PUF based IoT Device Authentication Scheme

Byoungkoo Kim, Seoungyong Yoon, Yousung Kang and Dooho Choi
Cryptographic Engineering Research Section, Cyber Security Research Division
Electronics and Telecommunications Research Institute (ETRI)
Daejeon, Korea
{bkkim05, syyoon, youskang, dhchoi}@etri.re.kr

***Abstract*—This paper propose a PUF(Physical Unclonable Function)-based IoT(Internet of Things) device authentication scheme to minimize the risk of authentication key exposure and the load of authentication server. The proposed technique can be realized by storing and updating only a single CRP(Challenge-Response Pair) through interaction with the device to be authenticated, unlike the existing technique of storing all the CRPs that can be generated by the PUF technology in the authentication server. In addition, it enables more secure device authentication through authentication message encryption by using the secret key generated from the CRP.**

Keywords*—*IoT Device, Device Authentication, PUF, CRP

I. Introduction

Currently, most IoT devices are operated with software-based security technologies. However, even if various types of security technologies are applied, many cases of damages caused by system hacking are being reported. In addition, most of such security software stores the secret key for authentication and encryption in the memory. Therefore, there is a high possibility that the leakage of the authentication key causes a great damage. Indeed, economic and industrial losses from illegal copying and counterfeiting due to key leaks are increasing day by day.

To solve this problem, a new technology called PUF emerged. PUF is a technology that allows each device to have unique characteristics, such as human fingerprint or biometric information, such as iris, that allows a device to have different characteristics even if it is made with the same process[1]. Above all, if the PUF technology is used for authentication, a unique identifier for identifying each device can be generated by itself without being injected from the outside. Thus, if you implement a non-replicable PUF using a variety of methods, you do not need to keep the keys created through that PUF separately. That is, the IoT device authentication technique using the PUF can provide a more secure device authentication technique.

This paper proposes an IoT device authentication scheme that is secure against security threats due to authentication key exposure. This paper is organized as follows: Section 2 presents a brief description of the existing PUF based device authentication techniques. Section 3 describes the proposed IoT device authentication scheme based on PUF technology and explains how the proposed scheme authenticates the device. Section 4 presents the prototype device to which the proposed scheme will be applied. Lastly, Section 5 presents the conclusion and future plan.

II. Background

PUF refers to a system that generates random digital values that are difficult to predict and implemented inside a chip, using process variations occurring in a semiconductor manufacturing process[2]. In other words, it is not possible to reproduce because a hardware-unpredictable value is output. Also, since it is generated in the chip, it can be generated by itself without injection from the outside, which can fundamentally block the leakage to the outside of the chip[3]. Therefore, the value can be appropriately utilized for device authentication and message encryption. In other words, when the digital value is used for device authentication, it is possible to have a unique identifier even if it is a device produced by the same process. In addition, since the device does not need to store its own identifier, it can be free from security threats due to authentication key exposure.

In the above, the digital value of the PUF value for the device authentication is used as a CRP composed of the input and output values of the PUF circuit, which is applied in various forms for authentication of each device. Here, the CRP for device authentication is generally extracted in the device manufacturing process and stored in advance in the database of the authentication server, and authentication for the device is performed by comparing the PUF value generated at the device with each authentication request.

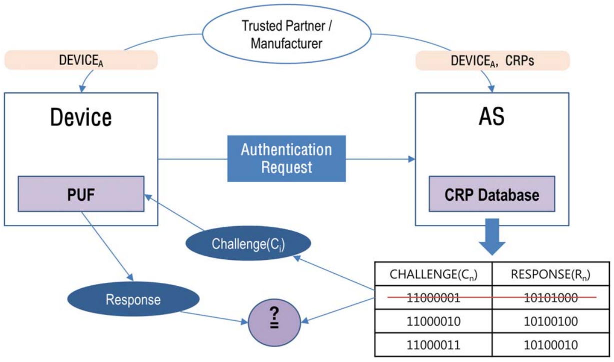

Fig. 1. Existing PUF-based device authentication

Fig. 1 briefly shows a general procedure of device authentication using the above PUF technique. As shown in the figure, the authentication server(AS) stores and manages CRPs for each device in a CRP database. Upon receipt in the authentication request of the device *A*, a randomly selected challenge value(C_i) is returned from the *n* CRP values belonging to the device *A*. The device generates a response value for the received challenge value through the PUF, and responds to the authentication server. The authentication server authenticates the device by matching the received PUF value with the response value in the CRP database corresponding to the challenge value transmitted to the device. Here, the CRP once used is deleted in order to prevent a sniffing or a reuse attack.

However, such a PUF-based authentication scheme must store and manage a fairly large amount of CRPs per device registered in the authentication server, which will increase proportionally to the devices registered in the authentication server. In addition, there are various security threats that may occur when the CRPs recorded in the authentication server are exposed through various paths. Accordingly, there is a need for an effective key management technique not only to reduce the burden on the authentication server but also to minimize the threat of exposure of the authentication key. That is, an IoT device authentication technique that can more effectively utilize the digital value of the PUF is required.

III. Proposed IoT Device Authentication Scheme

A. *Concept for IoT Device Authentication*

This paper presents our approach for IoT device authentication based on PUF. The proposed method provides a more effective device authentication scheme to prevent the load problem of the authentication server and the risk of exposing the authentication key with the existing PUF-based authentication technology. This is possible by storing and updating only a single CRP through interaction between the authentication server and the device, unlike the existing technique. In addition, since the secret key generated based on the CRP is used for message encryption, more secure device authentication can be assured.

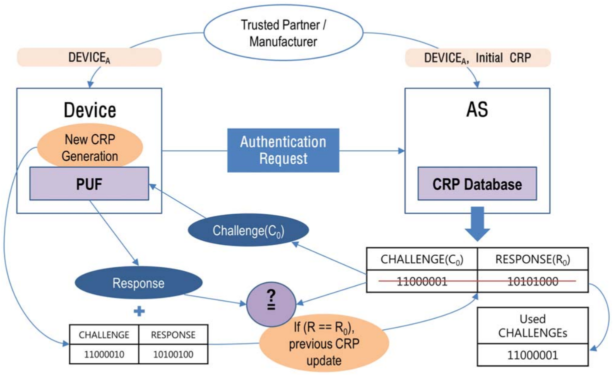

Fig. 2. Overall concept of the proposed method

Fig. 2 shows the overall concept of the proposed method. As shown in the figure, the authentication server registers only one initial CRP for each device in the CRP database in advance. Then, according to the authentication request of the device *A*, the authentication server responds to only the challenge value among the initial CRP belonging to the device. The device responds by generating a response value for the received challenge value through the PUF. Also, the device generates and delivers an new CRP to be used for subsequent authentication. Here, the authentication server performs authentication on the device by checking whether or not the received response value matches the response value(R_0) stored in the server. If the authentication for the device is successful, the initial CRP is updated with the new CRP received from the device. Finally, the previously used challenge value(C_0) is managed as a used challenge list in order to prevent a sniffing or a reuse attack.

B. *Detaild Description of the Proposed Method*

As described above, the proposed method has a method of storing and managing only one CRP for each device in the authentication server. Accordingly, it is possible not only to minimize the resources of the authentication server, but also to minimize the security threat due to exposure of the authentication key.

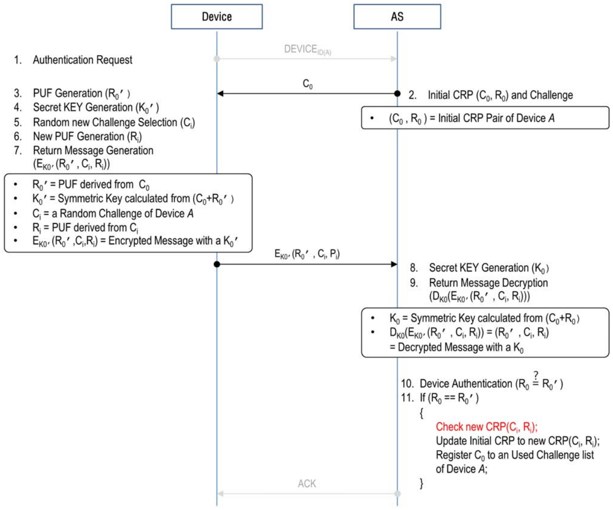
Device
AS
DEVICE$_{ID(A)}$
1. Authentication Request
C_0
2. Initial CRP (C_0, R_0) and Challenge
• (C_0 , R_0) = Initial CRP Pair of Device *A*
3. PUF Generation (R_0')
4. Secret KEY Generation (K_0')
5. Random new Challenge Selection (C_i)
6. New PUF Generation (R_i)
7. Return Message Generation ($E_{K0'}$ (R_0' , C_i, R_i))
• R_0' = PUF derived from C_0
• K_0' = Symmetric Key calculated from (C_0+R_0')
• C_i = a Random Challenge of Device *A*
• R_i = PUF derived from C_i
• $E_{K0'}$ (R_0' ,C_i,R_i) = Encrypted Message with a K_0'
$E_{K0'}$ (R_0' , C_i, P_i)
8. Secret KEY Generation (K_0)
9. Return Message Decryption (D_{K0}($E_{K0'}$ (R_0' , C_i, R_i)))
• K_0 = Symmetric Key calculated from (C_0+R_0)
• D_{K0}($E_{K0'}$ (R_0' , C_i, R_i)) = (R_0' , C_i, R_i) = Decrypted Message with a K_0
10. Device Authentication ($R_0 \overset{?}{=} R_0'$)
11. If (R_0 == R_0')
{
Check new CRP(C_i, R_i);
Update Initial CRP to new CRP(C_i, R_i);
Register C_0 to an Used Challenge list of Device *A*;
}
ACK

Fig. 3. Detailed authentication flow

Fig. 3 shows a detailed authentication flow including the authentication message encryption in the proposed method. As shown in the figure, the authentication server and the device to be authenticated are performed in the order of numbers with one initial CRP. The contents of each number are as follows.

1) Authentication Request: Device *A* requests its authentication server to authenticate itself with its own device identifier(DEVICE$_{ID(A)}$).

2) Initial Challenge: The authentication server returns only the challenge value(C_0) among the initial CRP(C_0, R_0) for authentication of device *A*.

3) PUF Generation: Device *A* generates a PUF value(R_0') derived from the received challenge value(C_0).

4) Secret Key Generation: The device *A* generates a secret key(K_0') for message encryption using the received challenge value(C_0) and the PUF value(R_0') generated in step 3. Here, the secret key is basically a symmetric key.

5) New Challenge Selection: The device *A* selects a random challenge value(C_i) to be used for the subsequent authentication.

6) New PUF Generation: Device *A* completes new CRP(C_i, R_i) to be used by the authentication server at the subsequent authentication by generating a PUF value(R_i) that is derived from the selected challenge value(C_i).

7) Return Message Generation: The device *A* encrypts the message composed of the PUF value(R_0') generated in step 3 and the CRP(C_i, R_i) generated in step 6 with the secret key(K_0') generated in step 4, and responds the message to the authentication server. Here, the encryption uses a symmetric key algorithm such as DES or AES.

8) Secret Key Generation: The authentication server receives the encrypted message from the device *A* and generates a secret key(K_0) for decrypting the message. Here, the secret key is generated using the pre-registered CRP (C_0, R_0) in the same manner as in step 4.

9) Return Message Decryption: The authentication server decrypts the encrypted message using the secret key generated in step 8, and extracts the PUF value(R_0')

generated in the device *A* and the CRP(C_i, R_i) to be used in the suqsequent authentication.

10) Device Authentication: The authentication server performs authentication for the device *A* by comparing the RUF value(R_0') obtained in step 9 with the pre-registered PUF value(R_0).

11) New CRP Update: If the device authentication is successful, it is checked whether or not the CRP(C_i, R_i) received from the device *A* is a new CRP that is not used in the authentication server through the used challenge list. As a result of the checking, if it is determined as a new CRP, the CRP(C_0, R_0) used for the current authentication is updated to the received CRP(C_i, R_i). Then, the challenge value of the already used CRP(C_0, R_0) is registered in the used challenge list for preventing reuse. Alternatively, if it is determined that the CRP has been used previously, the CRP is replaced with a new CRP after performing additional operations.

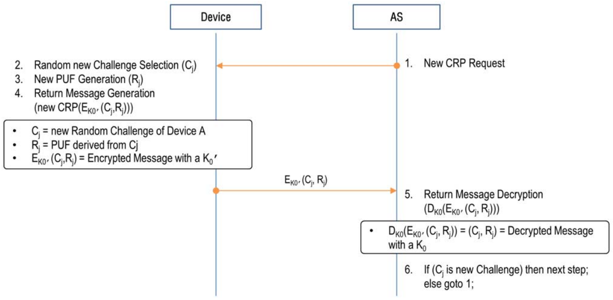
Device
AS
1. New CRP Request
2. Random new Challenge Selection (C_j)
3. New PUF Generation (R_j)
4. Return Message Generation (new CRP($E_{K0'}$(C_j,R_j)))
• C_j = new Random Challenge of Device A
• R_j = PUF derived from Cj
• $E_{K0'}$(C_j,R_j) = Encrypted Message with a K_0'
$E_{K0'}$(C_j, R_j)
5. Return Message Decryption (D_{K0}($E_{K0'}$(C_j, R_j)))
• D_{K0}($E_{K0'}$(C_j, R_j)) = (C_j, R_j) = Decrypted Message with a K_0
6. If (C_j is new Challenge) then next step; else goto 1;

Fig. 4. Additional flow for updating the CRP

Fig. 4 shows an additional flow for updating the CRP. This indicates that the challenge value of the CRP(C_i, R_i) received from the device *A* exists in the used challenge list, and the contents of each number are as follows.

1) New CRP Request: The authentication server requests a new CRP to be used for subsequent authentication.

2) New Challenge Selection: The device *A* newly selects a random challenge value(C_j).

3) New PUF Generation: Device *A* completes new CRP(C_j, R_j) by generating a PUF value(R_j) that is derived from the selected challenge value(C_j).

4) Return Message Generation: The device *A* encrypts the message composed of the new CRP(C_j, R_j) generated in step 3 with the secret key(K_0') generated in step 4 of Fig. 3 and responds it to the authentication server.

5) Return Message Decryption: The authentication server decrypts the encrypted message from the device *A* with the secret key(K_0) generated in step 8 of Fig. 3.

6) Challenge Availability Check: If the challenge value(C_j) of the CRP obtained through the decryption is a new value not belonging to the used challenge list, the authentication server selects the CRP as new CRP. Otherwise, the authentication server repeats the above steps.

As described above, since the proposed scheme performs device authentication by storing and managing only a single CRP, the execution load of the authentication server can be minimized. Above all, this technique can prevent security threats to key exposures by generating and using a new authentication key through the PUF whenever there is an authentication request. That is, the proposed method provides more effective and secure device authentication through effective use of CRPs.

IV. Implementation

The proposed IoT device authentication method was developed to solve the problems of existing PUF-based device authentication schemes. To verify the proposed mechanism, we implemented a prototype device.

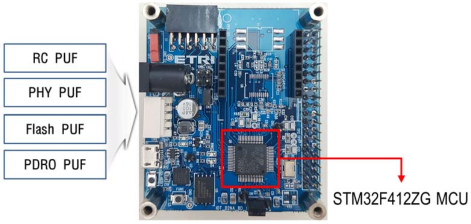

Fig. 5. Our prototype device

Fig. 5 shows the prototype device, which is an IoT device developed for installation and operation of the proposed mechanism. The device has a Cotex-M4 based STM32F4 MCU and provides various PUF technologies to support the proposed mechanism.

V. Conclustions

In this paper, we propose a PUF-based IoT device authentication scheme that minimizes the load on the authentication server and prevents the security threat by exposing the authentication key. The method stores and updates only a single CRP through the interaction between the authentication server and the device at each authentication request. In addition, it can perform more secure device authentication by encrypting the authentication message based on the CRP using the PUF. In the future, we plan to test the stability of the proposed method by actually installing and operating it in our prototype device.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evalution(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00230, Development on Autonomous Trust Enhancement Technology of IoT Device and Study on Adaptive IoT Security Open Architecture based on Global Standardization [TrusThingz Project])

References

[1] John Ross Wallrabenstein, “Practical and Secure IoT Device Authentication using Physical Unclonable Functions,” in Proc. of IEEE 4th Conference on FiCloud, pp.99-106, August 22-24, 2016.

[2] Yildiran Yilmaz, Steve R. Gunn and Basel Halak, “Lightweight PUF-Based Authentication Protocol for IoT Device,” in Proc. of IEEE 3rd IVSW, pp. 38-43, July 2-4, 2018.

[3] Prosanta Gope and Biplab Sikdar, “Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices,” IEEE IoTs Journal, Vol. 6, No. 1, pp. 580-589, February 2019.