

LGPKI 技術仕様書

第 1 . 2 版

平成 2 0 年 1 0 月

総合行政ネットワーク運営協議会

1	はじめに	1
1.1.	概要	1
1.2.	各章の位置付け	1
1.3.	前提	2
1.4.	見直し	2
2	LGPKI コンポーネント仕様	3
2.1.	概要	3
2.2.	組織 CA	3
2.3.	アプリケーション CA	4
2.4.	BCA	4
2.5.	公開リポジトリ	5
2.6.	証明書検証サーバ	5
2.7.	エンドエンティティ	5
2.8.	LGPKI における認証情報	7
2.8.1.	認証情報の公開	7
2.8.2.	相互認証証明書の格納・削除	8
2.8.3.	自己発行証明書の格納・削除	9
2.8.4.	失効情報の格納と更新	9
3	アプリケーション仕様	11
3.1.	概要	11
3.2.	証明書と失効情報 (CRL/ARL) のプロファイル	11
3.3.	証明書と失効情報 (CRL/ARL) の公開方法	13
3.4.	推奨署名アルゴリズム	13
3.4.1.	アルゴリズム	13
3.4.2.	鍵長	13
3.4.3.	認証パスの構築・検証方法	13
3.5.	LGPKI における名前と DIT の規定	14
3.5.1.	識別名、相対識別名	14
3.5.2.	エンコードタイプ	14
3.5.3.	issuerAltName 及び subjectAltName	14
4	証明書検証サーバの利用	15
4.1.	概要	15
4.2.	証明書検証サーバ用証明書 (VA 証明書)	15
4.3.	クライアント要件	15
4.3.1.	クライアント側の前準備	15
4.4.	証明書検証サーバ通信プロトコル	15

4.5.	証明書検証サーバのアクセス制御	15
5	ディレクトリプロファイル	16
5.1.	LGPKIにおけるDIT構造	16
5.2.	DITの名前形式	17
5.3.	公開リポジトリに格納される情報	17
5.3.1.	LGPKI コンテナ	17
5.3.2.	CA エントリ	19
5.4.	公開リポジトリのインタフェース仕様	24
5.5.	公開リポジトリのアクセス制御	25
5.5.1.	認証ポリシー	25
5.5.2.	アクセス制御ポリシー	25

1 はじめに

本仕様書は、LGPKI を構成する各 CA 及び LGPKI を利用するアプリケーションに関する技術仕様を定めるものである。本書の対象は、次の技術要件とする。

- LGPKI を構成する各コンポーネントが満たすべき技術要件
- LGPKI を利用するアプリケーションが満たすべき技術要件

1.1. 概要

LGPKI は、組織 CA 及びアプリケーション CA (以下、「組織 CA 等」という。) を中心とした認証基盤である。また、内部及び外部 CA との相互接続を効率的に実施する為にブリッジ CA (以下、「BCA」という。) を設置し柔軟な拡張性を有している。

本仕様書では、LGPKI を利用するために必要となる機能と仕様を定める。

1.2. 各章の位置付け

(2章) PKI コンポーネント仕様

2章では、LGPKI を構成する PKI コンポーネントの概要と基本的な仕様について記述する。対象とするコンポーネントの種類は以下のとおりとする。

- 組織 CA
- アプリケーション CA
- ブリッジ CA
- 公開リポジトリ
- 証明書検証サーバ
- エンドエンティティ

(3章) アプリケーション仕様

3章では、LGPKI を利用するアプリケーションが満たすべき機能を記述する。

また、この仕様書で記述するアプリケーションは、LGPKI を利用しデータに署名するソフトウェア (署名者) と署名されたデータを検証するソフトウェア (署名検証者) に位置付ける。

(4章) 証明書検証サーバの利用

4章では、LGPKI の提供する証明書検証サーバを利用する際に満たすべき仕様について定める。

(5章) ディレクトリプロファイル

5章では、公開リポジトリのプロファイルについて定める。

1.3. 前提

政府認証基盤相互運用性仕様書

(<http://www.gpki.go.jp/session/CompatibilitySpecifications.pdf>) 及び以下の標準等を考慮し、技術仕様を規定する。

- IETF : Internet X.509 Public Key Certificate Infrastructure and CRL Profile
- ITU-T : ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8

1.4. 見直し

本仕様書は情報通信技術の動向等を踏まえ必要に応じて見直すものとする。

2 LGPKI コンポーネント仕様

2.1. 概要

本章では、LGPKIを構成する各PKIコンポーネントの概要と基本的な仕様について記述する。
 なお、LGPKIを構成するPKIコンポーネントの概念図を以下に示す。

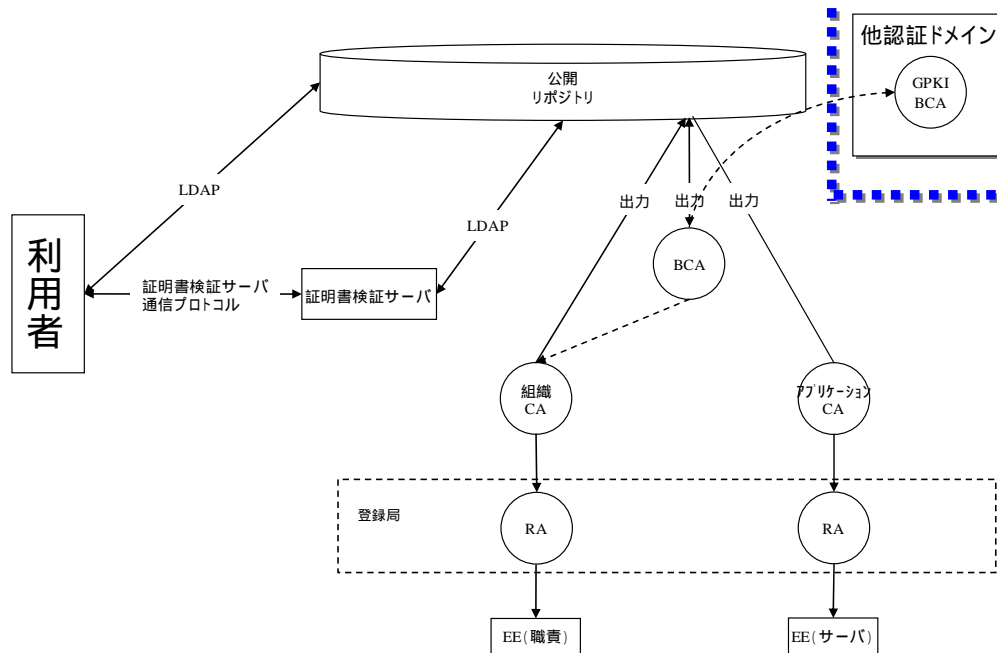


図 2 - 1 LGPKI 概念図

LGPKIは、組織CA（組織CA発行局及び登録局（以下、「RA」という。）、アプリケーションCA（アプリケーションCA発行局及びRA）BCA、公開リポジトリ及び証明書検証サーバから構成される。

2.2. 組織CA

組織CAは証明書の発行、失効、更新を行う。また各種証明書と失効情報（CRL/ARL）を公開リポジトリに格納する。

組織CAにはRAが存在する。RAは証明書所有者の身元を保証し、証明書に含まれる公開鍵と一致する秘密鍵を確実に証明書所有者が持っていることを保証する。

組織CAは以下の機能を備える。

- 自己署名証明書の発行
- 証明書の発行、更新

- BCA への相互認証証明書発行要求の作成、発行・更新された証明書の受け入れ
- BCA への相互認証証明書失効要求の作成
- エンドエンティティ（コンピュータシステムを含む）からの証明書発行要求の受け付け、証明書の発行、失効、更新
- 相互認証証明書の公開リポジトリへの格納
- 自 CA が発行した証明書に関する失効要求の受け付け、失効情報の発行と公開リポジトリへの格納
- 自 CA の各操作員の明確な権限分離が可能な証明書の発行・更新・失効
- 自 CA の鍵更新

2.3. アプリケーション CA

アプリケーション CA は証明書の発行、失効、更新を行う。また各種証明書と失効情報（CRL/ARL）を公開リポジトリに格納する。

アプリケーション CA には RA が存在する。RA は証明書所有者の身元を保証し、証明書に含まれる公開鍵と一致する秘密鍵を確実に証明書所有者が持っていることを保証する。

アプリケーション CA は以下の機能を備える。

- 自己署名証明書の発行と公開リポジトリへの格納
- 証明書の発行、更新
- エンドエンティティ（コンピュータシステムを含む）からの証明書発行要求の受け付け、証明書の発行、失効、更新
- 自 CA が発行した証明書に関する失効要求の受け付け、失効情報の発行と公開リポジトリへの格納
- 自 CA の各操作員の明確な権限分離が可能な証明書の発行・更新・失効
- 自 CA の鍵更新

2.4. BCA

BCA は二つの役割を担っている。一つは、LGPKI を構成する各 CA を束ねる役割であり、もう一方は、LGPKI が他認証ドメインと相互認証を行う際の橋渡しを行うという役割である。

BCA は各 CA と相互認証を行う。具体的には、お互いに相手の CA の公開鍵の証明書として相互認証証明書の発行と交換を行う。

BCA は他認証ドメインに属する CA（以下、「他 CA」という。）の公開鍵を含む相互認証証明書の発行、失効、更新を行う。

BCA は BCA が発行した相互認証証明書を含む相互認証証明書ペアと、失効情報を公開リポジトリの BCA エントリに格納する。ただし、組織 CA との間で発行された相互認証証明書は、BCA エントリに格納しない。

BCA の RA は各 CA の身元を保証し、相互認証証明書に含まれる公開鍵が確実にその CA の公開鍵であり、CA がこの公開鍵に一致する秘密鍵を持っていることを保証する。

BCA は以下の機能を備える。

- 自己署名証明書の発行と公開リポジトリへの格納
- 他 CA からの相互認証証明書発行要求の受け付けと検証、証明書の発行、更新
- 他 CA への相互認証証明書発行要求の作成、発行・更新された証明書の受け入れ
- 相互認証証明書ペアの公開リポジトリへの格納
- 他 CA への相互認証証明書失効要求の作成

- BCA が発行した証明書に関する失効要求の受け付け、失効情報の発行と公開リポジトリへの格納
- BCA の各操作員の明確な権限分離が可能な証明書の発行・更新・失効
- BCA の鍵更新

なお、鍵更新、リンク証明書の仕様については、政府認証基盤相互運用性仕様書に定められた仕様に準拠するものとする。

2.5. 公開リポジトリ

公開リポジトリは、証明書の有効性検証に必要な各種証明書、CRL/ARL を格納する。また、LDAPv3(389/tcp)を備えており、各 PKI コンポーネントからの証明書、CRL/ARL の検索、取り出し、格納が行える。

公開リポジトリは以下の機能を備える。

- BCA の自己署名証明書の格納
- 相互認証証明書及び相互認証証明書ペアの格納
- CRL/ARL の格納
- 証明書、CRL/ARL の検索
- 他リポジトリとの連携（リフェラル）

他のリポジトリとの連携に関しては、LDAPv3 リフェラルを使うことを前提とする。

2.6. 証明書検証サーバ

証明書検証サーバは、検証要求者が指定した証明書の妥当性を検証し、その検証要求者に証明書の検証結果を返すサーバである。OCSP レスポンダと異なる点は、証明書認証パス構築と各証明書の有効性に関する検証も可能なプロトコルを使用することである。

証明書検証サーバは相互運用に関する機能として以下を備える。

- 証明書検証サーバ自身の証明書の発行要求、発行された証明書の受け入れ
- 検証要求者からの証明書検証要求受け付け
- 認証パス構築
- 認証パス検証（指定されたポリシーによるチェック等）
- 検証要求者への証明書検証結果の送信

2.7. エンドエンティティ

エンドエンティティは、データに署名し送信する署名者と、認証パスを構築し署名を検証する署名検証者に大別できる。署名者は CA から証明書を発行され、その証明書に含む公開鍵と一致する秘密鍵でデータに署名することができる。

共通の機能として以下を備える。

- エンドエンティティ自身の証明書発行要求、発行された証明書の受け入れ
- エンドエンティティ自身の証明書の更新要求、発行された証明書の受け入れ

署名者のエンドエンティティは以下の機能を備えている。

- データに署名
- リポジトリから証明書の取得
- BCA までの認証パスの構築（オプション）

署名検証者のエンドエンティティは以下の機能を備えている。

- 署名の検証
- リポジトリから証明書、CRL/ARL の取得
- LDAPv3 リフェラルによる他のリポジトリへの連携
 エンドエンティティは、LDAPv3 をサポートすること。
- 証明書検証サーバへの問い合わせ機能、もしくは証明書検証サーバと同等の証明書検証機能

2.8. LGPKI における認証情報

LGPKI では、相互認証証明書や自己署名証明書及び失効情報を用いる。以降では、これら認証情報について記述する。

2.8.1. 認証情報の公開

BCA や組織 CA 等が発行する各種証明書と証明書の有効性を検証する情報である失効情報 (CRL/ ARL) の公開に関して記載する。ここでは、主に各種証明書と CRL/ARL の公開リポジトリへの格納・削除・更新について記述する。

証明書や CRL などの ASN.1 の構造をもつバイナリデータは、AttributeDescription での「Binary」オプションを指定し、バイナリ符号化して格納する。

表 2 - 1 に各認証情報と公開リポジトリへの格納・削除・更新の関係を示し、次節以降で格納形式等を記述する。

表 2 - 1 認証情報の公開リポジトリへの格納・削除・更新

		BCA の処理	組織 CA の処理	アプリケーション CA の処理
相互認証証明書	格納			-
	削除			-
	更新			-
エンドエンティティ用証明書	格納	-	-	-
	削除	-	-	-
	更新	-	-	-
自己発行証明書	格納		-	
	削除		-	
失効情報	格納			
	削除			
	更新			

ここで、自己発行証明書とは、自己署名証明書とリンク証明書のことをいう。

- : 処理対象
- : 処理対象外
- ・ エンドエンティティ用証明書は、公開リポジトリへの情報格納を行わない。

格納についての詳細な形式は RFC2587 の記述にしたがうものとする。

以下の証明書は、リポジトリ内の格納対象のエントリに複数格納される可能性がある。このため削除や更新時には、対象となる証明書以外に影響を与えないようにしなければならない。

- 各 CA エントリの cACertificate 属性に格納される、自己発行証明書もしくは相互認証証明書
- BCA エントリの crossCertificatePair 属性に格納される、相互認証証明書ペア

2.8.2. 相互認証証明書の格納・削除

(1)相互認証証明書の格納

ア) BCA の CA エントリへの格納

BCA が他 CA へ発行した、あるいは他 CA から発行された相互認証証明書の公開リポジトリへの格納は、次の形式で行う。

格納するエントリ	BCA の CA のエントリ	
格納するエントリが持たなければならないオブジェクトクラス	PkiCA (joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22))	
格納する属性名	crossCertificatePair 属性	
属性値の型	CertificatePair	
格納するフィールド ¹	Forward フィールド	他 CA が BCA の公開鍵に署名した相互認証証明書
	Reverse フィールド	BCA が他 CA の公開鍵に署名した相互認証証明書
複数 CA と相互認証した場合の属性値の扱い	相互認証した CA ごとに別の属性値として格納する	

相互認証証明書の格納は、相互認証証明書ペアを構成する 2 つの証明書が両方揃った時点で行っても、公開リポジトリへ格納できる状態になった方から随時格納してもよい。

また、BCA が組織 CA へ発行した相互認証証明書の公開リポジトリへの格納は、次の形式で行う。

格納するエントリ	BCA の CA のエントリ	
格納するエントリが持たなければならないオブジェクトクラス	PkiCA (joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22))	
格納する属性名	crossCertificatePair 属性	
属性値の型	CertificatePair	
格納するフィールド	Reverse フィールド	BCA が組織 CA の公開鍵に署名した相互認証証明書
複数 CA と相互認証した場合の属性値の扱い	相互認証した CA ごとに別の属性値として格納する	

イ) 組織 CA の CA エントリへの格納

BCA が発行した組織 CA に対する相互認証証明書に関しては、組織 CA のエントリに格納する。BCA のエントリには格納しない。公開リポジトリへの格納は、次の形式で行うものとする。

格納するエントリ	組織 CA の CA のエントリ	
格納するエントリが持たなければならないオブジェクトクラス	PkiCA (joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22))	
格納する属性	cACertificate 属性	
属性値の型	Certificate	

¹ International standard 9594-8 ITU-T RECOMMENDATION X.509 (03/2000)では、これまで forward / reverse と表記されていたフィールドを issuedToThisCA / issuedByThisCA と表記している。しかし、本書では政府認証基盤相互運用性仕様書との整合性を考慮し forward / reverse と表記する。

(2)相互認証証明書の削除

相互認証証明書の削除は、次のような場合に実施する。

- 自 CA が発行した相互認証証明書を失効した場合
- 相互認証する相手の CA が自 CA に対して発行した相互認証証明書を失効したことが、発行した相手の CA から通知された場合
- その他、公開リポジトリ内に存在する相互認証証明書が失効、もしくは、停止されたことを知った場合

2.8.3. 自己発行証明書の格納・削除

BCA 及び組織 CA 等の自己署名証明書は 10 年間有効である。そのうち、前半の 5 年間のみ相互認証証明書の発行に使用する。

(1)自己発行証明書の格納

BCA 及びアプリケーション CA が自らの公開鍵に自らの秘密鍵で署名して発行した自己発行証明書（自己署名証明書及びリンク証明書）の公開リポジトリへの格納は、次の形式で行うものとする。なお、組織 CA は、自己発行証明書の公開リポジトリへの格納は行わない。

格納するエン트리	CA のエン트리
格納するエント리가持たなければならないオブジェクトクラス	PkiCA (joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22))
格納する属性	cACertificate 属性
属性値の型	Certificate
複数の自己発行証明書を持つ場合の属性値の扱い	発行された自己発行証明書ごとに別の属性値として格納する

(2)自己発行証明書の削除

CA は、CA エン 트리内の cACertificate 属性中の証明書の削除を、次のような場合に実施する。

- 証明書内容変更、鍵長変更等で証明書を失効した場合
- 証明書の有効期限が切れた場合

2.8.4. 失効情報の格納と更新

(1)失効情報の格納

CA が発行した証明書の証明書失効リストは、各証明書の cRLDistributionPoint 拡張に示されたエン 트리、または CA のエン 트리に格納するものとする。

・ ARL

格納するエン 트리	各証明書内の cRLDistributionPoints で指定されたエン 트리または CA のエン 트리
格納するエント리가持たなければならないオブジェクトクラス	・ cRLDistributionPoints で指定されたエン トリの場合、 CRLDistributionPoint (joint-iso-itu-t(2) ds(5) objectClass(6) cRLDistiributionPoint(19))

	<ul style="list-style-type: none"> CA のエントリの場合、 pkiCA (joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22))
格納する属性	AuthorityRevocationList 属性 (ARL)
属性値の型	CertificateList

・CRL

格納するエントリ	各証明書内の cRLDistributionPoints で指定されたエントリまたは CA のエントリ
格納するエントリが持たなければならないオブジェクトクラス	<ul style="list-style-type: none"> cRLDistributionPoints で指定されたエントリの場合、 CRLDistributionPoint (joint-iso-itu-t(2) ds(5) objectClass(6) cRLDistributionPoint(19)) CA のエントリの場合、 pkiCA (joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22))
格納する属性	CertificateRevocationList 属性 (CRL)
属性値の型	CertificateList

(2)失効情報の更新

失効情報は、nextUpdate までに更新しなければならない。この時、実際の更新を nextUpdate より前に行い、新旧の失効情報がオーバーラップする期間を設けておく運用を行う。これにより、障害や事故等により次回の更新が遅れた場合に、失効情報がまったく公開されない期間が出現しないようにできる可能性が大きくなる。

3 アプリケーション仕様

3.1. 概要

本章では、LGPKI を利用するアプリケーションが満たすべき機能を記述する。前提として、以下の要件を満たすことが必要となる。

- アプリケーション間で関連する証明書と失効情報には互換性がなければならない
- 署名アルゴリズムは互換性がなければならない
- 相対する 2 者間で認証パスの構築と検証ができなければならない
- 証明書や失効情報を共有する方法には互換性がなければならない
- 相対する 2 者間で利用する署名フォーマットは互換性がなければならない
- ベンダー固有の依存性は避けなければならない

また LGPKI は、拡張性やアプリケーションの汎用性を確保する観点から、他の認証ドメインとの技術的な統一性を重視している。具体的には、公的な認証基盤である GPKI との技術的な統一を重視している。そのため、原則として、GPKI の技術仕様である、政府認証基盤相互運用性仕様書（平成 15 年 12 月 17 日改定）に定められた事項に準拠しなければならない。

以降では、LGPKI に固有な仕様について記述する。

3.2. 証明書と失効情報(CRL/ARL)のプロファイル

LGPKI で定義するプロファイルの詳細に関しては「地方公共団体組織認証基盤・プロファイル設計書」に示す。

なお、職責証明書、利用者証明書、メール用証明書、Web サーバ証明書及びコードサイニング証明書の証明書の名義（Subject）を各々表 3 - 1、表 3 - 2、表 3 - 3、表 3 - 4、及び表 3 - 5 に示す。

表 3 - 1 職責証明書

識別属性型	属性型	属性型説明	値の設定例
c	countryName	電子証明書所有者の国名	JP
o	organizeitionName	電子証明書所有者の組織名	Local Governments
l	localityName	電子証明書所有者の地域名	都道府県（英語）
ou	organizationalUnitName	電子証明書所有者の組織単位名	地方公共団体名（英語）
ou (*1)	organizationalUnitName	電子証明書所有者の組織単位名 (*2)	所属部門名（英語）
cn (*1)	commonName	電子証明書所有者の固有名義	役職名等（英語）

(*1) ou、cn の英語表記は 64 文字以内という制限があり、64 文字を超える名称等については 64 文字以内で収まるような名称に変更する。

(*2) 局、室、課名は、organizationalUnitName を用いて表現する。本 organizationalUnitName は、0 ~ 7 個の間で任意に用いることが出来る。

表 3 - 2 利用者証明書

識別属性型	属性型	属性型説明	値の設定例
c	countryName	電子証明書所有者の国名	JP
o	organizeitionName	電子証明書所有者の組織名	Local Governments
l	localityName	電子証明書所有者の地域名	都道府県（英語）

ou	organizationalUnitName	電子証明書所有者の組織単位名	地方公共団体名（英語）
ou (*1)	organizationalUnitName	電子証明書所有者の組織単位名 (*2)	所属部門名（英語）
cn (*1)	commonName	電子証明書所有者の固有名称	役職名等（英語）

(*1) ou、cn の英語表記は 64 文字以内という制限があり、64 文字を超える名称等については 64 文字以内で収まるような名称に変更する。

(*2) 局、室、課名は、organizationalUnitName を用いて表現する。本 organizationalUnitName は、0 ~ 7 個の間で任意に用いることが出来る。

表 3 - 3 メール用証明書

識別属性型	属性型	属性型説明	値の設定例
c	countryName	電子証明書所有者の国名	JP
o	organizeitionName	電子証明書所有者の組織名	Local Governments
l	localityName	電子証明書所有者の地域名	都道府県（英語）
ou	organizationalUnitName	電子証明書所有者の組織単位名	地方公共団体名（英語）
ou (*1)	organizationalUnitName	電子証明書所有者の組織単位名 (*2)	所属部門名（英語）
cn (*1)	commonName	電子証明書所有者の固有名称	役職名等（英語）
E	E-mail address	電子証明書所有者のメールアドレス	xxxxx@pref.xxxx.lg.jp

(*1) ou、cn の英語表記は 64 文字以内という制限があり、64 文字を超える名称等については 64 文字以内で収まるような名称に変更する。

(*2) 局、室、課名は、organizationalUnitName を用いて表現する。本 organizationalUnitName は、0 ~ 7 個の間で任意に用いることが出来る。

証明書発行要求（以下、「CSR」という）ファイルに設定できるサブジェクトが定型化されているなどの Web サーバ及び署名ツール側の技術的理由により、指定する識別名と同様の識別名構造での CSR ファイル作成が不可能な場合は、CSR ファイル作成側で指定する識別名の読み替えを行う。

表 3 - 4 Web サーバ証明書

識別属性型	属性型	読み替え	値の設定例
c	countryName	国コードをそのまま指定。	JP
st または s	state or province	指定しない。省略不可の場合、各地方公共団体の属する都道府県域を指定。	
l	localityName	各地方公共団体の属する都道府県域を指定。	都道府県（英語）
o	organizeitionName	Local Governments を固定で指定。	Local Governments
ou (*1)	organizationalUnitName	地方公共団体名を指定。サーバ管理組織名も任意で指定可能。	xxxxx Prefecture Soumubu IT Suishinshitsu
cn (*1)	commonName	サーバの完全修飾ドメイン名（FQDN）を指定。	www.pref.xxxxx.lg.jp

(*1) ou、cn の英語表記は 64 文字以内という制限があり、64 文字を超える名称等については 64 文字以内で収まるような名称に変更する。

表 3 - 5 コードサイニング証明書

識別属性型	属性型	読み替え	値の設定例
c	countryName	国コードをそのまま指定。	JP
st または s	state or province	指定しない。省略不可の場合、各地方公共団体の属する都道府県域を指定。	
l	localityName	各地方公共団体の属する都道府県域を指定。	都道府県（英語）
o	organizeitionName	Local Governments を固定で指定。	Local Governments
ou (*1)	organizationalUnitName	地方公共団体名を指定。サーバ管理組織	xxxxx Prefecture Soumubu

		名も任意で指定可能。	IT Suishinshitsu
cn (*2)	commonName	地方公共団体名及びコード管理責任者を表す CodeAdmin を指定 (*3)。組織名、アプリケーション名も任意で指定可能。	CodeAdmin of xxxxx Prefecture {組織名} {アプリケーション名}

(*1) ou、cn の英語表記は 64 文字以内という制限があり、64 文字を超える名称等については 64 文字以内で収まるような名称に変更する。

(*2) コード管理責任者の表記は、特段の理由がない限り CodeAdmin 固定とする。

3.3. 証明書と失効情報(CRL/ARL)の公開方法

LGPKI では基本的に公開リポジトリを使用し各種証明書や失効情報(CRL/ARL)を公開するものとする。そのため、登録内容に基準が無ければ、認証パスの構築や検証の際に参照できなくなってしまう。

本仕様書ではリポジトリ内のスキーマや DIT を示すディレクトリプロファイルを「5 ディレクトリプロファイル」に規定する。

3.4. 推奨署名アルゴリズム

LGPKI では以下に示す点に従って、推奨署名アルゴリズムを定める。

- 署名者があるアルゴリズムで署名したデータを、署名検証者が検証する際にはそのアルゴリズムを解釈できなければならない。
- 現状利用できるアルゴリズムが将来アルゴリズム危殆化により利用できなくなることも考慮しなければならない。
- 認証パス中に複数の署名アルゴリズムがある場合、認証パス中にある全てのアルゴリズムを解釈できなければならない。

3.4.1. アルゴリズム

エンドエンティティは署名検証する際、署名者が使用した署名アルゴリズムをサポートしていなければならない。

原則として署名アルゴリズムは sha1WithRSAEncryption (1.2.840.113549.1.1.5) を用いる。

3.4.2. 鍵長

署名検証者は、署名者が使用する鍵長の署名を検証できなければならない。また認証パスに含まれる全ての証明書の署名をその鍵長で検証できなければならない。そのため、少なくとも以下の鍵長での署名検証ができなければならない。

- RSA、2048 ビットまで

3.4.3. 認証パスの構築・検証方法

認証パスの構築・検証に関しては、政府認証基盤相互運用性仕様書(平成 15 年 12 月 17 日改定)に記載の技術要件に準拠することが必須である。

3.5. LGPKI における名前と DIT の規定

3.5.1. 識別名、相対識別名

LGPKI における名前としては、generalName の directoryName を用いる。issuer 及び subject もまた directoryName である。directoryName は識別名、すなわち一個以上の相対識別名のシーケンスである。

ここでは識別名及びその構成要素である相対識別名の使用可能文字、証明書へのエンコード方式について規定する。以後本書では、特に断らない限り「識別名」と記している場合、相対識別名も含むものとする。

これらは原則として RFC2459 の規定にしたがうものとするが、以下では LGPKI における特記事項について記述する。

3.5.2. エンコードタイプ

subject や issuer で使用される DN を記述する文字コードについては、原則として UTF8 String を用いる。ただし、アプリケーション CA から発行される証明書については、当面 Printable String で発行する。

3.5.3. issuerAltName 及び subjectAltName

issuerAltName 及び subjectAltName の directoryName については、日本語表記を格納することもできるものとする。その場合、エンコードタイプは UTF8String としなければならない。

4 証明書検証サーバの利用

4.1. 概要

LGPKI では、証明書検証サーバは BCA によって認証される。また、証明書検証サーバを利用するのは、BCA をトラストアンカーとする地方公共団体の職責者もしくはそれに準ずる証明書検証者である。また、検証対象となる証明書は、LGPKI が発行する証明書のほか、BCA が相互認証している他 CA が発行した証明書である。証明書検証サーバは、LGPKI が発行した証明書を検証する者に対し、証明書検証という非常に複雑な処理の代行や、さらに、LDAP による公開リポジトリへのアクセスが困難な者に対し、各種認証情報を提供する等、LGPKI が発行した証明書を検証する側の負担を軽減するものである。

本章では、まず BCA が提供する証明書検証サーバについて記述し、その後証明書を発行する BCA と、証明書検証サーバを利用する利用者クライアントが満たすべき仕様について記述する。

4.2. 証明書検証サーバ用証明書 (VA 証明書)

証明書検証サーバの証明書は、BCA が発行する。また、証明書検証サーバの証明書の extendedKeyUsage には id-kp-OCSPSigning を設定する。

4.3. クライアント要件

証明書検証サーバを利用するクライアントが備えているべき点を記述する。

4.3.1. クライアント側の前準備

証明書検証サーバのレスポンスデータに含まれる署名を検証する必要があるため、クライアントは自分の利用する証明書検証サーバの署名を検証するのに必要な下記の情報を設定する必要がある。

- トラストアンカーの証明書

4.4. 証明書検証サーバ通信プロトコル

証明書検証サーバとの通信プロトコルについては、別添 1 に示す。

4.5. 証明書検証サーバのアクセス制御

証明書検証サーバは、公的個人認証サービスから発行された証明書の検証を行う際、公的個人認証サービスの失効情報を参照する。LGPKI では、証明書検証サーバの利用に当たって、公的個人認証サービスから失効情報の参照が許可された地方公共団体からの利用だけに限ることとする。

5 ディレクトリプロファイル

5.1. LGPKI における DIT 構造

すべてのエンティティは、全体として矛盾のない一つのディレクトリ情報ツリー（DIT）を構成する。

DIT の第一階層は国、第二階層は LGPKI ("LGPKI") コンテナとする。第三階層は、LGPKI コンテナ以下にブリッジ CA およびアプリケーション CA、組織 CA とする。

なお LGPKI においては、発行する証明書の subject や issuer で使用される DN を記述する文字コードが Printable String の CA と UTF8String の CA がある。文字コードが UTF8String の CA に対しては、ou(organizationalUnitName)の識別名の最後に" U8"を付与している。

図 5 - 1 に LGPKI における DIT 構造を示す。

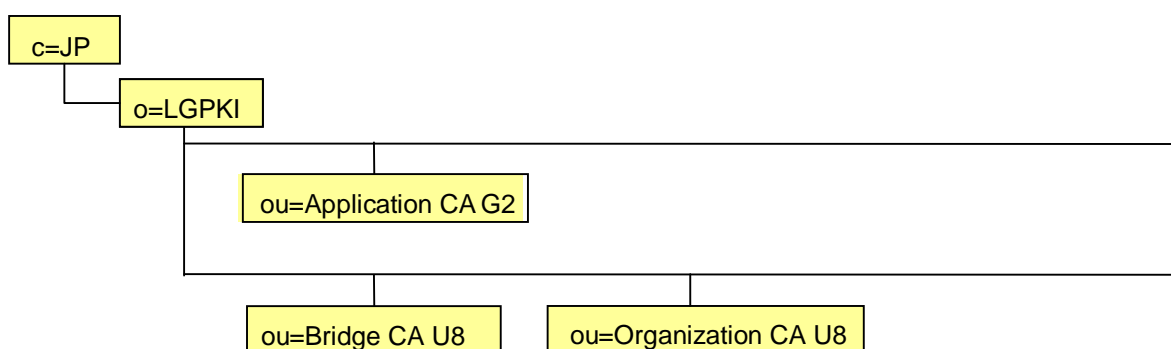


図 5 - 1 LGPKI における DIT 構造

5.2. DIT の名前形式

DIT 構造の各階層において、識別名に用いる属性を表 5 - 1 のように定める。

表 5 - 1 階層ごとの識別属性型と属性値

階層	識別属性型	属性値として取り得る値
第一階層	c	"JP"
第二階層	o	"LGPKI"
第三階層	ou	"Bridge CA U8"、"Application CA G2"、"Organization CA U8"

5.3. 公開リポジトリに格納される情報

LGPKI の公開リポジトリに格納されるエントリ及び属性について示す。

5.3.1. LGPKI コンテナ

LGPKI コンテナを表すエントリを構成するオブジェクトクラスとして、RFC2256 に定められている organization オブジェクトクラスを用いる。

organization オブジェクトクラスの定義、および organization オブジェクトクラスの上位オブジェクトクラスとして参照される top オブジェクトクラスの定義は、各々表 5 - 2、表 5 - 3 のとおりである。

表 5 - 2 organization オブジェクトクラスの定義

【オブジェクトクラス名】	organization
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) objectClass(6) organization(4)
【種別】	構造型
【上位オブジェクトクラス】	top
【必須属性】	o
【任意属性】	userPassword searchGuide seeAlso businessCategory x121Address registeredAddress destinationIndicator preferredDeliveryMethod telexNumber telexTerminalIdentifier telephoneNumber internationaliSDNNumber facsimileTelephoneNumber street postOfficeBox postalCode

	postalAddress physicalDeliveryOfficeName st 1 description
--	---

表 5 - 3 top オブジェクトクラスの定義

【オブジェクトクラス名】	top
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) objectClass(6) top(0)
【種別】	抽象型
【上位オブジェクトクラス】	なし
【必須属性】	objectClass
【任意属性】	なし

(1) LGPKI コンテナ

LGPKI のコンテナを表す名前("LGPKI")を格納する属性として organization オブジェクトクラスの必須属性である o 属性を用いる。

o 属性の定義、および o 属性の上位属性として参照される name 属性の定義は、各々表 5 - 4、表 5 - 5 のとおりである。

表 5 - 4 o 属性の定義

【属性名】	o
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) organizationName(10)
【上位属性】	name
【照合規則】	name 属性を継承
【属性構文】	name 属性を継承

表 5 - 5 name 属性の定義

【属性名】	name
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) name(41)
【上位属性】	なし
【照合規則】	< 完全一致 > caseIgnoreMatch

	< 部分一致 >	caseIgnoreSubstringMatch
	< 順序性 >	なし
【属性構文】		DirectoryString

5.3.2.CA エントリ

CA を表すエントリを構成するオブジェクトクラスとして、RFC2256 に定められている organizationalUnit オブジェクトクラスを用い、これに補助クラスとして、RFC2587 に定められている pkiCA オブジェクトクラスを付加する。

organizationalUnit オブジェクトクラスの定義、および organizationalUnit オブジェクトクラスの上位オブジェクトクラスとして参照される top オブジェクトクラスの定義は、各々表 5 - 6、表 5 - 3のとおりである。

表 5 - 6 organizationalUnit の定義

【オブジェクトクラス名】	organizationalUnit
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) objectClass(6) organizationalUnit(5)
【種別】	構造型
【上位オブジェクトクラス】	top
【必須属性】	objectClass ou
【任意属性】	userPassword searchGuide seeAlso businessCategory x121Address registeredAddress destinationIndicator preferredDeliveryMethod telexNumber telexTerminalIdentifier telephoneNumber internationaliSDNNumber facsimileTelephoneNumber street postOfficeBox postalCode postalAddress physicalDeliveryOfficeName st l description

pkiCA オブジェクトクラスの定義は表 5 - 7のとおりである。

表 5 - 7 pkiCA の定義

【オブジェクトクラス名】	pkiCA
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) objectClass(6) pkiCA(22)
【種別】	補助型
【上位オブジェクトクラス】	top
【必須属性】	objectClass
【任意属性】	cACertificate certificateRevocationList authorityRevocationList crossCertificatePair

(1) CA 名称

CA 名称 (ブリッジ CA ("Bridge CA U8")、アプリケーション CA ("Application CA G2") あるいは組織 CA ("Organization CA U8")) を格納する属性として、organizationalUnit オブジェクトクラスの必須属性である ou 属性を用いる。

ou 属性の定義、および ou 属性の上位属性として参照される name 属性の定義は、各々表 5 - 8、表 5 - 5 のとおりである。

表 5 - 8 ou 属性の定義

【属性名】	ou
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) organizationalUnitName(11)
【上位属性】	name
【照合規則】	name 属性を継承
【属性構文】	name 属性を継承

(2) 自己発行証明書

自己発行証明書を格納する属性として、pkiCA オブジェクトクラスの設定上追加必須属性である cACertificate 属性を用いる。

自己発行証明書は、BCA とアプリケーション CA だけが持つ。

この属性は "cACertificate;binary" と指定することによって属性値の受け渡しを行う。

cACertificate 属性の定義は表 5 - 9 のとおりである。

表 5 - 9 cACertificate 属性の定義

【属性名】	cACertificate
-------	---------------

【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) cACertificate(37)	
【上位属性】	なし	
【照合規則】	< 完全一致 >	certificateExactMatch
	< 部分一致 >	なし
	< 順序性 >	なし
【属性構文】	Certificate	

(3) 相互認証証明書

相互認証証明書は、次の二つの属性に格納される。

ア) crossCertificatePair 属性

BCA の CA エントリでは、他 CA との間で取り交わした相互認証証明書が、pkiCA オブジェクトクラスの設定上追加必須属性である crossCertificatePair 属性に格納される。

この属性は"crossCertificatePair;binary"と指定することによって属性値の受け渡しを行う。

crossCertificatePair 属性の定義は表 5 - 10 のとおりである。

表 5 - 10 crossCertificatePair 属性の定義

【属性名】	crossCertificatePair	
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) crossCertificatePair(40)	
【上位属性】	なし	
【照合規則】	< 完全一致 >	certificatePairExactMatch
	< 部分一致 >	なし
	< 順序性 >	なし
【属性構文】	CertificatePair	

イ) cACertificate 属性

組織 CA の CA エントリでは、BCA から発行された相互認証証明書が、pkiCA オブジェクトクラスの cACertificate 属性に格納される。

この属性は"cACertificate;binary"と指定することによって属性値の受け渡しを行う。

cACertificate 属性の定義は表 5 - 11 のとおりである。

表 5 - 11 cACertificate 属性の定義

【属性名】	cACertificate
-------	---------------

【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) cACertificate(37)	
【上位属性】	なし	
【照合規則】	< 完全一致 >	certificateExactMatch
	< 部分一致 >	なし
	< 順序性 >	なし
【属性構文】	Certificate	

(4) CRL

CRL を格納する属性として、pkiCA オブジェクトクラスの設定上追加必須属性である certificateRevocationList 属性を用いる。

この属性は"certificateRevocationList;binary"と指定することによって属性値の受け渡しを行う。

certificateRevocationList 属性の定義は表 5 - 1 2 のとおりである。

表 5 - 1 2 certificateRevocationList 属性の定義

【属性名】	certificateRevocationList	
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) certificateRevocationList(39)	
【上位属性】	なし	
【照合規則】	< 完全一致 >	certificateListExactMatch
	< 部分一致 >	なし
	< 順序性 >	なし
【属性構文】	CertificateList	

(5) ARL

CA に関する CRL を格納する属性として、pkiCA オブジェクトクラスの設定上追加必須属性である authorityRevocationList 属性を用いる。

この属性は"authorityRevocationList;binary"と指定することによって属性値の受け渡しを行う。

authorityRevocationList 属性の定義は表 5 - 1 3 のとおりである。

表 5 - 1 3 authorityRevocationList 属性の定義

【属性名】	authorityRevocationList	
【オブジェクト識別子】	joint-iso-itu-t(2) ds(5) attributeType(4) authorityRevocationList(38)	
【上位属性】	なし	

【照合規則】	< 完全一致 >	certificateListExactMatch
	< 部分一致 >	なし
	< 順序性 >	なし
【属性構文】		CertificateList

5.4. 公開リポジトリのインタフェース仕様

公開リポジトリに対して、民間側が利用する行政サービスアプリケーションからは、証明書・署名の検証のための認証情報の検索が行われる。公開リポジトリに対する一切の更新系の操作は許可されない。

民間側が利用する行政サービスアプリケーションは、LDAPv3 をサポートするものとする。

BindRequest

認証なしによる Bind のみを許可する。この場合、BindRequest パラメータは以下のとおりである。

version	2 あるいは 3
name	"" (長さ 0 の文字列)
authentication.simple	"" (長さ 0 の文字列)

Bind Response

RFC2251 4.2.3 の規定のとおりとする。

Unbind Request/Response

RFC2251 4.3 の規定のとおりとする。

Search Request

以下の事項以外については、RFC2251 4.5.1 の規定のとおりとする。

derefAliases パラメータは neverDerefAliases のみとする。

typesOnly パラメータは FALSE としなくてはならない。

filter 項目について、approxMatch 及び extensibleMatch はサポートしない。

SearchResEntry/ResDone/ResRef

RFC2251 4.5.2 及び 4.5.3 の規定のとおりとする。

SearchResponse

RFC1777 4.3 の規定のとおりとする。

5.5. 公開リポジトリのアクセス制御

5.5.1. 認証ポリシー

- 原則として認証しない。匿名によるアクセスを許可する。
この時、後述するアクセス制御ポリシーに従い、一部を除くすべての情報の参照が可能となるが、あらゆる情報の更新はできない。
- 公開リポジトリが管理・格納する情報を更新する操作員に対しては、最低限パスワードによる認証を行う。この時、後述するアクセス制御ポリシーに従い、許可された範囲の情報に対する参照・更新・削除が可能となる。

5.5.2. アクセス制御ポリシー

以下の方針によるアクセス制御を実施する。

- LGPKI 公開リポジトリにアクセス可能なすべてのクライアントに対して、すべての情報に対する参照権限を与える。ただし、一部公開すべきでない属性については、参照権限を与えない。さらに、各操作員を表すエントリに対しては、自分以外からの参照権限を与えない。
- 資格と権限のある操作員に対して、公開リポジトリが管理・格納するすべての情報について、参照・更新権限を与える。さらに、エントリの追加・削除・識別名変更権限を与える。

証明書検証サーバ通信プロトコル

1. はじめに

1.1. 証明書検証サーバの利用手順

証明書検証サーバの利用手順について説明する。

なお、証明書検証サーバと証明書検証サービスの利用者との通信は、SSL (https(TCP/443)) のサーバ認証を用いた暗号化通信とする。

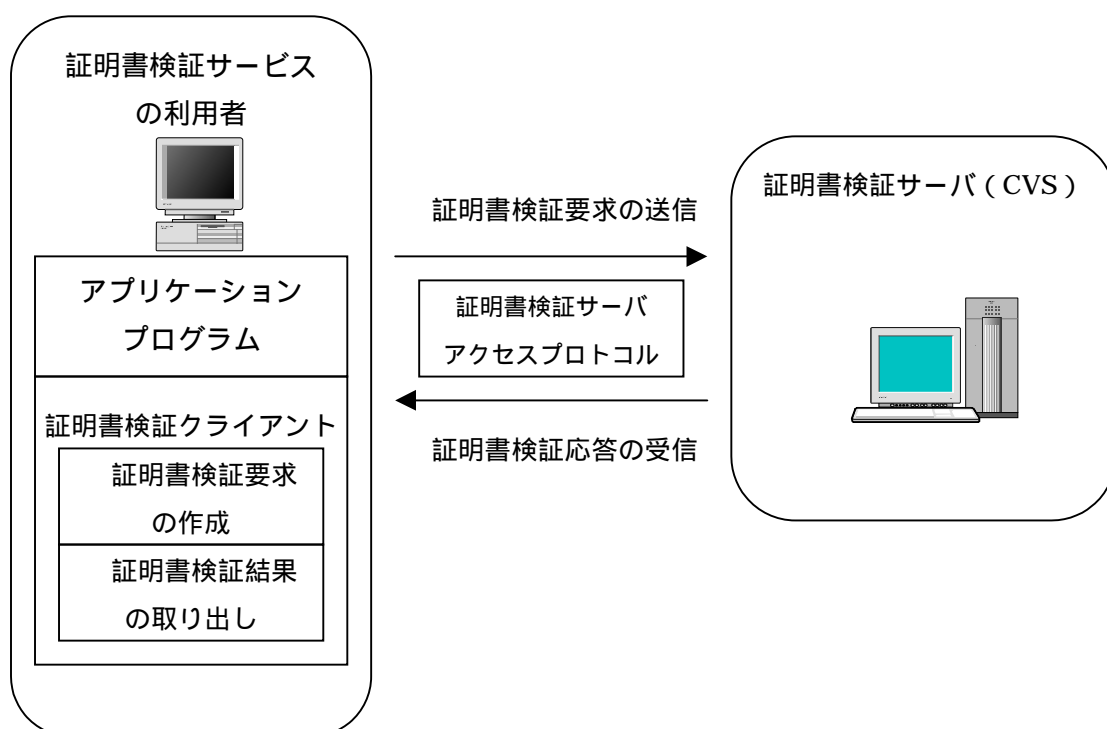


図 1-1 証明書検証の流れ

証明書検証要求の作成

アプリケーションプログラムは、証明書検証要求を 2 章以降で説明する証明書検証サーバアクセスプロトコルに従ったフォーマットで作成する。

証明書検証要求の送信

アプリケーションプログラムは、作成した証明書検証要求を証明書検証サーバ(CVS)に送信する。

証明書検証応答の受信

アプリケーションプログラムは、証明書検証サーバ(CVS)から送信された証明書検証応答を受信する。このとき受信した証明書検証応答は、2章以降で説明する証明書検証サーバアクセスプロトコルに従ったフォーマットとなる。また、証明書検証応答中の証明書検証サーバによる署名の検証を行う。

証明書検証結果の取り出し

アプリケーションプログラムは、証明書検証応答から証明書検証結果の取り出しを行う。

2. 通信プロトコル

OCSP(RFC2560)をベースとし、その拡張領域を証明書検証サーバ用に以下のように定義する。なお、拡張領域に付与するOID(オブジェクト識別子)は次のものをベースとして指定するものとする。

OID: 1.2.392.200010.10

2.1. 証明書検証要求(OCSP Request + 拡張)

2.1.1. OCSP 基本領域(OCSPRequest)

RFC2560の規定に従う。

ただし、requestExtensions に nonce を必須とする。

また、acceptableResponses, optionalSignature は用いないこととする。

2.1.2. OCSP 拡張領域 (singleRequestExtensions)

OCSP Request の拡張領域(singleRequestExtensions)に以下の情報を設定する。

2.1.2.1. 検証対象(subscriber)の証明書(必須)

検証の対象となる subscriber のエンドエンティティ証明書を指定する。

この Extension は必須である。

拡張部分への設定フォーマットを次表に示す。

表 2-1 検証対象の証明書の設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.1 :subscriberCert	独自定義
Critical flag	Critical	
Value	証明書(X.509)	

2.1.2.2. 中間証明書(オプション)

検証対象者(subscriber)側のルート認証局 (Trust Anchor)から subscriber のエンドエンティティまでの証明書等、認証パスの一部となる証明書を指定する。ただし、証明書検証サーバは、中間証明書を認証パス構築のためのヒント情報として利用するだけであり、必ずしも指定した証明書を含むパスを常に構築するわけではない。

この Extension はオプションであり、複数指定可能である。

また、複数指定する場合は、認証パスにおいて Trust Anchor に近い証明書から順に指定すること。

拡張部分への設定フォーマットを次表に示す。

表 2-2 中間証明書の設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.2 :intermediateCerts	独自定義
Critical flag	Critical	
Value	証明書(X.509)	

2.1.2.3. CA(Trust Anchor)の識別情報(オプション)

検証依頼者(relying party)が信頼するルート認証局 (Trust Anchor)を識別するための情報として次の情報を指定可能とする。

- CA 証明書

この Extension はオプションである。省略した場合、検証依頼者(relying party)が信頼するルート認証局 (Trust Anchor)は、証明書検証サーバ側で設定したルート認証局となる。

ただし、LGPKI では、必ず LGPKI ブリッジ認証局証明書を指定することとし、それ以外が指定された場合は、エラー(エラーコード:901)を返すこととする。

拡張部分への設定フォーマットを次表に示す。

表 2-3 ルート認証局 (Trust Anchor)証明書の設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.3 :trustAnchorCert	独自定義
Critical flag	Critical	
Value	証明書(X.509)	

2.1.2.4. 満たすべきポリシー(オプション)

検証依頼者(relying party)が、認証パスに対して満たすべきポリシーを要求する場合に、そのポリシーの object id を指定する。満たすべきポリシーが指定された場合、証明書検証サーバは、構築した認証パスが指定されたポリシーを満たすかどうかを検査する。

この Extension はオプションであり、複数指定可能である。

表 2-4 満たすべきポリシー設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.4 : requiredPolicy	独自定義
Critical flag	Critical	
Value	ポリシーの object-id	

2.1.2.5. require-explicit-policy の初期値(オプション)

検証依頼者(relying party)が、認証パスに対して require-explicit-policy を要求する場合に、範囲を指定する。require-explicit-policy が指定された場合、証明書検証サーバは、require-explicit-policy の範囲外の証明書にポリシーが存在することを検査する。

この Extension はオプションである。

LGPKI では、満たすべきポリシーが指定された場合、この Extension は常に“0(ゼロ)”を指定することとする。

表 2-5 require-explicit-policy の初期値設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.5 :require-explicit-policy	独自定義
Critical flag	Critical	
Value	0-xxx(integer)	

2.1.2.6. inhibit-policy-mapping の初期値 (オプション)

検証依頼者(relying party)が、認証パスに対して inhibit-policy-mapping を要求する場合には、範囲を指定する。inhibit-policy-mapping が指定された場合、証明書検証サーバは、inhibit-policy-mapping の範囲外の証明書にポリスマッピングが存在しないことを検査する。

この Extension はオプションである。

LGPKI では、ポリスマッピングを必須とするため、この Extension は常に指定しないこととする。

表 2-6 inhibit-policy-mapping の初期値設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.6 :inhibit-policy-mapping	独自定義
Critical flag	Critical	
Value	0-xxx(integer)	

2.1.2.7. 応答フォーマット(オプション)

検証依頼者(relying party)が、証明書検証応答として返却すべき情報のレベルを指定する。

この Extension はオプションである。(省略時は証明書の検証結果のみ返却する。)

指定可能な情報レベルを次に示す。

- 0: 証明書の検証結果のみ返却する(デフォルト)
- 1: 証明書の検証結果に加えて、認証パス及び CRL/ARL を返却する。

拡張部分への設定フォーマットを次表に示す。

表 2-7 応答フォーマット設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.7 : responseFormat	独自定義
Critical flag	Critical	
Value	0 または 1(integer)	

2.2. 証明書検証応答 (OCSP Response + 拡張)

2.2.1. OCSP 基本領域 (OCSPResponse)

RFC2560 の規定に従う。

ただし、responseExtensions に nonce を必須とする。

また、nextUpdate は用いないこととする。

2.2.2. OCSP 拡張領域 (singleExtensions)

OCSP Response の拡張領域 (singleExtensions) に以下の情報を設定する。

2.2.2.1. 証明書検証結果 (必須)

証明書検証 (認証パスの構築及び検証) の結果コードが設定される。

この Extension は必須である。

証明書検証の結果コードを次に示す。

- 0: 認証パスの構築が成功し検証結果が正しい(good)
- 101: 認証パス構築不可
- 202: 認証パスに署名が不正である証明書が含まれる
- 203: 認証パスに失効した証明書が含まれる
- 204: 認証パスに Policy Mappings に Any-Policy が設定された証明書が含まれる
- 205: 認証パスに制約に違反している証明書が含まれる
- 206: 認証パスに OCSP での CertStatus が unknown と応答される証明書が含まれる
- 901: 証明書検証サーバ側で要求の受け付けを拒否した
- 902: 要求がタイムアウトとなった

拡張部分への設定フォーマットを次表に示す。

表 2-8 証明書検証結果設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.8 : certPathStatus	独自定義
Critical flag	Critical	
Value	0-xxx(integer)	

2.2.2.2. 認証パス(オプション)

構築した認証パスの全証明書を設定する。

この Extension はオプションであり、1つまたは複数設定される。(証明書検証要求の応答フォーマットで返却が要求されている場合に限り設定される。)

拡張部分への設定フォーマットを次表に示す。

表 2-9 認証パス設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.9 : CertPath	独自定義
Critical flag	Non Critical	
Value	証明書(X.509)	

2.2.2.3. CRL/ARL(オプション)

構築した認証パスの証明書について、証明書発行者(OCSP レスポンダを除く)が発行した失効リスト(CRL/ARL)を設定する。

この Extension はオプションであり、1つまたは複数設定される。(証明書検証要求の応答フォーマットで返却が要求されている場合に限り設定される。)

拡張部分への設定フォーマットを次表に示す。

表 2-10 CRL/ARL 設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.10 : revocationList	独自定義
Critical flag	Non Critical	
Value	CRL(X.509)	

2.2.2.4. OCSP レスポンダからの応答(オプション)

構築した認証パスの証明書について、証明書発行者(OCSP レスポンダに限る)から取得した OCSP 応答情報を設定する。

この Extension はオプションであり、1つまたは複数設定される。(証明書検証要求の応答フォーマットで返却が要求されている場合に限り設定される。)

拡張部分への設定フォーマットを次表に示す。

表 2-11 OCSP レスポンダからの応答設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.11 : OCSPResponse	独自定義
Critical flag	Non Critical	
Value	OCSP レスポンスデータ(RFC2560)	

2.2.2.5. 適合したポリシー(オプション)

構築した認証パスの証明書について、検証結果として有効なポリシーを設定する。

この Extension はオプションであり、1つまたは複数設定される。(証明書検証要求の応答フォーマットで返却が要求されている場合に限り設定される。)

拡張部分への設定フォーマットを次表に示す。

表 2-12 適合したポリシー設定フォーマット

プロトコル要素	指定する値	備考
Object id	1.2.392.200010.10.12 : mappedPolicy	独自定義
Critical flag	Non Critical	
Value	ポリシーの object-id	

3. 電文フォーマット

証明書検証要求、応答はそれぞれ https(TCP/443)を通信プロトコルとする。証明書検証サーバとの通信で利用する HTTP ヘッダを以下のように定義する。

3.1. 証明書検証要求

表 3-1 証明書検証要求

項番	項目	設定値	区切り
1	Request-line	POST / HTTP/1.0	CR+LF
2	entity-header	Content-Type: application/ocsp-request	CR+LF
		Content-Length: N	CR+LF
3	(空行)		CR+LF
4	entity-body	要求内容	

注(1) CR+LF: ASCII コードの 0d(16 進)、0a(16 進)を続けることを示す。

注(2) N: entity-body のバイト長を示す。

注(3) : 空白(ASCII コード 20(16 進))を示す。

3.2. 証明書検証応答

表 3-2 証明書検証応答

項番	項目	設定値	区切り
1	Status-line	HTTP/1.0 200 OK	CR+LF
2	entity-header	Content-Type: application/ocsp-response	CR+LF
		Content-Transfer-Encoding: Binary	CR+LF
		Content-Length: N	CR+LF
3	(空行)		CR+LF
4	Entity-body	応答内容	

注(1) CR+LF: ASCII コードの 0d(16 進)、0a(16 進)を続けることを示す。

注(2) N: entity-body のバイト長を示す。

注(3) : 空白(ASCII コード 20(16 進))を示す。

4. 証明書検証要求・応答プロトコル

4.1. 証明書検証要求プロトコル

LGPKI における証明書検証サーバにアクセスする際の証明書検証要求プロトコルを以下に示す。

表 4-1 証明書検証要求プロトコル

フィールド	データ型	設定値	必須 (注 1)
-	OCSPRequest		
tbsRequest	TBSRequest		
requestList	SEQUENCE OF Request		
-	Request		
reqCert	CertID(注 2)		
hashAlgorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.3.14.3.2.26	
issuerNameHash	OCTET STRING	20 バイト以下の値を設定する。	
issuerKeyHash	OCTET STRING	20 バイト以下の値を設定する。	
serialNumber	CertificateSerialNumber	シリアル番号を設定する。	
singleRequestExtensions	[0]		
-	Extensions		
(subscriberCert)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.1	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		
-	Certificate	検証の対象となるsubscriberの証明書を設定する。	
(intermediateCerts)	Extension(注 3)		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.2	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		
-	Certificate	中間証明書を設定する。	
(trustAnchorCert)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.3	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		
-	Certificate	検証依頼者が信頼するルートCAの自己署名証明書を設定する。	
(requiredPolicy)	Extension(注 3)		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.4	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		
-	OBJECT IDENTIFIER	検証依頼者が受け入れるポリシーのOIDを設定する。	
(require-explicit-policy)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.5	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		
-	INTEGER	require-explicit-policyの範囲を設定する。 (注4)	
(inhibit-policy-mapping)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.6	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		

-	INTEGER	inhibit-policy-mappingの範囲を設定する。(注5)	
(responseFormat)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.7	
critical	BOOLEAN	TRUE	
extnValue	OCTET STRING		
-	INTEGER	応答フォーマットを指定する。	
requestExtensions	[2]		
-	Extensions		
(nonce)	Extension		
extnId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1.2	
extnValue	OCTET STRING		
-	Nonce (注6)	1バイト以上32バイト以下の乱数を設定する。	

- 注1 印のあるフィールドは必須である。 印のあるフィールドは、オプションである。 印のあるフィールドには、上欄にフィールドを設けたときは、必須である。
- 注2 証明書検証サーバは CertID ではなく Subscriber の証明書で検証対象となる証明書を特定している。
- 注3 複数指定する場合は、同じ OID を持つ Extension を複数設定する。
- 注4 設定する場合は、必ず「0」を設定する。
- 注5 本拡張は利用しない。
- 注6 OCTET STRING としてカプセル化されていること。

4.2. 証明書検証応答プロトコル

LGPKI における証明書検証サーバにアクセスする際の検証応答プロトコルを以下に示す。

表 4-2 証明書検証応答プロトコル

フィールド	データ型	設定値	必須 (注1)
-	OCSPResponse		
responseStatus	OCSPResponseStatus	0	
responseBytes	[0]		
-	ResponseBytes		
responseType	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1.1	
response	OCTET STRING		
-	BasicOCSPResponse		
tbsResponseData	ResponseData		
responderID	ResponderID([1])		
-	Name	署名を行った CVS の電子証明書の「subject」が設定される。	
producedAt	GeneralizedTime	この電文を作成した日時が設定される。	
responses	SEQUENCE OF SingleResponse		
-	SingleResponse		
certID	CertID		
hashAlgorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.3.14.3.2.26	
issuerNameHash	OCTET STRING	Request で設定した値が設定される。	
issuerKeyHash	OCTET STRING	Request で設定した値が設定される。	
serialNumber	CertificateSerialNumber	Request で設定した値が設定される。	
certStatus	CertStatus(CHOICE)		
unknown	[2] UnknownInfo	(注2)	

thisUpdate	GeneralizedTime	この電文を作成した日時が設定される。
singleExtensions	[1]	
-	Extensions	
(certPathStatus)	Extension	
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.8
critical	BOOLEAN	TRUE
extnValue	OCTET STRING	
-	INTEGER	証明書検証結果が設定される
(certPath)	Extension (注 3)	
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.9
extnValue	OCTET STRING	
-	CERTIFICATE	認証パスで構築された証明書が設定される
(revocationList)	Extension (注 3)	
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.10
extnValue	OCTET STRING	
-	CRL	認証パス検証で利用したCRL/ARLが設定される。
(OCSPResponse)	Extension (注 3)	
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.11
extnValue	OCTET STRING	
-	OCSPResponse	認証パス検証で利用したOCSPResponseが設定される。
(mappedPolicy)	Extension (注 3)	
extnId	OBJECT IDENTIFIER	1.2.392.200010.10.12
extnValue	OCTET STRING	
-	OBJECT IDENTIFIER	検証結果として有効なポリシーのOIDが設定される。
responseExtensions	[1]	
-	Extensions	
(nonce)	Extension	
extnId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1.2
extnValue	OCTET STRING	
-	Nonce	Request で設定した値が設定される。
signatureAlgorithm	AlgorithmIdentifier	
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.5
parameters	NULL	-
signature	BIT STRING	CVS の署名が設定される。
certs	[0]	
-	SEQUENCE OF Certificate	CVS の証明書が設定される。

注1 印のあるフィールドは必ず設定される。 印のあるフィールドは、証明書検証要求で指定された場合にのみ設定される。 印のあるフィールドには、上欄にフィールドを設定したときは、必ず設定される。

注2 必ず unknown が設定される。

注3 複数設定される場合は、同じOIDを持つExtensionが複数設定される。

5. 署名アルゴリズム

証明書検証サーバがサポートする証明書の署名アルゴリズムは以下の通りである。

- sha1WithRSAEncryption
- dsaWithSha1
- md5WithRSAEncryption

また、証明書検証サーバは、sha1WithRSAEncryption により署名を行い、レスポンスデータには証明書検証サーバ証明書が含まれる。