商業登記リモート署名ドライバソフト
API 仕様書
【CNG 編】

1.0 版

令和７年１０月
法務省民事局商事課

改訂履歴

版数	改訂日付	改訂内容
1.0 版	2025 年 10 月 24 日	・新規作成

目次

第 1 章　はじめに

商業登記リモート署名ドライバソフト（以降、署名ドライバとする）は、以下の機能を実現するためのApplication Program Interface(以下、API)を提供する。

- 証明書取得機能
- 電子署名生成機能
- 電子署名検証機能

以降、本書では署名ドライバのうち、CNG（Cryptography API: Next Generation）のAPI仕様について説明する。

第２章　動作環境

署名ドライバの動作環境は以下の通りとする。

表１　動作環境

項目	条件
プラットフォーム	Windows 11 Home/Pro
Web ブラウザ	Microsoft Edge

第３章　必要となるヘッダーファイル

署名ドライバで必要となるヘッダーファイルは以下の通りとする。

表１ ヘッダーファイル

ヘッダーファイル名	概要
CRPKIKSP_ex.h	KSP 外部公開ヘッダ

第４章　機能仕様

１．ソフトウェア構成図

本仕様書では、署名ドライバのうち、下図の太枠に示す署名ライブラリ(CNG/KSP)の仕様をまとめる。

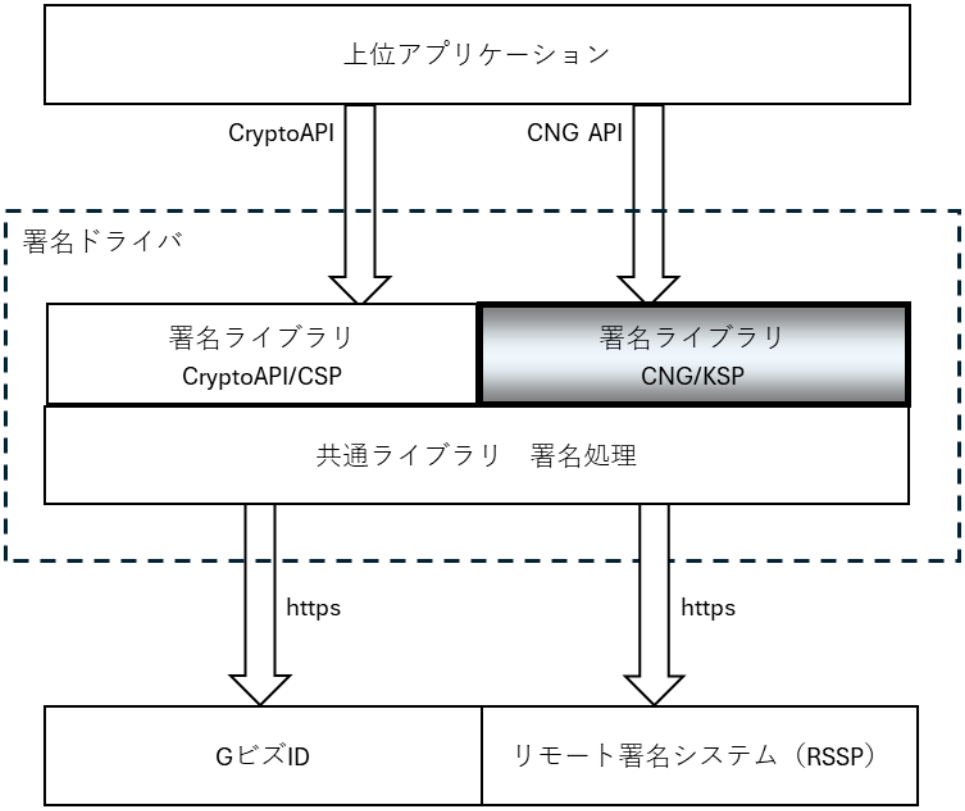


図1 ソフトウェア構成図

２．実現可能な機能の一覧

署名ライブラリ(CNG/KSP)で実現可能な機能の一覧を表 2に示す。

表2 実現可能な機能の一覧

NO	機能	概要
1	利用者証明書取得	証明書ストアから商業登記電子証明書（管理ツール*1 から設定したもの）を取得する。
2	認証局の自己署名証明書取得	証明書ストアから登記官証明書（自己署名証明書）（管理ツール*1 から設定したもの）を取得する。
3	署名生成	ハッシュ値に対して、商業登記電子証明書に紐づく RSSP 上の秘密鍵を使用して署名値を生成する。
4	署名検証	ハッシュ値、署名値、公開鍵を使用して署名値を検証する。
5	繰り返し署名生成	NO 3 の処理を繰り返し実行し、複数のハッシュ値に対する署名値を生成する。
6	繰り返し署名検証	NO 4 の処理を繰り返し実行し、複数の署名値を検証する。

*1 署名アプリケーションから電子署名を行う際に必要な電子署名用の鍵および証明書を設定するためのツール。

第５章　API仕様

１．サポート API 一覧

署名ドライバ用Key Storage Provider(以下、KSP)がサポートするCNG APIの一覧を表 3に示す。なお、サポートしていないAPIを呼び出した場合には、戻り値が常にFALSE(失敗)となる。

表 3 サポートAPI一覧

NO	CNG API	概要
1	NCryptOpenStorageProvider	プロバイダーを読み込む。
2	NCryptOpenKey	既存の鍵を開く。
3	NCryptGetProperty	プロバイダーまたは鍵のプロパティの値を取得する。
4	NCryptImportKey	鍵 BLOB から CNG キーをインポートする。
5	NCryptFreeObject	プロバイダーまたは鍵のハンドルを解放する。
6	NCryptSignHash	ハッシュ値に署名する。
7	NCryptVerifySignature	署名を検証する。

２．サポート API 仕様詳細

各関数の戻り値はBOOL型で、エラーが発生した場合はFALSEを返す。上位アプリケーションでエラー情報を取得する際は、GetLastError()関数をコールしてエラーコードを取得すること。なお、本仕様書に記述のないエラーコードがOSから返る場合があるが、それらのエラーコードについてはMSDNを参照のこと。

（1） NCryptOpenStorageProvider

API名	NCryptOpenStorageProvider		
概要	プロバイダーを読み込む。		
関数インターフェース	SECURITY_STATUS NCryptOpenStorageProvider(NCRYPT_PROV_HANDLE *phProvider, LPCWSTR pszProviderName, DWORD dwFlags);		
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）		
	型	I/O	内容
引数	NCRYPT_PROV_HANDLE *	OUT	ハンドル格納場所のポインタ
	LPCWSTR	IN	NULL文字で終了するKSP名称 ”CRPKI Key Storage Provider for Remote Sign”を指定すること。
	DWORD	IN	動作に関するパラメータ 0 を指定すること。
	値	内容	
エラーコード	NTE_BAD_FLAGS	dwFlags に不正な値が設定されている。	
	NTE_INVALID_PARAMETER	1 つ以上のパラメーターが無効。	
	NTE_NO_MEMORY	メモリ割り当てエラーが発生。	
備考			

（2） NCryptOpenKey

API名	NCryptOpenKey	
概要	既存の鍵を開く。	
関数インターフェース	SECURITY_STATUS NCryptOpenKey(PROV_HANDLE hProvider, NCRYPT_KEY_HANDLE *phKey, LPCWSTR pszKeyName, DWORD dwLegacyKeySpec, DWORD dwFlags);	
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）	
	型	内容

引数	NCRYPT_PROV_HANDLE	NCryptOpenStorageProviderで取得したハンドル
	NCRYPT_KEY_HANDLE	ハンドル格納場所のポインタ
	LPCWSTR	NULL文字で終了する鍵の名称 指定する値を表1に示す。
	DWORD	鍵の種類 0を指定すること。
	DWORD	動作に関するパラメータ 0を指定すること。
	値	内容
エラーコード	NTE_BAD_FLAGS	dwFlags パラメーターに無効な値が含まれています。
	NTE_BAD_KEYSET	指定したキーが見つかりませんでした。
	NTE_INVALID_HANDLE	hProvider パラメーターが無効です。
	NTE_INVALID_PARAMETER	1 つ以上のパラメーターが無効です。
	NTE_NO_MEMORY	メモリ割り当てエラーが発生しました。
備考		

表 1 NCryptOpenKey の pszKeyName で指定する値

#	値	内容
1	Private key of CRPKI_RSA_SHA256	RSA 署名の SHA-256 をサポートする。
2	Private key of CRPKI_RSA_SHA384	RSA 署名の SHA-384 をサポートする。

（3） NCryptGetProperty

API名	NCryptGetProperty		
概要	KSPまたは鍵のプロパティの値を取得する。		
関数インターフェース	SECURITY_STATUS NCryptGetProperty(NCRYPT_HANDLE hObject, LPCWSTR pszProperty, PBYTE pbOutput, DWORD cbOutput, DWORD *pcbResult, DWORD dwFlags);		
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）		
	型	I/O	内容
引数	NCRYPT_HANDLE	IN	キー ハンドル (NCRYPT_KEY_HANDLE)
	LPCWSTR	IN	NULL文字で終了する取得するプロパティ

			の名前。 CRPKI_CERT_PROPERTY_MY または CRPKI_CERT_PROPERTY_ROOT を指定する。 サポートする値を表2に示す。
	PBYTE	OUT	値を格納するバッファのポインタ NULLを指定した場合は値の書き込みは行われず、pcbResultに値の格納に必要な長さが設定される。
	DWORD	IN	*pbOutput* バッファーのサイズ (バイト単位)
	DWORD	OUT	値の長さを格納するバッファへのポインタ pbOutput パラメーターが NULL の場合、バッファーに必要なサイズ (バイト単位) が設定される。
	DWORD	IN	動作に関するパラメータ 0を指定すること。
	値	内容	
エラーコード	NTE_BAD_FLAGS	パラメーターに dwFlags 無効な値が含まれています。	
	NTE_INVALID_HANDLE	hObject パラメーターが有効ではありません。	
	NTE_INVALID_PARAMETER	1 つ以上のパラメーターが無効です。	
	NTE_NO_MEMORY	メモリ割り当てエラーが発生しました。	
	NTE_NOT_SUPPORTED	指定したプロパティは、オブジェクトではサポートされていません。	
備考			

表 2 NCryptGetProperty の pszProperty でサポートする値

#	値	内容
1	NCRYPT_NAME_PROPERTY	鍵の名称を返す。
2	NCRYPT_VERSION_PROPERTY	バージョン情報を返す。
3	NCRYPT_ALGORITHM_PROPERTY	使用中のアルゴリズム名を返す。
4	NCRYPT_IMPL_TYPE_PROPERTY	NCRYPT_IMPL_SOFTWARE_FLAG かつ NCRYPT_IMPL_REMOVABLE_FLAG を返す。(ソフトウェア実装かつリムーバブル実装)
5	CRPKI_CERT_PROPERTY_MY (CRPKIKSP_ex.h に定義された定数値)	証明書ストアの「個人」に登録された、プロバイダーと紐づけられた商標登記電子証明書を取得する。 定数値は「CrpkiMyCertificate」で定義する。
6	CRPKI_CERT_PROPERTY_ROOT	証明書ストアの

	（CRPKIKSP_ex.h に定義された 定数値）	「信頼されるルート証明機関」に登録された 認証局の自己署名証明書を取得する。 定数値は「CrpkiRootCertificate」で定義する

（4） NCryptImportKey

API名	NCryptImportKey		
概要	秘密鍵・公開鍵・対称鍵などをインポート（登録）する		
関数インターフェース	SECURITY_STATUS NCryptImportKey (NCRYPT_PROV_HANDLE hProvider, NCRYPT_KEY_HANDLE hImportKey, LPCWSTR pszBlobType, NCryptBufferDesc *pParameterList, NCRYPT_KEY_HANDLE *phKey, PBYTE pbData, DWORD cbData, DWORD dwFlags);		
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）		
	型	I/O	内容
引数	NCRYPT_PROV_HANDLE	IN	プロバイダーハンドル
	NCRYPT_KEY_HANDLE	IN	鍵ハンドル
	LPCWSTR	IN	インポートする鍵の形式（BLOB種別）。 サポートする値を表3に示す。
	NCryptBufferDesc	IN	インポート処理に渡すオプションパラメータ。NULLを指定すること。
	NCRYPT_KEY_HANDLE	OUT	生成された鍵ハンドル格納ポインタ
	PBYTE	IN	インポートする鍵データ本体（BLOB）へのポインタ
	DWORD	IN	pbData のサイズ（バイト単位）
	DWORD	IN	インポート動作に関するフラグ 0 または NCRYPT_OVERWRITE_KEY_FLAG を指定すること。
	値	内容	
エラーコード	NTE_BAD_DATA	BLOB データの形式不正、不正な鍵形式、破損など	
	NTE_BAD_TYPE	pszBlobType が KSP で未サポート	
	NTE_EXISTS	同名の鍵がすでに存在し、 NCRYPT_OVERWRITE_KEY_FLAG を指定していない	
	NTE_NO_MEMORY	メモリ不足	
	NTE_BAD_KEYSET	KSP が無効、またはプロバイダーに鍵セットが存在しない	

	NTE_INVALID_HANDLE	hProvider もしくは hImportKey が無効なハンドル
	NTE_INVALID_PARAMETER	NULL ポインタ・サイズ不一致などの不正な引数
	NTE_PERM	権限エラー
	NTE_NOT_SUPPORTED	KSP がこの形式（BLOB タイプや暗号アルゴリズム）に未対応
備考		

表 3 **NCryptImportKey** の pszBlobType でサポートする値

#	値	内容
1	BCRYPT_RSAPUBLIC_BLOB	RSA の公開鍵
2	BCRYPT_ECCPUBLIC_BLOB	ECC の公開鍵

（5） NCryptFreeObject

API名	NCryptFreeObject		
概要	プロバイダーまたは鍵のハンドルを解放する。		
関数インターフェース	SECURITY_STATUS NCryptFreeObject(NCRYPT_HANDLE hObject);		
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）		
	型	I/O	内容
引数	NCRYPT_HANDLE	IN	解放するオブジェクトのハンドル プロバイダーハンドル (NCRYPT_PROV_HANDLE) またはキーハンドル (NCRYPT_KEY_HANDLE)を指定する。
	値	内容	
エラーコード	NTE_INVALID_HANDLE	hObject パラメーターのハンドルが無効です。	
備考			

（6） NCryptSignHash

API名	NCryptSignHash
概要	ハッシュ値に署名する。
関数インターフェース	SECURITY_STATUS NCryptSignHash(NCRYPT_KEY_HANDLE hKey, VOID *pPaddingInfo, PBYTE pbHashValue, DWORD cbHashValue, PBYTE pbSignature,

	DWORD cbSignature, DWORD *pcbResult, DWORD dwFlags);		
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）		
	型	I/O	内容
引数	NCRYPT_KEY_HANDLE	IN	ハッシュの署名に使用する鍵のハンドル
	VOID	IN	埋め込み情報を含む構造体へのポインター RSAの場合：備考欄を参照。
	PBYTE	IN	署名するハッシュ値を格納したバッファのポインタ
	DWORD	IN	署名する *pbHashValue* バッファー内のバイト数
	PBYTE	OUT	署名値を格納するバッファのポインタ NULLを指定した場合は値の書き込みは行われず、pcbResultに値の格納に必要な長さが設定される。
	DWORD	IN	*pbSignature* バッファーのサイズ (バイト単位) *pbSignature* パラメーターが NULL の場合、このパラメーターは無視される。
	DWORD	OUT	署名値の長さを格納するバッファのポインタ *pbSignature* パラメーターが NULL の場合、バッファーに必要なサイズ (バイト単位) が設定される。
	DWORD	IN	動作に関するパラメータ RSA の場合：BCRYPT_PAD_PKCS1 を指定する。
	値	内容	
エラーコード	NTE_BAD_ALGID	hKey パラメーターで表されるキーは、署名をサポートしていません。	
	NTE_BAD_FLAGS	dwFlags パラメーターに無効な値が含まれています。	
	NTE_INVALID_HANDLE	hKey パラメーターが無効です。	
	NTE_INVALID_PARAMETER	1 つ以上のパラメーターが無効です。	
	NTE_NO_MEMORY	メモリ割り当てエラーが発生しました。	
	ERR_TOKEN_GET_FAILED	セッション情報（アクセストークン、リフレッシュトークン）の取得に失敗した。	
	ERR_TOKEN_REFRESH_FAILED	セッション情報の再取得（リフレッシュトークンを使用したアクセストークンの再取得）に失敗した。	
	ERR_AUTH_FAILED	G ビズ ID の認証エラーが発生した。	
	ERR_GBIZID_COMM_FAILED	G ビズ ID との通信に失敗した。	
	ERR_SAD_EXTEND_FAILE	SAD の延長に失敗した。	

	D	
	ERR_AUTHCODE_GEN_FAILED	認可コードの生成に失敗した。
	ERR_AUTH_REQUEST_FAILED	認可の要求に失敗した。
	ERR_AUTH_REQUEST_CHECK_FAILED	認可の要求に対するチェックが失敗した。
	ERR_SIGN_VALUE_GET_FAILED	署名値の取得に失敗した。
	ERR_SIGN_LIST_RETURN_FAILED	取得した署名値のリスト返却に失敗した。
	ERR_RSSP_COMM_FAILED	リモート署名システムとの通信に失敗した。
備考	pPaddingInfo に設定するパディング構造体（BCRYPT_PKCS1_PADDING_INFO）のメンバ pszAlgId には以下のいずれかのアルゴリズムを設定する。 RSA SHA-256 の場合：BCRYPT_SHA256_ALGORITHM: SHA-256 RSA SHA-384 の場合：BCRYPT_SHA384_ALGORITHM: SHA-384	

（本ドキュメント中の「ERR_～」から始まるエラーコードの表記は、今後定義予定の独自エラーコードを示します。正式なエラーコードは、次版以降のドキュメントで提供します。）

（7） NCryptVerifySignature

API名	NCryptVerifySignature		
概要	署名を検証する		
関数インターフェース	SECURITY_STATUS NCryptVerifySignature (NCRYPT_KEY_HANDLE hKey, VOID *pPaddingInfo, PBYTE pbHashValue, DWORD cbHashValue, PBYTE pbSignature, DWORD cbSignature, DWORD dwFlags);		
戻り値	SECURITY_STATUS（ERROR_SUCCESS:成功 ERROR_SUCCESS以外:失敗）		
	型	I/O	内容
引数	NCRYPT_KEY_HANDLE	IN	検証に使用する公開鍵のハンドル（NCryptOpenKeyやNCryptImportKeyで取得）
	VOID	IN	パディング情報の構造体 dwFlagsの値により、指定する構造体を変更する。
	PBYTE	IN	検証対象となる元のハッシュ値（署名前の元データのハッシュ）
	DWORD	IN	pbHashValueのバイト長

	PBYTE	IN	実際に署名されたバイト列(署名者が送信してきた署名データ)
	DWORD	IN	pbSignatureのサイズ（バイト単位）
	DWORD	IN	パディングの指定。 サポートする値を表 4 に示す。
	値	内容	
エラーコード	NTE_BAD_SIGNATURE	署名の検証が失敗	
	NTE_INVALID_HANDLE	ハンドルのパラメータが無効	
	NTE_NO_MEMORY	メモリ割り当てのエラーが発生	
	NTE_NOT_SUPPORTED	鍵ハンドルの作成に使用された署名アルゴリズムがサポート対象外	
備考			

表 4 **NCryptVerifySignature** の dwFlags でサポートする値

#	使用する署名方式	dwFlags の値	pPaddingInfo に指定する構造体
1	RSA-PKCS1	BCRYPT_PAD_PKCS1	BCRYPT_PKCS1_PADDING_INFO

３．コーリングシーケンス

「第３章 第２節 実現可能な機能の一覧」を実現するためのコーリングシーケンスを以下に示す。上位アプリケーションは、このコーリングシーケンスに沿って実装すること。

（1） 利用者証明書取得処理

NCryptOpenStorageProvider	プロバイダーを読み込む phProvider: プロバイダーハンドル格納領域アドレス pszProviderName: ”CRPKI Key Storage Provider for Remote Sign” dwFlags: 0

↓

NCryptOpenKey	鍵ハンドル取得 hProvider: NCryptOpenStorageProvider で取得したハンドル phKey: 鍵ハンドル格納領域アドレス pszKeyName: 以下いずれかの値を設定 Private key of CRPKI_RSA_SHA256 Private key of CRPKI_RSA_SHA384 dwLegacyKeySpec: 0 dwFlags: 0

↓

NCryptGetProperty	証明書サイズを取得する。 hObject: NCryptOpenKey で取得したハンドル pszProperty: CRPKI_CERT_PROPERTY_MY pbOutput: NULL cbOutput: 0 pcbResult: サイズ格納用ポインタ dwFlags: 0

↓

NCryptGetProperty	証明書を取得する。 hObject: NCryptOpenKey で取得したハンドル pszProperty: CRPKI_CERT_PROPERTY_MY pbOutput: 証明書格納用ポインタ cbOutput: 格納領域サイズ pcbResult: サイズ格納用ポインタ dwFlags: 0

↓

NCryptFreeObject	鍵ハンドル解放 hObject: NCryptOpenKey で取得したハンドル

↓

NCryptFreeObject	プロバイダーハンドル解放 hObject: NCryptOpenStorageProvider で取得したハンドル

（2） 認証局の自己署名証明書取得処理

NCryptOpenStorageProvider	プロバイダーを読み込む phProvider: プロバイダーハンドル格納領域アドレス pszProviderName: ”CRPKI Key Storage Provider for Remote Sign” dwFlags: 0

↓

NCryptOpenKey	鍵ハンドル取得 hProvider: NCryptOpenStorageProvider で取得したハンドル phKey: 鍵ハンドル格納領域アドレス pszKeyName: 以下いずれかの値を設定 Private key of CRPKI_RSA_SHA256 Private key of CRPKI_RSA_SHA384

	dwLegacyKeySpec: 0 dwFlags: 0

↓

NCryptGetProperty	証明書サイズを取得する。 hObject: NCryptOpenKey で取得したハンドル pszProperty: CRPKI_CERT_PROPERTY_ROOT pbOutput: NULL cbOutput: 0 pcbResult: サイズ格納用ポインタ dwFlags: 0

↓

NCryptGetProperty	証明書を取得する。 hObject: NCryptOpenKey で取得したハンドル pszProperty: CRPKI_CERT_PROPERTY_ROOT pbOutput: 証明書格納用ポインタ cbOutput: 格納領域サイズ pcbResult: サイズ格納用ポインタ dwFlags: 0

↓

NCryptFreeObject	鍵ハンドル解放 hObject: NCryptOpenKey で取得したハンドル

↓

NCryptFreeObject	プロバイダーハンドル解放 hObject: NCryptOpenStorageProvider で取得したハンドル

（3） 署名生成処理

NCryptOpenStorageProvider	プロバイダーを読み込む phProvider: プロバイダーハンドル格納領域アドレス pszProviderName: ”CRPKI Key Storage Provider for Remote Sign” dwFlags: 0

↓

NCryptOpenKey	鍵ハンドル取得 hProvider: NCryptOpenStorageProvider で取得したハンドル phKey: 鍵ハンドル格納領域アドレス pszKeyName:以下いずれかの値を設定 Private key of CRPKI_RSA_SHA256 Private key of CRPKI_RSA_SHA384 dwLegacyKeySpec: 0 dwFlags: 0

↓

NCryptSignHash	署名値長取得 hKey: NCryptOpenKey で取得したハンドル pPaddingInfo: 以下の値が設定された BCRYPT_PKCS1_PADDING_INFO 構造体を指定 pszAlgId=BCRYPT_SHA256_ALGORITHM または BCRYPT_SHA384_ALGORITHM pbHashValue: 署名対象データのハッシュ値 cbHashValue: 署名対象データのハッシュ値のバイト数 pbSignature: NULL cbSignature: 0 *pcbResult: 署名データ長格納領域アドレス dwFlags:0

	（※本 API 実行時、「第６章　画面仕様　１．画面一覧」に記載の画面が表示される。）

↓

NCryptSignHash	署名 hKey: 署名対象ハッシュオブジェクトのハンドル pPaddingInfo: RSA の場合：pPaddingInfo に設定するパディング構造体（BCRYPT_PKCS1_PADDING_INFO）のメンバ pszAlgId には以下のいずれかのアルゴリズムを設定する。 RSA SHA-256 の場合：BCRYPT_SHA256_ALGORITHM: SHA-256 RSA SHA-384 の場合：BCRYPT_SHA384_ALGORITHM: SHA-384 pbHashValue: 署名対象データのハッシュ値 cbHashValue: 署名対象データのハッシュ値のバイト数 pbSignature: 署名に使用するハッシュ値 cbSignature: 署名データ長 *pcbResult: 署名データ長格納領域アドレス dwFlags: RSA の場合：BCRYPT_PAD_PKCS1

↓

NCryptFreeObject	鍵ハンドル解放 hObject: NCryptOpenKey で取得したハンドル

↓

NCryptFreeObject	プロバイダーハンドル解放 hObject: NCryptOpenStorageProvider で取得したハンドル

（4）　署名検証処理

NCryptOpenStorageProvider	プロバイダーを読み込む phProvider: プロバイダーハンドル格納領域アドレス pszProviderName: ”CRPKI Key Storage Provider for Remote Sign” dwFlags: 0

↓

CertCreateCertificateContext (本 CSP の対象外。 OS 標準機能を使用する。)	証明書コンテキスト生成 dwCertEncodingType: X509_ASN_ENCODING pbCertEncoded: X.509 DER 形式の証明書データ cbCertEncoded: 証明書長

↓

NCryptImportKey	鍵インポート hProvider: プロバイダーハンドル hImportKey:鍵ハンドル pszBlobType: インポートする鍵の形式 *pParameterList: NULL *phKey: 生成された鍵ハンドル格納ポインタ pbData: インポートする鍵データ本体（BLOB）へのポインタ cbData: pbData のサイズ dwFlags0 または NCRYPT_OVERWRITE_KEY_FLAG

↓

CertFreeCertificateContext (本 CSP の対象外。 OS 標準機能を使用する。)	証明書コンテキスト破棄 pCertContext:破棄する証明書コンテキスト

↓

NCryptVerifySignature	署名検証

	hKey:検証に使用する公開鍵ハンドル pPaddingInfo:パディング情報 pbHashValue:検証対象のハッシュ値 cbHashValue:ハッシュ値のサイズ pbSignature: 検証対象の署名データ cbSignature:署名データのサイズ dwFlags:パディングの指定

↓

NCryptFreeObject	鍵ハンドル解放 hObject: NCryptOpenKey で取得したハンドル

↓

NCryptFreeObject	プロバイダーハンドル解放 hObject: NCryptOpenStorageProvider で取得したハンドル

（5） 繰り返し署名生成処理

「（3）署名生成処理」の網掛け部分を署名対象データの個数分だけ繰り返して呼び出す。

※（5）の繰り返し署名生成処理の回数の上限は 10 回とする。

（6） 繰り返し署名検証処理

「（4）署名検証処理」の網掛け部分を署名対象データの個数分だけ繰り返して呼び出す。

第６章　画面仕様

１．画面一覧

NO	機能名	画面名	機能概要
1	ログイン機能	G ビズ ID ログイン	G ビズ ID ログイン画面を表示する。（外部サイト）
2	署名認可	鍵パスワード入力画面	署名認可時に必要な鍵パスワードを入力する。
3	署名認可	認可コード表示画面	G ビズ ID の端末認可に入力する認可コードを画面に表示する。

２．画面仕様詳細

（１）G ビズ ID ログイン

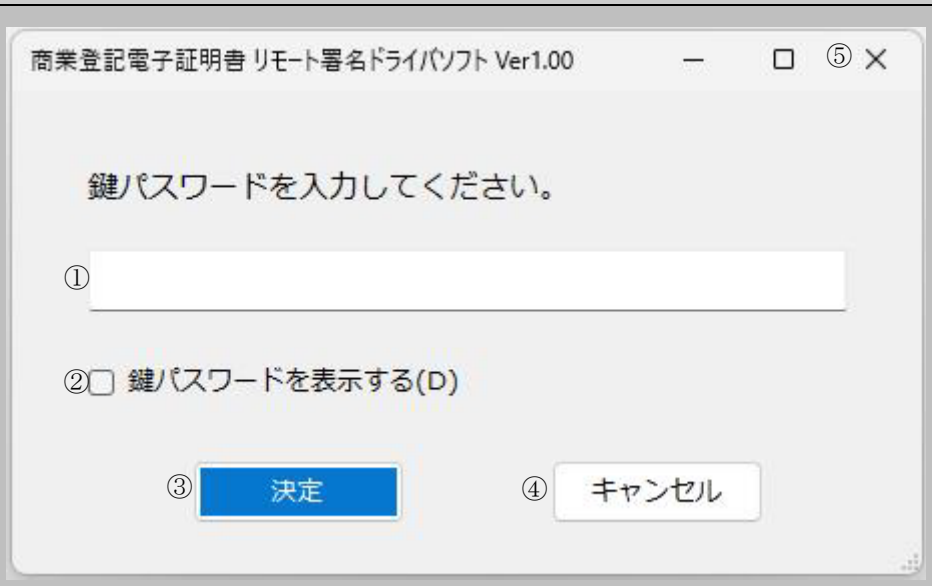
画面名	G ビズ ID ログイン（外部サイト）
概要	G ビズ ID のログインが必要な場合、ブラウザが起動されて G ビズ ID ログイン画面が表示され、G ビズ ID ログイン認証を行う。

画面レイアウト

（2） 鍵パスワード入力画面

画面名	**鍵パスワード**入力画面
概要	署名認可時に必要な鍵パスワードを入力する。

画面レイアウト



画面項目説明

NO	項目名	概要
①	鍵パスワード入力欄	鍵パスワードを入力する。（伏字(*)で表示する。）
②	鍵パスワード表示切替	入力した鍵パスワードの表示/非表示を切替える。
③	決定	リモート署名の認可要求を行う。3 回までリトライ可能。３回目失敗時はエラーメッセージを表示して署名処理を終了する。
④	キャンセル	画面を閉じて鍵パスワード入力処理を終了する。
⑤	×	画面を閉じて鍵パスワード入力処理を終了する。

（3） 認可コード表示画面

画面名	認可コード表示画面
概要	G ビズ ID の端末認可に入力する認可コードを画面に表示する。G ビズ ID の端末認可にて使用する。

画面レイアウト

画面項目説明

NO	項目名	概要
①	認可コード	認可コードを表示する。
②	キャンセル	画面を閉じて認可コード表示処理を終了する。
③	×	画面を閉じて認可コード表示処理を終了する。

禁・無断転載

商業登記リモート署名ドライバソフト API 仕様書

【CNG 編】

第１.０版

（注意事項）

※　署名ドライバの著作権は、法務省が保有しており、国際著作権条約及び日本国の著作権関連法令によって保護されています。

※　法務省は、利用者が署名ドライバを利用したことにより発生した利用者の損害及び利用者が第三者に与えた損害について、一切の責任を負いません。

※　署名ドライバの利用に当たっては、次に掲げる行為を禁止します。

(1)　　署名ドライバに対し、法務省に許可なく改造等を行うこと。

※　商標については次の通りです。

(1)　Microsoft Windows 及び Microsoft Edge は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

(2)　その他、記載されている会社名、製品名等は、各社の登録商標または商標です。