JKAU: Comp. IT. Sci., **Vol. 9** No. 1, pp: 17 – 27 (1441 A.H. / 2020 A.D.)
Doi: 10.4197/Comp. 9-1.2

Survey on Secure Identity Authentication Mechanisms for Fog-based Internet of Things (IoT) Applications

Arwa Bashanfar, Maram Alutebei and **Sabeen Tahir**

Department of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

abashanfar0002@stu.kau.edu.sa

Abstract. Fog computing enabled scalability and flexibility features to the Internet of Things (IoT) since it helped in reducing the communication traffic and enhanced the communication response time between the devices and the cloud. However, Fog computing inherited vulnerabilities for many attacks including Sybil attacks, impersonating attacks, and even Denial of service attacks. For that reason, it became essential to provide security to the Fog layer. Thus, implementing a secure authentication scheme as the first security line for a system is a necessity. In this paper, we are reviewing some recent approaches that have been developed for secure authentication in Fog computing and presenting a comparison between them considering different criteria such as user identification's methodology, the used cryptographic algorithm, and the mobility support.

Keywords: Cloud computing; Fog computing; IoT; Identity authentication protocol.

1. Introduction

Internet of Things (IoT) has been widely adopted in different applications recently, enabling objects like sensors, cellphones, or vehicles to interconnect with each other and exchange data freely to bring tremendous benefits to our daily life such as monitoring production line, controlling devices in a smart house, measuring heartbeat, informing cars about congested roads. Objects usually have limited capacity when it comes to storage, processing speed, or residual energy. These limitations are solved by using the Cloud services for processing or storing data and therefore, save the objects' resources.

Unfortunately, with the extensive growth of IoT, the amount of data sent by all connected devices to the cloud was growing exponentiality, causing network traffic and exhausting could resources. As a solution, the Fog layer was introduced as a new layer between the Edge layer and the Cloud layer. The new structure is shown in Fig. 1.

Fog computing concept was first introduced in 2012 by Cisco, for the purpose of complementing the Cloud computing functionality. It helped improving the service quality and reduced the communication delay in IoT applications [1]. However, Fog nodes are not fully secure and could face various types of attacks that threaten many components in the Fog layer. Many attacks related to user authentication can exists if the applied security mechanism is not strong enough, such as: impersonating, Sybil attack, man-in-the-middle attack, etc. Designing a secure approach to authenticate the user identity is a key factor in enhancing user's experience and encourage

users to engage in using IoT applications knowing that the application is secure, and their data won't be disclosed or leaked by unauthorized access.

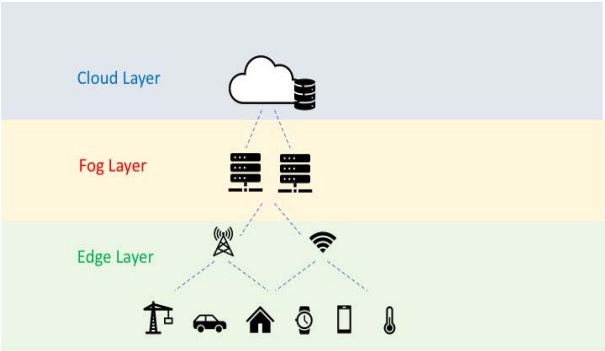

Fig. 1. IoT Fog-based Structure.

According to [2], designing a secure user authentication is entitled with challenges, many challenges are related to real-time services, where time is an important factor for IoT application users. Applying security and privacy mechanisms on IoT applications that intended to have a low-latency services could cause a delay in these services which will not be tolerated by users. Therefore, in the process of designing these mechanisms, researchers should consider a low-latency requirement and that could be assisted by adding mobility and cooperation features in these mechanisms.

The research objective is to evaluate and differentiate between recent proposed solutions that suggested mechanisms for secure identity authentication in Fog-based IoT applications. Moreover, highlight the advantages and limitations for each solution, and present a comparison between them for better understanding. The paper's main structure will be as follow: section II, recently proposed secure identity authentication mechanisms, section III will consist of a comparison and discussion of these previously introduced approaches. Section IV is a conclusion for this paper.

2. Recent Secure Identity Authentication Mechanisms

In this section, we categorized mechanisms based on the authentication type into: One-side and Mutual authentication mechanisms. A system that assumes the Fog nodes are trusted entities won't need to have both Edge device and Fog node to authenticate each other. On the opposite side, in a system were participated entities are assumed to be untrusted, then mutual authentication needs to be established to gain trust.

2.1 One-side Authentication Techniques

To create secure and difficult to impersonate identity, usually biometric is utilized. Since biometrics are almost unique for every individual, some researches benefit from biometrics and developed authentication scheme using voice pattern, fingerprint, etc. The scheme [3] applied confidentiality, integrity, and availability concepts to previously proposed face recognition framework by implementing secure identity authentication scheme, data encryption scheme, and data integrity checking scheme. It secures the communication between the different components using Advanced Encryption Standard (AES) symmetric encryption algorithm to encrypt the communication and prevent transmitted data to be visible to unauthorized users, Secure Hash Algorithm 1(SHA- 1) hash function used to ensure the integrity of the transmitted and stored data. During the processes of face identification and resolution, all the transmitted or retrieved data are encrypted using the previously issued key. Communications between Fog nodes and cloud are secure and encrypted, stored data are encrypted, and despite the added functionality to the framework, the proposed approach maintained desirable response time and network transmission amount.

Discussing limitations, data integrity checking scheme used the SHA-1 hash function,

which is vulnerable to collision attack, the integrity scheme could be stronger using the latest SHA hash function which is SHA-3. Another important point is client-side vulnerability to various attacks. Forging and impersonating may take place in the client side for publicly well-known people. This approach was not proved to be scalable: based on the size of face databases used with total of approximately two thousand images, the proposed framework not proved to be usable for large systems. It would be interesting if the experiment was run on larger database, e.g. universities to see if the approach would function the same way and maintain the same results.

Some approaches went further than authenticating end users and implemented authentication scheme for Data Centers (DC). At [4], it decomposes the Fog layer to Edge Datacenter (EDC) and Fog servers, with assumption that Cloud and Fog servers are trusted entities, unlike the EDC that are not fully trusted. For that reason, Cloud or Fog is responsible for initiating the authentication process among EDC's. Cloud will assign Identity (ID), key (Ki) linked to that specific ID, and a common shared key (Kc) for every single EDC in the network. These initialized values will be stored in the cloud's database. For the authentication process, every EDC will compose an authentication request packet consists of its ID encrypted using the shared key and broadcast the packet to all EDCs located locally in the same network. When EDC receives a request packet, it will decrypt it using the shared key, form another packet, and associate its own ID with the received ID and send it to the cloud to be verified and to check if it's registered. If found registered, both ECDs are authenticated to each other. Since all EDCs are registered at the cloud and all authentication processes are done through the Cloud, this reduces the chances for malicious attacks. However, if packets are not carefully broadcasted, collision problem may occur at some links in the network. The chances for a collision to occur will highly depend on the network structure and design. In addition to the previous limitation, if the shared key is compromised, the whole registration process should be re-initialized.

The mechanism in [5] added shibboleth protocol be- tween Fog client and Fog node to make the data access only to authenticated users, to prevent user's privacy and to protect the integrity of user's data. Shibboleth frame-work composes of two components: Service provider (SP) and Identity Provider (IdP). The public key is exchanged between SP and IdP to establish trust between them. IdP is responsible for authenticating the user's identity while SP is responsible for authorization.

The workflow of the proposed mechanism consists of several steps. It starts with Fog Node requesting an access to a service, the request is sent to the SP of Shibboleth. The SP then forwards the request to its Discovery Service (DS), which afterward asks the Fog node's ID from the Fog client. Then, the Fog client will send the Fog node's ID to DS, which thereafter is sent to SP. Then, SP requests from the Fog node authenticate Fog client. Fog node asserts that by requesting the authentication credential from the Fog client. After the client sends the credentials to the Fog node, the results are sent to SP, which finally decides to accept/decline the client's request. In this workflow, each Fog node with its clients registered with a unique ID in shibboleth's metadata. Therefore, database overloading will occur. Also, shibboleth is a single point of failure. In this state, the whole network will be in danger.

2.2 Mutual Authentication Techniques

Establishing a trust relation between participated par- ties is important to minimize security vulnerabilities in IoT. Thus, approaches

that consider a mutual authentication scheme where wildly proposed, where entities in IoT validate each other and share a common key [6], [7], [8], [9], [10] , [11].

Mutual authentication is essential in business applications. For instance, the approach in [6] was motivated by the need to secure sensor devices communications in industrial internet of things (IIoT) environments to prevent attacks that might threaten human life or cause physical damages. Users here are resource-constrained industrial devices equipped with Security Element (SE) that get authenticated by routers. It consists mainly of two phases: Registration and Authentication. During Registration, the Authentication server (AS) generates pre-shared secret key (PSK) for every router. A device sends its own identity ID to AS to be hashed with s secret key, then hash the resulted value again to get (f2i) value. Using the latest generated value, it will generate another value (f3i) by XORing it with the PSK of the corresponding router. Finally, AS sends both f2i and f3i to the device. When a device wants to get authenticated by a router, it generates a random number R1, and uses it to compute M1, M2, AID using hash and XOR operations, then sends these values to the router along with f3i value. Router will extract the used random number R1, then computes M2 and checks if it's equal to the received value. If it does match the received value, then Router will issue a random number R2, then router will hash both R1 and R2 to generate session key. Router will calculate new values M1, M2, AID then sends them to the device. Similarly, device will compute M2 and checks if it's equivalent to the received M2 value. If a match was found, the device generates a session key identical to the one generated by the router by hashing R1 and R2. Now, the device and router authenticated each other and can start communicating.

Current IoT applications demand the development of a secure authentication protocol that eliminates the need for one master key, weak pseudo-random number generators, and considers devices mobility in the IoT environment. In [7], authors proposed a less computational solution where Authentication Server (AS) facilitates the authentication process between edge devices and Fog servers and keeps a backup of keys list at different Fog areas to cope with the devices dynamically changing locations. It starts with the client requesting the Fog server's public key from AS, then creates a session key as the hash value of randomly generated number and timestamp, client will send his own public key encrypted with Fog server's public key to the desired Fog server along with the calculated session key in addition to other parameters. Fog server recalculates the session key and compares it to the one received, if they match, the client is authenticated. Same exact process is repeated by the Fog server to the client to fulfill the mutual authentication. It was proven to be resistant to some famous network attacks such as Man in The Middle (MITM) attack and traffic analysis. After the calculation of Fog server's session key, the key is transmitted to be stored in the Authentication Server in unencrypted format. While the latter two mechanisms were focusing on securing the communication between devices at the edge layer and the Fog nodes, other mechanisms took the authentication further than that, and secured the communication between user devices, Fog nodes, and the cloud.

The objective of [8] is to develop a three-party authentication key agreement scheme suitable for IoT-based health systems where users' information should maintain anonymity and confidentiality. All user's devices and Fog nodes should be registered at the cloud database. At the end of the registration, Cloud Service Provider (CSP) hands the user a smart card, and the Fog node is handed a calculated hash value R to be used during the authentication process.

Mutual authentication is carried out between the entities: user device, Fog node, and CSP, using the Fog node as a mediator. Fog node assumed to be untrustworthy, therefore, it does not store any user-related information. Authentication and key agreement are conducted using bilinear pairing. Thus, this scheme high in computational cost, also, the long key negotiation process leads to long executing time and an expensive communication cost. According to the authors, it resists famous network attacks such as dictionary attack, MITM attack, and replay attack, and it succeeds in maintaining the user's anonymity.

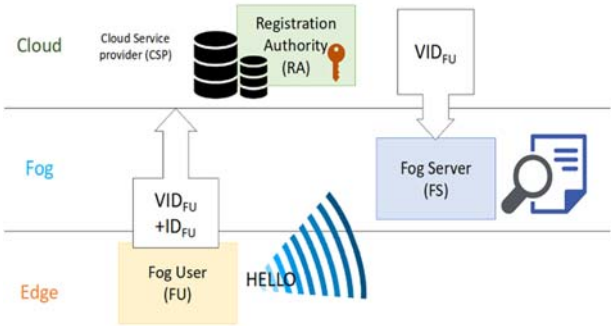

Fig. 2. Authentication scheme using virtual identity.

In [9], Fog server authenticates the virtual identity (VID) of user against the list of verified VID sent by the registration authority (RA) server in the cloud and the user will authenticate the Fog server using the challenge/response mechanism. (Refer to Fig. 2). As advantages, it provides mutual authentication between user and Fog servers. Since it uses virtual identity instead of the real identity to identify users, this tremendously contribute in preserving the user privacy and keeping the real identity hidden. In addition, cooperation between Fog servers helps avoid users' re-authentication when they change locations. As for the limitations, this scheme uses public key encryption cryptography which is known for having expansive computations that might not be suitable for devices with limited capabilities in Internet of Things environments. Moreover, the equation for calculating Fog Server private key consists of multiplication function of RA master key and the Fog Server public key, and since the public key is usually known, this eliminates the effort for an attacker to compromise the key and disclose the communications.

Other approaches were developed using the concept of Software Defined Networking (SDN), like the approach in [10], that implemented an authentication scheme using distributed Fog nodes as controllers with a built-in certificate authority (CA). In addition to the controllers, there are gateways that act as a mediator between controller and users. For the authentication process, there are three phases: First, controller authenticates gateway and creates a certificate for it. Second, users are registered in the controller's database and assigned IPv6 address and an identifier. Third: the final phase, when a user requests a service and sends the hash value of their own IPv6 address to controller via the gateway, if there address matches one of the registered at the controller database, then the user granted access to the service (Refer to Fig. 3).

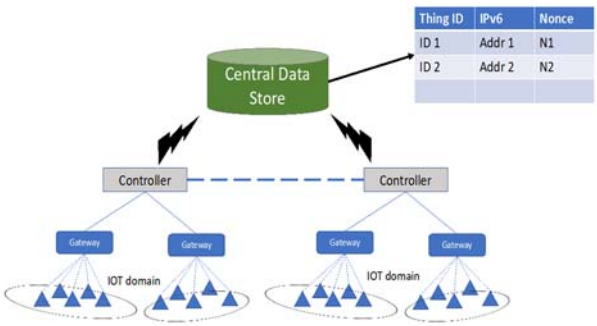

Fig. 3. Hashed IPv6 Address Authentication Scheme.

As an advantage, applying Software Defined Network added an efficiency in managing and controlling the communication which adds level of security to the scheme. Also, this scheme is considering the limited capabilities of the IoT devices and moving all the computation and storage overhead to the gateway and controller. In addition to that, this

scheme supports heterogeneity and mobility of users. For the limitation, this scheme is considered suitable only for large enterprises since SDN is usually deployed to take care of massive amount of network traffic with different purposes. (See Table 2).

Other approaches utilized Blockchain technology to benefit from its feature in managing decentralized and distributed systems. Authors in [11] proposed authentication scheme to authenticate users to gain access to IoT devices through a Smart Contract. Smart contract maintains a list of IoT devices linked to the corresponding Fog node they are managed by. In addition, it contains a list of authorized end users along with their Ethereum Address (EA) and access rights. Whenever a user wants to gain access to an IoT device, user should form an access request packet consists of its EA and the IoT device's EA. This packet is sent to the Smart Contract in which it checks if that specific user is authorized to access this IoT device or not. If authorized, an access token created that enables the user to authenticate themselves to the Fog node where the IoT device is located at, the same token is signed by the Fog node's private key and sent back to user, for the purpose of authenticating Fog node to the user. If both successfully authenticated each other, user will be granted the access to the IoT device.

This scheme is easily scalable without risking the security feature and enables central management for access request through the utilization of smart contract. How- ever, Blockchain is known for having a high processing overhead that may cause a delay in receiving packets throughout the authentication process.

3. Comparison and Discussion

In this section, we'll be highlighting the differences between the techniques using the criteria:

- User identification: how the devices at the edge layer get uniquely identified in the IoT system?
- Secure communication: since we mainly have three layers: Edge, Fog, and Cloud layers. Is the communication secured and encrypted between them all?
- Authentication: is the authentication process mutual between Fog and devices? Or is it one-sided authentication?
- Used cryptography algorithm: to secure the communication, a cryptography technique must be applied to encrypt exchanged messages.
- Mobility Support: when a device changes location and move from one area with specific Fog server to another area with a different Fog server, does it need to re-authenticate or re-register?

For every mechanism illustrated in the previous section, we'll be answering these five questions and discuss what they have in common.

3.1 User Identification

In [3], the face is used to identify users who want to benefit from Fog-based services, by assigning identity to the processed face image and store both face and identity in the cloud server via an encrypted communication. Other approaches like [5], [6], and [9] authenticate devices using alias identities instead of real device's identity to protect users' real identity from getting com- promised and used in illegal ways. For instance, in [9], virtual identity is used for user authentication instead of the real identity to preserve users' privacy and prevent attacks that might be issued if the real identity is com- promised such as the impersonating attack. Unlike the approach in [3], the virtual identities will be stored in

Registration Authority Server (RSA) at the cloud as well as in Fog nodes to reduce transmission time and reduce traffic between edge and cloud layers. Similar to[9], the approach in [4], [8] uses identity-derived generated values to create users' verifiers. Ref. [4] used ID and linked key assigned by the cloud server to every EDC in the network, these ID's and associated keys are what uniquely identifies datacenters. While in [8], create hash value using user identity and password and store it in a smart card handed to the user. Some other approaches used addresses such as [10] and [11]. In [10], the computed hash value of the assigned IPv6 address is used as an identifier instead of the traditional identifiers, the address is assigned by the controller to users once they are authenticated. Similarly, [11] uses Ethereum Address (EA) to authenticate users whether or not they are authorized. In [7], users identified using their public key which isn't the safest option since Public keys are usually are announce to the public and therefore could be easily compromised.

3.2 Secure Communication

To maintain confidentiality, techniques should secure connections between different layers in IoT by applying cryptography, but in [3] and [4] the communication was encrypted only between Fog layer and cloud servers using symmetric encryption algorithms. Moreover, [6] and [7] secure connection between Fog node and edge device. In [8],[9],[10], and [11] the communication was securely encrypted between the three entities cloud, Fog node and edge which make these techniques high in security level. However, the mention of an encrypted connection was absent in [5].

3.3 Authentication

Authentication of two entities communicating is mutually established in the schemes [6]–[11] between edge device and Fog server in which Fog server authenticate the virtual identity (VID) of an edge device against a pre-sent list of verified VID sent by the registration authority server located in the cloud and the user authenticate the Fog server using the challenge/response mechanism. While authentication in [3]–[5] is one sided authentication only in which the Fog node/cloud server authenticates the edge devices. Also, in [10] the authentication is mutual between the user and the gateway, the user only trusts the gateway if they are assigned a certificate by the controller, and vice versa, the gateway only trusts the user if they are registered at the controller DB. Moreover, in [11] end users are authenticated to the Fog node through the use of the token given by the Smart Contract, the Fog will use the same token and sign it with its private key to be authenticated to the end user.

3.4 Cryptographic Algorithm

To achieve confidentiality, cryptographic techniques were implemented in all proposed schemes except for [5]. In [3], [4], [6] and [7] Symmetric Encryption used to encrypt the transmitted data. In contrast, the schemes [8]–[11] used Asymmetric encryption which is when compared to the first technique will require more computational capabilities.

3.5 Mobility Support

Supporting mobility plays the main role in avoiding user re-authentication which causes an increase in transmission time which affects the quality of service provided by Fog nodes. In [9], whenever a Fog user wants to change location and move from a Fog server area to another, this movement is accomplished without the Intervention of an Authentication Server due to the fact that Fog servers cooperate to find if a certain VID is included in any Fog server's list of verified VIDs. In [7], at the Fog layer, the authentication server contains all the public keys of both Fog and edge devices, and the authentication server has several back up at different locations to avoid the need of re-

authentication. Unlike the approaches [3] and [6] where a device has to re- authenticate itself if it changed location. Similar to [9], in [4], [5], [8], and [10] users can move from one node to the other and still be authenticated without having to re- authenticate themselves. In contrast to all the previous, the approach in [11] doesn't need Fog layer to support mobility due to the existence of Smart Contract that manages all the authorization requests, See Table 1.

Table 1. Highlight of comparison between techniques.

Ref No	User Identification	Secure Communication	Authentication	Cryptographic Algorithm	Mobility Support
[3]	Face identity	Between Fog nodes and Cloud	One-side Authentication	Symmetric encryption (AES)	Yes
[4]	ID || Key	Between Fog nodes and Cloud	One-side Authentication	Symmetric encryption	Yes
[5]	Unique ID	-	One-side Authentication	-	Yes
[6]	AID= h(R) ⊕ ID	Between Fog and Edge	Mutual Authentication	Symmetric encryption	No
[7]	Public Key	Between Fog and Edge	Mutual Authentication	Symmetric encryption	Yes
[8]	h_2 = (ID|| s || h_1(ID||PW))	Between Cloud, Fog server and Edge.	Mutual Authentication	Asymmetric encryption (ECC)	Yes
[9]	Virtual identity	Between Cloud, Fog server and Edge.	Mutual Authentication	Asymmetric encryption	Yes
[10]	H(Ipv4)	Between Cloud, Fog server and Edge.	Mutual Authentication	Asymmetric encryption (ECC)	Yes
[11]	Ethereum Address	Between Cloud, Fog server and Edge.	Mutual Authentication	Asymmetric encryption	No

Table 2. Advantages and limitations for each mechanism.

Ref No.	Advantages	Limitations
[3]	▪ Secure communication between FN and cloud. ▪ Stored data are encrypted. ▪ Maintain desirable response time and network transmission amount.	▪ Uses SHA-1 hash function. ▪ Client-side vulnerability. ▪ Not proved to be scalable.
[4]	▪ Reduces the chances for malicious attacks since all EDC are registered at the cloud.	▪ Collision problem may occur at some links in the network, due to the packets constantly sent to the Cloud for authentication.
[5]	▪ Protecting user's data integrity. ▪ Protecting users' privacy. ▪ Providing an authentication to the network.	▪ Database overloading. ▪ Single point failure.
[6]	▪ Identity Confidentiality. ▪ Replay attack resistance ▪ Modification attack resistance. ▪ Impersonation attack resistance.	▪ XOR vulnerability to Plaintext_Known attack.
[7]	▪ Supports mobility of users. ▪ Doesn't communicate session keys. ▪ Resistance against many network attacks.	▪ Session keys are stored at the Authentication Server in unencrypted format.

[8]	▪ Maintain user' anonymity. ▪ Protection against: dictionary attack, replay attack, and MITM attack	▪ Fog node works as a mediator even though it's not trustworthy. ▪ High computational cost. ▪ High communicational cost. ▪ Long executing time.
[9]	▪ Provide mutual authentication. ▪ User is anonymous (real identity is hidden). ▪ Cooperation between Fog servers (no need to re-register).	▪ PK cryptography: ECC vs. expensive computation. ▪ Easy Fog Server's private Key creation: Sfs= s. Pfs
[10]	▪ Support devices heterogeneity. ▪ Support mobility of users. ▪ All computations complexity and storage are on gateway and controller.	▪ Not suitable for small IoT network with limited devices.
[11]	▪ Provide confidentiality, integrity, and availability. ▪ Scalable solution	▪ The use of Blockchain applies processing overhead.

From the previous discussion points, we can notice that most of these mechanisms support mutual authentication where both Edge devices and Fog nodes exchange messages and verify each other, thus, establishing trust relationship and start communicating. Applying mutual authentication may add a delay to the application due to the time it takes to exchange messages between two entities. Mutual authentication is suitable for scenarios where establishing trust between entities is a priority. Eliminating re-authentication process whenever the user changes location by supporting mobility feature is important to improve the service quality and to motivate users' participation.

4. Conclusions

In this paper, we presented an analysis of recent approaches developed to secure the authentication process in Fog-based IoT applications. Comparing the techniques, in terms of the user's identification, the cryptographic algorithm used, communication security, and mobility support. We found that the majority uses mutual authentication and supports mobility. Mutual authentication could take a longer time that the one- sided authentication, therefore it's suitable for application where Fog nodes assumed to be untrusted. On the other hand, supporting mobility eliminates the need to re-authenticate every time the user changes location, this will enhance the quality of service and save a lot of time.

References

[1] **Bonomi, F., Milito, R., Zhu, J.** and **Addepalli, S.,** "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12*, Helsinki, Finland, 2012, p. 13, doi: 10.1145/2342509.2342513.

[2] **Ni, J., Zhang, K., Lin, X.** and **Shen**, **X.** "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 601–628, 2018, doi: 10.1109/COMST.2017.2762345.

[3] **Hu, P. Ning, H., Qiu, T., Song, H., Wang, Y.** and **X. Yao,** "Security and Privacy Preservation Scheme of Face Identification and Resolution Framework Using Fog Computing in Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1143–1155, Oct. 2017, doi: 10.1109/JIOT.2017.2659783.

[4] **Puthal, D., Ranjan, R., Nanda, A. Nanda, P., Jayaraman, P. P.** and **Zomaya, A. Y.,** "Secure authentication and load balancing of distributed edge datacenters," *Journal of Parallel and Distributed Computing*, vol. 124, pp. 60–69, Feb. 2019, doi: 10.1016/j.jpdc.2018.10.007.

[5] **Zahra, S. *et al.*,** "Fog Computing Over IoT: A Secure Deployment and Formal Verification," *IEEE Access*, vol. 5, pp. 27132–27144, 2017, doi: 10.1109/ ACCESS.2017.2766180.

[6] **Esfahani, A. *et al.*,** "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 288–296, Feb. 2019, doi: 10.1109/JIOT.2017.2737630.

[7] **Pardeshi, M. S.** and **Yuan, S.-M.,** "SMAP Fog/Edge: A Secure Mutual Authentication Protocol for Fog/Edge,"

IEEE Access, vol. 7, pp. 101327–101335, 2019, doi: 10.1109/ACCESS.2019.2930814.

[8] **Jia, X., He, D., Kumar, N.** and **Choo, K.-K. R.,** "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Netw*, vol. 25, no. 8, pp. 4737–4750, Nov. 2019, doi: 10.1007/s11276-018-1759-3.

[9] **Ben Amor, A., Abid, M.** and **Meddeb, A.,** "A Privacy-Preserving Authentication Scheme in an Edge-Fog Environment," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, Hammamet, Oct. 2017, pp. 1225–1231, doi: 10.1109/AICCSA.2017.57.

[10] **Salman, O., Abdallah, S., Elhajj, I. H. Chehab, A.** and **Kayssi, A.,** "Identity-based authentication scheme for the Internet of Things," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, Messina, Italy, Jun. 2016, pp. 1109–1111, doi: 10.1109/ISCC.2016.7543884.

[11] A**lmadhoun, R., Kadadha, M., Alhemeiri, M., Alshehhi, M.** and **Salah, K.**, "A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, Aqaba, Oct. 2018, pp. 1–8, doi: 10.1109/AICCSA.2018.8612856.

دراسة استقصائية حول الآليات الآمنة للتحقق من الهوية في تطبيقات إنترنت الأشياء القائمة على الحوسبة الضبابية

أروى باشنفر، و **مرام العتيبي**، و **سابين طاهر**

كلية الحاسبات وتقنية المعلومات، جامعة الملك عبد العزيز، جدة، المملكة العربية السعودية
abashanfar0002@stu.kau.edu.sa

المستخلص. زوَّدت الحوسبة الضبابية انترنت الأشياء بعدة مميزات، من أهمها: القابلية للتوسع والمرونة، وذلك عن طريق تقليل ازدحام شبكة الاتصالات وتحسين وقت الاستجابة بين الأجهزة المتصلة والخدمة السحابية. من جهة أخرى، فإن حوسبة الضبابية قد ورثت نقاط ضعف للعديد من الهجمات، بما في ذلك هجمات سايبل (Sybil Attack)، وانتحال الهوية (Impersonating attacks)، وحتى هجمات الحرمان من الخدمة (Denial of Service Attack). ولهذا السبب؛ فإنه من الضروري إضافة خاصية الأمان إلى طبقة الحوسبة الضبابية عن طريق تأمين خاصية التحقق من الهوية، والتي تعتبر خط الدفاع الأول المهم في أي نظام. في هذه الدراسة الاستقصائية، نقوم بمراجعة بعض الآليات الحديثة التي تم تطويرها كآليات آمنة للتحقق من الهوية في الحوسبة الضبابية، وعلى إثر هذه المراجعة نقدم مقارنة بين هذه الأليات بناءً على معايير محددة كالطريقة المتبعة لتعريف المستخدمين، خوارزمية التشفير المستخدمة، ودعم التنقل السلس للأجهزة.

الكلمات المفتاحية: حوسبة سحابية؛ حوسبة الضباب إنترنت الأشياء؛ بروتوكول توثيق الهوية.