

第2回 システム環境設定ゼミ

ゼミ担当者 : 江上 透, 市川 親司, 佐藤 史隆
 指導院生 : 片浦 哲平, 米澤 基, 澤田 淳二
 開催日 : 2003 年 4 月 24 日 13:15-14:45

ゼミ内容: 本ゼミでは, インターネットを利用する際に必要となるプロキシサーバや, IP アドレス, DNS といった知識, SSH を用いたセキュリティの向上への取り組みとして, ポートフォワーディングや SCP, 暗号や認証の必要性について学ぶ。

1 プロキシサーバ

プロキシサーバとは, ローカルエリアネットワークとインターネットとの接続地点で, クライアントからのリクエストを代行して, 両者の通信を中継するコンピュータやアプリケーションのことである。プロキシとは「代理」の意味で, 通常はファイアウォール上で稼働させる。インターネットから社内ネットワークへの通信は遮断する一方, 社内ネットワークからインターネットに関しては, ユーザーやアプリケーションを指定してアクセスの制御を行なう。

プロキシサーバには, キャッシングというもう一つの大きな機能がある。HTTP プロキシでは, ユーザーがアクセスしたページを保管しておくことで, 2 回目以降のアクセスが高速化される。

FTP や Telnet, Gopher などのプロキシサーバもあるが, 単にプロキシと言った場合, HTTP での通信を中継する HTTP プロキシを指すことが多い。

2 IP アドレス

2.1 グローバルアドレスとプライベートアドレス

IP アドレスとは, PC1 台 1 台に割り振られた識別番号のことである。現在広く普及している IPv4 では, IP アドレスは 8 ビットずつ 4 つに区切られた 32 ビットの数値が使われており「192.168.6.18」などのように表現する。よって, 約 42 億台 (2 の 32 乗) がインターネットに接続できるが, 近年のインターネットの急成長により IP アドレスの不足が問題になっている。そこで特定の IP アドレス領域を LAN 用に確保して, 再利用するという規格が定められた。この LAN 用のアドレスをプライベートアドレスと呼ぶ。LAN の規模によってその領域は, クラス A, B, C と 3 つに規定されている (Table 1)。

2.2 ルータとネットマスク

プライベートアドレスはあくまでも LAN 用のものにすぎないため, これを利用する端末は直接インターネットに接続することができない。そこで, プライベートア

Table 1 各クラスのプライベートアドレス領域

クラス名	領域
クラス A	10.0.0.0 ~ 10.255.255.255
クラス B	172.16.0.0 ~ 172.31.255.255
クラス C	192.168.0.0 ~ 192.168.255.255

ドレスとグローバルアドレスを相互に変換する仕組みが必要になる。この機能を担うのがルータである。ルータは, ネットワーク上を流れるデータの行き先をみて, 同じ LAN の端末へ送るデータなのか, ほかのネットワークへ送り出すデータかを見分ける。ここで見分けるために使用されるのがネットマスクである。

まず IP アドレスにはネットワークを識別するネットワークアドレスと, 個々の端末を識別するホストアドレスがある。ネットマスクはこの IP アドレスのうち, ネットワークアドレスとホストアドレスの識別がどこかを定義する 32 ビットの数値である。サブネットマスク値から IP アドレスとビットの論理積を計算することによって, IP アドレスのネットワークアドレス部を取得できる。

例えば, 「192.168.6.121」という IP アドレスを「255.255.255.0」というサブネットマスク値を使って分割する。するとこの IP アドレスが示すのは, 192.168.6.0 というネットワーク上にある, ホストアドレス 121 の端末ということが分かる (Fig. 1)。

ところで, IP ホストは正確には「0」と「255」は使用できないことになっている。「0」はそのネットワーク自身のことを表すために使用され, また「255」はブロードキャストアドレスとして使用される。ブロードキャストアドレスとは, ネットワーク上の全てのコンピュータに対して通信を行うために使用されるアドレスである。例えば「192.168.6.255」とすると「192.168.6.0」のネットワーク全ての端末に通信が行われる。

IPアドレス : 192.168.6.121

11000000	10101000	00000110	01111001
----------	----------	----------	----------

∧

ネットマスク : 255.255.255.0

11111111	11111111	11111111	00000000
----------	----------	----------	----------

ネットワーク部 ホスト部

↓

ネットワークアドレス : 192.168.6.0

11000000	10101000	00000110	00000000
----------	----------	----------	----------

ネットワーク

Fig. 1 ネットマスク

2.3 プライベートアドレスとグローバルアドレス変換の仕組み

プライベートアドレスとグローバルアドレスの変換方法は次のとおりである。例えばルータが、グローバルアドレス(126.22.99.144)とプライベートアドレス(10.25.1.0)と、2つのネットワーク接続で構成されているとする(Fig. 2)。LAN内のホストA(10.25.1.5)がインターネットへ接続する場合、ホストAはルータにパケットを送信する。ルータは受け取ったパケットのヘッダに含まれる発信元アドレス(10.25.1.5)をポート番号と関連付けた後、グローバルアドレス(126.22.99.144)に書き換えてインターネットに送り出す。パケットには送信元のグローバルアドレスの他にポート番号も格納されているため、グローバルアドレス先に返ってくるパケットでも、ポートからプライベートアドレスを割り出し、ホストAに届けられる(Fig. 2)。

2.4 IP マスカレードと NAT

ルータの機能として実装されている技術がNATやIPマスカレードである。NATはNetwork Address Translationの略である。NATはひとつのプライベートアドレスに対してひとつのグローバルアドレスを対応させるため、同時に複数のパソコンからインターネットを利用することができない。一方IPマスカレードはNATと違ってTCP/UDPのポート番号まで動的に変換されるため、一つのグローバルアドレスで複数のマシンからの同時接続を実現することが可能である。ただし、ポート番号が変化するため、インターネット側からアクセスできない、ICMPが使えない、rshなど一部のサービスが使えないなどの欠点もある。

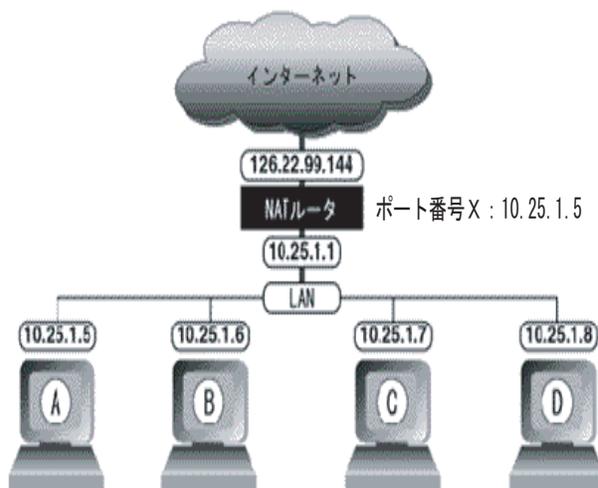


Fig. 2 ルータの仕組み

3 URL

URLとはインターネット上に存在するさまざまな情報が存在する場所を一意に特定できるようにするための情報、またはそのための情報の記述様式のことをいう。WWWブラウザでは、このURLを「アドレス」として指定することで、インターネット上の目的の情報にアクセスできる。例えばURLは以下のようなものである。

<http://www.doshisha.ac.jp/index.html>

<http://mikilab.doshisha.ac.jp/index.html>

ここで「http」はスキーム名といい、情報ソースの種類を指定する。他にもftp, wais, mailto, telnet等がある。次に「://」から「/」までの間をドメイン名という(ホスト名と呼ばれることもある)。ドメイン名は「ホスト部+ドメイン部」で構成されており、「www」や「mikilab」はホスト部、「doshisha.ac.jp」がドメイン部に当たる(Fig. 3)。ドメイン名は、インターネット上のIPアドレスは人間には覚えにくいので、わかりやすくつけた名前のことである。

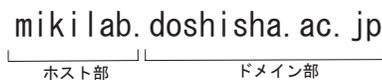


Fig. 3 ドメイン名 (=ホスト名)

「index.html」にあたる部分をパス名という。パス名ではサーバ内での情報の場所を示す。サーバ内で階層的に情報が管理されている場合には「/」を用いて階層ごとにパス名を指定する。

4 DNS サーバ

ドメイン名でコンピュータにアクセスすると、ホスト名が固有のIPアドレスに置き換わり通信が可能になる。

このドメイン名からそのドメイン名に対応する IP アドレスに変換する動作を行うのが DNS サーバである。

では次に、DNS サーバの動作を解説する。まず、DNS サーバの IP アドレスは、OS の TCP/IP の設定や、ダイヤルアップ接続の場合は、接続先から自動的に指定され、通常は接続先のプロバイダの DNS サーバや、LAN の DNS サーバが指定される。端末は、最も近い DNS サーバ A に変換要求を送る。DNS サーバ A は、要求されたドメイン名の IP アドレスを知っていれば、その情報を回答するが、もし知らなかった場合は、ドメイン名の広い範囲から順番に問い合わせを行う。まず、DNS サーバ A はトップレベルドメインを管理するルートサーバに「jp の DNS サーバは？」という要求を送る。ルートサーバから「jp」の DNS サーバの IP アドレスが回答されると、次に「ac」の DNS サーバの IP アドレス「doshisha」の DNS サーバの IP アドレスを次々と問い合わせていく。mikilab.doshisha.ac.jp の IP アドレスを doshisha.ac.jp の DNS サーバに問い合わせ、mikilab.doshisha.ac.jp の IP アドレスが判明する。最終的に、DNS サーバ A は、変換結果を端末に返答し、変換作業を終了する。

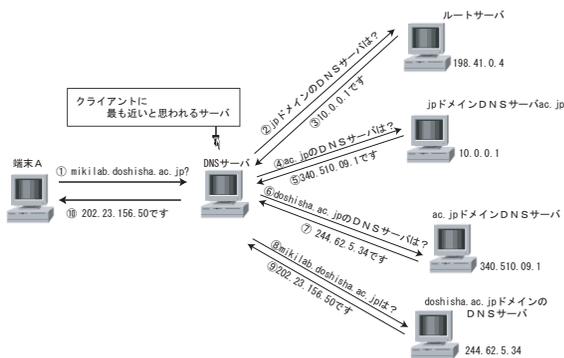


Fig. 4 DNS サーバの動作

5 SSH を用いたセキュリティ向上のための取り組み

UNIX 系 OS では従来、ネットワークを介して別のマシンにログインしたり、コマンドを実行したり、ファイルを転送するのに、Telnet や r 系コマンド (rlogin, rsh, rcp) と呼ばれる方法を用いてきた。しかし、この方法ではネットワークを流れる情報が暗号化されないで、パスワードややり取りされる情報が盗まれてしまう危険性がある。

そこで、三木研究室では前述のことを行う際に、セキュリティ向上のために SSH(Secure SHell) という方法を用いている。

SSH を用いた通信ではパスワードだけではなく、やり取りされるデータも暗号化されるので、通信の途中でデータを盗聴されたり、改ざんされたりする心配がなく

なる。また、ユーザ認証も通常のパスワードによる認証に加え、RSA 公開鍵暗号による認証も利用することができ、より通信の安全性を向上させることが可能である。

5.1 ポートフォワーディング

ポートフォワーディングとは、SSH の機能を用いて暗号化機能を持たないプロトコルでも通信路を暗号化させ、セキュリティを向上させるものである。

電子メールを送信するときのプロトコルである SMTP や受信するときのプロトコルである POP では、メール本文やパスワードの内容が暗号化されない。これでは、途中の通信路でデータを盗み見られる危険性がある。この様子を Fig. 5 に示す。



Fig. 5 通常の通信

そこで、ポートフォワーディングの登場となる。ポートフォワーディングとは、SSH に対応したプログラムを起動しておいて、ローカルホスト上にポートを開き、そこに送られてきたデータを SSH を用いてあらかじめ設定しておいたリモートホスト上の SSH サーバに送信し、SSH サーバがそのデータをさらにマシン内や別ホスト上で稼働しているサーバに送るといったものである。このことを図で表すと、Fig. 6 のようになる。こうすることで、データの暗号化がサポートされていないプロトコルでも途中の通信路では SSH を用いたデータ転送が行われるために安全にデータを送ることができる。

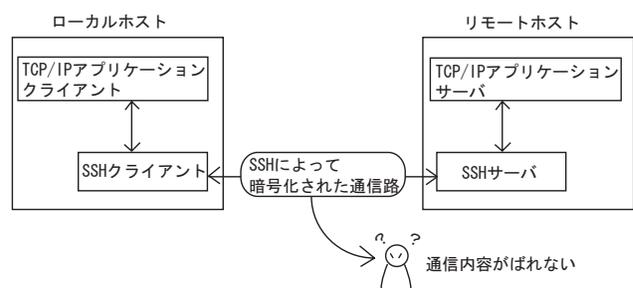


Fig. 6 ポートフォワーディングを用いた通信

ポートフォワーディングを行うためには、「自分のどのポートに送られたデータを」、「どの相手の」、「どのポートに」フォワーディングするかを設定する。たとえば、

8025:mikilab.doshisha.ac.jp:25

のような設定をすることで、ローカルホスト上の 8025 に送られたデータは SSH を経由し、最終的に mikilab の 25 番ポートに送り届けられることになる。

5.2 ファイル転送に関して

5.2.1 FTP をポートフォワーディングする方法

ここでは、FTP の PASV(Passive) モードというものをを用いる。FTP では、通常、こちらの IP アドレスとポート番号を指定することで相手側から接続が行われる。しかし、PASV モードではそれとは逆に相手側の IP アドレスとポート番号を受け取ってこちら側から相手側に接続を行う。こうすることで、サーバから指定されたポートを用いることにより、ポートフォワーディングを利用できるので、データを暗号化しつつ、安全にデータをやりとりすることができるようになる。しかし、このやり方ではサーバから指定されたポートをその都度フォワーディングする必要がある。これを手動でやるのは困難である。そこで、その処理を自動的にしてくれるソフトが必要になる。

mindterm では、サーバから伝えられたポート番号を監視し、サーバが伝えてきたポートをフォワーディングすることによって、FTP データポートのフォワーディングを実現している。

5.2.2 SCP を用いたファイル転送

SCP とは、SSH を用いたファイル転送コマンドである。従来、UNIX にはリモートホストにファイルを転送する方法として `rcp` というコマンドが使われていた。しかし、`rcp` はデータが暗号化されないために途中の通信路でデータを盗聴される危険性がある。そこで、`rcp` に変わるものとして、SCP が用いられるようになった。SCP を用いることでデータが暗号化されるため、途中で盗聴される心配がなくなる。

6 SSH で用いられる暗号技術

6.1 暗号化方式

SSH の中心的な機能である「認証」および「通信の暗号化」には暗号技術が使われている。暗号技術では、暗号化および復号化を行なう「鍵」の管理方法が大変重要であり、次の 2 通りの方式が主に使われる。

- 共有鍵暗号方式 (common-key cryptography)
1 つの秘密鍵で暗号化も復号化も行う。
- 公開鍵暗号方式 (public-key cryptography)
暗号化を行う鍵と復号化を行う鍵との異なる 1 組の鍵を用いる。SSH では、認証は公開鍵暗号方式で行ない、通信データの暗号化は共有鍵暗号方式を採用している。

6.2 共有鍵暗号方式

共有鍵暗号方式は、Fig. 7 のように、暗号化および復号化を 1 つの共通の鍵で行う方式である。

- 利点
公開鍵暗号方式と比べて、アルゴリズムが簡単であり、処理時間が短いので、大きなデータを暗号化するのに適している。
- 欠点
双方で鍵を共有し、第 3 者には秘密にする必要があるため、鍵を安全する方法が大きな問題であった。(現在では鍵の交換に公開鍵暗号方式を利用することにより解決)

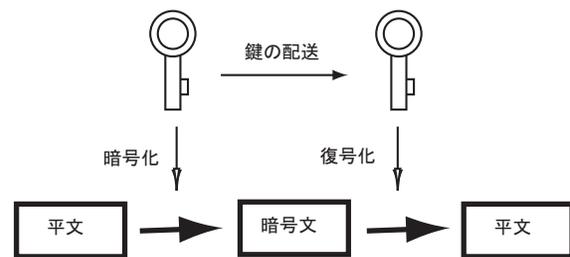


Fig. 7 共通鍵暗号方式

SSH が使用している主な共有鍵暗号の方式を紹介する。

- DES (Data Encryption Standard)
1970 年代に IBM が開発し、1997 年に米国連邦政府の暗号標準に定められ、広く利用されている。平文に対してデータの置換操作と、位置を入れ替える転置操作といった比較的単純な処理を組み合わせ、複雑な変換を施す。56 bit 長の鍵を利用し、データを 64 bit のブロック毎に処理する。
- 3DES
異なった鍵で DES 暗号化を 3 段階行うことにより、DES を強化したもの。SSH2 では標準になっている。
- IDEA (International Data Encryption Algorithm)
1991 年にスイスで開発された。128 bit 長の鍵を用いて、データを 64 bit のブロック毎に処理する。SSH1 では標準になっている。
- Blowfish
Bruce Schneier によって開発された高速の暗号。32 bit から 448 bit 長の鍵をサポートしている。SSH1 では 128 bit 長の鍵を使用し、データを 64 bit のブロック毎に処理する。

6.3 公開鍵暗号方式

共有鍵暗号方式を利用する場合、鍵の管理の問題を解決するために開発された。1976年に W. Diffie と M. Hellman により、暗号化を行う鍵と復号化を行う鍵を分離するアイデアとして提案され、その後実際のシステムが開発された。このシステムの概要は Fig. 8 の通りである。メッセージの受け手は、一組の暗号鍵と復号鍵を作成して、通信相手に暗号鍵を送る。通信相手はその鍵で暗号文を作成して、受け手に送る。受け手は、送られてきた暗号文を復号鍵で平文のメッセージに戻す。暗号文の復号化は復号鍵だけで可能であり、暗号鍵ではできないので、復号鍵を厳重に管理し他に漏れないようにすれば、暗号鍵を一般に公開して安全な暗号通信が可能である。この復号鍵を秘密鍵 (Private key) と、暗号鍵を公開鍵 (Public key) という。

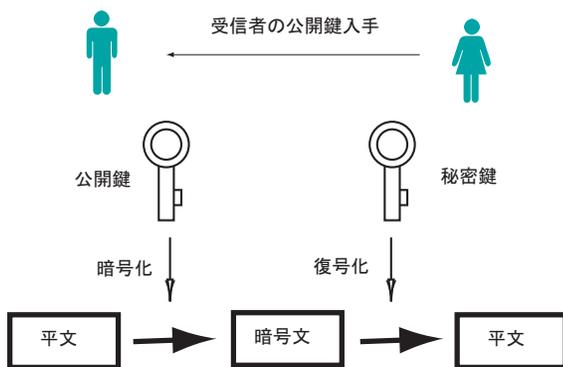


Fig. 8 公開鍵暗号方式

公開鍵暗号のシステムは、暗号通信の用途以外に、電子署名の用途に利用される。秘密鍵でメッセージを暗号化することができるのは本人だけであり、暗号化されたメッセージを受け取ったものは公開鍵で復号化して、発信者の認証を行なうことができるという特質を利用する。

SSH では公開鍵暗号システムとして RSA や DSA を使っている。

- RSA

1977年に R. L. Rivest, A. Shamir, L. Adelman の 3人によって考案された。名称は 3人の頭文字を取って付けられている。大きな整数を素因数分解することは非常に困難であるという性質を利用している。暗号化および電子署名に利用でき、SSH や PGP に利用されている。

- DSA(Digital Signature Algorithm)

1994年にアメリカ政府の標準に定められた、DSS(Digital Signature Standard) の中で定められており、電子署名のみに利用できる。SSH2では標準となっている。

公開鍵暗号システムは鍵の管理については強力ではあるが、一方、処理が遅いために、大きなデータを暗号化するのには向いていない。短いデータの暗号化に用いられることが多く、共有鍵暗号方式の暗号鍵の転送に利用される。

7 SSHでの認証

UNIX を利用する場合には、通常、パスワードを入力してユーザ認証を受ける。しかし、リモートアクセスについてホスト認証は非常に弱く、IP アドレス偽装などの不正アクセスの攻撃に弱い。

SSH は不正アクセスを防ぐために、暗号技術を使って、クライアントによるサーバホストの認証を行なう。サーバホスト認証が終了した後は、すべての通信を暗号化して通信内容の保護を行なう。続いてサーバホストによるユーザ認証が行なわれる。

この節では SSH による認証がどのように行なわれるかを説明する。ここでは、より広く使われている SSH1 を中心にして説明し、SSH2 については部分的にコメントを述べる。

7.1 サーバホスト認証

ホスト認証時には同時に、通信の暗号化を行なう共有鍵暗号システムで使用する session key の交換が行なわれる。

SSH1 によるログインでは、公開鍵暗号システム RSA を用いてサーバホストの認証および session key の交換が行われる (RSA 方式と呼ぶ)。SSH2 では RSA 方式に加えて、認証を公開鍵暗号システム DSA で、session key の交換は Diffie-Hellman 鍵配送方式で行う方式が使われており、後者が標準になっている。

ここでは SSH1 で使われている RSA 方式について説明する。RSA 方式ではホスト認証に、ホスト固有の host-key と、サーバ起動時に作られる server-key の 2つの鍵が使用される。この間の、処理の手順は以下の通りである。

1. クライアントがサーバに接続を要求する。
2. public host-key と public server-key、および自分がサポートする共有鍵暗号システムのリストをクライアントに送る。
3. クライアントは送られてきた host-key を、自分の known host key データベースに既に登録してある host-key と比較して、host-key が一致することを確認する。
4. 通信の暗号化に使う 256-bit の session key を、乱数を使って作成する。作成した session key をサーバから受け取った public host-key および public

server-key を用いて暗号化する。また、通信に使用する暗号システムを、送られてきたリストの中から選択する。暗号化した session key と選択した暗号システムをサーバに送る。

5. サーバは受け取った session key の復号化を、private host-key および private server-key を用いて行なう。復号化ができたことで、正当なサーバであるとクライアント側は判断する。

以上の手順でホスト認証が成立し、暗号化のための session key を双方で共有することができた。以下に、サーバホスト認証の過程で扱う暗号鍵を示す。

- Host key

1024-bit の RSA key あるいは DSA key でホスト認証のために使われる。SSH のサーバデーモン¹である sshd のインストール時に作成される。ssh-keygen を用いてキーの大きさを変更することができる。

SSH クライアントは、アクセスしようとする SSH サーバの public host-key を入手して自分の known host key データベースに登録する必要がある。実際には、クライアントが SSH サーバに最初にアクセスしたときに、サーバの public host-key を登録するかどうかの問い合わせがあり、確認して “yes” を答えると自動的に登録される。

Private host-key は SSH サーバだけが持ち厳重に管理する。

- Server key

Server-key は 768-bit の RSA key でホスト認証のために使われる。sshd 起動時に作成され、その後 1 時間毎に自動的に更新される。ファイルに保存されることがない。Server-key の目的は、host key で暗号化した通信を記録して解読されることを防ぐことにある。この一定時間ごとに変化する Server-key で修飾することにより暗号の解読を困難にしている。

- Session key

SSH ではセッションの通信が、共有鍵暗号方式により暗号化される。Session key はセッションの通信の暗号化に用いられる 256 ビット長の乱数である。

7.2 ユーザ認証

サーバによるユーザ認証は、4 つの方法が提供されている。

- UNIX 標準のパスワード認証

¹UNIX で、システムの機能を実現したり、何らかのバックグラウンドサービス（サーバーサービスなど）を実行するためのプロセス。OS の非常に基本的な機能から、各種用途向けのサーバまで、デーモンとして実装されていることが多い。

- 公開鍵暗号システムによる認証

- rhosts と公開鍵暗号システムを組み合わせた方式による認証

- rhosts による認証

実際に使用される認証方式は、SSH サーバの運用方針による設定と、ユーザ側の設定によって決まる。SSH サーバは、受け入れを許す設定になっている認証を順番に試みる。SSH1 では、rhosts による認証、rhosts 公開鍵認証、公開鍵認証、パスワード認証の順に試みる。SSH2 では、公開鍵認証、パスワード認証の順に試みる。例えば、ユーザが rhosts の設定も、公開鍵の設定もしていない場合には、最後のパスワード認証が行なわれ、公開鍵の設定をしている場合には公開鍵認証が行なわれる。ここでは、PortForwarder で実際に使用している、パスワード認証と公開鍵認証について解説する。

7.2.1 パスワード認証

UNIX 標準のパスワードによる認証の方法である。クライアントはユーザが入力したパスワードをサーバに送る。サーバは通常のパスワード認証ルーチンでチェックする。しかしパスワードは、session key を用いて暗号化して通信されるので、通常の telnet などより格段に安全である。

7.2.2 公開鍵暗号システムによるユーザ認証

サーバが公開鍵暗号システムを使用してユーザ認証を行なう方法である。これを利用しようとするユーザは、前もって SSH クライアントホストで ssh-keygen を使って 1 組の public-key と private-key を準備する。SSH1 では RSA key が、SSH2 では DSA key が標準になっている。

鍵を作成する時に passphrase の入力求められるが、これはファイルに保存する private-key を暗号化するためのパスワードであり、安全に保管するための措置である。passphrase には非常に長い文字列を指定できるので、単語ではなく、自分で覚えやすく、他人には分かりにくい文章となるような文字列を選ぶ。private-key は鍵を作成したホスト内だけで厳重に管理し、他のホストには転送しない。public-key は SSH サーバとする予定のホストに送り、authorized_ keys ファイルに登録する。公開鍵暗号システム RSA によるユーザ認証の手順を以下に示す。

1. ログイン要求時に、SSH クライアントはユーザ名と public-key を SSH サーバに送る。
2. SSH サーバは、そのユーザの authorized_ keys ファイルを探す。public-key を確認できると乱数を発生させ、その乱数を public-key で暗号化したデータ（challenge と呼ぶ）を作成してクライアントに送る。

3. クライアントは、private-key とユーザが入力する passphrase により challenge を復号化する。
4. 復号化したデータのチェックサムをハッシュ関数 MD5 で作成し、その計算結果をサーバに送り返す。
5. サーバ自身も乱数の MD5 を作成し、送られてきたデータと比較して、一致していればユーザ認証が成立する。

この認証方式は、private-key を持ち、そして passphrase を知っている本人だけが challenge を復号化できるということで、安全性の高い認証方式である。

8 SSH2 を用いたセキュアな環境

SSH1 では、弱いハッシュや十分でない暗号化アルゴリズムのために、脆弱性が生まれた。そこで、SSH2 が設計された。現在は SSH1 が主に使用されているが、SSH2 に移行することが考えられる。

そこで、ここでは SSH2 を用いたネットワーク環境の設定方法について述べる。

8.1 PuTTY とは

PuTTY は 1997 年から英国の Simon Tatham らによって開発されフリーの Windows 用の SSH クライアントである。商用非商用を問わず誰でも無料で使うことができ、自由に配布できる。動作環境は Windows 95/98/Me/NT/2000/XP である。

PuTTY は各機能ごとに別のプログラムになっており、Telnet および SSH クライアントである PuTTY、SCP クライアントである PSCP、SFTP クライアントである PSFTP、Telnet クライアント機能のみの PuTTYtel、PuTTY や PSCP の認証エージェントとなる Pagent、認証鍵対生成ツールの PuTTYgen がある。

8.2 PuTTY の環境設定

8.2.1 公開鍵暗号システムによるユーザ認証

ここでは、PuTTY を使うための設定について説明する。まず、マイネットワークのアドレスに ¥ ¥ museion ¥ shared ¥ softwares と入力する。

次に、putty をフォルダごと Program Files にコピーする。そして、そのフォルダの中にある putty.zip と puttykbin.zip を解凍する。

次に、putty のフォルダの puttygen.exe を起動する。

PuTTY Key Generator の画面が表示されるので、[Parameters] で SSH2 DSA をチェックする (Fig. 9)。

Generate をクリックすると鍵を作成する。このとき、この画面の上でマウスを移動させる。この移動によって自動的に乱数が作られる (Fig. 10)。

このとき、画面の Key fingerprint の部分が公開かぎになる。また、Key comment にはデフォルトでは日付

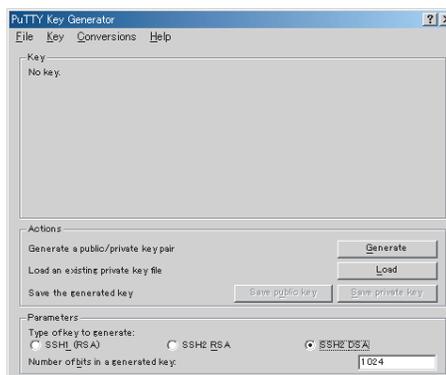


Fig. 9 鍵の作成

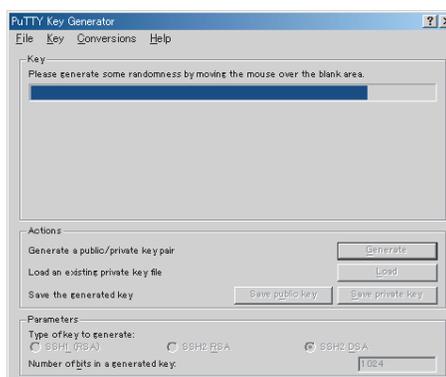


Fig. 10 乱数の作成

が入力される。

次に、Key passphrase にパスワードを各自で決め入力する。また、Confirm passphrase に同じパスワードを入力する (Fig. 11)。

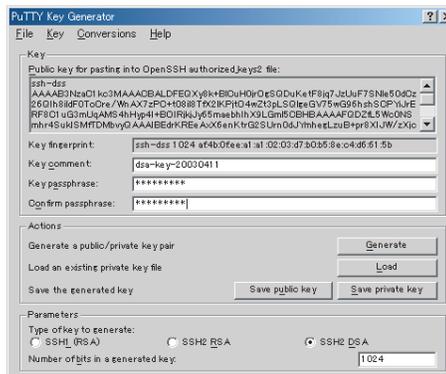


Fig. 11 passphrase の入力

次に、Save private key を押す。すると、保存の画面になるので、先ほどの putty のフォルダに保存する。名前は何でもよいが、ここでは id_dsa としておく (拡張子は .ppk となる)。

次に、ウィンドウを閉じずに作成したショートカットから putty.exe を起動する。ホスト名に

「mikilab.doshisha.ac.jp」と入力し、プロトコルでSSHを選択する。左から接続をクリックし、自動ログインのユーザ名に、申請したアカウントを入力する。

次に、セッションに戻り、保存されたセッションに保存名を入力し、保存ボタンを押す。ここでは保存名はmikilabとしておく。

次に、開くをクリックする。すると、PuTTY Security Alertのダイアログが出てくるので、「はい」をクリックする (Fig. 12)。



Fig. 12 PuTTY Security Alert のダイアログ

mikilab.doshisha.ac.jp-Putty の画面でパスワードを入力する。

次に、以前作成した.ssh というディレクトリに入るために、cd .sshと入力する。そして、vi authorized_keysと入力する (Fig. 13)。

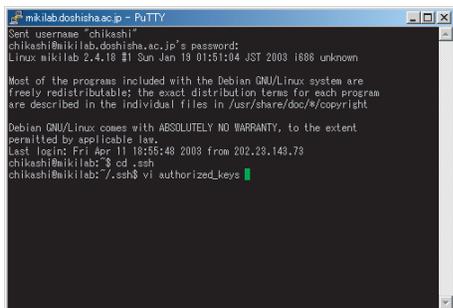


Fig. 13 公開鍵の登録

次に、PuTTY Key Generator の「Public key for pasting into OpenSSH authorized_keys2 file:」の中の文字列 (公開鍵) をすべてコピーする。この公開鍵をmikilab.doshisha.ac.jp-PuTTYに戻り、前に作成したキーの下にコピーする (Fig. 14)。

次に exit を入力し、一旦 PuTTY を終了する。また、PuTTY Key Generator を閉じる。

次に、デスクトップのショートカットを起動する。セッションで、保存されたセッションから mikilab を選択し読み込みをクリックする (Fig. 15)。

次に左から SSH を選択し、優先する SSH プロトコルバージョンで 2only(または 2) にチェックをつける。ま

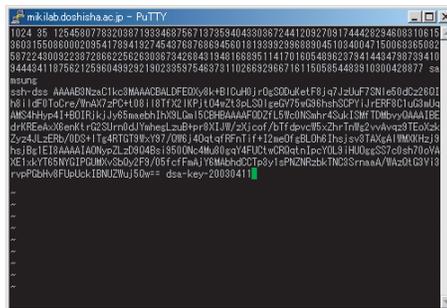


Fig. 14 公開鍵のコピー

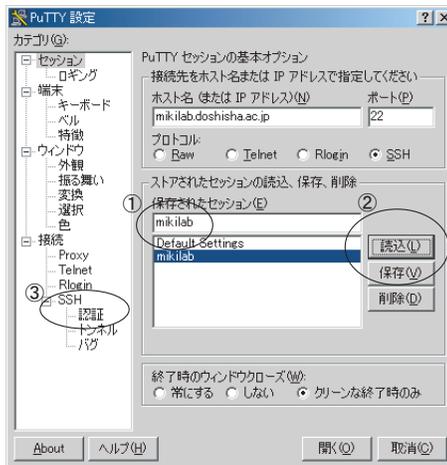


Fig. 15 Putty 設定での読み込み

た、左から認証を選択し、参照をクリックし先ほど作成した id_dsa.PPK を選択する (Fig. 15)。

次にセッションに戻り、保存をクリックする。

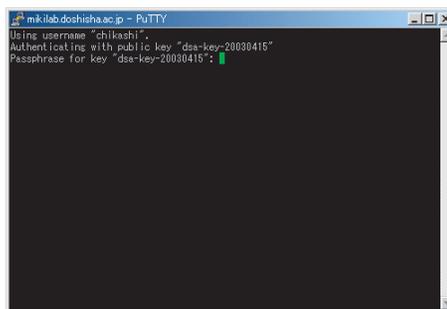


Fig. 16 Passphrase の入力画面

次に、開くをクリックする。開くと Passphrase を聞かれる (Fig. 16)。聞かれれば設定がきちんとできている。Passphrase を入力すればログイン完了となる (Fig. 17)。

8.2.2 ポートフォワーディング

次に、メールを送受信するためにポートフォワードの設定を行う。PortForwarder では SSH2 をサポートしていないため、PuTTY で行う。まず、puttyjp.exe を起動



Fig. 17 ログイン完了画面

する。次に、セッションで mikilab を読み込む。トンネルを左から選択し、源ポートに 8110 を入力し、送り先に mikilab.doshisha.ac.jp:110 と入力し、追加をクリックする (Fig. 18)。

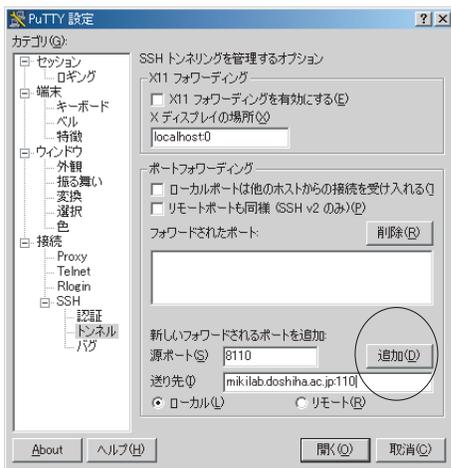


Fig. 18 ポートフォワーディングの設定 1

追加ボタンを押すと、Fig. 19 のような画面になる。次に、源ポートに 8025、送り先に mikilab.doshisha.ac.jp:25 を入力し、追加をクリックする Fig. 20。

また、追加すると Fig. 21 のような画面になる。次に、開くをクリックし、Passphrase を入力しログインする。これでポートフォワーディングされ、メールの送受信などができる。

8.3 WinSCP

8.3.1 WinSCP の利用

Windows 上で簡単に SCP を用いる方法として、WinSCP と呼ばれるソフトがある。ここではその設定について述べる。

まず、¥ ¥ museion ¥ shared ¥ softwares に移動し、フォルダの中の winscp のフォルダをコピーして、Program File などにコピーする。

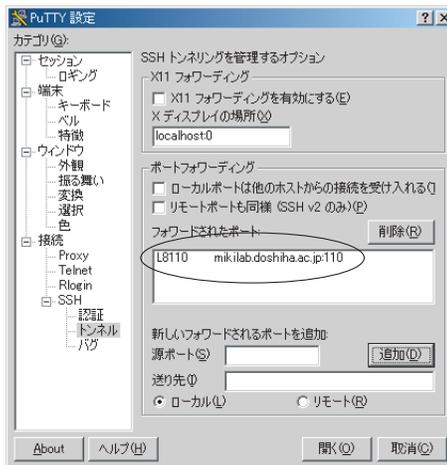


Fig. 19 ポートフォワーディングの設定 2

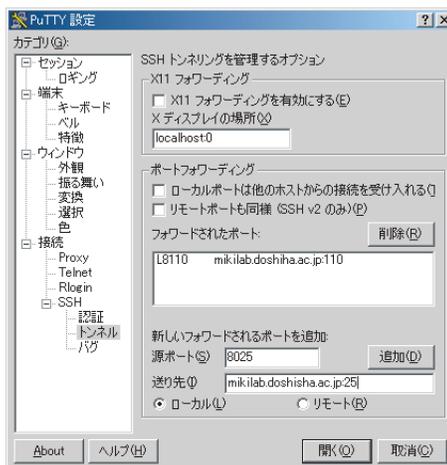


Fig. 20 ポートフォワーディングの設定 3

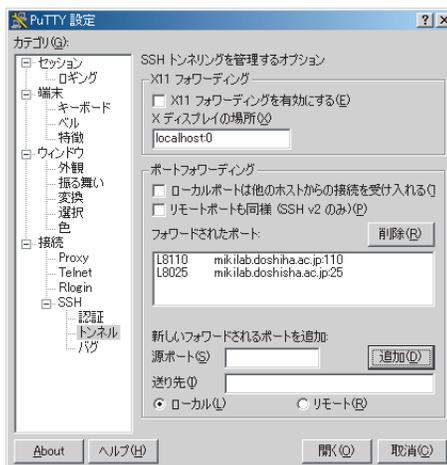


Fig. 21 ポートフォワーディングの設定 4

フォルダを開き、WinSCP2.exe のショートカットを作成し、ショートカットを開く。

Fig. 22 のように、Host name に mikilab、User name に自分のアカウント、Private key file に先ほど作成した秘密鍵を選択する (putty のフォルダの中の id_dsa.PPK というファイル)。なお、ここではリモートホストを mikilab として説明している。

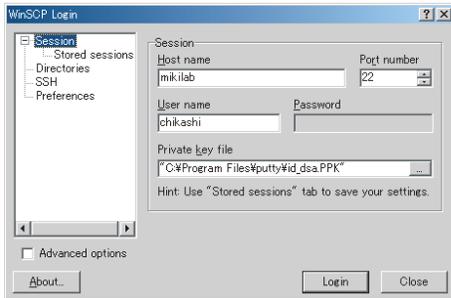


Fig. 22 WinSCP のログイン画面



Fig. 25 パスワード入力画面

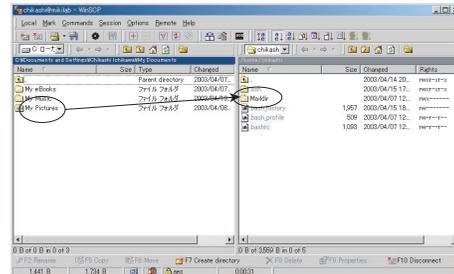


Fig. 26 WinSCP の画面

次に、Fig. 23 のように SSH で 2only を選択する。

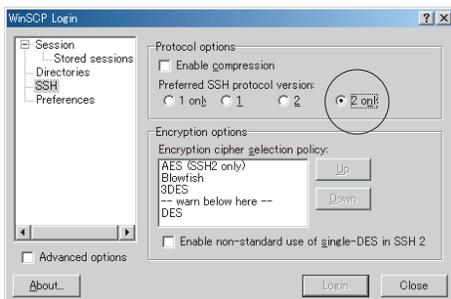


Fig. 23 SSH で 2only を選択

そして、Stored sessions で Save を選択する。セッションの名前を聞かれるので、セッション名を入力し、OK を押すと設定が保存される (Fig. 24)。

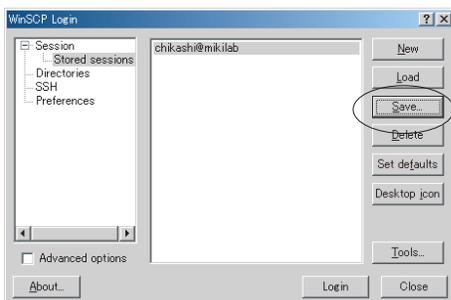


Fig. 24 設定を保存

login を押すと Passphrase を聞かれるので入力する。これでセッションが確立される (Fig. 25)。

ドラッグアンドドロップして、ファイルを転送することができる (Fig. 26)。

WinSCP は、Fig. 26 のような左側がローカルホストのフォルダで右側がリモートホストのディレクトリというような一般的な FTP クライアントソフトと同じの画面構成となっており、Windows ユーザに親しみやすいインターフェースになっている。

WinSCP を使えばファイルの転送以外にもファイル名の変更、ファイルの移動、ファイルの削除、ディレクトリの作成、パーミッションの設定といったことを GUI 上で簡単に行うことができ、非常に便利である。

以上で、暗号化された安全なファイル転送が可能となる。

9 LP-8700PS3 の設定

知的システムデザイン研究室では、Windows ユーザに対しては、印刷スピードが速く給紙カセットも大きいゼロックスの「DocuCentre 280」を使うことを推奨している。しかし「DocuCentre 280」が故障の場合、あるいは印刷が混み合っている場合なども考えられるので、その際は「LP-8700PS3」を使う。

まずはじめに [\\~~xxx~~\\museion\\~~xxx~~shared\\~~xxx~~softwares\\~~xxx~~driver] にある LP-8700PS3 というフォルダを、フォルダごと自分の PC にコピーする (Fig. 27)。LP-8700PS3 のフォルダ内の SETUPW2K.CMD を実行する (Fig. 28)。すると Fig. 29 のようなウィンドウが現れるので、LP-8700PS3 を選び、次に進む。Fig. 30 の画面になるので検索中止ボタンをクリックする。Fig. 31 の画面が出力されるので、OK ボタンをクリックする。

次にポートの設定を行う。マイコンピュータのコントロールパネルからプリンタのアイコンを選択し、EPSON LP-8700PS3 を右クリックし、プロパティを選ぶ。

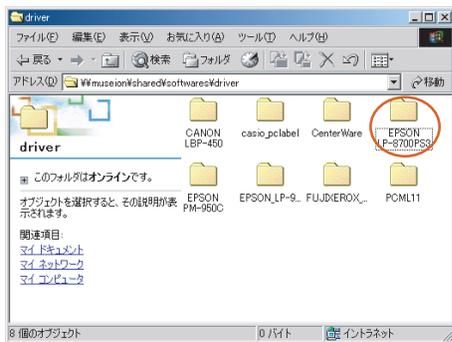


Fig. 27 \\museum\shared\softwares\driver

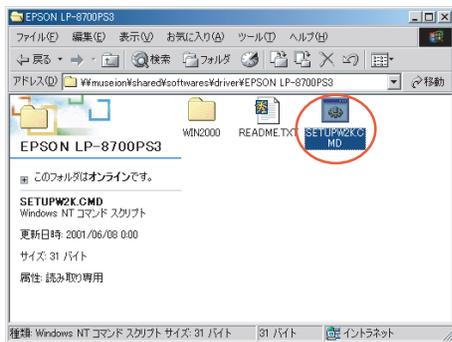


Fig. 28 SETUPW2K.CMD を実行



Fig. 29 LP-8700PS3 を選択

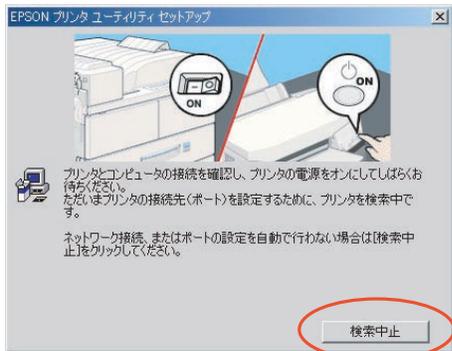


Fig. 30 検索中止をクリック

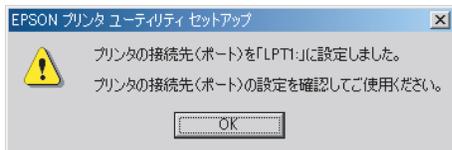


Fig. 31 ポートの仮設定

そこでポートのタブを選び、Fig. 32のようにポートの追加をクリックすると、Fig. 33の画面が現れるので、新しいポートをクリックする。TCP/IPプリンタポートの追加ウィザード開始の画面になるので (Fig. 34)、次へをク

リックする。Fig. 35のウィンドウが表示されるので「プリンタ名またはIPアドレス」入力欄に、192.168.6.20と入力して、次へをクリックする。Fig. 36のように追加ウィザードの完了の画面になるので完了ボタンをクリックする。Fig. 33の画面に戻るので閉じる。そして (Fig. 37)の画面で必ず先程追加したTCP/IP PORTの行を左クリックし、その後適用ボタンをクリック、そしてOKボタンをクリックする。

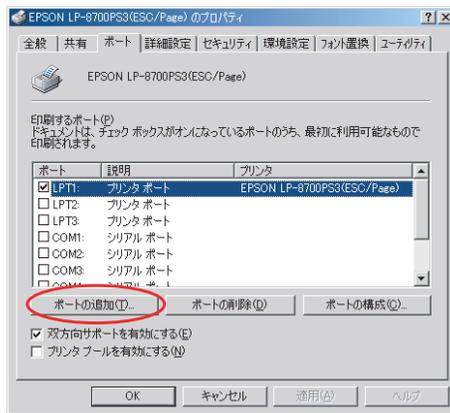


Fig. 32 ポートの追加



Fig. 33 新しいポートの追加

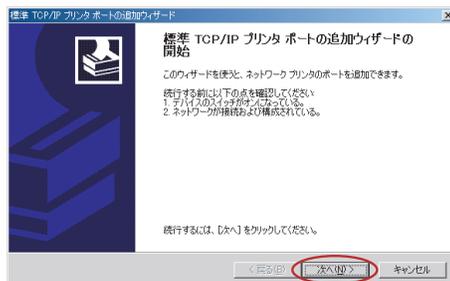


Fig. 34 追加ウィザードの開始画面

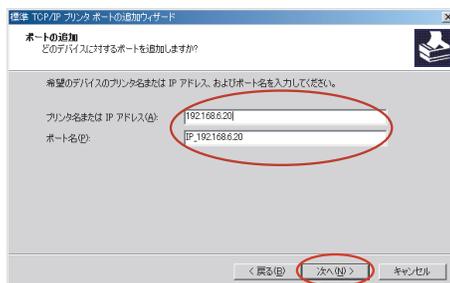


Fig. 35 IP アドレスの入力

ドライバをインストールすると、通常使うプリンタがEPSON LP-8700PS3 になるのでそれ



Fig. 36 追加ウィザードの完了画面

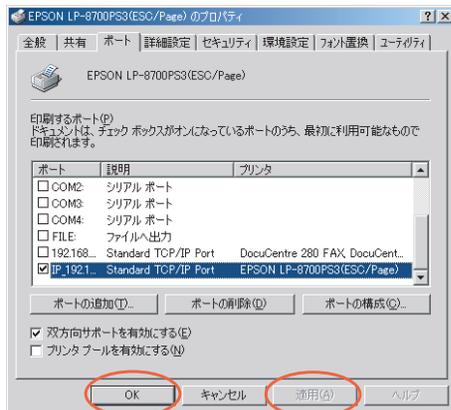


Fig. 37 ポートの選択画面

をDocuCentre 280 に戻す . マイコンピュータのコントロールパネルからプリンタのアイコンを選択し , DocuCentre 280 を右クリックし , 通常使うプリンタに設定をクリックする . 以上で EPSON LP-8700PS3 に関する環境設定は完了する .