Webを支える情報通信技術(10)
～ SSL/TLS ～

佐藤周行、中山雅哉、西村健

先週の復習

- 暗号技術
  - 共通鍵暗号
  - 公開鍵暗号
    - 鍵の受け渡しがキモ！

今週の内容

- ハッシュ関数
- SSL/TLS

（一方向性）ハッシュ関数

- 改竄防止のための技術
- 可変長の文から固定長の値（ハッシュ値）を導き出すもの
- 送信元で計算したハッシュ値と送信先で計算したハッシュ値を比較することにより、途中で改竄されていないことを確認できる

- アルゴリズム：MD5（RFC 1321、古い）、SHA-1（FIPS 180-2、RFC 3174）など

演習 : ハッシュ関数を使ってみよう(1)

- 「abc*[改行]*」という文字列のMD5ハッシュ値を計算する

```
$ openssl md5
abc
[Control + D]
…
```

- 「abc」という文字列のSHA-1ハッシュ値を計算する

```
$ openssl sha1
abc[Control + D][Control + D]
```

文中での入力終了は
Control+Dを2回

演習 : ハッシュ関数を使ってみよう(2)

- 「a」が1,000,000個並んだ文字列のSHA-1ハッシュ値を計算する
  - Safariで
    `http://www-sato.cc.u-tokyo.ac.jp/PKI-project/Komaba/hashfunction3.txt`
    を開き「ファイル」→「別名で保存」で保存する

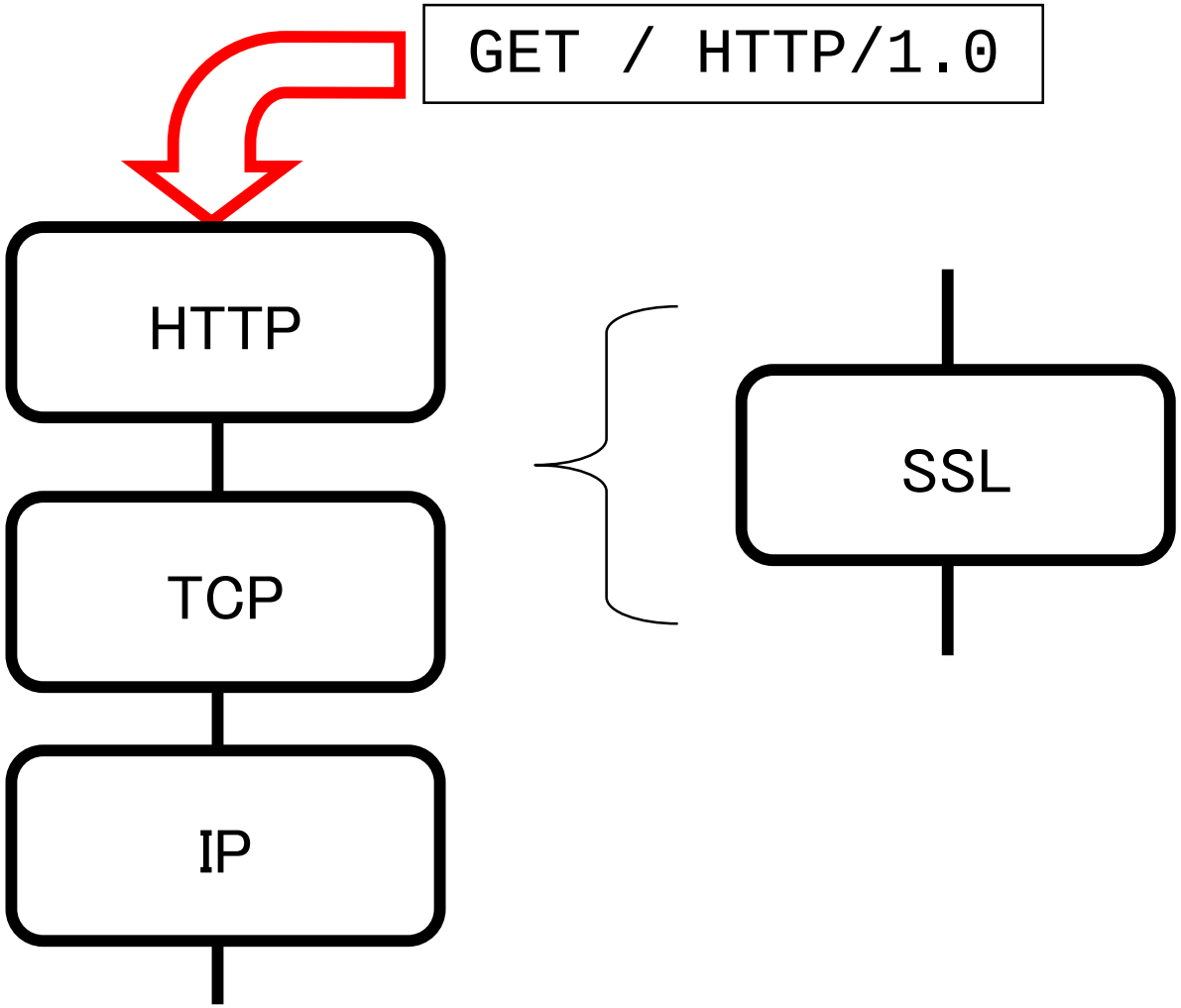
```
$ openssl sha1 < Desktop/hashfunction3.txt
...
```

- FIPSに記載されているハッシュ値と比較（A.1, A.3)

そしてSSLへ…

- 前回および今回あげた技術を組み合わせてSSLというプロトコルが定義されている
  - http://wp.netscape.com/eng/security/SSL_2.html
  - http://wp.netscape.com/eng/ssl3/
  - RFC 2246, RFC 4346
- TCPとHTTPの間にSSLを挟む ― 今までのプロトコルはそのままに通信を暗号化することができる

SSL (Secure Sockets Layer)



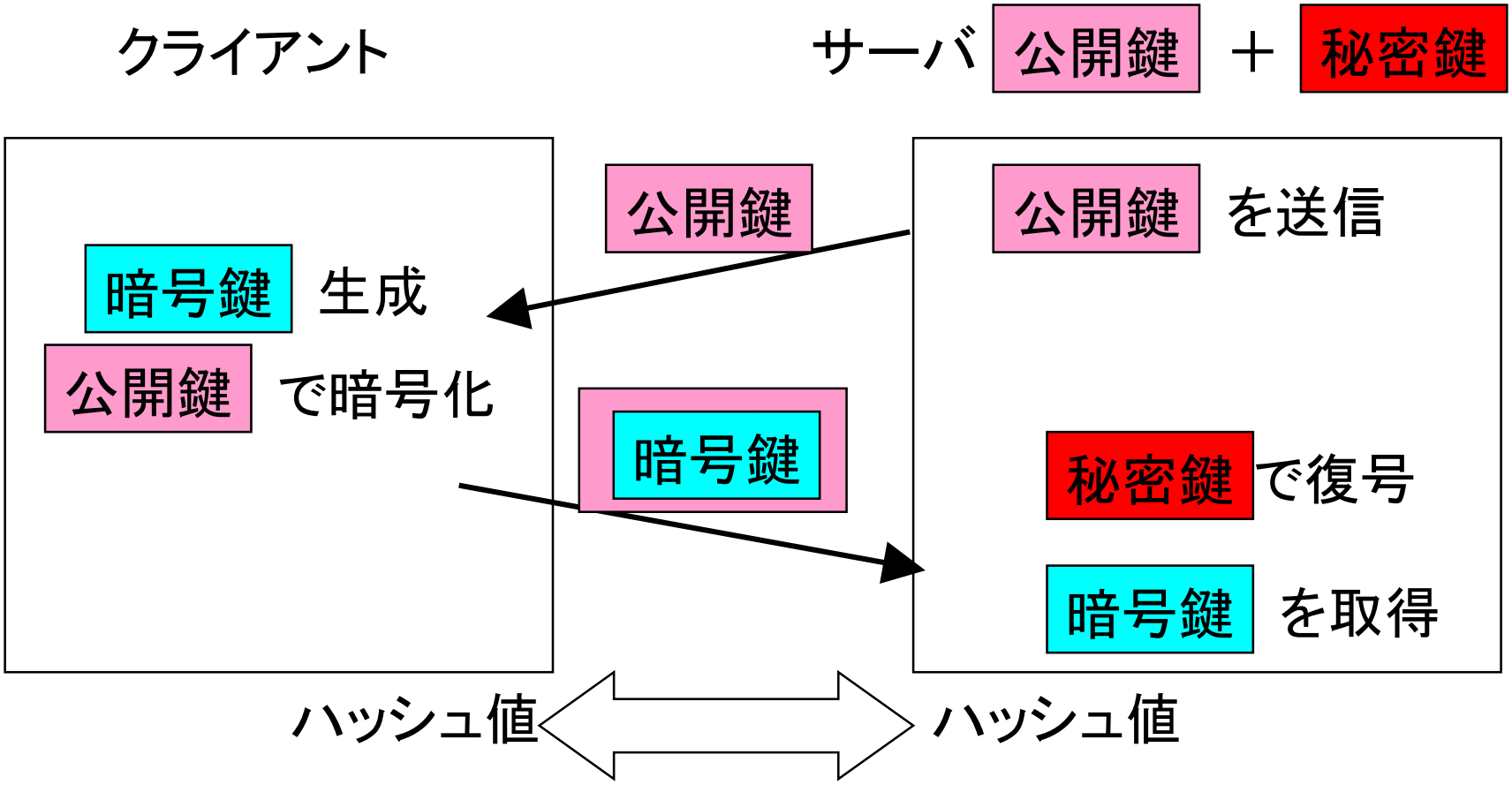
Handshake Protocol

SSL通信の最初に行なわれるやりとり

- 処理できる暗号方式の受け渡し
- 公開鍵の受け渡し（証明書の受け渡し）
- 暗号鍵の共有

  →暗号化には共通鍵暗号を使う
- ハッシュ関数でチェック

Handshake Protocol簡略版



以降は 暗号鍵 を使った共通鍵暗号で暗号化

応用例:HTTP → HTTPS

- HTTPでSSLを利用
- ポート番号 80 → 443
  - cf. RFC 2817
- RFC 2818

演習：SSL(HTTPS)を体験してみよう

- telnetの代わり

```
$ openssl s_client –connect
 secure.ecc.u-tokyo.ac.jp:443 –state
…
GET / HTTP/1.0
…
…
```

一行で書くこと