Computer Science Review 33 (2019) 69–90

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Usability, security and trust in password managers: A quest for user-centric properties and features[☆]

Sunil Chaudhary[a,b], Tiina Schafeitel-Tähtinen[a,*], Marko Helenius[a], Eleni Berki[c]

[a] Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland
[b] Deerwalk Institute of Technology, Kathmandu, Nepal
[c] Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

ARTICLE INFO

Article history:
Received 6 September 2018
Received in revised form 7 February 2019
Accepted 14 March 2019
Available online 8 June 2019

Keywords:
Password manager
Cognition
Usability
User experience
Security
Trust
Systematic literature review

ABSTRACT

A password manager stores and handles users' passwords from different services. This relieves the users from constantly remembering and recalling many different login credentials. However, because of the poor usability and limited user experience of password managers, users find it difficult to perform basic actions, such as a safe login. Unavoidably, the password manager holds the login credentials of many online services; as a result, it becomes a desired target for online attacks. This results in compromised security, which users often consider as an inevitable condition that must be accepted. Many studies analysed the usability and security of various password managers. Their research findings, though important, are rather incomprehensible to designers of password managers, because they are limited to particular properties or specific applications and they, often, are contradictory. Hence, we focus on investigating properties and features that can elevate the usability, security, and trustworthiness of password managers, aiming at providing practical, simple, and useful guidelines for building a useable password manager. We performed a systematic literature review, in which we selected thirty-two articles with coherent outcomes associated with usability and security. From these outcomes, we deduced and present meaningful suggestions for realising a useable, secure and trustworthy password manager.

Contents

1. Introduction 70
 1.1. Research objectives 71
 1.2. Paper outline 72
2. Research methodology: Systematic literature review 72
3. Literature review 72
 3.1. Comparative evaluation of usability and/or security of different PMs 72
 3.2. Proposed PM design/architecture and its evaluation 73
 3.3. Summary of the problems found in research literature 74
4. Features and properties of a user-friendly and useable PM 77
 4.1. Compatibility issues 77
  4.1.1. Follow practices in web-design and OS 77
  4.1.2. Facilitate transportability in PM 78
  4.1.3. Support for heterogeneous types of login 78
 4.2. Learnability issues 79
  4.2.1. Narrow semantic gaps between PMs and their users 79
  4.2.2. Increase learnability of the system 79
  4.2.3. Improve the memorability of the master password 79

☆ This work has financially been supported by TEKES (The National Research Agency of Finland) and is of the DIGILE Cyber Trust research program. The content is a deliverable of the WP1: Secure Services.

* Correspondence to: Tampere University, Hervanta Campus, Korkeakoulunkatu 1, 33720 Tampere, Finland.
E-mail addresses: sunil.chaudhary@deerwalk.edu.np (S. Chaudhary), tiina.schafeitel-tahtinen@tuni.fi (T. Schafeitel-Tähtinen), marko.helenius@tuni.fi (M. Helenius), eleni.e.berki@jyu.fi (E. Berki).

https://doi.org/10.1016/j.cosrev.2019.03.002
1574-0137/© 2019 Elsevier Inc. All rights reserved.

4.3. Design constraints ... 80
4.3.1. Simplify the deployability of the PM ... 80
4.4. User interface and performance issues ... 80
4.4.1. Include step-by-step migration of accounts ... 80
4.4.2. Use passive dialog box for user's confirmation ... 80
4.4.3. Include measures to improve the performance of PMs ... 80
4.5. Limited/error-prone password or master password control ... 80
4.5.1. Include necessary features and functionalities ... 80
4.5.2. Design a useable password generator ... 80
4.5.3. Include strength checker for passwords ... 81
4.5.4. Include recovery and reset features for the password and master password ... 81
4.6. Master password as single point of failure ... 81
4.6.1. Make mandatory master password ... 81
4.6.2. Not remembering and not saving the master password ... 81
4.6.3. Protect the master password ... 81
4.7. Weak user data security and privacy ... 82
4.7.1. Lock password database when not in use ... 82
4.8. PM Vendor/third-party dependence ... 82
4.8.1. Allow users to select the storage type by providing both the options (local and cloud storages) ... 82
4.8.2. Establish and build trust towards PM's vendors ... 83
4.9. Vulnerabilities to various attacks ... 83
4.9.1. Protection of the master password against phishing attacks ... 83
4.9.2. Only allow manual opening of a new tab ... 84
4.10. Erroneous security mechanism/design ... 84
4.10.1. Implement mandatory but secure auto-fill ... 84
4.10.2. Store passwords in encrypted form and use optimal iteration for key derivation ... 84
4.11. Underutilised security resources ... 84
4.11.1. Tap underutilised resources without affecting portability ... 84
5. Conclusions and future enhancements ... 85
Declaration of competing interest ... 86
Appendix. Research methodology: Systematic literature review ... 86
A.1. Literature sources and study selection ... 86
A.2. Data extraction and analysis ... 87
References ... 87

1. Introduction

People use many different kinds of online and offline services. In practice the most prevalent authentication mechanism in these services for authenticating users and protecting the data is the username–password pair [1]. The username provides identification of the user and the password verifies the user's legitimacy. Within the username–password authentication process, several types of passwords are available in practice. For the latter, well known examples in use are: *one-word password, PIN code, paraphrase password, cognitive password, associative password, gesture password, image password,* and *image-gesture password*. Passwords can be used in combination with other authentication methods (e.g., biometrics or e-token) during *multi-factor authentication*. Even when exclusively using passwords for authentication purposes, there can be *single-level* or *multi-level authentication*. Further, passwords can be *machine-generated* or *user-generated* (which is popular because of usability). Moreover, passwords can be *one-time passwords* or *time-based one-time passwords*, wherein the *password* remains valid only for a single login session or transaction; this is found to be in common use in online banking services. Irrespective of these password types, in order to afford improved protection of information for every service, it is highly recommended for users to select distinct and strong passwords, such as those with *high entropy* [2–4] or distinct passwords containing a *maximum guess number* [5–9].

As alternatives for the username–password pair, some may advocate implementation and dependence on other authentication methods, such as biometrics, smart cards/tokens and the list can go on. However, these authentication methods have their own functioning environments, limitations, and vulnerabilities [10–12]. More importantly, security comes at a cost. Hence, selecting a suitable authentication method should take into account the following factors: (i) the nature of the security concerns, i.e., the vulnerability of the information to be protected and the potential risks to the information, (ii) the end-users of the system, (iii) the method's compatibility with work practices and other aspects of the system, and (iv) the cost-effectiveness of the applied authentication method. Unlike other authentication methods (e.g., biometrics and e-tokens), no additional hardware is necessary for the use of username–password pairs, which makes it relatively economical and simple to implement this method. Furthermore, there is always room to strengthen security by choosing as strong a password as is required, depending on the context of the usage. In addition to password, strengthening the username/ user ID and keeping it secret [13] can also contribute to the overall security of authentication credentials. Apart from that, introducing "Three strikes" rule or a delay for every wrong entry [14], which is widely in practice for authentication can help in strengthening the security. For example, the recommended username and password for e-banking can and should be of higher entropy than that for social networking in social media. Further, the username–password pair offers a number of other benefits as well [*refer* [11,15]], which makes this method difficult to replace with the aforementioned alternative authentication methods.

As a matter of fact, none of the existing authentication methods have been successful in fully replacing passwords as the mainstream method for authentication. Apparently, the reason for this is that not a single alternative technology possesses the same combination of security, usability, and economy offered by passwords [*refer* [1,11]]. People still rely highly on passwords for authentication purposes and will undoubtedly continue depending on passwords for authentication purposes in the future, as long as a complete and trustworthy replacement is unavailable. At present, numerous researchers and organisations are experimenting with new user authentication alternatives in order to replace passwords [16,17]; this is a promising step towards strengthening security. Current research efforts may result in new authentication technologies that will scale up security, be cost-effective to implement, and at the same time be available, easy and convenient to use. However, some questions one may still ask include: *"Will they be able to kill passwords at any time soon, as claimed?"* and *"Will it be easy and fast to convince password users, that is both the owners of information systems implementing passwords and the end users using those systems, to migrate and embrace a new authentication technology?"* Regardless of any answers, one should accept that until a proper replacement for passwords is available and despite all passwords limitations, passwords will continue being essential and necessary for authentication purposes. For this reason, the use of passwords should still be improved in order to make them more useable, secure and, thus, trustworthy. For these reasons, in this research study, the authors have focused only on the username–password pair.

One of the major reasons for the aforementioned discussion of different password types and authentication methods is to relieve current users' cognitive load and to avoid future users' *cognitive overload*. In practice, memorising login credentials for a large number of services has proven to be a cognitively challenging task; on average, a user has around 25 password-protected accounts [18]. Meanwhile, security experts suggest various policies and rules for constructing a strong password. Yet, regrettably, those rules and policies are often unrealistic, time-consuming, and not worth complying with—and thus, often inoperative [19]. Similarly, it is recommended, as a security measure, that users routinely or at fixed intervals change their passwords. Such measures put extra pressure on users and often force them to choose easy-to-guess or weak passwords [20–22]. Unavoidably, there arises a conflict between IT/security professionals' recommended policies and users' actual needs. Attempting to adhere to standard policies and rules results in users emotional pressure, slowing them down from performing their primary work tasks [23]. To ease the login process and avoid forgetting their passwords, people are often led to insecure practices, such as constructing easy-to-crack passwords, writing down passwords, or reusing the same password for various different accounts [18,24]. Hence, to mitigate these insecure practices and at the same time help people to remember as many passwords as possible in a secure and manageable way, a password manager (PM) can be an effective, useable, trustworthy, and highly regarded solution.

Inevitably, as an alternative to a PM, some may propose the adoption of a *single sign-on* (SSO) mechanism, that is allowing users to gain access to multiple related but independent systems by logging in with a single username and password [25]. However, SSO mechanisms also have several limitations [*refer* [26]], so SSOs may be inappropriate when a person wants to share access only to an individual service, but not to all other related services.

A password manager (PM) is a software application that helps a user to store and organise passwords. A PM usually stores a large collection of passwords in an encrypted form, ideally protected by a strong master password. However, a PM can also operate without a master password's protection, likewise a built-in feature called *'auto sign-in'* for web browsers. The main purpose of a PM is to relieve its user from the need to memorise a large number of passwords, as it is simple enough to remember only one master password or no password in the case of *'auto sign-in'*. Password managers (PMs) range from independent mobile applications to browser extensions. Some PMs store passwords on the user's devices (offline), whereas others store passwords in the cloud (online), or provide the capability for both offline and online storage. Karole, Saxena, and Christin [27] classified PMs into three categories: (i) *desktop manager*, which stores passwords on the user's desktop; (ii) *online manager*, which uses third-party servers to store passwords; and (iii) *portable manager*, which stores the users' passwords in their portable devices, such as mobile phones or USB devices. Similarly, PMs can utilise (a) the *wallet approach*, in which a master password is used to encrypt a file of site-specific passwords; or (b) *the hashing approach*, wherein the master password is combined with site-specific information to generate site-specific passwords [28]. Depending on its type, a PM can also offer other features such as form-fillers, password-generators, data synchronisation, and safe password suggestions.

1.1. Research objectives

Although a PM relieves users from the hectic job of remembering a large number of passwords, it is not an ideal solution, since the PM users will still have usability and security concerns. These concerns seem to be the reasons why PMs have not been widely adopted by password users. Instead, most people still prefer to depend on their memory to recall and manage their passwords [29–32].

An analysis of 169 accounts was performed by Hayashi and Hong [29], which revealed that approximately 60.3% of the accounts do not make any use of password aids (e.g. PM) for storing and managing their passwords. Likewise, a survey conducted by the Password Boss with 2,030 adults from the United States of America (USA) revealed that 63% of the participants use their memory to manage their passwords [33]. Another similar survey carried out by the RoboForm with 1000 participants from the USA and the United Kingdom (UK) revealed another, rather more discouraging, result: only 8% of the participants reported using a PM so that they could have a strong and unique password for every website [31]. Two of the chief reasons that can encourage the potential password users to use a PM are the PM's security and usability properties [34,35].

Many of the studies considered in this systematic literature review evaluated the security and usability of PMs from different perspectives. However, their evaluations are still limited, because they focus on particular properties and features in specific application(s) and, thus, fail to take into account several polymorphic and significant software quality features and aspects that are essential to design and fully develop a *useable and secure PM*. Although these studies are valuable for the handling and exposure of the *specific situations* they consider, they are still not of much help to software practitioners and software designers interested in developing a PM that would more readily be used by people for the management of their passwords. Moreover, some studies are inconsistent and, in many cases, the research results contradict each other. As a consequence, someone interested in designing a PM may get confused when deciding which properties to adhere to or which features to build and integrate into their applications. Further, it is equally important to uncover the area(s) where more research is needed in order to make a useable and trustworthy PM. These are the principal reasons which prompted us as researchers to carry out our own investigation through this research study, by utilising a systematic literature review.

In this study, we included several past studies and performed a systematic literature review in order to determine the properties and features that can improve usability and user experiences in a PM. We also took into consideration the security of a PM, as usability and security are correlated, especially in software design for security purposes, as is the case with a PM. If the usability is associated with poor security, it will obstruct the PM users' primary tasks and also increase the time and effort required for them to log in to any service using a PM. This will de-motivate people from adopting a PM [23]; and even those who do *use it* will have to pay attention to unnecessary activities rather than the important ones.

Human attention is a scarce but useful resource [36], and it must be utilised for the primary task(s) and decisions related to it/them. An unmotivated PM user with divided attention has a greater likelihood of bypassing the complex security functions and making dangerous errors [37,38]. As security itself is a wide domain, it is best addressed in a separate study of its own; herein, we have essentially considered only those security and failure cases that may affect the usability and user experience of a PM. The final outcomes presented here were deduced mainly based on recommendations collected from the most relevant reviewed articles, according to the inclusion/exclusion criteria we set for the systematic literature review. We particularly aimed at making these results more practical and more comprehensible with valid, cross-checked and cross-referenced, logical explanations. The main outcomes comprise a consistent set of software quality features for those IT professionals who want to design and implement a PM with a particular focus on *both* security and usability.

Consequently, the main objective of the exposure of the research study's method and outcomes in this paper is to determine the properties and features that can elevate the usability and security in PMs, with the prospect of building a useable and trustworthy PM. In order to accomplish this objective, the authors answer to the following research questions:

- What are the *usability and security problems* in existing PMs?
- What *measures* and *suggestions* currently exist or have been proposed to mitigate those problems?

1.2. Paper outline

The rest of the paper is structured and organised as follows. Section 2 describes the research methodology; Section 3 contains the literature review of the past studies, that is, their (i) research questions, (ii) research methods, and (iii) research outcomes. Section 4 presents our findings as deduced from the literature review; and Section 5 concludes the work and mentions future enhancements.

2. Research methodology: Systematic literature review

Our systematic literature review took place during 6/2015–10/2015, considering the relevant published research papers until 9/2015. The selected sources for relevant literature searches were: IEEE Xplore Digital Library, ACM Digital Library, Springer Link, and ELSEVIER ScienceDirect. The selection of databases is engineering focused. This is a deliberate choice and serves the main goal of the research study, which is to provide useful information regarding properties and socio-technical features especially for the developers of password managers.

The following criteria were applied for selecting the most relevant research articles:

- The inclusion of articles evaluating usability and/or security of any PM, or proposing a solution for some usability or security problem in PM.
- The exclusion of overlapping journal or conference articles, or articles containing only one or two paragraphs on PM.

The number of relevant articles for the systematic literature review was 32, as listed in Tables 1 and 2 along with their subject of emphasis. It should be noted that the aforementioned 32 articles will serve as the primary source of knowledge herein, but other user-centric literature (including cognitive science, software quality engineering, and other) are also utilised. Throughout the research, the selected literature is reflected on against other user-centric literature not directly included in the reviewed literature. The engineering-focused perspective of this research and related reflections to other user-centric literature is a valuable contribution of this study, because, instead of just listing the properties and features, we try to synthesise engineering-focused perspective with user-centric one and to see why the features and properties suggested would be important for elevating both usability and security as built-in software quality features in password managers.

From the selected articles, we extracted the following data:

- Study description (issues studied, research study methodology);
- PM usability and/or security problems on which the study focuses; and
- Measures presented or suggested in the study to mitigate the particular problems related to usability and security.

Based on the extracted data and information, we formed a knowledge typology of 11 new knowledge categories for the different types of problems that we encountered in the PM research literature (see Sections 3.1–3.3). The authors scrutinised the new knowledge distilled and illustrated in the typology and subsequently summarised and condensed mitigation measures proposed in the literature for the particular PM problems considered in the context of this study. These final suggestions are expressed in terms of the quality features and properties that are indispensable for a PM to have/be in order to improve its usability (see Section 4).

Detailed description of the research methodology is presented in Appendix.

3. Literature review

In this research it was found that the researchers of the selected articles had either performed a comparative evaluation of different PMs or presented a new PM design or architecture. Hence, in a broad sense, the selected articles could be categorised as contributors into the following groupings:

- Comparative evaluation of usability and/or security of different PMs; and
- Proposed PM design/architecture and its evaluation.

We, in turn, have followed these concepts and divided our comparative literature reviews into these two headings. All the associated papers are presented in tabular form and are chronologically arranged in Tables 1 and 2. We have also listed a brief description of the studies, the problems identified by the researchers, and their notable suggested measures in Table 1, next.

Most of the selected articles accentuate prime security issues, but only a few of those draw attention to usability issues. As a result of the nature of PM applications, those articles that emphasise usability issues also consider a security perspective to some extent, since this generally makes the user's prospects more noticeable. Further, the evaluation methods followed in the articles, vary a lot. For instance, in some articles, the researchers performed a specific user study to evaluate usability whereas other researchers utilised cognitive walkthroughs.

3.1. Comparative evaluation of usability and/or security of different PMs

The authors of the articles in this category have evaluated usability and/or security of multiple PMs while comparing and contrasting formats. For instance, some studies, like those of Bicakci et al. [39] and Zhao and Yue [40], employed meta-knowledge derived from an evaluation of existing PMs in their own PM design.

Table 1
Summary of the articles that perform comparative evaluation of security and/or usability in PMs.

Authors	Descriptions of Study	Problems	Measures
Chiasson et al. (2006)	Evaluated PwdHash and Password Multiplier; conducted usability testing with 27 users; used 5-point Likert scale questionnaire and observation to collect participants' response	1. Semantic gap 2. Lack of trust 3. Inconsistent practices in web service development	1. Understand and include user's intentions and expectations 2. Suitable interaction between user and system 3. Short, accurate, understandable, and visible instructions/cues/feedback
Bicakci et al. (2009)	Designed and evaluated GPEX against PwdHash; conducted experiment in laboratory with 20 users to evaluate effectiveness, efficiency, user-satisfaction and security of their design. Used 5-point Likert scale questionnaire to collect participants' response	1. Mental model gap 2. Users are vulnerable to hotspot attack in click-based graphical password	1. Visual password interface 2. Gridded click-based graphical password
Karole et al. (2010)	Evaluated KeePassMobile and LastPass, RoboForm2Go; Conducted usability testing with 20 users; used questionnaire consisting 5-point Likert scale, open-ended and multiple choice questions to collect participants' response	1. Lack of trust Offline PMs are preferred over online PMs though the latter has better usability	1. Offline PMs, but with improved usability
Bicakci et al. (2011)	Evaluated Internet Explorer (IE) and Mozilla Firefox; conducted user study in semi-controlled lab experiment with 27 users; used questionnaire to collect participants' response	1. Intrusive dialog box increases the tendency to save password	1. Passive communication for password auto-complete to get users' confirmation
Gasti and Rasmussen (2012)	Evaluated Google Chrome, Mozilla Firefox, IE, 1Password, KeePass 1.x, KeePass 2.x, KeePassDroid, PINs, Password Safe, Password Gorilla, Roboform; performed mathematical analysis to analyse security of the selected tools' database	1. PM's databases storing users' passwords are vulnerable to attacks	1. Measures which can provide secrecy and data authenticity in the database
Bhargavan Delignat-Lavaud (2012)	Evaluated LastPass, PassPack, RoboForm, 1Password, Clipperz; performed security analysis using attacks that exploit standard web-application vulnerabilities to expose the flaws in the selected tools.	1. Lacks integrity protection for the cipher text 2. Metadata are unencrypted 3. Vulnerable to host attacks and tampering	1. Integrity checking 2. A scheme with authenticated encryption, includes associated data
Zhao et al. (2013)	Evaluated LastPass and RoboForm; performed security/ vulnerability analysis using threat model and formal arguments	1. Master password is optional 2. Inflexibility in resetting the master password 3. Remember and save the master password 4. No strength checking for the master password	1. Mandatory high-entropy master password 2. Do not remember and save the master password 3. Support resetting the master password 4. Proactive strength checker for the master password

(*continued on next page*)

3.2. *Proposed PM design/architecture and its evaluation*

The articles proposing a software/system design for a new PM are primarily found to focus on fixing security issues. For instance, these scientific works typically targeted all three mechanisms used for authentication: *what you know*, *what you have*, and *what you are* [(*see* e.g. [11]]. Some of these articles even proposed the use of a *two-factor authentication* procedure for the master password. As the research interests of this study are usability and security targeted to enhance the user experience, herein there is discussion only on the associated security factors. Table 2 features brief descriptions of these studies, including the highlighted problems and recognised measures.

Table 1 (*continued*).

Authors	Descriptions of Study	Problems	Measures
Fahl et al. (2013)	Analysed 21 PMs for Android platform (name not mentioned); performed manual and static code analysis for security vulnerabilities; proposed USecPassBoard for the Android platform	1. Lack of trust 2. Embedded PMs are limited to a single browser 3. Weak security for the users' data 4. Constraints imposed because of the mobile platform	1. Correctly implement key derivation or encryption 2. Ensure data confidentiality 3. Secure the OS's clipboard
Silver et al. (2014)	Evaluated desktop browser PMs (Google Chrome, Mozilla Firefox, Apple Safari, Microsoft IE), third party PMs (Keeper, 1Password, LastPass, KeePass), iOS PMs (Mobile Safari, Google Chrome, 1Password, LastPass Tab), Android PMs (Android browser, Chrome); conducted a survey of auto-fill policies, and also conducted security analysis for auto-fill policies	1. Treat equally to page using HTTP and HTTPS protocols 2. Vulnerable to rogue Wi-Fi and other attacks	1. Enforce user interaction before auto-filling a form 2. Show the domain name being auto-filled to the user before filling occurs 3. Always submit information over HTTPS even when the page is loaded over HTTP
Ziegler et al. (2014)	Evaluated eight PMs for Android platform (anonymous); performed manual examination of key derivation functions	1. Vulnerable to brute-force and other types of attacks 2. Error or weak design choices in the implementation of key derivation functions	1. Salt to increase the entropy of the password 2. Correctly implement when using established APIs (eliminate development errors)
Zhao and Yue (2014)	Designed and evaluated Cloud-based Storage-Free BPM; evaluated IE, Firefox, Google Chrome, Safari, Opera; performed manual analysis, source code and binary analysis; conducted laboratory experiment and usability testing with 30 participants	1. Master password is optional 2. Inconsistent user interface 3. Weak protection mechanism for the users' data	1. Mandatory master password 2. Preference for cloud-based PMs 3. Salt and 1000 iteration counts to derive the encryption key 4. Proactive password strength checker 5. Use established user interface principle 6. Associate the dialog box with the address bar
Li et al. (2014)	Manual security analysis (analysed bookmarklet, web, authorisation and user-interface vulnerabilities) of 5 web-based PMs: LastPass, RoboForm, My1login, PasswordBox, NeedMyPassword	1. Critical security vulnerabilities allowing credential stealing 2. Collaboration supporting PMs have flaws in their authorisation logic	1. Defence in-depth approach 2. Ask the user to manually open tab and log in 3. Use simple credential owner model for collaboration supporting PMs
Carnavalet and Mannan (2015)	Evaluated 1Password, KeePass, LastPass, RoboForm; performed empirical analysis for the evaluation of the selected tool's algorithm, strengths and weakness	1. Inconsistent password checker 2. Password checker provides a decent score even for a weak password	1. Extend and implement established password checker algorithm rather than designing a completely new one 2. Focus on detecting weak passwords rather than finding a strong password

3.3. *Summary of the problems found in research literature*

From our review of the literature, we grouped the problems that were found and analysed by the reviewed articles' researchers into 11 main categories, exposed next: (1) compatibility issues, (2) learnability issues, (3) design constraints, (4) user interface (UI) and performance issues, (5) limited/error-prone password or master password control, (6) master password as a single point of failure, (7) weak user data security and privacy, (8) PM vendor/third-party dependency, (9) vulnerabilities to various attacks, 10) erroneous implementation of security mechanism/design, and (11) underutilised security resources. These categories of problems cluster all the issues mentioned in the articles we reviewed and are presented in more detail in Fig. 1. These problems must be distinguished and properly dealt with in order to design a useable, trustworthy and reputable PM.

In addition to security and usability, some articles have raised the issue of *trust*. In the case of cloud-based or online PMs in particular, the lack of trust has been described as the foremost problem [27,41–43]. Trust has even been identified as one possible hindrance to a PM's usage and acceptance by and among users [42]. Thus, we further classified the chief problem categories into high-level clusters of usability, security, and trust, and we present the problems under their respective headings in Fig. 1. It can be seen that most of the problem categories are related to at least two of these headings. In summary, security and usability are the basic requirements for a useable PM and this, in turn, may facilitate the development of trust in PM' s users. Henceforth, a PM design should combine usability and security in a trust-enabling manner. *This requires an appropriate balancing of the properties of usability and security when they are in conflict with each other*. The latter meta-requirement of software quality design needs to be considered and incorporated as early as

Table 2
Summary of the articles that propose PM design / architecture.

Authors	Descriptions of Study	Problems	Measures
Halderman et al. (2005)	Designed and evaluated Password Multiplier; performed comparative analysis of security and usability features against Password Safe, LPWA, and PwdHash; performed security analysis using formal arguments	1. Transportability issue 2. Vulnerabilities for brute-force-attacks 3. Long delay to calculate hash function can be inconvenient for users	1. Hash-based mechanism 2. Increases iteration of hash computation 3. Two levels of computation, first-level computation is one time and its value is cached and used as input for second-level computation 4. Username as salt
Yee and Sitaker (2006)	Designed Passpet; performed security and usability analyses using formal arguments; the earlier version of Passpet was informally tested in a deployment to 15 users	1. Complete dependency on one centralised server 2. Compatibility and deployability constraints 3. Incompatible with website login's functionality 4. Expects users to migrate all accounts to a new scheme 5. Does not support password periodic change 6. Vulnerable to dictionary and phishing attacks	1. Stores cache in local storage and backup in remote storage 2. Flexibility to login into websites from anywhere 3. Supports practices in website design and login form 4. Only minimal changes in user's behaviour but improves their convenience 5. Site-by-site migration to a new scheme 6. Periodic change of the master password as well as individual websites' password 7. Black-list method to detect phishing
Zhang and Jones (2007)	Designed uKey	1. Applicable for homogeneous types of accounts	1. Middleware to manage credentials of heterogeneous account
Wang et al. (2008)	Designed PwdCaVe	1. No protection for the decrypted password stored in memory during the login 2. Available security resources are underutilised	1. Uses Trusted Platform Module (TPM)–based password caching and verification 2. Eliminates the time for which plaintext password is stored in the memory
Asokan and Ekberg (2008)	Designed OnBoard Credentials (ObCs)	1. Available security resources are underutilised	1. Utilises existing resources TPMs and hardware secured environment based on M-Shields
Bojinov et al. (2010)	Conducted two studies, a qualitative user interview (survey conducted with 87 participants) and quantitative analysis of real-world password databases (RockYou with 32 million passwords, and phpbb.com with 241584 passwords) to elicit requirements; Designed Kamouflage	1. Adversary with access to the database storing passwords compromises the security	1. Hides legitimate passwords in sets of decoy passwords in the database 2. Features to add, remove, update, and find passwords
Erdem et al. (2010)	Store credentials in Single Sign-On (SOS) smart card and use two-factor authentication; integrate the tool with Firefox	1. If the master password is compromised, all other accounts are compromised 2. Deployability issue	1. Smart card as information depository 2. Two-factor authentication 3. Can Add/Update/Delete accounts 4. Features for Backup/Recover/Reset of data 5. Ability to change PUK and PIN of Smart card
Bugiel et al. (2010)	Designed TruWalletM- a wallet-like PM	1. Security resources are underutilised 2. Resource constraints in mobile phone 3. Risk from malware or phishing attacks	1. Hardware-assisted wallet-based authentication, employs Trusted Execution Environment 2. Allows password add and change

(*continued on next page*)

Table 2 (*continued*).

Authors	Descriptions of Study	Problems	Measures
Al-Sinani and Mitchell (2011)	Designed and evaluated CardSpace-based PM; theoretical analysis of its security and usability	1. Dependency on trusted third parties	1. Utilises the security advantages of CardSpace 2. Transportability using backup facilities of CardSpace 3. Password management is transparent to all the parties 4, Does not change the default IE security settings
Biddle et al. (2011)	Designed and evaluated ObPwd; conducted user study (their hybrid study combined two lab sessions "at-home" environment) with 32 participants; used questionnaire after each lab session and post study questionnaire to get users' response	1. Transportability issue 2. Memorability issue 3. Disallow password reuse	1. Generates password from the object (image, video, text passage) by hashing followed by an appropriate string password 2. Carries password generating objects or has online access to the objects
Bicakci et al. (2011)	Designed and evaluated iPMAN; conducted usability evaluation using hybrid lab and field study with 20 participants; used questionnaire and interview to get users' response	1. Memorability issue 2. Transportability issue	1. Icon-based password 2. Use hashing approach
Wang et al. (2012)	Designed IDKeeper	1. PM is fixed to a single system 2. Vulnerable to malware and phishing	1. PIN-based USB Key to store credentials and a browser extension which communicates with USB Key through a trusted path 2. Two-factor authentication
Nasirinejad and Yazdi (2012)	Designed and evaluated SaSy; conducted feature-wise comparison of SaSy with Lastpass and social network plugins	1. Data stored in one server has risks from inside attacks 2. Data stored to one vendor has risk of monopoly over the data 3. Deactivation or deletion of the master password means losing all credentials	1. Stores user's data in three cloud servers from different companies using dissimilar encryption method 2. Minimum changes on web service authentication mechanism 3. Reset the master password needs identification such as ID card or fingerprint of the users 4. Review and recover account's password
McCarney et al. (2012)	Designed and evaluated Tapas; performed security analysis (formal arguments); conducted usability evaluation using an in-lab user study with 30 participants; used questionnaire to get users' response; compared with two browser PMs	1. Master password is a single point of break 2. Fails to consider users' mental model and attention 3. Lack of solution on ease of deployment and usability in Smartphone	1. Dual-possession authentication (manager-key holder and wallet-cipher text), possession of any one cannot leak the data 2. No master password 3. Stores data locally in strong encryption 4. Simple design
Moghaddam et al. (2013)	Designed and evaluated cloud-based SSO for software-as-a-service (SaaS); used simulation process to compare with similar products, and performed theoretical security analysis	1. Danger of storing data in a single server	1. Stores password and encryption keys on different servers 2. Secure Sockets Layer (SSL) to transmit data
Yang et al. (2014)	Proposed a cloud-based PM	1. Master password is a point of failure	1. Utilises SSO and Biometrics 2. Master password is two-factor authentication 3. Authentication to download, synchronise, create, delete passwords

(*continued on next page*)

possible in the software development lifecycle of a trustworthy PM. The measures proposed in the reviewed literature for achieving the objective are further discussed in the next section.

Table 2 (*continued*).

Authors	Descriptions of Study	Problems	Measures
Boukayoua et al. (2014)	Designed and evaluated a mobile password-management keyboard; performed security and usability analyses using formal arguments	1. Passwords outside the browser are disregarded 2. Log in to service using apps requires switching back and forth 3. Clipboard used by PM in Android is publicly visible to all apps 4. Underutilised resources are untapped	1. Custom keyboard which interacts with the relying apps through the Input Method Framework 2. The keyboard searches for decrypted credentials (if in storage), and user pinpoints username and password field to populate them with value 3. Data storage is hardware backed to hinder offline attacks
Stobert and Biddle (2014b)	Designed and evaluated Veripass; conducted user study (used scenarios analysis and think aloud methods), and cognitive walkthrough (used pluralistic walkthrough method) with 5 participants	1. Master password is a single point of failure 2. Discards what users are good at, i.e., reuse of password 3. Memorability issue 4. Mental model gap 5. Transportability issue	1. Employs PassTiles [55], a graphical password mechanism 2. Does not store any passwords but password cues 3. Allows reuse of password, same image for multiple accounts 4. Stores information in remote server so accessed from any machine
Aliasgari et al. (2015)	Designed and evaluated Sesame for mobile phone; performed security analysis using formal arguments	1. Master password is a single point of failure 2. Lack of trust	1. Speech recognition as the master password 2. Users' data are stored locally or in cloud depending on their choice 3. Voice command to state the name of the service so as to find its credentials

4. Features and properties of a user-friendly and useable PM

In a security-sensitive system such as a PM, it is common to encounter a trade-off between strengthening security and improving usability. Although the conflict between these parameters seems obvious, in reality the two aspects have a less distinct boundary and strive for a more common objective. Security restricts access to undesirable operations and averts dangerous errors, whereas usability facilitates access to essential and necessary operations [44]. Security and usability must, therefore, be considered to proceed together [37,44] and simultaneously. In this section, we list security features and properties that can assist in elevating the user-friendliness and usability of a PM system. The authors included 'trust' in the security section because the reviewed articles reveal that the reasons for user distrust towards PM are associated with security. These features and properties were compiled after the authors' synthesis of the reviewed articles. Moreover, we also explored and grouped together all similar and related features and properties in order to comprehend and explain them in a less complex and more understandable manner.

4.1. Compatibility issues

4.1.1. Follow practices in web-design and OS

Usability issues do not occur solely because of PM applications. Some are developed because of non-standard and inconsistent practices exercised in web services, such as an unfixed position of the login container or the use of different words (e.g. username, account name, email, and phone number) to convey the same meaning of 'username' [42]. Next, the identification of the fields for username and password from an account creation form can become more difficult because of the presence of the fields for many other parameters spread across multiple pages [45]. Notably, several constraints may be imposed by the OS. For example, Android does not offer an API to integrate PMs with the browser. Other applications and web browsers on smart phones and tablets often do not provide a plug-in interface that allows smooth integration of PMs [46].

Similarly, in a PM built as an add-on for a web browser, the dialog box for getting the user's confirmation for saving a password, is sometimes designed as a part of the address bar and sometimes as a floating dialog box [40]. Such floating dialog boxes are vulnerable to phishing attacks, as they can easily be faked. Moreover, locations, options, and functions of configurations may vary with a PM's role, creating confusion for users [40]. Some PMs possess login functionality that is incompatible with the website's login functionality [47].

Unfortunately, the inconsistencies in web-services cannot be controlled. All one can do is to hope that web-service designers and developers around the world comply with *established* and *standardised* rules and practices for software development and quality assurance, which can help in reducing the inconsistencies in the user interfaces of their web-services. However, while designing the user interface for a PM, the designers and developers should adhere to the established and standardised rules and practices for software development and quality assurance.

A few valid suggestions to mitigate this situation are: (i) always design the correct user interface [48]; (ii) support uniform practices in website design and login form [47]; and (iii) take into account the user's mental model and attention capabilities (cognitive style features) during design [49]. Similarly, Nasirinejad and Yazdi [50] suggested following web portal service rules for credential(s) creation, which provide(s) easy access from browsers and electronic devices, without any monopoly or restrictions. Further, they recommended making a minimum number of changes on web-service authentication mechanisms. Moreover, one should always associate the dialog box with the address bar, in order to make it technically hard to spoof, and adhere to established principles of usability and user-interface design [40]. The ultimate goal is to design simple, consistent and intuitive user interfaces [51].

USABILITY	SECURITY	TRUST
Compatibility Issues Inconsistent practices in web design and constraints imposed by OS, support homogenous types of login, transportability issue		
Learnability Issues Semantic or mental model gap, user's attention, memorability issue, poor memorability of the master password		
Design Constraints Deployability issue, constraints imposed by platform		
UI & Performance Issues Inconsistent UI, support only for bulk migration of accounts, optional auto-fill, long delays for hash calculation, time-consuming key derivation function, time taken for encryption and decryption, low login success rate		
Limited/Error-Prone PW or MPW Control Necessary features and functionalities missing (MPW reset; add, remove, update, find PW; deactivate, delete account; backup, recovery, download, synchronise data), inability to recover PWs from deleted or deactivated PM's account, unusable PW generator (generated PW is complex, incompatible to website PW requirements, difficult to memorise, or incompatible with the organisation's PW policies), inability to change the MPW, optional MPW, remember and save MPW, active and intrusive dialog box to get user's confirmation, missing PW/MPW strength checker, inconsistent or erroneous PW checker		
	MPW as Single Point of Failure Disclosure of the MPW risks all user data, accidental deletion of MPW means losing all user data	
	Weak User Data Security & Privacy Insufficient protection of user data (because of vulnerabilities and erroneous security mechanisms), insufficient user information on protection mechanisms	
	PM Vendor/Third-Party Dependence Lack of trust towards PM's vendor (because of weak user data security & privacy, erroneous security mechanisms), preference for local storage over cloud storage, risk of one vendor monopoly over data, risks of storing data to one server, dependency of the centralised services, dependency on trusted third parties	
	Vulnerabilities to Various Attacks Database vulnerabilities; vulnerabilities, e.g., to host, brute-force, rogue Wi-Fi, hotspot, dictionary, phishing and malware attacks, and tampering; critical security vulnerabilities allowing credential stealing; using a stored token instead of manually opening new tab	
	Erroneous Security Mechanism/Design Database kept unlocked even when PM not in use, insecure auto-fill (immediate auto-fill as website is loaded, treat HTTP and HTTPS equally), lack of integrity checking, unencrypted metadata, decrypted password in memory without protection, android clipboard publically visible to all apps, authorisation logic of collaboration supporting PMs, errors in the implementation of key derivation	
	Underutilised Security Resources Untapped underutilised security resources	

Fig. 1. PM problems relative to usability, security, and trust (PW=password, MPW=master password).

4.1.2. Facilitate transportability in PM

People nowadays use multiple different kinds of devices, such as laptops, tablets, and smartphones, which may all contain different operating systems and browsers. This is a challenge for the functionality and usability of a PM, because in order to fulfil user needs, it should provide access to passwords through all the users' devices. However, some PMs do not provide their users the freedom to login from anywhere and through any device [47, 52] or service platform. This limitation may be there because of either a platform-based dependency or the inaccessibility of the users' data. This portability limitation is one primary reason that discourages the use of PM applications [34]. Therefore, a *hashing approach* is suggested for better transportability [28,52]. To facilitate the accessibility of users' data, Erdem et al. [53] used a smart card as an information depository, and Wang et al. [54] used a USB key to store credentials. However, using hardware for authentication purposes contravenes the password benefit of having no physical object to carry [1].

To relieve people from the need to carry hardware, Biddle et al. [55] suggested an approach where users only have to carry password-generating objects with them or have online access to the objects. The users' data can be stored in the cloud and, consequently, the accessibility problem is solved. A completely different and unique approach was presented by Gaw and Felten [56], who proposed that, instead of just storing and auto-filling passwords, PMs should also help users to learn their password by displaying it with a low contrast background, as users have mostly relied on their memory to remember and manage passwords. Once the users have memorised the passwords, they can log in from anywhere using any device. In situations like this, assigning mnemonic passwords with better memorability [6] or personalised cognitive passwords for better recall and privacy [57] for online accounts will reduce the cognitive (over)load of the password users. Nevertheless, even these two password mechanisms have their limitations. For example, users may select and derive their passwords from a very common mnemonic phrase, thus making them easy to crack [6]. Additionally, the quality of questions used for cognitive passwords is often not personal enough, meaning that anyone who knows the users could guess the answer [57].

Therefore, a suitable solution could be to design a PM that supports the most widely used devices (e.g., desktop, mobile and tablet), OSs (e.g., Android, Windows, iOS, Mac OS), and web browsers (e.g., Chrome, Internet Explorer, Firefox and Safari) [58], and stores its data (i.e., stored credentials) in the cloud. This will allow the users to synchronise the PM installed on their device to the cloud, after authentication, and transport the PM to multiple different kinds of devices, irrespective of the device type and the OS and web browser installed on it.

4.1.3. Support for heterogeneous types of login

Most PMs support only homogeneous types of login procedures [56,59,60]. The PMs that are integrated with a web browser facilitate the login process into services only by using that particular web browser and disregard passwords outside it. Similarly, PMs designed for mobile phones or tablets lack auto-fill functionality for application-based services because of technical constraints [46]. Sometimes, these technical constraints are imposed because of an organisational policy or for security reasons [45]. Technical constraints —for example, the lack of auto-fill—compel users to manually transfer credentials from the PM to the application where the user wants to log in and, more alarmingly, often through an insecure clipboard. Thus, as technical constraints decrease usability and may also have an impact on security, they can simultaneously influence the level of users trust. From the point of view of software design and systems maintenance, *these limitations in PMs also adversely affect their acceptance by end-users* [56].

There have been multiple propositions made in the literature we reviewed to solve some of these technical constraints. For example, to overcome the auto-fill issue in mobile phones, Boukayoua, Decker, and Naessens [59] designed a mobile password management keyboard, which when initiated searches for credentials for that application the users have to point out the username and password fields to auto-fill the credentials. The main limitation of this approach is that every mobile platform does not support the overriding of their default keyboard function, which is a prerequisite for this approach. A rather more appropriate approach is illustrated by Zhang and Jones [60], in which a piece of middleware (i.e., uKey) is used to manage the authentication credentials of heterogeneous accounts.

Another appropriate solution, where there is no need of any middleware, could be to allow and facilitate the export of the data (i.e., stored credentials) of the PM but only after the needed authentication, so that the data could be imported or migrated to another PM. If a PM does not support a type of login procedure, the user can use an alternative PM that supports the login procedure and exports the data from the earlier PM, even if the two PMs are from two different vendors.

4.2. Learnability issues

4.2.1. Narrow semantic gaps between PMs and their users

A semantic or mental gap occurs because of a disparity between the ways various users perceive a system and the actual behaviour of the system [61]. These situations often create confusion and frustration to the system users, while these system situations can be dangerously exploited by attackers. These semantic gap conditions can also exist in PMs [39,42,48,49]. Software design components with poor usability or reduced user experience, in particular, are responsible for semantic gaps in PMs. Typical examples and causes of semantic gaps and consequent privacy/security threats include situations created by (a) a lack of suitable cues and feedback from the system, (b) unclear or hidden instructions, (c) confusing features and functionalities, (d) obscure rules and conventions, (e) a complex process for error recovery, and (f) a lack of a visual interface. Lack of understanding of the paramount importance functions and actions of a PM may lead to situations in which users do not understand the PM's benefits in terms of password security [19]. This alone will consequently decrease their level of trust and can affect the acceptance of the PM by the users. Thus, software developers should pay special attention to usability properties and other elements that help users understand what a PM does and how it enhances their password security. Along with usability, learnability of the PM's functions is an important quality property of the user experience. Problems attributed to learnability may further result in users having an incorrect and incomplete mental model of the PM. The reverse is also true; problems with the (incomplete and incorrect) mental model can influence the PM's learnability perception by the users.

To narrow down this semantic gap, the main suggestion is to improve the PM's usability and consequent user experience. In the software/systems design Berki, Isomäki and Jäkälä [62] and Chiasson, van Oorschot, and Biddle [42] encourage (i) *holistic communication modelling* through *shared models* and (ii) understanding and inclusion of users' *intentions* and *expectations*, respectively. This is possible through the involvement of actual users in the software/system development processes and by frequent feedback cycles. More specifically, Chiasson et al. [42] advised to design a suitable user interface and facilitate appropriate interactions between the user and the system through short, accurate, understandable, and visible instructions, feedback or cues that can communicate the complete mental model of the PM to its users. Likewise, Bicakci et al. [39] and McCarney et al. [49] recommended visual password interfaces and simple design, respectively, in order to mitigate the mental gap. Finally, Stobert and Biddle [48] suggested designing intuitive user-interface elements and including the PM features and functionalities that are simple to understand. These particular suggestions in prior studies are also supported by Wu et al. [61], Berki et al. [62] and Adams and Sasse [20], who emphasised the consideration of human factors and human involvement during the requirements analysis and design phases, in order to facilitate the expression and fulfilment of users' intentions and expectations through usability engineering [63] in the predesign phases.

4.2.2. Increase learnability of the system

A PM is supposed to be easy to use. Yet, on the contrary, usability studies performed on several of the existing PMs have revealed that users do not perceive them as easy to use [27,42,48]. For example, users lack understanding of how the PM system works [48] and fail to perform even basic activities such as using the PM to login to a website or to generate passwords [27]. Initially, these problems may be, at least partially, because of incorrect beliefs about the PM's functions, but they can also obstruct the users' further attempts to develop a correct and consistent mental model of the PM's functionality.

Therefore, PM designers should take into account their actual end-users (i.e., common people in particular along with other sophisticated stakeholders) and should design learning activities that explain 'the basics' of the PM as simply and intuitively as possible. These activities include installation, enabling and disabling a PM, registration for an online PM, logging in using the functions of a PM, changing passwords, correcting errors, the recovery process, generating passwords, and the list can go on. These activities should be aimed at supporting users in the process of learning and building a proper, non-complex mental model of the PM. While enhancing the learnability of the PM, developers could also find value in considering ways to improve users' awareness and understanding the benefits of PMs. Unfortunately, the literature sources reviewed here (as well as additional related sources) do not provide detailed insight into this matter. Nonetheless, as this subject may be related to the acceptance of multiple PM applications [19], it should be studied further. Ciampa et al. [34] revealed that raising the users' awareness about the advantages of a PM is directly associated to its acceptance.

4.2.3. Improve the memorability of the master password

Regarding usability of the master password, one rather distressing concern is memorability, which is necessary for master password use. For instance, iPMAN [28], ObPwd [55], and Versipass [48] have each attempted to solve this problem. Both Versipass and ObPwd do not require a master password but instead offer users the flexibility to generate passwords for any services simply by selecting image(s) and digital object(s). This means that there is nothing worth of protection in storage. iPMAN, however, uses an ordered set of icons as the master password. A common consideration in all these approaches is that humans are better at remembering a picture than words [64]. Notwithstanding, several approaches endeavour to improve the memorability of a textual master password and at the same time to strengthen its security; these are summarised in Section 4.6.

4.3. *Design constraints*

4.3.1. *Simplify the deployability of the PM*

At present, there is a lack of practical, easy to deploy, and useable PM solutions [47,49]. Some of the PMs' tasks demand new resources, and they may also contain several clumsy settings that are required to make the PM work properly. For example, the custom keyboard-based PM by Boukayoua et al. [59] was compatible only with the Android platform, as other mobile operating systems did not allow overriding their default keyboard. But now iOS 8.0 and onward versions also allow overriding the system keyboard with a custom keyboard [65]. Similarly, clipboard-based PMs are dependent on the OS platform for its clipboard [46]. The cost of such technical dependencies, sometimes, can be high. For example, the components on which a PM is dependent on can evolve independently, consequently affecting the deployment and functioning of the dependent PM, i.e., the PM may not work as intended or even at all. In order to overcome this problem, one simple suggestion could be to manage dependencies by eliminating the dependency on any platform or its resources. This means that the PM must not be dependent on any specific hardware or library file or OS platform found in only a certain device.

A principal concern then may arise is *"how feasible is to overcome such dependencies?"* It may not always be feasible to completely remove such dependencies because such dependencies may have occurred due to external factors, such as legal or policy limitations, accessibility and capabilities of a platform; however, designing a system with as few of them to the extent possible can be achievable. In software, if so desired, their dependencies can be more or less easily changed [66]. For this, dependencies and their sources can be identified and categorised into, for instance, mandatory and optional dependencies, or positive (required) and negative (forbidden) dependencies [67], and based on this classification the development team can determine what to allow or to prevent during the design and development phases. While doing so, it is suggested to decrease the dependency (or coupling) between two components so that changes to one do not impact the other [66].

4.4. *User interface and performance issues*

4.4.1. *Include step-by-step migration of accounts*

Some PMs set their own rules and expect users to change accordingly. For example, it is often expected or stated that users should migrate all their accounts to a new scheme [47]. Obviously, a certain level of changes can be expected in users' behaviour. However, those changes should be minimal and only for the sake of improving the use of the PM and for the user's convenience [47].

Therefore, a PM's functions should allow for site-by-site migration to a new scheme rather than bulk migration [47], and it should not demand any changes in the browser's default security or other settings [55]. While migrating site by site, a problem with many PMs is that they offer to save even incorrect passwords [45]. Sometimes people make mistakes while typing their password, but despite an error message informing the user of the incorrect password, the PM displays its dialog box offering to save the password. Similarly, if users reset or change their password and login to their accounts, the PM also saves the new password but does not remove the old one. In such case, the PM holds two passwords for the same account and when next trying to login to the account, the PM may get confused as to which one to auto-fill. Regarding the first problem, one solution can be for the PM to verify whether the response from the server contains the same destination as the one submitted, which implies that the login failed. However, this solution does not always work as it may result in false negatives (i.e., the PM does not offer to save even the correct password). Hence, in order to confirm password errors, the HTTP layer status code can be checked. For example, if in some case the response code was "200 OK", this implies that the password was incorrect [45].

4.4.2. *Use passive dialog box for user's confirmation*

Although the use of an intrusive dialog box in PMs to get users' confirmation before storing the password increases the tendency and perhaps the will to store passwords, it may also, ironically, lead users to make a habit of storing passwords while using a public computer. This often happens unconsciously, or without realising it [68]. Such an activity leaves a user's account open and exposed to others for misusing or performing malicious activities. Undoubtedly, it is necessary to get a user's confirmation before the PM stores her/his passwords [51], but it is also recommended to adopt passive communication [68].

4.4.3. *Include measures to improve the performance of PMs*

There are various factors that can increase the performance of a PM. These factors relate to performance times of a PM's actions, but also to how successfully users are able to perform their tasks with the PM. For instance, the time taken for encryption and decryption of data is crucial for the performance and overall efficiency of the PM; this has been discussed in Section 4.10.2. Other factors include login success rate and login errors, login times, ease of use, and password creation time. Each of these factors need considerable attention in order to improve the PM's performance [55].

4.5. *Limited/error-prone password or master password control*

4.5.1. *Include necessary features and functionalities*

A few PMs do not support periodic changing of individual websites' passwords [47]. This means that users cannot use these PMs to store these services' passwords that require periodical or scheduled by routine change.

To address this problem, it has been suggested that the PM should include features like *add, remove, update*, and *find passwords* [47,60,69]; *deactivate* or *delete* accounts, *backup* and *recover* data, *change* and *reset* the master password [53]; and *download* and *synchronise* data, but only after the needed authentication [70]. These features allow users to have control and management over their passwords and, thus, are important for usability reasons; but they may, in their part, also assist in the development of trust between users and the PM's utilities. User studies related to PMs have shown that users do not want to feel that they have lost control over their passwords [27,28,42]. Hence, adopting features that allow the user to make decisions over the passwords may help to diminish the feeling of having no control.

4.5.2. *Design a useable password generator*

Several PMs are equipped with a password generator, which saves users from creating passwords on their own for the various services they use. However, the problems with password generators are: (i) they are complex to use [42]; (ii) they generate passwords that may be incompatible with a website's password requirements [71]; and (iii) passwords generated by them are cognitively challenging and, thus, almost impossible for users to memorise [72]. Furthermore, different websites have different requirements for the length and character sets permitted for use in passwords [72]. Even different organisations can have their own policies and criteria for generating passwords. The incompatibility between generated passwords and organisations' practices can prevent users from using the password generator.

To overcome these limitations, Stobert and Biddle [71] suggested allowing the users to define and select an appropriate rule-set to generate password(s) for each website. For example, LastPass [73] allows users to select password rule-sets. The disadvantage of this approach is that users can cause problems by selecting the simplest combination, thus reducing the strength of the password. Therefore, it is important to provide continuous constructive feedback during the password generation process and to incorporate explanations on why an apparently weak password is rejected or is insecure [20]. Such feedback on the password's strength/weakness can make the system transparent and can also refresh the users' knowledge of password design and choice procedures.

4.5.3. Include strength checker for passwords

There are some PMs that do not enforce any requirement to the level of strength of users' master passwords [52,74], despite the fact that users often create easy-to-guess passwords if they are allowed to do so [75]. Furthermore, inconsistency in the rules and policies used by PMs to measure the strength of passwords can confuse users about the actual meaning of a strong password [52]. For instance, a high-entropy password created by some PMs may be assessed as a lower strength password by some websites' password checkers [52].

It has, therefore, been suggested to integrate a proactive password checker to check the master password strength [40,47,74]. To support their suggestions, Carnavalet and Mannan [76] advised extending or improving and implementing the existing password strength checker algorithm rather than designing a completely new one. They further suggest to focus on detecting weak passwords by applying *cracking algorithms*, common password dictionaries, and considering non-trivial challenges, such as finding patterns, coping with local cultural references, and dealing with leaked passwords. Moreover, providing constructive feedback on why the password is weak or insecure can make the procedure transparent and can also enhance users' knowledge about strong passwords and the reasons for constructing them so.

4.5.4. Include recovery and reset features for the password and master password

After deactivation or deletion of a master password or online account, all the login information stored in the PM is lost. Conventionally, this may seem correct; however, there is also the possibility that a user may have deactivated or deleted an account or master password mistakenly or in haste. Furthermore, there are some PMs that do not enable changing of the master password [40]. This is a severe usability issue and contradicts an important security policy, namely the recommendation to periodically reset the password [14].

Nasirinejad and Yazdi [50] suggested that PMs should include a provisional act to review and recover an account's password. An alternative option could be to have a backup feature [47,51,53]. However, the decision to retain a user's data after s/he has deleted or deactivated her/his accounts may raise questions of ethicality and legality. In such circumstances, it may be justifiable to include a set period for which the user's data will be retained after deactivation. During this period, it will be possible to recover the data; afterwards, it could permanently be deleted. Similarly, it has been suggested that a PM must allow its users to reset the master password [40,47]. To ensure that only the authorised user resets the master password, SaSy [50] suggested a requirement for some form of identification, such as an ID card or fingerprint, before the PM allows for changing or resetting of the master password.

4.6. Master password as single point of failure

4.6.1. Make mandatory master password

Regarding user convenience, some PMs do not enforce the use of a master password, leaving users unaware of its importance; thus, users' stored passwords remain frequently unprotected [40,74]. Moreover, one of the primary concerns among the users of PM is that all of their passwords would be exposed to attackers or anyone who can uncover the master password of the PM [34]. Therefore, a strong password (i.e., with high entropy or a large guess number) should be mandatory, and the process to activate such authentication mechanism(s) must be placed in a position that is visible to the users. Further, the users should also become aware of the consequences and possible vulnerabilities when they leave their PM without master password.

4.6.2. Not remembering and not saving the master password

Zhao, Yue, and Sun [74] found that some PMs remember and save the master password to the local machine and perform automatic auto-fill. One may view this as the laziness of the users to get away with an effort needed to remember even a single password, i.e., master password. Whereas, other can also argue that the users considered from a cost benefit standpoint [13], since they do not have to remember even the master password and thus makes the signing process convenient and faster. More importantly such automatic auto-fill makes the users less susceptible to typo-squatting and phishing attacks [77]. But such automatic auto-fill obviously exposes their stored passwords. Anyone with access to the users' device(s) also gets access to all their stored passwords, i.e., access to all their login accounts. Essentially, when a PM becomes a single point of failure, it cannot be left unprotected. Therefore, Zhao et al. [74] suggested to exclude a feature that remembers and auto-fills the master password. Another way that probably can be more user-friendly is to provide the users options to either enable or disable master password; however, when disabling the master password they must be alerted about the consequences of this action.

4.6.3. Protect the master password

Dependence on a master password to protect all the other stored passwords is equivalent to 'putting all eggs in one basket'. The master password is, often, recognised to be a single point of failure [41,48,49,53,70]. To overcome this problem, the reviewed papers suggested two types of measures, which are summarised next.

4.6.3.1 Strengthening the Master Password. Fortifying the authentication mechanism(s) used for the master password is one potential solution. In the place of a text-based master password, other authentication mechanisms can be used, such as *two-factor authentication, biometrics*, and *graphical passwords*; however, these authentication mechanisms come at a cost and possess their own limitations [1,11]. Moreover, security is contextual, so the type of authentication mechanism that may need also depend on the importance and sensitiveness of the credentials stored in the PM.

For two-factor authentication, the following mechanisms have been suggested: (i) a single sign-on (SSO) smart card protected by a secret PIN code, acting as the *depository* of the user's credentials [53]; and (ii) a USB Key protected by a secret PIN storing the user's accounts information [54]. Similarly, Yang et al. [70] proposed the use of a combination of SSO and biometrics as the master password. Something common to all these approaches is that they make use of a combination of *what you have* and *what you are* with *what you know*, i.e., two-factor authentication. Another promising approach is applying what you are (biometrics) authentication mechanism, i.e., replacing the master password with voice recognition [41]. As a departure from the aforementioned authentication mechanisms, Bicakci et al. [28] suggested employing an icon-based authentication mechanism. Within it, users would generate a master password each time by selecting an ordered set of icons, as this mechanism has both better security and better memorability because of the use of icons (images).

If a PM employs a textual password, Yee and Sitaker [47] and Zhao et al. [74] suggested to integrate a proactive password strength checker for the master password. This can help users to generate a high-entropy or maximum guess-number textual password. Nonetheless, one has to be careful of this password strength checker, so that it is not inconsistent and poorly designed (for more details see Section 4.5.3). Still, having a stronger master password will also increase the chance of user error when entering it to login to the PM [7]. Therefore, an alternative can be the approach offered by Bonneau and Schechter [15], who suggested applying a technique called spaced repetition (i.e., relaxing the time constraint to learn), through which they succeeded in making people learn and use 56-bit codes. As the user only has to know the master password, s/he can be trained to memorise and learn to use passwords of 56 bits in length. This can, relatively, provide better security. Another potentially suitable approach is to figure out and assign a mnemonic password (e.g., a password '4s&7yaoF' is derived from the phrase "Four score and seven years ago, our Fathers") [6] as the master password. Socio-cognitive personal associations can be a good way to increase the learnability of a PM and the number of people who can potentially use it.

4.6.3.2 Paradigm Where the master password is not required. Strengthening the master password is definitely a promising approach to protect a user's stored passwords from exposure; notwithstanding, some of our reviewed papers suggested mechanisms in which either there is no need for a master password, or attackers cannot benefit from it further, even if they succeed in compromising the master password.

In Kamouflage [72], a large collection of decoy data that are indistinguishable from the true data, are stored along with the true data, and this increases the difficulty of finding the true data. The main objective of this mechanism is to distract the attackers and force them to make several successive failed attempts to login to a service until the account is locked. Similarly, McCarney et al. [49] employed a technique of *dual-possession authentication*, in which both the manager-key holder and the wallet-cypher text (that is, two devices like a *smartphone* and a personal computer) must be accessed and reached simultaneously in order to access and use the data. Hence, the possession of one device cannot allow someone to access the data. These co-operative mechanisms do not implement a master password.

Other PMs that have attempted to remove the master password are *ObPwd* [55] and *Versipass* [48]. In ObPwd, users generate passwords by selecting a digital object (e.g., an image, video, or text passage); thereafter, they simply have to remember the selected object from a large pool of objects and generate the password whenever it is required. Likewise, Versipass uses a gridded image (graphical) based mechanism, called *PassTiles* [71], for authentication purposes. *PassTiles* does not store any passwords, but only password cues in order to help users recall their password so that they can generate their passwords and safely log in to online services.

ObPwd is susceptible to attacks from attackers who are equipped with personal cognate information about the potential victim, because such attackers may guess the user's online preferences, i.e., which object the user is more likely to select. Similarly, *Versipass* is susceptible to *hotspot attacks* (i.e., particular grid-tiles in the image are more often chosen as a part of passwords). In order to counteract the susceptibility of ObPwd, it has been recommended to use a large pool of personal objects, so that this increases the difficulty for attackers to guess the objects that the user may have selected. Regarding *hotspot attacks* in *Versipass*, it has been recommended to randomly assign the image for the password, instead of allowing the user to choose it.

4.7. Weak user data security and privacy

4.7.1. Lock password database when not in use

In the Android operating system, a few PMs leave account information visible in the recent applications view, even though the users have left the PM to perform other activities [46]. This can result in a serious security hazard, as anyone with physical access to the user's device can see the information.

Therefore, it has been recommended to lock the password database either immediately after the users exit the PM application or after a configurable amount of time, and also to replace the thumbnails in the recent applications view [46].

4.8. PM Vendor/third-party dependence

4.8.1. Allow users to select the storage type by providing both the options (local and cloud storages)

Whether to use local or cloud storage space to store users' data is a rather equivocal question. Both means of storage have potential advantages and disadvantages. Cloud storage has a trust issue, i.e., will the people/users trust the cloud service that keeps their stored passwords? However, at the same time PMs with cloud storage are found to be more popular among the users due to their portability [34]. Moreover, in cloud storage, availability can be a problem; if there is an interruption in the service, users will be unable to login to any account. On the other hand, storage in a cloud storage space offers transportability to users, similar to local storage when a smart card [53] or a USB key [54] is used to store data. More importantly, cloud storage can be a more attractive and frequent target for attackers. For example, in the data breach suffered by LastPass, attackers were successful in reaching and stealing data, including master passwords and other login details, thus potentially leading to exposure of data that had been stored with these particular passwords [19]. Local storage has also faced some serious security problems, and it is questionable whether users are able to provide sufficient security technology to protect their data. Some PMs allow users to choose whether to store passwords in an encrypted form or in plain text.

Aliasgari et al. [41] provided a rather neutral suggestion of supporting the option for users to store their data either locally or in the cloud, thus reflecting their personal choices and preferences. McCarney et al. [49] suggested using strong encryption when storing data locally. Other suggestions include (i) offering offline PMs but with improved usability [27]; (ii) storing in the cloud, but implementing integrity checking and authenticated encryption with associated data (e.g., unencrypted metadata); and (iii) adopting a robust host-proof application design pattern [78].

Some studies have even advocated for avoiding centralised external dependencies [47]. Data stored on one server presents risks [79] such as *host-attacks* and vendors gaining a monopoly over the data [50]. Thus, it has been suggested to put the data into cloud storage, but distributed among three servers from three different companies using dissimilar encryption methods [50], or to store passwords and encryption keys on two different servers and transmit the data through an SSL connection [79].

Whether the data is stored locally or in the cloud should be left to the users to decide by providing them with both options, that is, both cloud storage and local storage should be available. However, it is inevitable for the PM vendors, through correct and accurate information, to effectively communicate to the users about the choices of storing credentials locally or in the cloud and their benefits, risks and costs. Such communication influences the users' satisfaction and thus helps in building relationship and gain trust [80]. In addition, using strong encryption to hash the data before storage should be mandatory for both types of storage. Storing the encryption keys and passwords on separate servers will enhance the PM's security. Even so, one must also take into account the time and processing required for the encryption–decryption process. According to Nielsen's [63] response time study, 10 s is the time limit for keeping a user's attention focused on a *dialog box*. For situations where the necessary delay is longer than 10 s, providing feedback about what the system expects to be done can be helpful.

4.8.2. Establish and build trust towards PM's vendors

Apparently, users do not trust PM vendors and, thus, they do not agree to store passwords into PM vendors' cloud services [27,41,42,46,50]. Karole et al. [27] observed that users select offline PMs more often than online PMs, despite the fact that the usability and security of the latter are superior [40]; the reason for this choice is the distrust towards the PMs' vendors.

There may be two main reasons for this distrust. First, users may not be fully convinced that the PM's vendor is capable of protecting their passwords from inside as well as outside attacks. Second, users may be frightened that, without their knowledge and consent, the PM's vendor may misuse the users' personal information, including personal credentials such as individual passwords, in order to gain personal benefit.

Unfortunately, users' distrust towards PMs vendors is real and reasonable to some extent. Several PMs have been negligent with the security of users' data. For example, databases storing user data can be vulnerable to various attacks because of their weak protection mechanisms [40,81]. Further security problems can include: (i) encryption mechanisms used for the users' data are applied incorrectly [46,49]; (ii) *integrity protection* for cypher text is lacking or limited in PM databases that store users' data; and (iii) metadata, such as URLs, are stored in the PMs as plain text, which makes them susceptible to host attacks and tampering [78]. Likewise, the security of PMs that permit local storage of passwords is also lacking and is rather discouraged at present. This is so because they implement either very weak or no security measures in their databases; thus, they appear to be insecure [40,46,81] and consequently cannot be trusted.

According to the suggestion provided by Zhao and Yue [40], both users and PM vendors must take responsibility for data protection by, respectively, creating a strong master password and using proper mechanisms to protect users' data. PM vendors need to consider measures that provide confidentiality, privacy and data authenticity [81] and can correctly implement key derivation or encryption [46].

Likewise, other important suggestions associated with security and are indispensable for building up trust, are: (i) include integrity checking; (ii) implement a scheme that provides authenticated encryption for the main data as well as the associated data (e.g., unencrypted metadata) [78]; (iii) provide the users with better protection from both inside and outside attacks, through user-centred/specific information; (iv) follow a strict policy (to show respect) for the confidentiality of users' data [46]; (v) adopt password management practices that are transparent to all interested parties [51]; (vi) perform strong protection on the client side and assure users that their information cannot be obtained even by insiders by any feasible means [40]; and (vii) adopt a robust host-proof application design pattern [78].

Trust implies more than security and privacy. It also implies reliability [82] and usability or user-experience [83], which also partly help in building trust. By reliability, it means the PM vendors ability to recognise an attack and minimise the damage of any attack and subsequently recover from it [82]. Similarly, usability or user-experience conveys making the system easy to perform the right action, difficult to make an error, and easy to recover from an error if it happens anyway [84]. Further, it means making the system simple and intuitive that makes the system clear and understandable. Therefore, in addition to privacy and security recommendations, some user-centred suggestions that focus on user interaction, are: (a) improve usability and user experience of the system [42]; (b) build an offline PM with improved usability [27]; and (c) provide options for both local storage as well as remote storage of the user's data [41]. But whatever the options are provided to the users, they should be accompanied with relevant and sufficient information about each provided choice, particularly, about their security and economic benefits and risks. More importantly, this information should be framed in a way that appeals to the recipients and communicate in the way that is meaningful for them or cognitively not burdening to them, engages them positively and bring them on board with what is expected to be performed or achieved from them.

Although it is difficult to exactly pinpoint the factors that can instil trust in users, we believe that by incorporating the aforementioned measures in a PM, one can advocate in favour of the PM's improved security of data, clear policy for data usage, and safe options by which users can easily store their data. These can also be provided in a helpful way. Trust is built up gradually; it takes time and repeated good user experiences [85]. In the hope of establishing and building up user trust, some PMs have announced in their websites that they do not know their clients' master passwords and other stored passwords; hence, they cannot resend or reset a master password [74].

4.9. Vulnerabilities to various attacks

4.9.1. Protection of the master password against phishing attacks

Increasing the entropy or guess number of a textual password will make it robust only against attacks like brute-force attacks. Yet, it remains vulnerable to phishing attacks, especially those conducted using multifaceted social engineering techniques and keylogging [13]. For instance, *LostPass* is an example type of a phishing attack on PMs, in which the attacker mimicked the look and feel of the *LastPass* browser plugin and site, and tricked users into divulging their master password [86]. Moreover, a strong password also places considerable burden on user, i.e., effort needed to memorise it.

Although there is no guaranteed countermeasure for social engineering attacks, user knowledge and readiness can act as a preventive measure [87]. As a technical measure to protect users against falling for social engineering tricks, Bojinov et al. [72]

proposed the use of *Dynamic Security Skin* [88], an anti-phishing authentication mechanism, for the master password. Similarly, Yee and Sitaker [47] applied black-list methods to detect phishing attacks and used colour to display the reliability of websites. *Versipass* [48] prevents users from logging into any URLs that are absence in its database. Other mechanisms, for example, adopting other authentication methods, such as biometrics and/or e-tokens, as the master password can also help to deal with phishing attacks. Likewise, the use of one-time passwords can be another alternative for the master password.

However, these authentication methods come at a cost. These potential measures may be effective in protecting against phishing, but they do not provide any concession for improving the usability of the system or for eliminating the vulnerabilities of its user interface. For example, the aforementioned *LostPass* attack was possible simply because of the poor design quality and existing vulnerabilities in the user interface of the PM. The *LastPass* PM designed browser pop-up boxes or banners whose exact HTML and CSS were easy to find and placed them on the *browser viewport*; it was an unwise decision [86]. Therefore, the design of security related interactions, educational efforts and risk communication with users should align with the users' mental models and capabilities [89]. The users should first understand the concepts of vulnerabilities if they are expected to act and react safely and in a secure manner.

4.9.2. Only allow manual opening of a new tab

For protection against *iframe attacks* (i.e., loading a fake web page inside a legitimate web page), Li et al. [90] suggested that the system should ask the user to manually open a new tab and login rather than storing a token into local storage or cookies when the user is already logged in. This may not be very user-friendly, but for the sake of security, opening a new tab does not seem to create many problems and, thus, may be acceptable.

Moreover, limiting the use of links or buttons to open a new tab is also necessary, because it can cause disorientating especially for those who have difficulty in perceiving visual content [91]. However, opening a new tab by clicking a link or button may be justifiable in the following situations: (i) opening a page containing context-sensitive information, such as help instructions; and (ii) following a link outside the secured area of a site so that the first website login does not have to be terminated.

4.10. Erroneous security mechanism/design

4.10.1. Implement mandatory but secure auto-fill

PM's Auto-filling of a username and password to a website is convenient as it saves from the trouble of manually filling in these credentials. However, auto-fill has also encountered some security issues [92], due to which some websites have chosen to disable this feature. Similarly, some PMs also exclude the auto-fill feature [46], even though a secure implementation of the auto-fill feature is essential for a useable PM. Disabling the auto-fill feature often means that the user has to manually enter credentials into the website's textbox, which, in turn, creates a usability concern. Moreover, an auto-fill feature reduces the risk from shoulder-surfing attacks [51]. Hence, in addition to being convenient, the auto-fill feature can be a relevant security factor for the user in her/his social context. Therefore, implementing the auto-fill feature in PMs requires careful consideration of both usability and security aspects involved in user experience.

There are two types of PMs: those requiring user action before auto-filling the password values on a web page and those auto-filling the web page form as soon as the users visit the web page, relieving the users from typing [40,53]. Neither type differentiates between web pages using the HTTP and HTTPS protocols, though in reality web pages using HTTPS are more secure than those using HTTP. This flaw (i.e., treating HTTP and HTTPS as equal) can expose a user's information to a rogue Wi-Fi attack (known as Evil Twin) as well as to other types of attack [92].

Therefore, it has been suggested that PMs should integrate an auto-fill feature that forces users to interact with and give their confirmation before the PM auto-fills a form [40,92]. However, asking users' confirmation via a dialog box is not without problems, as it has been recently observed [68] that users may tend to ignore security-related messages and just answer 'OK', in order to proceed quickly to their main task. Further, it has been advised that PMs should submit data only over HTTPS, even when the login page is loaded over HTTP, and should always show the domain name being auto-filled to the user before filling occurs [92]. Moreover, Silver et al. [92] suggested storing the actions present in the login form along with the username and password when they are saved for the first time. In so doing, the next time that auto-fill is used, the actions of the login form can be compared with those stored by the PM, and the credentials would be submitted only if the two actions match.

In addition, Wang et al. [93] found that passwords from PMs are decrypted and stored in the device's memory without any protection during the login process. This is a serious security pitfall and potential risk. Such passwords are susceptible to memory viewer and phishing attacks during password verification. Hence, Wang et al. [93] suggested eliminating the time during which a plain-text password is stored in the memory.

4.10.2. Store passwords in encrypted form and use optimal iteration for key derivation

Some PMs store passwords in a plaintext whereas others in encrypted text. PMs store a large collection of user's passwords, thus, they can be a prime target for cyber attack. Therefore, it is advisable to store the passwords in encrypted text.

While employing an iterated hash, increasing the iteration count may strengthen the password's protection against *brute-force attacks* [52,74]; on the other hand, this correspondingly increases the computation time. Further, in order to strengthen the security, a *salt value* (in cryptography: random data as additional input) needs to be added every time.

The increased computation time reduces the usability of the PM [43]. Therefore, two-level computation has been proposed, in which the value obtained from the first-level computation is cached and always used as an input for the second-level computation performed for each password [52]. Zhao and Yue [40] also recommended limiting the iteration count to around 1000 iterations, so that the actual computation time is around 10 s. This, again, has been identified as the *upper time limit before users' attention is distracted* [63].

4.11. Underutilised security resources

4.11.1. Tap underutilised resources without affecting portability

Most PMs are accompanied with their own security resources. In reality, though, several resources already exist in the desktop or portable devices that could be utilised to strengthen security. The problem while using such resources is that the deployability of the system is limited (see Section 4.3.1). Therefore, when utilising such resources, one must realise that they must not restrict the deployment of the PM to only a certain type of devices.

Some PM approaches have taken the opportunity to tap into the benefits of the under-utilised security resources and facilities that exist in such devices; in so doing, they improved their authentication mechanisms. Such approaches can be beneficial for resource-constrained devices, for example mobile phones and tablets. Wang et al. [93] utilised the Trusted Platform Module (TPM) to cache passwords in order to eliminate the time during which a plain-text password is saved in memory during its loading from the PM to the service where the user wants to log in. This step prevents the theft of unattended plain-text passwords in the memory. Similarly, Asokan and Ekberg [94] utilised not just *TPMs* but also the hardware-secured environment *M-Shields* to secure the processing and storage of passwords. *TruWalletM* [69] is a piece of hardware that assists in wallet-based authentication and employs *Trusted Execution Environment* for better protection of users' data. Al-Sinani and Mitchell [51], in turn, converted *CardSpace* to a PM, realising its security advantages.

5. Conclusions and future enhancements

The username–password pair of personal identification credentials is, by and large, the predominant method of authentication. Even so, the increasing number of login accounts that an average person possesses makes it a challenging task for a password user to generate and maintain strong and unique passwords for each and every account login. This is a security and privacy concern because the humans take huge risks by potentially exposing their personal credential information to various cyber-attacks. A PM is designed with the primary objective of providing relief and peace of mind to protected by passwords account holders. The PM could help humans in constructing, storing, and organising their passwords. Paradoxically, the very idea of the PM has yet been unsuccessful, and most of the login account holders still continue relying on their memory to invent, remember, and manage their passwords. The notable scepticism towards PM exists, essentially, because of the current problems associated with PM as exposed in detail in the previous sections of this research study. Hence, we attempted to address the problems of the current PMs use and for this we primarily focused on concerns that deal with the usability and user experience while at the same time tried to find how to enhance the PM's security. By researching, confronting and finally resolving these issues, it could be possible to establish and build trust between PM users and vendors, thereby increasing the PM application's acceptance among people who need password use and protection.

In this literature survey, we systematically reviewed and evaluated the available research literature, before selecting thirty-two articles that dealt with security and usability problems in PMs. The selected articles have recognised and revealed various existing problems in PM technology. However, some of the issues identified by those articles remain unsettled, because an issue identified as a problem by one article is not said to be a problem by another article(s). To clarify this ambiguity, firstly, we explored the effect of the stated issue on the usability and user experience of PMs. Secondly, we took into consideration the results and logical explanations from several other studies, which were developed from dissimilar cases but whose justifications were equally relevant and applicable to the case of PM technology. Our review does not merely summarise the articles, but rather attempts to collectively determine suitable mitigations for each problem identified by the researchers and authors of the reviewed articles.

It is worthy to note that the research papers we found in this literature review are rather biased towards security, that is, it clearly appears that a greater research emphasis has been put on security considerations. This is mainly because of a lack of usability and user-experience research output regarding PM design. In reality, though, combined research that integrates and inter-relates both usability and security perspectives is essential for this kind of security-sensitive applications and their users. The selected papers for this study are also often focused on the engineering perspective of password managers' design. This can be considered both as the limitation but also as a focused contribution of this study. As a limitation it can be considered because it shifts the acquired results towards the engineering aspects of the software quality design. Notwithstanding, when the primary goal of this study has been to provide useful information for the password manager developers, this can be considered a limitation. On the other hand, because throughout this study, the acquired results are compared to and contrasted against literature focused on more user-centric aspect of design, it is possible to see how important for the engineering focused research it would also be to combine more user-centric with software quality aspects in the design phase of a PM.

Another limitation of this study is that, because of the relatively small size and great variance in the literature, meaning that the PMs in the reviewed literature differed widely in their approaches (e.g., cloud-based, browser plug-in, wallet-based) and in the quality of their usability or security analysis, we were not able to establish any particular weight of importance to the usability or security features we describe herein. For this reason, we can only say that the importance of these (weighted) features varies depending on the type of the PM developed; and as many of the suggested features are usability-related, the target user segment also affects the importance of the features in specific PMs. Thus, the results of this review serve best as a collection of features requiring the software developers' attention and careful consideration during PM design phases. From the developers' perspective, it would certainly be useful to have more guidance about the importance and prioritisation of PM features; however, this is an item for a future research agenda.

From this research study it can be seen that even though a PM is not a large application, it nevertheless merits diversified attention on aspects associated with security, usability, and trust, while a PM's acceptance could also be related to cognitive and user-psychological issues. More importantly, most of the problems in existing PMs relate to features and functionalities such as 'optional auto-fill', 'weak master password', 'active and intervening dialog box used to get user's confirmation', and 'bulk migration', while other apprehensive issues can easily be resolved if the designer and developer stay cautious and perform some trials and run tests with the established algorithms and principles in order to implement them correctly. Nonetheless, there are still unsettled issues concerning users' distrust towards PM applications, the mental gap between PM utilities and their users, and bad practices in web design involving multiple stakeholders, which are time-consuming and difficult to resolve and/or eliminate. These problematic situations can only be resolved or tackled by appropriate actions and improved measures (or countermeasures).

From user-centric perspective, it is rather worrying that in engineering-focused PM security literature, the old security versus convenience confrontation still subtly seems to exists. In the eyes of developers, usability, and user-centric features might be seen as a security reducing factor, instead of emphasising their possibilities to support each other. If this kind of subtle notion is not recognised, developers may not actively seek the solutions which could promote both perspectives. Even when discussing direct security issues, such as multiple passwords [11,95], user-centric aspects should be taken into account. For example, one of the major factors that compromise the use and security of multiple passwords is the cognitive awareness that relates e.g. to users' memory and the behaviours users adopt to compensate for the memory's failures. Through studying memory and other cognitive elements that influence users' relations to password management, PM designers can increase their understanding of the PM users' needs and therefore make proposals to increase the security of the password authentication mechanisms [95]. Thus, instead of contrast, usability and user convenience should be seen as an essential enabler for security. It could even be asked whether this kind of shift in view could also have an impact on other PM problems, e.g. regarding their acceptance among users.

From developers perspective, it is clear that a higher priority should be given to those aspects through which PMs' acceptance can be promoted. In their study concerning technology rejection, Murthy and Mani [96] presented several possible reasons behind technology rejection, including the complexity of the technology, fatigue caused by too many features or ever-updating models/versions, and switching costs for the users, i.e., the relative difficulty for users to start using the new technology. They also emphasise the importance of socio-technical factors, such as the different needs of different user groups [96]. These are all factors, which can be improved by focusing more to the user-centric perspective during development.

Of all these factors, the literature reviewed in this study mainly considered complexity-related issues, emphasising and outlining the difficulties in functions' understanding and usage of PMs. There is, however, no clear attribution to these issues as main reasons for the low acceptance of PMs. The possibility that other technology rejection reasons, as presented in [96], may contribute to the low acceptance of PMs has been rather neglected and thus not been analysed in the reviewed literature we selected for this study. Consequently, the influence of these and other factors could also be isolated and studied in further research.

According to the literature reviewed in this study, *learnability* (i.e., helping the user learn how to use the PM) is one concern that requires immediate attention, because it can possibly promote PM acceptance. The users of PMs do not necessarily have the correct mental models for the functions of the applications or the degree of security that a PM offers. Recently, user studies have reported low perceived necessity scores for PMs [27, 28,42]. Therefore, increasing users' understanding and awareness may have a positive effect on the acceptance of PMs [19, 42]. Even Ciampa's [97] research results showed that training and continuing education of the users increase their awareness of the benefits of PM applications. Similarly, a lack of trust is, possibly, another component affecting the acceptance of PM technology. Thus, there is a need for more research concerning how relationships of trust [98,99] can be established and built between end-users, security applications and their designers and what kind of impact this has on an application's acceptance also in contingency situations [100] in life and under cognitive (e.g. memorability) challenges [95]. In order to design a well-trusted PM, it would also be helpful to research more thoroughly how trust and cognition are related to the usability and security of this application. Finally, we are optimistic that, through improving the security, usability, and learnability of PMs, and by understanding and bridging the communication/cognitive gaps between PM designers and users, trust can be built between them. Consequently, users can be encouraged to accept significant PM applications and functions and take on board cognitive and mental challenges.

The results of this literature review outline what has already been researched and established as knowledge about the use of PMs, during the recent years. The authors support that "security by design" roles and responsibilities to improve usability and security of system access are not a panacea and do not constitute a prescriptive approach of success. Further research should also be carried out on revealing more about password management technologies especially considering laboratory work and field experimentation, along with other research methodologies.

In the future, it can be expected that more complex situations will arise, such as multiple websites sharing the same login web page, websites that disable auto-complete, services that demand one-time passwords, or services that require multi-factor authentication. These situations demand security and trust. Accordingly, an important question is: How could a PM cater to these and similar situations?

Declaration of competing interest

None.

Appendix. Research methodology: Systematic literature review

In essence, we followed Webster and Watson [101] advice, who suggest the following simple steps for conducting research through a literature review: (i) identifying the relevant literature, (ii) structuring the review, (iii) framing the theoretical development of the article, (iv) evaluating the developed theory, and (v) creating discussion and conclusions.

To complement our review of software engineering literature, we also utilised the three stages for systematic literature reviews in software engineering. This comprises the stages of planning (developing the review protocol), conducting (study selection, data extraction, synthesis), and reporting (formatting and evaluating the report) [102]. Finally, for the purpose of assessing our protocol, we applied the systematic review protocol template developed by Biolchini et al. [103].

A.1. Literature sources and study selection

Our systematic literature review took place during 6/2015–10/2015, considering the relevant published research papers until 9/2015. Collecting relevant pieces of published research of high quality that capture the essence of the set research objective(s) is of paramount importance for the study's validity and reliability; this has very often been emphasised in literature reviews [104, 105]. This is because the quality of the particular literature used for the review plays a significant role in augmenting the researchers' knowledge and advancing the ultimate outcomes of the original research contributions [104]. In order to determine such a quality literature sample, the following basic aspects have to be clarified beforehand: *where* and *how* to locate the relevant literature [105]; and on *what basis* these should be selected, i.e., selection criteria. In the present study, the authors resolved to identify the relevant research literature through *literature repositories and search keywords*. The literature sources were selected considering the following criteria:

- A literature repository with a reputation for storing high-quality scientific literature in the field of computer science/engineering.
- A literature repository with a web-based search engine with the ability to enable searches with/through subject-specific keywords.

On the basis of the aforementioned criteria, the selected sources for relevant literature searches were: IEEE Xplore Digital Library, ACM Digital Library, Springer Link, and ELSEVIER ScienceDirect. The selection of databases is engineering focused. This is a deliberate choice and serves the main goal of the research study, which is to provide useful information regarding properties and socio-technical features especially for the developers of password managers. The developers (e.g. designers, programmers, testers, maintainers) of the password managers have an interest in both usability and security features that are related to password managers. However, throughout the development process of a password manager these particular stakeholders need to be heavily focused on security issues. For this reason the engineering perspective of password managers is central in the design and delivery phases of a password manager's lifecycle. Thus, though not the main goal of this study, the authors had a special interest to see how usability and user-centric issues are seen in engineering and security focused literature. Throughout the research study reported here, the researchers have reflected on their findings gathered from the selected literature against other studies not directly included in the reviewed literature. For example, the latter consider usability and user-centric issues related to passwords or software quality design. We particularly consider the engineering-focused perspective of this research and related reflections to other user-centric literature as a valuable contribution of this study. Thus, instead of just listing the properties and features, we try to synthesise these perspectives and to see why the features and properties suggested would be important for elevating both usability and security as built-in software quality features in password managers.

The authors selected the following search keywords: 'usability AND password manager', 'user experience AND password manager', 'security AND password manager', and 'password manager', where 'AND' is a Boolean search operator. The authors considered that these keywords were sufficient for accumulating a relatively complete census of the relevant research literature. The keywords 'usability AND password manager' and 'user experience AND password manager' were used to collect those papers dealing with and attempting to address usability problems in PM. Similarly, the keyword 'security AND password manager' was used to gather papers considering security issues in PM. The last keyword, 'password manager', was used for collecting all papers dealing with PM, so that the authors could cross-check and verify that important papers were not left out.

In the same way, the following criteria were applied for selecting the most relevant research articles:

- The inclusion of articles evaluating usability and/or security of any PM, or proposing a solution for some usability or security problem in PM.
- The exclusion of overlapping journal or conference articles, or articles containing only one or two paragraphs on PM.

We took an additional preventative measure to increase the possibility that the selected articles would actually be relevant while at the same time ensuring that no important articles were missed. Two of the authors performed the search and screening activities, independently. During this phase, the authors assessed the articles based on their title and abstract, and by quickly perusing and examining the research articles. Next, the authors compared their chosen articles. The articles that the two researchers had identified in common were selected for the final review reported here. Furthermore, after reading through each non-in-common research article once more, the authors decided together whether to include or exclude the article. Initially, the authors agreed to consider 45 articles. However, after reading those selected articles during the systematic review, the authors also encountered the following issues: (i) some articles had only one or two paragraphs of content relevant to the research objective; and (ii) a few articles were from the same authors discussing the same issue but were presented in two different conferences. In the end, the authors decided to exclude those articles, which contained only one or two paragraphs describing PM. Regarding redundant articles (i.e., the same article presented in two different conferences), we selected only one article from each redundant set for the review. Thus, the number of relevant articles for the systematic literature review was reduced to 32, as listed in Tables 1 and 2 along with their subject of emphasis.

It should be noted that the aforementioned 32 articles will serve as the primary source of knowledge herein, but that several other articles will also be utilised in a compare and contrast manner for enriching the outcomes of our research. The research and knowledge domain of *useable security*, to which our research work belongs, is a multi-disciplinary field. Thus, it should include the valid knowledge from various other fields (including cognitive science, software quality engineering, and other) so as to logically and coherently explain the security and usability problems of PMs and their potential countermeasures.

A.2. Data extraction and analysis

From the selected articles, we extracted the following data:

- Study description (issues studied, research study methodology);
- PM usability and/or security problems on which the study focuses; and
- Measures presented or suggested in the study to mitigate the particular problems related to usability and security.

Based on the extracted data and information, we proceeded to classification, and we formed a knowledge typology of 11 new knowledge categories for the different types of problems that we encountered in the PM research literature (see Sections 3.1–3.3). The authors scrutinised the new knowledge distilled and illustrated in the typology and subsequently summarised and condensed mitigation measures proposed in the literature for the particular PM problems considered in the context of this study. These final suggestions are expressed in terms of the quality features and properties that are indispensable for a PM to have/be in order to improve its usability (see Section 4). Further, the authors considered the conceptual constructs and concepts as focal to the organising framework of the review [see e.g. [101]]; that is, we determined that identifying the key concepts and developing a logical approach to group them was requisite and essential for refining and clustering the existing knowledge. The authors were also inspired by the method used by DeLone and McLean [106] to summarise and simplify the literature with a set of tables. A tabular representation of the results often presents an overview of the compared and contrasted research outcomes with conciseness and clarity.

References

[1] J. Bonneau, C. Herley, P.C. van Oorschot, F. Stajano, The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes, in: Proceedings of the IEEE Symposium on Security and Privacy, 2012, pp. 553–567. San Francisco, CA, USA, May 20-23.

[2] W.E. Burr, D.F. Dodson, F.M. Newton, R.A. Perlner, W.T Polk, S. Gupta, E.A. Nabbus, Electronic Authentication Guidelines: Recommendation of the National Institute of StandArds and Technology, NIST Special Publication 800-63-1, 2011.

[3] S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, S. Egelman, Of passwords and people: Measuring the effect of password-composition policies, in: In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 2011, pp. 2595–2604.

[4] R. Shay, S. Komanduri, P.G. Kelley, P.G. Leon, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, Encountering Stronger Password Requirements: User Attitudes and Behaviors, in: Proceedings of the 6th Symposium on Usable Privacy and Security, 2010, Redmond, WA USA, July 14–16.

[5] B.F. Barton, M.S. Barton, User-friendly password methods for computer-mediated information systems, Comput. Secur. 3 (3) (1984) 186–195.

[6] C. Kuo, S. Romanosky, L.F. Cranor, Human Selection of Mnemonic Phrase-Based Passwords, in: Proceedings of the 2nd Symposium on Usable Privacy and Security, 2006, pp. 67–78. Pittsburgh, PA, USA, July 12-14.

[7] M.L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L.F. Cranor, P.G. Kelley, R. Shay, B. Ur, Measuring Password Guessability for an Entire University, in: Proceedings of the ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 173–186. Berlin, Germany, November 4-8.

[8] J.O. Pilam, On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks, in: Proceedings of the 1st International Conference in Cryptology, 2000, pp. 67–79. Calcutta, India, December 10-13.

[9] J. Yan, A. Blackwell, R. Anderson, A. Grant, Password memorability and security: Empirical results, IEEE Secur. Privacy 2 (5) (2004) 25–31.

[10] S.Z.S. Idrus, E. Cherrier, C. Rosenberger, J.J. Schwartzmann, A review on authentication methods, Aust. J. Basic Appl. Sci. 7 (5) (2013) 95–107.

[11] L. Li, E. Berki, M. Helenius, S. Reijo, New Usability Metrices for Authentication Mechanisms, in: Proceedings of the SQM and INSPIRE International Conference, 2012, Tampere, Finland, August 20-23.

[12] E.E. Schultz, R.W. Proctor, M. Lien, G. Salvendy, Usability and security an appraisal of usability issues in information security methods, Comput. Secur. 20 (7) (2001) 620–634.

[13] D. Florencio, C. Herley, B. Coskun, Do Strong Web Password Accomplish Anything?, in: Proceedings of the 2nd USENIX workshop on Hot Topics in Security, 2007, Boston, MA, USA. August 7, 2007.

[14] K. Scarfon, M. Souppaya, Guide To Enterprise Password Management: Recommendations of the National Institute of StandArd Technology, NIST Special Publication 800-118, 2009.

[15] J. Bonneau, S. Schechter, Towards Reliable Storage of 56-Bit Secrets in Human Memory, in: Proceedings of the 23rd USENIX Security Symposium, 2014, San Diego, CA, USA, August 20-22.

[16] R. Holly, Project abacus is an ATAP project aimed at killing the password, 2016, http://www.androidcentral.com/project-abacus-atap-project-aimed-killing-password, (Accessed 15 Novemeber 2016).

[17] T. Simonite, Google wants to replace all your passwords with a ring, 2013, https://www.technologyreview.com/s/512051/google-wants-to-replace-all-your-passwords-with-a-ring/, (Accessed 15 November 2016).

[18] D. Florencio, C. Herley, A Large-Study of Web Password Habits, in: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 657–666. Banff, Alberta, Canada, May 8-12.

[19] I. Ion, R. Reeder, S. Consolo, No One can Hack My Mind: Comparing Experts and Non-Experts Security Practices, in: Proceedings of the Symposium on Usable Privacy and Security, 2015, pp. 327–346. Ottawa, Canada.

[20] A. Adams, M.A. Sasse, Users are not the enemy: Why users compromise computer security mechanisms and how to take remedial measures, Commun. ACM 42 (23) (1999) 41–46.

[21] S. Chiasson, P.C. van Oorschot, Quantifying the security advantage of password expiration policies, Des. Codes Cryptogr. 77 (2–3) (2015) 401–408.

[22] B. Schneier, Changing password, 2010, www.schneier.com, (Accessed 15 Novemeber 2016).

[23] P. Inglesant, M.A. Sasse, The True Cost of Unusable Password Policies: Password use in the Wild, in: Proceedings of the ACM Conference on Human Factors in Computing Systems, 2010, Atlanta, GA, USA, April 10-15.

[24] M.A. Sasse, I. Flechais, Usable security: Why do we need it? how do we get it?, Secur. Usability (2005) 13–30, O'Reilly.

[25] M.F. Grub, R. Carter, Single Sign-on and the System Administrator, in: Proceedings of the 12th System Administration Conference, 1998, pp. 63–86. Boston, MA, USA, December 6-11.

[26] J.J. Kay, Self-reported password sharing strategies, in: Proceedings of the ACM CHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 2011, pp. 2619–2622.

[27] A. Karole, N. Saxena, N. Christin, A Comparative Usability Evaluation of Traditional Password Managers, in: Proceedings of the 13th International Conference on Information Security and Cryptology, 2010, pp. 233–251. Boco Raton, FL, USA, October 25-28.

[28] K. Bicakci, N.B. Atalay, M. Yuceel, P.C. van Oorschot, Exploration and Field Study of a Password Manager using Icon-Based Passwords, in: Proceedings of the Financial and Cryptography Workshops, 2011, pp. 104–118. Rodney Bay, St. Lucia, February 28 - March 4.

[29] E. Hayashi, J.I. Hong, A diary of password usage in daily life, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 2011, pp. 2627–2630.

[30] S. Riley, Password security: What users know and what they actually do, 2006, http://usabilitynews.org/password-security-what-users-know-and-what-they-actually-do/, (Accessed 15 November 2016).

[31] N.J. Rubenking, Survey: Hardly anybody uses a password manager, 2015, http://uk.pcmag.com/opinion/40161/survey-hardly-anybody-uses-a-password-manager, (Accessed 16 November 2016).

[32] E. Stobert, R. Biddle, The Password Life Cycle: User Behaviour in Managing Passwords, in: Proceedings of the Symposium on Usable Privacy and Security, 2014a, Menlo Park, CA, USA, July 9-11.

[33] Password Boss, Survey finds vast majority of americans memorize or write passwords on paper, 2015, https://www.passwordboss.com/news/survey-finds-vast-majority-of-americans-memorize-or-write-passwords-on-paper/, (Accessed 29 July 2015).

[34] M. Ciampa, M. Revels, J. Enamait, Online versus local password management applications: An analysis of user training and reactions, J. Appl. Secur. Res. 6 (4) (2011) 449–466.

[35] M. Fagan, Y. Albayram, M.M.H. Khan, R. Buck, An investigation into users' considerations towards using password managers, Human-Centric Comput. Inform. Sci. 7 (1) (2017) 12.

[36] R. Böhme, J. Grossklags, The Security Cost of Cheap User Interaction, in: Proceedings of the New Security Paradigms Workshop, 2011, Marin County, CA, USA, September 12-15.

[37] L.F. Cranor, S. Garfinkel, Secure or usable?, IEEE Secur. Privacy (2004) 16–18.

[38] A. Whitten, J.D. Tygar, Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0, in: Proceedings of the 8th USENIX Security Symposium, 1999, pp. 169–184. Washington, D.C., USA, August 23-26.

[39] K. Bicakci, M. Yuceel, B. Erdeniz, H. Gurbaslar, N.B. Atalay, Graphical Passwords as Browser Extension: Implementation and Usability Study, in: Proceedings of the 5th Symposium on Usable Privacy and Security, 2009, Mountain View, CA, USA, July 15-17.

[40] R. Zhao, C. Yue, Toward a secure and usable cloud-based password manager for web browsers, Comput. Secur. 46 (2014) 32–47.

[41] M. Aliasgari, N. Sabol, A. Sharma, Sesame: A secure and convenient mobile solution for passwords, in: Proceedings of the 1st Conference on Mobile and Secure Services, 2015, pp. 1–5. Gainesville, FL, USA, February 20-21.

[42] S. Chiasson, P.C. van Oorschot, R. Biddle, A Usability Study and Critique of Two Password Managers. In Proceedings of the 15th USENIX Security Symposium, 2006, pp. 1–16. July 31- August 3.

[43] D. Ziegler, M. Rauter, C. Stromberger, P. Teufl, D. Hein, Do You Think Your Passwords are Secure? In Proceedings of the International Conference on Privacy and Security in Mobile Systems, 2014, pp. 1–8. Aalborg, Denmark, May 11-14..

[44] K. Yee, Aligning security and usability, IEEE Secur. Priv. 2 (5) (2004) 48–55.

[45] F. Stajano, M. Spencer, G. Jenkinson, Q. Stafford-Fraser, Password-Manager Friendly (PMF): Semantic Annotations to Improve the Effectiveness of Password Managers, in: Proceedings of the International Conference on Passwords, 2014, pp. 61–73. Trondheim, Norway, December 8-10.

[46] S. Fahl, M. Harbach, M. Oltrogge, T. Muders, M. Smith, Hey, You, Get Off of My Clipboard- on how Usability Trumps Security in Android Password Manager, in: Proceedings of the Financial Cryptography and Data Security, 2013, Okinawa, Japan, April 1-5.

[47] K.P. Yee, K. Sitaker, Passpet: Convenient Password Management and Phishing Protection, in: Proceedings of Symposium on Usable Privacy and Security, 2006, Pittsburgh, PA, USA, July 12–14.

[48] E. Stobert, R. Biddle, A Password Manager that Doesn'T Remember Passwords, in: Proceedings of the New Security Paradigms Workshop, 2014b, pp. 39–52. Victoria, BC, Canada, September 15–18.

[49] D. McCarney, D. Barrera, J. Clark, S. Chiasson, P.C. van Oorschot, Tapas: Design, Implementation, and Usability Evaluation of a Password Manager, in: Proceedings of the 28th Annual Computer Security Applications Conference, 2012, pp. 89–98. Orlando, Florida, USA, December 3-7.

[50] M. Nasirinejad, A.A. Yazdi, SASy Username and Password Management on the Cloud, in: Proceedings of the International Conference on Cyber Security, Cyber Warfare and Digital Forensics, 2012, pp. 242–246. Kuala Lumpur, Malaysia, June 26-28.

[51] H.S. Al-Sinani, C.J. Mitchell, Extending the scope of cardspace, in: Proceedings of the 4th International Conference on Security of Information and Networks, 2011, pp. 235–238. Sydney, Australia, November 14-19.

[52] J.A. Halderman, B. Waters, E.W. Felten, A Convenient Method for Securely Managing Passwords, in: Proceedings of the 14th International Conference on World Wide Web, 2005, pp. 471–479. Chiba, Japan, May 10-14.

[53] E. Erdem, K.O. Kucukkurt, K. Samurkas, E. Kanargi, U. Celikkan, A Smart Card Based Single Sign-on and Password Management Solution as a Browser Extension, in: Proceedings of the International Conference on Education and Management Technology, 2010, pp. 539–543. Cairo, Egypt, November 2-4.

[54] X. Wang, Z. Han, D. Zhang, IDKeeper: A Web Password Manager with Roaming Capability Based on USB Key, in: Proceedings of the International Conference on Industrial Control and Electronics Engineering, 2012, Xi'an, China, August 23-25.

[55] R. Biddle, M. Mannan, P.C. Van Oorschot, T. Whalen, User study, analysis, and usable security of passwords based on digital objects, IEEE Trans. Inform. Forensic Secur. 6 (3) (2011) 970–979.

[56] S. Gaw, E.W. Felten, Password Management Strategies for Online Accounts, in: Proceedings of the Second Symposium on Usable Privacy and Security, 2006, pp. 44–55. Pittsburgh, PA USA, July 12-14.

[57] L. Lazar, O. Tikolsky, Personalized cognitive passwords: An exploratory assessment, Inform. Manag. Comput. Secur. 19 (1) (2011) 25–41.

[58] Net Market Share, Market share statistics for internet technologies, 2019, https://netmarketshare.com, (Accessed 1 January 2019).

[59] F. Boukayoua, B. De Decker, V. Naessens, A Keyboard that Manages Your Passwords in Android, in: Proceeding of the International Conference on Privacy and Security in Mobile Systems, 2014, pp. 1-4. Aalborg, Denmark, May 11-14.

[60] J. Zhang, S.F. Jones, uKey: A Unified Key Management and Authentication Automation Middleware for Heterogeneous System, in: Proceedings of the International Conference on Information Technology, 2007, pp. 943–944. April 2-4.

[61] M. Wu, R.C. Miller, G. Little, Web Wallet: Preventing Phishing Attacks by Revealing Users' Intentions, in: Proceedings of the Symposium of Usable Privacy and Security, 2006, Pittsburgh, PA, USA, July 12-14.

[62] E. Berki, H. Isomäki, M. Jäkälä, Holistic communication modelling: Enhancing human-centred design through empowerment, in: D. Harris, V. Duffy, M. Smith, C. Stephanidis (Eds.), Cognitive, Social and Ergonomic Aspects, Vol 3 of HCI International, 22-27 2003, University of Crete At Heraklion, Lawrence Erlbaum Associates Inc, 2003, pp. 1208–1212.

[63] J. Nielsen, Response times: The 3 important limits, 1993, http://www.nngroup.com/articles/response-times-3-important-limits/, (Accessed 15 November 2016).

[64] R.E. Gehring, M.P. Toglia, G.A. Kimble, Recognition memory for words and pictures at short and long retention intervals, Memory Cognit. 4 (3) (1976) 256–260.

[65] Apple Inc, Custom keyboard, 2016, https://developer.apple.com/library/content/documentation/General/Conceptual/ExtensibilityPG/CustomKeyboard.html (Accessed 15 November 2016).

[66] C. de Souza, E. Trainer, S. Quirk, D. Redmiles, Exploiting the relationship between software dependencies and coordination through visualization, 2008, https://pdfs.semanticscholar.org/16b3/96566f02ecd351e41098d66c36433366fc9f.pdf, (Accessed on 26 2018).

[67] M. Belguidoum, F. Dagnat, Dependency management in software component development, Electron. Notes Theor. Comput. Sci. 182 (2007) (2007) 17–32.

[68] K. Bicakci, N.B. Atalay, H.E. Kiziloz, Johnny in Internet Café: User Study and Exploration of Password Autocomplete in Web Browsers, in: Proceedings of the 7th ACM Workshop on Digital Identity Management, 2011, pp. 33–42. Chicago, IL, USA, October 21.

[69] S. Bugiel, A. Dmitrienko, K. Kostiainen, A.-R. Sadeghi, M. Winandy, TruWalletM: Secure Web Authentication on Mobile Platforms, in: Proceedings of the International Conference on Trusted System, 2011, pp. 219–236. Beijing China, December 13-15.

[70] B. Yang, H. Chu, G. Li, S. Petrovic, C. Busch, Cloud Password Manager using Privacy-Preserved Biometrics, in: Proceedings of IEEE International Conference on Cloud Engineering, 2014, pp. 505–509. Boston, MA, USA, March 10-14.

[71] E. Stobert, R. Biddle, Visual End-User Security, in: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, 2012, pp. 233–234. Innsbruck, Austria, September 30 - October 4.

[72] H. Bojinov, E. Bursztein, X. Boyen, D. Boneh, Kamouflage: Loss-Resistant Password Management, in: Proceedings of the 15th European Symposium on Research in Computer Security, 2010, 6345, pp. 286–302. Athens, Greece, September 20-22.

[73] LastPass, Lastpass: Generate password, 2015, https://lastpass.com/generatepassword.php, (Accessed 15 November 2016).

[74] R. Zhao, C. Yue, K. Sun, Vulnerability and risk analysis of two commercial browser and cloud based password managers, ASE Sci. J. 1 (4) (2013) 1–15.

[75] M. Bishop, D.V. Klein, Improving system security via proactive password checking, Comput. Secur. 14 (1995) 233–249.

[76] X. Carnavalet, M. Mannan, A large-scale evaluation of high-impact password strength meters, ACM Trans. Inform. Syst. Secur. 18 (1) (2015) 1–32.

[77] A. Herzberg, Why johnny can't surf (safely)? attacks and defenses for web users, Comput. Secur. 28 (1–2) (2009) 63–71.

[78] K. Bhargavan, A. Delignat-Lavaud, Web-Based Attacks on Host-Proof Encrypted Storage, in: Proceedings of the 6th USENIX Conference on Offensive Technologies, 2012, pp. 97–104. Bellevue, WA, USA, August 6-7.

[79] F.F. Moghaddam, O. Karimi, M. Hajivali, Applying a Single Sign-on Algorithm Based on Cloud Computing Concepts for SaaS Applications, in: Proceedings of the IEEE 11th Malaysia International Conference on Communications, 2013, pp. 335–339. Kuala Lumpur, Malaysia, November 26-28.

[80] R.K. Sinha, An effective communication helps building trust and improving performance of a service industry: A literature review and theory building, Paripex-Indian J. Res. 3 (8) (2014) 2014.

[81] P. Gasti, K. Rasmussen, On the Security of Password Manager Database Formats, in: Proceedings of the 17th European Symposium on Research in Computer Security, 2012, pp. 770–787. Pisa, Italy, September 10-12.

[82] Camp L.J., in: Rino Falcone (Ed.), Designing for Trust. Trust, Reputation and Security: Theories and Practice, Springer-Verlang (Berlin), 2003.

[83] J. Riegelsberger, M.A. Sasse, J.D. McCarthy, The mechanics of trust: A framework for research and design, Int. J. Human-Comput. Stud. 62 (3) (2005) 381–422.

[84] S. Chaudhary, The use of usable security and security education to fight phishing attacks, Ph.D. Thesis, University of Tampere, Finland, 2016, p. 55.

[85] P. Nikander, K. Karvonen, Users and trust in cyberspace, Secur. Prot. 2133 (2001) 24–35.

[86] S. Cassidy, Lostpass, 2016, (Accessed 15 November 2016). seancassidy.me.

[87] S. Chaudhary, E. Berki, L. Li, J. Valtanen, Time up for phishing with effective anti-phishing research strategies, Int. J. Human Capit. Inform. Technol. Prof. 6 (2) (2015) 49–64.

[88] R. Dhamija, J.D. Tygar, The Battle Against Phishing: Dynamic Security Skins, in: Proceedings of the Symposium on Usable Privacy and Security, 2005, pp. 77–88. Pittsburgh, PA, USA, July 6-8.

[89] M. Volkamer, K. Renaud, Mental models–general introduction and review of their application to human-centred security, in: In Number Theory and Cryptography, Springer, 2013, pp. 255–280.

[90] Z. Li, W. He, D. Akhawe, D. Song, The Emperor's New Password Manager: Security Analysis of Web-Based Password Managers, in: Proceedings of the 23rd USENIX Security Symposium, 2014, San Diego, CA, USA, August 20-22.

[91] W3C, G200: Opening new window and tabs from a link only when necessary, 2015, https://www.w3.org/TR/WCAG20-TECHS/G200.html, (Accessed 15 Novemeber 2016).

[92] D. Silver, S. Jana, E. Chen, C. Jackson, D. Boneh, Password Managers: Attacks and Defenses, in: Proceedings of the 23rd USENIX Security Symposium, 2014, San Diego, CA, USA, August 20-22.

[93] H. Wang, Y. Guo, X. Zhao, X. Chen, Keep Passwords Away from Memory: Password Caching and Verification using TPM, in: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications, 2008, GinoWan, Okinawa, Japan, March 25-28.

[94] N. Asokan, J.-E. Ekberg, A Platform for OnBoard Credentials, in: Proceedings of the 12th International Conference on Financial Cryptography and Data Security, 2008, pp. 318–320. Cozumel, Mexico, January 28-31.

[95] N. Woods, Improving the security of multiple passwords through a greater understanding of the human memory, Ph.D. thesis, University of Jyväskylä, Finland, 2016, https://jyx.jyu.fi/dspace/h{and}le/123456789/51882.

[96] S.R. Murthy, M. Mani, Discerning rejection of technology, SAGE open: 1-10, 2013.

[97] M. Ciampa, Are password management applications viable? an analysis of user training and reactions, Inform. Syst. Edu. J. 9 (2) (2011) 2–13.

[98] E. Berki, H. Isomäki, A. Salminen, Quality and Trust Relationships in Software Development, in: Proceedings of the Software Quality Management XV: Software Quality in the Knowledge Society, 2007, pp. 47–66. Tampere, Finland, August 1-2.

[99] R. Yahalom, B. Klein, T. Beth, Trust Relationships in Secure systems—A Distributed Authentication Perspective, in: Proceedings of the IEEE Symposium on Research in Security and Privacy, 1993, pp. 150–164. Oakland, CA, USA, May 24-26.

[100] L. Li, E. Berki, M. Helenius, S. Ovaska, Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate?, Behav. Inform. Technol. 33 (11) (2013) 1136–1147.

[101] J. Webster, R.T. Watson, Analyzing the past to prepare for the future: Writing a literature review, MIS Quart. 26 (2) (2002) 13–23.

[102] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering: EBSE technical report, 2007, EBSE 2007-001, Ver. 2.3.

[103] J. Biolchini, P.G. Mian, A.C.C. Natali, G.H. Travassos, Systematic review in software engineering, Technical Report ES 679/05, System Engineering and Computer Science Department COPPE/UFRJ, 2005.

[104] M.J. Grant, A. Booth, A typology of reviews: An analysis of 14 review types and associated methodologies, Health Inform. Librar. J. 26 (2) (2009) 91–108.

[105] Y. Levy, T.J. Ellis, A systems approach to conduct an effective literature review in support of information systems research, Inform. Sci. J. Int. J. Emerg. Tran Discip. 9 (1) (2006) 181–212.

[106] W.H. DeLone, E.R. McLean, Information systems success: The quest for the dependent variables, Inform. Syst. Res. 3 (1) (1992) 65–95.

Dr. Sunil Chaudhary is Head of the Department of Computer Science at the Deerwalk Institute of Technology, Kathmandu, Nepal. He received his Ph.D. degree on cybersecurity in 2016 and an M.Sc. in Software Development in 2012 from the University of Tampere, Finland. He completed a BEng in Computer Engineering in 2007 from Pulchowk Campus, Nepal. His research focuses on useable privacy and security, online identity, cyber trust, IoT security, and cybersecurity education and awareness. He has published several peer-reviewed journal and conference papers in the field of cybersecurity.

Tiina Schafeitel-Tähtinen M.Sc. is a Project Researcher at Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland. She received her M.Sc. on Astronomy in 2003 from the University of Turku, Finland, and B.Sc. (Tech) in Information and Knowledge Management in 2013 in Tampere University of Technology, Finland. She is currently finalising her M.Sc. (Tech) in Tampere University, Finland. Her research interests focus on IoT privacy and security, usable security, and cybersecurity education.

Dr. Marko Helenius is University Teacher at Tampere University, Tampere, Finland. He is responsible of the master's degree program in information security. He completed his Ph.D. from University of Tampere in 2002 about computer viruses. He became interested about usability aspects of information security in 2006. He has been active in many national research programs obtaining research funding including Future Internet, Cloud Software, Internet of Things, Need4Speed and Cyber Trust.

Dr. Eleni Berki is Adjunct Professor of Software Quality and Formal Modelling at the University of Jyväskylä, Finland. Her research areas are software/total quality engineering and learnability in diverse contexts. She belongs to more than 50 international committees and has organised/chaired many conferences. She worked in the UK, Greece, China and Finland. Current teaching and research focus on: security testing, confidentiality in e-health, trust/privacy, social engineering, free/open source software, social innovations. Dr Berki has around 150 multidisciplinary publications with hundreds of references/citations to them. She supervised 8 and examined 9 Ph.D. theses; she currently supervises many interdisciplinary Ph.D. researchers.