Secure Authentication using QR-Code and Randomized Image Password

R. Aarthi[1], Dr. D. J. Evanjaline[2]
[1, 2] Department of Computer Science
[1, 2] Rajah Serfoji Govt. College (A), Thanjavur-613 005.

Abstract- *A keylogger is software designed to capture all of a user's keyboard strokes, and then make use of them to impersonate a user in financial transactions. An authentication protocol is a type of computer communications protocol or cryptographic protocol specifically designed for transfer of authentication data between two entities. A secure authentication protocols designing is challenging. Involving human in authentication protocols, while promising, is not easy because of their limited capability of computation and memorization. Therefore, relying on users to enhance security necessarily degrades the usability. On the other hand, relaxing assumptions and rigorous security design to improve the user experience can lead to security breaches that can harm the users' trust. This paper, demonstrate how careful visualization design can enhance not only the security but also the usability of authentication. To that end, the design of propose two visual authentication protocols: one is a one-time-password protocol, and the other is a random image-based authentication protocol. Through rigorous analysis, verify that our protocols are immune to many of the challenging authentication attacks applicable to the literature. Furthermore, using an extensive case study on a prototype of new protocols, highlight the potential of this approach for real-world deployment: this authentication protocol able to achieve a high level of usability while satisfying stringent security requirements.*

Keywords- Authentication protocol, one -time -password, Quick Response code

I. INTRODUCTION

In every online application security is the major concern. Every public network offers security by the means of authentication. Authentication process is a way to protect the network for illegitimate access. In a Client-Server Architecture it is required to authenticate client, server and the network between them. The attackers can attack the network by using illegal means like spoofing, phishing, bot/botnet. Authentication process can be understood using Authentication Interface and Authentication protocols. The Authentication interface is human-computer interface (HCI)[1]. HCI is the way by which human interacts with the authentication process. It can be text based or graphic based. Threats against electronic and financial services can be classified into two major classes: credential stealing and channel breaking attacks. Credentials such as users' identifiers, passwords, and keys can be stolen by an attacker when they are poorly managed. For example, a poorly managed personal computer (PC) infected with a malicious software(malware) is an easy target for credential attackers .On the other hand, channel breaking attacks—which allow for eavesdropping on communication between users and a financial institution—are another form of exploitation. While classical channel breaking attacks can be prevented by the proper usage of a security channel such as IPSec and SSL (secure sockets layer), recent channel breaking attacks are more challenging. Indeed, "keylogging" attacks—or those that utilize session hijacking, phishing and pharming, and visual fraudulence— cannot be addressed by simply enabling encryption. For example, whenever a user types in her password in a bank's sign in box, the keylogger intercepts the password. The threat of such keyloggers is pervasive and can be present both in personal computers and public kiosks; there are always cases where it is necessary to perform financial transactions using a public computer although the biggest concern is that a user's password is likely to be stolen in these computers. Even worse, keyloggers, often root kitted, are hard to detect since they will not show up in the task manager process list. To mitigate the key logger attack, virtual or onscreen keyboards with random keyboard arrangements are widely used in practice. Both techniques, by rearranging alphabets randomly on the buttons, can frustrate simple key loggers. Unfortunately, the key logger, which has control over the entire PC, can easily capture every event and read the video buffer to create a mapping between the clicks and the new alphabet. Our approach to solving the problem is to introduce an intermediate device that bridges a human user and a terminal. Then, instead of the user directly invoking the regular authentication protocol, she invokes a more sophisticated but user-friendly protocol via the intermediate helping device. Every interaction between the user and an intermediate helping device is visualized using a Quick Response (QR) code Thus, in our protocols, a user does not need to memorize extra information except a traditional security token such as password or PIN, and unlike the prior literature that defends against should-surfing attacks by requiring complex computations and extensive inputs. More

specifically, our approach visualizes the security process of authentication using a smart phone aided augmented reality. The visual involvement of users in a security protocol boosts both the security of the protocol and is re-assuring to the user because she feels that she plays a role in the process.

II. SYSTEM STRUCTURE

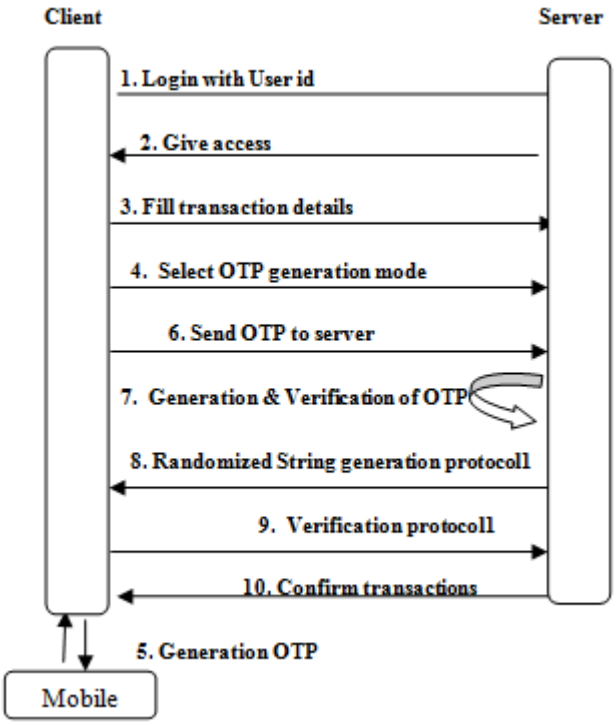

Figure 1. Authentication with random string

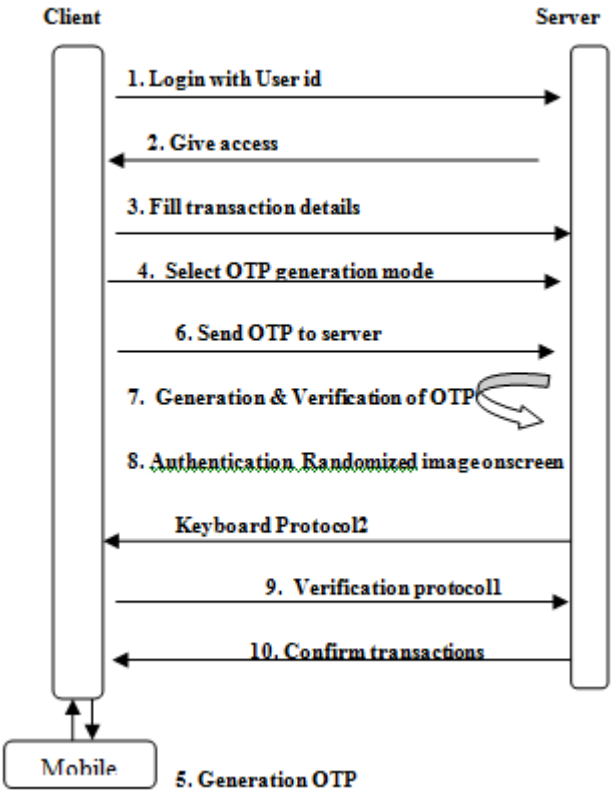

Figure 2. Authentication protocol with password and randomized onscreen keyboard

III. METHODOLOGY

Algorithm:

EncrK(.): an encryption algorithm which takes a key k and a message M from set M and output a cipher-text C in the set C.
DecrK(.): a decryption algorithm which takes a cipher-text C in C and a key K, and outputs a plain-text(or message) M in the set M.
OTP(.) : a one Time Password generation algorithm which takes a private key SK and a message M from the set M, and outputs to a signature σ.
ranStr(.): a signature verification algorithm which takes a public key PK as random String and signed message (M,σ), and returns valid or invalid.
ranImg(.): a signature verification algorithm which takes a public key PK as Image and signed message (M,σ), and returns valid or invalid.
QREnc(.): a QR encoding algorithm which takes a string S in S and outputs a QR code.
QRDec(.): a QR decoding algorithm which takes a QR code and returns a string S in S.

A. ONE TIME PASSWORD GENERATION (OTP):

A one-time password (OTP) is a password that is valid for only one login session or transaction. To make sure

that the user is an authorized user, OTP is generated and sent to his/her mobile number. This is usually the authentication method used when a transaction is verified with an OTP. It can be a list of passwords available with the user and each time user uses a different password. OTP which has been once used from the list is no longer valid for next session. One Time Password can also be generated every time the requests for it. One Time Password authentication helps preventing the access to unauthorized access to restricted areas. The bank system sends you an OTP and you then have few minutes to enter this OTP. This mechanism doesn't need any synchronization process as the OTP is originally generated by the server and send to a third party device. The server expects that you type the correct OTP.

B. QR CODE GENERATION

QR code (abbreviated from Quick Response Code) is the trademark for a type of matrix barcode (or two-dimensional barcode).Every interaction between the user and an intermediate helping device is visualized using a Quick Response (QR) code. The goal is to keep user-experience the same as in legacy authentication methods as much as possible, while preventing keylogging attacks. Thus, in our protocol, a user does not need to memorize extra information except a traditional security token such as password or PIN, and unlike the prior literature that defends against should-surfing attacks by requiring complex computations and extensive inputs. More specifically, our approach visualizes the security process of authentication.

C. AUTHENTICATION WITH RANDOM STRINGS

The user connects to the server and sends her ID. The server checks the ID to retrieve the user's public key (PKID) from the database. The server then picks a fresh random string and encrypts it with the public key. In the terminal, a QR code is displayed prompting the user to type in the string. The user decodes the QR code. The server checks the result and if it matches what the server has sent earlier, the user is authenticated. Otherwise, the user is denied.

D. AN AUTHENTICATION PROTOCOL WITH RANDOMIZED IMAGE ONSCREEN

It is used password shared between the server and the user, and a randomized image. The user connects to the server and sends her ID. The server checks the received ID to retrieve the user's public key from the database. The server prepares, a random permutation of a image arrangement, and encrypts it with the public key .Then, it encodes the cipher text with QR encoder .The server sends the result with a blank image matrix. When the user sees the blank image with the QR code through an application that has a private key, alphanumeric appear on the blank keyboard and the user can click the proper button for the password. The user types his password on the terminal's screen while seeing the keyboard layout through the application. The terminal does not know what the password is but only knows which buttons are clicked. Identities of the buttons clicked by the user are sent to the server by the terminal. The server checks whether the password is correct or not by confirming if the correct buttons have been clicked for the image.

IV. CONCLUSION

In this paper, a new approach to authenticate a user with a new family of QR-codes and graphical image passwords called. Trust in our protocols can be seen shifted from PC to smart phone to make authentication protocols secure against malware in a PC. However, considering that it is not easy to protect user's credentials when a malware resides in a PC without sacrificing usability , and that a user sometimes has to use an un trusted PC such as a public PC , this approach to move trust to the smart phone that is at least more trustworthy than public PCs is plausible. Also, in Protocol 2 that uses a password for authentication, a smart phone is not required to be trusted because a password is another factor for any successful authentication, proposed and analyzed the use of user driven visualization to improve security and user-friendliness of authentication protocols. Moreover, it's shown two realizations of protocols that not only improve the user experience but also resist challenging attacks, such as the keylogger and malware attacks. These proposed protocols utilize simple technologies available in most out-of-the-box smart phone devices. Traditionally static passwords can more easily be accessed by an unauthorized intruder given enough attempts and time. By constantly altering the password, as is done with a one-time password, this risk can be greatly reduced. In addition to offering protection against online guessing attack, random image when combined with QR login, is resistant to shoulder surfing attack and attacks using malwares. Random image and QR-code authentication is not a bullet proof system, but it offers reasonable security and usability and will fit well with practical applications for improving online security.

REFERENCES

[1] Ishupreet Kaur, 2 Gargi Narula . One Time Password Using Sphere Angle Based Random Password Generation for Online Portals - A Review. IJCAT International Journal of Computing and Technology, Volume 1, Issue6, July 2014 ISSN : 2348 – 6090.

[2] J. Brown. Zbar bar code reader, zbar android sdk 0.2. http://zbar. sourceforge.net/, April 2012.

[3] C.-H. O. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu. Gangs: gather, authenticate 'n group securely. In J. J. Garcia-Luna-Aceves, R. Sivakumar, and P. Steenkiste, editors, MOBICOM, pages 92–103. ACM, 2008.

[4] D. Crockford. The application/json media type for javascript object notation (json). http://www.ietf.org/rfc/rfc4627.txt?number=4627, July 2006.

[5] D. Davis, F. Monrose, and M. Reiter. On user choice in graphical password schemes. In Proc. Of USENIX Security, 2004.

[6] N. Doraswamy and D. Harkins. IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall, 2003.

[7] M. Farb, M. Burman, G. Chandok, J. McCune, and A. Perrig. Safeslinger: An easy-to-use and secure approach for human trust establishment. Technical report, CMU, 2011.

[8] H. Gao, X. Guo, X. Chen, L. Wang, and X. Liu. Yagp: Yet another graphical password strategy. In Proc. of ACM ACSAC, pages 121–129, 2008.

[9] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal, 1988.

[10] Google. Android. http://www.android.com/, 2011.

[11] E. Hayashi, R. Dhamija, N. Christin, and A. Perrig. Use your illusion: secure authentication usable anywhere. In Proc. of ACM SOUPS, 2008.

[12] C. Herley and D. Florencio. How to login from an internet caf´e without worrying about keyloggers. In Proc. of ACM SOUPS, 2006.

[13] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In Security and Privacy (SP), 2012 IEEE Symposium on, pages 553–567. IEEE, 2012.

[14] T. Holz, M. Engelberth, and F. Freiling. Learning more about the underground economy: a case-study of keyloggers and dropzones. In Proc. of ESORICS, pages 1–18, 2009.

[15] N. Hopper and M. Blum. Secure human identification protocols. In Proc. of ASIACRYPT, 2001. R. Housley. RFC3686: Using Advanced Encryption Standard (AES) counter mode with ipsec encapsulating security payload (ESP). http://www.ietf.org/rfc/rfc3686.txt, 2004.

[16] S. Josefsson. RFC 4648: the base16, base32, and base64 data encodings. http://tools.ietf.org/html/rfc4648, 2006.

[17] C. Karlof, U. Shankar, J. D. Tygar, and D. Wagner. Dynamic pharming attacks and locked same-origin policies. In Proc. of ACM CCS, pages 58–71, 2007. J. Katz and Y. Lindell. Introduction to modern cryptography. CRC Press, 2008.

[18] H. Krawczyk, M. Bellare, and R. Canetti. Hmac: Keyed-hashing for message authentication. RFC, 1997.

[19] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder- surfing by using gaze-based password entry. In Proc. of ACM SOUPS, pages 13–19, 2007.

[20] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder- surfing by using gaze-based password entry. In Proc. of ACM SOUPS, pages 13–19, 2007.

[21] L. Lamport. Password authentication with insecure communication. Communications of the ACM, 24(11):770–772, 1981. [31] J. Lim. Defeat spyware with anti-screen capture technology using visual persistence. In Proc. of ACM SOUPS, pages 147–148, 2007.

[22] Y.-H. Lin, A. Studer, Y.-H. Chen, H.-C. Hsiao, E. L.-H. Kuo, J. M. McCune, K.-H. Wang, M. N. Krohn, A. Perrig, B.-Y. Yang, H.-M. Sun, P.-L. Lin, and J. Lee. Spate: Small-group pki-less authenticated trust establishment. IEEE Trans. Mob. Comput., 9(12):1666–1681, 2010

[23] M. Mannan and P. C. van Oorschot. Leveraging personal devices for 1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE spermission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.