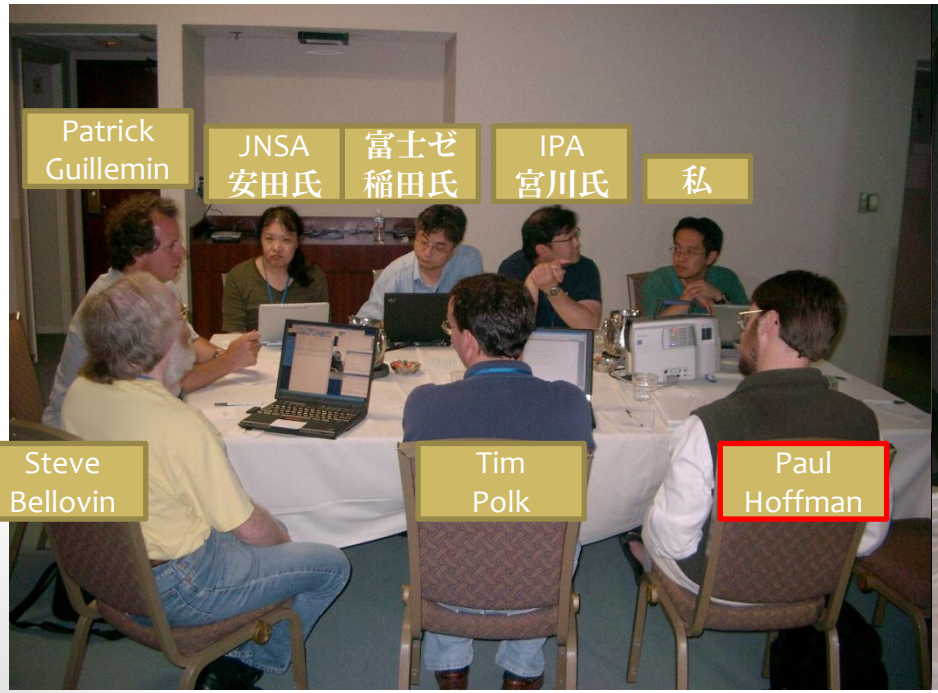
DANE update

セコムIS研究所
島岡　政基

自己紹介

- 2年目まではインターネット関連サービスシステム構築のSE
- 2000年以降はGPKI、AsiaPKI、UPKIと10年以上PKIどっぷり
  - 最近はPKI以外にもShibboleth, OpenIDなど
  - 最近は段々技術からポリシなどメタなレイヤへ。。。
- 2006年頃までIETF/PKIX WG中心に活動



dane-protocol(通称TLSA)

- RFC 6698: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA

- **概要**
  - TLS認証にCAの代わりにDNSSECを活用する技術
  - ドメイン名に証明書(またはそのハッシュ)を関連づけるリソースレコードとしてTLSAレコードを定義。

- History
  - 2010年12月に-00公開。DANE WGの前身、KIDNS ML時代。
  - 2012年8月、-23からRFC 6698として公開。1年半は順調な部類。

TLSAレコード

```
_443._tcp.www.example.com. IN TLSA (
        0 0 1 d2abde240d7cd3ee6b4b28c54df034b9
              7983a1d16e8a410e4561cb106618e971 )
```

Matching Type	0: Exact match 1: SHA-256 2: SHA-512

Selector	0: Full certificate 1: subjectPublicKeyInfo

Cert. Usage	0: CA constraint (**中間CA証明書でもok**) 1: Service certificate constraint (**CAから発行されたサーバ証明書**) 2: Trust anchor assertion (**ルートCA証明書**) 3: Domain-issued certificate (**自己署名証明書ok**)

1: 認証パスの検証必須
3: 認証パスの検証不要

Running code

- RFC化におけるrunning code
  - Datatracker→IESG Writeups
  - https://datatracker.ietf.org/doc/draft-ietf-dane-protocol/writeup/
- SWEDE
  - TLSAレコードの作成・検証ツール
  - Pythonベース
  - https://github.com/pieterlexis/swede
  - 実装してない機能も沢山あります。
- DANE for NSS
  - NSS(Mozilla)に対するパッチ
  - https://mattmccutchen.net/cryptid/#nss-dane
  - SSL_AuthCertificateでTLSA RRsetを取得する
  - PoCなので実運用には無理あり

DANE WGの動向

- アプリケーションプロトコルへの実装
  - smtp-04, mua-00, smime-03, (xmpp-02)
  - TLSプロトコルの中でTLSAレコードを使うための手順を定義している(smtp, mua)
- リチャーターに関する議論
  - 今後出てくることが予想されるであろう各アプリケーションへの実装方式をどこまでDANE WGで扱うべきか?
  - 原則論(General purpose edition)を示すべきか?

- 詳しくはJPRS森下さんの報告記事にて！
  - 第84回IETF Meeting報告（1）～DANEの動向に関する話題～
  - http://jpinfo.jp/event/2012/0831IETF.html

中間者攻撃とオレオレ証明書問題

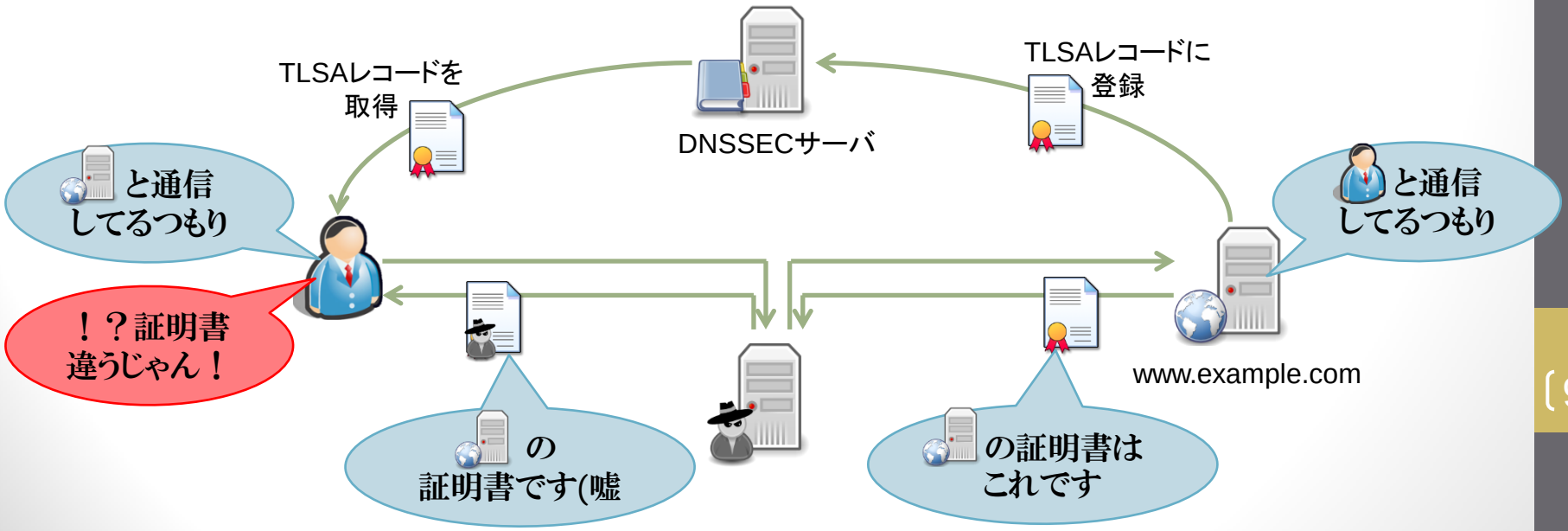
オレオレ証明書は中間者攻撃対策が不十分
∵TTP (Trusted Third-Party)不在

- パブリック証明書を入手して使う
  - お金かかる！
- 中間者攻撃のリスクを無視できる範囲で利用する
  - 判断難しい！
  - 責任取りたくない！
  - 説明難しい！
- 証明書を利用者に配付・インストールしてもらう
  - 面倒！
  - (相手によっては)頼みづらい！
  - 結局理由を説明しないといけない！

TLSAで何が変わる??

- 自己署名サーバ証明書が使いやすくなる
  - DNSSECで安全にサーバ証明書を配付できる
  - 中間者攻撃の脅威がない→つまり難しい説明不要！

- 認証局なしで安全な暗号化通信を実現できる



認証局が提供しているのは…

- サーバ証明書発行時の確認作業
  - Proof of Possession：公開鍵に紐づく私有鍵を所持していること
    - 申請者：私有鍵を用いて署名生成。公開鍵とともに認証局に送付。
    - 認証局：公開鍵で署名検証。
    - PKCS#10という証明書発行要求のデータフォーマットを用いる。
  - Domain validation：当該ホスト(管理者)からの申請であること
    - やり方は色々。
    - 例1：予め登録済のホスト管理者メールアドレスに連絡。
    - 例2: 予め登録済の権限者経由でのみ申請を受け付ける。
- サーバ証明書の失効
  - 漏洩した鍵の不正利用を防ぐ
    - なりすまし、盗聴、復号解読など

こうした一連の作業について、CPSによって透明性を確保している

TLSAの課題?

- 利用者のDNSSECサポートが大前提。
  - DNSSECのラストワンマイル問題
- (Web)サーバのDNSサーバがDNSSECを導入していること。
  - ラストツーマイル問題??

- 証明書失効ができない
  - 私有鍵漏洩したら盗聴される危険性
  - ただしなりすましリスクは低い?

- DNS管理者の負担
  - TLSAレコードの登録・更新
  - アプリケーションプロトコル増えれば責任も、、、

参考資料

- IETF/DANE WG
  - https://tools.ietf.org/wg/dane/
- RFC 6698
  - https://tools.ietf.org/html/rfc6698
- JPRS IETF Meeting報告
  - 第83回IETF Meeting報告 ～DNS関連の話題を中心に～
    - http://jpinfo.jp/event/2012/0425IETF.html
  - 第84回IETF Meeting報告（1）～DANEの動向に関する話題～
    - http://jpinfo.jp/event/2012/0831IETF.html
- TLSとDNSSEC
  - DNSSEC 2012 スプリングフォーラム資料
    - http://dnssec.jp/?page_id=913