Journal of Telecommunications and the Digital Economy

A Proposal for Dynamic and Secure Authentication in IoT Architectures Based on SDN

Younes Abbassi
Hassan II University, Casablanca, Morocco

Hicham Toumi
Chouaïb Doukkali University, El Jadida, Morocco

El Habib Ben Lahmar
Hassan II University, Casablanca, Morocco

Abstract: The connectivity of private resources on public infrastructure, user mobility, and the advent of new technologies have added new client and server-side security requirements. Security is the major element of the Internet of Things (IoT) that will certainly reinforce an even greater acceptance of IoT by citizens and companies. Security is critical in this context given the underlying stakes. This paper aims to advance the thinking on authentication of connected objects by proposing an authentication mechanism that meets the needs of IoT systems in terms of security and performance. It is based on SDN (Software-Defined Networking), which refers to a set of advanced technologies that allow for centralized control of network resources. OTP (One-Time Password) is a type of authentication that could be useful in connected object environments and smart cities. This research work extends the principle of OTP and proposes a lightweight authentication method using a new approach to OTP generation that relies on two parameters (Two-Factor Authentication, 2FA) to ensure the security of underlying systems. Subsequently, we leverage the combination of SDN and the 2FA algorithm to propose an adaptive authentication and authorization solution in the IoT network.

Keywords: Internet of Things (IoT), Software-Defined Networking (SDN), One-Time Password (OTP), Two-Factor Authentication (2FA).

Introduction

Internet of Things (IoT) and Cloud Computing are actually broad areas of interest and research. They will change the way we live and work by making different aspects of life smart. According to IoT Analytics estimates by Koohang *et al.* (2022), at the end of 2018 there were around 50.1 billion users connected to IoT devices.

The Internet of Things opens up new possibilities, such as the capacity to remotely monitor and manage devices, as well as to analyze and act on data from multiple real-time traffic data streams. Consequently, with the advent of cloud computing (Junior *et al.*, 2021), IoT products are changing habits and cities by enhancing infrastructures, improving municipal services to make them more effective and cost-effective, improving transportation services by reducing traffic congestion, boosting citizen safety, and delivering smart health services (Mitake *et al.*, 2021). IoT technologies, on the other hand, provide a slew of security and privacy concerns. Due to the hardware limitations of IoT devices, implementing and deploying complete and effective security and privacy solutions for the IoT environment remains a major challenge.

The IoT's immense power comes from the fact that its objects can communicate, analyze, process, and manage data without the need for human interaction. However, security issues are holding back the evolution and rapid deployment of this high technology. Identity theft, information theft, and data modification are a real danger for a parent system. Several cyberattacks have been blamed on flaws in the authentication procedures of connected door locks, computers, and phones. In 2016, Anna Senpai developed Mirai (Biggs, 2016), a malicious malware that allows attackers to take control of susceptible connected equipment such as surveillance cameras and routers, launching large-scale distributed denial-of-service (DDoS) assaults. Mirai turns infected objects into bots; in other words, it turns them into autonomous and intelligent computer agents controlled remotely.

In 2017, another malicious program, BrickerBot, appeared. It brute-force attacks objects using conventional password identification systems (Lagane, 2017) in order to eliminate them and thus delete their data. It is evident that the prosperity of the IoT can only be achieved when good security is provided for the objects and the communication networks used. It is crucial to implement a security policy that prevents any malicious or unauthorized object from accessing the IoT systems, reading their data, or modifying them. For an object to have the ability to operate a service or associate with a network, it must first prove its identity and have the necessary access rights. Connected objects are generally very limited in computing and storage capacity. They are also constrained by energy consumption. Therefore, we cannot use classical security mechanisms, such as authentication with digital certificates, or the use of asymmetric cryptographic asymmetric cryptographic algorithms, such as Rivest Shamir Adleman (RSA) or Diffie-Hellman (Kocher, 1996), because they are very costly or not even supported by the objects. As a result, a new lightweight and robust mechanism must be created to provide object authentication and data protection services, while adapting to object and communication technology capabilities.

This article describes a security system to ensure the authentication services of connected objects, the integrity of the inter-exchanged data, and the confidentiality of the information. This approach must take into account the constraints of the objects and the communication technologies used.

We used the SDN (Software-Defined Networking) technology in conjunction with the 2FA (Two-Factor Authentication) method based on the OTP (One-Time Password) algorithm in order to accomplish this goal. The remainder of the paper is laid out as follows. We first provide some background information on IoT, Cloud Computing, SDN, Access Control, and Authentication, as well as their principles. Then, we describe our proposed solution, its objective, and its functioning. We then provide a discussion of our findings, and, finally, we conclude this paper with some perspectives and further insights.

The objective of our IoT authentication solution is to enable intrusion detection (to prevent identity theft in a virtual environment by using mobile agents to collect malicious data and generate new signatures from this malicious data). We highlight IoT access control and authentication devices that can dynamically recognize and connect multiple real and virtual sensors in a unified secure system, and provide dynamic deployment of updates between clusters in an IoT cloud, using master and slave SDN.

What makes our work unique is that we use SDN architecture for dynamic deployment of a strong authentication mechanism and generation of inter-cluster updates in an IoT network, a framework never before dealt with by researchers, today offering access to the IoT network by registering and verifying the identity of objects, as well as updating the framework automatically.

Theoretical Foundations and Related Research

Internet of Things (IoT)

The CERP-IoT "Cluster of European Research Projects on the Internet of Things" represents the Internet of Things as "a dynamic backbone of a global network. This global network has self-configuration capabilities based on interoperable communication standards and protocols. Physical and virtual items in this network have identities, physical attributes, virtual personalities, and intelligent interfaces, and they're all perfectly interwoven" (Botta *et al.*, 2014).

Internet of Things is a continuously evolving system of interconnected devices based on a set of technologies, namely RFID, Barcode, Zigbee, WSN, Wi-Fi and Cloud Computing. It faces several challenges, of which security is a major challenge. The scope of application of IoT is almost unlimited, which will allow it to make the environment intelligent and favourable to any human activity (Abdellatif *et al.*, 2022).

Cloud Computing

The National Institute of Standards and Technology (NIST) claims that (Sturm, Pollard & Craig, 2017), Cloud computing is a concept that allows users to access a shared set of computer resources (such as servers, storage, and apps) on-demand over a telecommunications network. Cloud computing is a model that allows users to use a shared set of computing resources on demand (e.g., servers, storage, apps) that may be immediately put to use over a telecommunications network. There are four different kinds of clouds:

1) Public Cloud: dedicated to the general public, it is a set of free or paid services accessible via the Internet. It is offered by a company that manages an infrastructure that belongs to it.
2) Private Cloud: a set of resources available to a single customer that can be managed by the user company or by an external provider.
3) Community Cloud: cloud resources shared by several companies or organizations which can be managed by member organizations or by an external provider.
4) Hybrid Cloud: allows the company to be able to supply services in multiple clouds either public, private or community.

Some services offered by Cloud Computing are SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service), and NaaS (Network as a Service) (Hussain & Chun, 2022). IaaS provides computing and storage services on a rental basis; in addition to the data storage in IaaS, the data will be universally accessible on the Internet. PaaS offers a complete environment for the development and deployment of applications. SaaS allows remote access to software via the Internet. Users can have heterogeneous networks with NaaS since it gives virtual network(s) to them.

SDN (Software-Defined Networking)

SDN is a technology that is now mature, usable and “marketed” by operators. It allows centralizing the control logic in a controller. It also enables the separation of the control and data planes (Munther *et al.*, 2021). SDN is characterized by the following elements (El Kamel, Eltaief & Youssef, 2022):

- Separation of control plane and data plane: Control functionality is removed from the network nodes, which become simple (packet) forwarding elements.

- Forwarding (packet) decisions are flow-based (rather than destination-based):A flow is a collection of packets that travel from one point to another. All packets in the same flow are subject to identical service and processing policies at the transfer devices.
- The control logic (intelligence) is passed to a third party, the SDN controller or Network Operating System (NOS). The NOS is a software platform based on server technology that provides the tools and abstractions required to simplify the construction of forwarding devices using an abstract view of the network and logical centralization. Its purpose is therefore similar to that of a traditional operating system.
- The network is programmable: The network can be programmed using software applications that connect with the devices on the underlying data plane and run on the NOS. This is a core feature of SDN, as well as its primary value-added feature.

Functions of the SDN

SDN separates the data plane from the control plane, as previously indicated. In other words, the intelligence of the network is transferred to a controller, all computations are performed there, and many applications and features can be added as needed.

In Tok & Demirci (2021), the researchers discussed the basic modules of an SDN controller. They concluded that the modules of link discovery, topology management, storage, policy, flow table management and control data are the core modules of the SDN controller. Topology management is one of the essential and critical functions of the architecture. It is provided by two modules, the topology manager and the routing and link discovery manager, that also provide the routing service.

Authentication and Access Control

Access control is a security technique that can be used to determine which users or programs are allowed to see or use resources in an IT environment (Zhang & Hu, 2021). Physical and logical access control are the two main types of access control. Physical access control restricts access to campuses, buildings, rooms and computer equipment.

Connections to computer networks, system files, and data are all restricted by logical access control.

Access control systems handle approved identity, authentication, access approval, and entity responsibility using login credentials, such as passwords, PINs, biometrics, and electronic or physical keys. An access control model includes (Shan, Zhou & Hong, 2021):

- An access control policy (or rule) that specifies what access to data is allowed;

- An administration policy that specifies how the access controls policy can be updated. An access control mechanism is a software or hardware solution for enforcing an access control policy.

IoT now confronts numerous issues in authenticating the devices and sensors that connect to the network. Because the sensors' hardware IDs can be faked (Alizadeh, Tadayon & Jolfaei, 2021). As a result, there are few options for authenticating device sensors or home automation devices. Although standard security protocols, such as X10, ZWave, and ZigBee, have been embraced by the industry and can provide encryption methods, it is still a work in progress to develop acceptable mechanisms for authenticating devices.

Many security techniques based on private key cryptographic primitives have been developed due to quick computation and energy efficiency, as mentioned in Nait-Hamoud, Kenaza & Challal (2021). It is inefficient to keep keys for heterogeneous devices in the IoT because of the scalability issue and the memory requirement. IoT does not now solve all authentication requirements, such as mutual authentication, replay attack resistance, DOS (Denial of Service), MITM (Man in the Middle), and lightweight solutions.

OTP (One-Time Password)

A one-time password (OTP) is a random string of numbers or letters that is used to authenticate a user for a single transaction or login session (Lee, Kang & Cho, 2017). A password created by the user is less secure than an OTP, especially if it is weak and/or used across numerous accounts. OTPs can be used in place of or in addition to authentication login information to offer another degree of protection (Babkin & Epishkina, 2018).

Example:

OTP security tokens are microprocessor-based smart cards or pocket-size key fobs that generate a numeric or alphanumeric code to authenticate access to a system or transaction. This secret code varies every 30 or 60 seconds, depending on how the token is programmed.

Mobile device apps, such as Google Authenticator, rely on the token device and PIN to generate the one-time password for two-step verification. Hardware, software, or on-demand security tokens can all be used to implement OTP security tokens. Unlike regular passwords, which remain static or expire every 30 to 60 days, the one-time password is only used for one transaction or login session.

2FA (Two-Factor Authentication)

Two-factor authentication (2FA) is a security method that needs two different forms of identity to gain access to anything (Kemshall, 2011). Two-factor authentication can be used to strengthen the security of an online account, a smartphone, or even a door. 2FA needs two types of input from the user: a password or personal identification number (PIN), a fingerprint, or a code texted to the user's smartphone before anything being secured can be accessed.

Two-factor authentication is a security feature that prevents unwanted users from getting access to an account using only a stolen password. Users may be at greater risk of compromised passwords than they realize, particularly if they use the same password on more than one website. Downloading software and clicking on links in emails can also expose an individual to password theft (Sadri & Asaar, 2021).

Relevant Works and Limitations

Table 1. State of the art.

Research Work	Used Technology					Summary contributions
	IoT	Cloud	SDN	OTP	2FA	
Hammi *et al.*, 2020	✓	*	*	✓	*	Aim to advance the literature on IoT authentication by proposing three authentication schemes that satisfy the needs of IoT systems in terms of security and performance.
Baseri, Hafid & Cherkaoui, 2018	✓	✓	*	*	*	A method based on the Attribute-Based Encryption (ABE) technique, for designing secure and efficient data access control for mobile cloud
Hammi, Bellot & Serhrouchni, 2018	✓	*	*	✓	*	An approach that uses the asynchronous mode, which is based on the challenge/response method defined by RFC 1994
Botta *et al.*, 2014	✓	✓	*	*	*	Physical and virtual items in this network have identities, physical attributes, virtual personalities, and intelligent interfaces, and they are all perfectly interwoven
Abdellatif *et al.*, 2022	✓	*	*	*	*	Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data
Hussain & Chun, 2022	*	✓	*	*	*	Services offered by Cloud Computing are SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service), and NaaS (Network as a Service)
Mutlag *et al.*, 2019	✓	✓	*	*	*	Cloud4IoT is a platform that enables plug-and-play integration of additional sensor objects, as well as dynamic scalability, by automating the deployment, orchestration, and dynamic configuration of IoT support software components and data processing and analysis applications.

Research Work	Used Technology					Summary contributions
	IoT	Cloud	SDN	OTP	2FA	
Stergiou *et al.*, 2018	✓	✓	*	*	*	Cloud Things, IaaS, PaaS, and SaaS to help developers, deploy, and manage Internet of Things applications more quickly
Munther *et al.*, 2021	*	*	✓	*	*	Scalable and secure SDN-based Ethernet architecture by suppressing broadcast traffic
El Kamel, Eltaief & Youssef, 2022	*	*	✓	*	*	Attack mitigation in SDN using Deep Neural Network-based rate limiting
Lee, Kang & Cho, 2017	✓	*	*	✓	*	Design and implementation for Data Protection of Energy IoT utilizing OTP in a Wireless Mesh Network
Babkin & Epishkina, 2018	*	*	*	✓	*	One-Time Passwords: Resistance to Masquerade Attack
Kemshall, 2011	*	*	*	*	✓	Two-factor authentication makes sense in mobile
Sadri & Asaar, 2021	✓	*	*	*	✓	Two-factor authentication protocol for IoT-based applications

In the literature, only a few works use SDN, 2FA, and OTP in the authentication for IoT. Table 1 presents various studies, including a summary of their contributions. Some work is done on securing authentication of IoTs in the cloud with authentication principles like OTP or 2FA, which does not lead to secure and satisfactory results. In this section, we present three works. The first work (Hammi *et al.*, 2020) aims to advance the literature on IoT authentication by proposing three authentication schemes that satisfy the needs of IoT systems in terms of security and performance. One-Time Password (OTP) is a type of authentication that could be beneficial in Internet of Things and smart city applications. To ensure the security of such a protocol, this research effort extends the OTP principle and provides a novel way to produce OTP based on Elliptic Curve Cryptography and Isogeny.

The second work (Baseri, Hafid & Cherkaoui, 2018) is a method based on the Attribute-Based Encryption (ABE) technique, for designing secure and efficient data access control for mobile cloud. These methods allow data owners (enterprises or individuals) to ensure data security and provide mobile users with fine-grained access to data using defined policies and constraints.

The third work (Hammi, Bellot & Serhrouchni, 2018) is an approach that uses the asynchronous mode, which is based on the challenge/response method defined by RFC 1994 (Simpson, 1996). We chose this mode because it does not require any prior agreement between the communicating objects, in contrast to the synchronous mode, which requires an agreement between objects on some parameters, such as “time”, (for example, the Time-based One-Time Password (TOTP) algorithm (M'Raihi *et al.*, 2011)) or a "counter" (for example, the HMAC-based One-Time Password (HOTP) (M'Raihi *et al.*, 2005)).

There is no SDN-based research work for securing IoT authentication to date, as Table 1 shows. For this reason, our idea is to set up a framework for securing IoT authentication dynamically based on SDN and the two authentication principles OTP and 2FA.

Proposed Solution

The proposed solution concentrates on controlling the access to the devices or objects that are connecting in an IoT network based on SDN. In this section, we first highlight the objectives of the proposed framework, its overall architecture, its four main layers, and its overall operation. Finally, we explain the IoT authentication scenario as well as the role of each component and how it will react in case of an attack or otherwise.

Objectives of the Framework

The objectives of our framework are grouped into three main points as follows:

- Intrusion detection and identity usurpation in a virtual environment using IDS (Intrusion Detection System) Checker and IDS Analyser to collect malicious data, and generation of new signatures from malicious data.
- Internet of Things (IoT) access control and authentication devices, which can recognize and connect many real and virtual sensors dynamically into a unified secure system.
- Dynamic deployment of updates between clusters in an IoT cloud, using SDN master and slave.

Proposed Model of IoT Architecture Based on SDN

As shown in Figure 1, we define an IoT architecture based on SDN with front-end and back-end. The front-end is connected to both an external network as well as the internal network. It is represented in Figure 1 by the application layer.

IoT users are able to communicate with the Cloud via the front-end. The back-end consists of computer hardware and software that are designed for the delivery of services. It allows treatment of the user's query and executes it allowing access to sensors and actuators. It is represented in Figure 1 by the Data Processing Layer, Networking Layer and Physical Layer (sensors and actuators layer).

Application Layer: It is the application layer that defines all applications in which IoT has been deployed. It is the interface between the end IoT devices and the network. Examples of IoT Applications are smart homes, smart health, and smart cities.

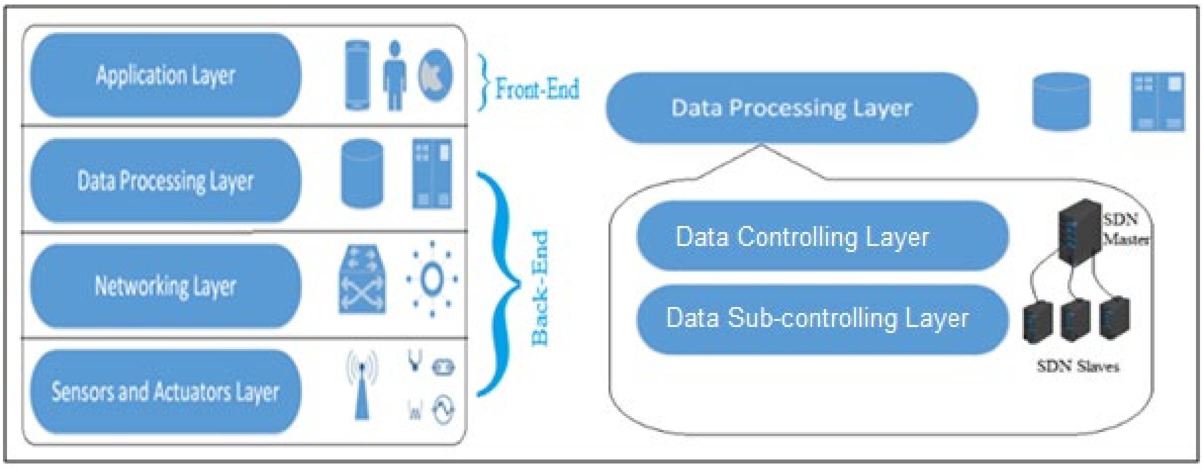

Figure 1. IoT architecture based on SDN with a front-end and back-end

The Application Layer has the authority to provide services to the applications. The services may be different for each application based on the information collected by the sensors. It is implemented at the device level by a specific application. The browser, for example, applies the application layer to a machine. It is the browser that executes application-layer protocols like HTTP, HTTPS, SMTP, and FTP. The application layer has numerous issues, the most important of which is security.

Data Processing Layer: In a three-layer system, data is transmitted directly to the networking layer. The likelihood of experiencing damage arises as a result of delivering data directly. In a four-layer architecture, data from a perception layer is transferred to this layer. The Data Processing Layer has two responsibilities: it verifies that data is forwarded by legitimate users and it protects the data from being tampered with.

Authentication is the most commonly used method to verify the users and the data. It is applied by using pre-shared keys and passwords for the concerned user. The second responsibility of the layer is to send information to the network layer. The medium through which data is transferred from the Data Processing Layer to the network layer can be both wireless and wire-based.

Data Controlling Layer: The SDN Controller serves as a bridge between the application and back-end layers. The northbound interface is the connection between the controller and applications, while the East/West interface is the connection between the controller and the data sub-controlling layer. This layer processes the instructions and requirements sent by the application layer (via northbound interface) and passes them to the networking components (via southbound interface). It also communicates back necessary information extracted from the networking devices to the application to function optimally.

Data sub-controlling Layer: also called the “control plane”, is mainly composed of one or more SDN controllers. Its role is to control and manage the infrastructure equipment through an interface called ‘southbound API’.

Network Layer: A transmission layer is another name for this layer. It functions as a bridge, carrying and transmitting data collected from physical things via sensors. The transmission medium can be wireless or wired. It also allows network devices and networks to communicate with one another. As a result, it is particularly vulnerable to attacks. It has important security issues regarding the integrity and authentication of data that is being transmitted to the network.

Perception layer/Sensor layer: The sensor layer is in charge of recognizing objects and collecting data from them. RFID, sensors and 2-D barcodes are just a few of the sorts of sensors that can be used to gather data from items. The sensors are selected as per the requirement of applications. The data that is collected by these sensors can be about location, changes in the air, environment etc. They are, however, mostly of interest to attackers who seek to use them to replace the sensor with their own.

Building a Solution Framework

After the presentation of the IoT architecture-based SDN, we now proceed to the establishment or distribution of the components of our framework according to our strategy for authentication security. Then, we shed light on the general architecture of our framework, shown in Figure 2.

In traditional networks, authentication security is mainly provided by firewalls to protect against attacks. With the advent of connected objects, these techniques are not enough to protect against attacks that are increasingly sophisticated and cover even larger perimeters with mobility. This is why we propose a new approach based on an intelligent and dynamic concept, using the functionalities of an SDN controller, the services of an IDS and an authentication server based on the 2FA mechanism, which we will call a Cluster of registration. This concept is a decentralized middleware to implement management and security policies for network devices via an SDN controller, IDS and a Dynamic Authentication Server (DAS). All network devices are located in a domain called a cluster. A cluster of the newest registrations is a main cluster where each new object must pass through to register in the IoT network in a direct or indirect way; i.e., if a new object tries to integrate with an already active cluster, it will be automatically redirected to the registration cluster, which makes our approach dynamic.

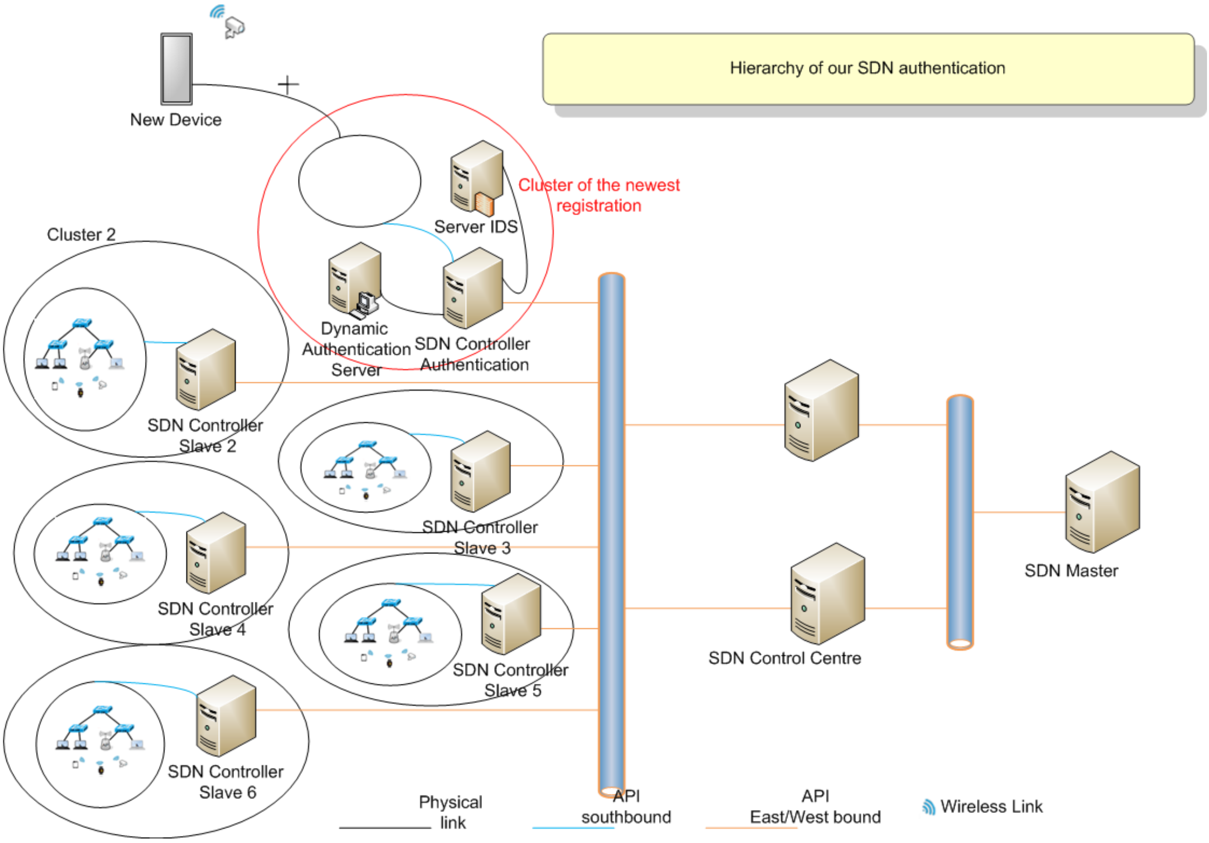

Figure2.General architecture of framework

Before registering or deregistering a node in a cluster, one must consider these registration rules:

- We need to enable the object on the registering SDN so that the registering SDN communicates its authentication table to all Slave SDNs up to the master SDN.
- A node cannot be registered in a cluster if it is already a member of another cluster. In this situation, the node must first be unregistered from its current cluster.
- Node registration is done directly and dynamically through the local management interface of the appliance we want to join to the cluster. The appliance we are registering must be able to communicate with the registering SDN.
- Deregistration of a node must occur on the primary master.
- A node cannot be deregistered if it is not registered on the entire SDN chain.

SDN Controller Authentication: consists of a physical machine running OpenFlow switches, an OpenDayLight controller, control elements of the distributed network. In order to ensure high programmability of control plane and a global view of the newest things in the network, it is decoupled from the data routing devices and centralized. This means that decision making is concentrated in a single (or redundant) location, the controller. Data routing is based on flow rules defined by controller instructions. The controller's decisions can be made based on a much wider range of criteria as well as on pre-programmed rules.

Dynamic Authentication Server (DAS): is an SDN Controller authentication client that implements the 2FA (Two Factor Authentication) algorithm based on OTP (One time Password). The DAS has been adapted to indicate whether the object is accepted or refused for authentication.

Server IDS: used to detect attacks against the registration cluster at an early stage and on the IoT network generally. It is the unit that reacts first in our approach whose purpose is to monitor and analyze all network activities, to detect unusual traffic and to notify the SDN Controller Authentication in such a case. This allows the latter to react to network access attempts by intruders and thus prevent an attack.

Other clusters: is a cluster of authenticated objects in the IoT network containing objects, nodes, network devices and mainly the OpenFlow Switch, communicating with the SDN.

SDN Controller Slaves: used to set the flow tables of the data plane switch (based on the OpenFlow protocol); such a feature set enables centralized and intelligent control and inspection of data packets that may be transmitted or received by any SDN switch connected to the network.

SDN Control Centre: complement of the SDN architecture, which aims at communicating the OpenFlow tables within the network and also the load balancing between SDNs, as well as a backup in case one of the slaves fails.

SDN Master: the major element of the architecture that has all the controls and authentication data, synchronizes with the Slave SDNs, and distributes the load in the SDN Controller centre.

Analyzing the Functioning of the Framework

We discuss how our framework works based on Figure 3.

Every time a new object wants to access an IoT network, it confronts our authentication concept based on the SDN and the authentication server adopting 2FA, as well as the IDS server.

The object sends a request for affiliation to the network; this request is redirected directly and dynamically to the registration cluster;the SDN Controller Authentication receives it and processes it, while ensuring the validity of the request. SDN Controller Authentication consults the data in its temporary log file to see if this object refers to a previous affiliation request: in the legitimate case, nothing is reported, so the event must be passed to the IDS server (based on Radius) so that it undergoes a thorough analysis todecide if the event is an attack or a straightforward authentication request. If it is an attack, the SDN Controller Authentication rejects the request; otherwise, the IDS server sends an authentication request to the DAS. The DAS comes to its process (that we will see later in detail) of authentication based on 2FA: if the two parameters are validated, a new affiliation is generated with the SDN. The SDN Controller Authentication communicates its authentication table to the SDN Control Centre, then to the SDN Master, so that the latter propagates this affiliation on all the SlaveSDNs. This is called anauthentication network update. The SDN Controller Authentication updates its log file for a new affiliation attempt temporarily (logging principle).

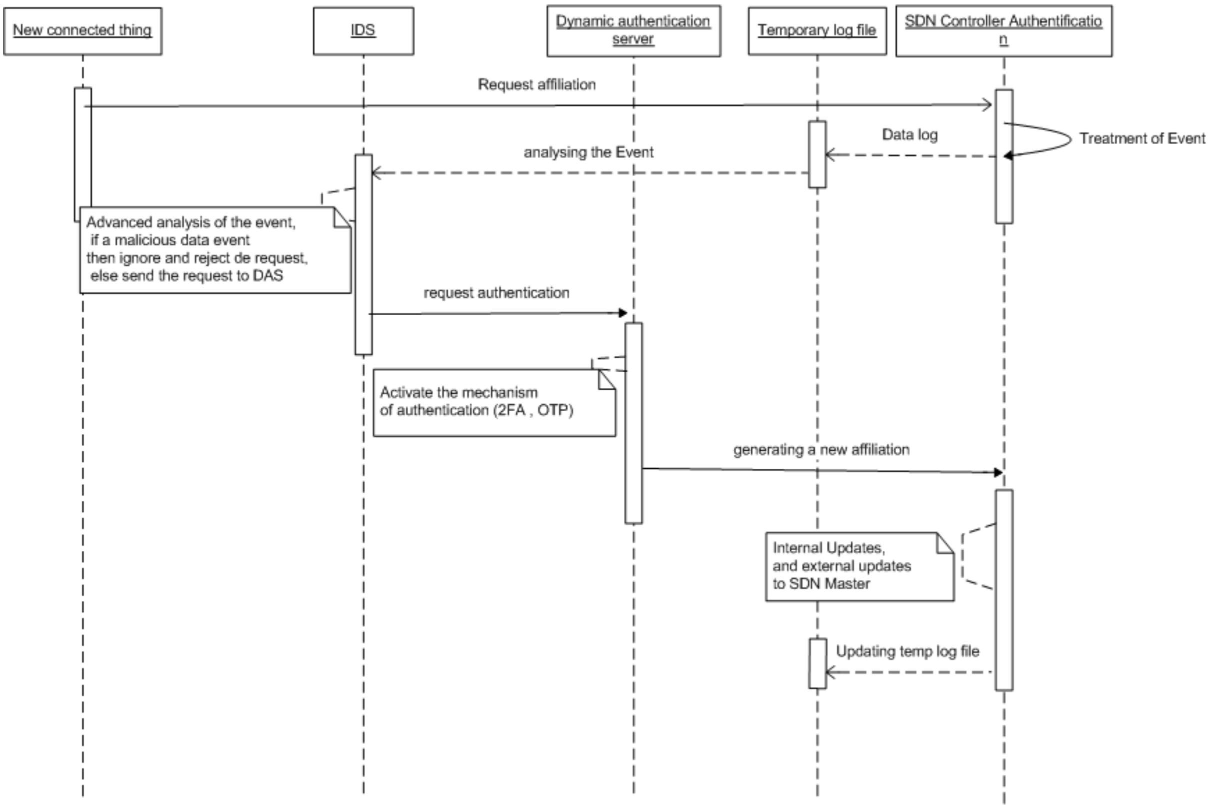
New connected thing
IDS
Dynamic authentication server
Temporary log file
SDN Controller Authentification
Request affiliation
Data log
Treatment of Event
analysing the Event
Advanced analysis of the event,
if a malicious data event
then ignore and reject de request,
else send the request to DAS
request authentication
Activate the mechanism
of authentication (2FA , OTP)
generating a new affiliation
Internal Updates,
and external updates
to SDN Master
Updating temp log file

Figure 3. Framework's general sequence diagram

IDS Process

In this section, we detail the exchange part of the SDN Controller Authentication and the IDS server. From the previous global sequence diagram, we can see that, when a new object arrives, the SDN receives its affiliation request. This request is obviously to be analyzed. So, the SDN Controller Authentication gives it to the IDS server to be analyzed and to pass the IDS process.

In our approach, specifically in the detection and analysis of events from new objects, our IDS server is subdivided into two virtual machines (VM). To assure a new degree of trust in the VMs, we use Virtual Machine Monitoring (VMM) in our framework. The IDS-Checker components are then deployed at the node (physical server) level to monitor virtual machines.

IDS Checker: it is in the form of a program analyzing a predefined attack database; this VM is self-powered according to its successor IDS analyzer.

IDS Analyzer: a VM implementing an intrusion detection and analysis system, such as RADIUS.

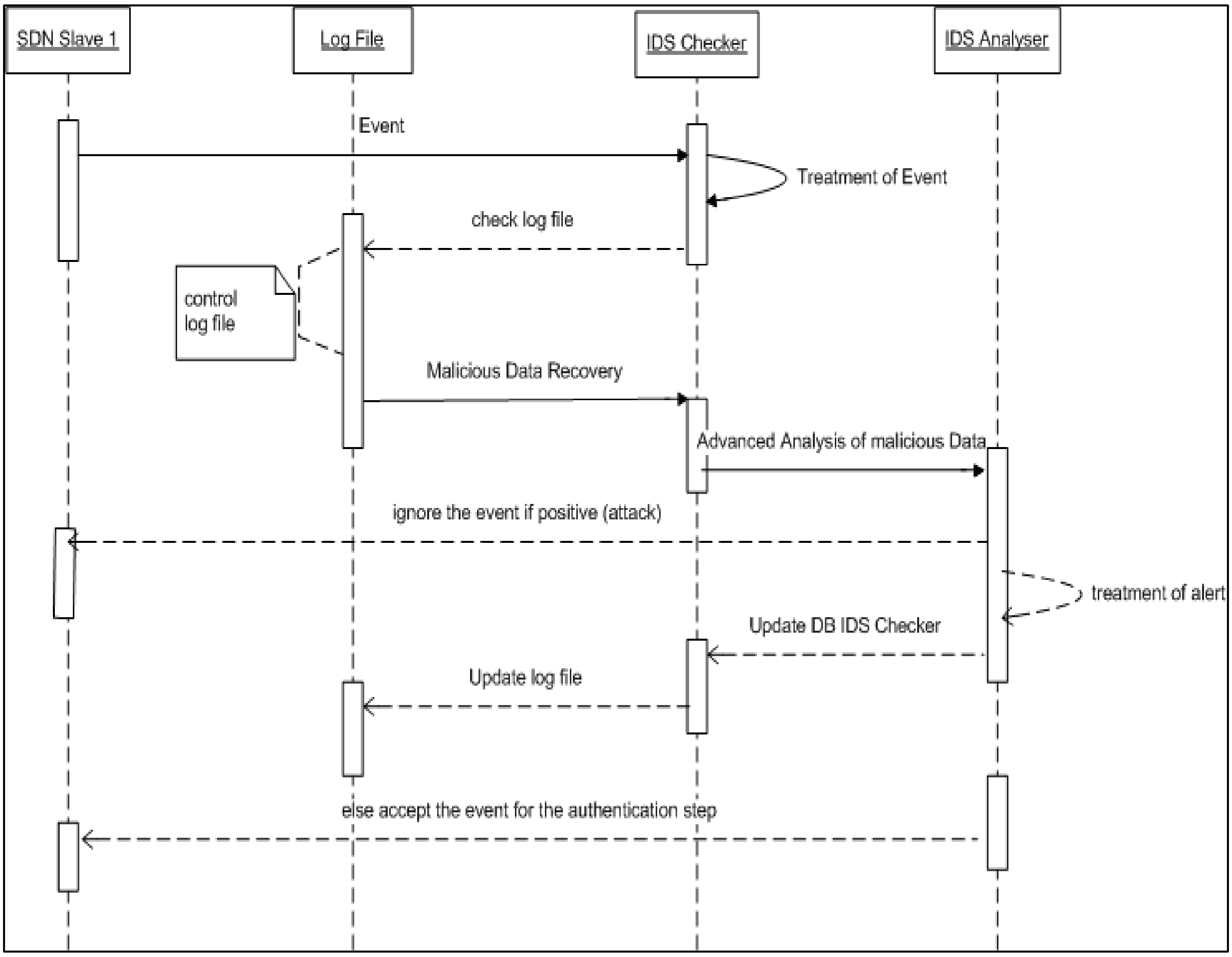

Figure 4. Sequence diagram between SDN Slave and IDS

As shown in Figure 5, the analysis process of the IDS server starts with SDN that sends the event to the IDS Checker, which processes this event and at the same time checks the IDS log file containing the access log of the objects. The log file returns a summary of the attempts of the objects that have launched attacks. Based on the summary and the processing of the IDS Checker, the IDS Analyzer receives the whole thing to decide if it is an attack or not. If it is an attack already predefined in the database, the IDS Analyzer ignores the event, sets an alert, and updates the IDS Checker and the log file. In the opposite case, that is, the event is a real affiliation request, the IDS Checker informs the SDN with an authentication acceptance.

Dynamic Authentication Server

When designing our approach, we first studied an initial version that ensures the authentication of a device during the association phase. The authentication mechanism used is based on the principle of 2FA, which takes two parameters into account to ensure authentication. The first parameter is the One Time Password (OTP), defined in RFC 2289 and RFC 4226, as well as the challenge/response principle described by RFC1994. Indeed, an OTP is a password that is valid only during a single authentication operation, which is why it represents a very resistant authentication mechanism against replay attacks and cryptanalysis attacks. One-time passwords can be used in synchronous or asynchronous mode.

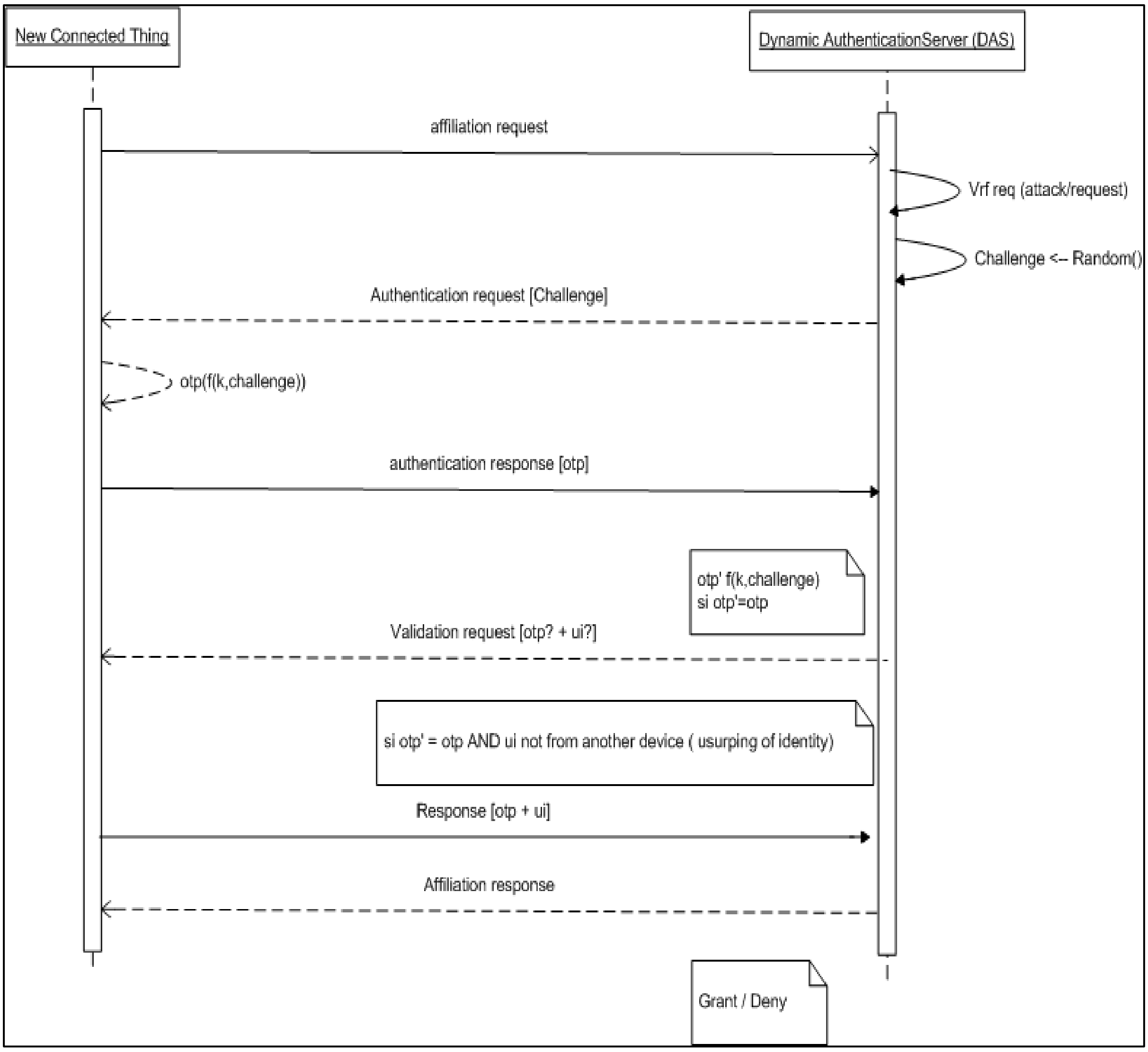
New Connected Thing
Dynamic AuthenticationServer (DAS)
affiliation request
Vrf req (attack/request)
Challenge <-- Random()
Authentication request [Challenge]
otp(f(k,challenge))
authentication response [otp]
otp' f(k,challenge)
si otp'=otp
Validation request [otp? + ui?]
si otp' = otp AND ui not from another device (usurping of identity)
Response [otp + ui]
Affiliation response
Grant / Deny

Figure 5. Sequence diagram between new thing and DAS

The synchronous mode is based on a shared secret (e.g., symmetric key) between two objects to prove the identity of one or both communicating entities and a pre-shared parameter, such as time or a counter, that changes in a synchronized way after each authentication operation. As for the asynchronous mode, it is based on a shared secret and a random number called a challenge sent by the authenticator. We used the asynchronous mode because, unlike the synchronous mode, it does not require any prior approval between the communicating entities (e.g., a counter). And because of the instability of most wireless networks, if a message is lost then this will cause a synchronization problem as the counter values become different. Moreover, many wireless technologies do not support absolute time.

To remedy the problem of identity usurpation not taken into account in asynchronous mode, our authentication protocol adds the exchange of two more messages, which are "authentication request" and "authentication response". Therefore, in order for a device to associate securely, it first sends an association request to the DAS. The latter responds with an authentication request containing a challenge. Then, via a cryptographic function described by RFC 4226, which we have adapted for the asynchronous mode, the device computes an OTP using a shared secret key and the received challenge. Then, it derives and stores a session key, k_u, that will be used to secure the session data exchanged in unicast mode. After that, it sends the OTP via an authentication response. Finally, upon receiving the message, the DAS calculates another OTP based on the same parameters and functions used by the device, and then compares the two OTPs. If they are identical, this proves the identity of the device. Thus, the authentication of the device is successful and its association is confirmed by a response association. After the authentication of the device, a symmetrical session key,identical to the one stored in the device is generated at the DAS level in order to ensure the integrity of messages in unicast mode.

Sharing the master key, used for object authentication and session key generation, is a challenge for security protocol designers. For a secure protocol, this key must be unique and personalized for each device. This personalization must not negatively influence the network performance and the proper functioning of the DAS. Therefore, we have created a key management mechanism called "Customization" of keys. The latter represents a secure, flexible and optimal method of distributing pre-shared keys that protects objects against internal spoofing attacks. Indeed, the principle of this mechanism is based on the fact of installing an initial key, k_i, at the DAS level and deriving from it a personalized key, k_d, for each device. The key k_d is calculated from k_i and the unique identifier (UI) of the device using a hash function (HMACSHA256). The latter represents a one-way function that prevents the input parameters (e.g., secret key) from being obtained from the result depending on the data provided; it generates a wide range of results. This customization offers many advantages:

- The DAS does not need to store the k_d key of each device belonging to its network, but rather to deduce it automatically thanks to its k_i key and the UI of the device requesting the association.
- When a new device with a k_d is added, the DAS does not need to be updated. This allows a great transparency and flexibility when adding new devices.
- Unlike some approaches that propose authentication based on a broadcast key, the fact that each device has its own key, which is linked to its identity, protects the system against internal identity theft.

Finally, once the association ends, a secure channel is created between the communicating entities. This channel ensures the integrity of the data by signing all messages with the messages using the key k_u. The signature represents the first n octets of the HMACSHA256 of the frame to be sent. Due to the limitation of the frame payload size in IoT networks, it is preferable that n does not exceed 16 bytes. This way, if a message is modified or tampered with, the system can automatically detect the problem.

Discussion

The IoT is an Internet-based computing technology, in which the necessary resources are provided on a rental basis to clients. Therefore, the existence of vulnerabilities in the IoT allows intruders to affect the confidentiality, availability and integrity of IoT network resources as well as services. Authentication intrusions, identity theft, and other malicious activity at the network level are major security issues in the IoT. To ensure a high level of access control in the IoT network, we propose a new framework based on the cooperation of SDN and 2FA and OTP mechanisms. This framework allowed us to achieve three objectives: intrusion detection (known and derived from known attacks) at the front-end and back-end of the IoT environment autonomously; dynamic deployment of updates between clusters in an IoT network, using SDN communication APIs; a dual parameter secure authentication, the first based on the OTP algorithm and the second based on a challenge calculation. We used the OpenFlow protocol to exchange updates between clusters to obtain new knowledge and detect new devices. Exceptional scalability is another strong point of this framework. When, for example, our object migrates from a Cluster1 to Cluster2, it is still possible to make exchanges in the network because our SDN Master can migrate any SDN slave table to another SDN slave. The strength also lies in our framework, which gives IDSs scalability and flexibility. Therefore, we have met almost all the challenges mentioned in our framework. Therefore, this framework has several advantages; for this reason, it can be considered as an effective solution for object authentication in an IoT network. Thus, it can be used to protect people and assets from the risks of intrusion and aggression.

Conclusions and Future Work

The IoT is enjoying undeniable success, which could be compromised by concerns about the risks associated with potential misuse of this model to conduct illegal activities. There is a major need to bring security, transparency and reliability into the IoT model for customer satisfaction. Therefore, one of the security issues is how to reduce the impact of any type of intrusion in this environment and mainly in the authentication process. Thus, in this paper, we propose a dynamic framework, which is based on the collaboration of IoT, SDN, IDS, 2FA, and OTP. As mentioned earlier, SDN controllers are used in our framework to examine devices, to transfer object data, and to update exchanges between different clusters in the IoT network; thus, SDN controllers could have the ability to examine objects and provide communication between hierarchical layers or clusters. Therefore, the Dynamic Authentication Server (DAS) is present to ensure secure authentication while relying on the 2FA mechanism, a two-parameter authentication where the first parameter is derived from the computation of an OTP and the second parameter is based on the computation of a challenge launched by the DAS. However, there are also the IDSs, which allow an analysis and detection of attacks at each affiliation of an object – that is, the analysis of an event and also the detection of an identity theft. Therefore, further development of mobile agent toolkits will facilitate their application in IDS systems. Finally, a dynamic deployment of a strong authentication mechanism and generation of updates between clusters in an IoT network, using SDN architecture, will also be of significant value. Further research can be undertaken to improve the presented work. Future directions are:

- Continuing to further develop the concepts and notions of this architecture and then proceed with an implementation to validate it.
- Taking into account minimizing the affiliation time between DAS and the object.
- The use of cooperation mechanisms between other authentication algorithms in order to reinforce the access control.
- Automatic improvement of the attack database at the IDS level.

References

Abdellatif, A. A., Mhaisen, N., Mohamed, A., Erbad, A., Guizani, M., Dawy, Z., &Nasreddine, W. (2022). Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. *Future Generation Computer Systems*, *128*, 406–419. https://doi.org/10.1016/j.future.2021.10.016

Alizadeh, M., Tadayon, M. H., & Jolfaei, A. (2021). Secure ticket-based authentication method for IoT applications. *Digital Communications and Networks*. [online]. https://doi.org/10.1016/j.dcan.2021.11.003

Babkin, S., & Epishkina, A. (2018). One-Time Passwords: Resistance to Masquerade Attack. *Procedia Computer Science*, *145*, 199–203. https://doi.org/10.1016/j.procs.2018.11.040

Baseri, Y., Hafid, A., & Cherkaoui, S. (2018). Privacy preserving fine-grained location-based access control for mobile cloud. *Computers & Security*, *73*, 249–265. https://doi.org/10.1016/j.cose.2017.10.014

Biggs, J. (2016). Hackers release source code for a powerful DDoS app called Mirai. *Tech Crunch*, October 11, 2016. Retrieved from https://techcrunch.com/2016/10/10/hackers-release-source-code-for-a-powerful-ddos-app-called-mirai/#:~:text=Hackers%20release%20source%20code%20for%20a%20powerful%20DDoS%20app%20called%20Mirai,-John%20Biggs%40johnbiggs&text=After%20doing%20heavy%20damage%20to,the%20source%20code%20on%20Github.

Botta, A., DeDonato, W., Persico, V., & Pescapé, A. (2014). On the integration of cloud computing and internet of things. Future internet of things and cloud (FiCloud), International Conference on, IEEE. https://doi.org/10.1109/FiCloud.2014.14

El Kamel, A., Eltaief, H., & Youssef, H. (2022). On-the-fly (D)DoS attack mitigation in SDN using Deep Neural Network-based rate limiting. *Computer* Communications, *182*, 153–169. https://doi.org/10.1016/j.comcom.2021.11.003

Hammi, M. T., Bellot, P., & Serhrouchni, A. (2018). BCTrust: A decentralized authentication blockchain-based mechanism. *IEEE Wireless Communications and Networking Conference (WCNC)*. https://doi.org/10.1109/WCNC.2018.8376948

Hammi, B., Fayad, A., Khatoun, R., Zeadally, S., & Begriche, Y. (2020). A Lightweight ECC-Based Authentication Scheme for Internet of Things (IoT). *IEEE Systems Journal*, *14*(3), 3440–3450. https://doi.org/10.1109/JSYST.2020.2970167

Hussain, A., & Chun, J. (2022). Cloud service scrutinization and selection framework (C3SF): A novel unified approach to cloud service selection with consensus. *Information* Sciences, *586*, 155–175. https://doi.org/10.1016/j.ins.2021.11.024

Junior, N. F., Silva, A. A. A., Guelfi, A. E., & Kofuji, S. T. (2021). Privacy-preserving cloud-connected IoT data using context-aware and end-to-end secure messages. *Procedia Computer* Science, *191*, 25–32. https://doi.org/10.1016/j.procs.2021.07.007

Kemshall, A. (2011). Why mobile two-factor authentication makes sense. *Network Security*, *2011*, 9–12. https://doi.org/10.1016/S1353-4858(11)70038-1

Kocher, P. C. (1996). Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) Advances in Cryptology — CRYPTO '96. CRYPTO 1996. Lecture Notes in Computer Science, *1109*, 104–113. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-68697-5_9

Koohang, A., Sargent, C. S., Nord, J. H., & Paliszkiewicz, J. (2022). Internet of Things (IoT): From awareness to continued use. *International Journal of Information Management*, *62*, 102442. https://doi.org/10.1016/j.ijinfomgt.2021.102442

Lagane, C. (2017). BrickerBot, un destructeur d'objets connectés qui agit… pour la bonne cause. Silicon (France), April 21, 2017. Retrieved from https://www.silicon.fr/brickerbot-destructeur-objets-connectes-bonne-cause-172891.html

Lee, S., Kang, B., & Cho, K. (2017). Design and Implementation for Data Protection of Energy IoT utilizing OTP in the Wireless Mesh Network. *Energy Procedia*, *141*, 540–544. https://doi.org/10.1016/j.egypro.2017.11.116

Mitake, Y., Tsutsui, Y., Alfarihi, S., Sholihah, M., & Shimomura, Y. (2021). A life cycle cost analysis method accelerating IoT implementation in SMEs. *Procedia CIRP*, *104*, 1424–1429. https://doi.org/10.1016/j.procir.2021.11.240

M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., & Ranen, O. (2005). HOTP: An HMAC-Based One-Time Password Algorithm. *IETF, RFC* 4226. https://www.ietf.org/rfc/rfc4226.txt

M'Raihi, D., Machani, S., Pei, M., & Rydell, J. (2011). TOTP: Time-Based One-Time Password Algorithm. *IETF, RFC* 6238. https://doi.org/10.17487/rfc6238

Munther, M. N., Hashim, F., Abdul Latiff, N. A., Alezabi, K. A., & Liew, J. T. (2021). Scalable and secure SDN based ethernet architecture by suppressing broadcast traffic. *Egyptian Informatics Journal*, *23*(1), 113–126. https://doi.org/10.1016/j.eij.2021.08.001

Mutlag, A. A., Abd Ghani, M. K., Arunkumar, N., Mohammed, M. A., & Mohd, O. (2019). Enabling technologies for fog computing in healthcare IoT systems. *Future Generation Computer Systems*, *90*, 62–78. https://doi.org/10.1016/j.future.2018.07.049

Nait-Hamoud, O., Kenaza, T., & Challal, Y. (2021). Certificateless Public Key Systems Aggregation: An enabling technique for 5G multi-domain security management and delegation. *Computer Networks*, *199*, 108443. https://doi.org/10.1016/j.comnet.2021.108443

Sadri, M. J., & Asaar, M. R. (2021). An anonymous two-factor authentication protocol for IoT-based applications. *Computer Networks*, *199*, 108460. https://doi.org/10.1016/j.comnet.2021.108460

Shan, L., Zhou, H., & Hong, D. (2021). Application of access control model for confidential data. *Procedia Computer Science*, *192*, 3865–3874. https://doi.org/10.1016/j.procs.2021.09.161

Simpson, W. A. (1996). PPP challenge handshake authentication protocol (CHAP). RFC 1994. https://www.rfc-editor.org/rfc/rfc1994.html

Stergiou, C., Psannis, K. E., Kim, B.-G., & Gupta, B. (2018). Secure integration of IoT and Cloud Computing. *Future Generation Computer Systems*, *78*, 964–975. https://doi.org/10.1016/j.future.2016.11.031

Sturm, R., Pollard, C., & Craig, J. (2017). *Application Performance Management (APM) in the Digital Enterprise*, Appendix C - The NIST Definition of Cloud Computing, 267–269. Morgan Kaufmann, Boston.

Tok, M. S., & Demirci, M. (2021). Security analysis of SDN controller-based DHCP services and attack mitigation with DHCPguard. *Computers & Security*, 109, 102394. https://doi.org/10.1016/j.cose.2021.102394

Zhang, R., & Hu, Z. (2021). Access control method of network security authentication information based on fuzzy reasoning algorithm. *Measurement*, *185*, 110103. https://doi.org/10.1016/j.measurement.2021.110103