インターネット上の信頼を確立する PKIの技術と運用 (応用編)

富士ゼロックス株式会社
稲田 龍
<Ryu.Inada@fujixerox.co.jp>

インターネット上の信頼の確立には？

- **確実な認証**
  - **どこでも、いつでも、ある程度、確実な認証**
  - **孤立した環境でも動けることが出来れば**better
- **安全な通信**
  - **経路上で盗聴が出来ない**
    - RFID**の様に無線環境にも適応できること**
  - **経路上で改竄されない**
    - **最低限、改竄の検出が出来ること**
- **情報の漏洩を最小限にしたい**

続き

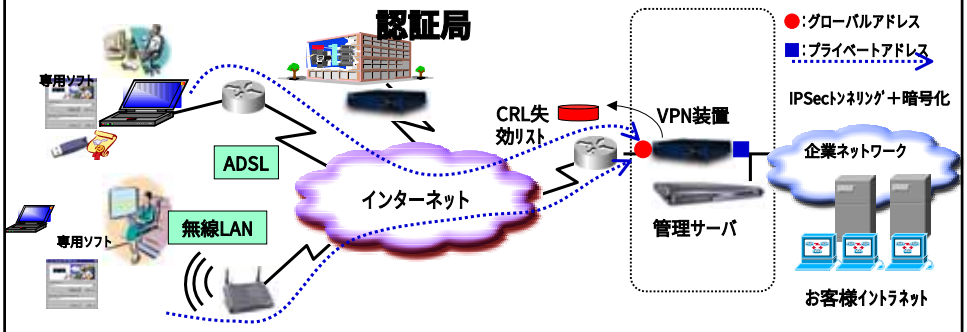
- 端末上でのリスク
- 端末間通信に対するリスク
- サーバ/端末間に置けるリスク
- サーバ上でのリスク

- パスワードからバイオメトリックス?
- 知識での認証からデバイスでの認証へ?
- クレデンシャルの内容の変異
  - 単なるパスワードからより強度のあるものへ

PKIの必要性

- インターネットで要求される便利で安全な認証
  - 安心、安全なインターネット環境 (PKI空間)のために、様々な用途の認証、様々なレベルの認証、広いドメインでの認証を実現したい
  - 広く採用するには、標準化された技術を採用したい
- 電子政府と政府認証基盤(GPKI)など動向
  - GtoBのための認証基盤　ー>GPKI
  - 3300の地方自治体のための認証基盤　-> LGPKI
  - GtoCのための認証基盤 -> 公的個人認証基盤
- Identrusのような国を超えたB2Bの認証基盤
  - サイバー世界では国境がない。
  - 世界に通用するセキュリティが必要
- GPKI、IdentrusのPKIの技術的要件
  - 否認防止が可能な署名

例えばInternet-VPNサービス



Anywhere, Anytime, Anyplace モバイルオフィスを実現するためには、確実な認証が必要

PKIはどう使われているのか?

- 現行ではHTTPSのホスト認証が主流
  - 伝送路の安全性の確保
  - ホストの正当性の保証

  判りやすく、利用者に負担をかけてない
  - 利用者に対して効果(伝送路の暗号化)がわかりやすい
  - HTTPSに対応したブラウザがあれば、利用者は何も意識せずに使える

PKIはどう使われているのか?

- 一方で、クライアント認証はほとんど行なわれていない
  - 金融系で一部使われている

  利用者の負担が大きい?
  - パスワードに比較して効能に差が見えにくい
  - 利用者の利便性がない?
  - 費用がかかる?

今後どう使われていくのか?

- インターネット＝ユビキタスネットワーク?
  - とはいえ、多くの状況ではインターネットを経由するであろう
  - 認証すべきものが増えてゆく?
    - サーバのみならずPDA/携帯電話/家電……
    - いわゆる機器認証が増えてゆく?
      - RFIDなどの情報の交換にもPKIはついて回る?
  - 「認証」に重きが行くのではないか?

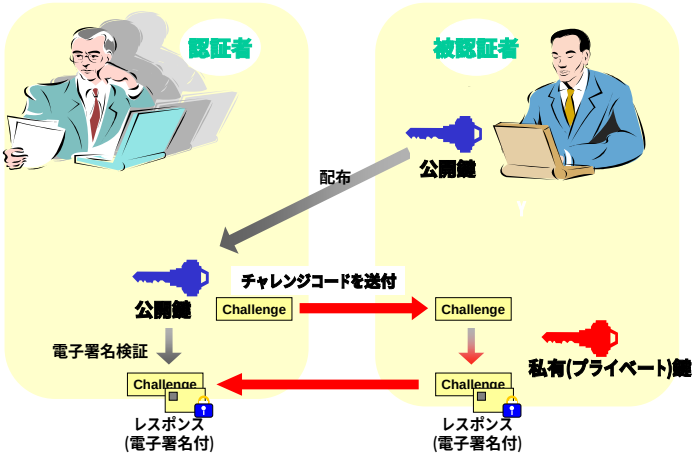
認証への応用

- **基本は**Challenge & Response

- PKIの利点
  - 認証サーバを必要としない
    - 孤立したネットワークでの利用が可能
    - 複数の拠点間での移動ノードに適している

- PKIの欠点
  - 演算が遅い
  - 厳密な失効確認をすると大変
    - 厳密な失効確認はどうするか？ → 後述

認証への応用の概念

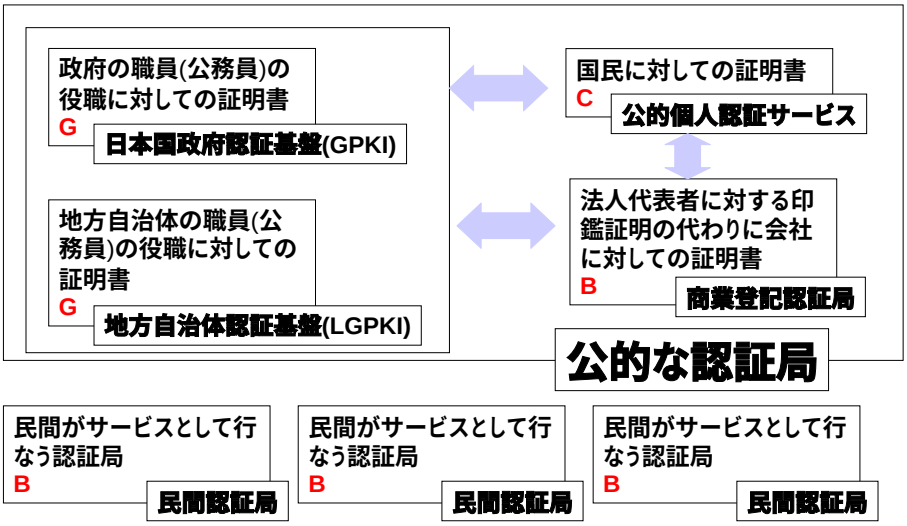


実社会での流れ

- 法的な環境の整備
  - いわゆる「電子署名法」→否認防止
  - e文書法
- 政府での利用
  - GPKI(政府)
  - LGPKI(地方自治体)
  - 公的個人認証サービス(国民)

- 公的機関において使われる認証システムにPKIが採用されている

日本における認証局



PKIが認証に使われるには？

- PKIでの認証は、ユーザ名･パスワードに比較して強固
- オフラインでの認証に利用できる可能性もある

- ただし……
  - 認証に利用する証明書の信頼性の検証が必要

電子署名アプリケーション

電子署名アプリケーション

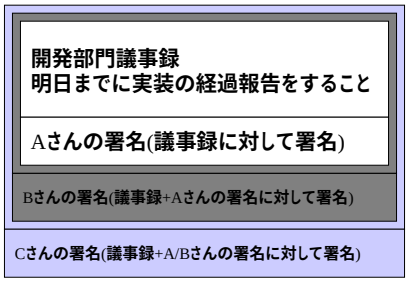
- ファイル/データ等に対して電子署名
  - 文書などのデータに署名する
  - コード署名といわれるプログラムへの署名
- 電子署名法の施行/電子政府での採用
- 専用アプリケーション
  - 電子申請など
- 汎用アプリケーション
  - Acrobat
  - Microsoft Office XP
  - DocuWorks

シリアル署名とパラレル署名

- 署名をどの部分に行うかの違い
- 用途により使い分けが必要
- Acrobat/Office XP/DocuWorksともにシリアル署名を実装

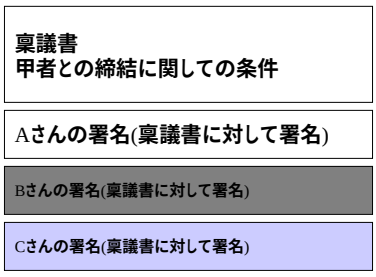
シリアル署名

- 署名を「追加」していくイメージ
- 長所
  - 署名の順番がわかる
- 短所
  - オリジナルの文書に対しての署名



パラレル署名

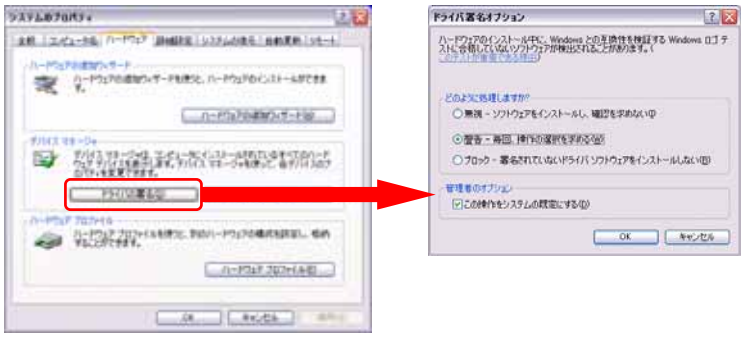
- 署名対象に対してのみ署名を行う
- 順番に関係なく署名検証可能



コード署名

- ダウンロードしたプログラムが正しいかどうかをどう確認するか?
  - 悪意のあるプログラム/ウィルスの排除
  - 正当なデバイスドライバであるかどうかの確認

Windowsのドライバの署名



マイクロソフトはWindowsのドライバに対して署名をすることにより、互換性の保障を行っている

Active-Xのコード署名

- IEの機能拡張を行うActive-Xモジュールに関してもコード署名を提供している

証明書の実効性/有効性

証明書の実効性/有効性

- 2001年の4月にいわゆる「電子署名法」が施行
  - 特定認証局が発行した電子証明書に実印と同様の権限を与えた
- 商業登記法の改正
  - 商業登記局が会社代表者に対して証明書を発行
    - 会社代表者に対しての印鑑証明に相当する
- 欧米では、バイオメトリックス情報を証明書に入れる動きもある
  - 身分証明書の代わりに使える証明書
  - 署名のイメージを入れる動きもある

QC(特定証明書)とは何か?

- 通常の電子証明書に対して、より高位の「保証」をつける事を目論んでいる証明書
  - 欧州における公的個人認証の必要性から、
    - 自然人（個人）を対象
    - 法的に認められるための証明書として通用すること

- セキュリティポリシーとそれを反映した証明書フォーマット（プロファイル）の制定が必要
- 欧州の標準化団体により提唱された標準が、IETF で採用され RFC 3039 として規定されたものが「**クオリファイド証明書**(Qualified Certificate)（**特定証明書**）」
- 現在はRFC 3739 として改訂版が出ている。

QC(特定証明書)とは何か?

- 特定証明書の特徴
  - X.509 v3 証明書プロファイルに準拠
  - 基本領域、拡張領域への記載内容にルールを設定
  - 特定証明書に特化した拡張領域を保持
  - 「人」を対象とした証明書
    - 必要となるポリシーを規定
- 記載内容に関するルールには、欧州電子署名指令案(EU-directive)の指示のもとETSIによる標準化検討
  - 実際には欧州における法律制度･社会制度にのっとり内容についてさらに詳細な規定を加えて
- 特定証明書を定義したRFC 3739
  - 利用される国や団体の幅広い要件に対応できるように汎用的な内容

標準化動向

- 欧州における電子署名の要件を満たすために検討が進められた
  - 「電子署名についての欧州指令(European Directive on Electronic Signature)」
    - Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures
- IETFでインターネットでの適用の必要性を認め、Standard Trackとして標準化が進行中
- 911以降、米国内においてバイオメトリック認証の必要性が向上
  - 米国政府内の標準的な認証用ICカードとしてバイオメトリックス情報の利用が検討されている

日本では?

- 公的個人認証サービスなど特定証明書を適用できる/すべきものはある
  - 現状の公的個人認証サービスにおいては、個人を特定する情報としていわゆる4大基本情報(氏名、生年月日、性別、住所) を入れているが、特にバイオメトリックス情報は入れてはいない
    - 個人情報に対する扱い
  - IPA(情報処理推進機構)が、特定証明書を適用すべきではないかと画策している模様
    - 経済産業省の思惑あり?
  - 状況によっては来年度の電子署名法改正に組み込まれるかも(参考情報としては既にインプット済みの模様)
    - 2006年度変更を見据え、2005年度中に見直し
    - タイムスタンプの概念は入るとの情報あり

QCの特徴 – 名前について

- 名前の一意性の保証(RFC 3739 2.4)

- 主体者ディレクトリ属性(subjectDirectoryAttributes)をもつ場合がある
  - クリティカルフラグを立ててはならない
  - dateOfBirth/placeOfBirth/gender/countryOfCitizenship/countryOfResidenceの各属性を解釈できること

QCの特徴 － 証明書ポリシー (certificatePolicies)

- **必須**
- **最小限1つのポリシーIDを持つこと**
- **クリティカルでも良い**
- **証明書発行目的がポリシーにより明確になっていること**
- **認証パス検証に必要なすべてのポリシー情報を含むこと**

QCの特徴 － バイオメトリック情報(biometricInformation)

- **オプショナル**
- **バイオメトリックテンプレートのハッシュとして格納**
  - **証明書内に含まず、URIで参照しても良い**
    - **ナイーブな情報が含まれることに注意**
    - **URIはhttp/httpsでなければいけない**
- **人間の検証にふさわしい情報の種類に限定することを推奨**
- **クリティカルにしてはならない**
- Picture/handewritten-signature**が予め登録済み**

製品への対応

- RSA Security**社**
  - RSA Keon Certificate Authority 6.5
    - http://www.rsasecurity.com/japan/news/data/200302181.html
    - http://www.rsasecurity.com/japan/products/keon/keon_certficate_authority.html
    - **ニュースリイリース内でクオリファイド証明書**(QC/**特定証明書のこと**)**に関してのサポートを記述している**

証明書の検証

証明書検証とは？

- Challenge / Responseの確認を行う際に、利用している証明書は「正しい」ものなのか？
  - 証明書の真贋はどうするか？

証明書の検証

- 証明書の信頼性をどう考えるか?
  - 証明書の偽造のチェックはどうするか?
  - 厳密な検証を行うには……
    - パス構築
    - パス検証

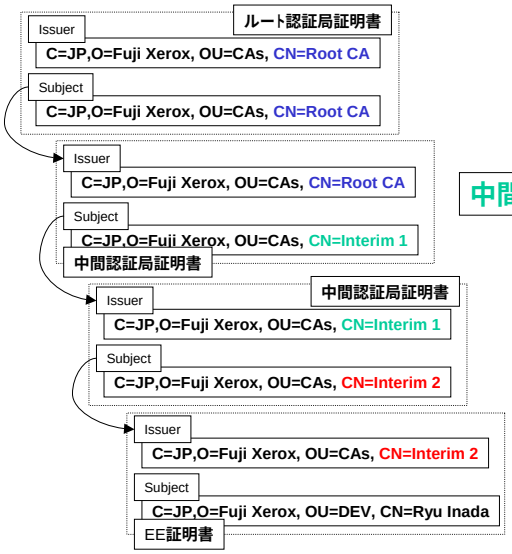
証明書の検証

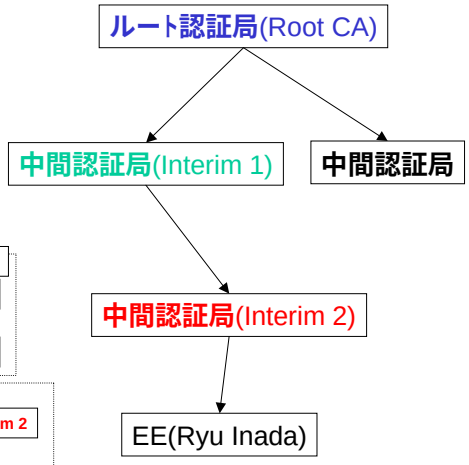
- 発行された証明書が有効なものか?
  - PKIの特徴との対比
    - 証明書の失効確認ができれば、認証サーバなどに接続せず、オフラインでの検証ができる
      - CRLが有利?
  - 証明書の検証は大変
  - オレオレ証明書
    - 自己署名証明書をどう扱うか?
      - Trust Anchor (Trust Point) をどう扱うか?
        - » 予め別経路で配る?
        - » CTL (Certificate Trust List)を利用する?

パス構築

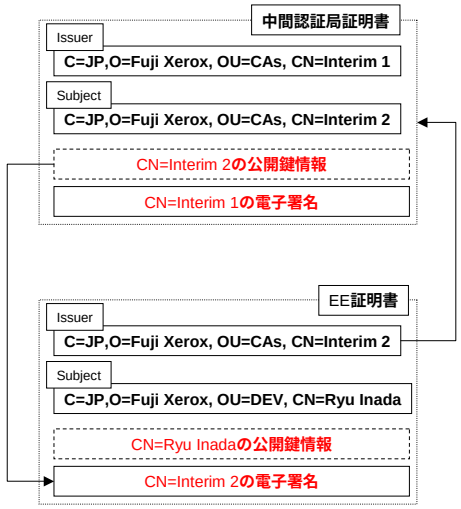
- 基本は、Issuer/Subjectを元にルート認証局からの木構造を再構成する
  - 詳細は……
    - RFC 4158: Internet X.509 Public Key Infrastructure: Certification Path Building

パス構築の概念



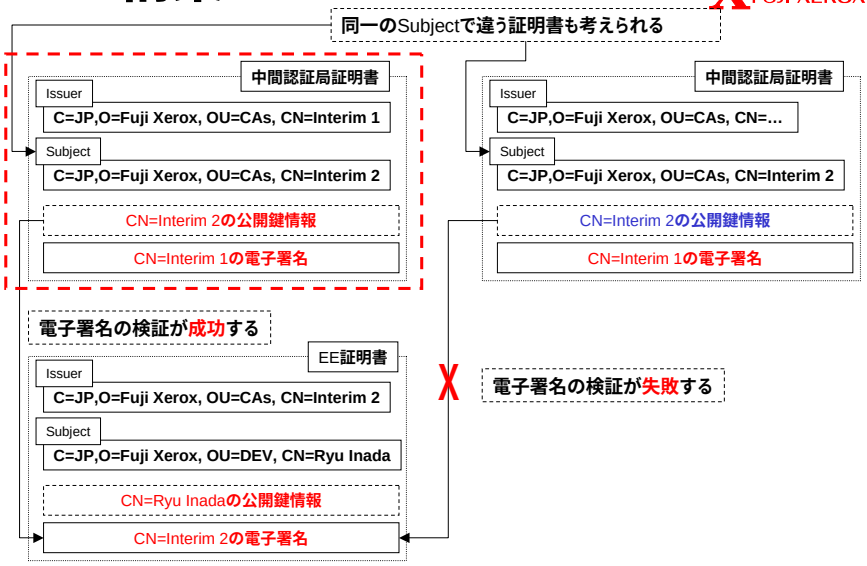


パス構築における署名の確認



1. Issuerから直上の認証局のSubjectを判断
2. 認証局の証明書より公開鍵情報を入手
3. 電子署名を確認

パス構築



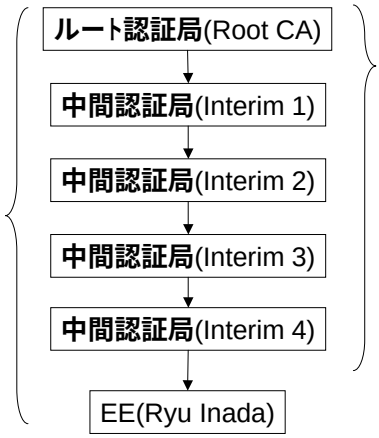
basicConstraintのチェック

- 証明書にはbasicConstraintというフィールドがある
  - isCA
    - 証明書が認証局のものであるかどうかを示すフィールド
  - pathLengthConstraint
    - 木構造が何段まで許されるかを規定

basicConstraintのチェック

THE DOCUMENT COMPANY FUJI XEROX

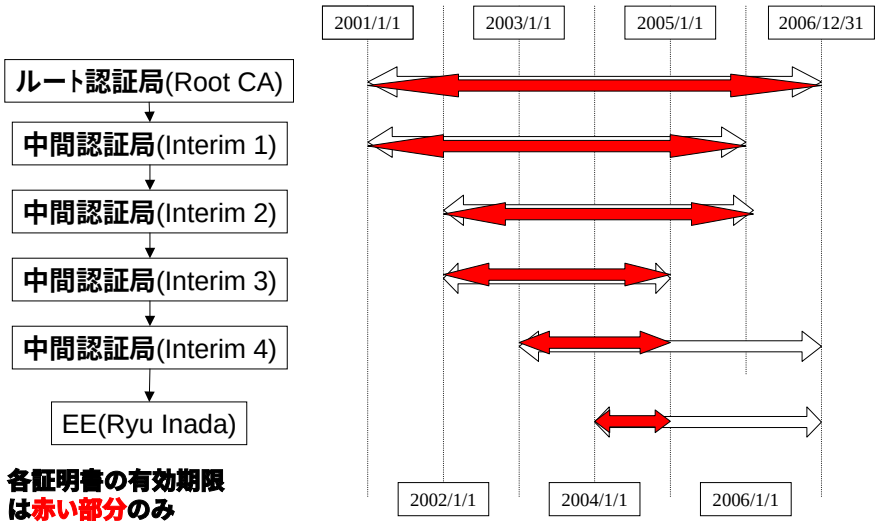
パスの長さ（認証局を経由する数）がpathLengthConstraint以下でないといけない



認証局の証明書はbasicConstraintのisCAがTRUEでなければならない

有効期限のチェック

THE DOCUMENT COMPANY FUJI XEROX



各証明書の有効期限は赤い部分のみ

keyUsageのチェック

- 証明書の使用目的を定義している
  - keyUsage
  - extendedKeyUsage
  - 使用目的に外れた利用はできない

拡張フィールドのチェック

- 拡張フィールド(X.509v3以降)
  - 拡張フィールドは、証明書の種々の特性を定義できる
  - プライベートな拡張もあり
- 証明書の検証では
  - isCriticalフィールドがTRUEなら、解釈できないものは失効と扱う

無効な証明書のチェック

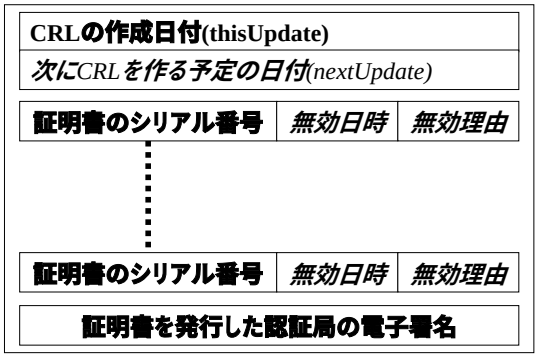
- 認証局は自分が発行した証明書を無効にする場合がある
  - 種々の理由で無効にする
    - 私有鍵（プライベート鍵）の紛失
    - 私有鍵（プライベート鍵）の盗難

    などなど
  - クレジットカードのブラックリストのように定期的に更新する場合が大半
  - 無効にしたの証明書のシリアル番号は、CRL(Certificate Revolation List)に登録

CRLのチェック

- CRLを入手
  - 認証局のリポジトリィ
  - 検証する証明書のCRLDistributionPoint
- CRLの電子署名を検証
  - CRLの偽造を検出するため
- CRLに検証対象証明書のシリアル番号がないことを確認

CRLの形式



イタリックはオプショナルフィールド

CRLの入手先は
1. 認証局のリポジトリィ
2. 検証対象のCRLDistributionPoint

示される

OCSPでの検証

- CRLでの証明書の無効チェックは基本であるが……
  - CRLは大きくなる(原則、無効にした証明書はすべて登録)
    - CRLの入手にコストがかかる場合がある
- チェックしたいのは、検証対象の証明書のみ
  - ほかはいらない
- CRLの代替としてOCSPプロトコルが提案されている
  - 証明書を指定してOCSPサーバ(OCSPレスポンダ)に問い合わせ
    - 有効、無効、不明が返る

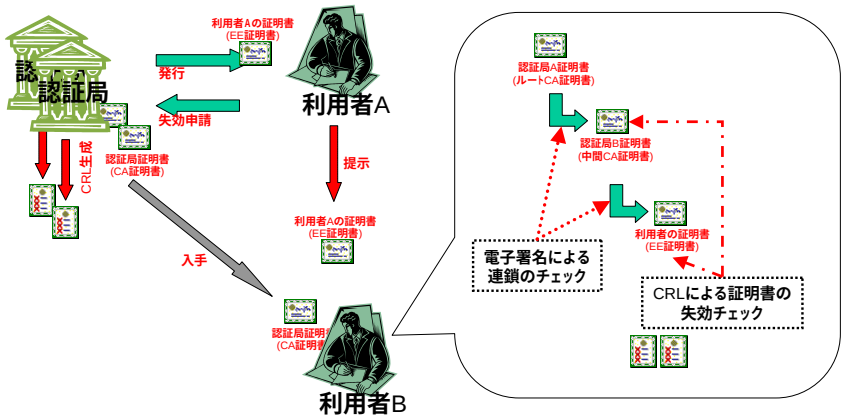
証明書検証のまとめ

- 証明書の利用による認証はユーザ名･パスワードに比較してより厳密かつ安全な認証の枠組みを提供できる
- 証明書の検証を得るためには……
  - 種々のチェックを実行
  - CRL (Certificate Revocation List)を参照するのが一般的
  - OCSPにより、証明書を指定して失効しているか否かを問い合わせる
  - 具体的な手段
    - 証明書にはCRL Distribution Point (CRLDP)がある!
      - 証明書が失効されたら、CRLDPに指定されている場所にあるCRLに登録されることを示している
    - CRLには、発行した認証局の電子署名がある!
      - 電子署名の検証により発行した認証局がわかっていれば、署名の検証により正しいCRLであることがわかる。
        - » Issuing Distribution Pointに発行した認証局の名前が入っていることもある
        - » CRLに証明書同様にAIA (Authority Information Access: 発行認証局情報)を入れるドラフトも出ている(CRL-AIA)
    - これらの情報を組み合わせて、CRLと証明書の関係を解釈

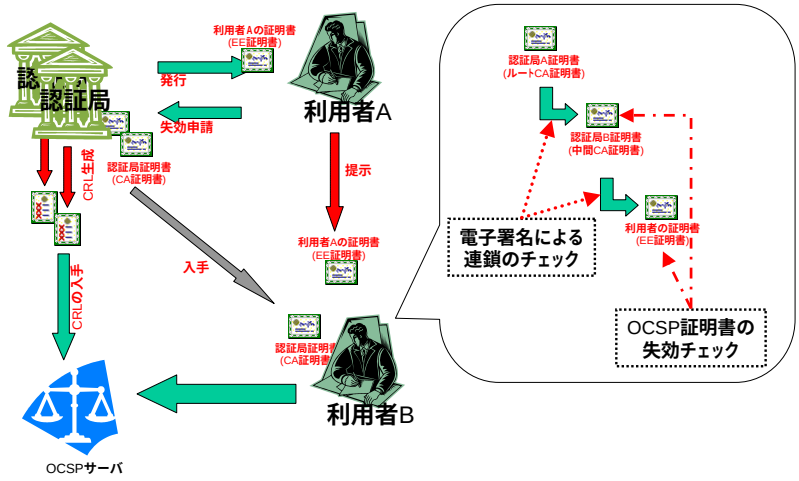
証明書の検証(続き)

- とまぁ、証明書の検証はかなり大変
  - Microsoft社のWindows/Sun Microsystems社のJAVAは、かなりがんばって検証可能であるが
    - 実際にプログラムを作るとなると大変
    - 種々のPKIモデルを理解するのは大変
      - 今回説明したものは、一番単純なモデル
    - CPUパワー、ネットワークのバンドワイズなどの資源が必要
- 一般のプログラマには無理?
  - とはいえ、証明書の検証は必要
    - いわゆるミドルウェアによるサポート
      - JAVAのPathBuilder/MicrosoftのCrypto APIなどのSecurity API
        - » IPAが「電子政府情報セキュリティ相互運用支援技術の開発」/「Security API」として報告書を掲載
        - » http://www.ipa.go.jp/security/fy14/development/pki/interop.html
        - » http://www.ipa.go.jp/security/fy15/reports/sec_api/index.html
    - 難しいところはサーバサイドで解決
      - SCVPなどの新たなサービス

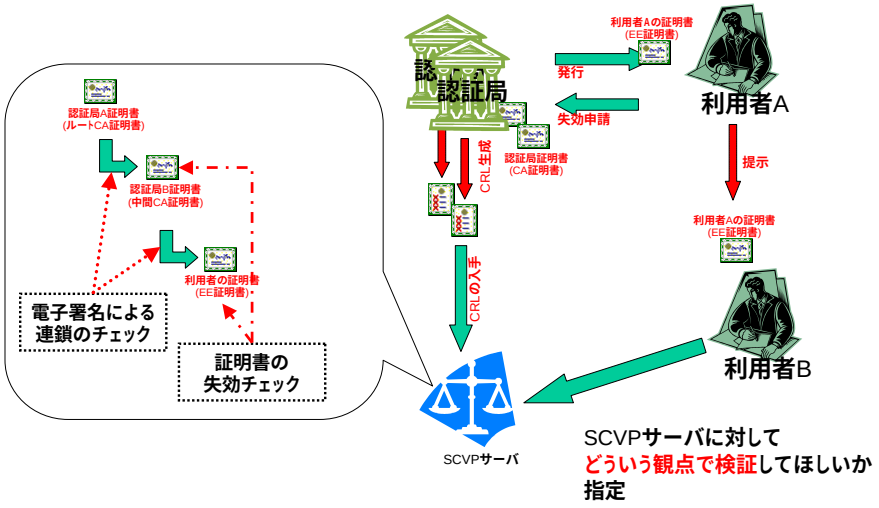
CRLでの証明書検証



OCSPでの証明書検証



SCVPでの証明書検証



SCVPサーバに対して
どういう観点で検証してほしいか
指定

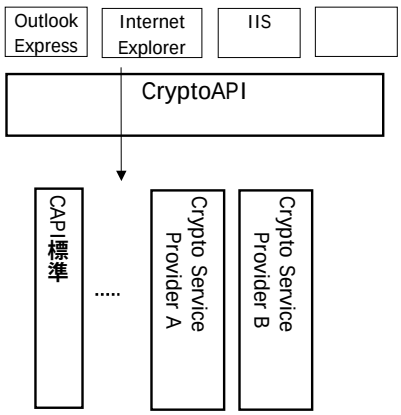
PKI実装

PKI実装面

- Windows系
  - Crypto API(Microsoft)
  - OpenSSL
- JAVA系
  - JDK/JCE
- UNIX系
  - OpenSSL

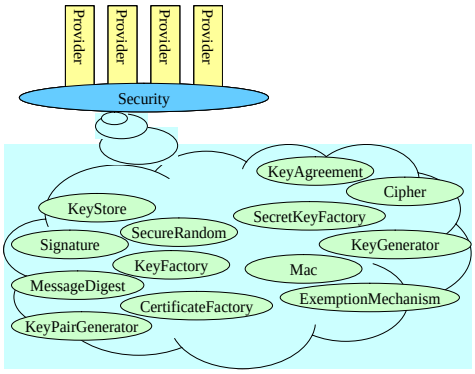
Windows Crypto API

- CSP(Crypto Service Provider)モデル
- IE 4.0以降から提供
- 暗号エンジンをモジュール化
- 複数の暗号エンジンを保持
- Third Party提供のCPSを利用可能
- 証明書の検証に関しても良く考えられている
- 暗号エンジンを作る場合、Microsoftにコード署名をしてもらう必要あり



JAVA/JCE

- JAVAの機能拡張モジュールとしてProviderモデルで実装されていた
  - 1.4より標準機能として実装されている
- 暗号機能/Hash機能/X509証明書操作機能を実装



JDK/JCE

- java.security.cert以下に実装されている。
- クライアントとして使う面では十二分な実装
  - JDK本体で証明書の基本的なハンドリングが可能
  - JCE(Java Cryptographic Extensions)で暗号周りの機能を提供
    - Windows同様Third PartyのJCEに差し替えることが可能
    - Sunより証明書を発行してもらい、その証明書でコード署名を行う必要あり
  - JSSE(Java Secure Socket Extensions)でSSL/TLSを提供
- RFC3280の証明書検証アルゴリズム相当のメカニズムを実装
- CertPathBulder/CertPathValidator/CertStoreの3つに仮想化
  - CertPathBulder
  - CertPathValidator
  - CertStore

OpenSSL

- 多くのUNIX系プラットフォームのデファクト実装
  - Linux/*BSD*に採用
- Windowsプラットフォームでも動作
- ApacheのSSL/TLSのエンジンとして広く使われている
  - Apache 1.X+mod_ssl+OpenSSL
  - Apache 1.X+Apache_SSL+OpenSSL
  - Apache 2.X(標準でSSL/TLSをサポート)

参考文献(国際標準関連)

- ITU-T Recommendation X.509 | ISO/IEC 9594-8: Information technology - Open Systems Interconnection - The Directory: Authentication framework. , 1997
- R. Housley, W. Ford, W. Polk, and D. Solo, RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, 1999
  - http://www.ietf.org/rfc/rfc2459.txt
- M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams , RFC 2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, 1999
  - http://www.ietf.org/rfc/2560.txt
- R. Housley, W. Polk, W. Ford, D. Solo, RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile . 2002
  - http://www.ietf.org/rfc/rfc3280.txt
- S. Santesson, M. Nystrom and T. Polk, RFC 3739: Internet X.509 Public Key Infrastructure: Qualified Certificates Profile
  - http://www.ietf.org/rfc/rfc3739.txt
- M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph and R. Nicholas, RFC 4158: Internet X.509 Public Key Infrastructure: Certification Path Building, 2005
  - http://www.ietf.org/rfc/rfc4158.txt
- T. Freeman, R. Housley, A. Malpani, D. Cooper and T. Polk, Internet Draft: Standard Certificate Validation Protocol (SCVP) , 2005
  - http://www.ietf.org/internet-drafts/draft-ietf-pkix-scvp-21.txt
- ETSI TS 101 862 V1.3.2 (2004-06)
  - Title: Qualified Certificate profile
- ETSI TS 101 456 V1.2.1 (2002-04)
  - Title: Policy requirements for certification authorities issuing qualified certificates
- ETSI TS 102 158 V1.1.1 (2003-10)
  - Title: Electronic Signatures and Infrastructures (ESI);Policy requirements for Certification Service Providers issuing attribute certificates usable with Qualified certificates

参考文献(報告書関連、書籍)

- **報告書関連**
  - IPA, PKI **関連技術情報**, 2004-2005
    - http://www.ipa.go.jp/security/pki/pki.html
  - IPA/JNSA, **セキュリティ**API**に関する技術調査**, 2004
    - http://www.ipa.go.jp/security/fy15/reports/sec_api/index.html
  - IPA/JNSA,**電子政府情報セキュリティ相互運用支援技術の開発** GPKI **相互運用フレームワーク**, 2004
    - http://www.ipa.go.jp/security/fy14/development/pki/interop.html

- **書籍**
  - **小松 文子**, PKI**ハンドブック**, ISBN 4883732053,**ソフトリサーチセンター**, 2004, 255p
  - **日本ネットワークセキュリティ協会**, **情報セキュリティプロフェッショナル総合教科書**, ISBN 479800880X, **秀和システム**, 2005, 575p
  - **青木 隆一**, **稲田 龍**, PKI**と電子社会のセキュリティ**, ISBN 4320120280, **共立出版**, 2001, 233p

登録商標等について

- Microsoft、MS、Windows、Windows 2000、Windows NT、Windows XP、Windows**ロゴ**、Internet Explorer、Outlook、Outlook Express**などは、米国**Microsoft Corporation **の米国およびその他の国における登録商標または商標である。**
- Sun Microsystems、Sun**ロゴ**、Java **コーヒーカップロゴ**、Solaris、Java、JDK**などは、米国**Sun Microsystems**の米国およびその他の国における登録商標または商標である。**
- **その他、本文小児記載されている会社名、商品名、製品名などは、一般に各社の商標または登録商標である。**
- **本書では、™、©、®などを記載しない**

THE DOCUMENT COMPANY
FUJI XEROX

ご清聴ありがとうございました