SLAP-IoT: A Secure Lightweight Authentication Protocol for IoT Device

CH.N.S. Abhishek, Chungath Srinivasan, Lakshmy K.V., and P. Mohan Anand

Abstract Internet of Things (IoT) has evolved on a large scale and is widely being used across all the industries in various sectors. The IoT devices have a limited capacity in terms of memory and computational ability. Compared to other network applications, providing security for IoT device communication is a relatively more difficult task. The risk of getting prone to attacks can be minimized by implementing a robust authentication mechanism. To achieve it, we are proposing a lightweight authentication protocol. The security analysis was conducted using the Scyther tool, which proves that the mechanism proposed is secure against replay, session key disclosure and impersonation attacks. Moreover, the performance of the proposed protocol has been analysed and evaluated with other protocols in terms of communication cost.

Keywords IoT authentication · Scyther tool · Lightweight protocol · Two party protocol

CH.N.S. Abhishek (✉) · C. Srinivasan · L. K.V.
TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: cb.en.p2cys19003@cb.students.amrita.edu

C. Srinivasan
e-mail: c_srinivasan@cb.amrita.edu

L. K.V.
e-mail: kv_lakshmy@cb.amrita.edu

P. M. Anand
Department of Computer science and Engineering, Indian Institute of Technology Kanpur, Kanpur, India
e-mail: pmohan20@iitk.ac.in

G. Ranganathan et al. (eds.), *Inventive Communication and Computational Technologies*, Lecture Notes in Networks and Systems 311,
https://doi.org/10.1007/978-981-16-5529-6_61

1 Introduction

The use of IoT has expanded in popularity, and it has become part and parcel of our daily lives. As the number of IoT devices is increasing day by day, in 2016, there were around 4.6 billion devices, and by the end of 2021, the number would reach 13.8 billion. By 2025, it is estimated that approximately 30 billion IoT devices will exist [1]. When IoT was first introduced, less emphasis was placed on its security. As the Internet of Things is built on data that is mostly private and highly sensitive, it has the potential to be exploited and, thus, violating the user's privacy [2–4, 24]. As a result, the security of IoT devices is being prioritized. IoT devices can be secured by employing various mechanisms where authentication plays a vital role. It can help to reduce risk and guarantee that IoT devices are trustworthy. In authentication, identity of the devices is recognized and further validated. Authentication is performed in the initial phase so that communication between devices may begin, and each device can learn about the identity of the other. If authentication is not performed securely, attackers can gain access to the machines and can steal the data generated and transferred, which results in various attacks.

In IoT networks, authentication attacks are a major concern. The classification of attacks is grouped into various categories namely denial-of-service attack, masquerade attack, forging attack, man-in-the-middle attack, guessing attack, physical attack and routing attack [5]. There are various challenges with IoT authentication that must be addressed [6]. The first challenge is to cut energy costs during the process of authentication. ECC is an authentication mechanism that leverages implicit certificates to reduce energy usage and computing overhead [7] in sensor networks for distributed IoT applications. The second challenge [8] is the implementation of IoT-adapted authentication mechanisms. Distinct network architectures are based on different concepts of IoT, and authentication mechanisms need to be implemented to secure the communication [9]. Another challenge is devising an authentication mechanism capable of identifying users in their devices while avoiding persistent interaction between those components [10]. The authentication protocols in the IoT environment should essentially avail the limited amount of memory or several bits. If an enormous amount of memory is consumed, then abundant resources are utilized for implementation, and the computational cost of the protocol increases. The stand out feature in our proposed model is a lightweight authentication protocol that consumes 2948 bits.

This paper is formatted as follows. Section 2 primarily concentrates on the related works, and Sect. 3 gives a detailed overview of the protocol that is being proposed. Section 4 describes the outcomes and properties, and finally, Sect. 5 summarizes our conclusions and lists the possible future work that can be done in this area.

2 Related Work

Over the past years, many authentication mechanisms were proposed with varied architectures in the IoT environment. The main motto behind this was to develop a secure IoT system which is resilient against attacks. Kumari et al. proposed an ECC-based authentication system for IoT and cloud servers [11]. Automated Validation of Internet Security Protocols and Applications tool was employed to formally examine the security features of the suggested scheme. Security and performance review demonstrated that the proposed model is more effective, reliable and stable than existing models in the face of a variety of known attacks. A lightweight authentication protocol for IoT devices with a three-tiered architecture was proposed by Ali et al. [12]. In their mechanism, the number of positive and hostile acts was used to calculate the device's trust using a fuzzy method. The findings demonstrate the suggested protocol's advantage over other techniques in terms of attack resistance. Yang et al. proposed an authentication scheme for multi-server architecture using a smart card [20]. This mechanism combines the benefits of biometrics and password authentication. Session key disclosure attack is possible in this scheme. An authentication protocol for multi-server architecture was proposed by Li et al. [22]. Unfortunately, this protocol is vulnerable to replay and impersonation attacks. Dammak et al. proposed a decentralized mechanism for group key management employing one key distribution centre and various subkey distribution centres [25]. Totally eight algorithms are presented in this approach to address the scalability issues in group key management. Nafi et al. used a matrix based scheme for developing a lightweight key management system [26]. This mechanism is suitable for networks that contain limited resources.

For multi-server architecture, an identity-based authentication protocol is discussed using smart cards [13]. It is a dual server model which imposes varying levels of trust on both the servers, which are the service provider and the control servers. The verifier's information of the user is distributed between these two servers. They asserted that their protocol could withstand various attacks, ensure session key agreement and user anonymity. But the protocol is vulnerable to impersonation attack, stolen smart card attack and leak-of-verifier attack. The authors of [14] have proposed RSA-based two-way IoT authentication techniques using the Trusted Platform Module (TPM). The drawbacks of this mechanism are the significant key size of RSA and the large packet header. The authors of [15] proposed an elliptic curve and symmetric cryptography-based authentication and key management scheme. Additionally, it enables mutual authentication with the network control centre, besides its resistance to denial of service, replay and impersonation attacks. But this mechanism is inefficient in terms of communication and computation. Xue et al. [16] proposed a lightweight authentication and key agreement protocol for multi-server architectures based on dynamic pseudonym identity. But this protocol is vulnerable to Impersonation attack and Session key disclosure attack.

Some of the models that were proposed earlier have some gaps in them and thereby not satisfying the different attack vectors. By considering all of these mechanisms and

their associated flow possibilities, in this paper, we have proposed an authentication protocol that is resistant to most of the attacks that were previously discussed in this section.

3 Proposed Mechanism

This section introduces a new authentication protocol that involves three entities: The IoT device, service provider and the trust centre. In this mechanism, various runs occur between IoT device, trust centre and service provider to establish a session key and authenticate each other. Here the trust centre holds the responsibility to authenticate the IoT device and the service provider. Later the trust centre generates a unique key for every session and can only be used by that particular device and the service provider.

As per the Fig. 1 in the first step, the IoT device sends its identity I_i, A_i, which is the hash value generated by the concatenation of password of the device P_i, Nonce N_i and the Nonce of IoT device N_i. All the values are encrypted by the public key of trust centre K_{pt}.

In Step 2, after receiving all the values from the IoT device, the trust centre decrypts them by using its private key. Also, the trust centre calculates M_i, C_i, D_i values. M_i = Hash ($T_i \parallel X$) is the hash value generated by concatenation of T_i and X, where is the secret number given by the trust centre for each IoT device. T_i is a hash value generated by concatenating I_i and N_i, T_i = Hash ($I_i \parallel N_i$). C_i is a hash value generated by the concatenation of T_i and A_i, C_i = Hash ($T_i \parallel A_i$). Finally, D_i = XOR(M_i, C_i) is generated by performing an XOR operation on M_i and C_i, and the trust centre will store this value. The trust centre will calculate the Hash of D_i, which is $D_i\prime$. The trust centre will send $D_i\prime$, registered device acknowledgement message, the nonce of trust centre N_T and nonce of IoT device N_i received in the previous step to IoT device by encrypting them with the public key of IoT device.

In the step 3, the service provider will register itself by sending its identity I_S, along with nonce N_S, and will concatenate these two values to generate S_i which is a hash value of I_S and N_S. It will send I_S, N_S, and S_i encrypted with the public key of the trust centre.

In Step 4, the trust centre will send the nonce of the service provider N_S and nonce of the trust centre N_T by encrypting them with the service provider's public key as described in the Fig. 1.

In Step 5, the IoT device will send a login request to the trust centre by sending its nonce N_i, $D_i\prime$, value, and the nonce of trust centre N_T, received in the previous step to the trust centre by encrypting them with the public key of trust centre.

In Step 6, the trust centre will use $D_i\prime$, values sent by the IoT device in step 5. This value is compared with Hash (D_i), this D_i which is previously calculated and stored by the trust centre. If both values are matched, then the trust centre will generate the

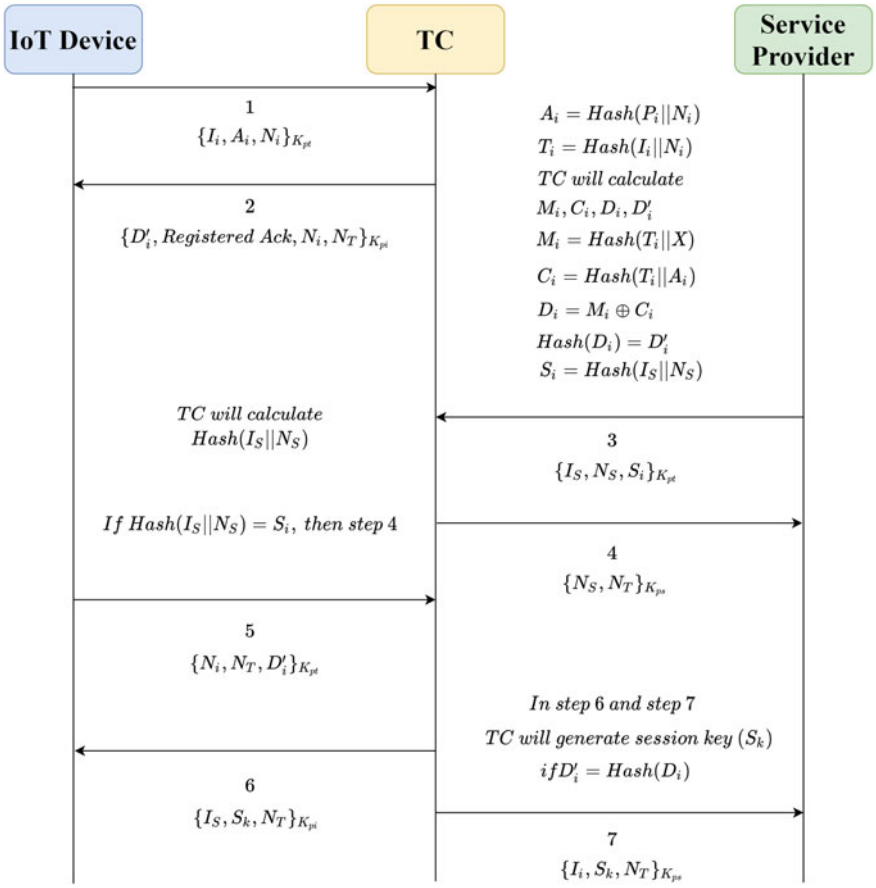


Fig. 1 Proposed protocol (SLAP-IoT)

session key S_k. Later, the session key S_k is sent to IoT device along with the nonce of trust centre N_T and identity of service provider I_S. These values are encrypted with the public key of the IoT device.

In the 7th step, the session key S_k is sent to the service provider along with the identity of IoT device I_i and nonce of trust centre N_T. These values are sent to the service provider by encrypting with the public key of the service provider.

The IoT device and service provider will decrypt the message which the trust centre sends by using their private keys and acquire the session key. This session key is used by the IoT device and service provider for the further transactions that occur in between them. The variables and their definitions used in the protocol are mentioned in Table 1.

Table 1 Variables and their definitions

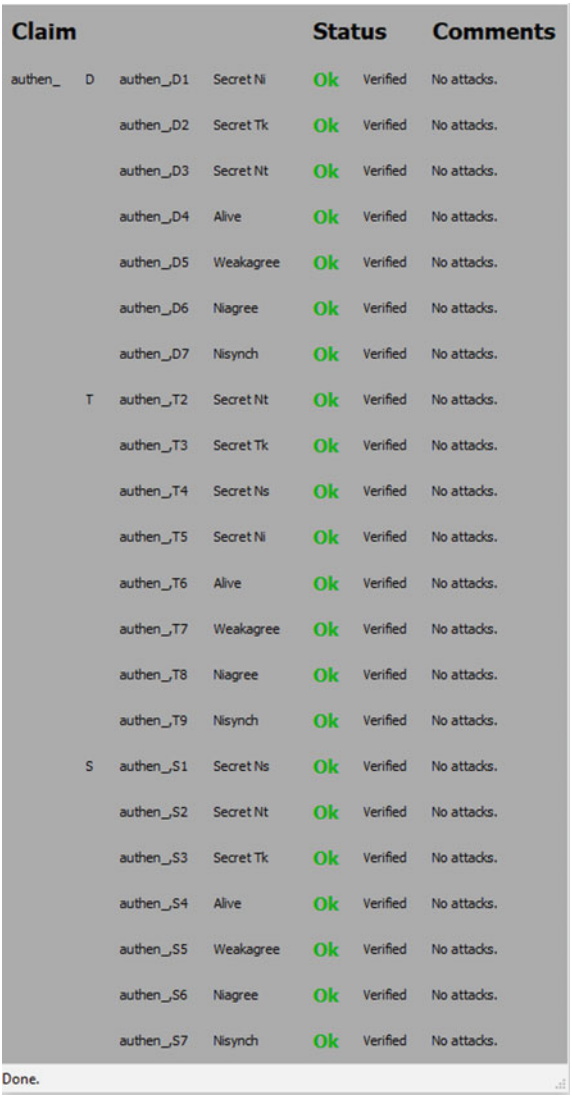
I_i	ID of i^{th} IoT Device
I_S	ID of Service provider
I_T	ID of Trust Centre
N_i	Nonce of IoT Device
N_S	Nonce of Service Provider
N_T	Nonce of Trust Centre
K_{pi}	Public key of IoT Device
K_{pt}	Public Key of Trust Centre
K_{ps}	Public Key of Service Provider
P_i	Password of IoT Device
X	Security number of device given by Trust Centre
A_i	Hash ($P_i \parallel N_i$)
T_i	Hash ($I_i \parallel N_i$)
M_i	Hash ($T_i \parallel X$)
C_i	Hash ($T_i \parallel A_i$)
S_i	Hash ($I_S \parallel N_S$)
D_i	XOR (M_i,C_i)
D_i'	Hash (D_i)
S_k	Session Key

4 Results

In this section, we will analyse the performance of the protocol and discuss the simulation results. For simulating this protocol, we have used a protocol verification tool called Scyther, which was developed in 2007 by Cas Cremersand [17]. The Scyther tool works on the adversary model proposed by Dolev-Yao [23]. It is very fast in terms of analysing the protocols formally and outperformed other state of art formal verification tools. In terms of protocol verification, this is a widely accepted tool. In Scyther [18], the verification of security properties can be done either by specifying the security properties as claims manually. If no claims are mentioned in the protocol, the tool can automatically generate the claims. The extension for protocol definition files is spdl (Security Protocol Description Language). In this tool, we can claim some security properties. After verifying the protocol, if the claims are not satisfied, then in the output console, we can see the status as Fail. Under the pattern section, scyther will generate various patterns describing the possibilities of an attack. If all the claims are satisfied, then the status is shown as OK, and the attack patterns are not generated. To model the intended security properties like Secret, Alive, Weakagree, Niagree, and Nisynch in Scyther, we use a keyword called claim.

Alive is a method of ensuring that an intended entity has completed certain acts [19]. Nisynch indicates that all messages received were sent by the communication

Fig. 2 Scyther output



Claim				Status		Comments
authen_	D	authen_D1	Secret Ni	Ok	Verified	No attacks.
		authen_D2	Secret Tk	Ok	Verified	No attacks.
		authen_D3	Secret Nt	Ok	Verified	No attacks.
		authen_D4	Alive	Ok	Verified	No attacks.
		authen_D5	Weakagree	Ok	Verified	No attacks.
		authen_D6	Niagree	Ok	Verified	No attacks.
		authen_D7	Nisynch	Ok	Verified	No attacks.
	T	authen_T2	Secret Nt	Ok	Verified	No attacks.
		authen_T3	Secret Tk	Ok	Verified	No attacks.
		authen_T4	Secret Ns	Ok	Verified	No attacks.
		authen_T5	Secret Ni	Ok	Verified	No attacks.
		authen_T6	Alive	Ok	Verified	No attacks.
		authen_T7	Weakagree	Ok	Verified	No attacks.
		authen_T8	Niagree	Ok	Verified	No attacks.
		authen_T9	Nisynch	Ok	Verified	No attacks.
	S	authen_S1	Secret Ns	Ok	Verified	No attacks.
		authen_S2	Secret Nt	Ok	Verified	No attacks.
		authen_S3	Secret Tk	Ok	Verified	No attacks.
		authen_S4	Alive	Ok	Verified	No attacks.
		authen_S5	Weakagree	Ok	Verified	No attacks.
		authen_S6	Niagree	Ok	Verified	No attacks.
		authen_S7	Nisynch	Ok	Verified	No attacks.

Done.

partner and received by another communication partner. We have implemented our proposed mechanism in the Scyther tool. As per the screenshot Fig. 2, we can see that this mechanism satisfies all the properties, and there is no scope for attacks. Here, we discuss some of the security features of the proposed protocol:

1. Resistance to Impersonation attack: An impersonation attack is not possible even if an attacker tampers the details of the IoT device because in the first step, identity I_i, A_i and nonce N_i are sent to the trust centre. Based on these values, the trust centre will calculate the values of M_i, C_i, and D_i. If the attacker tries to create an

Table 2 Proposed model results comparision with previous literature

Attack type	Yang et al. [19]	Sood et al. [12]	He et al. [20]	Xue et al. [15]	Li et al. [21]	Proposed
Impersonation attack	✓	✓	✓	✗	✗	✓
Replay attack	✓	✗	✗	✓	✗	✓
Session key disclosure attack	✗	✓	✓	✗	✗	✓

identical message as in step 1 and tries to send it to the trust centre, there will be a change in A_i, T_i, M_i, C_i, and D_i values. It will cause a mismatch so the attacker cannot impersonate a legitimate device. Also, in every step, nonce is being sent from one entity to other by encrypting them with public keys. In the next step, the concerned entity will echo the received nonce.

2. Resistance to Replay attack: In this attack, the attacker will forward the messages captured in the previous step. After validating the credentials, the IoT device will perform the login process. If the attacker tries to perform the replay attack after the IoT device is logged in, it will be of no use. Also, in this protocol, we are using nonce, which is a random number. As a result, the attacker cannot use the previously captured messages and pretend as a legitimate device because the nonce value will be updated at each step.
3. Resistance to Session key disclosure attack: Our proposed mechanism will generate the session key after validating the $D_i\prime$, value in the 6th Step. If the $D_i\prime$, value does not match, then the session key is not generated. If $D_i\prime$, value matches, then it is sent to the IoT device and the service provider by the trust centre after generating the session key. Before being sent, the session key is encrypted with the public keys of the IoT device and the service provider, respectively. The difficulty of the hash function and secret random nonces generated by the IoT device, trust centre and service provider, respectively, ensure the security of the session key in our protocol. So, session key disclosure attack is not possible in our proposed mechanism.

We conducted a comparison of our protocol's performance with other relevant publications such as Yang et al. [19], Sood et al. [12], He et al. [20], Xue et al. [15] and Li et al. [21] in terms of attack resistance and communication cost. From Table 2, it is clearly evident that our proposed protocol is better in terms of attack resistance when compared to related existing schemes where (✓) indicates that protocol resist the attack and (✗) does the opposite (Fig. 3).

We have calculated the login cost and authentication cost of our proposed protocol. In comparison with the related protocols, our proposed mechanism has more or less a similar communication cost. This is because, in our mechanism, each hash operation will consume 224 bits, considering the SHA-3 hashing algorithm, while the hash operation in other compared protocols utilizes 128 bits, and compromised hashing algorithms were used.

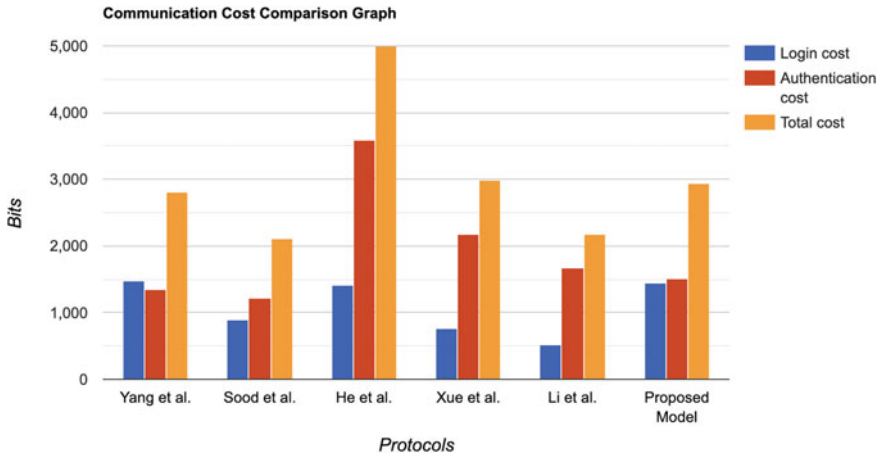


Fig. 3 Communication cost comparison graph

5 Conclusions and Future Work

In our work, we have implemented a protocol for authentication mechanism in IoT environment. The IoT environment contains devices, sensors that have limited resources in terms of memory and computation power, because of which they have a fragile security mechanism. In our approach, the trust centre plays a crucial role in authenticating, validating the IoT device, and establishing the session key between the IoT device and the service provider. The proposed protocol resists attacks like the session key disclosure, replay attack and impersonation attack. We have used a robust formal verification tool named "Scyther" to support our claim. In comparison with the related protocols, our proposed mechanism has more or less a similar communication cost satisfying the light weight property. Currently, the proposed protocol can be used for authentication when there are two parties involved. But, when it comes to group key agreement, our proposed approach is not applicable. So, as possible future work, we would like to extend this mechanism so that it can also be used in group key agreement protocols.

References

1. Statista: Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025, https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/ Accessed on April 7, 2021
2. H. Chunduri, T. Gireesh Kumar, P.V.S. Charan, A Multi Class Classification for Detection of IoT Botnet Malware, in *Computing Science, Communication and Security. COMS2 2021. Communications in Computer and Information Science*, ed by N. Chaubey, S. Parikh, K. Amin, vol 1416 (Springer, Cham, 2021)

3. M. Shriny, A. Ajisha, C. Srinivasan, Design and implementation of the protocol for secure software-based remote attestation in IoT devices, in *International Conference on Soft Computing and Signal Processing* (Springer, Singapore, 2019)
4. S.K.B. Hemanth, K.V. Lakshmy, Enhanced attach procedure for prevention of authentication synchronisation failure attack, in *Soft Computing and Signal Processing. ICSCSP 2019. Advances in Intelligent Systems and Computing*, ed by V. Reddy, V. Prasad, J. Wang, K. Reddy, vol 1118 (Springer, Singapore, 2020)
5. M. El-Hajj et al., A survey of internet of things (IoT) authentication schemes. Sensors **19**(5), 1141 (2019)
6. E.D.O. Silva, et al. Authentication and the internet of things: a survey based on a systematic mapping, in *International Conference on Software Engineering Advances* (2017)
7. H. Khemissa, D. Tandjaoui, A lightweight authentication scheme for E-health applications in the context of internet of things, in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies* (IEEE, 2015)
8. H. Khemissa, D. Tandjaoui, A novel lightweight authentication scheme for heterogeneous wireless sensor networks in the context of internet of things, in *2016 Wireless Telecommunications Symposium (WTS)* (IEEE, 2016) https://ieeexplore.ieee.org/abstract/document/9242592
9. M. Shahzad, M.P. Singh, Continuous authentication and authorization for the internet of things. IEEE Internet Comput. **21**(2), 86–90 (2017)
10. A.P. Haripriya, K. Kulothungan, ECC based self-certified key management scheme for mutual authentication in Internet of Things, in *2016 International Conference on Emerging Technological Trends (ICETT)* (IEEE, 2016)
11. S. Kumari et al., A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. J. Supercomput. **74**(12), 6428–6453 (2018)
12. A. Shahidinejad et al., *Light-Edge: A Lightweight Authentication Protocol for IoT Devices in an Edge-Cloud Environment* (IEEE Consumer Electron, Magaz, 2021)
13. S.K. Sood, Dynamic identity based authentication protocol for two-server architecture. J. Inf. Secur. **3**(04), 326 (2012)
14. T. Kothmayr, et al., A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication, **37th Annual IEEE Conference on Local Computer Networks-Workshops** (IEEE, 2012)
15. M. Qi, J. Chen, Y. Chen, A secure authentication with key agreement scheme using ECC for satellite communication systems. Int. J. Satell. Commun. Netw. **37**(3), 234–244 (2019)
16. K. Xue, P. Hong, C. Ma, A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J. Comput. Syst. Sci. **80**(1), 195–206 (2014)
17. C.J.F. Cremers, The scyther tool: verification, falsification, and analysis of security protocols, in *International conference on computer aided verification* (Springer, Berlin, Heidelberg, 2008)
18. C. Cremers, The scyther tool, www.cs.ox.ac.uk/people/cas.cremers/scyther/ [Online; Accessed on March 10, 2021]
19. G. Lowe, A hierarchy of authentication specifications, in *Proceedings 10th Computer Security Foundations Workshop* (IEEE, 1997)
20. D. Yang, B. Yang, A biometric password-based multi-server authentication scheme with smartcard, in *International Conference On Computer Design and Appliations*, vol. 5 (ICCDA, 2010), pp. 554–559
21. D. He, S. Wu, Security flaws in a smartcard based authentication scheme for multi-server environment. Wirel. Pers. Commun. **70**(1), 323–329 (2013)
22. X. Li, Y.P. Xiong, J. Ma, W.D. Wang, An efficient and security dynamic identity based authentication protocol for multi-server architecture using smartcards. J. Netw. Comput. Appl. **35**(2), 763–769 (2012)
23. D. Dolev, A. Yao, On the security of public key protocols. IEEE Trans. Inf. Theo. **29**(2), 198–208 (1983)
24. A. Bashar, Sensor cloud based architecture with efficient data computation and security implantation for Internet of Things application. J. ISMAC **2**(02), 96–105 (2020)

25. M. Dammak, et al. Decentralized lightweight group key management for dynamic access control in IoT environments. IEEE Trans. Netw. Serv. Manage. **17**(3 (2020): 1742-1757
26. M. Nafi, S. Bouzefrane, M. Omar, Matrix-based key management scheme for IoT networks. Ad Hoc Netw. **97**, 102003 (2020)