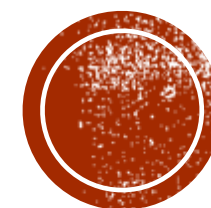


再配布禁止

情報科学概論 第3回

情報の伝達と通信



立教大学大学院 人工知能科学研究科

2023年5月1日

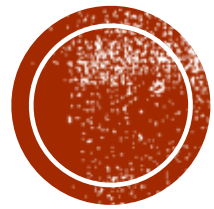
アンドラーデ ダニエル

andrade@rikkyo.ac.jp

本日の授業運用

- 授業時間：5 時限（17:10～18:50）
- 授業中、ChatやSlackが見えない可能性がありますので、キリのいいところで質問にまとめて答えます。スライド上では「質問タイム」で表示します。
- 「質問タイム」では是非音声でも質問をお願いします。
- **今回はBreakout Roomないが、「お試しタイム」を設ける。**
- **今回の授業はたくさんの用語が出ているが、すべて覚える必要はない。スライドのタイトルで出ている用語は重要で覚えてください。**
 - とはいえ用語を暗記するより、概念を理解することが重要。
- 授業に関する意見・要望はお気軽にご連絡をお願いします。
(Slackや私のメールアドレス (andrade@rikkyo.ac.jp))に)





前回の授業のおさらい

+

レポート課題の回答

前回の授業のおさらい (1/2)

- クラウドコンピューティング (Cloud Computing) :

従来は手元に整備する必要があったソフトウェア、データ、開発環境が、インターネットを通してサービスプロバイダーから提供される。例：Google Docs, iCloud, AWS

メリット：ユーザのニーズの増減に柔軟に対応でき、コスト削減につながる。

- 人工知能システムとデータ :

現在の人工知能には学習用のデータが不可欠。人工知能システムの計画・運用・保守において、データの調達・整備・管理も含めて考える必要がある。

- 並列コンピューティング :

並列・分散コンピューティングは複数のCPUやサーバに計算を分担する。大規模データ処理に不可欠。

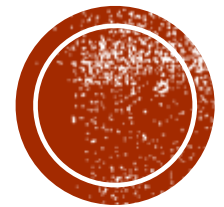


前回の授業のおさらい (2/2)

- 情報システムにおける安心・安全性の強化
 - リスク分析：そもそもどのようなリスクがあるかを列挙、優先順位をつける。
 - ソフトウェア品質の管理：バグが少ない、または、修正しやすいように設計する。
 - ハードウェアに関するリスク管理：HDDのバックアップなど
- 人工知能システムにおける安心・安全性の強化
 - 解釈性：なぜこの結果になったか説明できることにより、人工知能の誤判断を発見し介入する。
 - ロバスト性：外れ値やAdversarial AttackやData Poisoningにロバスト。
 - 確率的な評価：人工知能システムが分からないことを把握。
 - 学習データの整備・更新



質問タイム



本日の内容

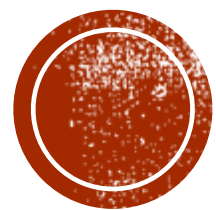
1. インターネットの仕組み

- インターネットのアドレス仕込み
- 通信規格

2. 通信におけるセキュリティ

- 暗号化と復号
- デジタル署名
- HTTPにおけるサイバー攻撃とSSHに関する補足





インターネットの 仕組み

インターネットの仕組み

概要

- インターネットには色々な機器がつながっている。
パソコン、携帯、印刷機、最近は冷蔵庫なども
- 機器同士のコミュニケーションを可能にするために、共通の言葉・ルールを利用する必要がある。
- そのために、標準化した規則（プロトコル）が必要となっている。
- 目的に応じて、様々なプロトコルがあり、歴史的に残っている、更新されているプロトコルも多い。
- プロトコル群の中ではTCPとIPが特に重要な役割を果たしている。



インターネットの仕組み

次のページ以降では以下の例を利用：

AさんがWikipediaで「桜」に関する記事を読覧したい。

例で分かるもの：

- AさんとWikipediaの間でどのような通信が行われているか。
- データの通信のためにどの基準が利用されているか。
- Web pageの閲覧だけでなく、インターネットで利用されている共通の技術は何か。



CLIENT/SERVER

AさんとWikipediaの立場は以下のようなものである。



URL (UNIFORM RESOURCE LOCATOR)

Aさんが「桜」の記事をWikipediaから要求する際には以下のアドレスをWeb Browserに入力：

`https://ja.wikipedia.org/wiki/サクラ`

プロトコル名* ホスト名 パス名

- Web pageを指定するためだけではなく、FTPなどにも利用される。
- ホスト名はDomain Nameとも呼ぶ。

* 一般的にはプロトコル名ではなく「スキーム名」



ホスト名とDNS (DOMAIN NAME SERVER)

まずは、ホスト名 (ja.Wikipedia.org) は世界 (インターネット) のどこにあるのかを調べる必要がある。

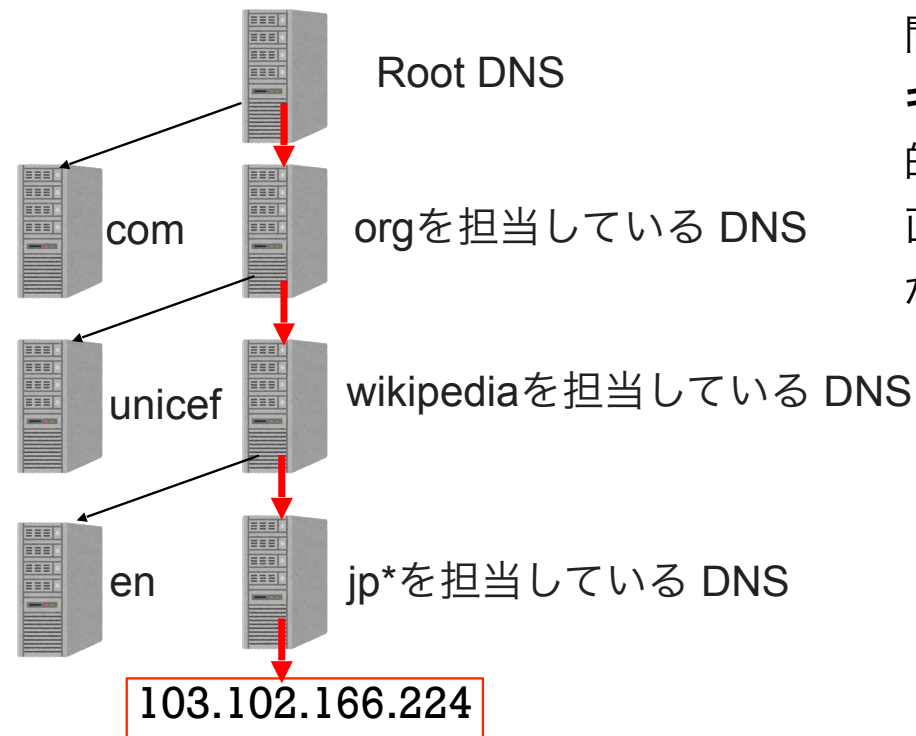
- ホスト名は人間にとって覚えやすい名前。
- ただし、インターネットにおける通信ではホスト名をIPアドレスに変換する必要がある。
- IPアドレスを調べるためにホスト名の構造が利用される。

DNS (Domain Name Server)群がホスト名のIPアドレスを以下のように調べている。

ja.wikipedia.org

← 調べる方向

理由：ホスト名の管理を効率よくするため。



毎回調べるのは時間がかかるため、**キャッシュ**（一時的な保存場所）に直近の問い合わせが保存される。



IPアドレス (IPV4 または IPV6)

IPアドレスによってServerとClientはどこにあるのかが分かる。

- IP (Internet Protocol)のアドレスは32Bitで表現されている。 (IPv4)
- 32Bitで表現できるユニークなアドレスが少なすぎるため、将来的には128Bitで表現される。(IPv6)
- インターネットに繋いでいる全ての機器が一意的なIPアドレスを持つ。(電話番号と同じ)
- 電話番号における国番号、市外局番と同じように、IPアドレスから地理的な大まかな場所の推定ができる。
- IPv4のアドレス例：

103.102.166.224

各8Bitが10進表記で表示され、「.」で区切る。

電話番号と違って、**IP**アドレスが頻繁に変わる可能性がある。
(そのため、前ページの**DNS**仕組みで調べる必要がある。)



ポート番号 (PORT)

ServerとClient内でApplication（この例ではWeb Browser上のWeb page閲覧）を指定するためのもの。

- IPアドレスは一つの機器(PC, 携帯等) までのアドレス。
- 機器内にどのアプリケーション宛なのかがポート番号で決まる。（＊）
- ポート番号が16BitでIPアドレスの後ろに「:」で表示することが多い。
- 例：

ポート番号
└──┬──┘
103.102.166.224:8080

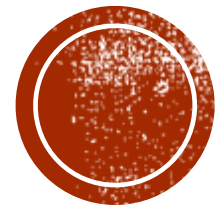
または

jp.wikipedia.org:8080

（＊） アプリケーション層のHTTP, SSHなど

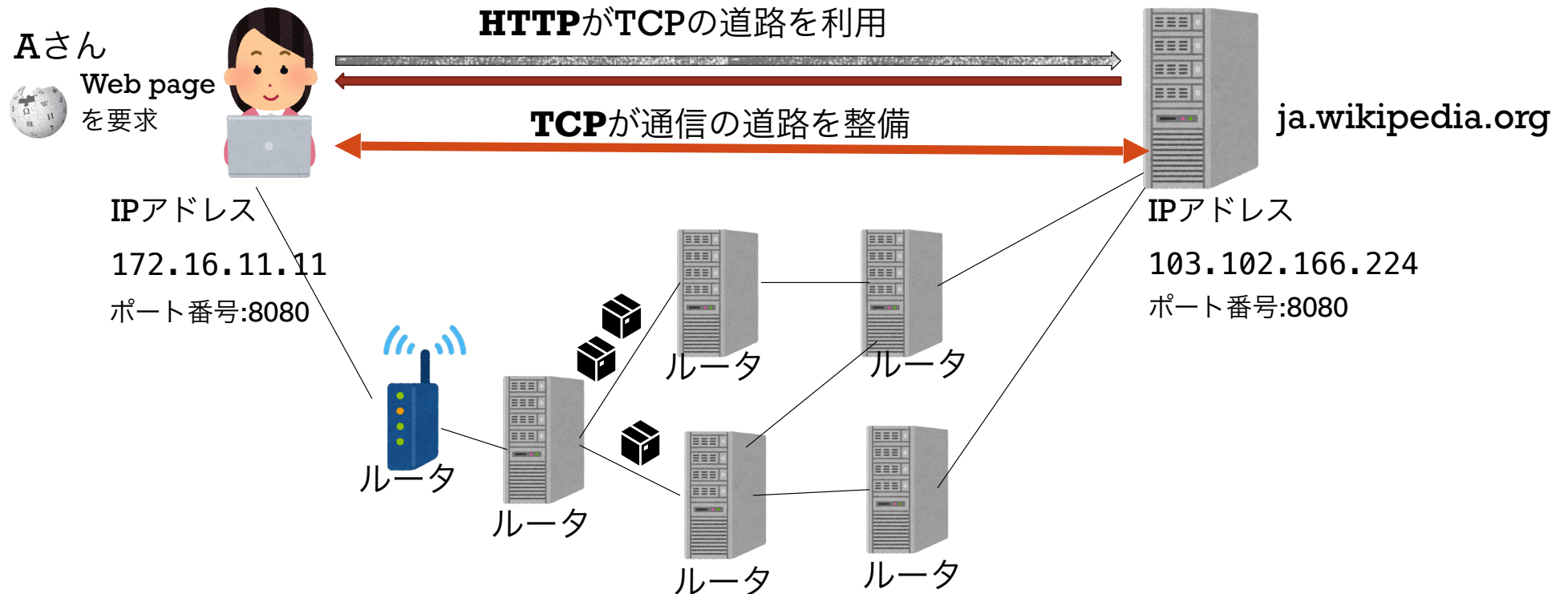


質問タイム



プロトコル (PROTOCOL)

- IPアドレスが分かれば、ServerとClientの間の通信が可能。通信がProtocolで制御される。
- Protocol = マシンやソフトウェア同士のやりとりに関する取り決め (通信規約)
- 例：HTTP, TCP, IP



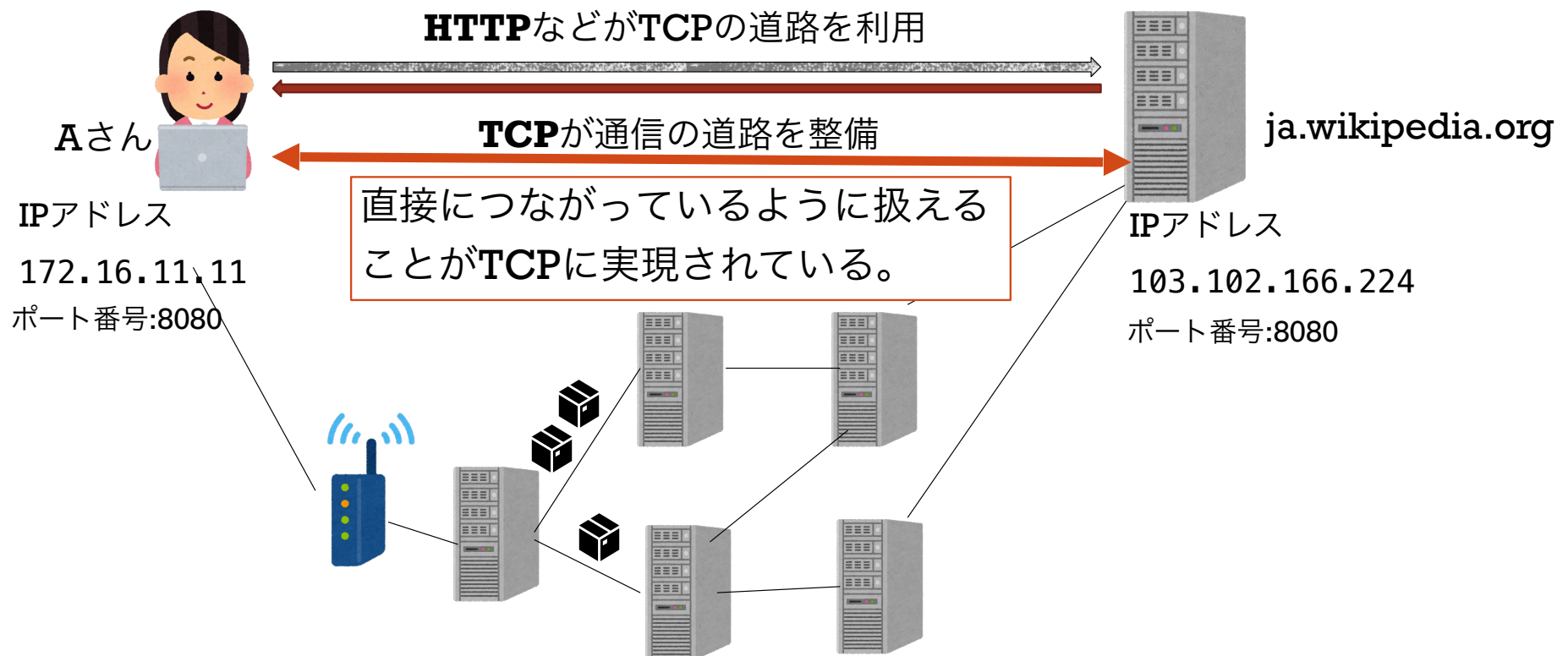
HTTP (HYPERTEXT TRANSFER PROTOCOL)

- アプリケーション層のプロトコルの一つ。
- ウェブページを要求するために利用されているプロトコル。
- ウェブページの表示自体はBrowserが担当している。
- 例：
 - 「GET」 コマンドで文章要求
 - 「POST」 コマンドでClientがWeb Serverに情報を提供。



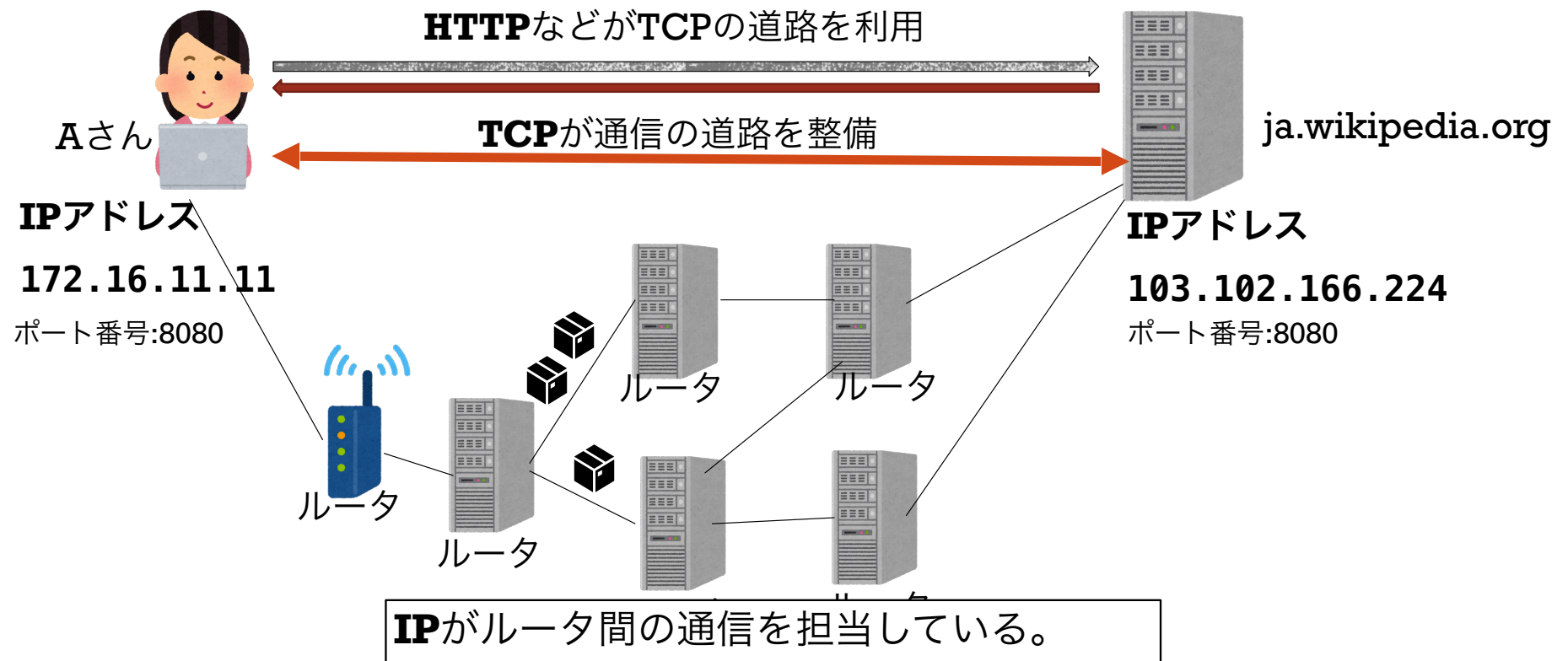
TCP (TRANSMISSION CONTROL PROTOCOL)

- HTTPなどのアプリケーションプロトコルに利用されているプロトコル。
- アプリケーションがTCPにデータ通信を任せている。
- TCPがデータを分割したりしたあとに、IPに通信の詳細を任せている。



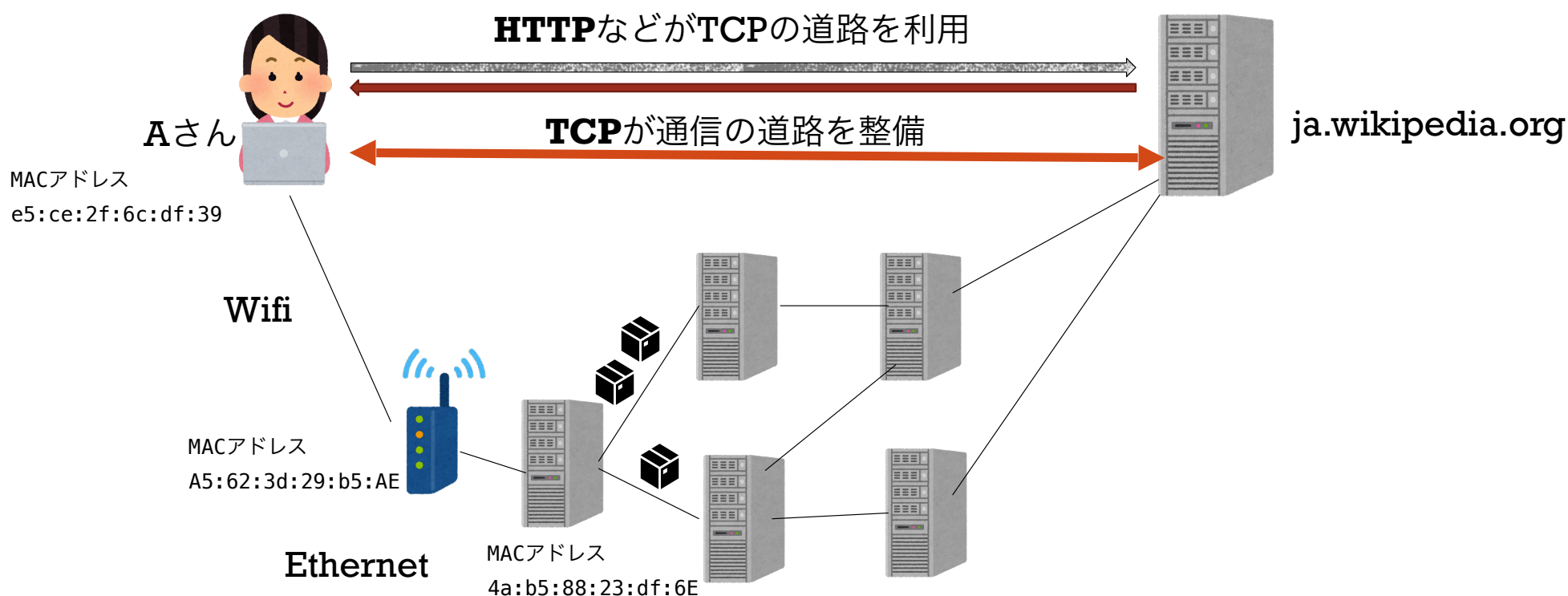
IP (INTERNET PROTOCOL)

- TCPに分割されたパケットの通信を担当している。
- TCPとIPが初期からInternetの中核プロトコルである。**TCP/IP**でまとめて表現することが多い。



媒体アクセス制御 (MEDIUM ACCESS CONTROL)

- ルータ間の物理的な通信方法に利用されているProtocol。
- 例えば、Ethernet, Wi-Fi, Bluetoothで利用されている。
- アドレスの仕込み：MACアドレス（次のページ）



MACアドレス (物理アドレス)

- ネットワークに接続可能な通信機器(PC, 携帯) に付いている一意的な48Bitの番号。
- 「:」で区切っている16進表記で表示することが多い。例：
e5:ce:2f:6c:df:39
- IPアドレスと違って、MACアドレスが固定に機器と結びついている。
- 応用例：例えば、セキュリティのために、ネットワークに接続可能な機器を事前に登録したMACアドレスに限定。
- 例：

```
$ ifconfig en1  
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500  
ether e5:ce:2f:6c:df:39  
...
```



インターネット・プロトコル・スイート (INTERNET PROTOCOL SUITE)

- インターネットにおけるプロトコルの階層モデル。
- TCP/IP Layer Modelとも呼ぶ。
- 下の層が上の層に機能を提供する。カプセル化
(モジュール化に似ている。参考：Pythonのプログラム -> Numpy -> BLAS)
- 効果：共通に利用されている機能が下の層に実装されているため、実際のハードウェア等からの抽象化。その結果、層の交換が簡単、基本的な機能の再実装が不要。



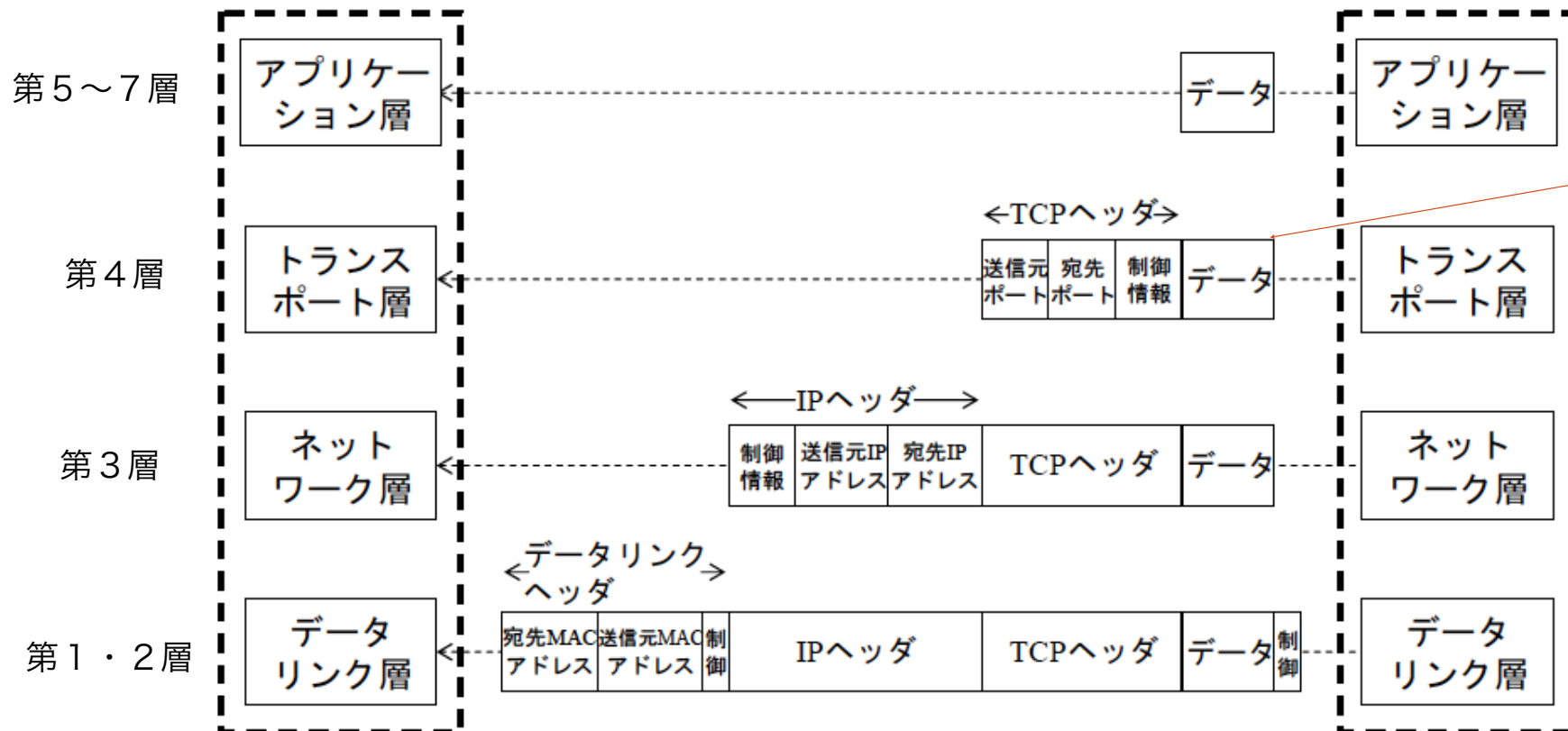
インターネット・プロトコル・スイート (TCP/IPの階層モデル)

	Layer名	主なプロトコル	宛先の主な指定方法
第5～7層	アプリケーション層	HTTP (ブラウザ) , HTTPS (ブラウザ) *, SSH (コンソール等) FTP (ファイル交換)	URL
第4層	トランスポート層	TCP , UDP	IP アドレスとポート番号
第3層	インターネット層 (ネットワーク層)	IP (IPv4, IPv6)	IPアドレス
第1・2層	ネットワークインタフェース層 (データリンク層)	媒体アクセス制御, Ethernet, Wifi	MAC アドレス

* 厳密に言えば、HTTPS自体はプロトコルではなく、SSL/TLSプロトコルによって提供されるセキュアな接続の上でHTTP通信を行うことをHTTPSと呼んでいる。 <https://ja.wikipedia.org/wiki/HTTPS> より



階層プロトコルにおける データ伝送



TCPの場合では、
データ自体も分割される。
次のページ

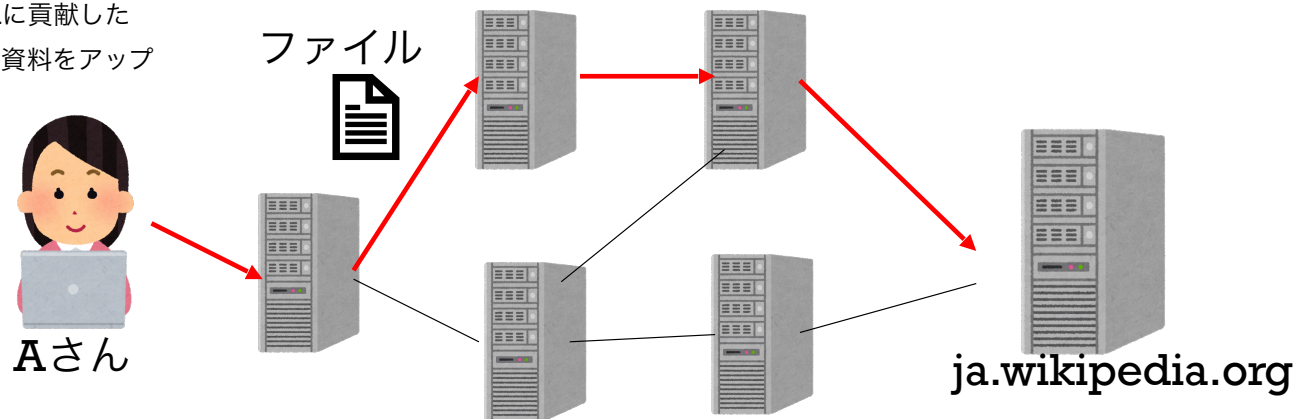
https://ocw.u-tokyo.ac.jp/lecture_files/engin_09/2/notes/ja/aida02.pdf より



TCPによるデータ分割

- もし大きいデータ（例えば1GBのファイル）を分割せずに送るとどうい問題があるでしょうか。

AさんがWikipediaに貢献した
く、大きいサイズの資料をアップ
ロードする。

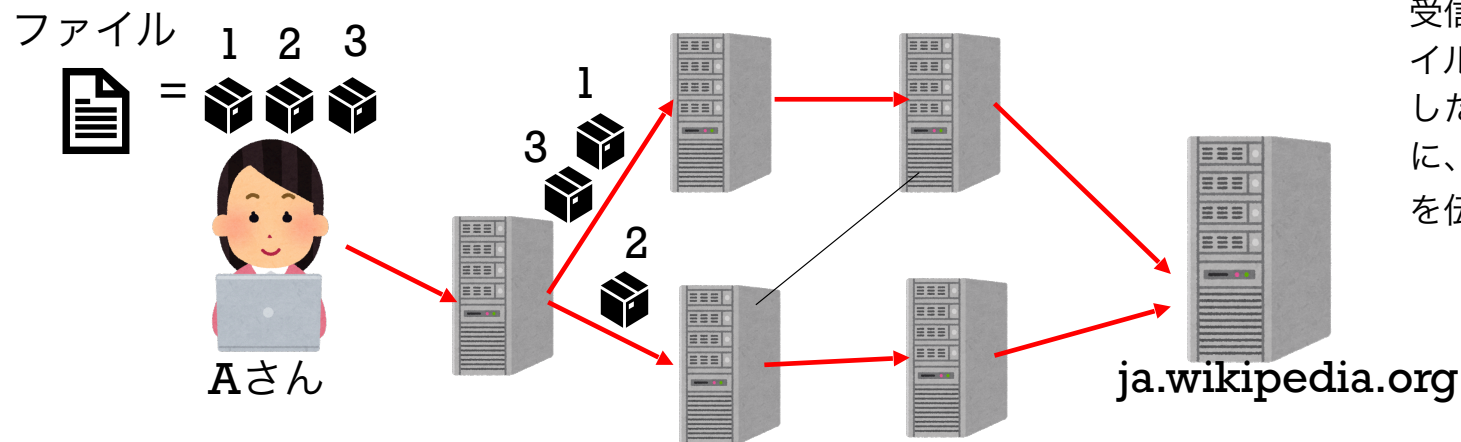


- データが一箇所で失ったら、すべてのデータを再送信する必要がある。→通信の時間が遅い。



TCPによるデータ分割

- TCPがデータを小さいサイズの packets (packet) に分割する。



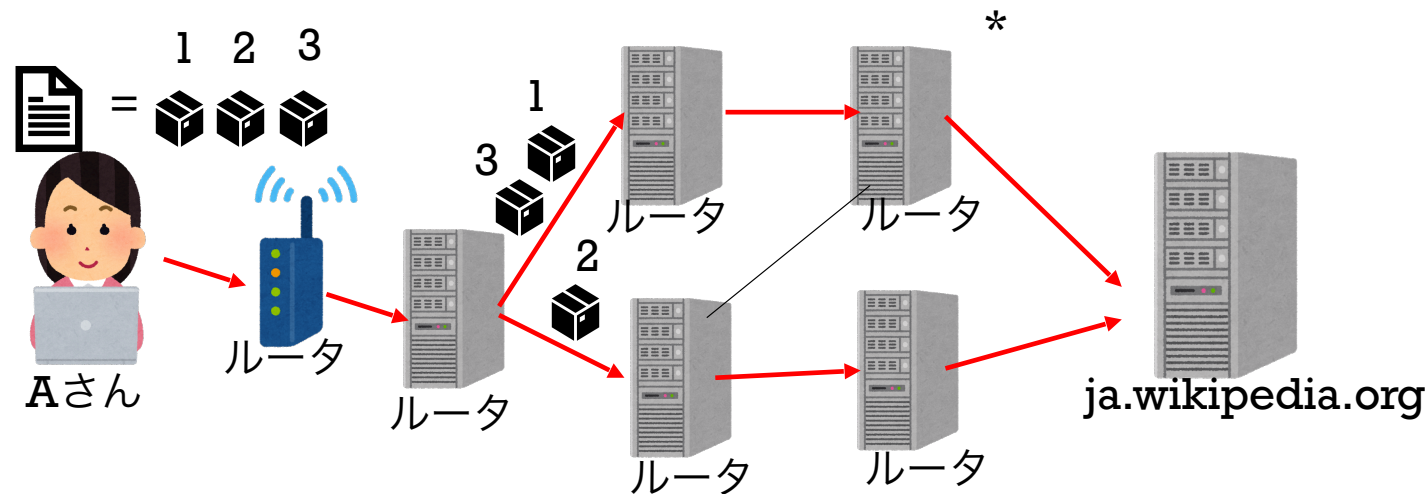
受信側では各パケットから元のファイルを復元。受け取っていない消失したパケットが再送信されるように、Aさんに受信済みのパケット番号を伝える。

- Packetが別の経路に送れるため、インターネットのルータの一部が不具合になっても、問題ない。
- TCPはデータが完全に受信されることを確保する。そのために失ったパケットを再送信したりする。
- ZoomなどでReal Time処理が求められる際にはTCPが不適切で、その代わりUDP等が利用される。



ルータ (ROUTER)

(1/2)



- ルータがインターネット(または社内のネットワークなど) でPacketの転送を担当
- ルータが受信したPacketを次にどこのルータに転送すればよいか**IPアドレス**によって決める。
- 方法：
 - 静的経路制御：転送先を静的に決める（手動で設定した経路表を参照）。
 - 動的経路制御：経路表を自動的に更新する。

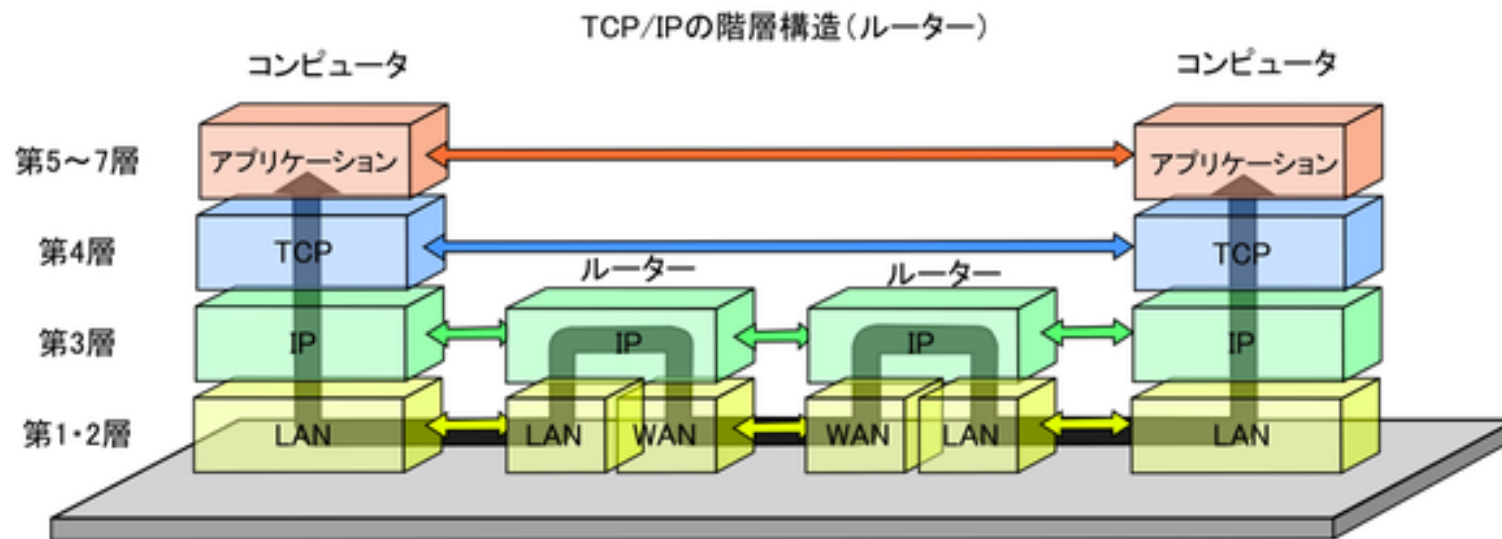
*図は主なインターネットにおける通信を説明するもので、実際にはSwitchなども利用されたり、第1・2層、または第4層以上の処理を行っているノードもある。



ルータ (ROUTER)

(2/2)

- ルータはネットワーク層で処理を行っている。IPアドレスの情報を利用しているから。

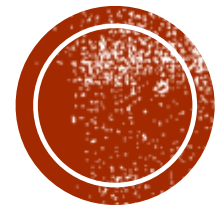


<https://ja.wikipedia.org/wiki/ルーター> より

一方、SwitchとHubはデータリンク層で処理する。

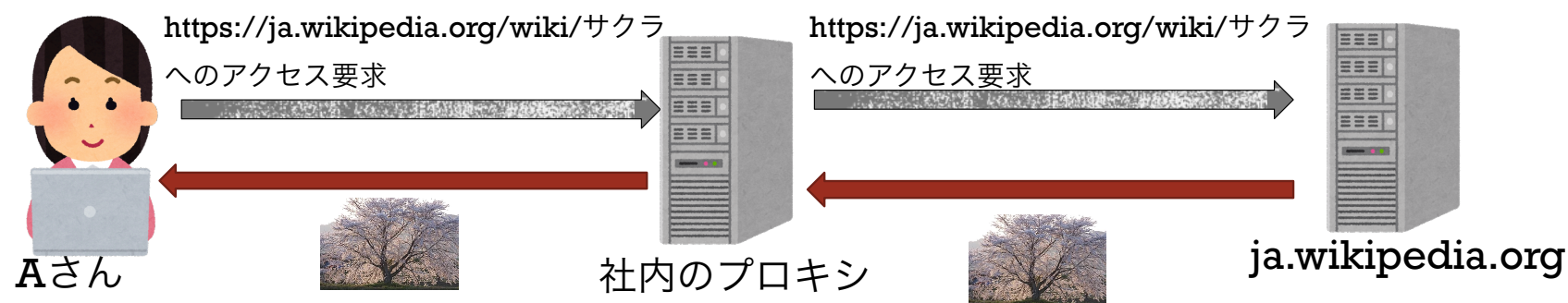


質問タイム



補足：プロキシ (PROXY)

Proxy とは代理人の役割を果たしているサーバ。



例えば、**Aさん**の要求された**Web page**がまず**Proxy**に送信されて、**Proxy**が**Aさん**の代わりに**Web page**を獲得して、転送する。

メリット：

- プライバシー（ウェブサイトのホストサーバ**B**が**Aさん**の代わり**Proxy**と通信するので、**B**が**Aさん**の存在知らない。）
- セキュリティ（危険なウェブサイトへのアクセスを拒否する）
- アクセス速度の向上（直近同じ要求があった場合キャッシュから応答できるから）

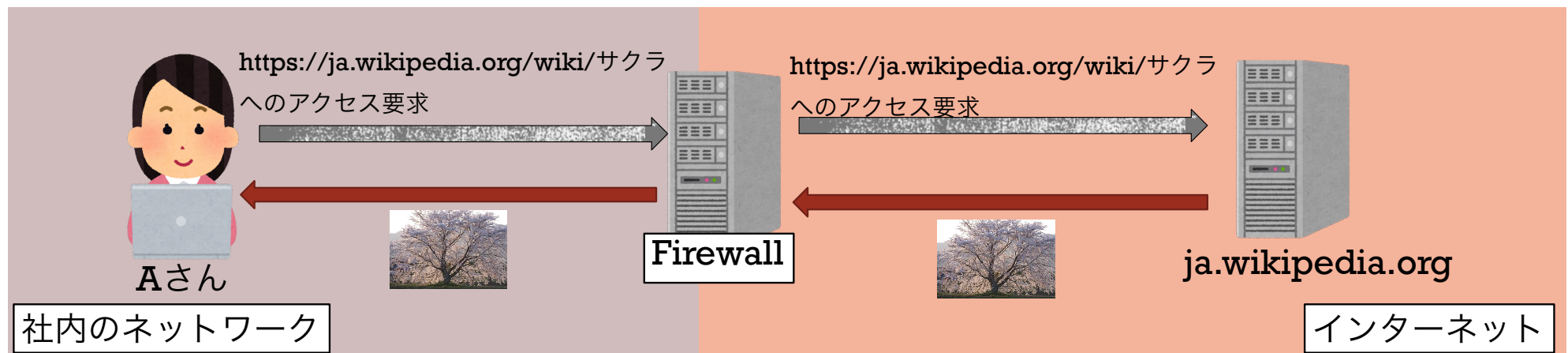
多くの企業に利用されている。

（私は社内からインターネットと繋がらない時にはほとんどいつもプロキシ設定の問題が原因だった☺）



補足：ファイアウォール (FIREWALL)

- Proxyと似ているが、役割はセキュリティの確保。
- Firewallはインターネット層以降(インターネット層・トランスポート層・アプリケーション層) で危険な通信を選別する。



例えば、

- 社内からの危険なウェブサイトへのアクセスを拒否する。
- 社外からの怪しい通信を遮断。



補足：レイテンシ(LATENCY)

- レイテンシ (Latency), 反応時間ともいう。
 - 物理的な限界がある。例えば、光ファイバーケーブルでは光の速さが限界。
 - あるノードAからあるノードBまでの通信の時間。
 - 一周時間 (Round Trip Time)では A->B->Aの時間, “ping”コマンドで測れる。

- 例:

```
$ ping www.google.com
PING www.google.com (172.217.26.4): 56 data bytes
64 bytes from 172.217.26.4: icmp_seq=0 ttl=49 time=180.541 ms
64 bytes from 172.217.26.4: icmp_seq=1 ttl=49 time=60.351 ms
64 bytes from 172.217.26.4: icmp_seq=2 ttl=49 time=78.893 ms
64 bytes from 172.217.26.4: icmp_seq=3 ttl=49 time=55.858 ms
64 bytes from 172.217.26.4: icmp_seq=4 ttl=49 time=74.487 ms
64 bytes from 172.217.26.4: icmp_seq=5 ttl=49 time=51.687 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 51.687/83.636/180.541/44.406 ms
```

例え：道路の整備
石畳 **vs** アスファルト舗装



補足：スループット

(THROUGHPUT, BANDWIDTH CONSUMPTION)

- どれぐらいのデータ量を一定時間内に通信できるかを示す。
 - ネットワークの状況（混雑など）によってスループットが悪くなる。
 - 最大のスループットは帯域幅(Bandwidth)*と呼ぶ。
-
- 例: G5のBandwidth は10 Gbit/s (gigabits per second)**

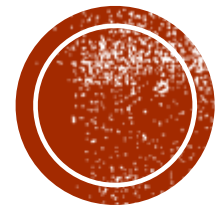
例え：車線の数
(Bandwidth)
交通状況によって
実際に使える車線
(Throughput)

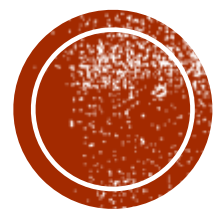
*信号処理分野ではBandwidthの意味が異なります。

** <https://en.wikipedia.org/wiki/5G> による



質問タイム





通信における セキュリティ

通信におけるセキュリティ

■課題

- 秘密を守る： 盗聴による情報漏洩を防ぐ
- 相手を認証する： なりすましからの防衛。
- 元のデータを確実に受け取る： 改ざんを検知

■対策方法

- 暗号化と復号
- デジタル署名



暗号化(ENCRYPTION)と 復号(DECRYPTION)

- 暗号化：読めないようにデータを書き換える。
- 復号：暗号化したデータを元のデータに戻す。

元のデータ：平文（ひらぶん, plain text）

暗号化と復号を実現するためには、以下の仕組みがある。

- 共通鍵暗号
- 公開鍵暗号



共通鍵暗号 (SYMMETRIC KEY ENCRYPTION)

- 送信者が鍵をもって、データを暗号化する。
- 受信者が受け取っているデータを**同じ鍵**で復号する。
- 例：
 - 入力：平仮名で書いている文書
 - アルゴリズム：五十音をx文字ずらして置き換える
 - 鍵：ずらしている文字数x
- 実際によく利用されているアルゴリズム： 3DES, AES



公開鍵暗号 (PUBLIC-KEY ENCRYPTION)

- 公開鍵暗号方式が**鍵のペア**を作成する。
 - 公開鍵 (**public key**) : 皆に知らせてよい。
 - 秘密鍵 (**private key**) : 秘密に保持する。
- 片方は暗号化のため、もう片方は復号のために利用される。応用によって役割が決まる。
- **公開鍵が分かったとしても、秘密鍵の推測が不可能。**
- 公開鍵で暗号化したデータは対応している秘密鍵でしか復元できない。
- よく利用されている公開鍵暗号方式：RSA, DSA
- 応用の例：**秘密文書の暗号化**
 - 秘密鍵：復号用
 - 公開鍵：暗号化用
 - AさんがRSAで鍵のペアを作成し、公開鍵を友達に渡す。友達がAさん宛にメールを送る際にはAさんの公開鍵を利用して、メールを暗号化する。効果：Aさんしか復号できない。他の友達も復号はできない。
- 他の応用例：**デジタル署名** (HTTPSやSSHで利用されている)

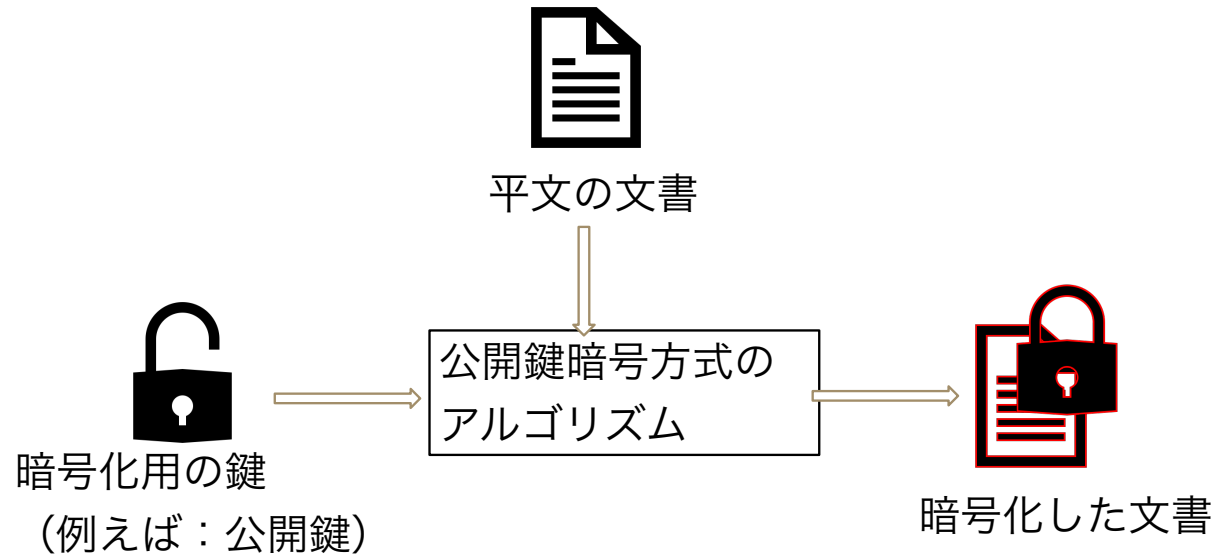


公開鍵暗号 (PUBLIC-KEY ENCRYPTION)

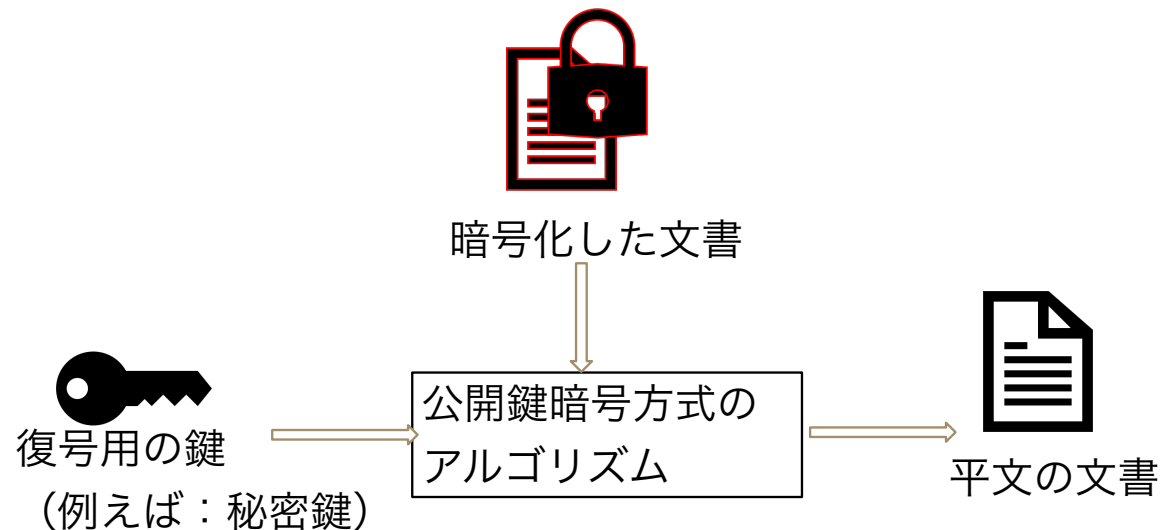
公開鍵暗号方式によって作成した鍵のペア：



暗号化



復号



お試し＋質問タイム（１０分）

共通鍵と違って、ユーザが公開鍵と秘密鍵を自分で決めるわけではなく、公開鍵暗号方式が自動的に作成してくれる。

- 以下のコマンドで鍵のペアを作成してみてください。
`ssh-keygen -t rsa -b 4096 -C "your_name@rikkyo.ac.jp"`
- ファイ名は「testKey」にしてください。
- 今回はPassphraseを空欄にしてもよい。
- 公開鍵(testKey.pub)と秘密鍵(testKey)が現在のDirectoryに保存される。
- testKey.pubとtestKeyの中身を確認してみてください。
(テキストエディタやlessのコマンドを使ってください。)

詳細：

<https://compbio.cornell.edu/about/resources/linux-ssh-keys-and-ssh-key-generation/>



デジタル署名

デジタル署名によって、改ざんの防止が可能。

デジタル署名を実現するために以下の技術が利用される。

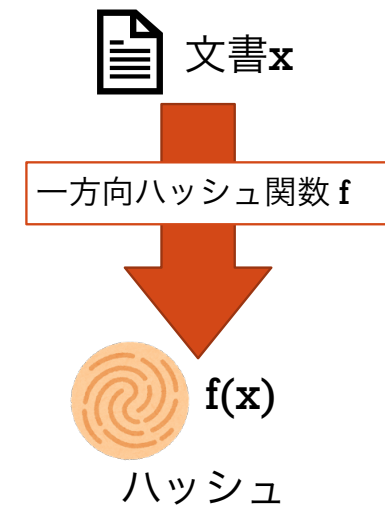
1. 公開鍵暗号 （前のスライドで説明済み, これから「g」で暗号化を表す）
2. 一方方向ハッシュ関数 （これから、「f」で表す）



デジタル署名の背景

一方向ハッシュ関数 (CRYPTOGRAPHIC HASH FUNCTION)

- 一方向ハッシュ関数 f が入力 x を受けて、値 $f(x)$ を返す。
この場合、 $f(x)$ はハッシュと呼ぶ。
- 特徴：
 - 元の入力の推測ができない。
 - 入力が元の x でなければ、高い確率で出力されたハッシュも違う。
つまり、 $x \neq y \Rightarrow f(x) \neq f(y)$ 。
- 改ざんの防止に利用される。
- よく利用されているアルゴリズム：MD5, SHA-1。

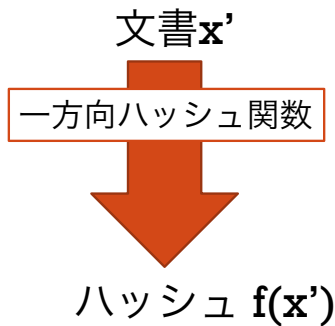
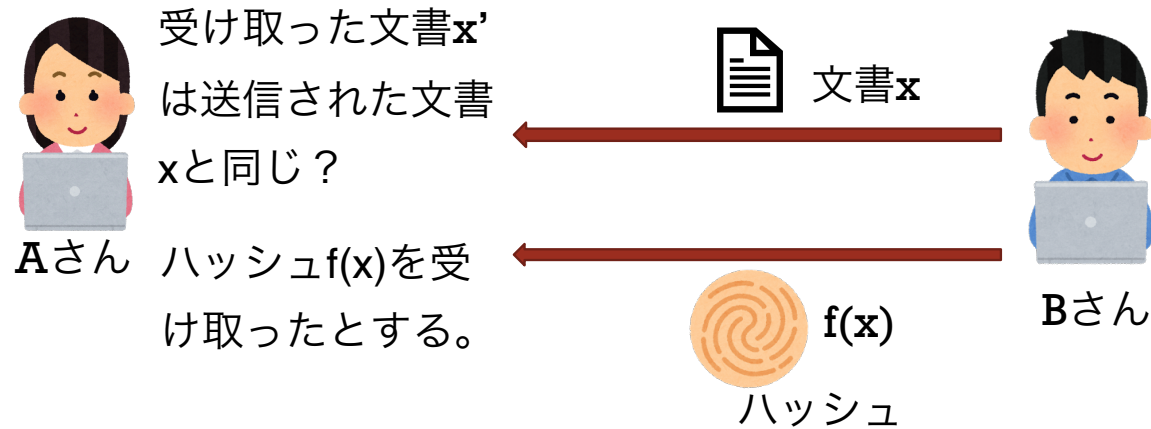


セキュリティ分野では $f(x)$ を
指紋(fingerprint)と呼ぶこ
とが多い。

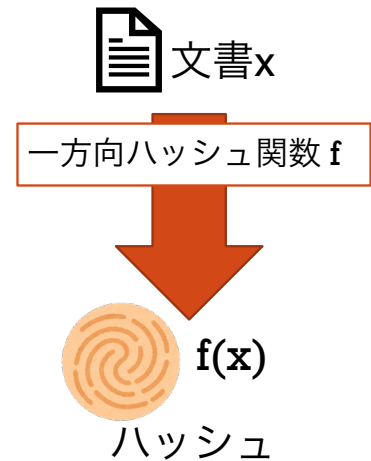


一方向ハッシュ関数



改ざん防止の例



$f(x') = f(x)$ かつ $f(x)$ が改ざんされなかったら、
受け取った文書 x' と送信された文書 x が一致している。



デジタル署名 (DIGITAL SIGNATURE)

- 秘密鍵：暗号化用 
 - 公開鍵：復号用 
 - 一方方向ハッシュ関数：文書の指紋を作成
- 40pの応用の例と比べると、公開鍵と秘密鍵の役割が変わった。

Aさんが確実にBさんの公開鍵を所持しているとする。


Bさんの公開鍵



受け取った文書 x' は
送信された文書 x と同じ？

 文書 x



Bさん

  $g(f(x))$

文書 x'

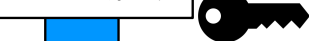
一方方向ハッシュ関数



ハッシュ h_1

$g(f(x))'$

Bさんの公開鍵



ハッシュ h_2

ハッシュ $h_1 =$ ハッシュ h_2 であれば

- 1.) ハッシュ h_1 はBさんの秘密鍵で暗号化された。
- 2.) 受け取った文書 x' と送信された文書 x が一致している。

文書 x

一方方向ハッシュ関数



ハッシュ $f(x)$

Bさんの秘密鍵 

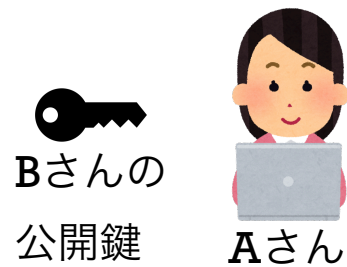


$g(f(x))$

デジタル署名

デジタル
署名の作
成

なぜかという、ハッシュ h_1 とハッシュ h_2 が等価になるように、
文書 x とデジタル署名 $g(f(x))$ を両方も改ざんするのが困難だから。



公開鍵への署名

問題：

- Aさんがどのように確実にBさんの公開鍵をもらえる？
インターネットを通して受け取った鍵が本当にBさんのものかを確認するためにはどうすればよいでしょうか。

解決：

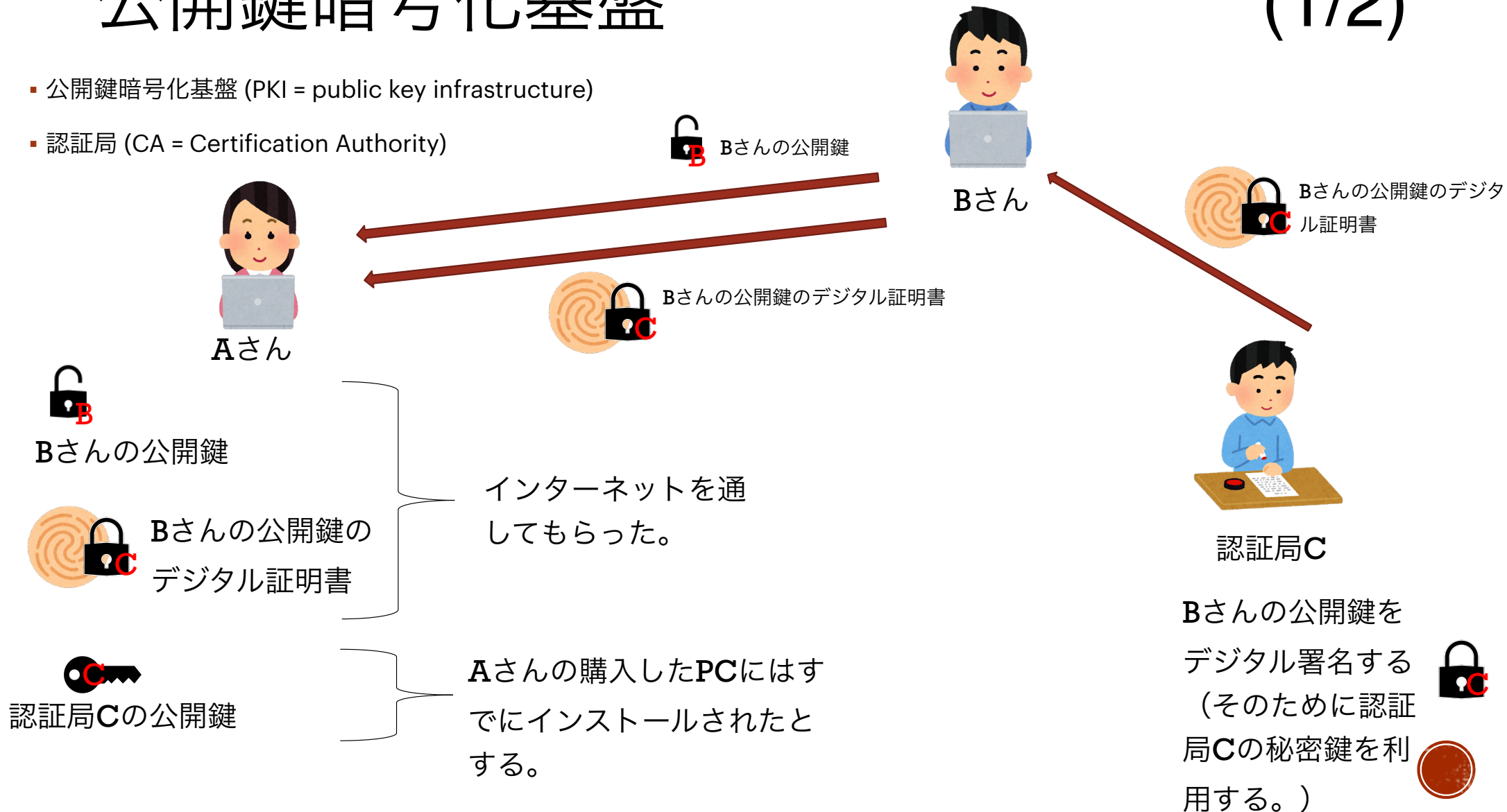
- Aさんが信頼している認証局Cの公開鍵を持っているとする。
(例えば、購入したPCにはすでにインストールされている。)
- 認証局CがBさんの公開情報（名前+Bさんの公開鍵）をデジタル署名する。
(つまり認証局Cが自分の秘密鍵を使って、Bさんの公開情報から作成したハッシュを暗号化する。)



公開鍵暗号化基盤

(1/2)

- 公開鍵暗号化基盤 (PKI = public key infrastructure)
- 認証局 (CA = Certification Authority)



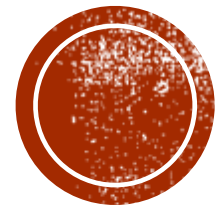
公開鍵暗号化基盤

(2/2)

- Bさんの公開鍵だけではなく、Bさんに関する情報もCにデジタル署名される。
たとえば、正式な名前、住所など。
- 認証局Cが正しくデジタル証明書を発行すると信頼できると、
認証局Cからデジタル署名されたBさんのデータ（公開鍵など）にも信頼できる。



質問タイム



補足：通信以外のセキュリティ問題

当たり前ですが、通信のセキュリティが保証されても、ClientやServerに保存したデータにおけるセキュリティのリスクがある。対策の例：

- データの管理（暗号化した状態での保存、データへのアクセス権限の管理など）
- パスワードの管理
- Virus Softwareによって悪意のソフトウェアや侵入者を早期に発見。
- など



本日のまとめ

- インターネットの仕組み
 - インターネットに繋いでいるすべての機器は一意的なIPアドレスを所有している。
 - IPアドレスが付与されて、変わる可能性がある。
 - DNSの仕組みでIPアドレスを調べることができる。
 - インターネットにおける通信規約 (Protocol)は四つのLayerに階層的に分類できる。
- 通信におけるセキュリティ
 - 暗号化と復号のために二種類の方式がある：共通鍵暗号と公開鍵暗号
 - 公開鍵暗号の場合では鍵のペア：「公開鍵 (public key)」と「秘密鍵 (private key)」が作成される。
 - 公開鍵暗号は秘密を守るための暗号だけではなく、デジタル署名にも利用されている。
- 次回：HTTPにおけるサイバー攻撃とその対策(HTTPS) と SSHに関する補足



次回（5月8日）のための準備

復習・予習のために以下を読んでおいてください。

- 第4章「情報の伝達と通信」
- 第5章「計算の方法」
- 第6章「計算の理論」

山口和紀, 情報第2版, 東京大学出版会(2017)



レポートの課題

- 締め切りは**5月7日の23時55分**です。
締切までは再提出可で、最後に提出されたものを採点します。
- レポートに関する質問は5月6日までにお願いします。
(締め切りの当日に私はメール・Slackのメッセージに返事できない可能性があるため。)
- 解答の方法：
最初の一行や一段落で、解答を簡潔に書いてください。
その後に、解答に至るまでの考え方・補足・詳細を簡潔に書いてください。
- docxファイルとして提出しないでください。pdfに変換したもののご提出をお願いします。



レポートの課題

1. IPアドレスとMACアドレスの共通点と違いを思い出して、
それぞれの発行者を書いてください。
2. AさんがBさんに自分の公開鍵をメールで送る。Bさんが受け取った公開鍵で社内の機密文書を暗号化する。どなたがその暗号化した文書を復号できるか。

Aさん、Bさん、第三者のCさん？

一つの正解はないので、**自分の回答に利用する仮説も明記にしてください。**

