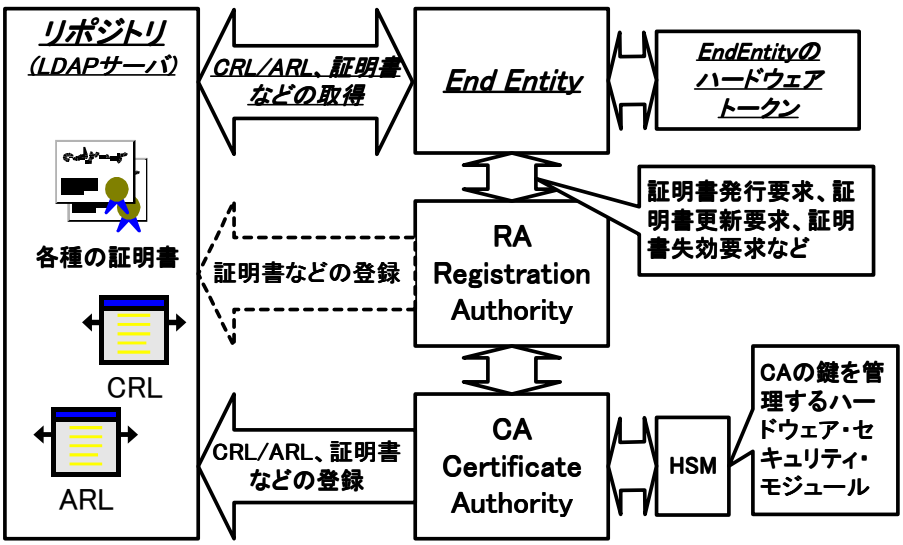
PKI　基礎編

セコムトラストネット株式会社
松本　泰
yas-matsumoto@secomtrust.net

PKI 基礎編

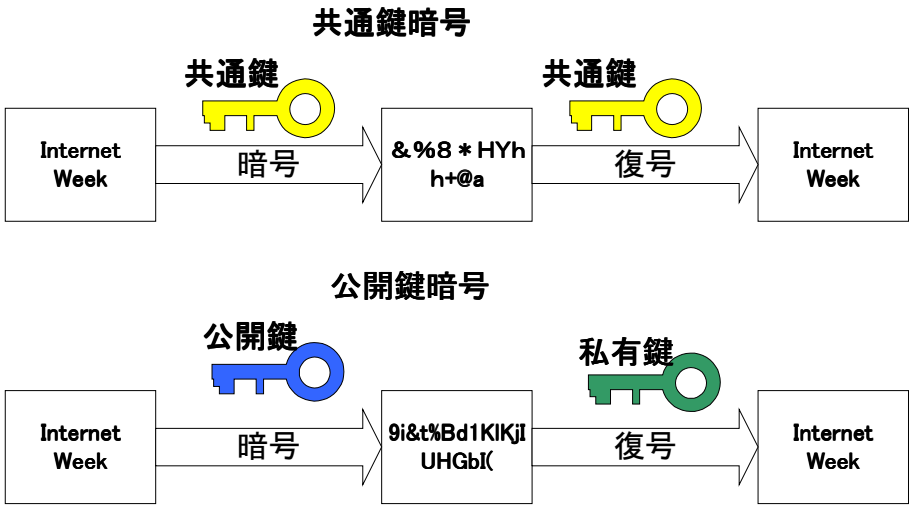
- 公開鍵暗号
- PKIの信頼モデル
- X.509 公開鍵証明書
- LADPによるリポジトリ
- ハードウェアトークン
- 証明書失効検証(CRL, OCSP)
- PKIアプリケーションの基本的な要件
- 証明書発行

PKIの基本コンポーネント



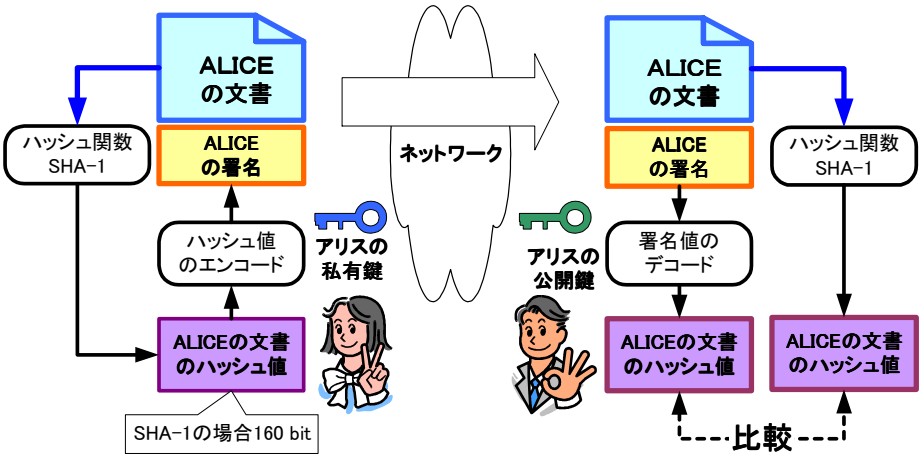
3

共通鍵暗号と公開鍵暗号



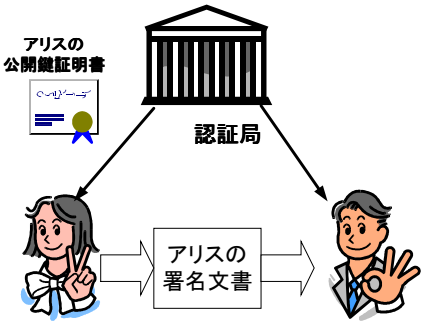
4

署名の仕組み

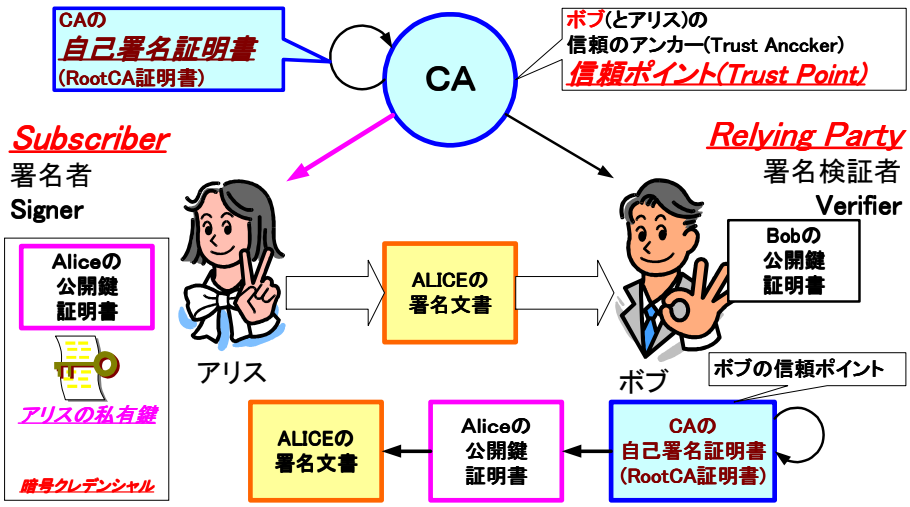


TTPよる認証
（アリスの公開鍵を信じるのか）

- TTP(Trusted Third Party)とは
  - 信頼できる第三者機関
  - TTPによって署名されたデータは信用できるものとする
  - 代表的な例はCA（Certificate Authority）
  - CAは印鑑証明を発行してくれる役所のイメージ

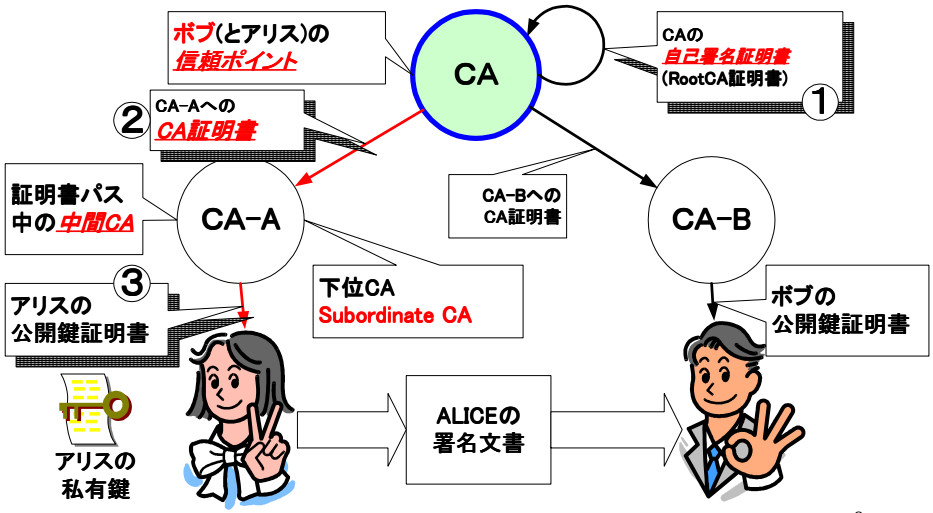


PKIの基本的な信頼モデル



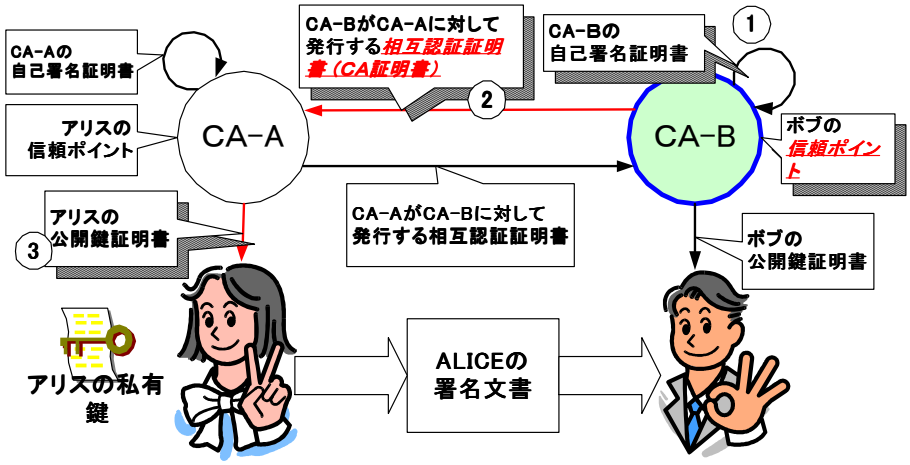
7

階層型CAモデル



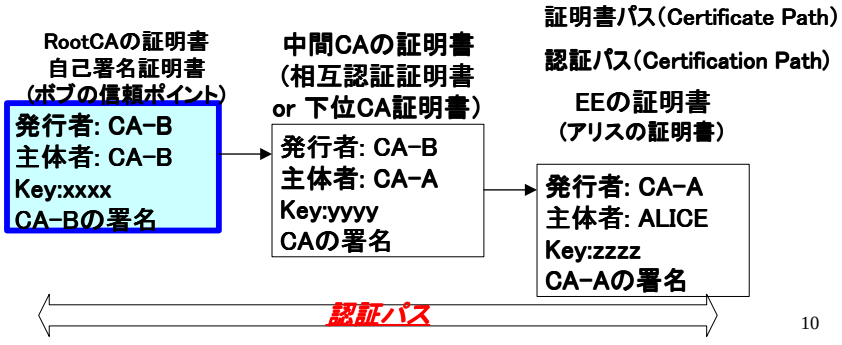
8

相互認証モデル（Cross certificate）



認証パスとは何か？

- ボブ(RP)はアリス(SC)からのメッセージを受け取った。
- ボブは、アリスからのメッセージの署名を検証したい
- 自分(ボブ)の“*信頼のポイント*”(ボブのRootCA)からの*認証パス*を検証する
- 検証は署名のチェーンの検証だけでなく、各証明書の失効チェック、そしてX.509証明書拡張に関する検証が行われる。



X.509証明書

証明書バージョン番号（V3）
証明書シリアル番号
デジタル署名アルゴリズム識別子
発行者名の識別名
有効期間
主体者（ユーザ）の識別名
主体者の公開鍵
　アルゴリズム識別子
　公開鍵値
V3の拡張
拡張フィールド（タイプ、フラグ、値）
拡張フィールド（タイプ、フラグ、値）
CAのデジタル署名
　アルゴリズム識別子
　署名

- **代表的な公開鍵証明書**
  - **主体者(アリス)と、主体者(アリス)の公開鍵や、その他の属性をCA鍵(アリスの証明書を発行したCAの署名鍵)の署名でバインドする。**
  - **この時、主体者(アリス)の公開鍵に対応した私有鍵は、主体者(アリス)しか使用できないことが理想。**
- **1997年版 X.509 3rd Edition**
  - X.509v3証明書フォーマット
    - X.509V3拡張
  - 14の標準拡張フィールド

＊ GPKIなどでは、X.509v3証明書フォーマットが使用されており、かつ拡張が、重要な意味を持つ。

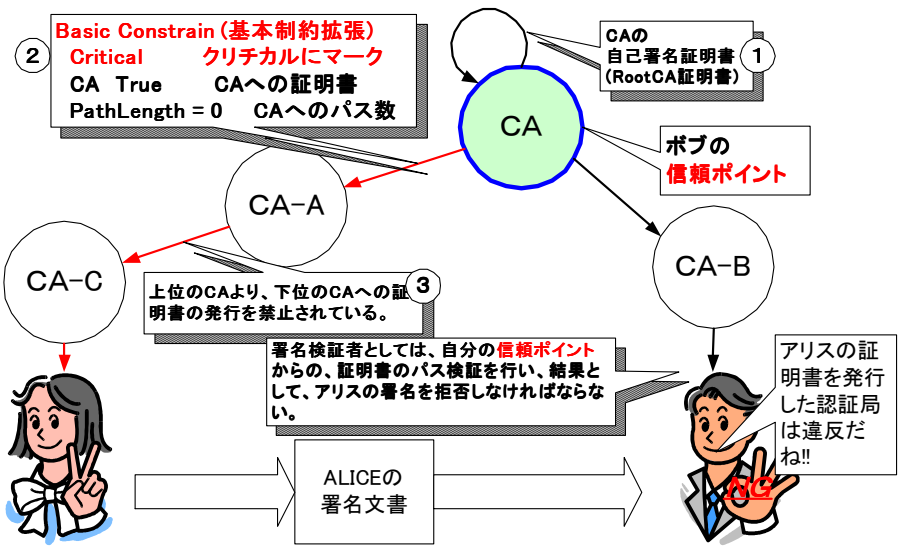
X.509証明書拡張(v3拡張)

No	標準拡張(X.509v3)	説明
1	発行者鍵識別子	発行者の鍵の識別に使用されCA鍵の更新に必要
2	主体者鍵識別子	主体者の鍵の識別に使用されCA鍵の更新に必要
3	***鍵使用方法***	私有鍵の使用方法。例えば署名用鍵で、暗号化を禁止する
4	私有鍵有効期間	証明書の有効期間に対して、私有鍵の有効期間。
5	***証明書ポリシ***	証明書ポリシIDなどが格納される。ポリシによる制御などに使用
6	***ポリシマッピング***	信頼ドメイン間のポリシのマッピングを行う
7	主体者別名	主体者の別名が格納される。例えばVPN装置の場合のIPaddress
8	発行者別名	発行者の別名が格納される。
9	主体者ディレクトリ属性	証明書の主体者のためのディレクトリ属性
10	***基本制約***	証明書の種類(CAorEE)。CAだった場合パス数の制限
11	名前制約	CA証明書で、相手のCAが発行する名前による制約
12	ポリシ制約	CA証明書で、相手のCAが発行するポリシ関係制約
13	拡張鍵使用方法	"鍵使用方法"以外の鍵使用方法のOIDが格納される。
14	CRL配布点	失効情報リストの配布点のDNやURLが格納される。

X.509 の証明書フォーマット

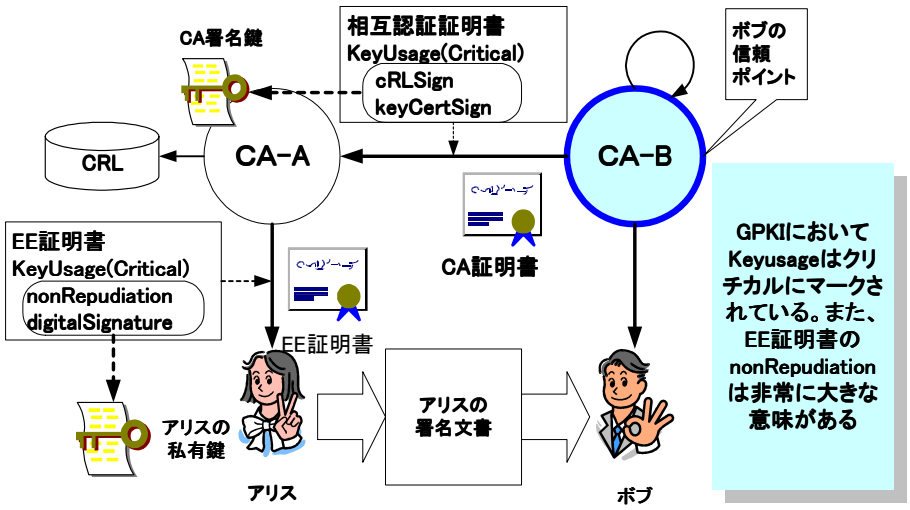
X.509 Edition	証明書 フォーマット	CRL フォーマット	備考
1st Edtion 1988	V 1	V 1	古いrootCAの証明書にV1 フォーマットのものがある
2nd Edtion 1994	V 2	V 1	ほとんど使用されていない??
3rd Edtion 1997	V 3	V 2	14個の(v3)標準拡張フィールド
4th Edtion 2000	V 3	V 2	標準拡張フィールドがひとつ追加された

X.509v3証明書の基本拡張基本制約拡張



** GPKIでは、CA証明書でクリチカルな基本制約拡張を必須としている。

鍵使用目的拡張(Key Usage)



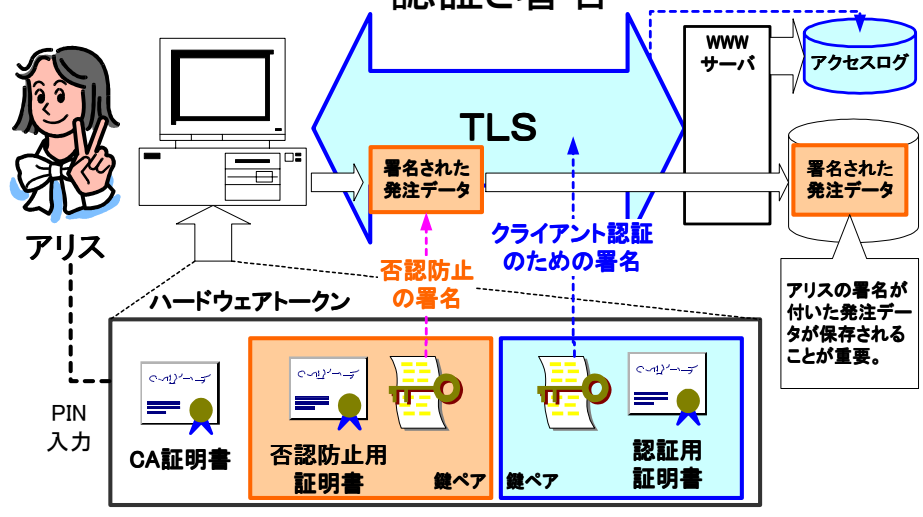
** 例えば、GPKIでは、CA証明書、EE証明書で、クリチカルな鍵使用目的拡張を必須としている

15

鍵使用目的拡張(Key Usage)

- 鍵使用目的拡張
  - 証明書に対応した私有鍵が使用目的をビット列で指定
  - デジタル署名、否認防止、鍵暗号化、データ暗号化、鍵交換、証明書署名、CRL署名、暗号化/複合化の指定。
- EEの私有鍵による署名と key Usage の関係
  - 否認防止 – Non Repudiation(NRビット)
    - 文書へのデジタル署名
  - デジタル署名 – Digital Signature(DSビット)
    - Nonce(乱数)へのデジタル署名
    - 認証(Authentication)用途

認証と署名



銀行を中心としたPKIであるIdentrusや、フィンランドの市民カードでのあるFINEIDでは、この例のように2つの証明書を発行している。

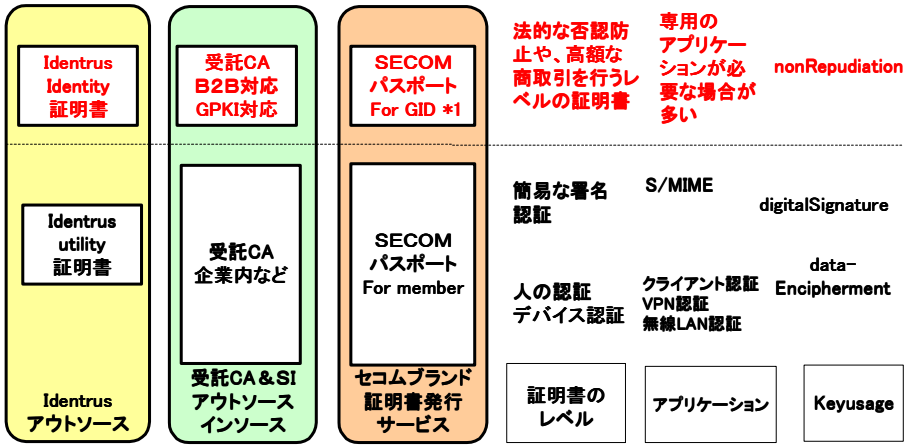
FinEIDの2つの保有者証明書と私有鍵

No	証明書の使用目的	私有鍵を使用するための認証方法	X.509証明書拡張鍵使用目的
1	認証や暗号化で使用する。 利便性が優先させる目的に使用される。 SingleSingOnなどにも使用される。	この私有鍵を使用するには、トークンの中のグローバルPINによる認証を行う。 グローバルPINは他のアプリケーションと共有される。	digitalSignature keyEncipherment dataEncipherment
2	法的な意味を持つ否認防止の署名のために使用される。	この私有鍵を使用するには、この私有鍵だけのためのローカルPINによる認証を行う。この認証は、私有鍵での署名操作でリセットされる。(毎回PINが要求される) *1	nonRepudiation

*1 こういった機能の実装のため、PKCS#15には、userConsent、PKCS#11には、CKA_SECONDARY_AUTH属性などといった新しい仕様がある。

3

鍵使用目的拡張と証明書発行の関係
(セコムトラストネットの証明書発行サービスの例)



*1 SECOMパスポートfor GIDは、電子署名法特定業務認定取得済み、GPKI相互認証を申請中

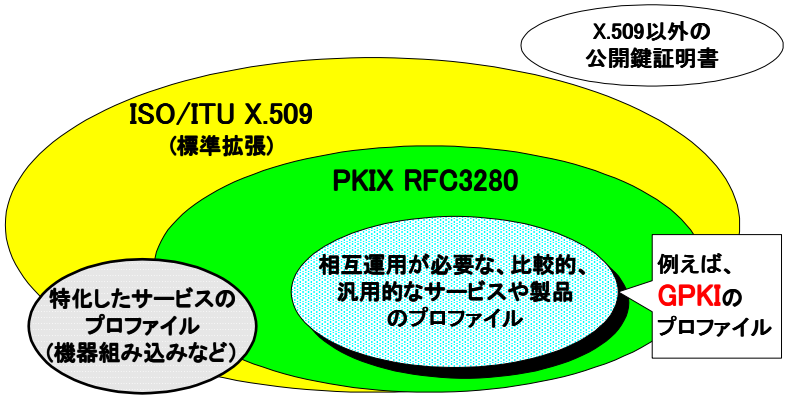
4

RFC3280（証明書プロファイルとは何か?)

- RFC3280（ RFC2459 ）
  - Internet X.509 Public Key Infrastructure Certificate and CRL Profile
- プロファイルとはなにか??
  - 汎用的なX.509証明書に対して、使用方法を限定することで、アプリケーションの実装を容易にする。
  - #それでも、RFC3280は、汎用的な証明書プロファイル
- 証明書プロファイルとして記述されること
  - 各証明書における、各証明書拡張の存在や使用方法
    - 存在がMUSTなのか、SHOULDなのか、MAYなのか??
    - フラグがクリチカルなのか?、ノンクリチカルなのか?
- 各証明書拡張のクリチカルフラグの扱い
  - 証明書拡張がクリチカルの場合処理が強制させる
- X.509標準証明書拡張にない証明書拡張の定義
  - AIA(Authority Information Access)。OCSPなどで使用される。

5

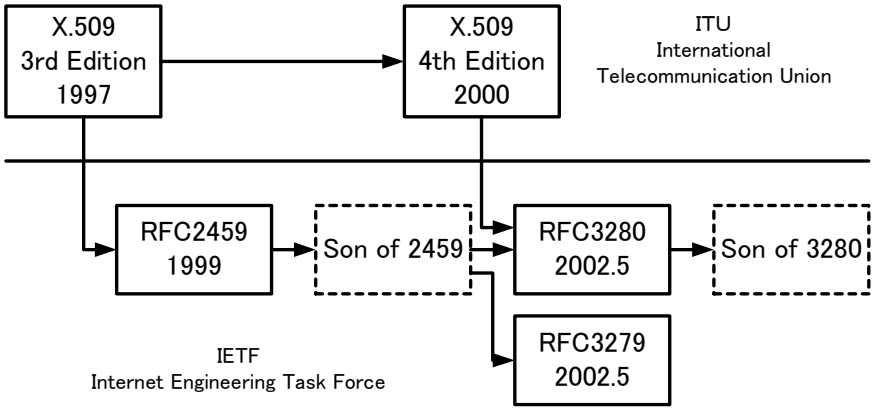
証明書のプロファイルの関係



- RFC3280（RFC2459）準拠の意味するもの
  - 証明書発行そのものよりも、そのプロファイルを解釈するアプリケーションの実装が格段に難しい。アプリケーションにおいて、100％ RFC3280サポートは、まずない。

6

X.509とRFC3280



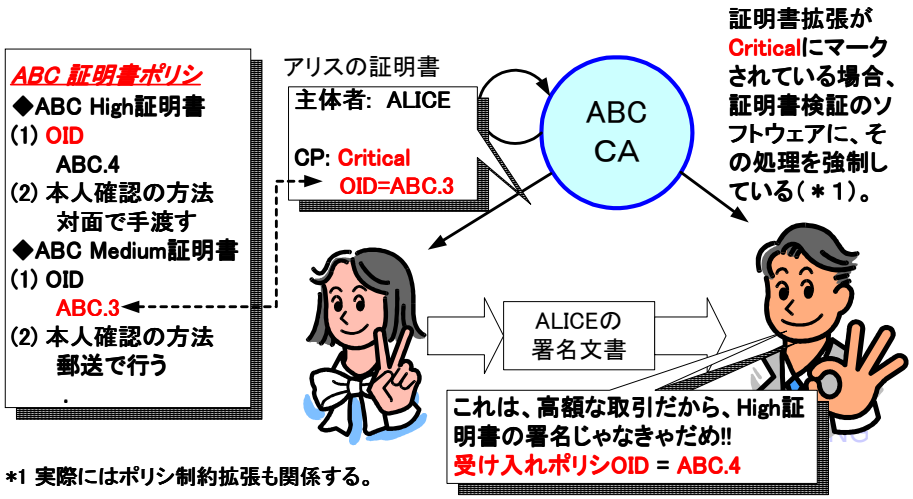
7

証明書ポリシとは何か？

- 証明書ポリシのX.509での定義
  - 特定のコミュニティ又はアプリケーションのクラスに共通のセキュリティ要求をもって、証明書の適用性を指定する規則の名前付けした集合
- RFC2527
  - インターネット X.509 PKI 証明書ポリシー(CP)と認証実施 フレームワーク
- 一般的な証明書ポリシの内容
  - 認証局の発行者。署名者の義務。証明書の正しい用途や署名確認に関するリライングパティへの要求事項
- 証明書ポリシ(CP)とCPS(Certificate Practice Statement)
  - CPは、“何を(what)サポート” 。CPSは、“どのよう(how)にサポート”
- X.509証明書の証明書ポリシ拡張
  - ポリシー識別子などが格納される
- GPKIでは、証明書ポリシ(CP)は存在がMUSTでクリチカル
  - ポリシ制約拡張の設定と合わせて証明書ポリシの処理を強制している。

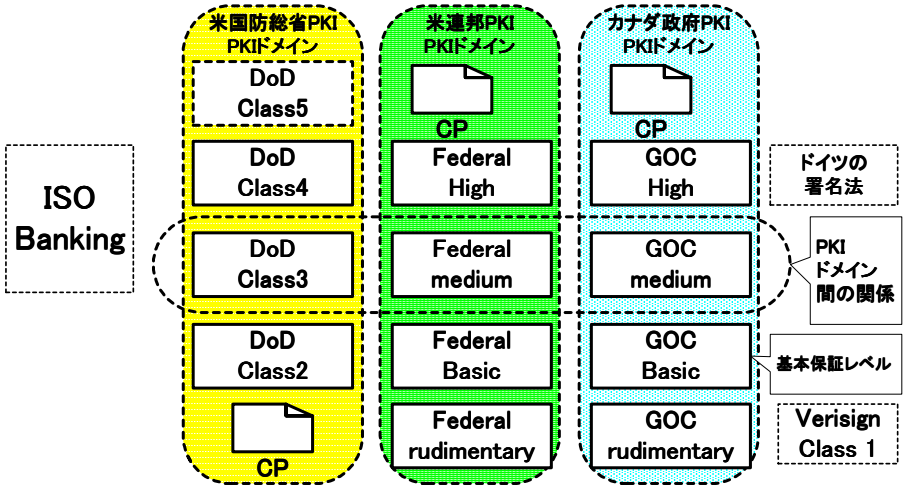
8

証明書ポリシとX.509v3証明書の証明書ポリシ拡張



＊1 実際にはポリシ制約拡張も関係する。

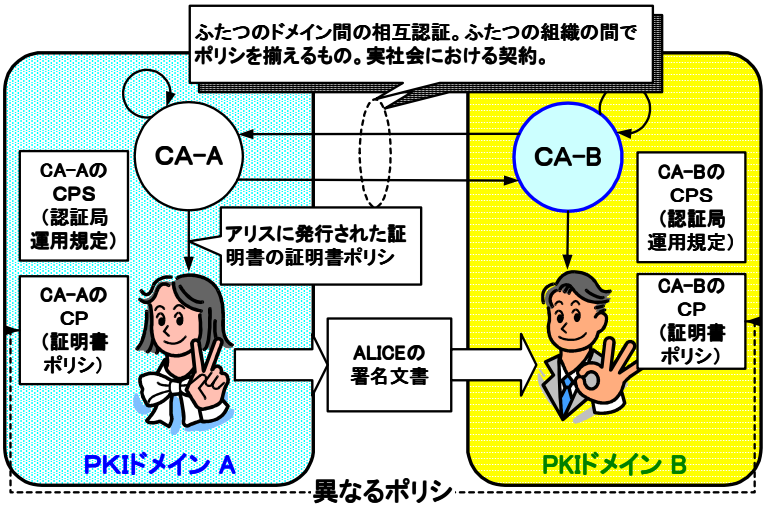
証明書ポリシと保証レベル



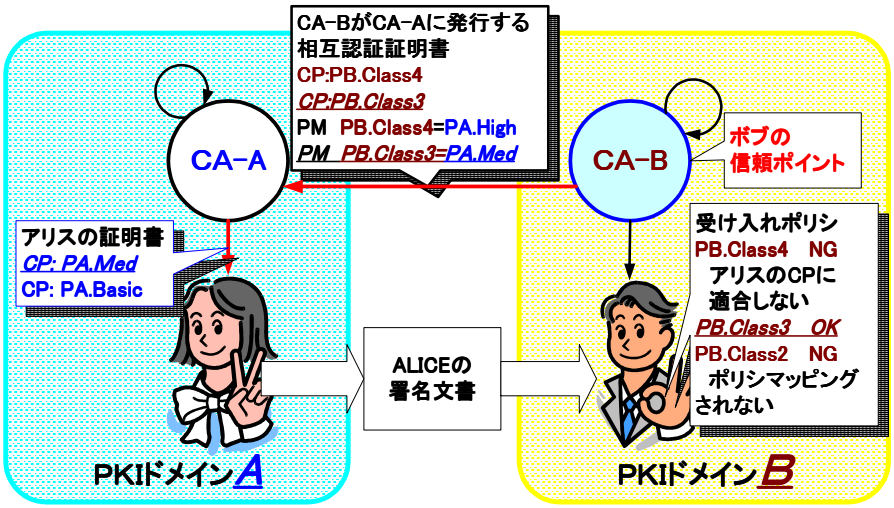
PKIドメインとは??

- PKIドメイン（PKIにおけるドメイン）
  - ひとつのポリシ管理機関などの管理下運用されるPKIの単位。
  - 典型的には、ひとつの組織のPKIなど。
  - 各ドメインは、少なくとも、ひとつのCP、ドメイン・ディレクトリを持つ。
  - ひとつのドメインで、階層型CAや、メッシュ型CAを持つ場合がある。
  - 信頼ドメイン、CAドメイン、ポリシドメインなど、ほぼ同義???
- ポリシドメイン(X.509 4th Editionに記述されている説明)
  - 証明書は、ひとつ以上のポリシに従って発行されるかもしれない。ポリシの定義、及び、識別子の割当は、ポリシ管理機関(PMA)によって行われる。ポリシ管理機関によって管理されたポリシの集合は、ポリシドメインと呼ばれる。

PKIドメイン間の相互認証

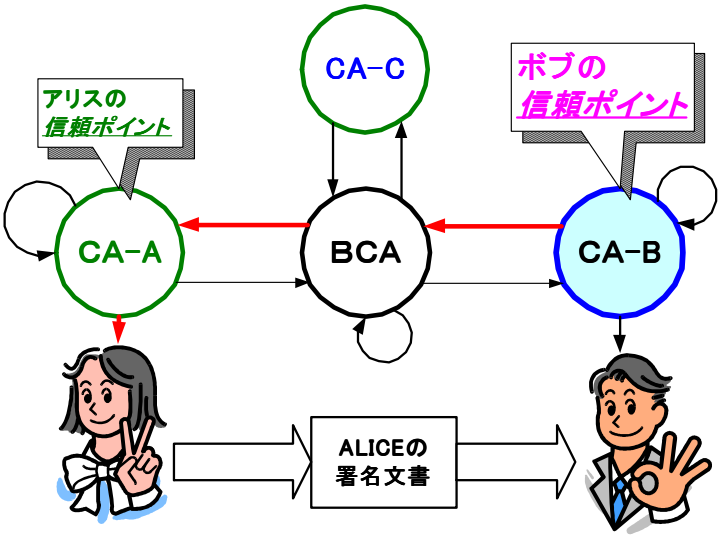


X.509v3証明書のポリシマッピング拡張



13

ブリッジモデル



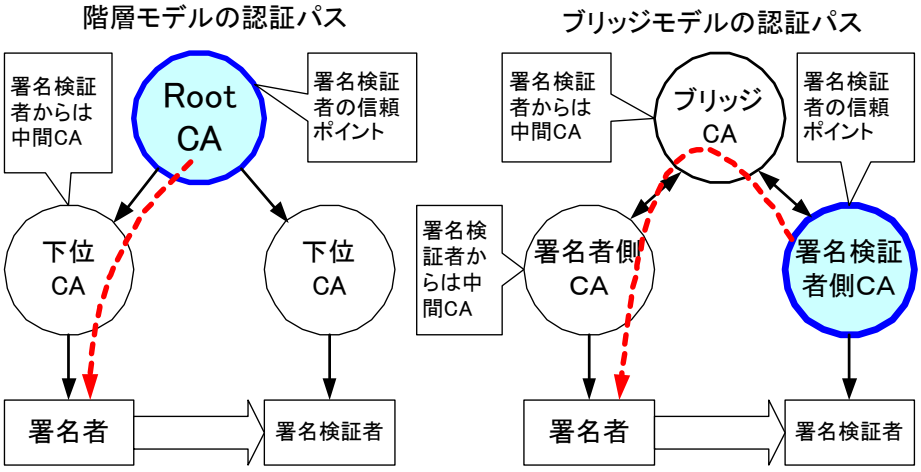
14

ブリッジモデル

- ブリッジCA
  - 各PKIドメインのPrincipal CAは、ブリッジCAと相互認証
- 別名ハブモデル
  - 認証局（PKIドメイン）のハブ
- ブリッジCAは、“信頼の橋”となる
- ブリッジCAは、証明書ユーザの信頼のアンカ（信頼ポイント）ではない
  - 署名検証者から見ると証明書パス中の中間CA
- ブリッジCAは、Root CAではない
- ブリッジCAとビジネスモデル
  - RoorCAから始まるトップダウンなモデルの階層型CAに対して、
  - ボトゥムアップに構築するブリッジモデル
  - 特定業界むけの垂直統合サイトに対応した階層型CA
    - Identrusなどが典型的
  - 水平統合モデルのB2Bには、ブリッジモデル??

15

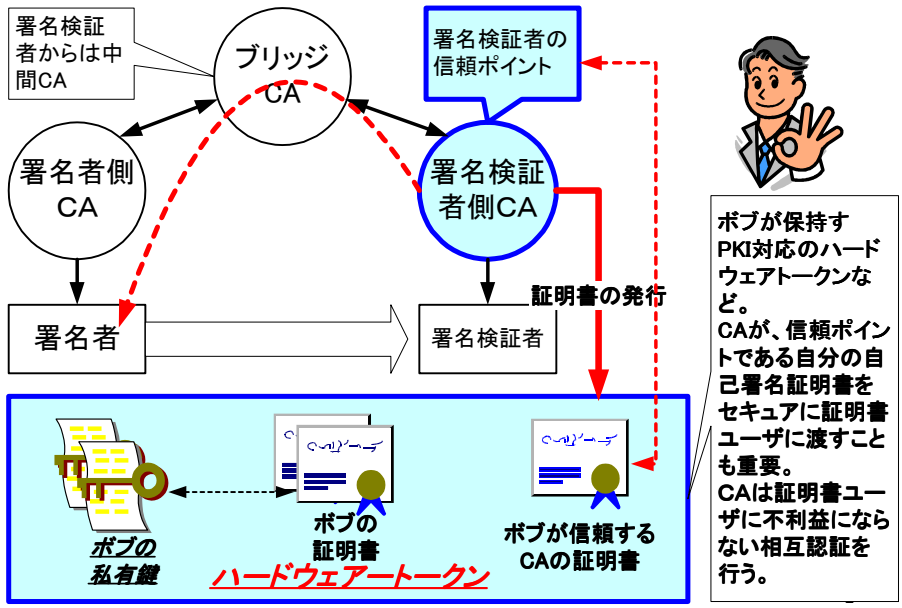
階層モデルとブリッジモデルの認証パス



RFC3280などに記述されている認証パス検証は、信頼モデルに依存しない。しかし、ブリッジモデルでは、その性格から、RFC3280の仕様の多くの部分の実装が要求され、高度なPKI相互運用技術が要求される。

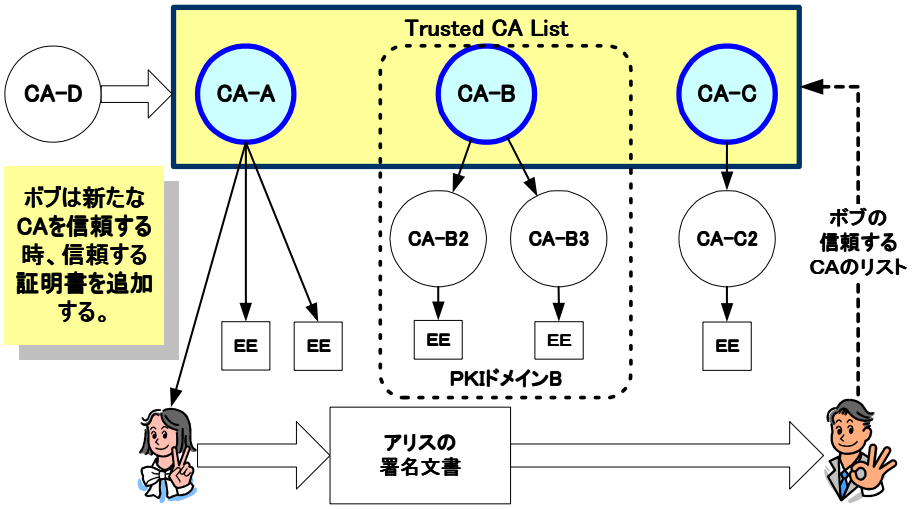
1

ブリッジモデルにおける信頼ポイントの扱い



ボブが保持するPKI対応のハードウェアトークンなど。
CAが、信頼ポイントである自分の自己署名証明書をセキュアに証明書ユーザに渡すことも重要。
CAは証明書ユーザに不利益にならない相互認証を行う。

証明書信頼リストによる方法（Webモデル）



マルチドメインPKIと課題

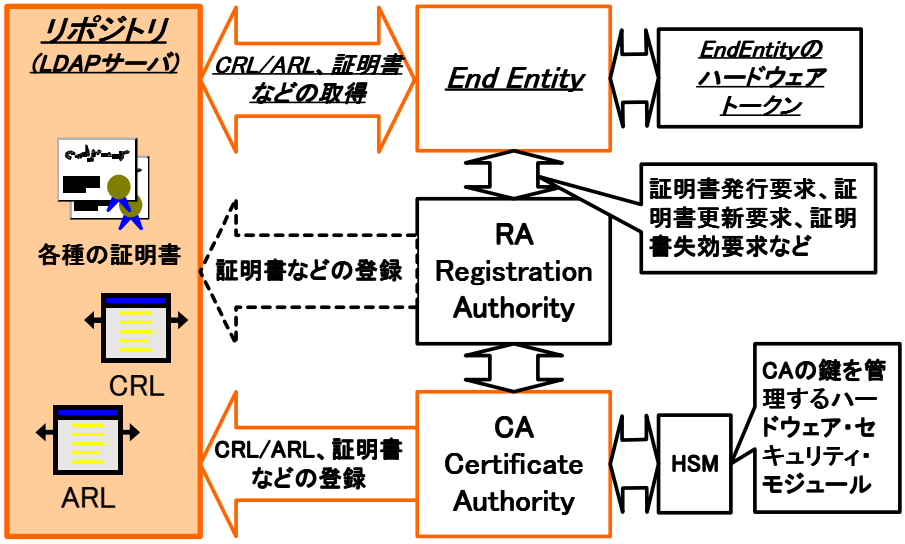
- マルチドメインPKIの要求
  - 色々なドメインの中で、色々なポリシの証明書が発行される中、必要なポリシの証明書の署名を検証できることが必要。
- ブリッジモデルの問題点
  - 署名のチェーンの検証だけでなく、各種の制約拡張の解釈が重要になり、多くのX.509v3証明書拡張の対応のための実装が要求される。また、実装例が少ない。このため、高度な相互運用が要求される
- Webモデル（信頼する認証局リスト）の問題点
  - 信頼リスト自体の信頼を維持するこが難しい
    - 実際には、単なるポリシの異なる認証局のリストでしかない場合が多い
  - 標準化の動きがない（ 下記しか見たことがない ^_^; ）
    - http://csrc.nist.gov/pki/twg/presentations/twg-99-76.pdf
  - ポリシ付き信頼認証局リストの提案などもある
    - ニュージランドの S.E.E. PKI
    - http://www.e-government.govt.nz/docs/see-pki-paper-4/chapter3.html

リポジトリ（LDAP）

- LDAPとは（超安直な説明）
- PKIにおけるLDAP
- PKIで使用するLDAPの標準と実装
- 相互認証とLDAP
- DIT(Directory Information Tree)
- リポジトリに格納されるもの（認証局のエントリ）
- LDAPサーバを使用したネットワーク構成例

5

PKIの基本コンポーネントとリポジトリ

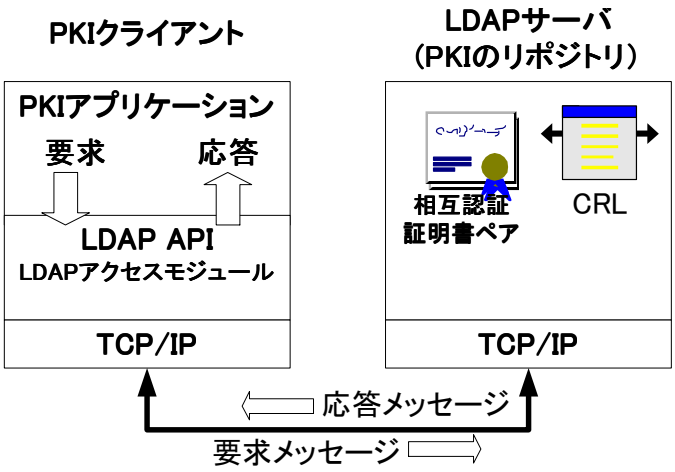


6

LDAPとは（超安直な説明）

- クライアントサーバモデル
  - 例えば、ブラウザ-Webサーバ、DBクライアント/DBサーバ
  - LDAPのモデルは、上記の二つを足して割ったようなもの??
- LDAP（Lightweight Directory Access Protocol）
  - 元になっているX.500の簡易版　#しかしVer3はLightweight??
  - 狭義には、プロトコル（HTTPなどと同じ）
  - 広義には、LDAPプロトコルを使用したクライアントサーバモデル全体のアークテクチャ??
- プロトコル以外も数々の標準が作成されている
  - 多くの標準的なスキーマ
    - PKIが利用するスキーマもある（RFC 2587、X.509）
  - 標準的なLDAPサーバアクセスAPI
  - 標準的なローディング形式（ LDIF　RFC2849 ）
  - #期待されながらも、ちっとも進まない複製プロトコルのLDUP

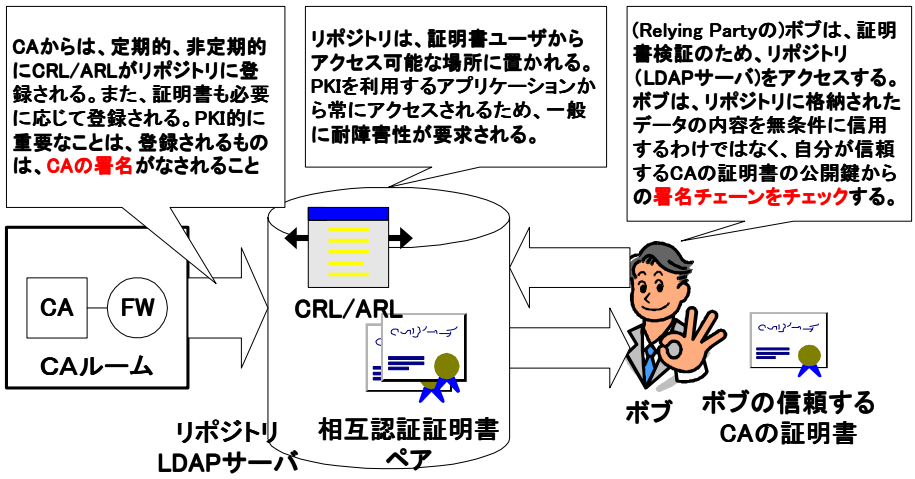
LDAPのクライアントサーバモデル



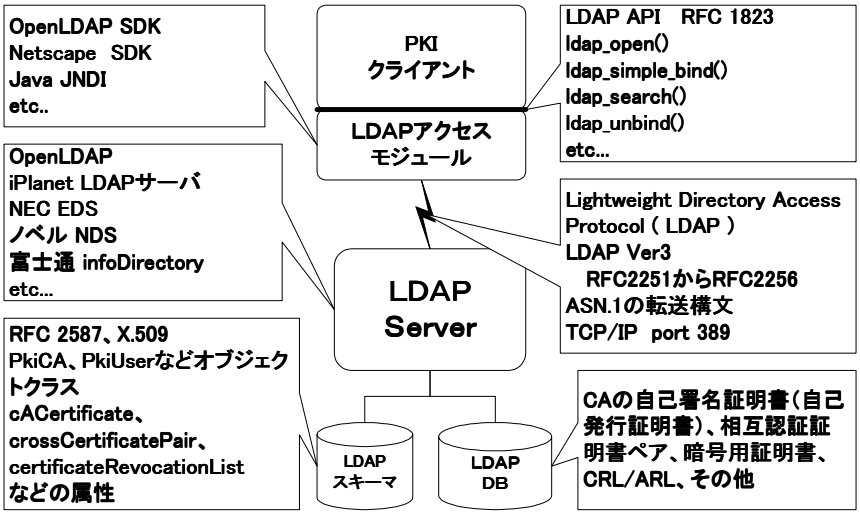
PKIにおけるLDAPの役割

- PKIのリポジトリの役割
  - 主にRelying Party側からアクセス
  - 署名検証
    - 相手の証明書の検証も行う。この証明書の検証が非常に重要。 Relying Party側にとって、証明書パス構築、証明書パス検証が重要
- LDAPサーバへのアクセス
  - 相互認証証明書ペアの取得
    - 証明書パス構築時に行う
  - 証明書失効リスト(CRL/ARL)の取得
    - 証明書パス検証時に行う

PKIにおけるLDAPの役割

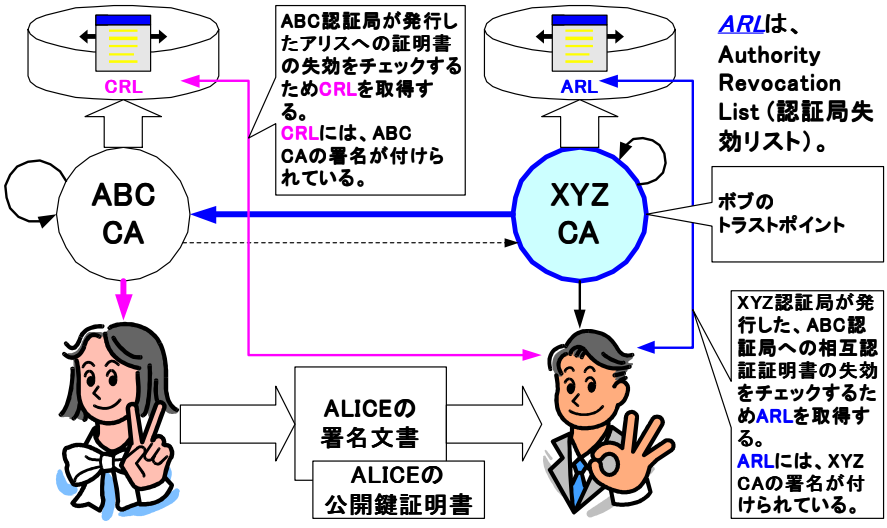


PKIで使用するLDAPの標準と実装



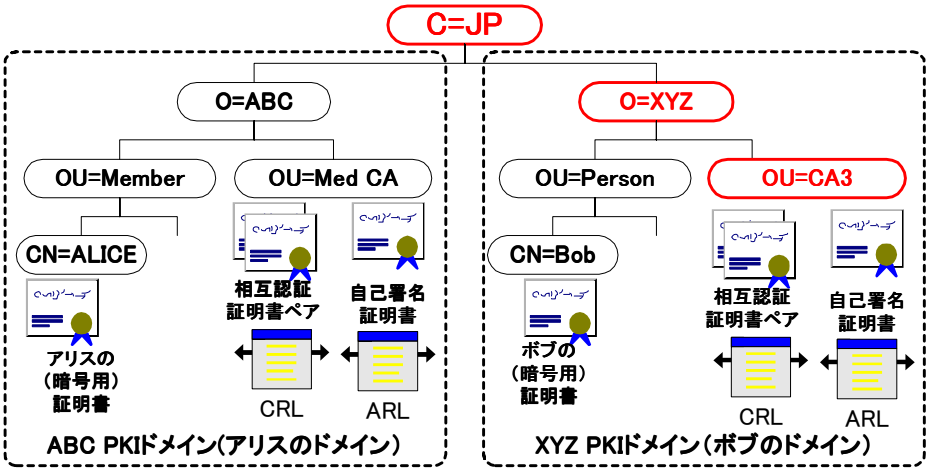
11

相互認証とLDAP



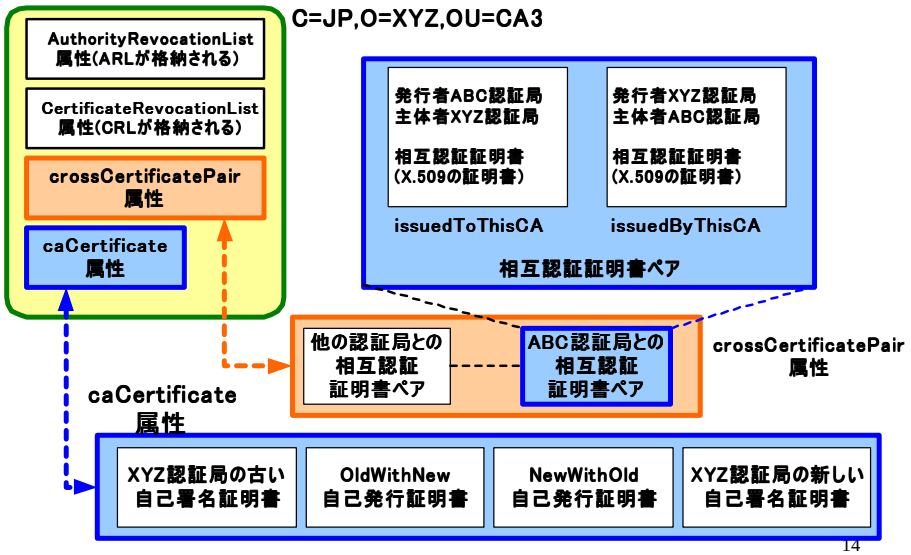
12

DIT(Directory Information Tree)



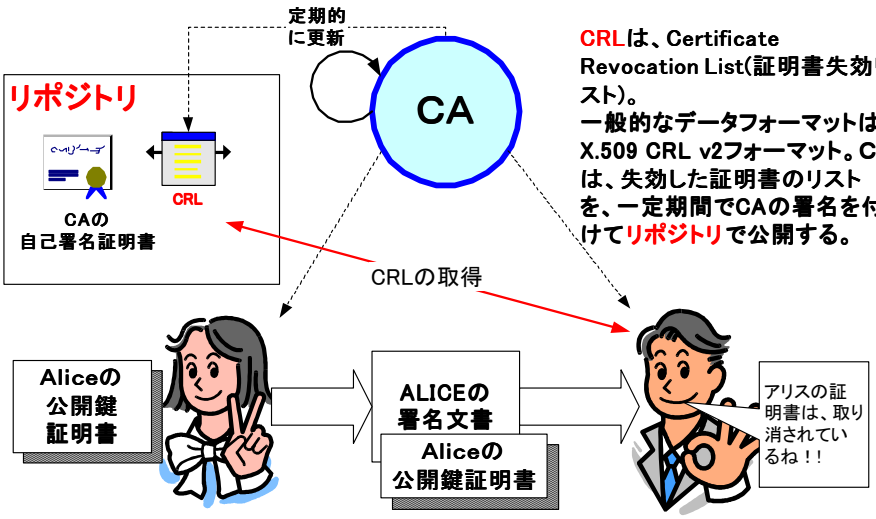
13

リポジトリに格納されるもの（認証局のエントリ）



14

証明書の失効



CRLは、Certificate Revocation List(証明書失効リスト)。
一般的なデータフォーマットは、X.509 CRL v2フォーマット。CAは、失効した証明書のリストを、一定期間でCAの署名を付けてリポジトリで公開する。

15

CRL（証明書失効リスト）

CRLバージョン番号（v2）
デジタル署名アルゴリズム識別子
発行者(CA)の識別名
今回の更新
次回の更新

証明書シリアル番号
失効日時
エントリ拡張(CRLv2の拡張)

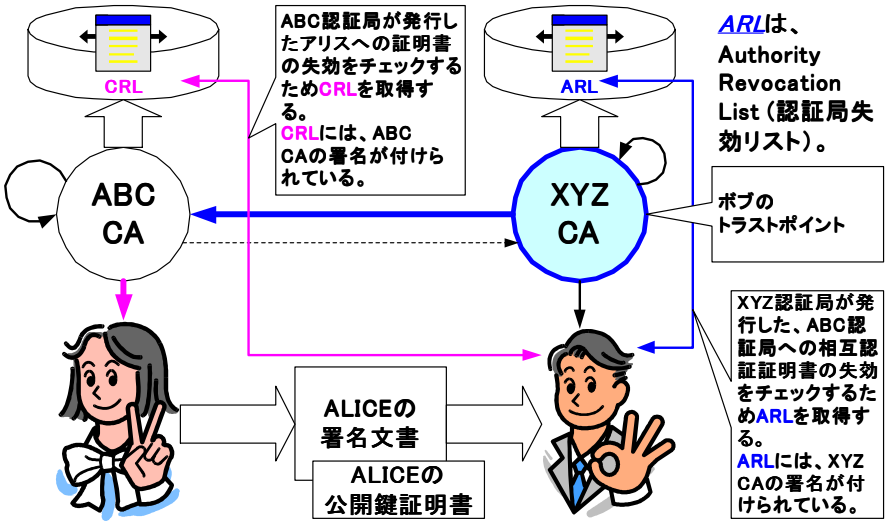
CRLv2の拡張
拡張フィールド(タイプ、フラグ、値)
拡張フィールド(タイプ、フラグ、値)

発行者（CA）のデジタル署名
アルゴリズム識別子
署名

- **CRL**
  - **あるCAが発行した証明書の有効期限内に証明書を失効したい場合、このCRLに、失効したい証明書のシリアル番号を入れて*リポジトリ（LDAPサーバなど）*で公開する。**
  - **CRLは一定期間毎にCAの署名を付けて発行される。**
- **1997年版 X.509 3rd Edition**
  - CRLv2フォーマット
  - X.509v3証明書と同じく拡張がある

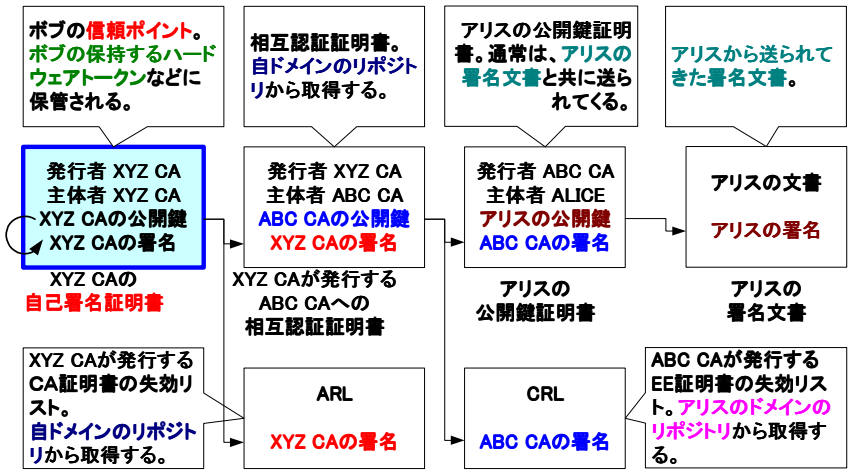
1

相互認証とLDAP



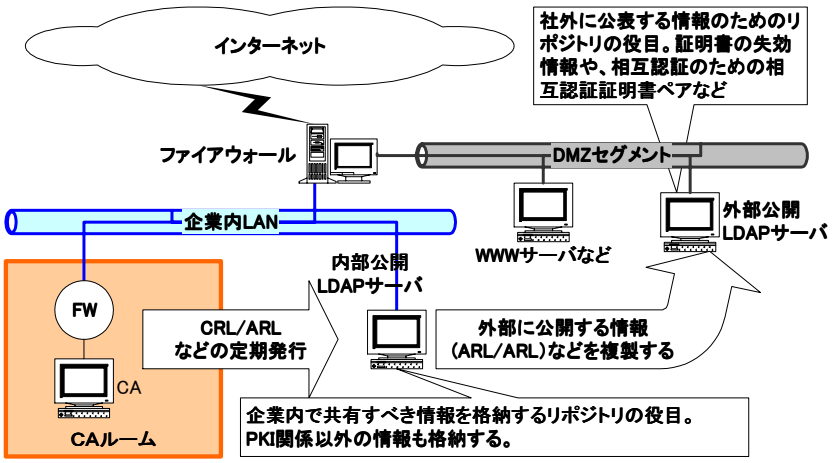
2

アリスの署名文書の検証（ボブが検証）



3

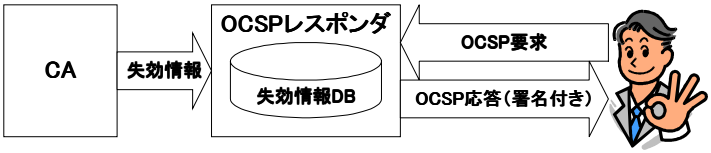
LDAPサーバを使用したネットワーク構成例



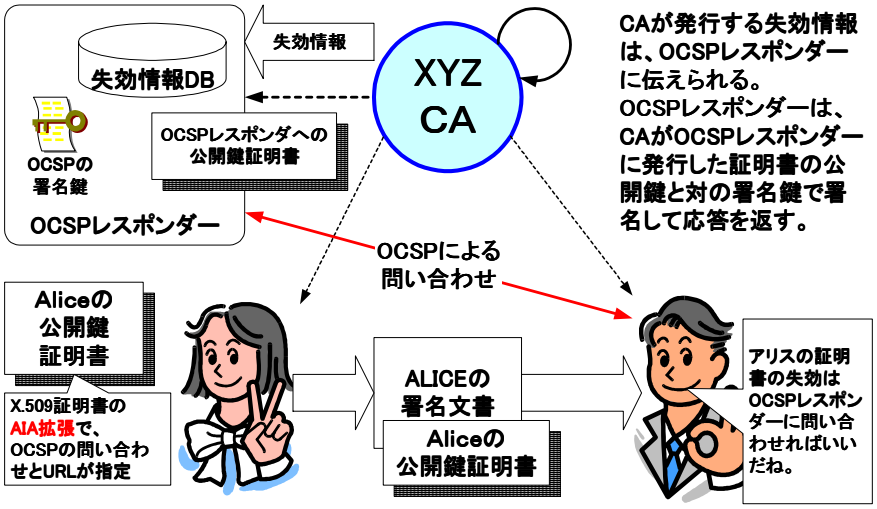
4

OCSP

- RFC-2560
  - Online Certificate Status Protocol – OCSP
  - RFCになったのは、99年6月
- IdentrusでのOCSP
  - Identrusは、OCSPを利用して証明書の失効情報を取得している。
  - Identrusの4コーナモデルでかなり込入った使い方をしている。
- 商業登記CAでのOCSP
  - 商業登記CAは、CAとは別にOCSPレスポンダを用意している訳ではなく、CA自体が、CAの署名鍵で署名してOCSPの応答を返す。

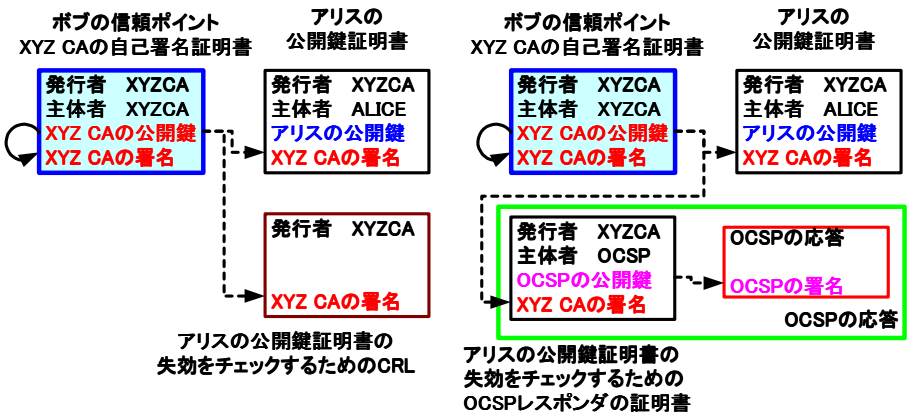


OCSPの仕組み



CAが発行する失効情報は、OCSPレスポンダーに伝えられる。
OCSPレスポンダーは、CAがOCSPレスポンダーに発行した証明書の公開鍵と対の署名鍵で署名して応答を返す。

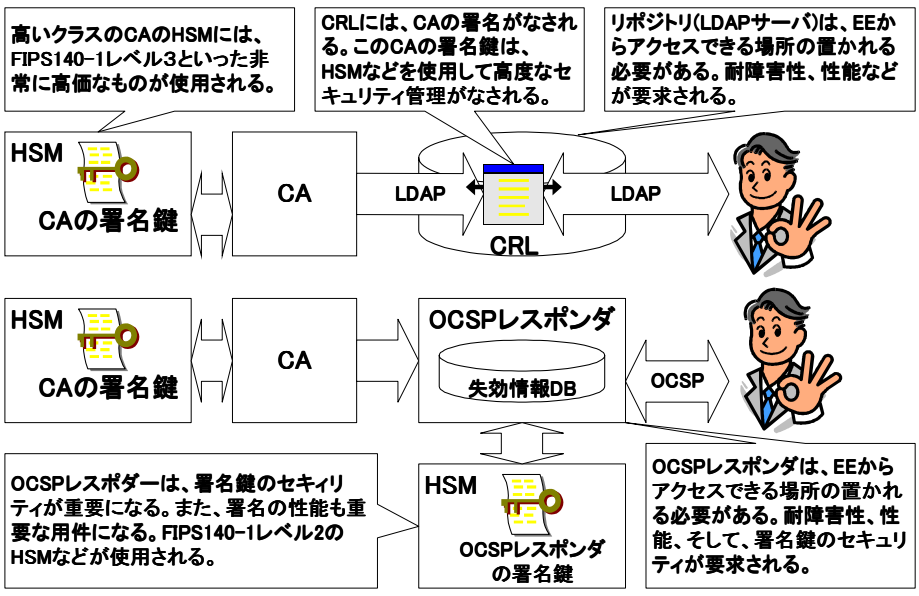
CRLとOCSPの検証の違い



OCSPレスポンダの証明書を失効の問題が面倒。OCSPでの応答にするとでは、OCSP自体の署名になってしまう。

7

LDAP,CRLとOCSPの比較



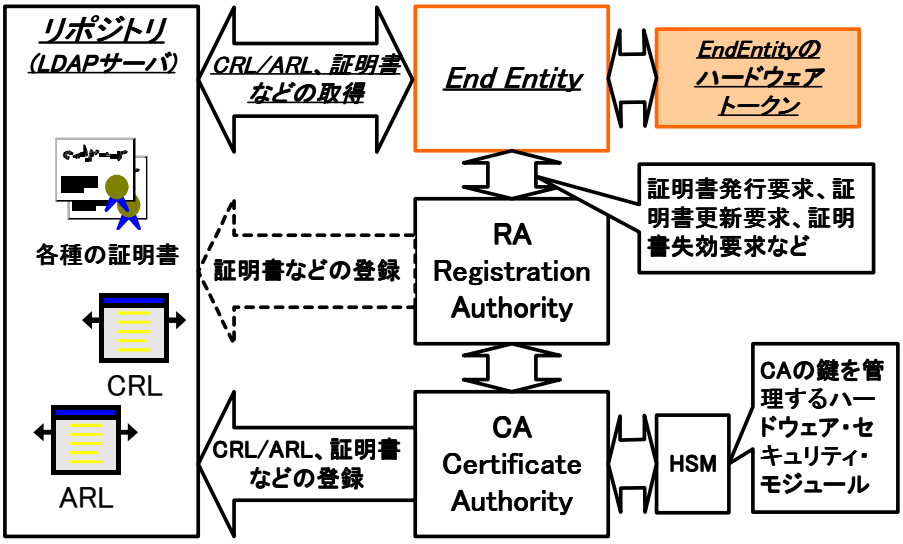
8

ハードウェアトークン（暗号トークン）とは??

- 主体者(アリス)の公開鍵に対応した私有鍵は、主体者(アリス)しか使用できないことが理想。
- そのためには、ハードウェアトークンの使用が有効になる。また、ハードウェアトークンは、色々なPKIアプリケーションで使用できるべき
- ハードウェアトークン
  - 所有者を識別するための暗号クレデンシャル（Cryptographic Credential）を格納することが可能で、かつ携帯性のあるデバイス
- “暗号クレデンシャル(Cryptographic credentials)”ってなに?
  - 鍵と証明書（群）

9

PKIの基本コンポーネントとハードウェアトークン



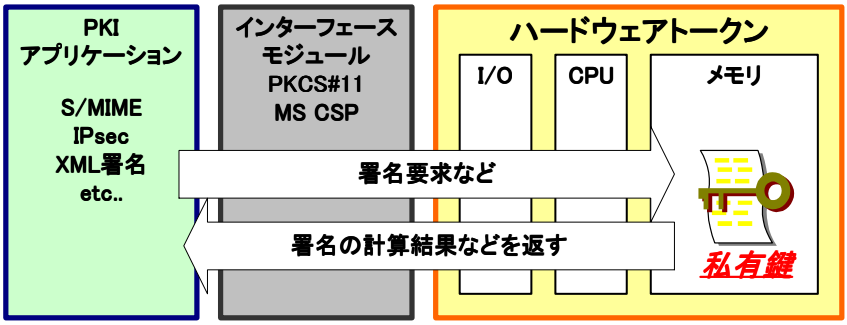
10

ハードウェアトークンの例

- スマートカード(ICカード)
- USB Token （Dongle）
- 生体認証と組み合わせたトークン
  - SonyのPuppy(FIU-710)など
- PCに内蔵されたセキュリティチップ
  - TCPA TPM (Trusted Platform Module)
  - 正確にはハードウェアトークンではないかもしれないが機能的には同じ
- MOPASS（ Mobile Passport ）
  - フラッシュメモリカード用のモバイルコマース拡張規格
  - SDカードなど
  - http://www.mopass.info/

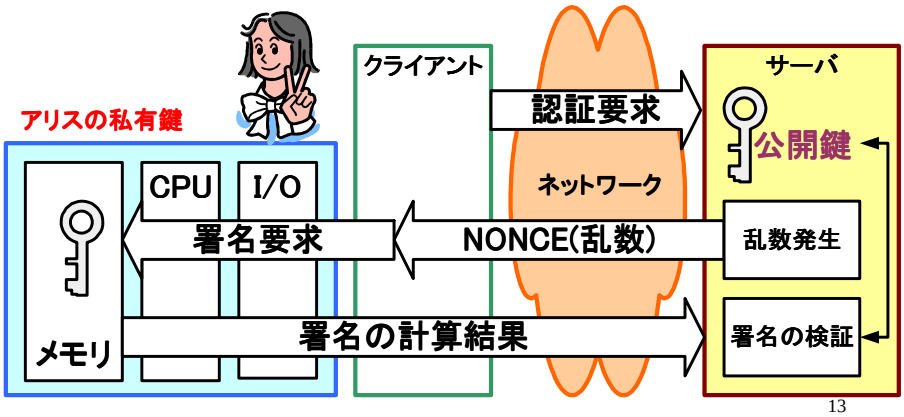
なぜハードウェアトークン

- 署名で使用する私有鍵（Private Key）を守る仕組みが可能
- 私有鍵のコピーを防ぐ。私有鍵がハードウェアトークンから外に出ない
- 私有鍵がハードウェアトークンのOSレベルで保護される
- ハードウェアトークンが盗難にあった場合を想定した耐タンパ性が重要
- PIN (Personal Identification Number)の入力や、指紋照合といった手段でハードウェアトークンにログインする。



ハードウェアトークンを使用した認証の例

- アリスの秘密情報（私有鍵）はハードウェアトークンから出ない
  - もちろんネットワークにも流れない
- アリスの秘密情報は、サーバには、格納されない
  - サーバは、アリスの秘密情報（例えばパスワード）を預かる必要がない
  - これは、アリスとっても、サーバの運用者にとってもメリット



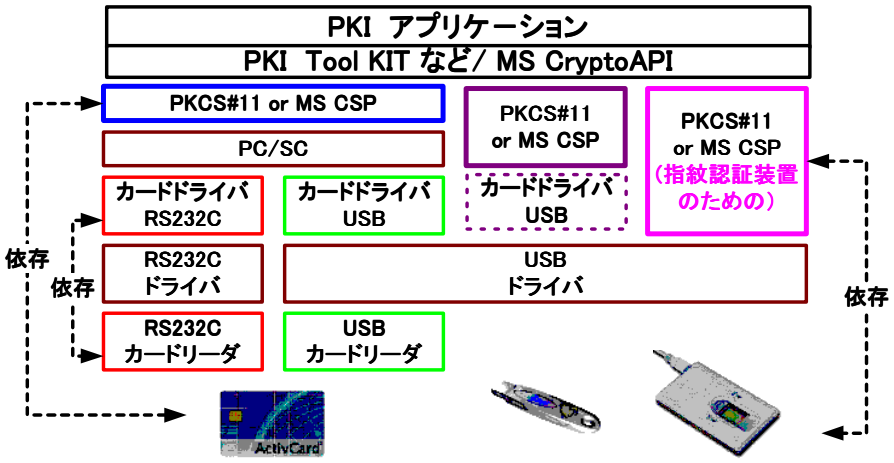
13

ハードウェアトークンとのI/F

- スマートカードリーダとのI/F
  - PC/SC
    - スマートカードリーダを仮想化する
      - USB、RS232C、PCMCIA、etc..
    - 主にWin32環境
- トークンとのAPI（PKCS#11、MS CryptoAPI）
  - PKIアプリケーションとハードウェアトークンとのAPI
  - 私有鍵などのトークンオブジェクトを保護するメカニズムを持っている
  - スマートカード以外のUSB Token などでも同じ

14

ハードウェアトークンとのI/F

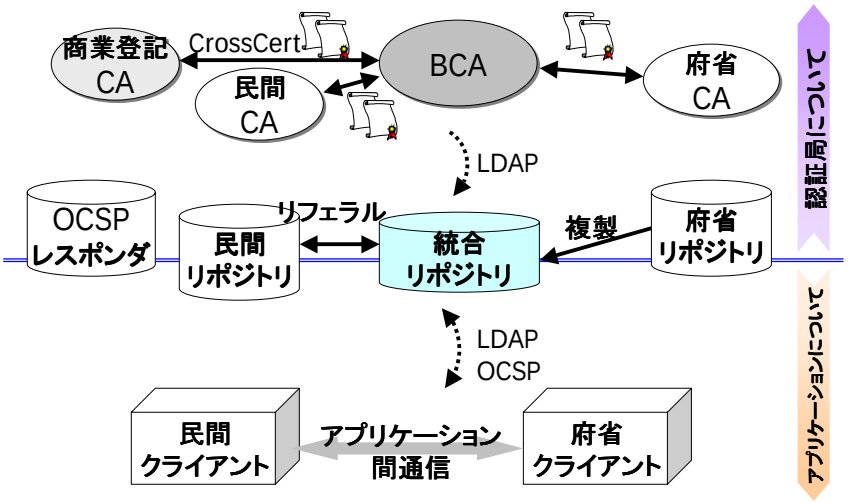


PKIアプリケーションの基本的な要件

- Subscriber側（アリス）の要件
  - セキュアな署名
    - なりすましをいかに防ぐか
    - 署名に使用する*私有鍵をいかに保護*するか??
    - セキュアなハードウェアトークンが有効
- Relying Party側（ボブ）の要件
  - 署名検証、証明書検証をいかに行うか
    - リポジトリ(LDAPサーバ)から必要な情報を取得
      - CRL、ARL、相互認証証明書ペアなど
    - ハードウェアトークン等に格納された*信頼ポイントの公開鍵*からのリポジトリなどから読み出した情報を元に証明書チェーンを構築、そしてパス検証を行う

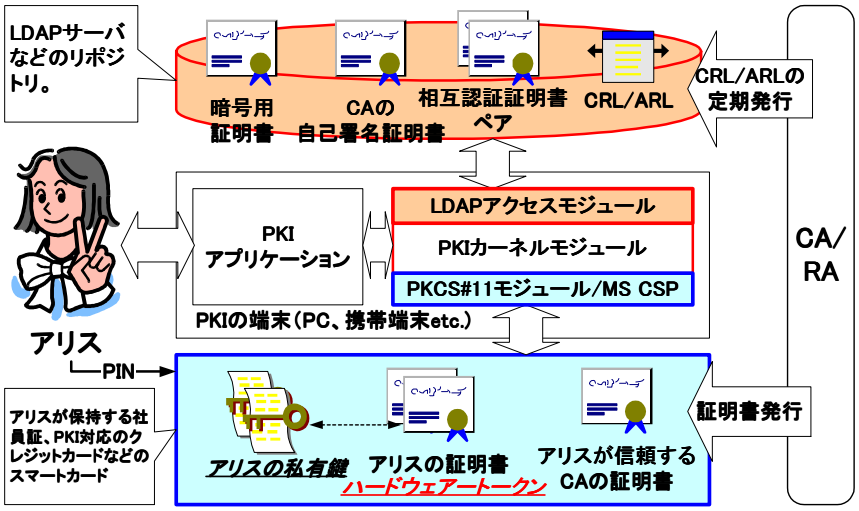
1

GPKIにおける認証局とアプリケーション



2

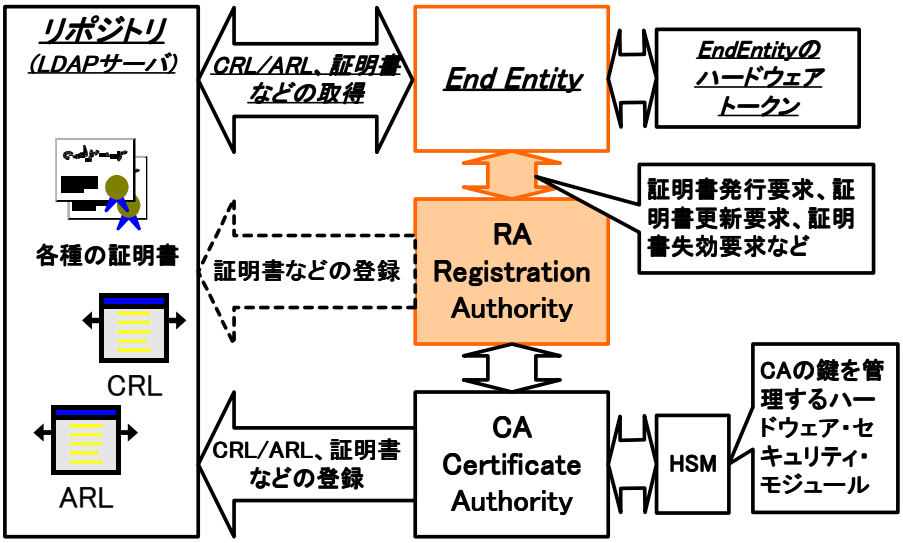
PKIアプリケーション環境の例



証明書発行

証明書発行のメカニズムとPKCS(Public-Key Cryptography Standards)の説明

PKIの基本コンポーネントと証明書発行



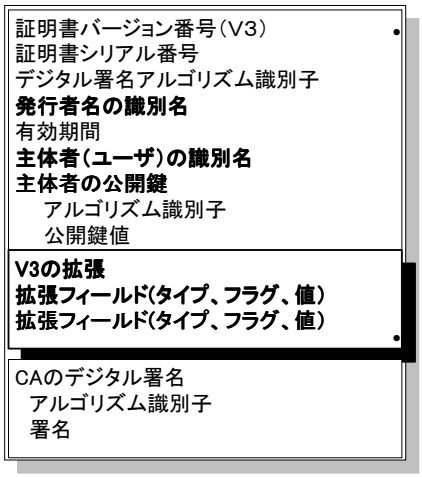
5

証明書の管理

- 鍵ペアの生成
  - EEでの生成とCA/RA側での生成
- 証明書の要求
  - PKCS#10などの証明書要求のフォーマット
- 証明書の配布
  - オンラインでの配布
    - PKIX-CMP, SCEPなど
  - オフラインでの配布
    - FDやSmartCardによる配布
    - PKCS#12, PKCS#7
- その他
  - 証明書の失効、証明書の更新、証明書の再発行とリカバリ

6

X.509証明書



代表的な公開鍵証明書

- 主体者(アリス)と、主体者(アリス)の公開鍵や、その他の属性をCA鍵(アリスの証明書を発行したCAの署名鍵)の署名でバインドする。
- この時、主体者(アリス)の公開鍵に対応した私有鍵は、主体者(アリス)しか使用できないことが理想。

1997年版 X.509 3rd Edition

- X.509v3証明書フォーマット
  - X.509V3拡張
- 14の標準拡張フィールド

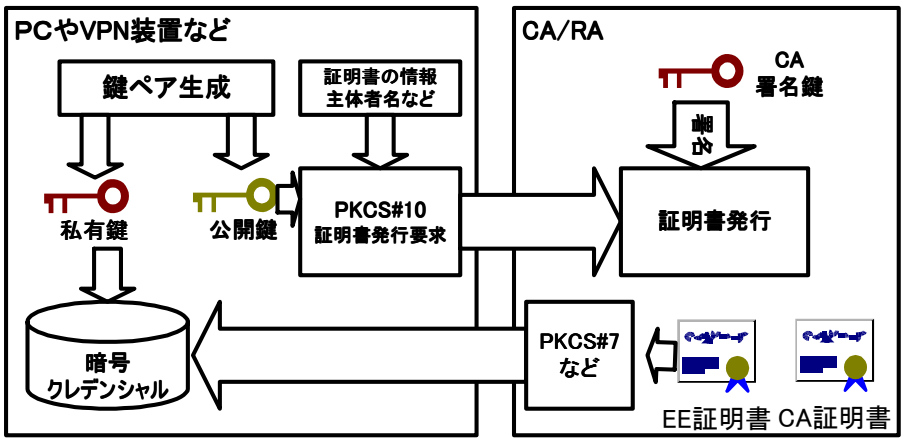
7

鍵ペアの生成と証明書発行

- EE側での鍵ペアの生成
  - SECOMパスポートforMemberなど
  - 鍵とPC上で生成
  - PKCS#10などで証明書要求を生成
  - CAで証明書を発行
  - CAからPKCS#7などで証明書を配布
- CA/RA側での鍵ペアの生成
  - SECOMパスポートfor G-IDなど
  - CAで鍵ペアを生成
  - PKCS#12などで私有鍵、証明書を配布
    - PINなどを別途配布（別経路）

8

証明書発行（EE側の鍵ペア生成）



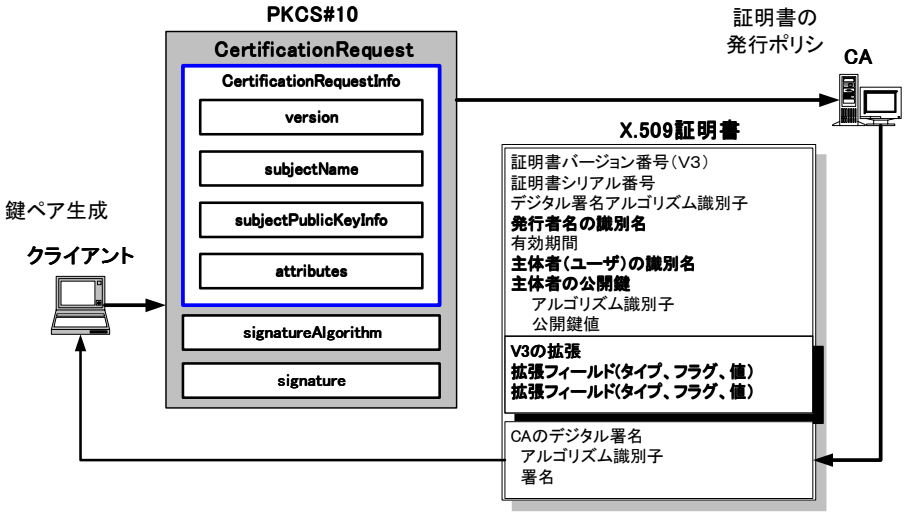
9

PKCS#10(RFC2413)

- 証明書要求のための特別なシンタックス
- PKIの製品で広く使用されている
- バージョン1.0 は、1993年にリリースされた現在までのバージョン
- X.509証明書の変更の適応に十分に柔軟であることが証明された
- RSA ラボラトリは、PKCS #10の新バージョンの計画がある
- 現在のバージョン
  - http://www.rsasecurity.com/rsalabs/pkcs

10

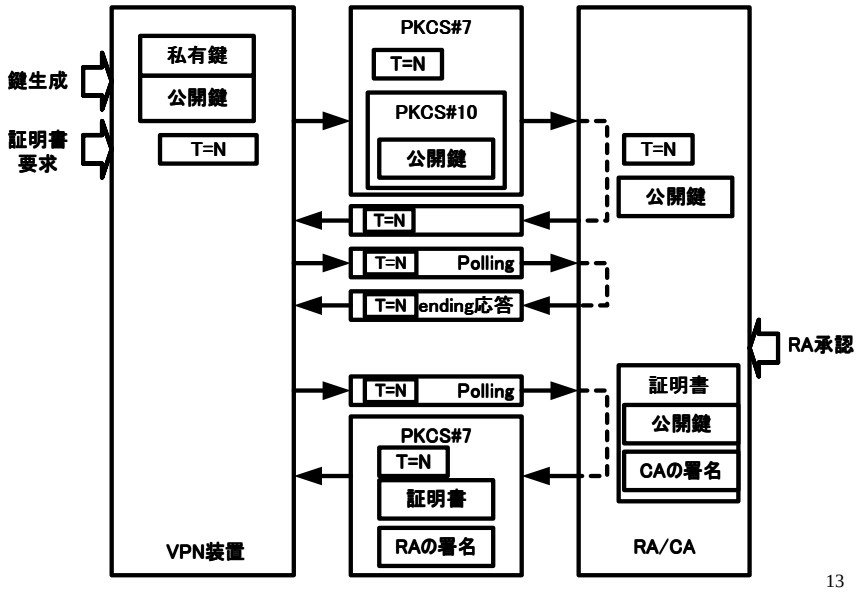
PKCS#10による証明書発行要求



SCEP
(Simple Certificate Enrollment Protocol)

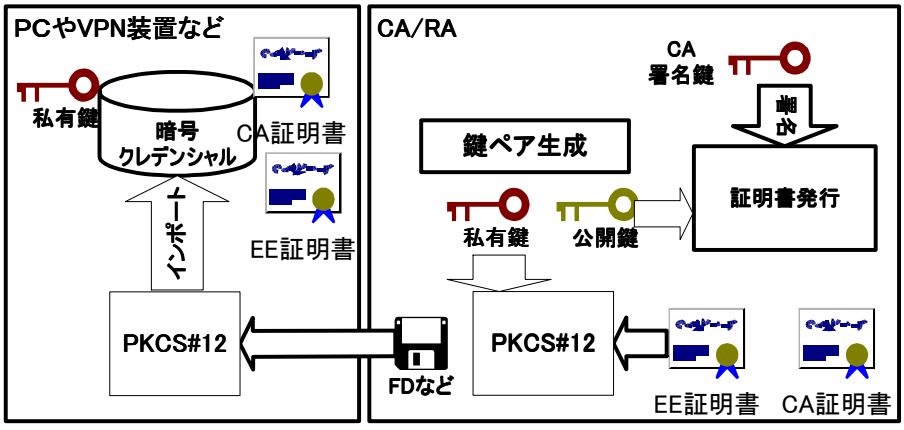
- SCEP
  - CISCOが仕様を作成。VPN装置などへの証明書発行が目的
- HTTPベース
  - SCEPデバイス側がHTTPのクライアトとして動作する
- メッセージに共通のデータ
  - メッセージには、トランザクションを一意に管理するための TransactionIDと、Dos攻撃を防ぐためのConsが含まれる
- PKCS#10,PKCS#7などを使用した証明書要求
  - 通常の証明書要求であるPKCS#10をPKCS#7で暗号化して、他のメッセージも合わせて証明書要求メッセージを作っている
- 生成した証明書
  - PKCS#7でRAの署名がついてSCEPデバイスに返される

SCEP



13

証明書発行(CA側の鍵ペア生成)



14

PKCS#12

- PKCS#12
  - 秘密鍵や公開鍵証明書のバックアップや他のマシンへの移行するために適した規格
  - RSA暗号のみサポート
  - インポート/エクスポート
- サポートされるアプリケーション
  - NetscapeCommunicator 4.04 以降
  - InternetExplorer 4.0 以降

PKCS
(Public-Key Cryptography Standards)

- PKCS
  - PKCSは、RSA Data Security Inc.が定めた規格
- PKCS #1:RSA Encryption Standard
  - RSA暗号を使った暗号化方法および署名方法。
- PKCS #7:Cryptographic Message Syntax Standard
  - 暗号化や署名を施したデータの構文。S/MIMEなどで使用されている。
- PKCS #10:Certification Request Syntax Standard
  - CA(Certificate Authority)に対する証明書発行要求メッセージの構文。
- PKCS #11:Cryptographic Token Interface Standard
  - ハードウェアトークンをアクセスするためのAPI
- PKCS #12:Personal Information Exchange Syntax Standard
  - 暗号クレデンシャルの交換を行うためのフォーマット

PKI 基礎編

END