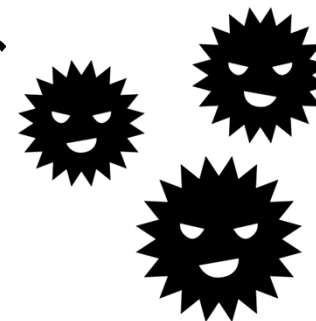


インターネット(Ethernet)を流れるパケットは、
経路上であれば誰でも傍受できる

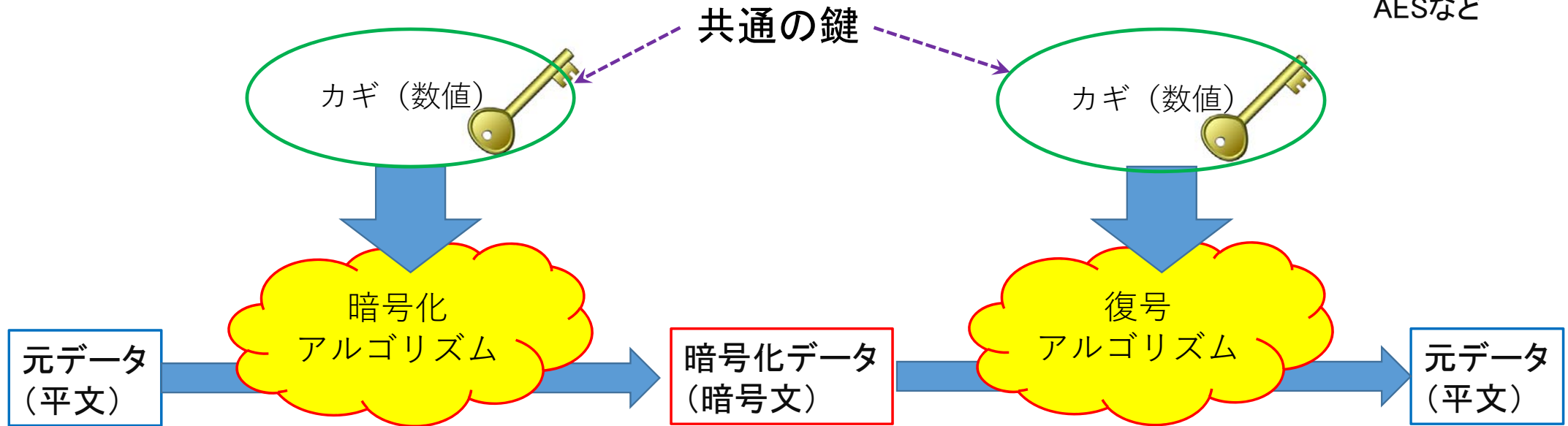


- ・通信内容を秘匿するための技術: **公開鍵暗号**
- ・通信相手の偽装(なりすまし)を防止する技術: **認証**

共通鍵暗号方式による暗号化

処理速度は速いが鍵の管理は困難

DES暗号(1977)
IDEA
AESなど



- ・暗号化／復号アルゴリズムは公知(20世紀半ばまではこれらも秘匿)
- ・第3者に対してカギを秘匿
- ・暗号の安全性は計算量理論に基づく

この条件を満たすのは難しい

N人の間の交信において

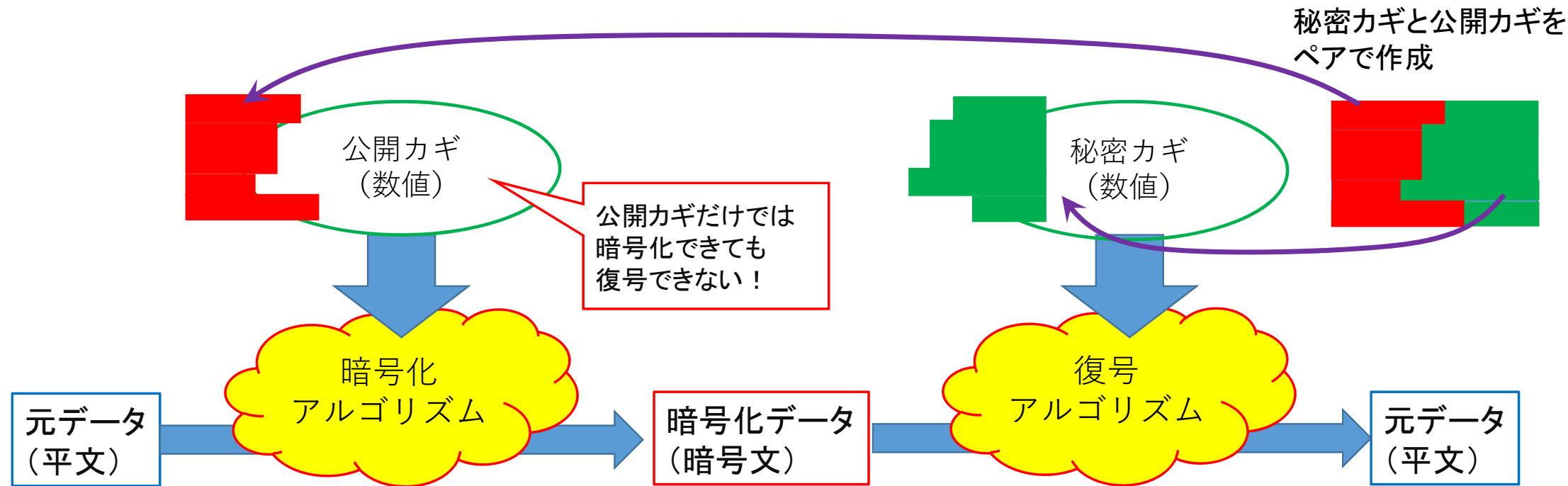
$${}_NC_2 = \frac{N(N-1)}{2}$$

の種類の鍵が必要

=1000人の通信に499500の鍵

公開鍵暗号方式による暗号化

鍵の管理はしやすいが処理がとても重い



- ・暗号化／復号アルゴリズムは公知
- ・秘密カギを秘匿、公開カギを相手に公開
- ・暗号の安全性は計算量理論に基づく (素因数分解の計算量)

秘匿すべき
カギの情報を
通信せずにすむ！

N人の間の交信において
 $2N$ の種類の鍵が必要
＝1000人の通信に2000の鍵

実用的公開鍵暗号: RSA暗号

公開鍵

$$N = pq$$

p, q は200桁以上の素数
これらは秘匿

$$L = \varphi(pq) = (p-1)(q-1)$$

1から pq の数の中で $pq=N$ に
対して互いに素な数の個数

公開鍵

$e : L$ とは互いに素な 1 より大きい自然数

秘密鍵

$d : e \cdot d$ を L で割ったとき余り1となるような数

・オイラーの定理

$m^L = m^{\varphi(N)}$ は N で割ると余り1

$e \cdot d = 1 + tL$ ただし t はある自然数
よって

$$m^{ed} = m^{1+tL} = m \cdot (m^L)^t$$

N で割ると余り m

平文 m (L より小さい数字で表す)

暗号化 (暗号文) $c = m^e \pmod{N}$

公開鍵 e, N と暗号文 c が
あっても平文 m は復元
できない

復号化 $m' = c^d \pmod{N} = m$

RSA暗号の簡単な計算例

公開鍵

$$N = 3 \times 13 = 39$$

1からpqの数の中でpq=Nに対して互いに素な数の個数

$$L = \varphi(pq) = (3 - 1)(13 - 1) = 24$$

公開鍵

$e = 5 : L$ とは互いに素な 1 より大きい自然数

秘密鍵

$d = 5 : ed$ を L で割ったとき余り1となるような数

平文 $m = 37$ (L より小さい数字で表す)

$$39 \times 1778050$$

暗号化 (暗号文) $c = m^e \pmod{N} = 37^5 \pmod{39} = 69343957 - 69343950 = 7$

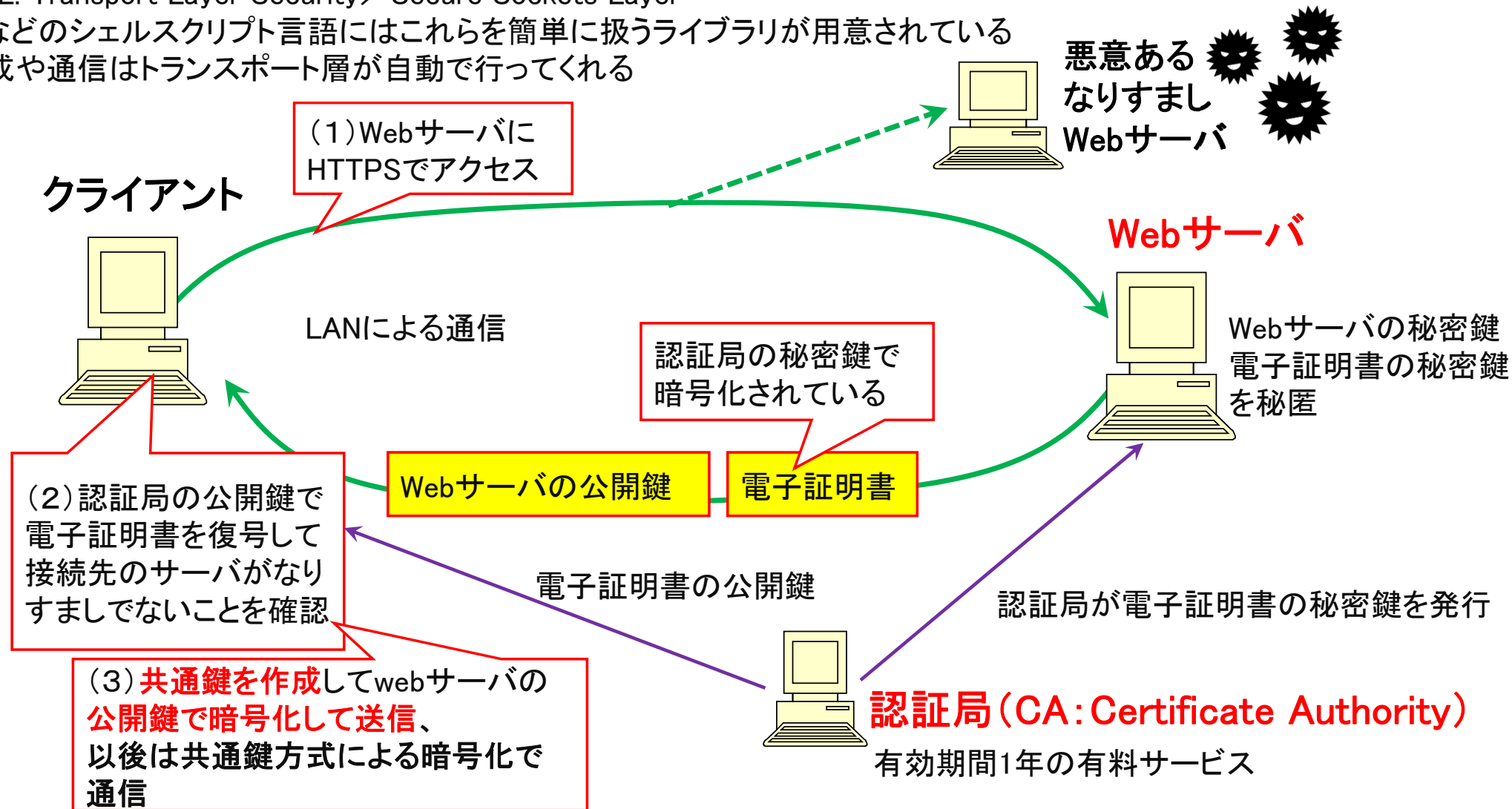
$$39 \times 430$$

復号化 $m' = c^d \pmod{N} = 7^5 \pmod{39} = 16807 - 16770 = 37$

TLS/SSLによるHTTP通信の暗号化: HTTPS

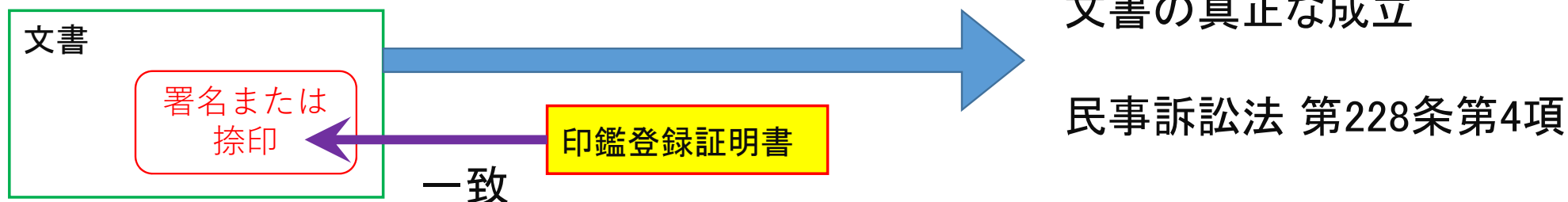
TLS/SSL: Transport Layer Security / Secure Sockets Layer

Pythonなどのシェルスクリプト言語にはこれらを簡単に扱うライブラリが用意されている
鍵の生成や通信はトランスポート層が自動で行ってくれる

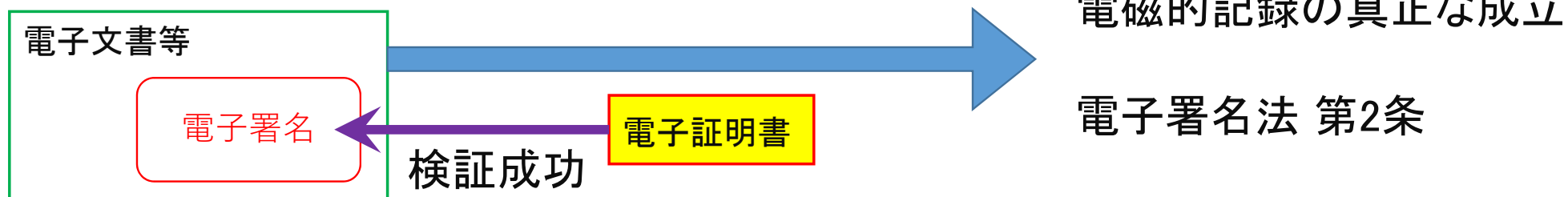


電子署名や電子証明書に関する法令

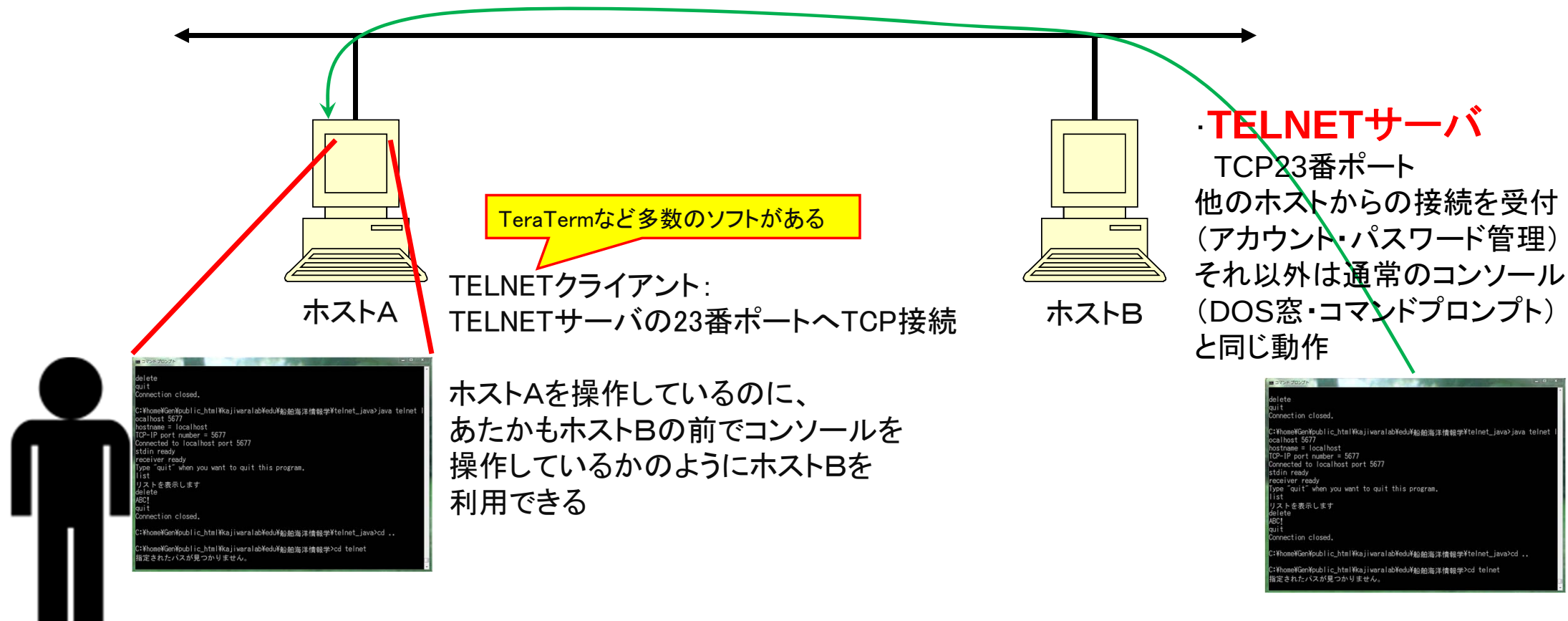
手書き署名・捺印



電子署名



【復習】遠隔ログイン(TELNET と SSH)



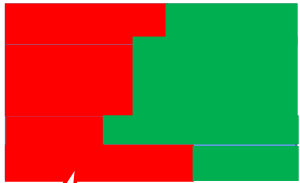
TELNETは、アカウントやパスワードも含めた全てのパケットが暗号化されずにそのままネットワークを流れていくため、**セキュリティ上危険**な通信

SSHは、これらの**通信を暗号化**

SSHによる遠隔ログイン

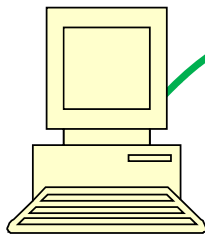
SSH: Secure Shell

秘密カギと公開カギを
ペアで作成



クライアント側で
秘密鍵を保持

SSHクライアント

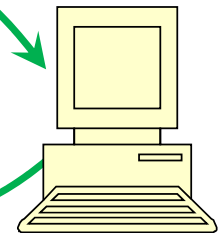


LANによる通信

SSHサーバに
公開鍵を送る



SSHサーバ



通信確立以後は共通鍵方式による
暗号化で通信(多分)

SSHで公開カギを作成した例

ssh-rsa

AAAAB3NzaC1yc2EAAAABIwAAAQEAwXO9g+7aM0d0CTZjh+wxa24pX58M7oUuvfd3/PIXojHb5zDE7mOWqyeonKuSMxalDcxNm1aDAJ5zI
p8tLZpqASCC3RkIVUTvuu9etxRAQkGb3l8GdwoDuRqd+RlZx86fpWPbk4lkkm7h7PYxdIX7/uomp5s4EQAiYCzii+MWn1QDBvIxWhZxItgEiRj+8
QPqpxZRJUsq4xPhfBHNjDH/CpWtzu1O2z0NLet6O3bEjC4DF96XJWhMsR6KiqtmMb9OohuWQOgADXfnwqj8kbXtJ++sbgWF+uCIMgozwYB+
xKoMwc0BndXLJEQNFzKRbGjqc7U2W18AyEjGW/RiV5kIcQ== kimura@BragZakato2

上記のペアになる秘密カギ

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4,ENCRYPTED

DEK-Info: AES-128-CBC,0CA460B2D8E6FB0571261B7EDE9E979E

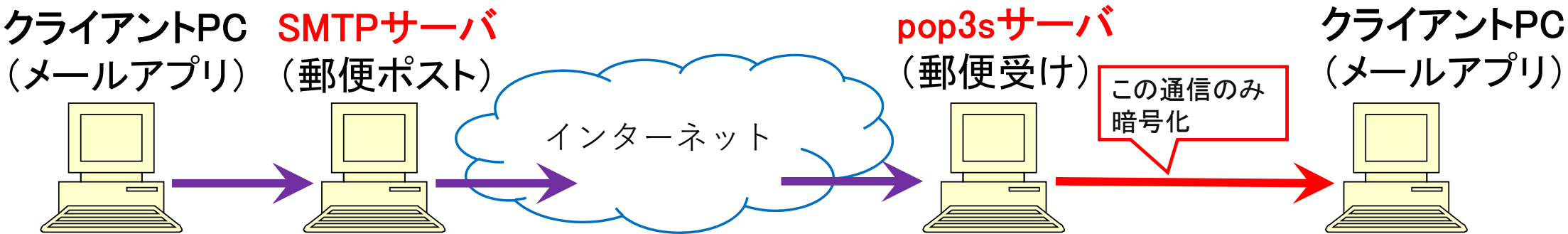
PjRn19wpXug8cufv2HQbsikyxJ/jwYlcZJnEPfO6ji4SR8JNDNiQamNDdK5fr8Ds
bbbV2Sbi6re1s4UYDJxTSKhBQm+9zNDNHH7vZWH+Z2B5hd8ziaJWamK7Q5SXR7Qq
hHtDaIco7kakVkcZQEG/Mp07HSeOyGm+oSIBbtn5QsHUJbOVYRIT+sVTd4YxwE0+
35TMvRK76Q7FqiyAErLw1S7dEHHmIECnXi4TGviAvwK7x9jeGPlxoMGozsXBf/DW
eyIxsBP9zebNaGSowiyukbhkk1SBnpJknT2IPAvWri7uAxBBWDO8NY7Tho5phUwo
4/a5RHkUWtQZ88VeWewXyoNCVUbjvWcJfZjdCbh+ff7ckvkdLGLjh88NgNis/mxs
rbuI9Oa48tIMioNitSmDZ9UIHCWRGoWQAtDriIW5ksX8x7EAFDJDikSTJuSz1LQC
zCQ3UGm/chfIS3TCG3no6OiCS1zAoVIuhC8pS+xn/c11X1nMAZP0hsHCMVHXjYDy
mn8eiLiqfI0IBm8Vp5dT9ypuIJRI8uEcZnLeWv6dLPA5HmAaZsVNBi++TE/OCfLj
BxSIlhPBVlsgz+cDAVJc36qfTI6TdNCuhN/aEVF2DwRj1hi/sv7PDVyeuBQyXr
zLMX0BtL0CfMIYXb8FIHi41IVYKkoUFXCbrOpbq3kPPa3/MJU2LN7R5zb8bDcF6
qyT3SKSL/2S185aGaJfN08tzDg4ak1MAvTpAGXmc5R3vSLZ77MzvneKnIfAZXkKS
+IKno56GJL0JFHbJIDHE2hEdUxjx1VVbJs4/PBAGQ8CB1vZfEtOBC6AINMUtT77u
doRKmPO0ITsZ5kGr9GI10QdAkZ9i9GoLokBqcGBzZcLQ6a83X6rbMfRFQ5mACyaB

電子メールの暗号化

TCPの
代表的な
ポート番号
とサービス

・25番	smtp	Simple Mail Transfer Protocol	郵便ポスト=メールを出す
・109番	pop2	Post Office Protocol Ver.2	郵便受け
・110番	pop3	Post Office Protocol Ver.3	郵便受け
・995番	pop3s	pop3 protocol over TLS/SSL	郵便受け(暗号化)
・143番	imap	Internet Message Access protocol v2,v4	郵便受け
・220番	imap3	Internet Message Access protocol v3	郵便受け
・993番	imaps	imap4 protocol over TLS/SSL	郵便受け(暗号化)

暗号化に対応するものが無い！



電子メールにメッセージの暗号化と本人認証の機能を取り込む仕様として PEM (Privacy Enhanced Mail) が規定

メールやHTTPなどMIMEデータ転送を暗号化に拡張したものとして S/MIME (Secure Multipurpose Internet Mail Extensions) 認証局を必要とする

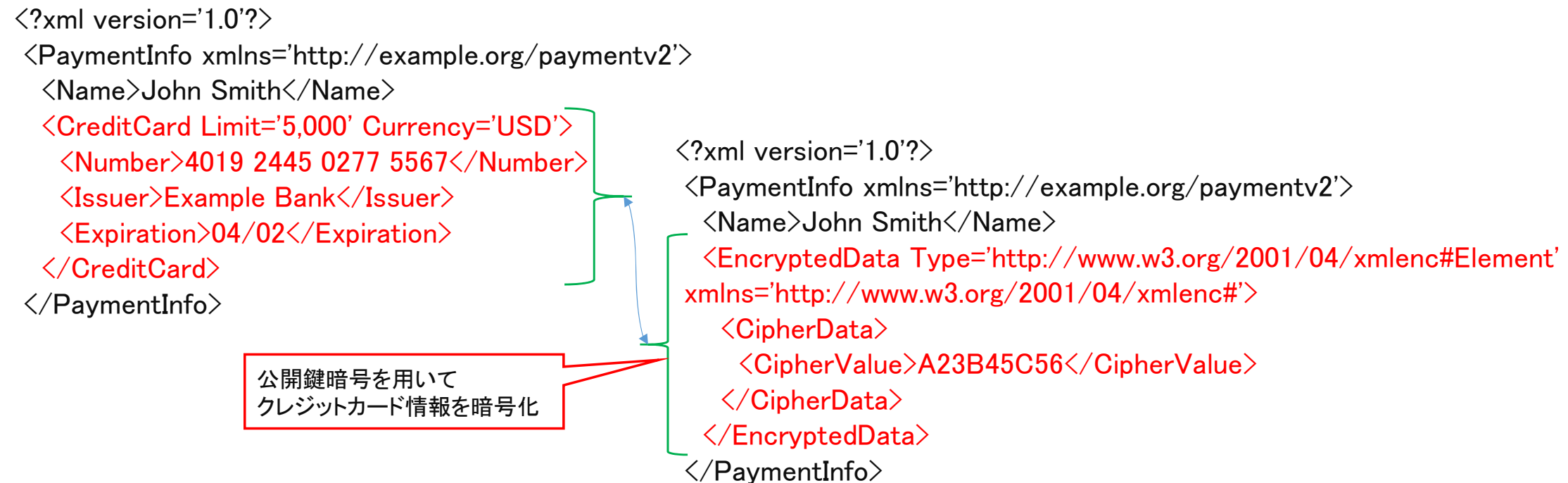
暗号パッケージソフト: PGP (Pretty Good Privacy)暗号化や署名ができる、認証局を用いない

メールソフトに
組み込んで使用

XMLの暗号化 (XML Encryption)

- ・ W3Cに提案されているXMLの暗号化技術で、ツリー構造XML文章のうち、ある要素以下を全て暗号化する仕組み。
ルートタグを暗号化すればXML文章全体の暗号化も可能。またデータ保護項目の優先度付けを行い、特定の要素のみ暗号化することも可能

W3Cの「XML Encryption Syntax and Processing」に紹介されているXMLエレメント暗号の例



まとめ

- (1) 現代の暗号:暗号化・復号アルゴリズムは公開し、カギにより秘密性を確保
- (2) **共通鍵暗号**: 計算は速い／鍵の管理が困難
- (3) **公開鍵暗号(RSA暗号)**: 鍵の管理は容易／計算が重い
- (4) **TLS/SSL**によるHTTPの暗号化: **HTTPS**
共通鍵暗号と公開鍵暗号の両方を用いて欠点を克服
第3者の「認証局」と暗号化された電子証明書により「なりすまし」を防止
- (5) 電子証明書に関する法令: 電子署名法
- (6) SSH(Secure Shell)ではクライアント側が秘密鍵を保持
- (7) 電子メールは通常セキュリティが無いので注意

演習課題 :

以下のサイトより公開鍵暗号に対応したファイル暗号化ソフト **FileCapsule Deluxe Portable** をダウンロード

<https://forest.watch.impress.co.jp/library/software/fcdx/>

ワード等で自分の氏名と学籍番号を記載したファイルを作成し、上記のソフトに以下のサイトにアップロードされている公開鍵ファイルを使って暗号ファイル化せよ。

<http://sysplan.nams.kyushu-u.ac.jp/gen/edu/NavallInformationProcessing/2018/index.html>

【提出方法】

念のためメール本文に各自の氏名を記述し、上記の暗号化ファイルを直接メールに添付して、
kimura@nams.kyushu-u.ac.jp宛てに「第8回船舶海洋情報学レポート」とのタイトルで送信せよ。

誤り訂正符号

多少ノイズで文字列やパターンが変化しても元のデータを復元できる

例) QRコード



<http://sysplan.nams.kyushu-u.ac.jp/gen/hobby/puzzle/puzzlej.html>

→ これらの符号化・復号化の理論を基に暗号化する研究もある