Vol.2016-DPS-166 No.21
Vol.2016-CSEC-72 No.21
2016/3/4

ワンタイムパスワード認証システムの利便性について

糸井 正幸[1,a)]　多田 充[2,b)]

概要：ワンタイムパスワード (OTP) 認証は，(固定) パスワード認証が抱える安全性の問題を解決するための 1 つの手段として，金融関係サイトなど一部のシステムで採用されている。しかし，通常のパスワード認証と比べて，安全性においては優位であるものの，その利便性については解決すべき問題が多い。我々の研究グループは，2011 年開催のコンピュータセキュリティシンポジウム (CSS2011) 以降，2 要素 3 者間による OTP 認証システムに取り組み，必要となる各処理に対するプロトコルを構築，その安全性の解析を行うと共に，システムの機能を拡張することが可能であることを示した。しかし，それらを実際構築しサービスとして提供するためには，ユーザから見た利便性も重要な要素となる。本稿では，他の OTP 認証方式と比較しつつ，我々の提案システムの利便性，およびその改良可能性について述べる。

キーワード：ワンタイムパスワード認証，2 要素認証，3 者間認証

On the user-friendliness of one-time password authentication systems

Masayuki Itoi[1,a)]　Mitsuru Tada[2,b)]

Abstract: One-time password (OTP) authentication is introduced to some systems like financial institution sites, as one of the solutions for the security problems usual (fixed) password authentication has. To be sure that OTP authentication has an advantage on the security, but there is no denying that it has some problems on user-friendliness. We have been investigating OTP systems with two-factor authentication among three parties since CSS2011, constructing protocols for each process, analyzing the security and presenting the possibility of expanding the functions. However, we can see that the user-friendliness is an important character when we think of the actual system being provided as a service. In this paper, we give how to improve the user-friendliness of our OTP authentication system, seeing those of other OTP systems.

Keywords: One-time password authentication, Two-factor authentication, Authentication among three-parties.

1. はじめに

現在，ネットワーク上には様々なサービスシステムが存在し，ユーザはその恩恵を受けている。個々のユーザに対応したサービスを提供するシステムの場合，ユーザがアクセスしたときにログインの処理をさせ，アクセスしてきた者が (サービスシステムに登録されている) どのアカウントに対するユーザなのかを識別・認証している。その際利用されている認証方法として最も普及しているのが「ユーザ ID&(固定) パスワード」によるもの ((固定) パスワード認証) である。

固定パスワード認証は「記憶情報による認証 (SYK)」であり，IPA が 2014 年 8 月に公開した「オンライン本人認証方式の実態調査報告書」[1] によれば，調査したサイトの全てで採用されている。確かに，固定パスワードによる認証システムは，導入が容易であり，ユーザにも馴染み深く，

[1] 株式会社セフティーアングル
Safety Angle Inc., Ichikawa, Chiba 272–0826, Japan
[2] 千葉大学 統合情報センター
Institute of Media and Information Technologies, Chiba University, Inage, Chiba 263–8522, Japan
[a)] m.itoi@safetyangle.co.jp
[b)] m.tada@faculty.chiba-u.jp

非常に便利な認証方法であることには間違いないが，総当たり (ブルートフォース) 攻撃や辞書攻撃，いわゆるリバースブルートフォース攻撃やリスト型アカウントハッキングなどのパスワードクラッキングの脅威に晒されていることも事実である。

固定パスワード認証の弱点は，ユーザ自身にパスワードの設定を任せていることに起因する。つまりユーザ自身が，単純な文字列をパスワードとして設定したり，同一のパスワードを複数のサービスシステムで設定したりすることによって，クラックされやすくなるのである。

ワンタイムパスワード (OTP) 認証は，(固定) パスワード認証の弱点を克服し安全性を高めるための手段の 1 つとして，一部のサービスシステム，特に金融サービスシステムで比較的高い割合で採用されている [1]。

我々の研究グループは，2011 年 [3] 以来，OTP 認証システムに関する研究発表を行ってきた [4][7]。これは，従来の OTP 認証システムとは違い，ユーザ，サービスシステムの他に，OTP の発行を行う発行センターが登場する 3 者間認証であり，ユーザが所有する携帯電話機器を利用し発行センターに対して (記憶および使用機器の)2 要素認証を行うことにより，発行センターにより発行された OTP がユーザの携帯電話機器 (およびサービスシステム) に通知されるというものである。提案システムは，OTP 認証システムとして比較的普及している (第 2 章で紹介する) ハード/ソフトトークンによる方式とは違い，OTP を生成するエンティティとして登場するのが発行センターのみなので，システム全体としては発行センターの運用コストがかかるものの，OTP 生成ロジックを秘匿にするのが容易であり，ロジック更新の際にユーザが行わなければならない作業は特にないなどの利点を持つ。

しかし，我々のシステムも含め，一般的に OTP 認証は，ユーザから見ると 面倒/手間のかかる/解りづらい ものである。特に，OTP 生成トークン紛失の際には，ほぼ再登録に近い処理が必要となる。ハードトークンの場合は，新しい機器の用意，その機器に対する処理，および，その機器の安全な配付は，サービスシステムが行わなければならないことであり，そのコストは決して無視できない。

本論文では，ユーザが OTP を取得するための携帯電話機器を，ユーザの意思，または意思とは関係なく，紛失/盗難/故障/破損 等の理由により機種変更する場合に，アカウント情報を再設定する必要なく，ユーザにとって手間のかからない方法を述べる。これにより，OTP 認証システムの利便性の向上が期待できる。

本論文は以下の通り構成される。第 2 章では現在普及している OTP 認証方法を紹介し，第 3 章で我々が発表してきた方法の概略を述べる。ここれは，表記を簡潔にするため，オプション扱いできるパラメータ等はなるべく省略して記述している。第 4 章において，ユーザにとって手間のかからない機種変更方法を述べる。第 5 章では，本提案手法が他の OTP 認証システムに適用できるかどうかを述べ，さらに，第 4 章で登場するユーザパラメータの選択・設定方法について述べ，第 6 章で本論文をまとめる。

2. ワンタイムパスワード認証

ワンタイムパスワードは，文字通り，一度きりの使い捨てのパスワードであるが，パスワードであるため，どのようにしてユーザ⇔サービスシステム間で正しく共有し認証できるようにするかが問題になる。それを解決する手段により，OTP 認証を下記のように分類することができる。

2.1 マトリクス認証方式

マトリクス認証は，ユーザがログインしようとするとき，サービスシステムが碁盤目状の数字の列を提示し，予めユーザが登録している盤面上の位置に従ってパスワード文字列を定めるというものである。しかし，固定パスワード方式と同様，記憶のみの認証になっているという弱点はあるが，以下で述べる「乱数表方式」「トークン方式」とは違い，配付物もクライアントソフトも必要としないという利点をもつ。

2.2 乱数表方式

乱数表方式は，登録時にサービスシステムが個別に配付した乱数表を使ったものである。ユーザがログインするとき，サービスシステムが表の行と列を指定し，ユーザはその乱数表に記載されている数値を入力するというものである。所有物認証の 1 つと考えることができるものの，その乱数表の複製や共有は容易であり，また，ログイン時にユーザが紙媒体である表を見ながら行と列を辿りながら数値を入力しなければならないという欠点もあるが，導入コストは一般的に低く抑えることができるという利点もある。

2.3 ハード/ソフト トークン方式

ユーザがサービスシステムに登録されるときに，サービスシステムが OTP の生成を行うプログラムを個別に配付し，ユーザはログインする際，そのプログラムを実行することにより OTP を入手するというものである。プログラムが専用デバイスに実装されたものや，スマートデバイス上で動作するアプリとして配付されたりするものがある。前者の場合，そのデバイスをハードトークンといい，後者の場合，そのアプリをソフトトークンという。ハードトークンの場合，ユーザはサービスを利用するとき，その専用デバイスが必要となるが，紛失や更新に伴う運用コストは無視できない。ソフトトークンの場合，そのようなコストは発生しにくい。

3. 3者間OTP認証システム [3]

本章では，[3] で示されている2要素3者間によるOTP認証方式の概略を述べる。登場するエンティティはユーザ ($\mathcal{U}$)，サービスシステム ($\mathcal{S}$) および発行センター ($\mathcal{C}$) の3者である。我々が発表した [2] では，複数のサービスシステムを $\mathcal{S}_i^{(j)}$ のように表したが，本論文では表記を簡単にするため，サービスシステムは1つ ($\mathcal{S}$) のみとする。ユーザは，サービスシステムを利用するための端末 $\mathcal{U}_\mathrm{P}$ と，発行センターにOTP発行等のリクエストを送るための携帯電話機器 $\mathcal{U}_\mathrm{M}$ を持つ [*1]。

ユーザは，まずサービスシステム $\mathcal{S}$ にユーザ登録する。これは，一般に普及しているサービスシステムで行っているのと同様の手続きで行ってもよい。この時点でユーザ $\mathcal{U}$ とサービスシステム $\mathcal{S}$ の間で，$\mathcal{U}$ が $\mathcal{S}$ を利用する際のID情報 (ユーザID，uID)，およびその識別符号 fPw を共有しているものとする。また，サービスシステム $\mathcal{S}$ は，サービスシステムID(sysID) を持ち，sysID は当該サービスシステムに関する情報と共に $\mathcal{C}$ に登録されているものとする。

$\mathcal{U}$ が本3者間認証システムに登録を済ませると，以下の状態になる。

(C1) $\mathcal{S}$ における $\mathcal{U}$ のアカウント (uID) に紐付けられたID情報 (管理マスタID，mID) が，$\mathcal{S}$ と $\mathcal{C}$ の間に共有される。

(C2) $\mathcal{U}$ が所有する $\mathcal{U}_\mathrm{M}$ と $\mathcal{C}$ の間にID情報 (アプリケーションID，aID) が共有され，$\mathcal{C}$ においてaIDは前項(1)におけるmIDと紐付けられる。

(C3) $\mathcal{U}_\mathrm{M}$ には，$\mathcal{C}$ が生成したパラメータ (pass) が暗号化された状態で保存され，正しく復号するためには $\mathcal{U}_\mathrm{M}$ を用いることが必要である。

以上の状態を作り出すため，$\mathcal{U}$ は $\mathcal{S}$ および $\mathcal{C}$ に対して，以下に示す2つのフェーズからなる登録手続きを行う。

3.1 登録手続き (その1)

$\mathcal{S}$ が ($\mathcal{S}$ に) 登録されているユーザに紐付けられるためのアカウントを $\mathcal{C}$ に追加するリクエストを送り，その ($\mathcal{C}$ における) アカウントに対して $\mathcal{U}$ が紐付け登録できるようにするための登録チケットを ($\mathcal{C}$ が) 発行する手続きである。

(R1) $\mathcal{U}$ は，$\mathcal{S}$ に，3者間認証登録リクエストとして，uIDを送る。

(R2) $\mathcal{S}$ は，$\mathcal{C}$ に，アカウント追加リクエストとして，sysIDを送る。

(R3) $\mathcal{C}$ は，そのデータベースに1つアカウントを追加し，それに対して一意的な管理マスタID(mID) を生成し，sysIDも併せて割り当てる。さらに $\mathcal{C}$ は，登録チケットticket，および，ticketの正しさを検証するのに必要な情報verTicketを生成 [*2] し，verTicketを当該アカウントに割り当てる。
なお，ticketには，それが $\mathcal{C}$ におけるどのアカウントに対応するかを示す情報 [*3] が含まれているものとする。

(R4) $\mathcal{C}$ は，mIDおよびticketを，$\mathcal{S}$ に返す [*4]。

(R5) $\mathcal{S}$ は，mIDとuIDを自身のデータベースに紐付けし，ticketを $\mathcal{U}$ に渡す。

3.2 登録手続き (その2)

前節において $\mathcal{C}$ が発行したticketを用いて，$\mathcal{U}$ が $\mathcal{C}$ に登録する手続きである。

(R6) $\mathcal{U}$ は，OTP発行依頼パスワードreqPwを定め，$\mathcal{U}_\mathrm{M}$ を用いて，ticketと共にreqPwを $\mathcal{C}$ に送る。

(R7) $\mathcal{C}$ は，送られてきたticketに該当するアカウントを探し，当該アカウントに対するverTicketを用いてticketの正しさを検証する。

(R8) 前項が正しい場合は，一意的なアプリケーションIDaID，reqPwを検証するのに必要な情報verReqPw，リクエスト依頼パスpass，および，passを検証するのに必要な情報verPassを生成し，verReqPwおよびverPassを当該アカウントに登録する。
なお，passには，aIDなど，$\mathcal{C}$ に登録されているどのアカウントに対するものかを示す情報が含まれているものとする。

(R9) $\mathcal{C}$ は，当該アカウントにaIDおよびverPassを登録し，passを $\mathcal{U}_\mathrm{M}$ に返す [*5]。

(R10) $\mathcal{U}_\mathrm{M}$ は，それ自身の機器固有情報を鍵としてpassを暗号化し保存する。
なお，暗号化の鍵にreqPwを含めてもよい。

以上の手続きにより，前述の (C1)〜(C3) の状態を作り出すことができる。実際，$\mathcal{U} \Leftrightarrow \mathcal{S}$ 間でuID，$\mathcal{S} \Leftrightarrow \mathcal{C}$ 間でmID，$\mathcal{C} \Leftrightarrow \mathcal{U}(\mathcal{U}_\mathrm{M})$ 間でaIDが共有されており，passは，$\mathcal{U}$ が登録手続きに使用した携帯電話機器以外では正しく復号されない。

[*1] ここでは，ユーザが2つの機器を用いるように記述しているが，$\mathcal{U}_\mathrm{P}$ は $\mathcal{U}_\mathrm{M}$ と同一の機器でも差し支えない。

[*2] verTicketはticketをどのように定めるかによって決まる。例えば，ticketが $\mathcal{C}$ によるMACであれば，verTicketはそのMACを検証するのに必要なメッセージおよび秘密鍵となり，公開鍵に基づく電子署名を用いて作成されたものであれば，verTicketはメッセージおよび署名検証鍵になる。また，ticketが単なる乱数列であれば，verTicketはその文字列を照合するのに必要となる，その文字列そのもの，または，そのハッシュ値等になる。

[*3] 連番や，一時的に生成された (一意的な) 文字列でよい。

[*4] $\mathcal{C}$ は，登録チケットに有効期限や独自パスワード等を定めて，それも併せて $\mathcal{S}$ に通知してもよいが，ここではそれらの表記は省略する。

[*5] 実際には，ユーザが $\mathcal{U}_\mathrm{M}$ を操作するとき，passがどのサービスシステムに対応するものかを識別できる必要があるため，passの他，当該サービスシステムの名称等も送信する必要があるが，ここでは表記の簡単のため，それらの表記を省略する。

3.3 OTP 発行手続き

ユーザは以下の手続きにより，$\mathcal{C}$ に OTP を発行してもらう。

(O1) $\mathcal{U}$ は，reqPw を $\mathcal{U}_{\mathrm{M}}$ に入力する。

(O2) $\mathcal{U}_{\mathrm{M}}$ は，自身の機器固有情報を用いて pass を入手し，reqPw と共に $\mathcal{C}$ に送る。

(O3) $\mathcal{C}$ は，pass から対応するアカウントを割り出し，当該アカウントの verReqPw および verPass を用いて，reqPw および pass の正しさを検証する。

(O4) 前項が正しい場合，$\mathcal{C}$ はワンタイムパスワード otp を生成し，当該アカウントに登録されているサービスシステム ID(sysID) に，mID および otp を送る。
ここで，sysID に対するサービスシステムを $\mathcal{S}$ とする。

(O5) $\mathcal{S}$ は，mID に対応するユーザアカウントに対して otp，および，その有効期限等 (info) を設定し，info を $\mathcal{C}$ に返す [*6]。

(O6) $\mathcal{C}$ は，otp および info を $\mathcal{U}_{\mathrm{M}}$ に送り，$\mathcal{U}$ は otp とそれに関する情報を知ることができる。

以上の手続きにより otp を入手した $\mathcal{U}$ は，uID および otp(&fPw) を用いて $\mathcal{S}$ にログインすることができる。

4. 機器認証における機器更新手法

本システムでは OTP の発行依頼には，ユーザ所有の携帯電話機器を用いる。ユーザの携帯電話なので，機種変更，場合によっては 紛失/盗難/故障/破損 などにより，機器を変更せざるを得ない場合がある。しかし，pass は携帯電話機器固有の情報を用いて暗号化されるため，たとえ $\mathcal{U}_{\mathrm{M}}$ 内の情報を新機器に移すことができたとしても，異なる機器で処理されるので，正しい pass が得られなくなる。

4.1 基本的なアイデア

我々の研究グループは，[7] において機種変更方法を述べたが，その方法は全く新規登録するものではないにしろ，プロトコルに $\mathcal{S}$ が含まれるため，ユーザが利用しているサービスシステムの個数分だけ手続きを行う必要があり，ユーザにとって多少面倒なものになっている。

そこで，$\mathcal{U}_{\mathrm{M}}$ 内の情報 (pass を含むテーブル，pList とする) を機器外に保存し，新しい機器にその情報を取り込むことを考える。安全性および完全性を満たすためには，最低限，以下の条件が必要であると考えられる。

(1) pList は，何かしらの鍵 K で暗号化された状態で (つまり，$\mathsf{Enc}_K(\mathtt{pList})$ として) エクスポートされなければならない。

(2) 前項で用いる鍵 K は，機器自体ではなく，所有者に紐付いた情報 (uInfo) で生成する必要がある。つまり，機器が変更されても，同一ユーザの所有物であるならば，同一の uInfo が割り当てられなければならない。

(3) 前項の uInfo は，悪意のあるユーザによるなりすましが困難でなければならない。

これらを満たす処理により，ユーザが適切にバックアップを取っていれば，機器が変更されても，$\mathcal{S}$ や $\mathcal{C}$ の助けがなくとも，使用し続けることができる。しかし，バックアップを取ることはユーザの負担ともなるため，次節において，保管システムを用いた自動バックアップのための手続きを述べる。バックアップが行われるタイミングは，ユーザの指示があったときの他，登録/削除 などにより pList が変更されたときが考えられる。

4.2 保管システムを用いた保管方法

ここでは保管システムを用いた保管 (バックアップ) プロトコルについて述べる。$\mathcal{U}$ が所有する携帯電話機器 $\mathcal{U}_{\mathrm{M}}$ には，$\mathcal{U}$ に紐付いた情報 (uInfo) が組み込まれており，$\mathcal{U}$ が機器を $\mathcal{U}'_{\mathrm{M}}$ に変更した場合でも，同じ uInfo が組み込まれるものとする。以下に述べるプロトコルにおいて，h は衝突困難な一方向性ハッシュ関数とする [*7]。

(D1) $\mathcal{U}_{\mathrm{M}}$ は自身の機器固有情報を用いて pass を復号し，pList を入手する。

(D2) $\mathcal{U}_{\mathrm{M}}$ は uInfo を用いて，鍵 K およびタグ $t = h(\mathtt{uInfo})$ を計算する [*8]。

(D3) $\mathcal{U}_{\mathrm{M}}$ は，dInfo$(= (t, \mathsf{Enc}_K(\mathtt{pList}))$ を保管リクエストとして $\mathcal{D}$ へ送り，$\mathcal{D}$ は dInfo を保管する。

また，$\mathcal{U}$ が機器を変更したとき，バックアップから以前の pList を新しい携帯電話機器 $\mathcal{U}'_{\mathrm{M}}$ に復元するために，以下の引き出し手続きを行う。

(W1) $\mathcal{U}'_{\mathrm{M}}$ は uInfo(および，ある場合は dPw) を用いて，鍵 K およびタグ t を計算し，引き出しリクエストとして t を $\mathcal{D}$ に送る。

(W2) $\mathcal{D}$ はタグ t をもつエントリ dInfo$= (t, E)$ が存在するならば，dInfo を $\mathcal{U}'_{\mathrm{M}}$ に返す。

(W3) $\mathcal{U}'_{\mathrm{M}}$ は K を用いて pList$(= \mathsf{Dec}_K(E))$ を計算し，自身の機器固有情報を用いて pList の一部または全部を適切に暗号化して保存する。

しかし， この引き出し手続きにより，同一ユーザであればクローン携帯電話機器を無制限に作成することができてしまう。それを防ぐためには，pass に生成回数 T を記載し，T が $\mathcal{C}$ 内の当該アカウントに登録されるようにし，引き出

[*6] info を $\mathcal{C}$ に知られないようにするためには，登録手続きにおいて，$\mathcal{U}_{\mathrm{M}} \Leftrightarrow \mathcal{S}$ 間で鍵を共有するなどの処理を行う必要があり，実際，そうすることにより，$\mathcal{S}$ は $\mathcal{U}$ に種々の情報を通知することができるが，本論文ではそれに関する記述は省略する。

[*7] 前述の鍵 K を算出する関数および関数 h は，登録されている各ユーザがそれぞれの携帯電話機器に導入するアプリに共通である。また，それらの関数は $\mathcal{D}$ や $\mathcal{C}$，$\mathcal{S}$ を含め，外部には秘匿であることが望ましい。

[*8] 鍵 K およびタグ t は，uInfo のみではなく，$\mathcal{U}$ が一時的に定めた保管パスワード dPw を用いて算出されてもよい。この場合，保管データの安全性は高まるが，dPw を忘れるとバックアップから pList を復元できなくなる。

し手続きが行われる度に，pList に含まれる各 pass を再発行するようにすればよい。そうすることにより，引き出し手続きが行われると T がカウントアップされ，たとえ旧機器に pass が残っていたとしても，それは無効化されることになる。

5. 考察

本章では，まず，第 4 章で述べた機種変更手法が，他のワンタイムパスワード認証システムでも適用できるか考え，その後，第 4 章に登場する uInfo の設定方法例について述べる。

5.1 他手法との比較

比較対象は，OTP 取得のために「機器」を用いる方式なので，ハードトークン方式およびソフトトークン方式とする [*9]。

ハードトークンは一般に通信機能を持たない。また，サービスシステムにおいて適切に設定され，各ユーザに個別配付されるものであるため，紛失等により機器を変更しなければならない場合は，例えば [6] のように，サービスシステムに相応の認証を経た安全な方法で処理してもらう必要がある。

ソフトトークン方式の場合，一般に普及しているシステムにおいては，OTP を得る機器はユーザの所有物であることが多い。その機器上で動作する専用アプリはサービスシステムが配布するものであり，ユーザはそれを自身用に設定するのである。機器におけるアプリ領域内にユーザ固有の情報が保存されるので，これを前章における pList のように扱い機器変更することが可能である。とはいえ，実際のソフトトークン方式においては，[5] のように再登録するものが多い。つまり，ユーザが利用するサービスシステム数が多くなると，機種変更はそのシステム数に応じた作業量を発生させる。

本システムの場合，ユーザが利用する各サービスシステムが同一の発行センターに接続されているのであれば，機種変更に伴うユーザの作業量はサービスシステム数に依存しない。また，同一の発行センターでない場合も，[2] の方法等により，単一の発行センターの場合と同じ作業量で機種変更が可能となる。

5.2 uInfo の設定方法

uInfo は，第 4.1 節の (2) および (3) を満たす情報にする必要がある。携帯電話機器の場合には，その候補として電話番号などが考えられる。携帯電話機器上で動作するアプリが，その機器に割り当てられている電話番号を直接取得できない場合は，アプリの初回起動時に電話帳から自身の番号を選択させたり，その番号に SMS を送信したりすることにより，所有者確認ができる。その他にも，初回起動時に契約者 ID(具体的な例としては，Gmail アドレスや Apple ID) に対して認証させることも考えられる。

[*9] マトリクス方式は機器を用いないので，機器変更そのものが起こりえない。なお，盤面上のパターンは記憶情報であるため，その変更は，(固定) パスワードの変更と同様の手法を取ることができると考えられる。乱数表は，サービスシステムから個別配付されるものであり，ユーザはそれに対して何の処理も行わない。したがって，乱数表を紛失した，または，その内容を他者に知られたなどの理由で，乱数表を更新する場合は，サービスシステムに相応の認証を経た安全な方法で再発行依頼する必要がある。

6. まとめ

本論文では，我々の研究グループが 2011 年より発表している 2 要素 3 者間 OTP 認証システムにおいて，保管システムを設置することにより，ユーザが OTP 発行依頼に使用する携帯電話機器を容易に変更できる方法を示した。保管システムは，秘密なパラメータを用いた特別な計算をするわけではないので，必ずしもネットワーク上のクラウドシステム等である必要はなく，ユーザの携帯電話機器とローカル接続されている PC やストレージなどでもよい。また，この方法は，ソフトトークンによる OTP 認証システムにも適用可能であると考えられる。

謝辞 本研究の一部は JSPS 科研費 15K00181 の助成を受けたものです。

参考文献

[1] IPA：オンライン本人認証方式の実態調査報告書, 2014. https://www.ipa.go.jp/security/fy26/reports/ninsho/
[2] 糸井, 多田：“共通 1-day パスワード認証システム”, 2015 年暗号と情報セキュリティシンポジウム (SCIS2015), 2C1-1, 2015.
[3] 垣野内, 木下, 多田, 糸井, 山岸：“発行センターを介したワンタイムパスワード認証システムの実装”, コンピュータセキュリティシンポジウム (CSS)2011, 3C4-2, 2011.
[4] 垣野内, 木下, 多田, 糸井, 山岸：“プライバシーを考慮したワンタイムパスワード認証システムの実装”, 2012 年暗号と情報セキュリティシンポジウム (SCIS2012), 1E2-5, 2012.
[5] 京都銀行：「ワンタイムパスワード」サービス. http://www.kyotobank.co.jp/directb/onetimepass.html
[6] 三菱東京 UFJ 銀行：ワンタイムパスワード. http://direct.bk.mufg.jp/secure/otp/info.html
[7] 佐々木, 多田：“発行センターを介したワンタイムパスワード認証システムの安全性に関する考察”, 2014 年暗号と情報セキュリティシンポジウム (SCIS2014), 3B1-4, 2014.