A PUF-based IoT Authentication Model in Edge Computing Environment

2023 9th International Conference on Computer and Communications (ICCC) | 979-8-3503-1725-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCC59590.2023.10507405

Qing Zhu
Zhengzhou Information Science and Technology Institute
Zhengzhou,China
zqoyna@163.com

Sensen Li*
Zhengzhou Information Science and Technology Institute
Zhengzhou,China
lss589@163.com

Bin Yu
Zhengzhou Information Science and Technology Institute
Zhengzhou,China
byu2009@163.com

***Abstract*—Edge computing proposes to solve the problems such as high bandwidth load and long delays when a great number of terminal devices connect to the IoT. However, the addition of the edge layer also poses new challenges to authentication security between devices at all levels of the IoT. The characteristics and communication requirements of cloud, edge, and terminal devices in the edge computing environment has been analyzed, and four types of authentication schemes is designed, which provides a basis for further research on the authentication problem of IoT devices in edge computing environment.**

Keywords—edge computing, IoT, authentication model, PUF

I. INTRODUCTION

With the continuous development of information technology, the Internet of Things (IoT) is widely used in many fields such as e-health, smart factory, smart home, and the Internet of Vehicles (IoV) [1]. According to the GSMA[2], global IoT connections will reach 24.6 billion by 2025. However, the access to massive heterogeneous terminal devices exposes the shortcomings of the cloud-end architecture, such as large bandwidth load and long communication delays. Edge computing extends the IoT from the two-layer architecture to the three-layer architecture of the cloud-edge-end. Analyzing and processing data on the edge nodes near terminal devices can effectively reduce the data transmission bandwidth, reduce system latency, alleviate the cloud center pressure, and improve availability. At the same time, the concept of data nearby processing also provides better-structured support for information security, which can protect data security and privacy to a certain extent[3].

Edge computing derives from the traditional cloud-end architecture and inevitably carries communication security risks in the general cloud environment. In addition, since edge computing presents a distributed security domain and its communication subjects are more vulnerable to attack, edge computing also has its communication security risks[4]. Authentication is an important basis for secure network communication and the first line of defense to ensure the safe operation of the IoT system[5]. The IoT authentication schemes with cloud-end architecture [6-8] have several shortcomings, such as terminal device registration and authentication relying on cloud center, edge nodes are vulnerable to replication attacks after being captured, and cannot be applied to the IoT in the edge computing environment. In recent years, many scholars have conducted relevant research to solve the mutual authentication issues among cloud, edge, and terminal devices in edge computing [9-12]. However, most of the proposed solutions are improvements and optimizations based on the two-layer architecture of IoT security protocols. While these solutions can avoid complex calculations on the terminal devices and achieve lightweight authentication, they do not fundamentally change the structure of the cloud center as the authentication core in terminal device authentication, which can easily result in single-point failure and communication latency issues.

In this paper, a cloud-edge-end integrated IoT authentication model based on the physical unclonable function (PUF) is proposed. The followings are the novelty and contributions of this paper:

(1) We analyze the characteristics and communication requirements of cloud, edge, and terminal devices in the edge computing environment.

(2) According to the characteristics of IoT devices in edge computing environment, a cloud-edge-end integrated IoT authentication model based on PUF is proposed, which provides a basis for studying the authentication security issues between IoT devices in edge computing environment.

(3) We describe the cloud-edge, edge-edge, edge-end and end-to-end authentication processes and key technologies in detail, which provides ideas for further designing authentication schemes.

The remainder paper is organized as follows. We briefly introduce the architecture and data transfer method of IoT in the edge computing environment in Section II. In Section III the characteristics and communication requirements of cloud, edge, and end layers are analyzed from the perspective of identity authentication. A cloud-edge-end integrated IoT authentication model based on PUF is proposed in Section IV. The authentication processes of cloud-edge, edge-edge, edge-end, and end-end are explained in Section V. Finally, we conclude this paper in Section VI.

II. PRELIMINARIES

A. The IoT in the edge computing environment

Compared with the cloud-end architecture, the IoT in the edge computing environment adds an intermediate edge layer, as shown in Figure 1. The IoT architecture in the edge computing environment includes three types of entities: cloud center, edge node, and terminal device. The cloud center is responsible for the authentication of edge nodes, collecting nodes data, and responding to edge nodes requests. Edge nodes are servers with certain storage and computing capabilities that are deployed near-end devices. They divide massive terminal devices into different security domains. Each edge node is responsible for authenticating the terminal nodes in its security domains. Terminal devices are IoT devices and mobile intelligent devices, responsible for data collection and task execution, such as cameras, sensors, control switches, etc.

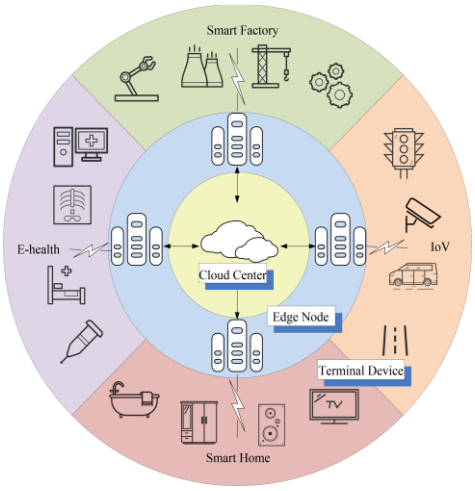

Fig. 1. The three-level architecture of IoT in the edge computing environment.

The communication process between cloud, edge, and end levels includes data uploading from the bottom up and command issuing from the top down. Terminal devices are deployed in hospital beds, factories, roads, and other locations to monitor the physical condition of patients, machine operation, and road operation and transmit relevant data to edge nodes. After receiving data, edge nodes perform operations such as discarding, computing, analyzing, storing, and forwarding important data to the cloud center for processing. On the contrary, in top-down data transmission, the cloud center sends the information to the edge node, and the edge node processes the data according to the computing capacity and receiving mode of the terminal devices, and then forwards the data to the corresponding terminal devices.

B. PUF

PUF was first proposed by Pappu et al. [13] in 2002. It refers to the unclonable mapping relationship between input challenges and output responses established by taking advantage of random physical characteristics of devices. PUF is the extraction of small differences in the circuit as the characteristics of electronic devices. It does not require storage and has low hardware costs. It can avoid security issues with traditional keys and is ideal for the security authentication of IoT devices[14].

The input of PUF is called a challenge, which can be denoted by C , and the output is called a response, which can be denoted by R . A challenge and its corresponding response form a challenge-response pair (CRP). PUF can be expressed as $PUF : C \to R$. An ideal PUF has the following properties:

(1) Uniqueness: For the same challenge, if the devices are different, the response is different.

(2) Unclonability: For any probability polynomial time (PPT) adversary $\mathcal{A}$, it is difficult to clone PUF through physical or mathematical methods.

(3) Unpredictability: For any PPT adversary $\mathcal{A}$, it is difficult to predict the response value of PUF.

(4) Stability: In different external conditions, the same devices respond to the same challenge.

III. DEMAND ANALYSIS

In the cloud-edge-end architecture, there are big differences in the quantity, deployment environment, computing power, and storage capacity between the cloud center, edge nodes, and terminal devices. In this section, we analyze the characteristics and communication requirements of the cloud, edge nodes, and terminal devices.

The cloud center is located at the top level of the three-layer architecture and has the following characteristics: (1) Strong computing and storage capabilities, capable of running complex calculations and storing large amounts of node and device data. (2) Fixed and secure location, low probability of physical attack. (3) Once the loss caused by intrusion is large, the intrusion attack on the cloud center will cause a large number of edge nodes and terminal devices to lose value, leading to the failure of the entire system. (4) The communication state with the edge node is relatively stable, the channel reliability is high, and the effective communication time after a successful authentication is long.

Terminal devices are located at the lowest level of the three-layer architecture and have the following characteristics: (1) The number of terminal devices is huge and has diversity and heterogeneity. If all of them are connected to the cloud center, it will lead to excessive communication delay, affecting the real-time performance of the IoT. (2) Limited computing and storage capacity, unable to run complex algorithms, only a small amount of information can be stored. (3) Most of them are deployed in open environments and are vulnerable to physical attacks leading to the disclosure of secret information. (4) The stability of communication with edge nodes is poor, and it is prone to frequent disconnection requiring re-authentication. (5) Divided into different security domains by edge nodes. Terminal devices in different security domains have different communication environments and modes, resulting in a high channel error rate and a high risk of disguised attacks.

Edge nodes are located between the cloud center layer and the terminal device layer, and have the following characteristics: (1) Computing and storage capabilities between the cloud center and terminal devices, and can run complex algorithms, analyze and store data of a certain scale. (2) They are often deployed in an open environment and are vulnerable to physical attacks leading to the disclosure of secret information. (3) The communication with the cloud center is relatively stable, and effective communication can be carried out for a long time after

a successful identity authentication; (4) Poor communication stability with terminal devices and high connection authentication frequency. (5) Edge nodes divide terminal devices into different security domains, and are responsible for authentication and data collection management of terminal devices within the domain.

IV. AUTHENTICATION MODEL

In this section, we distinguish devices at cloud, edge, and end layers, refine internal functional modules, and design the "cloud-edge-end" integrated IoT authentication model, as illustrated in Figure 2.

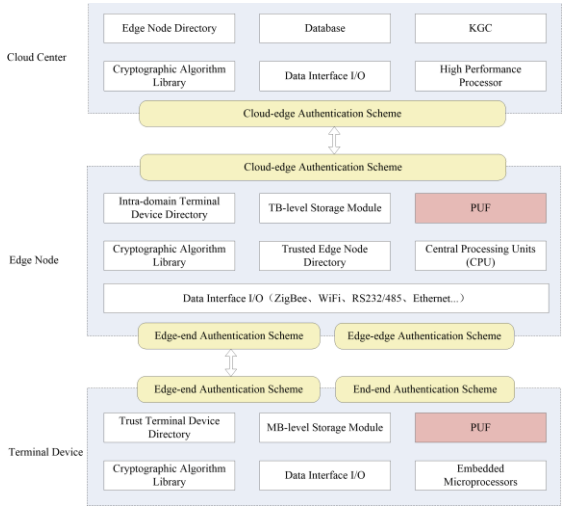

Fig. 2. The cloud-edge-end integrated IoT authentication model.

The authentication protocols in the cloud-edge-end integrated IoT authentication model can be divided into four categories: cloud-edge, edge-edge, edge-end, and end-to-end. The cloud-edge authentication protocol implements authentication between edge nodes and the cloud center. The edge-edge authentication protocol implements bidirectional authentication between edge nodes. The edge-end authentication protocol implements access authentication between terminal devices and edge nodes. And the end-end authentication protocol implements cross-domain authentication between terminal devices in different security domains. The functions of the internal modules at the cloud, edge, and end layers are as follows:

(1) Directory: The directory is used to store information about edge nodes and terminal devices. The Edge Node Directory is used by the cloud center to store the information of edge nodes that have been registered in the cloud center. The Intra-domain Terminal Device Directory is used by the edge node to store the information of terminal devices that have been registered in the edge node. The Trusted Edge Node Directory is used by the edge node to store the information of other edge nodes with which it has established a secure communication relationship. The Trust Terminal Device Directory is where the terminal device stores information about other terminal devices with which it has established a secure communication relationship.

(2) Database and Storage Modules: Database and storage modules are used to store other important information required for authentication. Storage capacity decreases in the order of cloud center, edge nodes, and terminal devices. In the cloud center is the database with the most powerful storage capacity. TB-level storage modules are in edge nodes, and MB-level storage modules are in terminal devices.

(3) Key Generator Center (KGC)： KGC is used to generate the partial key for authentication between edge nodes and the cloud center and is deployed in the cloud center.

(4) Cryptographic Algorithm Library: Cryptographic algorithm library provides cryptographic algorithms needed for authentication, such as hash functions, encryption/decryption algorithms, random number generation functions, etc.

(5) Data Interface I/O: Data interface I/O is used for data transmission between the cloud center, edge nodes, and terminal devices. Edge nodes have more kinds of I/O interfaces to communicate with massive heterogeneous terminal devices.

(6) Processor: The processor is used to analyze and process the received data. Computing power decreases in the order of cloud center, edge nodes, and terminal devices. High performance processor is located in the cloud center, central processing units (CPUs) are located in edge nodes, and embedded microprocessors are located in terminal devices

(7) PUF Module: PUF module is implemented in hardware form, which is used to extract the "hardware fingerprint" of edge nodes and terminal devices. PUF module establishes the mapping relationship between input challenge and output response and provides a hardware basis for the implementation of the authentication scheme.

V. AUTHENTICATION PROCESS

Based on the cloud-edge-end integrated authentication model for the IoT, we further design the authentication process for cloud-edge, edge-edge, edge-end, and end-end in this section. For ease of description, Table 1 lists the notations used in this scenario.

TABLE I. NOTATIONS

Notations	Description
CS	Cloud center
ES	Edge node
D	Terminal device
ID	Identity of ES
PID	Pseudo-identity of D
$Auth$	Authentication message
$PUF_D(\cdot)$	PUF of D
$PUF_{ES}(\cdot)$	PUF of ES
(C, R)	Challenge-response pair of D
(EC, ER)	Challenge-response pair of ES
(X, Y)	Public key pair of ES
(x, y)	Private key pair of ES
M	Auxiliary authentication information, such as pseudo-random numbers, timestamps, etc.

A. Cloud-edge authentication

We use certificateless authentication to deploy trusted third-party KGC as a private key generator in the cloud center. The specific process is shown in Figure 3.

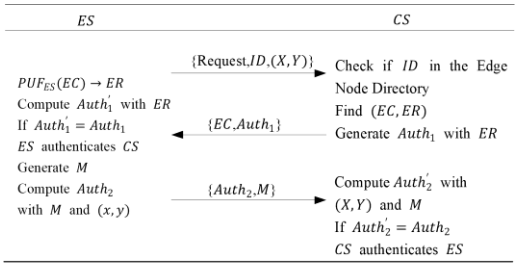

Fig. 3. The cloud-edge authentication process.

Step 1: ES sends authentication request, ID, (X,Y) to the cloud center CS.

Step 2: After receiving the message, CS finds whether ES exists in the Edge Node Directory based on ID. If ES exists, CS finds the corresponding (EC,ER) and computes $Auth_1$ with ER. Then CS sends ER and $Auth_1$ to ES.

Step 3: ES uses PUF module to generate the corresponding response ER from EC and computes $Auth_1^{'}$. Then ES checks whether the equation $Auth_1^{'}=Auth_1$ is true. If so, ES authenticates CS, and generates auxiliary authentication information M, computes $Auth_2$ from (x,y) and M. At last, ES sends M, $Auth_2$ to CS.

Step 4: CS uses (X,Y) and M to compute $Auth_2^{'}$ and checks whether the equation $Auth_2^{'}=Auth_2$ is true. If so, CS authenticates ES, it indicates that the authentication between ES to CS is successful.

B. Edge-edge authentication

Edge-edge authentication is required to ensure reliability when edge nodes work together. Edge nodes have partial keys distributed by KGC, so we use certificateless authentication in edge-edge authentication. The authentication between edge nodes is shown in Figure 4.

Step 1: ES_A generates M_A then sends authentication requests, ID, (X_A,Y_A) and M_A to ES_B.

Step 2: When ES_B receives the message, it uses (X_A,Y_A), (x_B,y_B) and M_A to compute $Auth_1$, and generates auxiliary authentication information M_B. Then ES_B sends M_B and $Auth_1$ to ES_A.

Step 3: ES_A computes $Auth_1^{'}$ and checks whether the equation $Auth_1^{'}=Auth_1$ is true. If so, ES_A authenticates ES_B, and it generates $Auth_2$ from (x_A,y_A), (X_B,Y_B) and M_B. At last, ES_A sends $Auth_2$ to ES_B.

Step 4: ES_B uses (X_A,Y_A), (x_B,y_B) and M_A to compute $Auth_2^{'}$, and checks whether the equation $Auth_2^{'}=Auth_2$ is true. If so, ES_B authenticates ES_A, it indicates that the authentication between ES_A to ES_B is successful. Finally, ES_A adds ES_B in its Trusted Edge Node Directory. Similarly, ES_B adds ES_A to its Trusted Edge Node Directory.

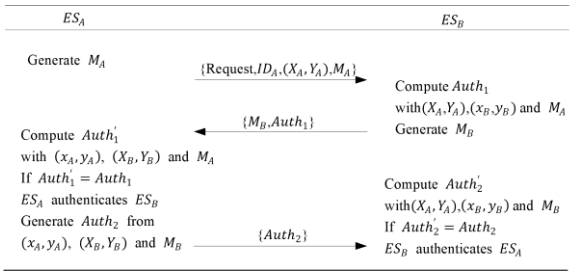

Fig. 4. The edge-edge authentication process.

C. Edge-end authentication

We use lightweight security primitives to design an authentication scheme to achieve lightweight and efficient access authentication for terminal devices to edge nodes. The specific steps are shown in Figure 5.

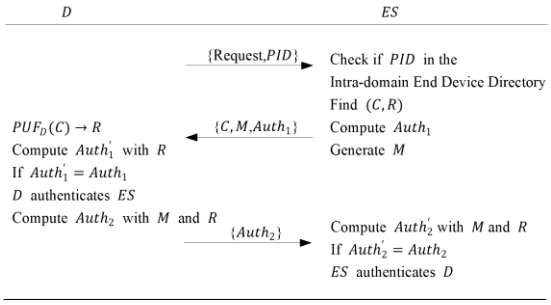

Fig. 5. The edge-end authentication process.

Step 1: D sends authentication request and PID to ES.

Step 2: After receiving the message, ES finds whether D exists in the Intra-domain Terminal Device Directory based on PID. If D exists, ES finds the corresponding CRP (C,R) and generates $Auth_1$ and auxiliary authentication information M. Then ES sends M and $Auth_1$ to D.

Step 3: D uses PUF module to obtain the response R of the challenge C. Then D computes $Auth_1^{'}$ from R, and checks whether the equation $Auth_1^{'}=Auth_1$ is true. If so, D authenticates ES, and it calculates $Auth_2$ according to R and M, then sends $Auth_2$ to ES.

Step 4: ES uses R and M to calculate $Auth_2^{'}$ and checks whether $Auth_2^{'}=Auth_2$ is true. If so, ES authenticates D, it indicates that the access authentication of D to ES is successful.

D. End-end authentication

Referring to authentication among multiple cloud service providers in cloud computing, we design a lightweight anonymous authentication scheme to solve the cross-domain authentication issues between terminal devices in different security domains. The specific process is shown in Figure 6.

Step 1: D_A sends authentication requests to D_B .

Step 2: D_B sends PID_B and ID_N to D_A .

Step 3: D_A sends PID_A , PID_B and ID_N to ES_M .

Step 4: ES_M finds whether D_A exists in the Intra-domain Terminal Device Directory, if not, D_A needs to be authenticated first. After successful authentication, ES_M generates auxiliary authentication information M_A .Then ES_M sends PID_A , PID_B and M_A to ES_N .

Step 5: ES_N finds whether ES_M exists in the Trusted Edge Node Directory, if not, ES_M needs to be authenticated first. Then ES_N finds whether D_B exists in the Intra-domain Terminal Device Directory. If so, ES_N sends PID_A and M_A to D_B .

Step 6: D_B computes $Auth_1$ with M_A , and generates auxiliary authentication information M_B . Then sends $Auth_1$ and M_B to D_A .

Step 7: D_A calculates $Auth_1^{'}$ and checks whether the equation $Auth_1^{'}$=$Auth_1$ is true. If so, D_A authenticates D_B . Then D_A generates $Auth_2$ with M_B , and sends M_B to D_B .

Step 8: D_B calculates $Auth_2^{'}$ and checks whether the equation $Auth_2^{'}$=$Auth_2$ is true. If so, D_B authenticates D_A , it indicates that the mutual authentication of D_A to D_B is successful. Finally, D_A adds D_B in its Trust Terminal Device Directory. Similarly, D_B adds D_A in its Trust Terminal Device Directory.

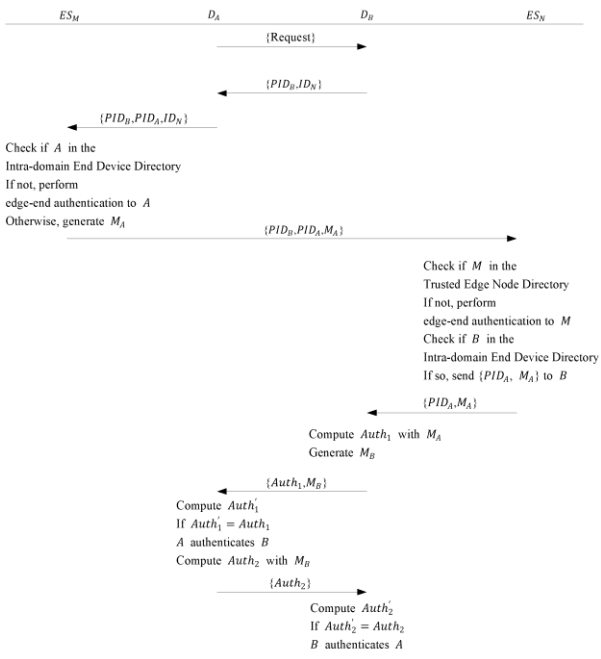

Fig. 6. The end-end authentication process.

VI. Conclusion

Applying edge computing to IoT can effectively reduce bandwidth pressure, shorten communication delay, and enhance service responsiveness, which is helpful to further expand the application field of IoT. Aiming at the identity authentication problem of IoT in edge computing environment, we deeply analyze the characteristics and communication requirements of all levels of devices. On the basis, this paper proposes an identity authentication model based on PUF, and designs the main processes of cloud-edge, edge-edge, edge-end and end-end authentication. It provides ideas for further research on the authentication protocol between devices at all levels of IoT in the edge computing environment.

References

[1] CAICT. Internet of things white paper(2020)[R/OL]. (2020-12) [2023-01-31]. http://www.caict.ac.cn/kxyj/qwfb/bps/202012/P020201215379753410419.pdf.

[2] GSMA. The mobile economy 2020[R/OL]. [2023-01-31].

[3] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges[J]. IEEE internet of things journal, 2016, 3(5): 637-646.

[4] Li X, Chen B, Yang D, et al. Review of Security Protocols in Edge Computing Environments[J]. Journal of Computer Research and Development, 2022,59(04): 765-780.

[5] Chen Q, Li Y, Li X, et al. Research on Application of Cryptography Technology for Edge Computing Environment[J]. Computer Science, 2020, 47(11): 10-18.

[6] Al-Riyami S S, Paterson K G. Certificateless public key cryptography[C]//Asiacrypt. 2003, 2894: 452-473.

[7] Wang Z, Han Z, Liu J, et al. ID Authentication Scheme Based on PTPM and Certificateless Public Key Cryptography in Cloud Environment[J]. Journal of Software, 2016,27(06): 1523-1537

[8] Mei Q, Xiong H, Chen J, et al. Efficient certificateless aggregate signature with conditional privacy preservation in IoV[J]. IEEE Systems Journal, 2020, 15(1): 245-256.

[9] Ibrahim M H. OCTOPUS: An edge-fog mutual authentication scheme[J]. Int. J. Netw. Secur., 2016, 18(6): 1089-1101.

[10] Amor A B, Abid M, Meddeb A. A privacy-preserving authentication scheme in an edge-fog environment[C]//2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2017: 1225-1231.

[11] Kumar V, Kumar R, Jangirala S, et al. An enhanced RFID-based authentication protocol using PUF for vehicular cloud computing[J]. Security and Communication Networks, 2022, 2022.

[12] Nyangaresi V O, Petrovic N. Efficient PUF based authentication protocol for internet of drones[C]//2021 International Telecommunications Conference (ITC-Egypt). IEEE, 2021: 1-4.

[13] Pappu R, Recht B, Taylor J, et al. Physical one-way functions[J]. Science, 2002, 297(5589): 2026-2030.

[14] Mall P, Amin R, Das A K, et al. PUF-based authentication and key agreement protocols for IoT, WSNs, and Smart Grids: a comprehensive survey[J]. IEEE Internet of Things Journal, 2022, 9(11): 8205-8228.