Security and Applications in Trusted Hardware

Portable Password Manager

João Silva Rui Fernandes Tiago Garcia

1 Problem

Current authentication schemes are based on something the user has (*e.g.* a smart card or a security token), something the user knows (*e.g.* a password or a PIN), or their physical characteristics (*e.g.* a fingerprint or iris pattern) [1], being password-based authentication the most widely used.

One problem with password-based authentication is that infrequently used passwords are easy to forget, and it is hard to remember secure passwords. In fact, to avoid memorizing numerous passwords, users tend to re-use the same password on different platforms [2]. As a result, discovering a user's password grants an attacker access to multiple of the user's systems. Some solutions address this issue by requiring users to remember only one password. In Section 2, we discuss two possible solutions to this problem, Single Sign On (SSO) and Password Managers.

2 Existing Solutions

In this section, we describe two different solutions to the problem of password re-use across multiple services, namely Single Sign On and Password Managers.

2.1 Single Sign On

User authentication can be delegated to an external service, known as Single Sign On (SSO). This service's purpose is to authenticate users and securely communicate successful authentication to the services the user wants to use. This way, the user only needs to remember the password he uses for the SSO service. Examples of SSO services include Shibboleth [3].

The main issue with this approach is their low portability and their availability, or lack thereof. For one thing, the service to which the user wants to authenticate must trust the SSO system. If this is not the case, the user still needs to remember a new password for the service. In other words, SSO systems only work with services that trust them.

On the other hand, a service may not trust in an SSO system due to the lack of availability guarantees. In other words, in case of a denial of service (DoS), the SSO may be disabled or rendered inaccessible, restricting user's ability to authenticate themselves.

2.2 Password Managers

Password managers are designed to store and provide access to a user's passwords. The user only needs to memorize a *master password* and store his passwords in the password manager. Examples of password managers include Bitwarden [4] and KeePass [5].

End-to-end encryption ensures that only the user has access to their master password and password vault – it cannot be accessed by the service provider or intercepted by hackers. However, since no one else can access the user's data, it is crucial that they remember the master password. Moreover, in order to safely store the passwords, symmetrical ciphers (*e.g.* AES) are used. These algorithms use either the master password, or a transformation of it such as its hash value, as a key to encrypt the stored data. However, other solutions exist. For example, by default, Firefox does not require a master password, using a default key to cipher passwords data base.

Finally, despite being more resistant to DoS attacks, password managers are still susceptible to brute-force attacks.

3 Proposed Solution

The major problems with the existing solutions discussed in Section 2 are their portability and availability issues, and the fact that they are not resistant to brute-force attacks, or both. In our proposed solution, this is circumvented by using the smart card as a means of securely storing a cryptographic key which is used to cipher externally stored passwords. As such, the card is responsible for generating a symmetric cryptographic key (*e.g.* AES) used to encrypt database containing the passwords, which will be stored in the cloud. Once the database is encrypted, we can assume that the passwords are stored securely. Whenever we want to access the password manager, the card reader will be responsible for storing and creating the passwords.

In this approach, the database could still be attacked, but the computational resources required to perform a successful attack are significantly greater given the size of the key. When it comes to access control, access to the card is protected by a PIN defined by the user, which cannot be brute-forced as the card will be locked after a certain number of unsuccessful attempts. In the case that the smart card supports multiple applications, no other application internal to the card can access the cryptographic key.

Finally, another factor that should be taken into account is that given the fact that a Java Card only persists data for about 10 years, a mechanism to export data to another card should be implemented.

3.1 Trust Model

In this work, we make the following assumptions:

- The card is tamper-proof. Thus, it is not possible for an attacker to access the cryptographic key without previous authentication.
- Only the user knows the smart card access PIN. This means that if somebody authenticates with the user PIN, he is the expected user.

References

[1] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, 2003.

[2] D. Florencio and C. Herley, "A large scale study of web password habits," Microsoft, Tech. Rep. MSR-TR-2006-166, November 2006. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-large-scale-study-of-web-password-habits/

[3] S. Suoranta, A. Tontti, J. Ruuskanen, and T. Aura, "Logout in single sign-on systems," in *Policies and Research in Identity Management*, S. Fischer-Hübner, E. de Leeuw, and C. Mitchell, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 147–160.

[4] Bitwarden Open Source Password Manager. https://bitwarden.com/. (Accessed on April 14, 2022).

[5] KeePass Password Safe. https://keepass.info/. (Accessed on April 14, 2022).