A Fog Computing-based Framework for Privacy Preserving IoT Environments

Dhiah el Diehn Abou-Tair[1], Simon Büchsenstein[2], and Ala' Khalifeh[1]
[1]School of Electrical Engineering and Information Technology, German Jordanian University, Jordan
[2]Embedded Systems Engineering, University of Freiburg, Germany

Abstract: *Privacy is becoming an indispensable component in the emerging Internet of Things (IoT) context. However, the IoT based devices and tools are exposed to several security and privacy threats, especially that these devices are mainly used to gather data about users' habits, vital signs, surround environment, etc., which makes them a lucrative target to intruders. Up to date, conventional security and privacy mechanisms are not well optimized for IoT devices due to their limited energy, storage capacity, communication functionality and computing power, which influenced researchers to propose new solutions and algorithms to handle these limitations. Fog and cloud computing have been recently integrated in IoT environment to solve their resources' limitations, thus facilitating new life scenarios-oriented applications. In this paper, a security and privacy preserving framework is proposed, which utilizes Fog and cloud computing in conjunction with IoT devices that aims at securing the users' data and protecting their privacy. The framework has been implemented and tested using available technologies. Furthermore, a security analysis has been verified by simulating several hypothetical attack scenarios, which showed the effectiveness of the proposed framework and its capability of protecting the users' information.*

Keywords: *Internet of thing, cloud computing, fog computing, privacy, security*.

Received July 10, 2019; accepted Novebmer 24, 2019
https://doi.org/10.34028/iajit/17/3/4

1. Introduction

The Internet of Things (IoT) have been proliferated into several services and applications, which span our daily activities. This proliferation has introduced many benefits and added values to the human society. However, IoT based services exposed several privacy indirect and hidden threats, which are normally ignored by the technology adopters and users, due to the IoT applications attractive style, which is ubiquitous by nature, and may have an ambiguous architecture that does not reveal its information workflow, security and privacy pitfalls, and information ownership [5]. In a survey of Ericsson in 2017, people were asked about their main concerns of different IoT based services. The survey results revealed that 11% believed that there is no tangible benefits of deploying IoT solutions, 17% admitted that they do not know how to use this technology, 21% feared that machines will take over the human control and influence, 24% did not trust IoT reliability, 27% believed that IoT based services are not physically safe, 54% raised security threats, while 62% had serious privacy concerns [4]. These results show that there is an urgent need for providing trustworthy solutions to overcome these security, privacy and reliability concerns, threats and challenges [21].

The main functionalities of IoT devices and applications are to gather data about the users and their surrounding environments for further processing and decision making. These devices normally have unrestricted access to the personal information such as users' information records, locations, activities, etc., which pose a threat to users' privacy and security [22]. In addition, IoT devices have limited storage, power and computational resources, which makes applying typical security and privacy mechanisms inefficient and not practical. Further, the increasing growth in the number of IoT devices coupled with the widespread availability of applications and scenarios, inflates the amount of data generated, and thus pushing security and privacy threats to new high levels. Cloud computing has enabled many new disruptive applications and scenarios [9] without taking into consideration the possible security and privacy concerns that can affect these cloud-enabled services and applications. All of these challenges have fueled the researchers for innovating new solutions and paradigms, that aim at providing secure and privacy preserving services and mechanisms, while taking into account the limited resources of the IoT devices [3, 7] and the open nature of cloud computing based services. Moreover, in some applications, the gathered data requires further processing and analysis that will be used to accomplish the designated services. Consequently, the demand of computing power is substantial. Therefore, IoT peripherals integration with the cloud service provider is crucial to fulfill the users' and applications' computing and storage demand. However, such an integration process postures diverse challenges with respect to privacy and security. This is due to the fact that cloud service providers have the full control and ownership of the users' data, which can be used for users' activities and actions' tracking, recognizing the used IoT devices, types, their usage patterns, access time and logging frequency, thus invading the users' privacy and turning them to transparent users. Hence, it is mandatory to establish mechanisms and methods to protect the users' privacy and security by anonymizing and preserving their information. Furthermore, IoT devices have limited connectivity to the Internet, due to its mobile nature and limited energy sources, thus preferring short distance communication protocols such as WiFi, Bluetooth, ZigBee [10].

Fog computing has been introduced to act as an intermediator layer between the IoT and cloud layers in many scenarios [13]. This proposition is due to the fact that this device can have much higher computing, storage and energy resources than the IoT devices, which makes it on one hand, capable of providing long range communication functionalities to the Internet, utilizes more sophisticated technologies that support longer distances such as LoRaWAN, LTE, LTE-M [18]. On the other hand, this device provides short range, low energy wireless connectivity to the IoT devices. Further, this device normally has much more computational and storage capabilities when compared with the IoT devices, which enablesit to run programs capable of executing more resource demanding security and privacy algorithms and functions, thus the IoT devices connect directly to them and are used as trusted gateways to the cloud environment. In this paper, an IoT-Fog and cloud based framework is illustrated that can be used to ensure IoT users' privacy and security. The proposed solution is a continuation of our previous work [1, 2], where a conceptual privacy preserving IoT framework that utilizes the Fog computing paradigm as a middle layer between the cloud service provider and IoT devices has been introduced, which enforces advanced privacy and security mechanisms, and provides longer range communication capabilities to the IoT devices. This work provides a more technical detailed description and an implementation validation for the proposed framework utilizing the up-to-date solutions and technologies. Further, to show the framework robustness and efficiency, a security analysis pertaining potential security threats and attacks has been discussed. The proposed framework is of paramount importance since it provides an end-to-end framework for providing security and privacy for the IoT users' data which has been implemented and tested using of the shelf hardware products and available technologies thus enhancing its adoption and implementation among IoT users and community.

The rest of the paper is organized as follows. Section 2 summarizes the most related work and points out the framework contribution. Section 3 describes the proposed framework architecture. Section 4 demonstrates the framework implementation. Section 5 discusses diverse security attacks and scenarios. Finally, the conclusion is presented in section 6.

2. Related Work

In the literature, several works have been published tackling the IoT privacy and security challenges. Alrawais *et al*. [3] investigated the security and privacy challenges associated with the IoT and Fog computing integration and highlighted the main security and privacy concerns in Fog based IoT environments, mainly access control and authentication and emphasized that more policies and strategies should be introduced. Ni *et al*. [14] looked into the privacy and security challenges and threats associated with IoT-Fog based environments, and identified privacy revelation risks as well as a set of security attacks threats, namely, Sybil, Tampering, Forgery, etc., As a result of the conducted research, the authors pointed out diverse security and privacy open issues related to IoT-Fog computing. Lee *et al*. [11] discussed the security challenges and threats introduced by the IoT-Fog computing environment. With the goal of ensuring the differential privacy and security for IoT-Fog computing services. Wang *et al*. [20] proposed a privacy preserving content based publish/subscribe scheme utilizing differential privacy in Fog computing. Sha *et al*. [16] studied the security and privacy challenges associated with IoT and based on that, the authors introduced a model named EdgeSec to ensure a secure access to IoT environment and services. The proposed model consists of an edge layer security service implemented in the Fog layer, which contains several security measures and uses the standard communication protocols. A mobile based homomorphic-encryption framework has been introduced by Makkes *et al*. [12] that takes into account the mobile phones limited power capacity. However, the authors did not consider the cloud and Fog-computing integration. The users' privacy and security while moving between different cloud service providers has been analyzed by Dang and Hoang [6] and proposed several mechanisms to ensure users' information privacy and security within different clouds. The challenges of providing jointly Quality of Service (QoS) and security assurance while having low processing and management overhead have been presented in [18, 19]. The authors [8, 15] proposed an open-source medical IoT-Fog based computing environment, where the cloud is used to gather the patients' vital health data. However, the authors did not focus on the privacy and security challenges.

Wang *et al*. [19] already pointed out that there are various schemes protecting privacy with the common defect that the Cloud Service Provider (CSP) has to be trusted but do not consider that the CSP can sell the users' data,they proposed a privacy preserving three layer framework based on the hash-solomon code and Fog-computing which splits the data in three parts and each part is saved on one of the layers, the lowest layer where the data was produced will save 1%, while 4%, 95%, of the data will be saved at the Fog, cloud layers, respectively. As can be deduced from the conducted literature review, most of the work focus on security issues in either the Fog or cloud layers but not jointly, which may not give a comprehensive solution for protecting the users' privacy. In the literature, most of the proposed work [12, 16, 20] do not propose a comprehensive end-to-end solution for protecting the users' data and privacy against possible attacks. This paper proposes a framework focused on securing all the IoT Fog-based network architecture components, starting from the user end device, the Fog-device, and ending up with the cloud platform, which provides a comprehensive and holistic view for protecting the IoT users' data and privacy against different possible attacks. Furthermore, the implementation and technical details of the proposed framework have been described, which facilitates the adoption of this framework by IoT service providers.

3. Privacy Preserving Framework

Figure 1 depects the proposed IoT-Fog based framework, with an emphasis on the users' privacy and security.

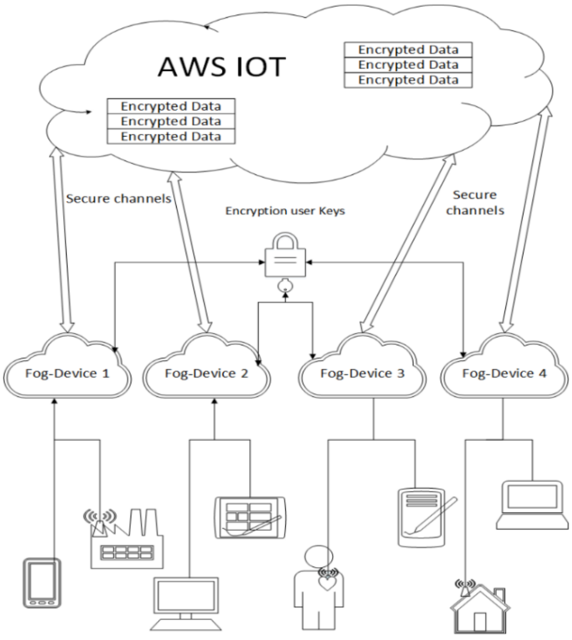

Figure 1. Privacy-Preserving Framework for IoT.

As shown in the figure, the framework consists of a set of IoT devics connected with th service prvider through a middle layer that consists of a set of Fog devices, which are responsible for establishing secure connections with the cloud service provider, taking into consideration the users' privacy. This scheme not only unloads the IoT devices from the heaviy and complex security computing processing, but also unveils the users' usage history and identities. The proposed framework is compsed from three main layers: IoT devices, the Fog and cloud computing layers. In what follows, a brief descrpition of each layer is provided.

3.1. Layer 1: IoT Devices

This layer is composed of several IoT devices with the purpose of providing the user with certain service. For instance, medical IoT devices can be used to enable patients of monitoring his/her health conditions. Further, these devices have limited computing and energy resources, that limits their ability to provide high security and privacy preserving mechanisms. In addition, due to the mobile nature of these devices and their prime functionalities in gathering users' data, it is important to protect them against potential attackers, who may intercept the users' gathered information, or even the IoT service providers themselves, who may be interested in learning more about the users' information nature, type, trends, access time, location, information history, etc., In order to achieve other business objectives (e.g., commercial advertising). Given these IoT devices limitations and security and privacy concerns, it is important to connect the IoT device to a more advanced computing resource (Fog layer) that can run more sophisticated security and privacy policies, which in one hand offloads the IoT devices and conserves their resources, and on the other hand, secures them from potential attackers and prevents the service provider from having the big picture of the users' information. This scheme puts the minimum-security functionalities on the IoT device, which includes providing the users with an authentication interface through which the users can enter his/her security credentials and utilize a secure communication channel with the Fog computing layer, which prevents the IoT generated data from eavesdroppers. Moreover, the IoT devices are stateless in terms of storing the users' data and can only save the captured data temporarily until data has been sent to the Fog layer for further processing and analysis. This makes the design and implementation of these devices economical, since they do not need to have high storage capabilities, thus cannot cause a security threat in case data was stolen or hijacked.

3.2. Layer 2: Fog Computing

The Fog layer acts as a high computing connection hub between the IoT and cloud layers, which is responsible for performing authentication, processing the IoT generated data, and establishing a secure communication channel with the cloud layer. The authentication flow for accepting an IoT user in the

Fog layer starts with receiving the users’ credentials through the IoT devices, which will be verified with the Fog layer users’ main database. To ensure a secure authentication process, the connection within the Fog layer is performed securely via the Hypertext Transfer Protocol Secure (HTTPS) protocol. Further, the database contains the users’ records which have the following main fields:

- Username and password stored as a Hash code.
- A 128-bit random generated identification code used for the purpose of encrypting the users’ credentials while accessing the cloud.
- Categories’ data types the have an ID and a sub-Key used to encrypt the categorized data ahead of storing it into the cloud data based.

After a successful authentication, the users’ security information needed to access the stored data in the cloud will be available through the Fog layer,which contains the following segments:

- The user identification code.
- The data type and its categories associated sub-Keys.

Users’ data can be classified into two categories according to the usage scenario. The first cataegory deals with data that needs to be stored without any processing,while the second category deals with data that needs to be processed before being stored. An example of the first category is a typical medical record data which contacins some vital information such as:users’ weight, tempertaure, blood pressure, blood sugare level, etc. For the second category, a medical image that needs to be analyzed to detect abnormality or to recognize certain observation, is an example of data that needs to be processed before being stored.

For both categories, the Fog layer acts as intermediator between the IoT and the cloud computing layers. In order to access the cloud layer, a secure channel is established between the cloud and Fog layers to securly transfer the data. If the data belongs to the first category, it will be encrypted using the Advanced Encryption Standard (AES) algorithm using the users' appropriate sub-Key, which hides the user data from the cloud, thus the user has the overhand of his/her data in the cloud since the key is saved among the users' security data in the Fog layer. In order to distinguish the users' encrypted data in the cloud, the identification code of the security database will be used. This code is generated at the Fog layer and determines the user identification and private information, which is only available at the Fog layer. For the second category, data will be sent via a secure channel to the cloud without encryption, since the cloud needs an unencrypted data in order to process it. After processing the data, the output data of the processing results are sent back to the Fog layer, where both the user data and the results are encrypted using the appropriate sub-Key, with the user identification code and data type. The encrypted data are then transferred to the cloud. Following this procedure, the cloud service provider will not be able to recognize the users’ identity, data hiostory, and the assocaite users’ activity paterns, since the data are encrypted using the users' keys, which are unknown to the cloud layer. Furthermore, the user-data association is performed via the identification code, which does not have any user information. Further, this mechanism ensures that the cloud layer will not be able to deduce any relationship between the unencrypted data and the user stored information and identity.

3.3. Layer 3: Cloud Layer

The cloud layer perfroms data storage functionlity as well as providing data processing and analysis services. to handle the aferomention data categories. For the first category, the data will be encrypted by the user defined keys, along with the user identification codes, and transferred securely to the cloud for storage purposes. For the second category, the cloud receives such data anonymized by the user reference code but unencrypted. Once the data is processed, the results will be sent back to the Fog layer to associate it with the user raw data and reference code. Both the raw data and the results will be encrypted by the user encryption keys, which will be sent encrypted to the cloud for storage purpose.

3.4. Framework Flow Diagram

This subsection elaborates on the framework operational flow depicted on Figure 2. The operation starts when the user accesses an IoT device for a certain service, which sends the users’ credentials to the Fog layer for authentication purpose. The Fog layer contains the users’ database which has a record for each user containing authentication and security information which consists of a 128-bit random number used as an identification code which acts as a user-ID as well as a user password. The user-ID hides the user identity while accessing the cloud services for the purpose of preserving the user privacy. Further, the user record contains a data type that determines the nature of services allowed to the user which are given by the IoT service provider. The data type includes multiple categories the user can access. For example, in a medical care scenario, an IoT service provider will enable the user to obtain different medical data, which can be classified into different categories such as: body pressure, temperature, blood sugar concentration, etc. For ensuring users’ privacy and security while accessing the data, a sub-Key is assigned to each data type category. Both, the sub-Key and the user-ID will be jointly used to encrypt the user data. Further, not only the user data is encrypted at the Fog layer, but also the communication channel between the IoT devices and Fog layer is secured via the HTTPS protocol. When the user is successfully authenticated, the Fog layer will enable the service flow by means of identification code and data category.

After the authentication, the user can capture and upload new data through the IoT device to the cloud or access a pre-existing data. The captured data is classified into two categories, according to the service nature. The first category is data that does not need processing, which will be directly encrypted and stored in the cloud. While the second category deals with data that needs further processing, which may trigger another service or/and store the processing results and data encrypted in the cloud for offline investigation. The encryption of the data and the processing results will be performed in the Fog layer utilizing the user reference code and data type category sub-Key. Furthermore, the data in this category will be linked with a temporary reference code used for disassociating the user-unencrypted data with the user actual identity given by the user-ID, which ensures the user privacy. In case accessing the user certain pre-stored data, an access request will be sent to the cloud, which in turns will be retrieved by means of the user reference code and data type with the associated sub-Key to be decrypted in the Fog layer.

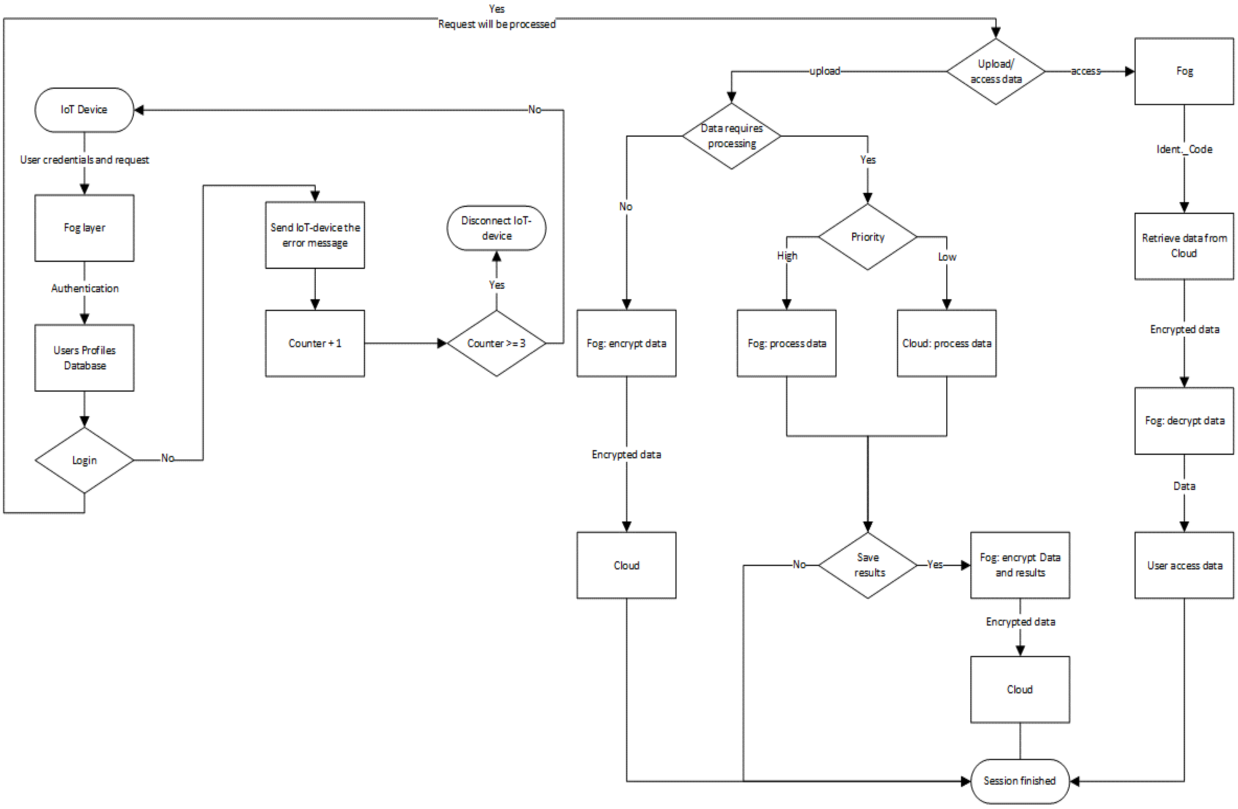

Figure 2. Framework flow diagram.

4. Implementation

This section describes the technical implementation details of the aforementioned framework. The implementation utilizes off the shelf technologies and serves two purposes. First, it validates the applicability of the proposed framework. Second, it acts as a research and development test-bed, which can be used to further investigate several research aspects in the Internet of Things secuirty and privacy domain. In what follows, a detailed implementation description of each layer is provided utilizing available software and hardware products.

4.1. IoT Layer

The IoT layer is modelled by a mobile device running a mobile application written in Java programming language that provides the user with an interface to upload and access the data. The application launches by allowing the user to enter his/her authentication credentials and gives the user the option of performing several actions on the data such as uploading new data or reading existing data. Further, the application simulates the process of continuously sending data by switching into the sensor mode. The mobile device is connected to the Fog layer via Bluetooth; the connection is established with the Android API for Bluetooth classic, for this, the serial port profile was used to establish the data communication pipe. The authentication process is preformed as follows. First, the user credentials will be authenticated, if it fails, the user will be directed again to the login page, if authentication succeed, then the user request will be processed. The user can choose one of four different options: Sensor data Generation process (Sen-Gen), Data Access (Data-Acc), and Data Addition (Data-Add). If the user chooses the Sen-Gen, then the mobile application will work in the sensor mode, where it simulates a sensor device that is transmitting data continuously. If the user chooses the Data-Acc option, then the user data will be retried from the cloud after being decrypted using the designated user decryption keys. If the user chooses Data-Add, the data will be sent to the cloud for inserting a new data entry to the user profile.

4.2. Fog Layer

The Fog layer is implemented using a Raspberry Pi 3 microcontroller which communicates with the IoT layer, security server and cloud, thus acting as a communication hub for the entire system. This communication paradigm is modelled by a state machine diagram as depicted in Figure 3, which consists of three main states: Idle, Authenticate and Process. If there is no active IoT devices, then the Fog device will be in IDLE status. As soon as a request is initiated by an IoT device, the status of the Fog layer will be changed to Authenticate, where it will validate the IoT credentials throughout the security server. In case of successful authentication, the Fog layer status will be changed to process, where it will handle the users' requests. In case of failed authentication, then the authentication process will be repeated for a determined number of trials. This mechanism is vital to protect the Fog layer from an unauthorized access that can result from possible brute force attack. If the authentication process did not succeed within the allowed trials, the IoT device will be rejected and reported to the system administrator for further investigation. To ensure a secure authentication process, HTTPS-requests will be invoked by the Fog device, which sends the Hash codes of the user credentials and the request type and information to the security server. The HTTPS communication with the security server is implemented on the server-side using SHA256 algorithm provided by Open Secure Sockets Layer (OpenSSL). Finally, if an error occurs while processing a request or the IoT-device performed its designated task, the Fog will returntothe IDLE state.

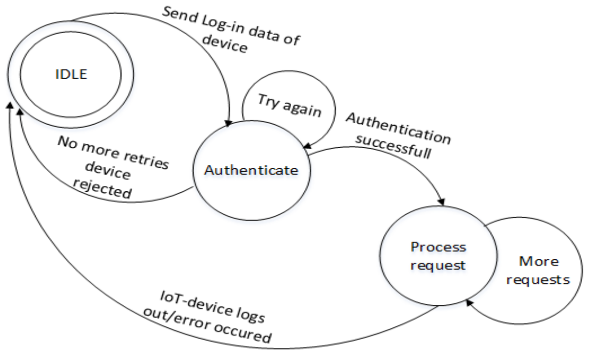
Send Log-in data of device
IDLE
Try again
Authentication successfull
No more retries device rejected
Authenticate
Process request
More requests
IoT-device logs out/error occured

Figure 3. Fog layer state machine model.

4.2.1. Cloud Layer

The implementation of the cloud layer is based on a public cloud service, which puts more challenge on assuring the users' data privacy. In particular, the Amazon Web Services (AWS) are used as the cloud service provider. For data storage and processing purposes, the AWS-DynamoDB database and the AWS Lambda service are used. Further, to assure a secure and trusted communication between the Fog and the cloud layers, the Fog layer should integrate the security certificate issued by the cloud service provider and utilize the HTTPS protocol in securing the data transmission functionality. The subsequent sections illustrate the implementation details of the cloud layer various building blocks.

4.2.2. AWS-DynamoDB

DynamoDB is a NoSQL database of the AWS through which security is established by signing the data with a predefined encryption key that identifies and authorizes the Fog layer. The user data is classified into several categories identified with a Data-ID that is associated with an encryption sub-Key, used to encrypt/decrypt the data into the user record stored into the DynamoDB. The Database record structureconsists of the user identification code, data categories and the encrypted data. It is paramount to emphasize that this structure anonymizes the user data from the cloud service provider, since the Database (DB) stores only the users' IDs, which consist of random numbers, associated with the users' encrypted data. Notice that the user data are classified into several categories, each of them is associated with an encryption sub-Key that is used to encrypt/decrypt the data according to their categories. This mechanism allows the user to access a particular data (by accessing the data category ID), in case of performing an update or read process, which saves the user from retrieving his/her entire record, thus making the process much more efficient.

4.2.3. AWS Lambda

AWS Lambda is a service that lets you run code in response to an event that is triggered into the DynamoDB automatically without the need of managing the backend server. It supports various programming languages such as Java, Go, C#, and Python, etc. The code written in this environment is named as Lambda function. The purpose of utilizing this feature in the proposed framework since this function can be used to trigger a certain processing event that can be used to analyze the users' data thus triggering a certain action. For example, it can be used for monitoring a patient health condition, where the user-generated data from medical IoT sensors (e.g. blood pressure, sugar level and temperature) are analyzed and an event is triggered notifying the patientmedical doctor, in case of detecting an abnormal situation.

4.3. Processing Requests

After successful authentication, the request is processed depending on the request number. This is described by the following algorithm depicted in Figure 4. In what follows, a brief description for the requests' processing cases is presented.

- *Authentication case:* This case occurs while authenticating the IoT-device especially if one of the parameters send by the IoT-device was invalid like:wrong username or password.The error-message received from the security server, after comparing these values with the ones saved there, will then be send to the IoT-device in case the device has its own error handling mechanism. Each time this case occurred, a counter will be incremented and if the counter reaches three, the IoT-device will be disconnected as a security measure. While the counter is less than three, the IoT-device has the chance to repeat the log-in process.
- *Processing case:* The authentication function will only give back 1 as the request number, if the authentication was successful without any ID or key, because no data had to be accessed yet. The protocol can either be the Message Queuing Telemetry Transport (MQTT) [17] or HTTPS, where MQTT is less power consuming and faster option and might therefore be better for Fog devices, both protocols are encrypted with Transport Layer Security (TLS). In case the data needs processing, the functions at the cloud layer will be executed with the help of the AWS Lambda service.
- *Get data case:* If a request-number 2 was given back, the answer from the security server will also be searched for user-ID, sub-ID and the sub-Key. The ID's and key are received as Hex-string because at the security-server side, data are stored in a JavaScript Object Notation (JSON) file which doesn't support binary data. The ID's will stay that way, but the sub-Key has to be decoded to a byte array before using it as a valid AES-key. To access DynamoDB database, a Python script was written which will take the user-ID and sub-ID and a binary file as arguments and then write the data found to the corresponding ID's in the cloud to a binary file,on success, the main program will get back the number of bytes written to the binary file. The program can now read out the data from the binary file, decrypt it with the sub-Key and the AES-algorithm. The Advanced Encryption Standard (AES) algorithm was implemented with the OpenSSL library.

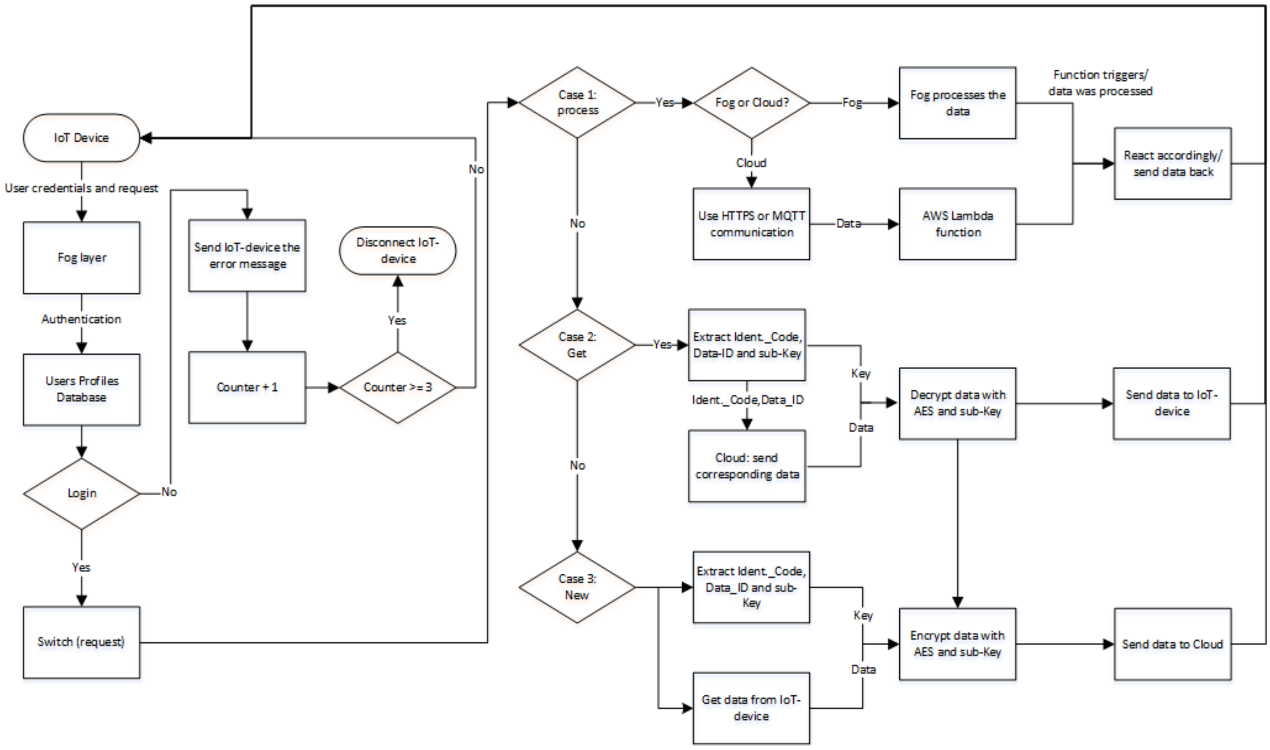

Figure 4. Fog requests handling.

- *New data case:* With the request-number 3, the user data has to be extracted as well. The IoT-device will be informed with a message when the authentication was successful and that it can send the data now. The data will be read in a buffer and if the data is not a multiple of 16, a zero padding will be used so that it will fit the block-size of the AES algorithm. The zero padding was used because of the easy reproducibility while decrypting. Afterwards, the AES and sub-Key will be used to encrypt the data and then write it to a binary file. For sending the data to DynamoDB, a Python script will be called with the IDs and the name of the binary file. A script will then read the data out and write it to the corresponding DynamoDB table using JSON-format. In fact, binary data is saved in Base64 format in the DynamoDB table.

4.4. Security Server

The security server was established with Node.js, which is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Because of that, the implementation of the server was completely written in JavaScript. The connections itself will be established with the HTTPS protocol, for the Rivest-Shamir-Adleman (RSA)-algorithm, a public and private keyare needed. Both of them were generated with the Open SSL library. The certificate to authenticate the server was also generated by OpenSSL and then it was self-signed.

The server uses the REST-API to communicate with the Fog client. With this server implementation, a client is able to use the post command to send the server its security credentials for validation. The security data itself are saved in a JSON-file but for bigger data, a database should be used. The JSON-format has limitations when it comes to binary data, that is problematic because the keys and IDs are only secured if binary data are used, where every byte has a value from 1-256 instead of using a JSON compatible UTF-8 string, which only has a value from 1-128.Because this would result in a weaker key, the data will be coded to a hexadecimal string before saving it in the JSON-file. For the authentication process, the server will loop through the JSON objects if there is a user with the same username and password, both hashed for security purposes. In what follows, a description of the three main cases is provided.

- *Case* 1: For processing only, the authentication process is needed and therefore the security server will only return a message if it was successful or not.
- *Case* 2: For getting data, the server will first check if the required data exist, otherwise it will return an error. If the data exist, it will send back the IDs and the sub-Key related to the data and user mentioned in the request.
- *Case* 3: For new data, the server will search if the data already exists and will inform the Fog layer about the data availability, if the data did not exist, a sub-ID and sub-Key are generated with a high entropy cryptography random number generator provided by Node.js.

5. Security Analysis

The purpose of this section is to examine the system security and identify the potential security threats. Further, an analysis for the known attacks is provided to estimate their severity level. The first part will focus on the system users' side and identify the consequences in case it is compromised by a hacker. While the second part will focus on the security of the utilized communication technology, how attacks on them may affect the system flow and what should be changed to further secure the system.

5.1. Hacking the IoT device

This scenario assumes the IoT-device is hacked with an attack on the device access security credentials. This threat could happen if the IoT-device's credentials were revealed by a hacker, which could occur in case of choosing a weak password or when having a brute-force attack. With this information, the hacker can send all the device generated requests to the Fog device, thus retrieving all the decrypted data from the cloud, and possibly adding new fake data or deleting existing data. Accordingly, the hacker would have full access to this IoT-device and the cloud services using the IoT-device credentials, consequently, the privacy of the user who is using this device will be compromised.

5.2. Hacking the Fog Layer

In this scenario, the hacker had access to the Fog layer device. In this case, the hacker can disable the communication between the Fog layer, IoT device and the cloud layer, thus affecting the system reliability and causing a Denial of Service (DOS) attack. However, the hacker can neither intercept the exchanged credentials with the IoT device nor access the security server or the cloud, since this information is encrypted.

5.3. Hacking the Security Server

Another potential attack can occur on the security server, where the hacker can retrieve all the data, which are saved there, i.e., all the IDs, sub-Keys and the users' credentials. Because the data are saved encrypted, the hacker will not get any users' credentials and therefore the attacker will not be able to query the cloud for data without the help of the Fog layer device, and without having the correct access users' credentials. However, similar to the previous attack, the hacker can cause a DOS attack by disabling the communication between the server and the Fog layer.

5.4. Hacking the Cloud

This scenario describes a security threat that could happen if the hacker was able to access the cloud and retrieve all the stored data. However, retrieving this data is not useful, since the hacker nor can use the users' IDs to request data neither can decrypt the data without the help of the associated sub-Key saved on the security-server. Thus, the main harmful action is to affect the communication traffic between the Fog and cloud layers, which may degrade or disable the service provided by the cloud layer.

5.5. Attacks Summary and Recommendations

After studying the various security threats, it is clear that the Fog layer and the IoT devices are the most critical parts that can be compromised. This is because the Fog and the IoT devices are in most cases, not as protected as the server and the cloud. Therefore, these devices should be carefully monitored, updated, maintained and inspected to avoid any potential security breach. On the other side, attacking the security server or the cloud could enable the hacker of retrieving the users' credentials and data, which are useless as they are encrypted. Regarding the potential DOS attack, it can be detected and stopped by regularly monitoring the network traffic and flow, thus eliminating this threat and stop it promptly. All of these potential threats have been taken into account while designing and implementing the proposed system. Regarding the ongoing traffic between the cloud, Fog and the security server, the HTTPS protocol is used and encrypted with TLS, which encrypts the communication and authentication data against vulnerability of the Man-in-the-Middle Attack (MITM). The Fog and cloud layers provide mutual authentication to eachother to secure both ends. The security server is located inside a private network that is usually protected with a firewall from external attackers. The big challenge in the current implementation is the reliance on Bluetooth as the wireless technology used to connect the IoT and the Fog devices. Bluetooth is insecure technology especially at the pairing process. The pairing process establishes key information on both devices, which then will be used for encrypting the communication. This process is vulnerable to MITM. There are some methods to prevent this potential threat, for example by using one of the suggested methods by the Bluetooth Special Interests Group (SIG). For instance, to use an out of band pairing which suggests using another band, e.g., Near Field Communication (NFC) to do the pairing process, which has a small range of approximately 10 cm, thus any MITM-attack can be detected.

6. Conclusions

In this paper, a Fog and cloud-based IoT framework is proposed that can be used to provide a secure and privacy-preserving environment for IoT oriented applications. In the proposed framework, users access the IoT device and cloud through a Fog layer that acts as a mediator, which handles the data processing and transfer between the IoT layer and the cloud by providing proper encryption and anonymization mechanism, which guarantees an end-to-end security and privacy handling for the exchanged information. Furthermore, the technical implementation details utilizing available hardware and software utilities along with a security analysis have been provided to facilitate the framework adoption among users and show its robustness and effectiveness.

References

[1] Abou-Tair D., Büchsenstein S., and Khalifeh A., "A Privacy Preserving Framework for the Internet of Things," *in Proceedings of the 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Busan, pp. 27-31, 2018.

[2] Abou-Tair D., Alouneh S., Khalifeh A., and Obermaisser R., "A Security Framework for Systems-of-Systems," *in Proceedings of the International Conference on Ubiquitous Information Technologies and Applications International Conference on Computer Science and its Applications*, Taichung, pp. 427-432, 2017.

[3] Alrawais A., Alhothaily A., Hu C., and Cheng X., "Fog Computing for the Internet of Things: Security and Privacy Issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34-42, 2017.

[4] Anonym, "IoT Security: Protecting the Networked Society," *Ericsson White Paper*, 2017.

[5] Chow R., "The Last Mile for IoTPrivacy," *IEEE Security and Privacy*, vol. 15, no. 6, pp. 73-76, 2017.

[6] Dang T. and Hoang D., "A Data Protection Model for Fog Computing," *in Proceedings of the 2nd International Conference on Fog and Mobile Edge Computing*, Valencia, pp. 32-38, 2017.

[7] Hiari O., Abou-Tair D., and Abushaikha I., "An IoT-Based Virtual Addressing Framework for Intelligent Delivery Logistics," *in Proceedings of the International Conference on Information Science and Applications*, Macau, pp. 698-705, 2017.

[8] Khalifeh A., Saleh A., AL-Nuimat M., and Tair D., "An Open Source Cloud Based Platform for Elderly Health Monitoring and Fall Detection," *in Proceedings of the 4th Workshop on ICTs for Improving Patients Rehabilitation Research Technique*s, Lisbon, pp. 97-100, 2016.

[9] Kiblawi T. and Khalifeh A., "Disruptive

Innovations in Cloud Computing and Their Impact on Business and Technology," *in Proceedings of the 4th the International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, pp. 1-4, 2015.

[10] Krejčí R., Hujňák O., and Švepeš M., "Security Survey Of The Iot Wireless Protocols," *in Proceedings of the 25th Telecommunication Forum (TELFOR)*, Belgrade, pp. 1-4, 2017.

[11] Lee K., Kim D., Ha D., Rajput U., and Oh H., "On Security and Privacy Issues of Fog Computing Supported Internet of Things Environment," *in Proceedings of the 6th International Conference on the Network of the Future*, Montreal, pp. 1-3, 2015.

[12] Makkes M., Uta A., Das R., Bozdog V., "P^2-SWAN: Real-Time Privacy Preserving Computation for IoT Ecosystems," *in Proceedings of the IEEE 1st International Conference on Fog and Edge Computing*, Madrid, pp. 1-10, 2017.

[13] Mukherjee B., Neupane R., and Calyam P., "End-to-End IoT Security Middleware for Cloud-Fog Communication," *in Proceedings of the IEEE 4th International Conference on Cyber Security and Cloud Computing*, New York, pp. 151-156, 2017.

[14] Ni J., Zhang K., Lin X., and Shen X., "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 1, pp. 601-628, 2018.

[15] Obermaisser R., Abuteir M., Khalifeh A., Abou-Tair D., "Systems-of-Systems Framework for Providing Real-Time Patient Monitoring and Care: Challenges and Solutions," *in Proceedings of ICTs for Improving Patients Rehabilitation Research Techniques. Communications in Computer and Information Science*, Oldenburg, pp. 129-142, 2015.

[16] Sha K., Errabelly R., Wei W., Yang T., Wang Z., "EdgeSec: Design of an Edge Layer Security Service to Enhance IoT Security," *in Proceedings IEEE 1st International Conference on Fog and Edge Computing*, Madrid, pp. 81-88, 2017.

[17] Shaout A. and Crispin B., "Using the MQTT Protocol in Real Time for Synchronizing IoT Device State," *The International Arab Journal of Information Technology*, vol. 15, no. 3A, pp. 515-521, 2018.

[18] Vallati C., Virdis A., Mingozzi E., and Stea G., "Exploiting LTE D2D Communications in M2M Fog Platforms: Deployment and Practical Issues," *in Proceedings of the IEEE 2nd World Forum on Internet of Things*, Milan, pp. 585-590, 2015.

[19] Wang T., Zhou J., Chen X., and Wang G., "A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 3-12, 2018.

[20] Wang Q., Chen D., Zhang N., Ding Z., Qin Z., "PCP: A Privacy-Preserving Content-Based Publish-Subscribe Scheme with Differential Privacy in Fog Computing," *IEEE Access*, vol. 5, pp. 17962-17974, 2017.

[21] Weber R., "Internet of Things: Privacy Issues Revisited," *Computer Law and Security Review*, vol. 31, no. 5, pp.618-627, 2015.

[22] Zhou W., Jia Y., Peng A., Zhang Y., and Liu P., "The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606-1616, 2019.

Dhiah el Diehn Abou-Tair, associate professor at the German-Jordanian University (GJU), received his PhD from the University of Siegen, Germany. During his PhD, Dr. Abou-Tair conducted research about the adoption of privacy laws and regulations in information systems through an ontology-based approach. His current research interests are in the areas of privacy-enhancing technologies, security and IoT. Dr. Abou-Tair has been involved in several EU and German-funded research and capacity building projects.

Simon Büchsenstein received his B.Sc. degree in electrical engineeringfromFurtwangenUniversity, Germany.Currently he is doing his M.Sc. in embedded systems engineering at theUniversity of Freiburg.His research interests include cryptography and embedded security.

Ala' Khalifeh received the PhD degree in Electrical and Computer Engineering from the University of California, Irvine -USA in 2010. He is currently an Associate Professor in the Communication Engineering department at the German Jordanian University and the department chair. His research is in communications technology, and networking with particular emphasis on optimal resource allocations for multimedia transmission over wired and wireless networks, Quality of Service, Internet of Things and Wireless Sensor Networks.