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

垣野内 将貴 [1)]　木下 誠 [2),3)]　多田 充 [2)]　糸井 正幸 [4)]　山岸 智夫 [4)]

[1)] 千葉大学大学院 理学研究科 / [2)] 千葉大学 総合メディア基盤センター
〒 263-8522 千葉市稲毛区弥生町 1-33.
{kakip.mstk, m.kinoshita}@chiba-u.jp, m.tada@faculty.chiba-u.jp

[3)] 外務省, 〒 100-8919 千代田区霞が関 2-2-1.

[4)](株) セフティーアングル, 〒 272-0826 千葉県市川市真間 5-18-21.
m.itoi@safetyangle.co.jp, tomoo.yamagishi@wd5.so-net.ne.jp

あらまし　ネットワーク上での認証にはパスワードを用いることが多いが，固定パスワードを用いた場合，キーロガー，スパイウェア，ハッキングなどにより，なりすましが行われる危険があるので，より安全に認証を行うためにはワンタイムパスワードの活用が有用である．本論文では，2 メディア，2 要素のダブル照合システムによるワンタイムパスワード認証システムの具体的構成法および我々の実装方針を紹介する．

Implementation of a one-time password authentication system

Masataka Kakinouchi[1)]　Makoto Kinoshita[2),3)]　Mitsuru Tada[2)]
Masayuki Itoi[4)]　Tomoo Yamagishi[4)]

[1)]Graduate School of Science / [2)]Institute of Media and Information Technology,
Chiba University, JAPAN.
Yayoicho 1-33, Inage-ku, Chiba 263-8522, JAPAN.

[3)]Ministry of Foreign Affairs of Japan.
Kasumigaseki 2-2-1, Chiyoda-ku, Tokyo 100-8919, JAPAN.

[4)]Safety Angle Corporation. Mama 5-18-21, Ichikawa, Chiba 272-0826, JAPAN.

Abstract　Passwords are usually adopted for authentication in an open network. However, since for fixed passwords, it is possible to impersonate with key-logger softwares, spywares or computer hacking, therefore we can say that one-time passwords are useful for more secure authentication. In this paper, we introduce how to concretely construct a 2-media, 2-factor and dual-collating authentication system, and also our policy for its implimentation.

1　背景

今日，認証はネットワーク上だけに限らず様々な環境において必要とされ，ネットバンキングやショッピング，オークションといった使い方から，企業内や官公庁，地方自治体などでの個人認証まで，幅広い目的で使用されており [9]，その安全性や利便性の確保/向上は重要な課題となっている．

認証方式には公開鍵暗号系を用いる方式と，共通鍵暗号方式や一方向性ハッシュ関数を用いる方法がある．公開鍵暗号系を用いる方式は安全性が高く電子署名など幅広い分野に応用できるが，比較的多くの計算量が必要という問題点がある．IC カードや携帯電話など計算能力の低い端末での利用を想定した場合，公開鍵暗号系に比べて計算量が格段に少ない共通鍵暗号方式や一方向性ハッシュ関数を用いた

認証方式が適していると思われる．共通鍵暗号方式や一方向性ハッシュ関数を用いた一般的な認証方式においては，固定されたパスワードを用いることが多い [9].

しかしながら，同一の固定パスワードを使用し続けると，攻撃者によってパスワードを特定される危険がある．例を挙げると，キーロガー，スパイウェア，ハッキングなどによりその固定パスワードを盗難されると，以降他者がなりすますことができてしまう．

これらの問題を解決する方法の 1 つとして，各認証セッション毎にパスワードを変化させるワンタイムパスワード (OTP) を使用することが考えられる．他にも，固定パスワードよりセキュリティレベルの高い認証手段として，生体認証などが挙げられるが，OTP を用いた認証は，被認証者 (証明者)，認証者 (検証者)，相互にとって運用が容易であることから実用上注目されていて，最近では多様な認証の応用が期待されることから様々な攻撃に耐性を持つように改良されている．

1.1 OTP 認証

OTP は，認証のために一度しか使えない「使い捨てパスワード」のことである．今日までに，OTP による認証を実現する様々な手段が提案されており，時刻同期方式，チャレンジ・レスポンス方式，数学的アルゴリズムを用いた方式および SMS(Short Message Service) を利用したものが挙げられる．主に採用されている手段は，時刻同期をした物理的なハードトークンを用意し，それを各ユーザに配布し，同期された時刻で OTP を生成するものである．

しかし，ハードトークンを用いた OTP 認証を利用する場合，そのトークンの管理に高いコストがかかってしまう．製品面でもシェアの面でも多く使われている RSA SecurID[1] に対して，2011 年 3 月，その製品を生産している EMC のシステムがクラッキングを受けて，その結果盗まれた技術情報が実際に悪用されたことを受け，EMC セキュリティ部門 RSA は，2011 年 6 月，4000 万台のトークンに対して交換プログラムを用意する事態となった [2, ?]. また，時刻同期によらない方式を採用した場合でも，カウンターなどの再同期のための手続きにコストが大きくかかってしまうという問題が残る．

また，ハードトークンを用意せず，携帯電話用アプリケーションにハードトークンと同様の機能を持たせたり，携帯電話の Web 機能や音声電話機能を利用して OTP を取得する方法もある．

新たにハードトークンを用意するのではなく，携帯電話に同様の機能を持たせるのは，「盗難/紛失を早期に気づく可能性が高い」という利点があるからである．

1.2 提案する方式

本論文では，[7, 8] に基づいた，携帯電話の通信キャリア/個体識別情報を認証に利用できる「2 メディア，2 要素のダブル照合」システムの具体的構成法を提案し，実際に構築したシステムの実装方針を紹介する．利用者を異なるメディアである携帯電話等の情報と関連づけることにより，利用者の所有物/環境による個人認証を実現できることが大きなメリットである．また，OTP の発行センターとサービスシステムを分けることにより，サービス側の管理の負担を減らすことが可能である，ハードトークンを用いた方法とは違い OTP 生成のアルゴリズムの容易な修正/変更が可能である，および，発行センターとサービスシステムのどちらか一方だけを攻撃してもなりすますことが不可能である，などの利点が得られる．

本システムにおいては，1 人のユーザに対して，そのユーザに関連づけできる ID 情報が 3 つ存在する．それぞれを `ID1`, `ID2`, `mID` とした場合，`ID1` はユーザと発行センターだけが知っておりサービスシステムには知らされず，`ID2` はユーザとサービスシステムだけが知っており発行センターには知らされず，`mID` は発行センターとサービスシステムだけが知っておりユーザには知らされない．即ち，3 つの ID 情報に対して，ユーザ，サービスシステムおよび発行センターが三竦みの状態を形成している．

また，携帯電話は，盗難/紛失時に気づく可能性が高く，仮に不正取得者が悪用しようとしても認証 ID(`ID1` および `ID2`) がわからないために OTP を不正取得できない点も，メリットとして挙げられる．

2 提案システムにおける諸手順

本章では，我々が提案・実装する認証システムにおける諸手順を述べる．

2.1 表記法

サービスシステム $\mathcal{S}$ の利用を希望するユーザを $\mathcal{U}$ とし，$\mathcal{U}$ が $\mathcal{S}$ へアクセスするための OTP の発行

を依頼する発行システムを$\mathcal{I}$とする．

$\mathsf{Sign}_{\mathtt{key1}}(\mathtt{info})$を情報 info に鍵 key1 を適用して生成した電子署名とし，Verify_*をその検証関数とする．つまり，検証鍵を key2 とするとき，$(\mathtt{info}, \mathtt{sig})$が正当な (情報，署名値) のペアであれば，$\mathsf{Verify}_{\mathtt{key2}}(\mathtt{info}, \mathtt{sig})$の値は true となり，そうでなければ false となる．また，$\mathsf{hash}(\mathtt{info})$を情報 info に対するハッシュ関数値とする．

2.2 事前設定

$\mathcal{S}$はユーザの OTP を$\mathcal{I}$に発行してもらうようにする際，自身のシステム情報$\mathtt{ID}_{\mathcal{S}}$を$\mathcal{I}$に登録し，そのとき$\mathcal{I}$との間で鍵 key0 を共有することとする．また，$\mathcal{I}$は独自の秘密鍵 key を持つものとする．

2.3 ユーザ登録手続き

$\mathcal{U}$が$\mathcal{S}$の利用を申し込んだとき，$\mathcal{U}, \mathcal{S}, \mathcal{I}$の 3 者は，その$\mathcal{U}$に関連付ける 3 つの情報 $\mathtt{ID1}, \mathtt{ID2}, \mathtt{mID}$ に対して，$\mathcal{U}$が $(\mathtt{ID1}, \mathtt{ID2})$，$\mathcal{S}$が $(\mathtt{ID2}, \mathtt{mID})$，$\mathcal{I}$が $(\mathtt{ID1}, \mathtt{mID})$ を所有するような三竦みの状態を作る．具体的には以下の操作を実行する．

(r1) $\mathcal{U}$は，ユーザ情報，サービス利用のためのアカウント文字列 ID2 ならびにチケット有効化 (ステップ (r3)) のためのパスワード pw を$\mathcal{S}$に送る．

(r2) $\mathcal{S}$は，$\mathcal{U}$の利用申し込みを受理する場合，$\mathcal{I}$にチケットの発行を依頼し pw を送る．

(r3) $\mathcal{I}$は，(r2) のリクエストが$\mathcal{S}$によるものであることを確認した上で，シリアル番号 n およびそのユーザに対する識別子 mID を定め，メッセージ m を任意に選び，その番号 n に対するチケット $t = \mathsf{Sign}_{\mathtt{key}}(m||\mathtt{pw})$ を生成し，そのチケットの有効期限 T を定める．$(\mathtt{mID}, n, t, T)$ を$\mathcal{S}$に送る．

(r4) $\mathcal{S}$は$\mathcal{U}$に (n, t, T) を送る．

(r5) $\mathcal{U}$は OTP 発行に使用するアカウント文字列 ID1 を定め，$(\mathtt{ID1}, n, t, \mathtt{pw})$ と第 1 認証のポリシー pcy を$\mathcal{I}$に送る[1]．

(r6) $\mathcal{I}$は，$\mathcal{I}$のリクエスト時刻がTを過ぎていないことを確認した上で，nに対応するメッセージ m を用いて $\mathsf{Verify}_{\mathtt{key}}(m||\mathtt{pw}, t)$ を計算し，true であれば ID1 と mID を関連付け，ポリシー pcy とともに ID1 をデータベースに登録する．さらに pcy に伴い$\mathcal{U}$の認証に必要となる情報 info があればそれを$\mathcal{U}$に送らせ，逆に OTP 発行依頼する際に$\mathcal{U}$が必要とすることになる情報 cer があればそれを$\mathcal{U}$に送る[2]．

2.4 ユーザ認証手順

$\mathcal{U}$が$\mathcal{S}$のサービスを利用するとき，以下の 2 種類の認証をパスする必要があり，ここではそれらを第 1 認証，第 2 認証と呼ぶことにする．

2.4.1 第 1 認証手順

$\mathcal{U}$は$\mathcal{I}$にによる第 1 認証をパスすることにより，$\mathcal{S}$を利用するための OTP を発行してもらう．

(i1) $\mathcal{U}$は，利用するサービスシステム情報$\mathtt{ID}_{\mathcal{S}}$，ID1 および pcy に伴い必要となる情報 info を$\mathcal{I}$に送る[3]．

(i2) $\mathcal{I}$は，$\mathcal{U}$が$\mathcal{S}$を利用できること，info の正当性，および$\mathcal{U}$が OTP の発行条件を満たしていることを確認した上で，$\mathsf{OTP}(\mathtt{otp})$ を生成する．さらに，ID1 に対応する mID を選び，$(\mathtt{mID}, \mathtt{otp})$ にその署名 σ $(= \mathsf{Sign}_{\mathtt{key0}}(\mathtt{mID}, \mathtt{otp}))$ を添えて$\mathcal{S}$に送り，$\mathcal{U}$には otp のみを送る．

(i3) $\mathcal{S}$は，$\mathcal{U}$と共有している鍵 key0 を用いて，$(\mathtt{mID}, \mathtt{otp})$ の正当性を確認した上で，mID に対応する ID2 に対してパスワード otp を設定し，その有効期限 T_2 を定める[4]．

2.4.2 第 2 認証手順

第 2 認証をパスすることにより，$\mathcal{U}$は実際に$\mathcal{S}$のサービスを利用することができるようになる．

(i4) $\mathcal{U}$は (i2) により$\mathcal{I}$から送られた otp を ID2 とともに$\mathcal{S}$に送る．

[1] ここでのポリシーは，$\mathcal{I}$に OTP の発行を依頼するときの$\mathcal{U}$の (アクセス) 状態や，$\mathcal{U}$が提示する情報の種類を意味する．具体的な例は第 3.1 節で述べられている．

[2] cer については第 3.1 節で言及する．

[3] info は，(r6) で cer が発行されていれば，その cer を含んだ情報となる．

[4] $\mathcal{U}$は$\mathcal{S}$を利用するにあたり，$\mathcal{I}$から発行された OTP の有効期間を承知しているものとする．

(i5) $\mathcal{S}$ は $\mathcal{U}$ のアクセス時刻が T_2 を過ぎていないことを確認した上で，otp の正当性を検証し，その結果に応じてサービスを提供する．

2.5 ユーザ再登録手順

$\mathcal{U}$ が，何らかの事情により，$\mathcal{S}$ に対する OTP 発行ポリシーを pcy1 から pcy2 に変更したい場合は，以下の手順を実行する．

(u1) $\mathcal{U}$ は，pcy1 に従い ID1 を $\mathcal{I}$ に送り，$\mathcal{S}$ に対する OTP 発行ポリシーを pcy2 に変更する旨，および pcy2 に伴い必要となる情報 info を $\mathcal{I}$ に送る．

(u2) $\mathcal{I}$ は，$\mathcal{U}$ が $\mathcal{S}$ を利用できること，および $\mathcal{U}$ が pcy1 に従ってアクセスしているかを確認した上で，データベース内の ID1 のポリシーを pcy2 に変更する．さらに pcy2 に伴い $\mathcal{U}$ が必要とすることになる情報があれば，その情報 cer2 を $\mathcal{U}$ に送る．

3 実装の概略

ここでは，我々が前章に基づいて実装した認証システムの動作概略を述べる[5]．システムの構成は，パスワード発行センター $\mathcal{I}$，サービスシステム $\mathcal{S}$ およびユーザ $\mathcal{U}$ のみであり，それら 3 者間の通信はすべて https(443) で行うものである．

$\mathcal{I}$ および $\mathcal{S}$ の OS および Web サービスのためのソフトウェアは「Debian Linux 6.0.x」および「Apache HTTP Server 2.x.x」を使用し，$\mathcal{S}$ のサービスのための認証方式は，簡単のため Basic 認証とした．

ユーザ登録の際に発行するチケット t は $\mathcal{I}$ による電子署名であるが，生成した署名を検証するのは $\mathcal{I}$ 自身である．また，$\mathcal{I}$ が OTP を発行し $\mathcal{S}$ に送るときに添える電子署名を検証するのは共有鍵をもつ $\mathcal{S}$ なので，ここで用いる電子署名方式は公開鍵暗号系に基づくものではなく，MAC(Message Authentication Code) を用いることにした．MAC 作成に用いるハッシュ関数は SHA256 とした．つまり鍵 key を用いてメッセージ m に対する MAC 値 σ は，最もシンプルな形と思われる $\sigma = \mathsf{sha256}(m||\mathsf{key})$ として計算した．key および key0 の値は 128 ビットの値としている．

[5] スペースの都合により，ここでは極めて簡潔に述べるに留めておきます．詳細については，本シンポジウムでデモンストレーション展示する予定なので，そちらまでご足労ください．

3.1 OTP 発行ポリシー

$\mathcal{U}$ が $\mathcal{I}$ に OTP の発行を依頼するとき，$\mathcal{I}$ はその依頼主が確かに $\mathcal{U}$ であることを認証しなければならない．発行依頼の際，$\mathcal{U}$ は少なくとも ID1 を送信しなければならないので，ID1 を秘匿にしている限り，他者になりすまされることはない．

ここでは，さらにセキュリティのレベルを上げるために，$\mathcal{U}$ が自分自身を証明するための手段を，後述の複数の項目から選択できるようにした．ここでは，$\mathcal{U}$ による OTP 発行依頼が，(i) 携帯電話からの通信で送られてくる場合と，(ii)PC からの通信で送られてくる場合，によって場合分けし，その (i) と (ii) は，$\mathcal{U}$ が使用している IP アドレスに基づき判別することにした．つまり，$\mathcal{U}$ の IP アドレスが携帯電話キャリア 3 社が公開している使用帯域 [5, 4, 6] の範囲内であれば携帯電話からの通信であると判定するものである．

携帯電話の場合

携帯電話の場合，機器の個体識別番号を使うことにより，$\mathcal{U}$ の登録時と OTP 発行依頼時に使用された機器が同一であるかを確認できる．しかし個体識別番号を送信できるようになっていない携帯電話機器の場合，この方法は使用できない．したがってこのような場合は，PC からの場合と同様にユーザ登録 (第 2.3 節) の際，$\mathcal{I}$ が電子署名 cer を発行し，$\mathcal{U}$ が OTP の発行依頼をするときに cer を提示させるようにする．cer の盗難/漏洩による他者のなりすましを防ぐために，携帯電話のキャリア，機種を限定させることにより，個体識別番号を用いなくても，「ID1, ID2, cer が知られる」,「使用する携帯キャリア・機種が知られる」ことがない限り，他者が $\mathcal{U}$ になりすまして $\mathcal{S}$ のサービスを使用することはできない．

以上のことから，OTP 発行ポリシーを以下の 4 通りとした．

(p1) cer のみの検証

(p2) cer および携帯キャリアの検証

(p3) cer および携帯キャリア・機種の検証

(p4) 携帯キャリアおよび電話機器個体識別番号の検証

PC からの場合

PC(携帯電話の IP 帯域外からのアクセス) の場合，HTTP ヘッダの偽装が容易なので，主にその PC が利用している IP アドレスに基づきポリシーを定めた．

(p5) `cer` のみの検証[6]

(p6) `cer` およびネットワークアドレスの検証

(p7) `cer` および IP アドレスの検証

3.2 事前設定

$\mathcal{I}$ は発行した OTP を利用できるシステムのリスト `S-List` を持つ．`S-List` にはシステム $\mathcal{S}$ の固有情報 $\mathtt{ID}_{\mathcal{S}}$ (システム名，システムの IP アドレス/FQDN，チケット発行を依頼できる IP アドレス $\mathtt{ip}_{\mathcal{S}}$，など) および $\mathcal{S}$ との共有鍵 `key0` が記載されている[7]．また $\mathcal{I}$ は，$\mathcal{I}$ を利用できるユーザのリスト `U-List` を持ち，そのリストには，シリアル番号，チケット発行時刻，`mID`，利用するシステム $\mathcal{S}$ に関する情報，`pw`，OTP 発行ポリシーおよび第 1 認証に必要な登録情報などが含まれる．

3.3 ユーザ登録手順

第 2.3 節で述べた手続きのうち，(r1) については，$\mathcal{S}$ が (r2) においてその申し込みを受理する/しないを判断するため，少なくとも $\mathcal{U}$ の確認をする必要があり，この申し込み手続きをオフライン処理にするということで，今回の実装には含めていない．また (r4) についても，$\mathcal{S}$ がチケットを $\mathcal{U}$ に知らせるだけであり，主に考えられる手段は，電子メールによるものや，オフラインによるものなので，この手続きについても今回の実装には含めていない．よって，ここでは (r1), (r4) 以外の手続きについて．その実装方針を述べる．

(r2) チケット発行依頼

$\mathcal{I}$ は，`S-List` に登録してある各 $\mathcal{S}$ に対する $\mathtt{ip}_{\mathcal{S}}$ からのみアクセスできる，`pw` の入力フォームを用意し，$\mathcal{U}$ が $\mathcal{S}$ の利用申し込みの際に記入した `pw` を送信させる．また，IP アドレスを偽装することによるチケット発行の不正依頼をされないよう，フォームに自身の署名を含ませておく．

[6]実質上，(p1) と (p5) は同一のポリシーとなる．

[7]`ip` は複数のアドレスのリストや特定のネットワークアドレスでもよい．

(r3) チケット発行

$\mathcal{I}$ は (r2) によるリクエスト内容に含まれる自身の署名を検証した上で，`pw` を受け取る．その後，シリアル番号 n，一意的な `mID` を生成し，対応するシステム $\mathcal{S}$ の情報，その有効期限 T，任意のメッセージ $m(\mathcal{S}, T, \mathtt{pw})$ を定め，`U-List` に登録する．また，チケット $t = \mathsf{Sign}_{\mathsf{key}}(m)$ を生成し，t を使用するための URL:

$$\mathtt{https://} \cdots \mathtt{?ticket=}n\_t$$

を `mID` とともに $\mathcal{S}$ の Web インターフェースに表示する[8]．

(r5), (r6) ユーザ登録

$\mathcal{I}$ はユーザ登録画面を用意し，$\mathcal{U}$ のアクセスの際に受け取る n の値から t の検証に必要な情報を取り出し，t の正当性を検証する．t が正しいならば，$\mathcal{I}$ は，$\mathcal{U}$ のアクセス媒体 (携帯電話/PC) に応じて，可能な OTP 発行ポリシー `pcy` を $\mathcal{U}$ の Web インターフェースに表示し，$\mathcal{U}$ にその中の 1 つを選択させ，さらに `pcy` に伴い必要となる情報 `info`(IP アドレスなど) を送らせる．$\mathcal{U}$ が選んだポリシーが (p4) 以外の場合は，OTP 発行の際に $\mathcal{U}$ を認証するためのメッセージ m およびその署名値 α を計算し，OTP 発行依頼の際にアクセスする URL:

$$\mathtt{https://} \cdots \mathtt{?cer=}m\_\alpha$$

を $\mathcal{U}$ の Web インターフェースに表示する[9]．

3.4 ユーザ認証手順

3.4.1 第 1 認証手順

第 1 認証においては，$\mathcal{U}$ は OTP 発行ポリシーに基づいた機器 (携帯電話/PC) で $\mathcal{I}$ にアクセスする．

(i1), (i2) OTP の発行・通知

$\mathcal{U}$ は OTP 発行 URL にアクセスする．$\mathcal{I}$ は `cer` の値がある場合はその値の正当性を検証した上で，`ID1` を入力させるフォームを表示する．`cer` の値がない (ポリシーが (p4) であるユーザの) 場合は，携帯電話の個体識別情報を送信させる画面を経由させ，識別情報を送信させた上で，`ID1` を入力させるフォームを表示し，送信させる．

[8]実装上は，携帯電話でのユーザ登録のために上記 URL に対応する QR コードも表示するようにしている．

[9]発行ポリシーが (p4) の場合は，同様の URL で GET 送信する値がないものになる．

$\mathcal{I}$ は $\mathcal{U}$ から送られた ID1 に対応する mID を選び，OTP(otp) を生成する．さらに $\sigma = \mathsf{Sign}_{\mathtt{key0}}(\mathtt{mID}, \mathtt{otp})$ を計算し，

https:// ⋯ ?request=mID_otp_α

にアクセスすることにより $\mathcal{S}$ にパスワード登録をリクエストする．また，$\mathcal{U}$ の Web インターフェースに otp を表示する．

(i3) OTP の登録

$\mathcal{S}$ は受け取った $(\mathtt{mID}, \mathtt{otp}, \sigma)$ により $(\mathtt{mID}, \mathtt{otp})$ の正当性を検証した上で，mID に対応する ID2 にパスワード otp とその有効期限 T_2 をセットする．

3.4.2 第 2 認証手順

(i4), (i5) 期限付き Basic 認証

通常の Basic 認証と同様であるが，パスワード照合の際，$\mathcal{U}$ のアクセス時刻が T_2 を過ぎていないことも検証する．

3.5 ユーザ再登録手順

ユーザが OTP 発行依頼に使用する携帯電話のキャリア/機種を変更した場合，ユーザが PC をネットワーク接続させるときの IP 帯域を変更する場合，OTP 発行ポリシーを変更せざるを得ないことがある．第 1 認証に使う情報が ID1 および cer だけとなるポリシー (p1) または (p5) であれば，そのような変更は必要ないが，それでも発行依頼の際にアクセスする URL が漏洩した場合，たとえ ID1 が漏洩していないとはいえ，その URL を変更したいという要望がでることは明白である．したがって，我々はそのような場合，ユーザが第 1 認証のポリシーを pcy1 から pcy2 に変更できるツール (Web フォーム) も用意した．IP アドレス詐称による不正リクエストを防ぐため，フォームには $\mathcal{I}$ による署名を含ませてある．

(u1) ポリシー変更依頼

$\mathcal{U}$ は pcy1 に従ってフォームにアクセスする．$\mathcal{I}$ は pcy1 の状態から変更可能なポリシーのリストを表示する[10]．$\mathcal{U}$ は表示されたポリシーリストから 1 つ選択肢，ID1 とともに $\mathcal{I}$ に送信する．$\mathcal{I}$ は pcy2 に伴い必要となる情報 — 携帯電話の場合はキャリア，機種，個体識別番号，PC の場合は IP アドレス，ネットワークアドレス — を取得する．

(u2) ポリシーの修正登録

$\mathcal{I}$ は (u1) で受け取った ID1 からそれに対応する mID を探索し，U-List 内の mID について，ID1 と同時に受け取った pcy2 およびそのポリシーに伴い必要となる情報に書き換える．

[10] IP アドレスまたはネットワークアドレスが限定されている PC から発行依頼するポリシーから携帯電話の機種名または個体識別番号の検証を利用するポリシーへの変更は，その場でその機種名や識別番号を正確に送ることが困難であるため，許可しないようにした．つまり，PC でアクセスした場合は (p3) および (p4) への変更はできないようにしてある．

4 まとめと今後の課題

本論文では，携帯電話の通信キャリア/個体識別情報を認証に利用できる「2 メディア，2 要素のダブル照合」システムの具体的構成法を提案し，我々が実際に構築したシステムの実装方針を紹介した．

今回の実装では，サービスシステムで利用する認証方式としてシンプルな Basic 認証を採用したが，比較的大規模なサービスシステムで利用されることの多い LDAP や Active Directory などの認証方式への適用，および，複数のサービスシステムに対応できる発行センターの実装を今後の課題としたい．

参考文献

[1] EMC ジャパン株式会社:「RSA SecurID」, http://japan.rsa.com/node.aspx?id=1156

[2] EMC Corporation: Open Letter to RSA Customers, http://www.rsa.com/node.aspx?id=3891

[3] Help Net Security: RSA admits SecurID tokens have been compromised, http://www.net-security.org/secworld.php?id=11122

[4] KDDI au:「技術情報 > IP アドレス帯域」, http://www.au.kddi.com/ezfactory/tec/spec/ezsava_ip.html

[5] NTT docomo:「作ろう i モードコンテンツ：i モードセンタの各種情報 | サービス・機能 | NTT ドコモ」, http://www.nttdocomo.co.jp/service/imode/make/content/ip/#ip

[6] Softbank:「WEB & NETWORK IP アドレス」, http://creation.mb.softbank.jp/web/web_ip.html

[7] 特許公報:「個人認証方法及びシステム」, 特許第 3678417 号, 特開 2003-323408, 特願 2002-126933.

[8] 特許公報:「個人認証方法及び個人認証システム」, 特許第 4583746 号, 特開 2005-128847, 特願 2003-364708.

[9] 鵜尾 健司, 白石 善明, 森井 昌克: “Stolen-Verifier attack に耐性のあるワンタイムパスワード方式の評価と提案”, コンピュータセキュリティシンポジウム 2006 (CSS2006) 予稿集, 8A-4, 2006.