Lightweight Anonymity-Preserving Authentication and Key Agreement Protocol for the Internet of Things Environment

Ahmed Mohammed Ibrahim Alkuhlani[1(✉)] and S. B. Thorat[2]

[1] S.R.T.M University, Nanded-Waghala 431606, India
Alkohlanyl@gmail.com
[2] I.T.M College, Nanded-Waghala 431602, India
suryakant_thorat@yahoo.com

Abstract. Internet of things (IoT) creates a world-wide network of interconnected objects or things that will have an active role in the Future Internet (FI). Such things will be readable, recognizable, locatable, addressable, and/or controllable via the Internet; in order for the IoT to expand there should be a trust in the IoT security infrastructure. The number of applications and services expected to be numerous so in order to access these applications and services a secure and robust authentication protocol is required. In this paper, we propose a robust and lightweight mutual authentication and key agreement protocol for the IoT environment. We have used lightweight computational cryptographic functions to maintain low computational, memory and energy consumption. The security analysis and performance evaluation prove the protocol is lightweight and resist most of known security related attacks Moreover, formal security verification was conducted using AVISPA tool. The result shows that the proposed protocol is secure and safe.

Keywords: IoT · IoT authentication · Biometric-based authentication
Constraints network · Lightweight authentication

1 Introduction

Nowadays, more IoT applications have been implemented, such as smart home systems [1], healthcare systems, connected cars, surveillance devices, environmental monitoring, and smart wearable devices [2–4]. Huge amounts of sensitive and personal information are exchanged.

It is very important to define how the IoT things could efficiently and securely communicate and exchange information among themselves and with remote servers. Security and privacy are a key challenge to IoT [5].

Things in IoT have limited computational capability, limited energy, and small memory. They communicate using low rate and low power wireless technologies such as IEEE 802.15.4 BLE ZigBee etc. [6, 7] meanwhile; existing traditional security techniques require a considerable amount of energy for processing. Therefore, we require efficient and robust security mechanisms that provide a similar level of security of the existing traditional techniques with the limited resources of the IoT devices. In IoT, we require authentication and key agreement techniques that allow two remote

G. P. Venkataramani et al. (Eds.): ICIIT 2017, CCIS 808, pp. 108–125, 2018.
https://doi.org/10.1007/978-981-10-7635-0_9

entities to mutually authenticate and negotiate secret keys that are used to protect the sensor data against various types of active and passive attacks [8].

Therefore, in this paper we proposed a secure and lightweight mutual authentication and key agreement protocol for IoT environment. We have used lightweight computational cryptographic functions such as hash function and Xor operator which is suitable to use on constrained platforms such as IoT and wireless sensor network (WSN) [9]. Also deep Security analysis and performance evaluation are conducted to prove the protocol is lightweight and robust.

The rest of the paper is organized into six sections; Sect. 2 presents a literature review of related schemes. In Sect. 3 preliminaries related to IoT authentication are discussed. In Sect. 4 we present our proposed protocol. In Sect. 5 we provide security and performance analysis. In Sect. 6 the formal security analysis using AVISPA software is conducted, and the paper is wrapped up with the conclusion in Sect. 7.

2 Literature Review

In 2012 Das et al. [10] proposed a new authentication scheme for hierarchical WSNs that support the feature of dynamic node. At the same year Liu et al.'s [11] proposed user authentication and access control scheme for IoT. The scheme uses RBAC access control. In 2013, Turkanović and Hölbl [12] and Xue et al. [13] claimed that the protocol of [10] is impractical, and proposed enhanced protocols to overcome its drawbacks. Li et al. [14] proved that the scheme of Xue et al. is prone to problems such stolen-verifier attack, off-line password guessing attack. In 2014, Turkanović et al. [15] proposed a lightweight authentication protocol for heterogeneous WSN based on the notion of IOT. The scheme proved to be computationally lightweight and consumes less memory and energy. At the same year, Ndibanje et al. [16] found some security weakness in [11] scheme; therefore, they propose an enhanced protocol that offers user anonymity and mutual authentication. In 2015, He et al. [17] showed that the Xue et al. protocol is susceptible to off-line password guessing attacks, and user and sensor node impersonation attacks. Amin and Biswas [18] claimed that the scheme of Turkanović is not efficient in terms of energy consumption, and proposed a user authentication and key agreement scheme in multi-gateway based on WSN. In 2016, Farash et al. [19] found some security weaknesses in Turkanović et al. [15] such as off-line password guessing attacks, and man-in-the-middle attacks. Then, they proposed an enhanced user authentication and key agreement scheme for heterogeneous WSN for the IoT concept. In the same year, Amin et al. [20] revealed that the scheme of Farash et al. is insecure and susceptible to stolen-smartcard attacks, off-line password-guessing attacks, user-impersonation attacks, and fails to preserve user-anonymity. Afterwards, Arasteh et al. [21] claimed that the scheme proposed by Amin et al. in [20] has security weaknesses and is prone to Replay attacks and DoS attacks, and proposed an enhanced protocol to overcome these drawbacks. Recently in 2017, Dhillon and Kalra [22] proposed an enhanced three-factor biometric authentication protocol for IoT network based on Turkanović et al. scheme. Jiang et al. [23] proposed Lightweight Three-factor Authentication and Key Agreement Protocol for Internet-integrated WSNs based on the idea of public key primitive Rabin cryptosystem.

3 Preliminaries

3.1 One-Way Hash and Bio-Hash Function

Hash function takes arbitrary input data and returns a string with a fixed size, which is referred to as a hash value or (a message digest). One of the important properties of one-way hash function is that it is very sensitive: any small change to the input data results in a totally different output hash value. Biometric is not always a constant value; it may change with time and environment. So, the general one-way hash function is not the proper choice for hashing biometric. To resolve this issue, researchers in [25, 26] have suggested Bio-hash function which proved its accuracy and flexibility with biometrics.

Bio-Hash function refers to a special type of one-way hash function that can be used to hash different types of biometrics such as (Fingerprint, iris, retina, and voice). The input data of biometric may vary a little bit, but the result hash value of Bio-hash function remains the same. In the contrary, if the variation is significant, the output becomes different.

3.2 Network Model

IoT is the concept of connecting smart devices to the global network (the Internet) which allows users to access the IoT services remotely. As depicted in Fig. 1 through an application on the remote user smartphone the user can directly connects to a specific IoT device inside the network (smart home). In order to lower the processing burden for the sensor node, the protocol uses the gateway node as a mediator for the authentication process [24].

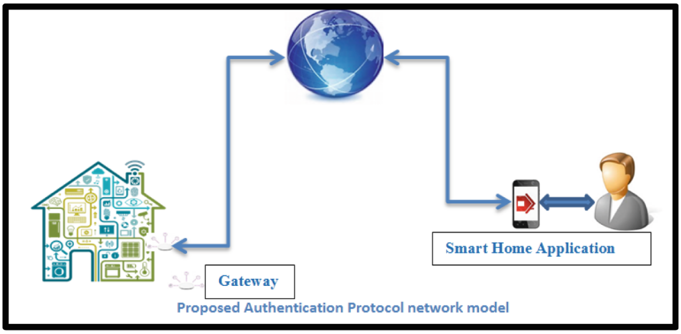


Fig. 1. Network model

4 Proposed Authentication and Key Agreement Protocol

In the following sections, we propose an authentication and key agreement protocol for IoT network. Our proposed scheme has a pre-deployment phase (system setup phase), registration phase, login phase, authentication and key agreement phase, and password change/update phase. In Table 1, there is a brief description of notations used within the protocol.

Table 1. List of notations used throughout the protocol

Symbol	Description
Ui	User
GWN	Gateway
Nj	IOT node
IDi	Unique identity for each user Ui
PWi	Password of the user Ui
Bi	Biometric key of fngi, where Bi = BK(H(fngi))
Fngi	Biometric template of user Ui
Xgui	Unique Shared secret key between each Ui and GWN
SP	Smartphone
Ksg	Shared secret key between Nj and GWN
Xgn	Master secret key and GWN secret password
N	High entropy Nonce generated by GWN to mask its secret Key
S	Used to masked Ui identity during communication
Y	Used to masked Nj identity during communication
IDj	Unique identity for each node Nj
CRj	Password of IoT node Nj
Ki	Random Nonce generated by Ui to construct the session key
Kj	Random Nonce generated by Nj to construct the session key
SK	Session key to encrypt communication between Ui and Nj
Ts1, Ts2, T1–T4	Timestamp used throughout the Scheme
ΔT, Tc	Time Range of allowed transmission delay, Current time
h(.), H(.)	One-way hash function, Bio-hashing function
$\|$, $\oplus$	Concatenate operation, X-OR operation
gi, fi, ei	Values used to protect the identity and password of the user

4.1 System Setup Phase

This is the pre-deployment phase in which each embedded device/sensor of IoT network has to be configured with certain parameters prior to authentication. This phase is executed by the system administrator (SA) in offline mode as follows:

- **Step 1.** SA assigns a master secret key (Xgn) for the gateway (GWN), the master secret key Xgn is known only to SA and the GWN.
- **Step 2.** SA assigns unique identity IDj for each IoT node Nj in the IoT network and also computes the password CRj CRj = h(IDj||Xgn). Therefore, each node will have a unique secret key CRj.
- **Step 3.** SA chooses a random secret number Ksg that is shared between the GWN and the Nj.
- **Step 4.** SA embeds (IDj, CRj, Ksg) into node's tamper-proof memory and (Xgn, Ksg, IDj) to GWN memory.

4.2 Registration Phase

Registration phase is divided into two phases. The first one is for the registration of nodes of IoT network, and the second is for the registration of the outside/remote users.

IoT Node Registration Phase. This phase is performed between the Nj and the GWN. Details of this phase are depicted in Fig. 2b.

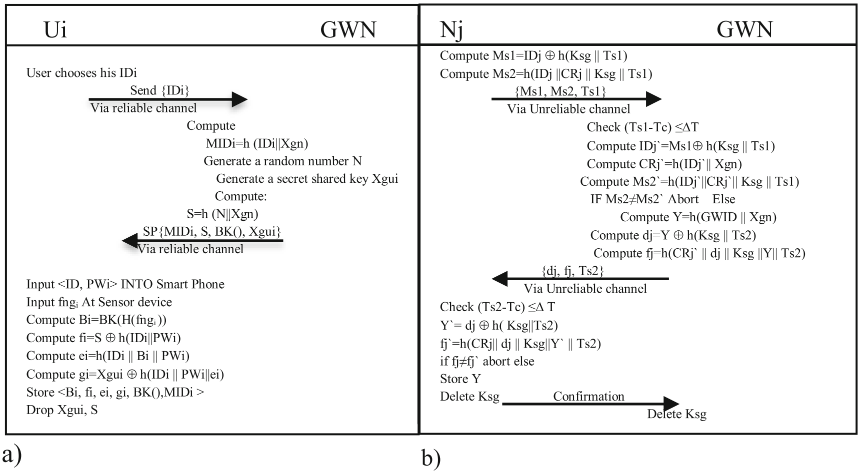


Fig. 2. Registration phase (a) user (b) IoT node

- **Step 1.** In order to provide ID anonymity, IOT node Nj computes $Ms1 = IDj \oplus h(Ksg||Ts1)$ and for message verification computes $Ms2 = h(IDj||CRj||Ksg||Ts1)$. In which T1 is a fresh timestamp and sends to GWN (Ms1, Ms2, Ts1) through an unreliable channel.
- **Step 2.** Upon the reception of the message from the Nj the GWN, first, verifies whether or not the time received T is within the allowed time span to avoid replay attack $(Ts1 - Tc) < \Delta T$. If it is not within the allowed time, the GWN refuses to accept the Nj; otherwise, if the verification holds, GWN computes $IDj` = Ms1 \oplus h(Ksg||Ts1)$, $CRj` = h(IDj`||Xgn)$, $Ms2` = h(IDj`||CRj`||Ksg||Ts1)$, and checks whether $Ms2` \neq Ms2$ then the Nj is not legitimate and session is aborted. If not, the GWN authenticates the Nj. The GWN continues and computes $Y = h(GWID||Xgn)$, $dj = Y \oplus h(Ksg||Ts2)$ and $fj = h(Y||dj||Ksg||Ts2)$ then the GWN sends to Nj{dj, fj, Ts2} through unreliable channel. When Nj received the message from the GWN, it first verifies the time for replay attacks if the time T is within the allowed time span. Then it continues with the registration process, or else, it rejects the message. If the verification holds, the Nj computes $Y` = dj \oplus h(Ksg||Ts2)$ and very fies if $fj` = fj$ holds then the GWN is legitimate and then the Nj stores Y and deletes the shared key Ksg from the device memory.
- **Step 3.** In the last step, the Nj sends a confirmation message to the GWN and deletes the shared key Ksg and IDj from the GWN memory.

User Registration Phase. The second phase of registration is done with the user Ui. At the end of this phase, the user will be authorized and registered with the GWN. Details of this phase are depicted in Fig. 2a.

- **Step 1.** Ui sends his identity IDi to the GWN via a reliable/secure channel. Upon the reception of message sent from the Ui, the GWN computes masked IDi with the GWN master secret key Xgn, $MIDi = h(IDi \| Xgn)$; then, the GWN generates a secret random key Xgui that will be shared between the Ui and the GWN for further secure communication.

  GWN also generates a random number N with high entropy, then computes $S = h(Xgn \| N)$ and customizes the user's smartphone (SP) with {Xgui, BK(), S, MIDi} where BK() refers to the biometric key generation and extraction function.
- **Step 2.** Upon the reception of the message sent from the GWN, the Ui inputs his IDi and credentials password PWi and fingerprint fngi using smartphone sensor device. Using the BK(), the user computes $Bi = BK(H(fngi))$, then computes $fi = S \oplus h(IDi \| PWi)$, $ei = h(IDi \| Bi \| PWi)$, and $gi = Xgui \oplus h(IDi \| PWi \| ei)$.
- **Step 3.** Finally user stores {Bi, MIDi, fi, ei, gi, BK()}in the SP and deletes Xgui and S from the SP memory. Note that Xgui is the secret key shared between the Ui and the GWN, and the value CRi needs it to be computed at the login phase $CRi = h(PWi \| Xgui)$. Hence, to be safe from smartphone breach/stolen attacks and offline password guessing attacks, Xgui is deleted from the SP and will be recomputed at login phase. Furthermore, the value S is used to preserve the identity anonymity of the Ui when the message is exchanged in the authentication phase.

4.3 Login Phase

This Phase is done between the Ui and the Nj. After the registration phase is completed, the user logs in to initiate a request to access the desired device in the IoT network. Our proposed protocol uses the user fingerprint, username and password for login. A detailed description of this phase is as follows:

- **Step 1.** Ui opens the IoT application (smart home App) on his smartphone (SP) then inputs his fingerprint (fngi) on the smartphone device sensor to compute $Bi` = BK(H(fngi))$, then compares the calculated Bi` with the stored Bi if ($Bi` \neq Bi$). Then the user is rejected. Otherwise, the user is asked to enter his identity IDi and the password PWi. Afterwards, Ui Computes $ei` = h(IDi \| PWi \| Bi)$ and checks whether ($ei` \neq ei$). If so, the session is aborted as the user is not a legitimate user. Once the user is proved to be legitimate and his fngi, IDi and PWi are correct, user proceeds to step 2.
- **Step 2.** Ui Computes $Xgui = gi \oplus h(IDi \| PWi \| ei`)$, $CRi = h(PWi \| Xgui`)$ and $S = fi \oplus h(IDi \| PWi)$, the Ui generates a random nonce Ki which is the user part of the session key to be used to encrypt the data. Also generates a fresh timestamp T1 to be used to avoid a reply attack. After generating Ki and T1, the user starts to prepare the authentication messages that are to be sent to the IoT node Nj that Ui wants to access. To provide identity anonymity and avoid user traceability attack for the Ui's IDi, the Ui computes $M1 = IDi \oplus h(S \| T1)$. The identity of the user Ui is kept secret. S is a highly secure value; it is a combination of the GWN master secret

key Xgn and a high entropy random number N which makes it difficult for an attacker to break. In M2 = CRi ⊕ h (MIDi||Xgui||T1), the user CRi is safely protected from man in the middle attack and replay attack by using the shared password Xgui and the fresh time T1. Note that these messages are sent through an unreliable channel and the one-way hash function h maintains the integrity of these messages, and any tiny change to the hash value is discovered. The third message M3 = Ki ⊕ h(CRi||Xgui||T1) carries the Ui part of the session key Ki, and eventually M4 = h(Ki||CRi||MIDi||Xgui||T1) verifies that the previously sent values M1, M2, M3 are not changed, modified, or deleted by any attacker.

- **Step 3.** Ui chooses the IoT node Nj he wishes to access and send {M1, M2, M3, M4, T1} to it via an unreliable channel.

4.4 Authentication Phase

After the deployment of the IoT network and registration of both users and IoT nodes, the user logs in and chooses the desired node he wants to access. The authentication phase comes to mutually authenticate a user with chosen node and the gateway. Moreover, manages a secure key agreement by securely exchanging key parts of the session key between the Ui and Nj. authentication phase is completed in 4 messages handshakes, a user who wants to access data from IoT network can directly access a specific IoT device without the need to access the gateway first. The gateway works as an authenticator for both the IoT node and user. Details of this phase is depicted in Fig. 3. Authentication steps are as follows:

- **Step 1.** Upon the reception of the login message {M1, M2, M3, M4, T1} form Ui, Nj verifies the time |T1 − Tc| < ΔT. If T is within the allowed time span, then Nj proceeds with the authentication. Otherwise, the user is considered illegitimate and the session is aborted.
- **Step 2.** After the verification of the freshness of T1 passes, Nj computes MIDj = IDj ⊕ h(Y||T2). The identity of the node IDj is masked with the value Y = h (GWID||Xgn), and fresh time stamp T2 to avoid any replay attack and to preserve the identity anonymity of Nj. Then Next Nj generates a random nonce Kj which is the Nj part of the session key to be used to encrypt the data in further communication with the Ui.
- **Step 3.** Nj continues to prepare the necessary values for authentication, and computes M5 = Kj ⊕ h(CRj||T1||T2), and the verification message M6 = h(CRj||IDj|| T1||T2||Kj).
- **Step 4.** As the node Nj is a constrained device, it delegates the authentication of the Ui to the GWN by sending the message received from Ui{M1, M2, M3, M4, T1}, along with its own message {MIDj, M5, M6, T2}.
- **Step 5.** After receiving the message sent from the Nj, the GWN checks the time freshness of the received messages |T2 − Tc| < ΔT. If the time difference between the sent time T2 and the current time of the GWN Tc is within the allowed time span, the GWN continues with the Authentication of the Nj, or else it aborts the session and sends a rejection message to the Nj.

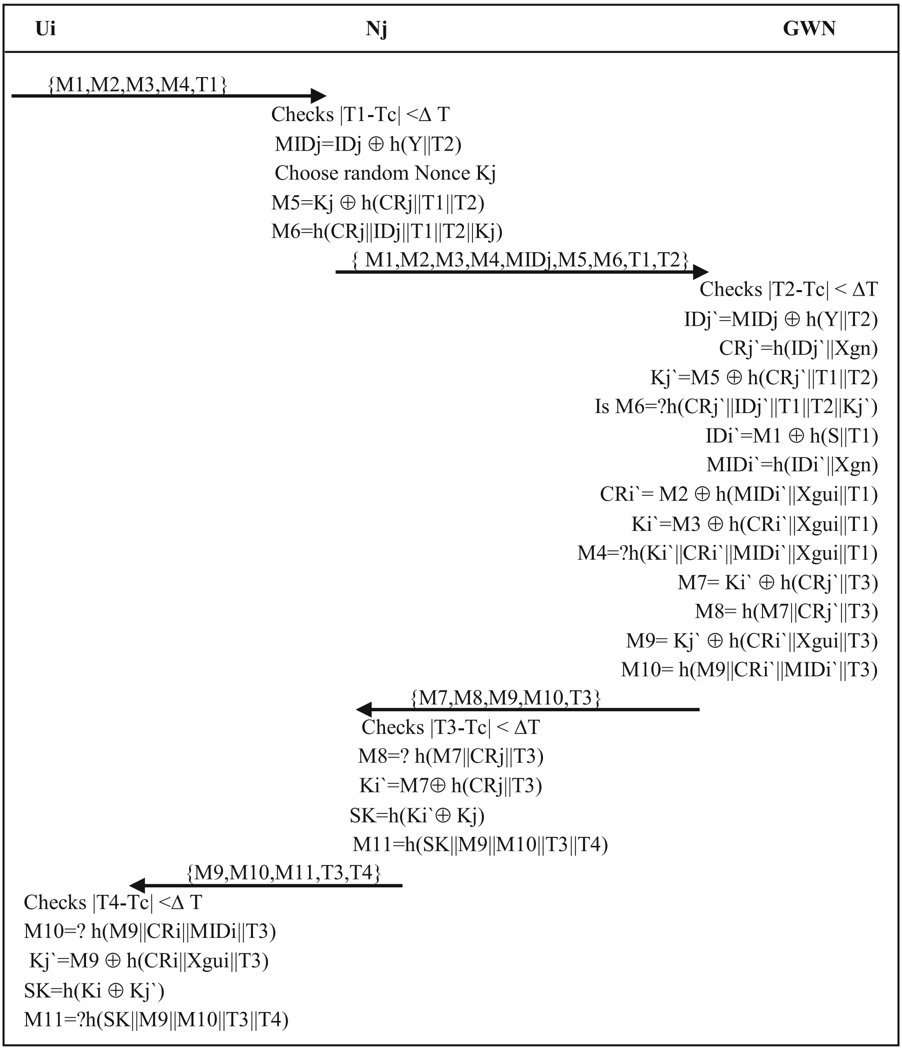


Fig. 3. Authentication and key agreement phase of the proposed protocol

- **Step 6**. After the time verification passes, the GWN first checks the legitimacy of Nj. The GWN computes the $IDj` = MIDj \oplus h(Y\|T2)$ using the secret value which was previously stored by the GWN in the Nj memory. It should be noted that only GWN can compute the value of Y user, as it is the only part that has the hashed value of Y. Using the newly computed IDj`, the GWN computes $CRj` = h(IDj`\| Xgn)$.
- **Step 7.** Using the newly computed IDj` and CRj`, the GWN extracts the Nj session key part by computing $Kj` = M5 \oplus h(CRj`\|T1\|T2)$ T1 and T2 are used to avoid the replay attack.

- **Step 8.** The GWN checks if the received value M6 = is equal to the GWN version of $M6'(h(CRj'\|IDj'\|T1\|T2\|Kj'))$, if so, then the Nj is authenticated and considered legitimate. Therefore, GWN proceeds to check the authenticity of Ui; otherwise, GWN rejects Nj and aborts any further transaction.
- **Step 9.** After the GWN verifies the legitimacy of the Nj, it has to check the authenticity of the Ui. The GWN extracts the identity of the Ui by computing $IDi' = M1 \oplus h(S\|T1)$. The identity of the user Ui is kept secret to maintain the ID anonymity and avoid user traceability attacks. S is a highly secure value; it is a combination of the GWN master secret key and the high entropy random number N. Using the newly computed IDi, the GWN computes the masked identity of the Ui $MIDi' = h(IDi'\|Xgn)$ using the GWN secret key Xgn. It should be noted that S can be computed only by the GWN and stored in a hash format in the Ui's Smartphone memory during registration.
- **Step 10.** Using the newly computed MIDi' and the GWN-Ui shared password Xgui, the GWN extract $CRi' = M2 \oplus h(MIDi'\|Xgui\|T1)$. Then using newly computed CRi' and the shared password Xgui, GWN extracts the Ui session key part $Ki = M3 \oplus h(CRi'\|Xgui\|T1)$.
- **Step 11.** The GWN checks if its version of $M4' = h(Ki'\|CRi'\|MIDi'\| Xgui\|T1)$ is equal to the M4 sent from Ui. If so, the user Ui is legitimate; if not, GWN declines the Ui and sends a message to Nj stating that Ui is not a legitimate user, then session is aborted.
- **Step 12.** After GWN verifies the authenticity of both Ui and Nj and extracts their session key parts Ki and Kj, GWN prepares the messages {M7, M8, M9, M10} and sends to the Nj, then to the Ui so that both the Ui and the Nj mutually authenticate with the GWN. Therefore, the Nj and the Ui can compute the session key (SK) and start encrypting their communication.
- **Step 13.** GWN computes $M7 = Ki' \oplus h(CRj'\|T3)$, $M8 = h(M7\|CRj'\|T3)$, $M9 = Kj \oplus h(CRi'\| Xgui\|T3)$, $M10 = h(M9\|CRi'\|MIDi'\|T3)$, M7 and M8 are used by the Nj, M7 is used to mask the user part of the session key Ki, and M8 to ensure the legitimacy of the GWN. The same applies to M9 and M10. They are used by the user Ui in which M9 is used to mask the Nj part of the session key Kj, and M10 to ensure the legitimacy of the GWN. The message {M7, M8, M9, M10, T3} is sent to Nj.
- **Step 14.** Upon the receipt of the message sent from the GWN, the Nj checks the time $|T3 - Tc| < \Delta T$. If T is within the allowed time span, Nj proceeds with the authentication; if not, the GWN is considered illegitimate and the session is aborted.
- **Step 15.** After the time verification passes, using the stored value of $CRj = h(IDj\| Xgn)$ and the lately received M7, Nj verifies if the received value of $M8 = h(M7\| CRj\|T3)$. If the verification holds, then GWN is legitimate, and thus Nj and GWN are mutually authenticated; otherwise, the message is intercepted and changed by an attacker, and the session is aborted, and a rejection message is sent to the GWN.
- **Step 16.** If the verification of the legitimacy of the GWN holds, Nj computes $Ki' = M7 \oplus h(CRj\|T3)$ to extract the Ui session key part Ki, and then construct the session key (SK) using its own session key part Kj and the newly computed Ki'.

- **Step 17.** The Nj computes the session key SK = h(Ki` ⊕ Kj) and M11 = h(SK|| M9||M10||T3||T4), and sends {M9, M10, M11, T3, T4} to Ui. M10 is used by the Ui to verify the legitimacy of the GWN, and M11 to verify the legitimacy of the Nj.
- **Step 18.** Upon the receipt of the message sent from the Nj, the Ui checks the time |T4 − Tc| < ΔT. If T is within the allowed time span, the Ui proceeds with the authentication. Otherwise, the Nj is considered illegitimate and session is aborted.
- **Step 19.** If the time verification holds, then using the values CRi and MIDi the Ui checks whether the received M10 = h(M9||CRi||MIDi||T3). If correct, then the GWN is legitimate; if not, the GWN is impersonated and session is aborted.
- **Step 20.** Ui extracts the session key part of the Nj using its secret values CRi and Xgui Kj` = M9 ⊕ h(CRi||Xgui||T3) and using its stored session key Ki and newly computed Kj` Ui constructs its version of the session key SK = h(Ki ⊕ Kj`).
- **Step 21.** Finally the Ui check if the received M11 = h(SK`||M9||M10||T3||T4) then, the Nj is legitimate. So, Ui authenticates Nj and GWN and starts using SK for further messages encryption between the user Ui and the IoT node Nj. Otherwise, the Ui rejects the Nj and considers it a malicious attacker.

4.5 Password Change/Update Phase

For reliability and security purposes, the facility of changing/updating the password should be considered when designing any authentication protocol in the case of IoT and constrained networks. It is preferred to keep messages exchanged and communication at minimum so, this phase is executed locally at the user side without interfering with SA or GWN.

- **Step 1.** The user opens the smart home application on his SP and using the password change form. He is asked to inputs his fingerprint on the SP's sensor device then verifies his fingerprint. If the verification passes, the user then is asked to enter his IDi and Password PWi and verifies if stored ei = h(IDi||Pwi||Bi). If verification holds, go to step 2.
- **Step 2.** The user is asked to enter his new password PWinew, in order to extract the values S and Xgui SP compute S = fi ⊕ h(IDi||PWi) and Xgui = gi ⊕ h(IDi||PWi|| Bi). Then Ui computes einew = h((IDi||Pwinew||Bi)) finew = S ⊕ h(IDi||PWinew), ginew = Xgui ⊕ h(IDi|| PWinew||Bi).
- **Step 3.** Replace the old values of ei, fi, gi, with the new values einew, finew, ginew.

5 Security Analysis and Performance Evaluation of the Proposed Protocol

In this section, we illustrate the security features and detailed security evaluation of the proposed protocol. The evaluation is conducted by two different methods. The first one proves the high security of the protocol through theoretical analysis and a comparison with some other related protocols. The second method of the evaluation conducted a formal security analysis using AVISPA simulation software.

5.1 Security Analysis of the Proposed Protocol

Security features and comparison with the related protocol is presented in Table 2.

Table 2. Security features comparison with other protocols

Security feature	Farash [19]	Yeh [30]	Amin [20]	Proposed scheme
Mutual authentication	Yes	Yes	Yes	**Yes**
Key agreement	Yes	Yes	Yes	**Yes**
Password protection	No	Yes	Yes	**Yes**
Password-change	Yes	Yes	Yes	**Yes**
Dynamic node addition	Yes	No	Yes	**Yes**
User anonymity	No	No	No	**Yes**
Node anonymity	Yes	No	Yes	**Yes**
Stolen SP&SC breach attack resilience	Yes	No	Yes	**Yes**
Traceability attack resilience	No	Yes	No	**Yes**
Replay attack resilience	Yes	No	No	**Yes**
Privileged-insider attack resilience	No	Yes	Yes	**Yes**
Stolen verifier attack resilience	No	Yes	yes	**Yes**
Impersonation attack resilience	Yes	No	Yes	**Yes**
Many logged-in with same id attack resilience	Yes	–	Yes	**Yes**
Password change attack resilience	Yes	–	Yes	**Yes**

Mutual Authentication. In the proposed protocol the Ui, the Nj and the GWN all of them authenticate each other. The GWN authenticates Ui and Nj by computing M4 and M6 respectively. In contrary Nj and the Ui both authenticate the GWN by computing M8 and M10 respectively and finally Ui receives M11 and authenticate the Nj and GWN.

Key Agreement. The Ui and the Nj contribute individually to produce a secure session key, in login phase, the Ui generate a nonce Ki and computes M3 = Ki ⊕ h(CRi||Xgui||T1), Ki is securely protected by the shared password Xgui and the one-way hash function. The IoT node Nj also generates a nonce Kj, its part of the session key and computes M5 = Kj ⊕ h(CRj||T1||T2) Kj is securely protected by the password CRj = h(IDj||Xgn) and the one-way hash function. Both the Ui and the Nj successfully compute SK = h(Ki||Kj).

User Anonymity. User anonymity means hiding the identity of the communicating parties during the authentication and key agreement process. The proposed scheme never transmits the identity of the user IDi without protection, and never saves inside the smartphone unmasked.

When Ui sends a message {M1, M2, M3, M4, T1} to the Nj, M1 = IDi ⊕ h (S||T1). The identity of the user IDi is protected with one way hash function h(S||T) where S = h(Xgn||N) and T1 is the fresh time sent by Ui, Xgn is GWN secret key which is known only by GWN, and N is a high entropy random number generated by the GWN to mask its secret key. The combination of both values with one-way hash function keeps them secure, and also keeps the identity of the Ui secure. On other messages, the identity is masked and sent only inside one way hash function h(MIDi||Xgui||T1), h(Ki|| CRi||MIDi||Xgui||T1) which makes it infeasible to retrieve Ui identity by any attacker. IDi is sent unmasked only one time during the registration through a secure channel.

Security Against Smartphone Stolen/Breach Attack. According to [27] a good hacker might use some power analyzing techniques to get the data inside the smart device. The proposed protocol is resistant to such attacks as we are going to explain.

Password Off-Line Guessing Attack. In the proposed protocol each value has the password (ei, fi, gi) is combined with other values and hashed by one way hash function making it hard to break or get the password. The values S = h(Xgn||N) which are sent from the GWN to the Ui during registration is combined with two values; Xgn which is the secret key of the GWN, known only by him, and N which is a highly entropy random number known only by the GWN. After computing fi and gi, both variables S and the secret shared key Xgui (which is known by the GWN and the Ui) will be deleted from the smartphone.

Identity Off-Line Guessing Attack. The Identity of the user is securely stored inside the smartphone, and each value has IDi(MIDi, ei, fi, gi) is secure with one way hash function. So, to get IDi we need to know PWi, S, Xgui, Xgn and Bi.

User/Node Impersonate Attack. Impersonating a legitimate user/node happens when an attacker uses the private values of a legitimate user/node such as identity or password or intercepts and forges a message sent from the Ui/Nj to other participants. IDi and PWi are secured as we mentioned in phone/card breach attacks. When Ui sends the login message to Nj {M1, M2, M3, M4, T1} the attacker needs to have IDi, CRi, S, Xgui, and Ki to compute (M1–M4). Each message in the login is hashed using different secret keys. Therefore, to calculate M1 the attacker needs to know IDi and S which both are known only by the Ui and the GWN. In M2 also, the attacker needs to know the shared secret key Xgui and MIDi which are known only by the Ui and GWN. The same applies to M3 in which the attacker needs to know Ki, CRi, and Xgui which are all kept secret from attackers also when the node Nj sent MIDj, M5, M6, T2 to the GWN the attacker doesn't know IDj, CRj, Y and Kj and is computationally infeasible to compute way hash function.

User Traceability. The attacker can trace user Ui when sending a login message. The attacker compares two different login messages and finds constant values, and hence can differentiate between users. In the proposed protocol, the user sends M1 = IDi ⊕ h (S||T1) where the user IDi is hidden and also M1 value is dynamically changed because of the time T1 which is different in every login.

Node Traceability. The same with the Nj, when sending the masked identity MIDj = IDj ⊕ h(Y||T2), the value of masked identity of the node is changeable in every login by the timestamp T2 so, The proposed protocol is safe against tractability attacks.

Privileged Insider and Stolen-Verifier Attacks. In the proposed scheme, GWN does not store user password PWi in any tables. It attaches its master secret key Xgn and the shared secret password Xgui to Ui verifiers (IDi and PWi) during the registration phase. Accordingly, a malicious privileged user can't get any user sensitive information. Therefore, an attacker cannot impersonate the user. Furthermore, when the Ui initiates the authentication phase, the Nj forwards the hashed message to the GWN, whereby a privileged user cannot extract Ui's password. The one-way property of the hash function prevents any attacker from getting any information. Consequently, the proposed scheme is resilient against both privileged Insider and Stolen-Verifier Attack.

Other Type of Attacks. *Many logged-in users with the same login-id attack*, *Password Change Attack and Replay attack*: Our proposed scheme uses a smartphone for a user's login or to Password Change. An attacker needs a legitimate smartphone to login or to change the password and also the user's fingerprint and password to successfully execute the login and change password phase. Timestamps are used in every message exchanged in login and authentication phase to prevent the replay attack. Therefore the proposed scheme is resilient against these attacks.

5.2 Performance Evaluation of the Proposed Protocol

Computational Cost of the Proposed Protocol. Computational cost varies from one scheme to another depending on the number of security features, number of attacks the scheme resists, and the type of cryptographic security primitives that the scheme uses. The proposed scheme uses the most lightweight cryptographic security primitives that are Xor and Hash; and thus provides a robust security against most of the well-known attacks. The security features comparison between our scheme and others authentication schemes is summarized in Table 2. In addition, the computational cost comparison of our scheme and others related schemes are summarized in Table 3.

Table 3. Computational cost of the proposed protocol with other related protocols

Protocol	User	IoT sensor	Gateway	Total computational cost
Farash [19]	11 T_h	7 T_h	14 T_h	32 T_h
Yeh [30]	1 T_h + 2 $T_{(d/e)}$	3 T_h + 2 $T_{(d/e)}$	4 T_h + 4 $T_{(d/e)}$	8 T_h + 8 $T_{(d/e)}$
Amin [20]	13 T_h	5 T_h	16 T_h	34 T_h
Proposed scheme	**13 T_h**	**7 T_h**	**13 T_h**	**33 T_h**

The proposed protocol uses a total number of 33 hashes. Although the protocol of farash used 1 hash operation less than our protocol but we have solved the security drawbacks in farash protocol as it fails to preserve user-anonymity, stolen-smartcard attacks, off-line password-guessing attack and user-impersonation attack. Our protocol also uses biometric for user login. Therefore for the extra security features that our protocol provides this difference can be neglected.

The author in [28] conducted an experiment to measure the energy cost on a sensor (i.e. CrossBow's MICA2) on an average message size of 24 bytes when hashed using SHA1 and for encryption/decryption using AES. The result was ≈ 0.075 J(Ws) and 0.241 J(Ws) for SHA1 and AES encryption/decryption respectively. Our scheme uses 7 hashes. Accordingly, the total energy cost consumed by the sensor is 0.525 J for each authentication cycle.

Storage Cost of the Proposed Protocol. Storage cost analysis is made for sensor and smartphone memory most of the protocols shown in Fig. 4 present the same storage cost for the smartphone memory. For sensor storage cost we have taken the measurements when the sensor has the maximums number of bits (moment of peak) it shows that the sensor in proposed protocol holds 256 bits as shown in Fig. 5 where its way far of typical sensor storage which is 128,000 bits.

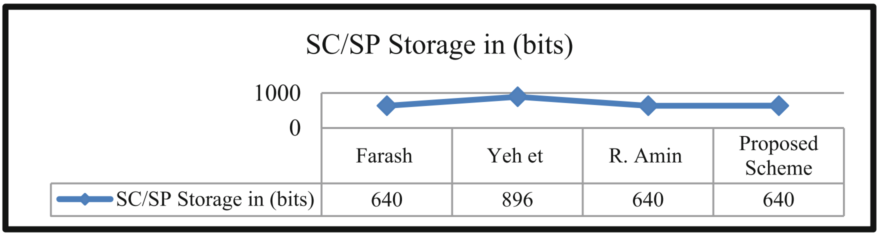


Fig. 4. Smartphone storage cost of the proposed protocol and other related protocols

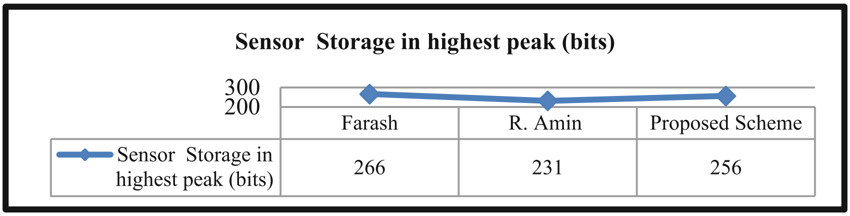


Fig. 5. Sensor storage cost of the proposed protocol and other related protocols

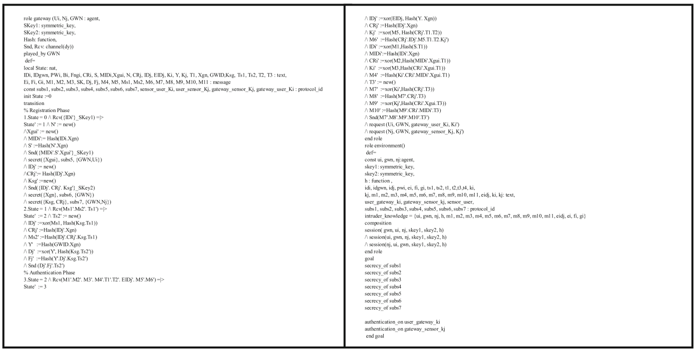
Communication Cost of the Proposed Protocol. In the proposed protocol four messages are exchanged between the Ui, the Nj and the GWN. In the first, third and fourth messages the packet size is 99 bytes and 98 bytes respectively. Their size is below the standard packet size (i.e. 127) and for that can be carried out without extra processing except for the second message that is sent from the Nj to the GWN as it carries both messages that come from the Ui and from the Nj as our protocol uses the direct approach where the user directly contacts the IoT device, not the gateway. The total number of bytes is 178 which can be handled by 6LoWPan (IPv6 over Low power Wireless Personal Area Networks) layer. The idea behind the design of 6LoWPan layer was for such situation where the packet size is more than 127 bytes of the regular standard size of IEEE 802.15.4. 6LoWPan compressed, fragmented and encapsulated large packets so they can fit into standard IEEE 802.15.4 packet frames.

6 Formal Security Analysis of Proposed Protocol

To support the result of the theoretical analysis we implemented our proposed protocol using AVISPA simulation tool. AVISPA (Automated Validation of Internet Security Protocols and Applications) is a strong simulation engine for automated security analysis of cryptographic protocols. It is used to confirm the security attributes of protocols and applications. AVISPA uses High Level Protocol Specification Language (HLPSL) [29].

6.1 HLPSL Specification of the Proposed Protocol

The proposed protocol is composed of three participants, namely, the user, the sensor, and the gateway. They are represented as Ui, Nj and GWN respectively. The implementation in Fig. 6 represents HLSPL specification of the gateway and the environment roles.



```
role gateway (Ui, Nj, GWN : agent,
SKey1: symmetric_key,
SKey2: symmetric_key,
Hash: function,
Snd, Rcv: channel(dy))
played_by GWN
 def=
local State: nat,
IDi, IDgwn, PWi, Bi, Fngi, CRi, S, MIDi,Xgui, N, CRj, IDj, EIDj, Ki, Y, Kj, T1, Xgn, GWID,Ksg, Ts1, Ts2, T2, T3 : text,
Ei, Fi, Gi, M1, M2, M3, SK, Dj, Fj, M4, M5, Ms1, Ms2, M6, M7, M8, M9, M10, M11 : message
const subs1, subs2, subs3, subs4, subs5, subs6, subs7, sensor_user_Ki, user_sensor_Kj, gateway_sensor_Kj, gateway_user_Ki : protocol_id
init State :=0
transition
% Registration Phase
1.State = 0 /\ Rcv({IDi'}_SKey1) =|>
State' := 1 /\ N' := new()
/\Xgui' := new()
/\ MIDi':= Hash(IDi.Xgn)
/\ S' :=Hash(N'.Xgn)
/\ Snd({MIDi'.S'.Xgui'}_SKey1)
/\ secret({Xgui}, subs5, {GWN,Ui})
/\ IDj' := new()
/\CRj':= Hash(IDj'.Xgn)
/\ Ksg' :=new()
/\ Snd({IDj'. CRj'. Ksg'}_SKey2)
/\ secret({Xgn}, subs6, {GWN})
/\ secret({Ksg, CRj}, subs7, {GWN,Nj})
2.State = 1 /\ Rcv(Ms1'.Ms2'. Ts1') =|>
State'  := 2 /\ Ts2' := new()
/\ IDj' :=xor(Ms1, Hash(Ksg.Ts1))
/\ CRj' :=Hash(IDj'.Xgn)
/\ Ms2' :=Hash(IDj'.CRj'.Ksg.Ts1)
/\ Y'   :=Hash(GWID.Xgn)
/\ Dj'  :=xor(Y', Hash(Ksg.Ts2'))
/\ Fj'  :=Hash(Y'.Dj'.Ksg.Ts2')
/\ Snd (Dj'.Fj'.Ts2')
% Authentication Phase
3.State = 2 /\ Rcv(M1'.M2'. M3'. M4'.T1'.T2'. EIDj'. M5'.M6') =|>
State'  := 3
```

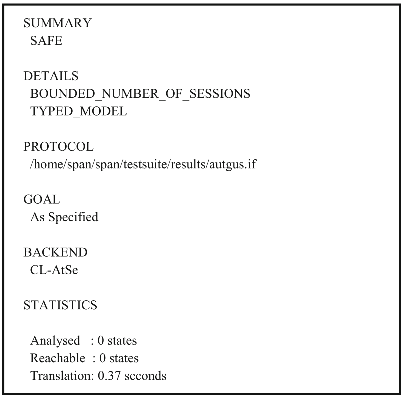
```
/\ IDj' :=xor(EIDj, Hash(Y. Xgn))
/\ CRj' :=Hash(IDj'.Xgn)
/\ Kj'  :=xor(M5, Hash(CRj'.T1.T2))
/\ M6'  :=Hash(CRj'.IDj'.M5.T1.T2.Kj')
/\ IDi' :=xor(M1,Hash(S.T1))
/\ MIDi':=Hash(IDi'.Xgn)
/\ CRi' :=xor(M2,Hash(MIDi'.Xgui.T1))
/\ Ki'  :=xor(M3,Hash(CRi'.Xgui.T1))
/\ M4'  :=Hash(Ki'.CRi'.MIDi'.Xgui.T1)
/\ T3' := new()
/\ M7'  :=xor(Ki',Hash(CRj'.T3))
/\ M8'  :=Hash(M7'.CRj'.T3)
/\ M9'  :=xor(Kj',Hash(CRi'.Xgui.T3))
/\ M10' :=Hash(M9'.CRi'.MIDi'.T3)
/\ Snd(M7'.M8'.M9'.M10'.T3')
/\ request (Ui, GWN, gateway_user_Ki, Ki')
/\ request (Nj, GWN, gateway_sensor_Kj, Kj')
end role
role environment()
 def=
const ui, gwn, nj:agent,
skey1: symmetric_key,
skey2: symmetric_key,
h : function ,
idi, idgwn, idj, pwi, ei, fi, gi, ts1, ts2, t1, t2,t3,t4, ki,
kj, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, eidj, ki, kj: text,
user_gateway_ki, gateway_sensor_kj, sensor_user,
subs1, subs2, subs3, subs4, subs5, subs6, subs7 : protocol_id
intruder_knowledge = {ui, gwn, nj, h, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, eidj, ei, fi, gi}
composition
session( gwn, ui, nj, skey1, skey2, h)
/\ session(ui, gwn, nj, skey1, skey2, h)
/\ session(nj, ui, gwn, skey1, skey2, h)
end role
goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
secrecy_of subs5
secrecy_of subs6
secrecy_of subs7

authentication_on user_gateway_ki
authentication_on gateway_sensor_kj
 end goal
```

Fig. 6. HLPSL specification of the gateway GWN and the environment role

6.2 Result of the Simulation

We have used the back-end CL-AtSe (Constraint-Logic-based Attack Searcher). It provides a translation from any security protocol specification written as a transition relation in an intermediate format (IF) into a set of constraints, which are effectively used to find security weaknesses of the designed protocol. The result of the proposed protocol as shown in Fig. 7 is **SAFE** indicating that the protocol is secure from different types of attacks.



```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/autgus.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 0 states
 Reachable  : 0 states
 Translation: 0.37 seconds
```

Fig. 7. AVISPA output result of the proposed protocol

7 Conclusion

This paper proposed a remote biometric mutual authentication and key agreement protocol for the IoT environment. The user contacts the IoT node directly without contacting the gateway at first. It is best for a scenario where data has to be retrieved on- demand directly from the IoT node. We have conducted a deep security analysis for possible security attacks also we have implemented the protocol using AVISPA tool to make sure of its robustness and security. In addition, we have also done a performance evaluation of the protocol to prove its efficiency for the IoT environment.

The result shows that the proposed protocol resists to most known security attacks and lightweight in term of computation, memory, and communication costs which is suitable for the IoT environment.

References

1. Bonino, D., et al.: ALMANAC: Internet of Things for smart cities. In: 2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud), pp. 309–316, August 2015
2. Alkuhlani, A., et al.: Internet of Things (IOT) standards, protocols and security issues. https://doi.org/10.17148/IJARCCE.2015.411109
3. Zhao, K., Ge, L.: A survey on the Internet of Things security. In: 2013 Ninth International Conference on Computational Intelligence and Security. IEEE (2013). https://doi.org/10.1109/CIS.2013.145
4. Miorandi, D., et al.: Internet of Things: vision, applications and research challenges. Ad Hoc Netw. **10**(7), 1497–1516 (2012)
5. Jing, Q., et al.: Security of the Internet of Things: perspectives and challenges. Wirel. Netw. **20**(8), 2481–2501 (2014)
6. Chatzigiannakis, I., et al.: True self-configuration for the loT. In: 2012 3rd International Conference on the Internet of Things (IOT). IEEE (2012)
7. Sethi, P., Sarangi, S.R.: Internet of things: architectures, protocols, and applications. J. Electr. Comput. Eng. **2017** (2017). https://doi.org/10.1155/2017/9324035

8. Saied, Y.B., et al.: Lightweight collaborative key establishment scheme for the Internet of Things. Comput. Netw. **64**, 273–295 (2014)
9. Kalirai, J., Kumar, I.: Lightweight cryptography by simplification of hardware – a comparison study. In: RFID Systems EE260, Spring, 18 May 2015
10. Das, A.K., Sharma, P., Chatterjee, S., Sing, J.K.: A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. J. Netw. Comput. Appl. **35** (52), 1646–1656 (2012)
11. Liu, J., Xiao, Y., Chen, C.P.: Authentication and access control in the Internet of Things. In: 2012 32nd International Conference on Distributed Computing Systems Workshops. IEEE (2012). https://doi.org/10.1109/ICDCSW.2012.23
12. Turkanović, M., Hölbl, M.: An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks. Elektronika Ir Elektrotechnika **19**(6), 109–116 (2013)
13. Xue, K., Ma, C., Hong, P., Ding, R.: A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. J. Netw. Comput. Appl. **36**(1), 316–323 (2013)
14. Li, C.-T., Weng, C.-Y., Lee, C.-C.: An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. Sensors **13**(8), 9589–9603 (2013)
15. Brumen, B., Turkanović, M., Hölbl, M.: A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. Ad Hoc Netw. **20**, 96–112 (2014)
16. Ndibanje, B., Lee, H.J., Lee, S.G.: Security analysis and improvements of authentication and access control in the Internet of Things. Sensors **14**(8), 14786–14805 (2014). https://doi.org/10.3390/s140814786
17. He, D., Kumar, N., Chilamkurti, N.: A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Inf. Sci. **321**, 263–277 (2015)
18. Amin, R., Biswas, G.P.: A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw. (2015). https://doi.org/10.1016/j.adhoc.2015.05.020
19. Farash, M.S., Turkanović, M., Kumari, S., Hölbl, M.: An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. Ad Hoc Netw. **36**, 152–176 (2016)
20. Amin, R., et al.: Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Comput. Netw. (2016). https://doi.org/10.1016/j.comnet.2016.01.006
21. Arasteh, S., et al.: A new lightweight authentication and key agreement protocol for Internet of Things. In: 13th International ISC Conference on Information Security and Cryptology, ISCISC 2016, 7–8 September 2016, Shahid Beheshti University, Tehran, Iran (2016)
22. Dhillon, P.K., Kalra, S.: A lightweight biometrics based remote user authentication scheme for IoT services. J. Inf. Secur. Appl. (2017). https://doi.org/10.1016/j.jisa.2017.01.003
23. Jiang, Q., et al: Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. IEEE Access. https://doi.org/10.1109/ACCESS.2017.2673239
24. Chen, T.H., Shih, W.K.: A robust mutual authentication protocol for wireless sensor networks. ETRI J. **32**(5), 704–712 (2010)
25. Lumini, R., Nanni, L.: An improved BioHashing for human authentication. Pattern Recognit. **40**(3), 1057–1065 (2007)

26. Jin, A.T.B., Ling, D.N.C., Goh, A.: BioHashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognit. **37**(11), 2245–2255 (2004)
27. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25
28. Chang, C.-C., Nagel, D.J., Muftic, S.: Assessment of energy consumption in wireless sensor networks: a case study for security algorithms. In: 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems, MASS 2007, pp. 1–6 (2007)
29. Armando, A., et al.: The AVISPA tool for the automated validation of internet security protocols and applications. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 281–285. Springer, Heidelberg (2005). https://doi.org/10.1007/11513988_27
30. Yeh, H.-L., Chen, T.-H., Liu, P.-C., Kim, T.-H., Wei, H.-W.: A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors **11**, 4767–4779 (2011)