EpisoPass: エピソード記憶にもとづくパスワード管理

EpisoPass: Password Management based on Episodic Memories

増井俊之 *

Summary. 忘れることがないエピソード記憶にもとづく秘密の質問を使って強力なパスワードを生成/管理するシステム「EpisoPass」を提案する。EpisoPass は、ユーザが作成した秘密の質問への回答にもとづいてシード文字列を換字することによってパスワードを生成する。シード文字列や回答のバリエーションにより異なるパスワードが生成されるので様々なサービスに対して異なるパスワードを生成できることに加え、シード文字列を逆計算することにより既存のパスワードの管理もできる。適切な運用により、パスワードに関連するあらゆる情報を秘密にすることなく強力なパスワードの生成/管理が可能である。

1 はじめに

個人認証のためにパスワードが現在広く利用されている。パスワード認証には多くの問題があることが知られているが [29]、今後も長期にわたって利用され続けると予想されるため [8]、問題点を認識しつつ適切に運用するための工夫が必要である。

パスワードの長期的記憶が難しいことはパスワード認証の大きな問題点のひとつである。安全に運用するためにはパスワードはランダムで長い文字列であることが望ましいが、そのようなものを頭の中に記憶しておくことは難しい。また複数のサービスを利用する場合、サービスごとに異なるパスワードを利用することが望ましいが、すべてのパスワードを記憶しておくことはほとんど不可能である。Florêncio の 2007 年の大規模な調査によれば、ユーザは平均 25 個のサイトで 6.5 個のパスワードを利用しており、3ヶ月間にユーザの 4.28%がパスワードを忘れていた [5]。また 2011 年の野村総研の調査によれば、一般的なユーザがパスワード認証を行なうサイトは平均 19.4 個で、利用しているパスワードは平均 3.1 個であった [32]。多数のパスワードを記憶することが困難であるため、多くのユーザが同じパスワードを複数サイトで使い回しているのだと思われる。

異なるパスワードをすべて記憶することは不可能なのでどこかに記録しておく必要があるが、パスワード文字列をそのまま記録するのは危険なので、複数のパスワードを秘密情報として扱うためのパスワード管理システムが利用されている。パスワード管理システムはひとつの「マスターパスワード」を利用して他のすべてのパスワードを管理するもので、暗号化されたデータベースにパスワードを格納するもの [1][3][12][14][18][21][24] が多いが、サービス名をもとにマスターパスワードを変換することによって複数のパスワードを生成するシステム [25] もある。両者ともにマスターパスワードの記憶は必須であり、マスターパスワードを盗まれたり忘れたりする危険がある。

一般にユーザはパスワードを忘れがちであるため、多くのサービスにおいてパスワードを復元したり初期化したりする手段が用意されている。ユーザが秘密の質問に対する答を登録し、質問に正しく回答することによってパスワードを復元したりリセットできるサービスは多いし、秘密の質問に答えることによってパスワード管理システムのマスターパスワードを復元するシステム [34] も提案されている。

新しく覚えた情報や新しく考えた情報はどうしても忘れてしまう可能性があるので、新しく作成したパスワード文字列を記憶して認証に利用することは本質的に無理がある。一方、既知で忘れることがないエピソード記憶を秘密の質問として認証のために直接利用することができれば、認証に必要な情報を忘れてしまうことがないはずである。多くの画像認証システム [2][26][10] は秘密の質問に対して適切な操作を行なうことによって認証を行なっているためパスワードのようなものを記憶する必要がない。画像認証システムはまだ普及しておらず利用できる環境は限られているが、忘れないエピソード記憶を利用した秘密の質問への回答を強力なパスワードに変換するシステムがあれば、通常のパスワード認証を用いた現在の様々なサービス上で、認証方法を忘れる心配なく安全に認証を行なうことができるようになる。本論文ではこのようなシステム「EpisoPass」について述べる。

2 EpisoPass

2.1 EpisoPass の原理

EpisoPass は、ユーザが忘れることがない個人的なエピソード記憶を文字列に変換することによって

* Toshiyuki Masui, 慶應義塾大学 環境情報学部

安全なパスワードを生成するシステムである。パスワード文字列は以下の手順で生成される。

1. パスワード生成の「種」となる文字列を用意する。以下ではこれを「シード文字列」と表現する。

2. 忘れることがない個人的なエピソード記憶にもとづく秘密の質問を複数作成し、それぞれについてひとつの正答と複数の偽答を用意する。

3. 質問と回答の組にもとづいてシード文字列に換字操作を行なう。すべてに正しく回答したとき生成される文字列をパスワードとして利用する。

2.2 EpisoPass 利用例

以下に EpisoPass の利用例を示す。

2.2.1 ブラウザでの利用

筆者が twitter のパスワードを生成するためにブラウザで EpisoPass を利用している例を図 1 に示す[1]。シード文字列として「Twitter123456」という文字列を指定しており、4 個の秘密の質問に対する回答選択に応じて「Mfveabn574923」のようなパスワード候補が生成される。異なる答をを選択すると全く異なる文字列が生成される。シード文字列の 8 文字目が数字である場合はパスワードの 8 文字目も数字になるなど、シード文字列の文字種に対応したパスワード候補が生成される。

最初の秘密の質問は筆者の小学校の同級生に関するもので、最後の質問は数年前の体験に関するものである。これらの質問は古いエピソード記憶にもとづいており、筆者が将来答を忘れることはほとんど考えられないが、本人以外がこのような質問に答えることは難しいので正しいパスワードを得ることはできない。

[1] http://EpisoPass.com/masui

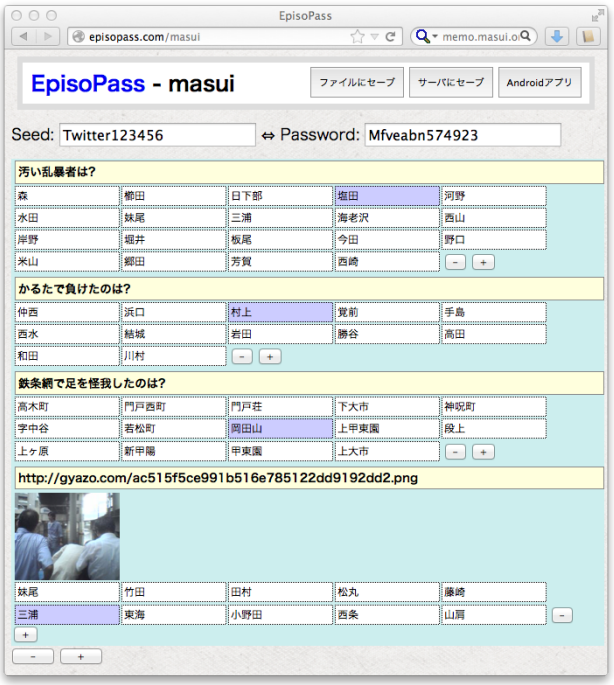

図 1. ブラウザ上で Twitter のパスワードを生成.

秘密の質問と答はブラウザで編集でき、右上の「サーバにセーブ」ボタンを押すことによりシード文字列、秘密の問題、答のリストがサーバにセーブされる。「ファイルにセーブ」ボタンを押すと JSON データをパソコンにダウンロードでき、パソコン上の JSON データをブラウザにドラッグ&ドロップするとサーバにアップロードできる。ユーザはどれが正答かを指定するわけではないので問題データを見てもユーザのパスワードはわからない。

シード文字列を「Facebook123456」に変更すると、生成されるパスワードは図 2 のように変化する。このように、サービスごとに異なるシード文字列を利用することによって様々なパスワードを簡単に生成できる。

図 2. FaceBook のパスワードを計算.

パスワードとして大文字/小文字英数字と記号をすべて利用しなければならないサービスの場合はシード文字列に「PassWord123!@#」のような文字列を指定すればよい。

2.2.2 回答選択によりパスワードを切り換え

サービスごとにシード文字列を変えるのではなく、図 3 のようにサービス名に応じて回答を変えることによって異なるパスワードを作成することも可能である。パスワードとしての長さや文字種に特種な制約が無いサービスに関してはこの方法が便利である。

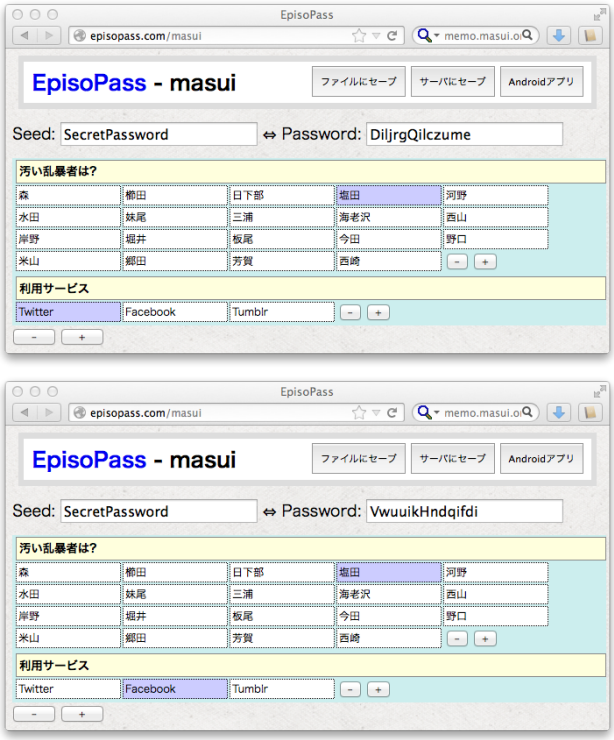

図 3. サービス名に対応する回答で異なるパスワードを生成.

定期的にパスワード変更を求められるシステムでは、「利用期間は?」という質問に対して「2013/1」「2013/2」のような答を用意しておけば、時期を選択することによって簡単にパスワードを変更することができるので便利である。

2.2.3 既存パスワードの利用

現在 “Masui1234” のようなパスワードを利用している場合、図 4 のようにパスワード欄に現在のパスワードを入力すればそれを生成するシード文字列が自動生成されるので、生成されたシード文字列を記録しておけばよい。

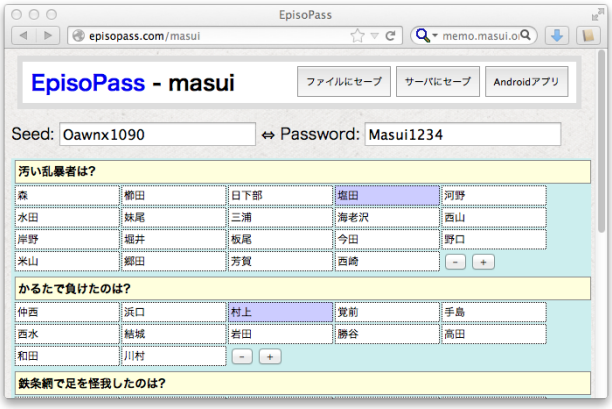

図 4. 既存のパスワードを利用.

パスワード以外の秘密の文字列も管理することができる。たとえば自転車の鍵番号「1234」を EpisoPass で管理したい場合、「jitensha-0000」というシード文字列を入力した後でパスワード枠の数字を「1234」に変更することにより、図 5 のように「jitensha-6145」というシード文字列で鍵番号を管理できるようになる。

図 5. 既存のパスワードを利用.

既存のパスワード管理システムは、利用中のパスワードを記憶するもの [1][3][12][14][18][21][24] と新しいパスワードを生成するもの [25] に分類されるが、EpisoPass はこの両方をサポートしている。

2.2.4 Android アプリ

Web サービスを利用する場合、ブラウザとサーバとの間の通信を記録されたり盗み見されたりされる心配を完全に払拭することはできない。前述の例において、パスワードはブラウザ内部で JavaScript により生成されるので、一度ページを表示した後はネットワークを遮断してもパスワード計算を行なえるようになっているが、最初から全く通信を行なわずにパスワードを作成できる方がより安心であろう。

このため、通信を全く行なわずにマシン単体でパスワード計算を行なうための Android アプリを用意した。ページの右上の「Android アプリ」ボタンを押すと、現在表示している秘密の問題と答を内蔵した Android アプリがサーバ上でビルドされてダウンロードされる。

Android 端末でアプリを実行すると図 6 のような画面が表示される。シード文字列を設定して「開始」ボタンを押すと図 7 のように質問がひとつずつ表示され、ボタンを押してすべて回答するとパスワードが計算され図 8 のように表示される。

図 6. Android アプリ初期画面.

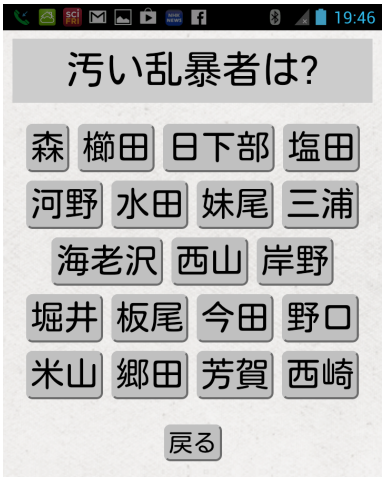

図 7. 最初の問題.

図 8. すべての問題に回答した後の状態.

回答入力とパスワード計算は Android 端末で実行されるため、端末を「機内モード」に設定するなどの方法でネットワーク接続を遮断した状態でもパスワードを計算することができる。EpisoPass をインストールした Android 端末を持っていれば常に各種のパスワードを計算できるので、他人のマシンや公共の場所に設置されたパソコンなどでも容易に twitter などのネットサービスを利用することができる。

前述の方法で EpisoPass アプリをサーバからダウンロードする場合は、ブラウザ上で秘密の問題をサーバに登録する必要があるが、秘密の問題を全くネット上に露出することなくアプリを利用することもできる。秘密の問題を含まない EpisoPass アプリを Google Play で公開 [2] しているので、これを端末にインストールした後、ローカルマシンで作成した秘密の質問を端末に転送すれば EpisoPass.com からダウンロードしたアプリと同様に利用できる。この手法を使うと秘密の質問が通信路を通ることがないので安全であるが、アプリのセットアップの手間は増える。

2.3 パスワード文字列の計算方法

問題と回答から文字列を生成し、その MD5 値によってシード文字列を換字することによりパスワードを生成している。パスワード文字列の計算方法は附録に示す。

3 議論

EpisoPass の利便性、安全性、運用の問題などについて考察する。

3.1 運用形態と安全性

パスワード管理システムにおいて最も重要なのは安全性である。EpisoPass は様々な運用方法が可能であり、運用形態によって安全性の評価が異なるので、利用例にもとづいて EpisoPass の安全性を考える。

3.1.1 EpisoPass の利用を秘密にする場合

パスワード管理に EpisoPass を利用していることを公開せず、単なるパスワード生成機として利用する場合は、普通にパスワードを利用する場合に比べて安全度が低下することは無い。パスワードとして充分強力な文字列をシード文字列として利用すれば EpisoPass によって換字された文字列もパスワードとして強力だと考えられるし、推測しやすい文字列をシード文字列として設定した場合であってもランダムな換字操作によってより強力なパスワードが生成されるので、EpisoPass を利用するデメリットは

[2] https://play.google.com/store/apps/details?id=com.pitecan.episo_pass

無く、通常のパスワード管理システムと同様の方法で利用できる。

3.1.2 シード文字列を秘密にする場合

EpisoPass の秘密の質問を公開した場合でも、シード文字列を通常のパスワードと同じレベルで秘密にしておけば SuperGenPass[25] と同じレベルの利便性と安全性が確保できる。2.2.2 で述べた方法を利用することにより、ひとつのシード文字列から複数のパスワードを生成することができる。

3.1.3 すべて公開する場合

秘密の質問を解くことが不可能であれば、シード文字列と秘密の質問をすべて公開しても安全である。この場合、ユーザは秘密の質問とシード文字列を通常のテキストデータと同じように管理できるし、パスワード管理のために新しく記憶しなければならない情報が皆無なので、ユーザはパスワード管理について注意を払う必要が無くなり気が楽になる。

秘密の質問を解くことを困難にするためには質問の数と偽答の数が多くなければならないし、自分だけが知っているエピソード記憶をうまく秘密の質問にするにはコツが必要である。良い秘密の質問を作る方法に関しては 3.3 で議論するが、すべての質問を公開するのが心配な場合や、秘密の質問の数や質がが充分でないと感じられる場合は 3.1.1 や 3.1.2 の手法で運用し、徐々に運用方法を変えていけば良いだろう。

3.2 秘密の質問の強度

パスワードは長年利用されているため強度や実際の運用に関して多くの研究が存在するが [7][13]、秘密の質問の強度に関しては充分研究されていない。EpisoPass の運用実績は長くないが安全性などについて考察を行なう。

EpisoPass で選択枝が 10 個の秘密の質問を 8 個使用する場合、総当たりでパスワードを生成するには 1 億 (10^8) 通りの試行が必要であり、エントロピーは 26.6 ビットとなる。英字からランダムに 8 文字を並べて作成したパスワードのエントロピーは 37.6 ビットになるが、“pmvixuzq” のように全く意味のないパスワードを記憶して利用することは少ないため、実際に利用されるパスワードのエントロピーは 20 ビット程度と考えられているので [16]、秘密の質問と選択枝の数を 10 個程度用意すれば通常のパスワードと同程度の強度が期待できることになる。総当たり攻撃が可能なオフライン運用ではエントロピーの大きさは重要であるが、オンラインサービスではパスワード入力を何度か間違えるとサービスがブロックされるのが普通なので、それほど長いパスワードを用意する必要は無いと考えられている [6]。

一方、秘密の質問を利用する認証の脆弱性を利用した攻撃が近年問題になっている。パスワードを忘れたときのために、あらかじめ設定した秘密の質問に答えることによってパスワードをリセットできるサービスがあり、「母親の旧姓は?」や「最初に飼ったペットの名前は?」のような質問に対してユーザが答を登録するようになっている。このような問題は他人が調べたり推測したりすることが容易であるうえに秘密の質問の数は一般的に少なく、パスワードよりも脆弱だといえる [17]。ユーザが作成した秘密の質問を使えばこのような問題はなくなるはずであるが、他人に解かれにくい問題をユーザが作成することは難しく、またユーザ自身が答を忘れてしまうことも多いと考えられている [11][20]。また、古い記憶にもとづいて作成した秘密の問題はユーザが想像するよりも解かれやすいという実験結果にもとづき、忘れない秘密の質問を複数利用する方法、問題と答を連想するために画像を利用する方法、複数の問題を連続的に利用する手法などが提案されている [19]。

EpisoPass では、他人には解くことが難しく自分では忘れないような秘密の質問を自由にいくつでも利用できるようになっている。問題作成に慣れていないユーザには有効な秘密の質問を作成することは難しいかもしれないが、次節で述べるように、適切な質問を選ぶことによりこの問題を解決できるはずである。

3.3 秘密の質問の選択

EpisoPass 利用において秘密の質問の選択は非常に重要である。他人が推測することが難しく、自分が決して忘れないようなエピソード記憶を秘密の質問として利用すべきであり、以下のような性質をもつ記憶は秘密の質問として利用すべきではない。

- 自慢になるもの (何かの機会にうっかり他人に話してしまう可能性がある)
- ネット上に記録が残っているもの
- 他人と情報を共有しているもの
- 趣味や嗜好に関連するもの (他人に推測されやすいうえに嗜好が変化する可能性がある)

このようなものではなく、「わざわざ人に話すことはないが自分の記憶に強く残っているような無難なエピソード記憶」を秘密の質問として利用するのが良いであろう。具体例としては以下のようなものがある。

- 昔のちょっとした怪我の場所や種類
- 昔のちょっと悔しい思い出
- 昔何かを見つけた場所

たとえば図 9 のような秘密の質問は他人に話したことが無いが、痛い思いをしたことは忘れないし、

偽答を作成するのも簡単なので、認証のための秘密の質問として適切であると考えられる。

図 9. 昔の怪我に関する秘密の質問.

3.4 偽答の作成方法

秘密の質問の種類によっては偽答の生成が難しい場合がある。図 10 は電子工作に関する秘密の質問の例であるが、正答と区別がつかない偽答を充分リストすることは難しい。

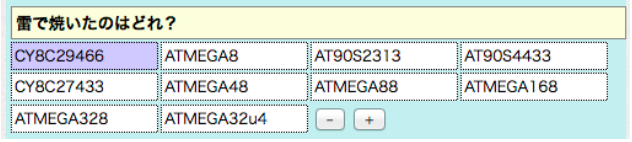

図 10. 偽答の生成が難しい例.

一方、正答として人名や地名を利用する場合、正答に似た人名や地名をリストすることは難しくない。「世田谷」が正答であるとき、「目黒」「杉並」のような偽答を用意するのは簡単である。正答と同じカテゴリに属する単語を自動的にリストすることができれば正答をもとにして簡単に偽答のリストを生成することができる。ひとつの単語もしくは単語の集合と同じカテゴリに属する単語を検索する手法は「同位語検索」と呼ばれ、Web のデータを利用した様々な同位語検索システムが提案されている [9][22][23][33]。たとえば「ライバルサーチ『やーやら』」[33] は、Web 上の「A や B が」のような表現を抽出することによって A と B のカテゴリが近いことを判断している。

人名や地名の偽答を作成したい場合は人名や地名のデータベースを利用して偽答を生成することができる。市町村の人口ランキングや位置関係のデータなどを利用すれば似た地名を偽答としてリストすることが可能であるし、人名ランキングを利用すれば似た苗字を偽答とすることができる。たとえば日本の名字ランキングの 40 位近辺に「長谷川」「近藤」「石井」「斉藤」「坂本」「遠藤」「藤井」などの名字があるので、「石井」が正答のときこれらの名字を偽答にすればよい。しかし、「小学生のとき〜だった同級生は誰?」という秘密の質問の正答が「石井」であるときこの方法で偽答を生成すると、「長谷川」「近藤」などの同級生の存在を確認することにより正答が「石井」であることが判明してしまう可能性があるので注意が必要である。

3.5 パスワード漏洩時の問題

秘密の質問を公開している場合、シード文字列とパスワードの対応がひとつでも漏洩してしまうと、総当たり計算でチェックすることにより、すべて秘密の質問の正答が判明してしまう。秘密の質問の正答を知っていればシード文字列からパスワードを計算することができるので、漏洩した秘密の質問は利用不可能になってしまう。3.1.1 や 3.1.2 のような運用をしている場合はパスワードがひとつ漏洩しても他のパスワードは安全だが、3.1.3 のような運用をしている場合はひとつでもパスワードが漏洩するとあらゆるパスワードが漏洩してしまうことになる。

通常の Web サービスなどのパスワードが漏洩することは考えにくいが、「自転車の鍵番号」のように家族などで共有する可能性があるものに対して 3.1.3 のような運用を行なうと、鍵番号を知っている人物が総当たり攻撃を行なうことによって秘密の質問の答が判明してしまう。シード文字列や秘密の質問を秘密情報として扱わない場合はパスワードを他人と共有しないように注意する必要がある。

3.6 正答の指定手法について

EpisoPass では秘密の質問の答をリストから選択する方法をとっているが、秘密の質問によってパスワードを回復できる Web サービスなどではユーザが回答文字列を入力する方法をとっているものが多い。リストから答を選択するよりも文字列を入力する方がより安全と思われるが、文字端末でのコマンド入力が GUI のメニュー操作より難しいのと同様に、入力するテキストを忘れたり間違えたりする可能性が高くなってしまうというデメリットがある。「日本の首都は?」のような簡単な質問であっても「東京」「Tokyo」「tokyo」「toukyou」のどれが正解か迷うかもしれない。自由なテキスト入力を行なうよりも、選択式の回答を多数提示して選択させる方が高速に誤りのない操作が可能であり、利便性が高く充分安全な運用が可能である。

3.7 画像認証の利用

忘れにくいエピソード記憶を利用する認証手法として様々な画像認証システム [2][10][26] が提案されている。複数の画像の中から正答を選択するもの (Cognometric 方式)、ひとつの画像の中の特定の場所を指定するもの (Locimetric 方式)、画像の上で描画操作を行なうもの (Drawmetric 方式) が広く利用されているが [2][10][27]、EpisoPass は図 1 のように文字列のかわりに画像 URL を秘密の質問として利用する点が異なっている。Cognometric 方式はエピソード記憶を効果的に利用できるが、多数の偽答画像が必要だという問題がある。Locimetric 方

式はエピソード記憶を効果的に利用できないことに加え、クリックしやすい「ホットスポット」は限られているため充分なエントロピーを確保できないことが問題になる [4]。また Drawmetric 方式もエピソード記憶を効果的に利用できないし、ユーザは似た傾向のストロークを選びがちであるため充分なエントロピーを確保しにくいことが知られている [15]。EpisoPass のように画像を秘密の質問として利用する場合、通常の秘密の質問の場合と同様に偽答を増やすことが容易であることに加え、画像に関連したエピソード記憶を有効に利用できるという利点がある [30]。

図 11 は本棚.org[28][31] のユーザのひとり [3] が利用している画像認証問題である。このユーザ以外には正答は見当もつかないが、本人にとっては忘れることがないエピソード記憶と結びついた画像だということであった。

図 11. エピソード記憶と深く結びついた画像.

4 結論

エピソード記憶に結びついた秘密の質問を利用してパスワードを生成/管理できるシステム EpisoPass を提案した。EpisoPass は単純な原理にもとづいており柔軟な利用形態が可能であり、強力な秘密の質問を用意することにより秘密情報を全く覚えることなく安全な認証を行なうことができる。強力な秘密の質問を作成して安全に運用が可能かどうかを長期的に評価したいと考えている。EpisoPass は http://Episopass.com/で運用しており、ソースは GitHub で公開している [4]。

参考文献

[1] AgileBits Inc. 1Password. https://agilebits.com/onepassword.

[2] R. Biddle, S. Chiasson, and P. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv.*, 44(4):19:1–19:41, Sept. 2012.

[3] Dashlane, Inc. Dashlane. https://www.dashlane.com/.

[4] A. E. Dirik, N. Memon, and J.-C. Birget. Modeling user choice in the PassPoints graphical password scheme. In *Proceedings of the 3rd symposium on Usable privacy and security*, SOUPS '07, pp. 20–28, 2007.

[5] D. Florêncio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pp. 657–666, 2007.

[6] D. Florêncio, C. Herley, and B. Coskun. Do strong web passwords accomplish anything? In *Proceedings of the 2nd USENIX workshop on Hot topics in security*, HOTSEC'07, pp. 10:1–10:6, Berkeley, CA, USA, 2007. USENIX Association.

[7] E. Hayashi and J. Hong. A diary study of password usage in daily life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2627–2630, 2011.

[8] C. Herley, P. C. Oorschot, and A. S. Patrick. Passwords: If We're So Smart, Why Are We Still Using Them? In R. Dingledine and P. Golle eds., *Financial Cryptography and Data Security*, pp. 230–237. Springer-Verlag, 2009.

[9] X. Huang, X. Wan, and J. Xiao. Learning to find comparable entities on the web. In *Proceedings of the 13th international conference on Web Information Systems Engineering*, WISE'12, pp. 16–29, Berlin, Heidelberg, 2012. Springer-Verlag.

[10] Internet Safety Project. Graphical Passwords. http://www.internetsafetyproject.org/wiki/graphical-passwords.

[11] M. Just and D. Aspinall. Personal choice and challenge questions: a security and usability assessment. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pp. 8:1–8:11, 2009.

[12] KING JIM. パスワードマネージャー「ミルパス」PW10. http://www.kingjim.co.jp/sp/pw10/, 2012.

[13] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. 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, pp. 2595–2604, 2011.

[14] LastPass.com. LastPass. https://lastpass.com/.

[15] D. Nali and J. Thorp. Analyzing user choice in graphical passwords. Technical Report TR-04-01, School of Computer Science, Carleton University, Ottawa, 2004.

[16] National Institute of Standards and Technology. Electronic Authentication Guideline. NIST Special Publication 800-63-1. http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf, 2011.

[3] http://hondana.org/Leiko
[4] http://GitHub.com/masui/EpisoPass

[17] A. Rabkin. 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, pp. 13–23, 2008.

[18] D. Reichl. KeyPass Password Safe. http://keepass.info/.

[19] K. Renaud and M. Just. Pictures or questions?: examining user responses to association-based authentication. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, BCS '10, pp. 98–107, 2010.

[20] S. Schechter, A. J. B. Brush, and S. Egelman. 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, pp. 375–390, 2009.

[21] Symantec Corporation. Norton ID Safe. http://jp.norton.com/portal-IDsafe/.

[22] R. C. Wang. Boo!Wa! http://boowa.com/.

[23] R. C. Wang and W. W. Cohen. Language-Independent Set Expansion of Named Entities Using the Web. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pp. 342–350, 2007.

[24] WoodenSoldier. ID Manager. http://www.woodensoldier.info/soft/idm.htm.

[25] C. Zarate. SuperGenPass. http://supergenpass.com/.

[26] 小池 英樹, 増井 俊之, 高田 哲司. 画像を用いた個人認証手法. 情報処理, 47(5):479–484, May 2006.

[27] 首相官邸. 電子政府ガイドライン作成検討会 セキュリティ分科会報告書. http://www.kantei.go.jp/jp/singi/it2/guide/security_guide_line/siryou2.pdf, 2010.

[28] 増井 俊之. 本棚通信: 控え目なグループコミュニケーション. インタラクション 2005 論文集, pp. 135–142, February 2005.

[29] 増井 俊之. マイ認証. *Unix Magazine*, 21(3), March 2006. http://www.pitecan.com/UnixMagazine/PDF/if0603.pdf.

[30] 増井 俊之. パスワードとの闘い - パスワードなし認証システムの運用報告. コンピュータセキュリティシンポジウム 2009 論文集, 2009.

[31] 本棚.org. http://hondana.org.

[32] 野村総合研究所. 利用者登録する商品・サービスを選別する傾向が強まった生活者と顧客情報の鮮度維持を望む事業者〜生活者と事業者を対象とした ID に関する実態調査〜. http://www.nri.co.jp/news/2012/120208.html, February 2012.

[33] 大島 裕明, 小山 聡, 田中 克己. Web 検索エンジンのインデックスを用いた同位語とそのコンテキストの発見. 情報処理学会論文誌. データベース, 47(19):98–112, December 2006.

[34] 平野 亮, 森井 昌克. パスワード運用管理に関する考察および提案とその開発. 電子情報通信学会技術研究報告. ISEC, 情報セキュリティ, 111(285):129–134, November 2011.

附録: パスワード計算アルゴリズム

パスワードは、秘密の質問と回答の組合せにもとづいてシード文字列を換字することによって計算される。換字は文字種ごとに行なわれる。たとえばシード文字列の 1 桁目が数字のときはパスワードの 1 桁目は数字に変換され、シード文字列の 1 桁目が記号のときはパスワードの 1 桁目は記号に変換される。

シード文字列内の数字 A は、換字関数 $f_N()$ によってパスワード内の数字 B に変換される。$f_N()$ は以下のような関数である。

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

ここで N はシード文字列と回答の組み合わせから計算される自然数で、答の選択により変化する。たとえば $N = 5$ のとき、$f_5(x) = (15 - x) \bmod 10$ となるので、x と $f_5(x)$ の対応は以下のようになる。

$$f_5(0) = 5$$
$$f_5(1) = 4$$
$$f_5(2) = 3$$
$$\ldots$$
$$f_5(8) = 7$$
$$f_5(9) = 6$$

$f_N()$ は N により変化するので、N がわからなければ $f_N()$ もわからない。N はシード文字列と回答の組み合わせから計算される自然数なので、シード文字列の答を知らなければ N を計算することはできず、問題と答の組み合わせを知らない限りシード文字列からパスワードを計算することはできない。

EpisoPass では以下のようにして $f_N()$ を計算している。

1. 問題文字列と選択した答の文字列の組を連結して長い文字列 S を生成
2. S の MD5 ハッシュ値 M(16 進 32 桁の文字列) を計算
3. シード文字列の k 桁目の文字に対し、M の $k \times 4$ 文字目から $k \times 4 + 3$ 文字目までの部分文字列 (16 進 4 桁) を取得し、それを N とする

   たとえばハッシュ値 M が 12345678... だったとき、1 桁目を計算する $f()$ については N=0x1234 となる。