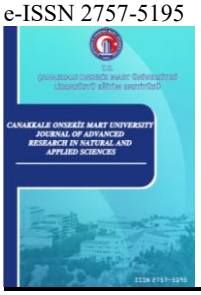
e-ISSN 2757-5195



Çanakkale Onsekiz Mart University

Journal of Advanced Research in Natural and Applied Sciences

Open Access

doi.org/10.28979/jarnas.1034534 2022, Vol. 8, Issue 4, Pages: 677-694 dergipark.org.tr/tr/pub/jarnas

Group-Based Authentication Methods in The OneM2M Ecosystem

İbrahim Uğur Aba[1,*], Erhan Taşkın[2]

[1]Department of Computer Engineering, Faculty of Engineering, University of Turkish Aeronautical Association, Ankara, Türkiye
[1]Department of Artificial Intelligence Technology, Graduate School of Natural and Applied Sciences, Ankara University, Ankara Türkiye
[2]Cloud DevOps Architect at Afiniti, İstanbul, Türkiye

Article History

Received: 09.12.2021

Accepted: 20.07.2022

Published: 15.12.2022

Research Article

Abstract − The essential element of the Internet of Things (IoT) environment, the number of devices has traditionally exceeded the number of devices connected to the Internet. This situation is considered positive for the IoT concept but still has negative consequences. Undoubtedly, the most prominent and most important among these results is the security of the devices and the constructed IoT environment. Group-based authentication and authorization methods are crucial to ensure the safety of many IoT devices and the environment. In this study, the "auth" mechanism that performs group-based authentication and authorization processes, serving from the first moment when the devices in the IoT environment are included in the system until they leave the system, has been developed. In the development process of the "auth" mechanism, the Mobius IoT platform, which is evaluated as a golden sample by the oneM2M global organization and developed as an open-source code, is taken as the basis. The "auth" mechanism tested in three different test environments, including simulation, physical, and cloud environments, were tested using five different test scenarios. By using the group management module provided by the IoT service platform and the "auth" mechanism's together, it has been observed that the computational overhead on the devices and the signal traffic in the environment provide up to 4 times efficiency according to performance measurements. With the development of the "auth" mechanism with a flexible structure, it can be operated independently from the IoT server platform, allowing interoperability between oneM2M-based IoT server platforms.

Keywords − *Group-based authentication, internet of things, oneM2M, open-source IoT server platform, security*

1. Introduction

The internet of things (IoT) is a new network structure in which all devices are connected and may communicate with each other via the internet. IoT can be relevant for numerous application fields such as smart transportation, tourism service, medical treatment, energy management, and education (Su, Wong, & Chen, 2016). In addition to these, smart homes, smart cities, and smart production stand out as other important application areas. IEEE defines this notion as "A network of items - each embedded with sensors - which are connected to the Internet" (Define IoT, 2015).

The Gartner research business shows the current and near-future situation of the Internet of Things through several studies. A study done in 2019 anticipated that the number of terminal units utilized in the corporate and automobile industries would reach 5.81 billion in 2020. That number translates to an increase of 21 percent compared to the previous year. Also, it is noted in the report that the sector where the tremendous increase will be in-building automation with 42 percent (Gartner, Inc., 2019). In another study by research company Gartner done in 2018, 20 percent of institutions surveyed stated that 2015 to 2018 be exposed to at least one IoT-based cyberattack. Researchers believe that the expenditure made in 2018 to assure security in the IoT will reach 1.5 billion dollars, and by 2021 this amount will climb to 3.5 billion dollars (Gartner, Inc., 2018). The analysis conducted by the IoT analytics research organization published in November 2020 indicated similar results

[1] ugur.aba@gmail.com
[2] erhantaskin@gmail.com
[*]Corresponding Author

with Gartner's investigations. The analysis found that despite the COVID-19 global pandemic's detrimental consequences, the IoT market continues to flourish and grow in 2020. The most interesting information revealed by the study was that for the first time in 2020, the number of IoT devices surpassed the number of devices that can typically connect to the Internet (computers, phones, tablets, etc.). At the end of 2020, it was anticipated that 11.7 billion of 21.7 billion active internet-connected devices world-wide, about 54 percent, will be IoT devices. It was also projected that the number of IoT devices would climb to 30 billion in 2025, and 4 IoT devices will be used per person according to the world population (IoT Analytics, 2020). In the study published by IoT Analytics in May 2022 and explaining the status of IoT in 2022, it was stated that terminal devices will increase by 18 percent and reach 14.4 billion end devices globally. While this supports the results of previous studies, it shows that the recovery in the IoT markets, where the chip shortage continues. In 2021, the number of global IoT connections increased by 8% to 12.2 billion active endpoints, a significantly lower rate than in previous years. Despite a surge in demand for IoT solutions and good attitude in the IoT community and most IoT end markets, IoT Analytics anticipates the chip shortage to have a long-term impact on the number of connected IoT devices. The ongoing COVID-19 pandemic and general supply chain interruptions are also challenges for IoT industries. The Internet of Things industry is estimated to expand 18% to 14.4 billion active connections by 2022. It is predicted that there will be around 27 billion linked IoT devices by 2025, as supply restrictions ease and demand increases (IoT Analytics, 2022).

The participation of so many IoT devices in the ecosystem poses numerous issues. In the study conducted by AIOTI, it was revealed that there are more than 100 protocol recommendations in 9 key categories (The Alliance for Internet of Things Innovation, 2019). The large amount of study and development on classical authentication and key agreement (AKA) and group-based AKA. The key objective of these research carried out is to perform mutual authentication and key exchange between end devices; It has been noticed as guaranteeing confidentiality and data integrity, and minimizing bandwidth usage, and defining the most efficient and uncompromising AKA method.

The earliest proposal of the group-based authentication and authorization strategy is the G-AKA technique (Chen, Wang, Chi & Tseng, 2010). In this protocol, a mobility management entity (MME) can authenticate additional end devices in the group using information comprising of authentication of the first end device. Thus, bandwidth usage for authentication for other end devices in the group has greatly lowered. However, it does not give a solution to the signal traffic that arises when several end devices wish to do authentication synchronously, which is vulnerable to the widely used man-in-the-middle (MitM) and denial-of-service (DoS) attacks. SE-AKA (Lai, Li, Lu & Shen, 2013) and EG-AKA (Jiang, Lai, Luo, Wang & Wang, 2013) have been presented, which are based on G-AKA and provide safety standards not found in G-AKA. Thus, the high computational burden due to asymmetric key operations attracts attention as the weak point of both protocols. The NOVEL-AKA protocol employs symmetric keys to lessen the computational cost, however it raises security difficulties (Lai, Li, Li & Cao, 2013). The GBAAM-AKA protocol proposes that to strengthen security in group-based AKA protocols (Cao, Ma & Li, 2015). However, the usage of asymmetric keys produces a high transaction burden and cannot secure privacy. PRIVACY-AKA employs asymmetric cryptography to provide secrecy (Fu, Song, Li, Zhang & Zhang, 2016). Although the protocol is robust to assaults, it creates a significant processing burden and does not offer forward and backward privacy. GLARM-AKA protocol has been designed to decrease the computational and communication overhead; protocol beneficial for resource-constrained end devices; It is not effective in terms of impersonation attack and privacy protection (Lai, Lu, Zheng, Li & (Sherman) Shen, 2016). The GR-AKA protocol has been suggested to assure security and privacy (Li, Wen & Zheng, 2016).). The sophisticated and time-consuming Lagrange Multiplier (LC) is employed in the GR-AKA protocol. GBS-AKA protocol, which is resistant to assaults and lowers the communication burden (Yao, Wang, Chen, Wang & Chen, 2016). However, the technique is subject to preserving secrecy, impersonation, and DoS attacks. The SEGB-AKA method has been suggested to strengthen the security of group-based protocols (Parne, Gupta & Chaudhari, 2018). However, the protocol cannot give protection against DoS assaults. GSL-AKA protocol, which has the same structure as GBS-AKA and SEGB-AKA, has been suggested by developing features (Modiri, Mohajeri & Salmasizadeh, 2018). It is noted that it successfully overcomes recognized security and non-security challenges while protecting the secrecy of end devices and groups.

The study, published in 2019 by Şahinaslan, looked at encryption technologies for protocols used on the internet of things. The article explains the Markov Chain and RSA asymmetric encryption approach for wireless IoT devices. The MAC session key provides cryptographic control over the information while also providing security against potential attacks. The paper also explains how to avoid the KRACK vulnerability

that happens during the key exchange phase by employing the DragonFly key exchange mechanism (Şahinaslan, 2019). Aydın et al. propose a lightweight Group authentication system (GAS) that significantly reduces device energy consumption, saving more than 80% when compared to state-of-the-art alternatives. Their approach is also resistant to replay and man-in-the-middle attacks. In mMTC situations, the proposed approach also tackles key agreement and key distribution concerns. That solution is also useful in both centralized and decentralized group authentication scenarios. The proposed solution can meet the rapid authentication requirements of the envisioned agile 6G networks, which will be supported by aerial networking nodes (Aydın et al, 2020). Padmashree et al. suggested Group Key Exchange and Authentication with Elliptic Curve Cryptography (ECC), or GKEAE for short, on the Internet of Things to establish safe key distribution and improve security. When an IoT device joins or leaves a group, ECC is used for authentication. Integrating access authentication and data transfer improves the serviceability of IoT devices. The GKEAE delivers a faster group key distribution computation time than the quick authentication system (Padmashree et al, 2022).

Suggesting different protocols for each application area and not settling on one protocol produces a highly undesirable standardization situation. To define technical standards for architectural structure, API, and security solutions in M2M and IoT technologies, in 2012, the world's main standardization authorities joined together to form an organization named OneM2M. OneM2M foundation represents nearly 200 enterprises and universities. The OneM2M standard offers a basic horizontal platform design based on a three-tier paradigm of applications, services, and networks (OneM2M, 2012).

Mobius, an open-source IoT service platform, was created by the Korean Electronic Technologies Institute (KETI, 1991), a member of the OneM2M organization, as part of an open alliance for IoT standard (OCEAN) investigations. Mobius IoT service platform gains notice by becoming the first application to acquire a oneM2M compliance certificate. It is also used as a gold sample to evaluate test cases and test systems. Mobius presents common service functions (CSF) as middleware for multiple service areas to IoT applications (IoT OCEAN, 2017).

In this study, based on the Mobius IoT service platform, a “auth” mechanism has been designed that executes group-based authentication and authorization operations from the moment the IoT devices in the IoT environment joined the ecosystem until they exit the system. The “auth” process has been evaluated in three separate test contexts: simulation, physical, and cloud environments. In the test scenarios, the benefit of the “auth” method with the group-based authentication procedure was examined. According to the results obtained by testing the “auth” mechanism using five different test scenarios in three different test environments, the computational overhead on the nodes and the signal traffic in the IoT environment have been significantly reduced by running the mechanism together with the group management module provided by the IoT service platform. It has been established that the proposed “auth” method contributes 1ms computational overhead to the IoT service platform, delivering an optimal benefit between 2 and 25 IoT devices and providing up to 4 times efficiency.

2. Materials and Methods

This section outlines the tools and apps utilized in the study. The environment produced by the introduction of open-source software and the enhancements made to it is presented in the architectural design. In addition, information about the methods used in the test scenarios on the generated environment is presented.

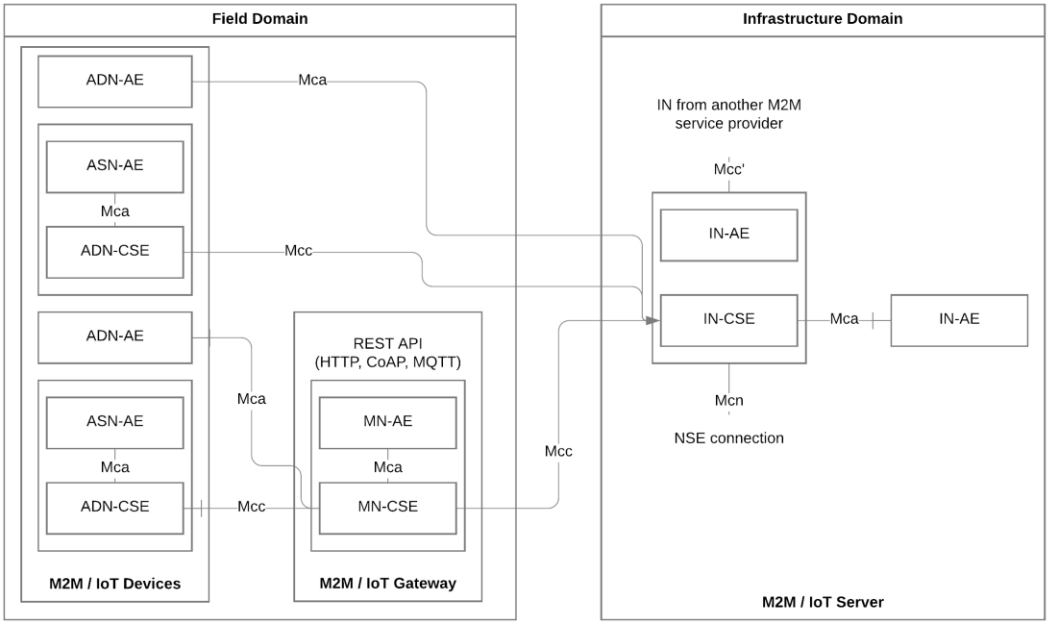
2.1. Open-Source Applications

2.1.1. Mobius IoT Server Platform

Mobius is an open-source IoT server platform based on the oneM2M standards developed by KETI as part of OCEAN studies. Mobius provides CSFs (enrollment, data management, subscription/notification, security) middleware for IoT applications of different service domains. Mobius can successfully connect oneM2M-compatible and non-oneM2M-compatible devices. Within the global oneM2M organization, Mobius has been awarded the “oneM2M compliance certificate” by Telecommunications Technology Association (TTA). This certification guarantees that Mobius meets the oneM2M specifications and testing requirements that ensure interoperability with oneM2M products. As it is the first application to receive oneM2M certification, it is used as a gold example for validating test scenarios and test systems (IoT OCEAN, 2017).

Functional Architecture of the Server Platform

The Mobius IoT server platform created by KETI is architecturally based on the OneM2M functional reference architecture stated in the document named "TR-0025 Application Developer Guide" issued with version 2A of the OneM2M global organization (TR-0025 Technical Report, 2018).



Nodes	Logical Entities	Reference Points
IN: Infrastructure Node MN: Middle Node ASN: Application Service Node ADN: Application Dedicated Node	CSE: Common Service Entity AE: Application Entity NSE: Network Services Entity	Mca: Interface point between the AE and the CSE Mcc: Reference point between two CSEs Mcn: Reference point between a CSE and the underlyting NSE Mcc': Interface point between two M2M service providers

Figure 1. Functional architecture of the IoT server platform (Kim, Choi, Yun & Lee, 2016).

As shown in Figure 1, the oneM2M architecture divides M2M and IoT environments into two different domains (Field Domain and Infrastructure Domain). It defines four different node types for use in these domains. An IN (infrastructure node) can exist in the infrastructure domain of any M2M (machine-to-machine) service provider, while any oneM2M node group, including MN (middle node), ASN (application server node), and ADN (application dedicated node), even non-oneM2M nodes can exist in the field domain.

Improvements on the IoT server platform

The Mobius IoT server platform created by KETI is architecturally based on the OneM2M functional reference architecture stated in the document named "TR-0025 Application Developer Guide" issued with version 2A of the OneM2M global organization (TR-0025 Technical Report, 2018).

The IoT service platform leverages access control policies (ACP) under oneM2M standards for authorization processes in security operations (TS-0001 Technical Specification, 2016). However, it does not give a solution for authentication processes. In this method, when a common service entity (CSE) seeks to access a resource in a working structure, it is sufficient to conduct merely ACP's. The fact that the element that attempts to get access during the creation of a new resource or accessing an existing resource does not perform the authentication process puts the system open to possible attacks.

Within the scope of the study, the "auth" mechanism, which will conduct the operations of two fundamental categories (Identification and Authentication, Authorization) for the "Security Functions Layer" of the oneM2M security architecture, has been created (TS-0003 Technical Specification, 2018). Passport.js (Passport.js) and jsonwebtoken (RFC7519) libraries are used for authentication and authorization operations as it is suitable with the IoT service platform created in the Node.js working environment using the JavaScript dynamic programming language. In the created "auth" method, MongoDB is used independently from the MySQL database used by the IoT service platform (MongoDB). The usage of a distinct database management

system permitted flexibility in the IoT network structure to be established. It was also feasible to establish a separate server that could execute authentication and authorization operations.

The "auth" method was created with group-based authentication and authorization operations in mind. Accordingly, when a group administrator initially authenticates, the system creates a token value that other members of the group can use for a limited period. By using this token value, which is specified as a Group Common Key (GCK), other members of the group can be added to the system avoiding the authentication stage if the GCK information is valid. After the validity of the produced token value expires, the group administrator must execute an authentication procedure again. The jsonwebtoken (RFC7519) library is used to construct the token structure that executes the GCK job.

Table 1
Comparison of OneM2M Token Structure and Generated Token Structure

OneM2M Token Structure	Generated Token Structure	Description
version	keyID	Token version
tokenID	jwtID	Token unique id
holder	subject	The ID of the token holder
issuer	issuer	The ID of the token issuer
notBefore	notBefore	The token is valid from this moment
notAfter	expiresIn	The token expires after this moment
tokenName	header	Token name (optional)
audience	audience	CSE's ID list (optional)
permissions	-	Associated permissions
extension	-	Application-specific information

The generated token value is supplied in the header named "authorization" which is appended to HTTP requests according to the format provided in the document named oneM2M "TS-0009 Protocol Binding" (TS-0009 Technical Specification, 2016).

Table 1 explains the token structure specified in the technical specification of oneM2M TS-0003 and the attributes used in the generated token structure. The generated token value is for sending in HTTPS requests.

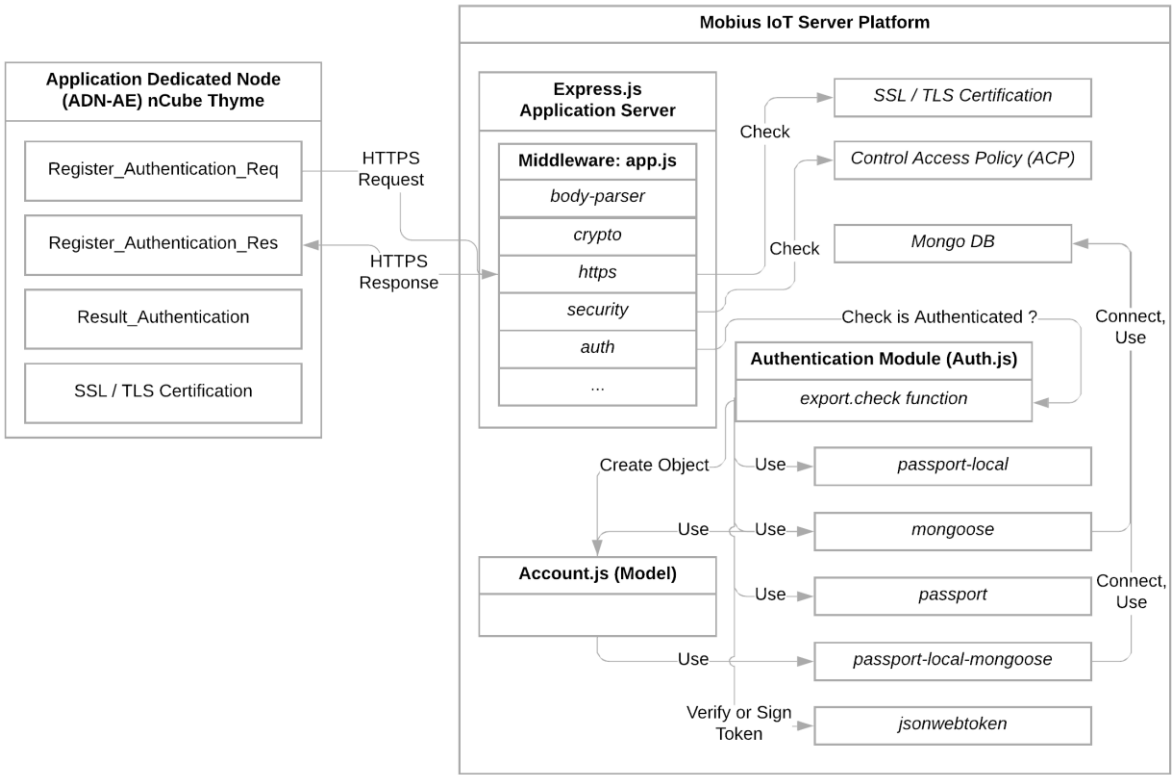


Figure 2. The architectural design of the IoT service platform after the inclusion of the auth mechanism.

To increase security, all sent and received HTTP requests must be transmitted over a secure channel. Therefore, the HTTPS protocol is for communication between the nodes in the field domain and IN. The architectural design formed after the operations performed on Mobius is shown in Figure 2.

The established "auth" method must not compromise the IoT platform's compliance with oneM2M requirements. For this reason, the token structure developed must also conform with the criteria described in the paper published by the worldwide oneM2M organization, named "TS-0003 Security Solutions". Accordingly, a token is for carrying authorization information, which can be roles given to the owner or ACPs valid for the owner (TS-0003 Technical Specification, 2018).

2.1.2. nCube

The nCube is specified as the general name of the nodes in the field domain, based on the oneM2M standards created by KETI within the framework of OCEAN research. Developed as open source, the nCube program contains five different versions (Rosemary, Thyme Node.js, Lavender, Thyme Arduino, Thyme Java) that may operate as three separate oneM2M node structures. Within the scope of this investigation, the first three node types were employed. The nCube Rosemary is an open-source IoT gateway platform based on oneM2M standards.

The nCube Rosemary application, used to deliver proximity based IoT services, gives CSFs to oneM2M apps and other oneM2M devices. Serving as the MN-CSE, nCube Rosemary also supports interoperability services using interworking proxy application entity (IPE), as stated in the oneM2M standards. It links to IN-CSE utilizing CSEs in ASN and ADNs, which are additional nodes in the field domain (nCube-Rosemary, 2018). The nCube Thyme is an open-source IoT device application element based on oneM2M standards. Thyme has three separate versions: node.js, java, and android. Node.js version is employed within the scope of this study. The nCube Thyme application may be linked to MN-CSE or IN-CSE (nCube-Thyme, 2018). The nCube Lavender is also an open-source IoT device platform based on oneM2M standards. The nCube Lavender, one of the oneM2M platforms, delivers CSFs to oneM2M device apps operating on the same device (nCube-Lavender, 2018). In this sense, although it is comparable to the nCube Rosemary application, nCube Rosemary acts as an MN in the oneM2M domain, whereas the nCube Lavender application serves as an ASN.

Improvements on the nCube application

Several provisions have been made for the nCube application to function in harmony with the "auth" mechanism created on the IoT service platform. These arrangements are based on the nCube Thyme application and the "auth_usr" and "auth_pwd" headers have been added to the HTTPS request submitted for registration to the "auth" mechanism. In addition, the IoT service platform is alerted that the "auth" mechanism established by the title "use auth" has the value of "enable". If the title "auth_useprotocol" is provided as "local", the system applies the authentication processes using the information on the MongoDB. If a group header also known as administrator will establish a connection with the "auth" mechanism for the first time, while transmitting the above-mentioned "auth_usr" and "auth_pwd" headers, the other members of the group will use to token value created by the "auth" mechanism, which is used as GCK, by the TS-0009 technical specification under the "authorization" heading (TS-0009 Technical Specification, 2016).

2.2. Test Environments

In the study, three distinct environments are built by employing the IoT service platform, in which the "auth" method is integrated. Performance measurements were done on equipment with varying technical parameters.

2.2.1. Simulation Environment

A simulation environment has been constructed to assess the performance of the "auth" method on hardware with restricted resources. Oracle VM VirtualBox program, a free and open-source hypervisor created by Oracle, was used to construct the virtual environment used. The technical parameters of the virtual machines utilized instead of the nodes in the oneM2M IoT ecosystem in the simulation environment are provided in Table 2.

Table 2
Technical Specifications of Machines in the Simulation Environment

Machine Name	CPU	RAM	Storage	Operating System
IN-Mobius	2 Virtual Cores	2048 MB	20 GB SSD	Ubuntu 20.04 (64-bit)
MN-Rosemary	2 Virtual Cores	2048 MB	20 GB SSD	Ubuntu 20.04 (64-bit)
ASN-Lavender	1 Virtual Core	2048 MB	20 GB SSD	Ubuntu 20.04 (64-bit)
ADN-AE-Thyme	1 Virtual Core	1024 MB	20 GB SSD	Ubuntu 20.04 (64-bit)
JMeter Xubuntu	2 Virtual Cores	2048 MB	20 GB SSD	Ubuntu 20.04 (64-bit)

The technical specifications of the computer, which runs virtual machines throughout the development phase, include Intel Core i7-5600U 2.60 GHz CPU, 8 GB RAM, 64-bit Windows 10 Pro operating system, and 500 GB SSD. While developing the "auth" method, the Postman program was used to transmit HTTPS requests from the host system to virtual machines and to execute unit tests. A NAT network has been setup in the Oracle VM VirtualBox program to correctly conduct HTTPS requests.

2.2.2. Physical Environment

In the simulation environment, test scenarios were achieved by allocating restricted resources to virtual computers with nodes and IoT service platforms. In the physical environment experiments, a laptop computer with technical characteristics that can function as an MN in the oneM2M IoT ecosystem and a desktop computer with technical capabilities that can work as an IN was utilized. The technical parameters of the physical devices utilized instead of the nodes in the oneM2M ecosystem in the physical environment are presented in Table 3.

Table 3
Technical Specifications of Machines in the Physical Environment

Machine Name	CPU	RAM	Storage	Operating System
MN-Windows	Intel i7 5600U	8 GB	500 GB SSD	Windows 10 (64-bit)
IN-Mobius	AMD FX-8320	32 GB	256 GB SSD	Ubuntu 20.04 (64-bit)

The machine named MN-Windows is in Lapseki, Çanakkale, whereas the machine called IN-Mobius is located in Kepez, Çanakkale. There is around 40 KM between the two mentioned locations. To perform the test scenarios, a test program named Postman was installed on the MN-Windows machine and the tests were run.

2.2.3. Cloud Environment

To test the established "auth" technique in the cloud context, an EC2 is built on Amazon Web Services (AWS). The built EC2 machine is picked in t2.large type, us-east-1e region, and North-ern Virginia location, and the IoT service platform is deployed and served. The technical parameters of the physical and virtual computers utilized instead of the nodes in the oneM2M ecosystem in the cloud environment are provided in Table 4.

Table 4
Technical Specifications of Machines in the Cloud Environment

Machine Name	CPU	RAM	Storage	Operating System
MN-Windows	Intel i7 5600U	8 GB	500 GB SSD	Windows 10 (64-bit)
IN-Mobius-EC2	Intel Xeon 2 vCPU	8 GB	20 GB EBS	Ubuntu 20.04 (64-bit)

HTTPS requests are made to the IoT service platform on the virtual machine name IN-Mobius-EC2 with IP number 54.227.195.58 utilizing the virtual private cloud (VPC) on the physical machined called MN-Windows. The HTTPS requests made from the MN-Windows computer is moved to the VPC when it reaches the internet gateway on AWS. Considering the security information, the HTTPS requests are sent to the EC2 machine with the t2.large type IoT service platform in the private subnet via the NAT gateway.

2.3. Testing Tools

2.3.1. Postman

During the creation of the "auth" mechanism, the Postman program was utilized to execute unit tests (Postman API Platform). During the testing utilizing Postman version 7.27.0, nCube open-source apps were emulated and operations were carried out on the IoT service platform. A collection of all HTTPS requests performed for unit testing of the "auth" method has been built.

2.3.2. Apache JMeter

In the study, Apache JMeter v5.3 application was utilized to assess the efficiency offered by the developed "auth" method (Apache JMeter). The test application was executed on the virtual machine named "JMeter Xubuntu" defined in the simulation environment section. The test application took part in various test scenarios as a group administrator or group member, which formed the foundations of the group-based authentication and authorization framework.

3. Results and Discussion

In this part, five distinct test situations in which we tested the "auth" method will be explained. Then, three distinct test methods that we devised utilizing these situations will be discussed. Finally, the outcomes seen in the test situations will be compared and assessed using the defined test methodologies.

3.1. Test Scenarios

3.1.1. Test Scenario 1: Determination of Core Values

In the technical standard named TS-0003 issued by oneM2M, the methods to be given for authentication and authorization are outlined. In addition, the IoT service platform, which is described as the golden example by oneM2M and produced by KETI, has been built by TS-0003 and other technical specifications released by oneM2M. In the technical specification designated TS-0003, it is recommended to employ ACPs for security and authorization processes (TS-0003 Technical Specification, 2018).

Since the IoT service platform is established by the given technical standards, no separate authentication and authorization module has been developed on the platform. In circumstances when ACP and the "auth" method are not employed, there is no security mechanism on the IoT service platform. The "auth" mechanism has been designed as a module that is meant to be used together with the ACP mechanism, not as a substitute for the ACP mechanism.

To estimate the contribution of the "auth" mechanism, which provides group-based authentication and authorization, the default results should be calculated as fundamental values. At this point, fundamental values were measured by deactivating the group feature of the "auth" mechanism produced owing to the absence of an authentication and authorization module operating on the IoT service platform.

In the test scenario carried out, the application entity (AE) registration request of the node units was issued, rising in floating slices between 100 up to 1000. The "auth" method, designed with each HTTPS request delivered independently, is offered to execute authentication and authorization. The outcomes when test case 1 is executed in the simulation, physical, and cloud settings are presented in Table 5, accordingly.

Table 5
Results of Test Scenario 1

	Simulation Environment			Physical Environment			Cloud Environment		
# Of nodes	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)
100	975	844	2104	504	468	2382	1029	982	1649
200	897	822	2157	486	457	2335	1047	980	1605
300	909	832	2049	495	466	2933	1105	982	2797
400	1088	834	2020	483	459	2120	1115	981	2957
500	1021	818	2565	492	467	2756	1048	979	1723
600	1111	826	2530	481	455	2395	1072	978	2612
700	1115	838	2315	478	457	2266	1142	979	6336
800	1120	926	2581	480	458	2555	1078	988	4568
900	1138	929	2338	478	458	2279	1017	978	2154
1000	1111	821	2318	477	455	2248	1036	983	1463

According to the results of test scenario 1, where the core values were determined, the operation of the "auth" mechanism for the first HTTPS request sent when the IoT service platform wat started for the first time and the response time of the platform to the request create the maximum value for each test. It is shown that the number of nodes employed in the functional design of the oneM2M ecosystem is directly related to the hardware on which the IoT service platform is operating and will cease to be stable if it is more than a specific quantity.

3.1.2. Test Scenario 2: Single Retrieve of CIN Source

A key benefit of group-based authentication and authorization procedures with the group management module is that activities may be conducted on many resources by submitting a single HTTPS request. To notice this benefit, a core data set should be constructed identical to the prior test case. In this test scenario, it is targeted to provide the <ContentInstance> resources described with the abbreviation CIN in the oneM2M ecosystem to the nodes that make the request from the IoT service platform. The CIN resource represents the resources containing application-specific data generated in cooperation with the oneM2M ecosystem.

There are varied numbers of sensors and nodes in apps created to function in the IoT ecosystem. In the research done to examine the performance gain offered by group-based authentication and authorization procedures in the IoT ecosystem, it was established that the most ideal outcome was the construction of 100 groups in an IoT ecosystem with 500 nodes (Su, Wong & Chen, 2016).

Considering the findings of the research indicated in the previous paragraph, the fact that the number of nodes creating a group is between 100 and 1000, rather than between 1 and 100, is in keeping with the applications created for the IoT environment. Considering this information, the number of nodes in the test scenario was determined as 1, 2, 3, 5, 10, 25, 50, 70, 90, and 100. The outcomes when test case 6 is executed in the simulation, physical, and cloud settings are presented in Table 6, correspondingly.

According to the results of test scenario 2, which was carried out in the simulation environment and where the CIN resource was retrieved individually without using group management, an irregular increase was observed in the maximum values obtained after the test using 25 nodes, as indicated by the column representing the maximum value. Although the average and minimum values do not reach the ideal values more than 25 nodes submit single HTTPS requests to result in the IoT service platform delivering irregular results.

Table 6
Results of Test Scenario 2

	Simulation Environment			Physical Environment			Cloud Environment		
# Of nodes	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)
1	100	100	100	282	282	282	712	712	712
2	84	66	102	147	66	229	459	186	733
3	89	43	170	111	62	210	357	180	711
5	75	34	144	91	58	216	292	184	723
10	50	30	124	75	57	212	248	190	752
25	49	30	103	62	52	200	207	183	730
50	53	15	342	57	49	197	203	190	755
70	45	20	126	54	49	185	196	183	730
90	44	16	248	55	46	204	196	185	738
100	36	15	102	54	50	205	189	182	743

In the test scenario established in the physical environment, it is apparent that there is an increase owing to network latency when the results are analyzed despite the increase in CPU power. When the average values of the results achieved in the simulation environment and the results obtained in the practical environment are compared, an increase in the range of 15-20ms was seen in the test cluster with more than 5 nodes. Considering that these values are caused by network delay, it has been proved that the IoT service platform operates reliably in both scenarios. When the findings are reviewed independently of the network latency, irregular results have been created after the test in which the number of 25 nodes is employed when the values acquired are executed on a virtual machine with restricted processing capacity in the simulation environment. However, it has been noticed that the results produced by operating the IoT service platform on hardware that is used in the actual world and has higher processing power, continue to be stable up to 100 nodes. In the test scenario done in the cloud environment, it was found that the efficiency fell, and the results rose owing to the variables such as

shared EC2 usage and network latency. When the data were evaluated, it was revealed that the efficiency reduced by 4 times compared to the results in the physical environment. The findings acquired in the minimum column coincide with the values in the maximum column obtained in the test scenario in the physical environment. Considering this condition, it should be recognized that the technical characteristics of the EC2 machine used in the cloud environment are raised and the efficiency will rise if it approaches the test machine that hosts the IoT service platform in the test scenario used in the physical environment.

3.1.3. Test Scenario 3: Group-Based Authentication and Authorization

The group feature of the "auth" mechanism produced in this test scenario has been enabled to measure the benefit offered by the "auth" mechanism designed for group-based authentication and authorization procedures and to compare it with the findings in test scenario 1. The test scenario 3 carried out is based on the G-AKA research, which is considered as the beginning of group-based authentication and authorization processes. According to the G-AKA research, a group authentication key (GAK) information is created when the administrator knew the header of the group enrolled in the system completes a full authentication and authorization procedure. After the GAK information is established is shared with the other members of the group through the group header. Other members of the group can be located and processed on the system utilizing this information (Chen et al., 2012).

The GAK information produced as a result of a full authentication authorization process of the group header was sent to the IoT service platform by the group members under the "authorization" heading of the HTTPS package transmitted over a secure channel, as specified in the TS-0009 technical specification published by oneM2M, in the test, based on the number of nodes specified in test scenario 1 (TS-0009 Technical Specification, 2016).

Table 7
Results of Test Scenario 3

	Simulation Environment			Physical Environment			Cloud Environment		
# Of nodes	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)	Avg. (ms)	Min. (ms)	Max. (ms)
100	30	15	165	69	59	250	204	195	749
200	26	13	139	62	50	332	206	197	749
300	22	11	113	78	61	277	205	195	779
400	21	10	103	64	59	264	209	192	809
500	20	9	98	67	61	263	205	193	856
600	19	9	81	68	59	301	197	189	758
700	19	7	98	66	60	231	204	189	798
800	19	9	101	69	59	363	203	189	832
900	20	8	150	66	59	364	200	191	774
1000	22	8	307	66	58	377	203	189	842

In the executed test scenario, AE registration requests were issued utilizing the group resources produced by the nodes, increasing in floating slices between 100 and 1000. The findings when test case 3 is executed in the simulation, physical, and cloud settings are presented in Table 7, accordingly.

In the executed test scenario, AE registration requests were issued utilizing the group resources produced by the nodes, increasing in floating slices between 100 and 1000. The results reported in Table 7 do not contain the values of the HTTPS request, in which full authentication and authorization are made by the group header. As noted in Table 7, optimal values have been attained in the operations performed over the group resource consisting of 600, 700, and 800 nodes. According to the findings of the test scenario established in the physical environment, which is described in detail in Table 7 and indicates the maximum values, is examined, it is seen that the maximum value occurs in the initial HTTPS request issued for each test owing to network delay. When the results in the physical environment and the results in the simulation environment are compared, it is notable that the average values created in the physical environment and the average values formed in the simulation environment rise by two or three times. Considering that the IoT service platform tries to minimize the difference between the processing power and the processing power of the hardware on which it works in the physical environment, and the distance between the client and server pair in the physical is approximately 40 KM, it is thought that the difference is due to network latency. Likewise, the findings of the test scenario performed in the cloud environment agree with the results of the actual environment. In the test scenario outlined in Table

7, average values were measured in the range of 197-209ms. In addition, the average and lowest values established created a 10 percent gap between themselves, comparable to those in the actual world.

3.1.4. Test Scenario 4: Retrieving CIN Resource Using Group Resource

The group management CSF is responsible for group-related operations. HTTP or HTTPS request is issued for batch actions such as reading, writing, subscribing, notification, device management enabled by the group, as well as controlling a group or its member. Group administration is responsible for gathering group answers and alerts when a request or subscription is made through the group (TS-0001 Technical Specification, 2016).

The designed "auth" mechanism provides authentication and authorization processes in all operations done on the IoT service platform, starting from the time the nodes are included in the system and during the full process, they are in system. In the test scenario where the group management module offered by the IoT platform, which was developed by the group management features specified in the technical specification named TS-0001 published by oneM2M, is used, it is aimed to call the previously created daemon resources using a single HTTPS request.

To assess the benefit offered by the group management and the "auth" mechanism, test scenario 4 was carried out based on the identical node counts as the previous test scenario 2. Accordingly, the efficiency offered by the "auth" method and group management was seen in the test scenario employing group resources consisting of 1, 2, 3, 5, 10, 25, 50, 70, 90, and 100 nodes, respectively.

Table 8
Results of Test Scenario 4

# Of nodes	Simulation Environment, Time Elapsed (ms)	Physical Environment, Time Elapsed (ms)	Cloud Environment, Time Elapsed (ms)
1	122	118	217
2	107	161	242
3	200	204	247
5	280	232	278
10	579	353	341
25	840	681	532
50	2120	1185	1006
70	2662	1450	1142
90	3590	1876	1445
100	3554	2033	1476

The results when test case 4 is run in the simulation, physical, and cloud environments are shown in Table 8, respectively.

According to the findings of test scenario 4, when the CIN resource is obtained collectively utilizing group management, the most efficient results were reached in the test phase carried out on a single group re-source in which 25 nodes were included as indicated in Table 8. The findings obtained in instances where group resources with more than 25 nodes are included offer efficiency compared to the case where the group management module is not employed, but the efficiency supplied by the growth in the number of nodes is seen to be inversely proportional. According to the findings of test scenario 4 performed in the simulation environment, efficiency cannot be attained in the test set if a group resource consisting of one node is employed. The lowest number of nodes that a group resource is efficient is determined to be 2. It has been observed that the optimum conditions in the results obtained with the growth in the processing capacity of the hardware employed in the physical environment compared to the simulation environment have altered. As the processing power of the actual hardware rises, the efficiency given by the established "auth" mechanism increases in direct proportion. When the column, which is presented in Table 8 and represents the time taken for the process, is inspected, it is apparent that the benefit offered by the mechanism starts with the test set consisting of 2 nodes and gives the optimal level of efficiency up to the test cluster in which 50 nodes are employed. However, although the efficiency given by the "auth" mechanism is de-creasing, it remains to be evident in the test set consisting of 100 nodes. In the fulfillment of the test scenario, which was produced utilizing the group management of the previously constructed CIN resources, on the cloud environment, the results were acquired in a way that verifies the values received from the prior two settings.

As observed in the test results performed in the physical environment, it has been observed that the "auth" mechanism and the group management module developed from the group resource using 2 nodes to the group resource with 50 nodes, and the group management module, are similarly efficient at the optimum level in the

test conducted in the cloud environment. However, the benefit given declines when more than 50 nodes are employed, as in the actual environment. According to the data provided in Table 8, the resultant times rise in direct proportion to the number of nodes. In group resources where more than 50 nodes are employed, forming more than one group by splitting the nodes that make up the groups allows the efficiency offered by the group module and the "auth" method to be delivered at the optimal level.

3.1.5. Test Scenario 5: Computational Overhead of Auth Mechanism

It is of considerable importance that the established "auth" mechanism maintains its compliance with the technical standards given by oneM2M, as well as maintaining security by successfully completing AKA transactions on the IoT service platform.

Table 9
Creation of group administrator resources in the simulation environment

# Of nodes	Average (ms)	Minimum (ms)	Maximum (ms)
100	816	764	985
200	809	764	954
300	824	764	3276
400	823	764	1035
500	834	764	1120
600	1115	767	6516
700	904	776	3810
800	1098	795	2120
900	1161	999	1704
1000	1198	788	2051

Table 10
AKA transactions performed by the group header and members in the simulation environment

# Of nodes	Average (ms)	Minimum (ms)	Maximum (ms)
100	1,12	0	985
200	1,2	0	967
300	1,18	0	1032
400	1,09	0	1032
500	1,03	0	1071
600	1,08	0	988
700	1,11	0	1028
800	1,08	0	1081
900	1,24	0	1068
1000	1,18	0	1044

Table 11
Creation of group administrator resources in the physical environment

# Of nodes	Average (ms)	Minimum (ms)	Maximum (ms)
100	282	264	387
200	278	264	351
300	277	265	378
400	277	264	359
500	277	264	361
600	279	264	383
700	282	264	440
800	276	261	357
900	277	261	378
1000	277	263	374

Table 12
AKA transactions performed by the group header and members in the physical environment

# Of nodes	Average (ms)	Minimum (ms)	Maximum (ms)
100	3,92	0	356
200	2,19	0	361
300	1,63	0	358
400	1,34	0	355
500	1,1	0	354
600	1,04	0	356
700	0,95	0	348
800	0,87	0	358
900	0,8	0	354
1000	0,76	0	340

Table 13
Creation of group administrator resources in the cloud environment

# Of nodes	Average (ms)	Minimum (ms)	Maximum (ms)
100	273	268	383
200	272	268	357
300	271	268	356
400	272	268	367
500	271	268	413
600	271	267	426
700	271	268	360
800	271	267	360
900	270	267	386
1000	272	267	36

Table 14
AKA transactions performed by the group header and members in the cloud environment

# Of nodes	Average (ms)	Minimum (ms)	Maximum (ms)
100	3,82	0	362
200	2	0	357
300	1,61	0	398
400	1,27	0	392
500	0,98	0	363
600	0,92	0	391
700	0,78	0	356
800	0,73	0	360
900	0,76	0	452
1000	0,59	0	360

However, the small computational overhead of the “auth” process is vital for the system to be accepted and employed in subsequent investigations. In group-based AKA procedures, only the group leader also known as an administrator executes a full AKA process. However, for each HTTPS request sent independently, a complete AKA process must be executed. In addition, the computational cost while completing AKA transactions by other members of the group using the token value described as GCK, which happens after a full AKA transaction of the group header was assessed.

While creating the group header resource of the developed “auth” mechanism, an average set of values in the range of 800-1200ms was produced in the test scenario 5 made in the simulation environment, as shown in Table 9. Table 10 shows the results of AKA transactions made by the group header and its members using the “auth” mechanism in the simulation environment. According to the results shown in Table 10, the computational overload of the “auth” mechanism on the system was measured as 1ms on average. The column showing

the maximum values shows the values resulting from a full AKA operation of the group header. When test scenario 5 is performed in the physical environment, the results obtained decreased with the increase of the processor power, as determined in the results of the previous tests. The average values obtained according to the results indicated in Table 11 were measured in the range of 276-282ms. According to these results, efficiency between two and three times is provided in the test performed in the physical environment compared to the simulation environment. In addition, the computational overhead of the "auth" mechanism on the hardware used in the physical environment and running the IoT service platform was measured as 1.4ms on average. When the column showing the average values in test scenario 5 performed in the physical environment is examined, the average of the values formed is like the simulation environment. In addition, when the maximum values specified in Table 12 are examined, it is observed that the results are 3 times less than in the simulation environment. In the tests carried out in the cloud environment, it has been revealed once again that the computational overhead brought to the system by the "auth" mechanism is the lowest level. It has been observed that the values obtained as a result of AKA operations performed in the AWS environment and performed by the header of the group in Table 13 are 3 times more efficient than in the simulation environment. When columns showing the minimum and average values in Table 13 are examined, EC2 and network structure used in the AWS environment stand out as another result that was found to have the most efficient and stable bandwidth among the three different test environments.

As with other findings, the maximum numbers always indicate the time taken for the delivered value in response to the initial HTTPS request at the start of the tests. As in the previous results, the "auth" mechanism maintains the level of efficiency it provided, as the GCK value of the developed "auth" mechanism is used by other members of the group as a result of the request made by the group header, and the results obtained from the structure forming the second part of the test. Also, the average of the column showing the aver-age values in Table 14, the value is determined as 1.34ms.

According to the numbers obtained with test scenario 5 executed in three separate test settings, the additional computational overload given to the system by the "auth" mechanism, which carries out group-based authentication and authorization operations, is 1.13, 1.46, and 1.34 accordingly.

3.2. Test Methods

3.2.1. Test Method 1: Group-Based and Non-Group-Based AE Enrollment Process

The results of test scenarios 1 and 3 are compared to measure the efficiency of the created "auth" method, which leverages the authentication and authorization module of more than one AE resource during the registration phase of the IoT service platform. While making this comparison, instead of making the system secure using just ACPs, the mechanism was controlled by performing AKA actions at the position where the group feature was switched off in the "auth" mechanism.

As indicated in the part where test scenarios 1 and 3 are presented, the comparison was done based on the G-AKA study in addition to the relevant measures (Chen et al., 2012). In Table 15, the efficiency given by the "auth" method established in the place where the group feature is active is noticed.

Table 15
Test Method 1: Group-Based and Non-Group-Based AE Enrolment Process

# Of nodes	Average with Group Feature Off (ms)			Average with Group Feature On (ms)		
	Simulation	Physical	Cloud	Simulation	Physical	Cloud
100	975	504	1029	30	69	204
200	897	486	1047	26	62	206
300	909	495	1105	22	78	205
400	1088	483	1115	21	64	209
500	1021	492	1048	20	67	205
600	1111	481	1072	19	68	197
700	1115	478	1142	19	66	204
800	1120	480	1078	19	69	203
900	1138	478	1017	20	66	200
1000	1111	477	1036	22	66	203

According to the findings presented in Table 15, by integrating the built "auth" mechanism with the IoT service platform, the outcomes in the off and on group feature were compared. In tests done in simulation, physical

and cloud settings, 897-1138, 477-504, and 1017-1142ms intervals were measured while the group feature turned off, accordingly. However, when the group feature was active, 19-30, 64,78, and 197-209ms intervals were recorded, respectively.

According to these data, the "auth" method created to execute group-based authentication and authorization operations give 4 times efficiency in the worst-case situation. In the oneM2M IoT environment, signal traffic is decreased by employing the created "auth" method, while AKA transactions between the nodes in the field and IN are carried out in a secure environment.

3.2.2. Test Method 2: HTTPS Requests Made with Single and Group Resource

It is of vital importance that the identities of the nodes registered to the system using the "auth" method be validated from the minute they join the system to the moment they exit the system and that they may only do the transactions they are permitted to accomplish. Working with the group management module of the "auth" mechanism built in HTTPS requests utilizing CSFs offered by the IoT service platform, it has become feasible to make transactions on the system in a safe, efficient, and collective method.

In this manner, the test results acquired in test scenarios 2 and 4, in which the group-based AKA transactions of the previously developed CIN sources are active, but the group management module is tested in both active and passive positions, are compared. In test scenario 4, one HTTPS request was issued to get numerous CIN resources utilizing only one group resource. However, in test scenario 2, distinct HTTPS requests are issued for each CIN resource. Therefore, to directly compare test cases 2 and 4, the real average value of test case 2 is computed using equation 3.2.2a.

$$Unique, Real\ Average = Number\ of\ Nodes * Average\ Value \quad (3.2.2a)$$

$$Efficiency = Unique, Real\ Average\ /\ Average\ with\ Group\ (ms) \quad (3.2.2b)$$

For a group to be efficient, as shown in Table 16, at least two nodes must be included in the group. However, the efficiency value was calculated for each test set as shown in equation 3.2.2b.

Calculated efficiency values were carried out for three separate test scenarios: simulation, physical, and cloud environments. In the simulation scenario, the greatest efficiency is assessed as 1,570ms, which happens when a group resource consists of 2 nodes. However, when the administrative efficiency of the IoT environment is taken into consideration, this number is estimated to be 1,458ms which happens when a group resource with 25 nodes is deployed.

Table 16
Test Method 2: HTTPS Requests Made with Single and Group Resource

# Of nodes	Single Average (ms)			Average with Group (ms)			Efficiency		
	Simulation	Physical	Cloud	Simulation	Physical	Cloud	Simulation	Physical	Cloud
1	100	282	712	122	118	217	0,819	2,39	3,281
2	168	294	918	107	161	242	1,57	1,826	3,793
3	267	333	1071	200	204	247	1,335	1,632	4,336
5	375	455	1460	280	232	278	1,339	1,961	5,252
10	500	750	2480	579	353	341	0,863	2,125	7,273
25	1225	1550	5175	840	681	532	1,458	2,276	9,727
50	2650	2850	10150	2120	1185	1006	1,25	2,405	10,089
70	3150	3780	13720	2662	1450	1142	1,183	2,607	12,014
90	3960	4950	17640	3590	1876	1445	1,103	2,639	12,208
100	3600	5400	18900	3554	2033	1476	1,012	2,656	12,808

When the results of the values acquired in the physical and cloud environments are analyzed, the first aspect to be noted is that the network latency in the tests done in these two settings affected all HTTPS requests. According to the calculation done in 3.2.2a, it is shown that the most efficient condition in the cloud environment is 2,656 efficiency values created by a group resource consisting of 100 nodes. However, in the cloud environment, this number appeared with an efficiency value of 12,808 in the test set when a resource consisting of 100 nodes was utilized.

According to these results, the effectiveness of the "auth" mechanism created in small, medium, and large-scale application regions for IoT settings that are meant to be constructed as oneM2M-based differs. If the

hardware with the IN utilized in the IoT environment to be constructed has restricted technological characteristics, the outcomes will mirror the simulated environment. As the technological qualities of the hardware with the IN in-crease, the efficiency will grow accordingly.

3.2.3. Test Method 3: Computational Overhead of the Designed Auth Mechanism

The computational overhead that the "auth" mechanism established within the scope of the study provides to the system during the AKA activities done on the IoT service platform must be at a low level for the mechanism to be utilized and accepted in further studies. As indicated before, there is no module supplied on the IoT service platform that executes AKA transactions. For this reason, to assess the computational cost of the created "auth" mechanism to the system, the results of the AKA transactions performed by the group header and other members of the group acquired in test scenario 5 are compared.

Table 17
Test Method 3: Computational Overhead of the Designed Auth Mechanism

# Of nodes	Single Average (ms)			Average with Group (ms)		
	Simulation	Physical	Cloud	Simulation	Physical	Cloud
100	816	282	273	1,12	3,92	3,82
200	809	278	272	1,2	2,19	2
300	824	277	271	1,18	1,63	1,61
400	823	277	272	1,09	1,34	1,27
500	834	277	271	1,03	1,1	0,98
600	1115	279	271	1,08	1,04	0,92
700	904	282	271	1,11	0,95	0,78
800	1098	276	271	1,08	0,87	0,73
900	1161	277	270	1,24	0,8	0,76
1000	1198	277	272	1,18	0,76	0,59

When the group management module offered by the IoT service platform is not used, each result will be the same as the result a group header would obtain after a full AKA operation. As noted in Table 17, conducting a full AKA operation by a group header was measured, on average, in simulated, physical, and cloud settings, and it was observed that it took a time in the range of 800-1198, 276-282, and 270-273ms, respectively. As shown in Table 17, when the members of the group using the "auth" mechanism designed to perform AKA operations transmit the GCK value, the computational overhead imposed by the mechanism on the system is measured 1.13, 1.46, and 1.34ms on average in simulated, physical and cloud environments, respectively.

4. Conclusion

Within the scope of the study, the "auth" mechanism that executes group-based authentication and authorization procedures were established based on the Mobius IoT service platform, which was issued a oneM2M compliance certificate by the oneM2M worldwide organization and produced as open source by KETI. By combining the "auth" method and the group management module supplied by the IoT service platform together, the computational overload and signal traffic on the nodes in the field domain are greatly decreased. According to the findings of the test scenarios carried out, the computational overhead of the "auth" mechanism on the IoT service platform is in the range of 800-1198, 276-282, and 270-273ms for single transactions in simulation, physical, and cloud environments, respectively. It was assessed as 1.13 – 1.46 and 1.34ms on average for processes using the source. In the testing for accessing CIN resources, it is noticed that HTTPS requests with the group feature enabled give up to 4 times efficiency, starting from 2 nodes to the test cluster employing 50 nodes. For an IoT environment to be developed utilizing restricted resources in the OneM2M ecosystem, it is advised to form groups of 25 nodes, provided that one of them is the group header.

The signal traffic on the internet of things environment has dropped as a result of the deployment of group-based authentication systems. It is anticipated that the nodes energy usage will go down as a result of the nodes' effective communication with one another. In this study, group-based transactions, particularly authentication and authorization procedures, maintain a high level of security on the Internet of Things contexts while maintaining a measurably low overhead in computing and communication. In future studies, the "auth" mechanism built based on this study can execute group-based AKA transactions on a standalone server. Developing a

structure that can interact with more than one IoT service platform in the oneM2M ecosystem would boost the possibilities of interoperability across IoT service platforms.

Author Contributions

İbrahim Uğur Aba: Graduated MSc student. Collected data and performed the tests. Performed statistical analysis and wrote the paper.

Erhan Taşkın: Thesis supervisor. Conceived and designed the analysis.

Conflicts of Interest

The authors declare no conflict of interest.

References

Aydin, Y., Kurt, G. K., Ozdemir, E., & Yanikomeroglu, H. (2020). A flexible and lightweight group authentication scheme. *IEEE Internet of Things Journal, 7(10),* 10277-10287. Doi::https://www.doi.org/10.1109/jiot.2020.3004300

Apache JMeter. Retrieved from: http://jmeter.apache.org

Cao, J., Ma, M., & Li, H. (2015). GBAAM: Group-based access authentication for MTC in LTE networks. *Security and Communication Networks, 8(17),* 3282-3299. doi: https://www.doi.org/10.1002/sec.1252

Chen, Y., Wang, J., Chi, K., & Tseng, C. (2010). Group-based authentication and key agreement. *Wireless Personal Communications, 62(4),* 965-979. doi: https://www.doi.org/10.1007/s11277-010-0104-7

Define IOT. (2015, May 25). Retrieved October 22, 2019, from https://iot.ieee.org/definition.html

Fu, A., Song, J., Li, S., Zhang, G., & Zhang, Y. (2016). A privacy-preserving group authentication protocol for machine-type communication in LTE/LTE-A networks. *Security and Communication Networks.* doi:https://www.doi.org/10.1002/sec.1455

Gartner says 5.8 billion enterprise and automotive IoT endpoints will be in use in 2020. (2019, August 29). Retrieved from: https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io

Gartner says worldwide IoT security spending will reach $1.5 billion in 2018. (2018, March 21). Retrieved from: https://www.gartner.com/en/newsroom/press-releases/2018-03-21-gartner-says-worldwide-iot-security-spending-will-reach-1-point-5-billion-in-2018

IoT Analytics, state of the IoT 2018: Number of IoT devices now at 7B – market accelerating. (2018, August 08). Retrieved from: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b

IoT Analytics, state of the IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. (2022, May 18). Retrieved from: https://iot-analytics.com/number-connected-iot-devices

IoT OCEAN. (2017, July 9). Retrieved from: http://developers.iotocean.org/archives/module/mobius

Jiang, R., Lai, C., Luo, J., Wang, X., & Wang, H. (2013). EAP-based group authentication and key agreement protocol for machine-type communications. *International Journal of Distributed Sensor Networks, 9(11),* 304601. doi: https://www.doi.org/10.1155/2013/304601

RFC7519. (2015, May). Retrieved from: https://datatracker.ietf.org/doc/html/rfc7519

KETI. (1991, August). Retrieved from: https://www.keti.re.kr

Kim, J., Choi, S., Yun, J., & Lee, J. (2016). Towards the onem2M standards for building IoT ecosystem: Analysis, implementation, and lessons. *Peer-to-Peer Networking and Applications, 11(1),* 139-151. doi: https://www.doi.org/10.1007/s12083-016-0505-9

Lai, C., Li, H., Li, X., & Cao, J. (2013). A novel group access authentication and key agreement protocol for machine-type communication. *Transactions on Emerging Telecommunications Technologies, 26(3),* 414-431. doi: https://www.doi.org/10.1002/ett.2635

Lai, C., Li, H., Lu, R., & Shen, X. (2013). SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks. *Computer Networks, 57(17),* 3492-3510. Doi:https://www.doi.org/10.1016/j.comnet.2013.08.003

Lai, C., Lu, R., Zheng, D., Li, H., & (Sherman) Shen, X. (2016). GLARM: Group-based lightweight authentication scheme for resource-constrained machine-to-machine communications. *Computer Networks, 99, 66-81.* doi: https://www.doi.org/10.1016/j.comnet.2016.02.007

Li, J., Wen, M., & Zhang, T. (2016). Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-A networks. *IEEE Internet of Things Journal, 3(3),* 408-417. doi:

https://www.doi.org/10.1109/jiot.2015.2495321

Modiri, M. M., Mohajeri, J., & Salmasizadeh, M. (2018). GSL-AKA: Group-based secure lightweight authentication and key agreement protocol for M2M communication. *2018 9th International Symposium on Telecommunications (IST)*. doi: https://www.doi.org/10.1109/istel.2018.8661145

MongoDB: The application data platform. (2007). Retrieved from: http://www.mongodb.com

nCube-Lavender. Retrieved from: http://developers.iotocean.org/archives/module/ncube-lavender

nCube-Rosemary. Retrieved from: http://developers.iotocean.org/archives/module/ncube-rosemary

nCube-Thyme Nodejs. Retrieved from: http://developers.iotocean.org/archives/module/ncube-thyme-nodejs

OneM2M, the global community that develops standards for IoT. (2012). Retrieved from: http://www.onem2m.org

Parne, B. L., Gupta, S., & Chaudhari, N. S. (2018). SEGB: Security enhanced group-based AKA protocol for M2M communication in an IoT enabled LTE/LTE-A network. *IEEE Access, 6*, 3668-3684. Doi: https://www.doi.org/10.1109/access.2017.2788919

Padmashree, M. G., Mallikarjun, Arunalatha, J. S., & Venugopal, K. R. (2022). GKEAE: Group key exchange and authentication with ECC in internet of things. *Intelligent Systems*, 1-10. Doi: https://www.doi.org/10.1007/978-981-19-0901-6_1

Passport.js. Retrieved from: http://www.passportjs.org

Postman API platform. Retrieved from: https://www.postman.com

Su, W., Wong, W., & Chen, W. (2016). A survey of performance improvement by group-based authentication in IoT. *2016 International Conference on Applied System Innovation (ICASI)*. doi:https://www.doi.org/10.1109/icasi.2016.7539800

Şahinaslan, O. (2019). Encryption protocols on wireless IOT tools. *AIP Conference Proceedings*. doi: https://www.doi.org/10.1063/1.5095121

The alliance for internet of things innovation. (2019, October). IoT LSP Standard Framework Concepts, Release 2.9 AIOTI WG03 - IoT Standardization

TR-0025 Technical Report. (2018, March 12). TR-0025 V2.0.2 Application Developer Guide.

TS-0001 Technical Specification. (2016, August 30). TS-0001 V2.10.0 Functional Architecture.

TS-0003 Technical Specification. (2018, March 12). TS-0003 V2.12.1 Security Solutions.

TS-0009 Technical Specification. (2016, August 30). TS-0009 V2.6.1 HTTP Protocol Binding.

Yao, J., Wang, T., Chen, M., Wang, L., & Chen, G. (2016). GBS-AKA: Group-based secure authentication and key agreement for M2M in 4G network. *2016 International Conference on Cloud Computing Research and Innovations (ICCCRI)*. Doi: https://www.doi.org/10.1109/icccri.2016.15