JNSA PKI相互運用WG・電子署名WG共催セミナー
PKI Day 2015 サイバーセキュリティの要となるPKIを見直す

パネル：
SSL/TLSの実装が進むべき道を語ろう
補足資料

2015年4月10日(金) 15:00-15:30
於：ヒューリックカンファレンス秋葉原ROOM1

富士ゼロックス株式会社　漆嶌　賢二, CISSP

FUJI Xerox

①ライブラリ実装の今後

暗号やSSL/TLSのライブラリと言語/環境と心配事(1/3)

Windows	標準ライブラリ (CryptoAPI、CNG、Schannel)
OSX, iOS	標準ライブラリ(Secure Transport)
C言語 (含むRuby, Python)	OpenSSL、NSS、GnuTLS、 LibreSSL、BoringSSL、RSA BSAFE
Java JCE JSSE	Oracle Java、BouncyCastle、 IAIK、RSA BSAFE
Go言語	標準ライブラリ

参考：http://en.wikipedia.org/wiki/Comparison_of_TLS_implementations

FUJI Xerox

暗号やSSL/TLSのライブラリと言語/環境と心配事(2/3)

Windows	標準ライブラリ (CryptoAPI、CNG、Schannel)
OSX, iOS	標準ライブラリ(Secure Transport)
C言語 (含むRuby, Python)	OpenSSL、NSS、GnuTLS、 LibreSSL、BoringSSL、RSA BSAFE
Java JCE JSSE	Oracle Java、BouncyCastle、 IAIK、RSA BSAFE
Go言語	標準ライブラリ

あまり心配しても仕方ないでしょうか？

これほど乱立してしまい
どうなるんでしょうか

(日本のJava)開発者の年齢の上昇は不安材料

質、量的にセキュリティ
ライブラリはかなりプア

FUJI Xerox

暗号やSSL/TLSのライブラリと言語/環境と心配事(3/3)

Windows	標準ライブラリ (CryptoAPI、CNG、Schannel)
OSX, iOS	標準ライブラリ(Secure Transport)
C言語 (含むRuby, Python)	OpenSSL、NSS、GnuTLS、LibreSSL、BoringSSL、RSA BSAFE
Java JCE JSSE	Oracle Java、BouncyCastle、IAIK、RSA BSAFE
Go言語	標準ライブラリ

みなさんが気にしているのはこの辺りでしょうか

←こちらは個人的に興味があります

FUJI Xerox

2015年3月のOpenSSLの明るいニュース NCCへの外部コードレビュー委託

- LinuxコンソーシアムはCIIという資金提供プロジェクトにより、重要なオープンソースプロジェクトを支援
- CIIはOpenSSLを支援、NCCというセキュリティ会社にOpenSSLソースコードのレビューを外部委託
- OpenSSH, NTPなど含め3年で5億円
- メモリ管理やASN.1・X.509パーザーのファジング検査などを中心に調査
- 監査結果公開は2015年夏予定

ZDNet Japan

OpenSSLのコード、NCC Groupが監査へ --脆弱性の早期発見を目指す

Steven J. Vaughan-Nichols (Special to ZDNet.com)

インターネットのセキュリティに不可欠のSSL（Secure Sockets Layer）とTLS（Transport Layer Security）を提供する「OpenSSL」は、間違いなくインターネットにおける最重要技術の一つだが、「Heartbleed」や「FREAK」など深刻な脆弱性の発見が後を絶たない。そうした中、OpenSSLの潜在的なセキュリティホールを事前に洗い出して悪用に歯止めをかけるべく、技術力に定評のあるセキュリティ企業NCC GroupがOpenSSLのコードを監査することになった。

OpenSSLのコードは以前から監査の必要性が叫ばれていたが、「火中の栗を拾う」役目に自ら手を挙げる者が現れずに時間だけが経過していた。今回、ついにこの役目を担うことになったNCC Groupは、Core Infrastructure Initiative（CII）からの資金提供を受けて監査を行う。CIIはLinux Foundationが開始した資金提供プ

出典：http://japan.zdnet.com/article/35061477/

参考１：http://www.zdnet.com/article/ncc-group-to-audit-openssl-for-security-holes/
参考２：https://threatpost.com/openssl-security-audit-ready-to-start/111538

FUJI Xerox

LibreSSL(forked on OpenSSL 1.0.1g) 2014年7月～

- OpenBSDのBob Beck氏の講演YouTubeより

- Beck氏の語るOpenSSLから分岐した理由
  - コードが汚く誰もOpenSSLで開発したがらない
  - OpenSSLはFIPS対応に莫大な予算をかけ肝心な所は???
  - OpenSSLは新しい機能追加に重き、バグ修正は二の次で放置
  - Beck氏が出したセキュリティパッチも4ヶ月放置
  - メモリ管理がデバッグ含みで酷くそれがHeartBleedの引き金に
  - 独自の擬似乱数生成のエントロピーに問題がある
- LibreSSLの特徴
  - 2014年7月初リリース後、約1ヶ月おき更新
  - 公開された関数はOpenSSLと完全API互換(移行が楽)
  - 安全で保守性が高い
  - 弱い暗号の排除(SSLv3無効化)
- OpenSSL→LibreSSLのポートでやった事
  - コーディングスタイルはOpenBSD準拠できれいなコード
  - 必要とされる環境のみサポートし余計なコード、ifdef文を排除
  - メモリ管理をシステムコールに変更し安全、メモリ再利用不可
  - 擬似乱数のシードはシステム関数を使うよう変更し安全に

参考：http://www.youtube.com/watch?v=GnBbhXBDmwU (LibreSSL with Bob Beck)

BoringSSL（forked on OpenSSL 1.0.2beta）2014年7月～

- OpenSSLから分岐した理由、経緯、状況
  - これまで、Chrome(for Android)でOpenSSLを使用するために数多くのセキュリティ対応を含むパッチを作成してきた。
  - GoogleのパッチはOpenSSLに反映されたが、多くは反映されていない。
  - 維持も大変なのでOpenSSL 1.0.2betaから分岐した
  - 今後もGoogleはLinuxコンソーシアムCIIプロジェクトへの資金を含む支援をするし、LibreSSLへのパッチ提供もやっていく。
  - Googleが試験的に導入したい機能の組み込み(False Start,Channel ID)

- 特徴
  - Chrome for Androidの暗号ライブラリはOpenSSL→BoringSSLへ
  - 他のChromeはMozilla NSSに独自パッチをあてたものを使う
  - CHACHA20_POLY1305_SHA256の暗号スイートがある
  - 前はChaCha優先だったけど、4/8時点でGCM優先？
  - あまりソースコードが整理された感はない、メモリ管理もそのまま？
  - ツールが少ない(bssl {speed,client} のみ)

参考１：http://www.imperialviolet.org/2014/06/20/boringssl.html
参考２：http://d.hatena.ne.jp/jovi0608/20140729/1406624449

OpenSSL/C言語 雑感

- OpenSSLが将来何かに置き換わるとは思えない。Libreで置き換えることもないのでは？
- 個人的な希望としてはOpenSSLを整理しLibreとマージしてほしい
- GoogleはLibreに協力しているので、BoringがLibreのフォークに変更されることはあるかも？
- OpenSSLのHeartBleedやCCSInjectionなどの脆弱性については、(あまり使われない)新しく追加され、誰もレビューできず放置された機能が問題の原因となったように思う。余計な機能、拡張はデフォルト「オフ」にすれば防げた問題もあるのでは？

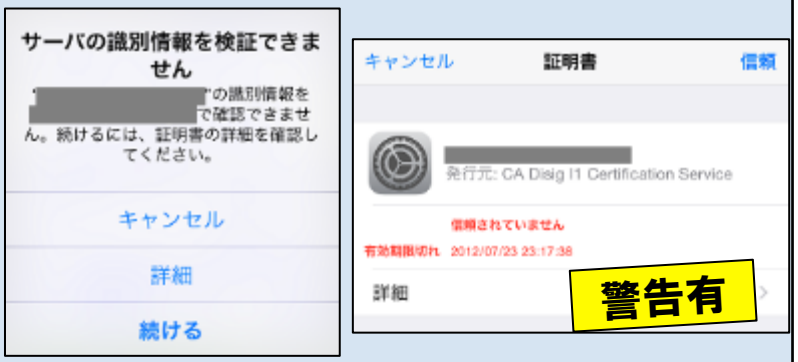
参考：http://www.youtube.com/watch?v=GnBbhXBDmwU (LibreSSL with Bob Beck)

Go言語の標準セキュリティライブラリ

- Go言語は簡単に並列実行可能で、バッファオーバーランやメモリ管理の心配もなく、簡単に高速で安全なプログラムが書ける
- ただ、PKI/SSL/暗号プログラミングに関する、いろいろな標準ライブラリが不足しており、とても問題
- Go言語のパッケージング/モジュールのアーキテクチャが良くなく、Javaのような良いサードパーティのライブラリが出にくい
- 例えば、Go言語で証明書の識別名からEV証明書かどうかをチェックしようとしたが、C、O、OUなどメジャーな属性しか対応せず他は捨ててしまうので、結局自分でバイナリをASN.1パーズする羽目になってしまいました

②ブラウザ実装の今後

モバイルブラウザ最大の懸念：の証明書検証(失効検証してない)

	CRLとOCSPの両方 利用可能なサイトの失効	期限切れ
Mobile Safari	エラーや 警告は 一切なし	
Chrome for Mobile	エラーや 警告は 一切なし	警告有 
Opera Mobile	エラーや 警告は 一切なし	警告有 

サーバーの秘密鍵が漏洩/盗難されたら他のどんな技術でも防げない
失効検証はPCと同様にきちんとやるべき

FUJI Xerox

③暗号移行の実装

暗号移行雑感

- 暗号移行は基本的に、ソフトウェアアップデートと、サーバー側の設定で対応するしかない。
  - クライアントで設定させるのはコストが高い
  - 一般ユーザが汎用ソフトでの暗号移行、トラストアンカ維持は、アップデートしかない

- セキュリティ要件の高いもの、例外要件に関しては、午前中のセッションで米丸先生も言及された、RFC 5698 DSSCが普及すると良い

- 本来は、「～はレガシーだから例外的に弱い暗号スイートを許す」としなければいけない。今は、弱い方に揃えるしかなく、RC4-MD5などが残っている。

RFC 5698 DSSC

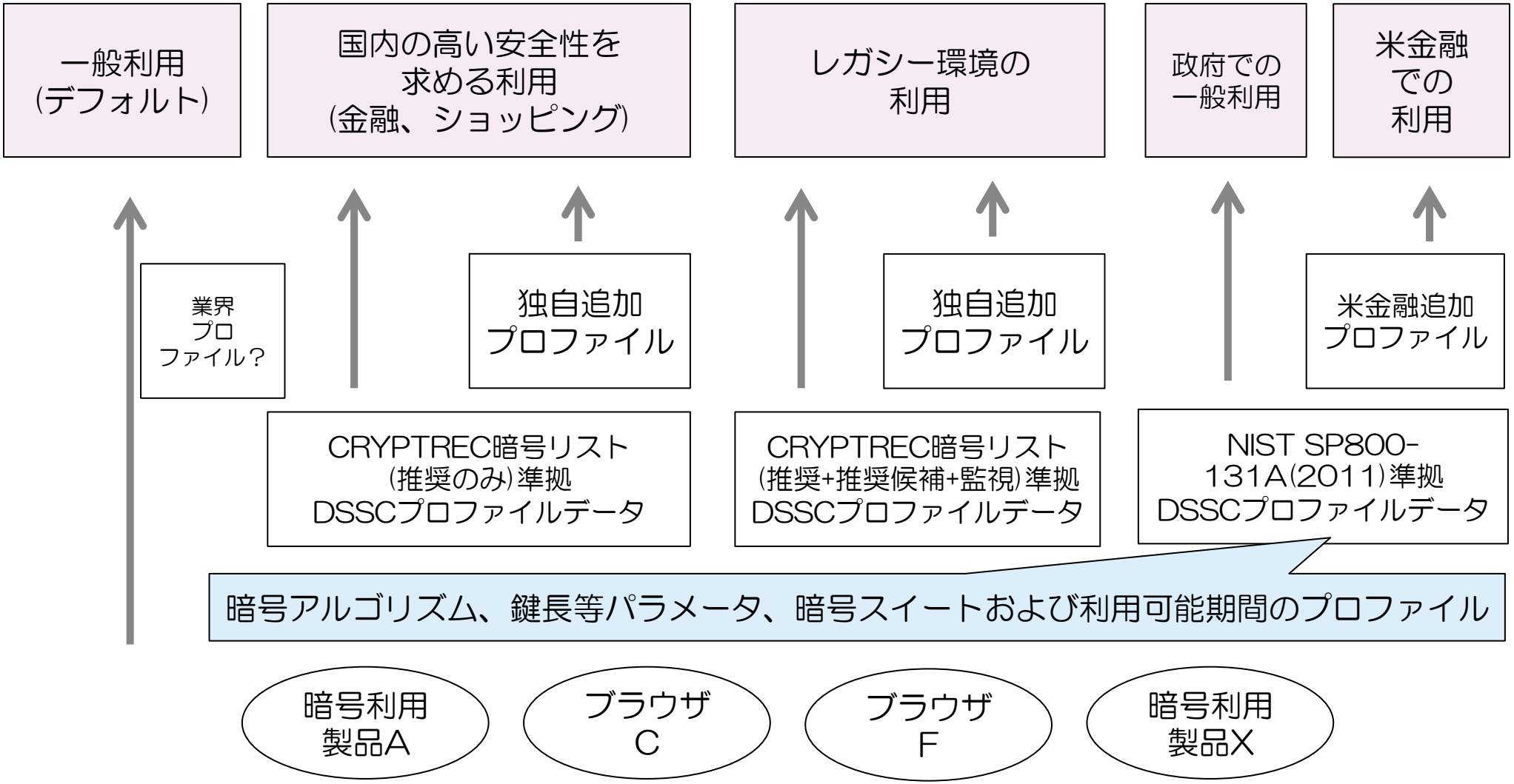
- RFC 5698 Data Structure for the Security Suitability of Cryptographic　Algorithms (DSSC)

- 暗号アルゴリズム利用に関するプロファイルを規定できる

- XMLやASN.1で以下の暗号利用ポリシーを表現できる
  - 使用可能なアルゴリズム
  - 使用可能な鍵長
  - そのアルゴリズムを使用可能な期間

- ただ、個人的には不満もある(後述)

DSSC(改)の活用(案)



※現行のDSSCには、プロファイル多段継承(or 追加/上書き)機能、暗号スイート対応がないので、DSSC標準への機能追加を期待したい。

FUJI Xerox

④非PKIによる補完関係

非PKIによる補完関係

- 残念ながらPKIや認証局単体では信用してもらえない世界になってしまった。(ただ、他なら信用できるのか？という)

- 以下の技術で包括的/総合的にサイトの正しさ、好ましさを見る必要がある
  - SSL/TLS
  - DANE/DNSSEC
  - Certificate Pinning
  - Certificate Transparency
  - WOTのようなユーザによるサイト評価集計

- Certificate Transparencyについては、個人的にいろいろ調査しており、PKI相互運用WGなどで、是非ディスカッションさせてほしい。その結果を別の場で発表できるとよい。

⑤その他(1)IoT時代のSSL実装

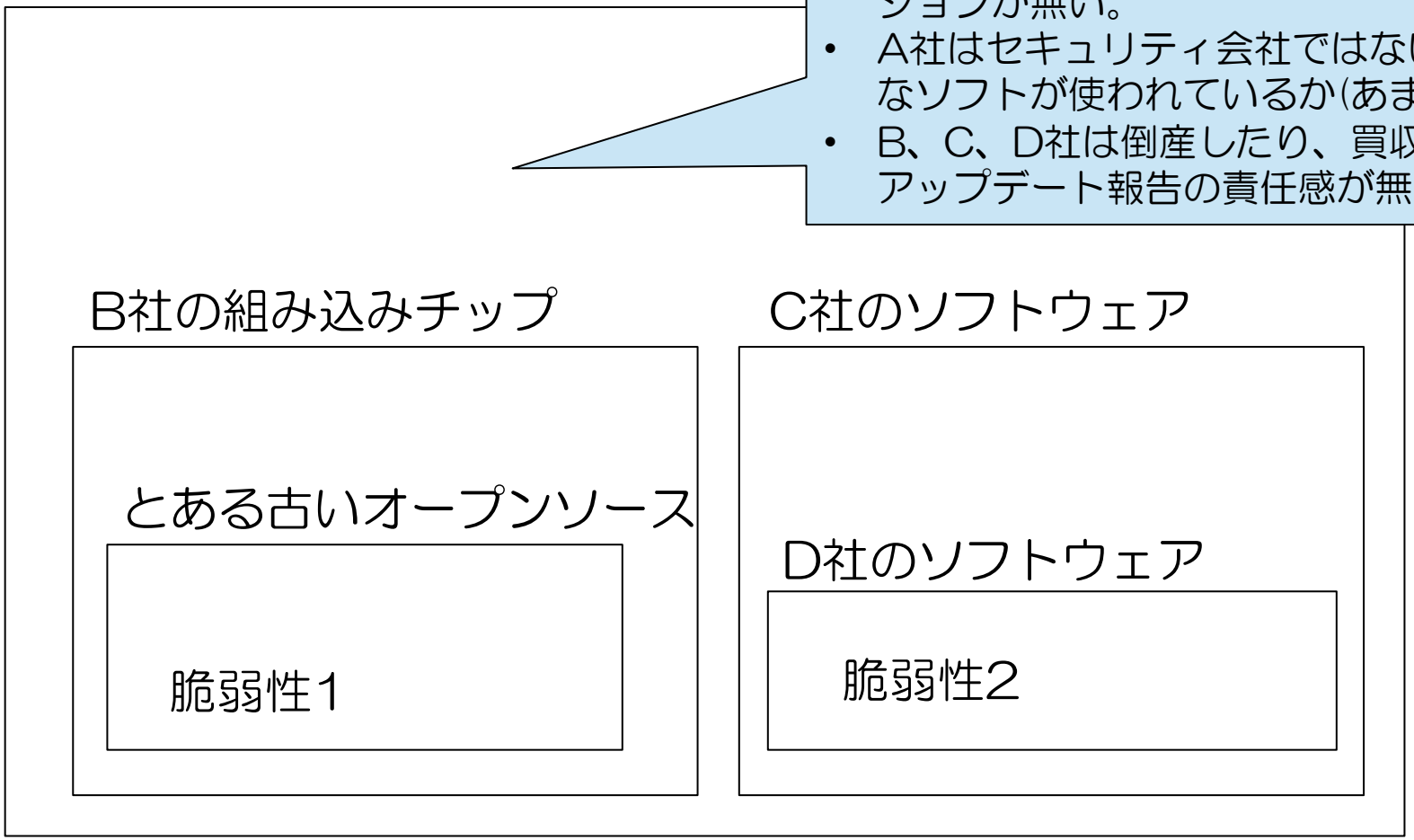
FUJI Xerox

IoT時代のSSL/TLSの実装 雑感

- ソフト、ファームのネットアップデート機能は必須
  - 暗号移行を特別視する必要ない
  - (アップデート機能があれば)IoTでは長くもつ暗号とかは気にしなくていいのでは？

IoT時代の安いIoTのSSL/TLS実装を含むライブラリの問題(ブルースシュナイアーの指摘)

- IoT製品は売れた時、組込まれた時にしかお金が入らないので、誰も脆弱性対策のモチベーションが無い。
- A社はセキュリティ会社ではないし、中にどんなソフトが使われているか(あまり)知らない。
- B、C、D社は倒産したり、買収されたりして、アップデート報告の責任感が無くなる。



安いIoTでは脆弱性対応アップデートを製品ベンダーに期待できないので
消費者団体/ユーザグループなどで外部調査し情報共有するようなスキームがあるとよい？

FUJI Xerox