SSHの
仕組みと利用

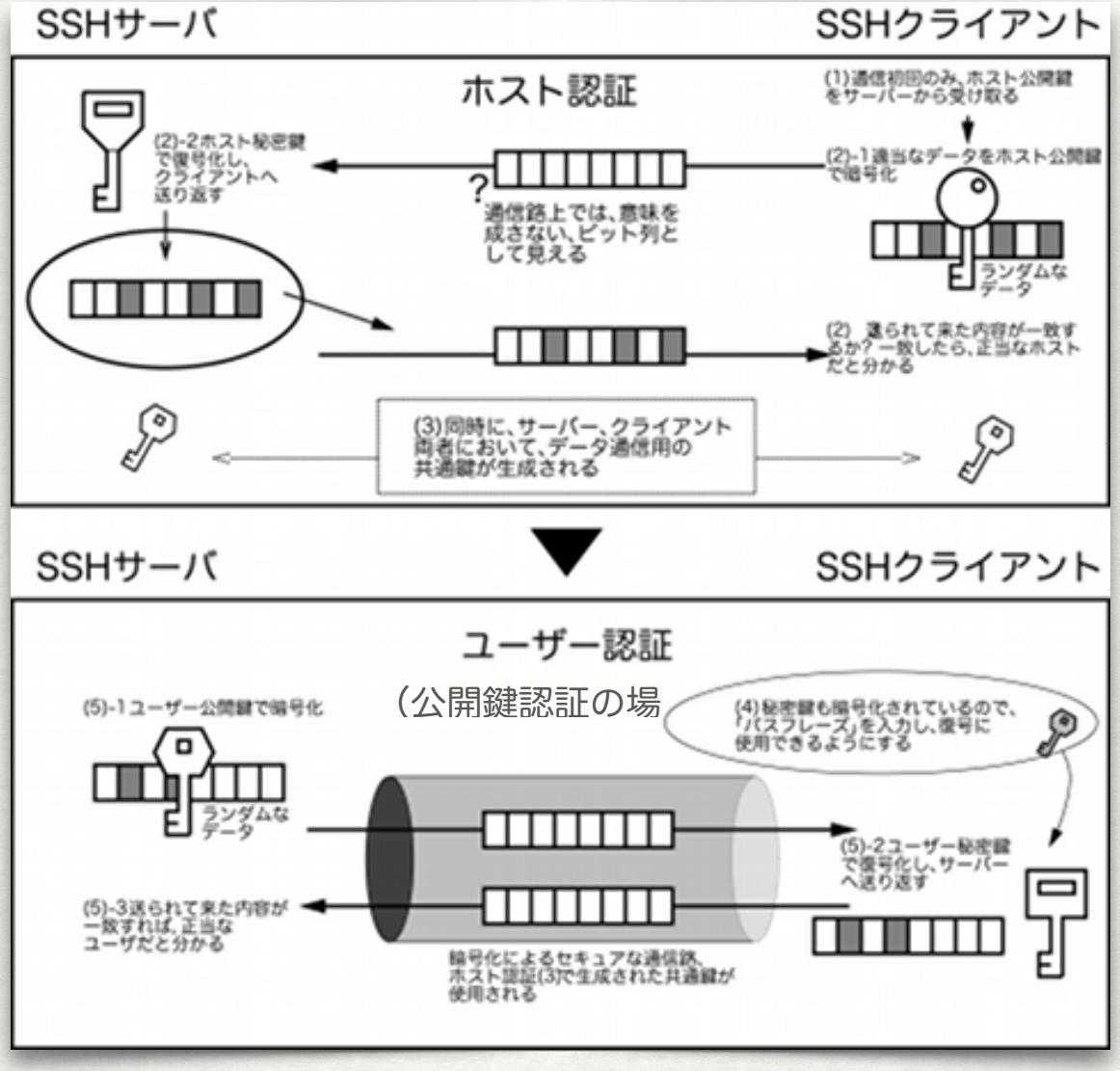
SSH2の仕組み

SECURE SHELL V2

(1

(2

(3



https://www.omoikane.co.jp/man30/w_ssh.htmlから引用

SSH2

(1)ホスト認証

1. 初めて接続するサーバなら、クライアントはサーバからそのホスト公開鍵を受けとり、鍵リスト ~/.ssh/known_hosts に加える
2. クライアントは、ランダムデータを作成，接続先ホストの公開鍵で暗号化してサーバに送る
3. サーバは自分のホスト秘密鍵で復号，ハッシュ値を求めクライアントに返す
4. クライアントは、送信したランダムデータのハッシュ値とホストからの返信が一致すれば，正しいサーバと接続していると認める (ホスト認証成功)

SSH2

(2)DIFFIE-HELLMAN鍵共有

1. クライアントとサーバで数 n, g を共有する
2. クライアントは秘密の値 a を生成し，
   $K_a = g^a \ mod \ n$ をサーバに送る
3. サーバは秘密の値 b を生成し，
   $K_b = g^b \ mod \ n$ をサーバに送る
4. クライアント，サーバで共通鍵
   $K = g^{ba} \ mod \ n = g^{ab} \ mod \ n$ を計算し共有する
5. 以降，共通鍵を用いた暗号化通信を行う

SSH2

(3)ユーザ認証

1. パスワード認証（非推奨）
    1. ホストはクライアントにユーザパスワードを聞く
    2. クライアントはパスワード（またはそのハッシュ値）をサーバに返す
    3. サーバ側でパスワードが正しいか確認する（ユーザ認証成功）
2. 公開鍵認証（推奨）
    1. ホストはランダムデータを生成し，あらかじめ保存してあるユーザの公開鍵で暗号化してクライアントに送る
    2. クライアントはパスフレーズで有効化した自分の秘密鍵で復号し，そのハッシュ値を返す
    3. ホストはランダムデータのハッシュ値と比較し同じであれば正当なユーザであると見なす（ユーザ認証成功）

SSH2

ホスト認証用鍵セットの生成（ホスト側）

1. 以下のコマンドを発行する
   - ssh-keygen -t <暗号化タイプ> -f <秘密鍵ファイル名> -N
   - 暗号化タイプ: rsa1, rsa, dsa
   - 秘密鍵ファイル名: ssh_host_rsa_key (公開鍵は .pub の拡張子がつく)など
   - -N: 秘密鍵を暗号化しない

2. 例
   - ssh-keygen -t rsa -f /etc/ssh_host_rsa_key -N

SSH2

ユーザ認証用鍵セットの生成（クライアント側）

1. 以下のコマンドを発行する
   - `ssh-keygen -t <暗号化タイプ>`
   - 暗号化タイプ: `rsa1`, `rsa`, `dsa`
   - 秘密鍵ファイルは `~/.ssh/` 以下に自動的に作成される
   - `rsa` の場合： `id_rsa`, `id_rsa.pub` が生成される
2. 公開鍵をホストに転送する（公開鍵認証の場合）
   - 転送する公開鍵 `~/.ssh/id_rsa.pub` など
   - ホストにログインして上記公開鍵を `~/.ssh/authorized_keys` に追加する（無ければ作成）

SSH2の利用

ログイン，X WINDOW

1. ログイン
   - `ssh` *user@hostname [コマンド]*
   - `ssh -l` *user hostname [コマンド]*
2. X Windowの利用
   - `ssh -X` *user@hostname [コマンド]*
   - X Window の設定が必要

SSH2の応用

SCP, SFTP

1. scp：ssh をベースにしたホスト間ファイルコピー
   - `scp [-p] [-r] [-C] <ユーザ名>コピー元(複数可)> <コピー先>`
     - `-p`：コピー元のタイムスタンプ・所有者・パーミッションを保持
     - `-r`：ディレクトリの中身をまとめてコピー
     - `-C`：通信内容を `gzip` 圧縮（ssh と同じ）

2. sftp：ssh をベースにしたホスト間複数ファイル転送
   - `sftp [-C] [<user>@]<hostname>`
     - `-C`：通信内容を `gzip` 圧縮（ssh と同じ）
     -

SSH2の応用

RSYNC

1. rsync：ssh をベースにしたファイル同期
   - rsync [<オプション>] <コピー元（複数可）> <コピー先>
     - -r：ディレクトリの中身をまとめてコピー
     - -u：新しいファイルのみコピー（アップデート）
     - --delete：コピー元にないファイルはコピー先から削除
     - -l, -H：シンボリックリンク・ハードリンクを保持
     - -L：シンボリックリンクを参照するファイルに変換
     - -D：デバイスファイル属性を保持（要 root 権限）
     - -o, -g：所有者・所有グループを保つ（-o は要 root 権限）
     - -p：パーミッションを保つ
     - -t：タイムスタンプを保つ
     - -a：-rlpogtD と等しい（要 root 権限）
     - -x：コピー元でファイルシステムをまたがった処理をしない
     - -n：処理内容を表示するだけで実際には処理しない

SSH2の応用

パスフレーズの省略

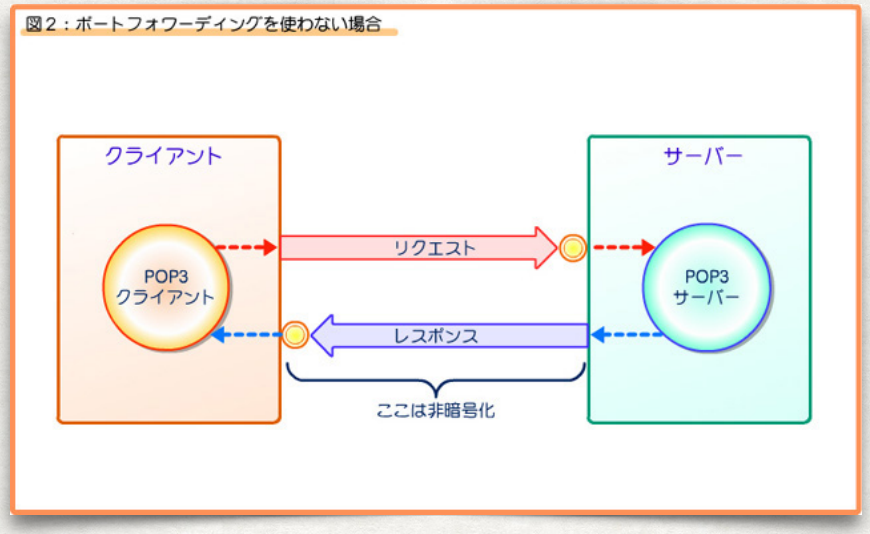
1. ssh-agent コマンドでシェルを起動する
   - `ssh-agent シェル名`
   - **例:** `ssh-agent /bin/bash`
2. ssh-add コマンドで利用する秘密鍵ファイルを指定する
   - `ssh-add 秘密鍵ファイル名`
     - この**時だけパスフレーズを聞いてくる**
   - **例:** `ssh-add .ssh/id_rsa`
3. 以降，シェルを終了するまでは ssh, scp, sftp 等のコマンドでパスフレーズを聞いてこなくなる

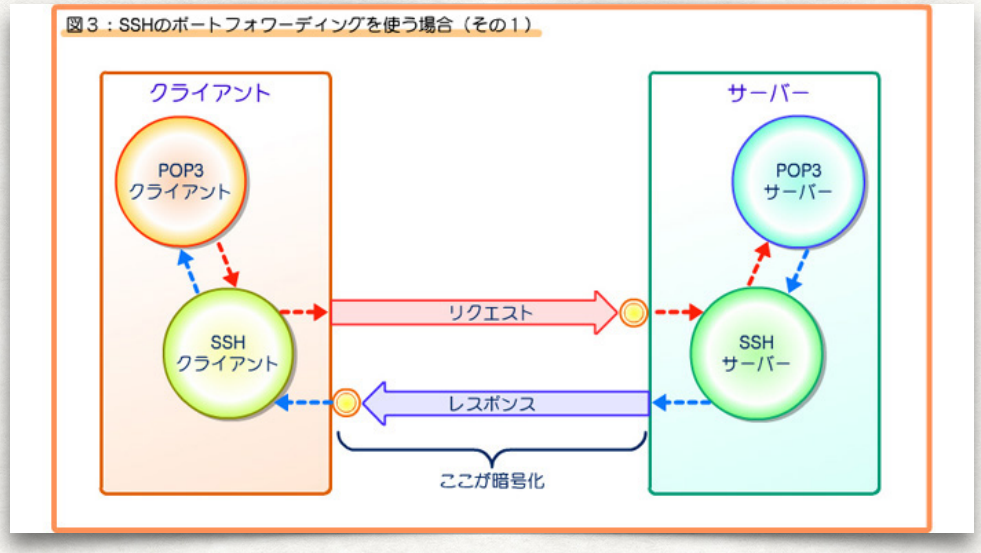
SSH2の応用

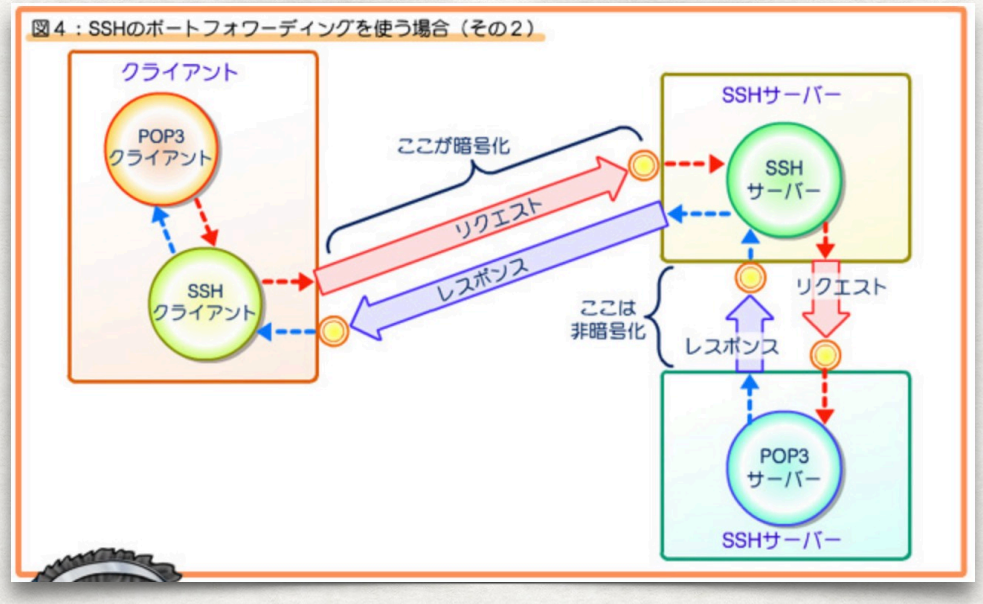
ポート・フォワーディング

通常暗号化されない通信を，sshプロトコルをトンネルとして暗号化してしまう

http://bb.watch.impress.co.jp/cda/bbword/8233.html から引用







SSH2の応用

ポート・フォワーディング

1. ssh 接続を経由して他のサービスに接続する
   - `ssh -L　ローカルポート番号:サービスホスト名:リモートポート番号　ユーザ名@SSHサーバ名`
   - 例:　`ssh -L 10110:mail.example.com:110 mail.example.com`
   - 

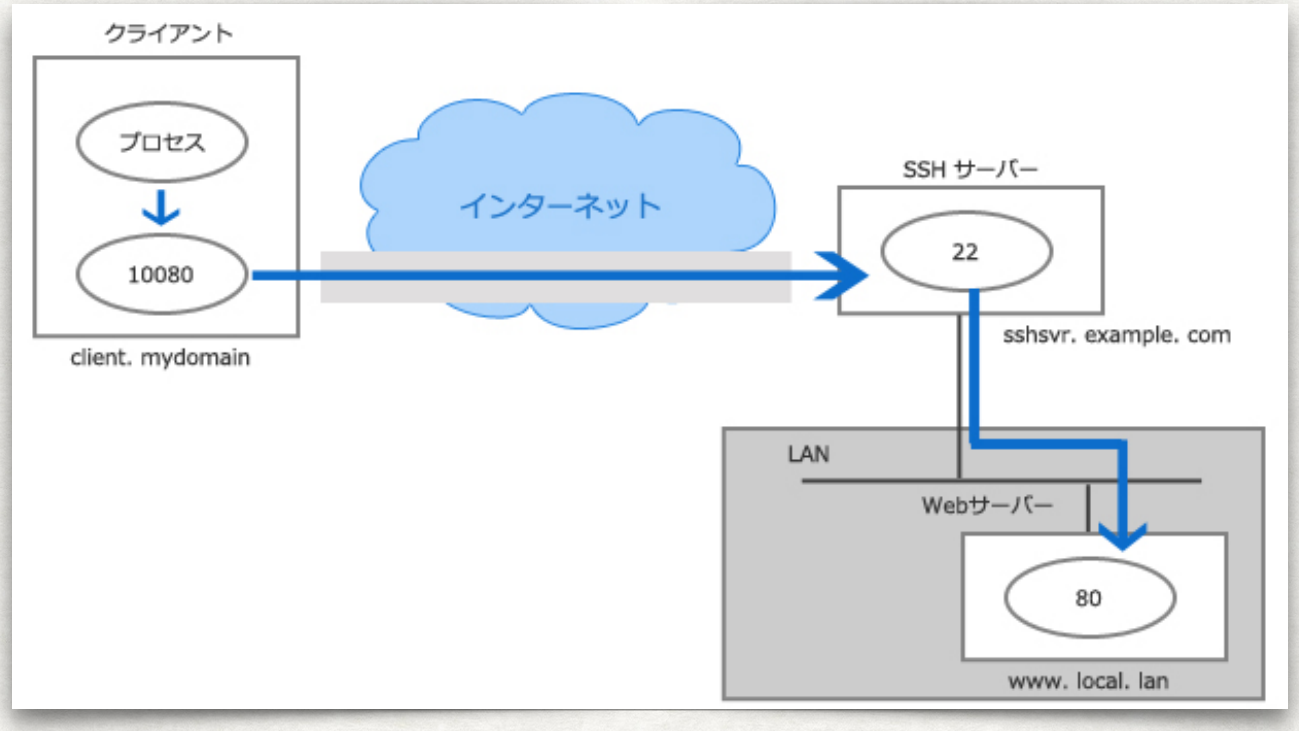
引用元 http://www.turbolinux.co.jp/products/server/11s/user_guide/x9016.html

SSH2の応用

ポート・フォワーディング

1. ssh 接続を経由して他のサービスに接続する

- 例:
  ```
  ssh -L 10080:www.local.lan:80 sshsvr.example.com
  ```



- 引用元 http://www.turbolinux.co.jp/products/server/11s/user_guide/x9016.html