電子認証システムのシステム更改に伴う「電子証明書の方式等に関する件（告示）」の変更についての補足説明資料

法務省民事局商事課
デジタル庁

２０２５年０５月１６日版

概要説明

電子認証登記所システムのシステム更改概要

■ 令和８年３月中のシステム更改を予定しています。
■ システム更改に伴う変更点は以下のとおりです。

１．電子証明書の有効性確認方法として失効リスト（CRL/ARL）が追加されます。

現行：OCSPのみ

次期：OCSPに加えて、失効リスト（CRL/ARL）での検証も可能

２．電子証明書の証明書ポリシーが一部変更となります。

現行：KeyUsage※１ および CRLDP※２ 設定なし

次期：①被証明者の電子証明書にKeyUsageを追加

②CRLDP※２を追加（項番１に関連）

上記更改に当たり、「電子証明書の方式等に関する件（告示）」の変更を予定しているため、現時点における変更案（1.1版）を公開しています。
（変更案資料内の着色部は現行からの変更箇所です。）

・変更案（「電子証明書の方式等に関する件（告示）」）【ＰＤＦ】
・変更案（「電子証明書の方式等に関する件（告示）」（補足））【ＰＤＦ】（ 政府認証基盤（GPKI）ブリッジ認証局との相互認証に関する仕様を示すもの）

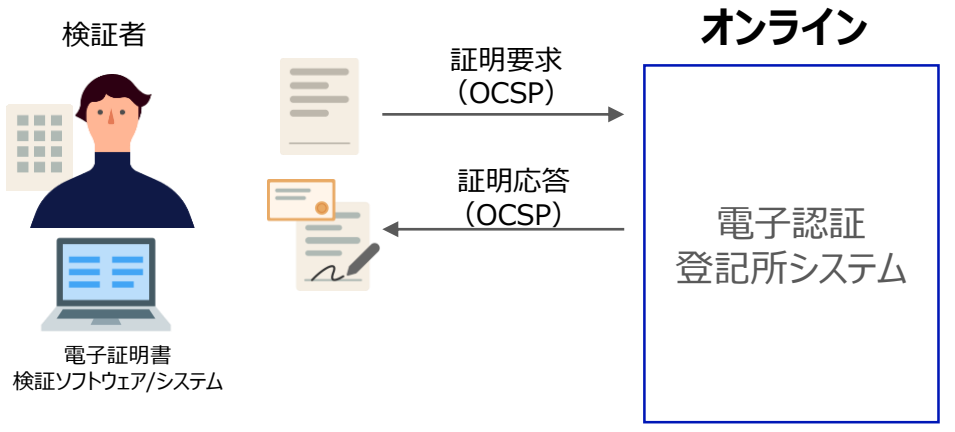
※１： KeyUsageは、電子証明書に含まれる拡張フィールドの一つで、公開鍵がどのような用途に使われるかを指定するものです。
※２： CRLDP（CRL Distribution Point）は、電子証明書に含まれる拡張フィールドの一つで、証明書失効リスト（CRL）のダウンロード先（ディレクトリ名/URL）を指定するものです。

電子証明書の有効性確認方法の追加（失効リスト）について

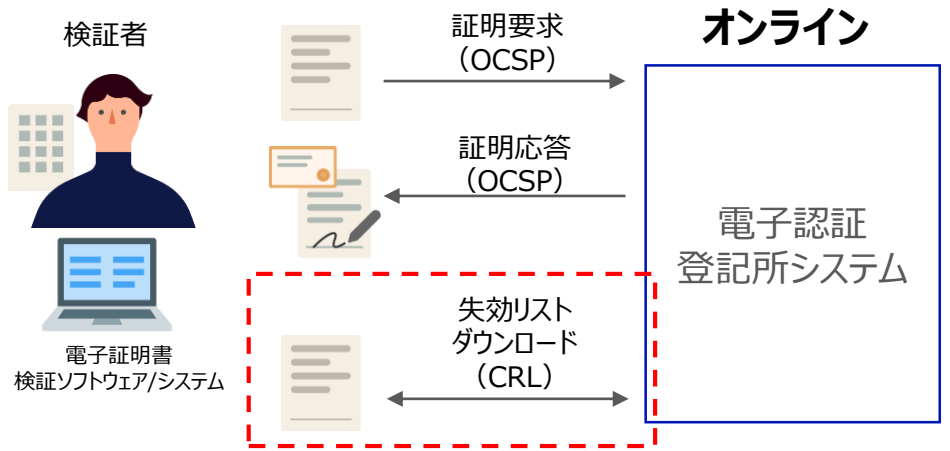
■ 電子証明書は、有効性※を確認しなければ信用できません。
※証明期間を経過していないか、記録された事項（会社の商号、本店、代表者の資格・氏名等）に変更が生じていないか等

■ 現状では電子証明書の有効性の確認方法としてOCSP方式のみが提供されていますが、
次期電子認証登記所システムでは、より世間一般的に利用されているCRL方式も選択できるようになります。

現行の商業登記電子証明書
有効性の確認方法：OCSP方式のみ



新商業登記電子証明書（令和8年度末以降）
有効性の確認方法：OCSP方式またはCRL方式



1 **既存のOCSP方式では、個別の電子証明書が有効であることの証明を要求** ←従来通り利用可能です

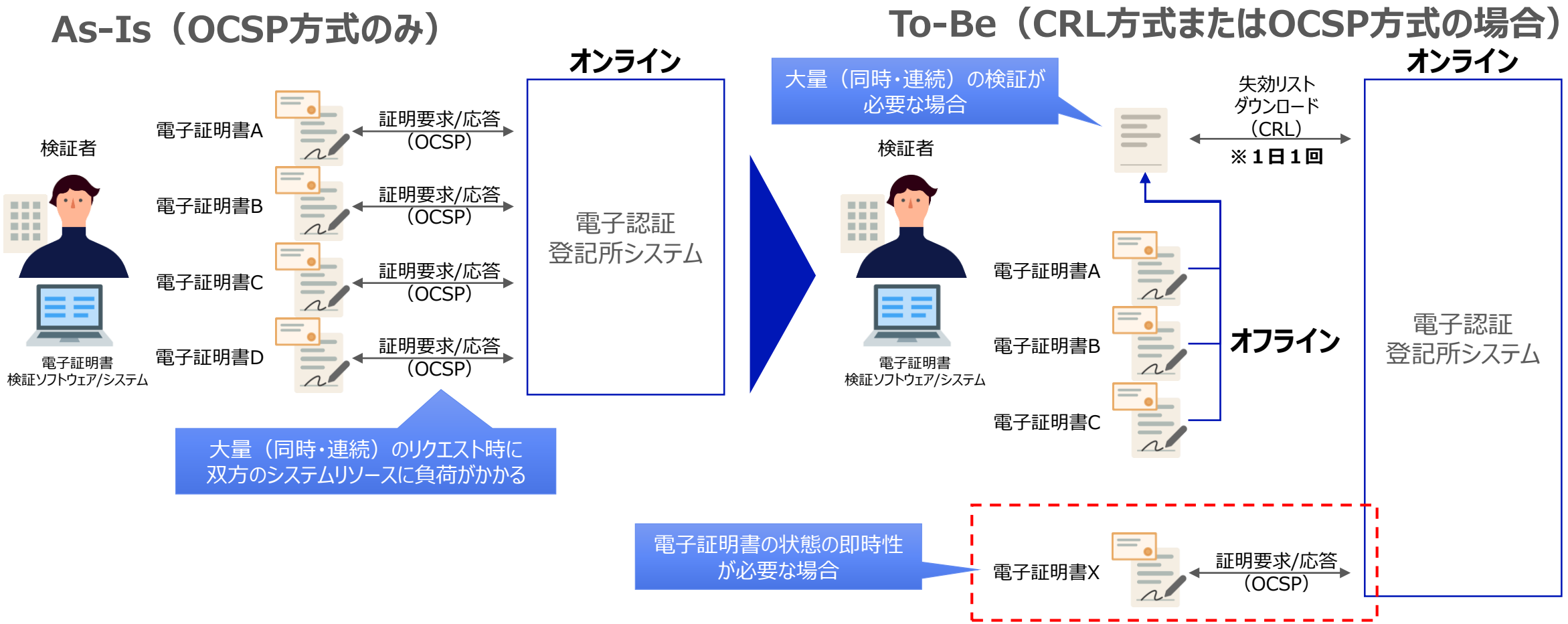
１つの証明要求に対して１つの証明応答となるため複数の電子証明書を検証する場合は複数回要求する必要がある

2 **新たなCRL方式では、失効している電子証明書のリスト（失効リスト）を取得できる**

失効リストに電子証明書のシリアル番号が掲載されていないことで有効であることが証明できる（CRL取得後にオフライン検証可）

電子証明書の有効性の確認方法の使い分けのお願い

- 電子認証登記所のシステムは、これまでの利用実績に基づき、必要十分なリソースを備えておりますが、リソースには限りがあり、利用者の皆様に公平かつ安定したシステム環境を提供するため、大量（同時・連続）の電子証明書の有効性の確認を行う場合は、CRL方式（※次ページ以降の補足参照）を優先しての利用を検討してください。
- 一方、電子証明書の状態の即時性が重要となる場合においては、従来通りOCSP方式を利用してください。
- ご理解とご協力をお願いいたします。



補足説明

補足説明①

【用語】

- OCSP: Online Certificate Status Protocol
  - 電子証明書の状態を取得するための通信プロトコル。電子証明書の状態（有効・無効・保留など）が応答され、応答には登記官の電子署名が付与される。
- CRL　: Certificate Revocation List
  - 電子証明書の有効期間中に、登記内容の変更や 秘密鍵の危殆化等の事由により失効された電子証明書のリスト。このリストには登記官の電子署名が付与される。有効期間 48 時間の CRL が 24 時間ごとに発行され、24時間365日公開サーバ上から取得できる。

【CRL方式を推奨する理由】

- 民間の認証局（認定認証事業者）や行政の認証局（GPKI・LGPKI）など、電子証明書の有効性を検証する方式としてCRL方式が一般的です。GPKI・LGPKIでは、別途、OCSP方式を拡張した証明書検証サーバ（CVS）を提供しており、用途に応じて使い分けることが可能です。
- OCSP方式は、個別の電子証明書の有効性の確認を行うたびに、電子認証登記所のサーバとの通信が発生するため、大量の電子証明書を検証する場合、双方に通信負荷がかかります。
- 一方、CRL方式は、電子証明書の有効期間内に失効（無効・保留）した全ての発行済み電子証明書が掲載されたリスト（CRL）を１日１回程度、電子認証登記所のサーバから取得するだけで、電子証明書の有効性の確認をオフラインで行うことができるため、双方の通信負荷を低減することが可能です。

補足説明②

【電子証明書を検証するソフトウェア／システムからの利用方法】

1. CRL方式の場合（優先・推奨）
   - CRLをダウンロード（１日１回程度）して、一定時間キャッシュして利用する。
   - 可能であれば、CRLが取得できない場合に備えOCSP方式も使えるようにしておく。
2. OCSP方式の場合
   - 大量（同時・連続）の利用を控え、ある程度間隔を空けて利用する。
   - 同じ電子証明書の有効性を複数回確認する必要がある場合は、OCSP応答を一定時間キャッシュして再利用することを検討する。
   - 可能であれば、OCSP応答が取得できない場合に備えCRL方式も使えるようにしておく。

補足説明③

【参考情報（ＵＲＬ）】

1. 商業登記に基づく電子認証制度（法務省）
2. 電子証明書の方式等に関する件（告示）（法務省）
   - 告示の変更案（証明書プロファイル等）が掲載されています（１．１版）
3. 商業登記法（e-Gov法令検索）
4. 商業登記規則（e-Gov法令検索）

商業登記に係る電子認証登記所システムの今後の予定（2025年5月現在）

■ 令和5年度末より次期電子認証登記所システムの調達と開発が開始されており新システムが令和７年度末に稼働予定。
■ 2028年末までに新暗号対応の次世代電子認証登記所システムの稼働を計画しており仕様を検討中。

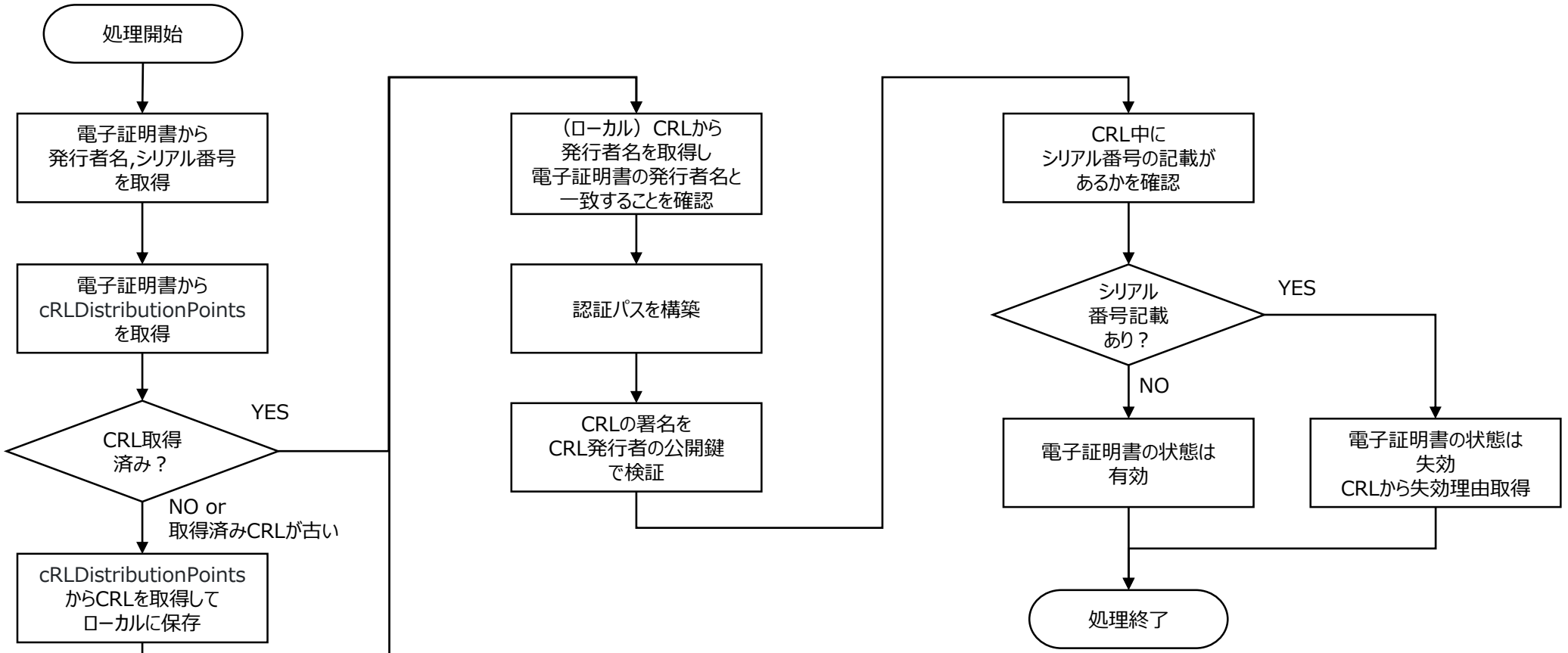
CRLによる電子証明書の検証方法について

■ 電子証明書のCRLによる失効検証方法の概略は以下のとおりです。
■ 詳細はRFC 5280（6章）を参照してください。

1. 電子証明書パス検証（RFC 5280 6章参照）
    1. 電子証明書の署名の検証
    2. 電子証明書の有効期間の検証
    3. 電子証明書の失効検証　→CRLまたはOCSPによる失効検証
    4. 電子証明書の発行者名の検証
2. CRLを用いた失効検証（RFC 5280 6.3章参照）
    1. CRLの取得（ローカルキャッシュの利用、またはオンライン取得＆ローカルキャッシュの更新）
    2. CRLの発行者名の確認
    3. CRLの発行者の認証パスの構築（トラストアンカが電子証明書のトラストアンカと同じであること）
    4. CRLの署名の検証（CRL発行者の公開鍵で署名を検証）
    5. CRL中のシリアル番号を検索

CRL検証の流れ

■ 電子証明書のCRLによる失効検証方法の一例（フローチャート）を以下に示します。
■ 詳細はRFC 5280（6章）を参照してください。



【cRLDistributionPoints】
・DistributionPoint: DNで指定（GPKI統合リポジトリからcertificateRevocationList.crlを取得）
・DistributionPoint: URLで指定（https://crca1.moj.go.jp/certificateRevocationList.crl）

■ 電子証明書のCRLによる失効検証方法の実装例（Java ソース）

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.BigInteger;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
import java.security.cert.CertificateException;
import java.security.cert.X509CRL;
import java.security.cert.CRLException;

public class CRLVerify {
 public static void main(String[] args) throws IOException {
  try{
   // X509 PKI管理ファクトリの生成
   CertificateFactory factory = CertificateFactory.getInstance("X.509");
   // CRLファイルの読み込み
   InputStream ins = new FileInputStream( new File("./fullCRL.crl") );
   // CRLのインスタンス生成
   X509CRL crl = (X509CRL) factory.generateCRL(ins);
   // チェック対象の証明書の読み込み
   InputStream ins2 = new FileInputStream( new File("./client.cer") );
  // 証明書のインスタンス生成
  X509Certificate cert = (X509Certificate) factory.generateCertificate(ins2);
  //証明書のシリアル番号の取得
  BigInteger serial = cert.getSerialNumber();
  String hexserial = serial.toString(16); //16進数表現
  //証明書の発行者名(DN)の取得
  X500Principal issuer_dn = cert.getIssuerX500Principal();
   // 証明書が失効されているかを判定する
   boolean revoke = crl.isRevoked(cert);
   if(revoke == true){
    System.out.println("失効！シリアル番号: " + hexserial);
   } else {
    System.out.println("有効！シリアル番号: " + hexserial);
   }
  } catch(CertificateException e){
   e.printStackTrace();
  } catch(CRLException e){
   e.printStackTrace();
  }
 }
}
```

※証明書からCRLDPを取得・保存する方法は省略
→CRLDPの取得はbouncy castleの利用が便利
※CRLのパス構築・署名を検証する方法は省略
→既存の証明書のパス構築ロジックを利用
※登記官の証明書の更新時には、CRLの署名に使用される登記官の証明書も更新されるため注意が必要