EpisoPass: Password Management based on Episodic Memories

1st Author Name
Affiliation
Address
e-mail address
Optional phone number

2nd Author Name
Affiliation
Address
e-mail address
Optional phone number

3rd Author Name
Affiliation
Address
e-mail address
Optional phone number

ABSTRACT

We propose a password manager *EpisoPass* that supports generating strong passwords based on the user's secret and unforgettable episodic memories. To use EpisoPass, a user first collects question-answer pairs related to his episodic memory and registers them with fake answers. When the user wants to generate a password, EpisoPass shows the questions and lists of possible answers and asks the user to select correct ones. Then EpisoPass generates a password by substituting characters of the seed string based on the selected answers. If the user is the only person who knows the correct answer, the password cannot be guessable by other people, and various strong passwords can be easily managed without using master passwords or secret devices that are usually required on conventional password managers.

Author Keywords

Passwords; password managers; episodic memories; EpisoPass;

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces; K.6.5. Management of Computing and Information Systems: Security and Protection

INTRODUCTION

Passwords have been used for various Web services and applications for a long time, and currently the most popular authentication method on the Internet. Since short passwords are easily guessed by attackers and using the same password for different services is unsafe, different long passwords should be used for different services. However, remembering many long passwords is almost impossible for ordinary humans. According to a research in 2007, people use 6.5 passwords on 25 Web services on average, and 4.28% of the users forget their passwords in 3 months[6].

Since passwords are difficult to handle, various other authentication methods have been proposed. For example, image-based authentication[2][11], biometrics authentication[1], behavior-based authentication[4], and many other authentication methods have been proposed.

However, password-based authentication is still the most convenient and strong method[3], and it is not supposed to get extinct in a short period of time[9].

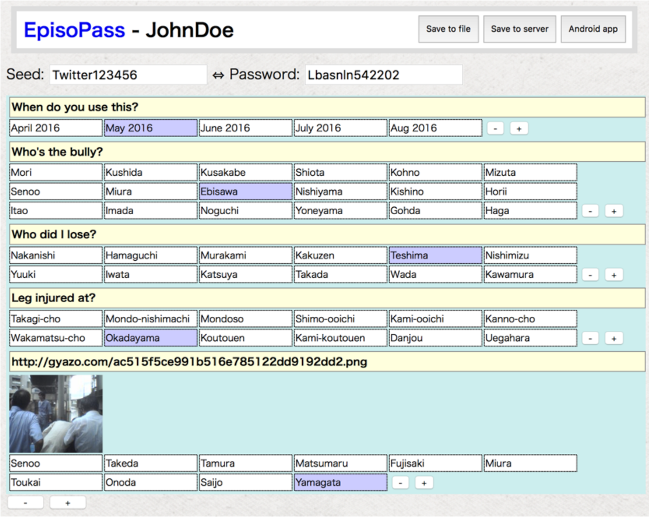
If we have to live with password-based authentication systems, we have to devise some ways to handle many passwords, and various "password managers" have been proposed [1][5][13][15][17][20][23]. Password managers remember users' passwords and help them enter passwords on various services. Most password managers can manage various passwords by asking users to remember a single "master password" to access the database. Although password managers are useful, users have to remember the master password or use a special hardware device for safely handle password managers, and password managers usually run on limited devices.

If we don't want to carry any special device for authentication, all the information required for the authentication should be kept in the brain. However, the biggest problem of brain-based authentication is that users cannot safely keep memories like long passwords or master password. For this reason, we believe that it is far better to "generate" something for the authentication, based on users' episodic memories which they can never forget. We propose a password manager *EpisoPass* that generates strong passwords based only on the user's secret episodic memories that the user can never forget.

EPISOPASS

EpisoPass is a password manager that supports generating strong passwords based on users' secret episodic memories. Memories in human brain are not uniform. Some memories are very short-lived, and others are unforgettable. When we have a very impressive experience, that memory will stay in the brain for a long time and cannot easily disappear. On the other hand, when we study math and try to remember a formula, it is usually hard to memorize it unless we practice a lot, since the knowledge about the formula is not related to any personal experiences. The former type of memory is called the episodic memory and people cannot easily lose it. Memories of passwords belong to the latter type and people cannot easily remember them, just like people cannot remember math formulas easily.

[1] `https://en.wikipedia.org/wiki/Biometrics`



EpisoPass - JohnDoe

Seed: Twitter123456 ⇔ Password: Lbasnln542202

When do you use this?

April 2016 | May 2016 | June 2016 | July 2016 | Aug 2016

Who's the bully?

Mori | Kushida | Kusakabe | Shiota | Kohno | Mizuta
Senoo | Miura | Ebisawa | Nishiyama | Kishino | Horii
Itao | Imada | Noguchi | Yoneyama | Gohda | Haga

Who did I lose?

Nakanishi | Hamaguchi | Murakami | Kakuzen | Teshima | Nishimizu
Yuuki | Iwata | Katsuya | Takada | Wada | Kawamura

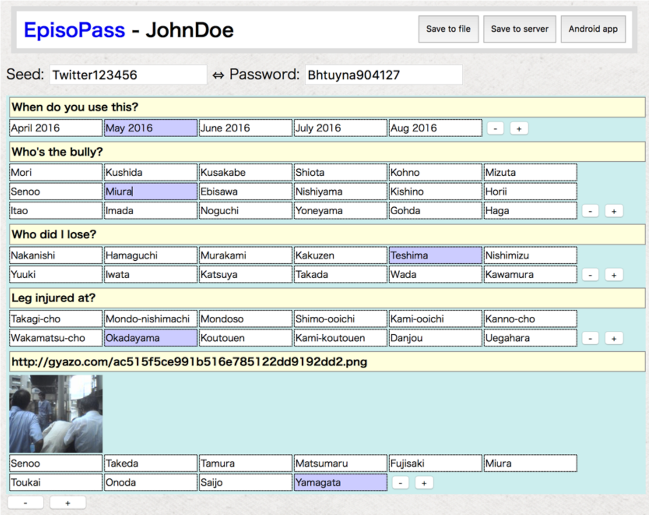
Leg injured at?

Takagi-cho | Mondo-nishimachi | Mondoso | Shimo-ooichi | Kami-ooichi | Kanno-cho
Wakamatsu-cho | Okadayama | Koutouen | Kami-koutouen | Danjou | Uegahara

http://gyazo.com/ac515f5ce991b516e785122dd9192dd2.png

Senoo | Takeda | Tamura | Matsumaru | Fujisaki | Miura
Toukai | Onoda | Saijo | Yamagata

Figure 1. Generating a Twitter password with EpisoPass.



EpisoPass - JohnDoe

Seed: Twitter123456 ⇔ Password: Bhtuyna904127

When do you use this?

April 2016 | May 2016 | June 2016 | July 2016 | Aug 2016

Who's the bully?

Mori | Kushida | Kusakabe | Shiota | Kohno | Mizuta
Senoo | Miura | Ebisawa | Nishiyama | Kishino | Horii
Itao | Imada | Noguchi | Yoneyama | Gohda | Haga

Who did I lose?

Nakanishi | Hamaguchi | Murakami | Kakuzen | Teshima | Nishimizu
Yuuki | Iwata | Katsuya | Takada | Wada | Kawamura

Leg injured at?

Takagi-cho | Mondo-nishimachi | Mondoso | Shimo-ooichi | Kami-ooichi | Kanno-cho
Wakamatsu-cho | Okadayama | Koutouen | Kami-koutouen | Danjou | Uegahara

http://gyazo.com/ac515f5ce991b516e785122dd9192dd2.png

Senoo | Takeda | Tamura | Matsumaru | Fujisaki | Miura
Toukai | Onoda | Saijo | Yamagata

Figure 2. Selecting a different answer.

Password generation on EpisoPass is performed through the following steps:

1. A user registers many question texts related to the user's personal secret episodic memories that the user never forgets, and provides a correct answer and additional fake answers.

2. The user provides a long "seed string" for each service that requires a password.

3. EpisoPass shows the data to the user so that the user can select the right answer for each question. Based on the user's selections, EpisoPass substitutes characters in the seed string and generates random-looking password candidate strings. After selecting all the right answers, the user copies the calculated string and register it as the password for the service.

Using EpisoPass on a browser

Figure 1 shows how to generate a password on EpisoPass running on a browser. 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 each question, the seed string shown at the top-left is converted to a random-looking string based on the selections. When the user selects the right answers to all the questions, the right password is calculated and shown at the top.

In Figure 1, "Twitter123456" is provided as the seed string, and according to the selections to the five questions, the seed string is converted to a string "Lbasnln542202", which can be used as the password for Twitter.

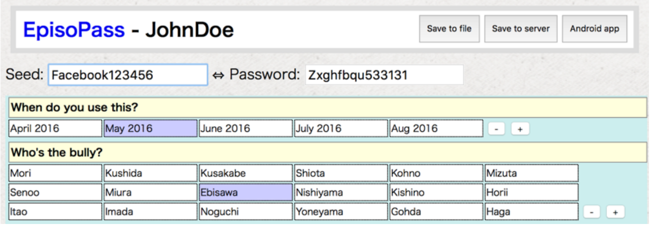
When the user clicks another candidate, the seed string is converted to another strings like "Bhtuyna904127" (Figure 2). In this way, different selections yields different password strings and the password string generated after selecting correct answers can be used as the password for the service.

Capital letters in the seed string are substituted to capital letters in the password, and digits in the seed string are substituted to digits, so that generated passwords conforms to password character restrictions sometimes requested by the service.

The second question in Figure 1 is based on the author's episodic memory at elementary school, and the last question is related to a more recent event which the author thinks he can never forget. All the questions are related to the author's episodic memories that he thinks he never forgets, and nobody else knows which one of the candidate answers is the right one.

Questions and answers can be edited directly on the browser, and they can be saved on the server by clicking the "save to server" button. The Q-A data sets are saved on the server, but no information about the correct answer or the generated password is saved on the server.

When we change the secret string to "Facebook123456", the generated password changes to "Zxghfbqu533131", as shown in Figure 3. In this way, we can generate different passwords for different services just by changing the seed string.



EpisoPass - JohnDoe

Seed: Facebook123456 ⇔ Password: Zxghfbqu533131

When do you use this?

April 2016 | May 2016 | June 2016 | July 2016 | Aug 2016

Who's the bully?

Mori | Kushida | Kusakabe | Shiota | Kohno | Mizuta
Senoo | Miura | Ebisawa | Nishiyama | Kishino | Horii
Itao | Imada | Noguchi | Yoneyama | Gohda | Haga

Figure 3. Generating the password for FaceBook.

Character substitution is performed based on the hash value calculated from concatinating all the selected answers. For example, if the user selects “**`Ebisawa`**” to the question “**`Who's the bully?`**”, a hash value calculated from “**`Who's the bully:Ebisawa`**” is used for the substitution. Details of the algorithm is shown in the Appendix.

Android application

If a user doesn’t like to use EpisoPass service on the Net, he can use an EpisoPass application on Android which does not require network connection. After registering questions and answers on the EpisoPass service, the user can download an Android application from the server by clicking the “Android app” button. The application contains all the information required for generating passwords. (The application is compiled and built on the server.)

When the user runs the application and selects the right answer to each question, he can eventually get the password and copy it to the password entry. (Figure 4,5)

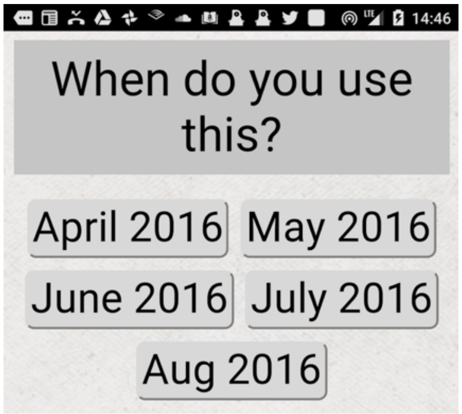

When do you use this?

Figure 4. Running EpisoPass on Android.

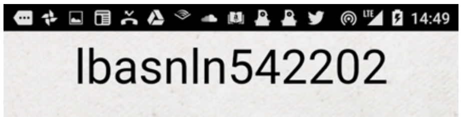

Ibasnln542202

Figure 5. After finishing selections.

DISCUSSIONS

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

Unforgettability

The biggest advantage of using EpisoPass is that users don’t have to remember passwords of any kind and they can stop worrying about password-related troubles. Users of EpisoPass can save the seed string and question-answer data at any place, and easily generate passwords by running EpisoPass and answering questions. If the question is based on old unforgettable episodic memories, there is little chance of losing passwords. If the user’s memory is related to an episode of 20 years ago and the user clearly remembers the episode now, it is unlikely that he forgets the episode 10 years later.

Security strength of EpisoPass

The strength of the generated password depends on the number of questions and the secret level of the questions. There have been many studies on the strength of passwords [8][14], but the strength of secret questions has not been studied enough. When we use 10 secret questions with 20 answers and there’s no clue for the answer, 1 billion (20^{10}) trials should be performed to check all the combinations of the answers, and the entropy of the system is 43.2 ($10 \times \log_2 20$) bits. It is almost the same entropy as using random 8-character English alphabets as a password, in which case the entropy is 45.6 bits. This level is considered to be strong enough for Web services, where online brute-force attack is impossible[7].

Selecting good questions

The quality of the questions is the key to using EpisoPass. If the episode is shared by someone else, that person can easily answer the question and generate the password just like the user. The episode related to each question should not be known to other people, and the episode should be unforgettable. Finding such episodes seems difficult at first, but when we try to remember experiences of old days, we can recall many trivial episodes which are unforgettable but not important to other people. Old experiences like the following are candidates for good questions used in EpisoPass.

- Memory of small injury. (Nobody cares it but you.)
- Memories of bad experience like blunders or defeat. (You don’t tell it to anybody.)
- Experience of finding a small special item which only you are interested in.

For example, a question like “who hit you when you were 6 years old?” is about a trivial experience that people do not mention, but a bad experience like this is not forgettable.

Questions like “Which food do you like best?” should be avoided, since some of the friends might know the user’s taste. Questions related to an episode which the user is proud of should also be avoided, since the user might talk about the episode to somebody else.

We usually don’t tell our bad trivial experiences to other people, but we might boast of good experiences or even write a blog about it. Also, our tastes (e.g. favorite food) might change in the long run. Using such episodes for questions should be avoided.

Creating fake answers

It is difficult to provide enough number of fake answers to a question like “what was your favorite sport?” because possible answers are 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 easy. 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.

In this way, fake 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[10][21][22].We can use such techniques for getting similar names.

Universality

Although everyone has to use authentication systems on the Internet these days, not all the people are good at handling passwords. Even experienced computer users have trouble with passwords, since choosing a good password is not intuitive and people forget passwords frequently. Using EpisoPass, people can use password-based systems without knowing techniques for creating and remembering strong passwords. They only have to provide questions and answers based on secret episodic memories. By integrating EpisoPass into existing password-based services, people can even use services without noticing that passwords are required for the service.

Frequent password update

On many services, users are requested to change passwords periodically to strengthen security. Using EpisoPass, users can just provide a date-related question like the first question shown in Figure 1, and generate different strong passwords based on the answer. Using this technique, people can easily manage old and new passwords.

Comparison with challenge-response authentication

In many services, users are adviced to define answers to secret questions like “what is your mother’s maiden name” so that he can log into the service even when he forgets the password. This kind of authentication is far less secure than simple password systems, since it is fairly easy for attackers to know the right answer. The variation of system-provided questions are usually small, and such challenge-response systems are considered to be insecure[16].

It would be better if the user can register his own secret questions based on his episodic memory. However, answers to user-generated challenge questions are more easily forgotten or cracked, and using simple challenge-responce is not considered to be safe enough [12][18]. Also, in this case, the user should register the answer to the system in addition to the question. Telling secret facts to service providers is like registering raw password string on the server, and it is not safe unless the strings are properly hashed and salted on the server.

Care for handling secret information

Users don’t have to be very careful about handling questions and answers. They can even be put on a public place if enough amount of questions and fake answers are provided. Keeping secrets is always a pain for many people including the authors, but if the whole questions and answers used on EpisoPass can be put on a public place, handling it becomes very easy. Usually we have to be very careful about handling secret information like passwords, secret key for SSH, etc. because they should not be copied or saved at unsafe places. On the other hand, we can save the EpisoPass data as a plain text file and put it at any place without great care, since malicious person cannot calculate passwords without having the owner’s episodic memories.

Risk of password leak

EpisoPass is just a string-conversion system and it does not have any information on which one is the correct answer. Also, it does not save any password information in any form, and it is almost impossible to get the right password without having the episodic memory.

Since the password is generated by the algorithm and the password string is not kept on a computer in any form, unexpected leak of the password is unlikely. The user doesn’t have to remember the password string, and there is no chance he will reveal it to somebody even when he is forced to tell the password.

Simplicity

The algorithm of generating a password (shown in Appendix) is fairly simple and it can be implemented on any browser JavaScripts or applications on smartphones and small devices.

Risk of password leak at the service side

If one of the passwords generated by EpisoPass is revealed for some reason, other passwords based on the same questions might also be revealed. For example, if Twitter is attacked by a cracker and the password for Twitter (“Lbasnln542202” in Figure 1) is revealed to the attacker, the attacker can try all the answer combinations and find out the answers to all the questions, if questions and answers are also known by the attacker. Once all the answers to the questions are known, the attacker can freely generate all the passwords generated with the same questions.

To prevent such troble, it is safer to keep all the questions and answers in a secret place or use sufficient number of questions so that brute-force does not work.

Using images

Old pictures can be used as the questions of EpisoPass, just like the 5th question shown in Figure 1. Even when people cannot create good secret questions, people can fairly easily select a secret image from their photo collections and use it as the question. For example, if you have an old picture of your friend, you can use it as the question and use his real name as well as other similar fake names.

Of course, the photo should not have information related to the person’s real name, since image search is fairly easy on the Web these days.

RELATED WORK

As shown in previous sections, there are many attemps for replacing password-based authentication systems, none of which has succeeded so far.

Various types of image-based authentication systems have been proposed recently, in the hope that images are easier to remember than text because images are usually more directly linked to episodic memories. However, on many systems, users have to remember new information related to the images used in the authentication process, or perform special operations on the image, and it is not much easier than using passwords. Image-based authentication based on episodic memories might work if anyone can prepare many images that are tightly linked to his episodic memories. However, finding such images is usually not easy, and image-only authentication systems would not take off until simple and effective technique is invented.

Even when a new ideal authentication method is invented, replacing all the password-based systems should take very long, and various password managers should be used until the ideal method prevails in the whole world. 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[19] helps the user to generate password strings using "visual cues" on an arbitrary image. Instead of directly using images for authentication, users use the system to generate a password text with the help of the image shown to the user.

EXPERIENCES

EpisoPass has been used by the authors for more than three years, and the authors are using it for various services including Twitter, Facebook, Amazon, Skype, etc. Since passwords for these services are remembered on the browser, EpisoPass is used only once in one week at most. Before using EpisoPass, taking care of many passwords was really a pain for the authors, but currently all the information for generating passswords is put on the cloud and no trouble has been found during the period.

Unfortunately EpisoPass is currently not used by many people. One big reason is that most people cannot fully understand the idea of EpisoPass and also cannot fully trust it because it is not operated by a big IT company nor supported by famous computer users. Another reason might be that users cannot estimate the strength of EpisoPass. Since all the questions and answers are quite obvious to the users, they might feel that everyone else also know the episode and easily solve all the questions. These kinds of psychological issues are quite important for an authentication system for everybody, and we hope to eliminate such obstacles in the future.

CONCLUSION

We introduced a password management system EpisoPass that converts a seed string into a password using the user's episodic memories represented as a set of questions and answers which can be solved only by the user. Using EpisoPass with well-defined questions and answers, a user can always retrieve various passwords without worrying about remembering secret information. We'll try to integrate the system with existing password-based services, and finally hope to eliminate all the problems derived from password-based authentication.

REFERENCES

1. AgileBits Inc. 1password. `https://agilebits.com/onepassword`.
2. Biddle, R., Chiasson, S., and Van Oorschot, P. Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv. 44*, 4 (Sept. 2012), 19:1–19:41.
3. Bonneau, J., Herley, C., van Oorschot, P. C., and 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* (2012), 553–567.
4. Dandapat, S. K., Pradhan, S., Mitra, B., Roy Choudhury, R., and Ganguly, N. Activpass: Your daily activity is your password. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15 (2015), 2325–2334.
5. Dashlane, Inc. Dashlane. `https://www.dashlane.com/`.
6. Florêncio, D., and Herley, C. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07 (2007), 657–666.
7. Florêncio, D., Herley, C., and Coskun, B. Do strong web passwords accomplish anything? In *Proceedings of the 2nd USENIX workshop on Hot topics in security*, HOTSEC'07 (2007), 10:1–10:6.
8. Hayashi, E., and Hong, J. A diary study of password usage in daily life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11 (2011), 2627–2630.
9. Herley, C., Oorschot, P. C., and Patrick, A. S. Passwords: If we're so smart, why are we still using them? In *Financial Cryptography and Data Security*, R. Dingledine and P. Golle, Eds. Springer-Verlag, 2009, 230–237.
10. Huang, X., Wan, X., and Xiao, J. Learning to find comparable entities on the web. In *Proceedings of the 13th international conference on Web Information Systems Engineering*, WISE'12, Springer-Verlag (2012), 16–29.
11. Internet Safety Project. Graphical passwords. `http://www.internetsafetyproject.org/wiki/graphical-passwords`.
12. Just, M., and Aspinall, D. Personal choice and challenge questions: a security and usability assessment. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09 (2009), 8:1–8:11.
13. KING JIM. Password manager "milpass" pw10. `http://www.kingjim.co.jp/sp/pw10/`, 2012.

14. Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., Cranor, L. F., and 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*, CHI '11 (2011), 2595–2604.

15. LastPass.com. Lastpass. `https://lastpass.com/`.

16. Rabkin, A. Personal knowledge questions for fallback authentication: security questions in the era of facebook. In *Proceedings of the 4th symposium on Usable privacy and security*, SOUPS '08 (2008), 13–23.

17. Reichl, D. Keypass password safe. `http://keepass.info/`.

18. Schechter, S., Brush, A. J. B., and Egelman, S. It's no secret. measuring the security and reliability of authentication via 'secret' questions. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, SP '09 (2009), 375–390.

19. Stobert, E., and Biddle, R. A password manager that doesn't remember passwords. In *Proceedings of the 2014 Workshop on New Security Paradigms Workshop*, NSPW '14 (2014), 39–52.

20. Symantec Corporation. Norton id safe. `http://jp.norton.com/portal-IDsafe/`.

21. Wang, R. C. Boo!wa! http://boowa.com/.

22. Wang, R. C., and 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*, ICDM '07 (2007), 342–350.

23. WoodenSoldier. Id manager. `http://www.woodensoldier.info/soft/idm.htm`.

Appendix: Algorithm for generationg a password

The password string is generated by substituting the characters in the seed string based on the user's selections of candidate answers. Character substitution is performed based on the character types. For example, if the first character of the seed string is a digit, the first character of the generated password string is also a digit.

A digit A in the seed string is converted to a digit B in the password string by the following character substitution function $f_N()$.

$$f_N(x) = (10 + N - x) \bmod 10$$

Here, N is a number calculated from the user's selection of candidate answers. If the value of N if 5, $f_5(x) = (15 - x) \bmod 10$, and the $f_5(x)$ value for each x is calculated like below:

$$f_5(0) = 5$$

$$f_5(1) = 4$$

$$f_5(2) = 3$$

$$...$$

$$f_5(8) = 7$$

$$f_5(9) = 6$$

$f_N()$ depends on the value of N, and it is impossible to know about $f_N()$ without knowing the value of N.

N is calculated by the following algorithm in EpisoPass.

1. Generate a string S by concatenating all the question strings and selected candidate answers.
2. Calculate the MD5 value of S and generate 32-byte hexadecimal string M.
3. For each k-th character of seed string, get the 4-character substring of M beginning at $k \times 4$, and use it as the value of N

   If the hash value M is 12345678..., $f_{\texttt{0x1234}}$ is used as the substitution function for the first character, $f_{\texttt{0x5678}}$ is used as the substitution function for the second character, etc.