Tapas: Design, Implementation, and Usability Evaluation of a Password Manager

Daniel McCarney David Barrera Jeremy Clark
Sonia Chiasson Paul C. van Oorschot
Carleton University, Ottawa, ON, Canada
{dmccarney,dbarrera}@ccsl.carleton.ca
{clark,chiasson,paulv}@scs.carleton.ca

ABSTRACT

Passwords continue to prevail on the web as the primary method for user authentication despite their well-known security and usability drawbacks. Password managers offer some improvement without requiring server-side changes. In this paper, we evaluate the security of *dual-possession authentication*, an authentication approach offering encrypted storage of passwords and theft-resistance without the use of a master password. We further introduce Tapas, a concrete implementation of dual-possession authentication leveraging a desktop computer and a smartphone. Tapas requires no server-side changes to websites, no master password, and protects all the stored passwords in the event either the primary or secondary device (*e.g.,* computer or phone) is stolen. To evaluate the viability of Tapas as an alternative to traditional password managers, we perform a 30 participant user study comparing Tapas to two configurations of Firefox's built-in password manager. We found users significantly preferred Tapas. We then improve Tapas by incorporating feedback from this study, and reevaluate it with an additional 10 participants.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*authentication*; H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors*

Keywords

Password managers, usable security, smartphones

1. INTRODUCTION

A large number of research contributions have been made toward increasing the security and usability of password-based authentication [5]. Many of these attempts require account providers to change how they handle authentication by augmenting or outright replacing passwords; *e.g.,* one-time passwords, dual-factor, single-sign on, biometrics, graphical passwords, *etc.* Recently, researchers have argued that despite the wide-held sentiment from the security and usability communities that passwords need to be replaced, the incumbency, familiarity, and low cost of traditional passwords continues to hamper widespread adoption of an alternative, as well as a lack of consensus on what exactly the alternative should provide [12].

We are interested in practical solutions combining easy deployability with security and usability. For this reason, we presently exclude from interest proposals requiring server-side changes. Previous research under this constraint focuses on storing and retrieving passwords for users (*e.g.,* password managers), strengthening password quality (*e.g.,* randomly-chosen, cryptographically processed, or site specific), and encoding alternative authentication mechanisms into passwords (*e.g.,* graphical or object-based passwords). These three classes of solutions tend to address orthogonal issues and can be complementary. We focus on the first, not necessarily excluding the others.

Password managers are designed to relieve password fatigue and reduce log-in time. They can also indirectly facilitate better password quality and a reduction in password reuse. A naive password manager simply stores the passwords, while security-conscious managers lock the stored passwords under a master password. Password managers may also integrate other techniques to strengthen or encode passwords, including those mentioned above.

Password managers have certain drawbacks. To use a password manager, existing accounts must be migrated into the manager and potentially replicated across multiple devices. In the event an adversary gains access to the manager's storage, a naive password manager offers no protection making it a high value target. With a master password, the manager provides at best a level of protection dependent on the strength of the master password against an offline attack. This is assuming the theft does not occur when the manager has unlocked the passwords for the duration of a session, in which case the protection offered is greatly reduced. Password managers that maintain unprotected passwords during use do not always clearly indicate to the user the current state (locked or unlocked) of the system.

In this paper, we present a type of password manager that combines usability advantages of the naive password manager with protected storage. Passwords are protected

ACSAC '12 Dec. 3-7, 2012, Orlando, Florida USA

against offline attacks with a strong encryption key which the user need not remember and decryption requires the control of two independent devices. Operation of this type of manager requires no master password, only control of both devices. If any one of these two devices is stolen, the adversary cannot recover the passwords in practice.

We consider a specific instantiation of this type of manager, Tapas,[1] and present its design, implementation, and analysis. Tapas is a smartphone-assisted password manager for a computer that requires no server-side changes. It maintains security of the managed passwords by encrypting and storing the passwords on a smartphone, and keeping the decryption key inside the browser on the paired computer. Tapas is resistant to theft in the following sense: an adversary must steal both the smartphone and the user's computer to gain access to managed credentials. Tapas is designed to provide a simple mental model of "sending" the password from the phone to the login screen on a separate device, maintaining no cached master password and not storing any credentials on disk. Unlike a hashing-based solution, Tapas does not preclude memorization of passwords and login outside of the Tapas system.

We present the results of a 30 participant user study evaluating a Tapas prototype and comparing it to the built-in Firefox password manager both with and without the use of a master password. Our study found that in general users have little knowledge of the benefits password managers provide or the means by which they protect passwords. This leads to an underutilization of browser password managers and low enrollment in opt-in master password protection. Participants selected to use Tapas rated their enjoyment of the process higher than participants' ratings for the other managers. Further they were able to utilize Tapas successfully and without error to store credentials and log into websites, despite any perceived initial difficulties.

Our primary contributions are as follows:

1. We study the notion of *dual-possession authentication* which has received little attention in the literature. We develop a threat model for using it in conjunction with a password manager and find it offers a practical set of security and usability properties.

2. To allow concrete evaluation of this notion, we implement a dual-possession password manager (Tapas) using a Firefox extension on a primary device and an Android app on a secondary device. Although the idea of requiring two devices for password retrieval is simple, the implementation involves several subtle security and networking details. Tapas requires no server-side changes, no master password, and offers theft-resistance for the managed passwords.

3. We validated the feasibility of Tapas through an in-person user study with 30 participants comparing Tapas to two browser password managers. Users of Tapas were successful in using the system, even without prior knowledge of password managers. Using insights from the initial study we improve the Tapas design and then conduct a 10 participant follow-up study to evaluate it, finding it improves user's understanding of the system.

[1] Tap-based authentication using a smartphone

2. RELATED WORK

Recently, researchers have been encouraged to consider that the persistence of passwords is not incidental [12]: their advantages as a well-known, firmly entrenched incumbent (*e.g.,* widespread familiarity, marginal cost per user) outweigh the costs of implementing an alternative, and this is asserted to be unlikely to change in the near-term. Despite a wide-held sentiment from the security and usability communities that passwords must be replaced, there is little consensus on the actual harm incurred by password breaches [11] (passwords may not be the last line of defence), what fraction of breaches is attributable to each threat vector, and thus, what alternative schemes should prioritize.

Password Managers.

The category of password managers (client-side tools to assist password-based authentication) is broad and contains many different, generally complimentary, techniques. Examples of services they provide are password strengthening through iterated hashing [9, 16, 6], phishing protection through site-specific passwords [16, 19], and converting other types of authentication into passwords [14, 17, 2].

The other main service a password manager can provide is the storage and retrieval of passwords, which is the focus of this paper. Major browsers (*e.g.,* Internet Explorer, Firefox, Chrome and Safari) offer a built-in password wallet. These wallets store the passwords on the user's computer in either plaintext (often by default), or encrypted under a master password. The browser may also offer cloud storage protected by a typical user account (*e.g.,* password and recovery questions). Third-party applications like LastPass,[2] and 1Password,[3] focus on cross-browser, cross-platform support and cloud synchronization.

Wallets protected under a master password have two drawbacks. First, many implementations do not use encryption correctly—many mobile password wallets were demonstrated to be insecure[4]. A second drawback is that a user-chosen password may not resist an offline attack if the wallet is stolen. To address this issue, Bojinov *et al.* [3] propose the use of password decoys to force the adversary back to using online attacks.

Device-based Authentication.

Several papers have explored *device-based authentication*; we restrict our coverage primarily to those involving possession of a smartphone. With dual-factor authentication, a secondary token is required in addition to a password. One use of a smartphone in authentication is to generate such tokens (*e.g.,* Google Authenticator) or to receive them over SMS. Phoolproof [15] uses a smartphone as an authentication token to augment traditional password authenticaiton with the goal of preventing phishing through the use of public key cryptography and end-to-end TLS. Pico [18] uses a cluster of devices, including smartphones and other smart devices, in proximity of each other to allow authentication. All of these require server-side changes.

[2] `https://lastpass.com`

[3] `http://1password.com`

[4] A. Belenko and D. Sklyarov. "Secure password managers and military-grade encryption on smartphones: Oh, really?" *Blackhat Europe*, 2012.

Usability & Comparison Frameworks.

A number of papers have compared password managers through user studies. Gaw and Felten [8] surveyed users on password use and found few users employed a password manager instead of relying on memory alone. Chiasson *et al.* [7] examined two password managers, finding significant usability and security failings related to entry of the master password as well as inaccurate/incomplete mental models of the software. Bicakci *et al.* [1] examined the user interface of browser-based managers and the tendency of users to inadvertantly save private information on a public computer.

Karole *et al.* [13] performed a comparative user study between an online, a mobile, and a portable USB password manager. They found non-technical users preferred keeping their credentials on mobile phone based password managers, but had difficulty entering passwords of sufficient strength on the mobile device.

Bonneau *et al.* [5] propose a framework for evaluating authentication solutions based on usability, security and deployability properties. They rank 35 representative schemes (including 2 password managers: Firefox and LastPass). We evaluate Tapas using this framework in Section 8.

3. DUAL-POSSESSION AUTHENTICATION

Storing passwords, whether software-based or a post-it note with passwords written on it, is based on the principle of authentication by something you have: the contents of the password 'wallet.' The primary security vulnerability of an unprotected wallet is theft. This is traditionally addressed by adding a master password, something you know, for additional protection. However this protection is best considered a deterrent, as theft allows offline attacks on the master password. Given a user-chosen master password this may mean fewer than 20 bits of security [4].

By contrast, password management that requires simultaneous access to multiple paired devices offers a level of theft-resistance. Strictly speaking, this is not dual-*factor* authentication because the factors are of the same type: 'something you have.' For comparative reasons, we refer to it as dual-possession authentication. We assume that for most users, a large proportion of log-ins occur on a small number of devices. Dual-possession authentication is designed to improve the usability of the log-in process from these devices without negative impact on the rest.

Dual-possession authentication involves two applications, a Manager and a Wallet, on different devices and offers the three depicted protocols for managing the passwords: Pair (Protocol 1), Store (Protocol 2), and Retrieve (Protocol 3). These protocols are designed to achieve a relatively simple goal: by stealing the data of either the Manager or the Wallet, an adversary cannot determine the stored password for any given account with any greater success than attacking the account directly. This is achieved by encrypting each password with a key held by the Manager and storing the resulting ciphertext on the Wallet. By stealing the Manager, the adversary obtains the decryption key but not the ciphertexts to decrypt, and by stealing the Wallet, the adversary only has a set of ciphertexts resistant to offline attacks.[5] The effect of malware which remains resident on the Manager is discussed in Section 5.

To ensure these devices can run Store and Retrieve over a potentially hostile network, we require Pair to be performed on an authenticated and secret out-of-band (AS-OOB) channel [10]. The pairing is essentially an assignment of public keys that will be used by each device to authenticate the other during network communication. In Tapas we instantiate the AS-OOB channel by having the Manager display a QR code which is scanned by the Wallet. Once paired, the devices will establish a mutually-authenticated end-to-end secure channel (*e.g.,* TLS with a Diffie-Hellman key exchange[6]) before exchanging any encrypted passwords. This allows the devices to securely tunnel their communication through various network devices that may assist them in establishing a connection.

4. TAPAS

We instantiate the protocols and general notion of dual-possession authentication (Section 3) to construct Tapas. In Tapas, password management is handled across both the user's desktop PC and a paired smartphone. In this Section we describe the implementation details of Tapas, and explain how the 3 protocols of dual-possession authentication are enacted.

While we have chosen to implement the components of Tapas using Mozilla Firefox and the Google Android platform, the architecture is independent of these choices. We expect that an extension for Chrome, Safari, and other extensible browsers could be developed for users who do not use Firefox as their primary browser. Similarly, non-Android smartphone platforms could be used.

Firefox Extension. In Tapas, the Manager device is implemented as a Firefox browser extension on the users's desktop PC. It is written in JavaScript and XML User Interface Language (XUL), utilizing interfaces exposed by Firefox for use by extensions. It is multi-platform and requires no native code, allowing the extension to be installed on Windows, Linux or OSX.

Android Application. The Wallet device is implemented as an application for the Android smartphone platform. The Wallet is written using Java for devices running Android versions 2.3 and above. Based on platform distribution statistics[7] Tapas is compatible with over 81% of Android devices worldwide (as of Sept 4, 2012).

Rendezvous Server. In order to allow direct communication between two devices potentially located on separate networks, the Tapas architecture employs a Rendezvous Server to facilitate network address translation (NAT) traversal and hole punching. The Rendezvous Server is considered untrusted and external to the management of passwords; no unprotected data is transmitted through it.

In addition to negotiating network connections, the Rendezvous Server is responsible for federating communication with the Google Cloud to Device Messaging (C2DM) service. Google requires all applications utilizing C2DM to

[5]For two devices, this approach seems more straightforward than using distributed/threshold decryption with key shares.

[6]The RSA-based key exchange in TLS does not provide perfect forward secrecy, which is necessary for security as discussed in Section 5.

[7]Google platform versions distribution: `http://goo.gl/rQ2gv`

Protocol 1: Pairing Manager and Wallet

User action: Upon a user choosing to set-up a new Wallet, the following protocol is initiated by the Manager.

Communication channel: A one-way authenticated and secret out-of-band (AS-OOB) channel from the Manager to the Wallet.

1. The Manager generates an authentication key pair for itself $\langle pk_m, sk_m \rangle$ and sends its public key pk_m to the Wallet.
2. The Manager generates an authentication key pair for the Wallet $\langle pk_w, sk_w \rangle$ and sends the pair to the Wallet.
3. The Manager generates a secret key k for a symmetric key authenticated encryption scheme $\mathsf{Enc}_k()$.

Output: The Manager stores $\langle pk_m, pk_w, sk_m, k \rangle$ and erases sk_w. The Wallet stores $\langle pk_m, pk_w, sk_w \rangle$.

Protocol 2: Storing a Password

User action: Upon a user choosing to save a password p_i, the following protocol is initiated by the Manager.

Communication channel: A mutually-authenticated secure channel with perfect forward secrecy between the Manager and the Wallet. The participants, respectively, identify themselves with pk_m and pk_w.

1. The Manager takes user password p_i (entered by user) and site information s_i and computes $c_i = \mathsf{Enc}_k(p_i \| s_i)$.
2. The Manager sends $\langle c_i, s_i \rangle$ to the Wallet.
3. The Wallet prompts the user to create a tag t_i for referencing the site, using s_i to suggest a value for the tag.

Output: The Manager erases $\langle p_i, s_i, c_i \rangle$. The Wallet stores $\langle t_i, c_i \rangle$ and erases s_i.

Protocol 3: Retrieving a Password

User action: Upon a user choosing a password for retrieval, the following protocol is initiated by the Wallet.

Communication channel: A mutually-authenticated secure channel with perfect forward secrecy between the Manager and the Wallet. The participants, respectively, identify themselves with pk_m and pk_w.

1. The Wallet retrieves the c_i value associated with the tapped t_i, and sends c_i to the Manager.
2. The Manager decrypts and authenticates c_i to retrieve s_i and p_i.
3. The Manager checks that s_i matches the site information for the current site that the browser is visiting.
4. The Manager transfers the user password p_i to the site.

Output: The Manager erases $\langle p_i, s_i, c_i \rangle$.

pre-register with the service to obtain an API authentication token allowing access to the service. In order to avoid embedding a C2DM API token into the Manager extension we defer C2DM pushes to the Rendezvous Server, allowing the Manager to send a push message to a device through it. Tapas relies on C2DM strictly as a means of launching the Wallet application automatically without requiring a long-running listener service on the smartphone.

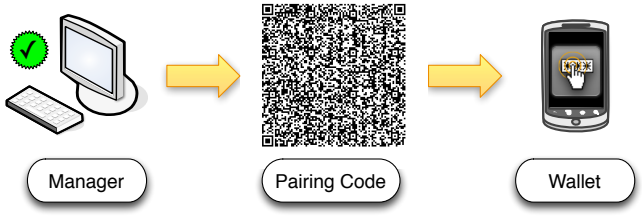


Figure 1: Setting up an out-of-band communication channel initiated (depicted by the checkmark) by the Manager, for pairing the devices.

4.1 Setup

To set up Tapas, the user installs the Firefox extension and the Android app using the standard software installation procedure for each respective platform. Once installed, the devices are paired using Protocol 1. The Manager computes the authentication key pairs and generates a self-signed TLS certificate for both public keys. It embeds networking information (IP address and port number), a fingerprint of its own certificate, and the Wallet's certificate and corresponding secret key into a QR code. The generated code is displayed on the computer screen, forming a unidirectional AS-OOB channel (Figure 1).

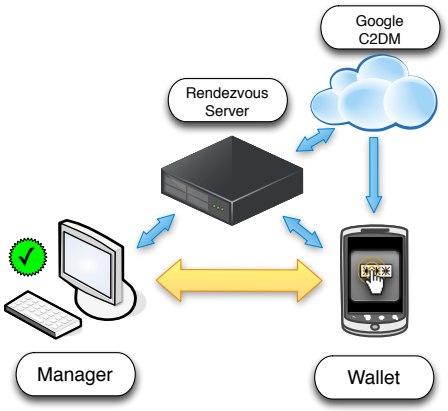


Figure 2: Setting up a two-way network communication channel initiated by the Manager, for password storage on the Wallet.

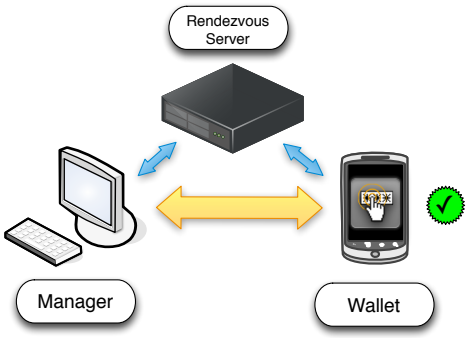


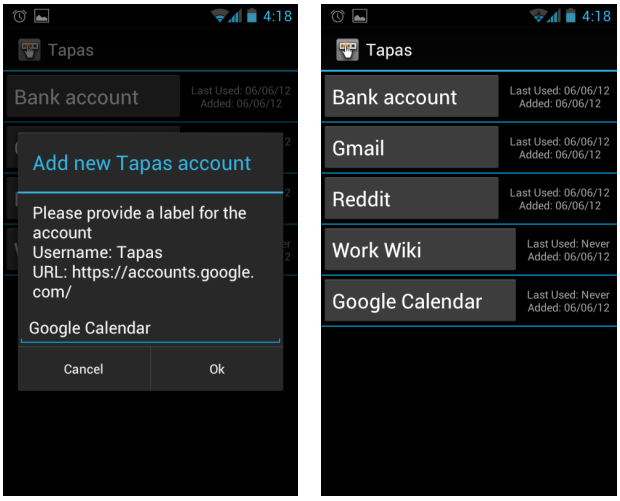
Figure 3: Setting up a two-way network communication channel initiated by the Wallet, for password retrieval from the Wallet.

The user now opens the Wallet app on the smartphone. The Android app defaults to displaying a "begin pairing" screen until the user successfully pairs the application with an instance of the Firefox browser extension. When the user presses the "pair" button on the Wallet app a QR code scan is initiated via the ZXing QR Code application.[8] ZXing utilizes vibration and auditory feedback to inform the user when a code has been successfully scanned in order to help make the process intuitive. After the Wallet app reads the QR code, the Wallet decodes from it the IP address and listening port of the Manager browser extension as well as the certificate material.

4.2 Account Import

When the Manager detects a username/password being submitted to a website, it temporarily saves the values as they are submitted and offers the user a chance to store the account credentials in the Wallet. This is done by presenting a non-obtrusive drop-down notification similar to the built-in Firefox password manager. If the user accepts the offer then the Manager contacts the Rendezvous Server to initiate a C2DM push to launch the Wallet application on

[8] `http://code.google.com/p/zxing/`



(a) Save account (b) Account list

Figure 4: Screenshots of the Tapas Wallet.

the paired smartphone (Figure 2). The smaller arrows represent communication used to launch the Wallet automatically (via C2DM and the Rendezvous Server) and to negotiate a direct network connection between the Manager and the Wallet (pictured as the larger arrow). Both the Manager and the Wallet rely on outgoing connections to the Rendezvous Server to negotiate direct communication through NAT, similar to traditional NAT hole punching techniques involving a third party.

At this point the Manager and Wallet follow Protocol 2 to securely transfer encrypted credentials. First the Manager encrypts the site information (URL, username, password) using AES in GCM mode with a symmetric encryption key known only to it. The encrypted ciphertext is then transmitted from the Manager to the Wallet over a mutually-authenticated TLS connection (using a Diffie-Hellman ciphersuite) where both certificates are pinned to the device certificates previously established during pairing.

When new account information is transmitted by the Manager to the Wallet the user is presented a chance to provide a meaningful label for the account (see Figure 4(a)). By default the label text is populated with the site URL; for privacy reasons the suggested label may be renamed. Each account in the Wallet has a large touch region displaying the user-chosen label for the account. Additionally, each account displays the date on which it was last used, and the date on which the account was added to the Wallet. Accounts are listed in order of most recent use (see Figure 4(b)).

4.3 Password Retrieval

When the user taps an account label in the Wallet on their smartphone, the stored credential associated with the account is transmitted to the paired Manager by Protocol 3. Figure 3 shows how the communication channel is set up. The Manager and the Wallet rely on communication with the Rendezvous Server (smaller arrows) to negotiate a direct network connection between one another (larger arrow). Assuming the user's browser is open to the correct website (*i.e.*, is viewing the URL associated with the tapped account) then the username and password field on the website are filled by

the Manager and submitted. The result of the login process is returned to the Wallet in order to display meaningful status messages to the user via the Wallet UI. All communication between the Manager and the Wallet is carried out over a mutually-authenticated TLS connection.

Tapas requires the user to signal their intent on both the browser and the smartphone before a login can occur. When a user transmits account credentials from the Wallet to the Manager, the latter decrypts the account information and verifies that the associated URL matches the currently open web page before filling the username and password. If a URL other than the one associated with the decrypted account details is open in the browser the Manager displays a message indicating that the correct URL must be opened before a login can occur (see Figure 5). This prevents accidental logins or a situation in which the user is away from their computer and accidentally triggers a login to a website by tapping their smartphone.

4.4 Limitations

The Tapas implementation is not without limitations. It relies on the availability of a network connection and the Rendezvous Server server to function. Given that the purpose of Tapas is authenticating with web resources the lack of an internet connection would likely preclude authentication regardless of Tapas. In order to use Tapas, both the paired devices must be present and usable. In the case of the smartphone Wallet this means the battery must be charged. The present implementation of Tapas allows pairing between only one computer and one smartphone, preventing use with multiple machines. Implementation of a full fledged secret sharing scheme could address the multiple device scenario.

5. SECURITY EVALUATION

We evaluate the security of Tapas relative to other types of password wallets, both with/without a master password. We assume the existence of an adversary with the ability to intercept, record, and modify any communication between the Manager and the Wallet except the one-time pairing process (Protocol 1) conducted over an AS-OOB channel (implemented in Tapas as a visible QR code). The pairing process allows the devices to establish public keys for authentication, enabling the devices to communicate confidently in the presence of an active adversary on standard communication channels (as in Protocols 2 and 3). In addition to granting access to the communication channel between the Manager and Wallet, we allow the adversary physical possession (theft) of either the Manager or the Wallet. Tapas offers no security against a loss of both.

Resistance to Theft.

If a device with an unprotected password wallet is lost, there is no inherent protection of the passwords stored in the wallet. The use of a master password offers some protection, however the adversary may still be able to conduct an offline attack that will recover all the passwords if the master password is not strong. On the other hand, a strong master password introduces usability issues related to memorability and accurate entry. In Tapas, theft-resistance is provided against offline attacks without the user having to remember any passwords.

Smartphones (which hold the Wallet in Tapas) are frequently lost and stolen. Passwords in the Wallet are encrypted in such a way as to be indistinguishable from randomness without the decryption key. This is a consequence of using the GCM mode of operation which provides indistinguishability under chosen plaintext attacks. The randomly generated 128 bit AES decryption key is held by the Manager and not contained on the smartphone, therefore the stored passwords are protected against even an offline attack. Further, aside from the user-chosen tag, all information about the sites that correspond to the stored passwords are also encrypted, providing privacy against individuals with passive access to the smartphone.

The Wallet also contains a wallet authentication key. Loss of this key to an adversary would allow the adversary to masquerade as the Wallet. The Wallet's only functionality is receiving and pushing encrypted passwords to and from the Manager. The ciphertext of each stored password is authenticated by the Manager's decryption key; a feature of GCM that prevents the decryption of any modified ciphertext. If a modified ciphertext caused the password portion to be submitted to a non-HTTPS site or one controlled by the adversary, the adversary could learn it. GCM does not allow the plaintext to be manipulated in structured ways, unlike other modes (*e.g.,* ECB or CBC). More generally, authenticated encryption ensures that the Manager cannot be a useful decryption oracle to the adversary.

The user's computer (which hosts the Manager in Tapas) may also be lost, stolen, or given away without the proper deletion of memory. In this case, the adversary recovers the symmetric AES encryption key k. k would allow the adversary to recover each password p_i given its ciphertext c_i, however the set of c_i are stored by the Wallet. Recall our assumption that the adversary can store all past communications observed over the secure channel in Protocols 2 and 3. Preventing such an adversary from learning the set of c_i is why it is essential that the encrypted passwords are communicated over an encrypted channel even though they are themselves already encrypted. Further, the adversary also learns the authentication key sk_m. If the design of the secure channel provided only authentication and encryption (using *e.g.,* the RSA-based ciphersuites in TLS), sk_m would be sufficient to derive the session key used in past executions of Protocol 2 and 3, allowing the adversary to recover the set of c_i. To thwart this line of attack, the secure channel in Protocols 2 and 3 have perfect forward secrecy (using a Diffie-Hellman key exchange in TLS) to ensure past session keys cannot be derived from a compromised sk_m.

Resistance to Malware.

Like other password managers, Tapas cannot protect stored passwords from persistent malware on the user's computer. The passwords must, at some point, be in plaintext for submission to the web service as per the current design of most web services (this is true if users memorize their passwords as well). With a traditional password manager, malware can immediately recover all the stored passwords as soon as the master password is entered. With Tapas, individual site passwords can only be recovered as they are used. If the malware is detected and removed, unused passwords will remain safe by repeating Protocol 1. Tapas also provides protection against specific forms of attack like hardware keystroke loggers and shoulder surfing.

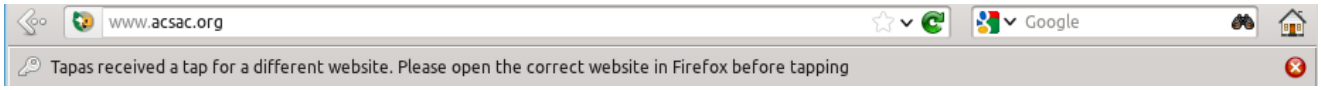


Figure 5: The notification the Tapas extension displays for mismatched user intent.

6. USABILITY EVALUATION

To evaluate the usability of Tapas, we conducted an in-lab user study with 30 participants. (Our later 10 participant follow-up study is presented in Section 7.2.) Our study design was approved by our university's Ethics Review Board.

6.1 Overview

We selected a between-subjects design where participants were randomly assigned to one of three conditions: Firefox with no master password (NMP), Firefox with a user-chosen master password (MP), and Tapas. Each participant was asked to complete a set of core tasks using the assigned password manager (see Section 6.5). We collected data through observation of the participants' interaction with the password manager as well as through questionnaires before and after each session. We did not mention to participants that Tapas was our own application so as to not bias participants.

We opted for an in-person study rather than a Mechanical Turk study for two reasons. First, Tapas requires the use of an Android phone and installation of an app not available in the market. In our study, we provided participants in the Tapas condition with an Android phone pre-loaded with the application. Second, conducting an in-person study allowed for direct observation of user behaviour when using the password managers.

6.2 Participant Demographics

We recruited a total of 30 participants (17 males, 13 females) through posters around the university campus and mailing lists. Most (age 18 to 42, $\bar{x} = 24.13$) were university students or staff. Participants had a wide range of backgrounds including accounting, psychology, theoretical physics, criminology, music, computer science and math.

Devices, operating systems and browsers. The majority of participants described themselves as Windows users (86%) and Google Chrome users (76%). A smaller number used MacOS and Linux regularly and one participant did not know how to tell what operating system he/she used. Chrome was the most popular browser, followed by Firefox which was used regularly by 50% of participants. Internet Explorer, Opera, and Safari were less popular. 28 participants owned a cell phone or smartphone. Smartphone OSs were approximately evenly split among Android, iOS and Blackberry.

Passwords and password managers. When asked to describe their use of passwords on the Internet, participants reported having between 3 and 40 ($\bar{x} = 11.53$) accounts that require passwords, and between 2 and 25 ($\bar{x} = 5.73$) unique passwords for those accounts, implying password reuse. 70% reported changing their passwords very rarely or never. 2 of 30 participants commented that the only time they change their passwords is if they are forgotten.

Participants in general had a poor understanding of the term *password manager*. Only 2 reported using a password manager, but several explained during their session that they do in fact use the browser's built-in password manager.

6.3 Study Setup

An Ubuntu Linux computer with Firefox pre-installed was used to perform the study. The Firefox history and settings were restored to defaults between sessions, so every participant saw the same "clean install" version of the browser.

Tapas requires that both a Firefox extension be installed on the desktop PC and an Android application be installed on the smartphone. We chose to avoid testing this common software installation process, and focused on the initial (post-install) setup and use. Thus, the Tapas Manager extension and Android Wallet app were installed, but not configured, before user sessions.

Three blog websites were created for the study (hereafter referred to as blog A, B and C). Each blog was designed to require an account (login) for posting comments.

6.4 Session

In-person sessions lasted 25 minutes on average and participants were paid $10. Participants were asked to read and sign an informed consent form which stated that their passwords would be logged, but not disclosed. Each participant was asked to read a short explanation of password managers in general, followed by a description of the specific password manager selected for their session. These text descriptions[9] were written with the objective of helping the user build an accurate mental model of the password manager rather than focusing on technical accuracy.

6.5 Tasks

During each session, participants were asked to perform the following tasks, after being given verbal instructions only at the start of each task (the examiner's involvement being minimal thereafter).

1. **Configure password manager:** If applicable (*i.e.,* in the MP and Tapas conditions), perform initial configuration. For MP, enable the master password protection in the Firefox settings and create a master password. For Tapas, scan the QR code displayed in the Tapas → Preferences pairing screen.

2. **Create and store accounts into the password manager:** Visit blogs A and B, find the register or create account section and select a username and password. When prompted, save the account into the password manager.

3. **Migrate an existing account into the password manager:** Participants were given a username and password and asked to pretend they already had an account on blog C. Proceed to log in to blog C and save the account into the password manager. Log out of blog C.

[9]Available at `http://pdox.ca/tapasscript`

4. **Log in to blogs:** After a distraction task,[10] participants were asked to log into and comment on the three blogs, in the following order. First log in to blog C. Next, close and reopen Firefox. Next, log in to blog A, followed by blog B. Closing and reopening Firefox was done to help users (particularly those in NMP and MP) identify when their passwords were accessible. Firefox, when configured with a master password, prompts the user for the master password on the first login after the browser is opened.

7. RESULTS OF FIRST USER STUDY

After completing the in-lab tasks, participants were given a post-test questionnaire designed to capture their comments and experience while interacting with the password manager. This section presents the questionnaire results and observations made by the examiner during the sessions.

Statistical tests. The Kruskal-Wallis non-parametric one way analysis of variance test was applied with a p-value of 0.05 considered significant to determine whether responses from participants in each condition were independent for a given question. If this test yielded a statistically significant p-value, one of the conditions was independent. To determine which condition(s) were independent, individual pairs were further analyzed with a non-parametric Mann-Whitney test to identify the specific conditions that differed.

7.1 Post-test Questionnaire

Perceived usability of password manager setup. Overall, there were no issues with the setup process in any of the three conditions. Participants in all conditions rated the ease of setup (on a 4-point Likert scale) as either easy or very easy with no obvious trend. Application of the Kruskal-Wallis test found no significant evidence that Tapas differed from the other managers in ease of setup.

Participants were asked if they thought they would be able to set up the password manager on their own. In all conditions participants responded positively. Participants in the MP condition noted that while the setup process was easy, finding the master password checkbox in the Firefox preferences was not straightforward, and that if the session information did not guide them to the right setting it would have been more difficult.

Comments from Tapas participants included the following: "This was a really easy step, I had never done it before but it was extremely simple", "It was pretty straightforward. It is easy to use", "The use of the QR code was a great tool to pair the devices. The set up was easy and quick".

Participants in the NMP condition (no setup required) were also allowed to enter comments regarding setup. A few voiced concerns about the simplicity of setup, stating that it was almost "too simple", and that you may actually end up accidentally saving passwords with the manager you didn't intend to save, a concern echoed in the literature [1]. Comments like these reinforce the Tapas design feature that requires signalled intent on both devices prior to saving account information or logging in using the password manager.

The QR code pairing method appears to be very intuitive and the audio and vibration feedback was verbally noted by some participants as useful. Tapas users mentioned feelings of accomplishment, as though they had achieved something complicated with little effort. On the other hand, the Firefox master password setup screen displays a password meter which no user was able to fill. Some users typed in two or three different passwords to try to increase the measure of the password strength bar.

Perceived usability of password saving. We asked participants to rate their agreement (on a 5 point Likert scale) with the following statement: "Saving a password was easy when I created a new account and migrated an existing account". Participants did not find Tapas any more difficult than the other two conditions, although some participants had to be reminded to complete the saving process on the phone after clicking the save button in the Tapas extension.

We also asked participants if they thought that using the password manager they were assigned made logging in easier than logging in without one. Based on verbal feedback from participants in the MP condition it appears some users perceive lower ease of use due to the master password being requested when the password manager is invoked for the first time. For the Tapas condition, one participant verbally noted that "logging in with Tapas would take longer since you would have to take out your phone and launch the app every time you log in".

User affectation. We asked users to rate how much they enjoyed using the password manager overall. Participants liked Tapas more than MP, and liked MP more than NMP. For this question, the Mann-Whitney test resulted in statistically significant difference between the Tapas condition and both the MP and NMP conditions ($p = 0.04891$ and $p = 0.006826$ respectively). For the NMP condition, 6 participants reported enjoying the password manager and one participant highly disliked it. In the MP condition, 5 participants enjoyed using the system, and 5 "somewhat enjoyed" it. Participants in the Tapas condition universally (10/10 participants) rated their enjoyment at the highest level of the Likert scale, demonstrating high user affectation in comparison to both the MP and NMP conditions.

General participant observations. In the free-form comment field at the end of the survey, several participants across all conditions expressed a desire to know where the passwords were stored. Some participants in the Tapas condition did not notice the pop-down message asking them to save their passwords. Considering this input, we modified Tapas as described in Section 7.2.

A second major theme of comments was related to losing access to the password manager. Several users stated that they probably would never use a password manager because if they lost access to it, they would lose access to all their accounts. While in reality users could still employ the password recovery mechanisms offered by individual websites, these comments highlight the importance of addressing a loss-of-access scenario. This motivates future work to enable an encrypted backup feature for Tapas.

7.2 Improving Tapas – Follow Up User Study

We revised the Tapas Firefox extension incorporating feedback from our 30 participant user study, specifically address-

[10]The distraction task had participants count down from 100 in decrements of 3 to help remove the recently created passwords from the participants' working memory.

ing issues with the poor visibility of the pop-down messages. For the message offering the user the chance to save an account (user ID, password) with Tapas the background was changed from gray to blue, and the label was changed from "Save with Tapas" to "Save to phone". The error condition pop-down messages were changed to have a red background. We revised the help text used in the Android Wallet application to clarify the goal of the pairing process.

To test the usability of the revised version of Tapas, we recruited 10 (6 female, 4 male) additional participants for the Tapas condition only. The study methodology was identical to the previous study, with the only change being the updated Tapas Firefox extension. For the most part, participants in the new study provided confirmation of the earlier ease of use and affectation findings. For brevity, we only present noteworthy results.

User attention. Observing participants during the second study confirmed that the blue message attracted participant's attention to the "save password" prompt. One participant remarked that the font size for the message was too small. Only one participant had to be reminded to look for the pop-down message after registering an account.

Mental model. The post-test survey attempted to capture the mental model participants had while using the password managers. All 10 participants in the second study answered correctly when asked "Where were your passwords stored?". In the first study, only 6 of 10 participants in the Tapas condition correctly mentioned the phone. While not statistically significant with 10 participants, we believe a larger sample would likely demonstrate that renaming the save button to "Save to phone" had a strong impact on the users' understanding of how the password manager works. The updated button label clearly explains where passwords are going when the button is clicked. In contrast, participants in the NMP and MP conditions answered this question correctly 50% of the time. Incorrect answers included some participants stating passwords were stored "in cyberspace", "on the website memory" and "no idea". We attributed this to Firefox's ambiguous "Remember password" button label.

7.3 Ecological Validity

Regarding demographics, most participants reported using Chrome as their primary browser, as well as not using a password manager. Thus, the lab study introduced these participants to both a new browser and a new password manager. This may have overloaded participants' memory, moving their attention away from the password manager or otherwise introduced a confounding effect.

The websites used were purpose-built blogs, and thus account integrity was not highly valued by users. Participant interaction with these sites, particularly in relation to password choice, may have been influenced by the lack of personal importance the blogs offered.

Some participants mentioned that the websites used in the study behaved strangely while logging in. Our sites were designed with minimal functionality requirements. Thus, when a user successfully logged in, the login form would be replaced with a message saying "successfully logged in", rather than returning to the password protected resource. Some participants failed to notice the change in login status, and were confused because they thought the login had failed.

8. COMPARISON SUMMARY

In this Section, we evaluate Tapas using the Usability-Deployability-Security (UDS) framework of Bonneau *et al.* [5]. Table 1 rates Tapas on the 25 benefits (properties) comprising the framework. For space, we cannot compare Tapas to all 35 authentication methods presented in the UDS paper. We focus on the incumbent (ordinary passwords) and the Firefox password manager from our user study.

Tapas addresses the usability issue of recalling an ever-growing number of passwords, without resorting to reuse. Relative to ordinary unmanaged passwords this benefit comes at the cost of interacting with a smartphone. In the event that access to the smartphone is lost, passwords need to be individually recovered using existing recovery mechanisms. Adding accessibility features to Tapas for disabled users is future work. When using a password manager, the stored passwords themselves can always be attacked directly. For this reason, password managers cannot improve on certain security properties of passwords. Tapas does provide phishing protection by ensuring stored passwords are only ever submitted to the exact site (determined by the site's URL and SSL/TLS certificate) they were registered with. Additionally the password cannot be observed externally when a user logs in with Tapas.

Surprisingly Firefox without a master password and Firefox with a master password rate equivalently using the UDS framework except for two properties: Memorywise-Effortless and Physically-Effortless. With a master password Firefox receives an empty circle rather than a filled circle due to the recall/entry of the master password. While a master password increases the security of stored passwords in the event of unauthorized access (*e.g.,* computer theft, in-person use, or through exposed backups) the UDS framework is not fine-grained enough to distinguish this security benefit. We opt to discuss just the master password version of the Firefox manager in Table 1.

Relative to Firefox MP, Tapas does not require a master password but does require interaction with a smartphone. On security, Tapas offers resilience to external observation. The framework does not distinguish that the stored ciphertexts in Tapas are resilient to an offline attack if the wallet is stolen, whereas in MP an offline attack on the master password coupled with access to the browser reveals all stored passwords at once. Similar to Firefox MP, Tapas receives an empty circle in the Physically-Effortless[11] and the Nothing-to-Carry[12] columns. Additionally, malware capable of recovering the Firefox master password can immediately learn all stored passwords, while malware on Tapas results in the gradual disclosure of passwords only as they are used.

For many cases, Tapas does not preclude composition with other mechanisms for improving security. For example, it can be used for per-site hash-based passwords, randomly generated passwords, or remembering the password portion for dual-factor authentication. Tapas could additionally generate a backup of the wallet's stored ciphertexts protected by a master password for recovery.

[11]We consider scrolling equivalent to pushing a button per the Physically-Effortless definition. Removing the phone from pocket is equivalent to removing a YubiKey or similar dongle.

[12]We consider the Tapas desktop component beyond the scope of Nothing-to-Carry, as Pico-siblings are likewise ignored.

Scheme	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party	Requiring-Explicit-Consent	Unlinkable
	Usability								Deployability						Security										
Tapas	●	●	○	○	●	○	●		○	●	●		●	●	●	○					●	●	●	●	●
Passwords			●		●	●	○	●	●	●	●	●	●	●		○						●	●	●	●
Firefox (MP)	○	●	○	○	●	●	●		●	●	●		●	●	○	○					●	●	●	●	●

Table 1: Evaluation of Tapas using the Usability-Deployability-Security framework for comparative evaluation of password alternatives [5]. The second and third rows are taken from [5] for reference comparison to Tapas.

9. CONCLUSION

We view the proliferation of "always keep me logged in" options and the interest in password managers and federated identity as evidence that the repetitive recall and typing of passwords is unpleasant for users. We designed Tapas to be compatible with password-based authentication, while relieving users of traditional password memory burdens. Tapas avoids the use of a master password— a setting which users found difficult to locate on existing password managers, does not offer strong protection against offline attacks, and may be inadvertently disclosed by users. Additionally with Tapas, users can walk away from their computers without exposing stored passwords that may be temporarily unlocked—every login requires an explicit action by the user. In the future, we intend on continuing the development of Tapas, as well as exploring other dual-possession schemes, specifically to facilitate logging in to accounts on mobile devices.

10. ACKNOWLEDGEMENTS

We thank Joseph Bonneau, Nitesh Saxena and the anonymous reviewers for useful feedback. This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)—the second author through a Canada Graduate Scholarship; the third through a Postdoctoral Fellowship; the fourth and fifth through Discovery Grants and as Canada Research Chairs. We also acknowledge partial support from NSERC ISSNet.

11. REFERENCES

[1] K. Bicakci, N. B. Atalay, and H. E. Kiziloz. Johnny in internet café: user study and exploration of password autocomplete in web browsers. In *Digital Identity Management*, 2011.

[2] R. Biddle, S. Chiason, and P. C. van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys*, 44(4):1–41, 2012.

[3] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh. Kamouflage: Loss-resistant password management. In *ESORICS*, 2010.

[4] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE Symposium on Security and Privacy*, 2012.

[5] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In *IEEE Symposium on Security and Privacy*, 2012.

[6] X. Boyen. Halting password puzzles – hard-to-break encryption from human-memorable keys. In *USENIX Security*, 2007.

[7] S. Chiasson, P. C. van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *USENIX Security*, 2006.

[8] S. Gaw and E. W. Felten. Password management strategies for online accounts. In *SOUPS*, 2006.

[9] J. A. Halderman, B. Waters, and E. W. Felten. A convenient method for securely managing passwords. In *WWW*, 2005.

[10] T. Halevi and N. Saxena. On pairing constrained wireless devices based on secrecy of auxiliary channels: the case of acoustic eavesdropping. In *CCS*, 2010.

[11] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *NSPW*, 2009.

[12] C. Herley and P. C. van Oorschot. A research agenda acknowledging the persistence of passwords. *IEEE Security & Privacy*, 10(1):28–36, 2012.

[13] A. Karole, N. Saxena, and N. Christin. A comparative usability evaluation of traditional password managers. In *ICISC*, 2011.

[14] M. Mannan and P. van Oorschot. Digital objects as passwords. In *HotSec*, 2008.

[15] B. Parno, C. Kuo, and A. Perrig. Phoolproof phishing prevention. In *Financial Cryptography*, 2006.

[16] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In *USENIX Security*, 2005.

[17] N. Saxena and J. H. Watt. Authentication technologies for the blind or visually impaired. In *HotSec*, 2009.

[18] F. Stajano. Pico: no more passwords! In *Security Protocols*, 2011.

[19] K.-P. Yee and K. Sitaker. Passpet: convenient password management and phishing protection. In *SOUPS*, 2006.