Advanced OTP Authentication Protocol using PUFs

Jonghoon Lee, Jungsoo Park, Seungwook Jung, and Souhwan Jung
School of Electronic Engineering
Soongsil University
Seoul, Republic of Korea
{ttaz, ddukki86, seungwookj, souhwanj}@ssu.ac.kr

***Abstract*—The One-Time Password (OTP) is an ephemeral password that can be used as a multi-factor authentication method when secure authentication is needed. This OTP is used to counter not only Man-in-the-Browser (MITB) attacks, but also memory hacking attacks. Alternatively, the financial systems use time synchronous OTP using Hash Message Authentication Code (HMAC)-based protocol to support secure authentication. However, it is possible to generate correct OTPs due to potential of stealing sensitive information of the OTP generator through intelligent phishing attacks. Therefore, it needs another scheme to prevent from generating the same OTPs. This paper proposes a new scheme using Physical Unclonable Functions (PUFs) to solve these problems. First, it is impossible to generate the same OTP values because of the physically unclonable features of PUFs. Moreover, sensitive information encrypted by hash and encryption function is exchanged through communication channel. Hence, the proposed protocol provides stronger OTP and robust authentication protocol by adding PUFs in the OTP generator.**

Keywords-OTP; authentication; PUF; HMAC

I. INTRODUCTION

The OTP [1] [2] [3] is an ephemeral password that is used as a strong and secure authentication method. Especially, financial systems utilize the OTP as an additional authentication factor to verify a user's identity. However, as social engineering and phishing attacks become more and more intelligent, various threats still exist. Recent attackers set up specific targets to collect privacy information related to public-key infrastructure (PKI) certificates [4], financial transaction, and the OTP generator, etc. These behaviors have enough availability to cause financial accidents. For instance, some information of SecurID, OTP generator, which is manufactured by RSA Security Inc. [5] was leaked in 2011 because of hacking in their systems. If this information was mixed with user's privacy information, an accident could have occurred. As above instance, there are various attacks. Therefore, it is urgent to make countermeasures to prevent those attacks.

First, we inquire about basic principles of the OTPs and look into their problems before proposing the countermeasure. Consequently, we propose an effective method to prevent its drawbacks. The OTP basically generates random values through an advantage of one-way functions, hash functions, to counter the replay attack. But the eavesdrop, social engineering, or active attacks still exist. There are many kinds of methods for generating OTP. First, an OTP authentication system such as S/KEY One-Time Password System was proposed by Bellcore Inc. [1]. The S/KEY uses hash function (md5 [4], SHA-1 [4], HMAC [4], etc) chains because it is impossible to invert the hash functions [1]. The Time Synchronized OTP [3], such as SecurID, uses the same time information between the server and the client. The Challenge-Response OTP uses the response corresponding with the challenge generated by the server. The Event Synchronized OTP [2] uses the shared counter that increases equally between the server and the client. Nowadays, the Time Synchronized OTP, among many methods, is generally used. However, as attacks become more and more intelligent, many threats still exist. An attacker can generate the same OTP value if he collects enough information of a targeted person and it is available to clone the OTP generator using hardware techniques. Therefore, it is necessary to consider secure measures because of these above reasons. This paper proposes a new secure OTP mechanism using the characteristic of PUF not to generate the same outputs of PUFs.

The remainder of this paper is organized as follows. In Section 2, security threats for OTP are described. Section 3 shows our proposed protocol. Section 4 analyzes the proposed protocol. Finally, Section 5 concludes this paper.

II. SECURITY THREATS

The principle of generating an OTP value is to use the output of a cipher function, such as hash function, using secret key and security token. Figure 1 describes the principle of the OTP.

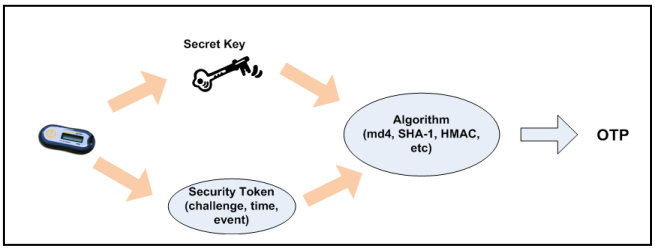

Figure 1. The Principle of Generating OTP

There are three types of the OTP generator approaches: Challenge-Response, Time Synchronous, and Event Synchronous approaches. First, the Challenge-Response OTP generator receives a challenge from the server. The user inserts the challenge, security token, into the OTP generator and then sends the output of the OTP generator, response, to the server. Figure 2 describes the principle of the Challenge-Response OTP. Time/Event Synchronous OTP is the

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-285-1

authentication approach using synchronous time/counter information between the OTP generator and the server as the security token. Figure 3 shows the principle of Time/Event Synchronous OTP. First, a user log in to the server. The server verifies the user's ID and password and request an OTP value to the user. The OTP generator creates the OTP value using time/counter information and secret key. The user sends it to server and the server compares it with the output of the server. Considering the principles of these approaches, we describe the pros and cons of these approaches in Table 1.

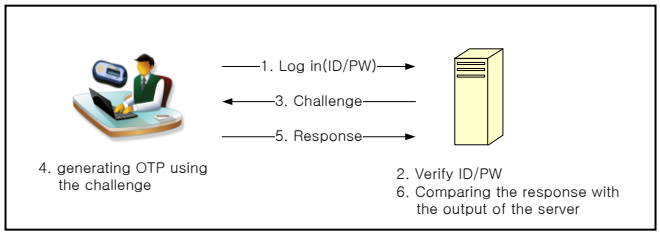

Figure 2. The Principle of Challenge-Response OTP

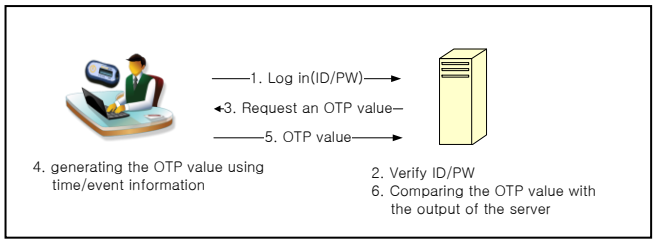

Figure 3. The Principle of Time & Event Synchronous OTP

TABLE I. PROS AND CONS OF APPROACHES GENERATING OTP

	Methods		
	Challenge-Response	***Time Synchronous***	***Event Synchronous***
Security Token	- Challenge from the server	- Time Sync	- Event Sync
Pros	- No need to maintain security token continuously	- Low error rate by users - Low traffic	- Low error rate by users Low traffic
Cons	- to need secure channel - to maintain CRPs	- to correct time sync deviation	- to correct event sync deviation

However, attackers can sufficiently generate the same values if they acquire information related in Security Token and Key using various and elaborate social engineering, phishing, and pharming attacks. It is possible to generate the same values because OTP values are generated by only software methods if they insert the same input information. In addition, it is an enough threat that attackers can clone OTP generators using hardware techniques. Thus, it is not desirable to count these threats through software methods alone. Therefore, by adding hardware components, such as PUFs, it is impossible for attackers to generate the same outputs even though they clone the OTP generator because of characteristics of the PUF. Also, it is impossible to discover its characteristics. In next section, we look into previous OTP protocols and then propose a new protocol using PUF to enhance security.

III. PROPOSED PROTOCOL

We first look into presenting the OTP protocol in financial systems and propose a stronger and more secure OTP protocol. The security model of Time Synchronous OTP generator is presented in Figure 4 [6].

There are three methods to insert Transaction Information in the OTP generator.

ⓐ The user directly inserts the Transaction Information using the keypad of the OTP generator.

ⓑ The user inserts the Transaction Information using sensor, 3D barcode reader, and Quick Response (QR) code reader of the OTP generator.

ⓒ The financial company inserts the Transaction Information through communication channel between the financial company and the OTP generator.

First, the OTP generator verifies the Personal Identification Number (PIN) the user inserts. If it isn't correct, the authentication is denied. If the PIN is correct, it prints the OTP value using the Secret Information (K) stored in the OTP generator, the Synchronous Information and Transaction Information (TI). The user inserts the OTP value in the user's terminal (Web Browser) and sends it to the server of the financial company. The server of the financial company compares this OTP value with the OTP value generated in the server using the function with the same information. If its value is matched, the server allows its transaction. Figure 5 describes the flow of Transaction Verification Protocol using OTP and Table II describes its notations. However, TI is not used in real financial systems. Problems can arise if attackers modify TI using the same OTP value by MITB or eavesdropping attacks. In other words, attacker can remit the user's money to the modified account. Financial systems allow its clients to use the OTP value once a minute to prevent this problem. A potential problem arises if attackers input the OTP value before the user inserts it. However, it is very difficult for the attackers to insert the OTP value through the man-in-the-middle attack (MITM), MITB, and sniffing, etc before the user uses it. To prevent these problems, this paper proposes a robust and secure authentication method using PUFs.

A. *PUFs*

PUFs utilize a hardware characteristic of an integrated circuit (IC) and this characteristic is different for each PUF. In other words, it is impossible to clone the characteristic of IC even if an attacker clones an IC of the PUF. Therefore, it is impossible to generate the same output even though the attacker clones the PUF. Since PUFs generate random outputs corresponding to each input, it is possible to use outputs corresponding to inputs as challenge-response pairs (CRPs). The Arbiter PUF creates two delay paths for each input, and produces an output based on which path is faster [7]. G. E. Suh and S. Devadas [7] also introduced PUF-based

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-285-1

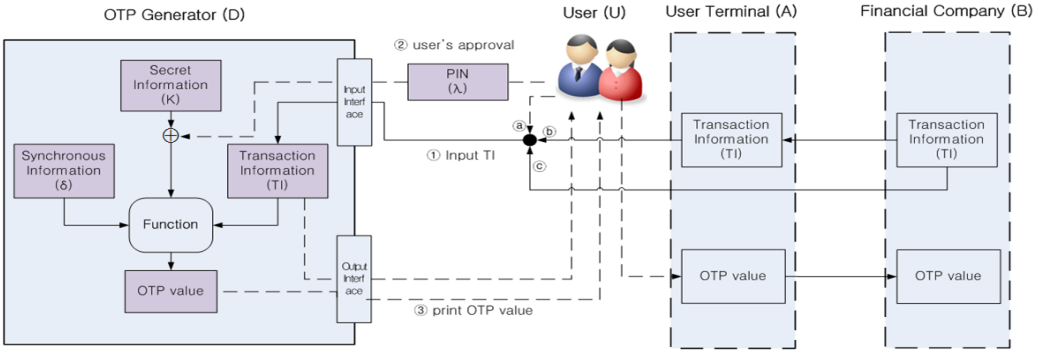

Figure 4. The Security Model of Time Synchronous OTP

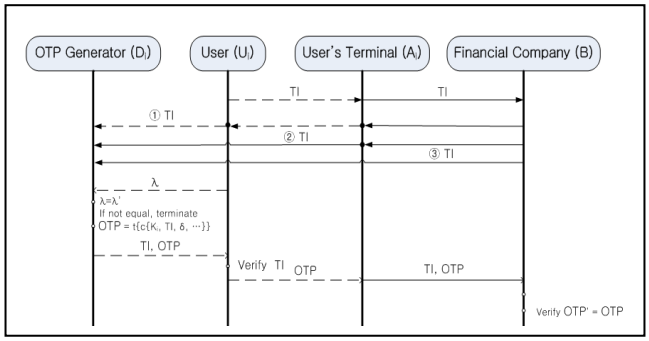

Figure 5. The Previous Protocol Flow

TABLE II. THE NOTATIONS OF THE PREVIOUS PROTOCOL

Notation	*Description*	*Notation*	*Description*
A_i	ith user's terminal	TI	transaction information
B	financial server	OTP	An OTP value
D_i	ith user's OTP generator	c{.}	cipher algorithm
K_i	ith user's secret information	t{.}	truncation algorithm
U_i	ith user	f{.}	OTP generation algorithm
δ	sync information	A→B: M	Send M from A to B through the communication channel
λ	PIN	A ⇢ B: M	Send M from A to B through the channel that user recognizes

authentication and cryptographic key generation with PUFs. The proposed protocol prevents from expecting the outputs of the OTP using the advantage of PUFs. However, the server has to maintain and store many CRPs for PUF-based authentication. L. Kulseng, Z. Yu, Y. Wei, and Y. Guan [8], M. Akgu¨n, M.S. Kiraz, and H. Demirci [9], S. W. Jung and S. H. Jung [10] proposed HMAC-based mutual authentication protocol using PUF in Radio-Frequency Identification (RFID) to solve this problem. By applying that protocol in OTP protocol, the proposed protocol in this paper could solve the above problem and assure strong authentication.

B. *Proposed Protocol*

We assume that the OTP generator is equipped with a communication channel to exchange challenge-response of the PUF. We add a PUF in the OTP generator and use the output of the PUF to generate the OTP values.

Figure 6 depicts the flow of the proposed protocol and Table III shows its notations. The main difference from the previous protocol is to use a PUF to assure secure transactions. The PUF basically utilizes cipher function to secure challenge-response pairs of the PUF and HMAC-based function to check the errors of PUF messages between the OTP generator and the server.

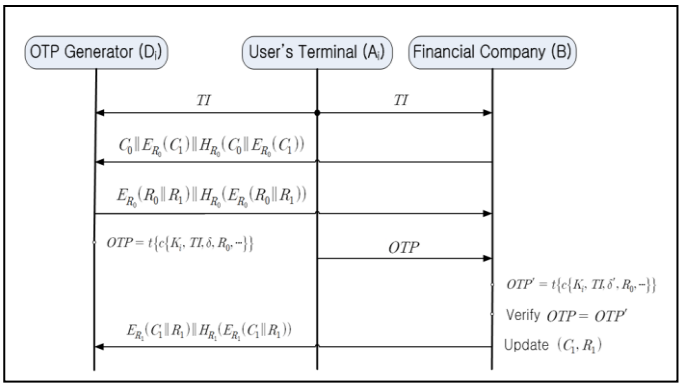

Figure 6. The Proposed Protocol Flow

Step 1: The user sends TI to the server and the OTP generator as a Hello message.

Step 2: The server sends the challenge and next challenge to the OTP generator

$$C_0' \| E_{R_0}(C_1') \| H_{R_0}(C_0' \| E_{R_0}(C_1'))$$

Step 3: The OTP generator sends the response and next response to the server.

$$E_{R_0}(R_0' \| R_1') \| H_{R_0}(E_{R_0}(R_0' \| R_1'))$$

Step 4: The OTP generator generates an OTP value and sends it to the server.

Step 5: The server verifies the OTP value and updates next challenge-response pair. The server sends ACK message after update.

$$E_{R_1}(C_1' \| R_1') \| H_{R_1}(E_{R_1}(C_1' \| R_1'))$$

TABLE III. THE NOTATIONS OF THE PROPOSED PROTOCOL

Notation	*Description*
C_n	nth challenge from the Financial Company
R_n	nth response of PUF from C_n
$E_K(\cdot)$	Encryption Function with K(Secret Key)

The server only stores initial CRP, (C_0, R_0), and updates next CRP, (C_1, R_1) in the authentication process to reduce loads of CRPs. By adding a response of PUF, it is difficult for the attackers to predict and re-create the same value.

IV. ANALYSIS OF THE PROPOSED PROTOCOL

B The attacks we mentioned in Section 2, such as phishing, pharming and social engineering attack, are serious issues. This section analyzes other threats, such as eavesdropping, blocking message, and replay attack.

The proposed protocol prevents eavesdropping attack and secures user information because sensitive values are protected by the cipher and hash function. Furthermore, this protocol also prevents blocking message because the server does not update the CRP unless the server verifies the OTP value from the OTP generator. The replay attack is impossible since this protocol uses fresh CRPs and OTPs every time. Moreover, financial information is sent through secure channel such as SSLv3. The secret information of the OTP generator and the PUF is only shared between the user and the server, thus spoofing attack is impossible unless this information is exposed. As the PUF is Physical Unclonable Function, it is also impossible to clone the OTP generator. Even though an attacker tries to clone an OTP generator using hardware technique, the outputs of the cloned PUF are totally different from an original one because of its characteristic. Therefore, cloning attack is impossible. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber [11] described the attack modeling on PUFs. This paper presume that an adversary Eve has collected a subset of all CRPs of the PUF, and tries to derive a numerical model from this data using machine learning techniques [11]. Our protocol protects the response of the PUF by encryption. Therefore, it is impossible for an attacker to collect CRPs of the PUF. The hacking memory attack does not have any impact on the OTP generator because the OTP generator does not have to store its output values. This feature of the PUF is the most benefit among its features. However, attacks such as intelligent phishing and pharming still exist as problems. To prevent the above problems, the proposed OTP protocol also uses transaction information that consists of account information, transaction time, and user information, etc.

V. CONCULSION

Existing Time Synchronous OTP protocol uses Secret Information and Sync Information shared between the OTP generator and the server to verify user's transaction in financial systems. It is also used as the multi-factor authentication in other systems. Attacks to acquire user's privacy information through various and intelligent social engineering, phishing attacks have increased in the past years. If attackers effectively use this sensitive information, it causes another financial incident. Many systems use the OTP generator to reduce these threats as a multi-factor authentication method. However, it is possible to clone an OTP generator and generate the same OTP values if an attacker acquires enough information about a user.

This paper introduced a new protocol using PUFs to assure more secure authentication. Moreover, our protocol not only prevent from cloning the OTP generator because of the characteristic of PUFs, but also phishing attack through Transaction Information. However, the proposed protocol requires the OTP generator, which is equipped with a communication channel to exchange information of PUFs. By using the OTP generator equipped with keypad, it is possible to implement a new protocol without communication channel. In conclusion, our protocol enhances security and provides more robust authentication method than existing ones.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center)) support program (NIPA-2013-H0301-13-1003) supervised by the NIPA(National IT Industry Promotion Agency).

REFERENCES

[1] N. Haller, C. Metz, P. Nesser, and M. Straw, A One-Time Password System, RFC 2289 IETF, Feb. 1998, pp. 1-8.

[2] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, HOTP: An HMAC-based One-Time Password Algorithm, RFC 4226 IETF, Dec. 2005, pp. 1-7.

[3] D. M'Raihi, S. Machani, M. Pei, J. Rydell, TOTP: Time-Based One-Time Password Algorithm, RFC 6238 IETF, May. 2011, pp. 1-7.

[4] W. Stallings, Cryptography and Network Security, 4th ed., Pearson Prentice Hall, 2006, pp. 318-372, 419-430.

[5] RSA SecurID, http://www.emc.com/security/rsa-securid.htm.

[6] H. W. Sim, W. J. Kang, and H. Y. Park, An One Time Password based e-Financial Transaction Verification Protocol, TTAK.KO-12.0167 TTA, Dec. 2011, pp. 1-9.

[7] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," Proc. 44th ACM Annual Design Automation Conference 2007, Jun. 2007, pp. 9-14.

[8] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight Mutual Authentication and Ownership Transfer for RFID systems," Proc. IEEE INFOCOM 2010, Mar. 2010, pp. 1-5.

[9] M. Akgün, M.S. Kiraz, and H. Demirci, "Cryptanalysis of Lightweight Mutual Authentication and Ownership Transfer for RFID System," Proc. IEEE Lightweight Security & Privacy: Devices, Protocols and Applications, Mar. 2011, pp. 20-25.

[10] S. W. Jung and S. H. Jung, "HRP: A HMAC-based RFID mutual authentication protocol using PUF," Proc. International Conference on Information Networking 2013, Jan. 2013, pp. 578-582.

[11] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling Attacks on Physical Unclonable Functions," Proceedings of the 17th ACM Conference on Computer and Communications Security, Oct. 2010, pp. 237-249.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-285-1