ShownPass: 表示されたパスワードを用いたアクセス制御

綾塚 祐二　河野 通宗　暦本 純一

ソニーコンピュータサイエンス研究所
インタラクションラボラトリー

要旨

アクセス制御は、いつでもどこでもネットワークが使えるような環境では重要な課題である。既存のアクセス制御手法は、主として予め登録された端末やユーザを認証するという考えに基づいていたため、その場にあるリソースの利用許可を訪問者に対して与えるといったようなことは手間が掛かる。我々は、リソースのそばに時刻とともに変化するパスワードを表示し、リソースの近くにいるユーザのみがそれを利用できるようにするという手法を提案する。この手法は特殊なハードウェアなどは必要とせずに、パスワードが見える場所に居るかどうかという物理的な制約に基づいたアクセス制御を簡単に行うことができる。

ShownPass: an Access Control with a Displayed Password

Yuji AYATSUKA, Michimune KOHNO, Jun REKIMOTO

Interaction Laboratory,
Sony Computer Science Laboratories

Abstract

Access control is one of the most important issue with ubiquitous networking environment. Traditional access control methods are mainly considering authentication of registered user or device. Therefore, it is troublesome to allow a visitor to use a networked resource in an office, without accessiblity to other resources. We propose a new access control method that employs alternative passwords displayed beside a resource. A user who can see the password is allowed to use the resource. This method can be implemented without any special equipment like a sensor.

1 はじめに

ネットワーク環境が整備されるにつれ、日常の様々な場所からのネットワークアクセスが極めて容易になってきた。IEEE802.11 などの規格の無線によるネットワーク接続も普及し、オフィス環境でも各人がノート型 PC を持ち歩き、会議室などからもネットワーク上のデータを参照することなどが当然のように行われるようになってきている。そのような環境に馴れたユーザが、訪問先などでも同じようにネットワークにアクセスしたいという要望を持つのは自然な成り行きである。

しかし、ネットワークが普及した環境ではセキュリティやアクセス制御のことも充分に考慮しなければならない (文献 [5] でユビキタスコンピューティングや仮想現実感を用いたインタフェースに関する様々なセキュリティの問題が分析されている)。例えば、オフィスでは、訪問者がそこからインターネットへアクセスすることは構わないが、オフィス内の他の計算機などネットワーク上のリソースに自由にアクセスできる状態にしてしまうことは好ましくないであろう。携帯電話などを利用したネットワークアクセスなどの手段もあるため、訪問者には LAN 内へのネットワークアクセスを提供しないという選択をすることは、より安全性を求めるためにもやむを得ないことが多い。

その結果、例えば、ユーザが訪問先で自分の端末にある資料を印刷したいと思ったときにも、そこにあるプリンタを直接使うことはできないという事態が生じる。そのため、ユーザはメモリカードなどの媒体を使ってデータを訪問先の人に渡し、印刷してもらうことになる。

このような状況は、訪問者に対し、その訪問期間のみ、特定のリソースへのアクセスを許すことのできる簡便な手段があれば改善することができる。ネットワークアドレスなど端末の ID を利用した制限やパスワードによる制限、訪問者用の独立した LAN のセグメントを用意することなどで解決することもできるが、それにはコストが掛かる。前者では、訪問者があるごとに端末の ID やパスワードを登録し、訪問が終了したら削除するということを繰り返さねばならない。また後者では、独立したセグメントを用意し、そこに訪問者用のリソースを用意しなければならない。

我々は、この訪問者のような例の場合、「ユーザがリソースに物理的に接近することが許されている」ことと「そのユーザに (一時的に) そのリソースの利用が許されている」ことがおおよそ一致する場合が多いと考え、特殊なハードウェアを用いることなくそのような利用許可を容易に実現する方法を提案する。具体的には、リソース自体もしくはその近傍に、数分程度の一定時間で自動的に変化するパスワードを表示する部分を設け、そのパスワードを用いることによりそのリソースにアクセスができるようにする。すなわち、物理的にリソースの近くにいれば容易に知ることができるが、そうでなければ知ることのできない情報を基準にしてリソースの利用を許可する。パスワードが一定時間ごとに変化することにより、そのリソースを一度利用したユーザでも、その場を離れた後は利用が不可能になる。

以下の節では、この手法の詳細と特性を論じ、いくつかの実装例を示す。

2 ShownPass

従来、ネットワーク上のリソースに対するアクセス制御は主に「予め登録されたユーザ・端末に対してアクセスを許可する」という方向で考えられてきた。登録は人手を介して行われ、また登録の解除も人手を介して行われる。そのため、「一時的な利用の許可」を頻繁に行うようなことには向いていない。また、アクセスが LAN の内側からか外側からかといった、ネットワーク上でのトポロジーに基づいた制御も行われるが、これは「物理的には近接しているが、まったく別のネットワークに接続している端末」に対応することができない。

「一時的な利用の許可」を与えたい場合とその対象となるリソースを考えると、離れた場所からネットワーク越しにサーバへのログインを許すというようなことよりも、前節で挙げたような、訪問者にプリンタなどの物理的出力デバイスの利用を許すという場合のほうが多いであろう。つまり、リソースの一時的な利用を許可したいユーザは、既に物理的にそのリソースの近くにいることが多いと考えられる。加えて、物理的にリソースに近づくことを許されているユーザには、その使用を許可して構わないことが多いと考えられる。そこで、この物理的な位置関係を利用して、リソースの近くにいるときのみアクセス許可することを考える。

我々は、リソースに対してリクエストを出すユーザが物理的に近くにいることをリソースに保証するための手段として、リソース自体もしくはその近傍にパスワードを表示し、そのパスワードをユーザが端末に入力し、リクエストに添えるという手法を提案する。すなわち、

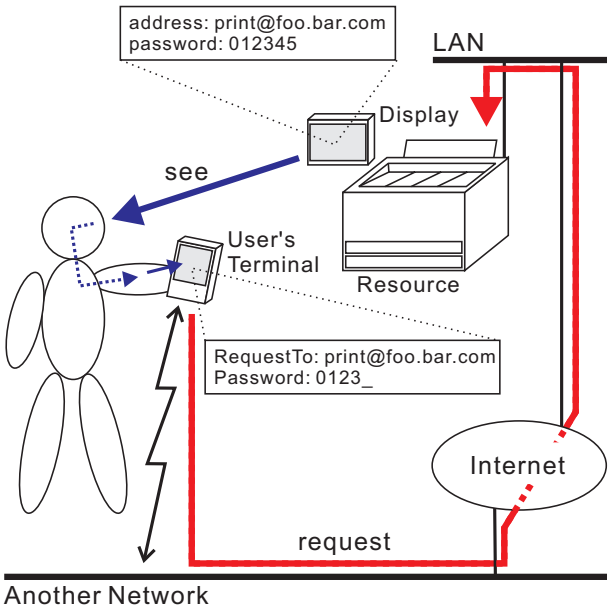

図 1: ShownPass の概念図

パスワードを見ることのできるユーザに対して、リソースの利用を許すという手法である。パスワードは一定時間 (たとえば数分) ごとに変化させ、過去のパスワードは使えないようにすることで、その場を去ったユーザは自動的にそのリソースが使えなくなる。逆に、パスワードを見ることができれば、ユーザは全く別のネットワークからリソースを利用することもできる。

この手法の特徴的な点は、通常ならばなるべく秘諾されるべきパスワードを、オフィス外などからは見えないような場所とはいえおおっぴらに表示しておく、というところである。そこで、この手法を “ShownPass” と呼ぶことにする。リソース自体、もしくはその近傍にパスワードを表示する部分さえあれば、付加的なハードウェアを必要としないということも大きな特徴である。そのため、既存の各種のリソースを既存のさまざまな端末から ShownPass を用いて利用することができる。

図 1 はこの手法の概念を表した模式図である。この図では、ユーザがリソースに対してリクエストを出すためのアドレスもパスワードとともに表示している。

2.1 リソースへのパスワードの提示方法

ユーザがリソースへのリクエストの際にパスワードを添える方法は二通り考えられる。

通信の開始時に伝える: TCP/IP などでリソースに接続した直後に、そのコネクション上でパスワードを伝え認証を行った後、そのリソース向けの通常のプロトコルでデータのやりとりを行う。

リソースへ送られるデータの一部に付加する: 送られるデータに、コメントなどの付加的な情報としてパスワードを付加しておく。

いずれの場合も、リソースはユーザが送付したパスワードと現在表示しているパスワードを比較し、合っていれば[1] その後の処理を続け、違っていればそのリクエストを却下する。パスワードを直接送らず、それを元にしたハッシュ値などを送ることも考えられる。

パスワードは、ユーザが見て端末に入力するので、あまり長い文字列ではないほうが望ましい。数分といった時間間隔で変更されるので、破られにくいように長くする必要もない。数桁の数字、もしくは英数字で充分なことがほとんどであろう。

2.2 適用範囲

ShownPass は訪問者のようなゲストユーザに一時的なリソースの利用許可、特に一回のリクエストのみの許可のようなものを与えるような場合に特に有効である。パスワードの表示をリソースの物理的な実体と関連付けた状態で行うことにより、物理的な空間に応じたアクセス制御、たとえば会議室内のリソースのみ利用を許すといったことが簡単に実現できる。これは、ネットワーク上でのユーザ登録などの管理のみで行おうとすると非常に繁雑な手間を要する。

ShownPass と同時に他のアクセス制御、認証を用いることもできる。そのオフィスに所属するユーザの利便性を考えると、LAN 内からのアクセスや登録された端末からのアクセスはパスワード無しで受け付けるのがよいであろう。また、場合によっては逆に制限を強くして、ユーザや端末の登録と ShownPass で提示されるパスワードの両方を必要とするリソースを置くことも考えられる。

ShownPass は、物理的にリソースの近くへ来ることが可能である時点でセキュリティの問題はある程度クリアされているという前提に基づいているので、それが成り立たない場合には単体で適用すべきではない。たとえばパスワードを見ることの可能な位置にカメラなどが取りつけられていて、それが外部から自由に見られるので

[1] 実用上は、ユーザがパスワードを見てからリクエストがリソースに届くまでの間にパスワードが次のものに変わってしまう場合を考慮して、一つ前のパスワードまでは有効とするのがよいであろう。

図 2: ShownPass を用いたプリンタ

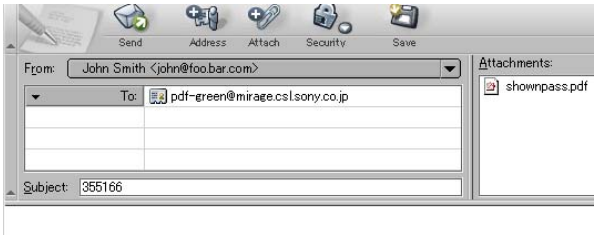

図 3: プリンタへの印刷要求のメイル

あれば、ShownPass は全く意味をなさない。逆に、表示される場所や、リソースへ渡す方法など、パスワードの取り扱いに留意すれば、この手法は比較的高度なセキュリティを要求されるものに対して用いることもできる。その際にはもちろん、求められる安全性に応じて、他の認証や送られるデータの暗号化などの技術を併用すべきである。

3 実装

我々は、ShownPass を用いたシステムとして、電子メイルによりリソースへリクエストを送り、件名として正しいパスワードが記されていればそのリクエストが処理される、というものを作成した。リソースへ送るデータ本体は、本文もしくは添付ファイルとして送る。メイルを用いることにより、既存の端末をそのまま付加的なハードウェアやソフトウェア無しに利用することができる。また携帯電話のような、LAN に直接接続することができない端末からも利用することができる。

ウェブページのフォームと CGI、HTTP を用いたファイルのアップロード [1] などを用いて同様な使い勝手のシステムを構築することもできる。その場合にはパスワードはフォームデータの一部として入力するようにすればよい。サーバへは、URL の引数部分の一部として埋め込むか、ヘッダの一部として渡すことができる。

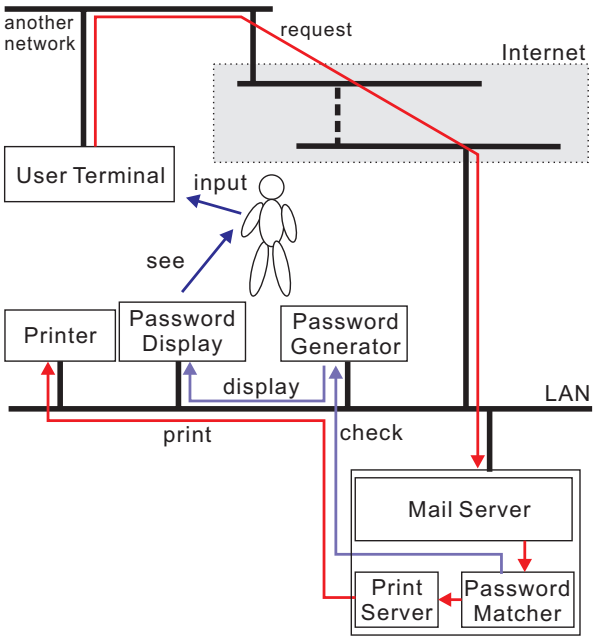

図 4: ShownPass を用いたプリンタの構成図

パスワードの生成、表示やその照合は、リソース自体が行ってもよいし、他の計算機がその部分を受け持ち、認証されたもののみをリソースに送るようにしてもよい。以下の例では、その両方の場合を紹介する。

3.1 プリンタ

図 2 は、ShownPass をプリンタに適用した例である。パスワードの生成や照合は他の計算機が行い、表示はプリンタのそばに置いた PDA (PocketPC) により行っている。この PDA は IEEE802.11b で LAN に接続されており、ウェブブラウザを用いてパスワードを表示している。PDA にはメイルアドレスも表示されており、ユーザはこのアドレスに、PDF (Portable Document Format) ファイルを添付し件名にパスワードを記したメイルを送ることにより (図 3)、このプリンタから印刷を行うことができる。パスワードは 3 分ごとに変わる、6 桁の数字である。LAN 内部からはもちろん通常通りにこのプリンタを使用することができる。

このシステムの具体的な構成は図 4 のようになっている。メイルサーバ (ここでは、PC の Linux 上で動く qmail を利用している) の、メイルが届く毎に設定した

図 5: ShownPass を用いた画像付き掲示板

図 6: カメラ付きの携帯電話から掲示板に投稿する

プログラムを起動する機能を用い、パスワード照合プログラムを起動している。照合プログラムはパスワード管理プログラムと通信して件名に記されているパスワードが有効であるかどうかを照合し、有効であれば添付されたデータをプリンタに印刷するプログラムを起動する。

この例の場合、リクエストが受理されたことはプリンタから印刷が開始されることで容易に知ることができる。一方、パスワードが違い、リクエストが却下されたときはそのままでは何も起こらず、リクエストの到達が遅延しているのと区別がつかない。メイルでエラーメッセージを送り返したり、Web ページを利用する場合は、結果としてエラーを表示するページを返したりすることが当然考えられるが、この実装ではパスワードを表示するディスプレイがエラーも表示する。そうすることにより、宛先のアドレスが間違っていたのではないことが確認できる。また、エラー表示もその場にいないと判らないということは、セキュリティ上も望ましいであろう。

3.2 掲示板

図 5 は我々のオフィスで用いている画像付き掲示板の例である。ユーザはカメラ付き端末で撮った写真やその他の画像ファイルに言葉を添えて投稿することができる。専用の端末からの投稿に加えて、掲示板の前にいるユーザに対しては自身のカメラ付き携帯電話などからの投稿を許している。

この例の場合は掲示板のプログラム自体がパスワードの生成・表示・照合を管理している。通常の状態ではパスワードは表示されておらず、ユーザが物理的なリクエスト (この例では、掲示板を表示している壁に対するノックを検出している) を行ったときに一定時間表示するようにしている。本文に文章を書き、そして添付ファイルとして画像を付け、プリンタの場合と同じように件名にパスワード (この例では 4 桁の数字及び * と # の記号) を記し表示されているアドレスにメイルを送ると、投稿が完了する。図 6 はカメラ付きの携帯電話から投稿しようとしている様子である。

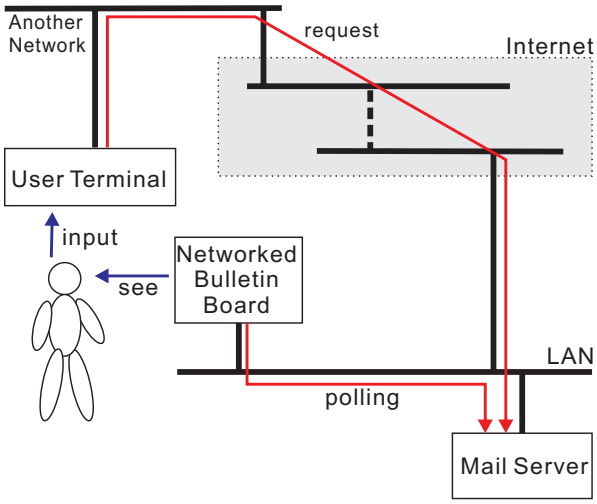

図 7: ShownPass を利用した掲示板の構成図

掲示板のプログラムは PC の Windows の上で動いているアプリケーションで、定期的に (もちろん、パスワードの更新よりは短い間隔で) 表示されているアドレスに対応したメイルのスプールにアクセスしている。メイルが届いていると件名を調べ、有効なパスワードが記されていればその投稿を受理し、それ以外のメイルは削除する。図 7 がシステムの構成図である。この場合はリ

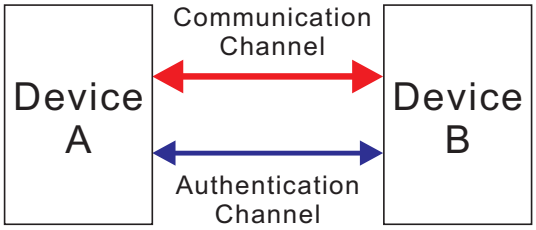

図 8: 通信チャンネルと認証チャネルを用いた通信のモデル: 多くの場合、通信チャネルのほうに広い帯域の通信路を、認証チャネルのほうには何らかの制限の強い通信路を用いる

ソース自身がパスワードの生成・表示・照合を管理しているので構成がシンプルになっている。

4 議論と比較

この節では、ShownPass の特性を考察し、またそのバリエーションや他のシステムなどとの比較を述べる。

4.1 隔離された別の通信路を用いた認証

ShownPass における重要なポイントは、通信のための正しい鍵 (パスワード) を通常の通信路とは隔離された別の通信路を使って渡しているという点にある。その「隔離された別の通信路」が物理的に制約の強い (リソースの近くにいないと使えない) ものであるということを利用してアクセス制御を行っているのである。我々は [8] において「通常の通信用の通信路 (通信チャネル, Communication Channel)」と「認証用の通信路 (認証チャネル, Authentication Channel)」を別々に用意し、認証チャネルには何らかの形で人間が介在することによりセキュリティとユーザビリティを両立させるというモデル (図 8) を提案している。ShownPass は、認証チャネル側に大きく人間が介在し特殊なハードウェアを必要としない、このモデルの一応用例と言える。

ある経路を通じてパスワードを見せる、あるいは聞かせるなどにより、別の経路でのアクセス制御を行うことは、実世界では日常的に行われていることである。たとえば、コンサートなどのチケットの販売で、電話で申し込み、予約が完了すると予約番号が音声で流れ、発券所に行きその予約番号を伝えると発券が行われるというシステムが使われている。また、菓子などの箱の内側にパスワードが記されていて、そのパスワードをウェブページで入力すると、懸賞に応募できる (パスワードは一回限り有効) という例も増えてきている。これらはそれぞれ、予約した本人であることやその菓子を買った人 (もしくはその権利を譲られた人) であることを認証するための手法である。より ShownPass に近い例としては、テレビ番組やラジオ番組の懸賞に郵便はがきを使って応募する際に、番組中で発表されるキーワードが必要であることが多いということが挙げられる[2] 。

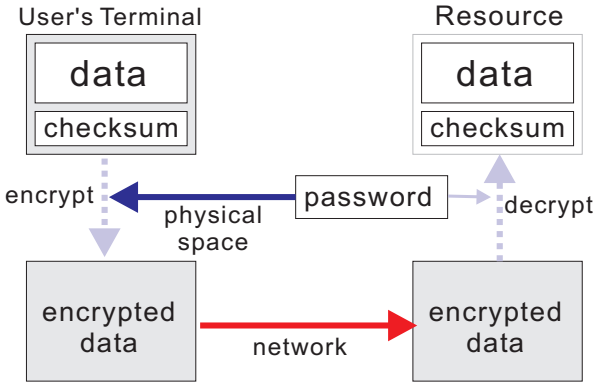

図 9: 表示されたパスワードで暗号化を行う場合

4.2 暗号化

本稿で紹介した実装例では、リクエストを送る端末側に付加的なソフトウェアを必要としないよう、表示されたパスワードをそのまま電子メイルに記すようにしている。しかし、より安全にするためには、パスワードがそのままの形でネットワーク上を流れるのは避けたほうがよい。そのためには、公開鍵暗号による電子メイル自体の暗号化や、HTTP ベースのシステムにして HTTPS[2] を利用することもできるが、ShownPass で表示されたパスワード自体を暗号化の鍵として使うこともできる。

まず、リクエストのためのデータの本体とそのデータのハッシュ値を用意する。そして、それらをまとめて、表示されたパスワードで暗号化しリソースに送る。それを受け取ったリソースは、表示しているパスワードで復号し、データのハッシュ値を計算し、復号されたハッシュ値と一致すれば、正しいリクエストとして処理を行う (図 9)。この過程においてパスワード自体は一切ネットワーク上を流れない。

[2] この例や菓子の例の場合は、アクセス制御が主目的ではなくキーワードが発表されるまで番組を見させる・聞かせるため、菓子を買わせるためという側面が強い。

4.3 自動的に変化するパスワード

時間とともに有効なパスワードが変わっていくという点は、市販されているワンタイムキーパッドを用いた認証システムと似ている。このシステムでは、パッド側と認証を受け付ける側が時刻で同期して同じパスワードを生成するようになっており、ユーザが正しいパッドを持っていることを保証することができる。これは、アルゴリズムと種となるパスワード及び時刻を揃えることにより、仮想的な「認証チャネル」を確立していると見なすことができる。ShownPass はこのパッドをリソースのそばに設置しておくという状態に近い。それにより、「ユーザが正しいパッドを持っている」という物理的制約の替わりに、「ユーザがそのリソースのそばにいる」という物理的制約を保証している。

なお、ShownPass の場合はパスワードを一定時間で変更するのではなく、ユーザから (物理的な) 要求があるごとに一定時間内に一回限り有効なパスワードを生成するようにしてもよい。そうすることにより、遠隔地からたまたま正しいパスワードを当ててしまう確率は激減し、また、有効時間を長めに取り、そこへ来た人にはその日のうちは遠隔地からの利用を許す、といった応用の仕方も可能になる。

4.4 「見る」ことを利用した機器制御

パスワードを「表示する」ということは、ユーザからそれが「見える」「見えない」でアクセス制御ができるので、管理する側・利用する側の双方にとって直感的に判りやすい。表示を一時的に止めたり隠したりすることで、管理する側は容易に特定のリソースの利用を禁止することもでき、その確認も容易である。利用する側も、利用が許されているリソースがどれであるか、認識がしやすい。また、3 節で述べたように、リクエストの結果が物理的に確認できることも重要である。パスワードが見える位置であれば、結果の物理的な確認も容易であろう。間違って他のプリンタに出力してしまうことを防止することもできる。

「見える」ということを直感的なアクセスのために使うというアイディアは、Gaze-Link[9] でも用いている。Gaze-Link は、リソースに張り付けた二次元コードを、カメラ付きの端末で認識することにより、目的のリソースを識別し、これらの端末とリソースのコネクションを確立するという手法である。これを拡張し、二次元コードを表示デバイスに表示し、機器の識別情報に加えてパスワードの情報も含めることにより、ShownPass を同時に利用することも可能である。この場合、ユーザが自分でパスワードを入力する手間を省くことができる。二次元コード化されたパスワードとともに通常の文字列としてもパスワードを表示しておけば、カメラ付きではない端末からの利用も行える。

4.5 物理的な場所を限定するアクセス制御

あるデータに対しアクセスできる場所及び時間を制限するというアイディアと実装が SpaceTag[7] で提案されている。また、携帯電話のサービスなどで、ユーザのいる場所に応じた情報を提供するようなことが行われ始めている。これらは GPS などを利用して得られる大域的な位置を元にデータを提示することが想定されているが、赤外線を発するビーコンを用いるなどの屋内での位置認識技術を用い、提示されるデータとしてパスワードを与えられるようにすれば、ShownPass と同じようなことが実現できる。

Pick-and-Drop[4] や mediaBlocks[6] のような、計算機上のデータをあたかも物理的な実体かのごとく扱う実世界指向ユーザインタフェースも、本質的に、物理的な場所が限定されたアクセスを提供できる。しかしこれらは特別なハードウェアを利用しており、現在の一般的な端末などで利用することはできない。また、広くネットワークを跨った状態で利用するためには、別途セキュリティの問題を考慮する必要がある。

4.6 近接通信を用いる場合との比較

ユーザの端末とリソースとの間の通信手段として Bluetooth[3] などの近接通信を用いることにより、リソースの近くにいる場合のみそれを利用可能にすることも考えられる。たとえば、Bluetooth を用いるならば、その電波の届く範囲が自然にその利用可能な範囲となる。しかし、電波の届く範囲は直感的に認識するのが難しく (壁を越えることもある)、確認にも手間が掛かり、制御が難しい。また、範囲内に複数の (特に同種の) リソースがある場合には、利用したいものを識別する方法を用意する必要がある。そして、ユーザの端末側もリソースが用意しているのと同じ種類の近接通信手段を備えている必要がある。

ShownPass の場合は、パスワード表示が見える範囲が利用可能な範囲なので、判りやすく、確認も容易であ

る。また、リソースとパスワード表示が対になっているので、リソースの識別も容易である。そして、端末やリソースが備えている通信手段によらず、既に多くの機器で使われているソフトウェアを使って利用できる。

さらには、人間を介するということを利用して、柔軟な運用を簡単に行うこともできる。たとえば、ユーザが電話でリクエストを送る先のアドレスとパスワードを他の人に伝え、ユーザのいる場所のプリンタから直接資料を出力してもらうことも可能である。

4.7 リソースに対し物理的な操作を要求する

リソースが提示するパスワードをユーザが自身の端末から入力するのではなく、逆にユーザが生成したパスワードをリソース側にキーボードなどで入力し、それをネットワーク経由のリクエストにも添えて伝えることにより、ShownPass と同様な機能を実現することもできる。この場合、「見える」だけではなく、「物理的に操作できる」必要があるため、より制限の厳しい方法であると考えられる。

掲示板の例のように物理的な要求を元にパスワードを表示する方法はこれらの中間的な方法である。すなわち、「見る」ために「物理的な操作」を要求している。リソース側にパスワードを入力するようにしたほうが、パスワードを表示する必要がないのでより安全ではあるが、利便性や適用可能なリソースなどを考慮すると、ShownPass を用いるほうが実用的であろう。

4.8 その他の物理的な制限による制御

データのアクセスやソフトウェアの利用などに物理的な制限を加える手法として、ハードウェアキーもしくはドングルと呼ばれる装置を利用する方法が使われている。この物理的なキーを一時的に貸し出すことにより、一時的なアクセス許可を与えることが考えられるが、キーを物理的に管理し、紛失などに対応しなければならないという手間が発生する。また、その物理的なキーを使える端末は種類が限られてしまうであろう。ShownPass のようにキーとなる部分が単なるデータであり自動的に変化するものであれば、そのような手間は生じないし、幅広い種類の端末を利用することができる。

5 まとめ

本稿では、たとえば訪問者に対してプリンタの利用を許可するといった、あるユーザに対し一時的にネットワーク上の特定のリソースの利用を許可するような場合に適した、ShownPass と呼ぶ手法を提案した。リソースの近くに時刻とともに変化するパスワードを表示しておき、そのパスワードを添えたリクエストのみが処理されるようにすることにより、リソースの近くにいるユーザのみがそのリソースを利用できるようにすることができる。ユーザは、物理的にリソースの近くにいればまったく別のネットワークに接続されている機器からでもそのリソースを利用できる。ShownPass を実装した例として、電子メイルを用いてリクエストを行うプリンタと掲示板を紹介した。

今後は、他の実装例も開発し、実際の利用における利便性や有効性を検証していく予定である。また、ShownPass で用いているような、物理的もしくは何らかの意味で制約のある「隔離された別の通信路」を利用した他の応用も探求していく予定である。

参考文献

[1] RFC 1867. Form based file upload.

[2] RFC 2818. Http over tls.

[3] Bluetooth. http://www.bluetooth.com.

[4] J. Rekimoto. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *UIST '97*, pp. 31–39, October 1997.

[5] F. Stajano. *Security for Ubiquitous Computing*. Wiley, 2002.

[6] B. Ullmer, H. Ishii, and D. Glas. mediaBlocks: Physical Containers,Transports, and Controls for Online Media. In *SIGGRAPH '98 Proceedings*, pp. 379–386, 1998.

[7] 垂水浩幸, 森下健, 中尾恵, 上林弥彦. 時空間限定型オブジェクトシステム：spacetag. インタラクティブシステムとソフトウェア VI 日本ソフトウェア科学会 WISS'98, pp. 1–10. 日本ソフトウェア科学会, 近代科学社, December 1998.

[8] 河野通宗, 長健二朗, 綾塚祐二, 暦本純一. ユーザインタフェースを活用したセキュリティモデル. インターネットコンファレンス 2002 論文集, pp. 43–51. 日本ソフトウェア科学会, November 2002.

[9] 綾塚祐二, 松下伸行, 暦本純一. 実世界指向ユーザインタフェースにおける「見ているものに接続する」というメタファ. 情報処理学会論文誌, Vol. 42, No. 6, pp. 1330–1337, June 2001.