Multimedia Tools and Applications (2022) 81:13823–13843
https://doi.org/10.1007/s11042-022-11927-y

A novel protocol for efficient authentication in cloud-based IoT devices

Irfan Alam [1] · Manoj Kumar [1]

Received: 14 July 2021 / Revised: 8 December 2021 / Accepted: 3 January 2022 /
Published online: 24 February 2022

Abstract
The Internet of Things (IoT) has emerged as one of the most revolutionary technological innovations with the proliferation of applications within almost all fields of the human race. A cloud environment is the main component of IoT infrastructure to make IoT devices efficient, safe, reliable, usable, and autonomous. Reduction in infrastructure cost and demand accessibility of shared resources are essential parts of cloud-based IoT (CIoT) infrastructure. Information leakage in cloud-assisted IoT devices may invite dangerous activities and phenomena. Various cloud-based systems store IoT sensor data and later on access it accordingly. Some of them are public, and some of them are private. Private cloud services must be secured from external as well as internal adversaries. Hence, there must be a robust mechanism to prevent unauthorized access to devices. This paper proposes a novel and efficient protocol based on the Elliptic Curve property known as Elliptic Curve Discrete Logarithm Problem (ECDLP) with hash and XOR functions for the authentication in cloud-based IoT devices. In comparison to the existing protocols, the proposed protocol is resistant to attacks and other security vulnerabilities. The one-way hash function and XOR function effectively ensure a reduction in computation cost. AVISPA and BAN logic have been used for formal analysis of the proposed protocol. As per the performance analysis results, it is clear that the proposed protocol is efficiently suitable for cloud-assisted IoT devices.

Keywords Authentication · Protocol · ECC · Attacks · XOR · AVISPA

1 Introduction

The Internet of Things (IoT) is a setup of different entities like objects, people, and animals capable of transferring data over networks without head-to-head or head-to-computer

✉ Irfan Alam
irfanrmamu@gmail.com

[1] Department of Computer Science and Engineering, Delhi Technological University, New Delhi 110042, India

connection. IoT is an acronym for Any time, Anything, Any place connections. In recent years, the Internet of Things (IoT) has risen to prominence as a critical technological component for overcoming interoperability, heterogeneity, and Internet-aware resistances.

The deployment of IoT devices has increased exponentially, resulting in a large amount of data handling and analysis [8]. There is a need for a standard platform to manage heterogeneous natured gadgets and data. Cloud-based infrastructures are well suited to this need. Ray, in his survey paper, describes applications for the IoT cloud platforms [39]. In general, cloud services are of two types: public cloud services and private cloud services. A person or organization owns private cloud services that make cloud security essential [53]. Analysis of the cloud-based IoT (CIoT) device mechanism [2, 7, 14, 36] shows that most applications carry essential information, and leakage of such information may lead to a significant loss and several attacks. Also, the nature of attacks changes from time to time. Computing resources are employed on-demand via a network in the Cloud services. It consumes a significant amount of energy, which does not measure easily with computational resources. Pete et al. [37] proposed an experimental energy-efficient model for Virtual Machines (VMs) scheduling over the Cloud; furthermore, Kumar et al. [27] illustrate an economically efficient model for virtualization over the Cloud using a docker container.

Authentication is a primary security mechanism for all network-based services that prevent unauthorized users or adversaries from gaining access [48]. It is compulsory to ensure that only authenticated users should access the resources. For information exchange through an unsecured network such as the Internet, IoT authentication can ensure the trust of IoT devices. A robust IoT authentication model is needed to protect unauthorized users and devices. Traditional public-key cryptosystems are commonly used in authentication procedures. Traditional public-key cryptosystems, such as RSA, have large key sizes and use a lot of computing resources [46]. As a result, most standard authentication systems are incompatible with the limited computational power of IoT devices. Recently, several research papers addressed IoT authentication and the possibility of attacks [4, 22, 30, 34, 51, 52]. However, many attacks are still unresolved, such as packet analysis, gateway forgery, and Sybil attacks. Choosing an appropriate protocol to address recent attacks is tough task, due to unavoidable resource constraints, such as low power support, small storage, low computational support, and low latency support. The proposed protocol is a solution for above said challenges. The significant contributions of this paper have been listed as follows:

- A novel protocol has been proposed for the IoT infrastructure based on a cloud environment.
- The proposed novel authentication protocol prevents adversaries from common attacks such as Man in the middle, Masquerade, Denial of Services (DoS), Forging, Guessing, and physical attacks.
- It can provide robust and secure authentication using the Elliptic Curve Discrete Logarithm Problem (ECDLP) property of ECC with hash and XOR functions. NP-hard complexity of ECDLP makes the protocol resistant to attacks such as Forging, guessing, and Man in the middle, and the inclusion of XOR and hash keeps it lightweight.
- The proposed protocol has been compared with the existing protocols to explore its ability to resist attacks, reduce computation power and storage requirements, etc. Its computational cost is very low (0.2 s) and resists more than fifteen modern attacks. It also shows its usefulness more than other existing protocols.

- The modular structure of protocols varies from application to application. We have tried to modal the proposed protocol in such a way so that it suits 90% of IoT infrastructures.
- Simulation of the proposed protocol using AVISPA [5, 44] and BAN logic [43, 50] proves its power against active adversaries.
- Description of BAN logic and AVISPA in the present article would allow readers to understand and apply these tools for other protocols easily.

The rest of the paper deals with the following way - Section 2 discusses the related work, followed by section 3, which describes the Preliminaries. Section 4 presents the details of the proposed protocol. Section 5 discusses the performance and security analysis. Finally, the paper presents the conclusion and future work in section 6.

2 Related works

The selection of proper methodology for securing the flow of information plays a vital role in secured and safe communication. This selection varies for different applications, such as polynomial congruence is used for multimedia encryption over the Cloud [26]. Similarly, the secret-sharing concept has been used to encrypt video [31, 40]. The proposed protocol meets the requirements of real-time applications by minimizing the required time and enhancing the efficiency of the encryption process. Similarly, Gadicha et al. [17] present a multimode approach of data encryption in images through Quantum Steganography. This work opens a new chapter in modern quantum cryptography dealing with images in multi-modal aspects.

In general, Confidentiality, Integrity, and Availability (CIA) are the main features to be taken care of in the case of traditional security. In contrast, in the case of IoT, security measures vary from application to application. In 2020, Iqbal et al. [21] assessed IoT security requirements, challenges, and remedies in-depth. From there, it is clear that authentication is mandatory for almost all applications related to cloud-based IoT infrastructure.

In their survey paper, Nandy et al. [35] reviewed all aspects of IoT authentication. It also explains the taxonomy of all possible attacks on IoT authentication, factors involved in choosing the authentication technique, and the tools for simulation of authentication protocols for IoT devices. Authentication protocols are usually based on three parameters: biometric features of users, credentials, and smart cards. In 1981, Lamport [29] proposed a password-based authentication protocol for untrusted networks. However, a stolen-verifier attack was found in password table-based protocols at the server end. After that, numerous protocols have been proposed using a single or combination of the techniques such as Smart card, Password, and Biometrics. In 1985, Victor Saul Miller [33] and Neal I. Koblitz [19] came up with the elliptic curve in cryptography. Elliptic curve cryptography (ECC) algorithms have been used since 2004 widely due to their small size and low computational requirements with the same security level as RSA [21, 48]. In [10, 23, 28, 38], authors have worked with the Computational hardness of ECC for IoT devices. In [24], Kim et al. have done cryptanalysis of the hash function. Protocols like [22, 42, 52], and others have used hashed-based algorithms, which are helpful to ensure the integrity of the messages and verify the sender as well.

In 2015, Li et al. [30] proposed a mutually authenticated protocol based on a smart card in cloud computing. Their protocol does not persist against physical attack, forging attack, and masquerade attack [50]. Furthermore, Sun et al. [43] proposed a novel remote user authentication and key agreement protocol for the mobile client-server Environment. Still, it does not provide a pleasant process of password update and biometric change [50].

After that, in 2016, an efficient authentication protocol was proposed for IoT-enabled devices in a distributed cloud environment by Amin et al. [4]. In this work, different attacks has been discussed, such as user anonymity, insider attack. In 2018, Challa et al. [26] analyzed that Amin et al. protocol [4] does not resist masquerade attacks such as privileged-insider attacks and impersonation attacks. An adversary can send a valid login message on behalf of the user. In 2018, Wu et al. [51] explained that Irshad et al. [22] and Amin et al. [4] are vulnerable to Privileged Insider (PI) attacks and Offline Password Guessing (OPG) attacks, respectively. They both do not guarantee User Anonymity (UA).

Recently, Wu et al. [52] analyzed Xu et al. protocol [51]. They concluded that Xu et al. protocol could not resist Privileged Insider (PI) and stolen Smart Card (SSC) attacks and did not provide pre-verification and Perfect Forward Secrecy (PFS). If we look deep into the hash functions, we can get an exciting result. If the attacker replaces the contents of two messages, such as msg1 with msg2, the receiver will not detect this change since both messages generate the message same hash. This defect may allow adversaries to play with the networks that lead to various attacks such as user and sensor impersonation attacks and guessing attacks. In this scenario, the protocols discussed above may not be able to resist modern attacks.

Considering the shortcomings of the above-discussed protocols, we have proposed this protocol. It uses ECC along with the hash and XOR operators. ECDLP makes the proposed protocol resist the above-discussed attacks and other recent attacks due to its NP-hard complexity. The use of hash and XOR functions keeps the proposed protocol computationally lightweight.

3 Preliminaries

3.1 Elliptic curve discrete logarithm problem (ECDLP)

For two points, P and Q, of an elliptic curve $E_P(a,\ b)$ the **ECDLP** is to find an integer $k \in [1,\ n\ -\ 1]$ such $Q\ =\ k.\ P$.

ECDLP is more complicated than most discrete logarithm and factorization problems in cryptography. There are several attempts to break this problem; however, it is still unbreakable.

3.2 One-way cryptographic hash function

Here we describe the brief properties of the hash function, which are as follows

- This function is deterministic h :$\{0,\ 1\}^* \rightarrow \{0,\ 1\}^n$
- h is the hash function, where output produces y = h(x) (hash digest)$y \in \{0,\ 1\}$.
- Any tiny modification to the input string potentially leads to an entirely different hash value (message digest).

3.3 System model

The modular architecture of IoT-based systems may vary from application to application. We have tried to consider the most common approach suited for 90% of IoT infrastructure. As shown in Fig. 1, the main components of the proposed system are following:

- User (U),
- Cloud Server (S_m) directly associated with IoT devices, and
- Trusted authority (TA) sometimes called a control server

Both user (U) and cloud server (S_m) must register with TA before U can access (S_m) data via a secure channel. Necessary Credentials for U and (S_m) are generated by a Trusted Authority (TA) and stored in their memory. A session key is established (After mutual authentication between U and (S_m) for future independent communication.

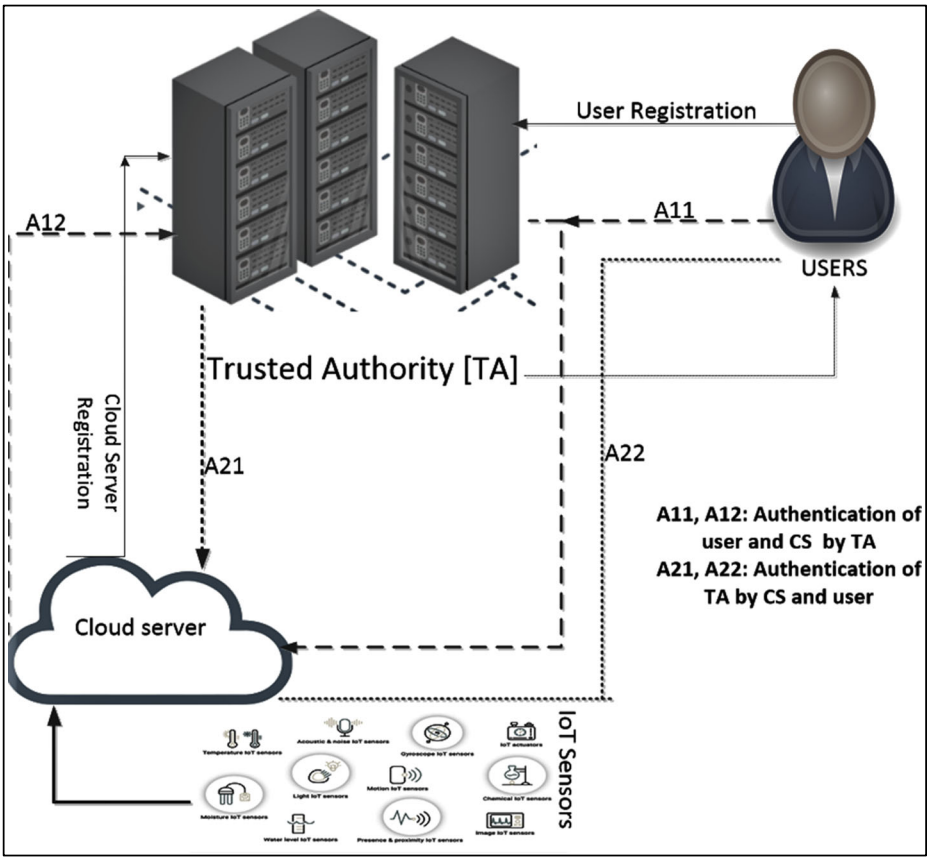

Fig. 1 Cloud-based IoT environment authentication model

3.4 Adversary model

If there is a protocol for some security purpose, there must be some adversary model against which the protocol is safe. A fundamental and most common adversary model was proposed by Dolev and Yao [16] in 1983. It is a broadly accepted model. According to this model, an adversary can read, alter and perform decryption (if got right keys) on messages. Adversaries are unable to carry out any statistical or cryptanalytic assaults. Due to rapid transformation in technologies, adversaries are now more advanced and have extraordinary capacities. Along with the Dolev and Yao model, we have considered that adversaries would know critical information from smart cards (if lost/stolen) using power analysis of smart cards [25, 32]. An adversary would also extract data from network flow using a network analyzer and recent AI techniques [15, 54].

4 Proposed protocol

This section presents an ECC-based protocol designed mainly to ensure that the devices are accessible in a secure manner on public channels. Cloud involvement in all entities and clock synchronization is our assumption for the proposed protocol. In Table 1, we listed necessary and frequent notations with their explanation for the proposed protocol. The proposed protocol comprises different phases, viz., pre-deployment, registration, login, and authentication. In the pre-deployment phase, the setup of the elliptic curve is discussed with the declaration of public and private keys. Registration of cloud server and user have been described in sections 4.2 and 4.3, respectively. After successful registration, verification of the user is done in the login phase via smart card by a trusted authority. In section 4.5, firstly, user authentication proceeded by cloud server authentication is done by a trusted authority. Later, trusted authority is authenticated by both user and cloud server consecutively. At the same time, a session key is established for further communications between the user and the cloud server.

Table 1 Notation used in the proposed protocol

Explanation	Notation
An elliptic curve	$E_P(b_1, b_2)$
Cloud server	S_m
Identity of the i^{th} User	U_i
Trusted authority	TA
Password of User	p_u
Card Reader	CR
Z_p= {0, 1, · ·, p−1}, a prime finite field	Z_p
Transmission delay	ΔT
Timestamps of CS, User and TA, respectively	TS_m, TS_u, T_{TA}
Pair of public key, private key of TA	$\{Q_{TA}, d_{TA}\}$
Concatenation operation	$\|\|$
Hash functions	H (.)
Session key	SK
XOR function	$\oplus$

4.1 Pre-deployment phase

An elliptic curve $E_P(b_1, b_2)$of non-singular nature over a finite field Z_p is selected by TA, where P is a large prime and $\mathbf{4}b_1^3 + \mathbf{27}b_2^2 \neq \mathbf{0}(mod\ p)$. At the same time, a base point P of order n over $E_P(b_1, b_2)$ where TA also chooses n.P = O, and a private key$d_{TA} \in Z_p$. Then TA computes the public key $Q_{TA} = d_{TA}$.P corresponding tod_{TA}. TA chooses the one-way cryptographic hash function. In the same way, cloud servers and the user also compute private keys and public keys using ECC properties.

4.2 Registration of cloud server

1. Cloud server S_m chooses identity (sid_m),$d_{cs} \in Z_{cs}$and $Q_{cs} = d_{cs}$. P after that sends {sid_m, d_{cs} and Q_{cs}} securely to TA as shown in Fig. 2.
2. TA computes$psid_m$= h ($sid_m||d_{cs}$) and$B_{sm} = h(psid_m||\ Q_{TA})$.
3. TA sends B_{sm}toS_m, which saves B_{sm} and d_{cs} in the server memory.

4.3 User registration phase

1. User chooses identity (U_i), password (P_u), $d_u \in Z_P$ and $Q_u = d_u$. P as shown in Fig. 3.
2. User calculate $A_u = h(p_u||d_u)$, $PID_u = h(U_i||Q_u)$ *and* $bb_u = Q_u \oplus A_u$
3. Then user send A_u and PID_u to a Trusted Authority (TA)
4. After receiving A_u and PID_u, TA calculates $h(A_u||PID_u)$, $D_u = h(PIDu||d_{TA})$ *and* $E_u = D_u \oplus A_u$
5. After step 4, TA send [C_u, E_u] to the user
6. User again calculate $DP_u = h(U_i||P_u) \oplus d_u$
7. User saves [C_u, E_u, DP_u]in the smart card.

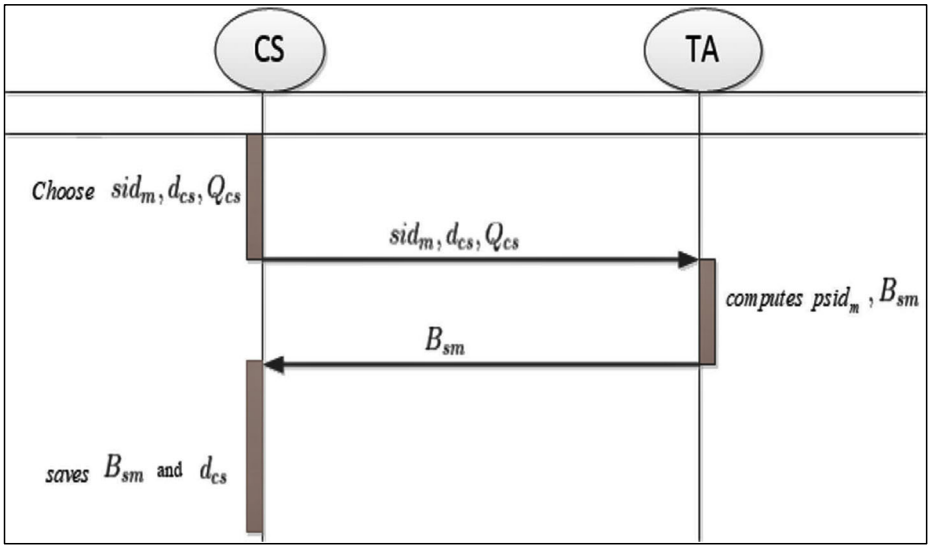

Fig. 2 Cloud Server Registration

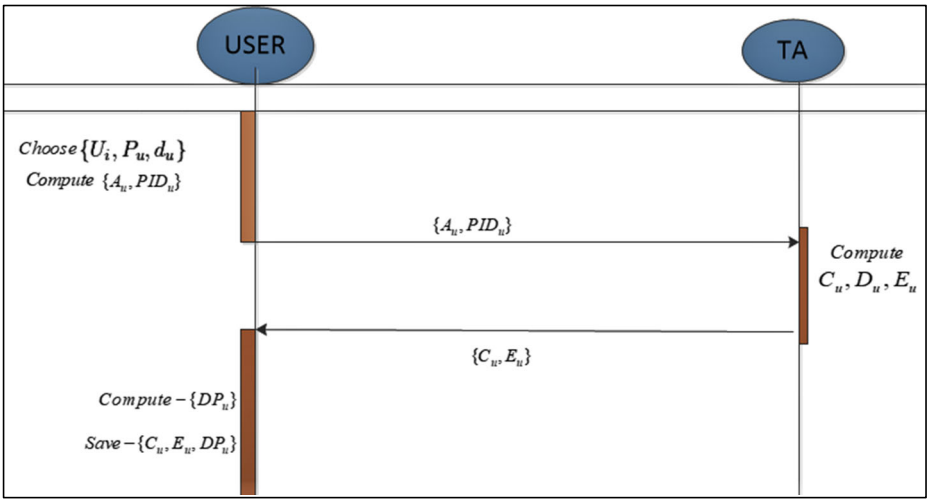

Fig. 3 User Registration

8. Finally, a smart card holds [C_u, E_u, DP_u, $h(.)$, *ECC parameters*]

4.4 Login phase

1. Firstly, a smart card is punched into the card reader by the legal user for accessing server resources, and then the user provides Id and password (U_i*, P_u*) in the card reader terminal, as shown in Fig. 4.

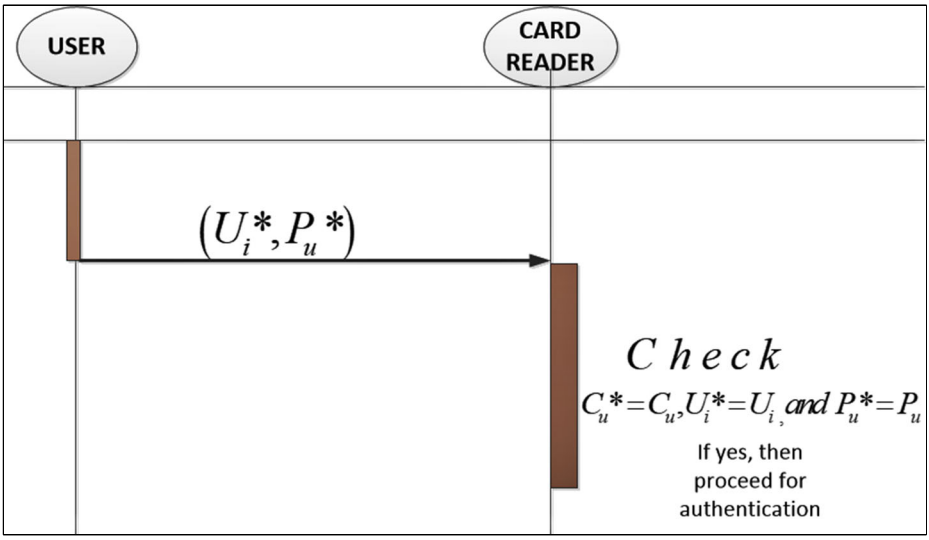

Fig. 4 pre-verification from card reader

2. After first step card reader calculates the following

(i)

$$d_u* = DP_u \oplus h\left(U_i*\left(P_u\right)\right) \quad and \quad A_u* = h\left(p_u*\left(d_u*\right)\right)$$

(ii)

$$Q_u* = bb_u* \oplus A_u*$$

(iii)

$$PID_u* = h\left(U_i*\left(Q_u*\right)\right)$$

(iv)

$$C_u* = h\left(A_u*\left(PID_u*\right)\right)$$

Card reader checks the condition $C_u * = C_u$, $U_i * = U_i$,*and* $P_u * = P_u$ if satisfies, then proceed for an authentication phase.

4.5 Authentication phase

After the login phase, the smart card produces a nonce N_u, Calculate $D_u = E_u \oplus A_u$, $F_u = D_u \oplus N_u$, *and* $Z_u = SID_m \oplus h(D_u \| N_u)$ and send $[F_u, Z_u, PID_u, TS_u]$,to the cloud-serverS_m, here (SID_m) is the cloud server identity chosen by the user and TS_u is the user's timestamp.

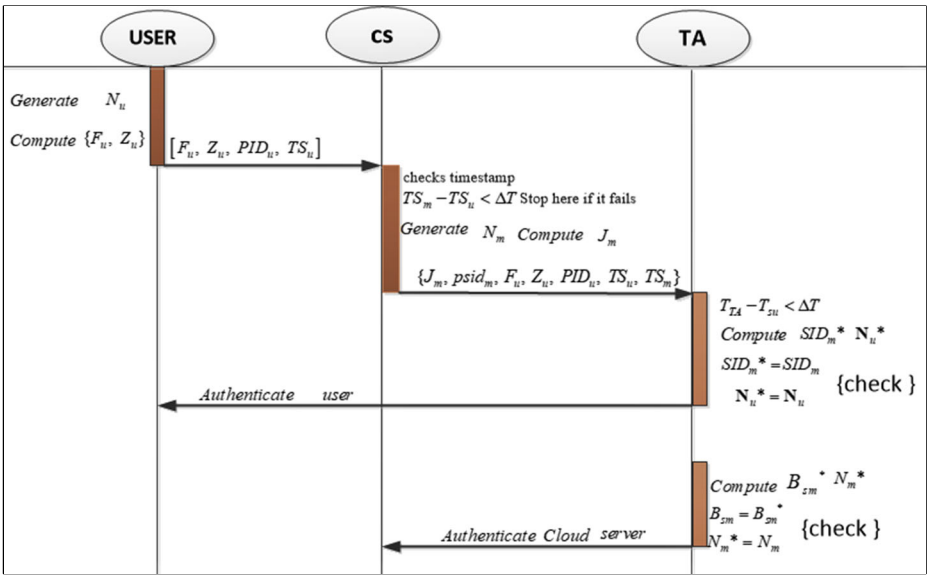

Fig. 5 Authentication of User and Cloud Server

The following steps are done on the Cloud server-side, as shown in Fig. 5.

1.1 At the cloud server, it checks timestamp $TS_m - TS_u < \Delta T$ (Stop here if it fails, where TS_m is a current timestamp and ΔT is the time interval to be expected during message transmission, respectively)
1.2 Cloud server produces 128 bits nonce N_m .

1. 3 Calculates $J_m = B_{sm} \oplus N_m$ and sends $[J_m, psid_m, F_u, Z_u, PID_u, TS_u, TS_m]$ to TA.
2. After receiving the above keys, TA confirms the validity of timestamp ($T_{TA} - T_{su} < \Delta T$) and calculates$N_u * = F_u \oplus D_u$, $SID_m * = Z_u \oplus D_u$ and check $SID_m * = SID_m$ and $N_u^* = N_u$. If this is ok then TA authenticates the user as the legal user.
3. After authenticated User, TA computes $B_{sm} = h(psid_m \| Q_{TA})$ and $N_m^* = B_{sm}^* \oplus J_m$ and checks $B_{sm}^* = B_{sm}$ and $N_m * = N_m$ if it is ok, TA authenticates the Cloud server.
4. After Authenticated User and Cloud server, TA chooses a 128-bit nonce $N_{TA}$and computes the following:

(v)

$$P_{TA} = N_m \oplus N_{TA} \oplus h\left(N_u\left(D_u\right)\right.$$

(vi)

$$SK_{TA} = h(N_u \oplus N_m \oplus N_{TA})$$

(vii)

$$R_{TA} = N_u \oplus N_{TA} \oplus h\left(B_{sm}^* \| N_m^*\right)$$

(viii)

$$Z_{TA} = h\left((N_m \oplus N_{TA})\left(\quad SK_{TA}\right)\right.$$

(ix)

$$V_{TA} = h\left(\left(N_u\left(\quad N_{TA}\right)\left(\quad SK_{TA}\right)\right)\right.$$

Finally SK_{TA} is the secret session key.
The TA sends $[P_{TA}, R_{TA}, Z_{TA}, V_{TA}]$to the S_m through public communication.

5. After receiving from TA the S_m computes following:

(x)

$$W_m = h\left(\mathbf{B_{sm}}\left(\quad N_m\right), N_u \oplus N_{TA} = R_{TA} \oplus W_m, SK_m = h(N_u \oplus N_{TA} \oplus N_m)\right.$$

(xi)

$$and \quad V_{TA}* = h\Big((N_u \oplus N_{TA})\Big(\quad SK_m\Big)$$

Then, S_m check the condition ($V_{TA}* = V_{TA}$). If yes, then proceed and send [P_{TA}, Z_{TA}] it to the user publicly.

6. On obtaining [P_{TA}, Z_{TA}] fromS_m, the user calculates the following:

$L_u = h(N_u \| D_u)$, *and* $SK_u = h(N_m \oplus N_{TA} \oplus N_u) Z_{TA}* = h((N_m \oplus N_{TA}) \| SK_u)$

A user checks the condition($Z_{TA}* = Z_{TA}$) *and* $N_m \oplus N_{TA} = P_{TA} \oplus L_u$, and an affirmative answer proves that S_m and TA are authentic.

Finally, we see that (as shown in Fig. 6)

- Mutual authentication is achieved among users S_m and TA.
- Session key SK_m=SK_u are established for future communication.

5 Security analysis

To demonstrate and validate the security and correctness of our proposed protocol, we use BAN logic [3, 9, 18, 41, 43, 50], followed by Automated Validation of Internet Security Protocols and Applications (AVISPA) [5, 6, 13, 45]. Furthermore, an informal discussion has also been done in the latter part of this section.

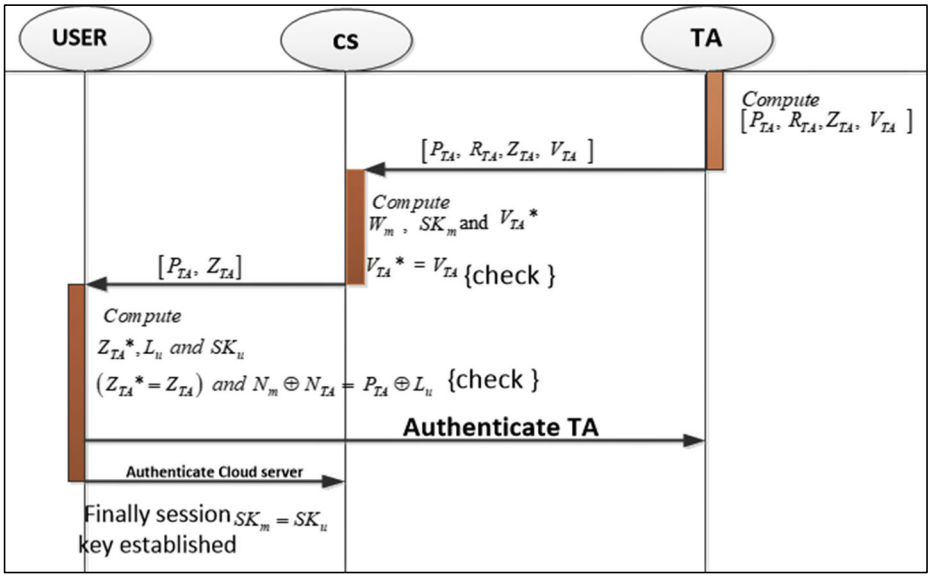

Fig. 6 TA authentication

5.1 BAN logic (a formal approach of security analysis)

It is very much essential to ensure the correctness of the authentication protocol. In Feb. 1990, MICHAEL BURROWS, MARTIN ABADI, and ROGER NEEDHAM came logically to verify the correctness of an authentication protocol, efficiency, and applicability, termed BAN Logic. Although detailed descriptions are illustrated by M. Burrows et al. [9] and more such as [3, 18, 41, 43, 50], we have tried to summarize BAN logic to understand the main concepts of BAN logic.

By using BAN logic, we can get the following important information about an authentication protocol:

1. The purpose of each protocol (Goal).
2. The cryptosystem that was used.
3. Whether secrets are used or not (other than the key).
4. Is there a guarantee that messages will arrive on time?
5. Whether protocol establishes each party's presence.
6. To remove redundancy.

The main goal of our protocol is that a user (U), Cloud server (S_m), and Trusted authority (TA) must authenticate one another. A session key (SK) is established for further communication between the server and the user.

Following are the steps used in the proposed protocol. Sentences with bold letters are common in all authentication protocols.

In First step, goals are written

G1: U believes $U \Leftrightarrow^{SK} S_m$

G2:UbelievesS_m believes $U \Leftrightarrow^{SK} S_m$

G3: S_m believes $U \Leftrightarrow^{SK} S_m$.

G4:$S_m$believes$U$believes $U \Leftrightarrow^{SK} S_m$.

G5:$S_m$believes$S_m \Leftrightarrow^{SK} TA$

G6:$S_m$believes TA believes $S_m \Leftrightarrow^{SK} TA$

G7: TA believes$S_m \Leftrightarrow^{SK} TA$

G8: TA believesS_m believes $S_m \Leftrightarrow^{SK} TA$

In the second step, the proposed protocols are idealized and written in the form of language of formal logic.

M1 (Message 1): $U \rightarrow S_m$: pid_u, TS_u, A_u, $E_u : \langle A \rangle_{(D_u)}$, $F_u : \langle N_u \rangle_{(D_u)}$, $Z_u : \langle SID_m \rangle_{\left(D_u\left(N_u\right)\right)}$.

M2 (Message 2):$S_m \rightarrow$TA: M1, $psid_m$, $J_u : \langle N_m \rangle_{(B_{sm})}$.

M3(Message3): TA$\rightarrow S_m$: $P_{TA} : \langle N_m \oplus N_{TA} \rangle_{\left(h\left(N_u\left(D_u\right)\right)\right)}$, $V_{TA} : \langle N_u \oplus N_{TA} \rangle_{(SK_{TA})}$.

M4(Message4):$S_m \rightarrow U$: $P_{TA} : \langle N_m \oplus N_{TA} \rangle_{\left(h\left(N_u\left(D_u\right)\right)\right)}$

In the third step, we identify the assumptions which show the initial state of the proposed protocol.

A1 (First assumption): $U \mid \equiv \#(N_u)$[U believes fresh(N_u)].

A2: $S_m|\equiv \#(N_u)$, A3: TA $|\equiv \#(N_u)$, A4: $S_m|\equiv\#(N_m)$,

A5: $U| \equiv \#(N_m)$, A6: TA $|\equiv \#(N_m)$.

A7: TA$|\equiv \#(N_{TA})$, A8: $U| \equiv \#(N_m \oplus N_{TA})$, A9: $U |\equiv \#(N_u \oplus N_{TA})$.

A10:$U|\equiv U \Leftrightarrow^{D_u} S_m$, A11:$S_m|\equiv(U \Leftrightarrow^{SK} S_m)$,

A12:$S_m|\equiv(S_m \Leftrightarrow^{B_{sm}} TA)$, A13: TA$|\equiv(U \Leftrightarrow^{SK} S_m)$. Many other assumptions can be added in the initial assumption, such as $S_m|\equiv U$ controls (N_u), TA $|\equiv S_m$ controlsN_m, but they are already evident, so we do not expand that.

In the fourth step (proof), the idealized forms (M1, M2… Mn) of the proposed protocol are analyzed. The basis of the analysis is based on the assumptions and BAN logic rules such as message meaning (MM), Freshness- Conjunction (FC), Belief (BL), Nonce -Verification (NV), Jurisdiction (JR), Session keys (SK). Based on the analysis in the fourth step, we reach the goals.

Using Message 1: $U \rightarrow S_m$: pid_u, TS_u, A_u, $E_u : \langle A \rangle_{(D_u)}$, $F_u : \langle N_u \rangle_{(D_u)}$, $Z_u : \langle SID_m \rangle_{\left(D_u\left(N_u\right)\right)}$ and seeing rule we get.

S1: $S_m \triangleleft pid_u$, TS_u, A_u, $E_u : \langle A \rangle_{(D_u)}$, $F_u : \langle N_u \rangle_{(D_u)}$, $Z_u : \langle SID_m \rangle_{\left(D_u\left(N_u\right)\right)}$.

Applying MM rule with S1, A11 results in S2: $S_m \mid \equiv U \mid \equiv N_u$.

Applying FC and NV rules A2, S2, results in S3: $S_m \mid \equiv U \mid \equiv N_u$ (hereN_u is a required parameter of the proposed protocol's session key.)

Applying JR rule with A14, S3 results in S4: $S_m \mid \equiv (N_u)$.

Applying SK rule with A2, S3 results in S5: $S_m|\equiv U \Leftrightarrow^{SK} S_m$**(G3).**

Applying the NV rule with A2, S3 results in S6: $S_m|\equiv U|\equiv U \Leftrightarrow^{SK} S_m$**(G4).**

Applying seeing rule on M2: S_m→TA: M1, $psid_m$, $J_u : \langle N_m \rangle_{(B_{sm})}$and results S7: $TA \triangleleft M1$, $psid_m$, $J_u : \langle N_m \rangle_{(B_{sm})}$.

Applying MM rule with A13, S7, and results in S8: $TA \mid \equiv S_m \mid \equiv N_m$.

Applying A6, S7, MM, and NV rule results S9: $TA \mid \equiv S_m \mid \equiv N_m$. ($N_m$is a required parameter of the proposed protocol's session key.)).

Applying JR rule with A15, S9 results in S10: $TA \mid \equiv N_m$.

Applying SK rule with A6, S10 results in S11: $TA|\equiv S_m \Leftrightarrow^{SK} TA$**(G7).**

Applying NV rule with A6, S11 results in S12: $TA|\equiv S_m|\equiv S_m \Leftrightarrow^{SK} TA$**(G8).**

Applying seeing rule with M3 results S13: $S_m \triangleleft P_{TA}$, V_{TA}. Results S14: $S_m \mid \equiv TA \mid \equiv (N_u \oplus N_{TA})$.

Applying FC and NV rule with A12, S14 results in S15: $S_m \mid \equiv TA \mid \equiv (N_u \oplus N_{TA})$.

Applying SK rule with A9, S15 results in S16: $S_m|\equiv S_m \Leftrightarrow^{SK} TA$**(G5).**

Applying NV rule with A9, S16 results in S17: $S_m|\equiv TA|\equiv S_m \Leftrightarrow^{SK} TA$**(G6).**

Applying seeing rule with M4: $S_m \rightarrow U$: $P_{TA} : \langle N_m \oplus N_{TA} \rangle_{\left(h\left(N_u\left(D_u\right)\right)\right)}$ results in S18: $U \triangleleft P_{TA}$.

Applying MM rule with S18, A8 results in S19: $U \mid \equiv S_m \mid \equiv (N_m \oplus N_{TA})$.

Applying FC and NV rule with S19, A10 results in S20: $U \mid \equiv S_m \mid \equiv (N_m \oplus N_{TA})$.

Applying SK rule with A10, S20 results in S21: $U|\equiv U \Leftrightarrow^{SK} S_m$ **(G1).**

Applying NV rule with A8, S21 results in S22: $U|\equiv S_m|\equiv U \Leftrightarrow^{SK} S_m$**(G2).**

Thus, the above steps are used in BAN logic to verify the authentication protocols.

5.2 AVISPA: A simulator for authentication protocols

It is vital to have tools that support the rigorous analysis of security protocols to speed up the next generation of security protocols and improve their security. For that, it detects weaknesses and establishes their correctness. In 2005, Armando et al. [6] came with a push-button tool for the

Table 2 Security features comparison

Protocols	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Xu et al. [51]	√	×	×	×	×	×	×	√	×	×	×	×	×	√	×
Wu et al. [52]	√	×	√	×	×	√	√	√	×	×	×	√	√	√	√
Chang et al. [11]	√	×	×	×	×	×	√	√	√	×	×	×	×	√	√
Amin et al. [4]	×	×	×	×	×	×	×	√	×	×	√	×	√	√	√
Sun et al. [42]	√	×	×	×	×	×	√	√	√	×	×	×	×	√	×
Challa et al. [10]	√	×	×	×	√	√	√	√	√	√	×	√	√	√	√
Proposed protocol	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

S1: User anonymity and un-traceability, S2: OGA, S3: PFS, S4: SA, S5: KSTIA, S6: SSCA, S7: PIA, S8: U-IA, S9: C-IA, S10: TA-IA, S11: Pre-verification in smart card, S12: K-KA, S13: Formal verification using AVISPA, S14: Mutual Authentication, S15: RA, ×: Insecure against a particular attack or does not support specific features, √: Secure against a specific attack or defend the specific features.

automated validation of Internet security-sensitive protocols and applications named AVISPA. The tool is used by a protocol designer who describes a security problem in the High-Level Protocol Specification Language HLPSL [20]. Moreover, detailed studies of AVISPA and its execution are available in [5, 6, 13]. In addition to this, some essential tricks to write the protocols in HLPSL language are reported in our work, followed by screenshots of our protocol result.

- First and foremost, the Role of each participant is written, which contains the Role name, declaration of local as well as constant variables, and transition.
- When the Roles of participants are defined, Roles are combined in a session.
- Define the Environment in which protocol is analyzed that contains prior knowledge about the intruder, the scenario to be executed, and the session instances to be run in parallel.
- Finally, we declare security properties of protocol to be executed.

We have passed our protocol with all four utilities provided by AVISPA tools that are following:

1. **On-the-fly Model-Checker (OFMC**): Performs protocol falsification and bounded verification.
2. **Constraint-Logic-based Attack Searcher (CL-AtSe):** It can identify type flaws and manage message concatenation associativity.
3. **SAT-based Model-Checker (SATMC**): constructs a propositional formula encoding a bounded unrolling of the IF's specified transition relation, the initial state, and the set of conditions denoting a breach of the security properties.
4. **TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols**): Regular tree languages and rewriting are used to approximate attacker knowledge. Our protocol passed all four tests.

5.3 Informal analysis of attacks

In these subsections that follow, we do a preliminary study to show the secrecy of the proposed protocol against a wide range of threats.

5.3.1 User anonymity and un-traceability

The identity of user must not be revealed during the exchange of keys among the user, cloud server, and trusted authority. User, cloud server, and trusted authority are generated for each fresh session nonce, respectively. At the same time, different timestamps (T_{TA},T_{su}andT_{sm}) validate the session. User identity is concatenated with Q_u and hashed with PID_u: ($PID_u = h(U_i \| Q_u)$). It is passed to the trusted authority. Hence, it is obvious that extracting information about the user's identity (U_i) is impossible by a third party. So, the proposed protocol preserves user anonymity and the un-traceability of the user's identity.

5.3.2 Offline guessing attacks (OGA)

The adversary can know various information saved in smart cards using power analysis or differential power analysis attacks [32, 54]. In the proposed protocol, the smart card holds $\{C_u,E_u,DP_u,\}$. Concatenation and XOR operation on user identity and password preceded by hashing results above parameters. One-way collision resistance property of hash function and uniqueness of ECC makes too hard to get knowledge about user identity and password. Wu.et. al [51] and Tsu et al. [52] are also using almost the same parameters as Amin's protocol [4]. Possibilities of the same attacks are there in these two protocols. In the proposed protocol, parameters such as user id (Ui), password, and server identity are concatenated with the private key generated on ECC. Then it is hashed with one-way hash function functions. So these consecutive steps eliminate the chance of power analysis attacks.

5.3.3 Perfect forward secrecy (PFS)

Some keys and parameters are readily available to adversaries. PFS ensures that session keys among users, cloud servers, and trusted authorities are unknown to adversaries at any cost. Here in the proposed protocol, assume an adversary wants to compute $SK_u = SK_m = h(N_m \oplus N_{TA} \oplus N_u)$.These three parameters $N_m N_{TA}$ and N_u are not available to the adversary because these parameters are always used in hashed conditions and concatenated with elliptic curve points. It is very tough to get a polynomial-time solution to this problem. Hence, the proposed protocol can provide PFS.

5.3.4 Sybil attack (SA)

In a Sybil attack, multiple accounts/nodes are created by an adversary to take over the network. In the proposed protocol, TA's elliptic curve Q_{TA} is involved in the registration and first phase of authentication. Thus ECDLP property of ECC eliminates the chance of private computing key from known parameters. So the uniqueness of private keys disaffects the harm of multiple accounts.

5.3.5 Known session-specific temporary information (KSTIA)

In KSTIA attack, session ephemeral secrets are exposed to the adversary, based on this an adversary would be able to get the session keys. In the proposed protocol, almost all information is hashed and concatenated with the private key. It is very tough to extract information from temporary information. Some researcher explains this attack with the name of Ephemeral Secret Leakage (ESL) Attack.

5.3.6 Stolen smart card attack (SSCA)

At the worst, there may be the situation that information $\{C_u, E_u, DP_{u,}\}$ are known to intruder A due to the loss or steal of a smart card. But due to h (·) protection secret entities of U, viz.,C_u,E_u and $DP_{u,}$, are not known to A. Therefore, our protocol is secure against this attack.

5.3.7 Privileged insider attack (PIA)

There is always the possibility of two types of adversaries in a system, outside adversaries and inside adversaries. An inside adversary may have privileged access to TA. In this case an adversary can get information such as A_u and PID_uwhile user is registering to TA, but $A_u = h(P_u \| d_u)$, $PID_u = h(U_i \| Q_u)$ and $\{d_u, Q_u\}$ are the points on elliptic curve, so to reveal P_u and U_i from A_u and PID_u is almost impossible. Furthermore, if the intruder fetches the information $\{C_u, E_u$ and $DP_{u,}\}$ from the smart card directly by smart card attack [43, 50], in this case also getting P_u and U_i is very difficult as discussed in stolen smart card attack.

5.3.8 User impersonation attack (U-IA)

An adversary may put P_u and U_i in authentication phase, to impersonate a user U, but to complete authentication by TA, there is need of SID_m and N_u. Getting SID_m and N_u is not possible because $N_u = F_u \oplus D_u$ and $D_u = E_u \oplus A_u$, forA_u, P_u is concatenated with d_u (point of elliptic curve) and transferred to TA in hashed form. Therefore getting N_u is too tough, similar is the condition with SID_m.

5.3.9 Cloud server impersonation attack (C-IA)

An adversary may get the identity of a cloud server (sid_m) to impersonate the cloud server, but to be authenticated from trusted authority (TA), $B_{sm} = h(psid_m \| Q_{TA})$ should be known where $psid_m$= h ($sid_m \| d_{cs}$). Here, in this case Q_{TA} and d_{cs} are the points of two different elliptic curves. Hence, by using the ECDLP property of ECC, we conclude that our protocol is safe from Cloud server Impersonation attacks.

5.3.10 Trusted authority (TA) impersonation attack (TA-IA)

To impersonate TA, an adversary may attempt to get the key $\{P_{TA}, R_{TA}, Z_{TA}, V_{TA}\}$ but at cloud server end $V_{TA} = h((N_u \| N_{TA}) \| SK_m)$and at user end $Z_{TA} = h((N_m \oplus N_{TA}) \| SK_u)$ are used for authentication, getting nonces N_u, N_{TA} and N_m in correct form is tough, also collision resistance property of hash function makes almost impossible to get correct keys. Thus, this protocol is safe from TA impersonation attack.

5.3.11 Pre-verification in the smart card

In the login phase several protocols such as [12, 28, 47], does not support smart card to verify the identity and password of user. It puts an extra burden on the server. While in the proposed protocol the smart card checks $C_u* = C_u$, $U_i* = U_i$, *and* $P_u* = P_u$ in the login phase. If it is found valid, the proposed protocol proceeds for the authentication phase. Otherwise, the session will be deferred until the right password and identity have been submitted. This means

Table 3 Comparison of the computational overhead of our protocols with related protocols

Protocols	User (U)	Trusted authority (TA)	Cloud server	Total overhead (sec)
Proposed protocol	$T_{ecm}+10Th$	$T_{ecm}+9Th$	$T_{ecm}+3Th$	$3T_{ecm}+22Th \approx 0.2002$
Wu et al. [52]	13Th	19Th	8Th	$40Th \approx 0.02$
Amin et al. [4]	9Th	10Th	19Th	$23Th \approx 0.0115$
Sun et al. [42]	$4T_{ecm}+3Th$	–	$6T_{ecm}+4T_{eca}+4Th$	$10T_{ecm}+7Th+4T_{eca} \approx 0.67775$
Li et al. [30]	$2T_{ecm}+6Th$	–	$6T_{ecm}+4Th+4T_{eca}$	$8T_{ecm}+10Th+4T_{eca} \approx 0.35667$
Chang-Le [11]	$4T_{ecm}+12Th$	–	9Th	$4T_{ecm}+21Th \approx 0.2628$
Challa et al. [10]	$2T_{ecm}+17Th+T_{fe}$	5Th	$T_{ecm}+5Th$	$3T_{ecm}+27Th+T_{fe} \approx 0.333$

that the proposed protocol saves computational and communication expenses whether there is inaccurate input or an unlawful user. As a result, the proposed protocol successfully provides pre-verification..

5.3.12 Known-key attack (K-KA)

If one session key is compromised, it does not necessarily mean that all other session keys would be as well. The accepted session key in the proposed protocol is based on three random ephemeral secrets N_u, N_{TA} and N_m, which are different for each session. Due to the difficulty of the ECDLP problem, the adversary may not be able to derive all of these at the same time. As a result, revealing one session key does not allow the adversary to learn about additional session keys.

Sometimes researchers call it No key control property.

5.3.13 Replay attack (RA)

In the authentication phase, we have used timestamps $\{TS_u, TS_m, TS_{TA}\}$, and accepted delay (ΔT) is sufficiently tiny, which makes replaying of old messages useless. Hence, our protocol is secure against replay attacks. In Table 2, the summary of the security features of recently published protocols and proposed protocols is given.

Fig. 7 Performance Comparison

5.4 Performance comparison

The computational cost of the proposed protocol is compared to that of other protocols in Table III. We have taken experimented values as reported in [1, 3, 10, 49] are following:

Time to compute hash function (T_h): 0.0005 s, time to compute ECC multiplication (T_{ecm}):0.06307 s, time to compute fuzzy extractor (T_{fe}):0.063075. Time to compute ECC point addition (T_{eca}): 0.010875 s. we have ignored time consumed in XOR and concatenation operations. At last, we have plotted a graph for better judgment between security features supported and time consumed by the proposed protocol and other protocols. The graph of summary of security analysis is presented for the ease of readers.

5.5 The efficiency of the proposed protocol

Table 2 compares the proposed protocol to other relevant existing protocols in terms of security and functional elements that are desired or required. Our protocol includes all essential security features. Existing protocols are missing several key features (as discussed in section 5.3, informal analysis of proposed protocols), such as protection against offline password guessing attacks, known key attacks, stolen card attacks, and impersonation attacks. In addition, the proposed protocol provides rigorous security analysis and formal security verification using the widely-accepted AVISPA tool. Table 3 shows that the total overhead of the proposed protocol is 0.2 s, which is much less than most of the existing protocols. Results from Tables 2 and 3 proves the efficiency of the proposed protocol. Furthermore, Fig. 7 makes our claim more claim in terms of performance comparison.

6 Conclusion

This paper first discusses crucial role of IoT devices in our daily lives, the advantages of using cloud computing in IoT devices, and authentication requirements in such an environment. Efficient authentication is the need for the hours for IoT devices. The proposed protocol is computationally very light and successfully resists attacks that are not covered by the currently existing protocol. Elliptic curve discrete logarithm problem (ECDLP) is used with one way hash function and XOR operator. ECDLP property makes the proposed protocols hard to break. The use of one way hash function and XOR operator maintains the efficiency of authentication along with secrecy. The proposed protocol is verified using BAN logic and simulated using HLPSL language for the AVISPA tool. We have added essential tricks to play with BAN logic and the AVISPA tool for any protocols related to the secure use of IoT devices. The generalized approach of the system model makes the proposed protocol implementable for the different scenarios of IoT devices such as Medical IoT, Industrial IoT, and cyber-physical systems. The proposed protocol can be extended for healthcare IoT devices and Industrial IoT by incorporating novel biometric, homomorphic encryption, and iterative learning techniques.

Acknowledgments We would like to thank the anonymous reviewers for their valuable comments, which helped us to improve the organization, content, and quality of this Manuscript.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Almuhaideb AM (2021) Re-AuTh: lightweight re-authentication with practical key Management for Wireless Body Area Networks. Arab J Sci Eng 46:8189–8202. https://doi.org/10.1007/s13369-021-05442-9
2. Alzahrani BA (2021) Secure and efficient cloud-based IoT authenticated key agreement scheme for e-health wireless sensor networks. Arab J Sci Eng 46:3017–3032. https://doi.org/10.1007/s13369-020-04905-9
3. Amin R, Biswas GP (2016) A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw 36:58–80. https://doi.org/10.1016/j.adhoc.2015.05.020
4. Amin R, Kumar N, Biswas GP, Iqbal R, Chang V (2018) A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment. Futur Gener Comput Syst 78:1005–1019. https://doi.org/10.1016/j.future.2016.12.028
5. Alessandro Armando, David Basin, Jorge Cuellar MR and LV (2001) The high level protocol specification language deliverable details. Avispa
6. Armando A, Basin D, Boichut Y, et al (2005) The AVISPA tool for the automated validation of Internet security Protocols and applications. Pp 281–285
7. Bae W, Kwak J (2020) Smart card-based secure authentication protocol in multi-server IoT environment. Multimed Tools Appl 79:15793–15811. https://doi.org/10.1007/s11042-017-5548-2
8. Banerjee S, Odelu V, Das AK, Srinivas J, Kumar N, Chattopadhyay S, Choo KKR (2019) A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of things deployment. IEEE Internet Things J 6:8739–8752. https://doi.org/10.1109/JIOT.2019.2923373
9. Burrows M, Abadi M, Needham R (1990) A logic of authentication. ACM Trans Comput Syst 8:18–36. https://doi.org/10.1145/77648.77649
10. Challa S, Das AK, Gope P, Kumar N, Wu F, Vasilakos AV (2020) Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems. Futur Gener Comput Syst 108:1267–1286. https://doi.org/10.1016/j.future.2018.04.019
11. Chang CC, Le HD (2016) A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans Wirel Commun 15:357–366. https://doi.org/10.1109/TWC.2015.2473165
12. Chen F, Tang Y, Cheng X, Xie D, Wang T, Zhao C (2021) Blockchain-based efficient device authentication protocol for medical cyber-physical systems. Secur Commun Networks 2021:1–13. https://doi.org/10.1155/2021/5580939
13. Chevalier Y, Compagna L, Cuellar J, et al (2006) A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols To cite this version: HAL Id: inria-00099882 A High-Level Protocol Specification Language for Industrial Security-Sensitive Protocols ∗. https://hal.inria.fr/inria-00100219
14. Chintala RR, Kallepalli H, Kotapati J, et al (2021) Implementing security framework for cloud based IOT network implementing security framework for cloud based IOT network
15. Do Xuan C, Dao MH, Nguyen HD (2020) APT attack detection based on flow network analysis techniques using deep learning. J Intell Fuzzy Syst 39:4785–4801. https://doi.org/10.3233/JIFS-200694
16. Dolev D, Yao AC (1983) On the security of public key Protocols. IEEE Trans Inf Theory 29:198–208. https://doi.org/10.1109/TIT.1983.1056650
17. Gadicha AB, Gupta VBB, Gadicha VB, et al (2021) Multimode approach of data encryption in images through quantum steganography. Pp 99–124
18. Haack C (2008) What is BAN logic ? Verification of security Protocols what are questions that BAN logic aims to answer ? What are limitations of BAN logic ? BAN logic ' s model of time formulas: syntax domains formulas: basic formulas formulas: idealized messages
19. Hankerson D, Menezes A (2011) Elliptic curve cryptography. In: Encyclopedia of cryptography and security. Springer US, Boston, MA, pp. 397–397
20. Internet A, Protocols S (2006) HLPSL Tutorial. In: Society

21. Iqbal W, Abbas H, Daneshmand M, Rauf B, Bangash YA (2020) An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. IEEE Internet Things J 7:10250–10276. https://doi.org/10.1109/JIOT.2020.2997651
22. Irshad A, Ahmad HF, Ramzan MS (2016) An efficient and anonymous Chaotic Map based authenticated key agreement for multi-server architecture KSII Trans Internet Inf Syst 10. https://doi.org/10.3837/tiis.2016.12.023
23. Islam SKH, Biswas GP (2012) An improved pairing-free identity-based authenticated key agreement protocol based on ECC. Procedia Eng 30:499–507. https://doi.org/10.1016/j.proeng.2012.01.890
24. Kim H, Kim D, Yi O, Kim J (2019) Cryptanalysis of hash functions based on blockciphers suitable for IoT service platform security. Multimed Tools Appl 78:3107–3130. https://doi.org/10.1007/s11042-018-5630-4
25. Kocher P, Jaffe J, Jun B (1999) Differential power. Analysis pp:388–397
26. Koppanati RK, Kumar K (2021) P-MEC: polynomial congruence-based multimedia encryption technique over cloud. IEEE Consum Electron Mag 10:41–46. https://doi.org/10.1109/MCE.2020.3003127
27. Kumar K, Kurhekar M (2016) Economically efficient virtualization over cloud using Docker containers. In: 2016 IEEE international conference on cloud computing in emerging markets (CCEM). IEEE, pp 95–100
28. Kumari S, Karuppiah M, Das AK, Li X, Wu F, Gupta V (2018) Design of a secure anonymity-preserving authentication scheme for session initiation protocol using elliptic curve cryptography. J Ambient Intell Humaniz Comput 9:643–653. https://doi.org/10.1007/s12652-017-0460-1
29. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24:770–772. https://doi.org/10.1145/358790.358797
30. Li H, Li F, Song C, Yan Y (2015) Towards smart card based mutual authentication schemes in cloud computing. KSII Trans Internet Inf Syst 9:2719–2735. https://doi.org/10.3837/tiis.2015.07.022
31. Manupriya P, Sinha S, Kumar K (2017) V⊕SEE: Video secret sharing encryption technique. In: 2017 Conference on information and communication technology (CICT). IEEE, pp 1–6
32. Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 51:541–552. https://doi.org/10.1109/TC.2002.1004593
33. Miller VS (1986) Use of elliptic curves in cryptography. In: Advances in cryptology — CRYPTO '85 proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 417–426
34. Muhammad G, Alhussein M (2021) Security, trust, and privacy for the Internet of vehicles: a deep learning approach. IEEE Consum Electron Mag 2248:1–1. https://doi.org/10.1109/MCE.2021.3089880
35. Nandy T, Yamani M, Bin I et al (2019) Review on security of Internet of things authentication mechanism. IEEE Access 7:151054–151089. https://doi.org/10.1109/ACCESS.2019.2947723
36. Naseer O, Ullah S, Anjum L (2021) Blockchain-based decentralized lightweight control access scheme for smart grids. Arab J Sci Eng 46:8233–8243. https://doi.org/10.1007/s13369-021-05446-5
37. Pete P, Patange K, Wankhade M et al (2018) 3E-VMC: an experimental energy efficient model for VMs scheduling over cloud. In: 2018 first international conference on secure cyber computing and communication (ICSCCC). IEEE:322–327
38. Rangwani D, Om H (2021) A secure user authentication protocol based on ECC for cloud computing environment. Arab J Sci Eng 46:3865–3888. https://doi.org/10.1007/s13369-020-05276-x
39. Ray PP (2017) A survey of IoT cloud platforms. Futur Comput Informatics J 1:35–46. https://doi.org/10.1016/j.fcij.2017.02.001
40. Sharma S, Kumar K (2018) GUESS: genetic uses in video encryption with secret sharing. In: Advances in Intelligent Systems and Computing. Springer Verlag, pp. 51–62
41. Sierra JM, Hernández JC, Alcaide A, Torres J (2004) Validating the use of BAN LOGIC. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 3043:851–858. https://doi.org/10.1007/978-3-540-24707-4_98
42. Sun H, Wen Q, Zhang H, Jin Z (2013) A novel remote user authentication and key agreement scheme for mobile client-server environment. Appl math. Inf Sci 7:1365–1374. https://doi.org/10.12785/amis/070414
43. Syverson P, Cervesato I (2001) The logic of authentication Protocols. Pp 63–137
44. Team A (2006) AVISPA v1. 1 user manual. Inf Soc Technol 1:1–88
45. Team A (2006) HLPSL Tutorial, A Beginner's Guide to Modelling and Analysing Internet Security Protocols Inf Technol Solut 1–52
46. Tsai JL, Lo NW (2015) A privacy-aware authentication scheme for distributed Mobile cloud computing services. IEEE Syst J 9:805–815. https://doi.org/10.1109/JSYST.2014.2322973
47. Wazid M, Das AK, Kumari S, Li X, Wu F (2016) Provably secure biometric-based user authentication and key agreement scheme in cloud computing. Secur Commun Networks 5:422–437. https://doi.org/10.1002/sec
48. Wazid M, Das AK, Hussain R, Succi G, Rodrigues JJPC (2019) Authentication in cloud-driven IoT-based big data environment: survey and outlook. J Syst Archit 97:185–196. https://doi.org/10.1016/j.sysarc.2018.12.005

49. Wazid M, Das AK, Bhat KV, Vasilakos AV (2020) LAM-CIoT: lightweight authentication mechanism in cloud-based IoT environment. J Netw Comput Appl 150:102496. https://doi.org/10.1016/j.jnca.2019.102496
50. Wessels J (2001) Applications of Ban-Logic
51. Wu F, Li X, Xu L, Sangaiah AK, Rodrigues JJPC (2018) Authentication protocol for distributed cloud computing: an explanation of the security situations for Internet-of-things-enabled devices. IEEE Consum Electron Mag 7:38–44. https://doi.org/10.1109/MCE.2018.2851744
52. Wu T-Y, Lee Z, Obaidat MS, Kumari S, Kumar S, Chen CM (2020) An authenticated key exchange protocol for multi-server architecture in 5G networks. IEEE Access 8:28096–28108. https://doi.org/10.1109/ACCESS.2020.2969986
53. Xue K, Hong P, Ma C (2014) Journal of computer and system sciences a lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J Comput Syst Sci 80:195–206. https://doi.org/10.1016/j.jcss.2013.07.004
54. Yang S, Greenberg A, Endsley M (2011) Social computing, Behavioral-Cultural Modeling and Prediction. Springer Berlin Heidelberg, Berlin, Heidelberg