SECURITY OPTIMIZATION OF IOT PLATFORMS BASED ON NAMED DATA NETWORKING

Dan-Andrei MARGIN[1], Denisa-Adina MOLDOVAN[1], Iustin-Alexandru IVANCIU[1], Jordi DOMINGO-PASCUAL[2], Virgil DOBROTA[1]
[1]*Communications Department, Technical University of Cluj-Napoca, Romania*
[2]*Computer Architectures Department, Universitat Politècnica de Catalunya, Barcelona, Spain*
Corresponding author: Virgil Dobrota (e-mail: Virgil.Dobrota@com.utcluj.ro)

Abstract: When it comes to developing a smart system involving sensors and actuators, there are two main problems to be addressed: what platform to be used as infrastructure and how to develop the security layer? In this paper, Orion Context Broker (OCB) is chosen for data management, while security layer is composed by Named Data Networking (NDN) and FIWARE modules (IdM KeyRock and Wilma PEP Proxy). The work was focused on data confidentiality and integrity because this is an important aspect of an IoT system. Store capabilities of OCB were extended from one value for an attribute to long term store archive. This approach is benefic for obtaining data necessary for statistics and predictions, where historical data must be used.

Keywords: *confidentiality, data archive, FIWARE, integrity, IoT devices, Named-Data Networking, security*

I. INTRODUCTION

Technology develops in a rapid manner in each domain. Specialists try to develop tools or technologies that are related to different kind of processes. There are now billions of interconnected devices, and all of them support different communication protocols and have different requirements or usage. Internet of Things (IoT) concept is growing continuously, and a massive amount of data needs to be processed by end-point applications. The most valuable resource of nowadays businesses are data. Having a large amount of data related to a certain subject opens various perspectives. Regarding this subject, big data analysis is a powerful tool because it gives you the possibility to generate statistical measurements, to predict phenomena or to overview a certain process.

According to [1], the number of connected devices in the network will reach 75 billion in 2025. The same research paper states that the amount of data that will be transmitted through the Internet will be around 25 ZB (zettabytes). Considering that nowadays sensors are used in almost every domain, including domestic usage, the price range of them is quite wide. In this situation, everybody can use them to develop a smart home system, by using the correct sensors and an integration platform [7]. There are many open-source platforms that can be used to implement IoT environments, which are able to manage enough sensors and let the user control the entire behavior of the components. Some open-source IoT platforms are described in: IoTvity [16], Kaa IoT [17], Zetta [18], ThingSpeak [19], FIWARE [13] etc. While ThingSpeak is mostly used for data analysis and visualization using Matlab, all the others are cloud-based architectures offering flexibility.

When we talk about IoT networks, there are two main topics that needs to be discussed. First of them is the infrastructure that will be used to implement the system. This means what kind of sensors will be installed, which protocol and what platform will be used to transmit data and, respectively, to manage the information. The second problem is the security of the system. In case of private and confidential data transmission, securing the system is a critical aspect. In case of a smart home system, temperature and humidity can be harmless for the owner, but the information related to the state of the doors and windows (closed or opened) or health information coming from medical devices must be protected. In the traditional way, TCP/IP stack can offer security by encrypting the channel. Within this paper, we considered an Information-Centric approach to offer protection to data itself. The solution for this problem can be solved using Named Data Networking (NDN), which is focused on naming the data instead of location, like in the case of IP addresses. The approach proposed herein uses an encryption mechanism based on Transport Layer Security (TLS) for TCP, and Datagram Transport Layer Security (DTLS) in case of UDP transmission.

The goal of this paper is to present an approach in developing a Smart IoT system which can be used by any type of sensors and protocols. The FIWARE platform was selected because it is recommended as an "alternative for proprietary solutions", when it comes to smart city development approaches [13] Our proposed approach makes use of the interoperability between the data broker (FIWARE) and NDN information-centric approach, both having similar data identification schemas. In this way, we consider that our solution offers a versatile solution to manage IoT data over a secure network.

This paper is organized as follows: Section 2 presents related work, while Section 3 discusses the general aspects regarding FIWARE and the principles of Named Data Networking. Section 4 describes the proposed architecture that includes Data Management and Data Security Layers.

Manuscript received February 15, 2021; revised April 26, 2021

Experimental results are presented in Section 5, and the paper ends with conclusions and future work.

II. RELATED WORK

In paper [2], the advantages of using FIWARE for remote patient monitoring are presented. This platform was chosen because it offers modularity, and it can address specific problems for each application. Considering that the application works with confidential data, an extra security layer was needed. The proposed solution is based on KeyRock IdM and Wilma PEP proxy server.

For monitorization of all the devices, Wirecloud web interface can be used, allowing OAuth2 authentication. The study realized in [3] presents a security approach for FIWARE-based applications composed by three components: IdM KeyRock, Wilma PEP proxy and AuthZForce. In this case, the authentication used is Single-Sign-On (SSO). AuthZForce module defined resource access permissions, with the same access token as IdM KeyRock and Wilma PEP proxy. The solution proposes a dual authentication mechanism for both users and devices, thus increasing the control to the data and offering consistency.

Whenever a design of an application based on open-source platform is realized, it is important to be aware of known vulnerabilities. According to [4], one of the issues in FIWARE framework is that not all the methods allowed at Orion Broker Level can be authenticated by the proxy server. There are two methods, Update Context and Register Context that are not able to use authentication. In this case, these requests cannot be verified by the Wilma PEP proxy. Another vulnerability found by [4] was related to the IoT Agent. This module does not encrypt user`s data, which can be easily intercepted. This issue is critical when transmitting commands to external actuators using the broker, because data tampering might occur.

Considering that IoT devices can transmit data frequently and they must work for a long time, energetic efficiency is a critical aspect. Paper [5] presents an overview of these problems and proposes a solution based on NDN. Usually, IoT packets have a small payload, which fits well with the Type-Length-Value format offered by NDN. As this approach offers scalability, the IoT packets will be as small as possible. Another problem is related to the resources of IoT devices. Considering that sensors update the information several times, they need to also have storage capabilities. Paper [5] presents a solution to the optimization of storage management and RAM usage by introducing the "caching" concept of an NDN node. This means that a consumer can retrieve the requested data packet not only from its original producer but also from neighbor nodes, which have cached that information previously.

III. GENERAL ASPECTS REGARDING FIWARE AND NAMED DATA NETWORKING

FIWARE framework was developed as an open-source platform that can be used in different architectures for data access and management of context-based information. It can be interconnected with various open Application Programming Interfaces (API) and it uses Next Generation Service Interface (NGSIv2) [6] as communication standard. The framework is mainly composed by Generic Enablers (GEs), which relate to the main component, the Orion Context Broker. To interconnect the broker to a network of sensors, IoT Agents are used on the southbound interface. One of the main roles of an IoT Agent is to convert from a device specific communication protocol, such as HTTP [8], MQTT [20], CoAP/ UDP [21] into one known by the broker. The security aspect is ensured by using KeyRock Identity Manager, based on OpenStack Keystone mechanism. FIWARE has become a standard in IoT development, being adopted by many cities for implementing smart city projects.

The development of the Information Centric Networks (ICNs) is an answer to the increasing demand for high efficiency and scalability distribution of content present in the Internet nowadays. This is a different approach than the legacy end-to-end communication between hosts. ICN is focused on the information objects, their properties and receiving interest to retrieve the objects in an efficient and reliable manner.

Two important parameters included herein are the Identification and the Security. The first one ensures that the naming scheme provides a unique name to all the content which is different. The second one is the most important parameter, and it is used to verify the requested content. Data objects need to be certified through self-certification. This procedure is done for the user to be able to authenticate the content, so it would not be corrupted or tampered, leading to different kind of attacks.

Named Data Networking uses a context-focused approach by naming the bits themselves, not the location of the bits. The centerpiece of this architecture includes data chunks and data names to communicate at Network Layer [12]. NDN changes the well-known significance of the network semantics (i.e., delivery of packets to a certain destination address), to data retrieval based on the name. The naming part of a packet can be used to address any kind of information: a data chunk in a song, a measurement unit of the results obtained from a sensor, a command in a smart house etc. The architecture added two new layers, Security and Strategy to its protocol stack. The first one provides security to every piece of content, unlike Internet, where the entire channel of communication must be secured end-to-end [10]. The Strategy Layer is used for the forwarding plane of NDN, making routing decisions for each incoming content request. Named Data Networking does not have a separate Transport Layer, all functions as in today's Internet being embedded into the forwarding plane Another important feature of NDN is caching [11], which allows simpler data retrieval process from the nearest node that has the information requested.

IV. PROPOSED SECURED IOT ARCHITECTURE

Our approach is based on FIWARE platform, including Generic Enablers (GEs) that are interconnected. The architecture has two distinct layers, Data Management and Data Security. The Orion Context Broker, acting as the central module of the system, runs on top of a non-SQL database MongoDB. As the IoT system can include many sensors, multiple brokers are involved, each of them dealing with different type of data. A simplified architecture diagram of the system, including one primary broker, one worker, IoT Agents and data tracking module is presented in Figure 1.

In this way, the workload is balanced between each of them and more important, the primary broker can collect data from all its workers, based on the interconnection called Context Broker Federation [14].

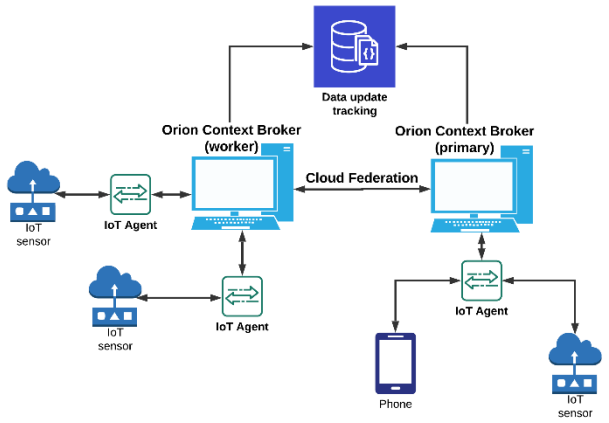

Figure 1. Data Management Layer architecture

This is based on subscriptions defined at worker level, using certain conditions. An example of a subscription definition is presented in Figure 2.

```
{"subject":
    {
    "entities":
[{"id":"Room1",
"type":"Room"}],
    "condition":{
  "expression":
{"q":"temperature"+">=25"}   }
    },
"notification":
{"http":{"url":"http://"+<IP>+":1026/v2/op/
notify"},
    "attrs":["temperature"]}}
```

Figure 2. Subscription object

When creating the subscription object, the only mandatory fields are "subject" and "notification". In the "url" sub-element, the IP address of the primary Broker must replace holder IP. Note that according to FIWARE platform, the Orion Context Broker will keep only the latest value for each attribute of an entity. In this way, when the sensors update the information, the previous value is overwritten. There are situations when users want to keep track of all data changes, so we need to save these values. The solution proposed here is to use the capabilities of the broker to send notifications. The notification including each updated value will be sent to another entity that will store the value in the same format as the broker. This entity can be queried using HTTP GET methods [9].

The functionality of each module was encapsulated into a Docker container, offering flexibility and portability.

Let us discuss now about the Data Security Layer that offers authentication and confidentiality for the user and its data that is transmitted through the network. The authentication system can be supported using FIWARE specific modules, like *Identity Manager (IdM) KeyRock* and *Wilma PEP Proxy Server*. These two modules are needed to generate an access token based on credentials (*KeyRock*) and to validate the requests coming from the user to the broker (*Wilma*). However, there is one more problem that needs to be addressed related to external devices authentication. All sensors and actuators entities that are connected to the broker can represent a potential attacker for the system. In such way, they have to authenticate as well, using the same mechanism as the users, based on tokens. In the end, every actor that participates in the process will use its own access token to send or request information. Figure 3 presents the authentication mechanism implemented by the proxy server to validate requests coming to the broker.

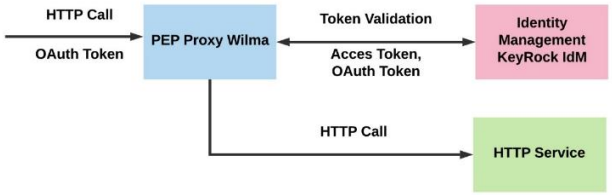

Figure 3. Wilma token validation

The other part of the security layer is represented by data confidentiality for the information centric approach proposed herein. Named Data Networking focuses on securing the application at bit level, compared to the traditional channel encryption mechanism, where IPsec and Virtual Private Networks (VPN) are used to secure the communication channel. In this paper the confidentiality is ensured using a Consumer–Producer architecture. Using this approach, the meaning of the payload carried by the *Data Packet* can be interpreted only by the two end entities of the communication. Note that the naming scheme is not implemented by the NDN standards and it is task of each developer to design it. Thus, each name corresponding to an *Interest* or *Data Packet* has a meaning only within the application it belongs to. When it comes to data tampering, the NDN standard states that the packet signature is mandatory to be used by any producer application. In this way, the consumer can trust its data provider and any modification at data packet level are noticed and the packet is rejected.

In our architecture we propose two independent entities, having the roles of a Consumer and a Producer. The communication between them is tracked using a database, where each entity updates its status. In this way, the communication channel is not overloaded by sending unsolicited *Interest* and *Data Packets*. The Consumer application is located at the user side, being the one that issued Interest packets based on the input parameters of the user. The workflow diagram of the Consumer entity is present in Figure 4.

The application runs as a service which requests the username and the password as input parameters. Based on them, it will retrieve the access token from *IdM KeyRock*. In a successful scenario, presuming the token is valid, when the correct status is present within the database (`READY_for_NDN_req`), the *Interest packet* is sent, containing the name corresponding to the information requested by the user. The other end of the communication,

the Producer, is represented also as a standalone application. Its main role is to fetch requested data from the Orion Context Broker and publish it in the network, such that it can be obtained by the Consumer, based on its name.

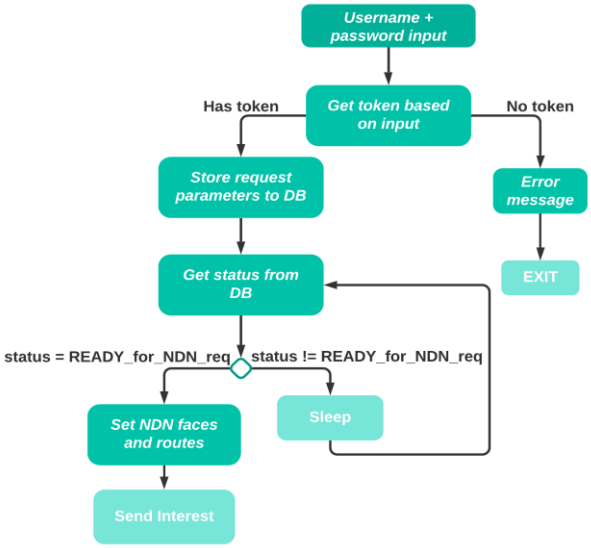

Figure 4. Consumer entity workflow

The workflow diagram of the Producer is presented in Figure 5.

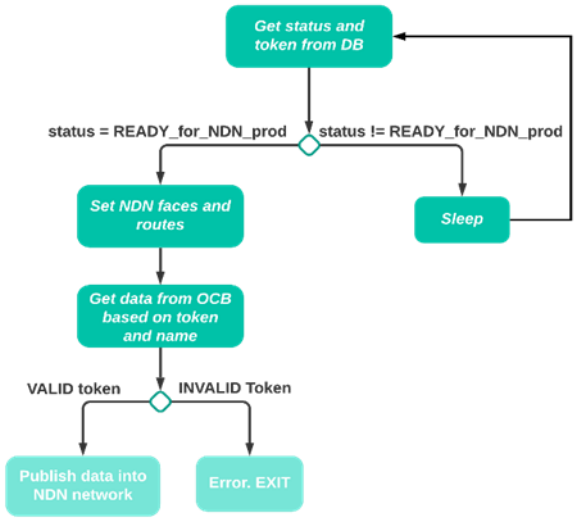

Figure 5. Producer entity workflow

In the same way as the Consumer, the Producer application verifies the communication status in the database (`READY_for_NDN_prod`). If the status is correct, it tries to fetch requested data from the OCB using the token. Note that only a valid token allows data to be published into the network.

Another important aspect regarding the Data Security Layer is related to the interconnection between Orion Context Broker and NDN Producer application (see Figure 6). Considering that OCB does not support HTTPS requests, we tried to minimize the HTTP requests that are issued over the Internet. In this way, the Producer application was integrated into the OCB module as an interface with the NDN network. The external communication was realized using only the NDN network, while the GET methods used to retrieve data (which was fetched in clear text from the broker) were performed locally by the Producer. We improved the security layer of the platform by adding an extra encryption layer besides the authentication one, which can be managed by using FIWARE components. It was out of the scope of this paper to develop a new naming scheme, as we relied on the default one provided by the Network Forwarding Daemon (NFD) server in NDN.

Finally, we obtained a secured IoT architecture, as in Figure 6. It includes the following: (1) FIWARE components (Orion Context Broker, IoT Agents, IdM KeyRock, Wilma PEP Proxy Server); (2) NDN entities (Consumer and Producer applications); and (3) Data update tracking (accumulator used for historical data and a common database to follow the communication status of each entity involved in the process).

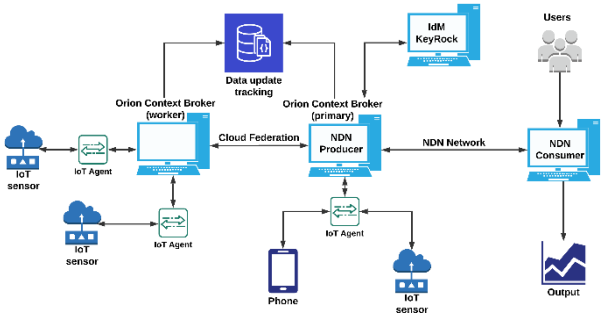

Figure 6. Secured IoT Architecture

We consider that the architecture has a fully functional secured data management system for IoT applications. The interconnection between FIWARE platform and NDN network was chosen because of the similarities that occurs at context identification and respectively naming convention. Both concepts use unique IDs at application level: FIWARE defines the application as “service”, while NDN defines the application as the communication between a Consumer and a Producer. The next section presents the results obtained after testing this proposed architecture for different scenarios.

V. EXPERIMENTAL RESULTS

The following experiments that will be presented are the following: (A) Legit retrieval of data from the broker; and (B) Data confidentiality using NDN network.

A. Legit data retrieval from the broker

According to FIWARE standard, the database behind Orion Context Broker can store only the latest value for an attribute. This is a drawback when the user needs to realize statistics based on a large set of data. Relying on the notification capabilities of the broker, we used a separate module that can collect all the values that are coming into the broker for a given entity. HTTP GET requests retrieved the data. For this experiment we obtained the average temperature around the earth in the last 150 years. We presumed a user called “test” was defined in the IdM Keyrock. Based on the username and password, token access can be retrieved and used to get the information

from the broker. The header of the GET method looks like in Figure 7.

```
headers = {"Fiware-Service": <service_name>,
           "Fiware-ServicePath": "/",
           "X-Auth-token'": <access_token>}
```

Figure 7. Subscription HTTP header

This method was implemented in Python programming language with HTTP requests. According to Figure 6, the request was sent locally by NDN Producer to the broker. Thus, we did not expose HTTP request, with no security layer on top of it. Figure 8 shows the result of the request, a `pandas.DataFrame` that is a multi-dimensional, heterogeneous tabular data furtherly used in data analytics computations.

```
In [27]: get_entity_info_history('192.168.1.45','Abidjan','City','thesis',"10","0")
```

	avgTemp	country	date
0	26.704	Côte D Ivoire	1849-02-01 00:00:00
1	28.101	Côte D Ivoire	1849-03-01 00:00:00
2	26.14	Côte D Ivoire	1849-04-01 00:00:00
3	25.427	Côte D Ivoire	1849-05-01 00:00:00
4	24.844	Côte D Ivoire	1849-06-01 00:00:00
5	24.058	Côte D Ivoire	1849-07-01 00:00:00
6	23.576	Côte D Ivoire	1849-08-01 00:00:00
7	23.662	Côte D Ivoire	1849-09-01 00:00:00
8	25.263	Côte D Ivoire	1849-10-01 00:00:00
9	26.332	Côte D Ivoire	1849-11-01 00:00:00

Figure 8. Historical data retrieval

Considering single attribute request, query time does not exceed 50ms for a maximum of 10,000 returned rows. This can be adjusted by changing the internal data organization of MongoDB database, which uses by default document type indexing.

B. Data confidentiality and integrity using NDN network

Data confidentiality is a critical aspect when it comes to IoT systems. As presented in paper [4], IoT Agents do not encrypt data, and the solution was to add an extra layer, TLS or DTLS. In this case, the overhead of the packets is also increased, lowering the performance. We had another approach, by using NDN networks. As it was presented before, TLV structure offers variable length, keeping the packet as small as possible. Regarding security, the NDN offers information encryption on application, meaning that only the Consumer which initiated the *Interest packet* can read the *Data Packet* published by the Producer.

To demonstrate this mechanism, we had a testbed scenario containing one intermediary node and an attacker that tries to scan the network to steal data. Based on present approach, the system is protected from two points of view. First, confidentiality is offered by the fact that the *Data Packet* is encrypted by the Producer. Note that by encrypting the *Data Packet*, the attacker is not able to see information about the source and destination of the packet, but it can read the payload. Secondly, both *Interest* and *Data Packet* contain a signature that is used to check the integrity of the packet. In this way, even if the attacker intercepts for example a command that the user wants to send to an IoT device over the network, he/she cannot tamper the data. Even if the data theft has been succeeded, the Consumer is not able to validate the signature, so the packet is discarded. The architecture is presented in Figure 9, where Consumer, Producer and Hop Node are part of the NDN network, while the attacker tries to scan the traffic between them.

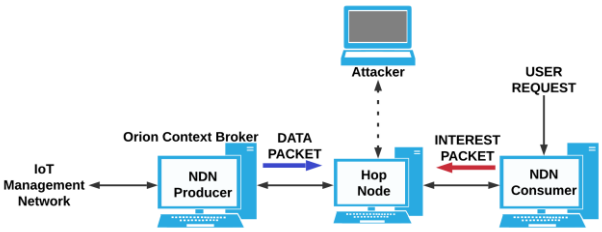

Figure 9. NDN secure communication

To encrypt the payload, we propose a pre-shared key approach [15]. It uses a set of keys that needs to be known by both Consumer and Producer. The list of keys can be retrieved from the OCB using SSL connection and X-Auth-Token. In this way, only authenticated devices can obtain the list of keys. When the Producer encrypts the data, it transmits in the *Data Packet* only the index of the key that needs to be used be the Consumer to retrieve the original data. Thus, the name contained in the *Interest packet* can be encrypted using the same mechanism. The entire algorithm is presented, as pseudocode script, in Figure 10.

```
Consumer Send Interest
getKeysFromBroker(accessToken)
InterestKey := getKeyFromList()
If (isValid(InterestKey) = true)
   Name := encryptName(InterestKey)
SendInteretPacket(Name + keyIndex)

Consumer Receive Data
KeyIndex := getKeyIndexFromPacket()
DataKey := getKeyFromIndex(KeyIndex)
If (isValid(DataKey) = true)
   Data := decryptData(DataKey)

Producer Receive Interest
getKeysFromBroker(accessToken)
KeyIndex := getKeyIndexFromPacket()
NameKey := getKeyFromIndex(KeyIndex)
If (isValid(NameKey)  = true))
   Name := decryptName(NameKey)

Producer Send Data
Data := fetchDataForName(Name)
DataKey := getKeyFromList()
If (isValid(DataKey) = true))
   Data := encryptData(DataKey)
SendDataPacket(Data + keyIndex)
```

Figure 10. Proposed algorithm

The advantage of this solution is that there is no need to send keys over the network. They must be present at each legitimate node within the NDN network. In this way, the key index that must be used by the other party is send together with the name, respectively data packets. This algorithm is independent of the encryption system chosen because the only thing that needs to be configured is the

mapping table between keys and indexes. Because the naming schema in NDN is application dependent, the index can be inserted into the Interest or Data packet in various ways, depending on the application choice. This algorithm does not affect neither the publish of Data Packets, nor the request of data using Interests, because if each node knows how to extract the key index from the data stream will be able to use to decode the payload.

The work with this algorithm is under development, so the performance evaluation is in a preliminary phase and it is out of the scope of this paper.

VI. CONCLUSIONS AND FUTURE WORK

This work addressed two major problems related to sensor-based systems and data management: selection of a platform and security issues. The FIWARE proved to be a good option as it provides its own communication mechanism already implemented, being flexible in integration with other technologies. Also, it fits very well with the security layer proposed herein: Named Data Networking. These two technologies successfully interworked because their information representation was similar, by using contextual elements. Regarding the security, the authentication methods provided by FIWARE modules, such as IdM KeyRock and Wilma, were combined with NDN data confidentiality at Application Layer. The proposed approach used a secure channel to load the list of keys from Orion Context Broker, whilst the NDN communication was based on key index exchanges.

As future development, we want to find a solution to integrate NDN capabilities at each node of the network, such that the entire communication within the system to be done by securing the information itself, instead of the channel. Another research direction will be focused on NDN forwarding strategies, to compare the performances of actual strategies regarding IoT systems and to choose the one that meets all the requirements, including small packet transmission, latency, and data confidentiality.

ACKNOWLEDGMENT

This work was carried out during an Erasmus mobility at Universitat Politècnica de Catalunya, Barcelona, Spain within academic year 2019-2020.

REFERENCES

[1] Statista Research Department, “Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025”, 2016

[2] M. Fazio, A. Celesti, F. Marquez, A. Glikson, M. Villari, “Exploiting the FIWARE Cloud Platform to Develop a Remote Patient Monitoring System”, Fifth Internaional Workshop on Management of Cloud and Smart City Systems, 2015, doi: 10.1109/ISCC.2015.7405526

[3] L. Barreto, A. Celesti, M. Villari, M. Fazio, A. Puliafito, “Identity Management in IoT Clouds: a FIWARE Case of Study”, Workshop on Security and Privacy in the Cloud, 2015, doi: 10.1109/CNS.2015.7346887

[4] C. Oliveira, R. Moreira, F. Silva, R. Miani, P. Rosa. “Improving Security on IoT Applications based on the FIWARE Platform”, IEEE International Conference on Advanced Information Networking and Applications, 2018, doi: 10.1109/AINA.2018.00104

[5] M. Hail, “IoT-NDN: An IoT Architecture via Named Data Networking (NDN)”, IEEE International Conference on Industry 4.0, Artificial Intelligence and Communications Technology (IAICT), 2019, doi: 10.1109/ICIAICT.2019.8784859

[6] J. Fonseca, F. Marquez, T. Jacobs, “FIWARE-NGSIv2 Specification”, 2018, [Online], Available: http://fiware.github.io/specifications/ngsiv2/stable

[7] S. Bergeon, L. Gurgen, B. Ortiz, “Activating Innovative IoT smart living environments for ageing well”, 2018, [Online], Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c3859613&appId=PPGMS

[8] “An Overview of HTTP”, MDN Contributors, 2021 [Online], Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

[9] “Using HTTP Methods for RESTful Services”, REST API Tutorials, 2021, [Online], Available: https://www.restapitutorial.com/lessons/httpmethods.html

[10] W. Shang, Z. Wang, A. Afanasyev, J. Burke, L. Zhang, “Breaking Out of the Cloud: Local Trust Management and Rendezvous in Named Data Networking of Things,” 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), Pittsburgh, PA, 2017, pp. 3-14

[11] D. Kim, Y. Kim, “Enhancing NDN Feasibility via Dedicated routing and caching,” Computer Networks, Elsevier BV, Vol: 126, 2017, pp. 218-228, doi: 10.1016/j.comnet.2017.07.011

[12] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, L. Zhang, “A Brief Introduction to Named Data Networking,” MILDCOM 2018 – 2018 IEEE Military Communications Conference, Los Angeles, CA, 2028, pp. 1-6, doi: 10.1109/MILOCOM.2018.8599682.

[13] S. Garcia, “FIWARE: A standard open platform for smart cities”, 2015 [Online], Available: https://www.fiware.org/2015/03/25/fiware-a-standard-open-platform-for-smart-cities/

[14] “Context Broker Federation” [Online], Available: https://fiware-orion.readthedocs.io/en/1.2.0/user/federation/index.html

[15] D. Harkins, “Secure Pre-Shared Key (PSK) Authentication for the Internet Key Exchange Protocol (IKE)”. 2012 [Online], Available: https://tools.ietf.org/html/rfc6617

[16] “IoTivity Architecture”, 2021 [Online], Available: https://iotivity.org/about/iotivity-architecture

[17] “Kaa IoT Cloud”, 2021 [Online], Available: https://www.kaaproject.org/kaa-iot-cloud

[18] “An API-First Internet of Things”, 2021 [Online], Available: https://www.zettajs.org/

[19] “ThingSpeak for IoT Projects” [Online], Available: https://www.thethingsnetwork.org/docs/applications/thingspeak

[20] R.A. Atmoko et al, “IoT real time data acquisition using MQTT protocol”, Journal of Physics: Conference Series, ISSN: 1742-6588, Vol. 853, IOP Publishing 2017, doi:10.1088/1742-6596/853/1/012003.

[21] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1-7, doi: 10.1109/SysEng.2017.8088251.