Comparing Password Management Software

Toward Usable and Secure Enterprise Authentication

Patricia Arias-Cabarcos and Andrés Marín, *University Carlos III of Madrid*
Diego Palacios, *Telefónica España, Madrid*
Florina Almenárez and Daniel Díaz-Sánchez, *University Carlos III of Madrid*

Password managers (PMs) automate password generation and login processes, but might not be secure or usable. The authors analyze the four most popular free PMs from both security and usability perspectives and make recommendations on enterprise PM selection.

Nowadays, we are constantly asked to prove our digital identities—for instance, when checking in for a flight, concluding a purchase via credit card, or logging onto a computer or secure website. This panorama does not change inside corporate walls, where employees must access a huge number of services daily, whether on-premises, cloud hosted, or run by partner companies. To face the mental burden of managing an increasing number of accounts and passwords, users commonly devise dangerous password strategies, such as reusing them or writing them down. These bad habits can cause harm to individuals, but the impact is much worse for companies, where a breach on an employee password could leak highly sensitive data and cause significant business and reputation damage.

Stories of such breaches have been front-page news in recent years,[1] so companies are taking measures, the implementation of strict password policies being the most popular. Generally, best practices and recommendation guidelines for password policies include restricting minimum length and character sets, imposing frequent changes, and prohibiting reuse.[2,3] Although these mechanisms should suggest an increase in security, studies unveil that the stricter the policies, the more users are prone to develop insecure practices to cope with the mental burden these policies can impose.[4]

Published by the IEEE Computer Society **1520-9202/16/$33.00 © 2016 IEEE**

Related Work in SSO Protocols and Smart Authentication Clients

The traditional silo-based identity scheme in which users are asked to set up one login per service has become inefficient for the current demands of cross-organizational cooperation, partnership, and collaboration. Several technological mechanisms are available for coping with the identity problems that arise in current distributed ecosystems, including unmanageability and scalability. We can classify these mechanisms into two categories.

Single sign-on (SSO) protocols are oriented to provide credential reuse functionality—that is, a user authenticates once and gains access to multiple sites without having to re-authenticate because information is transmitted between all the involved parties in a seamless manner. Based on this SSO concept, the past decade witnessed the development of several identity protocols built around Web services and beyond,[1] which include the Security Assertion Markup Language, OpenID Connect, and OAuth. The problem with SSO protocols is that the involved parties must create a circle of trust before authentication data can be shared. Thus, because it is impossible to have a unique federation in which all services trust one party for authentication, users end up being part of different federations and having to actively authenticate many times. An interesting approach here would be to combine and complement SSO technologies with mechanisms that enhance password-based authentication, but these solutions are still emerging.[2]

Another group of proposals is dedicated to alleviating the number of repetitive tasks required to manage multiple accounts by moving this load to devices that become *smart authentication clients*. The most salient work along this line is Pico,[3] which envisions users carrying a dedicated authentication device that stores all user cryptographic secrets or credentials and performs automatic logins to Web applications and other protected systems on the user's behalf based on a new protocol. But the most promising research lines are those centered on implicit authentication,[4] the foundations of which lie in determining user authentication by analyzing behavioral patterns. The main problem for short-term Pico adoption is that it requires important software modifications on services, aimed at totally replacing password-based authentication. In turn, implicit authentication allows for better session management by detecting user presence, but neither does it completely eliminate multiple password-based logins. Therefore, the most realistic strategy nowadays is the adoption of password managers (PMs), which are software programs that can automatically fill password-based login forms and can be used in conjunction with SSO protocols and implicit authentication solutions.

References

1. A. Pérez-Méndez et al., "Identity Federations Beyond the Web: A Survey," *IEEE Comm. Surveys & Tutorials*, vol. 16, no. 4, 2014, pp. 2125–2141.
2. P. Arias-Cabarcos et al., "Blended Identity: Pervasive IdM for Continuous Authentication," *IEEE Security & Privacy*, vol. 13, no. 3, 2015, pp. 32–39.
3. F. Stajano et al., "Bootstrapping Adoption of the Pico Password Replacement System," *Proc. 22nd Int'l Workshop Security Protocols XXII*, 2014, pp. 172–186.
4. K. Hassan, U. Hengartner, and D. Vogel, "Usability and Security Perceptions of Implicit Authentication: Convenient, Secure, Sometimes Annoying," *Proc. 11th Symp. Usable Privacy and Security*, 2015, pp. 225–239.

Some new authentication approaches have emerged to provide more security, such as those relying on single sign-on (SSO) protocols (see the "Related Work in SSO Protocols and Smart Authentication Clients" sidebar). However, because passwords continue to dominate the authentication landscape, password managers (PMs) are an alternative that can be immediately adopted without important infrastructure modifications. PMs automate password generation and login processes, which has direct economic benefits: it both reduces the cost of helpdesk calls related to password issues and minimizes the time employees dedicate to tasks that are not related to their work (login tasks are considered interruptions).

Considering the aforementioned benefits and recent studies recommending the use of these tools,[5] we wanted to evaluate the real security and usability levels of the most popular PMs. Our goal here is to perform a comparative analysis, based on expert knowledge and empirical studies with users, and also to derive new directions toward secure and usable authentication schemes for enterprise environments.

A PM is a software application that aids users in creating, storing, and organizing passwords. Basically, the application creates a local database to save all user passwords, which are encrypted to guarantee an adequate protection level. Thus, the user just needs to memorize a single, ideally very strong password, the *master key*, which grants user access to the entire password database. Additionally, many PMs include extra functionalities to improve usability, such as browser plugins to automatically fill login forms on the user's behalf, or synchronization of the password database among multiple user devices based on cloud server storage.

Among all existing free PM software applications, four stand out because of their popularity: Dashlane, KeePass, 1Password, and LastPass. Next, we analyze and compare these four applications.

Usability Analysis

For the usability study, we consider the set of evaluation criteria known as the 5 E's.[6] According to these criteria, usable software must be as follows:

- *Efficient*. Efficiency can be described as the speed (with accuracy) at which users can complete the tasks for which they use the product.
- *Effective*. Effectiveness is the completeness and accuracy with which users achieve specified goals. It is determined by looking at whether the user's goals were met successfully and whether all work is correct.
- *Engaging*. An interface is engaging if it is pleasant and satisfying to use. The visual design is the most obvious element of this characteristic. Equally important is the style of the interaction, which might range from a game-like simulation to a simple menu-command system.
- *Easy to learn*. An interface that is easy to learn allows users to build on their knowledge without deliberate effort.
- *Error tolerant*. An error-tolerant program is designed to prevent errors caused by interactions with users, and to help users in recovering from any errors that do occur.

Based on this set of criteria, we tested the usability of the four PMs empirically through questionnaires with real users. The questionnaires were structured around a set of tasks based on a pioneering study that analyzed two emerging PMs in 2006.[7] These tasks, which cover the main functionalities of a PM, are described as follows:

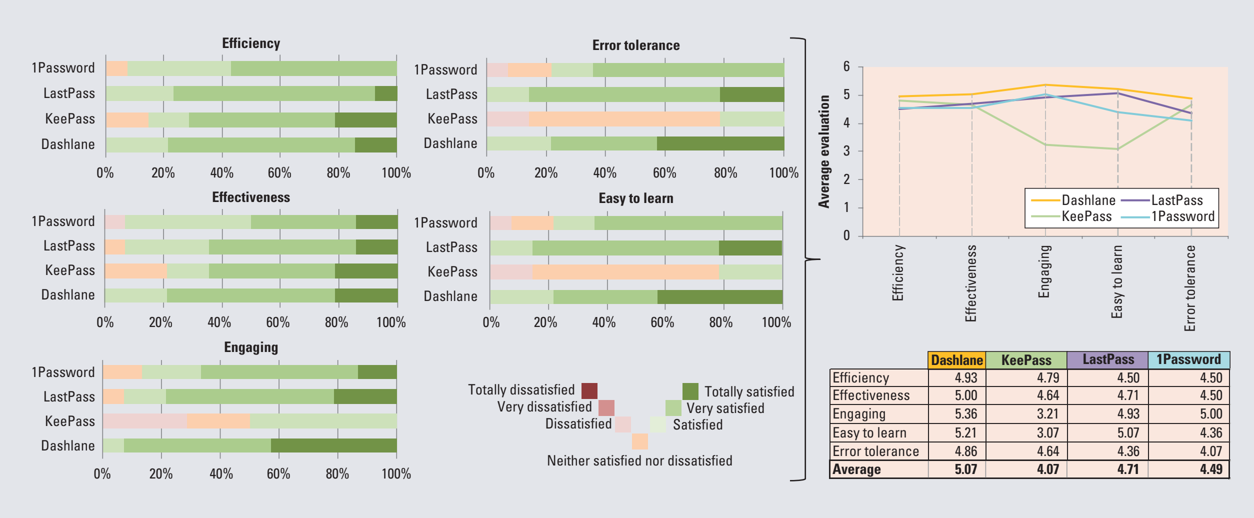
- Task 1—initialization: register with the PM software and store a couple of personal passwords associated with frequently visited websites.
- Task 2—login: start a session on a website for which the PM has a stored password.
- Task 3—remote login: start a session on a website for which the PM has a stored password using a different user device. This task serves to understand portability and synchronization between devices in the user's personal network.
- Task 4—password change: use the PM to change a personal password for a particular website.
- Task 5—log in with changed password: complete a login procedure after a password change.

Each respondent followed the set of tasks for each of the four PMs under study and gave an evaluation regarding the usability criteria. For rating each criterion, participants were presented with a seven-dimension Likert scale containing the following labels: totally satisfied (TS), very satisfied (VS), satisfied (S), neither satisfied nor dissatisfied (NSND), dissatisfied (D), very dissatisfied (VD), and totally dissatisfied (TD). Furthermore, scale dimensions were assigned a numerical value (a decreasing integer from 6 to 0) to compute an average quantitative punctuation for each PM.

The usability study involved 14 participants (a number high enough to identify most usability problems[8]), and data were collected and processed through online questionnaire tools. The evaluation results are illustrated in Figure 1a.

The study reveals that users rate all the PMs high on the scale in regard to efficiency, effectiveness, and error tolerance. Rating values are, in most cases, close to 5 out of 6, and no significant variance exists between the evaluations of the different managers. However, important differences arise when users rate the PMs according to the engaging and easy-to-learn features. For these latter aspects, we can see a difference of more than two points between the highest- and lowest-rated PM, with KeePass being the worst evaluated manager in both categories. However, we should highlight the generally positive nature of the evaluations: no PM gets an average evaluation under 3 points for any of the 5 E's.

From the usability analysis, we conclude that users accept PMs, so moving toward adopting this technology is a good choice for enhancing password-based authentication in the short term, especially in the enterprise arena. Nevertheless, it is crucial that the PM a corporation selects for employees is engaging and has an easy learning



	Dashlane	KeePass	LastPass	1Password
Efficiency	4.93	4.79	4.50	4.50
Effectiveness	5.00	4.64	4.71	4.50
Engaging	5.36	3.21	4.93	5.00
Easy to learn	5.21	3.07	5.07	4.36
Error tolerance	4.86	4.64	4.36	4.07
Average	**5.07**	**4.07**	**4.71**	**4.49**

Figure 1. Comparative analysis of the usability of password managers (PMs). For usability, Dashlane stands out over the other three PMs analyzed.

process to make the transition smoother. Figure 1a graphically summarizes the comparative results, where it can be seen that Dashlane stands out over the rest of the analyzed PMs.

Security Analysis

For the security study, we decided to use a heuristic evaluation approach and analyze the PMs based on how well they comply with widely accepted security guidelines. More specifically, we build on US National Institute of Standards and Technology (NIST) guidelines.[9,10] The set of criteria that guide the evaluation process are based on the general PM architecture, and take into account weak points where security must be enforced:

- *Security of the master key*. Because the master key gives access to all user passwords stored in the PM, it must be strong enough to prevent leaks and attacks. We check whether the PMs impose a mandatory minimum length for the master key; whether the PMs force users to create strong master keys by applying a secure policy; and whether the master key is securely stored.
- *Security of the credentials database*. This is the valuable asset protected by the PM. We check the strength of the algorithm used for encrypting the database; whether the PMs give quantitative or qualitative feedback on the stored passwords' security level; whether they can automatically generate strong passwords on users' behalf; whether multifactor authentication is permitted; and whether the PMs enable scheduling password validity periods and generate new passwords when the configured passwords expire.
- *Security of communications*. Security must be guaranteed when communicating credentials between PMs and applications using passwords. We evaluate the communication security between PMs and external cloud servers, and the communication security between PMs and browser plugins.

Table 1 summarizes the results of the security study, which are also graphically represented in Figure 2. To compare the different PMs, we assigned quantitative values to each criterion. The majority of the criteria are evaluated using a binary scale, so they are given a value equal to 1 if the security feature is supported and 0 otherwise. For those criteria in which a security algorithm in use must be specified, we use a continuous scale [0–1] and assign the value depending on the algorithm strength according to well-established security recommendations.[9,10]

According to the information gathered during the study, the security of the master key is protected in all the analyzed PMs in relation to criterion SM#3: the master key is never stored, either locally or in cloud servers. Instead, a password-based key derivation function is used, which consists of applying a pseudorandom function (hash, cipher, or keyed-hash message authentication code) to the master key along with a salt value and repeating the process many times to obtain a derived key to be used for the rest of the PM operations. The added computational work makes password cracking much more

Table 1. Security evaluation of password managers, including quantitative value (inside parentheses).

Security goal	Criterion	Dashlane	KeePass	LastPass	1Password
Security of the master key (SM)	SM#1: minimum mandatory length	Yes (1)	No (0)	No (0)	No (0)
	SM#2: user must apply policy for strong master key	Yes (1)	No (0)	No (0)	No (0)
	SM#3: master key securely stored	Yes (1)	Yes (1)	Yes (1)	Yes (1)
Security of the credentials database (SDDBB)	SBBDD#1: algorithm used for database encryption	AES-256 (1)	AES/Twofish, 256-bit key (1)	AES-256 (1)	AES-256 (1)
	SBBDD#2: the PM gives feedback on the security level of the stored passwords	Yes (1)	Yes (1)	Yes (1)	Yes (1)
	SBBDD#3: automatic generation of strong passwords on the users' behalf	Yes (1)	Yes (1)	Yes (1)	Yes (1)
	SBBDD#4: multifactor authentication	Yes (1)	Yes (1)	Yes (1)	Yes* (1)
	SBBDD#5: can schedule password validity periods and generate new passwords upon expiration	No (0)	Yes (1)	Yes (1)	No (0)
Security of communications (SC)	SC#1: security of the communication algorithm between the PM and external servers	HTTPS with Transport Layer Security (TLS) v1.2, AES-128, and Ephemeral Diffie-Hellman (1)	Not applicable (NA)	HTTPS with TLS v1.2, AES-128, and Ephemeral Diffie Hellman (1)	Not applicable (NA)
	SC#2: security of the communication algorithm between the PM and the browser plugin	AES-256 (1)	Not applicable (NA)	Not applicable (NA)	Not applicable (NA)

** only for MAC OS and iOS; AES: Advanced Encryption Standard*

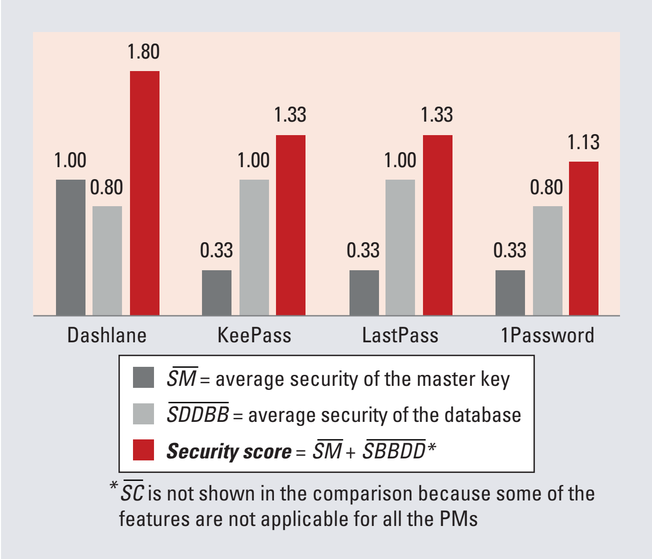
difficult, and is known as *key stretching*. However, only Dashlane provides additional security measures for securing the master key. It forces the user to choose a password with a minimum length of eight characters, including an uppercase letter, a lowercase letter, and a number. This policy leads to an increased entropy value for the selected word, so security against brute force attacks is enhanced.

As regards the security level of the credentials database, all the PMs provide comparable solutions, supporting the Advanced Encryption Standard (AES) with a 256-bit key as the cipher, which are the algorithm and key size currently recommended for the highest security. The only difference in this security category lies in the support of automatic scheduling of passwords, which only KeePass and LastPass provide. This feature is important because it facilitates an easy way to limit the validity of passwords, and frequent password changes lead to better security because exposure is for a shorter time period.

Finally, the security level of communication exposes some differences. On one hand, only Dashlane and LastPass provide in-cloud storage.

For these two PMs, communication with storage servers is secured using HTTPS with cryptographic algorithms currently considered as highly secure. More specifically, the protocol version is Transport Layer Security (TLS) v1.2 used with AES for confidentiality combined with Ephemeral Diffie-Hellman for key exchange between client and server. Such ciphersuites provide perfect forward secrecy, ensuring long-term confidentiality of the session—that is, the compromise of a long-term private key used in deriving a session key subsequent to the derivation does not compromise the session key. On the other hand, although all the PMs support browser plugins, only Dashlane provides public documentation about the security of the communication between the plugin and the application (based on AES-256).

Figure 2 shows the security comparison and the total quantitative security evaluation given to each PM. Note that we have not included the security of communications dimension in the comparison because some features were not supported by all the applications, or there was missing information. However, because cloud storage and plugins are additional functionalities that are not required for a PM to be fully functional, we can look at this comparison as a security evaluation of core PM functionality. As we can see in Figure 2, the best security score is obtained by Dashlane, with a 1.8 out of 2. Nevertheless, it is important to highlight that all the evaluated PMs demonstrate good general security features, the lowest security score being 1.13 points out of 2.



*$\overline{SC}$ is not shown in the comparison because some of the features are not applicable for all the PMs

Figure 2. Comparative analysis of the security of password managers. Dashlane achieves the best security score.

We analyzed and compared the most popular PMs with free versions in terms of usability and security and observed two interesting facts: usability is perceived in a positive way by users, and all the analyzed PMs demonstrate architectures that are theoretically secure according to current best-practice recommendations. More specifically, Dashlane offers the best combination of security and usability characteristics.

Related work has shown a user perception of poorer usability,[7] probably because PM applications were still emerging. Nevertheless, our study reveals that PMs have reached a level of maturity that is adequate for their successful mass adoption in both personal and corporate environments. In regard to the latter scenario, there are some future research lines that would boost PM adoption—namely, investigating how to automatically integrate and enforce company security policies with PMs; combining PMs with SSO technologies used inside corporate environments, such as the Security Assertion Markup Language and OAuth; and researching usability improvements by merging implicit authentication with PMs as substitution mechanisms for the master key. Finally, as future work, we would like to develop automatic testing tools that evaluate the security of PMs empirically to complement the theoretical analysis we provide here and cover specific attack scenarios and vulnerabilities, as described elsewhere.[11]

Acknowledgments

This work was supported partly by the Spanish Ministry of Science and Innovation under project INRISCO (TEC2014-54335-C4-2-R), and by the state of Madrid (Spain) under contract number S2013/ICE-2715 (e-Madrid).

References

1. D. Mirante and J. Cappos, *Understanding Password Database Compromises*, tech. report TR-CSE-2013-02, Dept. Computer Science and Eng., Polytechnic Inst. of New York Univ., 2013.
2. K. Scarfone and M. Souppaya, “Guide to Enterprise Password Management (draft),” NIST special publication 800-118, 2009.
3. *Password Protection Policy*, SANS Inst. report, 2014.

4. P.G. Inglesant and M.A. Sasse, "The True Cost of Unusable Password Policies: Password Use in the Wild," *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2010, pp. 383–392.
5. I. Ion, R. Reeder, and S. Consolvo, "'... No One Can Hack My Mind': Comparing Expert and Non-Expert Security Practices," *Proc. 11th Symp. Usable Privacy and Security*, 2015, pp. 327–346.
6. W. Quesenbery, "What Does Usability Mean: Looking Beyond Ease of Use," *Proc. Ann. Conf. Society for Technical Comm.*, vol. 48, 2001, pp. 432–436.
7. S. Chiasson, P.C. van Oorschot, and R. Biddle, "A Usability Study and Critique of Two Password Managers," *Proc. 15th Conf. Usenix Security Symp.*, vol. 15, 2006, article no. 1.
8. R.A. Virzi, "Refining the Test Phase of Usability Evaluation: How Many Subjects is Enough?" *Human Factors*, vol. 34, no. 4, 1992, pp. 457–468.
9. E. Barker, *Recommendation for Key Management*, NIST special publication 800-57, part 1, rev. 4, 2016.
10. T. Polk, K. McKay, and S. Chokhani, Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations, NIST special publication 800–52, 2014.
11. Z. Li et al., "The Emperor's New Password Manager: Security Analysis of Web-Based Password Managers," *Proc. 23rd Conf. Usenix Security Symp.* (Usenix Security), 2014.

***Patricia Arias-Cabarcos** is a researcher at the University Carlos III of Madrid (UC3M). Her research interests include identity management, trust and reputation models, and risk assessment. Arias-Cabarcos received a PhD in telematics engineering from UC3M. Contact her at ariasp@it.uc3m.es.*

***Andrés Marín** is an associate professor at the University Carlos III of Madrid (UC3M). His research interests include ubiquitous computing, limited devices, and trust and security in next-generation networks. Marín received a PhD in telecommunication engineering from the Technical University of Madrid. Contact him at amarin@it.uc3m.es.*

***Diego Palacios** is a cybersecurity analyst at Telefónica. His research interests include cybersecurity, security incidents, and vulnerability management. Palacios received an MSc in cybersecurity from the University Carlos III of Madrid. Contact him at diego@diego.hk.*

***Florina Almenárez** is an associate professor at the University Carlos III of Madrid (UC3M). Her research interests include trust and reputation management models, identity management, and security architectures in ubiquitous computing. Almenárez received a PhD in telematics engineering from UC3M. Contact her at florina@it.uc3m.es.*

***Daniel Díaz-Sánchez** is an associate professor at the University Carlos III of Madrid (UC3M). His research interests include distributed authentication, authorization, and content protection. Díaz-Sánchez received a PhD in telematics engineering from UC3M. Contact him at dds@it.uc3m.es.*