ITPASS
Informational Training program
with a spirit of self-help

Network Computing & Internet Security

服部 蒼紀
神戸大学理学研究科惑星学専攻 M2

2024 年 9 月 5 日

ITPASS 実習 3 日目

ここで話すこと

- ✓計算機はネットワークを介してどのように情報をやり取りしているのか?
- ✓離れた計算機との通信にはどのような種類があるか?
- ✓インターネットセキュリティの原則
- ✓安全に情報をやり取りするための仕組みは?

目次

- Network Computing の基礎
  - サーバ, クライアント, ポート, デーモン
- 最低限リモートアクセス
  - リモートログイン, ファイル転送
- インターネットセキュリティ
  - 被害に遭わないために, ユーザーを守るために
- 公開鍵暗号による通信と ssh 認証

計算機の呼び方

- ホスト
  - 個々の計算機
  - それぞれにホスト名 (hostname) が存在する
- ローカルホスト
  - 手元で操作している計算機
  - IPv4 では “127.0.0.1”
  - IPv6 では “::1” が割り当てられている
- リモートホスト
  - ネットワーク上に存在する計算機 (ローカルではない)

目次

- Network Computing の基礎
  - サーバ, クライアント, ポート, デーモン
- 最低限リモートアクセス
  - リモートログイン, ファイル転送
- インターネットセキュリティ
  - 被害に遭わないために, ユーザーを守るために
- 公開鍵暗号による通信と ssh 認証

Network Computingとは？

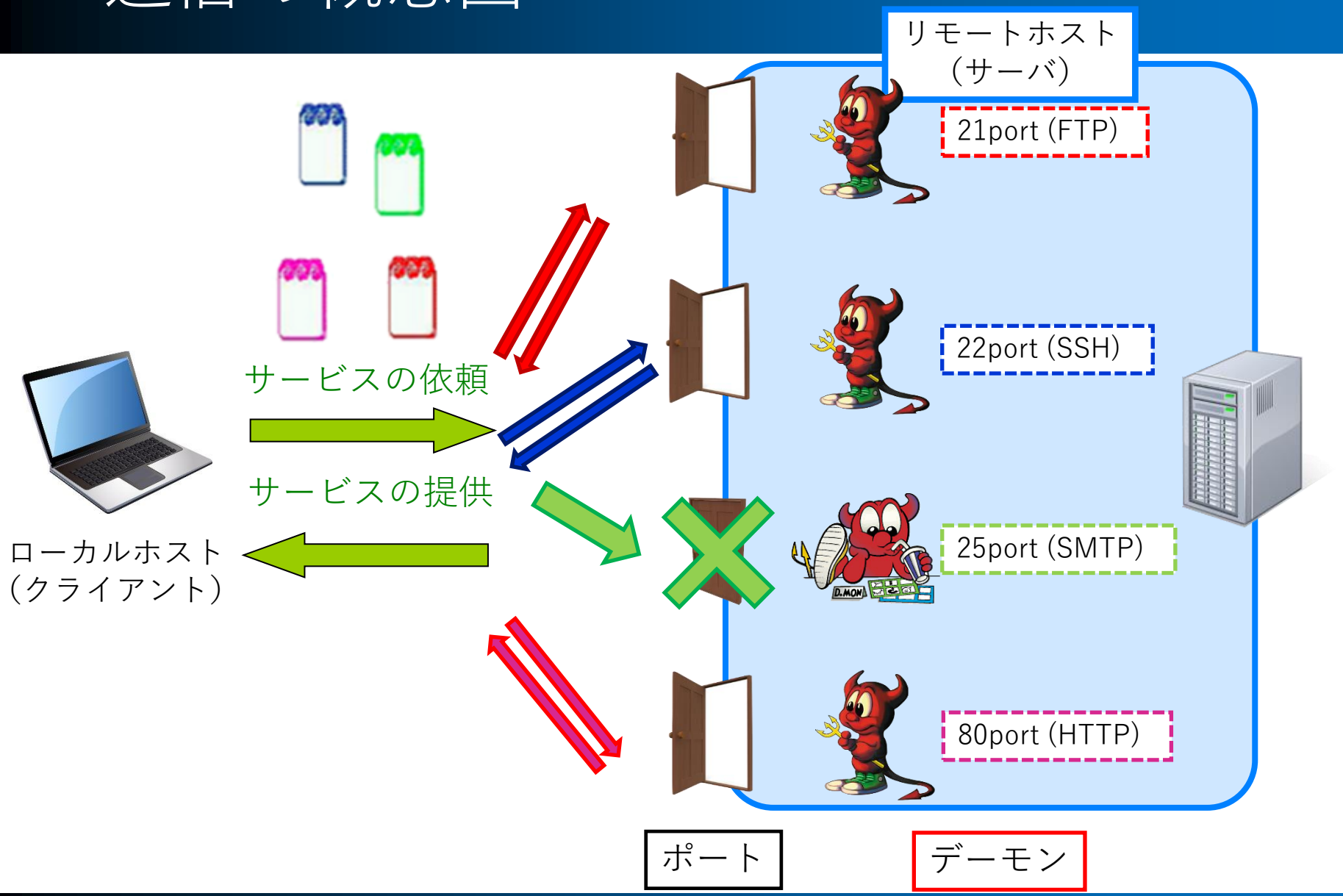
- ここでは，「ネットワークを介した計算機同士の情報のやり取り」の意味で使う．
- 例えば，
  - ftp
  - telnet, ssh
  - Mail の送受信
  - WWW

  …等々は全て “network computing”

Network Computing 登場人物

- サーバ (server)
  - ネットワーク上でサービスを提供する側の計算機またはプログラム
- クライアント (client)
  - ネットワーク上でサービスを提供される側の計算機またはプログラム
  - サーバに対して接続し, サービスの提供を受ける
- ポート (port)
  - サーバで情報を送受信する窓口
    - どのサービス宛の情報なのかを指定する番号 = ポート番号
- デーモン (daemon)
  - ポートを監視し, 要求に応じてサービスを提供するプログラム

通信の概念図



ポートの種類

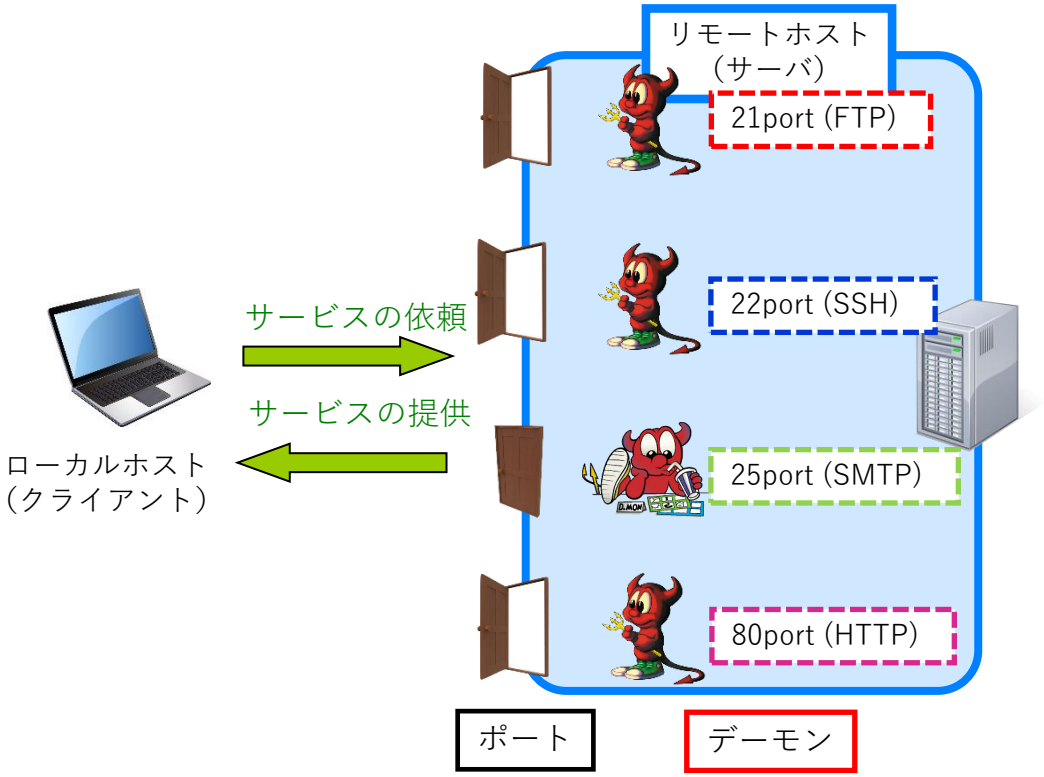
- 割り当て済みのポート (0～49151)
  - IANA (**I**nternet **A**ssigned **N**umbers **A**uthority, http://www.iana.org/) が管理
  - well-known port (0～1023)
    - 主要なサービスのため IANA が割り当て済みのポート
      - 例えば, 21: ftp,　22: ssh,　23: telnet,　80: http など
    - privileged port / 特権ポートとも呼ばれる
  - registered port (1024～49151)
    - アプリケーション用にベンダーが申請したポート
- 特定の役割に割り当てられないポート (49152～65535)
  - dynamic and/or private port
    - クライアント等が動的に使用するためのポート

主なサービスとそのポート番号

- リモートアクセス
  - 端末・プロセスとの通信: TELNET (Telecommunication Network) (23)
  - ファイル転送: FTP (File Transfer Protocol) (20, 21)
  - 暗号化通信: SSH (Secure Shell) (22)
- WWW サービス
  - HTTP (Hyper Text Transfer Protocol) (80)
  - HTTPS (HTTP over SSL (Secure Socket Layer)) (443)
- Mail サービス
  - 送信 / 転送
    - SMTP (Simple Mail Transfer Protocol) (25)
  - 受信
    - POP3 (Post Office Protocol ver.3) (110)
    - IMAP4 (Internet Message Access Protocol ver.4) (143)

Network Computing の基礎まとめ

- クライアント:
  - サーバのポートへ接続し, サービスを受ける
- サーバ:
  - デーモンを使ってポートを監視し, クライアントの依頼を受け, サービスを提供

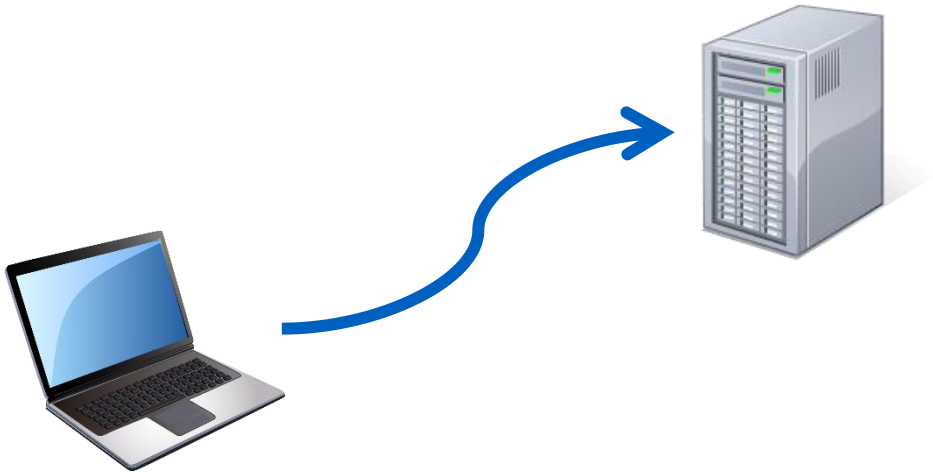


目次

- Network Computing の基礎
  - サーバ, クライアント, ポート, デーモン
- 最低限リモートアクセス
  - リモートログイン, ファイル転送
- インターネットセキュリティ
  - 被害に遭わないために, ユーザーを守るために
- 公開鍵暗号による通信と ssh 認証

リモートログイン

- ローカルホストからリモートホストへネットワーク経由でログインすること
  - 遠隔地のコンピュータにリモートログインすることによって, そのコンピュータを, 目の前にある時と同じように直接操作することができる.
- 代表的なコマンド
  - telnet
  - rlogin
  - slogin
  - ssh



リモートログインのイメージ

ホスト名: joho03
アカウント名: hoge

ssh コマンドを用いてリモートログインを要請

ログインパスワードを要求

ホスト名: joho14
アカウント名: hoge

```
hoge@joho03:~$ ssh hoge@joho14
hoge@joho14's password:
```

リモートログインのイメージ

ホスト名: joho03
アカウント名: hoge

ログインパスワードを
入力

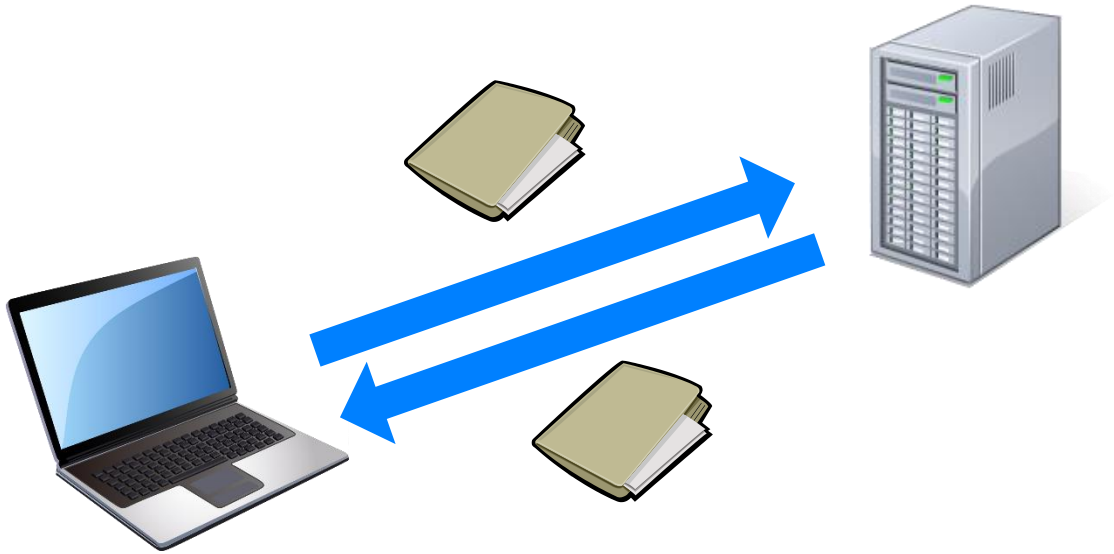
ログインを許可

ホスト名: joho14
アカウント名: hoge

```
hoge@joho03:~$ ssh hoge@joho14
hoge@joho14's password: （パスワードを入力）
…
hoge@joho14:~$
```

ファイル転送

- ローカルホストとリモートホストの間でネットワークを介してファイルを転送すること
- 代表的なコマンド
  - ftp
  - scp
  - sftp



リモートアクセスを用いたファイル転送のイメージ

ホスト名: joho03
アカウント名: hoge

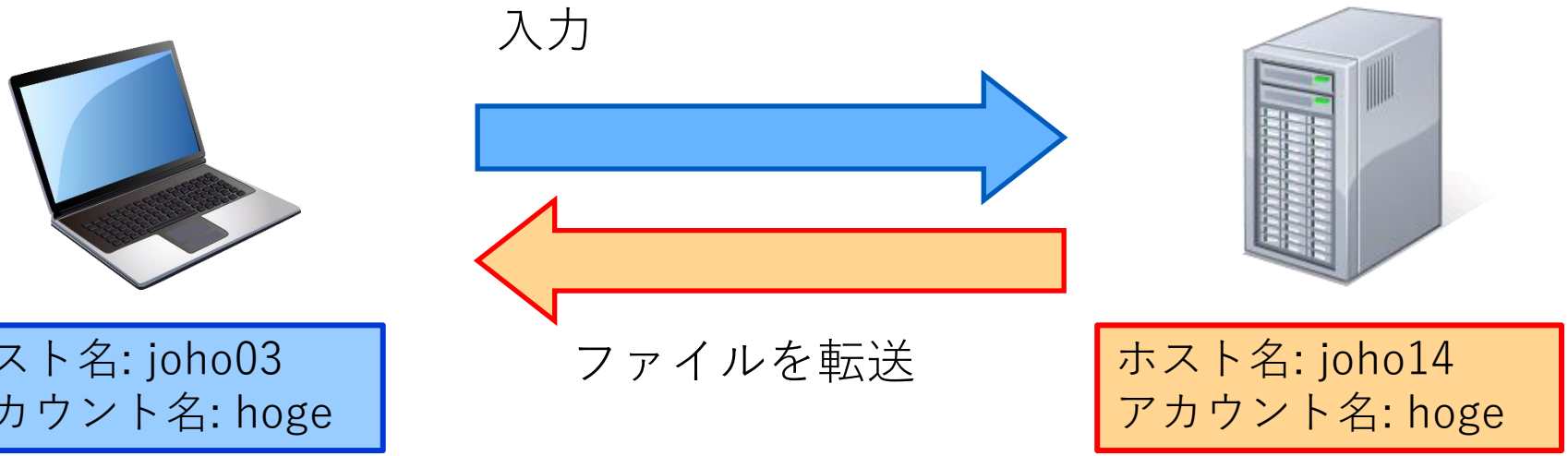
scp コマンドを用いて
ファイル転送を要請

ログインパスワード
を要求

ホスト名: joho14
アカウント名: hoge

```
hoge@joho03:~$ scp joho14:/home/itpass/hoge.txt ./
hoge@joho14's password:
```

リモートアクセスを用いたファイル転送のイメージ




```
hoge@joho03:~$ scp joho14:/home/itpass/hoge.txt ./
hoge@joho14's password: （パスワードを入力）
hoge.txt                      100%   14KB  14.0KB/s   00:00
hoge@joho03:~$ ls
hoge.txt
```

リモートアクセスの仕組み

- リモートログイン, リモートコマンドの実行, ファイル転送の際に, サーバの決められた番号のポートにアクセス
- 通信では, それぞれ決められたプロトコル(通信規約) を使用
  - 代表的なアプリケーション層のプロトコル(TELNET, FTP, SSH)について述べる

TELNET （ポート番号 23）

- **Tel**ecommunication **Net**work
  - ネットワークを介して端末間およびプロセス間の通信を提供するためのプロトコル
    - この規約ができたことでリモートアクセスが可能になった
    - 端末間通信では, ログインしたあとにシェルを起動する
- 短所
  - データがそのまま（平文のまま）ネットワークを流れる
    - パスワードも平文のまま流れる
- コマンド
  - telnet

FTP （ポート番号 20, 21）

- **F**ile **T**ransfer **P**rotocol
  - ホスト間でファイル転送を行うためのプロトコル
    - データ転送用: 20 番
    - 制御用: 21 番
- 短所
  - データがそのまま（平文のまま）ネットワークを流れる
    - パスワードも平文のまま流れる
- コマンド
  - ftp

SSH （ポート番号 22）

- **S**ecure **Sh**ell
  - リモートホストにログインしたり, コマンドを実行したり, ファイルを転送するためのプロトコル
  - 通信内容が暗号化される
    - パスワードが盗聴されても解読できない
- 短所
  - 暗号化される分, 処理時間が（少しばかり）長くなる
  - パケットサイズも（少しばかり）大きくなる
- コマンド
  - ssh, slogin (=secure login), scp (=secure copy)

Windows の場合…

- telnet, ftp
  - 最初から実装されている
    - ただし, telnet については, Windows Vista 以降では, 有効化する必要がある
- Windows に含まれないクライアント
  - リモートログイン
    - MobaXterm
    - Tera Term
    - PuTTY
    - …
  - ファイル転送
    - MobaXterm
    - WinSCP
    - FFFTP
    - …

Mac の場合…

- telnet, ftp
  - ターミナルアプリケーションに付属していない
  - 2017年にコマンドが廃止された

- ssh
  - 「ターミナル」アプリで ssh コマンドを入力すると使える

リモートアクセスまとめ

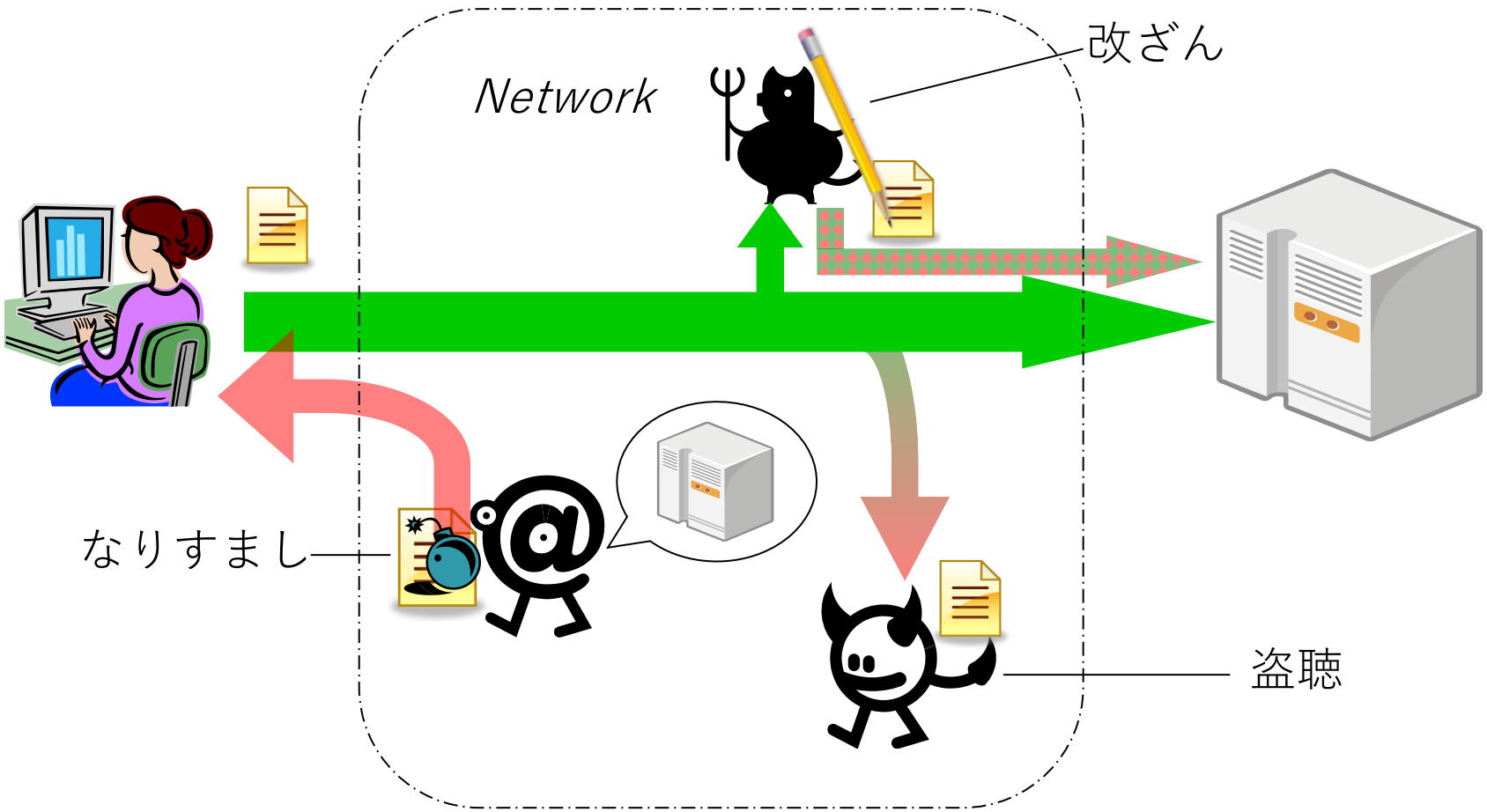
- リモートアクセスの種類
  - リモートログイン, ファイル転送
- TELNET
  - リモートログインのためのプロトコル
  - 通信内容は暗号化されない
- FTP
  - ファイル転送のためのプロトコル
  - 通信内容は暗号化されない
- SSH
  - リモートログイン, ファイル転送などに使用するプロトコル
  - 通信内容が暗号化される

TELNET, FTP は使わないように

目次

- Network Computing の基礎
  - サーバ, クライアント, ポート, デーモン
- 最低限リモートアクセス
  - リモートログイン, ファイル転送
- インターネットセキュリティ
  - 被害に遭わないために, ユーザーを守るために
- 公開鍵暗号による通信と ssh 認証

インターネット上の脅威



脅威は，大きくわけて 3 種類

ネットワークセキュリティの原則 ～一般ユーザー編：被害に遭わないために～

- 有害なデータを受け取らないように予防する
  - 不要なアプリケーションのインストールはしない
  - セキュリティホールがあるとわかっているソフトウェアはインストールしない/アップデートする
  - メールの添付ファイルを確認せずに開いたり，書かれている URL に無闇にアクセスしたりしない
    - 例: JTB の情報流出（ http://mainichi.jp/articles/20160616/k00/00m/040/042000c ）…「企業の情報を盗むため、取引先を装ってウイルス感染させる『標的型メール』が送りつけられたのが原因」
- パケット盗聴の予防策を講じる
  - 暗号化通信プロトコル(SSH, SSL/TSL) を用いた通信を利用する

SSL/TLS

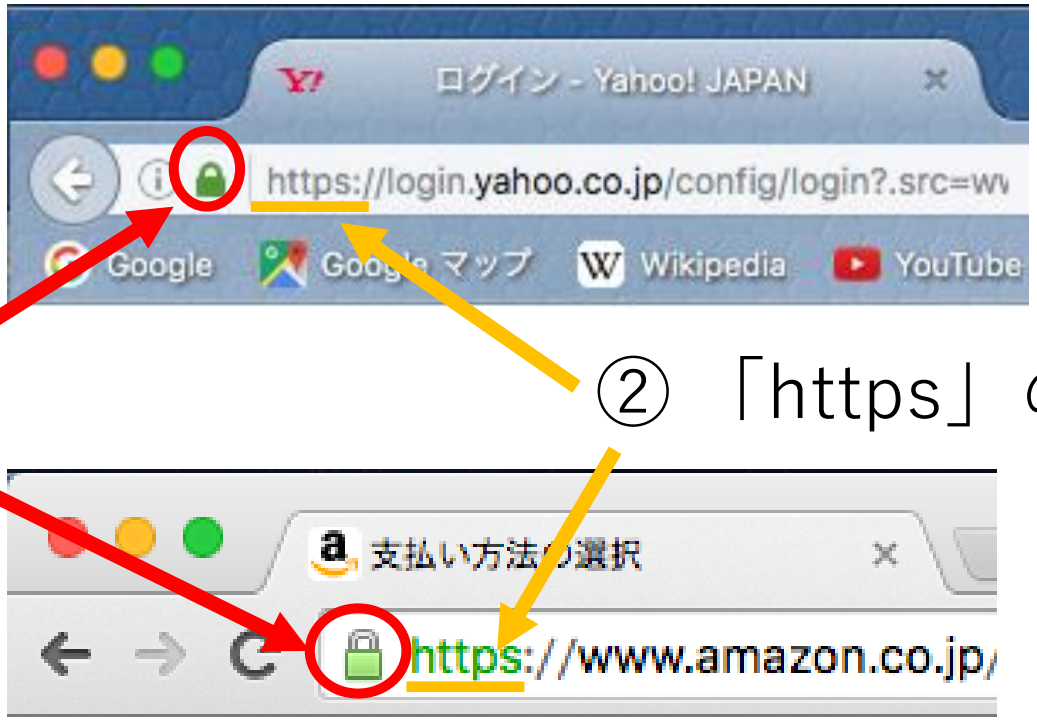
- Secure Socket Layer / Transport Layer Security
- 通信データを暗号化するために利用
- アプリケーション層とトランスポート層の間のプロトコル
  - TCP の代替として使用可能
- HTTP と組み合わせて利用：HTTPS (HTTP over SSL/TLS)
  - Web サイトで認証情報や個人情報, 決済情報などを安全にやり取りするために利用

TCP/IPの階層
アプリケーション層
トランスポート層
インターネット層
リンク層

SSL サーバ証明書

- Web サイトの身元の証明と SSL による通信の暗号化に使われるデジタル証明書
  - ウェブサイト所有者の確認：ページなどを管理する人/組織が実在し, 信頼に足ることを証明
  - 適切な SSL/TLS を利用した通信であることを証明
  - 通信の「なりすまし」「盗聴」「改ざん」を防ぐ
- 認証局（CA：Certificate Authority）が発行
  - 通信しようとする相手の身元を発行元の認証局に照会し, 確認することができる

HTTPS 通信の目印



ログイン画面や決済方法を入力する画面では，
以上のようになっていることを確認してから
入力・送信を行うこと!

怪しい SSL 証明書

- 信頼できる CA が発行したわけではない証明書
  - 技術的には誰でもCAになってSSLサーバ証明書を発行することができる
  - 悪意のある攻撃者が自らが認証局となって発行した証明書を提示している可能性がある
  - 商用認証局から証明書の交付を受けるには費用がかかるため, 自らが認証局となって証明書を発行し, 暗号化通信などを利用するサイトもある.
    - Web ブラウザからの警告が表示される
  - 信頼に足る管理者/組織が提供しているページかを確認しなければならない.

怪しい SSL 証明書が利用されている例

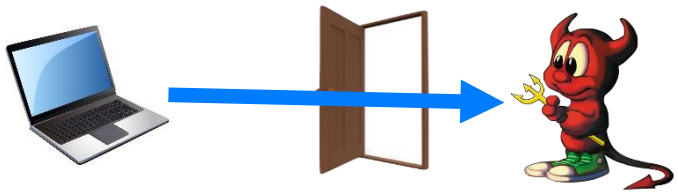
ネットワークセキュリティの原則 ～計算機管理者編：ユーザーを守るために～

- 計算機への不正アクセスを未然に防ぐ
  - ネットワーク空間との接点を最低限にする
    - ポートの管理
      - 不要なポートを閉める
    - アクセス制限
      - 必要外のホストによるアクセスを制限する
  - セキュリティホールを無くす
    - 最新のセキュリティ情報の取得
      - 8月8日の「Linuxインストールに必要な基礎知識」で解説済み

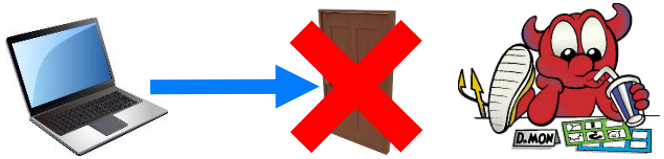
ポートの管理

- 各ポートには, パケットを取り扱うデーモンがいる
- ポートを開閉するには, ポートのデーモンを操作する
  - デーモンの起動と停止

ポートを開いた状態



ポートを閉じた状態



デーモン

- daemon（守護神）
  - メモリに常駐し, バックグラウンドで稼働し続けるプログラム
  - ポートデーモン：ポートを監視
    - ドア（ポート）の門番のような存在
    - ポートにデータが来たら, 要求に応じてサービスを提供
- スーパーデーモン：inetd/xinetd
  - デーモンを呼び出すためのデーモン
    - 常にポートを監視
    - 要求に応じて, 対応するデーモンを呼び出す
  - 使用頻度の低いデーモンを登録すれば, メモリの節約になる
  - /etc/inetd.conf で設定

起動デーモンの確認

- 「ps ax (ps aux)」コマンドを用いる
- 実行例(一部抜粋)

```
$ ps ax
PID TTY      STAT   TIME COMMAND
1   ?        Ss     0:06 /sbin/init
2   ?        S      0:00 [kthreadd]
223 ?        S      0:00 [kauditd]
569 ?        Ss     0:00 /usr/sbin/sshd -D
618 ?        Ssl    0:04 /usr/sbin/rsyslogd -n
```

- [ ]が付いているもの：カーネルデーモン
  - ハードウェアを直接制御するデーモン
- 赤字：ネットワークに関連したデーモン
- デーモン名の最後は「d」で終わることが多い

デーモンの停止方法

- systemctl コマンドを用いる
  - systemctl コマンド: デーモン管理用コマンド
  - 例えば, ssh のデーモンを停止する場合…

```
# systemctl stop sshd.service
```

    - 開始する場合は,「stop」-> 「start」
  - ただし, 計算機やアプリケーションを再起動すると, デーモンは復帰
- デーモンを含む不要なアプリケーションをアンインストールする
  - 不要なものはそもそもインストールしない

tcp_wrapper によるアクセス制限

- tcp_wrapper
  - inetd 経由で呼び出されるサービスについて, アクセス可能なホストやドメインを設定するアプリケーション
  - 設定方法

    ```
    デーモン名: アクセスを禁止/許可するホスト名(IPアドレス)
    ```

    と記述
  - アクセスの拒否 : /etc/hosts.deny に記述
    - 例 : すべてのアクセスを禁止

      ```
      ALL: ALL
      ```
  - アクセスの許可 : /etc/hosts.allow に記述
    - 例 : itpass.scitec.kobe-u.ac.jp からは ssh 接続を許可

      ```
      sshd: itpass.scitec.kobe-u.ac.jp
      ```

インターネットセキュリティまとめ

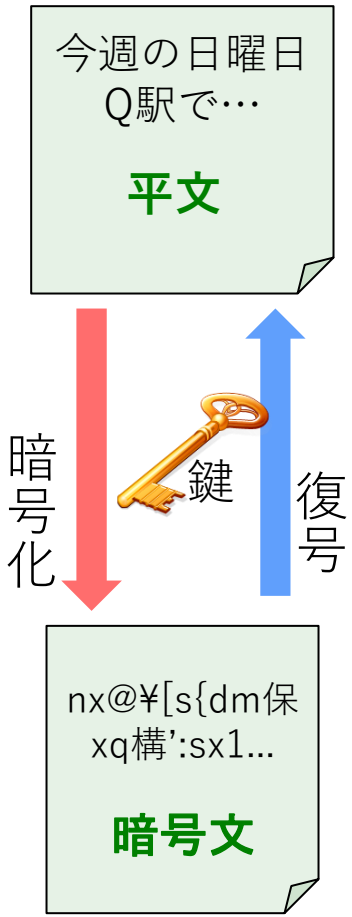
- 一般ユーザーとして：被害に遭わないために
  - 送信元がわからないファイルを開かない
  - よくわからない URL にはアクセスしない
  - 暗号化通信を使う
- 管理者として：ユーザーを守るために
  - ネットワーク空間との接点を最低限にする
    - ポートの管理, アクセス制限
  - 最新のセキュリティ情報を取得する

目次

- Network Computing の基礎
  - サーバ, クライアント, ポート, デーモン
- 最低限リモートアクセス
  - リモートログイン, ファイル転送
- インターネットセキュリティ
  - 被害に遭わないために, ユーザーを守るために
- 公開鍵暗号による通信と ssh 認証

暗号化

- 暗号
  - 情報を安全にやりとりするための技術
- 暗号化
  - 元のデータ (平文) を別のデータ (暗号文) に変換すること
- 復号
  - 暗号文を平文に戻すこと
- 鍵
  - 暗号化, 復号の際に用いるデータ
  - たいていは bit 列の長いものが使われる



暗号の例: シーザー暗号

- アルファベットを **n** 文字後にずらす
  - 例: **n** = 3 のとき
    - deepconv → ghhsfrqy
    - Kobe Hanako → Nrdh Kcqcmr
  - 暗号化のアルゴリズム
    - 「**n** 文字後にずらす」という操作
      - この **n** は任意に選ぶことができる
  - 鍵
    - 「ずらす文字数 **n**」が鍵として扱われる
  - 暗号化のアルゴリズム と 鍵 がセット
- 暗号化のアルゴリズム：公開されている
- 鍵：バレるとマズい

暗号化と復号
- Alice が Bob にメッセージを送るとき -

Alice

今週の日曜日
Q駅で…

平文

暗号化処理

暗号化アルゴリズム

＋

鍵

Network

nx@¥[s{dm保
xq構':sx1...

暗号文

復号処理

復号アルゴリズム

＋

鍵

今週の日曜日
Q駅で…

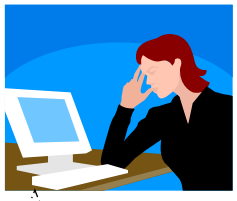
平文

Bob

共通鍵暗号

- 特徴
  - 同じ鍵（共通鍵: common key）を使って暗号化・復号
  - 対称鍵暗号, 秘密鍵暗号とも呼ばれる
- メリット
  - 暗号化・復号の処理が速い
- デメリット
  - 鍵を事前に共有する必要がある (鍵配送問題)
- 共通鍵暗号の例
  - DES, 3DES, AES

鍵配送問題



Alice

今週の日曜日
Q駅で…

平文

暗号化処理

暗号化アルゴリズム

＋

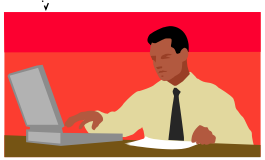
鍵

Network

nx@¥[s{dm保
xq構':sx1...

暗号文

あらかじめ鍵を秘密裏に
交換しておく必要がある
→ **鍵配送問題**



Bob

今週の日曜日
Q駅で…

平文

復号処理

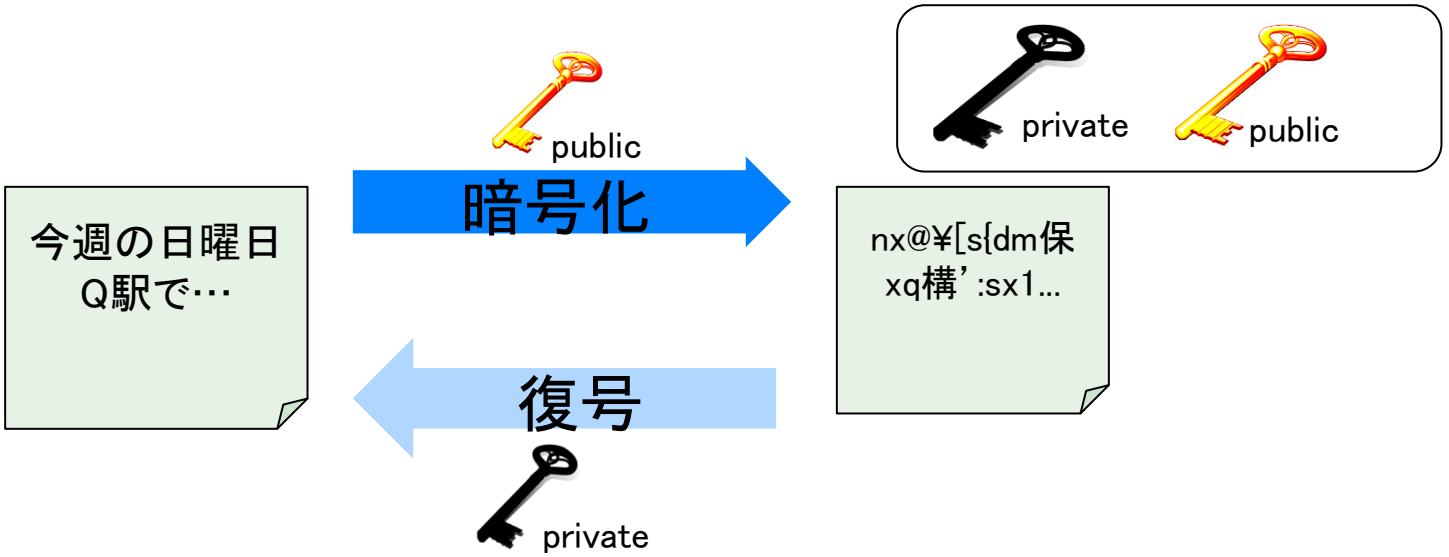
復号アルゴリズム

＋

鍵

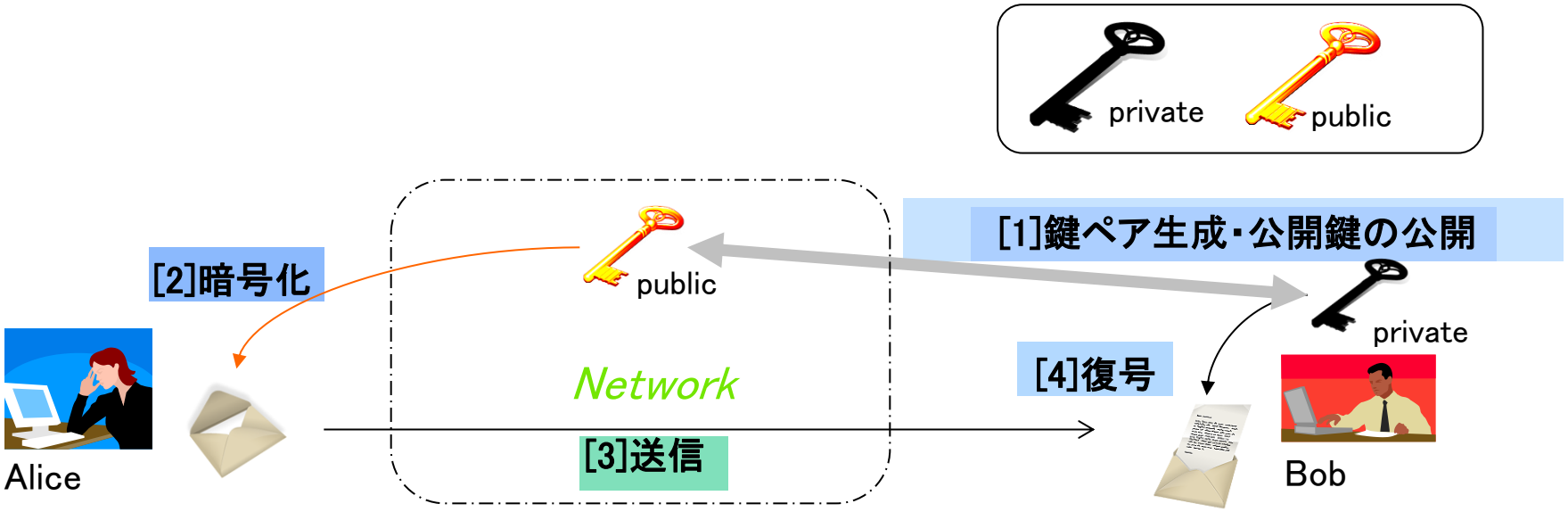
公開鍵暗号

- 鍵配送問題を解決できる
- 秘密鍵（private key）と公開鍵（public key）のペアとして鍵が作られる
  - 公開鍵で暗号化されたものは，ペアの秘密鍵でしか復号できない



公開鍵暗号の使い方

- Alice が Bob にメッセージを送るとき
  - Bob はあらかじめ鍵のペアを作っておき, 公開鍵をネットワーク上に公開[1]
  - Alice は, Bob の公開鍵でメッセージを暗号化[2] したのち送信[3]
  - Bob は自分の秘密鍵でメッセージを復号[4]

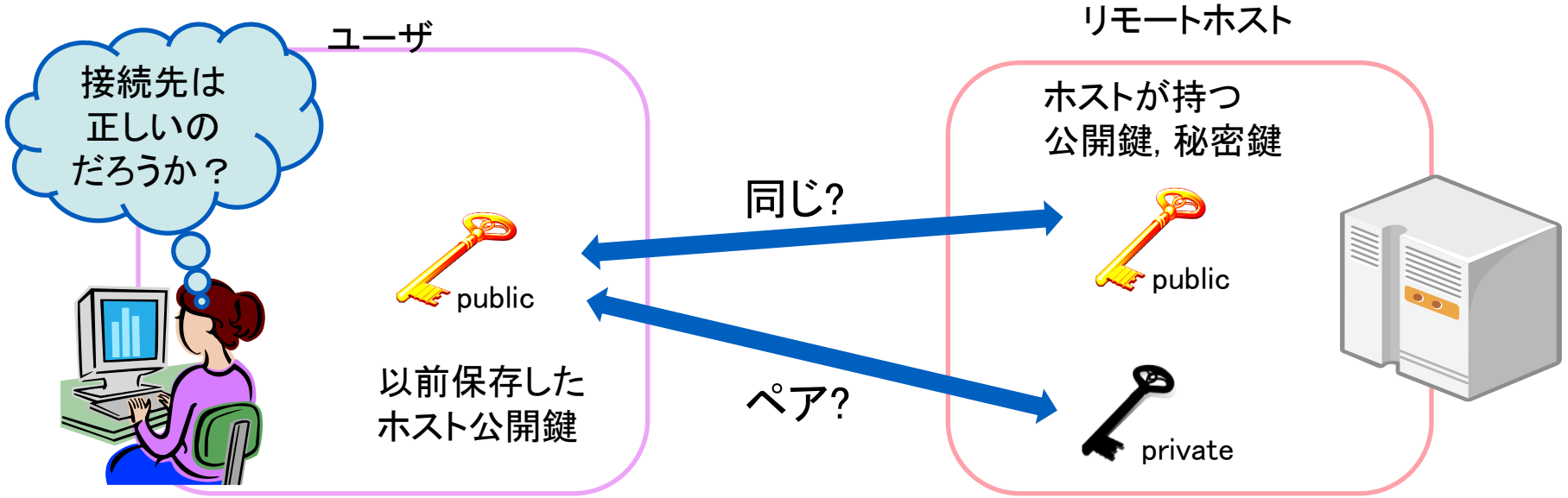


公開鍵暗号

- 特徴
  - 秘密鍵と公開鍵のペアとして鍵がつくられる
- メリット
  - 鍵の配布が簡単
    - 公開鍵は誰に見られても良い
    - 解読に必要な秘密鍵は, ネットワークに流す必要がない
    - 不特定多数の相手とやり取りができる（全員が同じ公開鍵で暗号化してくれる）
- デメリット
  - 同じ鍵で暗号化・復号する暗号（共通鍵暗号）に比べて処理速度が遅い
  - 公開鍵の持ち主を確かめる必要がある → 認証の必要性
- 公開鍵暗号の例
  - RSA, ECDSA, DSA, ElGamal, …

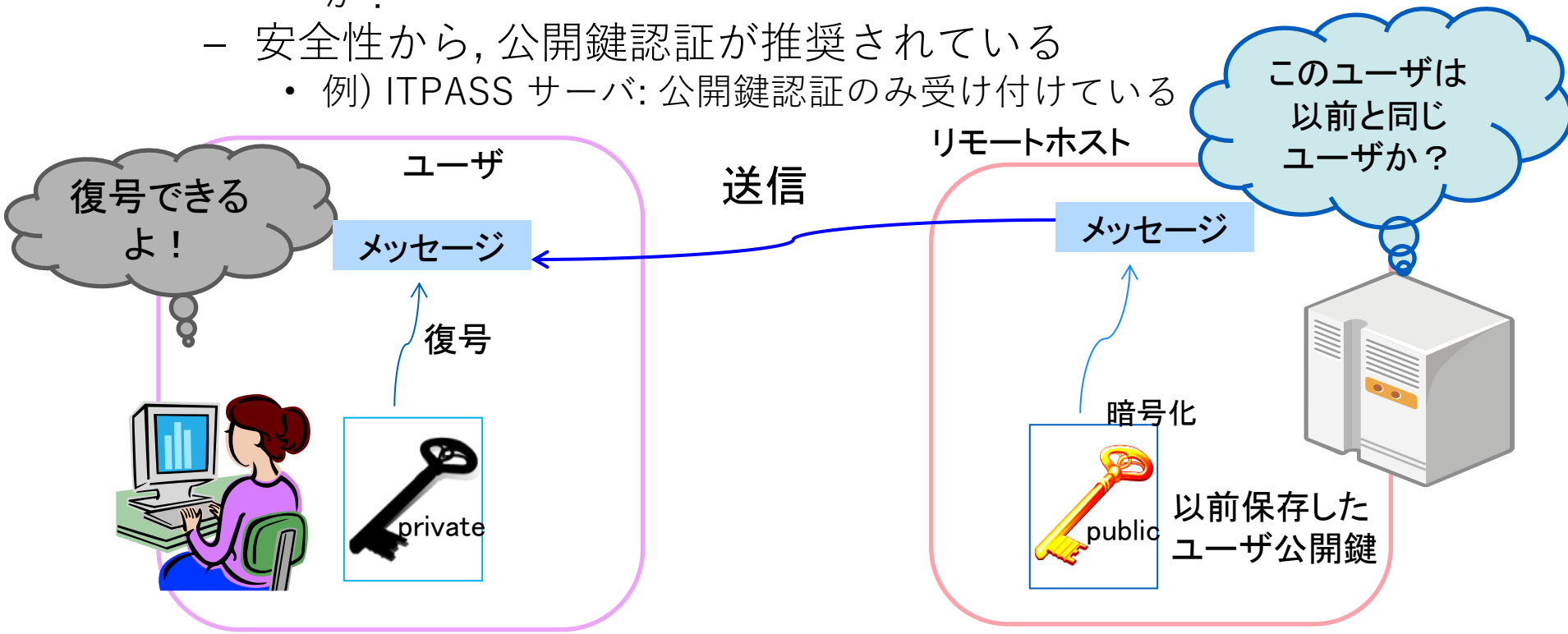
ssh 認証 – ホスト認証 -

- ユーザ側が, リモートホストの「なりすまし」の有無をチェック
  - ユーザ認証がパスワード認証の場合, 認証先をなりすまされると, パスワードを相手に教えることになる
  - 公開鍵暗号の仕組みを利用して認証
    - リモートホストの公開鍵（ホスト公開鍵）が以前のアクセス時と変わっていないか？
    - リモートホストは, ユーザが持つホスト公開鍵に対応した秘密鍵を持っているか？



ssh 認証 – ユーザ認証 -

- リモートホスト側が, ユーザの「なりすまし」の有無をチェック
  - たいていはどちらかの方法で認証
    - パスワード認証: ユーザはパスワードを知っているか？
    - 公開鍵認証: ユーザが設置した公開鍵に対する秘密鍵を持っているか？
  - 安全性から, 公開鍵認証が推奨されている
    - 例) ITPASS サーバ: 公開鍵認証のみ受け付けている



公開鍵暗号と ssh 認証まとめ

- 共通鍵暗号と公開鍵暗号
  - 共通鍵暗号
    - 暗号化と復号に同じ鍵を用いる
    - 処理速度は速いが, 鍵配送問題が存在する
  - 公開鍵暗号
    - ペアとして生成される「秘密鍵」と「公開鍵」を用いる
    - 公開鍵の持ち主を確認する必要あり　→ 認証の必要性
- ssh 認証
  - 通信を行うもの同士がそれぞれ「なりすまし」ていないかチェックする
  - 認証には主に公開鍵暗号が用いられる

ITPASS
Informational Training program
with a spirit of self-help

補足資料

インターネットセキュリティ

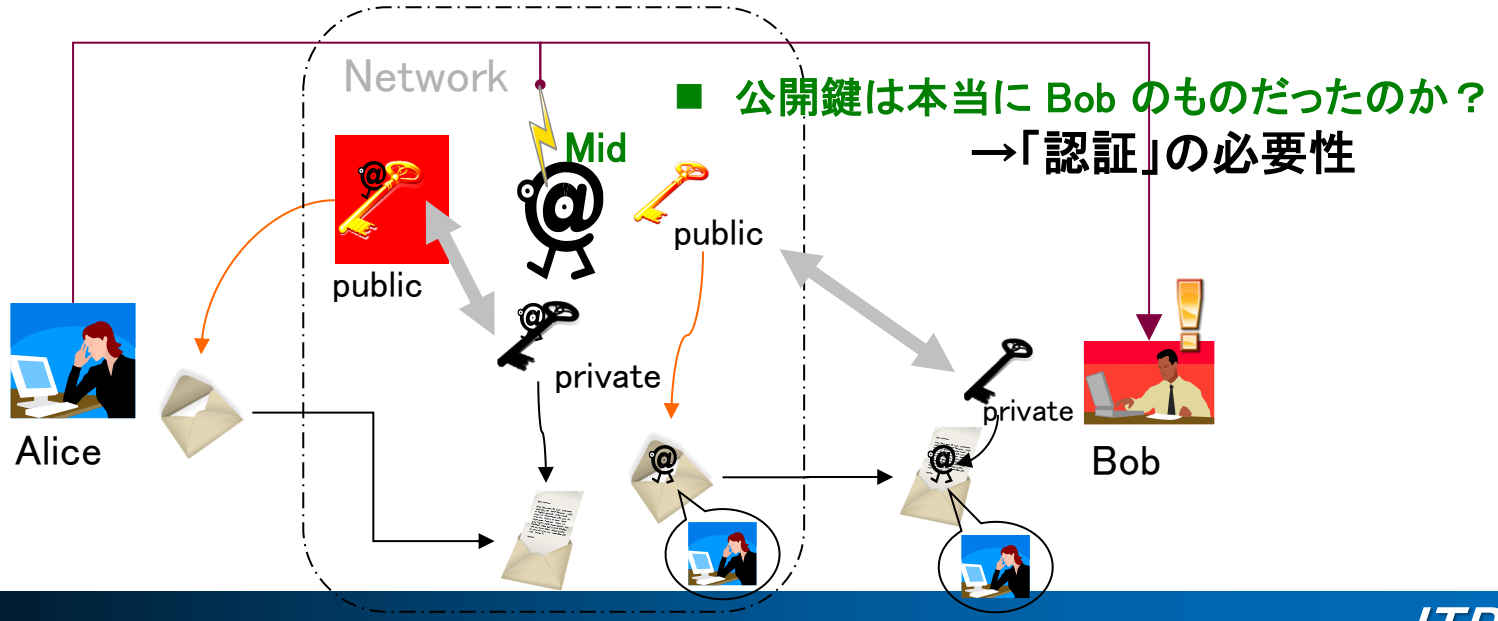
公開鍵認証における認証の必要性

- Man-in-the-Middle 攻撃 (次項) などにより, ネットワーク上に公開している公開鍵が改ざんを受ける可能性がある

  → 公開鍵が本当に本人のものなのか, 認証する必要がある

Man-in-the-Middle 攻撃

- Alice が Bob にメッセージを送る
  - Alice: Bob へメッセージを送るために公開鍵を要求　→ 盗聴者 Mid に傍受された
  - Mid:　Bob からの公開鍵を奪い, Alice に届かないようにする
    代わりに, 自分の公開鍵を Bob のものと偽って Alice に送りつける
  - Alice: (Bob のものだと思って) Mid の公開鍵でメッセージを暗号化, 送信
    → 復号できるのは Mid だけ
  - Mid:　さらに奪った公開鍵を使って, Alice のふりをして Bob に悪意的メッセージを送る
  - Bob: (公開鍵は Alice に送ったはずだから) Alice からのメッセージと思ってしまう

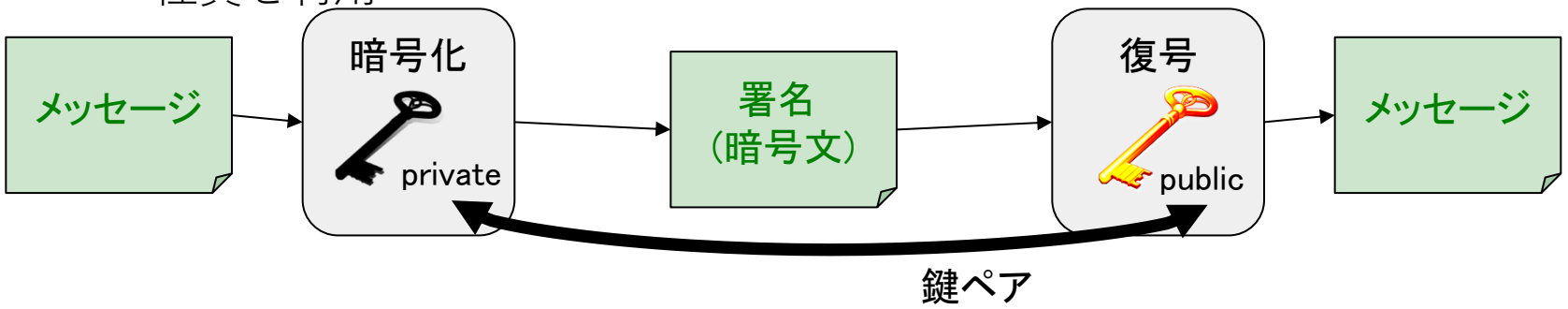


認証の手段

- 本人から直接 (物理的に) 公開鍵をもらう
- 第三者機関 (認証局) の発行する公開鍵証明書を利用する
  - 「この公開鍵は確かに Bob のものだ」という証明が交付される
- 電子署名を利用する

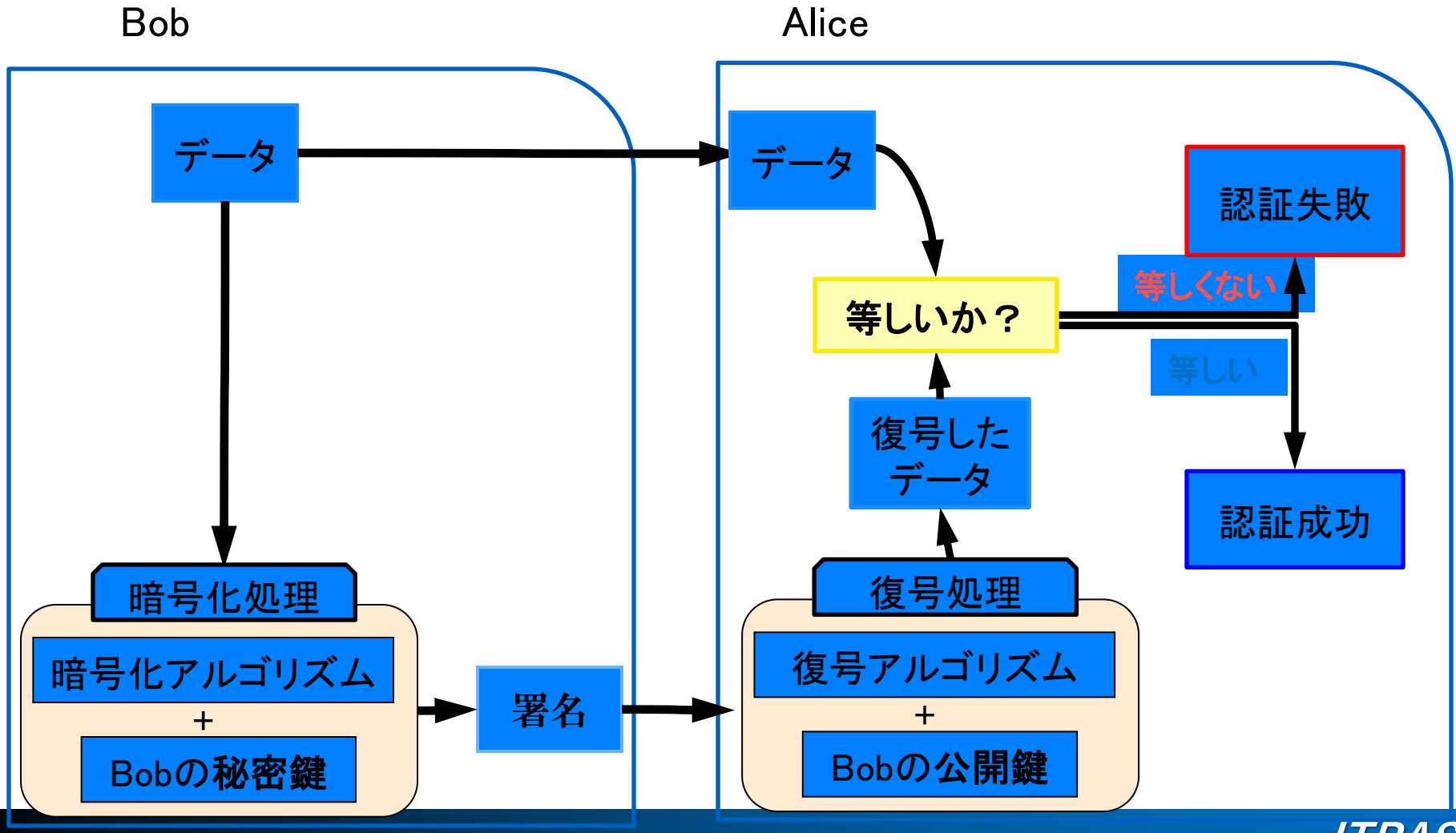
電子署名

- 電子署名は「メッセージを書いた人がその人自身」であることを保証する
- 電子署名アルゴリズム
  - 主なものに, RSA (RSA 暗号を利用), DSA (ElGama 暗号を利用) など
- RSA 公開鍵暗号を使った電子署名
  - 暗号として使ったときの「逆」を考える
  - RSA 公開鍵暗号の「秘密鍵で暗号化したものは, 公開鍵で復号できる」性質を利用



  - 「Bob の公開鍵で復号できるような暗号文を作成できるのは, ペアとなる秘密鍵を持っているであろう Bob ただひとり」

電子署名のしくみ

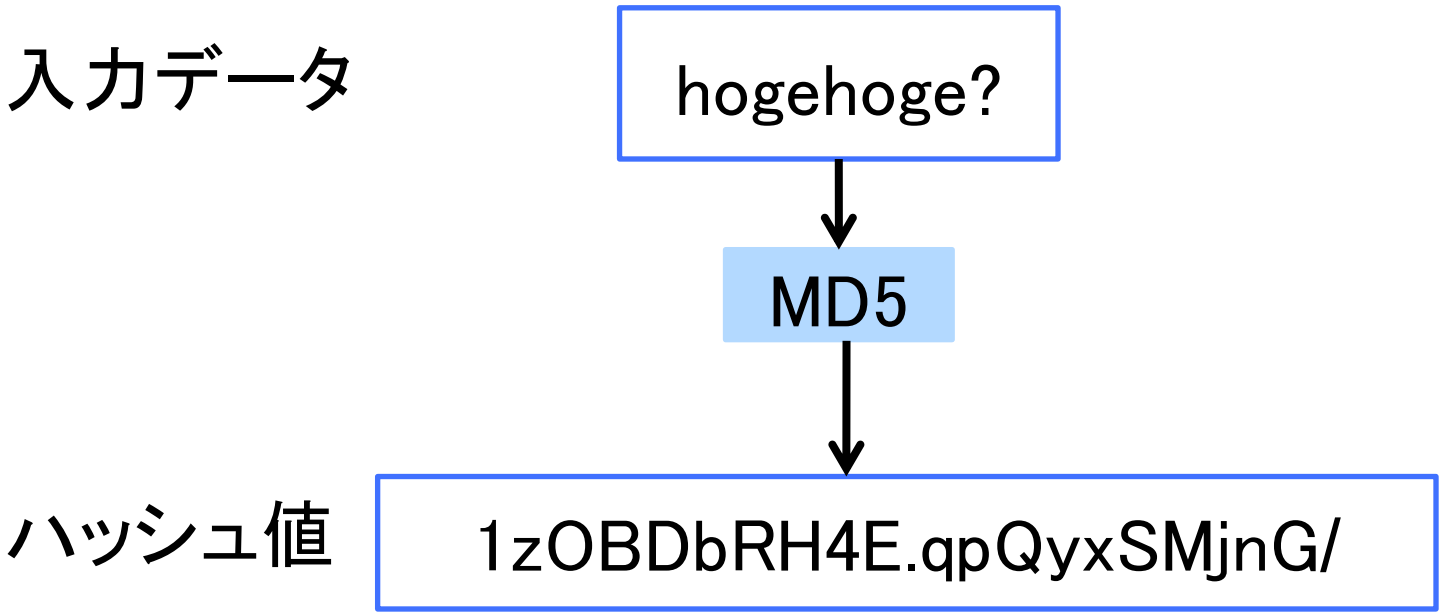


一方向ハッシュ関数

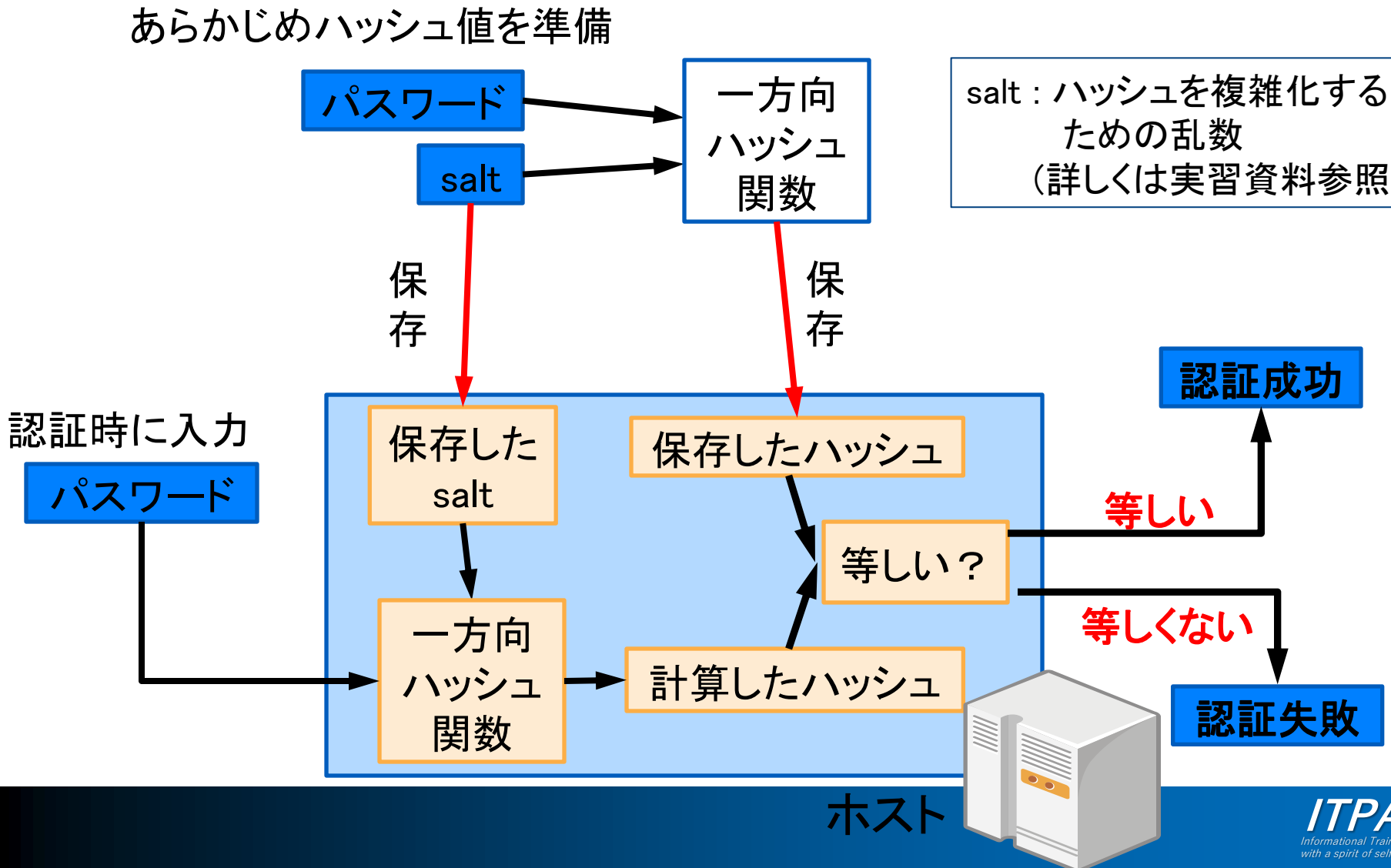
- データの代表となる bit 列を得るための関数
- 入力データから「ハッシュ値」を計算
  - ハッシュ値から入力データは原理的に求められない (一方向性のため)
  - 入力データを少し変えると全く違うハッシュ値を出力
  - 入力データの大きさによらない, 決まった長さのハッシュ値を計算
- 種類
  - MD5, SHA1, SHA256, …
- 用途
  - パスワード認証
  - 公開鍵認証
  - ファイルの改ざん検出
  - データの検索

一方向ハッシュ関数の例

MD5 による例



一方向ハッシュ関数を用いたパスワード認証



参考資料・参考文献

- 全般
  - 神戸大学 ITPASS 実習 2022 年度「Network Computing & Internet Security」
    - https://itpass.scitec.kobe-u.ac.jp/exp/fy2022/220809/lecture_networkcomputing/pub/
  - 北海道大学 情報実習 2019 年度 「リモートアクセス/ネットワークセキュリティ 」
    - http://www.ep.sci.hokudai.ac.jp/~inex/y2019/0621/lecture/pub/inex20190621.pdf
  - 小悪魔女子大生のサーバエンジニア日記
    - http://co-akuma.directorz.jp/blog/
  - IT 用語辞典 e-Words
    - http://e-words.jp/
- Network Computing の基礎
  - 「分かりそう」で「分からない」でも「分かった」気になれるIT用語辞典
    - https://wa3.i-3-i.info/index.html
- インターネットセキュリティ
  - ネットワークエンジニアとして
    - https://www.infraexpert.com