HPAKE HONEY PASSWORD AUTHENTICATED KEY EXCHANGE FOR FAST AND SAFER ONLINE AUTHENTICATION

[1]T. Supraja, [2]Balusani Virinchi, [3]Gunda Uday Kumar, [4]Jadav Naresh, [5]Segu Shriya

Assistant Professor in Department of CSE Sreyas Institute Of Engineering And Technology

[2.3.4,5]UG Scholar in Department of CSE Sreyas Institute Of Engineering And Technology

Abstract

Password-only authentication is one of the most popular secure mechanisms for real-world online applications. But it easily suffers from a practical threat - password leakage, incurred by external and internal attackers. The external attacker may compromise the password file stored on the authentication server, and the insider may deliberately steal the passwords or inadvertently leak the passwords. So far, there are two main techniques to address the leakage: Augmented password-authentication key exchange (aPAKE) against insiders and honey word technique for external attackers. But none of them can resist both attacks. To fill the gap, we propose the notion of honey PAKE (HPAKE) that allows the authentication server to detect the password leakage and achieve the security beyond the traditional bound of aPAKE.

Keywords: Hpake, Password Leakage, Attacker

I INTRODUCTION

The username/password paradigm is the most commonly used authentication mechanism in security applications. Alternative authentication factors, including tokens and biometrics, require purchasing additional hard-ware, which is often

considered too expensive for an application. However, passwords are low-entropy secrets, and subject to dictionary attacks. Hence, they must be protected during transmission. The widely deployed method is to send passwords through SSL/TLS. But, this requires Public Key Infrastructure (PKI) in place; maintaining a PKI is expensive. In addition, using SSL/TLS is subject to man-in-the-middle attacks. If a user authenticates himself to a phishing website by disclosing his password, the password will be stolen

even though the session is fully encrypted. Since passwords are inherently weak, one logic solution seems to replace them with strong secrets, say, cryptograph-icily secure private keys. This approach was adopted by the UK National Grid Service (NGS) to authenticate users. In the UK, anyone who applies to access the national grid computing resource must first generate a private/public key pair of his own, and then have the public key certified by NGS. The authentication procedure for the grid computing environments in the USA is similar. However, developments in the past ten years reveal that users – most of them are non-computer specialists – encounter serious difficulties in managing their private keys and certificates. This has greatly hindered the wider acceptance of the grid computing technology. Hence, weak passwords are just a fact of life that we must face. Researchers have been actively exploring ways to perform password-based authentication without using PKIs or certificates – a research subject called the Password-Authenticated Key Exchange (PAKE). The first milestone came in 1992 when Beloin and Merrit introduced the EKE protocol. Despite some reported weaknesses the EKE protocol first demonstrated that the PAKE problem was at least solvable. Since then, a number of protocols have been proposed. Many of them are simply variants of EKE, instantiating the "symmetric cipher" in various ways. The few techniques that claim to resist known attacks have almost all been patented – most notably, EKE was patented by Lucent Technologies, and SPEKE by Phoenix Technologies. As a result, the scientific community and the wider security industry cannot readily benefit from the implementations of these techniques.

The security with EKE and SPEKEprotocols is only heuristic. Given the way the two techniques were designed, formal security proofs seems unlikely without introducing new assumptions or relaxing requirements; we will explain the details in Section 4. In the following section, we will introduce a different approach to solve thecae problem, and show that our solution is free from the security issues reported with the EKE and SPEKE protocols.

II LITERATURE SURVEY

Against Insiders. In Figure 1a, Augmented password authentication key exchange (aPAKE) is designed to allow a client and a server to establish a session key based on a password, where the client has the password plaintext and the server only holds the verifier. This technique prevents the server from knowing the password, and therefore resists the insider attacks. Since Bellovin and Merrit introduced this notion, many researchers proposed various aPAKE schemes in order to improve the security and efficiency performance. Among them, OPAQUE is the most well-studied scheme with the strongest security and thus, it recently is standardized by the Crypto Forum Research Group of the Internet

Engineering Task Force (IETF). Against Outsiders. Honeyword technique (see Figure 1b) is proposed to detect the password leakage for the most common password-only authentication systems, passwordover-TLS. This approach associates t−1 decoy and plausiblelooking passwords (i.e., honeywords) to each account. The honeywords and the real password are collectively called sweet words. If an attacker steals the password file, she cannot tell the real one and probably (with 1−1/t probability) log in with a honeyword. Then, the server can detect the password leakage from the "wrong" login. The follow-up works focus on the honeyword generation algorithms so as to produce more plausible-looking decoys and the detection methods to improve reliability. Others. Password less authentication or multi-factor authentication systems make good use of other factors, e.g., smartphone and fingerprint. They significantly reduce the risk of password leakage. If an attacker steals the password, she still needs additional factors to compromise account. Besides, in some of these designs, authentication server does not need to store the password-related data, so that even if the attacker compromises the storage file on server, she cannot carry out offline password guessing as long as other factors are secure. A typical design can be seen in that a smart device (as an authentication factor) is used to store the password-related data, making systems resist offline guessing in the case of server compromise. Shortcomings. The techniques above, unfortunately, have the following shortcomings.The honeyword mechanism requires the client to send the password plaintext to the server (via a server-authenticated secure channel), otherwise the server cannot tell if the login password is real. Thus an insider can directly steal the plaintext of the login password without any guessing attacks. In aPAKE, the server has to store the verifiers in the password file for authentication. But an external attacker may steal the file and carry out guessing attacks [24] to recover the password. This vulnerability is inherent in aPAKE. And neither of these methods can provide a solution maintaining security against both insiders and outsiders. As for other (passwordless or multi-factor) approaches, they may provide stronger security relying on extra factors, which may bring disadvantages to deployability and usability. In this paper, we do not consider them and only focus on passwordonly authentication. According to the above discussion, we thus raise a question: "How could one design a fast and secure password-only authentication scheme that can resist both the insider and external attackers.

III EXISTING SYSTEM

PAKE protocols are commonly classified as either balanced or augmented. A balanced PAKE assumes that the two parties share a secret, which is a password or derived from a password. Typical requirements for a secure balanced PAKE include:

1) resisting offline dictionary attacks

2) limiting online attacks to one password guess per protocol execution

3) ensuring session-key security

4) providing forward secrecy

Disadvantages

An attacker must perform an offline dictionary attack that cannot make use of any pre-computed table. Note that these requirements increase the burden on attackers, but do not stop attacks; once a server is compromised, an offline dictionary attack should be expected (one response is to update all passwords).

IV PROBLEM STATEMENT

This project is an online shopping website where users can register with application by entering valid user name and password along with text file data which is used for every time login. User must give same text file every time and apply Honey Password-Authenticated Key Exchange when he login to application which will encrypt and send key to authentication server who will verify and validate user. If Password-Authenticated Key Exchange is success then only user is considered as normal user else he is considered as attacker.

V PROPOSED SYSTEM

We propose the notion of honey PAKE (HPAKE) that allows the authentication server to detect the password leakage and achieve the security beyond the traditional bound of aPAKE. Further, we build an HPAKE construction on the top of the honeyword mechanism, honey encryption, and OPAQUE which is a standardized aPAKE. We formally analyze the security of our design, achieving the insider resistance and the password breach detection.

Advantages

The honey word mechanism requires the client to send the password plaintext to the server (via a server-authenticated secure channel), otherwise the server cannot tell if the login password is real. Thus, an insider can directly steal the plaintext of the login password without any guessing attacks. In aPAKE, the

server hasto store the verifiers in the password file for authentication. But an external attacker may steal the file and carry out guessing attacks to recover the password.

VI IMPLEMENTATION

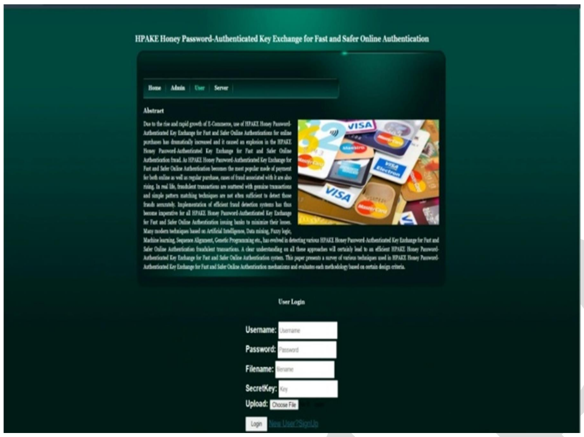
Online Shopping Website: Using this module web application is developed which has online shopping features where seller can use admin module to upload products and buyer can view products and purchase. This application provides option for payment, add products to cart, view products, search products get conformation from admin on purchase, use attacker module to show internal attacks. Show security methods to secure authentication process.

Admin Module: This module is part on online shopping website where admin and login to application add products with cost and product details and verify users as attackers or normal users and block users who are attackers. Admin can verify users for purchasing products and get confirmation.

User Module: This module is part of online shopping website where users can register with application by entering valid user name and password along with text file data which is used for every time login. User must give same text file every time and apply Honey Password-Authenticated Key Exchange when he login to application which will encrypt and send key to authentication server who will verity and validate user. If Password-Authenticated Key Exchange is success then only user is considered as normal user else he is considered as attacker.

Authentication Server Module: This module is used as middle layer between user registration process and login process verification for verifying Honey Password-Authenticated Key Exchange process. Every time new user registers this server will store a security key which is unique based on user input data. If same data is uploaded by user while login then only authentication server will give validation else authentication exchange will be failed

VII RESULTS


HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication
Abstract
User Login
Username:
Password:
Filename:
SecretKey:
Upload:
Login


Purchase Products
Product Id :
Product Name:
Product Description :
Quantity :
Total Cost :
Card Number :
Expiry Date :
CVV :
submit


HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication
View Users

VIII CONCLUSION

We propose the notion of HPAKE, which is the first of its type, achieving the advantages of the honeyword and aPAKE techniques, i.e., detecting the password leakage caused by external attackers and preventing the insider from getting the password plaintext. Using OPAQUE, honeyword mechanism, and honey encryption, we build a concrete HPAKE construction. To analyze the security of our design, we propose a game-based security model and formally prove the security of our design in this model. We implement and deploy the proposed scheme in the real-world environment. The experimental results show that our design is efficient for the realworld application.

REFERENCES

[1] J. Bonneau, C. Herley, P. C. Oorschot, and F. Stajano, “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes,” in Proc. IEEE S&P 2012, pp. 553–567.

[2] N. Huaman, S. Amft, M. Oltrogge, Y. Acar, and S. Fahl, “They would do better if they worked together: The case of interaction problems between password managers and websites,” in Proc. IEEE S&P 2021, pp. 1367–1381.

[3] D. Pasquini, A. Gangwal, G. Ateniese, M. Bernaschi, and M. Conti, "Improving password guessing via representation learning," in Proc. IEEE S&P 2021, pp. 265–282.

[4] W. Li and J. Zeng, "Leet usage and its effect on password security," IEEE Trans. Inform. Foren. Secur., vol. 16, pp. 2130–2143, 2021.

[5] "Have i been pwned?" [Online]. Available: https://haveibeenpwned.com

[6] "Yahoo! data breaches." [Online]. Available: https://en.wikipedia.org/w iki/Yahoo! data breaches

[7] "Yahoo tries to settle 3-billion-account data breach with $118 million payout." [Online]. Available: https://arstechnica.com/tech-policy/2019/0 4/yahoo-tries-to-settle-3-billion-account-data-breach-with-118-million -payout/

[8] Z. Whittaker, "Github says bug exposed some plaintext passwords," ht tps://www.zdnet.com/article/github-says-bug-exposed-account-passwor ds/, 2018.

[9] S. M. Bellovin and M. Merritt, "Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise," in Proc. ACM CCS 1993, pp. 244–250.

[10] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure passwordauthenticated key exchange using diffie-hellman," in Proc. EUROCRYPT 2000. Springer, pp. 156–171.

[11] C. Gentry, P. MacKenzie, and Z. Ramzan, "A method for making password-based key exchange resilient to server compromise," in Proc. CRYPTO 2006. Springer, pp. 142–159.

[12] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks," in Proc. EUROCRYPT 2018. Springer, pp. 456–486.

[13] M. Abdalla, M. Barbosa, T. Bradley, S. Jarecki, J. Katz, and J. Xu, "Universally composable relaxed password authenticated key exchange," in Proc. CRYPTO 2020. Springer, pp. 278–307.

[14] S. Smyshlyaev, N. Sullivan, and A. Melnikov, "[cfrg] results of the PAKE selection process," 2020. [Online]. Available: https://mailarchiv e.ietf.org/arch/msg/cfrg/LKbwodpa5yXo6VuNDU66vt Aca8/

[15] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in Proc. ACM CCS 2013, pp. 145–160.

[16] D. Wang, H. Cheng, P. Wang, J. Yan, and X. Huang, "A security analysis of honeywords," in Proc. NDSS 2018, pp. 1–18.

[17] Akshima, D. Chang, A. Goel, S. Mishra, and S. K. Sanadhya, "Generation of secure and reliable honeywords, preventing false detection," IEEE Trans. Depend. Secur. Comput., vol. 16, no. 5, pp. 757–769, 2019.