



Lightweight multi-factor mutual authentication protocol for IoT devices

Reem Melki¹ · Hassan N. Noura^{1,2} · Ali Chehab¹

Published online: 13 December 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

The Internet-of-Things (IoT), which refers to the interconnection of heterogeneous devices, has gained a lot of interest lately, and it witnessed a large growth in the number of IoT devices due to the importance of such systems in today's communication networks. On the other hand, the authentication of entities (devices) is a major concern and a main security challenge in IoT systems since any weakness in the identification or authentication process will allow a compromised entity to establish communication, inject false data and launch dangerous attacks leading to system malfunction. Currently, most IoT authentication mechanisms are based on single-factor cryptographic solutions. These techniques are not practical for IoT devices that have limited computational capabilities. In this paper, we propose a lightweight and secure multi-factor device authentication protocol for IoT devices. The scheme is based on two concepts, configurable physical unclonable functions (PUF) within IoT devices, and channel-based parameters. It uses few and simple cryptographic operations such as the bit-wise exclusive-OR operation and a one-way hash function. The unique PUF value serves as the mutual secret identifier between a pair of users, which frequently changes for every session. Moreover, the proposed protocol exploits the random channel characteristics to provide high robustness against different kinds of attacks, while maintaining low complexity. To the best of the authors' knowledge, this is the first work that combines physical layer security with PUFs to authenticate communicating devices, dynamically. Security and performance analysis prove the security and efficiency of the proposed protocol, which is designed with minimum overhead in terms of computations and communication costs.

Keywords Lightweight mutual authentication · PUF · Dynamic keys · AVISPA · Physical layer security

1 Introduction

Recently, IoT technology has been introduced as a new paradigm that enables different physical devices that feature an IP (Internet Protocol) address, to communicate and interact with each other via the Internet. A large class of these devices uses wireless channels to connect to the network and relay data. However, the broadcast nature of wireless

channels makes it easier for adversaries to eavesdrop and conduct different attacks, which exposes the IoT system to a wide range of threats (vulnerabilities) and compromises the communication among IoT devices. Hence, ensuring robust security is of utmost importance for realizing and deploying IoT systems [1]. Some of the security requirements for IoT systems include secure booting, authentication, access control, data integrity and privacy [2]. Authentication in IoT systems and the challenges associated to this security service have been thoroughly discussed in the literature, and the importance of finding appropriate solutions to address these concerns has been outlined in [2–9]. In principle, communicating entities should be able to authenticate each other and verify that the received data are indeed legitimate. Therefore, mutual authentication (device authentication) is the first step toward establishing secure communication among different IoT devices.

The task of designing authentication protocols for IoT devices is extremely difficult and challenging since these

✉ Reem Melki
rmm71@aub.edu.lb

Hassan N. Noura
hn49@aub.edu.lb

Ali Chehab
chehab@aub.edu.lb

¹ Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon

² Department of Computer Sciences, Arab Open University, Beirut, Lebanon

devices are usually low cost and simple in nature with limited processing, memory and energy resources. Consequently, any security protocol or application designed to run on IoT devices should be very efficient in terms of computational complexity and energy requirements [6,10]. Moreover, most schemes presented in the literature, which use only one-factor authentication fail to prevent device authentication attacks (such as device forgery or impersonation), since this single factor is vulnerable to being captured and acquired by adversaries [11].

1.1 Contributions

The main goal of this work is to secure end-to-end communication systems and prevent existing authentication attacks. Hence, in this paper, we present a lightweight and secure mutual authentication protocol for IoT systems. The contributions of this paper are summarized as follows:

- *A multi-factor authentication protocol* The proposed scheme is based on multiple factors to increase the authentication accuracy. To the best of our knowledge, this is the first work that combines PUF-derived secret keys and dynamic physical channel parameters, to design an efficient and secure mutual authentication scheme. The physical layer parameters are introduced in the proposed protocol to increase the accuracy, randomness and dynamicity. More specifically, communicating entities (IoT devices and servers) rely on a PUF-derived secret, and channel randomness [Physical layer security (PLS)] to authenticate each other and establish a secret session key, which will later be used to ensure data confidentiality (encryption), data integrity and message authentication (message authentication algorithm).
- *Dynamic key derivation function* For each new session, new user-credentials are used to minimize the risk of having an exposed key and to prevent tracking; the PUF-derived secret and the channel-based nonce are both updated (dynamicity) to obtain a new session key. Then, this session key is used to produce a set of confidentiality and authentication dynamic keys. This increases the security level since each input message is encrypted and authenticated using a different key. The dynamic key approach is presented to complement the proposed authentication protocol and to enhance its security level.
- *Security and efficiency* Due to the usage of lightweight cryptographic computations, mainly the XOR operation and a one-way hash function, the proposed technique is very efficient in terms of computations, communication overhead, and required resources, compared to other authentication schemes in the literature; the technique takes into account energy constraints, processing capabilities, and practical limitations of low-power IoT devices.

Moreover, several simulation tests have been performed to prove that the proposed protocol achieves the desired security and performance (efficiency) requirements, in comparison with several PUF-based authentication protocols present in the literature.

1.2 Organization

The rest of the paper is organized as follows. Section 2 briefly reviews the related work in the literature. Section 3 presents the network and threat models. Section 4 describes the proposed authentication protocol. Section 5 analyzes the security of the proposed mechanism in the context of different authentication attacks. Section 6 evaluates the security of proposed authentication technique using the AVISPA tool. Section 7 analyzes the security of the dynamic key generation scheme. Section 8 discusses the PUF-based threats. Section 9 assesses the performance and efficiency of the proposed protocol in comparison with existing schemes. Finally, Sect. 10 concludes the paper.

2 Related work

Recently, several authentication protocols based on PLS have emerged in the literature. These protocols are either key-less schemes or key-based schemes. While the former ones depend on specific features related to the shared channel between a pair of devices, the latter ones use a key that is only known to the communicating entities in the authentication procedure, which makes it more reliable and secure [19].

In the literature, there are four main approaches to device authentication: (1) comparing the channel properties of two consecutive frames [Channel state information (CSI), carrier frequency offset (CFO)], (2) relying on a third-party authority (relay) and performing XOR and simple multiplication operations, (3) using secret keys derived from the channel and (4) utilizing the concept of “tags” which can be generated using encryption or hashing [13–23]. However, these schemes suffer from several limitations, which hinders their efficient and secure deployment in current wireless systems. More specifically, the first approach is considered ineffective and insecure since users, within a small geographical area, experience similar channel conditions (CSI, CFO) and hence adversaries are able to deceive legitimate receivers into thinking that the sent frames are transmitted by a legitimate sender. On the other hand, the second approach requires a high level of trust between users and the third party, which is not always applicable since they are vulnerable to impersonation. In the third scheme, the secret key, which is derived from the channel, is a weak proposition since it can be generated and acquired by adversaries if present within the same geographical area. Finally, using “tags” in the authentica-

Table 1 A summary of the PLS device authentication schemes

Device authentication schemes	Comparison of the channel properties of two consecutive frames (CSI, CFO) [12, 13]	Third-party authority (relay) using XOR and simple multiplication operations [14]	Using secret keys derived from the channel for encryption [15, 16]	Concept of “tags” which can be generated using encryption or hashing [17, 18]
Advantage	No additional cost and overhead	Non-repudiation	Utilizing the notion of a secret key	Utilizing the notion of a secret key
Limitation	Ineffective when attacker is near legitimate user, that is when both experience same channel conditions	The third-party authority is vulnerable to being impersonated. Moreover, a high level of trust should exist between legitimate users and the third party	A secret key derived from the channel is a weak proposition since it can be easily generated and acquired	Computationally complex. It requires additional operations at the transmitter and receiver
Resource and communication cost	No additional cost and overhead	Authentication is verified through multiple rounds of communications (3 rounds) and performing simple operations such as XORing	Two steps are done prior to data transmission: (1) Key extraction. (2) Encryption and decryption operations are performed	(1) Key extraction and (2) “tag” generation. (3) Append “tags” to the transmitted messages
Complexity	Not computationally complex: performing comparison	Not computationally complex: performing XOR operations	Computationally complex: encryption	Computationally complex: encryption

tion process is computationally complex where additional operations at the transmitter and receiver are required [24]. Table 1 summarizes and compares the PLS device authentication schemes present in the literature.

On another note, the authors in [25] proposed an RFID-based authentication architecture for distributed IoT applications, suitable for the future smart city environments. The proposed protocol (independent of PLS) is lightweight and efficient, compared to existing schemes. It also provides forward secrecy, anonymity and untraceability of RFID tags and secure localization. Similarly, a lightweight authentication protocol for IoT-enabled devices in distributed cloud computing environments is presented in [26]. The authors propose an architecture with an authentication protocol based on smart-cards, in which registered users are able to access all private information securely, from the private cloud servers. Also, the authors in [27] discuss the three-layered data flow architecture for fog computing and present several novel architectures such as energy lattices, MediFog, UXFog, connected parking system and FoAgro within the paradigm of fog computing.

In contrast, authors in [28] present a secure PUF-based device authentication protocol for wearable devices, **independent of PLS**. The presented scheme allows wearable devices and mobile terminals, worn or carried by the same user, to mutually authenticate each other and share a secret session key, which will later be used to secure communication. Lightweight cryptographic computations, mainly the hash function and XOR operation, are utilized toward achieving high security and low complexity, simultaneously. However, this scheme is considered inefficient since it requires a large number of computational operations and a large execution time. In fact, 17 hash functions are needed to ensure secure authentication between the two devices, which is quite exhaustive for resource- and power-limited devices. A different PUF-based approach is applied in [29], where a three-factor anonymity authentication scheme is presented for wireless sensor networks (WSNs) in Internet-of-Things (IoT) systems. This scheme mainly depends on two simple operations which are multiplication and hashing; however, it suffers from high computation costs (21 hash functions). Similar, but less efficient user authentication, schemes are also presented in [30,31]. The presented protocols require the exchange of four messages and a total of 31 and 19 hash functions, respectively. All of the aforementioned PUF-based authentication schemes have been proven to be secure in the literature; however, these schemes suffer from high computational complexity and communication costs, as it will be shown in Sect. 9.

In order to secure the mutual authentication process between IoT users and gateways over wireless links, we propose a lightweight protocol based on simple cryptographic operations, random channel parameters (PLS) and a secret

session identifier. The proposed protocol outperforms the protocols presented in [28–31] in terms of communication and computation costs.

3 System models

In this section, we describe the proposed network model, the threat model, the utilized fuzzy system and the basic properties of PUFs.

3.1 Table of notations

Table 2 represents all of the notations used in this paper.

3.2 Network model

Figure 1 represents the IoT communication system, where different IoT devices (such as smartphones) are able to com-

Table 2 Table of notations

Notation	Definition
ID_S	Secret session identifier
PUF	Physical unclonable function
SK	Shared secret session key
σ_i	Secret channel-based parameter which is obtained from the $Gen(\cdot)$ function
$N_{0,A}$	Channel-based nonce of User A
$N_{0,B}$	Channel-based nonce of User B
$Gen(\cdot)$	A probabilistic function that generates a uniform string of bits given a specific input
$Rep(\cdot)$	A function that recovers a string of bits from an input slightly different from the original input
TS	Time stamp; time of message transmission
R	Random number
τ_i	The second output of the $Gen(\cdot)$ function
HD	Hamming distance
TS*	Time of the received message
ΔT	Transmission delay
DK	Dynamic key
K_C	Data confidentiality sub-key
IV_C	Data confidentiality initial vector
K_A	Authentication sub-key
IV_A	Authentication initial vector
P	Plaintext
EP	Encrypted plaintext
MAC	Message authentication code
$M = \langle X, Y \rangle$	Message containing elements X and Y
$h(\cdot)$	Hash function
\parallel	Concatenation operation
\oplus	Exclusive-OR (XOR)

municate directly with each other [32]. The system consists of several mobile devices and a fixed number of communication units (gateways). Here, it should be noted that the presence of aggregation nodes is not mandatory since in most cases, IoT devices are able to communicate with the gateway directly (for example: the Long Range radio (LoRa) technology [33]).

In this paper, we assume that:

- IoT devices communicate with the gateway directly over public wireless links (star topology).
- Throughout this paper, we assume that the gateway resides in the network server. If not, users can perform authentication with the network server, using the proposed protocol.
- The proposed solution targets single-hop networks (no aggregation nodes).
- Channel between two users is non-reciprocal.
- The gateway has high computational power and large memory. This unit is responsible for storing different input values (to the PUF), each corresponding to a different IoT user in the cell.
- Secret session identifiers (ID_S) are derived from the output of the PUF.
- The gateway stores the initial challenge/response of each device. For each authentication session, a fresh challenge/response is produced and updated in a secure and synchronized manner.
- Secret session identifiers ID_S are kept secret by the corresponding parties.
- The initial PUF inputs are embedded within each IoT device and are saved in the memory of the gateway/network server (initial configuration). Otherwise, each IoT device relays its initial challenge (input of PUF) to the gateway/network server, using a secure exchange scheme. Consequently, the gateway will be able to obtain the ID_S values of all IoT devices, including new devices joining the network.

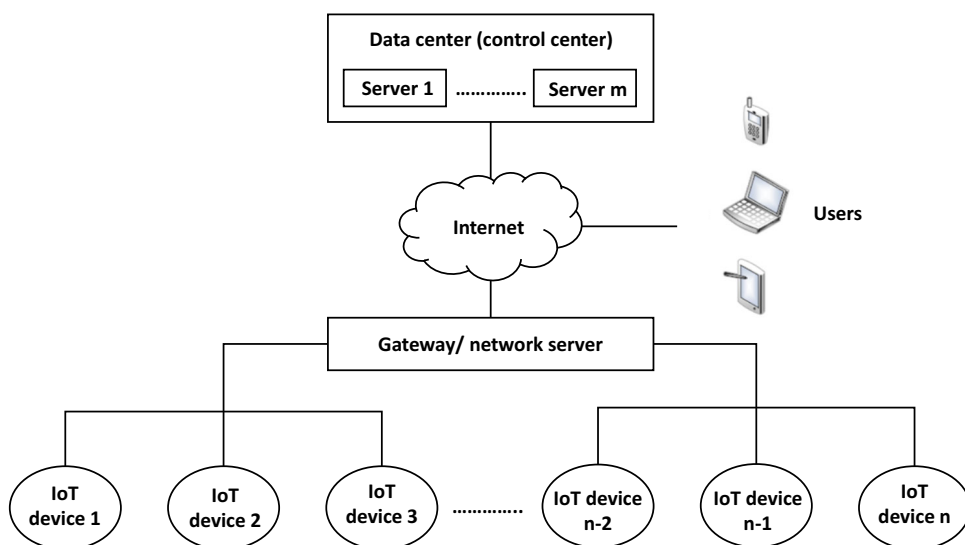
The main advantage of introducing PUFs is to eliminate the need of secret symmetric keys.

It should be noted that the procedure for storing the different challenge/response values corresponding to each IoT user at the gateway is not the main focus of this paper; hence, it will not be discussed further.

3.3 Threat model

In the proposed scheme, we consider that end-points are untrustworthy nodes and that the adversary is able to read, forge, manipulate, reply, delay and delete messages. Moreover, end-points communicate with the gateway (gateway

Fig. 1 IoT system model, having n IoT devices and m servers (example: LoRaWAN System)



residing in the network server) directly, as is the case in the LoRaWAN system [33].

In addition, we assume that the only way to compromise the authentication session is by obtaining the long-term and short-term secrets. However, this is not possible since the attacker will have to guess the values of these secrets. Based on the work presented in [34], we assume that the underlying cryptography is perfect: each cryptographic primitive is modeled as an abstract symbolic function with strong properties [35]. For example, hash functions are irreversible (one-way) [35].

3.4 Fuzzy system

In the proposed protocol, we exploit the advantages of fuzzy systems to overcome channel non-reciprocity. Generally, a fuzzy system, which is based on a collision-resistant extractor, takes as input a binary string, N , of some metric space $SP \in \{0, 1\}^n$ (n is a positive number) and outputs a random string $\sigma \in \{0, 1\}^l$ (l is a positive number) and an auxiliary string, $\tau \in \{0, 1\}^r$, where r is a positive number that can be equal to l or n [36]. This mapping procedure is denoted by:

$$Gen(N) = (\tau, \sigma). \tag{1}$$

Typically, τ is a public reproduction parameter, that is known by all users. However, in the proposed scheme, this is not the case since the input to the $Gen(.)$ function is a parameter only known by communicating entities. Therefore, the resulting outputs, τ and σ , will not be public parameters.

Another procedure that is also used in fuzzy systems is the recovery function, in which a different string, N' , of the same metric space $SP \in \{0, 1\}^n$ is fed, along with $\tau \in \{0, 1\}^r$, to produce $\sigma \in \{0, 1\}^l$ [36]. This mapping procedure is denoted

by:

$$Rep(N', \tau) = \sigma. \tag{2}$$

Both of these functions are used in the proposed mutual authentication process, which combines the secrecy of the PUF output values, and the randomness and dynamicity of wireless channels (PLS).

3.5 PUFs: basic properties and characteristics

A PUF circuit has a specific architecture, which is typically added to a chip to extract its unique fingerprint. The input to a PUF circuit is a sequence of bits, referred to as challenges, and the output is another sequence of bits, referred to as responses. Each chip (IoT device) has its own fingerprint related to the unique pairs of challenges/responses; that is no two chips (IoT devices) are able to produce the same response for the same challenge [20,37], which is mainly due to the variability within the manufacturing process.

In general, PUFs are widely used for the authentication of resource-limited devices since no cryptographic operations are required in this case. The use of PUF circuits is a very popular technique for authenticating IoT devices. Specifically, PUF-based authentication is divided into two phases: enrollment and authentication [20,37].

In the first phase, the chip, which contains the PUF circuit, is physically linked to the server (connected). The server generates challenges, and the PUF circuit returns back the corresponding responses which are stored in the server. Next, the chip is attached to the IoT device [20,37].

During the second phase, the server sends a dynamic random PUF challenge to the device. If the device produces and transmits the correct corresponding response, the device is authenticated [20,37].

The enrollment and authentication steps are slightly modified in the proposed protocol. Here, we assume that the gateway (or network server in case the gateway does not reside in the network server) already has the PUF inputs (challenges) of each IoT device in the network.

During the enrollment phase, the chip of each IoT device is connected to the gateway (in case it resides in the network server). The gateway generates and stores the device's secret session identifier, ID_S , based on the responses to the challenges of the IoT device. This identifier is then used during the authentication process.

4 Proposed protocol

The proposed protocol is based on two main authentication factors, the secret session identifier (ID_S), and the secret channel-based parameter, σ_i . The secret session identifier ID_S is derived from a PUF output value that is only known by the communicating entities (IoT user and gateway). More specifically, the gateway keeps a list of input PUF values (initial challenge/response), each corresponding to a different IoT user. This list is private and is only accessible by the gateway. The parameter, ID_S , serves as the common shared secret. This allows the gateway to distinguish the different IoT devices.

The proposed scheme depends only on two lightweight operations, a cryptographic one-way hash function ($h(\cdot)$) and the XOR operation. Moreover, the widely used fuzzy extractor technique is used to overcome the issue of channel non-reciprocity. In general, most PLS techniques in the literature assume that the channel between a pair of users is reciprocal, consequently, both communicating entities extract the same channel parameters and use them for data encryption (data confidentiality) [38]. However, this is not always true. In fact, channel characteristics and features of the same channel may differ slightly between the transmitter and receiver, and they may change from time to time. As result, a channel-based nonce extracted by User A is not always equal to a channel-based nonce extracted by User B ($N_{0,A} \neq N_{0,B}$). Common sources of channel-based nonces are the channel state information (CSI), the received signal strength (RSS) and angle of arrival (AoA) [39]. For this purpose, the fuzzy extractor, which depends on two functions, $Gen(\cdot)$ and $Rep(\cdot)$, is used in this scheme.

- $Gen(\cdot)$: This function is a probabilistic function that generates a uniform string of random bits given a specific input. In the proposed scheme, the input is the channel nonce extracted by User A (transmitter of authentication request) and is represented by $N_{0,A}$. The produced outputs are the l -bit channel-based key σ_i and the repro-

duction parameter τ_i . Hence, we have $Gen(N_{0,A}) = (\sigma_i, \tau_i)$.

- $Rep(\cdot)$: This function recovers the uniform string of random bits from an input that is slightly different from the original input (Hamming distance less or equal to a predefined threshold value t): $HD(N_{0,B}, N_{0,A}) \leq t$. In particular, $N_{0,B}$ and τ_i are given as inputs to produce the channel-based key σ_i . Hence, we have $\sigma_i = Rep(N_{0,B}, \tau_i)$.

Cryptographic hash functions are employed in the proposed protocol to ensure the one-way property of exchanged messages, in addition to a high input sensitivity. This prevents attackers from recovering any secret information (irreversibility) from the collected traffic. Only legitimate entities, sharing similar features and unique parameters (secret), are able to calculate the same hash digest. Hence, eavesdroppers will not be able to acquire any useful information from the transmitted messages, unless they have all of the correct parameters, which is very unlikely. This step is crucial for ensuring proper and secure authentication.

Finally, in order to guard against replay attacks, both a time stamp (TS) and a random number (R) are used.

4.1 Authentication and key agreement phase

This phase is executed by the IoT device (User A) and gateway (User B).

1. *Step 1* User A first extracts a channel nonce, $N_{0,A}$, and generates (σ_i, τ_i) from $N_{0,A}$ using $Gen(N_{0,A}) = (\sigma_i, \tau_i)$. User A also generates a random number R_A . The stored secret session identifier ID_S derived from the PUF is concatenated with the current time stamp TS_A . The resultant is hashed, then XORed with R_A to generate $M_1 = h(ID_S || TS_A) \oplus R_A$. A second message M_2 is calculated by XORing the random number R_A and τ_i , $M_2 = R_A \oplus \tau_i$. Finally, User A transmits the authentication request composed of $\langle M_1, M_2, TS_A \rangle$, to User B over a public channel. Here, it should be noted that the reproduction parameter, τ_i , is used to protect and securely transmit R_A to User B . In this protocol, τ_i is derived from the shared physical channel between legitimate users, hence, this parameter is not publicly available and cannot be acquired by adversaries. On the other hand, σ_i is kept as a secret by the transmitter.
2. *Step 2* Once the authentication request is received, User B validates the currency of TS_A using $|TS_A - TS_A^*| \leq \Delta T$, where ΔT is the maximum transmission delay, and TS_A^* is the received time of the message. If this condition fails, User B terminates the connection.
3. *Step 3* User B calculates $M_3 = h(ID_S || TS_A)$ using the stored information ID_S and the received informa-

tion TS_A . Next, User B calculates $R_A = M_1 \oplus M_3$ and $\tau_i = M_2 \oplus R_A$. Using τ_i , $\sigma'_i = Rep(N_{0,B}, \tau_i)$ is generated.

4. *Step 4* Afterward, User B generates $M_4 = h(ID_S || TS_A || TS_B || R_A) \oplus R_B$ using a random number R_B and the current time stamp TS_B . In addition, a secret session key SK is derived using the stored and received information such that $SK = h(ID_S || \sigma'_i || R_A || R_B || TS_A || TS_B)$. User B replies to User A using the message $\langle M_4, M_5, TS_B \rangle$, where $M_5 = h(SK \oplus R_A \oplus R_B)$.
5. *Step 5* User A receives the message $\langle M_4, M_5, TS_B \rangle$ and checks the currency of the messages based on $|TS_B - TS_B^*| \leq \Delta T$. If this condition fails, User A terminates the connection.
6. *Step 6* Then, User A extracts TS_B and generates $M_6 = h(ID_S || TS_A || R_A || TS_B)$. R_B , which is obtained by XOR-ing M_6 with M_4 , is used to derive $SK' = h(ID_S || \sigma_i || R_A || R_B || TS_A || TS_B)$. Using SK' , R_A and R_B , User A calculates $M_7 = h(SK' || R_A || R_B)$. If $M_7 = M_5$, User A authenticates User B and verifies that both users were able to derive the same secret session key $SK = SK'$.
7. *Step 7* User A sends a message $\langle M_8, TS_{A'} \rangle$ to User B as an acknowledgment, where $M_8 = h(SK' || ID_S)$ and $TS_{A'}$ is the new time stamp.
8. *Step 8* Finally, User B checks $|TS_{A'} - TS_{A'}^*| \leq \Delta T$ and generates $M_9 = h(SK || ID_S)$. If $M_8 = M_9$, then User B verifies that User A has produced the same secret session key, hence, User A is authenticated. If any of the above steps fails, the connection will be immediately terminated.

At the end of this phase, both users A and B reserve the same secret session key SK , which will be used for secure communication, after performing the mutual authentication phase (Fig. 2).

In order to increase the robustness of the proposed protocol, an additional factor is used to enhance the mutual authentication of users. This factor employs non-cryptographic parameters such as physical channel parameters, traffic and energy consumption. Different features (parameters) are chosen for each entity to generate a unique user profile (fingerprints). This means that device fingerprinting is generated from a set of features that can be obtained from different layers as shown in Fig. 3. For example, the network traffic of each device is frequently monitored and compared to its history log (or any feature in the device's profile). Upon any change in network traffic (for example sudden increase), both devices will have to re-authenticate each other. Moreover, new user-specific credentials are used in every new authentication session. The proposed protocol mainly depends on

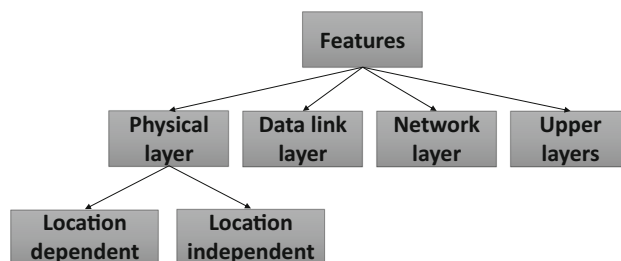


Fig. 3 Existing device features

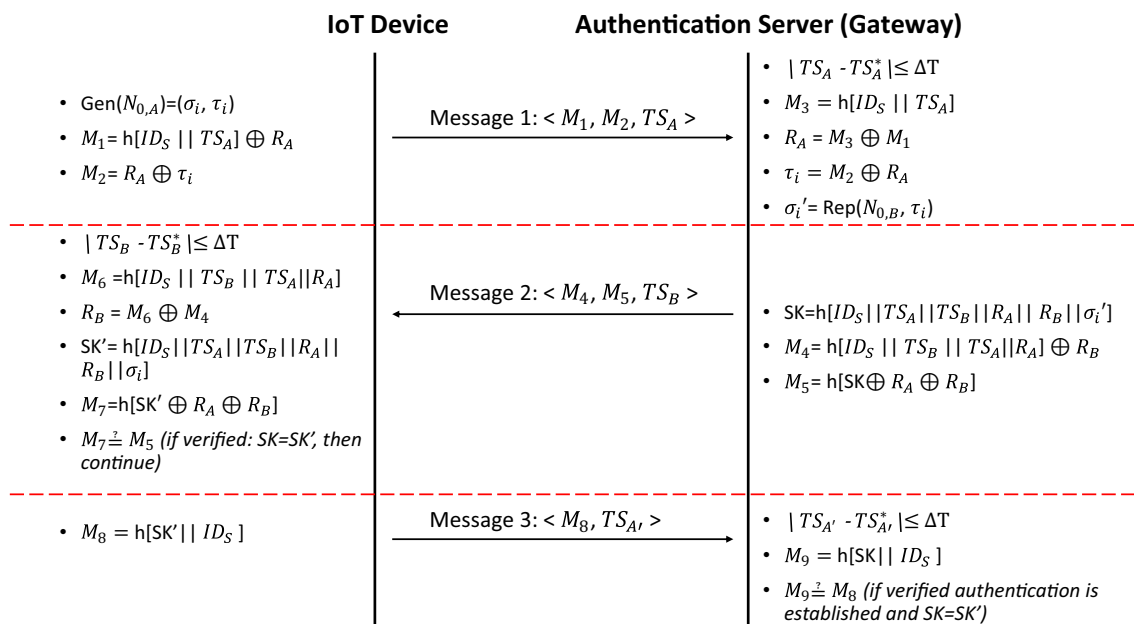


Fig. 2 Proposed PLS authentication protocol

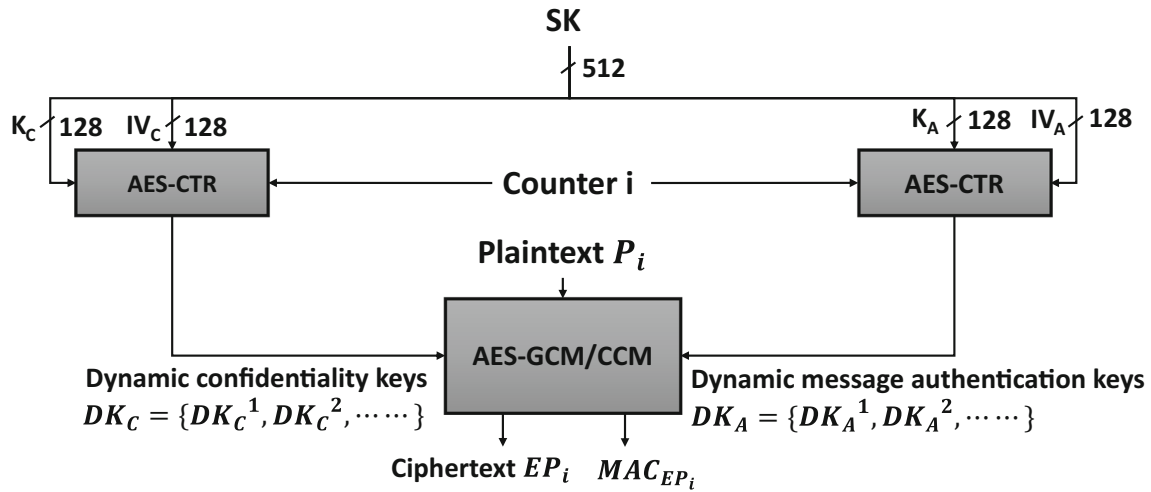


Fig. 4 Proposed dynamic key derivation scheme for data confidentiality and message authentication

the shared channel parameters between a pair of devices and a secret session identifier. Since wireless channels are random and dynamic, channel-based parameters will differ greatly from one time period to another. On the other hand, the secret identifier which is derived from the device’s PUF is constant. To enhance security even further, we propose employing variable challenges and responses (using configurable PUFs). More specifically, input values are updated based on the previous output (recursive function). As such, the secret identifier will be dynamic and able to resist different kinds of attacks.

4.2 Data confidentiality and message integrity

Following the mutual authentication and key agreement phase, the IoT device and gateway will both have the same secret session key $SK = SK'$, which is used for multiple purposes such as data confidentiality, message integrity in addition to source authentication. SK is divided into four sub-keys (128 bits each) which are: K_C , IV_C , K_A , and IV_A as shown in Fig. 4. K_C and IV_C represent the data confidentiality sub-keys and are used to derive the set of dynamic keys $DK_C = \{DK_C^1, DK_C^2, \dots\}$. Whereas K_A and IV_A represent the source authentication keys, which are used to derive the set of dynamic keys $DK_A = \{DK_A^1, DK_A^2, \dots\}$.

For every input message, i , new data confidentiality and message authentication keys are required in order to reach a high level of security. The proposed dynamic key generation technique is mainly based on the Advanced Encryption Standard (AES) with counter mode (CTR). This mode ensures the best performance according to Table 6 and Fig. 7. Moreover, it can be applied in parallel. K_C and IV_C are the initial data confidentiality key and the data confidentiality initial vector, respectively. The i th dynamic confidentiality key DK_C^i

is produced using the initial data confidentiality key K_C and the i th initial vector (IV_C^i). In order to obtain IV_C^i , IV_C is XORed (updated) with the counter ($i - 1$) according to the following equation:

$$IV_C^i = IV_C \oplus (i - 1) \tag{3}$$

The i th dynamic confidentiality key is produced according to the following equation:

$$DK_C^i = AES_{CTR}(IV_C^i, K_C) \tag{4}$$

Similarly, K_A and IV_A represent the initial message authentication key and the message authentication initial vector, respectively. A set of dynamic message authentication keys $DK_A = \{DK_A^1, DK_A^2, \dots\}$ is produced using the same approach (AES in counter mode) as that of the dynamic confidentiality keys. This means that the i th dynamic message authentication key is produced using K_A and the i th initial vector (IV_A^i), which is generated using IV_A and counter ($i - 1$).

At the source, the i th message P_i is encrypted using the dynamic confidentiality key DK_C^i to produce EP_i which represents the corresponding encrypted plaintext (ciphertext). Then, EP_i is authenticated (hashed) using the dynamic message authentication key DK_A^i to produce MAC_{EP_i} which is concatenated with the i th ciphertext EP_i . After passing through the channel, the legitimate receiver separates the received message into two parts, EP_i and MAC_{EP_i} . First, the dynamic confidentiality and message authentication keys (DK_C^i and DK_A^i) are generated as described, previously. Then, EP_i will be used along with DK_A^i to compute the corresponding message authentication code \widehat{MAC}_{EP_i} . If $\widehat{MAC}_{EP_i} = MAC_{EP_i}$, then, EP_i will be decrypted using DK_C^i

to obtain \hat{P} , which should be equal to P . Otherwise, an alert will be issued indicating an active modification attack.

In order to achieve data confidentiality, message integrity, in addition to source authentication, simultaneously, either the Counter with CBC-MAC (CCM) mode or the Galois/Counter Mode (GCM) mode can be used as modes of operation for encryption. Both, the CCM mode and the GCM mode use secure block ciphers with the counter mode. The main difference between the two modes is that CCM ensures data integrity and source authentication using block cipher with CBC mode, while GCM uses Galois multiplier with CBC operation mode. GCM is very fast compared to CCM and it is preferable for real-time IoT applications. Both modes are defined for block ciphers with a block size of 128 bits and are patent-free [40].

5 Security analysis of the proposed authentication scheme

In this section, the proposed scheme is analyzed in the context of different authentication attacks.

5.1 Resistance against privacy threats

In this part, several metrics are presented to prove that the proposed authentication scheme exhibits a high level of immunity against privacy threats.

5.1.1 Indistinguishability

The identity-related information (ID_S) provided by both entities is guaranteed by the one-way property of the hash function and the randomly generated numbers and time stamps (R_A , R_B , TS_A and TS_B). Here, the attacker cannot obtain plaintext information related to the users' identity or the transmitted data, hence, it is indistinguishable.

5.1.2 Anonymity

In this scheme, both entities do not show their true identity in either the authentication stage or later in the data communication stage. In other words, data are anonymous during transmission across the public channel. Even if data are stolen, it is difficult to identify the data owner. Assuming that the adversary intercepts the exchanged messages: $Message_1 = \langle M_1, M_2, TS_A \rangle$, $Message_2 = \langle M_4, M_5, TS_B \rangle$ and $Message_3 = \langle M_8, TS_{A'} \rangle$. Since, TS_A , TS_B , R_A and R_B are unique and dynamic, all of the three messages are distinct where M_1 , M_2 , M_4 , M_5 and M_8 vary greatly with any slight change in the above parameters.

5.1.3 Identity privacy

One of the main security requirements for exchanging information is privacy. Using a set of secret parameters hides the real identity of communicating entities. The real identity of the IoT users is thus preserved and threats related to the user location tracking attacks are not possible in this case.

5.2 Man-In-the-Middle (MIM) attack

The MIM attack is a form of active eavesdropping, where the attacker initiates independent connections with victims and relays messages between them. Assume that the adversary intercepts the first message issued by the sender, $\langle M_1, M_2, TS_A \rangle$. Afterward, he creates another message using the current time stamp TS_E and a randomly generated number R_E . The resulting message will be $\langle [h(ID_S^a || TS_E)] \oplus R_E, (R_E \oplus \tau_i^a), TS_E \rangle$. However, ID_S and τ_i are unknown to the adversary, consequently, another secret identifier ID_S^a and a different channel parameter $Gen(N_{0,E}) = (\sigma_i^a, \tau_i^a)$ will be used to generate M_1^a and M_2^a . In this case, connection will be terminated since both users will not be able to authenticate each other and derive the same secret session key SK. Therefore, the proposed approach is immune against MIM attacks since it relies on a secret and on channel parameters which are unknown to adversaries.

5.3 Resistance against replay attacks

Even if the attacker was able to intercept authentication credentials and resend these credentials back to the legal entity, it is difficult to pass legal authentication due to the validity of the random numbers and time stamps. Consequently, replay attacks are easily prevented.

5.4 Camouflage attack and tracking prevention

At the authentication stage, adversaries shouldn't acquire information related to the real user's identity or their secret credentials. For this purpose, random numbers, fresh time stamps and a one-way hash function are used, in which every new challenge is updated with a fresh time stamp.

Moreover, the authentication mechanism uses a new channel-derived parameter ($Gen(N_0) = (\tau_i, \sigma_i)$), which makes it impossible for attackers to get the content of previous authentication sessions. This is attributed to the fact that the wireless channels are random and dynamic, hence, extracted channel parameters vary greatly from one session to another. Accordingly, the proposed protocol effectively resists camouflage attacks and prevents tracking.

5.5 Masquerading, forgery and impersonation attacks

In the impersonation/masquerading attack, adversaries try to deceive users by pretending to be a legitimate sender/receiver. In the proposed scheme, all of the exchanged messages require a valid secret identifier ID_S , which only known to the legitimate users. Consequently, the impersonation attack is only feasible if the adversary acquires ID_S (very unlikely).

5.6 Forward secrecy

Forward secrecy is achieved by the one-way property of the hash function. Even if the adversary acquires the used channel-based parameters at the authentication stage, he will not be able to derive the same secret session key SK since a secret session identifier is utilized.

It should be noted that having a **malware** embedded in one of the communicating device compromises device's credentials (ID_S). This issue is not the focus of this paper. Moreover, if the adversary is present on the same subnet, he will also be able to extract similar channel-based parameters. Security and authentication are threatened in this case, and other alternate solutions should be taken into consideration.

5.7 Security of secret session keys

In this subsection, we provide a brief formal (mathematical) analysis to assess the security of the proposed protocol. Similar to [28,30,31], the Real-Or-Random (ROR) model is used to prove the robustness of the produced secret session key SK. It should be noted that not all attacks are captured by mathematical modeling.

We assume that the adversary is able to eavesdrop, modify, inject, and fabricate messages using the following queries [28,30,31]:

- $Execute(\Lambda^v, \Lambda^w)$: This represents a passive attack, where an adversary is able to read the transmitted messages between legitimate participants at instances v and w (Λ^v and Λ^w).
- $Reveal(\Lambda^v)$: This query reveals SK to the adversary.
- $Send(\Lambda^v, MSG)$: This models an active attack, where an adversary sends a message MSG to a participant instance Λ^v , and receives a reply back.
- $Test(\Lambda^v)$: This corresponds to the security of the secret session key SK between the IoT user and gateway following the indistinguishability style in the ROR model [28]. Here, an unbiased coin c is flipped before the experiment starts. Λ^v returns SK if $c = 1$ otherwise, it returns a random number.

The adversary initiates $Test$ queries to either the IoT device or the gateway. If the guessed bit c' is equal to the random bit c , the adversary wins the game ($Succ$). According to [28,41], the adversary's advantage in breaking the security of the proposed approach and deriving SK is $Adv_{proposed} = |2 \cdot Pr[Succ] - 1|$. Using the ROR model, the proposed scheme is secure if $Adv_{proposed} \leq \epsilon$, where $\epsilon > 0$ is very small.

Theorem 1 *The secret session key SK is secure against adversaries. Using the ROR model, $Adv_{proposed} \leq \frac{q_h^2}{|H|}$, where q_h and $|H|$ are the number of access times to a collision-resistant hash function $h(\cdot)$ and the range of space of a hash function $h(\cdot)$, respectively.*

Proof We modify the approaches in [28,30,31], where three games, G_i ($i = 0, 1, 2$) are defined.

- Game G_0 : This represents the original attack on the protocol using a random bit test. Since c should be guessed by the adversary before the game starts, by definition, we have:

$$Adv_{proposed} = |2Pr[Succ_0] - 1|. \quad (5)$$

- Game G_1 : G_0 is transformed to G_1 . Here, the adversary intercepts (eavesdropping) the transmitted messages between the sender and receiver ($Execute$ query). The adversary uses the $Test$ and $Reveal$ queries to test whether the $Test$ query gives the real value of SK. Since the secret session key contains short and long-term secrets, the adversary's chance of winning this game is not increased by eavesdropping the exchanged messages. Hence, it is clear that

$$Pr[Succ_0] = Pr[Succ_1]. \quad (6)$$

- Game G_2 : G_1 is transformed to G_2 , which is an active attack. The adversary performs several $Send$ queries in order to guess the output of the hash functions of the transmitted messages. However, these messages also include long- and short-term secrets. As a result, this will lead to no collision which gives the following:

$$Pr[Succ_1] \leq \frac{q_h^2}{2|H|} + Pr[Succ_2]. \quad (7)$$

Since the adversary has no choice other than guessing the bit c in order to win the game, we get

$$Pr[Succ_2] = \frac{1}{2}. \quad (8)$$

From Eqs. (7) and (8), it follows that:

$$Pr[Succ_1] \leq \frac{q_h^2}{2|H|} + \frac{1}{2}, \tag{9}$$

$$Pr[Succ_0] \leq \frac{q_h^2}{2|H|} + \frac{1}{2}. \tag{10}$$

Using Eq. (5), we get:

$$Adv_{proposed} \leq \left| 2 \left[\frac{q_h^2}{2|H|} + \frac{1}{2} \right] - 1 \right|, \tag{11}$$

$$Adv_{proposed} \leq \frac{q_h^2}{|H|}. \tag{12}$$

Since the range of space of a hash function $|H|$ is much greater than the number of $Test$ queries, $\frac{q_h^2}{|H|}$ is negligible. Consequently, $Adv_{proposed} \leq \epsilon$, which proves that SK and data transmitted using the proposed scheme are secure. For a detailed discussion, refer to [28,30,31]. \square

6 Security evaluation of the proposed authentication protocol using AVISPA

In this section, we verify the security of the proposed authentication scheme using the widely used AVISPA tool [36,42–47]. The AVISPA tool includes four backends which are: (1) On-the-Fly Model Checker (OFMC), (2) Constraint-Logic-based Attack Searcher (CL-AtSe), (3) SAT-based Model Checker (SATMC) and (4) Tree Automata based on Automatic Approximation for the Analysis of Security Protocols (TA4SP). Here, the results of the proposed scheme under the SATMC and TA4SP backends are omitted since these backends don't support the bit-wise XOR operation (result: "inconclusive"). The proposed scheme is implemented using the High-Level Protocol Specification Language (HLPSL) and is simulated in Security Protocol Animator for AVISPA (SPAN). It should be noted that this tool captures the replay and the Man-In-the-Middle (MIM) attacks only.

Figure 5 proves that the proposed protocol is "safe" against the replay attacks and the MIM attack; hence, it satisfies the design properties of a secure authentication protocol.

7 Security analysis of the proposed dynamic key function

The proposed key generation scheme enhances the security of data confidentiality and message authentication keys. More specifically, communicating entities perform device authentication first and then generate the same key SK. Using this key, the initial vectors, IV_C and IV_A , which are used to

achieve data confidentiality and message integrity along with the initial keys K_C and K_A , are obtained. In traditional AES with CTR/CCM/GCM operation modes, initial vectors are exchanged along with the transmitted data, so that both are able to perform correct source authentication, data integrity and/or encryption/decryption. Using the proposed approach, exchanging the initial vectors is no longer a requirement since both ends are able to obtain and update the same initial vectors in a synchronized and secret manner. This increases the security level of produced dynamic keys (data confidentiality and message integrity) against different key-related attacks such as weak keys and key disclosure attack and reduces the communication overhead and required power.

8 PUF-based threats

In [20], the authors consider two main attack models. The first model assumes that the adversary is able to intercept the communicated messages between devices (Man-In-the-Middle attack), and the second one assumes that the adversary has physical access to the device (side channel attack).

8.1 Man-In-the-Middle attack

The MIM attack allows adversaries to capture messages that are communicated between two devices such as the exchanged challenges and responses. However, it has been proven earlier that the proposed protocol is immune against this type of attacks.

8.2 Side channel attack

In such an attack, the adversary has physical access to the device. The attack can be invasive, semi-invasive or non-invasive, and it be either passive or active, according to [11].

In general, invasive (active) attacks are complicated and costly since adversaries have to move the compromised IoT device to a specialized laboratory, where expensive laboratory equipment is available. This type of attacks is not convenient for IoT devices, especially when devices are located in public places; hence, bringing them to a laboratory is not possible.

On the other hand, semi-invasive (active) attacks require the emission of photonics and electromagnetic sampling, and they depend on much simpler techniques compared to invasive attacks. In particular, the semi-invasive attacker should have access to the chip surface, which will not be damaged by the attack. However, this technique also requires moving the IoT device to the laboratory and utilizing special equipment.

Unlike invasive and semi-invasive attacks, the equipment needed for conducting non-invasive attacks can be trans-

Fig. 5 Simulation results using the AVISPA tool under **a** OFDM and **b** CL-AtSe backends

<pre>% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/protocol.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.02s visitedNodes: 18 nodes depth: 4 plies</pre>	<pre>SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/protocol.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 14 states Reachable : 4 states Translation: 0.01 seconds Computation: 22.83 seconds</pre>
--	---

(a) OFMC

(b) CL-AtSe

ported and installed near the attacked IoT devices. These equipment are relatively small and inexpensive.

Moreover, non-invasive (passive) attacks do not require direct access to internal components. In this technique, secret information is extracted by exploiting data related to power consumption and time delay. Non-invasive attacks use analysis tools based on machine learning algorithms. Therefore, IoT devices are prone to this type of attacks. To prevent non-invasive attacks, we have used a dynamic challenge-response authentication protocol that is based on physical channel parameters. The introduced dynamic physical properties prevent side channel attackers from recovering any useful information (channel parameters have a high level of randomness).

In order to reinforce the resistance against side channel attacks, proper defense strategies (restrictions on physical access) for IoT devices should be taken into account.

9 Performance evaluation of the proposed authentication scheme

In this section, we evaluate the performance of the proposed protocol in comparison with the protocols presented in [28–31]. The tested parameters include communication cost, computational cost and execution time. Although, the protocols in [28–31] do not utilize the notion of PLS, they use PUFs in the authentication process. In contrast, the proposed scheme combines both PUFs and PLS, to increase the robustness and efficiency of multi-homed systems, which is the main contribution of this paper.

9.1 Communication costs

For comparatives purposes, we assume that the identity or the secret session identifier in our case is 160 bits, the random number is also 160 bits, the time stamp is 32 bits, and the hash digest is 160 bits (using the SHA-1 hash function as in [28]).

The protocol presented in [28] requires the exchange of three messages, which consist of 512, 512, 192 bits, respectively. Consequently, the total communication cost is 1216 bits. On the other hand, the total communication cost of the scheme presented in [29] is 1856 bits, where four messages are needed to achieve mutual authentication. Similarly, the authentication protocols in [30,31] require four messages as a communication overhead, and a total of 2752 bits and 2080 bits, respectively. The proposed protocol involves the exchange of three messages: (1) $\langle M_1, M_2, TS_A \rangle$, (2) $\langle M_4, M_5, TS_B \rangle$ and (3) $\langle M_8, TS_{A'} \rangle$, however, it requires a fewer number of bits. The first message consists of $(160 + 160 + 32) = 352$. The second message is also composed of $(160 + 160 + 32) = 352$ bits whereas the final message requires $(160 + 32) = 192$ bits only. Hence, the total number of required bits is $(352 + 352 + 192) = 896$ bits, which less than 1216 bits [28]. In other words, the proposed scheme is more efficient than the scheme presented in [28] in terms of communication cost (Table 3).

9.2 Computational cost

In order to asses the computational costs of the proposed scheme, we identify the following parameters: T_h , T_{xor} , T_f and T_E , which denote the time of the hash function, the time of the XOR operation, the time of the fuzzy extractor, and

Table 3 Communication cost

Scheme	Required messages	Required bits
Protocol in [28]	3	1216
Protocol in [29]	4	1856
Protocol in [30]	4	2752
Protocol in [31]	4	2080
Proposed	3	896

Table 4 Communication cost

Scheme	Computational delay
Protocol in [28]	$17T_h + 8T_{xor} + 1T_f$
Protocol in [29]	$21T_h + 3T_E$
Protocol in [30]	$31T_h + 4T_E + 1T_f$
Protocol in [31]	$19T_h + 7T_{xor}$
Proposed	$10T_h + 10T_{xor} + 1T_f$

the time of the elliptic curve cryptosystem (ECC) point multiplication, respectively. The total computational delay of the schemes presented in [28–31] is:

$$Delay_{[28]} = 17T_h + 8T_{xor} + 1T_f, \tag{13}$$

$$Delay_{[29]} = 21T_h + 3T_E, \tag{14}$$

$$Delay_{[30]} = 31T_h + 4T_E + 1T_f, \tag{15}$$

$$Delay_{[31]} = 19T_h + 7T_{xor}. \tag{16}$$

On the other hand, the total computational delay of the proposed scheme is only:

$$Delay_{proposed} = 10T_h + 10T_{xor} + 1T_f. \tag{17}$$

Since the time required by the XOR operation is much less than that of the hash operation (negligible), we can conclude that the proposed scheme outperforms the schemes in [28–31].

Table 4 summaries the computational cost in terms of delay.

9.3 Execution time

In order to evaluate the execution time of (1) the proposed authentication protocol, (2) the dynamic key generation scheme and (3) the data confidentiality and message integrity procedure, we use “OpenSSL”, which is a very popular tool

Table 5 Execution time (sec) of the SHA-256 and SHA-512 hash functions

Type	16 bytes	64 bytes	256 bytes	1024 bytes
SHA-256	2.1209e-07	3.7359e-07	7.7219e-07	2.3952e-06
SHA-512	2.9697e-07	2.7289e-07	6.3527e-07	1.7382e-06

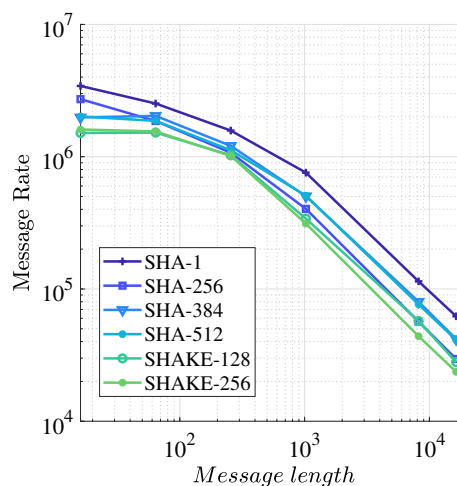


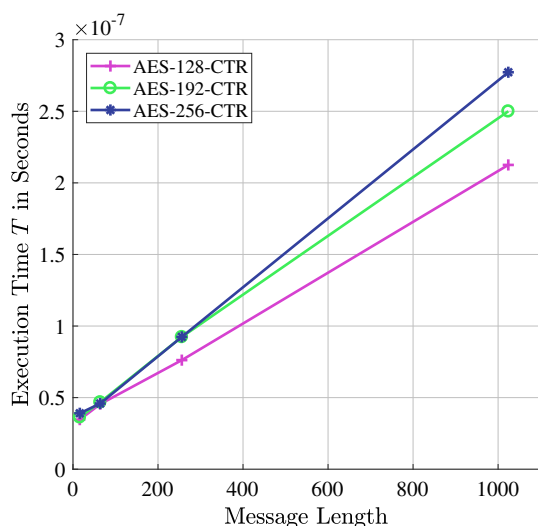
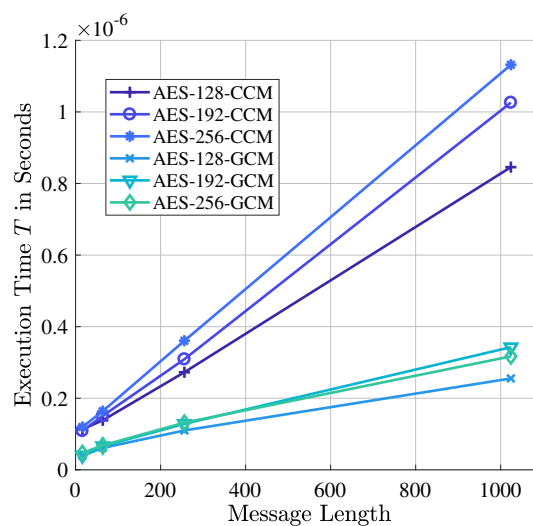
Fig. 6 The number of hashed messages in one second using different hash functions, versus message length

and is widely used since it is considered as one of the most important and efficient cryptographic libraries that provide robust, commercial-grade and full-featured toolkit for Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. Moreover, OpenSSL is implemented on a common IoT hardware which is Raspberry Pi 2, which has a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor. To show the efficiency of the proposed authentication protocol, we compute the time required by the different hash functions, mainly the SHA-256 and SHA-512 functions (Table 5 and Fig. 6). For different message sizes (block size in bytes), it is evident that the SHA-512 requires less time, hence, it is more efficient. In contrast, the SHA-256 is efficient for small-size messages only (16 bytes). On the other hand, the execution time of AES OpenSSL with the counter mode (CTR) is analyzed to show the efficiency of the proposed key derivation function. The obtained results indicate that there exists a trade-off between the security level (increasing the size of the secret key) and the required latency (Fig. 7).

Finally, AES OpenSSL is used to compare two authentication/encryption operation modes, CCM and GCM, which ensure data confidentiality and message integrity in addition to source authentication. In general, CCM (Counter with CBC-MAC) and GCM (Galois/Counter Mode) modes are preferred over CTR (Counter) since both modes support data confidentiality and message integrity simultaneously, unlike the CTR mode. Table 6 illustrates the numerical values of the

Table 6 Execution time (sec) of different cryptographic modes

Type	16 bytes	64 bytes	256 bytes	1024 bytes
AES-128-CTR	3.4632e-08	4.5463e-08	7.6184e-08	2.1253e-07
AES-192-CTR	3.6296e-08	4.6812e-08	9.2378e-08	2.5000e-07
AES-256-CTR	3.9062e-08	4.5866e-08	9.2423e-08	2.7728e-07
AES-128-CCM	1.1330e-07	1.3839e-07	2.7262e-07	8.4573e-07
AES-192-CCM	1.0878e-07	1.5083e-07	3.0909e-07	1.0260e-06
AES-256-CCM	1.1967e-07	1.6399e-07	3.6056e-07	1.1317e-06
AES-128-GCM	3.9161e-08	6.1339e-08	1.0996e-07	2.5529e-07
AES-192-GCM	4.0285e-08	6.6155e-08	1.2841e-07	3.1697e-07
AES-256-GCM	4.6874e-08	6.8451e-08	1.3173e-07	3.4262e-07

**Fig. 7** Execution time versus message length of CTR mode**Fig. 8** Execution time versus message length of CCM and GCM operation modes

execution time for different cryptographic operation modes using AES OpenSSL implementation. The obtained results prove that the GCM mode is more efficient than the CCM mode and thus recommended since it requires less execution time for different AES (Advanced Encryption Standard) symmetric key sizes (Fig. 8). As such, it is more suitable for IoT applications. It should be noted that as the message size increases, the time required to perform encryption and message authentication also increases. This is logical since as the number of bytes increase, more time is required to process (encrypt) them.

10 Conclusion

In this paper, a novel and lightweight mutual multi-factor authentication protocol is proposed for IoT systems. The proposed scheme allows IoT entities to mutually authenticate each other and agree on a shared secret session key,

which will later be used for securing and authenticating the data transmitted over public wireless channels. In particular, a secret identifier based on a re-configurable PUF in each IoT device is employed, along with a channel-derived nonce to ensure high authentication accuracy. The lightweight of the proposed scheme is attributed to the fact that a low number of simple operations and communication messages is required, mainly the bit-wise exclusive-OR operation and the one-way hash function, which is convenient for resource-limited IoT devices. In addition, a novel physical data confidentiality and message integrity scheme is proposed. Informal security analysis, security evaluation and mathematical proofs confirm that the proposed scheme is secure and robust against different authentication attacks and outperforms the existing solutions since it has low communication and computational costs.

Funding This research is supported by the Maroun Semaan Faculty of Engineering and Architecture at the American University of Beirut.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Aman, M., Chua, K., Sikdar, B.: A lightweight mutual authentication protocol for IoT systems. In: Proceeding of IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2017)
- Barbareschi, M., Bagnasco, P., Mazzeo, A.: Authenticating IOT devices with physically unclonable functions models. In: IEEE Proceeding of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), pp. 563–567. IEEE (2015)
- Granjal, J., Jorge, M., Monteiro, E., Silva, J.: Security for the internet of things: a survey of existing protocols and open research issues. *IEEE Commun. Surv. Tutor* **17**(3), 1294–1312 (2015)
- Zamfir, S., et al.: A security analysis on standard IoT protocols. In: IEEE Proceeding of International Conference on Applied and Theoretical Electricity (ICATE), pp. 1–6. IEEE (2016)
- Bauer, T., Hamlet, J.: Physical unclonable functions: a primer. *IEEE Secur. Priv.* **6**, 97–101 (2014)
- Aman, M., Chua, K., Sikdar, B.: Secure data provenance for the internet of things. In: Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, pp. 11–14. ACM (2017)
- Wallrabenstein, J.: Practical and secure IoT device authentication using physical unclonable functions. In: IEEE Proceeding of International Conference on Future Internet of Things and Cloud (FiCloud), pp. 99–106. IEEE (2016)
- Che, W., Saqib, F., Plusquellic, J.: PUF-based authentication. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 337–344. IEEE (2015)
- Xu, Q., Zheng, R., Saad, W., Han, Z.: Device fingerprinting in wireless networks: challenges and opportunities. *IEEE Commun. Surv. Tutor.* **18**(1), 94–104 (2016)
- Aman, M., Chua, K., Sikdar, B.: Physically secure mutual authentication for IoT. In: IEEE Proceeding Conference on Dependable and Secure Computing, pages 310–317. IEEE, 2017
- Aman, M., Chua, K., Sikdar, B.B.: Mutual authentication in IoT systems using physical unclonable functions. *IEEE Internet Things J.* **4**(5), 1327–1340 (2017)
- Pospl, M., Mark, R.: Experimental study of wireless transceiver authentication using carrier frequency offset monitoring. In: International Conference on Radioelektronika (RADIOELEKTRONIKA), pp. 335–338 (2015)
- Liu, M., et al.: TBAS: enhancing wi-fi authentication by actively eliciting channel state information. In: IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–9 (2016)
- Du, X., et al.: Physical layer challenge-response authentication in wireless networks with relay. In: Proceeding IEEE International Conference on Computer Communications (INFOCOM), pp. 1276–1284 (2014)
- Caparra, G., et al.: Energy-based anchor node selection for IoT physical layer authentication. In: Proceeding IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2016)
- Wen, H., et al.: A novel framework for message authentication in vehicular communication networks. In: Proceeding IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2009)
- Verma, G., et al.: Physical layer authentication via fingerprint embedding using software-defined radios. *IEEE Access* **3**, 81–88 (2015)
- Wu, X., et al.: A channel coding approach for physical-layer authentication. In: IEEE Proceeding of Wireless Communications and Signal Processing (WCSP), pp. 1–5. IEEE (2016)
- Wu, X., et al.: Artificial-noise-aided physical layer phase challenge-response authentication for practical OFDM transmission. *IEEE Trans. Wirel. Commun.* **15**(10), 6611–6625 (2016)
- Babaei, A., et al.: Physical unclonable functions in the Internet of Things: state of the art and open challenges. *Sensors* **19**, 3208 (2019)
- Liu, F., et al.: A two dimensional quantization algorithm for CIR-based physical layer authentication. In: 2013 IEEE International Conference on Communications (ICC) pp. 4724–4728 (2013)
- Zhang, J., et al.: Using basis expansion model for physical layer authentication in time-variant system. In: IEEE Conference on Communications and Network Security (CNS), pp. 348–349. IEEE (2016)
- Wang, W., et al.: Privacy-preserving location authentication in Wi-Fi networks using fine-grained physical layer signatures. *IEEE Trans. Wirel. Commun.* **15**(2), 1218–1225 (2016)
- Melki, R., et al.: A survey on OFDM physical layer security. *Phys. Commun.* **32**, 1–30 (2019)
- Gope, P., et al.: Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment. In: Sloom, P., Cambria, E., Abramson, D., Altintas, I. (eds.) *Future Generation Computer Systems*, vol. 83, pp. 629–637. Elsevier, Amsterdam (2018)
- Amin, R., et al.: A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. In: Sloom, P., Cambria, E., Abramson, D., Altintas, I. (eds.) *Future Generation Computer Systems*, vol. 78, pp. 1005–1019. Elsevier, Amsterdam (2018)
- Kunal, S., et al.: An overview of cloud-fog computing: architectures, applications with security challenges. In: *Security and Privacy*, vol. 2, pp. e72. Wiley, New York (2019)
- Das, K., Wazid, M., Kumar, N., Khan, M., Choo, K., Park, Y.: Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE J. Biomed. Health Inform.* **22**(4), 1310–1322 (2018)
- Li, X., et al.: A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *J. Netw. Comput. Appl.* **103**, 194–204 (2018)
- Das, A., et al.: Provably secure user authentication and key agreement scheme for wireless sensor networks. *Secur. Commun. Netw.* **9**(16), 3670–3687 (2016)
- Chang, C., Le, H.: A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Trans. Wirel. Commun.* **15**(1), 357–366 (2016)
- Atzori, L., et al.: The internet of things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)
- Van den Abeele, F., et al.: Scalability analysis of large-scale LoRaWAN networks in ns-3. *IEEE Internet Things J.* **4**(6), 2186–2198 (2017)
- Dolev, D., Yao, A.: On the security of public key protocols. *IEEE Trans. Inform. Theory* **29**(2), 198–208 (1983)
- Delignat-Lavaud, A.: On the security of authentication protocols on the web. PhD thesis, Paris Sciences et Lettres (2016)
- Amin, R., et al.: A software agent enabled biometric security algorithm for secure file access in consumer storage devices. *IEEE Trans. Consum. Electron.* **63**(1), 53–61 (2017)

37. Mesaritakis, C., et al.: Physical unclonable function based on a multi-mode optical waveguide. *Sci. Rep.* **8**, 9653 (2018)
38. Hamamreh, J., Arslan, H.: Secure orthogonal transform division multiplexing (OTDM) waveform for 5g and beyond. *IEEE Commun. Lett.* **21**(5), 1191–1194 (2017)
39. Badawy, A., et al.: Unleashing the secure potential of the wireless physical layer: secret key generation methods. *Phys. Commun.* **19**, 1–10 (2016)
40. Szalachowski, P., Ksiezopolski, B., Kotulski, Z.: CMAC, CCM and GCM/GMAC: advanced modes of operation of symmetric block ciphers in wireless sensor networks. *Inf. Process. Lett.* **110**(7), 247–251 (2010)
41. Abdalla, M., Fouque, P., Pointcheval, D.: Password-based authenticated key exchange in the three-party setting. In: *International Workshop on Public Key Cryptography*, pp. 65–84. Springer (2005)
42. Odelu, V., Das, A., Goswami, A.: SEAP: secure and efficient authentication protocol for NFC applications using pseudonyms. *IEEE Trans. Consum. Electron.* **62**(1), 30–38 (2016)
43. Wazid, M., et al.: A novel authentication and key agreement scheme for implantable medical devices deployment. *IEEE J. Biomed. Health Inform.* **22**(4), 1299–1309 (2018)
44. Wazid, M., et al.: Secure authentication scheme for medicine anti-counterfeiting system in IoT environment. *IEEE Internet Things J.* **4**(5), 1634–1646 (2017)
45. Challa, S., et al.: Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access* **5**, 3028–3043 (2017)
46. Chatterjee, S., et al.: Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. *IEEE Trans. Dependable Secure Comput.* **15**(5), 824–839 (2018)
47. Amin, R., et al.: Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. *Wirel. Pers. Commun.* **84**(1), 439–642 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.