Windows用(.exe,.cab,.dll)形式編

改版履歴			
版数	日付	内容	担当
V.1.0	2015/4/1	初版	NII
V.2.0	2018/2/26	動作環境の変更に伴う修正	NII
V.2.1	2018/7/9	タイムスタンプ利用手順の追加	NII
V.2.2	2020/6/4	中間CA証明書のURLとリポジトリのURLの変更	NII
V.2.3	2020/12/22	中間CA証明書のURL変更	NII
V.2.4	2020/12/24	鍵長の変更	NII

目次

1. コード署名用証明書の利用

1-1. 前提条件

OpenSSLコード署名用証明書を使用する場合の前提条件について記載します。適宜、コード署名用証明書をインストールする利用管理者様の環境により、読み替えをお願いします。
（本マニュアルではVisual Studio 2015での実行例を記載しております。）
コマンドプロンプト上で実行するコマンドは、「＞」にて示しています。

前提条件
1. OpenSSLがインストールされていること

CSR作成時は既存の鍵ペアは使わずに、必ず新たにCSR作成用に生成した鍵ペアを利用してください。更新時も同様に、鍵ペアおよびCSRを新たに作成してください。鍵ペアの鍵長はRSA 3072bitまたは 4096bitにしてください。

1-2. PKCS#12ファイルの作成

本章ではPKCS#12ファイルの作成方法について記述します。

1-2-1. 事前準備

事前準備として、「ルートCA証明書」、「中間CA証明書」、「コード署名用証明書」を取得してください。

事前準備

1. 「.証明書の申請から取得まで」で受領したコード署名用証明書を任意の名前で任意の場所に保存してください。
2. 「ルートCA証明書」と「中間CA証明書」を準備し、この2つを連結させます。下記URLより、リポジトリへアクセスしてください。

   「中間CA証明書」を下記リポジトリより取得してください。
   セコムパスポート for Member 2.0 PUB リポジトリ：
   https://repo1.secomtrust.net/spcpp/pfm20pub/index.html

   リポジトリ内にある「証明書の種類」より中間CA証明書を取得してください。
   https://repo1.secomtrust.net/spcpp/pfm20pub/codecag2/CODECAG2.cer

   次に、「ルートCA証明書」を下記リポジトリより取得してください。
   SHA-256 Security Communication RootCA2 リポジトリ：
   https://repository.secomtrust.net/SC-Root2/index.html

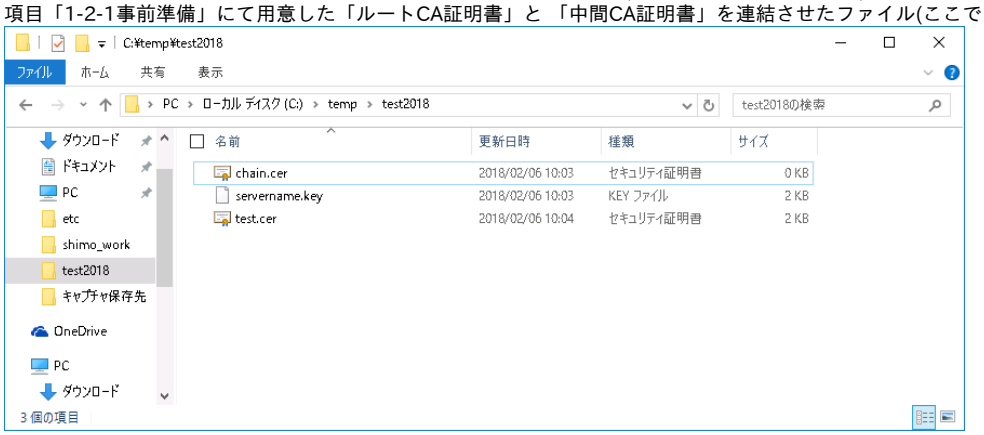
   SHA-256 Security Communication RootCA2 証明書：
   https://repository.secomtrust.net/SC-Root2/SCRoot2ca.cer

1-2-2. PKCS#12ファイルの作成

本項目ではWindowsOS上で任意のフォルダにPKCS#12ファイルを作成する方法を記述します。
以下は、例としてWindows10上での作成方法を記載します。

PKCS#12ファイルの作成

1. 任意のフォルダ(ここではC:\temp\test2018とします)にて以下の3つのファイルを用意してください。
   a. 項目「鍵ペアの生成」にて作成した鍵ペアのファイル(servername.key)
   b. 項目「証明書の申請から取得まで」にて取得したコード署名用証明書(ここではtest.cerとします)
   c. 項目「1-2-1事前準備」にて用意した「ルートCA証明書」と 「中間CA証明書」を連結させたファイル(ここではchain.cerとします)



2. CAfile に指定する証明書をDER形式からPEM形式に変換します。

```
・Security Communication RootCA2の場合
openssl x509 -inform der -in SCRoot2ca.cer -outform pem -out SCRoot2ca.cer

・中間CA証明書SHA-256の場合
openssl x509 -inform der -in nii-odcacssha2.cer -outform pem -out nii-odcacssha2.cer
```

3. コマンドプロンプト上にて上記で取得した「ルートCA証明書」と「中間CA証明書」を以下のコマンドにより、連結させてください。中間CA証明書の下部にルートCA証明書が併記されるファイルとなります。

```
＞ type (中間CA証明書のパス) (ルートCA証明書のパス) > (出力するファイル名)
```

4. 連結したファイルがPEM形式になっていることを確認してください。
例）PEM形式の証明書

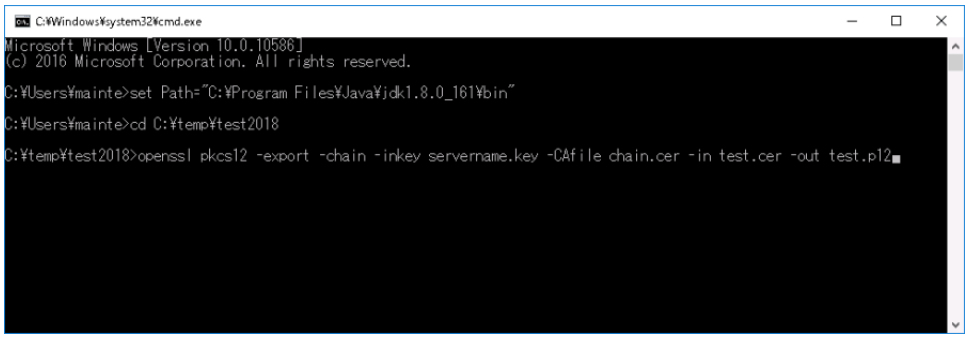
```
----BEGIN CERTIFICATE----
MIIEcTCCA1mgAwIBAgIIasWHLdnQB2owDQYJKoZIhvcNAQELBQAwbzELMAkGA1UE
BhMCSlAxFDASBgNVBAcMC0FjYWRlbWUtb3BzMSowKAYDVQQKDCFOYXRpb25hbCBJ

bnN0aXR1dGUgb2YgSW5mb3JtYXRpY3MxHjAcBgNVBAMMFU5JSSBPcGVyYXRpbmcg
Q0EgLSBHMjAeFw0xNTAzMTIwMTA4MDJaFw0xNzA0MTEwMTA4MDJaMHAxCzAJBgNV

（中略）
LmeW0e/xkkxwdmKv5y5txLIFcp53AZl/vjn3BHp42PFkkTISEmAUiCtQ2A25QDRR
RG33IaC8E8Tl/SnOA8h95XQtGWm47PrIjXyYtIe0rFousbpIoW8MZw4gDXVQ3485
XEftqwwIMcLNxttJ6i6f9XVyPMRhHy9rdDPseHiXayxcBxJMuw==
----END CERTIFICATE----
```

5. コマンドプロンプトを開き、ファイルのある任意のフォルダ(ここではC:\temp\test2018)へ移動します。

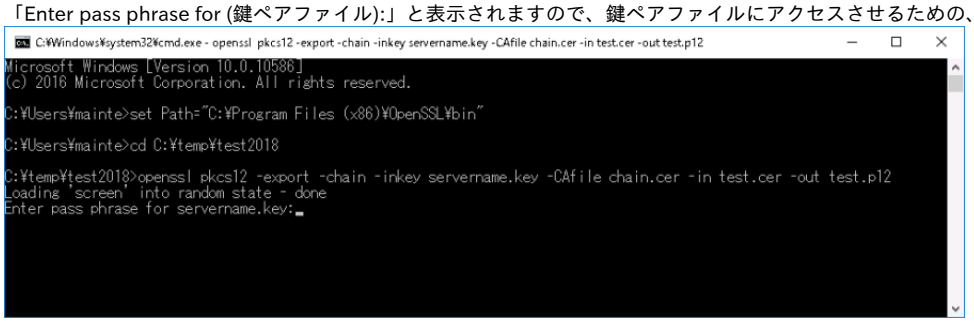
```
＞ set Path=(OpenSSLインストールディレクトリ)\bin
※OpenSSLインストールディレクトリをプログラムを探すディレクトリに指定します。
＞ cd (作業ディレクトリ) ←作業ディレクトリ
```

6. 移動後、以下のコマンドを入力しPKCS#12ファイルを作成してください。

```
＞ openssl pkcs12 -export -chain -inkey (鍵ペアのファイル名) -CAfile (ルートCA証明書と中間CA証明書を連結させたファイル) -in (コード署名用の証明書ファイル名) -out (PKCS#12形式で出力するファイル名)
```

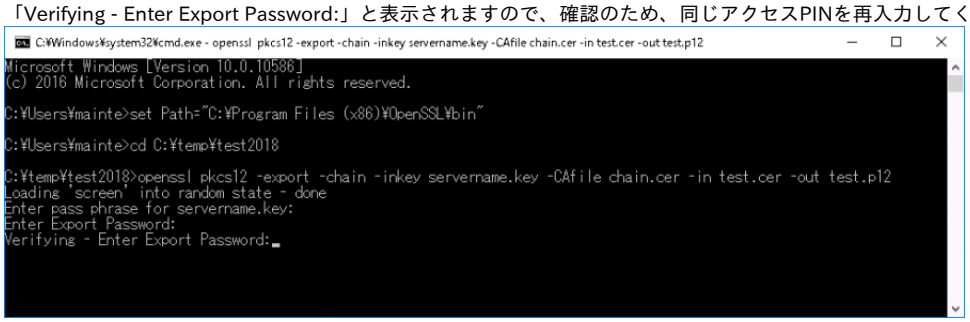


7. 「Enter pass phrase for (鍵ペアファイル):」と表示されますので、鍵ペアファイルにアクセスさせるための、パスフレーズを入力してください。

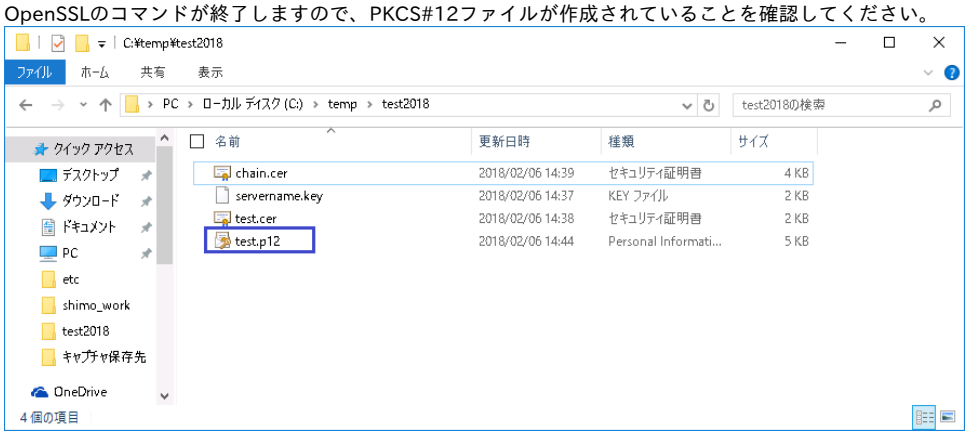


8. 「Enter Export Password:」と表示されますので、PKCS#12形式のファイルを保護するためのアクセスPINとして任意の文字列を入力してください。

9. 「Verifying - Enter Export Password:」と表示されますので、確認のため、同じアクセスPINを再入力してください。



10. OpenSSLのコマンドが終了しますので、PKCS#12ファイルが作成されていることを確認してください。

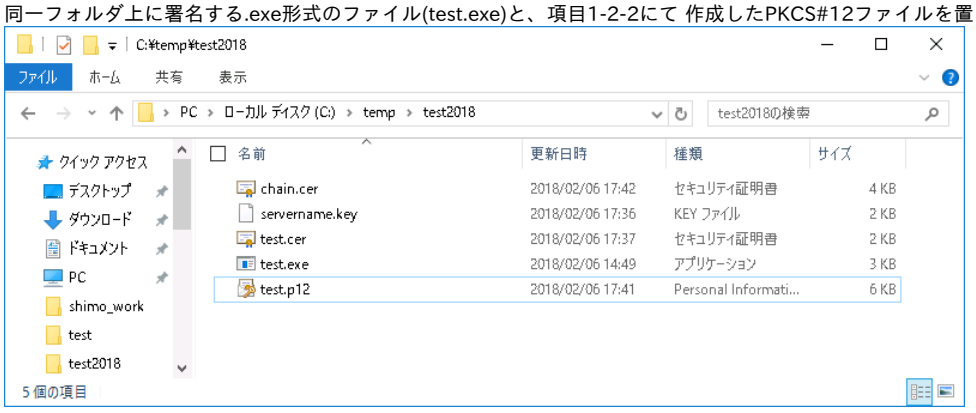


1-3. 署名

本章ではWindows用(.exe.cab.dll)形式のファイルにWindowsOS上にてデジタル署名をする方法について記述します。以下は、例としてWindows10上での作成方法を記載します。
署名のみを行う場合と署名に加えてタイムスタンプを付与する場合に分けて記載してあります。

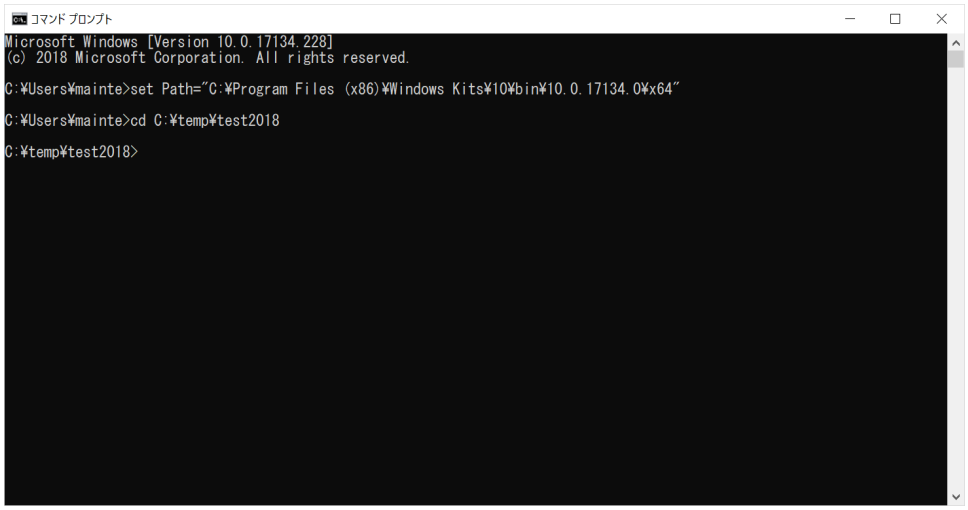
署名作業（併せてタイムスタンプを付与しない場合）

1. 同一フォルダ上に署名する.exe形式のファイル(test.exe)と、項目1-2-2にて 作成したPKCS#12ファイルを置きます。



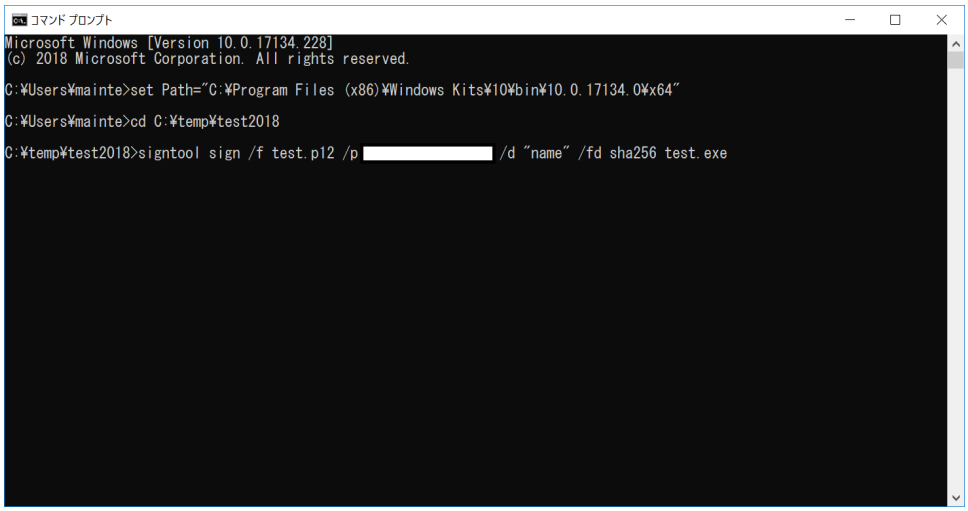
2. コマンドプロンプトを実行し、作業を行うフォルダへ移動します。

コマンド例：＞ set Path=**(Visual Studioインストールディレクトリ)**\bin
※Visual Studioインストールディレクトリをプログラムを探すディレクトリに指定します。
＞ cd **(作業ディレクトリ) ←作業ディレクトリ**



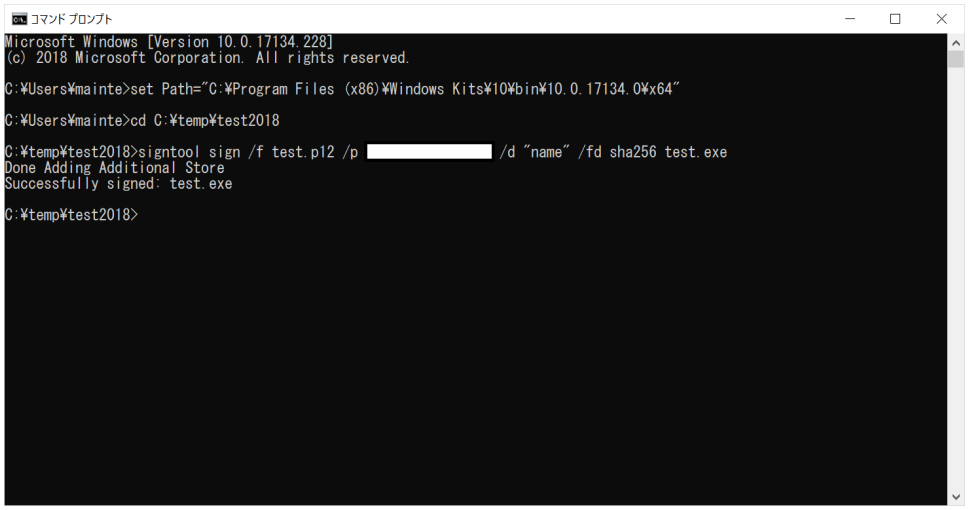
フォルダ移動後、署名したい.exe形式のファイル(ここではtest.exe)に対して以下のコマンドにて署名を実行してください。

signtool sign /f (PKCS#12ファイルへのパス) /p (PKCS#12ファイルのアクセスPIN) /d "(署名の説明)" /fd sha256 (署名したい.exe形式のファイルのパス)
※上記のコマンドはハッシュ関数sha256を用いて署名する場合です。オプション/fd sha256を指定しない場合、sha1にて署名しますので用途に応じて使い分けください。



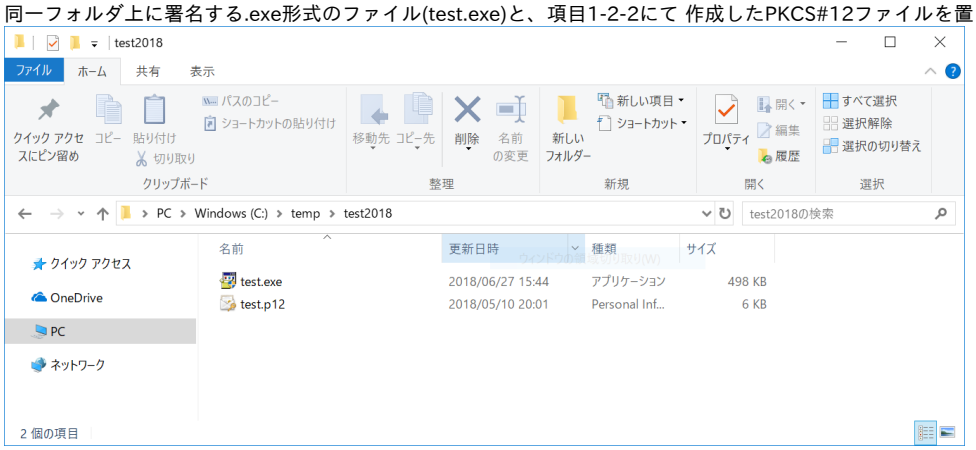
3. コード署名が成功すると下記のメッセージが表れます。

```
Successfully signed: (署名したexeファイル名)
```



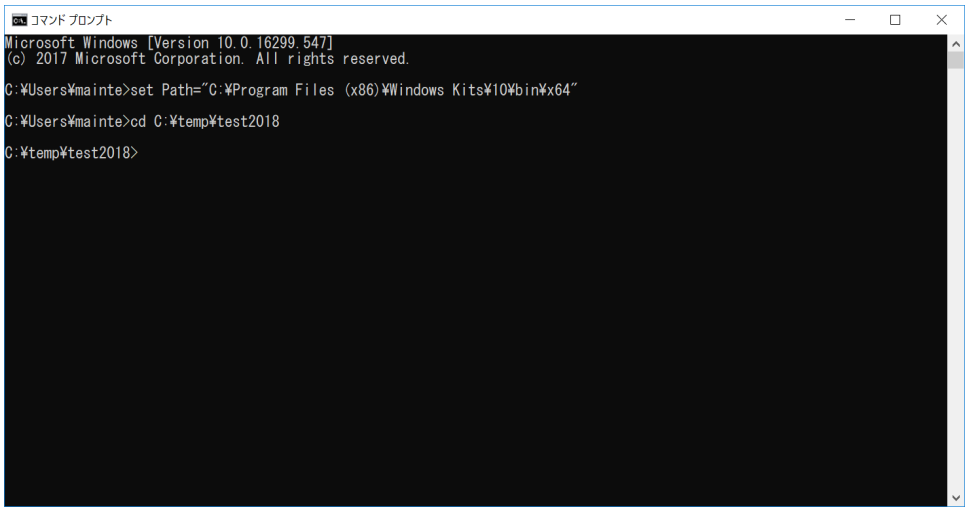
署名作業(併せてタイムスタンプを付与する場合)

1. 同一フォルダ上に署名する.exe形式のファイル(test.exe)と、項目1-2-2にて 作成したPKCS#12ファイルを置きます。



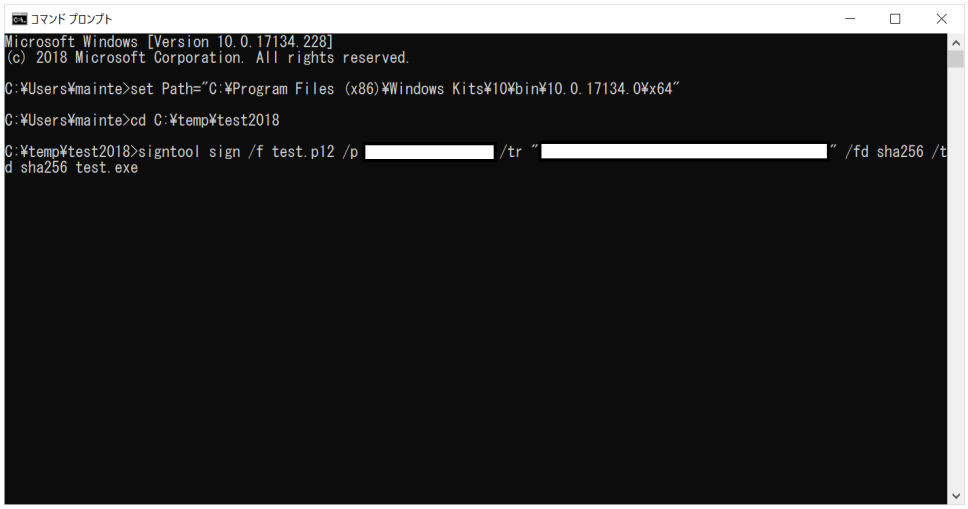
2. コマンドプロンプトを実行し、作業を行うフォルダへ移動します。

コマンド例：＞ set Path=**(Visual Studioインストールディレクトリ)**\bin
※Visual Studioインストールディレクトリをプログラムを探すディレクトリに指定します
＞ cd **(作業ディレクトリ)** **←作業ディレクトリ**



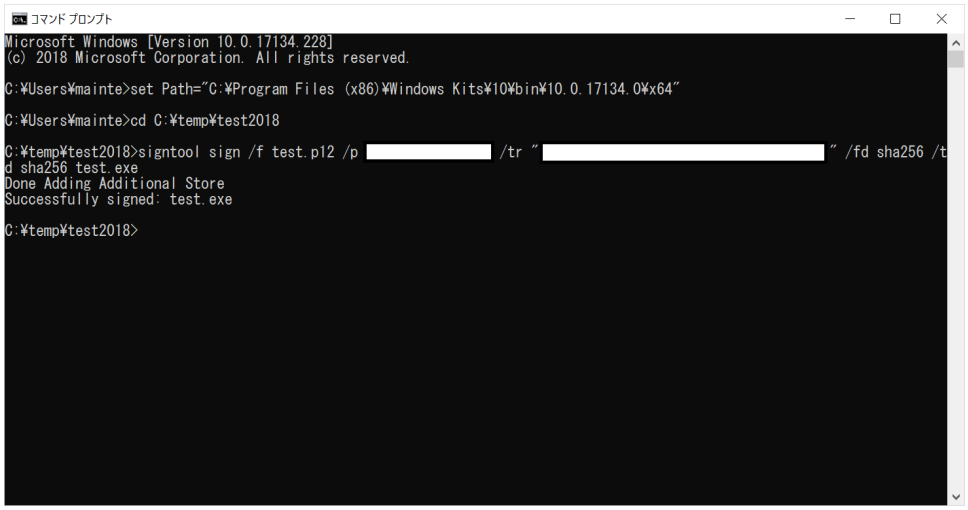
フォルダ移動後、署名したい.exe形式のファイル(ここではtest.exe)に対して以下のコマンドにて署名を実行してください。

signtool sign /f (PKCS#12ファイルへのパス) /p (PKCS#12ファイルのアクセスPIN) /tr （タイムスタンプURL※）/fd sha256 /td sha256 (署名するファイル)
※タイムスタンプURLに関しては登録担当者（各利用機関において、証明書発行のための審査と電子証明書自動発行支援システムの操作をする方）にお問い合わせください。
※タイムスタンプURLは「" (Double Quotation Marks) 」で囲んでください。
※上記のコマンドはハッシュ関数sha256を用いて署名する場合です。オプション/fd sha256を指定しない場合、sha1にて署名しますので用途に応じて使い分けください。
但し、ps1ファイルに対してタイムスタンプを付与する際は必ず/fd sha256を指定してください。sha1で署名した場合、正しくタイムスタンプが付与されません。



3. コード署名が成功すると下記のメッセージが表れます。

```
Successfully signed: (署名したexeファイル名)
```



1-4. コード署名確認作業

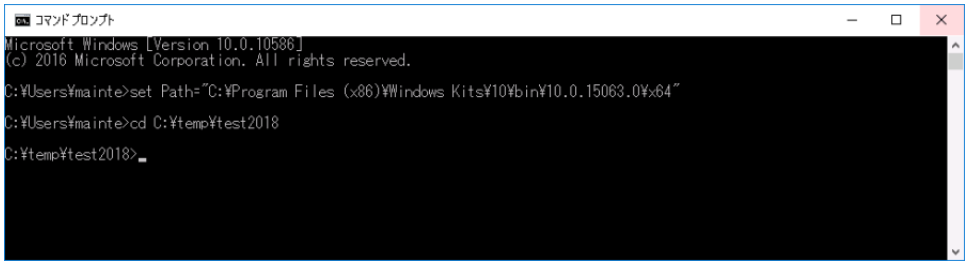
本章ではデジタル署名した.exe形式のファイルについて署名確認作業について記述します。
署名はsigntool、もしくはGUI操作にて確認可能です。
付与したタイムスタンプに関する詳細はGUI操作よりご確認ください。

1-4-1. signtoolを使用したコード署名確認

署名確認作業

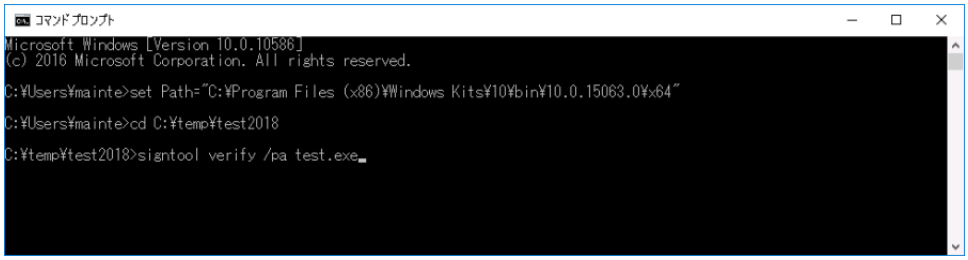
1. コマンドプロンプトを実行し、作業を行うフォルダへ移動します。

   コマンド例：＞ set Path=**(Visual Studioインストールディレクトリ)**\bin
   ※Visual Studioインストールディレクトリをプログラムを探すディレクトリに指定します
   ＞ cd **(作業ディレクトリ) ←作業ディレクトリ**

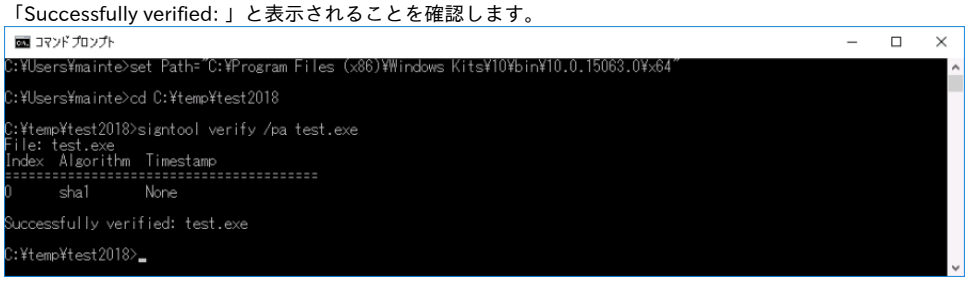


2. フォルダ移動後、以下のコマンドにて署名検証を実行してください。

   ＞ signtool verify /pa （署名したファイル名）



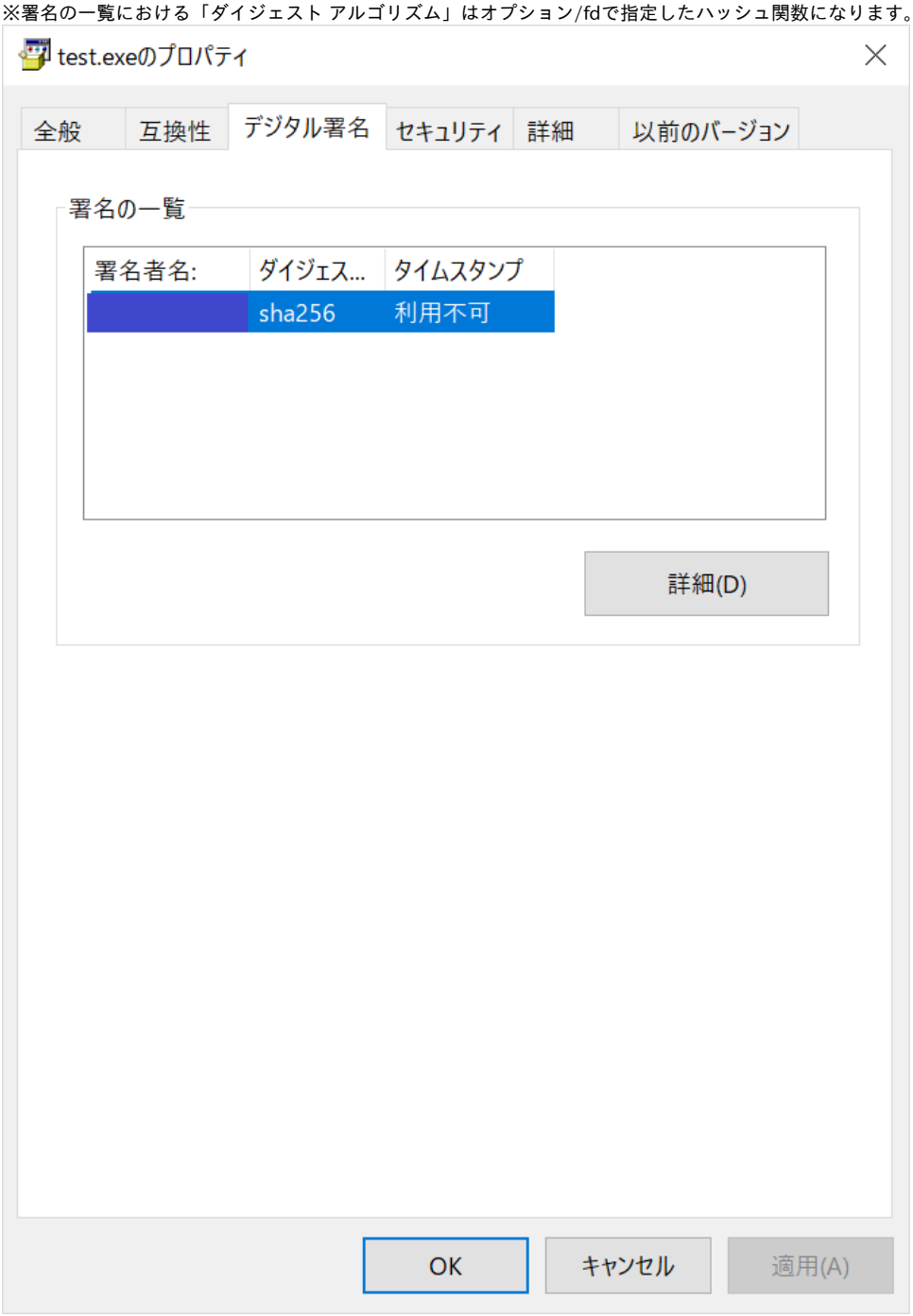
3. 「Successfully verified: 」と表示されることを確認します。



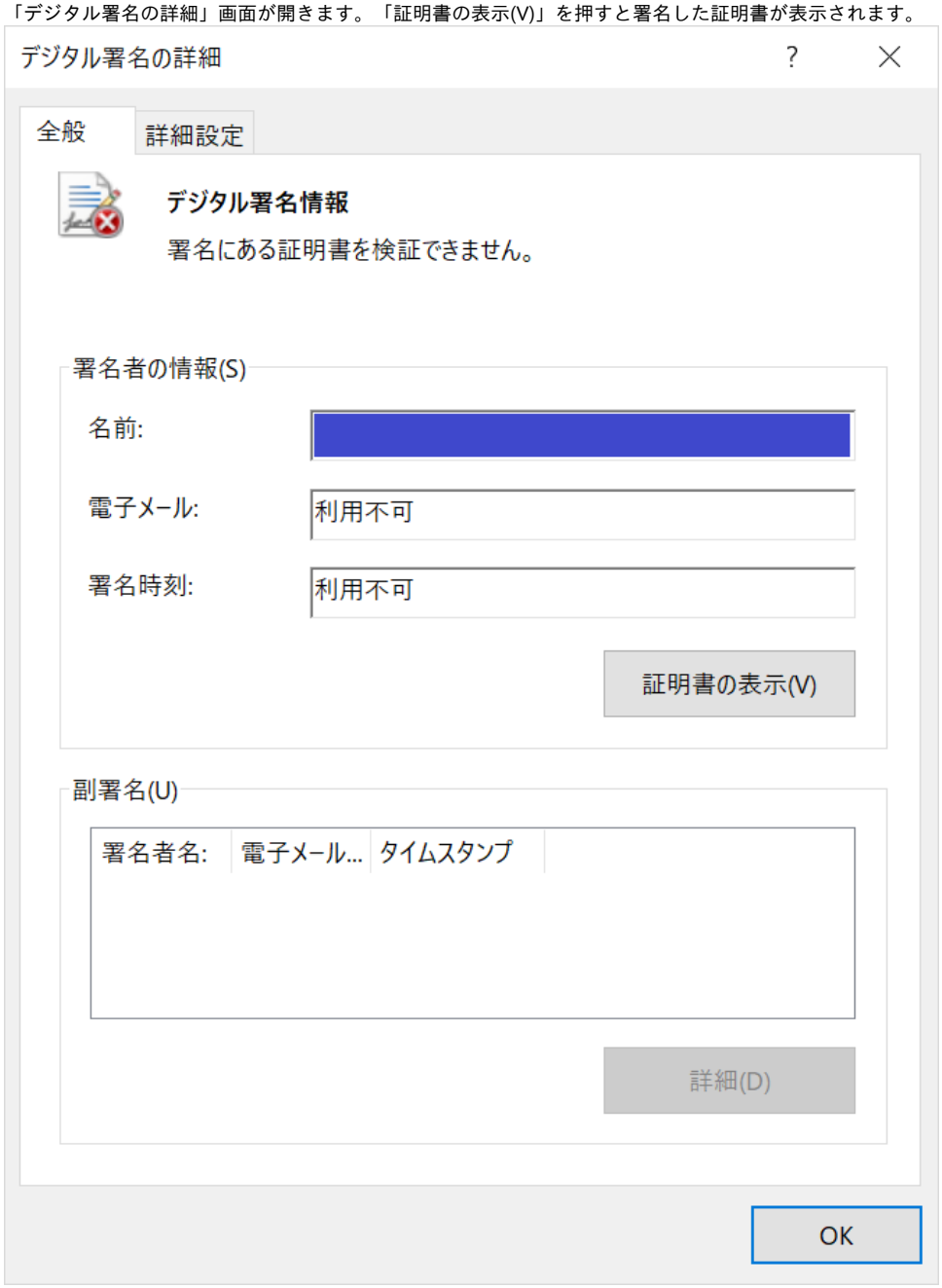
1-4-2. GUI操作によるコード署名確認

署名確認作業（併せてタイムスタンプを付与していない場合）

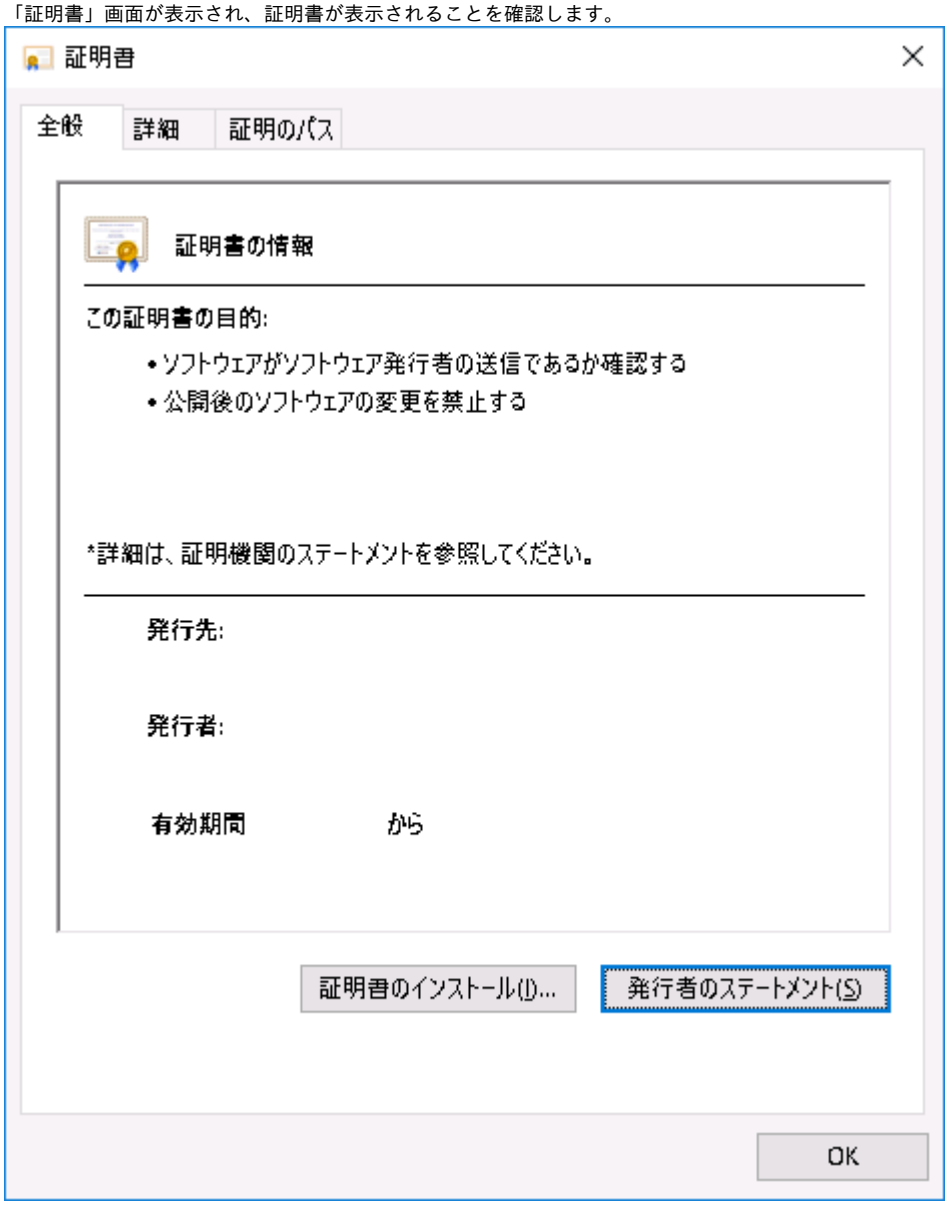
1. 署名したファイルを右クリックにて「プロパティ」を開き、「デジタル署名」タブを開きます。「詳細(D)」を押しますと署名した証明書の内容について確認できます。
※署名の一覧における「ダイジェスト アルゴリズム」はオプション/fdで指定したハッシュ関数になります。



2. 「デジタル署名の詳細」画面が開きます。「証明書の表示(V)」を押すと署名した証明書が表示されます。

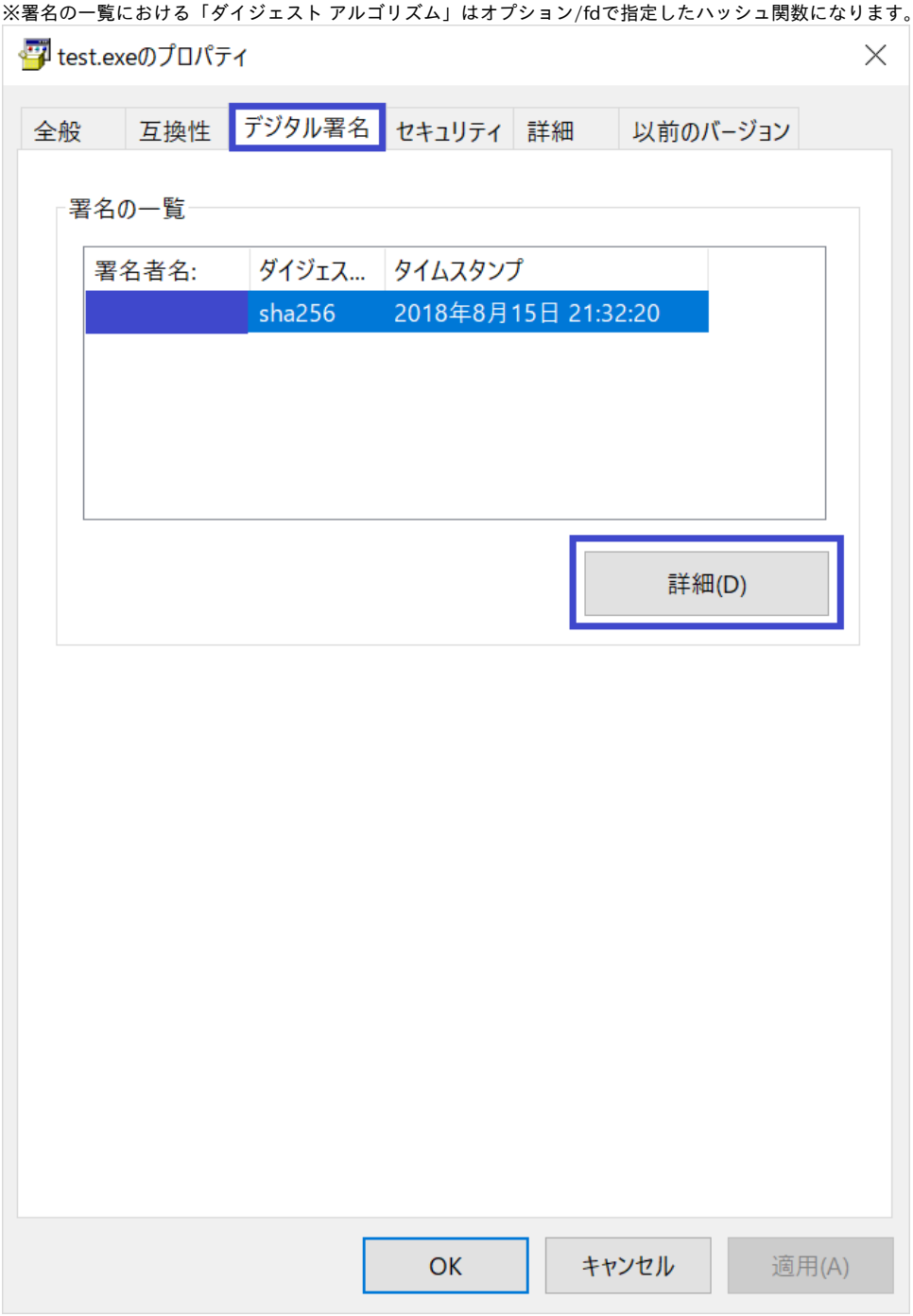


3. 「証明書」画面が表示され、証明書が表示されることを確認します。



署名確認作業（併せてタイムスタンプを付与した場合）

1. 署名したファイルを右クリックにて「プロパティ」を開き、「デジタル署名」タブを開きます。「詳細(D)」を押しますと署名した証明書の内容について確認できます。
※署名の一覧における「ダイジェスト アルゴリズム」はオプション/fdで指定したハッシュ関数になります。



2. 「デジタル署名の詳細」画面が開きます。副署名の項目からタイムスタンプを確認してください。