Regular paper

Journal of the KIECS. pp. 671-678, vol. 12, no. 4, Aug. 31 2017, t. 84, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2017.12.4.671

IoT 환경에서의 안전한 키 업데이트를 위한 하드웨어 연동 보안 시스템

잠시드 사이드오브* · 김봉근* · 이종협** · 이 광***

Hardware Interlocking Security System with Secure Key Update Mechanisms In IoT Environments

Jamshid Saidov* · Bong-Keun Kim* · Jong-Hyup Lee** · Gwang Lee***

요 약

최근 사물인터넷(IoT)의 발전에 따라 IoT장비가 실생활에 적극적으로 사용되고 있다. 하지만 IoT장비의 사용이 늘어남에 따라서, IoT 보안 사고에 의한 사생활 침해의 문제 또한 늘어나고 있다. 키 관리는 보안 서비스에서 기본적이면서도 중요한 작업이다. 보안성 강화를 위해 인증 과정에서 동일한 키의 재사용은 제한되어야 하지만 다양한 키들을 기억하며 수동으로 업데이트하는 일은 어려운 일이다. 본 논문에서는 자동화된 키 관리 하드웨어 보안 모듈인 HSM을 제안한다. 제안하는 HSM은 IoT장치에 부착하여 장비와 직접 통신하며, 사용자의 개입 없이 안전하고 자동화된 키관리 과정을 제공한다. 제안된 기법을 통해서 제공되는 키는 인터넷 서비스에서의 사용자와 기기의 인증에 사용될 수 있다.

ABSTRACT

Recent advances in Internet of Things (IoT) encourage us to use IoT devices in daily living areas. However, as IoT devices are being ubiquitously used, concerns onsecurity and privacy of IoT devices are getting grown. Key management is an important and fundamental task to provide security services. For better security, we should restrict reusing a same key in sequential authentication sessions, but it is difficult to manually update and memorize keys. In this paper, we propose a hardware security module(HSM) for automated key management in IoT devices. Our HSM is attached to an IoT device and communicates with the device. It provides an automated, secure key update process without any user intervention. The secure keys provided by our HSM can be used in the user and device authentications for any internet services.

키워드

Authentication, IoT, Hardware Security Module, Secure Keys
인증, 사물 인터넷, 하드웨어 보안 모듈, 보안 키

* 한국교통대학교 소프트웨어학과 (jamshidtuit@gmail.com, bkkim@ut.ac.kr)
** 가천대학교 금융수학과(jonghyup@gachon.ac.kr)
*** 교신저자 : 한국교통대학교 소프트웨어학과
· 접 수 일 : 2017. 07. 10
· 수정완료일 : 2017. 07. 13
· 게재확정일 : 2017. 08. 01
· Received : July 10, 2017, Revised : July 13, 2017, Accepted : Aug 01, 2017
· Corresponding Author : Gwang Lee
Dept. of Software, Korea National University of Transportation,
Email : gwang@ut.ac.kr

Ⅰ. INTRODUCTION

In the era of Internet of Things (IoT), we bring internet-connected computing devices into our life. However, the more use the IoT devices in our life, the more possibility of exposed to cyber-attacks.

To ensure the security of IoT devices, essential security mechanisms should be well implemented. In this sense, encryption and authentication are representative primitives of security protocols and the security of them are heavily dependent on the security of keys. However, managing keys is still a manual job, which are influenced by human factors. For the purpose, we should devise a unique key of our own and memorize it to reuse in later sessions. Recently, password management software, such as Last Pass and One Password, helps users to automatically generate a new password for each website, but the software itself is being subject of attacks and it does not provide secure updating mechanisms.

We can sum up the requirements for practical authentications for IoT devices. First, the password should be leaked from software attacks. Second, the passwords should be dynamically generated and updated to prevent unintended exposing. Third, the whole process should be easily to use. In this paper, we propose WhiteKey, a Hardware security Module (HSM) for secure key management. In addition, the main purpose to share three steps.

1. Mutual authentication step between device and user.
2. Biometric authentication step.
3. Key update step using a hash chain with authentication.

Since WhiteKey generates a key based on the user input, an unauthorized user cannot generate a legitimate key. The HSM of WhiteKey provides an encrypted secret to IoT devices after it is authenticated in advance. Thus WhiteKey can provide a seamless authentication process on user and device at the same time.

II. BACKGROUNDS

2.1 Biometric Authentication

Biometrics becomes a practical alternative to traditional identification methods in many application areas[1]. The device fingerprint, which is equipped with all modern smart phones, can facilitate the solution of this problem. Since two or more people may not have the same biometric data. Biometric systems offer several advantages over traditional authentication methods, namely, 1) biometric information cannot be acquired by direct covert observation, 2) It is impossible to share and difficult to reproduce, 3) It enhances user convenience by alleviating the need to memorize long and random passwords, 4) It protects against repudiation by the user. In addition, biometrics provides the same level of security to all users unlike passwords and is highly resistant to brute force attacks[2].

2.2 Hash Chain for Authentication

The use of one-way chains in authentication was initially proposed by Lamport in order to achieve entity authentication in a one-time password scheme[3]. One implementation of such a scheme is in the S-Key system by Haller[4], other one-time password schemes with distant relation to the proposal of Lamport can be found in [5-6].

A one-way chain $(X_0 \cdots X_n)$ is a set of values such as each value X_i (suppose the last value X_n) is a one-way function of the next value X_{i+1}. Particularly, as a result, we have that $X_i = H(X_{i+1})$ where $0 \le i < n$. H is one way cryptographic hash function. Therefore, the structure is also often called hash chain. One-way hash chain, the generator randomly selects the root or seed of the chain[7]. The value X_n, and drives

all previous values X_1 by iteratively applying the hash function H as indicated above. The value X_0, which we refer to as the end-value, is generally made public, and theoretically related to the identity of the user possessing the corresponding root value. In particularly, Figure 1 shows standard hash chain.

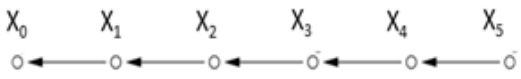


Fig. 1 One-way hash function

Traditional one-way chains have many advantages[8]. In particularly, given only a trusted value X_i of the chain, it is intractable to and a value X_j where $j > i$, such that $H^{j-i}(X_j) = X_i$. Assuming that H is a secure one-way function and that the output of H is quite large. However, it is easily assessing the validity of the value X_j where $j > i$, by verifying that $H^{j-i}(X_j) = X_i$, provided that H provide resistance to weak collisions.

III. THE PROPOSED APPROACH FOR CRYPTOGRAPHIC KEY UPDATING AND MULTIDIMENSIONAL AUTHENTICATION

3.1 Project Overview

The security module is connected to the other sensors, and it receives from them the necessary information. Notably, after connecting to Smartphone, WhiteKey device starts to work taking into account the circumstances surrounding and place of the use. Various sensors send information through the device hub to the gateway. The gateway receives the information and divides it into two sides, private and public information. Private information remains into a gateway, processed and informs the user. All pieces of information should be encrypted before storing database. There are many ways to encrypt data and information, such as random hash lock protocol (hash function), hash chain protocol, extract key from an infinite channel, Encrypted identifier and so on [9 - 10]. The network architecture is to integrate the needs of IoT applications and offer seamless integration. In this case, the IoT and device communication may require for fast networks with high-capacity. Also, the popular binary protocols such as MQTT will also be integrated to enable remote interaction with devices with low resources. Fig. 2 shows the connection structure.

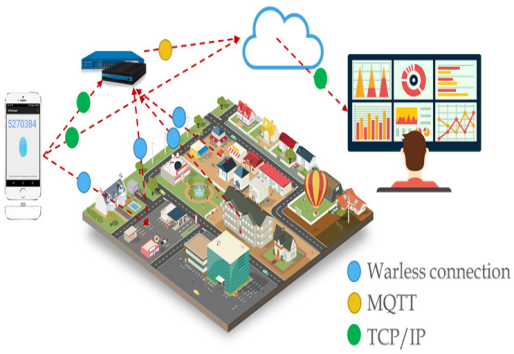


Fig. 2 Design of network configuration

According to the principle of operation of this design can be said that it is widely used in everyday life. For example: in hospitals, pharmacies, shops, offices, homes and public transport. The advantage of this project is that it can use the authentication, synchronization, and randomly changing key methods. Precisely this aspect of paying the most attention to IoT security industry.

3.2 Initialization of Fingerprint Certification

In this section, we propose authentication based on the approach I n biometric data. We take advantage of a biometric identification scheme, encryption, and key authentication. Fingerprint certificate is typically created by using the

following steps:

1. Data is encoded as a byte sequence.

2. The data obtained in the previous step is hashed with a cryptographic hash function, such as MD5.

3. If desired, the output of the hash function should be reduced to provide a shorter, more suitable fingerprint size.

When displaying for user inspection fingerprints tend to hexadecimal encoded string. These strings are then formatted on a group of characters for easy reading. For example, a 128-bit MD5 fingerprint for device is displayed as Fig. 3.

```
73:64:43:a7:b3:fc:5b:b6:0a:6a:a9:b4
:0f:66:73:a7
```

Fig. 3 Example of 128-bit MD5 fingerprint for device

Following the approach, we assume that the sensor is capable of capturing the biometric data of the user, extracting it in the 256 bytes data, and completing the transfer to next cryptographic operations, such as encryption key (bio_id), shows in Figure 4. After that, the device generates a hash value which is equal to AES encryption key.

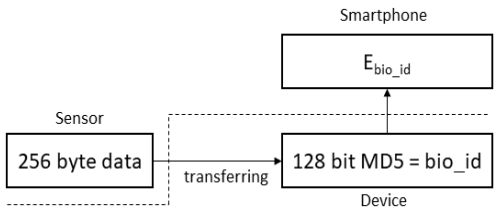


Fig. 4 Biometric data stream

3.3 Key Update and Authentication Mechanisms

This section provides information about the description of our new hardware authentication based on a hash chain, which can be used as authentication, key generation and other section of our projects. In our scheme performance, AU (authentication users) and AD (authentication devices) are employed for authentication and authority.

Something the user knows is the most popular for user authentication in a computer-security system. Quite often, the most systems employ a simple PIN (Personal Identification Number) or password as a data authenticator. Because password-based authenticators typically tend to be software-based, they are prone to various attacks and vulnerabilities from both human and software [11-12]. Hardware-based authentication equipment includes a wide range that includes cryptographic co-processors, randomly key generators, encryption, biometric devices and the trusted computing.

Table 1. Pre-shared information for key agreement phase

WhiteKey	Android	Server
d_id	d_id	d_id
Bio_id	Bio_id	Bio_id
kc	kd	kc
		kd

According to the initialization, Hardware Security Module generates a hash chain and used the elements of the hash chain as his identification[13]. When a function $h^{n-1}()$ is iteratively applied $n-1$ times to an argument k, the result is denoted as $h^{n-1}(k)$. When a process is started, client and server generate a new key using a hash chain which is (K_g) and using as the element of the hash chain; it makes the one-time password. After that, the client generates the (K_g) as a seed for the next chain $X_3 = h^{n-1}(K_g))$, in here X_3 as

X_4 same value of the hash chain, which is we mentioned before. At each hash chain process, the client sends an element of the hash chain together with device identification number to the server. The server authenticates the client by verifying the validity of the element. A prominent feature is that cryptographic key updating and multidimensional authentication. In our scheme, d_did(device id), b_did(biometric id), K_c(key1) and K_d(key2) are key agreements which are for authentication, encryption and other purposes are shown in Table 1.

The symbols in figure 5, will be explained in the following sections, and also we have described an authentication and all process scheme through the scenario.

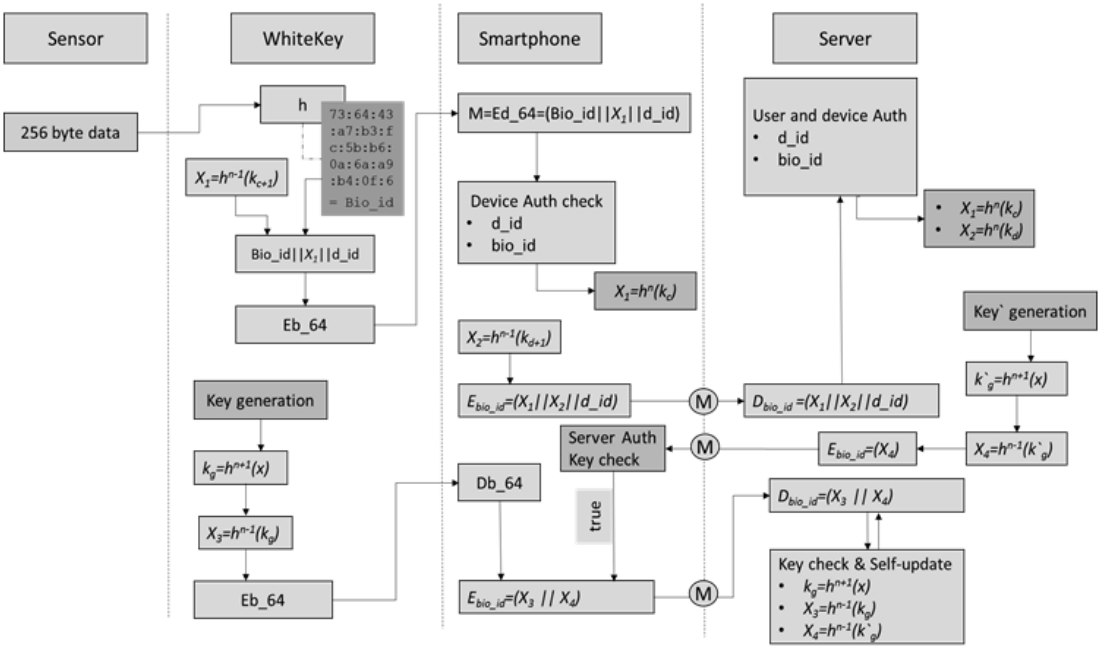


Fig. 5 Key update and authentication algorithm

1) Construction of hash chain traversal with pebbles for key synchronization.

It is a common phenomenon and indisputable fact that the problem of a hash chain is (n) a length. Consequently, $h^n(k)$ one-way hash chain n length, k is key, as a result: $h^{n-1}(k)$. To receive x_1, $(=h^n(k))$ from k, we estimate the hash function $h()$ for $n-1$ times. Particularly, besides for synchronization method we need to use k_{c+1} as pebbles function estimates to a calculate $x_1 = h^{n-1}(k_{c+1})$ by k.

For convenience, we adopt the same citations in Table 2.

Table 2. Notations of Key update and authentication algorithm

Value	Meaning
d_id	Device identification
b_id	Biometric key for AES
kc , kd	Keys (seed).
H	Hash function (e.g. MD-5)
M	Encrypted message or Cipher text
n-1	Length of hash chain or time
X1	Value of hush. Generated by hardware device for authentication
X2	Value of hush. Generated by smart phone for authentication.
X3	Value of hush. Generated by hardware device for checking key and key updating on server side
X4	Value of hush. Generated by server for checking key and key updating on client side
kg	Key generator
E	AES encryption
D	AES decryption
Eb_64	Encode Base 64
Db_64:	Decode Base 64

2) Multidimensional authentication.

Currently, the problem of the non-authentication system is becoming unavoidable. In that case, we have two main authentications, 1) device authentication which is X_1. 2) User authentication which is X_2. Moreover, we must emphasize that, in the same way, we used key authentication. Which are X_3 and X_4 includes key authentication method inside of this value? So it is advised to go multidimensional authentication scheme.

3) Hash-chain based key updating mechanism

K_g is value of hash chain which makes a key for key updating,

$k_g = h^{(n+1)}(x)$ using one-way hash chain is shown in figure 6.

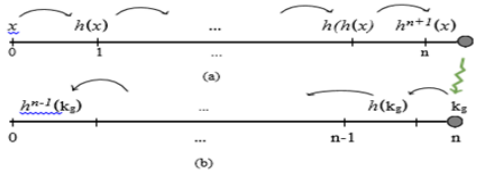


Fig. 6 Key generating and authentication mechanisms using hash chain

For a given X^3, it is the value of hash chain which is responsible for keying authentication or key checking method, $X_3 = h^{n-1}(k_g)$.

4) Encryption.

Bio_id used as a key for AES encryption algorithm that key size is 128 bit. AES encrypt the following important variables such as X_1, X_2, X_3, X_4, d_did between client and server.

IV. IMPLEMENTATION

It is widely acknowledged and known that mobile devices have access to some of your most sensitive personal and private information. Most of users and businesses servant smart phones as a communication device, besides as a means of planning, writing and checking personal accounts their work and private life. Within society, these technologies are causing deep changes in the society of information systems, and they have become the source of new risks consequently.

Our smart phone application is considered for keeping inside and outside information. In order to improve security and privacy of user information, our system does not work without whitekey device. Thus, user will not be able to get access to any personal thing without attaching the device to smart phone. When the device connected to the smart phone, the app will generate a new password to the user needs are shown in figure 7.

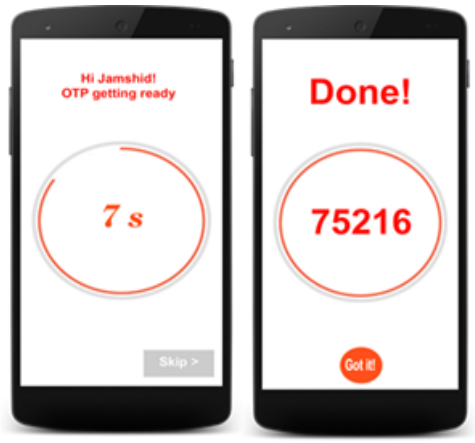


Fig. 7 One-time password generator for personal accounts

In order to connect to a smart phone in this device has a micro-USB and Bluetooth. SP232L microcontroller performs the task of data converter between smart phones and Atmega328P-PU.

The resulting data it encrypts and sends it to a smart phone application. The main processes Atmega328P-PU performs preparation for the device authentication, key generation, and data encryption. Figure 8 shows the WhiteKey device map.

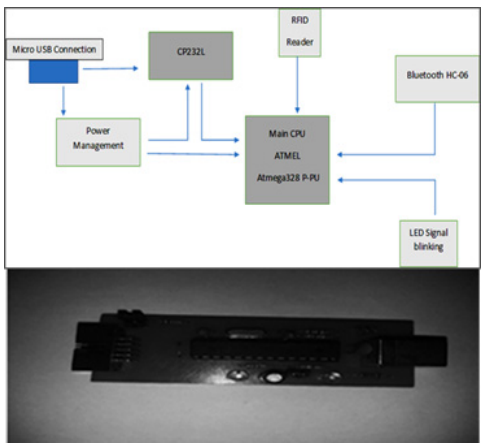


Fig. 8 WhiteKey(HSM) device map

V. CONCLUSIONS

WhiteKey management is the process of key update and authenticating users and devices across IoT network system and controlling their access to the service provided by that system. Thanks to the Belgian cryptographer Kirchhoffs, we followed one of the guidelines which indicated in his security principles. Experimentation was used in this research to investigate the secure key update and authentication scheme without any user intervention. The study has given a strong indication that the data transmit without authentication method vulnerable to the attacks that have been described in the experiment.

This research involves the integration of hardware and software to help users more secure service. We have focused the key update problem in two-ways. Firstly, authentication through biometric identification and device authentication as we mentioned before. Secondly, self-updating key. We have proposed an efficient key generation a mechanism that reach an essentially large key rate than that other HSM devices. As for our future work, will be a useful device for IoT environment.

References

[1] J. Yoo, J. Ko, S. Jung, Y. Chung, K. Kim, K. Moon, and K. Chung, "Design of an Embedded Multimodal Biometric System," *Electronics and Telecommunications Research Institute-Information Security Research Division*, Dec, 2007, pp. 988-992.

[2] S. Yoon and G. Kim, "Personal Biometric Identification based on ECG Features," *J. of the Korea Institute of Communication and Information Sciences*, vol. 10, no. 4, 2015, pp. 521-526.

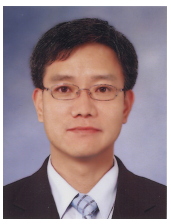
[3] L. Lamport, "Password Authentication with Insecure Communication," *Communication of*

the ACM, vol. 24, no. 11, Nov, 1981, pp. 770-772.
[4] N. Haller, C. Metz, P. Nesser, and M. Straw, "A One-Time Password System," *RFC 2289, Bellcore, Kaman Sciences Corporation, Nesser and Nesser Consulting*, Feb, 1998, pp. 1-25.
[5] C. Mitchell, "Remote user authentication using public information," *9th Institute of Mathematics and Application Conf. on Cryptography and Coding, Lecture Notes in Computer Science 2898*, Berlin, Heidelberg, Dec, 2003, pp. 360-369.
[6] H. Chien and J. Jan, "Robust and Simple Authentication Protocol," *Oxford J., The Computer J.*, vol. 46, no. 2, 2003. pp. 1-9.
[7] J. Soo and K. Park, "An Efficient data management Scheme for Hierarchical Multi-processing using Double Hash Chain," *Digital Fusion Research*, vol. 13, no. 10, 2015, pp. 271-278.
[8] H. Park and J. Seo, "Implementation of Mobile Authentication System for Context-Awareness based on Near Field Communication," *J. of the Korea Institute of Communication and Information Sciences,* vol. 12, no. 1, 2017, pp. 39-45.
[9] G. Montenegro and C. Castelluccia. "Crypto-based identifiers (CBIDs): Concepts and applications," *ACM Trans. Information and System Security,* vol. 7, no. 1, 2004, pp. 97 - 127.
[10] E. Blass, O. Elkhiyaoui, K. Molva, and R. "Tracker: security and privacy for RFID based supply chains," In *Proc. the 18th Network and Distributed System Security Symp.*, San Diego, California, Feb, 2011.
[11] V. Griffith, M. Jokobsson, and Messin, "Deriving mother's maiden names using public records," *Applied Cryptography and Network Security (ACNS), Springer, Heidelberg, vol. Lecture Notes in Computer Science* 3531, 2005, pp. 91 - 103.
[12] D. Klien, "A survey of and improvements to password security," *UNIX Security II: USENIX Workshop Proc.,* Portland, Oregon, Aug, 1990.
[13] W. Jeong and S. Lee, "A Study on the Self-Key Generation Algorithm for Security Elevation in Near Field Communications," *J. of the Korea Institute of Communication and Information Sciences*, vol. 7, no. 5, 2012, pp. 1027-1032.

저자소개

잠시드 시이드오브 (Jamshid Saidov)

2014년 타쉬켄트 TUIT대학 소프트웨어학과 졸업(공학사)
2017년 한국교통대학교 대학원 소프트웨어학과(공학석사)
※ 관심분야 : 임베디드 소프트웨어, 보안



김봉근(Bong-Keun Kim)

1991년 숭실대학교 대학원 전자계산학과 졸업(공학석사)
1997년 숭실대학교 대학원 전자계산학과 졸업(공학박사)
1993~현재 한국교통대학교 소프트웨어학과 교수
※ 관심분야 : 컴퓨터비전, 패턴인식, ITS

이종협(Jong-Hyup Lee)

2004년 연세대학교 대학원 컴퓨터과학과 졸업(공학석사)
2009년 연세대학교 대학원 컴퓨터과학과 졸업(공학박사)
2015~현재: 가천대학교 금융수학과 조교수
※ 관심분야 : 금융보안, 소프트웨어 보안

이 광(Gwang Lee)

1995년 조선대학교 대학원 컴퓨터공학과 졸업 (공학석사)
2000년 조선대학교 대학원 컴퓨터공학과 졸업 (공학석사)
1997~현재 한국교통대학교 소프트웨어학과 교수
※ 관심분야 : 시스템소프트웨어, ITS