SSL/TLSの20年を振り返る ～ダウングレード攻撃からHeartbleedまでの脆弱性を中心に～

一般社団法人日本ネットワークインフォメーションセンター
木村泰司

内容

- SSL/TLSのおさらい
- SSL/TLSの歴史
- 脆弱性との戦いの歴史
- 対策の方向性

SSL/TLSのおさらい

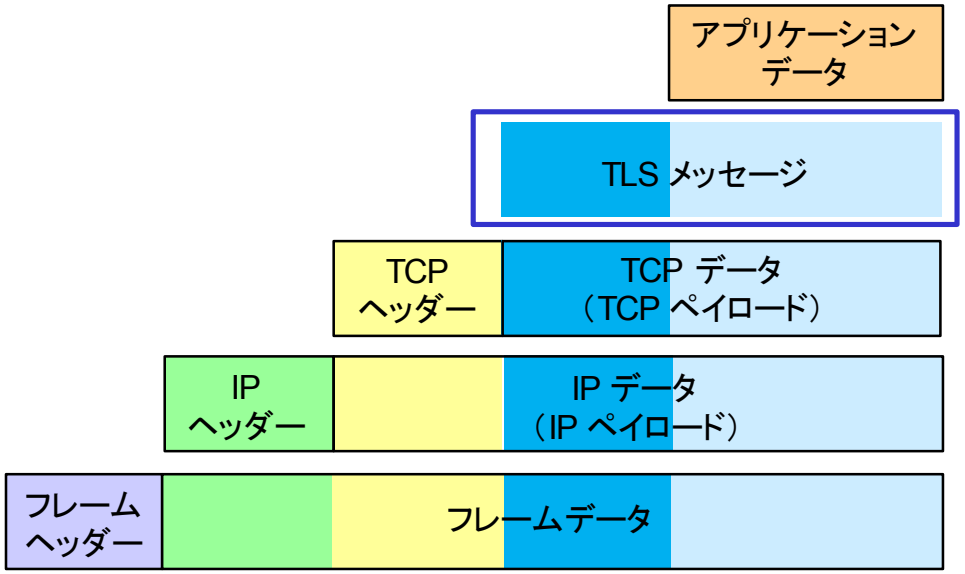
JPNIC 一般社団法人 日本ネットワークインフォメーションセンター

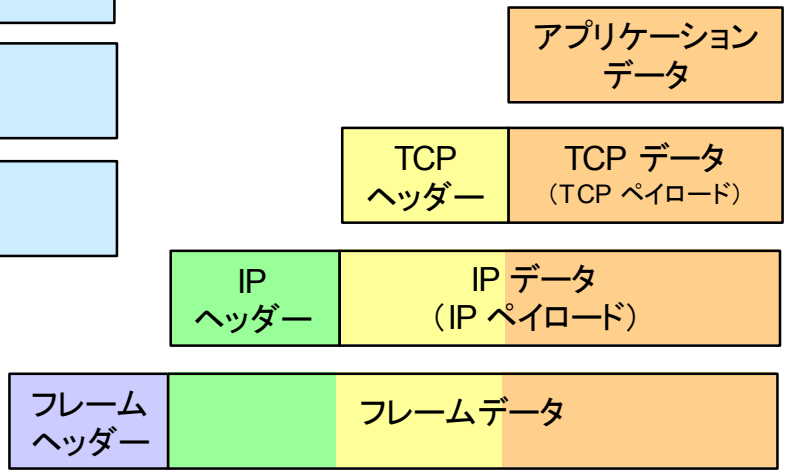
SSL/TLSプロトコル

SSL = Secure Sockets Layer
TLS = Transport Layer Security

- 暗号化
- 改ざん検出
- 通信相手の認証

- ソケットライブラリ

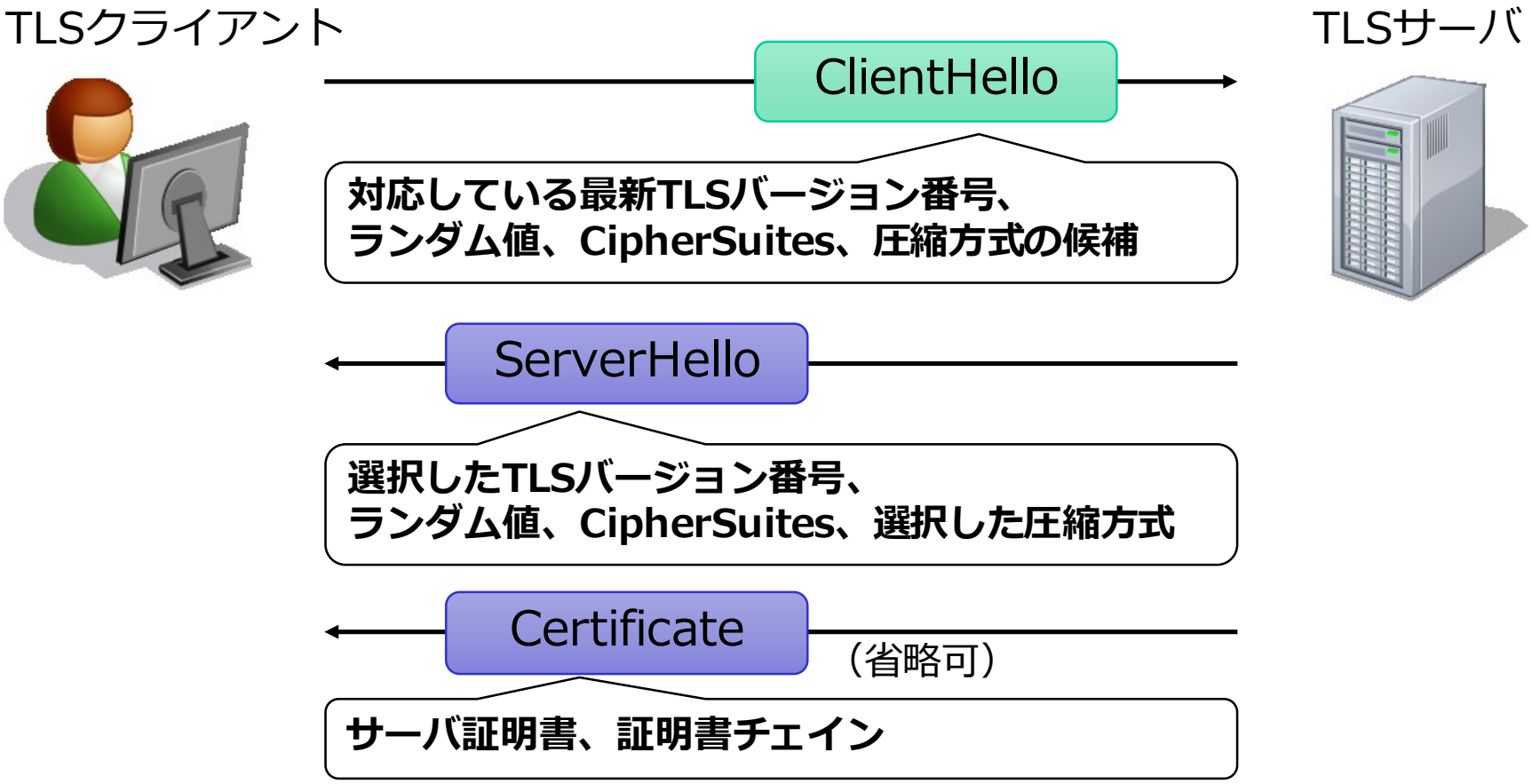




SSL/TLSのやること

- **暗号化と復号（confidentiality）**
  - 共通鍵暗号を使ったアプリケーションデータの暗号化と復号
  - 鍵共有
- **改ざん検出（Integrity）**
  - MAC=Message Authentication Code
- **通信相手の認証（Authentication）**
  - X.509証明書

TLSハンドシェイク(1/4)



- クライアントが扱えるものに応じてTLSバージョンを決定する。互いが扱えるCipherSuiteを情報交換しておく。

TLSハンドシェイク(2/4)

TLSクライアント

TLSサーバ

（鍵共有の方法によって）

ServerKeyExchange

一部の鍵共有（DHE, DH_anon）のためのデータ

（クライアント認証する場合）

CertificateRequest

サーバが扱える証明書の鍵の種類・署名アルゴリズム・認証局証明書

ServerHelloDone

- 鍵共有と認証のために使えるサーバ証明書をクライアントに渡す、もしくは鍵共有に必要なパラメーターを渡す。

TLSハンドシェイク(3/4)

TLSクライアント

(クライアント認証する場合)

ClientCertificate

クライアント証明書、証明書チェイン

ClientKeyExchange

サーバの公開鍵で暗号化されたプレマスターシークレット、またはDHパラメータ

(クライアント認証する場合)

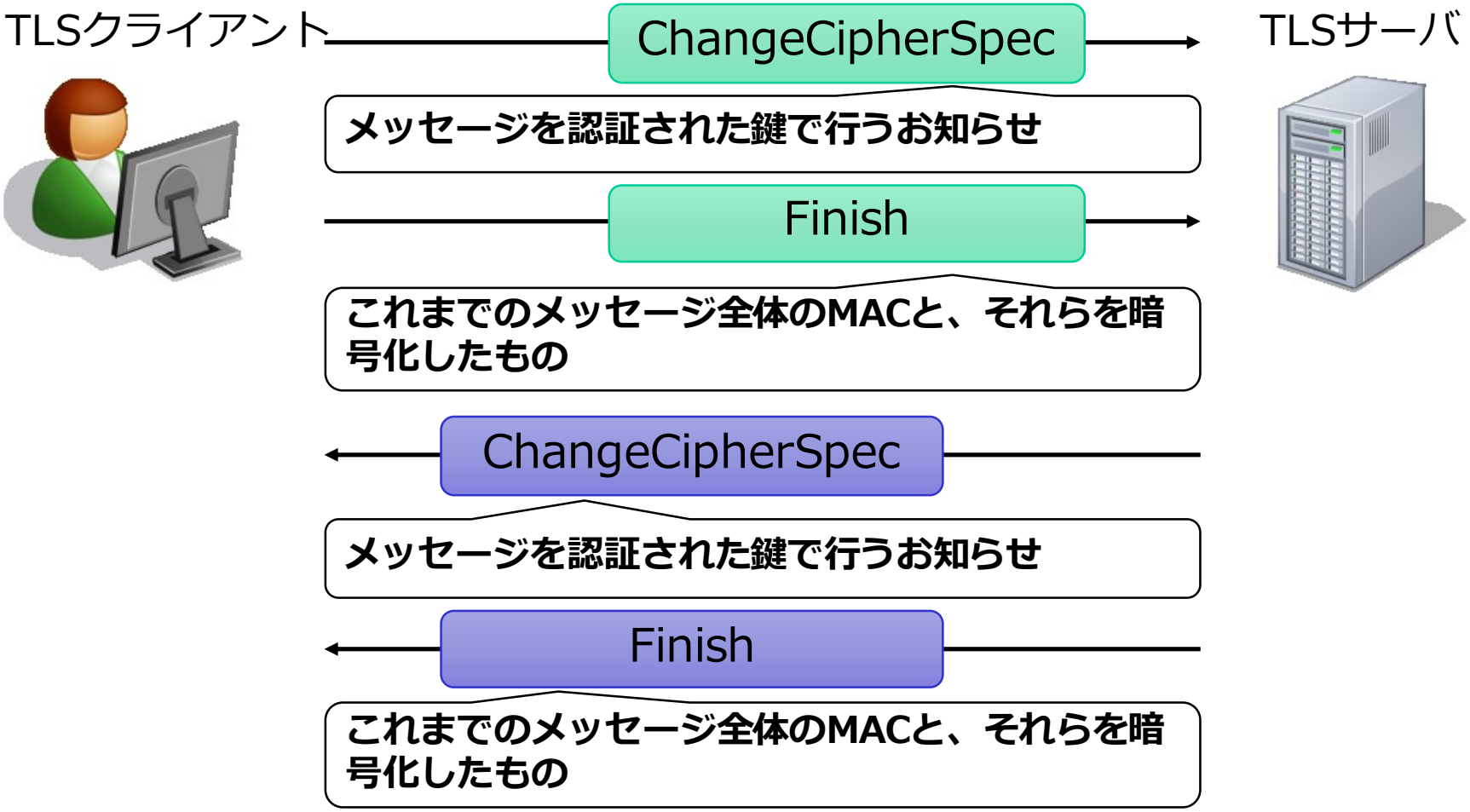
CertificateVerify

これまでのメッセージ全体への電子署名

TLSサーバ

- サーバはプレマスターシークレットから共通鍵を復号。
- クライアント認証する場合、サーバは電子署名を通じてクライアントが私有鍵を使えることを確認する。

TLSハンドシェイク(4/4)



- 相互に復号とMACの検証を行ってみて確認。完了。

TLSハンドシェイクにある意図

- クライアントとサーバの両方が扱える最もバージョンの新しいTLSを選ぶ
- クライアントとサーバの両方が扱える鍵交換・暗号・署名アルゴリズムを選ぶ
- クライアントとサーバが、証明書に入っている公開鍵に対応する私有鍵を使えることを確認する

しかし…

- クライアントとサーバの両方が扱える最もバージョンの新しいTLSを選ぶ。

  ⇒ 本当はもっと新しいバージョンを扱えるのに古いのを選んでしまっているのでは？

- クライアントとサーバの両方が扱える鍵交換・暗号・署名アルゴリズムを選ぶ。

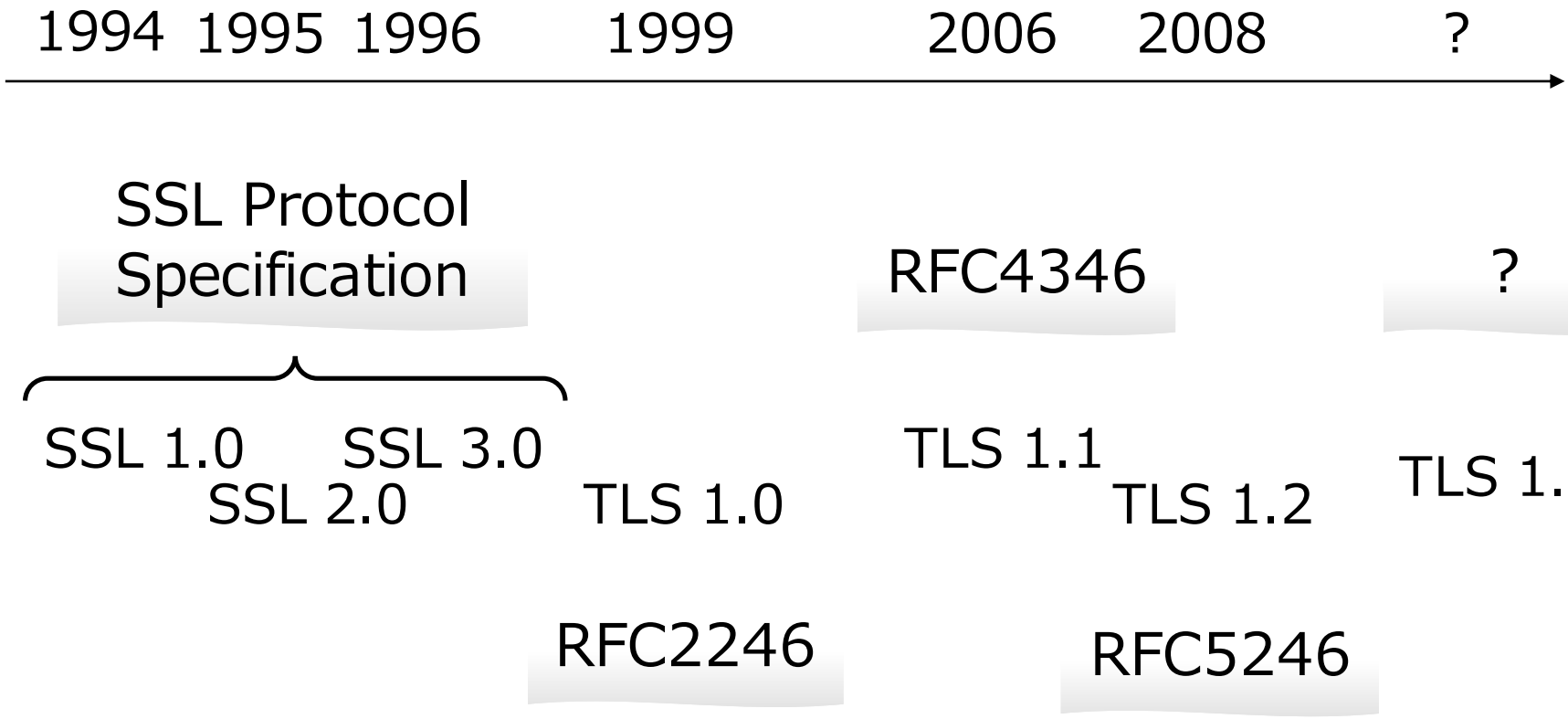
  ⇒ 弱い鍵や脆弱なアルゴリズムを選んでしまっていないか？

- クライアントとサーバが、証明書に入っている公開鍵に対応する私有鍵を使えることを確認する。

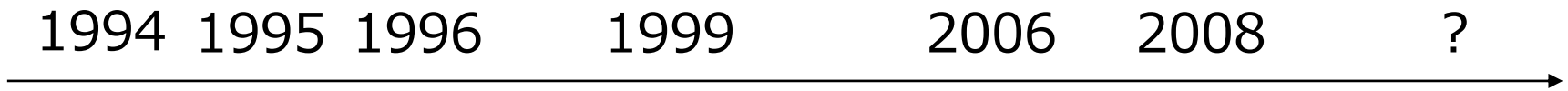
  ⇒ その証明書自体は本物か？

SSL/TLSの歴史

SSL/TLSの歴史



SSL/TLSの歴史



- Netscape社開発
- リリースなし

SSL 1.0

- Netscape Navigator 1.0に実装
- 中間CAは未サポート

SSL 2.0

- SSL2.0の修正のため再設計
- 再ネゴシエーション/圧縮対応
- 証明書チェイン対応

SSL 3.0

- 初のRFC策定
- SSL3.0とは詳細が異なるが似ている
- AES対応
- Camellia対応

TLS 1.0

- TLS 1.0の修正 IV/Padding改善
- ECC対応

TLS 1.1

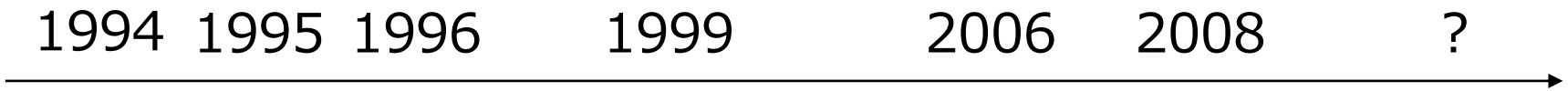
- SHA-256対応
- GCM,CCM対応

TLS 1.2

- 圧縮の非サポート
- PFSでないものの廃止?

TLS 1.3

SSL/TLSの歴史



- Netscape社開発
- リリースなし

SSL 1.0

- Netscape Navigator 1.0に実装
- 中間CAは未サポート

SSL 2.0

- SSL2.0の修正のため再設計
- 再ネゴシエーション/圧縮対応
- 証明書チェイン対応

SSL 3.0

- 初のRFC策定
- SSL3.0とは詳細が異なるが似ている
- AES対応
- Camellia対応

TLS 1.0

暗号アルゴリズムの危殆化が話題

- TLS 1.0の修正 IV/Padding改善
- ECC対応

TLS 1.1

- SHA-256対応
- GCM,CCM対応

TLS 1.2

- 圧縮の非サポート
- PFSでないものの廃止?

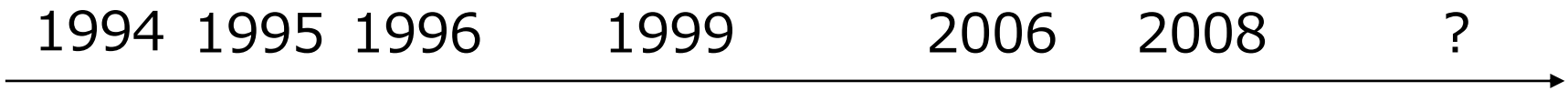
TLS 1.3

米国における暗号輸出規制の緩和(～2000年)

大規模な通信傍受の話題

脆弱性との戦いの歴史

脆弱性との戦いの歴史



- 改ざん検知なし
- リプレイアタックに弱い

SSL 1.0

SSL 2.0

- CipherSuite差し替え可
- パディング長がMAC対象でない

- ブロック暗号のCBCモードでの脆弱性

SSL 3.0

TLS 1.0

- 擬似乱数の関数でMD5,SHA-1のみ

TLS 1.1

TLS 1.2

(TLS 1.3)

最新

プロトコルの脆弱性の他に、アルゴリズムや実装の脆弱性をつく攻撃方法が指摘される。

SSL/TLS関連の事件年表(1/2)

1996 擬似乱数の予測攻撃

1998 Bleichenbacher
“Million Question Attack”

1996-2000 パディングオラクル攻撃

2006 選択平文攻撃

2008 Debian OpenSSL bug

2009 再ネゴシエーション攻撃

実装に関係する脆弱性は多く、パッチ当てが対策となるケースが多い。

SSL/TLS関連の事件年表(2/2)

2011　　2012　　2013　　2014

2011 トルコの認証局 誤発行

2011 コモドハッカー事件

2012 オランダDigiNotar事件

2012 ラッキーサーティーン

2013 CRIME/BREACH攻撃

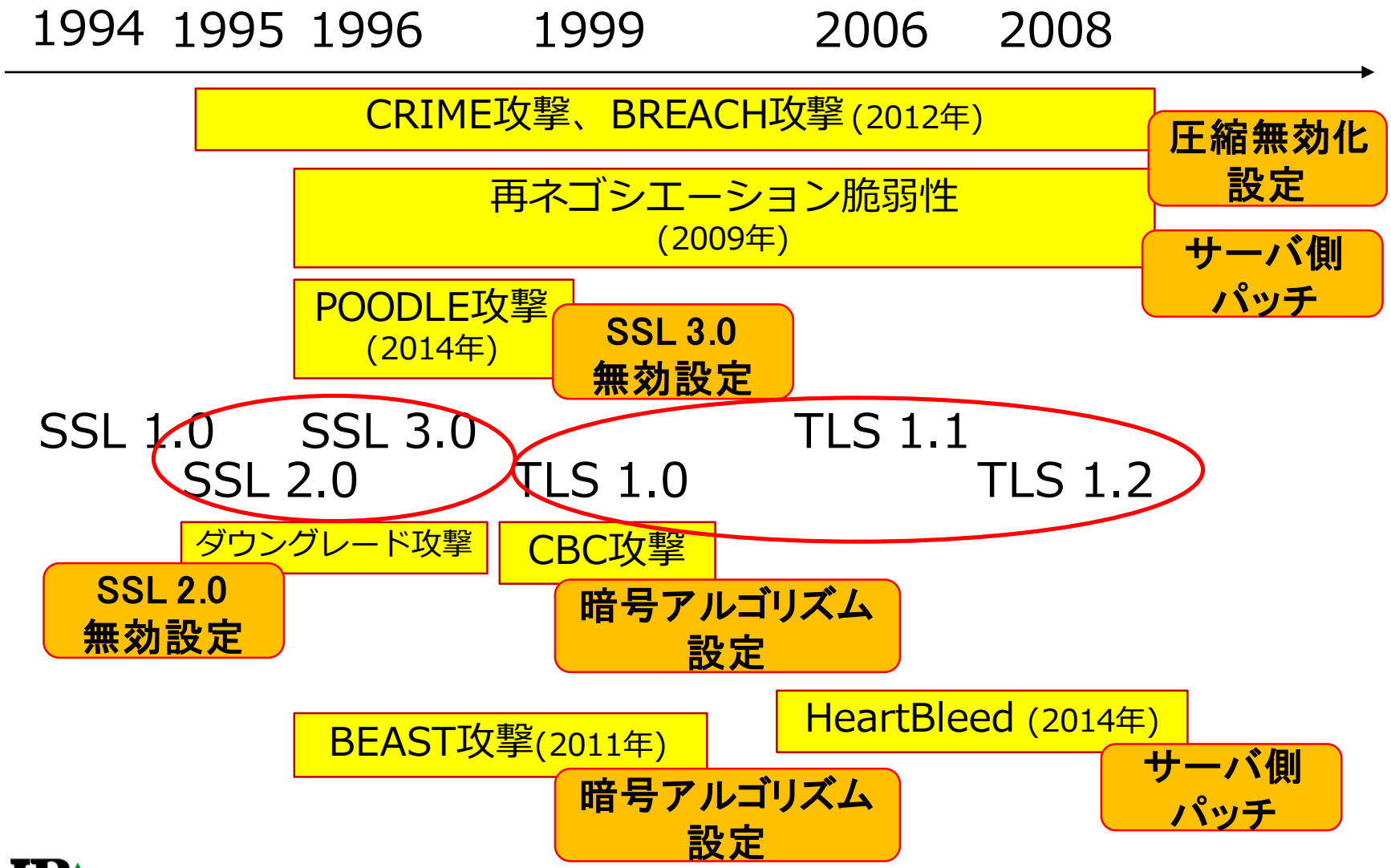
認証局関連の事件はプロトコルや実装の脆弱性ではない。（クライアント環境におけるパッチで対応）
2014年は脆弱性のねらい方が巧妙化

2014 Heartbleed

2014 POODLE攻撃

対策の方向性

脆弱性とバージョンと対応



まとめ

- **SSL/TLSハンドシェイクの意図に対して攻撃者によって狙われるポイントがある。**
- **SSL/TLSプロトコルにおける脆弱性の発見と共に改善とバージョンアップが行われてきた。**
- **実装や暗号アルゴリズムにも脆弱性対応が行われてきた。**

SSL/TLSは「最新のバージョンを使えば大丈夫」ということではなく、サーバの実装や設定、暗号アルゴリズムといった要素を確認していくことが重要だと考えられる。

おわり

資料

- **SSL/TLS**
  - RFC5246: “The Transport Layer Security (TLS) Protocol Version 1.2”
    http://tools.ietf.org/html/rfc5246
  - RFC4346: “The Transport Layer Security (TLS) Protocol Version 1.1”
    http://tools.ietf.org/html/rfc4346
  - RFC2246: “The TLS Protocol Version 1.0”
    http://tools.ietf.org/html/rfc2246
  - RFC6101: “The Secure Sockets Layer (SSL) Protocol Version 3.0”
    http://tools.ietf.org/html/rfc6101

資料

- **解説**
  - “20 Years of SSL/TLS Research, An Analysis of the Internet’s Security Foundation”, Christopher Meyer, Feb 2014, http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/MeyerChristopher/diss.pdf
  - “標準はどのように実装されているのか？ -- OpenSSLにおけるSSL/TLSの実装に関して –”, 富士ゼロックス株式会社, 稲田龍, Jun 2006 http://www.jnsa.org/seminar/2006/20060607/inada.pdf