IoT Framework for Effective and Fine–Grain Access Control

Zina Houhamdi
Cybersecurity Department
College of Engineering, Al Ain University
Al Ain, United Arab Emirates
zina.houhamdi@aau.ac.ae

Belkacem Athamena
Business Administration Department
College of Business, Al Ain University
Al Ain, United Arab Emirates
belkacem.athamena@aau.ac.ae

***Abstract*—The standardized protocols and new generation of hardware for communications increased interoperability more than ever. However, the increasing interoperability causes a rise in offensives from vicious people and hardware. Therefore, there is a necessity to apply encryption algorithms to guard the communications between clients and servers. Nevertheless, the current encryption techniques do not protect the service access at a fine–grain level. Furthermore, in wireless sensors and actuators, each network endpoint is integrated constrained–resource; thus, the interoperability increase necessitates a high computation rate. On the other hand, the endpoints inherent to processing and memory restrictions negatively affect communication delays and power consumption, leading to a shorter battery lifetime. Consequently, there is a need for new methods to increase interoperability, dependability, scalability, security, and energy efficiency. This study proposes a theoretical design of a new and effective IoT model that supports authentication, authorization, and fine grain access control with no network configuration and dynamic reconfiguration. The proposed framework demonstrates the possibility of the integration of IoT devices powered by batteries and a functional System of Systems.**

Keywords— internet of thing, interoperability, communication security, access control

I. INTRODUCTION

The main feature of application in industry is interoperability, which minimizes the cost and maintenance of the tasks since the model improvement by increasing its interoperability necessitates low effort and cost compared to the improvement of a non–interoperable model. Moreover, there is a possibility of integrating distinct interoperable models to allow service, data, and resource sharing without duplication. An interoperable model supporting different kinds of a device takes advantage of the best characteristic of a particular device in different circumstances, such as collecting information from one device and managing this data separately on a powerful server. These useful properties are very beneficial in industrial applications; however, in a context where the Internet of Thing (IoT) devices transmit sensitive data or offer actuators access, interoperability is highly risky. Thus, security is a key issue for the deployment of IoT models in the industry. Mainly, security is essential for constrained–resource devices, especially for devices powered with a battery. The implementation of the security method increases unavoidably the consumption of power; thus, a new framework strikes an appropriate balance between power consumption and security. In Service-Oriented Architecture (SOA), services are available to anyone on the network, and each network endpoint operates as a Service Provider. The protection of each service against unauthorized access requires some mechanisms. Accordingly, the security concerns are decomposed into two aspects: Access Control and Communication Security. This paper overviewed the existing methods of communication security in IoT and suggested a new approach that enables effective, fine-grain access control. The remaining of this paper is organized as follows. Section 2 introduces the IoT and its historical evolution until the WSANs. Sections 3 and 4 propose a solution to communication security and access control to make IoT technology applicable in the industry by concentrating on security and efficiency issues.

II. INTERNET OF THINGS

The IoT concept is hard to define; thus, different research domains define the IoT concept differently. The IEEE IoT team collected definitions from several Internet research teams and organizations in [8]. This study defines the IoT as: "IoT devices are constraint–resources integrated systems capable of carrying out multiple and clear operations, for example, interaction processing, signal processing, and sensing. Generally, IoT is battery-powered and possesses wireless communication abilities". The definition of IoT concept is updated with software and hardware development. Accordingly, a brief historical evolution of IoT devices is introduced along with Wireless Sensor Networks description. Finally, some application examples of IoT are given.

a) *Historical Evolution*: the evolution of the IoT concept begins with the establishment of Internet Protocols. The technology evolutions concern software and hardware. For the software, the IoT includes various components; however, the more significant evolutions are carried out in operating systems (TinyOS, FreeRTOS, Contiki, Embedded Linux, OpenWS, RIOT), link–layer (Bluetooth and WiFi, 6LoWPAN) and application protocols (RESTful HTTP, MQTT, Jabber an open-source community, XMPP, MQTT–SN, WebSockets, Constrained Application Protocol). For the hardware, over the last ten years, there has been a burst in the applications of embedded devices for industry goals and different commercial items, like smartwatches, mobiles, and computers have encouraged the production of several different devices types (such as Microcontrollers and microprocessors, Wireless technologies, Sensors)

b) *Wireless Sensor and Actuator Networks* (WSAN): were created during the actuator's integration in industry and home Wireless Sensor Networks (WSNs). Nevertheless, the incorporation of the actuators necessities important modifications in the WSN framework. The actuator needs information concerning the task to be executed. Thus, the actuator requires the ability of information reception. In implementing this property, the framework should allow communication in two ways (from servers to actuators and from actuators to servers). To enhance actuators usage, a

WSAN node exploits information from various sensors to determine its actuator actions; therefore, the WSAN necessitates the M2M communications. Furthermore, the integration of the IP into a WSAN changes nodes to IoT devices; but, as reported in many kinds of research, still without the IP, WSAN nodes are considered as IoT devices.

c) *CoAP*: is a specialized IP application that provides web–services working with constrained resource devices. CoAP is efficacious for microcontroller devices with small ROM and RAM sizes and runs over the 6LoWPAN network with high packet error rates. The protocol allows network endpoints to move to the sleeping mode in order to increase battery life because CoAP is aimed at networks with low power consumption. CoAP uses the client/server model, provides the current service discovery, contains Key-Web concepts (for instance, RESTful [10] and URIs [2]), and includes extendable header alternatives. CoAP interfaces easily with HTTP for web integration. CoAP performs through User Datagram Protocol, contrary to HTTP that runs over TCP. Actually, many research teams carrying CoAP to perform through TCP, such as [3].

d) *SOA*: is a designed framework that creates and develops systems based on decentralized modules. Each module offers and consumes a set of services in order to execute its activities. The primary advantages of SOA are flexibility, reusability, and scalability. This design allows to decompose a complex system into a set of simple modules to make its implementation, validation, and verification fast and easy since each module is implemented separately. The system extension or update necessities only the increase of modules number. In IoT industry domain, the SOA facilitates better utilization of scalability and interoperability. Supervision and management in the industry are considered complex tasks that should be decomposed into simple modules. Each module represents an IoT device that performs a single and simple task, for example, measuring a variable or offering a service. In industrial applications, the important modules are duplicated or substituted. SOA needs several complex methods to operate, containing configuration, orchestration, and Quality of Service. The presented study is conducted to research, develop, and improve interoperability in complex industry contexts.

III. COMMUNICATION SECURITY

The communication over the network requires security to protect against multiple kinds of assaults, particularly forging packets, sniffing attacks, "man-in-the-middle" attacks, and "Denial–of–Service" attacks. The model described in this paper uses CoAP and is ideally suited for 6LoWPAN network. To preserve interoperability, security methods should be normalized. Therefore, we need to analyze existing standard methods that are compatible with 6LoWPANCoAP stack. A summary of such analysis is described in [13].

A. End–to–End Security Method

Interoperability allows communication between multiple devices. Generally, this communication necessitates the utilization of additional intermediate devices, for example, router, switch, and server. Accordingly, the security of end–to–end communication is mandatory. The leading methods providing end–to–end security are Datagram Transport Layer Security (DTLS) and Internet Protocol security (IPsec).

DTLS is mainly a User Datagram Protocol based on the Transport Layer Security (TLS) protocol, which provides protection to communication over computer networks by providing the data source authorization, authentication, data confidentiality, and integrity. Originally, DTLS was composed of two stages. The first stage is a connection between two communicated computers, where both machines authenticate themselves, and the second stage is the transfer of the encoded information. Nevertheless, the initial edition of DTLS is inefficient for constrained resources since the certificates used and the overhead decline the low-power performance. Consequently, a compacted DTLS was developed [5], and the certificates were replaced by the key [6], creating a conventional and effective DTLS edition.

IPsec is the secure version of the IP. It represents coordination between multiple distinct protocols and supports different encryption types [7]. IPsec incorporates two methods: Authentication Header (that authenticates and protects the data source against "man–in–the–middle" attacks and the integrity of independent data) and Encapsulating Security Payloads (that support data privacy). In case data privacy is a primacy issue, then Encapsulating Security Payloads is a rational option. Encapsulating Security Payloads encodes the initial IP packet into the new IPsec packet payload, which will be decoded merely based on the accurate earlier negotiated or utilized keys. For the keys' negotiation, IPsec provides the Internet Key Exchange protocol, version 2 [9] that is beneficial to avoid the utilization of long-term and static keys, therefore increasing the security.

B. Access Control Review

The methods mentioned in the previous section to secure End-to-End communications support multiple services to control the devices' access. However, these methods support device access at different levels (lack of granularity). Table I presents an overview of the access control type provided by different technologies. The access control by ID and Address enables the supervision of who is allowed to interact with the service provider. At the Method level, the access control provides services with distinct utilities according to users' type, such as the administrator can update the service time while the normal user obtains the time. Finally, access control at the Service level allows the creation of customized services for individual users and users' types. Consequently, the application of fine-grain access control is necessary.

TABLE I. ACCESS CONTROL ANALYSIS

	Access Control			
Technologies	**Fine-grain**		**Large grain**	
	Method	**Service**	**ID**	**Address**
IP	No	No	No	No
IPsec	No	No	No	Yes
IPsec + IKEv2	No	No	Yes	Yes
DTLS	No	No	Yes	No
CoAP	No	No	No	No
Black-list	No	No	No	Yes

IV. ACCESS CONTROL

Access control is a vital aspect of data security. It allows access based on the presented credentials. It monitors service demands sent to a particular Service Provider and manages the communication approval. It also enables identifying the client of service and providing the appropriate information related to that client to the service to allow the opportunity of the provision of customs services. Again, there is a need for

standard methods to preserve interoperability. Nevertheless, existing methods suffer from limitations. Accordingly, we need to implement a new effective Access Control model applicable in IoT context. In the following, we summarize the existing access control methods along with their limitations and suggest an effective access control method.

A. Standard Solutions

RADIUS and Kerberos are the most well-known standards. They are protocols providing the functionalities of access control. However, they have different principles and work differently. They offer some advantages and disadvantages, which are utilized to create our proposed effective access control method.

a) *Remote Authentication Dial-In User Service* (RADIUS): RADIUS is a client/server protocol and software that allows remote servers to interact with a central server for authenticating and authorizing Dial-In users requesting access to a service or a system. The process of access control involves three modules: RADIUS Server, Service Consumer, and Service Provider. The Service Consumer claims a particular service from the Service Provider, which requests the authentication information to verify the access state with the RADIUS Server. The latter responds with three possibilities: Accept, Reject, and Challenge. The challenge answer requires more information from the Service Consumer to the RADIUS Server through the Service Provider. The authorisation and authentication processes in RADIUS use simple encryption algorithms, and communication requires an insignificant quantity of transferred data. Consequently, RADIUS is a suitable and effective protocol for IoT devices regarding the complexity and the processing; however, RADIUS necessitates numerous interactions, particularly for the Service Provider, which jeopardizes the low–power principle.

b) *Kerberos*: Kerberos is an authentication protocol that runs over User Datagram Protocol and works according to tickets allowing communication through an unsecured network. The access control process involves three modules: Key Distribution Center, Service Consumer, and Service Provider. Every object possesses its private key, except the Key Distribution Center, which has all keys. In the beginning, the Service Consumer requests a ticket by sending a partial encoded message. The Key Distribution Center uses the Service Consumer's key to decode the message. In case of a successful scenario, the Key Distribution Center returns an encoded packet with timeout and other parameters. The Key Distribution Center stores this packet and uses it for requesting an authentic ticket from the Service Provider. Thus, the Service Consumer creates a ticket, encoded by the Service Provider's key that the Key Distribution Center uses to access the Service Provider. Finally, the Service Consumer requests access using this ticket, and the Service Provider uses its key to decode the ticket and extracts all information regarding the Service Consumer and the access control policy. Kerberos protects passwords that are not communicated. It uses a centralized Key Distribution Center, which makes the maintenance of the database convenient. Kerberos eliminates the interaction between the Service Provider and the Key Distribution Center; thus, it supports the low-power criterion. Nevertheless, Kerberos is not perfect for an IoT device because of the ticket size (the ticket contains the entire encrypted information) and processing complexity.

c) *Need for IoT standard solution*: in IoT, the devices are resources with limited processing ability. Any processing improvement increases power consumption, which is a crucial restriction for battery devices. Furthermore, wireless communications increase to a great extent power consumption. RADIUS necessitates a lot of message exchanges, particularly on the service provider's side, which also increases the device's power consumption. On the other hand, Kerberos needs to utilize a significant number of encrypted tickets (including data such as services, time, client, etc.). The ticket processing and transmission require considerable energy, and the model with limited communications is an inefficient solution. Consequently, the existing standard solutions are not suitable for implementation in IoT. Thus, a different and more effective method is needed to ensure energy efficiency and access control at fine–grain.

B. Ticket–based Access Control

The access levels monitoring defines the granularity of the access control method. Here we propose a fine-grain approach to control the access that allows multiple access to be monitored by end-user, method, and service. This control level is impossible with existing technologies (see table 1). We suggest using the CoAP as an application protocol by offering resources (services) accessible by several techniques, for example, DELETE, PUT, POST, and GET. Thus, we worked on a new approach, which is barely explained in [4]. This approach objective is to restrict the supplementary communication overheads of CoAP, which increases the power consumption or raise communications delay. To achieve this objective, we designed a hybrid approach for access control over CoAP, combining together the authorization/ authentication methods of RADIUS and the ticket system of Kerberos. CoAP offers different packet alternatives. The idea is the utilization of one option to transmit the ticket information. Contrary to Kerberos, the ticket, includes only required information, which is a set of bytes to determine the identities (servers and clients). The ticket size is dependent on the implementation and symbolizes a balance between the performance of the power consumption and security level. Consequently, this approach centralizes the tickets validation and authentication for decentralized services. This approach supports either several access control methods for particular applications or a centralized method to avoid inconsistency, thus to improve the system extensibility. In order to minimize the overheads on the IoT devices, the authorization and authentication procedures are modeled as CoAP services.

1) *Approach Requirements*: The suggested protocol supposes the availability of the following information for the 3A server and the IoT devices:

- *Id*: it represents the identity of the device.
- *Password*: it is a common characteristic of the 3A server and the IoT device. It is never sent during the Authorization and Authentication processes.
- *Confidential Key* (CK): it is a set of 16 bytes known by the 3A server and the device.

2) *Approach Description*: The suggested access control method contains two separate phases, particularly, Authorization and Authentication. These phases are performed through distinct services on the 3A server.

a) *The Authentication Process*: is executed by all IoT devices monitored by the access control method, comprising

consumers and providers. This phase guarantees device identification by the 3A server. The server creates a separate ticket for each device. This ticket is the identification tag used in succeeding information transmissions between the 3A server and other entities. Figure 1 illustrates the authentication process where the device initiates the process by sending to the 3A server a GET request (possessing MAC and IP addresses). The Challenge-Response process starts. The 3A server generates an authenticator, which is a set of 16 bytes and valid only for the next 15 sec. After the reception of the authenticator, the IoT device encrypts the password according to RADIUS process for Challenge-Response. The Confidential Key (CK) and the authenticator lengths are 16 bytes. In case the CK size is small (<16 bytes), we fill the leftover values with 0. The CK is returned to the 3A server along with the device Id; then, the 3A server reiterates the algorithm and matches the two outcomes. If the CKs match, the 3A server generates a ticket, defines a timeout, and returns it back to the IoT device to complete the Authentication process.

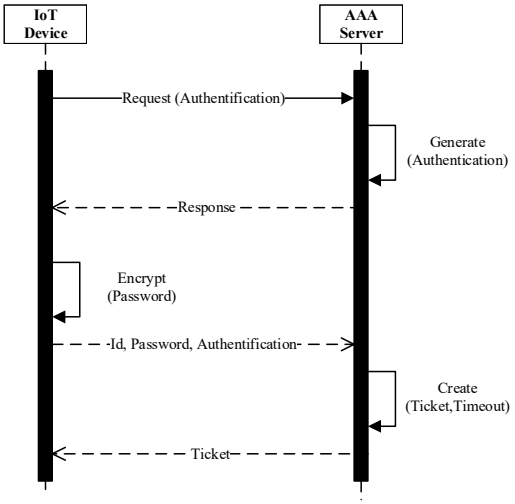

Fig. 1. Authentication sequence diagram.

b) *The Authorization Process*: is performed by the service provider to identify the service consumer or to implement doubled authentication where the IoT device performs the authorization process to check the validity and trustworthiness of the service provider. Initially, the IoT device sends a request to the 3A server asking for the validness of the ticket. The request contains the ticket in CoAP format, the ticket in the payload, and the IP address. In the successful scenario, the 3A server returns a validity confirmation as well as the IoT device name, latest-login, expire time, protocols, ticket timeout. At this stage, two possibilities to handle access approvals: the appropriate policies are incorporated in the Authentication request, or distinct user types with various rights are established. The last option is implemented and tested, and it is more effective, however, less manageable than the policies incorporation option. Figure 2 shows the authorization process.

c) *The Accounting Process*: works either by access instances or by time. The access instances type restricts the allowed number of accesses to a particular for a specific time (for example, allow the number of access to a particular service to 10 times for 30 minutes). Thus, if the access number attains the limit or if the timeout window expires, then these situations should be reported to the 3A server. On the other hand, the accounting by time enables the provider to offer services to an IoT device for a limited time, and then the authorization access expires, and this should be again reported to 3A server by the service provider. The Accounting phase characterizes the access to a particular service regarding the number of accesses or access duration. The Accounting phase is represented here as an exciting quality for business models.

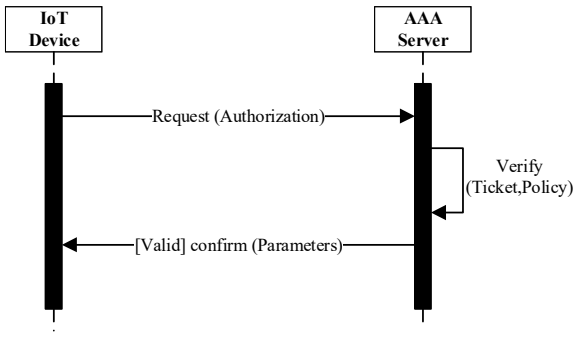

Fig. 2. Authentication sequence diagram.

3) *The Ticket Information*: The ticket aims to minimize power consumption and communication overheads whenever it is feasible by applying simple procedures. Consequently, the actual ticket implementation is basically 64 bits produced by the 3A server. The ticket is unique. The IoT device in the network is determined by its ticket information. The ticket information is defined by a hexadecimal number making it clear for humans. In case of the success of a Challenge-Response process of a ticket request, the 3A server replies with the ticket and the timeout. The timeout defines the ticket validity. The Authentication process needs the utilization of encoded channels. Therefore, the static ticket usage is not problematic; furthermore, the minimized number of communications, and the ticket timeout assist its protection. Nevertheless, to increase the protection of the system, the ticket can be dynamic. Accordingly, the ticket consists of hashed data of the initial ticket and other parameters (such as message Id) for each transmission between providers and consumers. In this study, the dynamic ticket is not implemented because we suppose the confidentiality trust at the IPsec level.

4) *Decentralized Access Control*: A network with Consumer, Provider and 3A Server is the simplest schema for access control method. Figure 3 illustrates the three alternative schemas for a successful service request:

- Service request without access control: The Provider delivers the service with no supplementary processing.
- The consumer's first attempt or an expired ticket: The Provider validates the ticket before offering the service.
- The service request from the consumer with a valid ticket: The Provider demands just to verify the timeout ticket and then offer the service.

The integration of CoAP and RADIUS protocols makes the authentication process more flexible. This model does not need support for the RADIUS on the consumer. The required resources and the overheads are reduced in comparison to the application of the protocols together simultaneously. This is particularly essential for the sensor with constrained

resources. Actually, it is possible to convert RADIUS packet to RADIUS–CoAP packet. Here we propose two alternatives: compacted RADIUS–CoAP packet and CoAP packet with a RADIUS payload. The compaction excludes duplicated information like the Length fields, Identifier, and Code, that are directly included in the CoAP Code and Id fields.

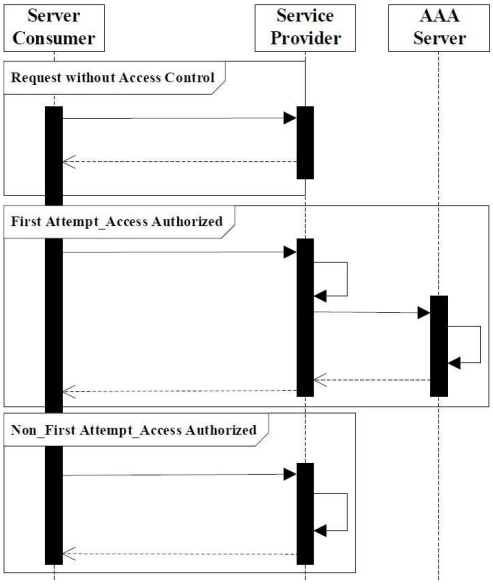

Fig. 3. Access control schemas.

a) *Multiple Protocol Framework*: The study focus on interoperability maximization in the industry context. The objective is to provide an intelligent method to commute services transparently between IoT devices, which possess distinct properties, meanings, protocols, etc. Such as, the IoT devices interact using CoAP protocol while one consumer service communicates using MQTT. The existing standards do not allow communication between these devices, and the current access control methods are ineffective for these technologies.

The access control method described here is appropriate for applications with multiple protocols communication using a converter representing a dependable mediator. The suggested 3A server is intended to deal with requests from distinct protocols (see Figure 4).

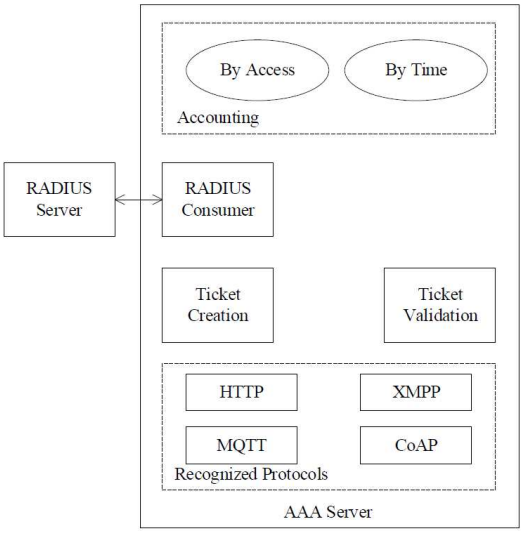

Fig. 4. 3A server framework with multiple protocols.

During the communication between two entities, the entities use different protocols represented by different colors (black and red). The access control procedure is similar to communication with a unique protocol (see Figure 3), with the addition of the converter that directly links the Consumer and the Provider (see Figure 5).

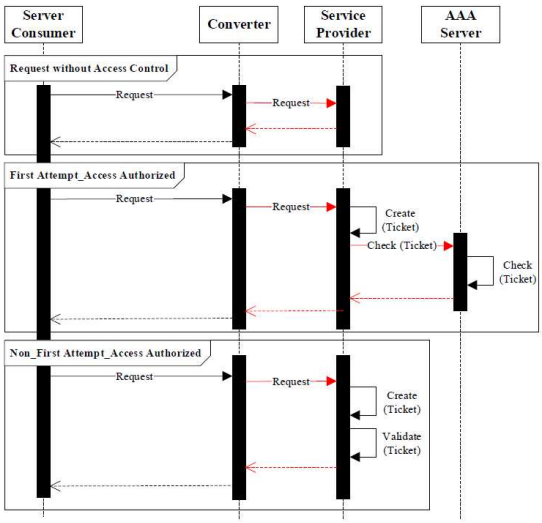

Fig. 5. Access control schema with multiple protocols.

The protocols' conversion is not discussed in this study since it is irrelevant to the presented outcomes. Accordingly, the converter is addressed as a black box allowing the conversion between two different protocols. Also, the multiple protocols communication implementation is represented as an extension of our model, together with an investigation on methods for avoiding the "man-in-the-middle" attack while the converter is used.

V. Similar Works

The ACE (Authentication and Authorization for Constrained Environments) team focuses on developing solutions for access control for constrained–resource devices. This ACE team developed OAuth 2.0 and OSCOAP by using CBOR (CBOR Object Signing and Encryption (COSE)) as a semantic protocol to minimize the message size [11]. COSE defines the representation of the confidential keys in CBOR format and also specifies the signature and processing, encryption, and message authentication code.

OAuth 2.0 [1] which is an access control method defining the Authorization and Authentication model that enable the consumer to get limited use of specific resource provided by a Service Provider. OAuth 2.0 needs a dependable me- diator server providing "intraspect" and "token". OAuth 2.0 is similar to ticket–based access control in the authentication and authorization resources. However, OAuth 2.0 differs from ticket-based access control in two main points: (1) OAuth 2.0 utilizes an access token rather than a ticket. The token, named a Proof-of-Possession, includes encoded data that are legible by only the AA server and the provider. (2) the absence of communication between the AA server and the service provider since the access rights are encrypted in the token. Initially, the consumer asks for an access token by specifying an access type; if allowed, the AA server creates an encrypted token where the provider knows the key.

OSCOAP [12], an Object Security of CoAP that is end-to-end security method that extends CoAP communication by

adding a supplementary protection layer. In addition to replay protection, encryption, and end-to-end security, the OSCOAP verifies the message's integrity. The OSCOAP main idea is the encapsulation of the header, CoAP payload, and different alternatives into a COSE encrypted object. This later represents a new CoAP packet payload.

VI. Conclusion

The IoT is a discipline that concerns integrating several research areas, for instance, design, big data, cloud computing, information security, machine learning, hardware, sensors, actuators, networking protocol, and wireless communications. The multidisciplinary character of IoT necessitates collaboration between researchers with different experiences and settings. This study represented an extension of existing IoT standards, particularly in security. This paper presents a significant improvement that focuses on a new method for the access control to constrained–resource IoT devices, eventually, in the suggestion of a CoAP–based networks access control method that is effective in energy consumption and allows fine-grain access control. This paper gives a short description of IoT and discusses its advantages. Also, it describes the new challenges to be tackled for IoT technology for its applicability in industrial applications. Finally, the paper attempts to propose a solution to these issues to apply IoT in the industry in the limelight of security issues.

The model proposed in this paper shows the theoretical validity of employing IoT technology in the industry. Consequently, the implementation of the access control method must be proved and validated in several situations and for multiple goals, for example, mobile machine monitoring (Arrowhead), mining conveyor belts, and smart rock bolts (IPSO Challenge). Also, the proposed model represents a foundation for further researches, such as Quality of Service, scalability, and efficiency.

References

[1] Bertin, E., Hussein, D., Sengul, C. and Frey, V., "Access control in the Internet of Things: a survey of existing approaches and open research questions," Annals of Telecommunications, vol. 74, no. 7-8, pp.375–388, 2019.

[2] Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B. and Raymor, B., "(constrained application protocol) over TCP, TLS, and ," Internet Requests for Comments, RFC Editor, RFC, 8323, 2018.

[3] Bormann, C., "A TCP and TLS Transport for the Constrained Application Protocol (CoAP)," draft-ietf-core-coap-tcp-tls-01, 2016.

[4] J. Delsing, Ed., "Arrowhead Framework: IoT Automation, Devices, and Maintenance," CRC Press, 12 2016.

[5] Tschofenig, H. and Fossati, T., "Transport layer security (tls)/datagram transport layer security (dtls) profiles for the internet of things. In RFC 7925. Internet Engineering Task Force, 2016.

[6] Halabi, D., Hamdan, S. and Almajali, S., "Enhance the security in smart home applications based on IOT-CoAP protocol," In 2018 Sixth Inter- national Conference on Digital Information, Networking, and Wireless Communications (DINWC), pp. 81-85, IEEE, 2018.

[7] Hoffman, P., "suites for," RFC 4308 (Proposed Standard), 2005.

[8] IEEE Internet of Things, "Towards a definition of the internet of things (iot)," IEEE, Tech. Rep., 2015.

[9] Kaufman, C. and Perlman, R., "Key exchange in IPSec: analysis of IKE," IEEE Internet Computing, vol. 4, no. 6, pp.50–56, 2000.

[10] Schiekofer, R. and Weyrich, M., "Introduction of Group–Subscriptions for RESTful OPC UA clients in IIoT environments," In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1355–1358, IEEE, 2019.

[11] Schaad, J., "object signing and encryption (cose)," RFC 8152, DOI 10.17487/RFC8152, July 2017, https://www. rfc-editor. org/info/rfc8152.

[12] Randhawa, R.H., Hameed, A. and Mian, A.N., "Energy efficient cross-layer approach for object security of CoAP for IoT devices," Ad Hoc Networks, vol. 92, p.101761, 2019.

[13] Gomes, T., Salgado, F., Pinto, S., Cabral, J. and Tavares, A., "6LoWPAN accelerator for Internet of Things endpoint devices," IEEE Internet of Things Journal, vol. 5, no. 1, pp.371–377, 2017.