SSHの仕組み

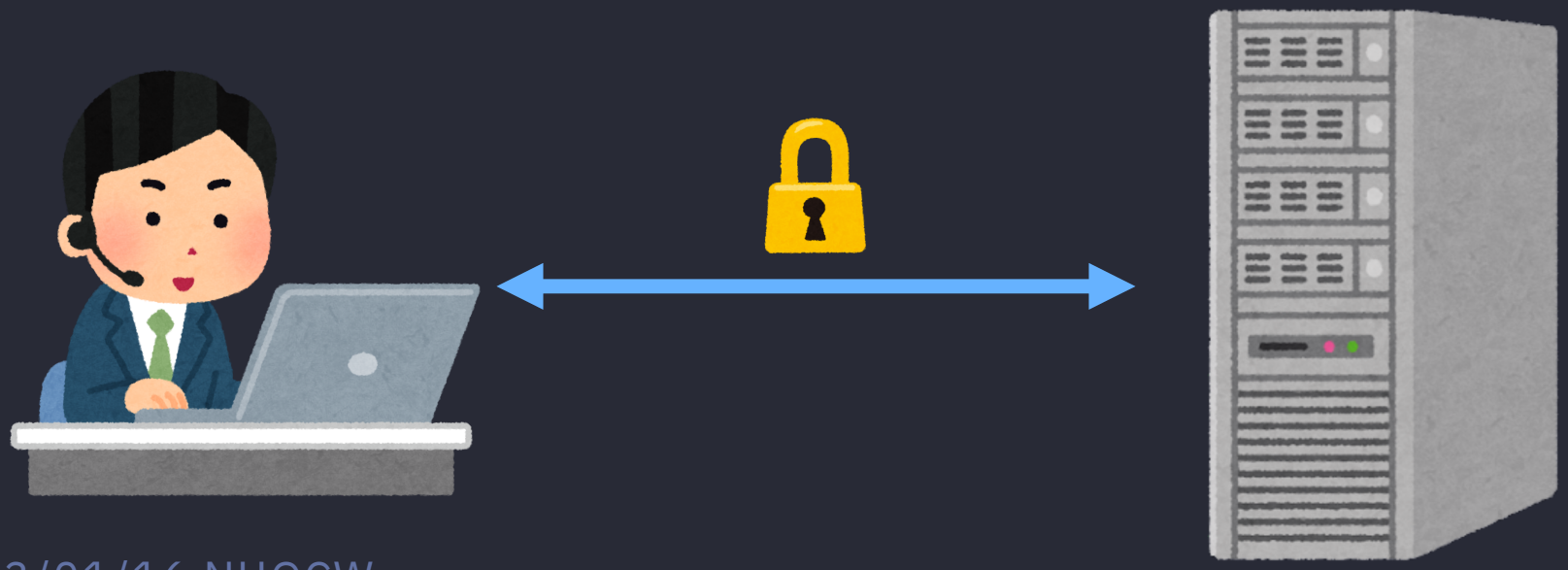
サーバサイド班　甲本健太 🔗

目次

SSHとは

SSHとは**Secure Shell**（安全なシェル）の略称であり、安全にリモートコンピュータと接続するためのプロトコル。

- サーバに接続して作業するときなどに利用する



実演

コマンド

```
$ ssh -i {秘密鍵} {ユーザ名}@{接続先のIPアドレス}
```

ssh-configを使う

```
Host {登録名}
    HostName ocw.nagoya-u.jp  # 接続先
    User {ユーザ名}
    IdentityFile {秘密鍵のパス}
```

SSH接続の流れ

認証段階

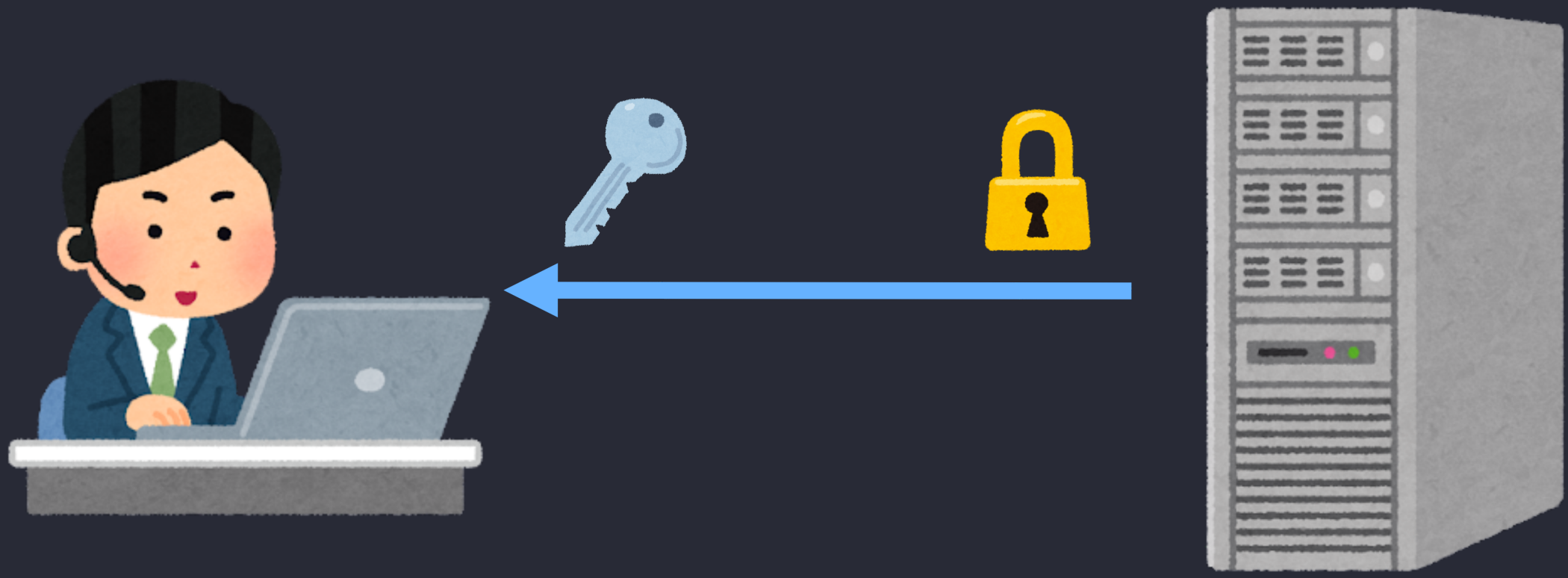
1. クライアントによるサーバの認証
2. セッションキーの生成
3. サーバによるクライアントの認証

通信段階

4. データを暗号化して通信

1. クライアントによるサーバの認証

クライアント側が、サーバが怪しいものでないかを調べる



初めてサーバにアクセスするとき

サーバの公開鍵を検証し、**手動で**検証を行う

→ `Are you sure you want to continue connecting (yes/no)?`

ローカルの `known_hosts` ファイルにその情報を保存する。

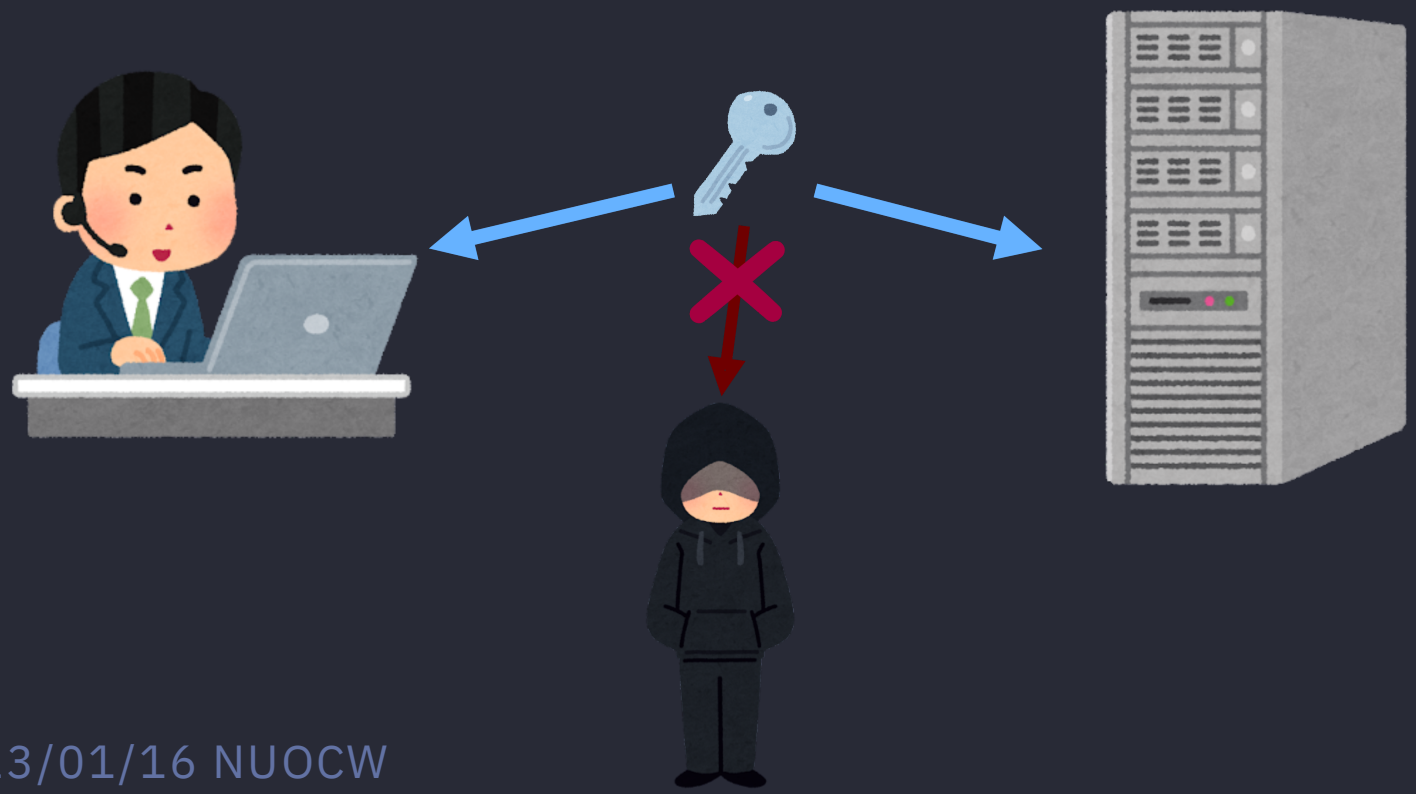
（他にも `ssh-keyscan` コマンドを用いるなどの方法もある）

2回目以降

ローカルの `known_hosts` にある公開鍵情報からサーバの身元を確認

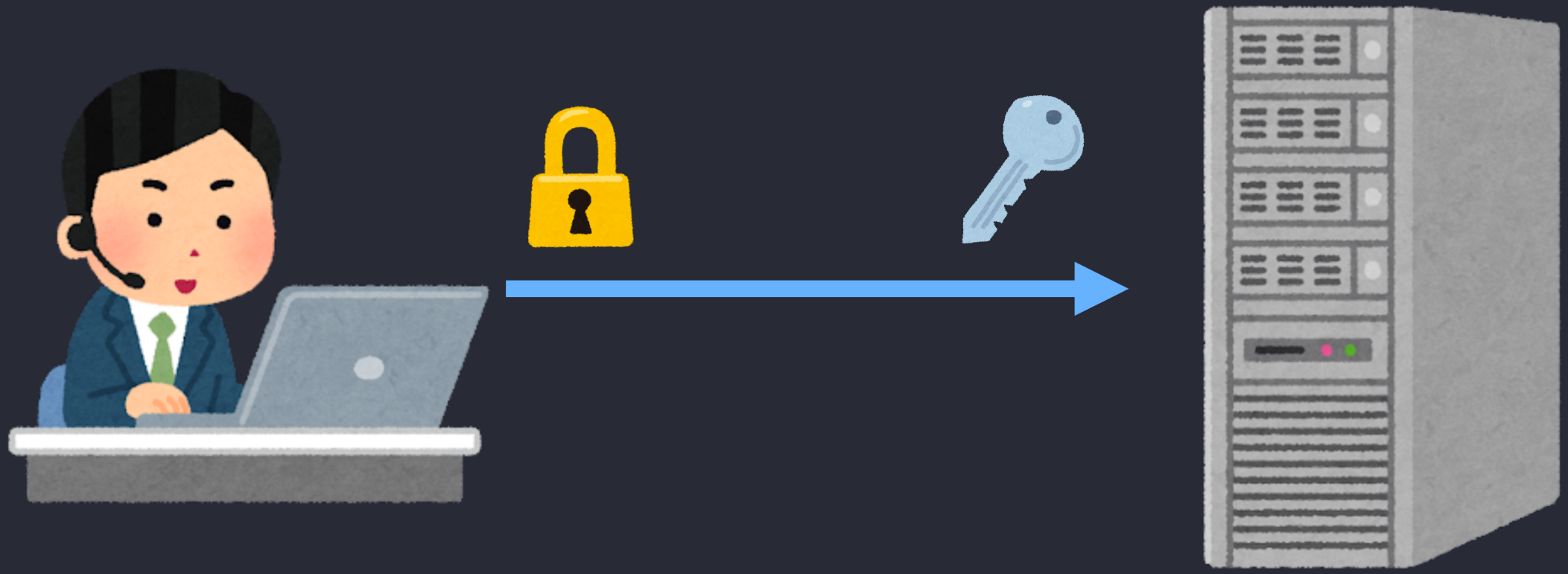
2. セッションキーの生成

通信の暗号化をするための鍵を生成する。
ディフィー・ヘルマンの鍵交換を用いて、中間者攻撃に対処



3. サーバによるクライアントの認証

サーバ側が、正しいクライアントに接続しているかを調べる



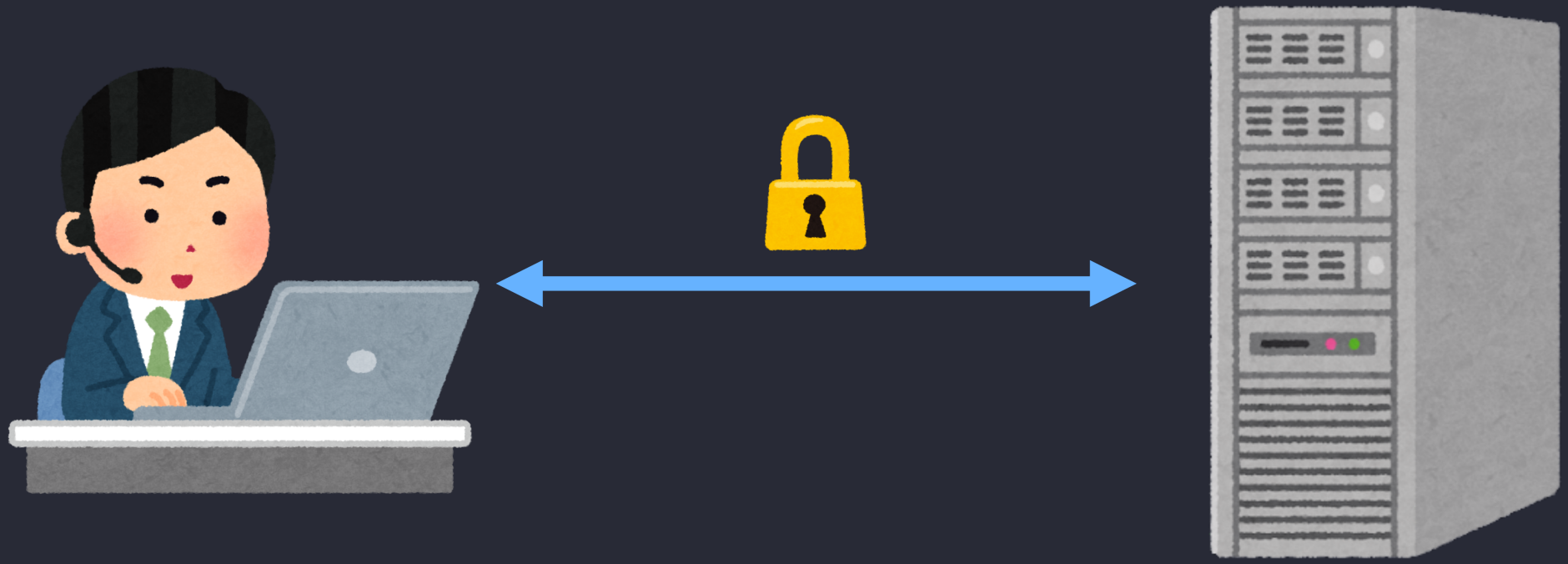
クライアント認証の流れ

1. 鍵ペア（公開鍵、秘密鍵）を生成する
2. サーバ側に公開鍵を渡す
3. サーバがトークン（乱数）を生成しクライアントに送る
4. 自分の秘密鍵を用いてトークンを暗号化し
   暗号化したトークンをサーバに送り返す
5. サーバ側で公開鍵を用いて5を復号し、
   その結果が3と一致すれば「**認証成功**」

c.f. 電子署名

4. データを暗号化して通信

- 2．セッションキーの生成 で生成された鍵を使って暗号化通信をする
- 高速な暗号化、復号が可能な**共通鍵暗号**を使う



SSHのログを見てみよう

`ssh` コマンドに `-v` オプションをつけると接続の際のログを確認できる

（おまけ）RSAとECDSA

- RSA
  - 「素因数分解の困難性」ベース
  - 公開鍵暗号の最初の実装
- ECDSA
  - 「楕円曲線上での離散対数問題」ベース
  - 比較的最近開発された方法

同じ長さの鍵では、ecdsaの方が強度が高い
→ 最近はecdsaが推奨されています！

まとめ

- SSHは`公開鍵暗号`、`共通鍵暗号`、`電子署名`、`鍵交換アルゴリズム`といった暗号技術をフルに使ったすごい技術！
- サーバにSSH接続をするときは、より安全な`公開鍵認証`を使おう！

参考

- Secure Shell (Wikipedia)
  https://ja.wikipedia.org/wiki/Secure_Shell
- OpenSSH (公式)
  https://www.openssh.com/
- Understanding SSH Workflow
  https://medium.com/@hellomudit/understanding-ssh-workflow-66a0e8d4bf65