

# EpisoPass: Password Management based on Episodic Memories

Toshiyuki Masui

Faculty of Environment and Information Studies, Keio University  
5322 Endo, Fujisawa, Kanagawa 252-8520, Japan  
[masui@sfc.keio.ac.jp](mailto:masui@sfc.keio.ac.jp)  
<http://pitecan.com/>

**Abstract.** We propose a password manager *EpisoPass* that supports the generation of strong passwords based on a user's secret episodic memories. To use EpisoPass, a user first collects question-answer pairs related to their own episodic memories. Each is registered with several possible answers: a single correct answer and multiple fake answers. When the user wants to generate a password, EpisoPass shows each question and list of possible answers and asks the user to select those that are correct. EpisoPass then generates a domain-unique password by substituting the characters of a seed string based on the selected answers. Through careful selection of memories and answers, EpisoPass provides an authentication step using memories that are easy to recall, but difficult for others to guess. In this way various strong passwords can be easily managed without the need for the master password or secret device that is otherwise required by conventional password managers. Using a browser extension, users can use EpisoPass directly on the login page of conventional Web services like Facebook, removing the need to type or copy a password string.

**Keywords:** Passwords; password managers; user authentication; episodic memories; EpisoPass

## 1 Introduction

Passwords have been used as a means of authenticating to Web services and applications for a long time, and remain the most popular authentication method on the Internet. Since short passwords are easily guessable by attackers and using the same password for multiple services is unsafe, a different long password should be used for each service a person uses. However, remembering numerous long passwords is almost impossible for ordinary humans. According to research conducted by Florêncio *et al.* in 2007, people use on average only 6.5 different passwords to access 25 Web services, and 4.28% of users forget their passwords within 3 months [9].

Since passwords are difficult to handle, various other authentication methods have been proposed. For example, image-based authentication [3, 15], biometrics<sup>1</sup>, behavior-based authentication [6], and many others.

However, password-based authentication is still the most convenient and widely deployed method [4], and is not expected to disappear any time soon [13].

Given we will have to continue to live with password-based authentication systems for the foreseeable future, we have to devise practical methods for handling many passwords, and various “password managers” have been proposed [1, 8, 16, 18, 21, 24, 27]. Password managers remember users’ passwords and aim to directly enter them into the login pages of the various services a user wants to access. The burden on the user is reduced by requiring just a single “master password” to access the database of stored passwords. Although password managers are widely used and a clear step forward from having to remember multiple passwords, users nonetheless have to remember the master password or use a special hardware device for safely handling the password database, and password managers usually run on only a limited set of devices.

If we want to avoid having to carry any special device for authentication, either we must use some form of biometrics, or all the information required for the authentication must be stored in the user’s memory. However, the biggest problem with memory-based authentication is that users cannot reliably remember passwords or master password that are long and complex enough to be secure. For this reason, we believe that it is far better to “generate” something for the authentication, based on a user’s episodic memories. This has the benefit that unlike a password, a person is highly unlikely to forget such episodic memories. We therefore propose *EpisoPass*, a password manager that generates strong passwords based only on a user’s secret and unforgettable episodic memories.

## 2 EpisoPass

EpisoPass is a password manager that supports the generation of strong passwords based on a user’s secret episodic memories. Our brains store numerous memories, but different memories have different characteristics. Some memories are very short-lived, while others are recallable for long periods of time. When we have a particularly impressive experience, the memory of it will stay in our mind for a long time and can’t be easily forgotten. On the other hand, other topics are more difficult to remember. For example, when studying mathematics and trying to remember a new formula, it can be hard to memorize without significant practice, since knowledge of the formula is unrelated to any personal experiences. The former type of memory is called an *episodic* memory and most people find such memories easy to recollect and hard to forget. Memories of passwords belong in the latter category. People find them hard to remember and easy to forget, in the same way people find it difficult to remember mathematical formulas.

---

<sup>1</sup> <https://en.wikipedia.org/w/index.php?title=Biometrics&oldid=736189000>

**EpisoPass - JohnDoe@example.com** Save to file Save to server Android app

Seed: Amazon123456 ⇌ Password: Tblgeq808187

**Who was dirty and rude?**

Nishizaki	Kushida	Kusakabe	Shiota	Kouno	Mizuta	Senoo
Miura	Noguchi	Nishiyama	Kishino	Horii	Itao	Imada
Ebisawa	Yoneyama	Gouda	Haga	Nakazono	-	+

**Grandmas phone?**

0798	7799	1233	9876	2525	4553	3435
2301	3678	5838	6594	9008	3904	2381
2435	6253	3238	7473	-	+	

**Who beat me?**

Nakanishi	Noguchi	Murakami	Kakuzen	Teshima	Nishimizu	Yuuki
Iwata	Katsuya	Takada	Wada	Kawamura	-	+

**Who loved Kwansei Gakuin?**

Nishizaki	Kushida	Kusakabe	Shiota	Kouno	Mizuta	Senoo
Miura	Noguchi	Nishiyama	Kishino	Horii	Itao	Imada
Ebisawa	Yoneyama	Gouda	Haga	Nakazono	-	+

**Best trail in Rokko?**

Hokura-san	Twenty-Cross	Shijuu-hachitaki	Somadani	Totoyamichi	Tokugawamichi	Momijidani
Okuke	Sunrise	-	+			

**Where did I BBQ?**

Hama-Koushien	Sengari	Mukogawa	Amagatouge	Minoo	Kitayama-dam	Seppiko-san
Sakasegawa	Doujou	Kidugawa	Soni-kogen	Shukugawa	-	+


**Who taught trumpet?**

Yamamoto-Sensei	Moriya-Sensei	Mizukuchi-Sensei	Hatano-Sensei	Tainoshou-Sensei	Uchikoshi-Sensei	Murakami-Sensei
Suzuki-Sensei	Itoh-Sensei	Fujiwara-Sensei	Wakamatsu-Sensei	Shimizu-Sensei	-	+

**Where did I hurt my leg?**

Takagicho	Mondo-nishimachi	Mondosou	Shimo-oiichi	Kannouchi	Aza-nakatani	Wakamatsucho
Okadayama	Kami-koutouen	Danjo	Uegahara	Shin-kouyou	Koutouen	Kami-oiichi
					-	+

<http://gyazo.com/ac515f5ce991b516e785122dd9192dd2.png>



Yokota	Yoshizumi	Nagata	Senoo	Takeda	Tamura	Matsumaru
Yoroi	Fujisaki	Miura	Toukai	Onoda	Saijo	Yamagata
					-	+

**When do you use this?**

April 2016	May 2016	June 2016	July 2016	August 2016	September 2016	October 2016
November 2016	December 2016	January 2017	February 2017	March 2017	-	+
					-	+

Fig. 1. Generating an Amazon password with EpisoPass.

The idea of using episodic memories for authentication has a long history and early papers suggested using episodic memories for creating passwords [2]. Authentication using secret questions is sometimes referred to as “cognitive passwords” [28], and such approaches have been used as an alternative to password-based authentication systems. While cognitive passwords were aimed to be alternatives to password-based systems, EpisoPass is a system for supporting password-based systems using episodic memories.

Password generation on EpisoPass is performed through the following steps:

1. A user registers multiple question texts related to their own personal secret unforgettable episodic memories. For each question they must provide a single correct answer and multiple additional incorrect answers.
2. The user provides a long “seed string” for each service that requires a password.
3. EpisoPass shows the questions and answers to the user allowing them to select the correct answer for each question. Based on the user’s selections, EpisoPass substitutes characters in the seed string and generates a strong password candidate string. After selecting all of the correct answers, the user copies the calculated string and registers it as the password for the service.

## 2.1 Using EpisoPass in a browser

Figure 1 shows how to generate a password on EpisoPass running in a browser by accessing the EpisoPass site. Many questions related to the user’s episodic memories are shown to the user, and many candidate answers are also shown for each question. When a user clicks and selects one of the answers for a question, the seed string shown at the top-left is converted to a candidate password string based on the selections. If incorrect answers are chosen the wrong password will be generated. On the other hand, when the user selects the correct answers for all of the questions, the correct password is calculated and shown at the top of the page.

In Figure 1, “Amazon123456” is provided as the seed string, and based on the selections to the ten questions, the seed string is converted to the string “Tb1geq808187”, to be used as the password for Amazon.com.

When the user clicks on different candidate answers the seed string is converted to a completely different string, such as “Xvdkzb940345”, as shown in Figure 2.

In this way, different selections yield different password strings and the unique password string generated after selecting the correct answers should be used as the password for the service.

Capital letters in the seed string are substituted for capital letters in the password, and digits in the seed string are substituted for digits. In this way the generated password can be arranged to conform with any password character restrictions the site might impose.

The first question in Figure 1 is based on the author’s episodic memory at elementary school, and the question with a photo at the bottom is related

to a more recent event which the author believes he will never forget. All the questions are based on the author’s episodic memories that he’s unlikely to ever forget, and that he believes nobody else will know the correct answer for.

The screenshot shows the EpisoPass interface for a user named JohnDoe@example.com. It includes buttons for 'Save to file', 'Save to server', and 'Android app'. Below these are input fields for a 'Seed' (Amazon123456) and a 'Password' (Xvdkzb940345). The main content area displays three sets of questions and answers, each with a table of options and a selected answer highlighted in blue.

**Who was dirty and rude?**

Nishizaki	Kushida	Kusakabe	Shiota	Kouno	Mizuta	Senoo
Miura	Noguchi	Nishiyama	Kishino	Horii	Itao	Imada
Ebisawa	Yoneyama	Gouda	Haga	Nakazono	-	+

**Grandmas phone?**

0798	7799	1233	9876	2525	4553	3435
2301	3678	5838	6594	9008	3904	2381
2435	6253	3238	7473	-	+	

**Who beat me?**

Nakanishi	Noguchi	Murakami	Kakuzen	Teshima	Nishimizu	Yuuki
Iwata	Katsuya	Takada	Wada	Kawamura	-	+

**Fig. 2.** Selecting a different set of answers.

Usability has been considered as a key consideration, and the questions and answers in the figures can be edited directly in the browser. They can be saved on the server by clicking the “save to server” button or saved locally if the user prefers. While the Q-A data sets are saved, no information about the correct answer or the generated password is saved on the server. Consequently there is no requirement for the user to trust the server to keep the data secret. The Q-A data can be downloaded to the user’s local machine in JSON format by clicking the “save to file” button. To upload the data back to the server, the user simply drags the JSON file to the EpisoPass page.

When the secret string is changed to “Facebook123456”, the generated password changes to “Onjbrppy030937”, as shown in Figure 3. In this way, the user can generate different passwords for different services just by changing the seed string. Seed strings like “Amazon123456” and “Facebook123456” are used in the examples because users can easily remember for which service the set of seeds and Q-A pairs are used. Users don’t have to remember the seed strings, just like users don’t have to remember the Q-A pairs.

Character substitution is performed based on the value calculated by hashing a concatenation of all the questions and selected answers. For example, if the user selects “Ebisawa” as the answer to the question “Who’s the bully?”, a string including “Who’s the bully:Ebisawa” is used for calculating the hash value.

The last question in Figure 1 is not intended as a secret question based on an episodic memory, but is instead a question for generating different passwords

**EpisoPass - JohnDoe@example.com** Save to file Save to server Android app

Seed: Facebook123456 ↔ Password: Onjbrppy030937

**Who was dirty and rude?**

Nishizaki	Kushida	Kusakabe	Shiota	Kouno	Mizuta	Senoo
Miura	Noguchi	Nishiyama	Kishino	Horii	Itao	Imada
Ebisawa	Yoneyama	Gouda	Haga	Nakazono	-	+

**Grandmas phone?**

0798	7799	1233	9876	2525	4553	3435
2301	3678	5838	6594	9008	3904	2381
2435	6253	3238	7473	-	+	

**Who beat me?**

Nakanishi	Noguchi	Murakami	Kakuzen	Teshima	Nishimizu	Yuuki
Iwata	Katsuya	Takada	Wada	Kawamura	-	+

**Fig. 3.** Generating a new password for Facebook.

for different situations. By providing a question like this, the user can generate completely different passwords for different months and years, say.

## 2.2 Android application

If a user prefers not to use the EpisPass service on the Web, there is an alternative EpisPass application for Android which requires no network connection. After registering questions and answers on the EpisPass service, the user can download an Android application from the server by clicking the “Android app” button at the top of the page. The application unique to the user is compiled and built on the server and contains all of the information needed for the particular user to generate passwords using the Q-A sets entered.

When the user runs the application and selects the correct answer to each question, the password will be generated in the same way as on the site, as shown in Figures 4 and 5. This can then be copied to the password entry for access to a particular service.

The calculation is performed on Android without the need for a network connection, so the user can safely generate passwords without being concerned about an attacker observing any data transfer.

## 2.3 Using the browser extension

One of the difficulties of the previous two approaches is that the user must re-type or copy the password string into the service’s password field after it’s been generated. This isn’t ideally convenient, so we have also developed a browser extension that allows EpisPass to be used directly with the login page of services on the Web.

A browser extension is a JavaScript add-on program which runs on top of existing Web pages. Figure 6 shows the login page of Amazon.com, with the

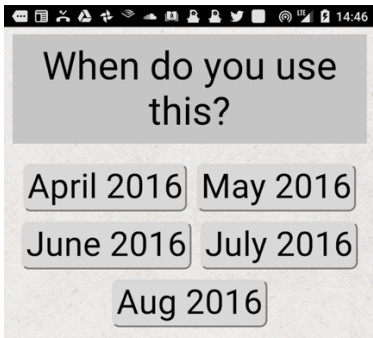


Fig. 4. Running EpisoPass on Android.

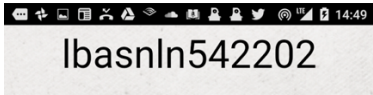


Fig. 5. After finishing selections.

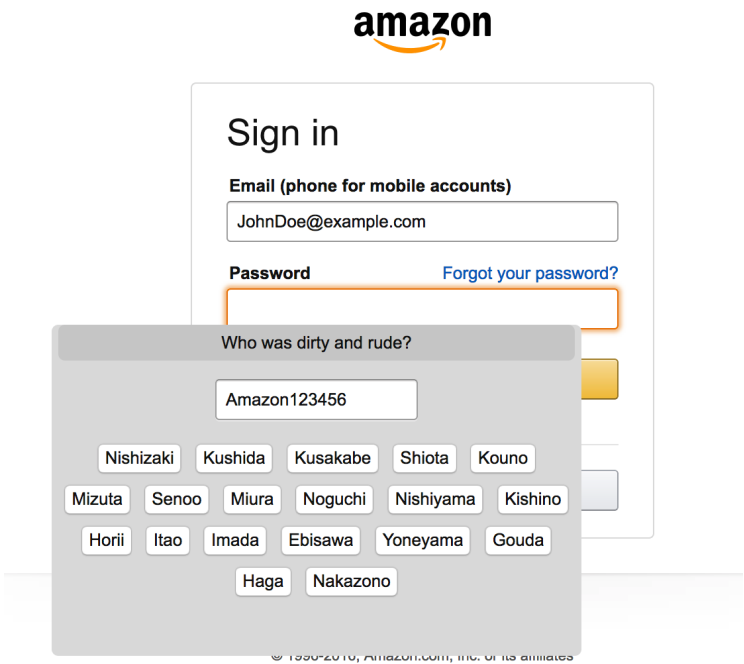


Fig. 6. Using EpisoPass on the login window.

EpisoPass questions automatically overlaid directly onto the page by the browser extension. When the user accesses the Amazon login page, the browser extension automatically runs on the page, acquires the questions and answers from the EpisoPass server, and displays them using the same style as for the Android application. After answering all of the questions, a password is calculated from the selected answers and automatically pasted into the password field of the page by the extension. In this way users can log in to various services just by answering the questions presented, and without even seeing the password string. It appears as if the EpisoPass-based authentication is being provided directly by the services themselves.

### 3 Discussion

In this section, we discuss the advantages and potential disadvantages of using EpisoPass.

#### 3.1 Recallability of answers

The biggest advantage of using EpisoPass is that users don't have to remember strong password strings. The seed string and question-answer data is essentially public, and users of EpisoPass can save them wherever is convenient. They can then easily generate passwords by running EpisoPass and answering the questions. If a question is based on old unforgettable episodic memory, there is little chance of losing the password for a service as long as the seed string and questions are available. If the user's memory is related to an episode that took place 20 years ago, say, and the user clearly still remembers the episode now, it is unlikely the episode will be forgotten in the future.

#### 3.2 Selection of seed strings

In the previous examples showing the author's EpisoPass questions, a seed string "Amazon123456" was used for Amazon.com, and "Facebook123456" was used for Facebook. Such strings were chosen because they are easily memorable. In practice any seed string could be used as the seeds for the various services, and automatic generation of seed strings is a possibility, although generating service-specific unique seed string is not trivial. For example, users can use the same username and password both for Amazon.com and Amazon.co.uk, but not for Amazon.co.jp.

#### 3.3 Security strength of EpisoPass

The strength of the generated passwords depends on the number and level of secrecy of the questions. There have been many studies on the strength of passwords [12, 17], but measuring the strength of secret questions is an area where more study is needed. Using ten secret questions each with twenty answers, where



each answer is considered equally likely, an attacker would need to cover a search space of over ten trillion ( $20^{10}$ ) values to check every potential combination of answers, and the entropy of the system is 43.2 ( $10 \times \log_2 20$ ) bits. This represents roughly the same entropy as using a random 8-character Latin-alphabet password, for which the entropy is 45.6 bits. This level is considered to be strong enough for Web services, where it's not possible to perform online brute-force attacks [10].

### 3.4 Selecting good questions

The quality of the questions is key to using EpisoPass effectively. If the episode is shared by someone else, this other person might easily be able to answer the question and generate the user's password. The episode related to each question should therefore not be known to any other person, and the episode should be unforgettable. Finding such episodes seems difficult at first, but our experience has been that we try to remember such experiences from our personal past, we can soon recall many trivial episodes which are unforgettable but nonetheless not important to other people. The following list shows some examples of experiences that we believe make good candidates for EpisoPass questions.

- Memories of very minor injuries that nobody would have cared about apart from you.
- Memories of bad experience such as embarrassing blunders or failures, especially in the case where it was never admitted to anyone else.
- Experiences of finding inconsequential but special items which only you are interested in.

For example, a question such as “who hit you when you were six years old?” is about a trivial experience that most people would never have reason to discuss, but an experience that may have been unpleasant at the time is hard to forget.

Questions such as “Which food do you like best?” should be avoided, since friends or family might know the user's tastes, thereby allowing others to select the correct answer. Questions related to an episode which the user is proud of should also be avoided, since this is a something the user might prefer to discuss with others.

We don't usually talk about trivial bad experiences with other people, but we might boast about good experiences (some might even write blog posts about them). Similarly our tastes (*e.g.* favorite foods) might change in the long run. Using such episodes for questions should therefore be avoided.

### 3.5 Creating fake answers

It can be difficult to provide a large number of false answers to a question like “what is your favorite sport?” because the number of possible answers is limited. On the other hand, if the right answer to the question is a name of a place or a person, generating similar answers is straightforward. For example,

if the answer to the question is “Colorado”, we can easily provide fake answers like “California”, “Utah”, *etc.* because we can use the list of states in the U.S. as possible answers.

In this way, false answers can be easily generated if it is possible to collect words which belong to the same category as the right answer. Various methods have been proposed for collecting words in the same category, mainly for information retrieval tasks [14, 25, 26]. We can use such systems to provide a list of potential false answers almost automatically.

### 3.6 Universality

Although everyone has to make use of authentication systems, not everyone is good at handling passwords. Even experienced computer users have trouble with passwords, since choosing a good password is unintuitive and remembering complex passwords is hard. Using EpisoPass, people can use password-based systems without having to invent techniques for creating and remembering strong passwords. The Q-A configuration step is a one-time process, after which the system becomes straightforward to use. By integrating EpisoPass into existing password-based services, people can even use services without noticing that passwords are required for the service. Our experience of using browser extensions suggests that this approach offers the most seamless experience.

### 3.7 Password requiring frequent updates

Many services still require users to change passwords periodically in an attempt to strengthen security. Although this is no longer considered good-practice, since humans struggle to generate a stream of strong passwords [22], the practice is still widespread. Using EpisoPass, users can provide a date-related question similar to that shown as the final question in Figure 1. This allows different strong passwords to be generated depending on the answer chosen for this question. Using this technique, people can easily manage both old and new passwords efficiently.

### 3.8 Care for handling secret information

Users don't have to be very careful about handling questions and answers. They can even be stored in a public place given a sufficient number of questions and fake answers are provided. Keeping secrets is a considerable effort for most people (including the authors), but if the entire set of questions and answers used for EpisoPass can be stored publicly, handling it becomes straightforward. This is in contrast to the care needed when handling secrets like passwords, secret keys for SSH, *etc.* which should never be copied to or saved in an unsafe place. The EpisoPass data can be stored as a plain text file wherever is convenient for the user, since a malicious attacker isn't able to generate a valid password without also knowing the owner's relevant episodic memories.

### 3.9 Risk of server-side password leaks

If one of the passwords generated by EpisoPass were to be revealed for some reason, there is a danger that other passwords based on the same questions might also be revealed. For example, if Twitter is attacked by a hacker and the password for Twitter (“Lbasnln542202” in Figure 1) is revealed to the attacker, the attacker could test all answer combinations. If the attacker also had access to the question and answer strings used, they could then establish the correct answers used to generate the password. Once all the answers to the questions are known, the attacker can then freely generate all of the user’s passwords generated from the same set of questions.

To prevent this, users can prefer saving questions and answers in a secret place or use sufficient questions to prevent such a brute-force attack from being viable. The entropy of 12 questions with 30 candidate answers is  $58.9 (12 \times \log_2 30)$  bits, and this is considered to be safe enough at this moment.

### 3.10 Using images

Pictures can also be used as EpisoPass questions, similar to that shown at the bottom of Figure 1. Even if people find it difficult to create good secret questions, selecting an image from their photo collections and using it as the question should be straightforward. For example, if you have an old special picture of a place you visited, you can use it as the question and use the name of the place as well as other similar fake place names. For a secret question, a low-quality photo is better than a high-quality photo because a low-quality photo like the one shown in Figure 1 are not easily recognizable by other people.

Of course, it’s important the photo should have no information related to the location, especially given how sophisticated Web-based image searching has become.

### 3.11 Password entry time

Answering to many questions takes much longer time than typing a remembered password, so people may prefer typing password rather than using EpisoPass if they have to log in to a service frequently. The tradeoff between security and usability has long been a big issue of authentication systems [5], but we believe we can approach the best solution by improving the user interface of answering questions.

## 4 Related Work

As discussed in previous sections, there have been many attempts to replace password-based authentication systems, but none of them have yet become as widely deployed as passwords [4]. Although cognitive password systems have been an area of study for many years [19, 28], implementations up to now have

all been intended as replacements to password-based systems, requiring support from the services involved. As far as the authors are aware, ours is the first password-generation systems to be based on cognitive authentication.

The idea of using episodic memories for authentication is also not new. In fact, using episodic memories for selecting passwords has generally been recommended, and many current computer users are probably using password that are in some way informally related to their episodic memories.

Recent work has considered the use of mobile devices for authentication, in particular harnessing the fact that a mobile device captures large quantities of personal information. A mobile device may therefore be able to identify its user based on their previous behavior; events which only the user would be able to recall [6, 7, 11]. For example, if a user can answer questions such as “who did you call last night?” correctly, the authentication will succeed. Using mobile or wearable devices for authentication will become more usable in the future, but in order for this to work effectively users must be able to remember potentially arbitrary behavior over the long-term. This may be a challenge for many people, and the benefit of EpisoPass in comparison is that the user is able to select memories that they know to be memorable, rather than having their mobile device choose them arbitrarily.

Various types of image-based authentication systems have been proposed recently [3, 15], based on the realisation that images are easier to remember than text given they are usually more directly linked to episodic memories. However, some image-based authentication systems request users to remember new secrets related to the images instead of using the users' old episodic memories. This turns out to be not much easier to remember than passwords.

Image-based authentication based on episodic memories might work if users can prepare many images that are tightly linked to their episodic memories. However, finding such images is usually not straightforward, and image-only authentication systems are unlikely to become popular until simple and effective techniques for doing so have been developed.

Even when a new ideal authentication method is invented, replacing all the password-based systems will still be a lengthy process. Password managers will therefore remain important until this hypothetical ideal solution has become ubiquitous. In the age of password-based authentication, using password managers seems to be the only way to tackle the problem of passwords. While most of the password managers only remember passwords given by the user, generating passwords with a password manager is a new approach for handling password-based systems. Just like EpisoPass generates passwords, Versipass [23] helps the user to generate password strings using “visual cues” from an arbitrary image. Instead of directly using images for authentication, users use the system to generate a password string with the help of the image shown to the user.

## 5 Experiences

EpisoPass has been used by the authors for more than three years for various services including Twitter, Facebook, Amazon, Skype, *etc.* Before using EpisoPass, managing multiple passwords was a significant challenge for the authors. We now have all the information for generating passwords stored in the cloud and no problems have arisen during this period. Since the introduction of our browser extension, visiting the EpisoPass service has now become unnecessary, making authentication to the various services even easier.

The EpisoPass service is available via the Website and the source code is available on GitHub<sup>2</sup>. EpisoPass currently has only a small user-base, one major reason being that most people cannot fully understand the idea behind EpisoPass, and may not trust it without the support of a well-known IT company. Another reason is that it's difficult for users to assess the security of EpisoPass. Since the intention is for the questions and answers to be obvious to those that know them (but not to others), users may lack confidence in their answers being unknown to others. Any authentication system expecting widespread use must take such psychological issues seriously, and we hope to address such issues in future work.

## 6 Conclusion

We introduced a password management system *EpisoPass* that converts a seed string into a password using the user's episodic memories. These memories are represented as a set of questions and answers which can be solved only by the user in order to generate a site-specific password. Using EpisoPass with well-defined questions and answers, a user can always retrieve a service's passwords without worrying about having to remember any secret information, other than the episodic memories that have been chosen specifically to be easy to recall. In future work we hope to integrate the system with more existing password-based services, and ultimately aim to address the problems derived from password-based authentication.

**Acknowledgments.** This paper was written while on sabbatical at the University of Cambridge, hosted by Frank Stajano and the Pico group, and supported by EPSRC grant number EPSRC IRIS grant EP/M019055/1 on "Future Authentication Systems". We would like to thank Frank Stajano and David Llewellyn-Jones for giving many valuable advices to our work.

**Note.** This paper is largely a translation of a 2013 paper published by the author in Japanese [20]. This is the first time that the work is presented to an international audience in English. Additionally, in the intervening three years the author has developed the browser extension described in section 2.3.

<sup>2</sup> <https://EpisoPass.com/>, <https://github.com/masui/EpisoPass>

## References

1. AgileBits Inc.: 1password. <https://agilebits.com/onepassword>
2. Barton, B.F., Barton, M.S.: User-friendly password methods for computer-mediated information systems. In: *Computer & Security*. vol. 3, pp. 186–195 (1984)
3. Biddle, R., Chiasson, S., Van Oorschot, P.: Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv.* 44(4), 19:1–19:41 (Sep 2012), <http://doi.acm.org/10.1145/2333112.2333114>
4. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. pp. 553–567 (2012)
5. Braz, C., Seffah, A., M'Raihi, D.: Designing a trade-off between usability and security: A metrics based-model. In: *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*. pp. 114–126. *INTERACT'07*, Springer-Verlag, Berlin, Heidelberg (2007), <http://dl.acm.org/citation.cfm?id=1778331.1778344>
6. Dandapat, S.K., Pradhan, S., Mitra, B., Roy Choudhury, R., Ganguly, N.: Activpass: Your daily activity is your password. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. pp. 2325–2334. *CHI '15* (2015), <http://doi.acm.org/10.1145/2702123.2702457>
7. Das, S., Hayashi, E., Hong, J.I.: Exploring capturable everyday memory for autobiographical authentication. In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. pp. 211–220. *UbiComp '13*, ACM, New York, NY, USA (2013), <http://doi.acm.org/10.1145/2493432.2493453>
8. Dashlane, Inc: Dashlane. <https://www.dashlane.com/>
9. Florêncio, D., Herley, C.: A large-scale study of web password habits. In: *Proceedings of the 16th international conference on World Wide Web*. pp. 657–666. *WWW '07* (2007), <http://doi.acm.org/10.1145/1242572.1242661>
10. Florêncio, D., Herley, C., Coskun, B.: Do strong web passwords accomplish anything? In: *Proceedings of the 2nd USENIX workshop on Hot topics in security*. pp. 10:1–10:6. *HOTSEC'07* (2007), <http://dl.acm.org/citation.cfm?id=1361419.1361429>
11. Gupta, P., Wee, T.K., Ramasubbu, N., Lo, D., Gao, D., Balan, R.K.: Human: Creating memorable fingerprints of mobile users. In: *PerCom Workshops*. pp. 479–482. *IEEE Computer Society* (2012), <http://dblp.uni-trier.de/db/conf/percom/percomw2012.html#GuptaWRLGB12>
12. Hayashi, E., Hong, J.: A diary study of password usage in daily life. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp. 2627–2630. *CHI '11* (2011), <http://doi.acm.org/10.1145/1978942.1979326>
13. Herley, C., Oorschot, P.C., Patrick, A.S.: Passwords: If we're so smart, why are we still using them? In: *Dingledine, R., Golle, P. (eds.) Financial Cryptography and Data Security*, pp. 230–237. Springer-Verlag (2009), [http://dx.doi.org/10.1007/978-3-642-03549-4\\_14](http://dx.doi.org/10.1007/978-3-642-03549-4_14)
14. Huang, X., Wan, X., Xiao, J.: Learning to find comparable entities on the web. In: *Proceedings of the 13th international conference on Web Information Systems Engineering*. pp. 16–29. *WISE'12*, Springer-Verlag (2012), [http://dx.doi.org/10.1007/978-3-642-35063-4\\_2](http://dx.doi.org/10.1007/978-3-642-35063-4_2)

15. Internet Safety Project: Graphical passwords. <http://www.internetsafetyproject.org/wiki/graphical-passwords>
16. KING JIM: Password manager “MilPass” PW10. <http://www.kingjim.co.jp/sp/pw10/>
17. Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F., Egelman, S.: Of passwords and people: measuring the effect of password-composition policies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2595–2604. CHI '11 (2011), <http://doi.acm.org/10.1145/1978942.1979321>
18. LastPass.com: Lastpass. <https://lastpass.com/>
19. Lazar, L., Tikolsky, O., Glezer, C., Zviran, M.: Personalized cognitive passwords: an exploratory assessment. In: Information Management & Computer Security. vol. 19, pp. 25–41 (2011)
20. Masui, T.: EpisoPass: Password management based on episodic memories. In: Proceedings of the 21st Workshop on Interactive Systems and Software (WISS2013). pp. 109–114. Japan Society for Software Science and Technology (2013), in Japanese
21. Reichl, D.: Keypass password safe. <http://keepass.info/>
22. Schneier, B.: Changing passwords. [https://www.schneier.com/blog/archives/2010/11/changing\\_passwo.html](https://www.schneier.com/blog/archives/2010/11/changing_passwo.html) (2012)
23. Stobert, E., Biddle, R.: A password manager that doesn’t remember passwords. In: Proceedings of the 2014 Workshop on New Security Paradigms Workshop. pp. 39–52. NSPW '14 (2014), <http://doi.acm.org/10.1145/2683467.2683471>
24. Symantec Corporation: Norton id safe. <http://jp.norton.com/portal-IDsafe/>
25. Wang, R.C.: Boo!wa! <http://boowa.com/>
26. Wang, R.C., Cohen, W.W.: Language-independent set expansion of named entities using the web. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining. pp. 342–350. ICDM '07 (2007), <http://dx.doi.org/10.1109/ICDM.2007.104>
27. WoodenSoldier: Id manager. <http://www.woodensoldier.info/soft/idm.htm>
28. Zviran, M., Haga, W.J.: User authentication by cognitive passwords: An empirical assessment. In: Proceedings of the Fifth Jerusalem Conference on Information Technology. pp. 137–144. JCIT, IEEE Computer Society Press, Los Alamitos, CA, USA (1990), <http://dl.acm.org/citation.cfm?id=100512.100538>