Accessing Cloud Services Using Token based Framework for IoT Devices

Dr.N. Sudhakar Yadav
Department of Information Technology, VNR Vignana Jyothy Institute of Engineering and Technology, Hyderabad, Telangana, India.

Dr.Ch. Mallikarjuna Rao
Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, Telangana, India.

Dr.D.V. Lalitha Parameswari
Department of Computer Science and Engineering, G. Narayanamma Institute of Technology Science (For Women), Hyderabad, Telangana, India.

Dr.K.L.S. Soujanya
Department of Computer Science and Engineering, CMR College of Engineering & Technology, Kandlakoya, Hyderabad, Telangana, India.

Dr. Challa Madhavi Latha*
Department of Computer Science and Engineering, CMR College of Engineering & Technology, Kandlakoya, Hyderabad, Telangana, India.

Received March 17, 2021; Accepted July 14, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V18I2/WEB18316

Abstract

Nowadays cloud environments are used by many business service sectors like healthcare, retail marketing, banking, and many business fields. At the same time, the usage of Internet of Things (IoT) devices in different sectors also increasing tremendously. So, there is a general problem for securing any business service in enterprise cloud environments restricting by only authorized devices. We are proposing cryptographic techniques with the help of a token-based framework by enabling a secure handshake between consuming applications and the source business service which aims to authorize the target end consumers of the respective business service. The proposed work aims to achieve the desired secure handshake so that any consuming application or device requests the desired business service with a secret token and an input combination. The source business service creates a secure token using any latest robust cryptographic algorithm on the above input combination and returns the token to the consuming application. The consuming application requests to the source business service, it must pass the above token which if validated then only would receive the required data.

Hence, in this paper, we propose the delegation of the authorization task to the end consumers, who are responsible to fetch the security tokens and use them in their application lifecycle.

Keywords

Clouds, Security, Privacy in Cloud, Token, Privacy, IoT.

Introduction

Internet of things means connecting the millions of devices having with sensors, actuators, and different communication devices. With the cloud platform where millions of devices are getting connected we need some mechanism to validate the request after any device is registered or authenticated. So, to authorize the subsequent request we have implemented the token-based authorization which needs to pass with every request. In this way, we can secure any request originating from any device in the IoT domain without any third-party tools. It will help us to protect from the malicious attack as the generated key will be hashed key which is one-way encryption and cannot be decrypted.

To update the regular life of people with connected devices, we are connecting and using the services of cloud computing. The smart devices however use the services of the cloud via Open Data Protocol (OData) Restful APIs, and the authentication and authorization of millions of devices is the challenge for the server (Madhavi & Soujanya 2021a). In addition to this, the security and privacy of data of devices is also an additional challenge. In this work, we are going to present secure token-based authorization service architecture. With the help of tokens, the devices can access the services of the cloud. The token lifetime is valid for a specified amount of time, afterwards, it will not be worked. Again the token has to raise with the help of the authentication center. This token was also dispatched by the third-party center to improve the security and reliability features. Furthermore, the token is valid only within a period or it will not work after the token count exceeds the threshold defined by the system, thereby lowering the devices' risk of being hacked. The token cannot be decrypted as it has been created by hashed mechanism and its one-way encryption. The framework proposed in this article is applied to any connected devices. The advantages of this framework are practical and secure. The interoperability between cloud services and IoT devices is realized by Web services. Specifically, OData-based Restful API (application programming interface) is the most common way to realize this in the services. OData systems usually have communication through HTTPS (Secured Hypertext Transfer Protocol).

Related Work

Malani, Saurav, et al., (2019) proposed a certificate-based device access control technique to restrict the security attacks in the IoT environments. Chaudhry & Shehzad Ashraf, (2020) improved the certificate-based scheme and provided security for device impersonation as well as man-in middle attacks.

Tamanna, (2020) developed a model based on key management and symmetric and asymmetric-based approach for secure IoT environments. Anajemba (2020) proposed a model based on physical layer security in IoT environments. Yadav et al., (2018) proposed a framework based on real sense architecture that integrates wireless sensor networks and IoT environments (Madhavi et al., 2020). Yadav et al., (2019) developed a framework for IoT in the health care domain. Shekh-Yusef et al., (2015) proposed an HTTP digest authentication scheme. M. Jones et al [8] discussed how to transfer the information between two parties with the help of JSON Web Token. Campbell et al., (2015) used Security Assertion Markup Language in their authentication process. Kevin E. Foltz et al., (2016) proposed a secure technique for accessing digital objects (Madhavi et al., 2020). Yan & Lu, et al., (2019) discussed the limitation for the transport layer, especially in content delivery networks. To authenticate the proxy server they have proposed a delegation method that is based on the token. Kubovy et al., (2016) proposed CAAS (Central Authentication and Authorization System). This system uses an encryption-based token management system. This one leads to securing the transmission without depending on the server-side party (Madhavi & Soujanya 2021b). Indu et al., (2017) worked on providing security for web services on the cloud. For this, they proposed the method which is based on fine-grained authentication technique in addition to security assertion markup language for cloud web services (Madhavi et al., 2020). Ethelbert et al., (2017) proposed authentication with the help of JSON Web Tokens. Their main aim is providing authentication access to cloud resources especially SaaS ones. Proposes security for open source cloud platforms. In their paper, they discussed token-based systems, and in addition to this biometric systems were also introduced for open source cloud platforms. Yamin et al., (2019), proposed a blind approach for providing the security of the Internet of Things. They found the effectiveness of the system with simulation results. They have applied the blind approach to various use cases such as smart homes, smart transportation, and e-health (Madhavi & Soujanya 2021a). Badii et al., (2020), proposed a snap4city solution that provides the full range of security from IoT Devices to IoT applications (Madhavi & Soujanya 2021b). Martínez-Peláez, et al. (2019), proposes mutual authentication technique on IoT Cloud environment. Their

paper represents that, the method restricts the various attacks in the environment and also providing enhanced security compare to the mentioned one in their paper. Yuliana & Mike. et al., (2019) proposed a signal strength exchange security method for Internet of Things Devices. They have shown their method is efficient in terms of communication overhead and computing time. Chikouche et al., (2019) introduced a code-based authentication protocol for Internet of Things environments, and also they developed it in such a way that the system opposes the quantum attacks. Mahbub et al., (2020) have done a detailed analysis on security on IoT applications. They have investigated different domains like fog and edge computing and machine learning to widen the IoT security. Das et al., (2018) discussed several security necessities for IoT environment, authentication, Key management and also discussed challenges of IoT security protocols. Yadav et al., (2018) proposed a healthcare system that is based on the cloud. Here, the authors compared the results with the help of big data analytics. Sudhakar et al., (2020), proposed Domestic animals health monitoring based on IoT. In this, the authors applied various machine learning algorithms also to predicting heart abnormalities in domestic animals.

Proposed Work

In today's world the main focus area is to protect your every request which can originate from any device, it can be a medical device, fitness device, In Today's world, IOT is in every device and it is connected with every device and to protect the web request we communicate over the internet, so it is very important even after the device authenticates the subsequent request and response date can be protected over the request, by any means it should not be compromised, since it deals with the privacy of data of an individual. There is a mechanism to securely authenticate your devices. In this paper, the proposed approach helps to secure your subsequent request over the network as in the IoT world every device is connected, and to avoid the malicious attack, keeping in mind the data shared among the multiple devices in a cloud platform can be secured using the proposed approach.

Steps for Token-based Authorization Implementation

This article proposes a token-based framework and provides private access control and an authorization framework which is integrated by invoking create token request to the cloud services. The token reads the session information along with the other information and it will generate the HASHED KEY based on the logged in users and the registered devices details, server application which request for the token has been already protected by SSO

or OAUTH, the communication to create or to request for the token is server to server. So, any request that arises from application one to the token-based application services is always secured. The hashed key will now be shared in the subsequent OData request and will be used to validate the token. The generated token is hashed using the SHA-256 algorithm, which is a one-way hash and it can't be decrypted. Using this approach, we can easily exchange information over the network in a secure way. The token can be stored in the database along with the unique identifier which will be validated against the input token and proceed on success. The token will be deleted after regular intervals via the scheduler job as we store the token along with the last access token time which can be used to delete the token at regular intervals.

The client should send the authenticated data only one time and there is a single chance of access to the database of the user. The remaining transactions accessing done without touch the user data or information. The performance of this approach improved because of not depending on every transaction access with user permission. In the following figure 1, the system security part which is key to the system was missed. In this process, we require a token management part that can store the issued tokens information like in figure 1. When the client raised the request to the server, then the server should verify the validation of the token with the token management system. If any issue would come, the revocation of the token done and the same thing dispatched to the token management by the server.

The working principle of the proposed approach for token-based authorization is explained as below:

1. The client first registers the device, it can be any device. First, authenticate the request originating from the browser or any devices.
2. Once the devices are registered, the client sends an access token request to the Token Management Service.

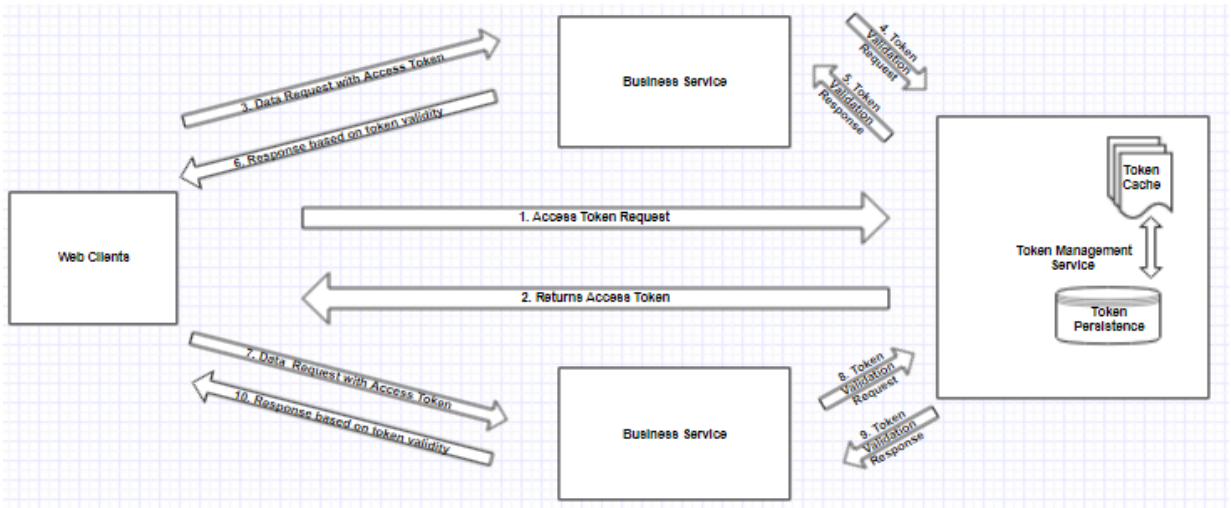


Fig. 1 Service access using Token-based authorization

3. Token Management Services generates a unique hashed token, which is hashed using the SHA-256 also it is the one-way hash mechanism and it is extremely difficult to decrypt the HASHED key. After the generation of the token, it stores the token in the database as well as a cache at the application level for performance improvement and returns the token to the client for subsequent requests.

$$\text{SHA-256: } \begin{matrix} B^1U \ldots UB^{2^{64}} & \rightarrow & B^{256} \\ M & \mapsto & H \end{matrix}$$

The algorithm uses the functions:

$$\text{Ch}(A, B, C) = (A \wedge B) \oplus (A \wedge C),$$
$$\text{M aj}(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$

4. Generated tokens are periodically cached at the application level & thereafter the tokens will be periodically deleted from the cache as well as from the database token table, based on the last access time. So, that the table size should not grow exponentially.
5. When a client requests the data, it sends the token along with the request object.
6. The business service first validates the token at the filter class from the cached Object, if the token is available then the required data is served in the request.
7. If the token is not available in the cache, then the client needs to send a token to generate the request again.

Sequence Diagram to Represent Token-based Authorization

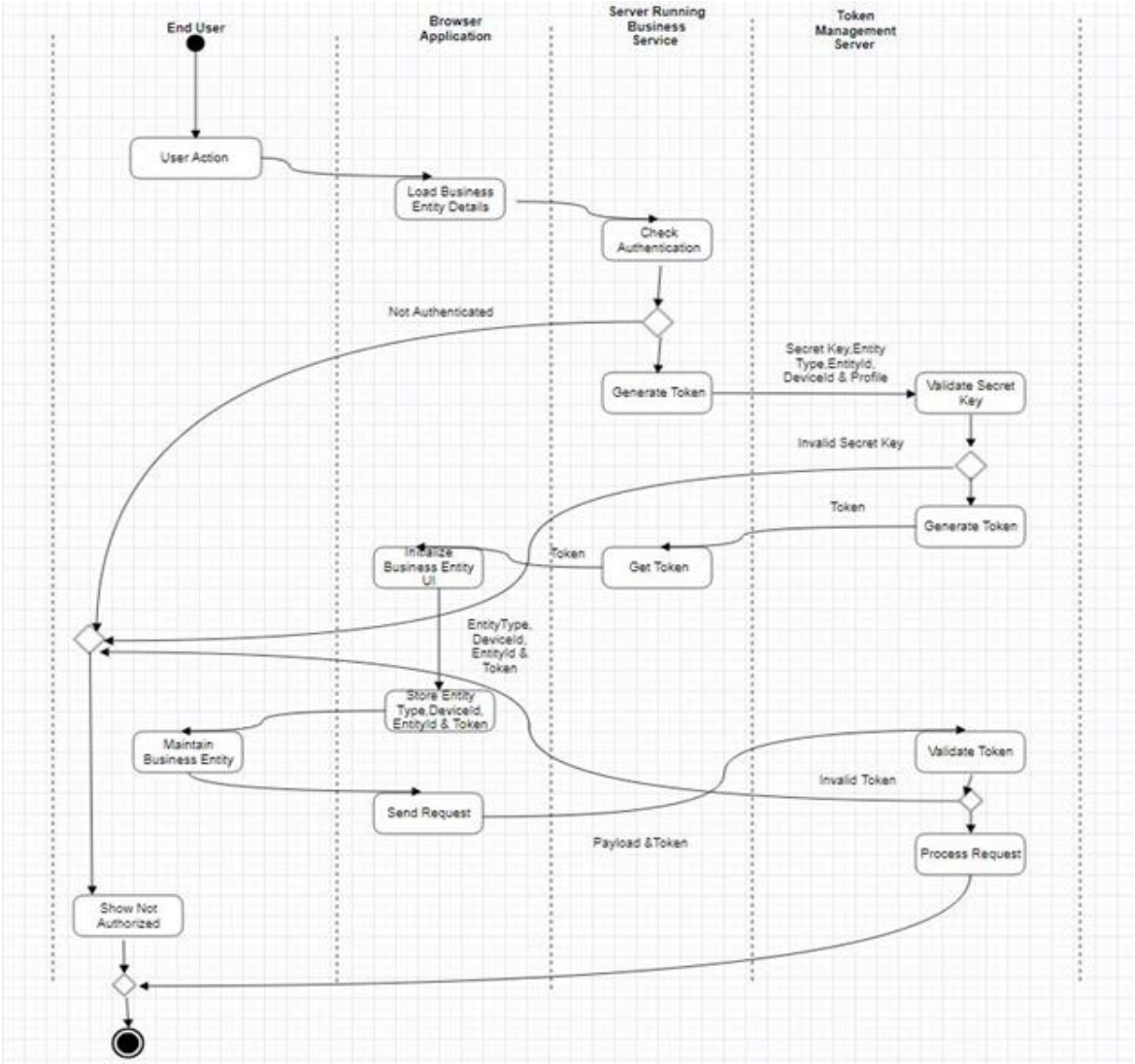


Fig. 2 Sequence diagram

The working principle to propose the token-based authorization of Apache Olingo OData Services, the above Fig-2 has proposed the implementation using sequence diagram.

1. The end-user of the mobile/web application acts on the user interface of the application to trigger the loading of the business entity.
2. The service request goes from the mobile/web client to the server running business service & checks for authentication.
3. If the authentication fails then the service request fails, if it passes then a generate token request goes to the token management server with parameters such as Secret Key, Entity Type, Entity Id, Device Id & User Profile.

4. If the secret key validation fails then the service request fails & if it passes then a token is generated and sent to the mobile/web client in the service response.
5. The mobile/web client sends the received token along with Entity Type, Entity Id, Device Id to the server running the business service.
6. The token is validated by the token management service, if it passes the required business entity details is sent to the end-user.
7. If the token validation fails a not authorized error is received by the end-user.

Results

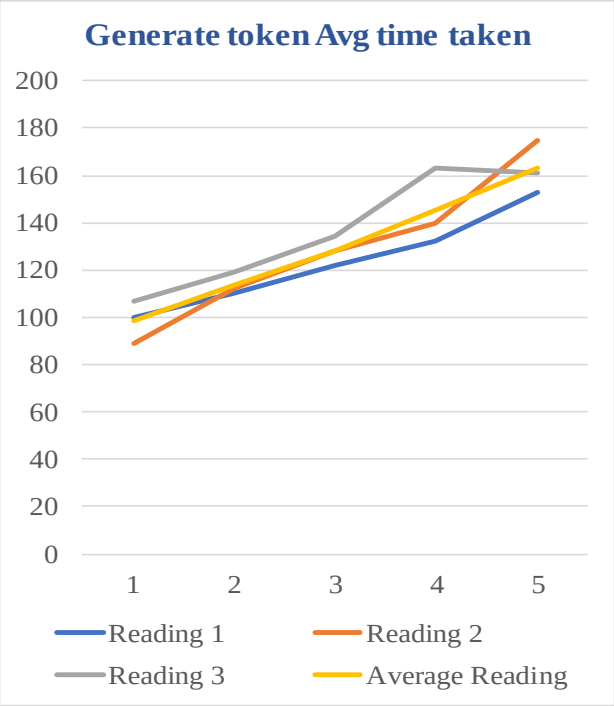
This part is providing the details about results that are numerical. We have obtained these results after the performance. In this section are presented the numerical results obtained with performance test conditions as per the proposed system. These tests were performed with multiple threads running on a client. Different requests related to the service made by each thread. We used the load runner tool for creating multiple sequence requests for the token and then validation case validating to token when token-based solutions are used to create the request, Table 1 represents the required time for each transaction value in terms of milliseconds.

Generate Token Approach and Analysis with Numerical Results

Table 1 Number of Generate Token Clients Request

Reading	10	20	30	40	50
Reading 1	100	110	122	132	153
Reading 2	89	112	128	140	175
Reading 3	107	119	134	163	161
Average Reading	98.67	113.67	128	145	163

As shown above in Table-1, to generate the token we have taken 3 sets of reading with the different sets of the user using the load runner tool to request for the token generation with multiple client requests at the same time. Which helped to analyze the actual environment when the request for token generation will be used to generate the token from the token management services. The below graph (Graph-1) explains the same thing using a graph about the multiple set of the client request.



Graph 1 Token Generation Time Analysis

Validate Token Time taken Analysis

This document proposes that validation can be implemented using two ways, First through Servlet Filter by overriding the do Filter method and validate the token in Filter class. In the second approach, we can achieve the same on the OData JPA Factory class by overriding the create Service class and the token can be passed in every subsequent call as a transient field in the JPA Entity class, and then validation can be done by reading the transient filed of every request and can be validated against the cached hash table (where the token is stored for the performance reason).

i. Token validation using Servlet Filter

In this approach, the token is validated using the do Filter class, and validation of the token can be implemented. Table-2 and Graph-2 explain the performance of the number of a request originating from the client for the token validation.

Table 2 Number of validate token request from Client

Readings validate using Servlet Filter	10	20	30	40	50
Reading 1	297	360	478	691	758
Reading 2	356	399	536	758	869
Reading 3	265	347	403	621	983
Average Reading	306	368.67	472.33	690	870

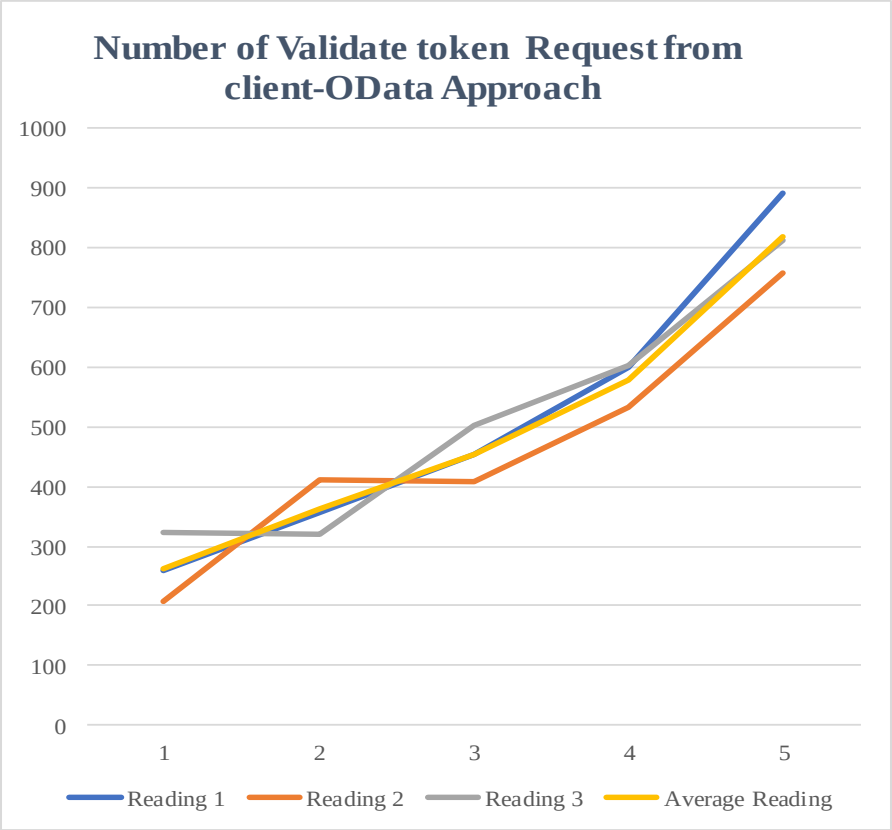
Graph 2 Token validation-ServletFilter approach

ii. Token Validation using Apache Olingo OData Approach

In this approach the token is passed as a trainset parameter in the entity and in creating Service of ODataServiceFactory class, a transient parameter can be read and validation can be implemented. Below Table-3 and Graph-3 explained the performances of no. of client request and time is taken to validate the request.

Table 3 Number of validating token requests from client

Reading Odata validate using Transient parameters	10	20	30	40	50
Reading 1	258	356	453	598	891
Reading 2	208	411	407	532	756
Reading 3	323	321	502	601	811
Average Reading	263	362.67	454	577	819.33



Graph3 Token validation-Client-OD approach

The above-mentioned Table-2 and Table-3 is the detailed analysis about the validation using different mechanism and it comparative analysis of time taken.

Conclusion

Finally, this paper presented OData Services which are based on the token for authorization of the consuming application. The algorithm is to achieve the desired secure handshake so that any consuming application requests the desired business service a secret token with an input combination. The combination will register device details, session object along with the other information that can be fetched from the server applications. The source business service creates a security token using the SHA-256 algorithm to create a hashed key, which is a secured way to create a token and can share with the consuming application for further request. Here a secured token-based authorization provides a two-way handshake, which makes the communication secure. In this approach, we authorize the server request on the above input combination and return the token to the consuming application. The consuming application whenever requests to the source business service.

References

Anajemba, J.H., Tang, Y., Iwendi, C., Ohwoekevwo, A., Srivastava, G., & Jo, O. (2020). Realizing efficient security and privacy in IoT networks. *Sensors, 20*(9), 2609.

Campbell, B., Mortimore, C., & Jones, M. (2015). Security assertion markup language (SAML) 2.0 profile for OAuth 2.0 client authentication and authorization grants. *Internet Engineering Task Force (IETF).*

Badii, C., Bellini, P., Difino, A., & Nesi, P. (2020). Smart city IoT platform respecting GDPR privacy and security aspects. *IEEE Access, 8,* 23601-23623.

Chaudhry, S.A., Yahya, K., Al-Turjman, F., & Yang, M.H. (2020). A secure and reliable device access control scheme for IoT based sensor cloud systems. *IEEE Access, 8,* 139244-139254.

Chikouche, N., Cayrel, P.L., & Boidje, B.O. (2019). A privacy-preserving code-based authentication protocol for Internet of Things. *The Journal of Supercomputing, 75*(12), 8231-8261.

Das, A.K., Zeadally, S., & He, D. (2018). Taxonomy and analysis of security protocols for Internet of Things. *Future Generation Computer Systems, 89,* 110-125.

Ethelbert, O., Moghaddam, F.F., Wieder, P., & Yahyapour, R. (2017). A JSON token-based authentication and access management schema for Cloud SaaS applications. *In IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, 47-53.

Foltz, K.E., & Simpson, W.R. (2016). Simplified key management for digital access control of information objects. *In Proceedings of the World Congress on Engineering,* 413-418.

https://www.redalyc.org/jatsRepo/4115/411557165001/html/index.html

Indu, I., Rubesh Anand, P.M., & Bhaskar, V. (2017). Encrypted token based authentication with adapted SAML technology for cloud web services. *Journal of Network and Computer Applications, 99,* 131-145.

Kubovy, J., Huber, C., Jäger, M., & Küng, J. (2016). A secure token-based communication for authentication and authorization servers. *In International Conference on Future Data and Security Engineering,* 237-250.

Sheffer, Y., Hardt, D., & Jones, M.B. (2020). JSON web token best current practices. *RFC, 8725,* 1-13. http://www.ietf.org/rfc/rfc7519.txt

Madhavi Latha, C., & Soujanya, K.L.S. (2020). Secure IoT Framework Through FSIE Approach. *In International Conference on Futuristic Trends in Networks and Computing Technologies,* 17-29. https://doi.org/10.1007/978-981-16-1480-4_2

Challa, M.L., & Soujanya, K.L.S. (2021). Secured smart mobile app for smart home environment. *Materials Today: Proceedings, 37,* 2109-2113. https://doi.org/10.1016/j.matpr.2020.07.536

Challa, M.L., Soujanya, K.L.S., & Amulya, C.D. (2020). Remote Monitoring and Maintenance of Patients via IoT Healthcare Security and Interoperability Approach. *In Cybernetics, Cognition and Machine Learning Applications,* 235-245.

Mahbub, M. (2020). Progressive researches on IoT security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics. *Journal of Network and Computer Applications, 168,* 102761.

Malani, S., Srinivas, J., Das, A.K., Srinathan, K., & Jo, M. (2019). Certificate-based anonymous device access control scheme for IoT environment. *IEEE Internet of Things Journal*, *6*(6), 9762-9773.

Martínez-Peláez, R., Toral-Cruz, H., Parra-Michel, J.R., García, V., Mena, L.J., Félix, V.G., & Ochoa-Brust, A. (2019). An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors, 19*(9), 2098.

Shekh-Yusef, R., Ahrens, D., & Bremer, S. (2015). HTTP Digest Access Authentication, Internet Requests for Comments, *Internet Engineering Task Force (IETF), RFC* 7616.

Tabassum, T., Hossain, S.K., Rahman, M., Alhamid, M.F., & Hossain, M.A. (2020). An efficient key management technique for the Internet of Things. *Sensors, 20*(7), 2049.

Yadav, N.S., Reddy, B.E., & Srinivasa, K.G. (2018). An efficient sensor integrated model for hosting real-time data monitoring applications on cloud. *International Journal of Autonomic Computing*, *3*(1), 18-33.

Yadav, N.S., Reddy, B.E., & Srinivasa, K.G. (2018). Cloud-based healthcare monitoring system using storm and Kafka. *In Towards Extensible and Adaptable Methods in Computing,* 99-106.

Yadav, N.S., Srinivasa, K.G., & Reddy, B.E. (2019). An iot-based framework for health monitoring systems: A case study approach. *International Journal of Fog Computing (IJFC), 2*(1), 43-60.

Yadav, N.S., Reddy, M.P.B., & Sreenivasulu, G. (2020). ML and IOT based Real-Time Health Monitoring System for Domestic Animals. *Journal of Critical Reviews, 7*(19), 10111-10117.

Yamin, M., Alsaawy, Y., Alkhodre, A.B., & Sen, A.A.A. (2019). An innovative method for preserving privacy in Internet of Things. *Sensors, 19*(15), 3355.

Yan, L., Chen, X., Deng, H., & Ye, X. (2019). A delegation token-based method to authenticate the third party in TLS. *International Journal of High Performance Computing and Networking, 13*(2), 164-174.

Yuliana, M. (2019). An efficient key generation for the Internet of Things based synchronized quantization. *Sensors*, *19*(12).

Votinova, E.M., & Votinov, M.V. (2019). Information society: Analyzing problems and prospects of using information technologies, computers and communication networks. *Webology, 16*(1), 86-113.