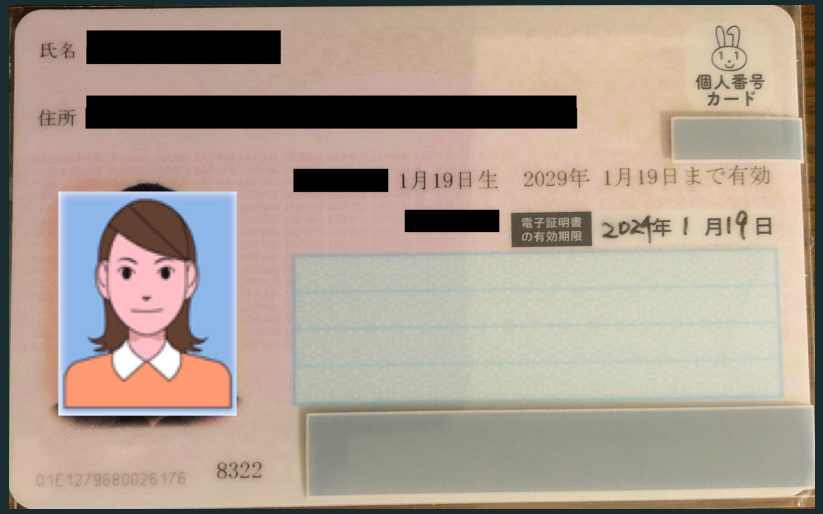
マイナンバーカードで署名する

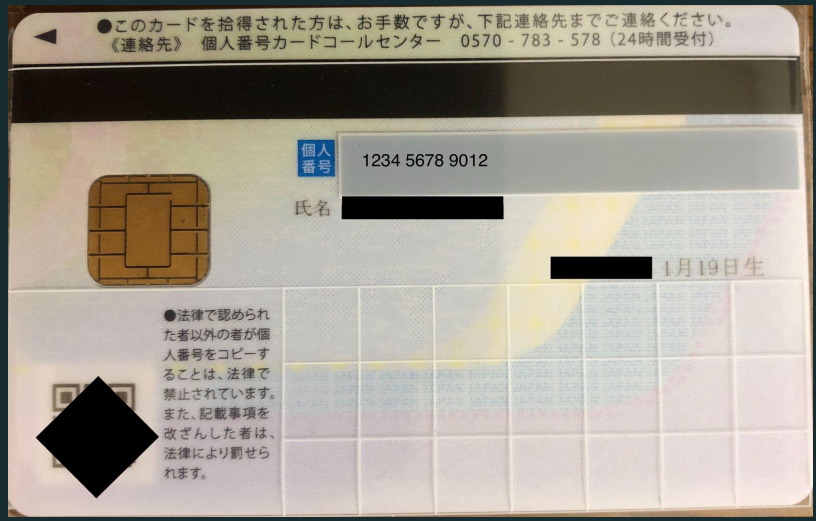
@tex2e

2021/8/9

マイナンバーカード

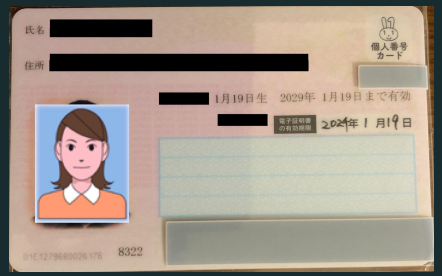
個人番号が書かれたICカード。内部に認証用・署名用の証明書 (公開鍵)と秘密鍵を持つ

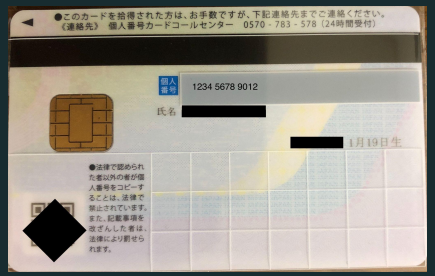




↑※QRコードにも個人番号がある点に注意

マイナンバーカードでできる処理

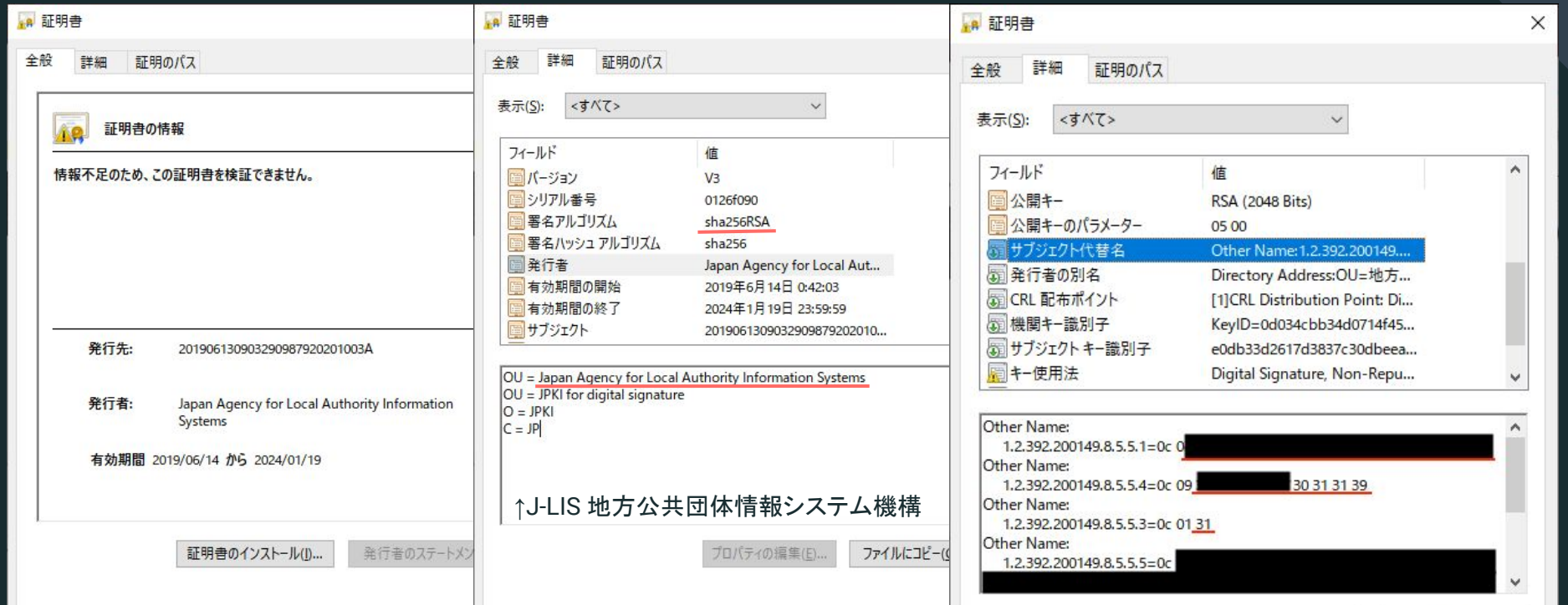




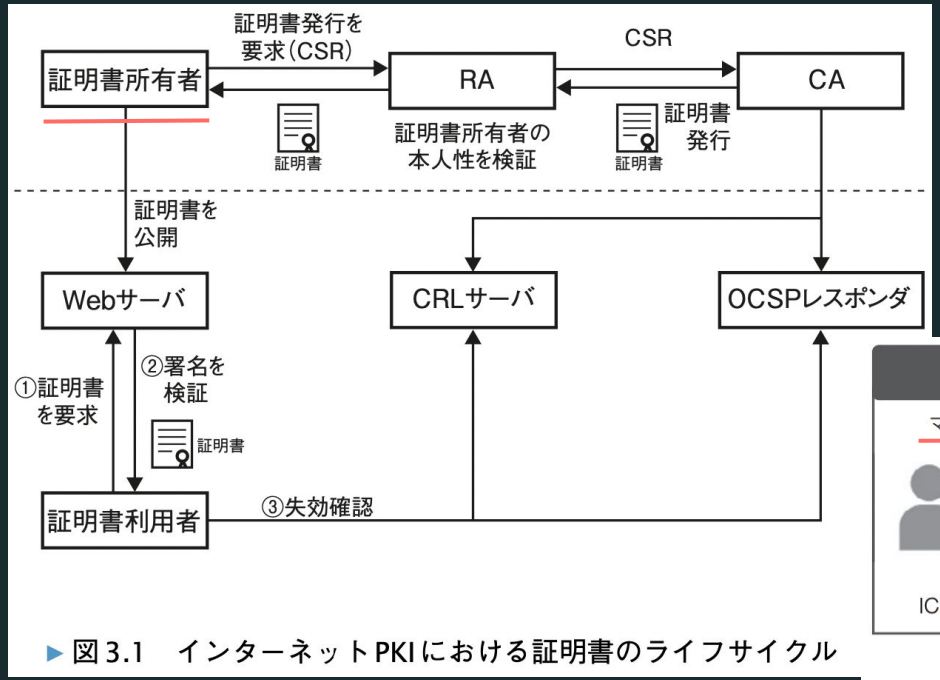
＜処理＞	＜出力＞
認証用証明書の取得	（DER形式のデータ）
認証用秘密鍵による署名	（データに対する署名）
署名用証明書の取得	（DER形式のデータ）
署名用秘密鍵による署名	（データに対する署名）
個人番号の取得	（文字列）
基本4情報の取得	（UTF-8文字列）

認証用証明書・署名用証明書

署名用証明書のみ基本 4情報(名前, 生年月日, 性別, 住所)あり↓



公開鍵基盤



▶ 図 3.1　インターネット PKI における証明書のライフサイクル

←HTTPS証明書

↓認証用・署名用証明書

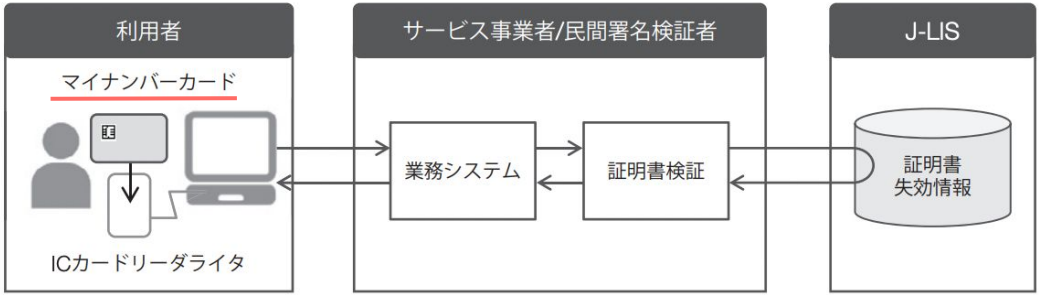


図-2　証明書の有効性検証の仕組み

左図：プロフェッショナルSSL/TLS, p. 62
https://www.lambdanote.com/collections/ssl-tls

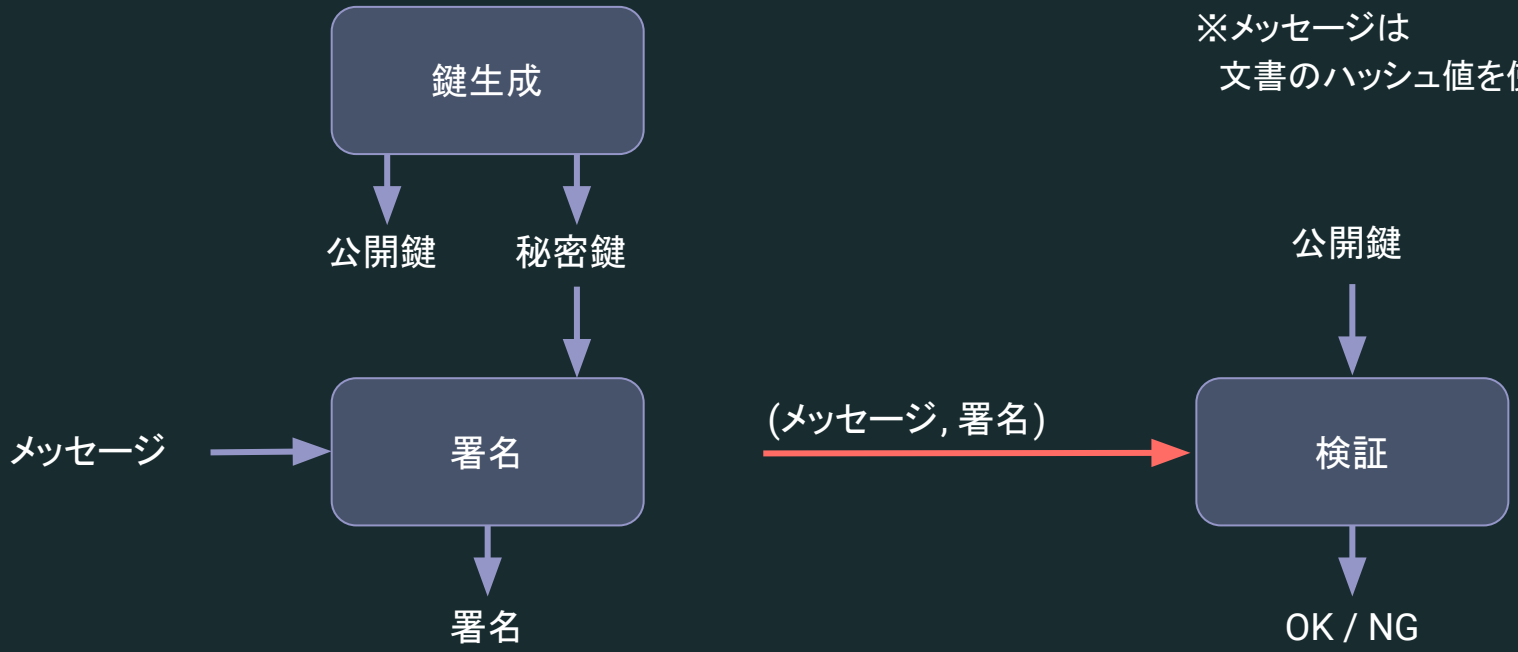
右図：富士通 2017-7月号 マイナンバー特集マイナンバーカードの技術仕様と利活用方式p.62
https://www.fujitsu.com/jp/about/resources/publications/magazine/backnumber/vol68-4.html

デジタル署名

改ざん・なりすまし・否認を防ぐための技術

※RSA署名のため、
検証鍵＝公開鍵
署名鍵＝秘密鍵

※メッセージは
文書のハッシュ値を使う



マイナンバーカードで署名する手順

公的個人認証AP を選択

署名用PIN を選択

署名用PIN送信

認証

署名用秘密鍵 を選択

署名対象データ送信

署名データ


```
root/
+-- 公的個人認証AP/
    +-- 署名用証明書
    +-- 署名用PIN
    +-- 署名用秘密鍵
```

通信プロトコル (APDU)

SELECT FILE

00 A4 04 0C 0A D3 92 F0 00 26 01 00 00 00 01 →

00 A4 02 0C 02 00 1B →

VERIFY

00 20 00 80 06 31 32 33 34 35 36 →

← **90 00**

00 A4 02 0C 02 00 17 →

COMPUTE DIGITAL SIGNATURE

80 2A 00 80 33 (...署名対象データ...) 00 →

← **(...署名データ...) 90 00**

←SHA256RSA署名に必要な
データの構造は「DigestInfo」

SHA256RSA 署名に必要な DigestInfo

マイナンバーカードに署名させるデータを送るためのデータ構造（ RFC, ANS.1参照）

(1) RFC 2315 (PKCS #7: Cryptographic Message Syntax Version 1.5)

```
DigestInfo ::= SEQUENCE {
  digestAlgorithm DigestAlgorithmIdentifier,
  digest Digest }

Digest ::= OCTET STRING
```

```
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
```

(2) RFC 5280 (Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile)

```
AlgorithmIdentifier  ::=  SEQUENCE  {
     algorithm               OBJECT IDENTIFIER,
     parameters              ANY DEFINED BY algorithm OPTIONAL  }
```

以上をまとめると:

```
DigestInfo ::= SEQUENCE {
  SEQUENCE {
    algorithm   OBJECT IDENTIFIER,
    parameters  ANY DEFINED BY algorithm OPTIONAL
  }
  digest  OCTET STRING
}
```

ANS.1 とSHA256のOID

```
DigestInfo ::= SEQUENCE {
  SEQUENCE {
    algorithm   OBJECT IDENTIFIER,
    parameters  ANY DEFINED BY algorithm OPTIONAL
  }
  digest  OCTET STRING
}
```

algorithm: RFC 5754 (Using SHA2 Algorithms with Cryptographic Message Syntax)

```
id-sha256 OBJECT IDENTIFIER ::= {
  joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101)
  csor(3) nistalgorithm(4) hashalgs(2) 1 }
```

```
id-sha256: 30 0b 06 09 60 86 48 01 65 03 04 02 01
```

parameters: NULL必須

05 00　← タグがNULL(0x05)、長さが0x00のデータ、内容はなし（ANS.1参照）

digest: 文書 (ファイル) のハッシュ値

以上でDERエンコードするとDigestInfoは:

```
30 31 30 0B 06 09 60 86 48 01 65 03 04 02 01 05
00 04 20 22 D6 28 B5 3B C6 B3 56 F5 91 3E 98 C5
A3 BC 8A E1 A5 BE 91 C2 91 68 02 35 8E 0E C2 BC
FE 71 E7
```

(Tag, Len, Value)

Tag:
0x30 : SEQUENCE
0x05 : NULL
0x04 : OCTET STRING

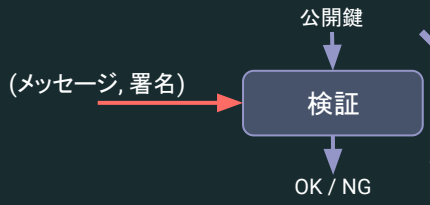
署名データ

SHA256RSA (SHA256 with RSA Encryption)

メッセージ→SHA256ハッシュ値(32byte)→RSA暗号化→署名(256byte)

```
0000000 6d 79 3e d3 0a d5 25 84 58 63 66 06 9c 0e 87 ad
0000010 87 82 65 16 6e 87 b6 35 5a 0a a2 6f 09 8b f6 43
0000020 45 9c 4b 73 65 93 2c 3b b9 7b 19 56 54 d7 0b 63
0000030 6c 13 e4 9d 0c b9 f5 1f d4 7b 03 88 95 73 6b d5
0000040 1a 27 fc 70 05 9e 62 45 69 22 6b 8b 3a 93 a4 6c
0000050 24 8a 33 be d3 84 3e d1 6e f6 f7 e3 0b da 46 7b
0000060 10 be 97 1a 28 81 8c 0e 75 9d 25 8b 24 1e 8d ec
0000070 28 d2 35 9d c6 e5 02 ad cd 77 2c 59 1a 47 62 91
0000080 8f e6 eb 6f 52 ed c9 17 b1 a6 43 81 9a b6 60 e3
0000090 ff 94 13 94 03 52 87 c6 bf c5 8c ed ed 17 ed 75
00000a0 da 6e d1 54 45 6a 20 24 ed 4b 41 dc d8 4f 16 be
00000b0 a5 0e e5 7a e4 29 22 60 64 10 3d b4 1a 23 bf 5e
00000c0 45 86 ce 32 18 d9 40 49 b7 43 af b3 44 41 63 40
00000d0 8a a0 94 d1 6b c5 5d 49 e0 7d 83 54 70 5c a0 6c
00000e0 02 7f ae 83 20 ef a2 5c 1c af e2 d5 2e cf ac 1e
00000f0 8b a3 95 76 fa cc 82 a0 3b 82 46 d5 10 77 9a aa
0000100
```

署名データの検証



検証者には (文書, 文書のハッシュ, 署名データ) と 証明書 が与えられる

1. マイナンバーカードから署名用証明書を取り出す
2. 署名用証明書から公開鍵を取り出す

```
openssl x509 -in SignCert.der -inform der -out SignCert.pem -outform pem
openssl x509 -pubkey -noout -in SignCert.pem -out SignPub.pem
```

3. OpenSSLに公開鍵と文書と署名データを渡して検証する

```
openssl dgst -verify SignPub.pem -signature TARGET.pdf.sig TARGET.pdf
```

4. 「Verified OK」で検証成功

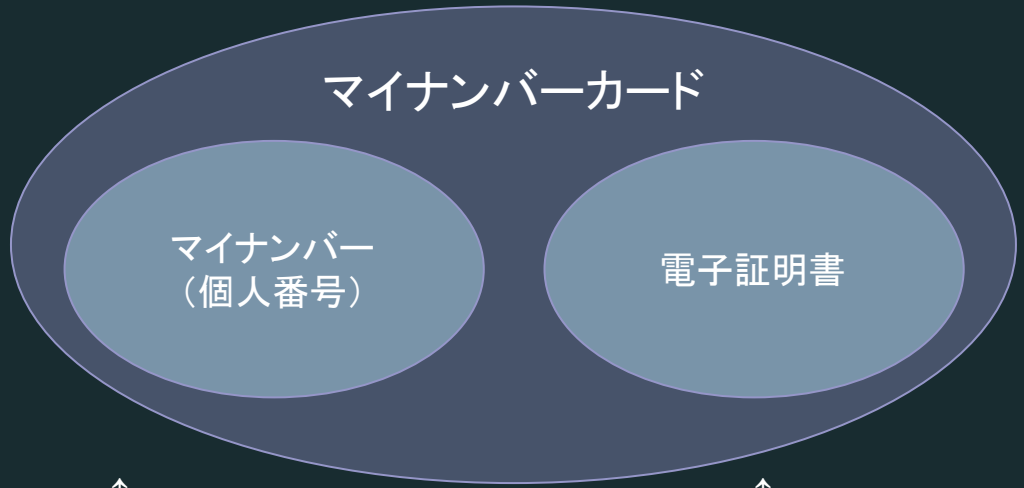
より詳細な解説

マイナンバーカードとAPDUで通信して署名データ作成 | 晴耕雨読
https://tex2e.github.io/blog/protocol/jpki-mynumbercard-with-apdu

ANS.1 のタグ一覧 | 晴耕雨読
https://tex2e.github.io/blog/protocol/ans1-tags

おわりに

マイナンバーカードは民間利用が可能です！どんどん使いましょう！



↑
個人番号は社会保障、税、
災害対策にしか使えない

↑
認証用証明書、署名用証明書は
民間でも利用可能

Thank you!