クライアント認証を使った アプリケーション開発

～PKI とパスワードレスなアプリ開発～

畑中貴之

自己紹介

- 名前は畑中貴之。31歳です。
- Javaでの業務アプリ開発とかフレームワーク開発がメインです。たまに技術ドキュメントとか書いています。
- 今後取り組みたいテーマ
  - Enterprise Integration Pattern
  - ポータル (MS SharePoint, JSR286)
  - 論理は「結論」と「根拠」から

Copyright 2006 Takayuki Hatanaka All rights reserved.

講習の内容

- 1. PKI とクライアント認証とは？
- 2. クライアント認証の環境構築方法
- 3. Java APIつかって証明書へアクセス

Copyright 2006 Takayuki Hatanaka All rights reserved.

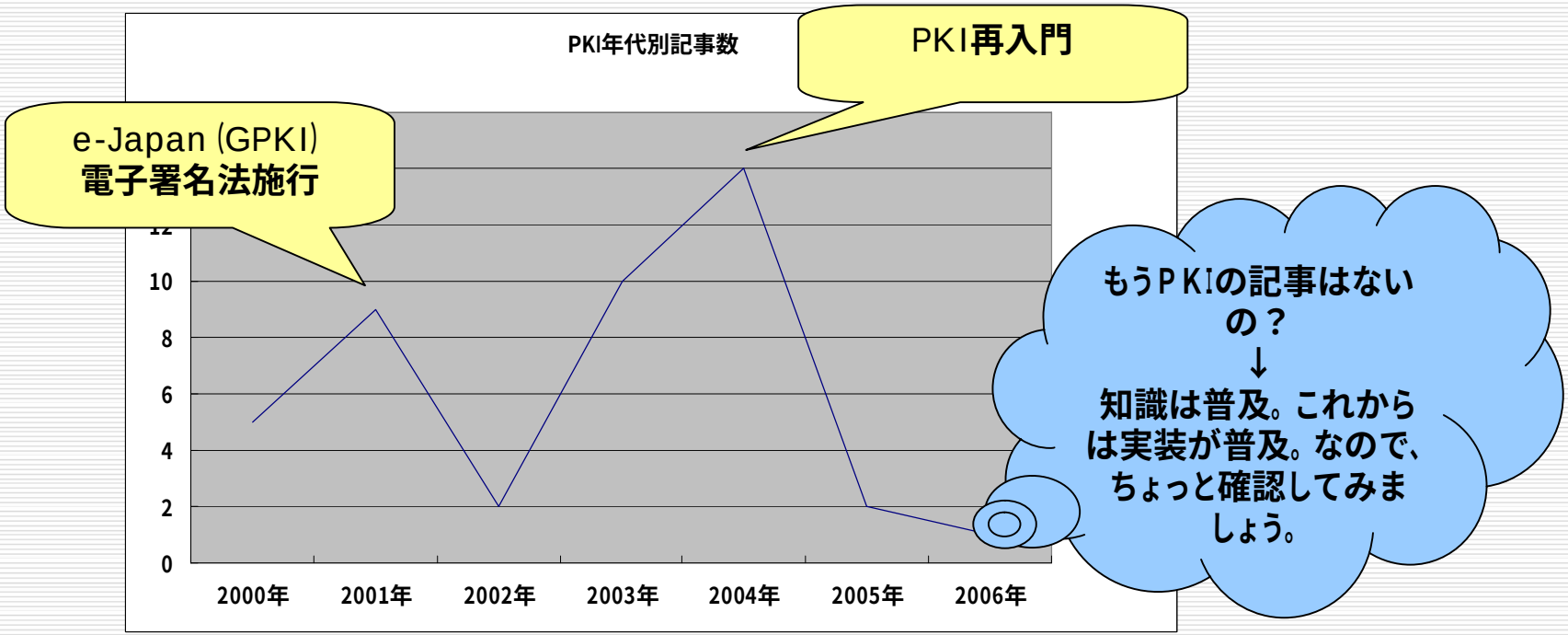
Warmup - 暗号使ってラブレター -

- これはなんと読むのでしょうか？

Copyright 2006 Takayuki Hatanaka All rights reserved.

もう当たり前なの、PKIって？

- ITサイトのPKI関連の記事数



Copyright 2006 Takayuki Hatanaka All rights reserved.

第1部　PKIとは

Copyright 2006 Takayuki Hatanaka All rights reserved.

便利なネットワークと脅威

- インターネットが普及して、何もかもネットワーク経由。便利な時代だなと。
- でも悪い人はいつの時代もいる。
  - 盗聴
  - 不正アクセス
  - なりすまし
  - 改ざん
  - 否認

＜＜度重なる不正アクセス＞＞

2006.6.13 KDDI 400万人顧客情報流出

2006.8.30 AT&T 1万9千人顧客情報流出

・オレオレ詐欺、ソーシャル・ハッキング

＜＜制度化されていくセキュリティ＞＞

個人情報保護法

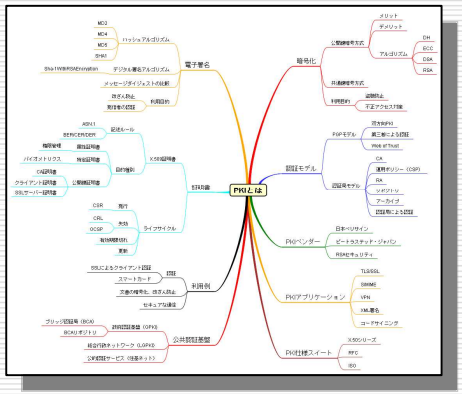
プライバシーマーク

日本版SOX法

Copyright 2006 Takayuki Hatanaka All rights reserved.

PKIとは

- Public Key Infrastructure
- 暗号化とデジタル署名を用いた認証インフラ
- 暗号化を公開鍵と非公開鍵で行う



Copyright 2006 Takayuki Hatanaka All rights reserved.

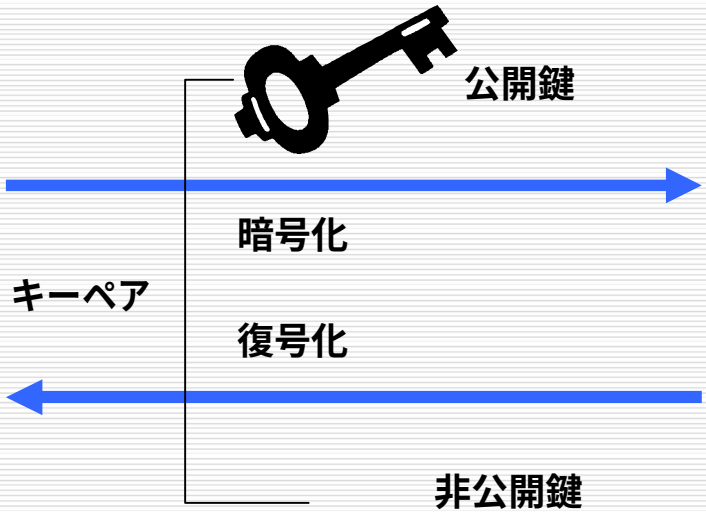
PKIによって何ができるの？1

暗号化

・データを暗号化して、内容を確認できないようにする。(盗聴、不正アクセス対策)

・暗号化と復号化で異なる鍵を使う。

・メジャーなアルゴリズムはRSA



Copyright 2006 Takayuki Hatanaka All rights reserved.

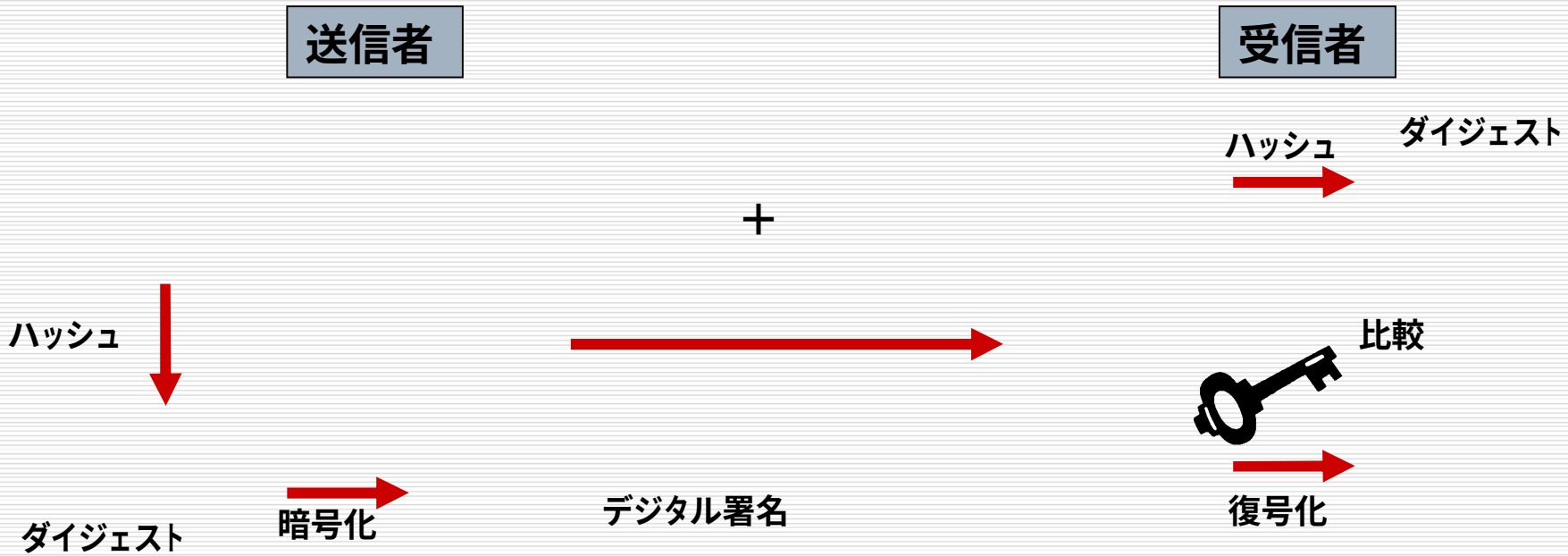
PKIによって何ができるの？2

デジタル署名

・ダイジェスト同士を比較

・改ざんや否認防止、送信者の身元確認（公開鍵の特性を利用）

・MD5とかSHA1とか。



Copyright 2006 Takayuki Hatanaka All rights reserved.

公開鍵と証明書

- 公開鍵は単なるビット列
- 認証局が発行する電子証明書によって、公開鍵の持ち主が証明される。
「確かに畑中さんの公開鍵です。」

```
RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          db:d8:2a:26:5c:9a:1b:21:c4:c4:d3:1b:ef:e6:b1:26:77:e2:86:e7:
          e1:c6:71:07:7b:a5:84:e0:5d:74:b3:19:09:fd:a1:26:62:e6:21:9e:
          df:6e:72:6a:ec:65:ca:40:ea:07:05:2e:61:c6:97:1b:4c:a7:0d:b1:
          19:05:d4:8b:46:2b:62:76:7b:94:04:e6:8f:a4:48:63:36:57:c6:79:
          24:3c:71:b2:31:73:a7:8e:b8:96:3d:27:7b:1c:5d:ea:64:5c:28:c0:
          e4:6a:8c:03:05:59:64:e7:95:04:42:d2:85:67:eb:df:6b:fc:0b:b5:
          5b:ff:e0:35:4a:ec:12:c9:
        Exponent:
          00:01:00:01:
```

Copyright 2006 Takayuki Hatanaka All rights reserved.

X.509 証明書

- ITU-Tが作成した X.509 (X.509 v3)
- RFC2459 → RFC3280
- 公開鍵証明書
  属性証明書
  クオリファイド証明書
- ASN.1 形式で記述
  pem, der エンコーディング

X.509 フォーマット

X.509のバージョン番号
証明書のシリアル番号
電子署名のアルゴリズム
発行者
電子証明書の有効期限
電子証明書の所有者 (プリンシパル)
所有者の公開鍵
その他、ユニークな識別子や拡張領域

Copyright 2006 Takayuki Hatanaka All rights reserved.

認証局 (CA)

- 電子証明書を発行する機関
- 運用ポリシー (CP/CPS) RFC 2527/3647
- CA自身が上位のＣＡによって証明される。

Copyright 2006 Takayuki Hatanaka All rights reserved.

PKI で実現されていること

- TLS/SSL
  - クライアント・サーバー間のセキュアな通信
- S/MIME
  - 電子メールの暗号化とデジタル署名
- VPN
  - インターネットでのネットワークの暗号化
- XML 署名
  - XML文書へのデジタル署名
- コードサイニング
  - Webで公開されるプログラムに電子署名

Copyright 2006 Takayuki Hatanaka All rights reserved.

クライアント認証とは？

- クライアントを識別するための認証方法
  - よくあるのがID、パスワード（単因子認証）
  - サーバー認証はサーバ証明書
    クライアント認証ではクライアント証明書
  - 証明書が奪われたら一緒？
    →PIN（二因子認証）
    →ICカード、USBトークン

Copyright 2006 Takayuki Hatanaka All rights reserved.

クライアント認証の導入事例

- 身分証明書にICカード（with 証明書）
  - 東北電力
  - 14,000 人の出退勤＋リアル･バーチャルの身分証明
- 法人向けオンラインバンキング
  - 東京三菱UFJ銀行
  - クライアントを認証するだけでなく、権限との紐付けを実現

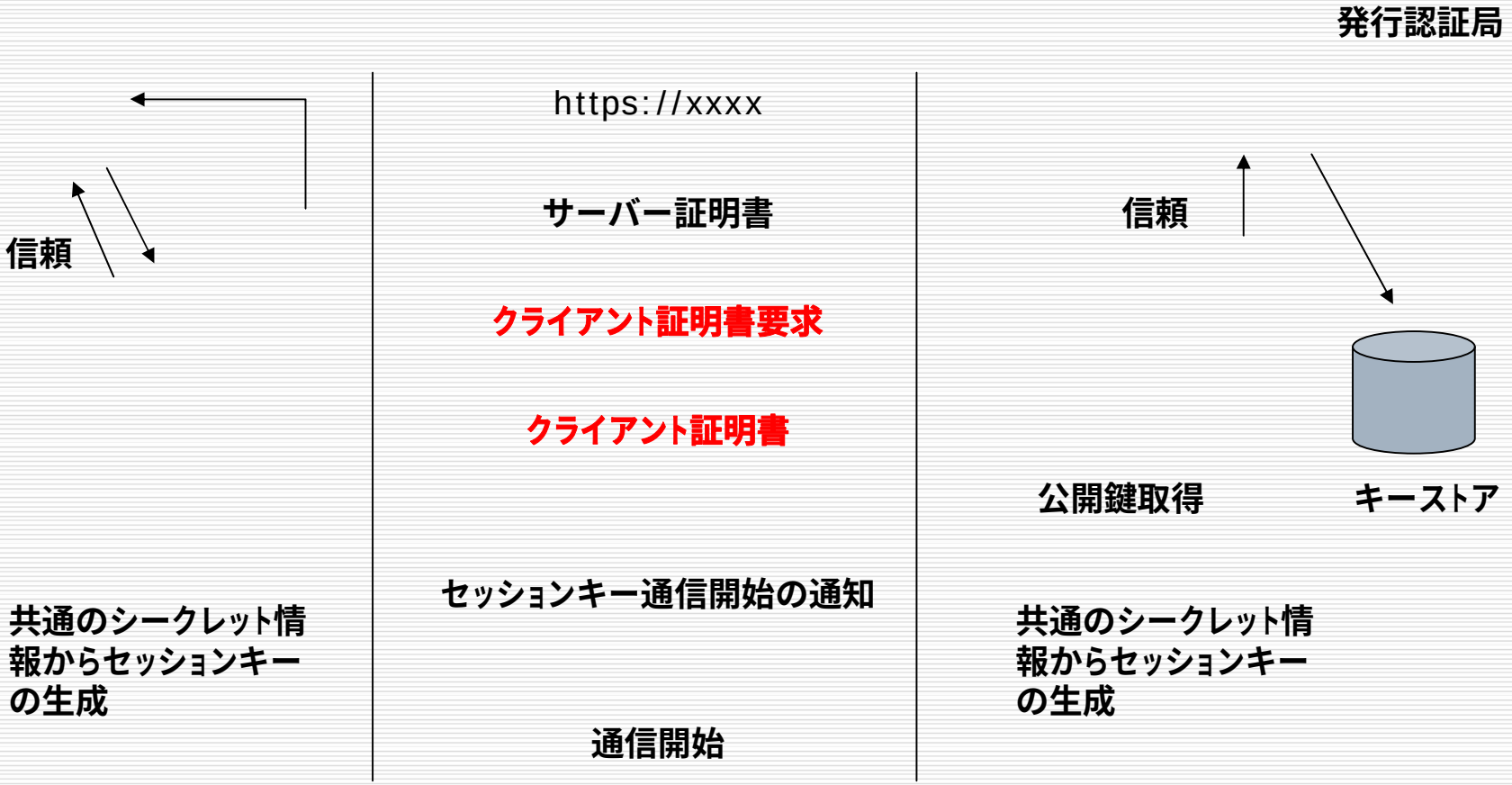
（PKIベンダー導入事例より引用）

Copyright 2006 Takayuki Hatanaka All rights reserved.

第2部　クライアント認証の環境構築

Copyright 2006 Takayuki Hatanaka All rights reserved.

クライアント認証の手順



Copyright 2006 Takayuki Hatanaka All rights reserved.

クライアント認証の環境構築手順

●CA構築
CA証明書の作成

●サーバーのための設定
サーバー証明書の作成
キーストアの作成(CA証明書のインポート)
httpリスナーの設定

●クライアントのための設定
クライアント証明書の作成
クライアント証明書のブラウザへインポート

Copyright 2006 Takayuki Hatanaka All rights reserved.

ＣＡ構築

- OpenSSL
  - オープンソースのSSL実装ツールセット
- CAの鍵生成
  - openssl genrsa –out ca.key 1024
- CA証明書作成
  - openssl req –new –x509 –key ca.key –out ca.crt
- CA情報定義
  - ca.config

Copyright 2006 Takayuki Hatanaka All rights reserved.

keytool

- 鍵と証明書の管理ツール
- キーストア
  - キーエントリと信頼する証明書エントリ
  - 別名
- キーペアの作成と自己署名証明書
  - keytool -genkey -alias myserver -keyalg RSA -keystore .keystore
- 信頼する証明書エントリの登録
  - keytool -import -v -trustcacerts -alias myca -file ca.crt -keystore .keystore

Copyright 2006 Takayuki Hatanaka All rights reserved.

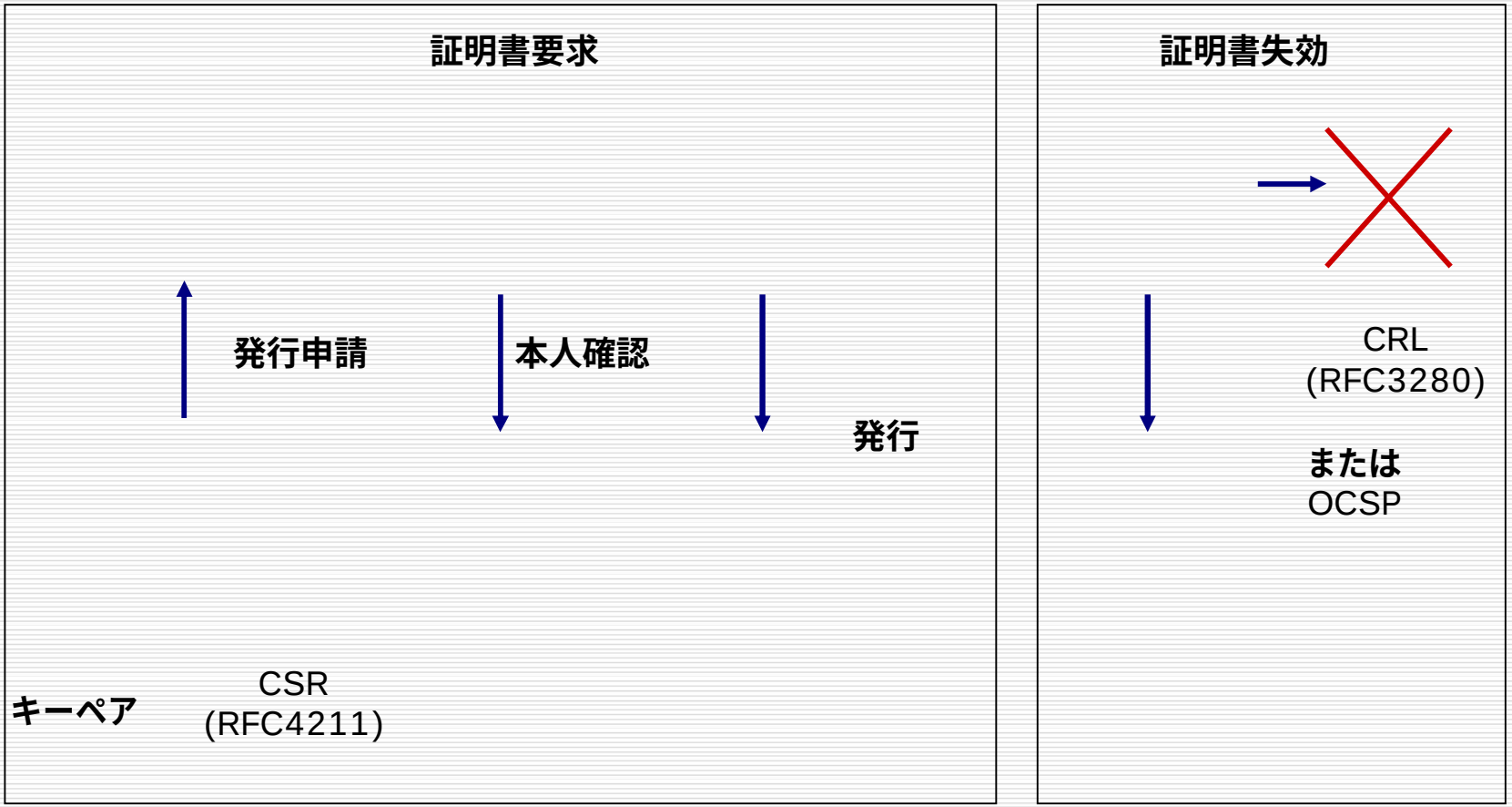
HTTPリスナーの設定(Tomcat)

- server.xml
  - clientAuth=“true”
  - keystore, truststore

```
<Connector port="8443"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" disableUploadTimeout="true"
        acceptCount="100" debug="0" scheme="https" secure="true"
        clientAuth="true" sslProtocol="SSL"
        keystoreFile=".keystore"
        keystorePass="changeit"
        truststoreFile= ".keystore"
        truststorePass="changeit" />
```

Copyright 2006 Takayuki Hatanaka All rights reserved.

証明書のライフサイクル



Copyright 2006 Takayuki Hatanaka All rights reserved.

クライアント証明書の作成

- 秘密鍵の生成
  - openssl genrsa -out client.key 1024
- 証明書要求 (CSR) の作成
  - openssl req -new -key client.key -out client.csr
- 証明書の作成
  - openssl ca -config ca.config -out client.crt -infiles client.csr

Copyright 2006 Takayuki Hatanaka All rights reserved.

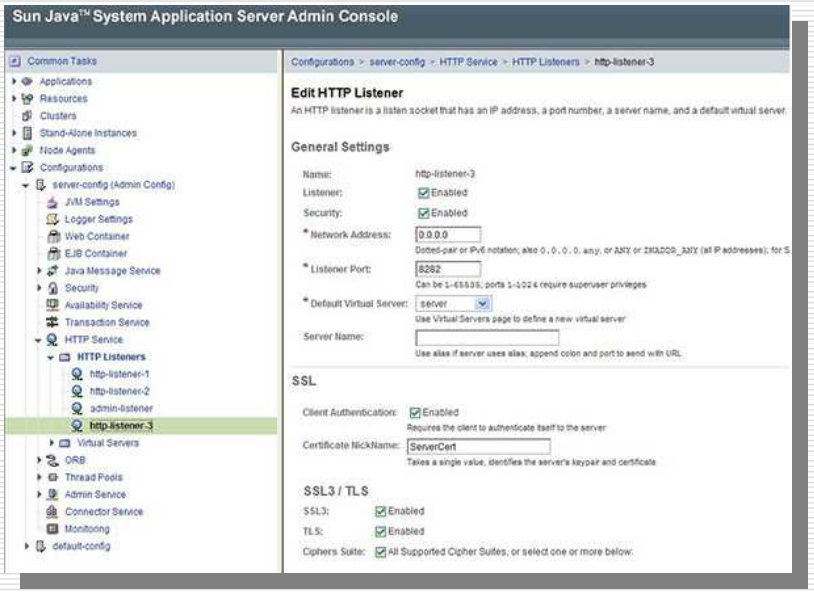
ブラウザへ取り込む（IE）

- PKCS#12形式にエクスポート
  - openssl pkcs12 -export -out client.p12 -in client.crt -inkey client.key -name myclient -certfile ca.crt
- PKCSって？
  - RSAセキュリティが定義した証明書や鍵のフォーマットを定めた規格セット。1ー15（14は未定義）
  - PKCS#12（秘密鍵＋証明書連鎖）

Copyright 2006 Takayuki Hatanaka All rights reserved.

(参考) Sun Java Application Server の場合

- CertUtil
  - Sun Java Application Server が提供する鍵、証明書管理ツール
  - Enterprise Edition で提供
- HttpListener



Copyright 2006 Takayuki Hatanaka All rights reserved.

当たり前かもしれませんが。(環境構築編)

- クライアント認証では、クライアント証明書のCA証明書を信頼せよ。
- Webサーバーとの連携方法がいつの間にか変わってしまった！？(Sun App)
  - <property name="authPassthroughEnabled" value="true"/>

Copyright 2006 Takayuki Hatanaka All rights reserved.

第3部　証明書へアクセスする

Copyright 2006 Takayuki Hatanaka All rights reserved.

証明書アクセス Java API (JSSE)

- java.security.cert package
  - CertificateFactory
  - Certificate
  - CRL
  - X509Certificate
  - X509Extension
  - X509CRL
  - X509CRLEntry

Copyright 2006 Takayuki Hatanaka All rights reserved.

実装例

□ リクエスト属性にいます。

```
// 証明書情報取得
X509Certificate[] certs =
    (X509Certificate[])request.getAttribute("javax.servlet.request.X509Certificate");
X509Certificate cert = certs[0];

// シリアル番号を取得
BigInteger bigInteger =cert.getSerialNumber();
String hexSerial = encode(bigInteger.toByteArray());

// 有効期限
String notBefore = cert.getNotBefore().toString();
String notAfter = cert.getNotAfter().toString();
```

Copyright 2006 Takayuki Hatanaka All rights reserved.

キーストアへアクセス

- java.security.KeyStore クラス
  - KeyStore#getInstance()
  - KeyStore#load()
  - KeyStore#aliases()
  - KeyStore#getCertificate()
- クライアント認証で取得した証明書でさらに細かいチェックをかけたいときなど。

Copyright 2006 Takayuki Hatanaka All rights reserved.

実装例

□ キーストアの読み込みとエイリアスの取得

```
// キーストアの取得とロード
store = KeyStore.getInstance("JKS", "SUN");
store.load(new FileInputStream(KEYSTORE_PATH),
                   KEYSTORE_PASSWORD.toCharArray());

// キーストアの証明書の取得、エントリ種別の判定
entry.setCertInfo(CertUtil.createCertInfo(
    (X509Certificate)store.getCertificate(alias)));
entry.setKey(store.isKeyEntry(alias));
entry.setTrust(store.isCertificateEntry(alias));
```

Copyright 2006 Takayuki Hatanaka All rights reserved.

どんな実装が考えられる？

- クライアント認証＋権限チェック
  - ServletFilterで実装
- クライアントと権限のマッピングは？
  - 権限テーブルとのマッピング
  - 属性証明書は使える日は来るの？
- キーストア経由で生成した証明書を書き換え

Copyright 2006 Takayuki Hatanaka All rights reserved.

当たり前かもしれませんが。（アプリ開発編）

- javax.security.cert.X509Certificate のほうが古いの？
- 128ビット対応の無制限強度？（米国の陰謀）
  - local_policy.jar
  - US_export_policy.jar

Copyright 2006 Takayuki Hatanaka All rights reserved.

最後に今後さらに注目される？PKI

- 属性証明書 (RFC3281)
  - 証明書に権限情報などを持たせることが可能
  - AA (Attribute Certificate Authority) が必要
  - まだまだ世間的には広まっていない
- クオリファイド証明書 (RFC3739)
  - 自然人を対象とした証明書
  - 指紋などの生体情報が格納可能

Copyright 2006 Takayuki Hatanaka All rights reserved.