Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 198 (2022) 454–459

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Internet of Smart Things (IST 2021)
November 1-4, 2021, Leuven, Belgium

Physical Unclonable Function Based Identity Management for IoT with Blockchain

Zhixiang Li[a,b,c],Yunxia Chu[a,b],Xuning Liu[a],Yuekui Zhang[a,b,c]*,Jiu Feng[a],Xiaoyang Xiang[a]

[a]*Shijiazhuang University, No. 288 Zhufeng Avenue, High tech Zone, Shijiazhuang 050000, China*
[b]*Hebei Province Key Laboratory of IoT Blockchain Integration,Shijiazhuang 050000,China*
[c]*IoT Security and Sensor Test Engineering Research Center of Hebei,Shijiazhuang 050000,China*

Abstract

As a result of the increasingly pervasive deployment of the Internet of Things(IoT), the cybersecurity of IoT has already attracted more and more research efforts. Identity management is believed to be the fundamental keystone to build security mechanisms. The traditional centralized identity management scheme suffers from a single point of failure and identity forgery. A secure IoT system framework was proposed leveraging blockchain as the basic infrastructure with Physical Unclonable Functions (PUFs) identifying the sensors uniquely. We brought up a scheme to improve the identity authentication protocol. Experiments showed that our approach was more effective and secure against attacks.

Keywords: Identity management; IoT; PUF; Blockchain;

1. Introduction

The Internet of Things with countless sensors and actuators connected has profoundly changed the world. A new study from Juniper Research found that the total number of IoT connections will reach 83 billion by 2024, rising from 35 billion connections in 2020[1]. While this can be viewed as technical advancement, it also raises a broad

* Corresponding author. Tel.:+86-188-3065-0600; fax:+86-311-6661-7198.
E-mail address: 1062908417@qq.com

10.1016/j.procs.2021.12.269

concern about security. For example, the Mirai IoT botnet broke down a large number of Internet sites[2]. The attacker may also use the access vulnerability of the smart door lock to remotely open the door and break into the house[3].

Access control is considered a practicable approach to build secure IoT. Nevertheless, the traditional centralized Internet access control scheme, based on Certificate Authority (CA) or other trusted servers, has the risk of a single point of failure, poor scalability, and resource consuming. On the other hand, identifying the terminal node uniquely is also a basiclly important issue. Traditional product serial number based identity suffers from identity forgery. Identity authentication in IoT has distinctive complexity given limited resources, massive nodes, and dynamic topology. On the identity representation of things, Roel Maes[4] divided the identity of things into assigned and inherent identities. The inherent identity of things, in analogy with the fingerprint of human being, could be derived from PUF(Physical Unclonable Function), an unpredictable irreproducible but response-repeatable module[5].

The prevailing PUF based authentication applications still use star topology and rely on centralized authentication servers to verify identity. This topology suffers from a single point of failure, and lack of scalability. Therefore, the concept of self-sovereign identity is put forward[6]. Blockchain is viewed as a promising technology to build a decentralized self-sovereign identity. However, blockchain gateway, which is the critical component in IoT identity management, is rarely illustrated exhaustively in literature.

On communication protocol between IoT nodes, Chatterjee et.al.[7] designed a PUF based device enrollment, device verification, and secure communication protocol. Braeken et.al.[8] improved it and proposed a new communication protocol between two nodes. However, the main scenario studied in this paper is communication between IoT devices and the blockchain gateway, in which two terminal nodes does not involve simultaneously. In order to alleviate the nodes' computing burden, the verification protocol and secure communication protocol are simplified to fit gateway-node communicating mode without compromising the security.

The main contributions of the paper are as follows:

- We proposed an IoT system framework leveraging blockchain as the fundamental infrastructure with PUF as the node identity management module.
- A blockchain gateway was designed and implemented on Raspberry Pi 4.
- We brought up an improved PUF-based identity enrollment and authentication scheme, reinforced the security level against replay attacks and DoS attacks.

2. System Architecture

The IoT system architecture based on blockchain could be divided into four layers: terminal node layer, gateway aggregation layer, blockchain layer, and application layer. As shown in the Fig. 1.

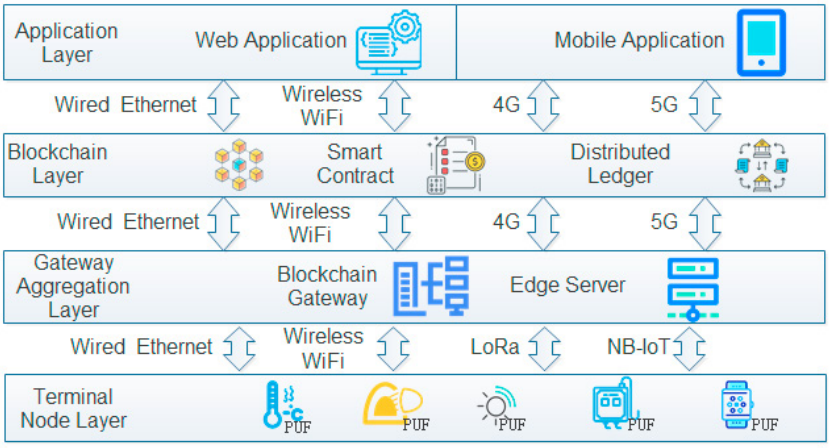

Fig. 1. IoT system architecture based on blockchain and PUF

The terminal node layer contains all kinds of sensor and actuator nodes. The gateway aggregation layer includes different types of blockchain IoT gateways. From the perspective of blockchain, the gateway is a full node, which keeps a complete backup of all block data. The blockchain layer refers to the tamper-proof decentralized computing and storage infrastructure based on the consensus algorithm, which solves the problems of node trust and data distributed storage in the IoT. The application layer includes the web application and the mobile application. We emphasize the core component blockchain gateway and PUF in terminal nodes.

2.1. Blockchain Gateway

Blockchain gateway is responsible for the aggregation of terminal node data in its coverage region and stores the data in the blockchain, of which function modules include node registration, identity recognition, data decryption, data ownership consolidation, and data upload module, as shown in Figure 2.

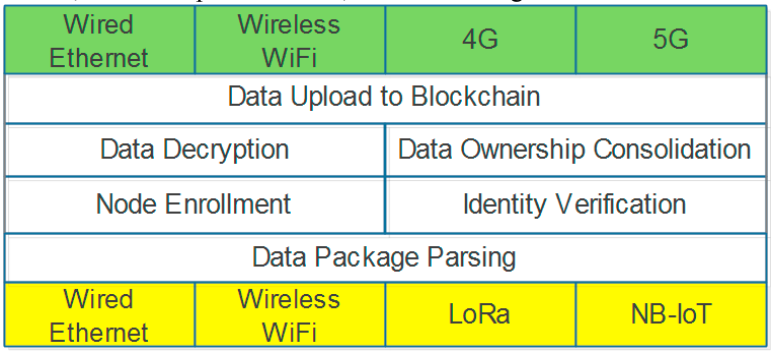

Fig. 2. Modules in Blockchain Gateway

Due to the diversity of IoT terminal nodes, blockchain gateway should be compatible with one or more communication modes such as wired Ethernet, wireless WiFi, LoRa, and NB-IoT.

We implemented the blockchain gateway on Raspberry Pi 4b with LoRa shield and WiFi connection. IoT terminal nodes connect to the gateway through LoRa. The gateway checks the identity of the node, decrypts the message, aggregates multiple messages to form a denser data package, signs the data package with the node's identity, uploads the data package to the blockchain.

2.2. PUF in Terminal Nodes

The irrefutable identity of the IoT terminal nodes can be generated by PUF, but not all of the chips are with PUF. At present, only a few microcontrollers have PUF modules. In order to implement PUF based identity, the following methods are usually adopted:

- During the designing stage of the device, choose a secure microcontroller with PUF function, such as Maxim's Max32520.
- The PUF chip, such as DS28E38, can be considered to be an identity co-processor to enhance the security of the device when designing the circuit.
- For a large number of legacy devices without PUF function, General Purpose Input/Output(GPIO) and Analog Digital Converter(ADC) of Micro Control Unit (MCU) can be used to simulate PUF[9].

3. PUF Based Identity Management Protocol

The protocols discussed include device enrollment protocol, device verification protocol. In order to fit the node-server communication scenario, the verification and key exchange protocol in [8] is tailored and improved by adding

initial encryption to gain more security. The server mentioned in the following protocol description can be regarded as the blockchain gateway in the system framework. Elliptic Curve Cryptography(ECC) with the sextuple parameters (p,a,b,G,n,h) [10] over finite domain F_q is used, where G is the base point. In the proposed protocol we denote H as a hash function, the symbol || as the concatenation of two messages, the symbol $\oplus$ as the exclusive-or operation. The addition or multiplication in the protocol denotes elliptic curve point arithmetic.

3.1. Enrollment

Each node in the system must register and enroll to its responding server under the control of its owner before deploying in the communication network. The enrollment process should be undertaken in a secure and trusted environment. The server randomly chooses $d_s \in F_q$ as its private key, then calculates the public key $P_s=d_sG$. During the enrollment phase, the server collects multiple Challenge-Response Pairs(CRPs) from the device, stores them in the database, and shares the public key P_s to the device. The steps are explained as follows:

(1)Enrollment starts with the node sending an enrollment request.
(2)The server sends P_s to the node. The node stores it. The server generates a random challenge C, then sends it to the node.
(3)After received C, the node derives PUF response R= PUF(C). Then the node sends (SN, C, R) to the server, where SN denotes the node's unique serial number.
(4)The server appends <SN, C, R> to the database.
(5)Repeat steps 2-4 for k times, where k is decided by the system designed depending on the storage space and the demanding security level. If k is set to be a larger number, the CRPs database requires more storage resources, while the system has more candidate CRPs, and it is more difficult for the attackers to track the authentic CRP.

3.2. Verification and Key Generation

(1)To start the verification phase, Node1 sends to the server a communication request containing the unique serial number SN_1 and the current timestamp TS, encrypted with P_s:

$$Req = E\left(SN_1 \,||\, TS, P_s\right) \tag{1}$$

where *E(m,k)* denotes the ECC encryption over the plaintext *m* with the key *k*. In the meantime, Node1 saves SN_1 and TS for later usage.
(2)On the arrival of the request, the server starts a session, firstly decrypts the message with SK:

$$\left(SN_1 \,||\, TS\right) = D(Req, d_s) \tag{2}$$

where *D(c,d)* denotes the ECC decryption over the ciphertext *c* with the private key *d*. Then the server parses SN_1 and TS out of the message, searches for items with the key equals SN_1 in the CRPs database. If the result is not null, it then searches TS with SN_1. The existence of TS indicates that the same timestamp has been used in an earlier session. So the request should be regarded as a replay attack and rejected. If not exist, it randomly chooses two CRPs <C_1, R_1>,<C_2, R_2> of Node1. After that, it computes

$$C_2^{'} = C_2 \oplus H\left(R_1 \,||\, TS \,||\, SN_1\right) \tag{3}$$

In addition, a hash value needs to be appended to ensure the integrity of the message. The value for Node1 is computed as follow:

$$h_1 = H\left(C_1 \,||\, C_2 \,||\, TS\right) \tag{4}$$

The information C_1,C_2',h_1 is sent to Node1. The server stores the parameters SN_1, R_1, R_2, TS for later usage.

(3) When the message is received, Node1 first tries to parse the message for three parameters C_1, C_2', h_1 with expected size. If so, the parameters are stored. Then, Node1 computes R_1 for C_1 with PUF embedded in the node device. C_2 can be computed:

$$C_2 = C_2^{'} \oplus H\left(R_1 \,||\, TS \,||\, SN_1\right) \tag{5}$$

With C_2 and C_1, h_1 can be verified. If $H(C_1||C_2||TS)=h_1$ is false, it can be inferred that the message may be counterfeit from an attacker. If true, Node1 derives R_2 and randomly selects $q_1 \in F_q$ to compute $Q_1=q_1G$. Next it computes $h_2=H(R_1||R_2||TS||Q_1)$. The message(h_2, Q_1) is sent to the server, and R_1, R_2 are additionally stored in the session.

(4) Once the message consisting of (h_2, Q_1) of the expected length is received, the server immediately verifies h_2 for the Nodes1 identity. If failed, the node seems to be an illegal participant. So the communication is interrupted. Otherwise, the server stores Q_1 and uses it to calculate the communication encryption cipher K:

$$K = d_s \times Q_1 = (x_k, y_k) \tag{6}$$

where K is the shared key with the node, x_k or y_k could be used to encrypt the communication.

(5) On the node side, the shared key can also be calculated by:

$$K = d_1 \times Q_s = (x_k, y_k) \tag{7}$$

Henceforth, the messages between the node and the server are encrypted with the key x_k or y_k. The exclusive-or operation is adopted as the encryption to simplify the calculation.

4. Experiments and Results

An identity management system was implemented according to the above research. We used Hyperledger Fabric to construct a blockchain network. The blockchain genesis block and initial network with four nodes were generated on Cloud Server with 2 CPU cores, 8GB RAM,40GB high-performance cloud disk, and 5Mbps network bandwidth. The blockchain gateways built on Raspberry Pi 4b with 8GB RAM and 32GB TF-card, which merged into the blockchain through WiFi with the help of the inside Fabric Client module. We designed a PUF-enabled terminal node with DS28E38 as the PUF authenticator and STM32L5RBT6 as the processor(Node Type 1). We also had several legacy IoT nodes without PUF in them, in which we used GPIO and ADC to simulate PUF(Node Type 2). We used the elliptic curve secp256k1[11] for ECC.

4.1. Performance

We implemented two versions of identity enrollment and verification protocol on the terminal nodes and blockchain gateway, respectively, according to [8] and this paper. We tested the average enrollment time and verification time for the two types of node and server, as shown in Table 1.

Table 1. Performance comparision.

Characteristics	Enrollment Phase		Verification Phase		Improvement
	[8]	This	[8]	This	relative to[8]
Average Time for Node Type 1	0.954μs	1.306μs	12.768 μs	10.433 μs	18.3%
Average Time for Node Type 2	1.296μs	1.477μs	14.230 μs	12.309 μs	13.5%
Average Time for Server	2.164 μs	3.548 μs	25.404 μs	22.837 μs	10.1%

We found that our approach is slightly less effective than [8] in the enrollment phase as a result of initial key storage and longer message. While in the verification phase, our approach improved up to 10.1% relative to [8].

4.2. Security Evaluation

Resistance against session-hijacking attacks. The node chooses different keys in different session. The communication encrypted with one-time-cipher is secure, since the attacker can neither derive the cipher from the message nor decrypt it.

Resistance against identity-forgery attacks. The identity of the node is implemented with PUF the hardware module. There are not two devices with the same PUF. So the malicious node can not generate legal CRPs to pass the identity verification. Also, the series number(SN) of the node device is a handle to choose CRPs from the database, the encryption avoid the risk of disclosure of serial number plaintext.

Resistance against replay attacks. In the identity verification phase, timestamp (TS) is stored at the beginning of a session. If the timestamp TS exists in the database, the connecting attempt is rejected immediately; thus the replay attack is stopped.

Resistance against DoS attacks. In the verification phase, the server firstly checks the existence of SN in database. When a request flood comes, the server will discover illegal requests with the affordable cost.

5. Conclusions

In this paper, we have presented an IoT system framework based on blockchain as the computing infrastructure, using PUF as the node identity module. The core component blockchain gateway has been designed. We have improved the enrollment and verification protocol by adding initial encryption. Finally, we have evaluated that the proposed approach is more effective in the verification phase and secure against session-hijacking attacks, identity-forgery attacks, replay attacks and DoS attacks.

Acknowledgements

This work is funded by Scientific Research and Development Program of Shijiazhuang(Grant 211130221A), and Scientific Research Platform Construction Program of Hebei (Grant SG20182058, SZX2020033).

References

[1]*Juniper Research, IoT Connections to Reach 83 Billion by 2024, Driven by Maturing Industrial Use Cases*. 31st March 2020 [cited 2021 14th July]; Available from: https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024.

[2]Kolias, Constantinos, et al., *DDoS in the IoT: Mirai and other botnets.* Computer, 2017. **50**(7): p. 80-84.

[3]Pavelić, Marko, et al. *Internet of things cyber security: Smart door lock system*. in *2018 international conference on smart systems and technologies (SST)*. 2018. IEEE.

[4]Maes, Roel, *PUF-Based Entity Identification and Authentication*, in *Physically Unclonable Functions: Constructions, Properties and Applications*, R. Maes, Editor. 2013, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 117-141.

[5]Pappu, Ravikanth, et al., *Physical one-way functions.* Science, 2002. **297**(5589): p. 2026-2030.

[6]Tobin, Andrew and Drummond Reed, *The inevitable rise of self-sovereign identity.* The Sovrin Foundation, 2016. **29**(2016).

[7]Chatterjee, Urbi, Rajat Subhra Chakraborty, and Debdeep Mukhopadhyay, *A PUF-Based Secure Communication Protocol for IoT.* ACM Trans. Embed. Comput. Syst., 2017. **16**(3): p. Article 67.

[8]Braeken, An, *PUF Based Authentication Protocol for IoT.* Symmetry, 2018. **10**(8).

[9]Vaidya, G., T. V. Prabhakar, and L. Manjunath. *GPIO PUF For IoT Devices*. in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 2020.

[10]SECG. *SEC 1: Elliptic Curve Cryptography, Version 2.0*. 2014 [cited 2021 21 Aug]; Available from: https://www.secg.org/sec2-v2.pdf.

[11]Cook, John D. *Bitcoin key mechanism and elliptic curves over finite fields*. 2018 14 August [cited 2021 21 Aug]; Available from: https://www.johndcook.com/blog/2018/08/14/bitcoin-elliptic-curves/.