e., N_s^*= $K_{ds} \oplus N_s$, the device can identify that by using the key-hash response V_0= $h(N_d||\ K_{ds}||N_s^*)$ which can not be constructed without knowledge of the secret key K_{ds}. On the other hand, if the attacker attempts to change the contents of message M_3, the server can easily comprehend that by checking the key-hash response V_1= $h(N_s||k||R_{new}^*||hd^*)$, where only the legitimate server can reconstruct the keying element k=FE.Rec($R,\ hd$).
7) *Protection Against Replay Attacks*: In the proposed scheme, an adversary cannot replay the message M_1 : $\{AID, N_d^*\}$ since AID changes in each session. The adversary cannot reuse the message M_2 since a new challenge C is used in each session. Similarly, an adversary also cannot resend the message M_3 since a new response $R_{new}^{'}$ is used in each session. In this way, we ensure the security against replay attacks.

V. Performance Analysis and Comparison

To show the advantage of our proposed scheme, now we first compare the proposed scheme with three recently proposed *user's centric* two factor authentication schemes. From Table II, we can see that, the proposed scheme is secure against all the imperative security threats and accomplishes diverse features. On the other hand, according to [24] the scheme presented in [23] cannot ensure the untracebility support and the scheme presented in [22] is vulnerable to password guessing attacks. In addition, since to ensure prevention against replay attacks the schemes presented in [22] and [24] are merely rely upon the timestamp. Hence, they are susceptible to clock synchronization problem. Nevertheless, none of these schemes ([22-24]) can guarantee the security of the user's device, where the devices are vulnerable to physical and cloning attacks. On the other hand, Table II also shows that all these user's centric authentication protocols ([22-24]) are based on the computationally expensive elliptic-curve cryptosystem (ECC). Whereas our proposed scheme is based on the computationally efficient symmetric key crypto- systems such as *PUF* and *fuzzy extractor*, etc. which are suitable to resource limited IoT devices.

Next, we compare the proposed lightweight and privacy-preserving two-factor authentication scheme with a recently proposed PUF-based mutual authentication scheme for IoT devices [14]. In [14], the IoT devices only use PUFs and do not maintain any secret key for authentication. Hence, it does not provide two-factor secrecy. Moreover, in the scheme presented in [14], the devices use their original identity during the execution of the authentication phase. Accordingly, an outside adversary can monitor the activities of the IoT devices. Therefore, Aman et al.'s scheme cannot guarantee the privacy of the IoT devices. Furthermore, even though differential design methodologies can improve reliability, noise is still an important factor in PUF design [19]. In this regard, for any given challenge, noise may result in one or several bits of the output to be incorrect. However, this important issue has been overlooked in [14].

Table II
PERFORMANCE COMPARISON WITH EXISTING USER'S CENTRIC TWO-FACTOR AUTHENTICATION PROTOCOLS

Security Property	Amin et al. [22]	Han et al. [23]	Xie et al. [24]	Proposed Scheme
Resilience to the Impersonation Attack	Yes	Yes	Yes	Yes
Anonymity and Untraceability	Yes	No	Yes	Yes
Resilience to the Password Guessing Attack	No	Yes	Yes	-
Prevents Clock Synchronization Problem	No	Yes	No	Yes
Device Security	No	No	No	Yes
Deployed Security Algorithm	ECC	ECC	ECC	PUF and FE

Table III
PERFORMANCE COMPARISON WITH AN EXISTING IOT DEVICE'S CENTRIC AUTHENTICATION PROTOCOL BASED ON SECURITY FEATURES

Comparison Matrices	Aman et al. [14]	Proposed Scheme
Mutual Authentication	Yes	Yes
Two-Factor Secrecy	No	Yes
Privacy of the IoT Devices	No	Yes
Consideration of noise in the PUF	No	Yes
Protection Against Physical Attacks	Yes	Yes

Table IV
PERFORMANCE COMPARISON WITH AN EXISTING IOT DEVICE'S CENTRIC AUTHENTICATION PROTOCOL BASED ON COMPUTATION COST

Schemes	IoT Device	Server
Aman et al. [14]	$2N_H$+ $3N_{MAC}$+ N_{SD}+$2N_{PUF}$	$2N_H$+ $3N_{MAC}$+ N_{SE}
Proposed Scheme	$5N_H$+ $N_{FE.Gen}$+$2N_{PUF}$	$5N_H$+ $N_{FE.Rec}$

Table V
EXECUTION TIME OF VARIOUS CRYPTOGRAPHIC OPERATIONS

Operations	IoT Device	Server
MAC (CBC-MAC)	2.9 ms	1.23 ms
H(SHA-256)	0.026 ms	0.011 ms
SE(AES-CBC Encryption)	-	0.14 ms
SD(AES-CBC Decryption)	0.37 ms	-
PUF (128-bit Arbiter)	0. 12 ms	-
FE.Gen (.)	2.68 ms	-
FE.Rec (.)	-	3.34 ms

On the contrary, in the proposed scheme, each IoT device maintains *two factors* (i.e., secret key K_{ds}, and its PUF PUF_{D_i}) for proving its legitimacy to the server. In addition, in the proposed scheme, the devices use their one-time alias identity or unused fake identity for each session. Therefore, it will be difficult for an outside adversary to comprehend the activities of the IoT devices. Furthermore, we address the noise issue in PUF operation in the proposed scheme by using the concept of *reverse fuzzy extractor*. From Table II and Table III, we can see that the proposed scheme can support all the desirable security properties, which are of great importance for the security of IoT devices.

Next, we consider the computation cost for comparing the proposed scheme with respect to [14]. Table IV shows the number of hash (N_H), message authentication code (MAC) (N_{MAC}), symmetric-key-based encryption/decryption ($N_{SE/SD}$), PUF (N_{PUF}), key generation algorithm FE.Gen ($N_{FE.Gen}$), and reconstruction algorithm FE.Rec ($N_{FE.Rec}$) operations required by the proposed mutual authentication protocol and the protocol proposed by Aman et al. [14]. Now, for rigorously analyzing the performance of the proposed protocol with respect to [14], we conducted simulations of the cryptographic operations used in the proposed scheme and [14] on an Ubuntu 12.04 virtual machine with an Intel Core i5-4300 dual-core 2.60 GHZ CPU (operating as a server). To simulate an IoT device, we used a single core 798 MHz CPU with 256 MB of RAM. Our simulations used the JCE library [20] to evaluate the execution time of the cryptographic primitives (shown in Table IV) used in the proposed scheme and [14]. For these results, we considered the 128-bit arbiter PUF for PUF operation and for FE.Gen and FE.Rec operations, we adopted the code offset mechanisms using BCH [21].

Based on the simulation results of Table IV, we see that in [14] an IoT device takes 9.36 ms to compute $2N_H+3N_{MAC}+N_{SD}+2N_{PUF}$ operations and the server takes 3.85 ms for executing $2N_H+3N_{MAC}+N_{SE}$ operations. Therefore, the overall computational cost of the scheme presented in [14] is 13.21 ms. On the other hand, in the proposed scheme the computation cost at the IoT devices is 2.92 ms for executing $5N_H+N_{FE.Gen}+2N_{PUF}$ operations and server takes 3.39 ms to compute $5N_H+N_{FE.Rec}$ operations. Therefore, the overall computational cost of the proposed scheme is 6.31 ms, which is significantly lower than [14]. Hence, it can be argued that the proposed scheme is secure and more efficient for resource limited IoT devices.

VI. Formal Analysis of the Proposed Scheme Using BAN Logic

In this section, we present a formal analysis of the proposed scheme using the Burrows-Abadi-Needham logic [25], generally known as BAN logic. The BAN logic model provides primitives that describes the beliefs of the principles involved in a crypto system.

A. BAN logic and its Enhancement

The BAN logic is based on the a set of postulates and assumptions and it uses three objects: principals, encryption keys, and logic formulas. The main construction of BAN logic is described as follows.

- $P \models X$ represents P believes X.
- $P \lhd X$ represents P sees X.
- $P \mid\sim X$ represents P said X.
- $P \Rightarrow X$ represents P has jurisdiction over X.
- $\#(X)$ represents that the formula X is fresh and X has not been sent in a message at any time before the current execution of the proposed scheme.
- $P \overset{K}{\leftrightarrow} Q$ represents P and Q share a secret K.
- $P \ni X$ represents P is capable of processing formula X.
- $\{X\}_K$ represents that formula X is encoded/encrypted using key K.

The set of inference rules of BAN logic that are required in the analysis of our proposed scheme are described below.

1) Message-meaning rule R1: $\frac{P \models P \overset{K}{\leftrightarrow} Q,\ P \lhd \{X\}_K}{P \models Q \mid\sim X}$;
2) Nonce-verification rule R2: $\frac{P \models \#(X),\ P \models Q \mid\sim X}{P \models Q \models X}$;
3) Jurisdiction rule R3: $\frac{P \models Q \Rightarrow X,\ P \models Q \models X}{P \models X}$;
4) Seeing rules R4: $\frac{P \lhd (X,Y)}{P \lhd X}$; R5: $\frac{P \models P \overset{K}{\leftrightarrow} Q,\ P \lhd \{X\}_K}{P \lhd X}$;
5) Fresh rule R6: $\frac{P \models \#(X)}{P \models (X,Y)}$;
6) Belief rule R7: $\frac{P \models (X,Y)}{P \models X}$;

To analyze the properties of the proposed scheme, we need to extend the conventional BAN logic with the following rules: ER1: $\frac{P \models Q \overset{K}{\leftrightarrow} P,\ P \lhd f(X,Y)}{P \models Q \mid\sim Y}$ and ER2: $\frac{P \models \underset{\rightarrow}{X} Q,\ P \lhd X}{P \models Q \mid\sim X}$. Extension rule ER1 specifies that if key K is shared between P and Q, and function f that compares formulas X and Y is satisfied, then it also verifies the originality of principle Y. Extension rule ER2 denotes that the verification root of X.

B. Analysis of the Proposed Scheme

The initial security assumptions on IoT device D_i, and the server S are described as follows:

1. $\mathrm{D}_i \models \mathrm{D}_i \overset{K_{ds}}{\leftrightarrow} \mathrm{S}$; and $\mathrm{D}_i \models \mathrm{D}_i \overset{R}{\leftrightarrow} \mathrm{S}$
2. $\mathrm{S} \models \mathrm{D}_i \overset{K_{ds}}{\leftrightarrow} \mathrm{S}$; and $\mathrm{S} \models \mathrm{D}_i \overset{R}{\leftrightarrow} \mathrm{S}$;
3. $\mathrm{D}_i \models \mathrm{D}_i \overset{AID}{\leftrightarrow} \mathrm{S}$, where $\mathrm{S} \ni AID$;
4. $\mathrm{S} \models \mathrm{D}_i \overset{AID}{\leftrightarrow} \mathrm{S}$, where $\mathrm{S} \Rightarrow AID$ as $\mathrm{S} \ni AID$;

Now, we first show response message M_2 received by the IoT device D_i in the proposed scheme is valid. For each device D_i, we can write $\mathrm{D}_i \models \mathrm{S} \mid\sim \mathrm{M}_2$, $\exists \mathrm{D}_i \models \#(\mathrm{M}_2)$. Furthermore, when D_i receives M_2, we can use belief rules R7 and ER1 to derive the following statements for authentication:

$$\frac{\mathrm{D}_i \models (V_0, K_{ds})}{\mathrm{D}_i \models V_0};$$

$$\frac{\mathrm{D}_i \models (V_0, N_d)}{\mathrm{D}_i \models V_0},$$

$$\frac{\mathrm{D}_i \models \{\mathrm{M}_2, (N_d, K_{ds})\}}{\mathrm{D}_i \models \mathrm{M}_2},$$

$$\frac{\mathrm{D}_i \models \mathrm{S} \overset{K_{ds}}{\leftrightarrow} \mathrm{D}_i,\ \mathrm{D}_i \lhd f((h(\mathrm{N}_d || K_{ds} || N_s^*),\ V_0)}{\mathrm{D}_i \models \mathrm{S} \mid\sim V_0};$$

Now, we show that $\mathrm{D}_i \models \mathrm{S} \mid\sim \mathrm{M}_2$ and for that we use ER2 to derive the following statement:

$$\frac{\mathrm{D}_i \models \underset{\rightarrow}{M_2} \mathrm{S},\ \mathrm{D}_i \lhd \mathrm{M}_2}{\mathrm{D}_i \models \mathrm{S} \mid\sim \mathrm{M}_2};$$

$$\frac{\mathrm{D}_i \models \mathrm{S} \Rightarrow V_0,\ \mathrm{D}_i \models \mathrm{S} \models V_0}{\mathrm{D}_i \models V_0};$$

Similarly, when the server S receives the message M_3 from the IoT device D_i, for the validation of the message M_3 and authentication of D_i we utilize R6, R7, ER1 and ER2 to derive the following statements:

$$\frac{\mathrm{S} \models \{(\mathrm{M}_3),\ V_1\}}{\mathrm{S} \models (\mathrm{M}_3)};$$

$$\frac{\mathrm{S} \models \{(V_1), K_{ds}\}}{\mathrm{S} \models (V_1)};$$

$$\frac{\mathrm{S} \models \{(V_1), N_s\}}{\mathrm{S} \models (V_1)};$$

$$\frac{\mathrm{S} \models \{(k), (N_s, R)\}}{\mathrm{S} \models (k)};$$

$$\frac{\mathrm{S} \models \underset{\rightarrow}{\mathrm{M}_3} \mathrm{D}_i,\ \mathrm{S} \lhd \mathrm{M}_3}{\mathrm{S} \models \mathrm{D}_i \mid\sim \mathrm{M}_3};$$

$$\frac{\mathrm{S} \models \{(\mathrm{M}_3), \mathrm{V}_1\}}{\mathrm{S} \models (\mathrm{M}_3)};$$

$$\frac{\mathrm{S} \models \mathrm{D}_i \overset{k}{\leftrightarrow} \mathrm{S}, \mathrm{S} \lhd f((h(N_s || k || R_{new}^* || hd^*),\ V_1)}{\mathrm{S} \models \mathrm{D}_i \mid\sim V_1};$$

$$\frac{\mathrm{S} \models \#(N_s)}{\mathrm{S} \models (N_s, \mathrm{M}_3)};$$

Now, we consider the *session key security* of the proposed scheme. We utilize R6, R7, ER1 and ER2 to derive the following statements for validating the session key:

$$\frac{\mathrm{S} \models (sk,\ V_1)}{\mathrm{S} \models sk};$$

$$\frac{\mathrm{D}_i \models \#(hd^*)}{\mathrm{D}_i \models (hd^*,\ V_1)};$$

$$\frac{\mathrm{S} \models \#(K_{ds})}{\mathrm{S} \models (K_{ds},\ V_1)};$$

Similarly for the IoT device we can to derive the following statements:

$$\frac{\mathrm{D}_i \models \{\mathrm{V}_0, \mathrm{sk}\}}{\mathrm{D}_i \models sk};$$

$$\frac{\mathrm{D}_i \models \#(N_d)}{\mathrm{SM}_i \models (sk, N_d)};$$

Next, we note that $\mathrm{S} \models \mathrm{D}_i \mid\sim \{AID\}$ since $\mathrm{S} \models\!\Rightarrow AID$ and $\mathrm{S} \mid\sim AID$. Therefore, using ER1 and R3 we can show that , the proposed scheme achieves identity authentication with the following statements:

$$\frac{\mathrm{S} \models \overrightarrow{AID}\, \mathrm{D}_i,\ \mathrm{S} \lhd AID}{\mathrm{S} \models \mathrm{D}_i \mid\sim AID};$$

$$\frac{\mathrm{S} \models \mathrm{D}_i \models\!\Rightarrow AID,\ \mathrm{S} \models \mathrm{D}_i \models tid_{ij}}{\mathrm{S} \models AID};$$

This proves the correctness of the proposed two-factor authentication scheme.

VII. Conclusions

In this paper we presented a novel privacy-preserving two-factor authentication protocol for IoT devices, which allows an IoT device to anonymously communicate with the server located at the data and control unit. We showed that the proposed scheme remains secure even if an adversary has physical access to an IoT device. The proposed protocol provides the desired security characteristics efficiently by exploiting the inherent security features of PUFs. Hence, we argue that the proposed scheme is be a viable and promising solution for the security of IoT devices.

Acknowledgment

This research was supported by the Ministry of Education, Singapore under a Tier 1 grant (number R-263-000-C13-112).

References

[1] P. S. Ravikanth, Physical One-Way Functions, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[2] G. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Design Automation Conference, *DAC '07, 44th ACM/IEEE*, 2007, pp. 9–14.

[3] V. Shivraj, M. Rajan, M. Singh and P. Balamuralidhar, "One time password authentication scheme based on elliptic curves for Internet of Things (IoT)," *Proceedings of NSITNSW*, pp. 1-6, Riyadh, KSA, February 2015.

[4] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications," *Proceedings of IEEE WCNC*, pp. 2728-2733, Istanbul, Turkey, April 2014.

[5] Y. Kim, S. Yoo, and C. Yoo, "DAoT: Dynamic and Energy-aware Authentication for Smart Home Appliances in Internet of Things," *Proceedings of IEEE ICCE*, pp.196-197, Las Vegas, NV, Jan 2015.

[6] V. Petrov, S. Edelev, M. Komar, and Y. Koucheryavy, "Towards the Era of Wireless Keys: How the IoT Can Change Authentication Paradigm," *Proceedings of IEEE WF-IoT*, pp.51-56, Seoul, South Korea, March 2014.

[7] E. Ozturk, G. Hammouri, and B. Sunar, "Towards Robust low cost authentication for pervasive devices", *Proceeding of IEEE PerCom*, pp. 170-178, 2008.

[8] K. Frikken, M. Blanton and M. Atallah, "Robust Authentication Using Physically Unclonable Functions", In: P. Samarati et al. (eds.): *ISC* 2009, LNCS 5735, pp. 262-277, Springer, Heidelberg 2009.

[9] E. Simpson, P. Schaumont, "Offline hardware/software authentication for reconfigurable platforms", In: L. Goubin, M. Matsui, (eds.) *CHES* 2006, LNCS, vol. 4249, pp. 311-323, Springer, Heidelberg 2006.

[10] J. Guajardo et al. "Physically Unclonable functions and public key crypto for FPGA IP protection," *International Conference on Field Programmable Logic and Applications*, pp. 189-195, 2007.

[11] A.-R. Sadeghi, I. Visconti, C. Wachsmann, "PUF-enhanced RFID security and privacy," *in: Secure Component and System Identification–SECSI*'10, Cologne, Germany, 2010.

[12] M. Akgun, M.U. Caglayan, "Puf based scalable private RFID authentication," *in: Proceedings of the 20 11 Sixth International Conference on Availability, Reliability and Security, ARES* '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 473–478.

[13] S. Kardas , S. Elik, M. Yıldız, A. Levi, "Puf-enhanced offline RFID security and privacy," *J. Netw. Comput. Appl*. 35 (6) (2012) 2059–2067.

[14] M. N. Aman, et al. "Mutual Authentication in IoT Systems Using Physical Unclonable Functions," *IEEE Internet of Things Journal*, vol. 4(5), pp. 1327-1340, 2017.

[15] Y. Dodis, J. Katz, L. Reyzin, A. Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," *In: Advances in Cryptology (CRYPTO)*, LNCS, vol. 4117, pp. 232-250. Springer (2006)

[16] Y. Dodis, L. Reyzin, A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *In: Advances in Cryptology (EUROCRYPT)*. LNCS, vol. 3027, pp. 523–540 (2004)

[17] C. Bosch, J. Guajardo, A.R. Sadeghi, J. Shokrollahi, P. Tuyls, " Efficient helper data key extractor on FPGAs," *In: Cryptographic Hardware and Embedded Systems (CHES)*. LNCS, vol. 5154, pp. 181–197. Springer (2008)

[18] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M-D Yu, "Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications, " *In: Cryptographic Hardware and Embedded Systems (CHES*). LNCS vol. 8913 pp. 412-430, Springer (2016).

[19] C. Herder, M. D. Yu, F. Koushanfar and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial,"*In Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126-1141, Aug. 2014.

[20] Oracle Technology Network. Java Cryptography Architecture (JCA). [Online]. Available: http://docs.oracle.com/javase/6/docs/technotes/guides/crypto/CrypoSpec.html, accessed Apr. 20, 2017.

[21] Y. Dodis et al., "Fuzzy extractors: How to generate strong keys from from biometrics and other noise data," *SIAM J. Compt*. vol. 38, no. 1, pp. 97-139, 2008.

[22] R. Amin, S. Islam, M. K. Khan, A. Karati, D. Giri, and S. Kumari, "A two-factor rsa-based robust authentication system for multiserver environments,"*Security and Communication Networks*, vol. 2017, 2017.

[23] L. Han et al. "An efficient and secure two-factor authentication scheme using elliptic curve cryptosystems,"*Peer-to-Peer Networking and Application*, vol. 11(12), pp. 1âĂŞ11, 2016.

[24] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382âĂŞ1392, 2017.

[25] Michael Burrows, Martin Abadi, Roger Needham. "A Logic of Authentication,"*DEC SRC Research Report 39.*

Prosanta Gope received the M.Tech. degree in computer science and engineering from the National Institute of Technology (NIT), Durgapur, India, in 2009, and the PhD degree in computer science and information engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2015. He is currently working as a Research Fellow in the department of computer science at National University of Singapore (NUS). Prior to this, Dr. Gope served over one year as a Postdoctoral Research Fellow at Singapore University of Technology and Design (SUTD) established in collaboration with Massachusetts Institute of Technology (MIT). His research interests include lightweight authentication, authenticated encryption, access control system, and security in mobile communication and hardware security of the IoT devices. He has authored over 50 peer-reviewed articles in several reputable international journals and conferences, and has three filed patents. He received the Distinguished Ph.D. Scholar Award in 2014 given by National Cheng Kung University, Tainan, Taiwan.

Biplab Sikdar (S'98-M'02-SM'09) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He was on the faculty of Rensselaer Polytechnic Institute from 2001 to 2013, first as an Assistant and then as an Associate Professor. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include computer networks, and security for IoT and cyber physical systems. Dr. Sikdar is a member of Eta Kappa Nu and Tau Beta Pi. He served as an Associate Editor for the IEEE Transactions on Communications from 2007 to 2012. He currently serves as an Associate Editor for the IEEE Transactions on Mobile Computing.