Journal of Engineering Science and Technology
Vol. 16, No. 5 (2021) 3987 - 4005

A NEW LIGHTWEIGHT AUTHENTICATED KEY AGREEMENT PROTOCOL FOR IOT IN CLOUD COMPUTING

AHMED H. ALY*, ATEF GHALWASH, MONA NASR,
AHMED ABD EL-HAFEZ

Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt
*Corresponding Author: ahmed71.aly@gmail.com

Abstract

Internet of Things (IoT) is a pervasive technology that grants authorized users the ability to communicate with sensors and devices. This technology connects millions of devices, exchanges sensitive information with users, and off-loads classified information to the cloud. This technology is evolving to encompass time-critical applications. In IoT-time critical applications, legitimate users may require accessing the real-time data directly from the IoT devices rather than requesting data stored in the cloud. These IoT devices are prone to distinct threats and security breaches. Authentication mechanisms are substantial to control access to IoT devices in cloud computing, as authorized users and IoT devices should ensure the authenticity of each other and generate a session key for securing the exchanged traffic. As different IoT devices are resource-constrained, traditional security mechanisms will not be appropriate for these devices, as they need considerable computational power and consume excessive energy. Cryptographic researchers are exerting a worthy effort to develop lightweight security mechanisms to cope with resource-constrained IoT systems. In this paper, we propose a novel lightweight protocol (Light-AHAKA) for authenticating IoT-cloud elements and establishing a key agreement for encrypting the exchanged sensitive data. Security analysis of the (Light-AHAKA) is carried out to assure the protocol immunity to different security attacks.

Keywords: Cloud, Cryptography, IoT, Key agreement, Lightweight authentication.

1. Introduction

Internet of Things (IoT) is a proliferating technology that allows users to interact and exchange sensitive data with sensors and devices while enabling these sensors and devices to connect to cloud computing servers without human intervention [1]. Connected IoT devices are enormously increasing, by 2025 more than 30.9 billion IoT devices will be connected, and researchers will spend greater than three trillion US dollars for developing the IoT hardware [2]. It is also promising to see that the new wireless generation (5G) shall encompass people-to-machine and machine-to-machine communications [3], as this generation supports the high demanding speeds needed for the vast growing IoT devices.

Based on the application used, IoT devices send a tremendous amount of data, real-time videos, and photos to cloud servers that need huge storing areas, and high processing. Traditional platforms are considered a meager solution to cope with the extensive amount of data offloaded by IoT devices. From the perspective of storage and processing, using cloud computing as a large resource pool is the appropriate solution to manipulate the massive data generated. In many use cases, authorized users need to access the real-time data directly from designated IoT devices for instant and critical decisions (i.e., Healthcare- Fire ignition -Traffic Congestion-etc.).

Before establishing the communication between the remote users and the targeted IoT devices, the cloud servers are responsible for the mutual authentication between the communicating parties, as the authentication is considered as the forefront defense against cyber-attacks. However, the constrained nature of the IoT devices makes the devices susceptible to different vulnerabilities and security attacks. Meanwhile, hackers are developing new methods to exploit poor security mechanisms implemented in IoT devices [4]. As traditional security mechanisms consume excessive energy, large memory storage, and high processing, researchers are adopting the topic of 'Lightweight Security Mechanisms' to implement it in a resource-constrained IoT system, exerting a big effort to overcome the attackers' developing technologies. Research is considering strong lightweight authentication mechanisms as the most important aspect of IoT-Cloud networks to guarantee the appropriate functionality of these networks [5-7].

1.1. Research questions

In the context of designing a new lightweight authenticated key agreement protocol, it is of our concern to identify the security flaws of existing authentication mechanisms. To examine this, our research questions are as follows:

Q1: What is the state of art proposed lightweight authentication protocols in IoT-Cloud computing?

Q2: What are the drawbacks and gaps in the proposed protocols?

Q3: To fill the gaps, what is the best approach to design a new lightweight authenticated key agreement protocol for IoT in cloud computing?

1.2. Research objective

The research objectives are as follows:

- Surveying the lightweight authentication protocols in the context of IoT- Cloud computing, security, performance, and identifying the existing drawbacks.
- Designing a new lightweight authenticated key agreement protocol to enhance the resiliency of IoT-Cloud computing.
- Conducting a security analysis to ensure the security of the (Light-AHAKA).

2. Organization of Paper

The upcoming sections of the paper are structured in the following way. Section 3 of the paper, surveys the prominent and recent authentication schemes proposed specifically for IoT networks and identifies the research gap. The network model and assumptions are presented in section 4. The (Light-AHAKA) is introduced thoroughly in section 5. Security analysis of the (Light-AHAKA) has been performed against various security attacks in section 6. Finally, section 7 discusses the conclusion of the paper.

3. Related Work

In an endeavor to protect different IoT systems from potential attacks, researchers are exerting a significant effort, as these attacks can lead to drastic havoc in different IoT applications. Accordingly, authentication is a pivotal requirement in IoT systems, as it is considered as the first forefront in the security system. All participants in the IoT systems should be trusted, as jeopardizing one participant can lead to the downfall of the entire system.

Based on Public Key Infrastructure (PKI), which is considered a heavy cryptographic mechanism from the aspect of the computation and communication cost, Ning [8] and Pranata et al. [9] proposed a lightweight PKI framework to adapt to the IoT-constrained nature. Kothmayr et al. [10] enhanced their previous work [11] and presented a two-way IoT authentication protocol. Their protocol is based on the handshake of the Datagram Transport Layer Security (DTLS) protocol with the exchange of (X.509) certificates. Qi et al. [12] studied the case of Low-Earth-Orbit Satellite (LEOs) communication systems and proposed a new authentication and key management scheme for devices working in this environment. The scheme is based mainly on Elliptic Curve Cryptography (ECC) and Symmetric Cryptography.

Based on multi-factor authentication, Alizai et al. [13] presented a novel factor called the device capability and combined this factor with the digital signature. They considered the combination as a two-factor authentication scheme. Device capability is used to test if the device can perform certain operations and provide the expected result on the provided data. Using one authentication factor like passwords or smart cards makes the network susceptible to various security attacks. Hence, Lu et al. [14] proposed an enhanced biometrics-based remote user authentication scheme based on empowering smart cards and passwords with personal biometrics.

To enhance the security and access control in the IoT- cloud computing, Yu et al. [15] analyzed the proposal of He et al. [16] and showed that it is prone to insider attack and DoS attacks. For overcoming these vulnerabilities, they presented an enhanced scheme using a 'fuzzy verifier' to reinforce the registration and authentication phases. Additionally, user validation is transferred to be the

responsibility of the cloud server. Zhou et al. [17] provided a lightweight two-factor authentication scheme based on the one-way hash function and XOR operation. The authors performed formal verification for their proposed protocol using the Prover if tool to ensure that the protocol is immune to the attacks imposed by the tool.

Sahoo et al. [18] proposed a lightweight authentication protocol to support cloud-based IoT applications. They used a one-way hash function, XOR operation, and symmetric encryption/decryption. The security analysis showed that their protocol is immune to replay attacks, insider attacks, and server spoofing attacks. The protocol achieves user anonymity, mutual authentication, and perfect forward secrecy. The security analysis did not address other security attacks the system is prone to. Shah and Venkatesan [19] proposed a protocol depending on storing keys in vaults stored in IoT devices and cloud servers. The set of stored keys is changed after every successful session. To increase the security of the protocol, the key size stored in the vault must be increased, which requires more energy and high processing; this is considered the main drawback of the protocol. Besides, the protocol is susceptible to security attacks as shown in Table 1. Esfahani et al. [20] designed a lightweight authentication mechanism for machine-to-machine in an industrial IoT environment, based only on the hash function and XOR operations. The proposed scheme is not immune to the attacks shown in Table 1.

For Mobile Edge Computing (MEC) which is inherited from cloud computing, Kaur et al. [21] proposed a lightweight authentication protocol based on (ECC) and a one-way hash function, using timestamps to resist replay attacks. Rabiah et al. [22], provided a lightweight authentication and key exchange protocol for the case when IoT devices and gateway are connected through a wireless channel. The protocol relies on loading the IoT device and the gateway during the configuration time with two unique keys: a master key and an initial session key. The session key changes from one session to another and is used as the key for securely exchanging the frames during each session. The authors assumed that the MAC layer protocol is error-free and delivers packets in-sequence which is not true in the real environment. Additionally, the protocol is prone to security attacks shown in Table 1. Neither formal analysis nor performance evaluation for the proposed protocol was conducted. Amin et al. [1] developed a lightweight authentication protocol for IoT devices in the distributed cloud computing environment. They analyzed and pointed out the security vulnerabilities in the multi-server cloud environment of the protocols proposed by Chuang and Chen [23] and Xue et al. [24]. But Wang et al. [25] scrutinized their protocol and showed that if the data in the smart card is revealed, this scheme will not withstand offline password guessing attacks. This eliminates the main objective of "two-factor security"; furthermore, it provides no forward secrecy and cannot guarantee user un-traceability.

3.1. Research gap

According to our survey presented in section 3, the research gaps are as follows:

- The work dedicated to authenticating legitimate users to IoT devices in cloud computing applications is not considering authenticating the IoT devices as well as authenticating the users that require access to these devices.
- Additionally, this work is not resilient enough against IoT network potential attacks.

- In this regard, there is still a gap between the presented lightweight authentication mechanism and the security breaches.
- To fill this gap, we propose a novel lightweight authenticated key agreement protocol (Light-AHAKA) to enhance IoT security in the cloud computing environment.

3.2. Our Contribution

The major contributions of our proposal are as follows:

- Designing a lightweight authentication protocol based on the challenge-response mechanism achieving the following:
  - Mutual authentication considering the constrained nature of the IoT devices and generating a fresh session key.
  - Securing the exchanged parameters and the messages, with no need for secure channels in any step.
  - Updating the pre-shared keys, passwords, and participant identities, to empower the resiliency of the IoT-Cloud network.
- Conducting a security analysis is to verify the immunity of the (Light-AHAKA) against distinct security attacks.

4. Network Model and Assumptions

4.1. Network model

In Fig. 1 our network model is presented. In this model, the IoT devices are deployed in the IoT environment (Healthcare-Industrial Plant-Smart Traffic Signal Monitoring- Connected Vehicles). The IoT devices are connected to the cloud servers via the internet. Authorized users may require accessing the data directly from assigned IoT devices for instantaneous and crucial decisions. The authentication server is the element part responsible for authenticating the network participants. The authentication server must authenticate the legitimacy of the user as well as the intended IoT device before establishing the communication between them.

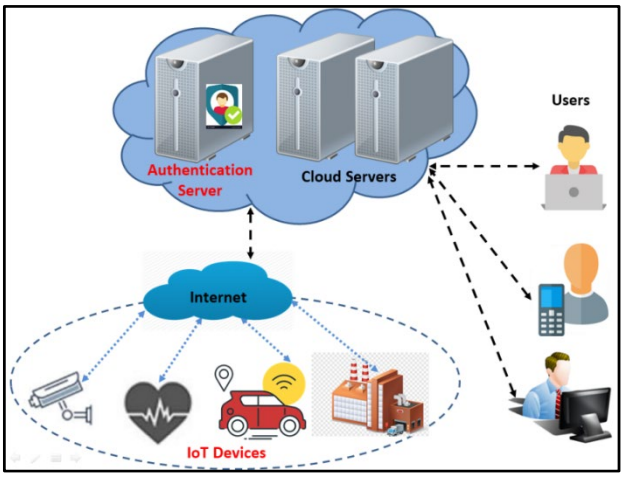

Fig. 1. Network model.

4.2. Assumptions

The fundamental assumptions of Light-AHAKA are as follows:

- The IoT devices have a tamper-proof memory. Any attempt to disassemble the IoT device, the tamper-proof memory will be erased.
- The data is exchanged securely between the IoT micro-controller and the tamper-proof memory.
- The authentication server plays the role of the trusted party and has no constraints on its resources.
- The Acknowledgment message is a packet containing the sequence number of the expected packet with the ACK flag is set to '1'.
- The pre-shared key is used as the key for encryption/decryption and the HMAC function.
- All random numbers are generated from a pseudo-random number generator (PRNG).

5. Light-AHAKA

The (Light-AHAKA) is a lightweight authentication protocol, not based on the heavy traditional cryptographic functions. The (Light-AHAKA) is based on Lightweight Symmetric Key Cryptography, Lightweight Hash Function, Lightweight Hash-Based Message Authentication Code (HMAC) [26], and Exclusive-Or operation. The phases and the details of the (Light-AHAKA) will be detailed in the upcoming sub-sections.

5.1. Initialization

The initializations before starting the network operation are as follows:

- The Network Administrator (*NA*) assigns a unique identity (ID_{Ui}), password (PW_{Ui}), and a pre-shared key (PSK_{Ui}) for each user (U_i) stored safely in the client application.
- The (*NA*) loads (ID_{IoTi}), (PW_{IoTi}), and pre-shared key (PSK_{IoTi}) in a tamper-proof memory for each IoT device.

5.2. General

The (Light-AHAKA) Fig. 2 consists of five phases:

- First Phase: The protocol is initiated when the user sends an access request to the authentication server and supplies his/her credentials (step A).
- Second Phase: The authentication server initiates mutual authentication with the user based on the challenge-response mechanism (step B).
- Third Phase: The authentication server starts to initiate the mutual authentication with the intended (IoT_i) using the challenge-response mechanism (step C).
- Fourth Phase: The user and the (IoT_i) start the key agreement phase and end up with a fresh session key to encrypt the upcoming traffic (step D).

- Fifth Phase: The protocol terminates with updating the parameters (Pre-shared keys-Passwords-ID_{IoT}-ID_U) (step E).

5.3. Light-AHAKA Procedure

In the upcoming phases, the (Light-AHAKA) will be illustrated in detail.

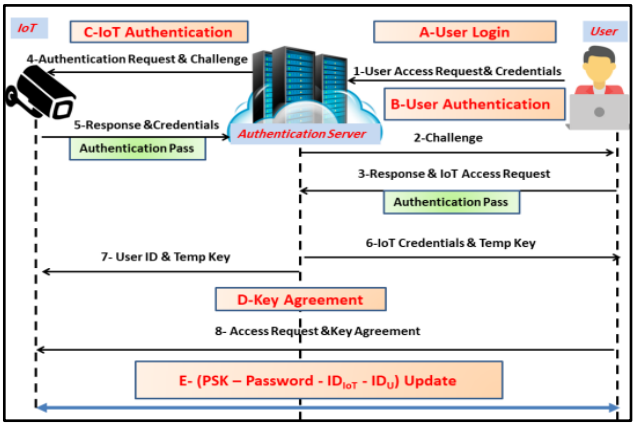

Fig. 2. Proposed protocol.

5.3.1. First phase: User login phase (Fig. 3)

The user sends an access request to the authentication server, contains the following:

$(ID_U \| H(PW_U) \| T_{U1}) \| (HMAC_{U1})$

1. The authentication server will do the following:
   - Checks if $(T_{U1} < \Delta T)$.
   - Calculates $(HMAC_{U1})$ and compares it with the received one.
   - Extracts (ID_U).
   - Searches in the database if (ID_U) exists or not.
   - Searches for the hash of the password and compares it with received $H(PW_U)$.

5.3.2. Second phase: User authentication phase (Fig. 3)

After the user successful login, the authentication server will start the authentication phase with the user according to the following steps:

1. The authentication server will do the following:
   - Generates a random number (R_{S1}).
   - Sends the following message encrypted with the pre-shared key (PSK_U) as a challenge for the user:

     $E_{PSK_U}(IDs \| R_{S1}) \| T_{S1} \| (HMAC_{S1})$.

2. The user receives the message and will do the following:
   - Checks if $(T_{S1} < \Delta T)$.
   - Calculates $(HMAC_{S1})$ and compares it with the received one.

- Decrypts the message using (PSK_U).
- Checks ID_S (received) =ID_S (stored).
- Extracts (R_{S1}).

3. From step (2) the user authenticates the server.
4. The user, after authenticating the server, will do the following:
   - Generates a random number (R_U).
   - Sends the following message as a response to the server:

   $E_{PSK_U}(R_{S1}||R_U||ID_{IoT}) ||T_{U2} || (HMAC_{U2})$.

5. The authentication server receives the message and will do the following:
   - Checks if ($T_{U2} < \Delta T$).
   - Calculates ($HMAC_{U2}$) and compares it with the received one.
   - Decrypts the message using (PSK_U).
   - Checks if R_{S1} (sent) = R_{S1} (received).
   - Stores R_U.
   - Searches for the (ID_{IoT}).
6. From step (5) the server authenticates the user.
7. The server sends the user acknowledgment message and the *HMAC* of the acknowledgment using (R_U) as a key for the *HMAC* as a response for the user:

   $ACK || HMAC_{S2}$

8. The user receives the messages and calculates the $HMAC_{s2}$ of the *ACK* using (R_U) as a key and compares the result with the received one.
9. In the next phase, the server will start mutual authentication with the intended IoT device.

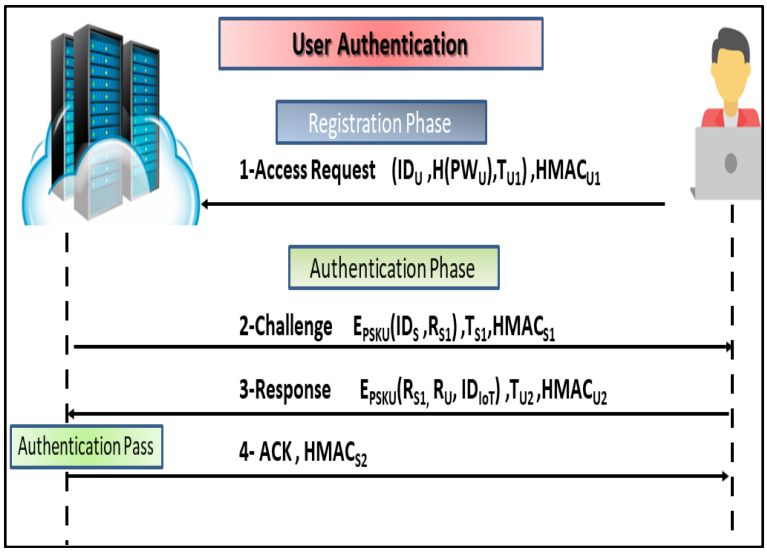

Fig. 3. User authentication.

5.3.3. Third phase: IoT authentication phase (Fig. 4)

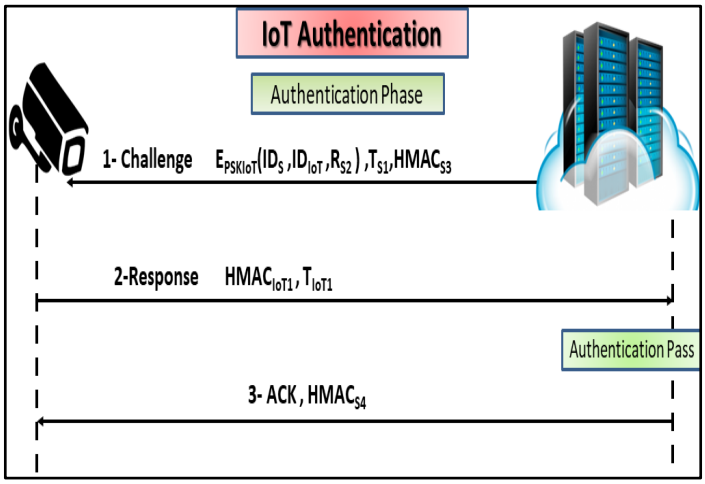

Fig. 4. IoT authentication.

1. The server generates a random number (R_{S2}).
2. The server sends a challenge message to the intended IoT device, contains the following:

   $E_{PSK_{IoT}}(ID_S || ID_{IoT} || R_{S2}) || T_{S1} || HMAC_{S3}$
3. The IoT receives the message and will do the following:
   - Checks if ($T_{S1} < \Delta T$).
   - Calculates ($HMAC_{S3}$) and compares it with the received one.
   - Decrypts the message using (PSK_{IoT}).
   - Checks if ID_S (received) = ID_S (Stored).
   - Extracts (R_{S2}).
4. From step (3) the IoT device authenticates the server.
5. The IoT device will do the following:
   - Calculates the response message for the challenge of the server:

     $HMAC_{IoT1}(T_{IoT1} \oplus R_{S2} \oplus PW_{IoT})$.
   - Calculates the nonce R_{IoT} as follows:

     $RIoT = Hash(RS2 \oplus IDS \oplus IDIoT)$
   - Sends the following response message to the *server*:

     ($HMACIoT1 || TIoT1$).
6. The server receives the message and will do the following:
   - Checks if ($T_{IoT1} < \Delta$ T).
   - Retrieves the *IoT* device password from the database.

- Calculates $HMAC_{IoT1}$ ($T_{IoT1} \oplus R_{S2} \oplus PW_{IoT}$) and compares it with the received one.
- Calculates R_{IoT} as follows:

$$RIoT = Hash\ (RS2 \oplus IDS \oplus IDIoT)$$

7. From step (6) the server authenticates the IoT device.
8. The server sends the IoT device acknowledgment message and the *HMAC* of the acknowledgment using ($RIoT$) as a key for the *HMAC* as a response for the IoT device:

$$ACK\ ||\ HMACS4$$

9. The IoT receives the messages and calculates the $HMAC_{S4}$ of the *ACK* using (R_{IoT}) as a key and compares the result with the received one.

5.3.4. Fourth Phase: Key Agreement Phase (Fig. 5)

In this phase the user and the IoT device will generate the session key according to the following steps:

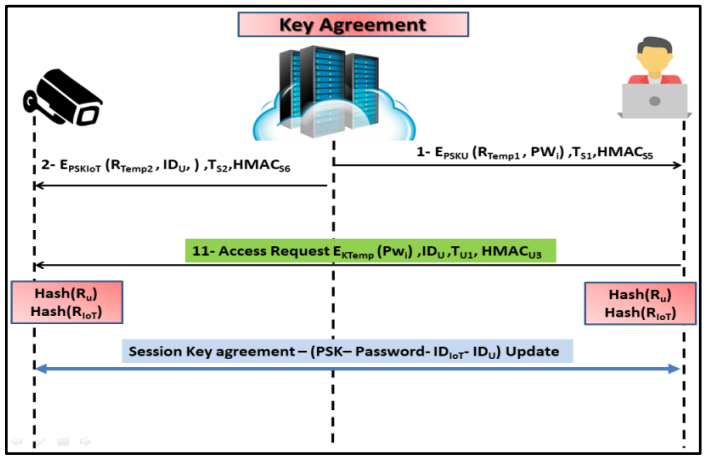

Fig. 5. Key agreement phase.

1. The server sends to the user the following data:

- $R_{Temp1} = (R_{IoT} \oplus ID_S)$
- IoT device password (PW_{IoT}), and R_{Temp1} are encrypted with the user pre-shared key (PSK_U) as follows:

$$E_{PSKU}\ (R_{Temp1}\ ||\ PW_{IoT})\ ||T_{S1}||HMAC_{S5}$$

2. The user receives the message and will do the following:

- Checks if ($T_{S1} < \Delta\ T$).
- Verify $HMAC_{S5}$.
- Calculates $R_{IoT} = R_{Temp1} \oplus ID_S$
- Stores R_{IoT} and PW_{IoT}

- Calculates $K_{Temp1} = R_{IoT} \oplus R_U$.

3. The server sends to the IoT device the following data:
   - $R_{Temp2}=R_U \oplus ID_S$
   - User ID (ID_U)
   - R_{Temp2} and user ID encrypted with IoT pre-shared key (PSK_{IoT}) as follows:

     $E_{PSKIoT}(R_{Temp2} || ID_U) ||T_{S2}||HMAC_{S6}$
4. The IoT device receives the message and will do the following:
   - Checks if ($T_{S2} < \Delta T$).
   - Verify $HMAC_{S6}$.
   - Calculates $R_U=R_{Temp2} \oplus ID_S$
   - Stores (R_U) and (ID_U).
   - Calculates $K_{Temp1} = R_{IoT} \oplus R_U$.
5. The user sends an access request to the IoT device as follows:

   $E_{K_{Temp1}}(PW_{IoT}) ||ID_U||T_{U1}|| MAC_{U3}$
6. The IoT device receives the message and will do the following:
   - Checks if ($T_{U1} < \Delta T$).
   - Verify $HMAC_{U3}$.
   - Checks if ID_U (received) = ID_U (stored).
   - Decrypts the message using (K_{Temp1}) calculated in step (4.e).
   - Extracts the password and checks that it is a valid password.
   - Sends acknowledgment to the user.
7. The server, the IoT device, and the user store two shift registers, the first shift register is a linear feedback shift register (L_1) connected to a primitive polynomial. The second is a non-linear feedback shift register (L_2) connected to a non-linear Boolean function. The output of the two registers is connected to a vectorial Dobbertain function (F) as a combiner function [27]:
   - $Output=F(L_1, L_2)$
8. Dobbertain function is an almost perfect non-linear function that achieves the following cryptographic properties:
   - High resistance to linear attacks.
   - High resistance to differential attacks.
9. The IoT device and the user will do the following for generating the session key:
   - IV_1=Hash (R_{IoT}).
   - IV_2=Hash (R_U).
   - IV_1 and IV_2 will be used to fill (L_1) and (L_2), respectively.
   - The session key (SK) will be calculated using the key agreement and parameters update module (Fig. 6) as follow

$SK = F\ [Shift\ (L1(i)),\ Shift\ (L2(j))]$

- As an example, let's assume that i is predetermined as the 20^{th} byte of R_U, and the decimal value of i is (126), so the L_1 will be shifted (126) shifts.

5.3.5. Fifth phase: Update phase (Fig. 6)

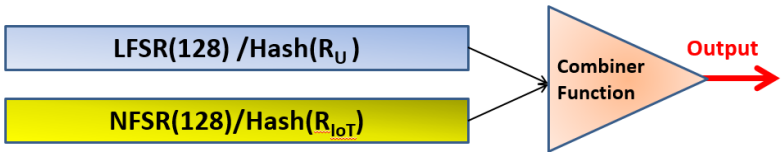

Fig. 6. Key agreement and parameters update module.

1. After finishing the mutual authentication and key agreement phases, the server, user, and the IoT device will update the following:
   - Pre-Shared Keys (PSK_U & PSK_{IoT}) according to the following:

     $\boldsymbol{PSK_U} = F\ [\ Shift\ (L_1(k),\ Shift\ L_2(l))]$

     $\boldsymbol{PSK_{IoT}} = F\ [Shift\ (L_1(m)),\ Shift\ L_2(n))]$
   - User and IoT device passwords according to the following:

     $\boldsymbol{PW_U'} = F\ [Shift\ (L_1(o)),\ Shift\ L_2(p))]$

     $\boldsymbol{PW_{IoT}'} = F\ [Shift\ (L_1(q)),\ Shift\ L_2(r))]$

     $\boldsymbol{PW_{U\text{-}New} = H\ (PW_U' \parallel PW_{U\text{-}Old})}$

     $\boldsymbol{PW_{IoT\text{-}New} = H\ (PW_{IoT}' \parallel PW_{IoT\text{-}Old})}$
   - User and IoT unique Identity according to the following:

     $\boldsymbol{ID_U'} = F\ [Shift\ (L_1(s)),\ Shift\ L_2(t))]$

     $\boldsymbol{ID_{IoT}'} = F\ [Shift\ (L_1(u)),\ Shift\ L_2(v))]$

     $\boldsymbol{ID_{U\text{-}New} = H\ (ID_U' \parallel ID_{U\text{-}Old})}$

     $\boldsymbol{ID_{IoT\text{-}New} = H\ (ID_{IoT}' \parallel ID_{IoT\text{-}Old})}$

6. Security Analysis

The security analysis of the (Light-AHAKA) will be discussed in the upcoming subsections. The (Light-AHAKA) is compared with [1, 19, 20, 22] in terms of security features and well-known attacks on IoT systems. The comparisons are demonstrated in Table 1.

6.1. Proposition 1: The (Light-AHAKA) provides mutual authentication

Proof: The (Light-AHAKA) fulfills one of the most security demands which is mutual authentication, this achievement is based on the challenge-response mechanism according to the following:

- The authentication server is responsible for authenticating both the user and the IoT device (Second phase - step5) (Third phase-step6).

- The user and the IoT device authenticate the authentication server (Second phase-step 2) (Third phase-step 3).
- Moreover, the user and the IoT device check the authenticity of each other and establish a fresh session key to encrypt the exchanged traffic (Fourth phase- step 5 and step 6).

6.2. Proposition 2: The (Light-AHAKA) achieves key agreement

Proof: The user (U_i) and the IoT device (IoT_i) generate a session key using the two random numbers generated in the authentication phase (R_{IoT} and R_U).

- These two random numbers are hashed and used to fill two shift registers.
- The shifting values differ from one session to another, depending on pre-determined indices selected from the random numbers (R_{IoT} and R_U).
- The value of the shifting index differs from one session to another, as the random numbers (R_{IoT} and R_U) are freshly generated in every session.

6.3. Proposition 3: The (Light-AHAKA) is immune against impersonation attack

Proof: In an impersonation attack, the attacker captures the authentication request message in an earlier session of a legitimate user and tries later to authenticate himself/herself impersonating a legitimate user. In the (Light-AHAKA), all the data of each authentication request change from one session to another (*Pre-shared keys -Identities- Passwords- Random Numbers*). Hence, the attacker will not get any use of the intercepted data from previous sessions.

6.4. Proposition 4: The (Light-AHAKA) is immune against privileged insider attack

Proof: Long and strong passwords are difficult to be memorized and as a common human behaviour, most users employ the same passwords for different applications [28]. However, if the network administrator or a privileged insider knows the passwords of a registered user (U_i), s/he may try illegitimately to impersonate (U_i). In the (Light-AHAKA), during the user login phase (step 1), the user sends the hash of his password to the *authentication server*, which is stored hashed not plain in the *server*. Accordingly, there is no feasible way for a privileged insider to know the plain password of a user.

6.5. Proposition 5: The (Light-AHAKA) is immune against man in the middle attack (MITM)

Proof: MITM attack [29], is the attack where an adversary illegitimately intercepts the sensitive data exchanged between the network parties. The adversary resends these data later trying to impersonate any party by pretending himself/herself a legitimate user.

- In the (Light-AHAKA) timestamps are used with each message, the time difference is checked at each end. If the time difference is out of the accepted range, the session terminates.

- Moreover, random numbers in the challenge-response mechanism are freshly generated and encrypted with the pre-shared keys, so the adversary cannot capture any message nor replay it.

6.6. Proposition 6: The (Light-AHAKA) is immune against DOS Attack

Proof: An attacker floods the network with an excessive number of consecutive fake requests targeting either the *server* or the *IoT* device, aiming to disrupt the regular service of the IoT network [30].

- From the *server*-side, the user access request could not be replayed as the user identity (ID_U), the user password ($H(PW_U)$) and the pre-shared key (PSK_U) change from one session to another, accordingly the *HMAC* value will change.
- From the IoT-side, the attacker has to possess one of the keys (PSK_{IoT} -K_{Temp}) to send an access request to the IoT device. These two keys change from one session to another.
- Additionally, the use of timestamps in the (Light-AHAKA) provides the ability to deny any request out of the accepted time range.

6.7. Proposition 7: The (Light-AHAKA) is immune against Replay Attack

Proof: In a replay attack, an attacker intercepts and acquires the data sent by the sender and resends it later to the destination as an original sender. To overcome this type of attack:

- Fresh random numbers and timestamps are used to generate unique login and reply to messages.
- Therefore, if the attacker replays a previous intercepted message, the system will reject the request as the timestamp is out of the time limit.

6.8. Proposition 8: The (Light-AHAKA) provides perfect forward secrecy

Proof: A protocol is considered to achieve perfect forward secrecy if compromising of the current session key will not lead to compromise past session keys. In the (Light-AHAKA), suppose the adversary has obtained the pre-shared keys of the two participants (User-IoT); the adversary still cannot get the session key of the two participants due to the following:

- The session key depends on two fresh encrypted random numbers, fed as an input to a one-way hash function and used as initial values for two shift registers.
- Moreover, the two shift registers are shifted differently in every session depending on random indices, then their output is fed to a combiner function.
- For every session, the two shift registers will be filled with the hash of two random numbers. After each successful session, the pre-shared keys will be updated depending on two different random indices. Therefore, the (Light-AHAKA) provides perfect forward secrecy.

6.9. Proposition 9: The (Light-AHAKA) is immune against offline guessing attack

Proof: In the (Light-AHAKA):

- User and IoT devices' passwords are sent hashed. For the offline guessing attack, the attacker has to find a collision in the used hash function or to use the dictionary attack to crack the password.
- Moreover, the passwords are updated after each session, which is considered a one-time password (*OTP*).

6.10. Proposition 10: The (Light-AHAKA) achieves data integrity

Proof: Data integrity implies that a message receiver has to be sure that the message has not been tampered with during the transmission.

- The (Light-AHAKA) adopts an *HMAC* function with each transmitted message to achieve data integrity.
- In the authentication phase, if an attacker tries to tamper with transmitted messages, the attacker needs to figure out the pre-shared keys (PSK_U - PSK_{IoT}) to produce a valid *HMAC*.
- Also, the attacker needs to learn the fresh nonce (R_U-R_{IoT}) to forge the *HMAC* in the stages of the pre-key agreement phase.
- All the aforementioned secret values are updated every session so the brute force attack on these values is infeasible.

6.11. Proposition 11: The (Light-AHAKA) is immune against parallel session attack

Proof: In this attack, the attacker keeps tracking and recording the ongoing transmission of messages between the legitimate network participants. Immediately, the attacker will start a parallel session by retransmitting the recorded messages to the server within the accepted time limit [31].

- To impersonate a legitimate user (Ui), the attacker retransmits the message (ID_U ||H (PW_U) ||T_{U1}) || ($HMAC_{U1}$) within the correct time limit. The attacker must guess:

  H (PW_U).

  $HMAC_{U1}$.

- Guessing the two previous parameters to initiate a new session within the time frame is extremely difficult.

6.12. Proposition 12: The (Light-AHAKA) is immune against session key discloser attack

Proof: The security of the session key in the (Light-AHAKA) relies on the following:

- The secret random numbers (R_U) and (R_{IoT}).
- The hardness of the hash function used to hash the two random numbers.

- The primitive polynomial and the Boolean Function used in connecting the (*LFSR*) and (*NFSR*) taps, respectively.
- The Dobbertain Function.
- The decimal values of the two indices, i^{th} and j^{th} bytes of (R_S) and (R_U) respectively, used to determine the number of shifting cycles of the two shift registers. The decimal values of the two indices differ from one session to another depending on the values of (R_S) and (R_U).

6.13. Proposition 13: The (Light-AHAKA) achieves user anonymity and untrace ability

***Proof*:** To achieve user anonymity, the (Light-AHAKA) should prevent the attacker from neither knowing the identity of the participants nor tracking them.

1. For identity protection:
   - The identity of (User-IoT) changes from one session to another, so the eavesdropper cannot find identical identities in consecutive sessions.
2. For user untrace ability:
   - All the parameters transmitted on the channel are changing with the fresh random numbers from one session to another.

Therefore, (Light-AHAKA) achieves user anonymity and user untrace ability.

Table 1. Comparison of security features.

Security Feature	(Light-AHAKA)	[19]	[20]	[22]	[1]
Mutual Authentication	✓	✓	✓	✓	✓
Key Agreement	✓	✓	✓	✓	✓
Impersonation Attack	✓	✗	✗	✓	✓
Privileged Insider Attack	✓	✗	✗	-	✓
Man in The Middle Attack	✓	✓	✓	✓	✓
DOS Attack	✓	✗	✗	✗	✗
Replay Attack	✓	✗	✓	✗	✓
Perfect Forward Secrecy	✓	✓	✓	✓	✗
Offline Guessing Attack	✓	✓	✗	-	✗
Data Integrity	✓	✗	✗	✓	✗
Parallel Session Attack	✓	✗	✓	✗	✗
Session Key Discloser Attack	✓	✓	✗	✓	✓
User Anonymity and User Untraceability	✓	✗	✗	✗	✗

7. Conclusion and Future Work

In this paper, a lightweight authenticated key agreement protocol for IoT in cloud computing is proposed based on the challenge-response mechanism, using symmetric key encryption/decryption, hash function, and hash-based message authentication code. The (Light-AHAKA) generates a new fresh session key to encrypt the traffic; it also updates the identities, passwords, and the pre-shared keys of the network participants every session. The (Light-AHAKA) provides mutual authentication, and perfect forward secrecy. Security analysis is performed to

ensure that the (Light-AHAKA) is secure against known attacks. In the future, we plan to work on formal security analysis using the strand space model to prove the correctness of the (Light-AHAKA) and to ensure that it is not prone to any substantial attacks. Moreover, we will work on the practical implementation of the (Light-AHAKA). A test-bed network will be constructed to test and evaluate the communication cost, computational cost, execution time, and storage cost.

Nomenclatures

E (.)	Symmetric Lightweight Encryption
F	Dobbertain Function
H (.)	A Lightweight Collision Free One-Way Hash Function
ID_{IoT}	IoT Identity
ID_S	Authentication Server Identity
ID_U	User Identity
i, k, m, p, r, t, v	Indices selected from R_U – ex: i= decimal value of i^{th} byte of R_U
j, l, n, q, s, u	Indices selected from R_{IoT} – ex: j= decimal value of j^{th} byte of R_{IoT}
PW_{IoT}, PW_U	The password of the IoT device, User respectively
PSK_{IoT}	The pre-shared key between the user and the IoT device
PSK_U	The pre-shared key between the user and the server
R_S, R_U, R_{IoT}	Random numbers of Sever, User, IoT devices respectively
T_S, T_U, T_{IoT}	Timestamps of Sever, User, IoT device respectively
ΔT	Time range allowed for delay
$\|\|$	Concatenation
$\oplus$	XoR Operation

Abbreviations

HMAC	Hash Message authentication Code
LFSR	Linear Feed- Back Shift Register
NFSR	Non-Linear Feed-Back Shift Register
SK	Session Key

References

1. Amin, R.; Kumar, N.; Biswas, G.; Iqbal, R.; and Chang, V. (2018). A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. *Future Generation Computer Systems*, 78, 1005-1019.
2. Zhang, S.; Wu, Q.; Xu, S.; and Li, G.Y. (2016). Fundamental green tradeoffs: Progresses, challenges, and impacts on 5G networks. *IEEE Communications Surveys Tutorials*, 19(1), 33-56.
3. Osseiran, A.; Boccardi, F.; Braun, V.; Kusume, K.; Marsch, P.; Maternia, M.; Queseth, O.; Schellmann, M.; Schotten, H.; and Taoka, H. (2014). Scenarios for 5G mobile and wireless communications: the vision of the METIS project. *IEEE Communications Magazine*, 52(5), 26-35.
4. Hossain, M.; Hasan, R.; and Skjellum, A. (2017). Securing the internet of things: a meta-study of challenges, approaches, and open problems. *IEEE* 37*th*

International Conference on Distributed Computing Systems Workshops (*ICDCSW*). Atlanta, USA, 220-225.

5. Atzori, L.; Iera, A.; and Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15), 2787-2805.
6. Gubbi, J.; Buyya, R.; Marusic, S.; and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
7. Medaglia, C.M.; and Serbanati, A. (2010). *The internet of things*. Chapter: An overview of privacy and security issues in the internet of things. Springer, 389-395.
8. Ning, H. (2013). A security framework for the internet of things based on public key infrastructure. *Advanced Materials Research*, 671-674, 3223-3226.
9. Pranata, H.; Athauda, R.I.; and Skinner, G. (2012). Securing and governing access in ad-hoc networks of internet of things. *Proceedings of the IASTED International Conference on Engineering and Applied Science*, *EAS*. Colombo, Sri Lanka, 84-90.
10. Kothmayr, T.; Schmitt, C.; Hu, W.; Brünig, M.; and Carle, G. (2013). DTLS based security and two-way authentication for the Internet of Things. *Ad Hoc Networks*, 11(8), 2710-2723.
11. Kothmayr, T.; Schmitt, C.; Hu, W.; Brünig, M.; and Carle, G. (2012). A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. 37*th Annual IEEE Conference on Local Computer Networks-Workshops*. Clearwater, USA, 956-963.
12. Qi, M.; Chen, J.; and Chen, Y. (2019). A secure authentication with key agreement scheme using ECC for satellite communication systems. *International Journal of Satellite Communications Networking*, 37(3), 234-244.
13. Alizai, Z.A.; Tareen, N.F.; and Jadoon, I. (2018). Improved IoT device authentication scheme using device capability and digital signatures. *International Conference on Applied and Engineering Mathematics* (*ICAEM*). Taxila, Pakistan, 1-5.
14. Lu, J.Z.; Chen, T.; Zhou, J.; Yang, J.; and Jiang, J. (2013). An enhanced biometrics-based remote user authentication scheme using smart cards. 6*th International Congress on Image and Signal Processing* (*CISP*). Hangzhou, China, 1643-1648.
15. Yu, Y.; Hu, L.; and Chu, J. (2020). A secure authentication and key agreement scheme for IoT-based cloud computing environment. *Symmetry*, 12(1), 150.
16. He, D.; Zeadally, S.; Kumar, N.; and Wu, W. (2016). Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Transactions on Information Forensics Security*, 11(9), 2052-2064.
17. Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; and Chiu, W. (2019). Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Generation Computer Systems*, 91, 244-251.
18. Sahoo, S.; Sahoo, S.S.; Maiti, P.; Sahoo, B.; and Turuk, A.K. (2019). A lightweight authentication scheme for cloud-centric IoT applications. 6*th International Conference on Signal Processing and Integrated Networks* (*SPIN*). Noida, India, 1024-1029.

19. Shah, T.; and Venkatesan, S. (2018). Authentication of IoT device and IoT server using secure vaults. 17*th IEEE International Conference On Trust*, *Security And Privacy In Computing And Communications*/12*th IEEE International Conference On Big Data Science And Engineering* (*TrustCom*/*BigDataSE*). New York, USA, 819-824.
20. Esfahani, A.; Mantas, G.; Matischek, R.; Saghezchi, F.B.; Rodriguez, J.; Bicaku, A.; Maksuti, S.; Tauber, M.G.; Schmittner, C.; and Bastos, J. (2017). A lightweight authentication mechanism for M2M communications in industrial IoT environment. *IEEE Internet of Things Journal*, 6(1), 288-296.
21. Kaur, K.; Garg, S.; Kaddoum, G.; Guizani, M.; and Jayakody, D.N.K. (2019). A lightweight and privacy-preserving authentication protocol for mobile edge computing. *IEEE Global Communications Conference* (*GLOBECOM*). Waikoloa, USA, 1-6.
22. Rabiah, A.; Ramakrishnan, K.; Liri, E.; and Kar, K. (2018). A lightweight authentication and key exchange protocol for IoT. *Proceedings Workshop Decentralized IoT Secure Standards* (*DISS*). San Diego, USA, 1-6.
23. Chuang, M.C.; and Chen, M.C. (2014). An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Systems with Applications*, 41(4), 1411-1418.
24. Xue, K.; Hong, P.; and Ma, C. (2014). A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *Journal of Computer System Sciences*, 80(1), 195-206.
25. Wang, P.; Li, B.; Shi, H.; Shen, Y.; and Wang, D. (2019). Revisiting anonymous two-factor authentication schemes for IoT-enabled devices in cloud computing environments. *Security Communication Networks*, 2019, 1-13.
26. McKay, K.; Bassham, L.; Sönmez Turan, M.; and Mouha, N. (2016). *Report on lightweight cryptography*. National Institute of Standards and Technology, Gaithersburg, USA.
27. Dobbertin, H.J.I.; and Computation. (1999). Almost perfect nonlinear power functions on GF (2n): the Niho case. *Information and Computation*, 151(1-2), 57-72.
28. Khan, M.K.; and Alghathbar, K. (2010). Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. *Sensors*, 10(3), 2450-2459.
29. Baig, A.F.; ul Hassan, K.M.; Ghani, A.; Chaudhry, S.A.; Khan, I.; and Ashraf, M.U. (2018). A lightweight and secure two factor anonymous authentication protocol for Global Mobility Networks. *PloS one*, 13(4), 1-21.
30. Anirudh, M.; Thileeban, S.A.; and Nallathambi, D.J. (2017). Use of honeypots for mitigating DoS attacks targeted on IoT networks. *International Conference on Computer*, *Communication and Signal Processing* (*ICCCSP*). Chennai, India, 1-4.
31. Shieh, W.G.; and Wang, M.T. (2008). A new parallel session attack to Khan-Zhang's authentication scheme. 3*rd International Conference on Innovative Computing Information and Control*. Dalian, China, 154-154.