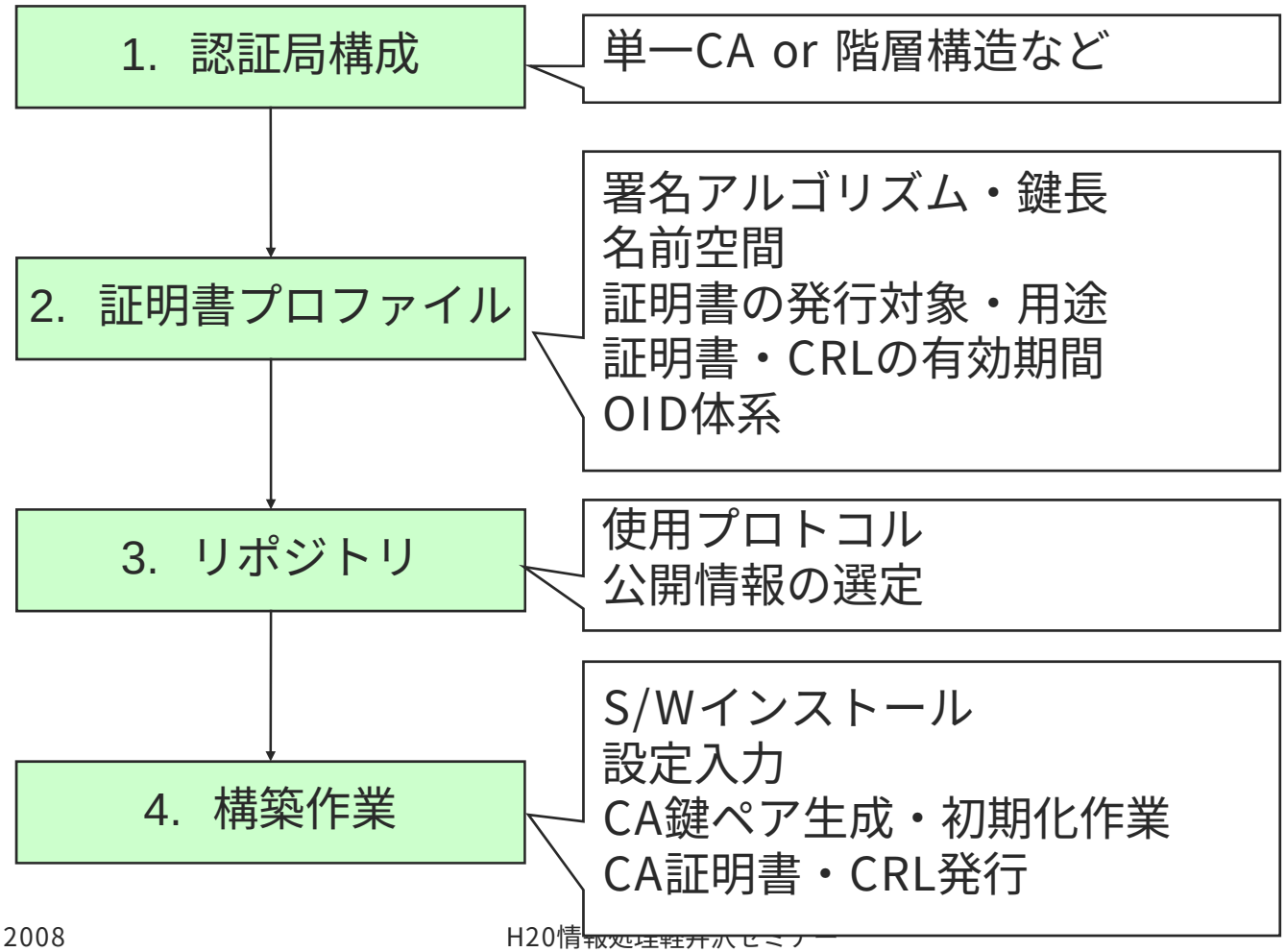
認証局を実際に構築するには

国立情報学研究所
学術ネットワーク研究開発センター
島岡　政基

認証局構築の流れ


1. 認証局構成
単一CA or 階層構造など
2. 証明書プロファイル
署名アルゴリズム・鍵長
名前空間
証明書の発行対象・用途
証明書・CRLの有効期間
OID体系
3. リポジトリ
使用プロトコル
公開情報の選定
4. 構築作業
S/Wインストール
設定入力
CA鍵ペア生成・初期化作業
CA証明書・CRL発行

認証局構成の決定

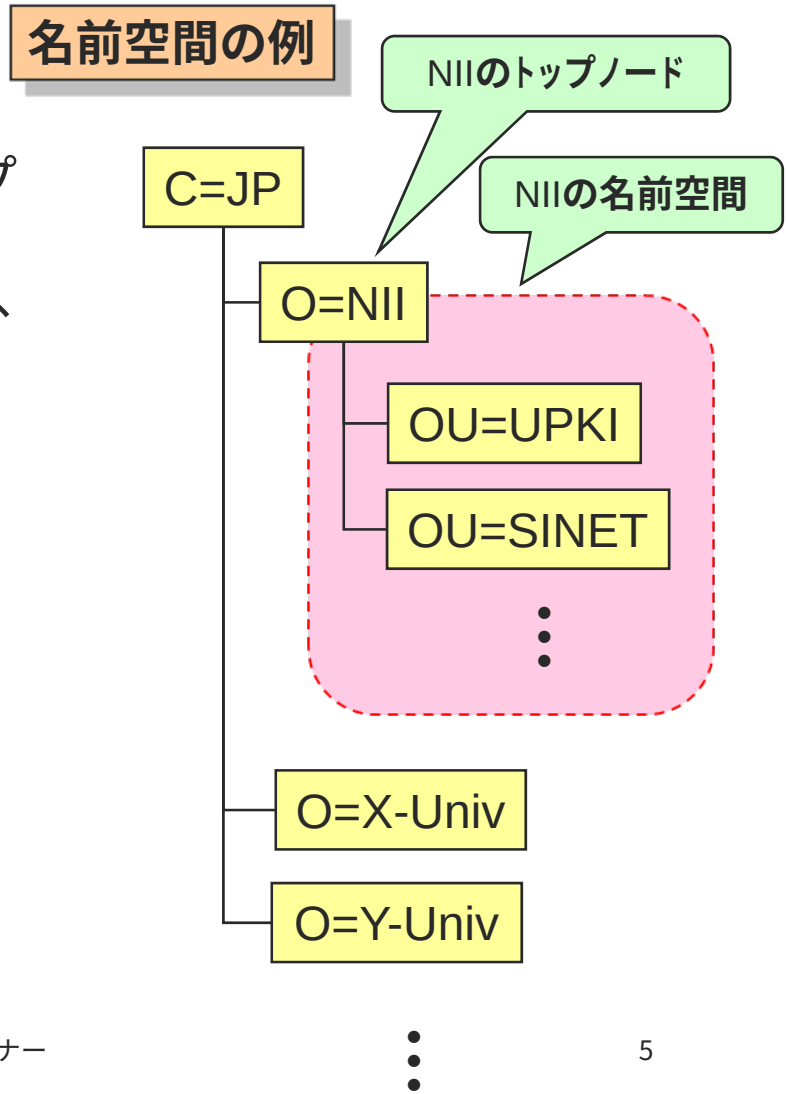
- 単一CA
  - 評価用途なら単一CAでも十分
  - 長期運用、大規模展開を考慮しなくてよければ単一CAでOK
    - ただし本格運用すれば長期運用は必至
- 階層構造
  - 評価: 多段認証局構造などの検証
  - 本格運用では、認証局の鍵ペア更新や拡張性などを考慮して階層構造をとることが望ましい
    - ルート認証局はあくまでトラストアンカ
    - 実際の証明書発行は下位認証局から行う
    - 単一認証局だと、都度トラストアンカが増えることに

署名アルゴリズム・鍵長

- 署名アルゴリズム
  - いくつか選択肢はあるものの、、、
    - SHA1/RSA、SHA256/RSA、etc.
    - SHA1/DSA、SHA1/ECDSA
  - 相互運用性などを考慮すると、実質的にはSHA1/RSA(またはSHA256/RSA)
- 鍵長
  - RSA1024bitまたは同2048bit
    - CAは基本的に2048bitとすべき
    - 可能であればEE証明書も2048bitが好ましい
    - 無闇に長くしてもあまり効果なし

名前空間

- ベースDN
  - 認証局が扱う名前空間のトップノードを決める
  - 認証局が発行する主体者DNは、基本的にベースDNの下に収まるようにする
- 主体者DN命名規則
  - 認証局は、主体者DNの一意性を保証する必要がある
    - 例1: CNは主体者の教職員IDとする。
    - 例2: CNは主体者ローマ字表記とし、同一表記が存在する場合、serialNumber属性に2桁のsuffixを記載する。



証明書の発行対象

- 自然人
  - 職権などの可変属性を証明書に記載しない
  - 静的な属性のみを記載する
- 職権
  - 職権に対して証明書を発行する
  - 例: 課長以上の職級や窓口業務など
- サーバ
  - WebサーバやVPNなど、IPアドレスまたはFQDNが固定のもの
  - センターなど一定の物理セキュリティが確保された環境で運用されていることが暗黙の前提
- (クライアント)機器
  - パブリックスペースなど一定のセキュリティを確保できない環境に設置され無人運用される機器
  - IPアドレスやFQDNが固定でなかったり、特定の管理者がいないなどの特徴が挙げられる
- その他

証明書の用途(1)

keyUsage	用途
digitalSignature(DS)	署名や認証 SSL/TLS、PKCS#7、XMLSignatureなど
nonRepudiation(NR)	否認防止を伴う署名 主体者以外が私有鍵のバックアップを持っている場合、これを指定してはいけない
keyEncipherment(KE)	鍵の暗号化 SSL/TLSやS/MIME暗号では、データを暗号化する共通鍵を公開鍵で暗号化する仕組み。
dataEncipherment	実質的に使用せず
keyAgreement(KA)	鍵合意 DHなどに用いる
ketCertSign	証明書への署名(認証局のみ)
CRLSign	CRL発行
encipherOnly	実質的に使用せず
decipherOnly	実質的に使用せず

証明書の用途(2)

extendedKeyUsage	用途	keyUsage要件
serverAuth	TLSサーバ認証	DS&KE or DS&KA
clientAuth	TLSクライアント認証	DS and/or KA
codeSigning	コード署名	DS
emailProtection	S/MIME	DS, NR and/or (KE or KA)
timeStamping	タイムスタンプ(TSAのみ)	DS and/or NR
OCSPSigning	OCSP署名(OCSPレスポンダのみ)	DS and/or NR

- keyUsageよりもextendedKeyUsageを先に決めるとわかりやすい
- 発行対象によって用途も異なる場合が多い

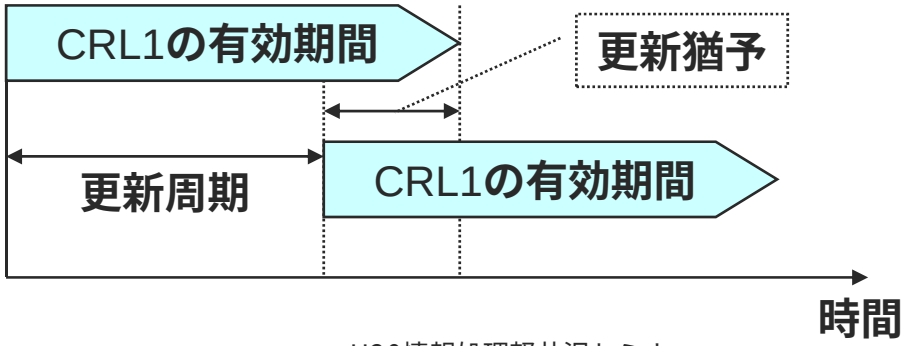
証明書の有効期間

- CA証明書とEE証明書に分けて考える
  - CAは鍵ペア管理が厳密であることが前提
- CA証明書
  - RSA2048bitなら～20年が目安
  - RSA2048bitの解読時間に依存
- EE証明書
  - 鍵ペア管理が厳密ではないので解読時間にはあまり依存しない
    - 鍵ペア管理の強度に依存する話だが、適切な評価軸がないので、あくまで目安レベルで決めているのが実態か
  - サーバ証明書など一定のセキュリティが期待できればある程度長め(それでもせいぜい2～3年程度)でも大丈夫と考えられる
  - SWトークンで管理するなど紛失・漏洩の可能性が高い状況では短めに設定する
  - エンドユーザでもICカードなどHWトークンなら3～5年程度としている例もある

10億円の解読コストをかけると…
・RSA1024bitは2022年には6年で、
・RSA2048bitは2038年には20年で
それぞれ解読可能と言われている

CRLの有効期間

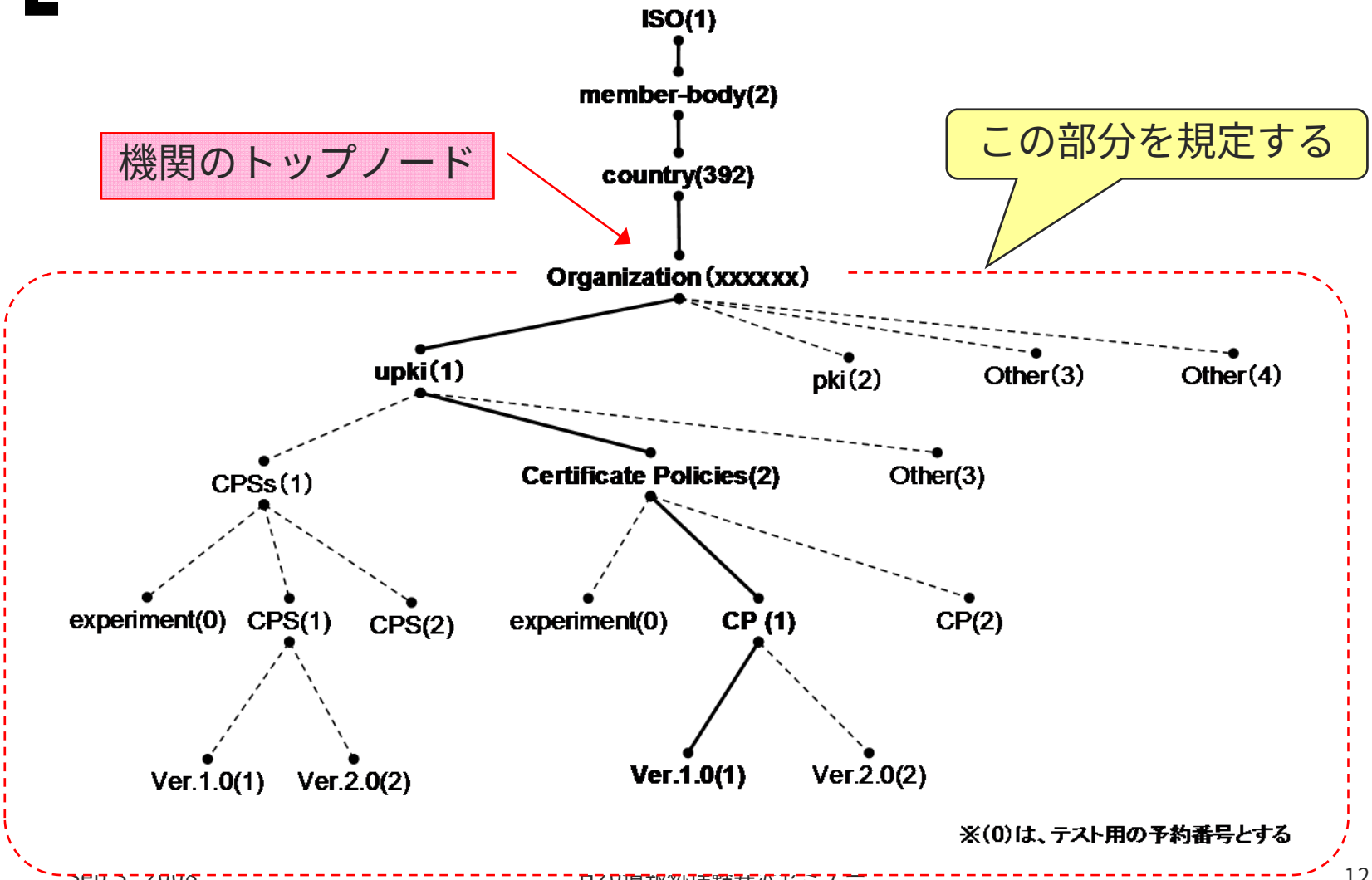
- 基本的には有効期間が短い方が好ましい
  - GPKIなどは48時間
- ただし障害時の復旧時間も短くなるため、更新猶予も含めて検討すべき
  - 例：有効期間7日間、更新周期3日
    - 4日間の更新猶予(ダウンタイム)を確保できる



OID体系

- OID: Object Identifier
  - CP/CPSなどの識別子として用いる。
    - 認証業務に限らず、ネットワーク管理などにも利用可能。
    - どちらかというとSNMPのMIB管理に用いるものとして知られている。
  - 下記から機関単位で割り当ててもらう
    - 総務省: 日本語可、有償、2〜4週間程度
    - 次世代電子商取引推進協議会(ECom): 日本語、有償、2〜4週間程度
    - IANA: 英語のみ、無償、一カ月以上?
- OID体系
  - 機関内部での管理体系を規定する。

OID体系例(1)


ISO(1)
member-body(2)
country(392)
機関のトップノード
この部分を規定する
Organization(xxxxxx)
upki(1)
pki(2)
Other(3)
Other(4)
CPSs(1)
Certificate Policies(2)
Other(3)
experiment(0)
CPS(1)
CPS(2)
experiment(0)
CP(1)
CP(2)
Ver.1.0(1)
Ver.2.0(2)
Ver.1.0(1)
Ver.2.0(2)
※(0)は、テスト用の予約番号とする

OID体系例(2)

OID	オブジェクト
1.2.392.xxxxxx	Organization（NII学術NW R&Dセンター）
1.2.392.xxxxxx.1	upki 関連で利用
1.2.392.xxxxxx.1.1	CPS
1.2.392.xxxxxx.1.1.x	各認証局のCPS
1.2.392.xxxxxx.1.1.x.x	各 CPSのバージョン
1.2.392.xxxxxx.1.2 （※2以降）	各 CP （※ CPは2以降、1はCPSで利用）
1.2.392.xxxxxx.1.2.x （※ 2以降）	CP毎に設置（用途、本人性確認レベル等）
1.2.392.xxxxxx.1.2.x.x （※ 2以降）	各CPのバージョン
1.2.392.xxxxxx.1.3	3以降はその他用途(未定) (独自拡張鍵用途の定義など)
1.2.392.xxxxxx.2	所内PKI関連で利用(未定)
1.2.392.xxxxxx.3	3以降はその他用途(未定)

リポジトリ

- プロトコル
  - 基本的にはWebサーバ(HTTPS)で十分
  - EE証明書も公開する場合などにはLDAPサーバ
- 公開する情報
  - トラストアンカ情報(オプション)
    - CA証明書、同フィンガープリント
  - CRL←証明書プロファイル:CRL配布点
  - CP/CPS等規定類
  - 各URLは認証局運用中および場合によっては運用終了後も一定期間保持する必要がある
- その他
  - 各URLへのアクセス制御
  - LDAPサーバの場合ディレクトリスキーマの設計が必要
    - DIT、オブジェクトクラス、属性