厚生労働省 御中

中間CAにおける鍵更新対応の考え方

2024年6月11日

新井　聡

ご相談事項の解決に向けたアプローチ

ご相談事項

- HPKIのような中間CAを介したEEへの証明書発行という運用を行っているケースにおいて、使用期限満了に伴う「中間CAの鍵更新」は、どのように実施されるのが望ましい手法なのか
- 上記のような鍵更新を実施するにあたり注意点や運用側で定めておかなければいけないルールや条件は何か

基本的な考え方

中間認証局の鍵が更新された場合でも、新旧双方の認証局で発行された利用者電子証明書に対して、すべて検証可能であることが重要となる。

＜今回の議論のポイントと考えられる点＞

- 利用者証明書の検証には、信頼できるルート証明書(トラストアンカー)の設定が必要となる。
- 信頼できるルート証明書(トラストアンカー)の設定方式は、主流が２つあり、どちらかで運用方針が決定される。
- 信頼できるルート証明書(トラストアンカー)の設定方式の決定に関する検討指標の比較が必要である。

＜勘違いが懸念される点＞

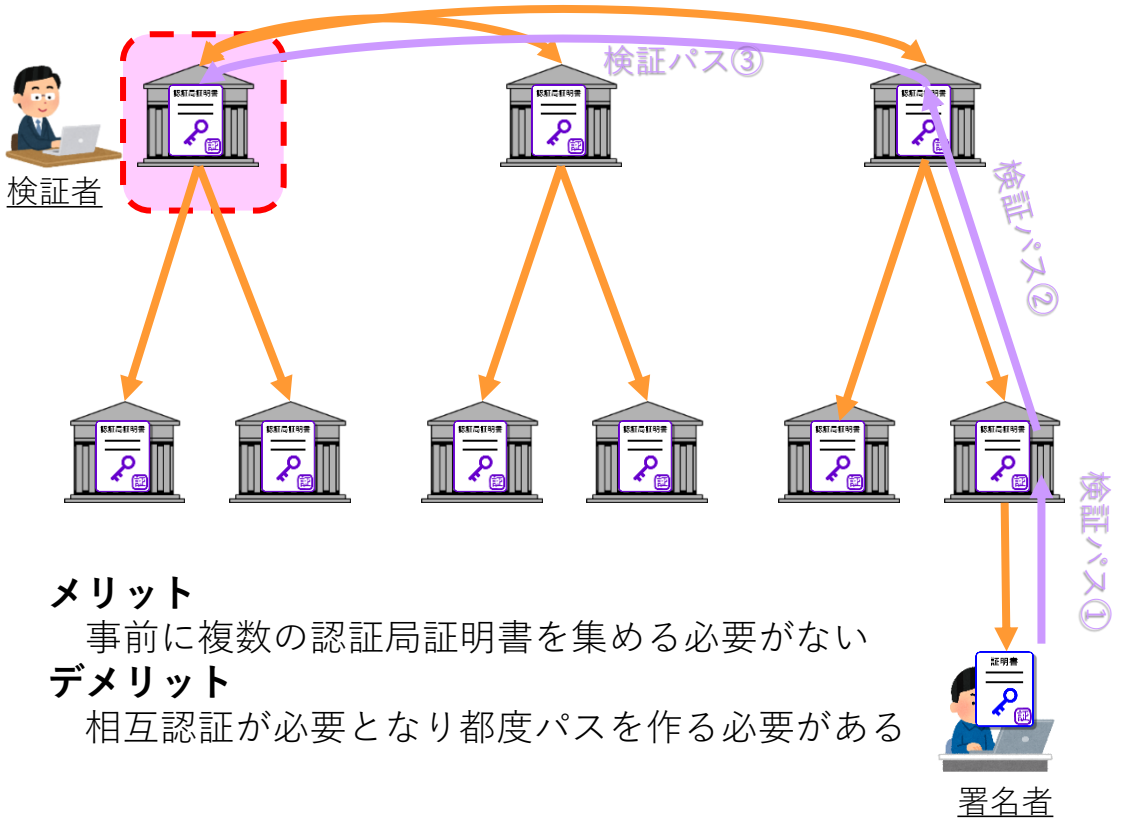
- 証明書失効リスト(CRL)は、認証局から発行されるとは限らないため、これを考慮した規約が存在する。
- モノ(Entity)の識別は、X.500シリーズが定めるDirectory Nameで示され、認証局自体の名前もこちらで決定される。

トラストアンカーの方式

- PKI電子証明書におけるトラストアンカーとは、検証時に**信頼の基点**となる自己署名証明書を持つ認証局を指す。
- 一つの認証局を信じるシングルトラスト方式と複数の認証局を信じるマルチトラスト方式がある。
- シングルトラストは**相互認証**にて、マルチトラストは**トラストリスト**として、他認証局発行の証明書とのパスを作る。

シングルトラスト方式

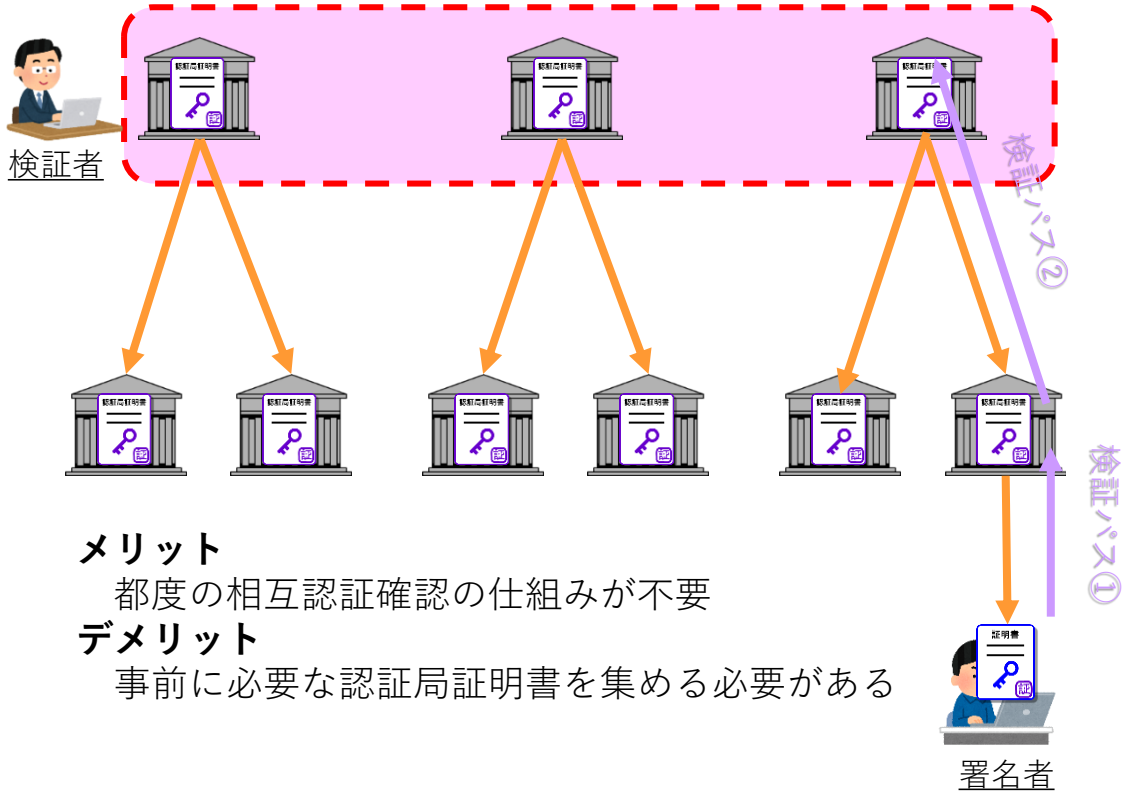
検証者が、一つの認証局を信頼する方式



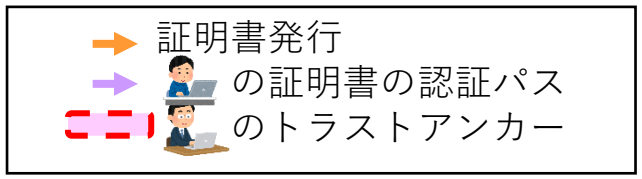
メリット
事前に複数の認証局証明書を集める必要がない
デメリット
相互認証が必要となり都度パスを作る必要がある

マルチトラスト方式

検証者が、複数の認証局を信頼する方式



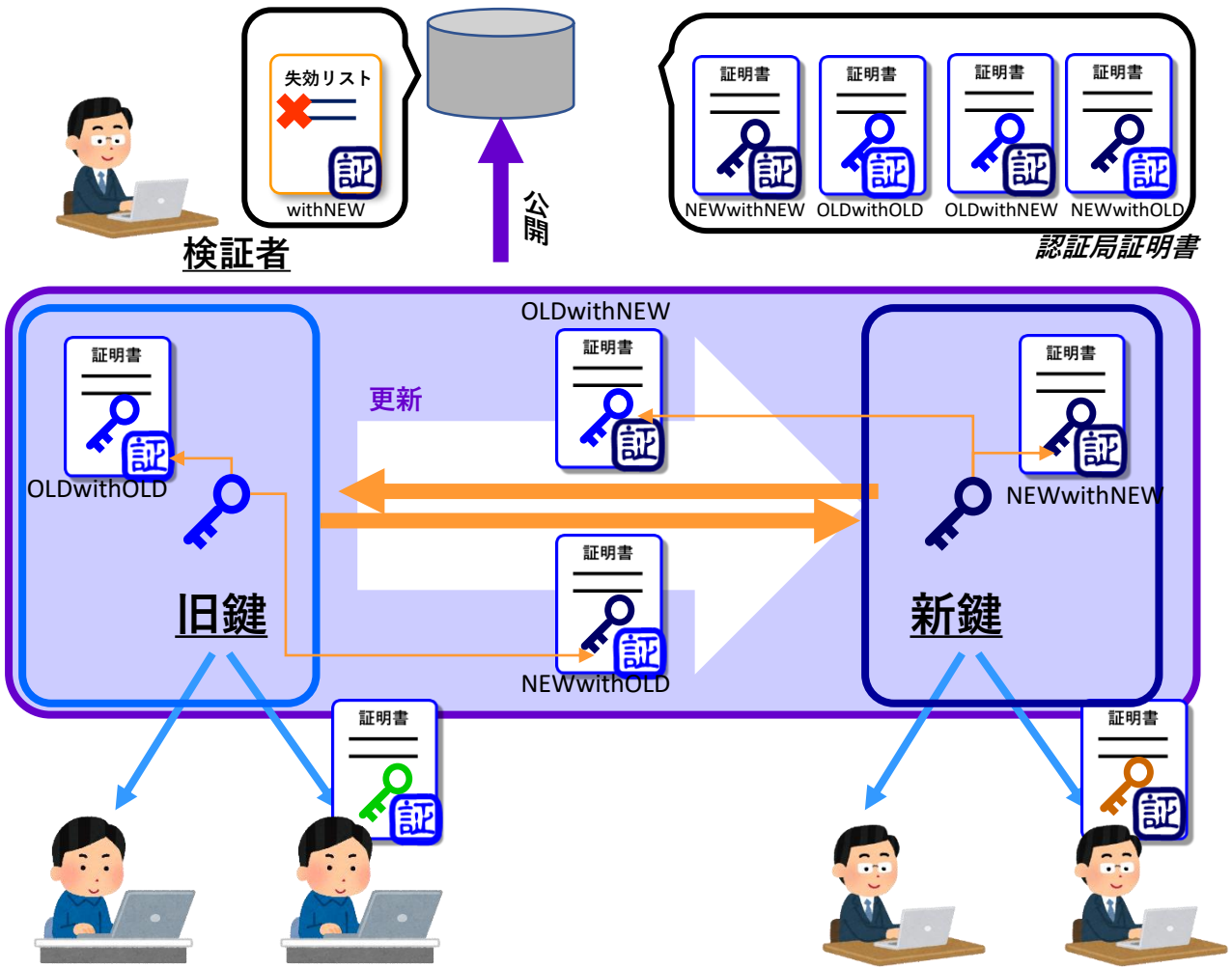
メリット
都度の相互認証確認の仕組みが不要
デメリット
事前に必要な認証局証明書を集める必要がある



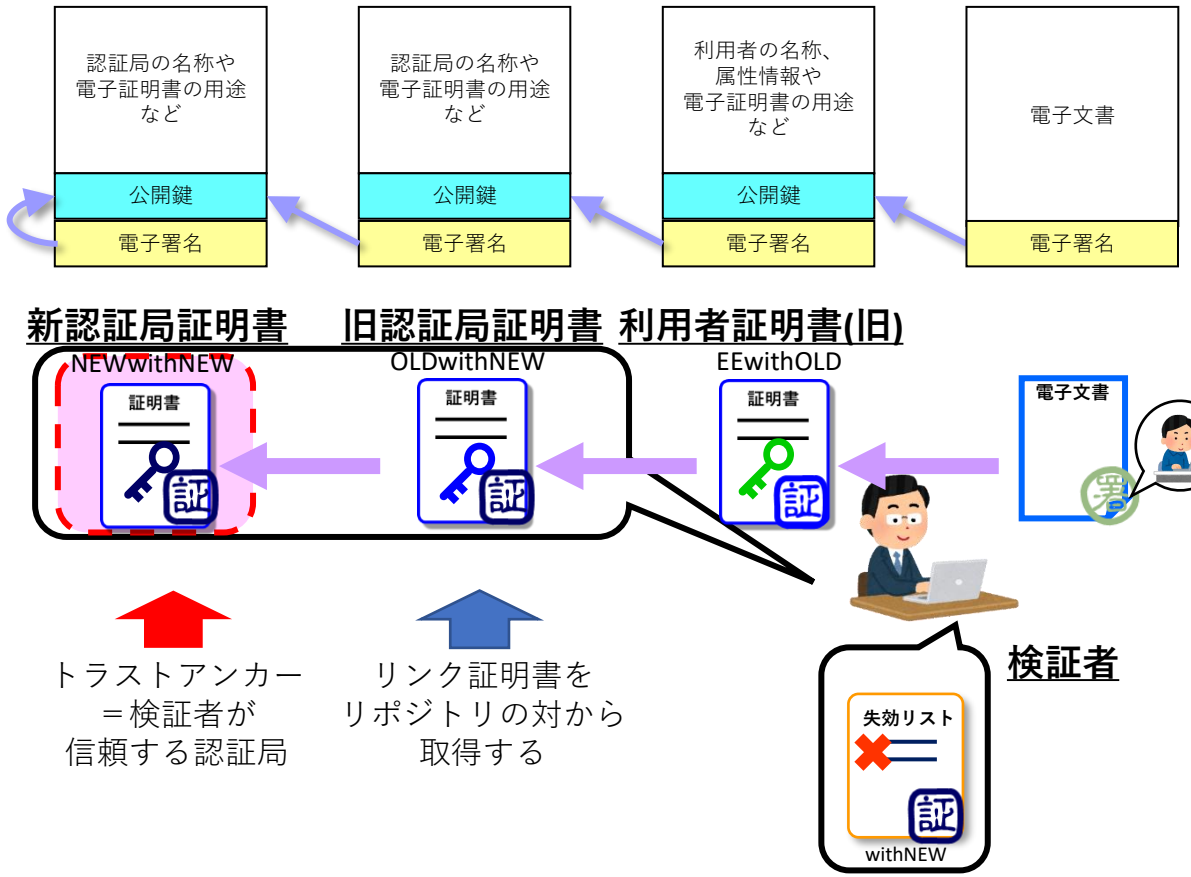
リンク証明書による証明書更新(シングルトラスト方式)

- **旧証明書と新証明書の関係**を結ぶために、**リンク証明書**にて認証パスを作る方式があり、ルート認証局では必須である。
- リンク証明書は、**新旧の鍵に対して証明書**を作り、その証明書を**認証局証明書**のひとつとしてリポジトリに公開する。
- 本仕組みにより、検証者が**新旧どちらをトラストアンカー**としても**すべての証明書につながる**認証パスが構築できる。

鍵更新のしくみ



認証パス(古い利用者証明書を新しい認証局証明書で検証)



トラストアンカーの方式を選ぶための観点

方式の比較

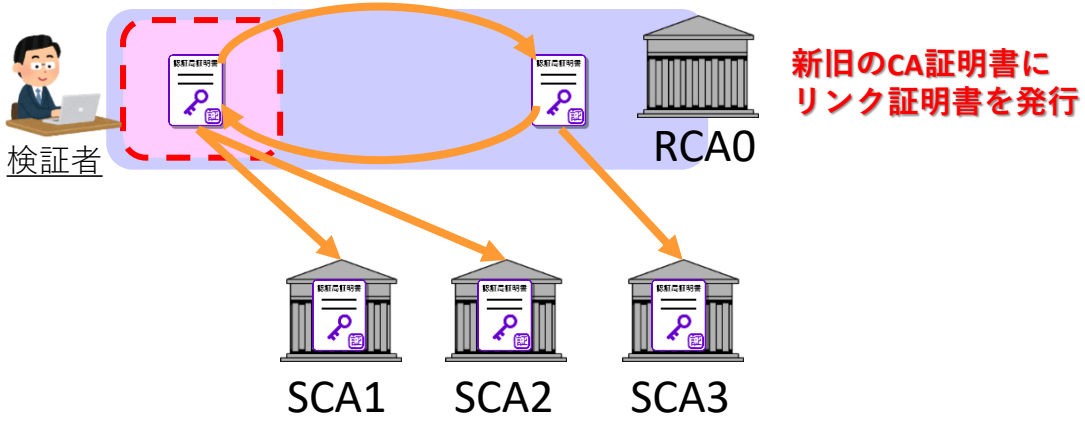
	シングルトラスト方式	マルチトラスト方式
Issuer(発行者名)の変更	**変更しない**	認証局鍵が新しくなるたびに**変更する**
ルート認証局の有効期間終了に向けた対処方法	旧認証局と**同一のIssuer**にて新しい有効期間の電子証明書を発行し、旧認証局証明書の間に**リンク証明書**を作る	新しい有効期間にて**新たなIssuerの認証局**を構築する。
サブ認証局の有効期間終了に向けた対処方法	旧認証局と**同一のIssuer**にて新しい有効期間の電子証明書を**有効なルート証明書から発行**する。 リンク証明書は任意であり、必然性はない。	新しい有効期間にて**新たなIssuerの認証局**に対して、**有効なルート証明書から発行**する。
トラストパスの構築	最新のルート認証局の証明書を基点として構築する。サブ認証局にリンク証明書がある場合、代替手段としてパスの構築が可能となる。	現在有効なルート認証局の証明書のいずれかを基点として構築する。
失効リスト(CRL)の発行について	ルート認証局やサブ認証局に限らず、**新旧双方の証明書の失効リスト**を、**最新の認証局の署名**を付して公開する。	ルート認証局やサブ認証局に限らず、**その認証局から発行した証明書**に対する証明書の失効リストを**発行元の認証局の署名**を付して公開する。
利点	Issuerを変える必要がない 管理する認証局(トラストアンカー)が増えない	ポリシ変更が柔軟に対応できる。 トラストパスの構築が単純にできる。

(補足)トラストアンカーの方式による更新イメージ

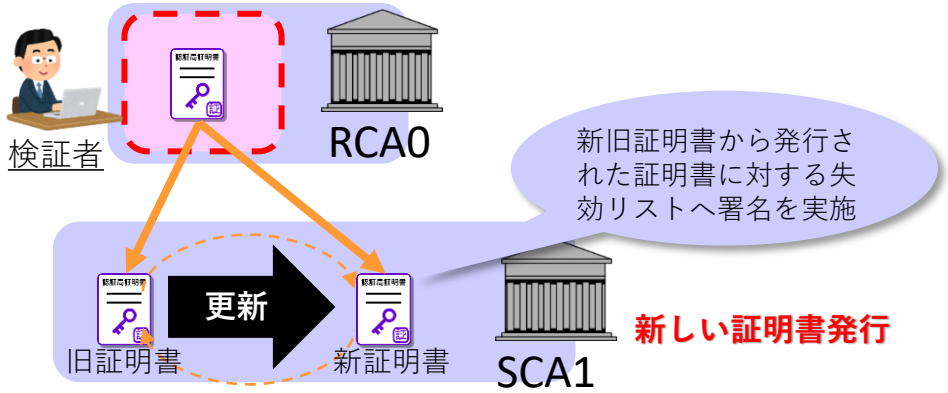
- シングルトラス方式は、狭義では信頼する電子証明書自体が一つとなるため、RootCAではリンク証明書が必要となる。
- マルチトラスト方式は、広義で信頼する認証局を拡張できるため、必要に応じて新しいRootCAをリストへ加える。
- シングルトラストとマルチトラストは、検証者視点ではあるが、認証局からは名前空間(DN)設計や失効リスト設計に影響する。

シングルトラスト方式

RootCAの更新はリンク証明書を用いて鍵更新する。



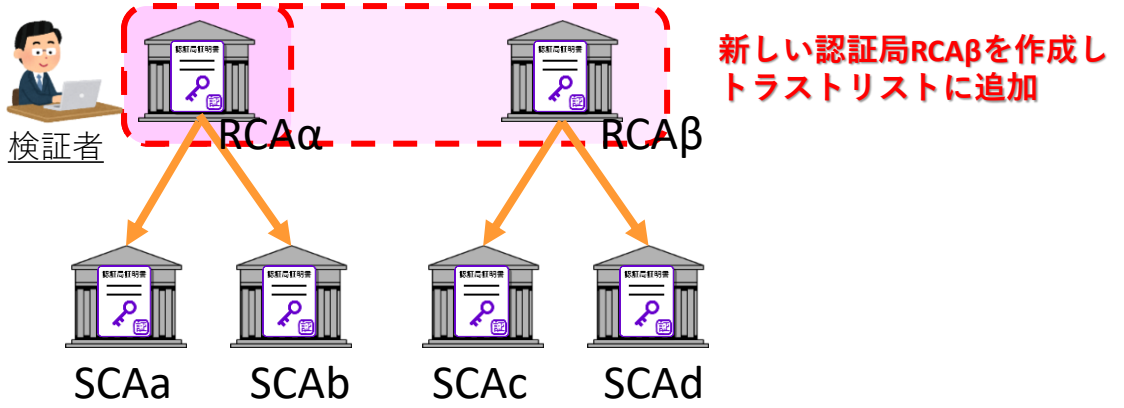
SubCAの更新は、同一のSubCA名で有効なRootCAよりCA証明書を発行し更新する。リンク証明書は必須ではない。



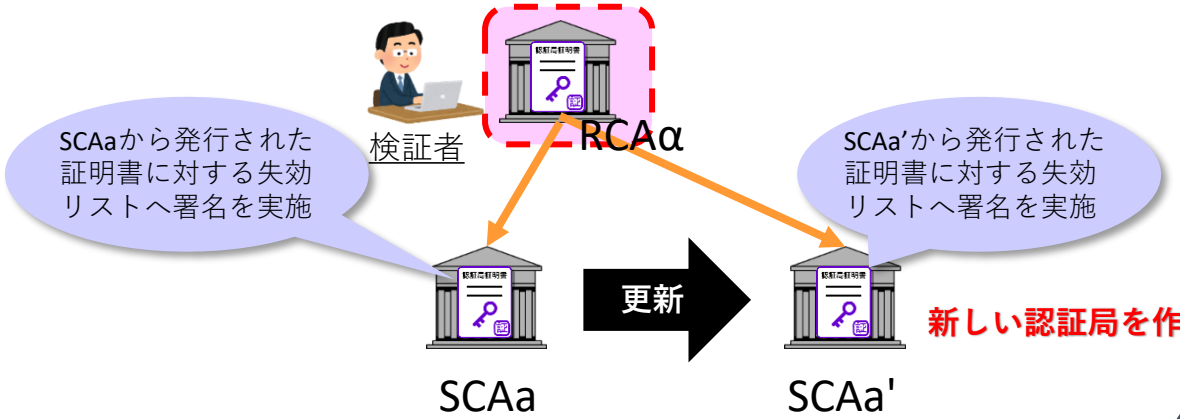
※リンク証明書は必須ではない

マルチトラスト方式

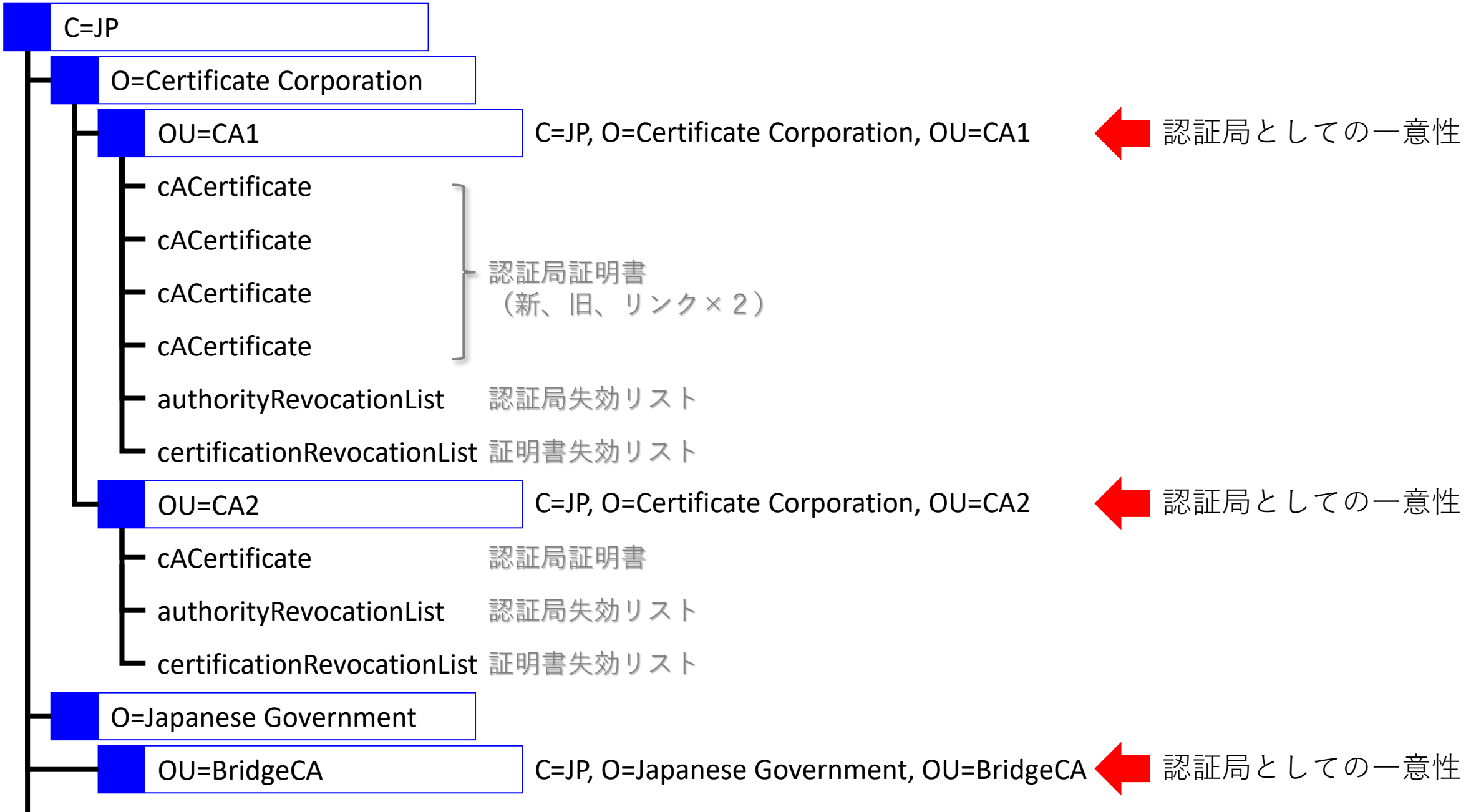
RootCAの更新は、別のRootCA名で認証局を立ち上げる



SubCAの更新は、別のSubCA名で有効なRootCAよりCA証明書発行し更新する。



X.500の世界におけるDirectory Treeによる一意性



RFC4210によるCA更新後の検証について

RFC4210

3.1.2. PKI Management Requirements The protocols given here meet the following requirements on PKI management. ~中略～ 10. A graceful, scheduled change-over from one non-compromised CA key pair to the next (CA key update) must be supported (note that if the CA key is compromised, re-initialization must be performed for all entities in the domain of that CA). An end entity whose PSE contains the new CA public key (following a CA key update) must also be able to verify certificates verifiable using the old public key. End entities who directly trust the old CA key pair must also be able to verify certificates signed using the new CA private key (required for situations where the old CA public key is "hardwired" into the end entity's cryptographic equipment).
3.1.2. PKI 管理要件 ここに示すプロトコルは、PKI 管理に関する以下の要件を満たす。 ～中略～ 10. 危殆化されていない CA 鍵ペアから次の CA 鍵ペアへの猶予のあるスケジュールされた変更（CA 鍵更新） がサポートされなければならない（CA 鍵が危殆化された場合、その CA のドメイン内のすべてのエンティ ティに対して再初期化が実行されなければならないことに注意）。PSE に新しい CA 公開鍵が含まれているエンド・エンティティ（CA 鍵の更新後）は、古い公開鍵を使用して検証可能な証明書も検証できなければならない。旧 CA 鍵ペアを直接信頼するエンド・エンティティは、新 CA 秘密鍵を使用して署名された証明書 も検証できなければならない（旧 CA 公開鍵がエンド・エンティティの暗号化機器に「ハードワイヤード」 されている場合に必要）。

新旧の認証局から発行された証明書は、どちらも検証できなければならない。

RFC2585,RFC4210,RFC5820のCRL公開について

RFC5280

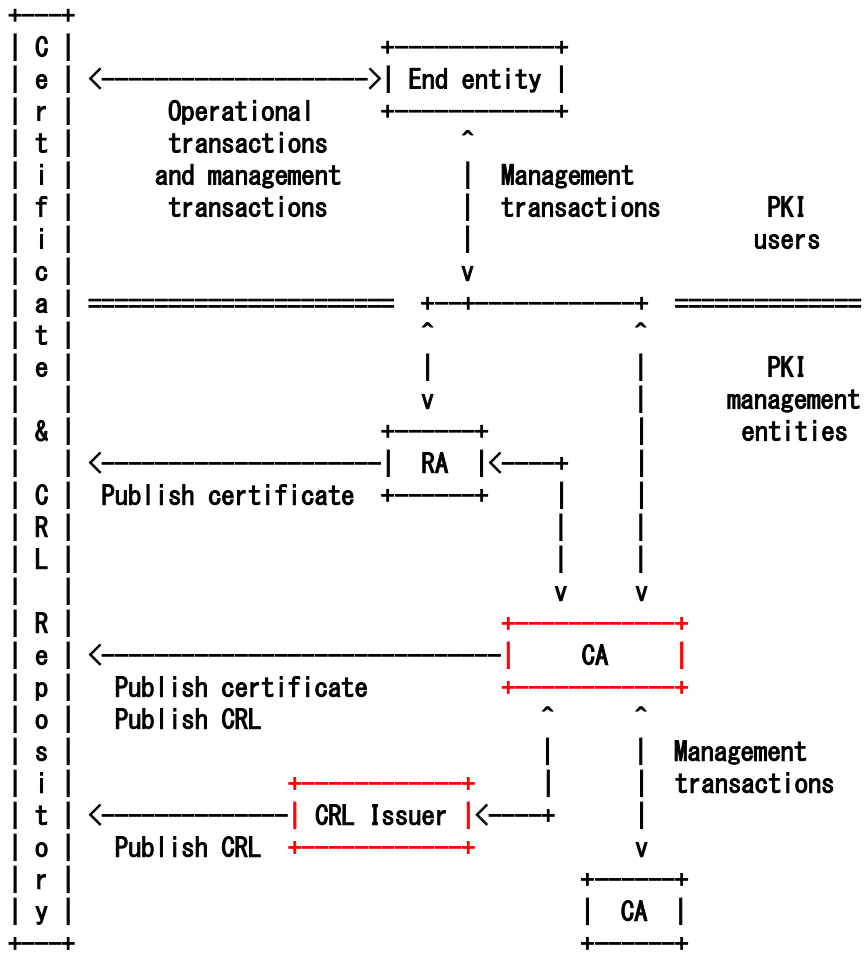


Figure 1. PKI Entities

証明書の発行者(Issuer)と失効リストの発行者(CRL Issuer)は異なっていてもよいモデル

※異なってもよいだけで異ならないモデルが多い

RFC5280の認証局の特定方法について

4.1.2.4. Issuer
The issuer field identifies the entity that has signed and issued the certificate. The issuer field MUST contain a non-empty distinguished name (DN). The issuer field is defined as the X.501 type Name [X.501].

4.1.2.4. 発行者
発行者フィールドは、証明書に署名し発行した**エンティティを識別**する。発行者フィールドは、空でない**識別名(DN)**を含まなければならない(MUST)。発行者フィールドは、X.501 タイプの Name [X.501]として定義される。

4.2.1.13. CRL Distribution Points
～略～
The cRLDistributionPoints extension is a SEQUENCE of DistributionPoint. A DistributionPoint consists of three fields, each of which is optional: distributionPoint, reasons, and cRLIssuer. While each of these fields is optional, a DistributionPoint MUST NOT consist of only the reasons field; either distributionPoint or cRLIssuer MUST be present. If the certificate issuer is not the CRL issuer, then the cRLIssuer field MUST be present and contain the Name of the CRL issuer. If the certificate issuer is also the CRL issuer, then conforming CAs MUST omit the cRLIssuer field and MUST include the distributionPoint field.
asserted. Otherwise, verify that the CRL issuer matches the certificate issuer.

4.2.1.13. CRL 配布ポイント
～略～
cRLDistributionPoints拡張はDistributionPointのSEQUENCEである。DistributionPointは3つのフィールドから構成され、各フィールドは任意である。distributionPoint、reasons、cRLIssuerである。各フィールドは任意であるが、DistributionPointは reasonsフィールドのみで構成されてはならない(MUST NOT)。証明書発行者が**CRL発行者でない場合**、cRLIssuerフィールドが存在し、**CRL発行者名を含まなけれ ばならない**(MUST)。**証明書発行者**が **CRL 発行者**でもある場合、適合する CA は cRLIssuer フィールドを省略し、distributionPoint フィールドを含めなければならない。

6.3.3. CRL Processing
～略～
(b) Verify the issuer and scope of the complete CRL as follows:
(1) If the DP includes cRLIssuer, then verify that the issuer field in the complete CRL matches cRLIssuer in the DP and that the complete CRL contains an issuing distribution point extension with the indirectCRL boolean asserted. Otherwise, verify that the CRL issuer matches the certificate issuer.

6.3.3. CRL 処理
～略～
(b) 以下のように、完全なCRLの発行者と範囲を検証する：
(1) DPにcRLIssuerが含まれている場合、CRLのissuerが DP の cRLIssuer と一致すること、およびCRLにindirectCRLが設定された発行配布ポイント拡張が含まれていることを確認します。それ以外は、CRL発行者が証明書発行者と一致することを検証する。

認証局の特定は、基本的には識別名(DN)を用いる。

失効リスト発行者の識別名(DN)が、認証局の識別名(DN)と異なる場合は、証明書および失効リストにそれぞれ記載する必要がある。

失効リスト発行者の識別名(DN) が、証明書の識別名(DN)と異なる場合は、それぞれの識別名(DN)の一致を確認する。
一緒の場合は、失効リスト発行者の識別名(DN)と認証局の識別名(DN)の一致を確認する。

認証局の一意性(証明書プロファイル)

項目	情報種別	説明	設定例
Certificate			
tbsCertificate			
version	証明書	電子証明書情報のフォーマットバージョン	2(v3)
serialNumber	証明書	電子証明書のシリアル番号	16進数16桁、前８桁は認証局証明書発行日を16進数化、後８桁はシリアル
signature	証明書	電子証明書の署名アルゴリズム	SHA256withRSA2048
issuer	**発行者**	**電子証明書の発行者名称**	**C=JP,O=Certificate Corporation, OU=CA1**
validity	証明書	電子証明書の有効期間	発行日および発行日から5年
subject	主体者	電子証明書の主体者名称(所有者、利用者)	C=JP,CN=Ninsho Taro
subjectPublicKeyInfo	主体者	公開鍵データ	RSA2048bit
issuerUniqueID	発行者	電子証明書の発行者ID	－
subjectUniqueID	主体者	電子証明書の所有者ID	－
extensions			
Authority Key Identifier	発行者	発行者の公開鍵の識別子	認証局証明書のDNおよびシリアル番号 認証局公開鍵のSHA1ハッシュ値
Subject Key Identifier	主体者	主体者の公開鍵の識別子(ハッシュ)	主体者公開鍵のSHA1ハッシュ値
Key Usage	証明書	鍵の利用方法(基本)	digitarSignature
Certificate Policies	証明書	電子証明書のポリシ	ANY
Policy Mappings	証明書	電子証明書のポリシマッピング	－
Subject Alternative Name	主体者	主体者の別名	C=JP, CN=認証　太郎
Issuer Alternative Name	発行者	発行者の別名	C=JP,O=Certificate Corporation, OU=CA1 サービス
Subject Directory Attributes	主体者	主体者の属性情報	－
Basic Constraints	証明書	電子証明書の基本制約	－
Name Constraints	証明書	電子証明書の名称制約	－
Policy Constraints	証明書	電子証明書のポリシ制約	－
Extended Key Usage	証明書	鍵の利用方法(拡張)	－
CRL Distribution Points	証明書	失効リストの配布場所情報	－
Inhibit anyPolicy	証明書	他ポリシーの禁止規定	－
Freshest CRL	証明書	差分失効リストの情報	－
Authority Information Access	証明書	機関情報のアクセス情報	－
Subject Information Access	主体者	主体者情報のアクセス情報	－
signatureAlgorithm	証明書	自認証局証明書の鍵情報より取得	SHA256withRSA2048
signatureValue	証明書	CTB11と自認証局の署名鍵により生成	バイナリデータ(SHA256withRSA2048)

認証局としての一意性

認証局**証明書(鍵)**としての一意性

主体者情報　発行者情報　利用情報（証明書）

認証局の一意性(CRLプロファイル)

項目	情報種別	説明	設定例
CertificateList			
tbsCertList			
version	ＣＲＬ	失効リスト情報のフォーマットバージョン	2(v3)
signature	ＣＲＬ	失効リストの署名アルゴリズム	SHA256withRSA2048
issuer	**発行者**	**失効リストの発行者名称**	**C=JP,O=Certificate Corporation, OU=CA1**
thisUpdate	ＣＲＬ	失効リストの発効日(有効期間開始)	－
nextUpdate	ＣＲＬ	次の失効リストの発効予定日(有効期間終了)	－
revokedCertificates			
userCertificate	主体者	発行者の公開鍵の識別子	失効された証明書のシリアル番号
revocationDate	主体者	失効日(認証局の失効手続きが完了した日)	証明書の失効日
crlEntryExtensions			
Reason Code	主体者	失効理由(コード)	0(unspecified)
Invalidity Date	主体者	失効日(実際に危殆化が起きた日)	－
Certificate Issuer	主体者	**利用者証明書の発行認証局(本issuerと違う際)**	**(C=JP,O=Certificate Corporation, OU=CA1)**
crlExtensions			
Authority Key Identifier	発行者	発行者の公開鍵の識別子	認証局証明書のDNおよびシリアル番号 認証局公開鍵のSHA1ハッシュ値
Issuer Alternative Name	発行者	発行者の別名	C=JP,O=Certificate Corporation, OU=CA1 サービス
CRL Number	ＣＲＬ	失効リストの番号(CRLのシリアル番号)	－
Delta CRL Indicator	ＣＲＬ	以前に配布された失効情報の更新	－
Issuing Distribution Point	ＣＲＬ	失効リストの配布場所情報	失効リストの配布場所
Authority Information Access	ＣＲＬ	機関情報のアクセス情報	－
signatureAlgorithm	ＣＲＬ	自認証局証明書の鍵情報より取得	SHA256withRSA2048
signatureValue	ＣＲＬ	CTB11と自認証局の署名鍵により生成	バイナリデータ(SHA256withRSA2048)

失効リスト発行局としての一意性
一般的には認証局と同じ

利用者証明書の発行認証局
issuerと同じなら記載しない

認証局**証明書(鍵)**としての一意性

主体者情報　発行者情報　利用情報（ＣＲＬ）

Memo
保健医療福祉分野PKI認証局 署名用証明書ポリシ
表7.2.2 証明書失効リストのプロファイル
→HoldInstructionCodeは、RFC3280からRFC5280に変更時に消滅なので削除が望ましい