On Securing the Communication in IoT Infrastructure using Elliptic Curve Cryptography

Hugo Bourreau, Emeric Guichet, Amine Barrak, Benoît Simon, and Fehmi Jaafar
Dpt. informatique et mathématique, UQAC, Saguenay, Canada
hugo.bourreau1@uqac.ca, emeric.guichet1@uqac.ca, mabarrak@uqac.ca,
benoit.simon1@uqac.ca, fehmi.jaafar@uqac.ca

***Abstract*—Internet of Things (IoT) is widely present nowadays, from businesses to connected houses, and more. IoT is considered a part of the Internet of the future and will comprise billions of intelligent communication. These devices transmit data from sensors to entities like servers to perform suitable responses. The problem of securing these data from cyberattacks increases due to the sensitive information it contains. In addition, studies have shown that most of the time data transiting in IoT devices does not apply encrypted communication. Thus, anyone has the ability to listen to or modify the information. Encrypting communications seems mandatory to secure networks and data transiting from sensors to servers.**

In this paper, we propose an approach to secure the transmission and the storage of data in IoT using Elliptic Curve Cryptography (ECC). The proposed method offers a high level of security at a reasonable computational cost. Indeed, we present an adequate architecture that ensures the use of a state-of-the-art cryptography algorithm to encrypt sensitive data in IoT.

***Keywords*—Cryptography, Elliptic-curve, Internet Of Things, IoT**

I. Introduction

In the last decade, the number of businesses that use the IoT technologies are increasing every year. In 2021 there were more than 13 billion active IoT devices and are projected to reach 43 billion by 2023 [1]. Such equipment requires robust algorithms to ensure a secure communication, which means cryptography. However, the level of cryptography in IoT architecture communication is not secure enough. Nowadays, at least 98% of IoT devices do not use encrypted communication [2], due to resource limitations [3]. Moreover, computing power as well as network complexities have evolved and reached high efficiency, especially with the arrival of quantum computing. This increase the exhibition of IoT devices to threats of cyber attacks.

IoT architectures possess numerous points of entrance that can be used by hackers in diversified attack scenarios. These scenarios include entering in a server without access to collecting data from database illegally, intercepting information used to communicate between two entities, especially while using IoT devices, switching the information transiting between two or more equipment, etc. Thus, the number of possible entrance and attacks are varied and numerous [4].

In this work, we propose a distributed and secure architecture of connected IoT devices, which leads to the following problem : How can we ensure secure, encrypted communication and data storage between IoT devices ?

II. Methodology

In this section, we explain the architecture used in this work. The main goal is to ensure data security in the Internet of Things, from data collection to its usage. Between these two points, there are many steps in which data are transmitted and saved. During this work, we take into consideration that every machine or network that will carry the data can be a threat. We therefore must take the necessary steps to cover these different aspects.

Figure 1 describes the overall proposed architecture of the cloud server. We observe several sensitive transfer points between the different entities. The data will be shared across the internet at least twice before being used. First, they are sent to the server to be stored, and then when the user has access to them. We used a distributed architecture using asymmetric encryption (pair of public and private keys). The first part is a local network where sensors will be able to collect data on their environment and will transmit them to controllers. These can be low-power components such as Arduino or ESP boards.

The local gateway will be registered to the cloud server and the user will log in using his email address, password, and private key. Here, the private key is not sent to anyone to ensure authenticity. If the user is successfully logged in, the cloud server will send the public key to the client. Once registered the data from the sensors will be sent to a local gateway which will encrypt the data using the user's public key and then send it to the remote server.

This implementation ensures several advantages, such as the ability to use any type of sensor without having to manually update a list or even adjust formats. Scaling (adding/removing sensors) is also favored since only the number of requests will vary. Indeed, it is necessary to be careful about network saturation and build an architecture accordingly.

The collected dataset from the different devices is safely stored. As presented in Figure 1, the remote server will receive the requests sent by the local network. Its roles are to save the received data from each user and to resend it when it is requested. It includes the following different components:

- An SQL database where we can find user accounts, people with whom we agree to share the data, and the data itself.

2693-9371/22/$31.00 ©2022 IEEE
DOI 10.1109/QRS-C57518.2022.00121

- An API to allow users to authenticate and receive encrypted data.
- A complete website that allows us to manage the account, the persons allowed to share the data with, and visualize the data in a graph form.

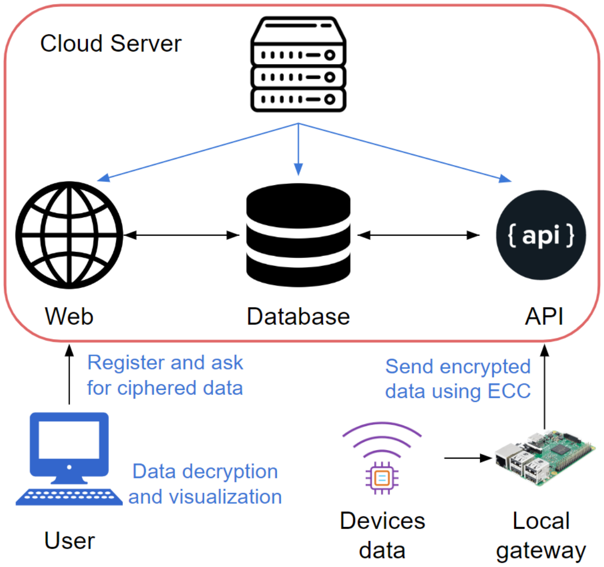

Figure 1. Architecture of the cloud server

To ensure secure communication between the different stakeholders, we use an HTTPS site using an RSA-SSL certificate (2048 bits). This will allow a secure data transfer between the local gateway and the remote one. However, this is not enough. If a hacker compromised the remote server, he could see the data arrive and bypass the security used for communication. This protection secures the connection between the client and the server. However, we still need to protect the data stored on the server and send it to the correct user.

At this stage, we need to choose an encryption algorithm to secure communication between IoT devices. We found that the Elliptic Curve and RSA algorithms are highly used in the communication between IoT devices [5]. Gura et al. [6] discuss how ECC is efficient over RSA on 8-bit processors and how to improve ECC performance (almost useful for embedded systems). Therefore, we chose to use elliptic curves to secure the data from end to end, since the ECC keys are shorter than the RSA keys. To be able to encrypt the data, we generate a pair of keys for the user during registration. The public key is recorded, and will be used to encrypt the data, as for the private key, it is communicated to the user but is not stored on the server. Thanks to this method, the data is encrypted before being sent to the remote server. This allows us to fix the problem if the server is compromised. Indeed, even if a hacker succeeded in recovering all the information from the SQL tables, he could not recover the data from the sensors. However, it could collect certain information such as user's email addresses. It would still not be possible to have access to their accounts, since the password is encrypted using the BLOWFISH algorithm as well as the user's private key (which is not saved on the server). Note that it is not possible to decrypt the data on the server.

For the user part, as mentioned before, the private key is never sent to the server. We still ask it to authenticate the user through a challenge. This situation allows us to be sure that only the user can see the data since it is encrypted and only he has the private key. When the user requests access to the data, he will receive everything encrypted that is deciphered locally. Since the protection of the private key is under the responsibility of the user, physical access to his device should be protected. Our proposed approach did not cover this aspect.

III. Conclusion

The process of securing the communication among devices is an important part of IoT. Sensitive data communication, transmission and storage are key points of this process. Every communication on the internet contains data that has different sensitivity level that it needs an appropriate methods to protect them. In this paper, we present an architecture that ensure data protection when transmitted and when stored. We presented how ECC can be used to protect data and how encrypted storage offers a high level of security against server hacking. Our proposed architecture can be used among an existing IoT implementation and completed with its encryption features.

As a future work, we will evaluate the efficiency of the proposed method regarding the security and the cost of additional computational level. We will extend our study to other devices and propose an API that can easily be integrated in an existing IoT project.

References

[1] Statista, "Internet of things (iot) and non-iot active device connections worldwide from 2010 to 2025." [Online]. Available: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/

[2] "2020 unit 42 iot threat report 2020 unit 42 iot threat report," https://unit42.paloaltonetworks.com/iot-threat-report-2020/, (Accessed on 09/07/2022).

[3] S. Zahoor and R. N. Mir, "Resource management in pervasive internet of things: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 8, pp. 921–935, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157818305858

[4] Y. Shah and S. Sengupta, "A survey on classification of cyber-attacks on iot and iiot devices," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2020, pp. 0406–0413.

[5] Z. Vahdati, S. Yasin, A. Ghasempour, and M. Salehi, "Comparison of ecc and rsa algorithms in iot devices," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 16, 2019.

[6] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 119–132.