On the Security of Multi-Factor Authentication: Several Instructive Examples*

Yun Huang, Weijia Xue, Geshi Huang, Xuejia Lai

Department of Computer Science & Engineering , Shanghai Jiaotong University, Shanghai 200240, China
hycnsjtu@hotmail.com, icelikejia@sjtu.edu.cn, huang.ge.shi@foxmail.com, lai-xj@cs.sjtu.edu.cn

Abstract - How to propose a secure multi-factor authentication system remains a major concern. In this work we review several instructive examples from a practical and a theoretical point of view. From these observations we extract important guidelines for future works on multi-factor authentication.

Index Terms - Multi-factor authentication, security, one-time password, biometrics.

I. Introduction

Multi-factor authentications are considered stronger authentication as they combine several of the following authentication alternatives: 1) knowledge-based, that is something only known by the user, like a password or a PIN. 2) object-based, that is something the user possesses, like a physical token. 3) identity-based, that is a user's feature, i.e. biometrics, which relies on the uniqueness of a physical characteristic of a person such as fingerprints, facial features, iris, and voice. Each of those authentication options has its own advantages and disadvantages.

The increasing number of authenticated applications and the constant growing of attacks seem to call for multi-factor authentication technologies. To date the two most common multi-factor authentications are OTP authentication, based on a static password and a dynamic one-time password, and biometrics authentication based on a memorized password coupled with a biometric feature. The idea of OTP was first suggested by Leslie Lamport [1] in the early 1980s, and had developed into many patented OTP tokens [2][3][4] over the years. A few standards [5][6][7] have been introduced to facilitate the interoperability of OTP authentication. Furthermore, over the past few years, the use of a two-factor biometric authentication have been often suggested [8][9][10][11][12][13].

On the other hand, major issues in multi-factor authentication have been pointed out. In this paper, we review several instructive examples from a practical and a theoretical point of view. From these observations we extract important guidelines for future works on multi-factor authentication.

The rest of this paper is organized as follows. Section 2 presents the first example: Different solutions of two online banks, where, in the first case, the customer is forced to download a specific software, supposed to improve the security, in order to connect to the online bank; in the second case the online bank chooses to only adjust the login process. Section 3 gives the second example: RSA SecurID® One-Time Passwords, where the core of the authenticator is the proprietary SecurID® hash function and the block cipher at the heart of the function can be broken in milliseconds. Section 4 introduces the third example: GridCode One-Time Passwords, where the OTP from a human computable keyed MAC is an easily invertible hash function. In fact such poor choices can render a multi-factor authentication even more vulnerable to attacks than a common strategy based on a static password. Section 5 brings the fourth example: Multi-Factor Biometrics for Authentication, where a compromised transformation key from a password or a token, permits an imposter to be falsely accepted by the biometric system with no more than two attempts on average. Conclusions are then presented in Section 6.

II. Example I: Different solutions to anti-phishing of two online Banks

Phishing is the act of attempting to acquire information such as usernames, passwords, and credit card details by masquerading as a trustworthy entity in an electronic communication. It is different from the Man-In-The-Middle (or parallel session) Attack where the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to one another over a private connection, while in fact the entire conversation is controlled by the attacker.

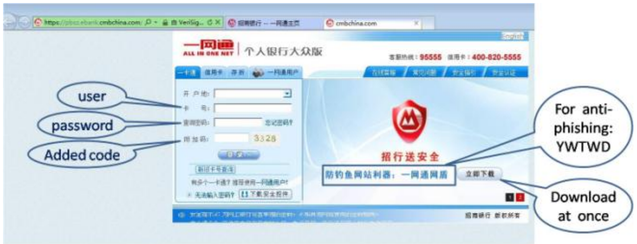
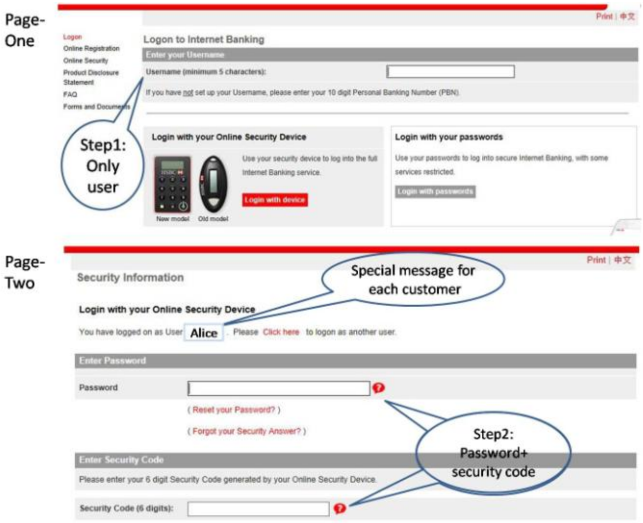


Fig. 1 Online Bank A: download the special software to anti-phishing

Both online banks consider phishing and propose a solution as shown on Fig.1 and Fig.2. It is mandatory for a customer from Bank A to download a special anti-phishing software while Bank B adjusts the login process by adding an extra step in the communication in order to improve the

*This work was supported by the National Natural Science Foundation of China (61073149 and 61272440), State Key Laboratory of ASIC & System (11KF0020), Key Lab of Information Network Security, Ministry of Public Security (C11603).

security. As in Page-Two of Fig.2, a special message is displayed for each customer. Once in the scenario of phishing, the Page-Two with the special message could not jump up at all.



Logon to Internet Banking

Enter your Username

Username (minimum 5 characters):

If you have not set up your Username, please enter your 10 digit Personal Banking Number (PBN).

Login with your Online Security Device

Use your security device to log into the full Internet Banking service.

Login with your passwords

Use your passwords to log into secure Internet Banking, with some services restricted.

Security Information

Login with your Online Security Device

You have logged on as User Alice . Please Click here to logon as another user.

Enter Password

Password

Enter Security Code

Please enter your 6 digit Security Code generated by your Online Security Device.

Security Code (6 digits):

Fig. 2 Online Bank B: logon through two-steps common communications to anti-phishing

III. Example II: RSA SecurID® One-Time Passwords

This example is taken from the article "Recent attacks on alleged SecurID® and their practical implications" by Alex Biryukov, et al. [14]. SecurID® tokens are developed by SDTI/RSA Security to authenticate users' login into a corporate computer infrastructure. Every minute (or every 30s), the device generates a new pseudorandom token code. By combining this code with his PIN, the user can gain access to the computer infrastructure of his company. More than 12 million employees in more than 8000 companies worldwide use SecurID® tokens. Institutions that use SecurID® include the White House, the U.S. Senate, the NSA, the CIA, the U.S. Department of Defense, and a majority of the Fortune 100 companies.

The core of the authenticator was the proprietary SecurID® hash function, developed by John Brainard in 1985. SDTI/RSA has never made this function public, but the source code for an alleged SecurID® token emulator was posted on the Internet in December 2000 (Wiener, Sample SecurID® token emulator with token secret import). This code was obtained by reverse engineering, and its correctness was confirmed by the reactions to this post.

Based on this implementation of the ‘‘Alleged SecurID® Hash Function’’ (ASHF), some cryptanalysis results were reported by Alex Biryukov, et al., see [14] for details. In their paper "Recent attacks on alleged SecurID® and their practical implications", they applied vanishing differentials in the block cipher to break the hash function at the core of ASHF in only a few milliseconds. This adaptively chosen plaintext attack rendered ASHF vulnerable, therefore at the end of their work, they recommend to replace all the ASHF-based tokens by tokens implementing open reviewed algorithm such as AES. From February 2003 onwards RSA has started phasing in AES-based tokens (RSA, security website).

IV. Example III: GridCode One-Time Passwords

This example can be found in the article "Attack on the GridCode One-Time Password" by Ian Molloy, et al. [15]. GridCode One-Time Passwords is a challenge-response protocol proposed by SyferLock. The system aims at providing a secure, device-less OTP with greatly reduced total cost of ownership. It is also purposed to be easy deployed. Customers and partners include managed business IT solutions (mbits), CA, and the Australian Government. Note that the system has also attained the NIST FIPS 140-2 certification.

In GridCode OTP, the user's response OTP is generated based on a human computable keyed MAC of the challenge state sent by the authenticator and a human memorable secret. The major issue is related to the human computable keyed MAC which is an easily invertible hash function. This poor choice makes a multi-factor authentication even more vulnerable to attacks than a static password as shown in the analysis results gathered in Table I (Ian Molloy, et al. [15]).

TABLE I Summary of results. Standard values are assumed: n=94, m=10, r=8, k=4, H (P) is 18-30 bits, and the OTP has a shared secret of 128 or 54 bits.

Attack		Password	OTP	GridCode
Brute Force	KR	2^{52} Queries	$2^{128}(2^{54})$ Queries	2^{27}Queries
	Imp		2^{26} Queries	2^{26}Queries
Dictionary	KR	2^{18}-2^{30} Queries	$2^{128}(2^{32})$ Queries	2^{20}-2^{26} Queries
	Imp		2^{26} Queries	
Key Logging	KR	1 Authentication	$2^{128}(2^{54})$ Queries	2^{27}Queries
	Imp		2^{26} Queries	2^{26}Queries
Shoulder Surfing	KR	2^{18}-2^{30} Cognition	$2^{128}(2^{54})$ Queries	2^{26+42} Cognition
	Imp		2^{26} Queries	
Eavesdropping/ Phishing	KR	1 Authentication	1 Auth., $2^{128}(2^{54})$ Time	2-4 Pairs, O(knr) Time
	Imp		2^{26} Queries	
KR: Key Recovery; Imp: Impersonation				

V. Example IV: Multi-Factor Biometrics for Authentication

The fourth example is taken from the paper "Multi-Factor Biometrics for Authentication" by Hisham Al-Assam, et al. [16]. In multi-factor biometric authentication a common approach consists in applying User-Based Transformations (UBTs) on the biometric features. Typically, UBTs relies on generating user-based transformation keys from a password/PIN or retrieved them from a token. One significant advantage of employing UBTs is its ability to achieve zero or near to zero Equal Error Rate (EER), i.e. it clearly differentiate a genuine user from an impostor.

However, the effect of compromised transformation keys on the authentication accuracy need to be tested rigorously. The experimental results presented in [16], are quite different from the expected behavior often reported in other publications [8][9][10][11][12][13] (Table II). In the case of a compromised transformation key, the other literatures [8]-[13] showed some accuracy drops while still remaining close to the accuracy of the biometric system only (upper six rows in Table II). Hisham Al-Assam, et al. [16] showed that in such a scenario, the same Operating Point (OP) setup to operate at a zero EER caused the False Acceptance Rate (FAR) of the system to reach unacceptable levels: FAR of 56.67% for Fingerprint and FAR of 66.69% for Face (last two rows in table II).

TABLE II Authentication results

source	Biometric Type	Biometric only	Two-Factor (secure)	Two-Factor (insecure)
[8]	Fingerprint	EER=5.66	EER=0	N\A
[9]	Iris	EER=3.2	EER=0	EER=8.6
[10]	Fingerprint	FAR=1 at FRR=7	EER=0	FAR=1 at FRR=7
[11]	Face	EER=15.63	EER=0	EER=16.21
[12]	Palm	EER=2.75	EER=0	N\A
[13]	Face	EER=7.19	EER=0	EER=7.19
[16]	Fingerprint	FAR=0.1 at FRR=16	EER=0	FAR=56.67 at FRR=0
	Face	FAR=0.67 at FRR=21.5	EER=0	FAR=66.69 at FRR=0
EER: Equal Error Rate; FAR: False Acceptance Rate; FRR: False Rejection Rate				

The main reason behind the biased evaluation of a compromised key is due to the simulation which is performed at operating point(s) whose values are completely different from the operating point(s) in the case of a secure key. This assumption is totally unrealistic, as it implicitly assumes that the biometric system knows it is a compromised key and automatically changes its OP. However, note that there is no way to distinguish a genuine key from a compromised key.

VI. Conclusions

Multi-factor authentications have been proposed with the purpose of strengthening security in authentication systems. However, the effect of multi-factor authentications should be considered carefully, as security could be greatly weakened if not properly set and used. In this paper, we present several examples of multi-factor authentications which misbehave, compared to the initial expectations. From these examples, we extract some guidelines for future works.

1) As a general rule, that can be extracted from examples I and II and III, an open and standard cryptographic primitive such as a protocol, a block cipher or a hash function, should be preferred to any special or hidden design.

2) Through Example I, we emphasize the need for the client to be simpler in the client-server applications. Compared to the server, usually handled by security specialists, common clients have less insight on the security of their environment. Thus the secure solution should be obvious to the common client, like the Page-Two jumping up with the special message in Fig. 2. Note especially that the client may be even the attacker. Hence the special software downloaded in client environment is more vulnerable, ruining the security of the whole authentication process.

3) Example III and IV highlight the fact that in multi-factor authentication, all the factors must be independent; that is: password, token key and biometric transformation key must be independent. Indeed if a factor is compromised, it does not influence the others, ruining the whole security. In turn the independence of the factors forces the attacker to break every single component, making it much harder.

Acknowledgment

The authors would like to deliver special thanks to Manuel Charlemagne for his helpful proofreading work.

References

[1] L. Lamport：Password Authentication with Insecure Communication, In: Comm. ACM, vol. 24, No 11, 1981, pp. 770-772

[2] RSA SecurID - Two Factor Authentication, Security Token – EMC, http://www.rsa.com/node.aspx?id=1156

[3] Kensuke Sawada: Authentication System, US8074075, Dec. 2011.

[4] Zhou Lu, HuaZhang Yu: One time password generating method and apparatus, US8184872, May 2012.

[5] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O. Ranen: HOTP: An HMAC-Based One-Time Password Algorithm, Dec. 2005, http://tools.ietf.org/html/rfc4226.txt

[6] N. Haller, C. Metz, P. Nesser, M. Straw: A One-Time Password System, Feb. 1998, http://www.ietf.org/rfc/rfc2289.txt

[7] RSA Laboratories - OTP-PKCS #11: PKCS #11 mechanisms for One-Time Password tokens, Dec. 2005, http://www.rsa.com/rsalabs/node.asp? id=2818

[8] Teoh, A.B.J., Ngo, D.C.L, Goh, A. 2004. BioHashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognition. Vol. 37(11), pp. 2245-2255.

[9] Lumini Alessandra and Loris Nanni. 2006. Empirical tests on BioHashing. Neurocomputing. Vol. 69, pp. 2390-2395.

[10] Anil K. Jain, Karthik Nandakumar and Abhishek Nagar. 2008. Biometric Template Security. EURASIP Journal on Advances in Signal Processing. pp. 1-17.

[11] Andrew B. J. Teoh, Kar-Ann Toh, and Wai K. Yip. 2007, 2^N Discretisation of BioPhasor in Cancellable Biometrics. Advances in Biometrics. pp. 435-444.

[12] Connie, T. and Teoh, A. and Goh, M. and D. Ngo. 2004, PalmHashing: a novel approach for dual-factor authentication. Pattern Analysis & Applications. Vol. 7(3),pp. 255-268.

[13] Wang, Y. and Plataniotis, KN. 2007, Face Based Biometric Authentication with Changeable and Privacy Preservable Templates. Biometrics Symposium. pp. 1-6.

[14] A. Biryukov, J. Lano, and B. Preneel. Recent attacks on alleged securid and their practical implications. Computers & Security, 2005. pp. 304-370

[15] Ian Molloy, and Ninghui Li. Attack on the GridCode One-Time Password, AsiaCCS2011, Hongkong, China. pp. 306-315

[16] Hisham Al-Assam, Harin Sellahewa, and Sabah Jassim. Multi-Factor Biometrics for Authentication: A False Sense of Security, MM&Sec2010, Roma, Italy. pp. 81-87.