Computational Sciences and Engineering 2(1) (2022) 41-52

University of Guilan

Computational Sciences and Engineering

journal homepage: https://cse.guilan.ac.ir/

A Mutual Lightweight Authentication Protocol for Internet of Things Using smart card

Rahim Asghari [a,*], Reza Semyari [a]

[a] Faculty of Electrical and Computer engineering, Malek-Ashtar University of Technology, Tehran, Iran.

ARTICLE INFO

Article history:
Received 1 May 2021
Received in revised form 17 September 2021
Accepted 14 October 2021
Available online 1 April 2022

Keywords:
Internet of Things
Security
Lightweight Authentication Protocol
Smart Card

ABSTRACT

One of the most important and essential requirements for Internet of things is security of its limited resources. The simple nature of many devices on the internet of things makes them the main purpose of a variety of attacks. To deal with these attacks, there are many protocols for authentication for internet of things. In fact, an appropriate authentication protocol plays an important role in ensuring secure communications for internet of things. In this paper, we propose an authentication scheme with key agreement on elliptic curve cryptography (ECC). The simulation results using SCYTHER show that our protocol is secure against active and passive attacks.

1. Introduction

Internet of things is a system of computing devices, mechanical and digital machines, objects or people who have unique identities and the ability to transfer data on a network without the need for human interaction with human or human with a computer. IOT has evolved from the convergence of wireless technologies, micro-electromechanical systems and the internet. There is an example of IoT in figure.1. The term Internet of things was presented by Kevin Ashton in 1999, but since 2005 until now it is growing fast.

* Corresponding author.
E-mail address: meisam.mathhome@gmail.com (R. Asghari)

http://dx.doi.org/ 10.22124/cse.2021.19498.1013

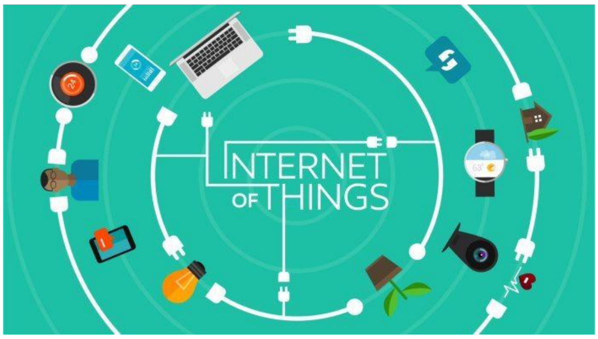

Figure 1. Internet of Things

However, making a secure connection on IOT creates a lot of challenges that need to be addressed to launch these networks on a large and commercial scale. Key management plays an important role in any communication systems. Between the sensor node in a smart environment and a remote user, it is possible to create common encryption keys in a secure way via the Internet [3, 4]. Additionally, mutual authentication between a sensor node and a remote user can prevent potential attacks. The most important security and privacy issues in IoT are shown in Fig2.

Due to the specific features of such networks, such as limited computing and processing resources, traditional key management and authentication schemes can't directly be used in the internet of things. In recent years, many authentication protocols have been proposed but they couldn't resist against the most attacks are on IOT[15].

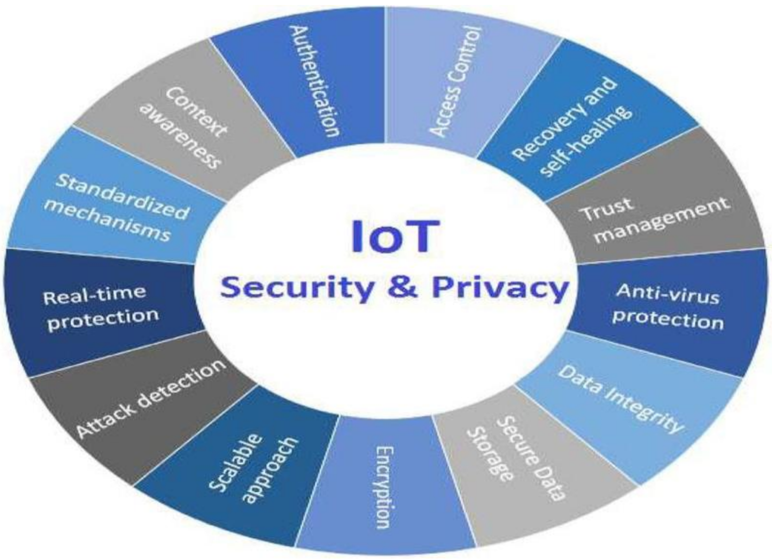

Figure 2. IoT security and privacy

In 2011, Yeh [14] presented the first user authentication protocol that uses elliptic curve cryptography in WSN environments. However, Yeh et al.'s protocol has some security weaknesses; it does not provide perfect forward secrecy. Yoon and Yoo proposed a three-factor authentication scheme in [16] based on Elliptic Curve Cryptosystem for the multi-server environment. In [17] Reddy et al. presented an ECC-based authentication protocol with anonymity for mobile computing environment but we found that their protocol was unsuccessful in achieving mutual authentication and key agreement.

In this paper, we propose an authentication protocol and key agreement for IOT on elliptic curve, hash functions, and random number generators. We will show that our proposed protocol is secure against dangerous attacks, such as denial of service, replay, and known session key attack. We also show that this protocol will have security features such as user anonymity, mutual authentication and forward secrecy [1, 2].

In the remainder of this paper, in the second part, the mathematical background of elliptic curve is presented. In third section, we will propose our protocol and in the fourth section, we will analyze and evaluate the protocol.

2. Mathematical background

In the middle of 1980s, Victor Miller and Neal Koblitz firstly used elliptic curve for cryptography. Elliptic curve cryptography computation built on finite fields, which can either choose a prime field or a binary field. Point addition and Point doubling are the basic arithmetic of elliptic curves and the basic operations of scalar point multiplication $Q = kP$, where $k \in Z$, point $Q,P \in E(Fq)$, Fq is a prime finite field. An elliptic curve is a cubic equation of the form E: $y^2 + m_1xy + m_2y = x^3 + m_3x^2 + m_4x + m_5$. Where, m_1, m_2, m_3, m_4 and m_5 are real numbers. The singular elliptic curve can be of the form $Ep(m, n): y^2 = x^3 + mx + n \ (mod\ p)$ over a prime finite field Fp, where $m, n \in Fp, p > 3$, and $4m^3 + 27n^2 \neq 0 \ (mod\ p)$. In general, the security of elliptic curve is dependent on the following hard issues.

Problem 1: Let E be an elliptic curve defined over a finite field Fq. P and Q be points in $E(Fq)$, and suppose that P has prime order n, assuming that $Q = dP$, where d is an integer from the interval [1, n-1]. The problem of determining d given the domain parameters and Q is the elliptic curve discrete logarithm problem (ECDLP) [12, 13].

Problem 2: The elliptic curve Diffie-Hellman problem *(ECDHP)* is: given an elliptic curve E defined over a finite field Fq, a point $P \in E(Fq)$ of order n, and points $A = aP$, $B = bP \in <P>$, find the point $C = abP$.

3. Proposed Authentication Protocol

In this section we will propose our protocol. For the proposed protocol to be more practical, we assume that the protocol consists of three parts of the user, the sensor node and the gateway node (*GWN*). The gateway node is in the role of the service provider. Our lightweight authentication protocol contains 4 steps, which are as follows. The used notations of this protocol are shown in Table 1.

A. System initialization phase

B. User registration phase

C. Sensor registration phase

D. Log in, authentication and key agreement.

In the following, we will explain the details of these steps.

Table1. Notations description

Notations	Description
GWN	The gateway node
ID_i and PSW_i	Identity and password of user U_i
SC_i	Smart card
ID_G	Identity of GWN
TS_i	timestamps
K_i and K_j	Random keys generated by sensor and user
USN	Counter for user U_i
DID_{GWN}	Dynamic identity for GWN
E_k	Symmetric cryptography
SK	Session key
H(.)	Hash function

A. System initialization phase

At this phase, GWN is responsible for the initialization of the system and should provide the system's required parameters from the ECDL problem. For this, GWN first chooses an elliptic equation E over a finite field Fp and a base point $P \in E(Fp)$ of order n. In the next step, GWN selects a random value X for itself, and value of $Y = X.P$ is computed. According to the second part, obtaining the value of X from the value of Y is a hard problem and can't be solved in a polynomial time. Therefore, GWN considers the value of X as its secret parameter and the value of Y as its public parameter, and publishes the values $\{E, P, Y, H(.)\}$ For the whole system.

B. User registration phase

At this phase, the user who intends to use the sensors information must register. To do this, it must send a request message that contains its own identity through a secure channel to GWN. After receiving the message, GWN first check the existence of ID_i in the database. If it exists, GWN requests a fresh identity; otherwise GWN calculates a parameter called $L_i=h(USN \,||\, h(x))$. In this term, the amount of USN is a counter to indicate how often the user is trying to access the system Finally, GWN places $\{L_i, USN\}$ values on a smart card and sends it through a secure channel to the user.

After receiving the smart card, user first enters his or her username and password, and the smart card calculates the following values.

$T_i = L_i + h(h(ID_i) \,||\, h(PSW_i))$

$e_i = h(h(ID_i) \,||\, h(PSW_i))$.

User saves these values in the smart card. Then the smart card contains $\{T_i, USN, e_i\}$ now. At the end of this step, it should be noted that GWN encrypts the value of ID_i according to the following term.

$ID_i^{\#} = ID_i + h(ID_G \,||\, X \,||\, USN)$.

A copy of the encrypted ID_i, with USN, is kept in its memory. The details of this phase are shown in Fig. 3.

C. Sensor registration phase

At this phase, sensors are in the network should be known to GWN. For this purpose, we assume that a sensor called S_j first selects a random number called b, which $b \in Z^*_{p-1}$, and computes the values $B = b.p$ and $B` = B.Y$. In the following, a request for GWN is made as follows.

$$SR=h(B \,||\, B` \,||\, TS_1 \,||\, h(S_j \,||\, K_{GWN\text{-}Sj}))$$

In the above term, the TS_1 is the sensor timestamp, S_j is the sensor identity and $K_{GWN\text{-}Sj}$ is the secret key between the GWN and the sensor node. The sensor node sends the message $\{SR, TS_1, B, S_j\}$ to GWN. After receiving the message, GWN first checks the timestamp TS_1 and then computes the value of $B`` = x.B$. GWN should also check the SR message and compute the REG_j value. Therefore, the message $\{REG_j, TS_2, B\}$ is sent to the sensor. After receiving the message, sensor should check the validity of the time stamp TS_2 and the REG_j. The details of this phase is shown in Fig. 4.

$$REG_j = h(TS_2 \,||\, B \,||\, B`` \,||\, h(S_j \,||\, K_{GWN\text{-}Sj}))$$

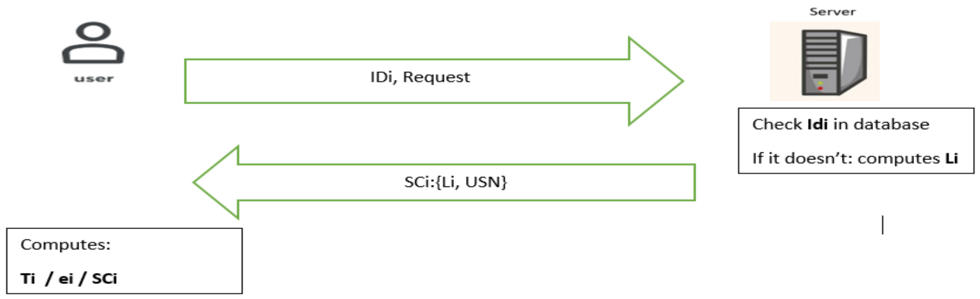

Figure 3. User registration phase

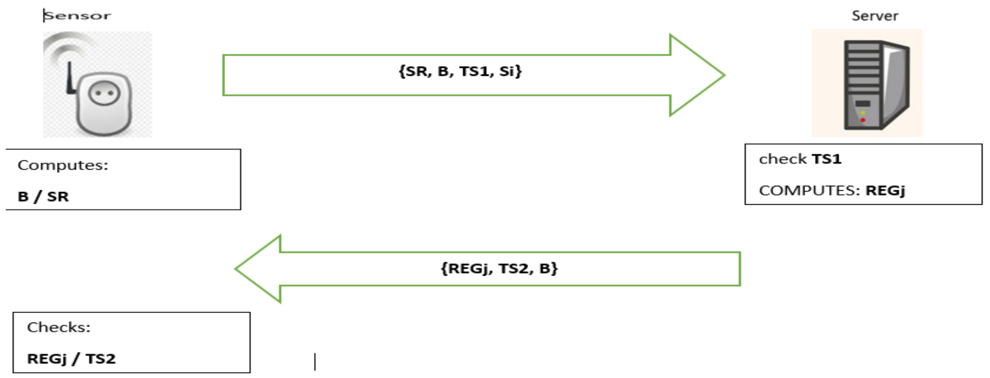

Figure 4. Sensor registration phase

C. Log in, authentication and key agreement

At this phase user re-enter his/her username and password, and the smart card computes the value of e_i^* to determine whether the user is correct or not and the smart card computes the value of L_i^* as follows:

$$L_i^*=T_i + h(h(ID_i^*)||h(PSW_i^*))$$

Now user chooses a random number $u \in Z^*_{p-1}$ and sends *{u, USN, request}* to GWN. GWN first checks the USN that is equal to the amount in its database. If they are not equal, GWN will terminate the connection, otherwise GWN will authenticate the user in the first step. After that GWN chooses a random number $d \in Z^*_{p-1}$ and computes the value $M_0=h(USN \,||\, h(u||d))$.

Finally GWN sends $\{d, M_0\}$ to the user. After receiving the message, the user must compute the value of M0, and if it is equal to the amount he has received, he can authenticate the gateway node. The user chooses a random number $c \in Z^*_{p-1}$ and computes values $C_i = c.P$, $D_i = c.Y$ and with these two numbers can computes $R_i= C_i + L_i^* + h(D_i \,||\, C_i)$. Finally, user can compute M_U, M_1 and M_2.

$M_1=h(ID_i \,||\, L_i^* \,||\, USN \,||\, D_i \,||\, C_i)$

$M_2=h(h(M_1) \,||h(Ri) \,||\, L_i^* \,||\, TS_3)$

$M_U=K_i + h(TS_3 \,||\, D_i)$

The user sends the values $\{M_1, M_U, M_2, R_i, C_i, TS_3\}$ to GWN.

After receiving the message, GWN should compare the value of M_2 *with* $h\,(h(M_1) \,||\, h\,(R_i) \,||\, L_i^{\,*} \,||\, TS_3)$ and if it the conditions were right, user would be authenticated to GWN secondly and GWN should get K_i value from M_U as follows.

$K_i= M_U + h(TS_3 \,||\, D_i)$.

In the following, GWN must have TS_4 timestamp for itself, and computes DID_{GWN}, TC_j, M_3, and M_4 values.

$TC_j=E_{K\ GWN\text{-}Sj}\,(TS_4|| \,S_j)$

$M_3=h(h(M_1) \,||\, TC_j \,||\, TS_4\,)$

$DID_{\ GWN}=ID_i+h(TS_4|| \,TC_j)$

$M_4=h(TS_4 \,||\, h(M_3) \,||\, TC_j) + K_i$. Finally, GWN sends the message $\{M_1, M_3, M_4, DID_{GWN}, TS_4\}$ to S_j . After receiving the message, S_j must check the TS_4 timestamp and compute $TC_j^{\,*}$ value and check whether it is equal to TC_j. If conditions are ok, then M_3 must be checked and if it is equal to $h\,(h\,(M_1) \,||\, TC_j \,||\, TS_4)$, GWN will be authenticated for S_j. After that, S_j computes the K_i value and ID_i from the following equations.

$K_i=M_4 + h(TS_4 \,||h(M_3) \,||\, TC_j)$

$ID_i=DID_{\ GWN}+ h(TS_4|| \,TC_j)$.

After this step, S_j selects a random number $K_j \in Z^*_{p-1}$, and computes the following values.

$S_k=h(K_i + K_j)$

$TC_j + h(K_i + K_j)=S_k`$

$M_5=h(TC_j \,||\, TS_5 \,||\, S_K`)$

$M_6=h(TC_j \,||\, TS_5)+K_j$

In the above terms, the value of SK is a session key. Finally, S_j sends the *message* $\{SK`, M_6, M_5, TS_5\}$ to GWN. After receiving the message, GWN checks the TS_5 timestamp and get the K_j value from M_6.

$K_j = M_6 + h(TC_j \,||\, TS_5)$. After finding the value of K_j, Sk has to be obtained. Finally, GWN also checks the validity of M_5 whether it is equal to $h\ (TC_j \,||\, TS_5 \,||\, SK`)$. If conditions are ok, S_j will be authenticated for GWN. On the other hand, GWN should add one unit to the USN, and then, GWN computes M_7, M_8 and M_9 using TS6 timestamp.

$M_7=h(R_i \,||\, TS_6)+ USN_{new}$, $M_8=h(R_i \,||\, h(M_1) \,||TS_6)$, $M_9=h(TS_6 \,||\, h(M_1))+K_j$.

GWN sends the message $\{M_7, M_8, M_9, TS_6\}$ to the user. The user first checks the validity of the timestamp and then checks whether the M_8 is equal to $h\ (R_i \,||\, h\ (M_1) \,||\, TS_6)$. If the conditions are ok, the user can compute the session key.

$USN_{new}= M_7 + h(R_i \,||\, TS_6)$, $K_j= M_9 + h(TS_6 \,||\, h(M_1))$. The summery of this phase is shown in Figure. 5.

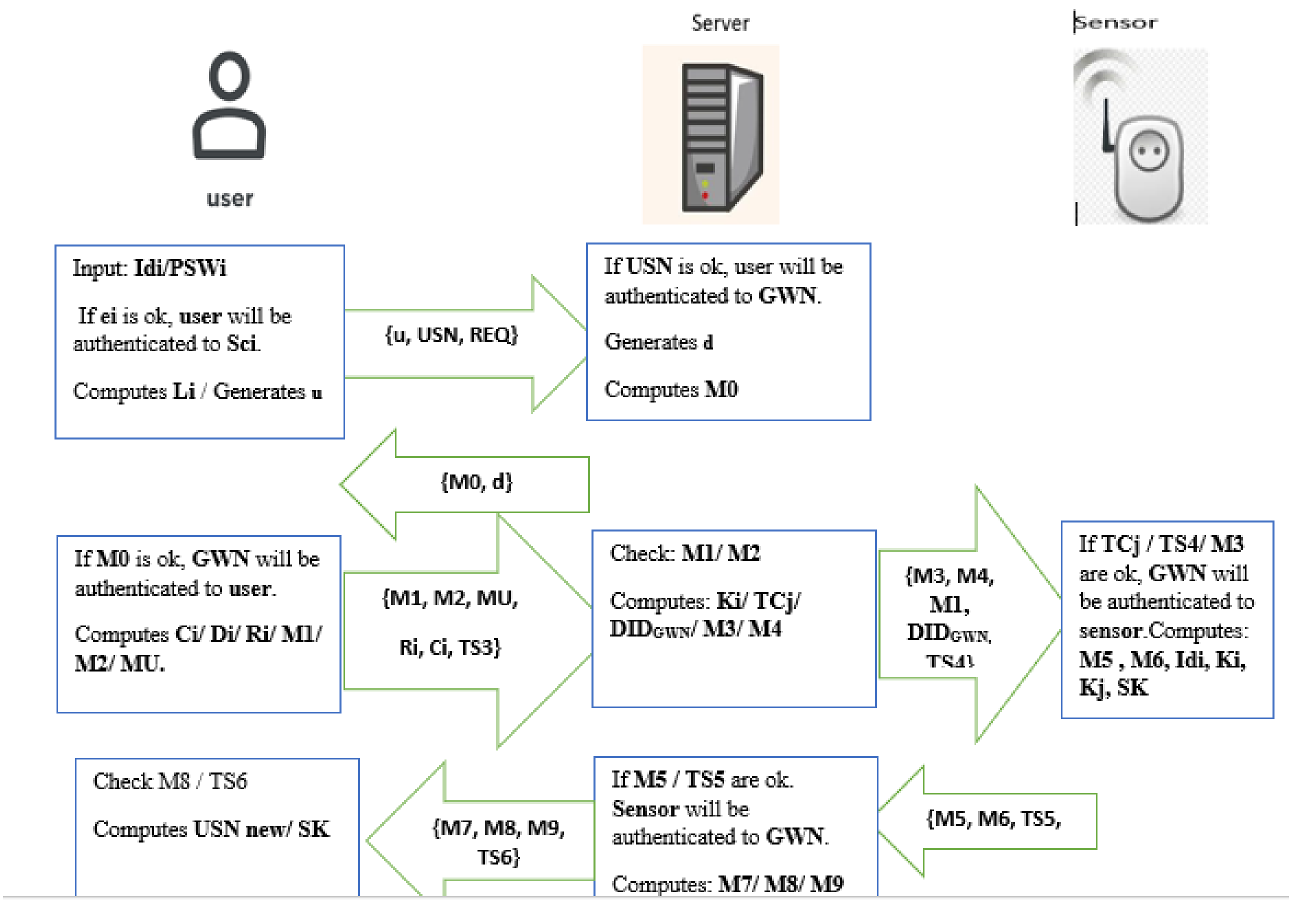

Figure 5. Log in, authentication and key agreement phase

4. Security analysis

We present the evaluation of the proposed protocol in two sections. In the first section, we will describe the security features and in the second part we use the SKYTHER software and we will describe its results.

i. Security features

A. Mutual authentication

Mutual authentication [10] in this protocol occurs. All three sides will be authenticated. GWN by checking the USN and the M2 authenticates the user. The user authenticates GWN in the second phase of authentication by checking M0. On the other hand, GWN by checking M_5, authenticates S_j and S_j by checking M_3 authenticates GWN.

B. Anonymity

In proposed protocol, we have complete anonymity [11]. All of the identities used in this protocol for the user and S_j are unrecognizable.

- Use of one-way hash function.
- Due to the use of D_i and C_i, which are based on the ECDL problem, and in the M_1 message that includes the user's identity, the attacker can't easily reach the desired identity because he must form a valid M_1 message.
- Because of the use of TC_j in the DID_{GWN} that it's based on the shared key between the sensor and GWN, the attacker can't achieve the identity of the user used in the DID_{GWN}.
- Even if an attacker accesses the GWN's private key, he can still not access the user's identity. Because in the $ID_i^{\#} = ID_i + h(ID_G \| X \| USN)$, the attacker needs to know the ID_G. Because of we have not sent ID_G in any of the phases of this protocol, so the attacker can't access the encrypted data of the GWN database.

C. Forward/Backward secrecy

This scheme has forward / backward secrecy [9]. A process is called forward security of session key, if getting a session key does not affect the security of the previous and the next keys. Because of our protocol uses random numbers for the session key and these numbers will be updated in each round and also because we have used the one-way hash function in session key and we have not sent the session key directly, the proposed authentication scheme has forward secrecy.

D. Resistance to replay attack

We claim that our scheme is resistant to replay attack [8], because of in this design, we used a USN counter and timestamps. The concept of counter is mainly used to speed up the authentication process as well as to prevent any replay attempt from any adversary.

E. Resistance to denial of service attack (DOS)

Assume that the attacker receives the {u, USN, request} message, and sends it several times. GWN calculates M_0 and sends it to the user (attacker). It should be noted that in the proposed scheme, M_0 execution are very light and do not affect the entire network. Therefore, the proposed authentication scheme is resistant to denial of service [7].

F. Known session key attack

In our authentication scheme, the agreed session is based on ECDLP, and the key of the session is a short key, so this attack will not work on this protocol.

G. Resistance to man-in-the-middle attack

Because the proposed scheme provides mutual authentication between all participating members, so this attack can't be implemented.

H. User impersonation attack

Assume that an attacker wants to introduce himself as a legitimate user, he must have a valid password in order to be able to generate the valid message $\{M_1, M_2, M_U R_i, C_i, TS_3\}$. For this purpose, the attacker should be able to calculate the R_i value, which is based on ECDL problem. Therefore, the proposed authentication scheme is resistant to user impersonation attack.

ii. Simulation Results

It is difficult to analyze the security protocols by humans, because humans mind can't consider all attack scenarios. In order to we usually look for software that makes it easy for us to do this. One of this software is SKYTHER. The advantage of SKYTHER software over other software is that it does not need to define a scenario for the application, while SKYTHER considers all different modes of attack on a protocol [5, 6]. The simulation results are shown in table 2. Some of the security features of this software are as follows:

The following features demonstrate that one scheme can resist against attacks.

Alive: has Two-way authentication feature.

Niagree: The replay attack does not apply to this protocol.

Weak Agree: Has complete authentication and good for against “No man-in-the-middle attack”.

Nisynch: It is good against impersonation and denial of service attack.

Secret x: confidentiality and integrity.

Reachable: This feature indicates that there is no pattern to track the important characteristics of the parties and the attacker has not been able to trace them.

SKR: The conditions for this claim are equal to the conditions for secret. Once this claim works correctly, the session key has not been attacked. Therefore, SKR states that known session key attack on the protocol is not applied. In table 3 and table 4, we compare our scheme with related work. These comparisons show that our proposed protocol is very suitable for IOT.

Table2. SKYTHER results

SKYTHER results	user	GWN	S_j
Secret ID_i	✓	✓	Not checked
Secret K_{GWN-Sj}	Not checked	✓	✓
Secret S_j	Not checked	✓	✓
SKR SK	✓	✓	✓
Alive	✓	✓	✓
Nisynch	✓	✓	✓
Niagree	✓	✓	✓
Weak agree	✓	✓	✓
Reachable	✓	✓	✓

Table3. Comparison of security features

Security features	Lu(17)	Yoon(14)	Arshad(15)	Choi(16)	SLAP (our scheme)
Resist replay attack	✓	✓	✓	✓	✓
Complete anonymity	-	-	-	-	✓
Mutual authentication	✓	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✓	✓
Resist known session key attack	✓	-	-	-	✓
Resist man-in-the-middle attack	✓	-	✓	✓	✓
Resist impersonation attack	✓	✓	✓	✓	✓
Resist DOS attack	-	-	-	-	✓

We have two parameters for comparison of computational costs for these authentication protocols. T_h is defined as the time for hash function cost and T_e is defined for elliptic curve cryptography point multiplication. According to [18] T_e and T_h are 0.427576 and 0.0000328 ms respectively.

Table4. Comparison of computational costs

	User	**GWN**	**S_j**	**Total cost**
Lu(17)	$4T_h + 2T_e$	$4T_h + 2T_e$	**-**	$8T_h + 4T_e$=**1.7105**
Arshad(15)	$4T_h + 2T_e$	$4T_h + 3T_e$	**-**	$8T_h + 5T_e$=**2.1381**
Yoon(14)	$2T_h + 2T_e$	$2T_h + 4T_e$	**-**	$4T_h + 6T_e$=**2.5655**
Choi(16)	$9T_h + 3T_e$	$5T_h + 1T_e$	$6T_h + 2T_e$	$20T_h + 6T_e$=**2.566**
SLAP(our scheme)	$16T_h + 2T_e$	$18T_h + 1\ T_e$	$6T_h$	$40T_h + 3T_e$=**1.284**

5. Conclusion

Internet of things is evolving every day. However, this environment is vulnerable to many security threats. Therefore, security protocols are necessary to ensure the success of these devices. In this paper, we propose an ECC based lightweight authentication for internet of things. ECC is a very efficient public key cryptography mechanism as it provides privacy and security with lower computation overhead. In the next step, we express the security features for our protocol and proved that the protocol is resistant to major attacks on the Internet of things. On the other hand, SKYTHER proves these features about our authentication protocol.

References

[1] Debiao, H, Jianhua, C, & H. Jin, (2012), " An id-based client authentication with key agreement protocol for mobile client–server environment on ecc with provable security", Information Fusion, 13-2 , 223–230.

[2] Xue, K, Ma, C, Hong, P, Ding, R,(2012), "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks", Netw. Comput. Appl, 3- 6, 316–323.

[3] Jiang, Q, Zeadally, S, Ma, J & He, D, (2017) " Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks ",IEEE Access ,2-5, 3376-3392.

[4] Granjal, J, (2015) " Security for the Internet of Things: A Survey of Existing Protocols and Open Research issues" , IEEE Communications Surveys & Tutorials , 3-17, 1294 – 1312.

[5] Yang, H, (2016), " Verifying Group Authentication Protocols by Scyther", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications.

[6] Baeten, M, (2006)" Scyther - Semantics and Verification of Security Protocols", ISBN 90-386-0804-7. – ISBN 978-90-386-0804-4 NUR 993.

[7] Gope, P, Lee, J, (2016), "Resilience of DoS Attacks in Designing Anonymous User Authentication Protocol for Wireless Sensor Networks ",IEEE Sensors Journal 1-6, 498 - 503 .

[8] Jaiswal, S, Gupta, D, (2017), "Security Requirements for Internet of Things (IoT)", Springer.

[9] Conti, M, Dehghantanha, A, Franke, F, Watson, S, (2018), "Internet of Things security and forensics: Challenges and opportunities", Elsevier.

[10] Hwang, J & Yeh, T, (2002) "Improvement on Peyravian-Zunic's Password Authentication Schemes ", IEI CE TRANSACTIONS on Communications, 2-5, 823-825

[11] Park, N, & Kang, N, (2016), "Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle", mdpi/journal/sensors.

[12] Balasubramaniam, R, Sathya, R, Ashicka, S & Senthil Kumar, S, (2016), "an analysis of RFID. authentication schemes for internet of things (IoT) in healthcare environment using elgamal elliptic curve cryptosystem", IJRTER.

[13] Lu Y, Li L, Peng H, Yang, Y. (2016), " A secure and efficient mutual authentication scheme for session initiation protocol" , Peer-to-Peer Networking and Applications 9(2), P 449.

[14] Yeh, H.L.; Chen, T.H.; Liu, P.C.; Kim, T.H.; Wei, H.W. (2011), A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors, 11, 4767–4779

[15] Melki, R., Noura, H. N., & Chehab, A. (2020). Lightweight multi-factor mutual authentication protocol for IoT devices. International Journal of Information Security, 19(6), 679-694.
[16] Yoon, E. J., & Yoo, K. Y. (2013). Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. The Journal of supercomputing, 63(1), 235-255.
[17] Reddy, A.G.; Das, A.K.; Yoon, E.J.; Yoo, K.Y. (2011), A secure anonymous authentication protocol for mobile services on elliptic curve cryptography. IEEE Access, 4, 4394–4407.
[18] Choi, Y, Lee, D, Kim, J. Jung, J. Nam, D, (2014) " Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography", Sensors 14 10081–10106.