MassPass: an Alternative to Password Managers

Alex Lesman, Giuseppe Zingales and Adrian Miguel

Table of Contents:

Introduction

Passwords have been the staple of authentication for as long as computing has existed. However, as users use more and more web based services they are being required to remember more and more passwords. This leads to users forgetting and reusing passwords, among other problems. One solution to this problem has been password managers which solve the problem by storing encrypted versions of the users authentication information and then decrypt them using some master password. This works, but has various drawbacks. Our goal was to address this problem by designing a novel alternative to password managers which would generate passwords based on publicly known credentials (domain and username) and the addition of a master password.

Design

Masspass differs from other password managers in that it stores absolutely no information about the user, and generates credentials on the fly. This works by using a secure one-way hash of the users credentials and master password to create a domain specific password without leaking information or inconveniencing the user.

Password Generation

The core of MassPass is the password generation algorithm. This takes place in three stages. First, MassPass collects the domain, username and master password and simply concatenates them. Then, this concatenated string is processed by a bCrypt hash. BCrypt is an application of the Blowfish encryption scheme and is specifically designed to be computationally intensive and difficult to implement on GPU. Specifically, MassPass runs bCrypt with a load factor of 11 which equates to 2048 rounds of Blowfish, which takes approximately half a second on a modern CPU. This generates a 184 bit hash represented as a 31 character, base 64 string. This string is then adapted to conform to various password formats and requirements while losing as little

information as possible.

A websites password requirements are specified through the plugin or website UI. The user selects the allowed character sets (lower-case, upper-case, numbers and symbols), the required number of characters from each character set and finally a maximum length for the password. The password formatting algorithm starts with this ruleset and the 184 bit hash. It first generates the required characters by dividing the hash by the cardinality of that character set and uses the remainder to select a character. This process is repeated to generate the required characters. Then a complete character set is formed by combining all of the allowed characters and then the division and remainder is applied until either the maximum password length is met of the 184 bit hash is depleted. Finally this newly formed password is deterministically shuffled using the hash in order to re-arrange the characters. This is done to re-arrange the required characters which would otherwise always end up at the front of the password string. Once this is done MassPass will have generated a password that uses as much as possible of the available password space or the full 184 bits generated by the hash.

Chrome Plugin

To use the chrome plugin, the user must follow three simple steps. Firstly, he must navigate to the website he would like to log in to. Secondly, he must click on the MassPass plugin icon. And lastly, he must input his master password (MassPass) and username. The username is website dependent, while the MassPass should always remain the same. Once the user completes these steps, the plugin will generate the password on the fly and enter your username and password into the website for you. If the website has no password field to fill in, the plugin will generate a popup that allows the user to copy the password onto the clipboard. The situation that generates the popup is suboptimal because the user is no longer resilient to physical observation. For convenience as well as additional security from physical observation of the master password the MassPass plugin caches the master password in javascript for the duration of the chrome session. This may carry some security risk, but because this is only stored in ram and is removed when chrome exits this risk is relatively small and the benefits are well worth it. For the particularly paranoid user this could be an optional feature in the future.

Website

If a user is unable to use the chrome plugin the user can visit a website. This website does much of the same things as the chrome plugin in client side javascript. Unlike the plugin however the website asks the user to copy-paste the url in which is inconvenient and potentially vulnerable to attack (html5 allows the modification of site address). Also, the website puts the generated password in the field for the user to copy-paste into a site. It is then up to the user to ensure that they are protected from physical observation, and to ensure that the password is cleared from the clipboard if they are on a public machine. However, having the website provides the user with additional freedom when using MassPass.

Analysis

The paper “The Quest to Replace Passwords” developed a scheme for judging authentication schemes across a variety of usability, deployability and security metrics. We analyzed MassPass on how well it satisfies these criteria.

Usability

MassPass is *Quasi-Memorywise-Effortless* because users are only required to remember one password, and *Scalable-for-Users* because it can generate passwords to authenticate the user for an arbitrary number of sites. Because MassPass does not store user data there it is *Nothing-to-Carry*, the user need to either install the plugin on the other machine or visit the MassPass website. It is *Quasi-Physically-Effortless* as it only requires typing in your master password once and is *Easy-to-Learn* because it closely follows the standard login flow. MassPass automatically populates password field after the username is entered in the plugin making it more efficient than regular passwords. Because the user always uses the same infrequently typed password it also has *Infrequent-Errors*. Finally, MassPass does not have *Easy-Recovery-From-Loss*, if the user forgets their master password there is absolutely no way to recover their account specific passwords.

Deployability

Because MassPass is a client side applcation it provides almost full deployability benefits. It does not have *Maturity*, which is not very relevant client side. It also only provides *Quasi-Browser-Compatibility* because without the chrome plugin the user must use the website version which only provides the core features.

Security

MassPass is *Quasi-Resilient-to-Physical-Observation* because the master password only needs to be typed in once per session. While the master password is secure, the underlying passwords are only *Quasi-Resilient-to-Throttled-Guessing* and *Quasi-Resilient-to-Unthrottled-Guessing* because the generated passwords are extremely difficult to guess, although restrictive password requirements and poor security policy on a website could compromise this. MassPass is not *Resilient-to-Internal-Observation* because a keylogger would be able to capture a user's master password. It is *Resilient-to-Leaks-from-Other-Verifiers* because the leaking of one generated password does not compromise any other passwords. It is *Resilient-to-Phishing* because we assume that websites are using TLS and the chrome extension pulls the domain name from the page. It is obviously *Resilient-to-Theft* and *No-Trusted-Third-Party* by design. MassPass *Requires-Explicit-Consent* from the user because it does not hit the login button but merely fills in the password field. Finally MassPass is completely *Unlinkable*, it is impossible from a single generated password to determine the master password or ascertain any relation to another generated password.

Comparison to other password mechanisms

While MassPass is not strictly better than any of the other password alternatives, it does have a unique set of advantages. The key advantage for usability is that MassPass is *Nothing-to-Carry*, at most a user will need to visit the MassPass website to be able to login from another computer. The largest usability flaw is that the loss of the master password is absolutely catastrophic, the user would need to go through the account recovery flow for all the accounts for which MassPass was used.

Property	Password	Firefox	LastPass	MassPass
Memory Wise Effortless	✗	−	−	−
Scalable for Users	✗	✓	✓	✓
Nothing to Carry	✓	−	−	✓
Physically Effortless	✗	−	−	−
Easy to Learn	✓	✓	✓	✓
Efficient to Use	✓	✓	✓	✓
Infrequent Errors	−	✓	✓	✓
Easy Recovery from Loss	✓	✗	−	✗

MassPass compares favorably against other password managers in terms of deployability by being *Quasi-Browser-Compatible*. Because there is a website version available, MassPass can be used with any browser, albiet with some loss of convenience and functionality. This is an advantage over other password managers which leave the user completely unable to authenticate if they are using a non-compatible browser. MassPass is not *Mature*, however this

should not affect deployability.

Property	Password	Firefox	LastPass	MassPass
Accessible	✓	✓	✓	✓
Negligible Cost-Per-User	✓	✓	−	✓
Server Compatible	✓	✓	✓	✓
Browser Compatible	✓	✗	✗	−
Mature	✓	✓	✓	✗
Non-Proprietary	✓	✓	✗	✓

MassPass is universally more secure than other password derived authentication schemes. The master password itself is both *Resilient-to-Throttled-Guessing* and *Resilient-to-Unthrottled-Guessing* because it uses a computationally intensive hash function and is required to be 16 characters long. By only allowing the user to use generated passwords which are designed to be extremely difficult to guess MassPass is provides more security than typical passwords, but poorly implemented security policy on the target website could still compromise a particular password. The hashing scheme also makes the passwords *Resilient-to-Leaks-from-Other-Verifiers* and *Unlinkable*. Finally, MassPass is *No-Trusted-Third-Party* which protects the user from a malicious or compromised third party. However, as with all other password schemes, MassPass is not *Resilient-to-Internal-Observation* as a keylogger would be just as effective.

Property	Password	Firefox	LastPass	MassPass
Physical Observation	✗	−	−	−
Targeted Impersonation	−	−	−	−
Throttled Guessing	✗	✗	−	−
Unthrottled Guessing	✗	✗	−	−
Internal Observation	✗	✗	✗	✗
Leaks from Other Verifiers	✗	✗	−	✓
Phishing	✗	✓	✓	✓
Theft	✓	✓	✓	✓
No Trusted Third Party	✓	✓	✗	✓
Requires Explicit Consent	✓	✓	✓	✓
Unlinkable	✓	✓	✓	✓

Conclusions and Future Work

MassPass solves a handful of the problems of other password schemes with a few drawbacks. The primary advantage of MassPass is in security, it generates extremely hard to guess passwords for websites and even if a password is leaked, it is infeasible to obtain the master password. The plugin and website are not as polished as they could be and for this reason to not give the best user experience, although both are reasonably convenient. It would be valuable to also have a native mobile version for paranoid or traveling users. The major flaw with MassPass is that it is not possible to change the password for one specific site. Although this is not something that a user would need to typically do, this may occur if a particular website is compromised and that password needs to be changed. In this case the user will need to come up with a new master password and change their passwords on all of the websites for which they use MassPass. It seems difficult to address this problem without compromising the store-nothing policy at the core of MassPass. Despite this, MassPass is a viable and useful alternative to existing password managers that users seeking additional security or easier portability may find valuable.