Poster: Secure Storage of Masked Passwords

Peter Mayer
SECUSO - Security, Usability, Society
Technische Universität Darmstadt
peter.mayer@secuso.org

Melanie Volkamer
SECUSO - Security, Usability, Society
Technische Universität Darmstadt
Privacy and Security Research Group
Karlstad University
melanie.volkamer@secuso.org

***Abstract*—Passwords are used to secure our daily digital lives. Some websites on the Internet use a technique called *masked passwords* to combat observation attacks (e.g. keyloggers or shoulder-surfers). This technique requires the users to enter only a random subset of the characters of their password, thus preventing an observer to capture the entire password. However, to verify the individual characters, the password must be stored as plaintext or reversibly encrypted on the website's servers. In this work, we present a method, which allows storing the users' passwords in hashed form when using masked passwords.**

1. Introduction

There are many ways a password can fall into the wrong hands. Among them are observation attacks like keyloggers and shoulder-surfing. Such attacks capture the input on the users' machines with the goal to extract passwords and other sensitive information (e.g. credit card details). Some websites employ a technique called *masked passwords* to counter this threat. These websites require their users to enter only a randomly chosen subset of the password's characters instead of the complete password. Fig. 1 depicts this procedure as employed on a banking website. Sometimes this technique is also used in two-factor schemes [1].

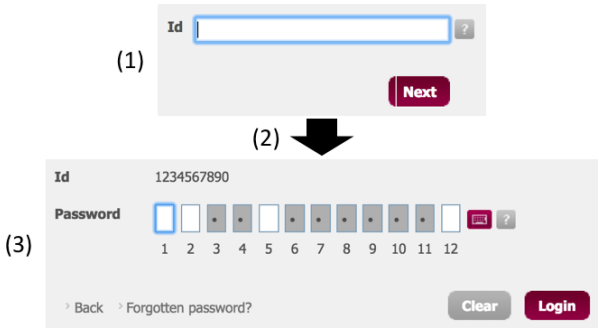

Figure 1. A typical login procedure using masked passwords: (1) the user specifies her/his user name (Id in this example), (2) the user clicks "Next" to proceed to the password entry, (3) randomly selected characters of the password have to be entered (in this example, the first, second, fifth, and twelfth character). Screenshots from https://aliorbank.pl/hades/do/Login.

While this technique can help to prevent leaking entire passwords to keyloggers or shoulder-surfers, it requires the password to be stored either in plain text or reversibly encrypted (i.e. encrypted with the key being available on the system). Thus, in case the servers of the website using masked passwords are compromised or a disgruntled administrative employee turns into an insider threat, the users' passwords are unprotected.

The goal of this work is to provide a method which allows storing only a hash[1] derived from the masked password, as is best practice for password storage. It is based on (t, n)-*threshold verification* [2], an approach using secret sharing to securely and efficiently store secrets in different portfolio authentication scenarios. We first describe the general ideas of portfolio authentication as well as t-n-threshold verification and then how we apply these ideas to masked passwords.

2. Portfolio Authentication

The basic idea of portfolio authentication is regarding the password as a set of elements (a portfolio). Masked passwords are one application of portfolio authentication based on text passwords. Another application of portfolio authentication is increasing the resistance of graphical recognition-based authentication schemes to naive shoulder-surfing attacks [3].

Generally speaking, in portfolio authentication a password P of length n is represented as $P = \{e_1, e_2, \ldots, e_n\}$. During each authentication attempt the user enters only a random subset $P' \subseteq P$ of these elements. We use the nomenclature of [2] and denote a subset P' as authorised if it has at least t elements, where the parameter t is set depending on the desired security properties. Thereby, the elements can have any form or even different forms (e.g. textual characters/strings, images, electronic certificates, etc.). Portfolio authentication usually follows a challenge-response procedure. To authenticate, the user is challenged by the system to provide a specific authorised subset.

1. Note that when we refer to hashes or hashing in this work, we always assume the use of a secure KDF (e.g. PBKDF2-SHA256, scrypt, Argon2, etc.) with appropriate configurations, salts and where appropriate peppers.

3. (t, n)-threshold Verification

One challenge with portfolio authentication is to verify in a secure and efficient manner whether the user's input represents an authorised subset P' of the password. This challenge is addressed by (t, n)-threshold verification. It uses Blakley (t, n)-threshold secret sharing [4] and key derivation functions (KDF) to derive the same secret from all authorised subsets. Blakley secret sharing is based on hyperplane geometry. Its internal structure is a linear system of equations $Mx = y$, where x is a secret t-dimensional point, y the n-dimensional vector of shares and M holds randomly chosen parameters that do not need to be kept secret. The details are left out here due to space constraints, but can be found in [2].

4. Applying (t, n)-threshold Verification to Masked Passwords

Fig. 2 depicts the procedure, when applying (t, n)-threshold verification to masked passwords. We first describe the enrolment and then the authentication and verification.

4.1. Enrolment

First, the textual password is split up into its characters $c_{i\in\{1,2,\ldots,|P|\}}$ and each character c_i is concatenated with its index i in the password to create the elements e_i. This step ensures, that in order to guess a share, not only the right character, but also its correct position in the password is required. Following the procedure of (t, n)-threshold verification, these elements are hashed to generate the shares y_i. Thereafter, a random secret x is chosen and its hash s stored for later verification. Finally, the matrix M is generated using the method described in [2] and stored alongside the hash s and the length of the password.

4.2. Authentication and Verification

To authenticate a user, the system generates a challenge by randomly selecting t positions $i \in \{1, 2, \ldots, n\}$, where $n = |P|$. This challenge is displayed to the user who has to enter the respective characters c_i of the password. Then, analogously to the enrolment, the pair of c_i (supplied by the user) and the respective i (supplied by the server) are concatenated and hashed on the server to derive the shares y'_i. These shares are then used to solve the linear system of equations $Mx' = y'$ for x'. In the last step, it is verified that the hash of x' matches the previously stored s. If the two hashes match, the user has entered the correct authorised subset and the authentication attempt is successful.

5. Conclusion

In this work, we apply (t, n)-threshold verification to masked passwords. Thereby, we increase the server-side strength of the technique. While the guessing resistance is determined by the parameter t (any attacker has to guess only t elements of the password correctly) the use of our proposal in conjunction with modern password hashing functions can greatly increase the cost for an attacker.

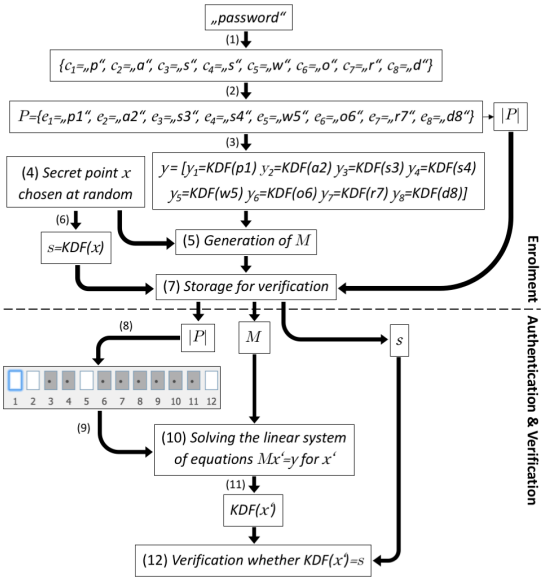

Figure 2. Enrolment and verification procedures, when applying (t, n)-threshold verification to masked passwords: (1) text password is split up into its characters, (2) characters are concatenated with their index, (3) each element is hashed to create the vector of shares y, (4) secret point x is chosen at random, (5) matrix M is generated as described in [2], (6) x is hashed to generate the verification information, (7) the elements required for verification are stored, (8) challenge is generated by choosing t indices in the password at random, (9) user responds to the challenge with the respective characters and the server concatenates the respective indices to the characters and hashes the pair to generate the shares y', (10) linear system of equations $Mx' = y'$ is solved for x', (11) x' is hashed, (12) verification whether KDF(x')=s is performed.

Acknowledgment

This work has been developed within the project 'KMU AWARE' which is funded by the German Federal Ministry for Economic Affairs and Energy. Moreover, it has been supported in part by the German Federal Ministry of Education and Research (BMBF) within CRISP (www.crisp-da.de/).

References

[1] N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Computers & Security*, vol. 30, no. 4, pp. 208–220, 2011.

[2] P. Mayer and M. Volkamer, "Secure and Efficient Key Derivation in Portfolio Authentication Schemes Using Blakley Secret Sharing," in *Annual Computer Security Applications Conference*, 2015, pp. 431–440.

[3] P. Dunphy, A. P. Heiner, and N. Asokan, "A closer look at recognition-based graphical passwords on mobile devices," in *SOUPS '10: Proceedings of the Sixth Symposium on Usable Privacy and Security*, 2010.

[4] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the national computer conference. Vol. 48*, 1979.