ユーザインタフェースを活用したセキュリティモデル

河野 通宗　長 健二朗　綾塚 祐二　暦本 純一
株式会社 ソニーコンピュータサイエンス研究所

{mkohno,kjc,aya,rekimoto}@csl.sony.co.jp

情報家電やユビキタスコンピューティングの実現のためにはセキュリティの確保が必須である。しかしパスワード入力等の煩雑な操作は不適当であり、一般にセキュリティとユーザビリティを両立させることは難しい。
本論文では、認証手続きに人間が本質的に介在することの重要性を指摘した上で、それをシステム技術として理解するための通信モデルを提案する。そしてこのモデルと直感的なユーザインタフェース技術を使うことで、セキュリティとユーザビリティの両立が可能であることを示す。

A Security Model with User Interface Techniques for Ubiquitous Computing

Michimune Kohno,　Kenjiro Cho,　Yuji Ayatsuka,　and　Jun Rekimoto
Sony Computer Science Laboratories, Inc.

1　はじめに

コンピュータネットワークの著しい発達と普及により、どこにいても広いバンド幅と有線・無線を問わない接続性が提供されつつある。Web 技術、VPN、ホットスポットサービスなどが普及したことで、人は物理的な位置の制約なしに、どこからでもコンピュータ通信を利用できるようになった。

しかしウイルスやポートスキャン、パケット盗聴（タッピング）、なりすましなどのいわゆるクラッキングの問題が大きくなって来ている。これからもネットワークの拡大にともなってますます被害は増加して行くものと思われ、セキュリティの確保は現在もっとも重要な研究テーマの１つとなっている。

一方、最近では情報家電の実現がいよいよ現実的となり、UPnP や SLP などの家電制御を意識したプロトコルも多数提案されている [1, 2]。これらのプロトコルは、PC、携帯電話、携帯端末などで遠隔地からでも家電を制御できるようにすることを主な目的の１つとしている。

家電がネットワークに接続するようになると、セキュリティの問題は一層深刻になる。例えば自宅のテレビを見ず知らずの他人が勝手に操作できてしまうようでは、まったく使い物にならない。かといって自分の所有物を操作する度にいちいちパスワードを入力しなければならないのであれば、それは極めて不便である。指紋認識などのバイオメトリクス認証による解決を試みるアプローチは多いが、セキュリティにはいろいろな側面が含まれているため、一概にそれだけで解決するのは困難である。

本研究は、上記の問題を解決し、セキュリティとユーザビリティを両立することを目的とする。以下ではまず認証手続きにおける本質的な課題を指摘し、それをシステム技術的な視点から理解するための通信モデルを提案する。そのモデルに従って何種類かの通信形態を分析し、ユーザインタフェース技術を適用することで目的を達成可能であることを示す。

2　セキュリティの現状と問題

様々なアプライアンスがコンピューティング機能を持ち、かつそれらがネットワークに接続して相互に通信することは現実になりつつある。しかしユビキタスコンピューティング (Ubiquitous Computing) [3] は、セキュリティ問題の解決なしには

実用になりえない。Stajano は過去のユビキタスコンピューティングや Augmented Reality に関する研究に触れ、それらに関する様々なセキュリティの問題を分析している [4] が、解決のための提案は少ない。

ネットワークセキュリティについては、暗号化方式、プロトコル、認証などについて盛んに研究されており、既に実用として利用されているものは数多い [5, 6]。これらはユビキタスコンピューティングにおいてもセキュリティ確保のために必須の技術であるが、パスワード入力などの煩雑な操作を常に要求されるようではユーザビリティの観点からは望ましくない。

つまり、セキュリティとユーザビリティの両立が実用的なユビキタスコンピューティングの実現のために必須であるが、未だそれは達成されていない。本研究ではセキュリティに関する問題の中で認証手続きについて着目する。

2.1 高次の認証

パスワード入力による認証は、今も広く使われている認証方法である。しかし指紋やパスワードで認証できるのは、あくまでそのユーザが利用を許可されたユーザであることだけである。したがって、たとえば通信しようとしている相手先ノードが本当に正しい相手であることを保証するわけではない。これはデジタル証明書を相互に交換することで解決できるが、**最初にお互いを登録する時だけは、必ず人間が介在しなければならない。**つまり、メーラやブラウザを使って証明書を交換する行為自体が、正しい相手と通信できていることの本質的な検証になっている。

別の例としては、あるウェブサイトにユーザ登録する場合を考える。多くのサイトがメールアドレスの入力を求め、そのアドレスに対して仮パスワード（またはテンポラリ URL）を書いたメールを送付する。ユーザはその仮パスワードを使ってログインすることで、正式登録される。この場合は、ウェブサイトがユーザ確認のために電子メールの到達確認をしている。ここでユーザが介在することで本人確認を行っているのである。

これらの認証手続きは、むしろ人間が介在することが重要であると言える。これを本論文では OSI7 階層より上位の層という意味で「高次の認証」と呼ぶ。

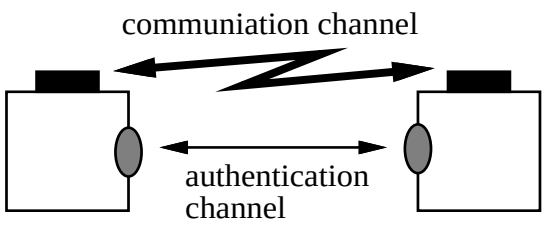

図 1: 高次の認証を実現する通信モデル: 高次の認証用に、人間が通信に寄与する通信媒体を利用する（位置依存デバイスなど）。通常の通信にはイーサネットなどの大容量メディアを使う

2.2 物理的な存在証明

認証の問題には、「物理的な存在証明」の問題も含まれる。前述のように、コンピュータネットワークは、物理的な位置の制約なく通信ができることを目的として発達してきた。現在のインターネットのアーキテクチャは、プロトコルまで含めて本質的に物理的位置と独立である。言い替えれば、物理的位置を隠蔽する構造になっている。しかし一方では、物理的位置に依存したサービスやユーザインタフェースを提供することで利便性を向上させるコンテキスト・アウェアネスの有効性が認識されてきている [7, 8]。コンテキスト・アウェアなシステムを現在のネットワークアーキテクチャ上で実装する際は、たとえ通信する対象ノードが目の前にあったとしても、そのノードの論理的アドレスを得るためには何らかのマッピング作業が必要になる。つまり、通信しようとしているアドレスが本当に「それ」のアドレスなのかは、論理的アドレスだけからは絶対にわからない。本論文ではこれを**物理的な存在証明問題**と呼ぶ。物理的な存在証明問題を解決するためには、本人確認の問題と同様に高次の認証が必要になる。すなわち、正しいノードにアクセスしていることを、人間自身が知覚できることが必要である。

3 高次の認証のための通信モデル

このように、認証手続きにはプロトコルだけでは解決が難しく、本質的に人間が介在することが重要な側面がある。高次の認証を行っている時の通信フローについてよく考えると、人間が物理的に通信に寄与する通信媒体を使うことが重要であ

ることがわかる。これを図1に示す。図では認証用の通信路（以後認証チャネルと呼ぶ）と通常の通信用（通信チャネル）の2経路を示している。認証チャネルは概念的なものであり、物理的なネットワークインタフェースに限らない。先の例で考えれば「ウェブ」という通信チャネルに対して「メール」という認証チャネルを使っている。

本研究ではこの考えにもとづき、さらに直感的な操作ユーザインタフェースの技術を取り入れて、セキュリティとユーザビリティを両立させることを目的とする。

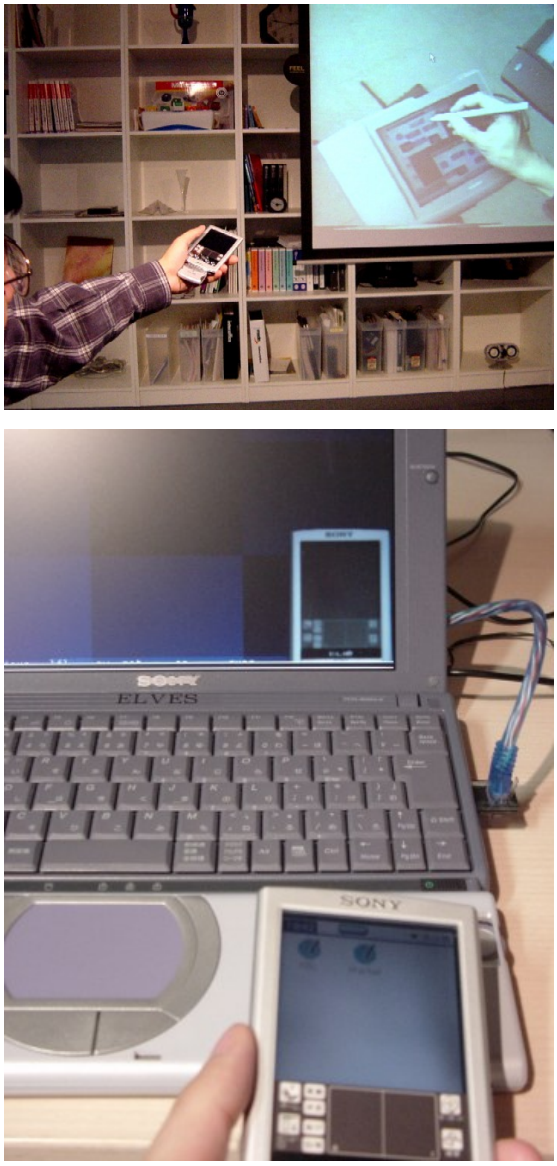

図 2: 直感的な操作による接続先指定: “向ける”、“近づける” などの直感的な操作が接続先を指定している

3.1 コネクションの種類と認証

以下では、ユーザが携帯可能マシンを使用して別のノードと接続する場合に絞って考慮する。まず、コネクションの種類を以下の項目について場合分けする。

- 接続先ノードの種別（固定／モバイル）
- ノード間の物理的距離（近接／遠隔）
- 認証を要するリソースへのアクセス（あり／なし）

2番目の項目は、ノード同士の距離関係に加えて、実際に近接させることが可能な場合のみ「近接」と考える。したがって、デスクトップ端末同士の場合は、仮に近くに存在していても近接とはならない。また3番目の項目は認証が求められるリモートコンテンツへのアクセスが発生するか否かを示す。これらの項目の組合せによりコネクションの種別を定義する。以下ではこの場合分け毎に、どのようにユーザインタフェースの技術を取り入れるかを説明する。

3.2 近接ノードへの接続

はじめに近接ノードへの接続について考える。まずこれに関連するユーザインタフェース技術の概要と背景について簡単に説明する。

一般に、インターネット上であるノードからあるノードに接続を開始するためには、接続先のIPアドレスやホスト名などの、いわゆるエンドポイントのアドレスを指定する必要がある。ユーザはそれをキーボードで入力したり、GUIで選択したりする。これをPCで行う分には何ら問題ないが、PDA等の小型の携帯端末ではやや扱いづらい。

また、接続対象のPCが携帯端末のすぐ近くにあることも少なくない。これは、PDAとPCがすぐ間近にあるにも関わらず、そのホスト名を選択する必要があることを意味する。自分のネットワーク環境ならまだしも、出張先などでは問題が起きることがある。

この問題に対して暦本らは、携帯端末上のアプリがPCと無線LAN経由で接続する際に、端末をPCに近づけることで通信相手を特定するというユーザインタフェース技術を提案している[1]。「近づける」または「機器を向ける」という直感的な操作で接続先ノードを指定できるため、携帯端末

[1]COMDEX2001 にて発表済み。

での近接ノードへの接続に適している（図 2）。

これは RFID や IrDA を用いて実現されている。RFID は近接でのみ通信でき、IrDA はトランシーバ同士がある偏角以内で向かい合った時にのみ通信できる。このような物理的な制約を持ったメディアを使ってエンドポイント情報を交換した後は、無線 LAN などの広帯域で位置制約の少ないメディアで通信する。

この技術はもともと通信相手を容易に指定するためのユーザインタフェースであるが、ここではこれをセキュリティ的な視点から考える。すなわち、近接することで通信を開始するということは、逆に言えば通信できる時にはノード同士が近接していることを意味する。つまり、近接通信デバイスが第 3 節で述べた認証チャネルとして機能していることになる。

このシステムを使って、近接通信デバイスでエンドポイントのアドレスや証明書を交換すれば、ユーザ自身によって通信相手の物理的な存在証明を確立でき、かつセキュリティに関連するデータをイーサネットを一切通さずにやり取りできる。もちろん近接通信デバイス経由も暗号化通信を行い、タッピングされても問題ないようにすることが望ましい。

重要なのは、認証チャネルの通信発生が直感的なユーザインタフェースと密に結び付いていることである。この特徴はセキュリティとユーザビリティの両立に極めて有効である。

3.3 遠隔ノードへの接続

次に遠隔ノードへの接続における認証について考える。

最近では、PDA のような小型の携帯でも、802.11a/b や BlueTooth などの広帯域なネットワーク通信機能を持つようになってきている。今後ホットスポットサービスのような通信インフラが増加するに伴い、これらの携帯機器からリモートサイトにある自分のプライベートデータにアクセスしたいという要求が増えると考えられる。

一方で、そのような携帯端末を VoIP による IP 電話として利用し、電話網を経由せず音声対話をすることも可能である。実際にはネットワーク層でのモビリティサポート、高速ハンドオーバ、消費電力の問題など解決すべき点は多いが、音声対話機能が PDA に統合されることで、電話とコンピュータが融合した新しいタイプの携帯端末が実現できると考える。

電話とコンピュータが融合することは、単に複数の機能が同じ機器に搭載されるということ以上の意味を持つ。例えば現在では、電話で話している相手にファイルを送信するためには、コンピュータから相手のメールアドレスに送信しなければならない。しかし電話とコンピュータが融合していれば、**電話をかけていることで通信相手は既に特定している**ので、単にその接続先にファイルを送信すればよく、メールアドレスを間違えるというようなこともない。その上 IP での音声通信には 3 人以上の会話における制約もなく、複数の相手にまとめてデータを送ることも容易である。

上記の機能は既にインスタントメッセージアプリケーションで実現されているが、携帯端末から自分のプライベートデータにアクセスする機能は統合されていないし、携帯端末で使いやすくするためのユーザインタフェースはまったく考慮されていない。

電話とコンピュータの融合をセキュリティの観点から見てみると、大変興味深い。電話と融合することで人間が本質的に介在し、高次の認証問題を暗黙的に解決できることを、以下に具体的に説明する。

電話で音声対話している相手にファイルを送付することを考える。ネットワーク的な観点で考えると、前述したように、電話をかけていることで通信相手は既に特定しているので、新たにアドレス解決をする必要はない。これをセキュリティ的な観点で考えると、ユーザが音声で相手と会話しているので、**通信相手が本当に正しい相手であることがあらかじめわかっている**。つまり「音声による対話」が認証チャネルそのものと考えることができる。これはネットワークアーキテクチャとはまったく異なるセマンティクスであるが、だからこそそれだけでは解決が難しい認証問題を解決することができると考える。

3.4 リモートデータへのアクセス

次に、携帯機器から遠隔地にある自分のプライベートデータにアクセスするときの認証問題について考える。この場合は遠隔地のサイトに人間が

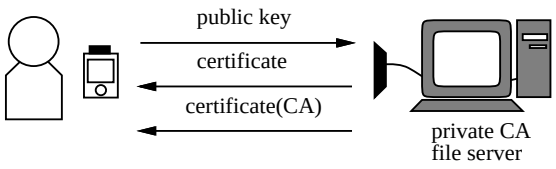

図 3: 認証局としてのデータストレージへの登録: プライベートデータのサーバがプライベート CA としても機能し、ユーザは自身の端末の証明書を申請する際に近接通信デバイスを使用する。

存在しないため、今まで述べたアプローチをそのまま適用することはできない。ここではリソースへのアクセス権のレベルを、そのデータの所有者、グループ、公開の 3 段階として検討する。なお、ISP にホスティングされているストレージのデータへのアクセスは、ここでは考慮しない。また自分の端末を他の人間が許可なく使うことを防ぐためには、バイオメトリクスなどの技術を利用することを想定する。

ここでまず、ネットワーク通信における盗聴防止や不正アクセスを防ぐ技術として現在広く利用されている公開鍵暗号方式（非対称鍵方式）について考えてみる。この暗号化方式は非常に強力で、man in the middle 攻撃に弱いが PKI があればそれも防げる。

しかし、認証局へ公開鍵を登録して証明書の発行を依頼する時には、結局高次の認証が必要である。例えば、あるユーザが証明書の発行を依頼するときには、その認証局にすでに証明書を発行してもらっている第三者の証明書を添付したり、あらかじめ電子メールや電話などで認証局の管理者と信頼関係が築かれたりしているのが普通である。つまり、ネットワークにおけるプロトコルだけで解決されているわけではなく、電子メールや電話が認証チャネルとして使われている。このことから、認証局への登録における認証チャネルを工夫することで、遠隔地からのプライベートデータの参照時の認証問題を解決できると考える。

本研究では、所有者からのみのアクセスは、所有者はプライベートデータのストレージに物理的に近づくことが可能であるという前提に基づくことで解決する[2]（図 3）。この前提を利用して、プライベートデータのストレージをプライベートな認証局とし、鍵の登録・交換は近接通信を通してのみ行うようにする。これにより、認証局（ストレージ）に証明書を発行してもらえるのは近接通信可能なユーザの持つノードのみになる。同時にストレージの公開鍵も受け取ることで、man in the middle の介在する余地をなくす。

ユーザビリティの観点からは、ユーザが持ち歩く携帯端末をストレージに近づけるだけで、証明書を安全に交換できる。鍵ペアを更新したい時には、単にまたストレージに近づければ済む。遠隔地からアクセスするときには、登録された鍵ペアとランダム生成したセッションキーを使えば、かなり安全に通信することができる。

「公開」のアクセス権を与えられたデータについては、その所有者が明示的に公開したものであるので、単にその URL にアクセスすればよい。しかし特定のグループにのみ公開するデータについては近接通信の前提も利用できないため、別の手段で鍵を共有するか、そうでなければ個別にアクセス制御リストを作成しなければならない。

1 つの手段としては、データへのアクセスが発生した時にその所有者の端末にイベントが送られ、そこで所有者を介在して鍵が送られるという方法が考えられる。しかしこれでは所有者の負荷が高すぎて実用的ではない。現在のところこのアクセス権に関する有効な手法は見出せておらず、今後の課題である。

3.5 まとめ

以上で、種々の通信における認証チャネルの活用方法について説明した。まとめると以下のようになる。

- 近接・固定ノード: 近接通信による接続
- 近接・モバイルノード: 近接通信による接続
- 遠隔・モバイルノード: VoIP セッションの利用

遠隔・固定ノードへの接続については人間が介在する余地がなく、今後の課題である。

4 IP 電話とユビキタスコンピューティング

さて、通信機能を持ったアプライアンスが生活環境に沢山存在するユビキタスネットワーク環境

[2] スーパーユーザのログインをコンソールからのみに限定するイメージに近い。

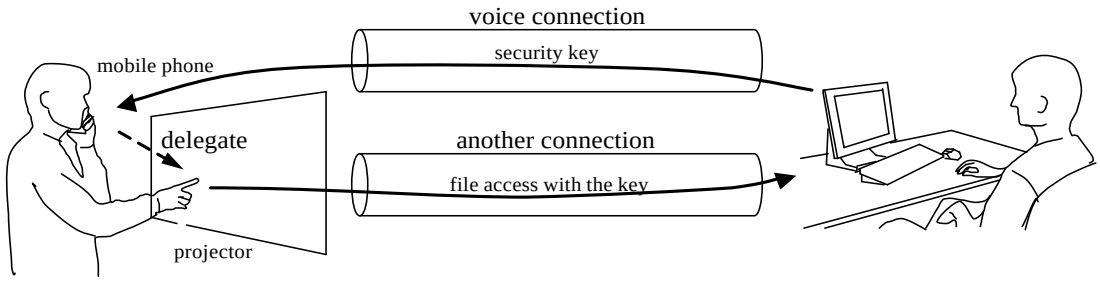

図 4: IP 電話を利用したアクセス権の委譲

になると、それらと携帯機器が協調することで、新たな応用が産まれる。例えば携帯機器で電話をしている状態から、車やオフィスのスピーカとマイクを使った対話に切り替えたり、電話の相手から送信されたファイルを近くのプロジェクタで表示することなどができると考えられる。ここではそれを実現するための具体的な問題点とその解決方法について議論する。

まず、IP 電話での対話について詳細に考える。ユーザが IP 電話で対話を行おうとするとき、相手先の端末へ発呼するパケットが送られる。相互に公開鍵が交換され、接続が成功するとセッションが開始する。このセッションで盗聴防止のために使われる暗号鍵は一時的なセッションキーであり、セッションが終了するとセッションキーは破棄される。

例えば携帯電話で話していた音声対話セッションを、近くにある anonymous なスピーカとマイクに切り替えたいような場合は、このセッションキーを近接通信で渡してやればよい。これによって、セッションキーを無線 LAN を通すより安全で、かつ直感的な操作で渡すことができる。

一時的なアクセス権の譲渡

しかし、相手から送信されたファイルをプロジェクタに表示するような場合は、単純にセッションキーを共有するだけでは望ましくない。このセッションキーを使って勝手にプロジェクタを管理するノードがサーバからデータを取得してしまうかもしれないからである。

これを避けるためには、プロジェクタに対しては別のセッションを張り、そこを通してファイルを転送することが必要である。プロジェクタノードは「近接・固定」ノードなので、近接通信デバイスを使ってコネクションを開けばよい。ファイル自体ではなくアクセス権のみを転送する場合は、短時間の有効期限（あるいは通話が継続している間のみ）を持つ新規のアクセス権を生成し、それを送信する。図 4 にこの処理の概要を示し、動作を以下に説明する。

1. 対話相手（図中右側）が、ユーザ（左側）に見せたいファイルへの一時的なアクセス権を生成する。
2. ファイルへの URL とアクセス権が書かれたファイルをユーザに送信。送信処理は音声対話のコネクションを通して行われる。
3. ユーザは近接通信を使ってプロジェクタとのコネクションを開き、そのコネクションを通して今受け取ったアクセス権ファイルを送信する。
4. プロジェクタノードはアクセス権ファイルを使ってリソースにアクセスし、データを出力する。

ファイルのオーナの立場からは、電話で話している相手が近接できるデバイスならアクセスする権利を許すということになり、認証チャネルを使った認証の継承関係が成り立っていると言える。アクセス権には様々なエントリが含まれると考えられるが、現在は有効期限だけしか考慮していない。

5 実 装

前節までに述べた機能を Windows2000 と PocketPC 上のアプリケーションとして実装した。図 5 に、その使用例を示す。Visual C++と eMbedded

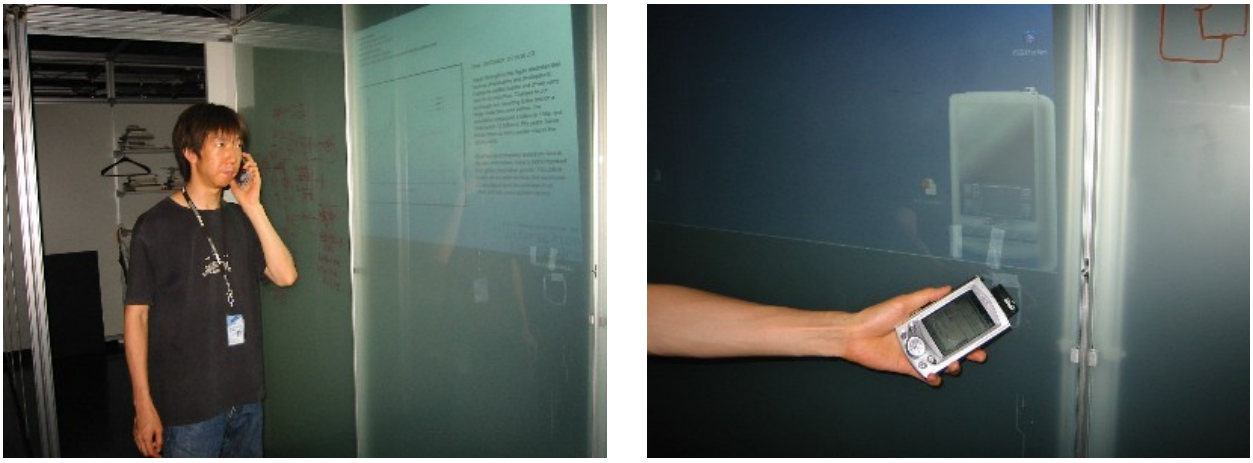

図 5: システムを利用している様子: IP 電話と近接通信の組合せによって、近接・遠隔双方の認証チャネルを利用している

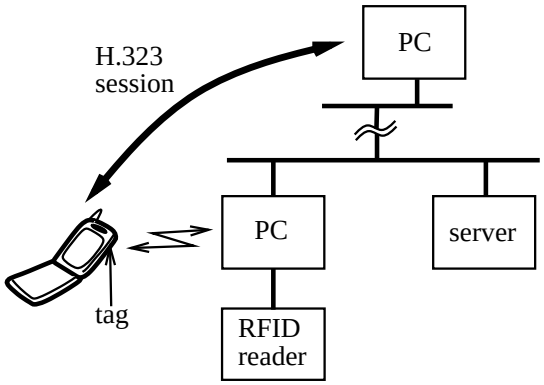

図 6: デモシステム概要

Visual Tools で記述し、ライブラリは OpenSSL, OpenH323, UPnP を使用した [9, 10]。プロジェクタに接続した PC（以後プロジェクタ PC と呼ぶ）上のシステムの実装には DirectX も使用した。

図 6 にシステムの全体図を、図 5 に実際に使用している様子を示す。端末は PocketPC デバイス（CASSIOPEIA E-2000）に無線 LAN カードを差し、近接通信のために RFID タグを端末の先端に取り付けたものを使った。VoIP 機能は OpenH323 ライブラリで実装した。

壁面には PC の画像を投影して RFID リーダのアンテナを設置し、PC に接続した。RFID リーダとタグは Texas Instruments 社の TIRIS シリーズを使った。このタグは樹脂に封入された小型のトランスポンダで、リーダから出力された電波を電源として使って 64 ビットの ID を返す。近距離（10cm 程度）でしか通信できないが、本研究のような用途においては近距離でのみ通信できる方がむしろ適しているので、これを利用した。

5.1 端 末

端末上のアプリは起動時に、まず自分の RFID と IP アドレスをサーバに登録し、次にサーバ上にあるユーザプロファイルとコンテンツリストを取得する。ユーザプロファイルにはユーザのアドレス帳が含まれており、LCD 上で選択することで電話をかけることが出来る。コンテンツリストはプライベートデータのディレクトリである。

サーバ上のコンテンツの格納方法について説明する。ユーザにはサーバによって割り振られた 1 つのユーザ ID があり、1 つのユーザ ID ごとにプロファイルとコンテンツリストがそれぞれ対応付けられている。また 1 つ以上の RFID が 1 つのユーザ ID に対応付けられている。これにより、複数の端末を一人のユーザが持つような場合に、両方の端末から同じプロファイルが参照されるようにした。この登録処理は第 3.4 節で述べたように近接デバイスを使って行われるべきだが、そのためのハードウェアが未実装なため、今回は手動で設定した。

端末上のアプリは、コンテンツウィンドウとコントローラウィンドウの 2 つのウィンドウを持つ。コンテンツウィンドウにはアドレス帳かコンテンツリストが、コントローラウィンドウには電話や各種アプライアンスの制御用インタフェースが表示される。通話中で、かつどのアプライアンスと

も接続していない時は、電話操作用のコントローラが表示されている。このコントローラには電話の切断やファイルの転送、エンドポイントの転送などのボタンを HTML で記述して実装した。

5.2 ファイル転送

コンテンツリスト中の 1 つのエントリを通話中に選択することで通話相手に送信できる。この際の転送処理には H.323 の文字列送信機能を使った。この機能を使って転送するデータはリソースにアクセスするために必要な情報（アドレスとアクセス権）だけなので、その長さが極端に長くなることはない。

通話相手からコンテンツの URL が送られると、自分のコンテンツリストが更新され、ユーザにはベル音で知らされる。ユーザが端末を RFID リーダのアンテナに近づけるとプロジェクタ PC が RFID タグを検出し、サーバに問い合わせて RFID から端末の IP アドレスを得て通信を開始する。これでセッションが確立する。この時端末には HTML で記述されたプロジェクタのコントローラ画面がプロジェクタ PC からダウンロードされ、コントローラウィンドウに表示される。ユーザはそれを使ってプロジェクタを制御できる。これは UPnP を使って一般的な機能として実装したので、プロジェクタに限らず情報家電一般に適用可能である。

このコントローラを使ってユーザがコンテンツを選択することで、ファイルエントリがプロジェクタに対して送信される。プロジェクタはそのファイルの URL とアクセス権データを使ってファイルの実体を取得し、出力する。出力は、ファイルの拡張子に応じたアプリケーションを起動することで行われる。これは Windows が標準で提供している機能をそのまま利用した。ただしコントロール画面は PowerPoint 用だけを実装してある。ファイル型に応じたコントロール画面の提供は今後実装する予定である。

5.3 エンドポイントの転送

音声セッションのエンドポイントをユーザの行動や場所に応じて転送する例として、デスクトップ PC と端末との間の転送を実装した。

デスクトップ PC に端末のクレイドルを接続し、マイクロスイッチの状態が RS232 を通して出力

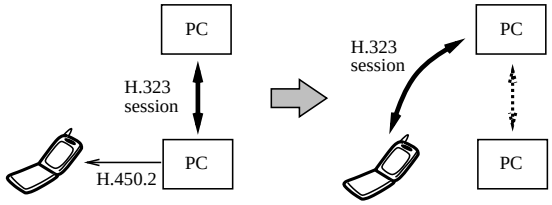

図 7: エンドポイント転送

されるハードウェアを作成してクレイドルに付加した。端末がクレイドルにささっている時はマイクロスイッチは ON を、それ以外では OFF を出力する。

デスクトップ PC で音声対話をしていて、かつ端末がクレイドルにささっている状態の時に端末をクレイドルから外すと、そのイベントがデスクトップ PC に伝わる。このイベントを受信すると端末に対して H.450.2 エンドポイント転送のメッセージが送られる（図 7）。これによりエンドポイントが端末に切り替わり、ユーザは端末を使って音声対話を継続できるようにした。

6 まとめと今後の課題

人間が本質的に介在する認証手続きの重要性について述べ、それを認証チャネルを使った通信としてモデル化した。このモデルに基づき、種々の通信形態における認証チャネルにユーザインタフェース技術を取り入れることでユーザビリティとセキュリティの両立に寄与できることを示した。特に電話を使ったセキュリティモデルと言うものはこれまで提案されていない。これらはユビキタスコンピューティングと親和性が高く、今後ますます有用になるであろうと考えられる。

今回はデモシステムを実装して有効性を確認したが、今後はプロトコルやアクセス権についてさらに詳細に固めて行く予定である。また初期登録された証明書入りの端末が物理的に盗まれてしまう場合の問題は未解決であり、revocation などについても検討が必要である。

実装における課題としては、RFID タグが読み取り可能領域に複数存在する場合に PC の接続対象が激しく切り替わってしまう問題がある。RFID リーダがどのタグを認識するかはハードウェア的に不確定なため、現状ではデバイスの改善を待つ必要がある。また今回の実装ではプロジェクタな

どのアプライアンスにRFIDリーダを接続したが、これだと個々のアプライアンスにリーダを設置しなければならない。しかしPDAにリーダを持たせることができればアプライアンスにはタグだけを貼れば済む。最近CompactFlash型のリーダが数社から発売されたので、今後はそれを利用して環境への導入負荷を軽減したシステムにしていきたい。

謝 辞

この研究を進めるにあたり多大かつ貴重な意見をいただきました、株式会社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ならびにネットラボの皆様に感謝いたします。

参考文献

[1] IETF. Session Initiation Protocol. `http://www.cs.columbia.edu/sip/`, 1999.

[2] Microsoft Corporation. Understanding Universal Plug and Play: A White Paper. `http://www.upnp.org/`, 2000.

[3] M. Weiser. Some Computer Science Issues in Ubiquitous Computing. *In Special Issue, Computer-Augmented Environments*, Vol. 36, No. 7, July 1993.

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

[5] 有賀征爾, 南正樹, 江崎浩. IP Security ソフトウェア処理の性能評価. インターネットコンファレンス'99 論文集, pp. 57–66, December 1999.

[6] 村上陽子, 小川浩司, 大川恵子, 村井純. 電子証明書を用いたインターネット成績通知証明システム の設計と実装. インターネットコンファレンス'99 論文集, pp. 39–46, December 1999.

[7] J. Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'98*, pp. 31–39, 1998.

[8] S. Holland and D. Oppenheim. Direct Combination. In *Proceedings of CHI99*, pp. 262–269, April 1999.

[9] OpenSSL Project. OpenSSL Documents. `http://www.openssl.org/`, 1998.

[10] OpenH323 Project. H.323 standards. `http://www.openh323.org/`, 1998.

[11] ユーリス・ブラック. インターネット・セキュリティガイド. ピアソン・エデュケーション, 2001.

[12] G. Zimmermann, G. Vanderheiden, and A. Gilman. Prototype Implementations for a Universal Remote Console Specification. In *Proceedings of CHI2002*, April 2002.

[13] M. Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *Proceedings of UbiComp2002*, September 2002.

[14] S. Capkun, L. Buttyan, and J.P. Hubaux. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. Technical report, EPFL/IC/200234, 2002.

[15] 下遠野亨, 野口哲也. 対面無線アドホック通信に適した暗号通信路構築方法. マルチメディア、分散、協調とモバイル (DICOMO2002) シンポジウム論文集, pp. 559–562, July 2002.