Руководство программиста. SDK

2024 год

Оглавление

Аннотация

Настоящее руководство содержит описание входящего в состав КриптоАРМ ГОСТ модуля, реализующего программный интерфейс trusted-crypto.

Интерфейс trusted-crypto является встроенным в ПО КриптоАРМ ГОСТ программным интерфейсом языка Javascript, включающим в себя набор объектов с методами работы с сертификатами, подписи и шифрования в формате CMS, установки TLS-соединения.

Описание API модуля trusted-crypto в составе продукта КриптоАРМ ГОСТ

В данном документе описывается typescript интерфейс модуля trused-crypto. Для использования этого SDK в составе продукта КриптоАРМ ГОСТ.

В модуле представлены пространства имен (namespace):

1. CMS – набор классов для выполнения операций с цифровой подписью.
2. PKI – набор классов для работы с криптографическими объектами, такими как цифровые сертификаты, списки отзыва, идентификаторы, ключевые контейнеры и т.п.
3. PKISTORE – набор классов для работы с криптографическими провайдерами, посредством предоставляемого API.
4. UTILS – набор классов для работы с лицензией и логированием операций.
5. COMMON – набор методов для управления контекстом библиотеки OpenSSL.

1. Пространство имен CMS

В данном разделе описан namespace CMS. Интерфейсы, свойства и методы

1. Класс SignedData

Метод	Возвращаемый тип	Описание
load	void | SignedData для static	Чтение подписанного файла
sign | signAsync	void	Создает подпись
import | importAsync	Void | SignedData для static	Импорт подписи из памяти
export | exportAsync	Buffer	Возвращает подписанное сообщение
save | saveAsync	void	Запись в файл
verify | verifyAsync	boolean	Проверка подписанного сообщения
get content	ISignedDataContent	Возвращает контент подписанного сообщения
set content	undefined	Задает контент подписанного сообщения
get policies	string[]	Возвращает политики
set policies	undefined	Задает политики
freeContent	void	Освобождение подписанного контента
isDetached	boolean	Проверка откреплённая ли подпись
certificates	CertificateCollection | Certificate	Возвращает коллекцию сертификатов или сертификат подписчика
signers	SignerCollection | Signer	Возвращает подписчиков или подписчика
get signParams	TimestampParams | CadesParams	Возвращает параметры подписи
set signParams	void	Задает параметры подписи
writeContent	void	Запись контента в файл (асинхронно)
isContentAvailable	boolean	Проверка доступности контента

1.1.1. Метод load

Чтение подписанного файла.
load(filename: string, format?: DataFormat): void

Асинхронный метод: loadAsync(filename: string, format: DataFormat, done: (message: string) => void): void

Параметр	Тип	Описание
filename	string	Полный путь до файла

format?	DataFormat	Необязательный параметр. Тип кодировки

Статичный метод load возвращает SignedData

1.1.2. Метод sign

Формирование подписанного сообщения.

sign(cert: pki.Certificate): void

Асинхронный метод: signAsync(cert: pki.Certificate, done: (message: string) => void): void

Параметр	Тип	Описание
cert	pki.Certificate	Сертификат подписчика

1.1.3. Метод import

Импорт подписанного сообщения из памяти

import(buffer: Buffer, format: DataFormat = DEFAULT_DATA_FORMAT): void

Асинхронный метод: importAsync(buffer: Buffer, format: DataFormat, done: (message: string) => void): void

Параметр	Тип	Описание
buffer	Buffer	Буфер с подписанным сообщением
format?	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.DER

Статичный метод import возвращает SignedData

1.1.3. Метод export

Экспорт подписанного сообщения в буфер

export(format: DataFormat = DEFAULT_DATA_FORMAT): Buffer

Асинхронный метод: exportAsync(format: DataFormat = DEFAULT_DATA_FORMAT, done: (error: string, result: Buffer) => void): void

Параметр	Тип	Описание
format	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.DER

1.1.4. Метод save

Запись подписанного сообщения в файл.

save(filename: string, format: DataFormat): void

Асинхронный метод: saveAsync(filename: string, format: DataFormat, done: (message: string) => void): void

Параметр	Тип	Описание
filename	string	Полный путь до файла
format	DataFormat	Тип кодировки (BASE64 | DER)

1.1.5. Метод verify

Проверка подписи.

verify(signer?: cms.Signer): boolean

Асинхронный метод: verifyAsync(done: (error: string, result: boolean) => void, signer?: cms.Signer): void

Параметр	Тип	Описание
signer?	cms.Signer	Объект подписчика, подпись которого будет проверяться. Необязательный параметр

1.1.6. Метод content

Получение или установка контента в объект подписи (getters/setters).

get content(): ISignedDataContent

set content(data: ISignedDataContent)

Параметр	Тип	Описание
data	ISignedDataContent	Контент (исходное сообщение)

1.1.7. Метод policies

Получение или установка политик в объект подписи (getters/setters).

get policies(): string[]

set policies(values: string[])

Параметр	Тип	Описание
values	string[]	Массив политик (SignedDataPolicy)

1.1.8. Метод freeContent

Освобождение контента (памяти)

freeContent(): void

1.1.9. Метод isDetached

Проверка, является ли подпись откреплённой

isDetached(): boolean

1.1.10. Метод certificates

Получение всех сертификатов подписчиков или одного сертификата, если передан индекс

certificates(index?: number): CertificateCollection | Certificate

Параметр	Тип	Описание
index?	number	Индекс сертификата, который надо получить. Необязательный параметр

1.1.11. Метод signers

Получение всех подписчиков или одного, если передан индекс

signers(index?: number): SignerCollection | Signer

Параметр	Тип	Описание
index?	number	Индекс подписчика, который надо получить. Необязательный параметр

1.1.12. Метод signParams

Получение или установка дополнительных параметров подписи (getters/setters).

get signParams(): TimestampParams | CadesParams

set signParams(params: TimestampParams | CadesParams)

Параметр	Тип	Описание
params	TimestampParams | CadesParams	Параметры подписи, отвечающий за установку штампов времени и cades

2. Класс Signer

Метод	Возвращаемый тип	Описание
get certificate	Certificate	Возвращает сертификат подписчика
set certificate	void	Задает сертификат подписчика
get index	number	Возвращает индекс подписчика
set index	void	Задает индекс подписчика
get signingTime	Date	Возвращает время подписи (из подписанных атрибутов)
get signatureAlgorithm	string	Возвращает алгоритм подписи
get signatureDigestAlgorithm	string	Возвращает хэш алгоритм подписи
get issuerName	string	Возвращает имя издателя сертификата подписчика
get serialNumber	string	Возвращает серийный номер сертификата подписчика
timestamp	TSP	Возвращает штамп времени по его типу
verifyTimestamp	boolean	Проверка штампа времени
isCades	boolean	Возвращает true если это CAdES
get certificateValues	CertificateCollection	Только для CAdES. Возвращает коллекцию сертификатов из атрибута certificateValues
get revocationValues	Buffer[]	Только для CAdES. Возвращает массив с закодированными значениями отзыва (ответ OCSP или CRL)
get ocspResp	OCSP	Только для CAdES. Возвращает OCSP ответ
verifySignatureValue	boolean	Проверка подписи по хешу

1.2.1. Метод certificate

Возвращает сертификат подписчика.

get certificate(): Certificate

1.2.2. Метод index

Получение или установка индекса подписчика (getters/setters).

get index(): number

set index(ind: number)

Параметр	Тип	Описание
ind	number	Индекс подписчика

1.2.3. Метод signingTime

Возвращает время подписи (из подписанных атрибутов).

get signingTime(): Date

1.2.4. Метод signatureAlgorithm

Возвращает алгоритм подписи.

get signatureAlgorithm(): string

1.2.5. Метод signatureDigestAlgorithm

Возвращает хэш алгоритм подписи.

get signatureDigestAlgorithm(): string

1.2.6. Метод issuerName

Возвращает имя издателя сертификата подписчика.

get issuerName(): string

1.2.7. Метод serialNumber

Возвращает серийный номер сертификата подписчика.

get serialNumber(): string

1.2.8. Метод timestamp

Возвращает штамп времени по его типу.

timestamp(tspType: number): TSP

Параметр	Тип	Описание
tspType	number	Тип штампа времени

1.2.9. Метод verifyTimestamp

Проверка штампа времени по типу.

verifyTimestamp(tspType: number): boolean

Параметр	Тип	Описание
tspType	number	Тип штампа времени

1.2.10. Метод isCades

Проверка является ли подпись CAdES.

get isCades(): boolean

1.2.11. Метод certificateValues

Только для CAdES. Возвращает коллекцию сертификатов из атрибута certificateValues

get certificateValues(): CertificateCollection

1.2.12. Метод revocationValues

Только для CAdES. Возвращает массив с закодированными значениями отзыва (ответ OCSP или CRL).

get revocationValues(): Buffer[]

1.2.13. Метод ocspResp

Только для CAdES. Возвращает OCSP ответ.

get ocspResp(): OCSP

1.2.14. Метод verifySignatureValue

Проверка подписи по хешу.
verifySignatureValue(hashValue: Buffer, signValue: Buffer, signAlg: string | undefined, signCertOrKeyValue: pki.Certificate | Buffer, keyAlg?: string, namedCurve?: string): boolean

Параметр	Тип	Описание
hashValue	Buffer	Значение хеша
signValue	Buffer	Значение подписи
signAlg	string | undefined	OID алгоритма подписи
signCertOrKeyValue	pki.Certificate | Buffer	Сертификат подписчика или закодированная структура SubjectPublicKeyInfo
keyAlg	string	Алгоритм ключа. Необязательный параметр

3. Класс SignerCollection

2. Метод	Возвращаемый тип	Описание
items	Signer	Возвращает объект коллекции по его индексу
get length	number	Возвращает количество элементов коллекции

1.3.1. Метод items

Возвращает объект коллекции по его индексу.

items(index: number): Signer

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

1.3.2. Метод length

Возвращает количество элементов коллекции

get length(): number

1.4. Класс CadesParams

Метод	Возвращаемый тип	Описание
get cadesType	number	Возвращает тип (стандарт) cades
set cadesType	void	Задает тип (стандарт) cades
get connSettings	ConnectionSettings	Возвращает настройки соединения для создания tsp
set connSettings	void	Задает настройки соединения для создания cades
get tspHashAlg	string	Возвращает хэш алгоритм
set tspHashAlg	void	Задает хэш алгоритм (OID)
get ocspSettings	ConnectionSettings	Возвращает настройки соединения для создания ocsp
set ocspSettings	void	Задает хэш алгоритм (OID)

1.4.1. Метод cadesType

Получение или установка типа стандарта cades (getters/setters). В текущей версии всегда 1 (CAdES-X Long Type 1)

get cadesType(): number

set cadesType(signType: number)

Параметр	Тип	Описание
signType	number	В текущей версии всегда 1 (CAdES-X Long Type 1)

1.4.2. Метод connSettings

Получение или установка настроек соединения для создания cades (getters/setters).

get connSettings(): utils.ConnectionSettings

set connSettings(connSett: utils.ConnectionSettings): void

Параметр	Тип	Описание
connSett	utils.ConnectionSettings	Набор параметров, используемых для установки сетевого соединения (адрес, аутентификация, пароль и т.д.)

1.4.3. Метод tspHashAlg

Получение или установка используемого алгоритма хеширования (getters/setters).

get tspHashAlg(): string

set tspHashAlg(hashAlgOid: string): void

Параметр	Тип	Описание
hashAlgOid	string	OID используемого хеш-алгоритма. Если не будет задан, то используется значение из сертификата

1.4.4. Метод ocspSettings

Получение или установка настроек соединения для создания cades (getters/setters).

get ocspSettings (): utils.ConnectionSettings

set ocspSettings (connSett: utils.ConnectionSettings): void

Параметр	Тип	Описание
connSett	utils.ConnectionSettings	Набор параметров, используемых для установки сетевого соединения (адрес, аутентификация, пароль и т.д.)

1.5. Класс TimestampParams

Метод	Возвращаемый тип	Описание
get stampType	number	Возвращает тип штампа времени
set stampType	void	Задает тип штампа времени
get connSettings	ConnectionSettings	Возвращает настройки соединения для создания tsp
set connSettings	void	Задает настройки соединения для создания cades
get tspHashAlg	string	Возвращает хэш алгоритм
set tspHashAlg	void	Задает хэш алгоритм (OID)

1.5.1. Метод cadesType

Получение или установка типа стандарта cades (getters/setters). В текущей версии всегда 1 (CAdES-X Long Type 1)

get cadesType(): number

set cadesType(signType: number)

Параметр	Тип	Описание
signType	number	В текущей версии всегда 1 (CAdES-X Long Type 1)

1.5.2. Метод connSettings

Получение или установка настроек соединения для создания cades (getters/setters).

get connSettings(): utils.ConnectionSettings

set connSettings(connSett: utils.ConnectionSettings): void

Параметр	Тип	Описание
connSett	utils.ConnectionSettings	Набор параметров, используемых для установки сетевого соединения (адрес, аутентификация, пароль и т.д.)

1.5.3. Метод tspHashAlg

Получение или установка используемого алгоритма хеширования (getters/setters).

get tspHashAlg(): string

set tspHashAlg(hashAlgOid: string): void

Параметр	Тип	Описание
hashAlgOid	string	OID используемого хеш-алгоритма. Если не будет задан, то используется значение из сертификата

1.5.4. Метод ocspSettings

Получение или установка настроек соединения для создания cades (getters/setters).

get ocspSettings (): utils.ConnectionSettings

set ocspSettings (connSett: utils.ConnectionSettings): void

Параметр	Тип	Описание
connSett	utils.ConnectionSettings	Набор параметров, используемых для установки сетевого соединения (адрес, аутентификация, пароль и т.д.)

1.6. Перечисляемые типы

1.4.1. SignedDataContentType

Определяет вид контента для объекта подписи

Идентификатор	Значение	Описание
`url`	0	Путь до файла
`buffer`	1	Буфер в памяти

hash	2	Хеш

1.6.2. StampType

Определяет виды штампов времени на подпись

Идентификатор	Значение	Описание
stContent	1	Штамп времени на подписываемые данные
stSignature	2	Штамп времени на подпись
stEscStamp	4	

1.6.3. CadesType

Определяет формат CAdES. Ограничение: только CAdES-X Long Type 1

Идентификатор	Значение	Описание
ctCadesXLT1	1	CAdES-X Long Type 1
ctCadesT	2	CAdES-T

1.7. Интерфейсы

1.7.1. ISignedDataContent

Свойство	Тип	Описание
type	SignedDataContentType	Вид контента
data	string | Buffer	Путь до файла или буфер данных

1.8. Примеры

В данном разделе приведены примеры работы с электронной подписью, в том числе CAdES.

1.8.1. Чтение подписанного файла

```
const cms = new trusted.cms.SignedData();
cms.load("signedfile.sig", trusted.DataFormat.PEM);
```

1.8.2. Подпись

Откреплённая подпись данных:

```
const cert = trusted.pki.Certificate.load("./certfile.cet",
trusted.DataFormat.DER);
const sd = new trusted.cms.SignedData();
sd.policies = ["detached"];
```

```
sd.content = {
      type: trusted.cms.SignedDataContentType.buffer,
      data: "Hello world"
};

sd.sign(cert);
```

Формирование подписи без аттрибутов:

```
const cert = trusted.pki.Certificate.load("./certfile.cet",
trusted.DataFormat.DER);
const sd = new trusted.cms.SignedData();
sd.policies = ["noAttributes"];

sd.content = {
      type: trusted.cms.SignedDataContentType.buffer,
      data: "Hello world"
};

sd.sign(cert);
```

Прикрепленная подпись файла:

```
const cert = trusted.pki.Certificate.load("./certfile.cet",
trusted.DataFormat.DER);
const sd = new trusted.cms.SignedData();
sd.policies = [];

sd.content = {
      type: trusted.cms.SignedDataContentType.url,
      data: "./file_for_sign.txt"
};

sd.sign(cert);
sd.save("./outfile.sig");
```

1.8.3. Усовершенствованная подпись

```
const cert = trusted.pki.Certificate.load("./certfile.cet",
trusted.DataFormat.DER);

const connSettings = new trusted.utils.ConnectionSettings();

connSettings.Address = "http://qs.cryptopro.ru/tsp/tsp.srf";

const sdCades = new trusted.cms.SignedData();

sdCades.content = {
      type: trusted.cms.SignedDataContentType.buffer,
      data: "CAdES test 1"
};
```

```
const cadesParams = new trusted.cms.CadesParams();
cadesParams.cadesType = trusted.cms.CadesType.ctCadesXLT1;
cadesParams.connSettings = connSettings;
cadesParams.tspHashAlg = "1.2.643.7.1.1.2.2";
sdCades.signParams = cadesParams;

sdCades.sign(cert);

sd.save("./outfile.sig");
```

1.8.4. Подпись со штампом времени

```
const sdTspContent = new trusted.cms.SignedData();
sdTspContent.content = {
      type: trusted.cms.SignedDataContentType.buffer,
      data: "Signature with time stamp 1."
};

const connSettings = new trusted.utils.ConnectionSettings();
connSettings.Address = "http://qs.cryptopro.ru/tsp/tsp.srf";

const tspParams = new trusted.cms.TimestampParams();
tspParams.connSettings = connSettings;
tspParams.tspHashAlg = "1.2.643.7.1.1.2.2";


tspParams.stampType = trusted.cms.StampType.stContent;
sdTspContent.signParams = tspParams;

sdTspContent.sign(cert);
sd.save("./outfile.sig");
```

1.8.5. Соподпись

Добавление подписчика (соподпись):

```
const cms = new trusted.cms.SignedData();
cms.load("./outfile.sig");
const secondSignerCert = trusted.pki.Certificate.load("./certfile2.cet",
trusted.DataFormat.DER);

cms.sign(secondSignerCert);

sd.save("./outfile.sig");
```

1.8.6. Проверка подписи

Проверка откреплённой подписи:

```
const cms = new trusted.cms.SignedData();
cms.load("./outfile.sig");
cms.content = {
      type: trusted.cms.SignedDataContentType.url,
      data:  "./data.docx"
```

```
};
const res = cms.verify();
```

Проверка прикрепленной подписи:

```
const cms = new trusted.cms.SignedData();
cms.load("./outfile.sig");
const res = cms.verify();
```

Проверка конкретного подписчика:

```
const cms = new trusted.cms.SignedData();
cms.load("./outfile.sig");
const signers = cms.signers();
const signer = signers.items(0);
const res = cms.verify(signer);
```

2. Пространство имен PKI

2.1. Класс OID

Метод	Возвращаемый тип	Описание
get value	string	Возвращает текстовое представление значения OID
get longName	string	Возвращает длинное имя OID
get shortName	string	Возвращает короткое имя OID

2.1.1. Метод value

Возвращает текстовое представление значения OID.

get value(): string

2.1.2. Метод longName

Возвращает длинное имя OID.

get longName(): string

2.1.2. Метод shortName

Возвращает короткое имя OID.

get shortName(): string

2.1.3. Примеры

Создание объекта OID и получение свойств:

```
const oid = new trusted.pki.Oid("keyUsage");
oid.value; // 2.5.29.15
oid.shortName; // keyUsage
```

2.2. Класс Extension

Метод	Возвращаемый тип	Описание
get typeId	Oid	Возвращает OID
set typeId	void	Задает OID
get critical	boolean	Возвращает критичность расширения
set critical	void	Задает критичность расширения

2.2.1. Метод typeId

Возвращает или задает OID расширения (getters/setters).

get typeId(): Oid

set typeId(oid: Oid)

Параметр	Тип	Описание
oid	Oid	OID

2.2.2. Метод critical

Возвращает или задает критичность расширения (getters/setters).

get critical(): Oid

set critical(critical: boolean)

Параметр	Тип	Описание
critical	boolean	Критичность расширения

2.2.3. Примеры

Формирование расширения (критичное, цифровая подпись):

```
const oid = new trusted.pki.Oid("keyUsage");
const ext = new trusted.pki.Extension(oid, "critical,digitalSignature");
ext.critical; // true
```

2.3. Класс ExtensionCollection

Метод	Возвращаемый тип	Описание
items	Extension	Возвращает объект коллекции по его индексу
get length	number	Возвращает количество элементов коллекции
push	void	Добавляет элемент в коллекцию
pop	void	Удаляет один элемент коллекции
removeAt	void	Удаляет элемент коллекции по его индексу

2.3.1. Метод items

Возвращает объект коллекции по его индексу.

items(index: number): Extension

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

2.3.2. Метод length

Возвращает количество элементов коллекции

get length(): number

2.3.3. Метод push

Добавляет элемент в коллекцию

push(ext: Extension): void

Параметр	Тип	Описание
ext	Extension	Добавляемый Extension

2.3.4. Метод pop

Удаляет один элемент коллекции

pop(): void

2.3.5. Метод removeAt

Возвращает объект коллекции по его индексу.

removeAt(index: number): void

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

2.3.6. Примеры

Добавление extension в коллекцию и получение количества элементов:

```
const exts = new trusted.pki.ExtensionCollection();
const oid = new trusted.pki.Oid("keyUsage");
const ext = new trusted.pki.Extension(oid, "critical,digitalSignature");
exts.push(ext);
exts.length; // 1

const ext2 = exts.items(0);
```

2.4. Класс CRL

Метод	Возвращаемый тип	Описание
load	void	Чтение CRL из файла
import	CRL	Чтение CRL из памяти
get version	number	Возвращает версию CRL
get issuerFriendlyName	string	Возвращает CN имени издателя
get issuerName	string	Возвращает полное имя издателя
get lastUpdate	Date	Возвращает дату последнего обновления CRL
get nextUpdate	Date	Возвращает дату следующего обновления CRL
get thumbprint	string	Возвращает SHA-1 отпечаток CRL
get signatureAlgorithm	string	Возвращает алгоритм подписи CRL
get signatureDigestAlgorithm	string	Возвращает хэш алгоритм подписи CRL
get authorityKeyid	string	Возвращает Authority Key Identifier
get crlNumber	string	Возвращает номер CRL
compare	number	Сравнивает два COC (CRL)
equals	boolean	Проверка являются ли два CRL эквивалентными
hash	string	Возвращает вычисленный отпечаток CRL
duplicate	CRL	Создает копию объекта CRL
export	Buffer	Экспорт CRL в память (буфер)
save	void	Запись CRL в файл

2.4.1. Метод load

Чтение CRL из файла

static load(filename: string, format?: DataFormat): CRL

load(filename: string, format?: DataFormat): void

Параметр	Тип	Описание
filename	string	Полный путь до файла
format?	DataFormat	Необязательный параметр. Тип кодировки

2.4.2. Метод import

Импорт CRL из памяти

static import(buffer: Buffer, format: DataFormat = DEFAULT_DATA_FORMAT): CRL

import(buffer: Buffer, format: DataFormat = DEFAULT_DATA_FORMAT): void

Параметр	Тип	Описание
buffer	Buffer	Буфер с контентом CRL
format?	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.DER

2.4.3. Метод version

Возвращает версию CRL

get version(): number

2.4.4. Метод issuerFriendlyName

Возвращает CN имени издателя

get issuerFriendlyName(): string

2.4.5. Метод issuerName

Возвращает полное издателя

get issuerName(): string

2.4.6. Метод lastUpdate

Возвращает время последнего обновления COC (getters/setters).

get lastUpdate(): Data

2.4.7. Метод nextUpdate

Возвращает время следующего обновления COC (getters/setters).

get nextUpdate (): Data

2.4.8. Метод thumbprint

Возвращает SHA-1 отпечаток CRL.

get thumbprint(): string

2.4.9. Метод signatureAlgorithm

Возвращает алгоритм подписи CRL.

get signatureAlgorithm(): string

2.4.10. Метод signatureDigestAlgorithm

Возвращает хэш алгоритм подписи сертификата.

get signatureDigestAlgorithm(): string

2.4.11. Метод authorityKeyid

Возвращает Authority Key Identifier.

get authorityKeyid(): string

2.4.12. Метод crlNumber

Возвращает номер CRL.

get crlNumber(): number

2.4.13. Метод compare

Сравнивает два CRL. Для различных вернет 1 или -1. Для идентичных 0

compare(crl: CRL): number

Параметр	Тип	Описание
crl	CRL	CRL с которым требуется сравнить текущий

2.4.14. Метод equals

Проверка идентичности двух СОС (CRL).

equals (crl: CRL): boolean

Параметр	Тип	Описание
crl	CRL	CRL с которым требуется сравнить текущий

2.4.15. Метод hash

Возвращает вычисленный отпечаток CRL. По умолчанию используется алгоритм SHA-1

hash(algorithm: string = "sha1"): string

Параметр	Тип	Описание

algorithm?	string	Необязательный параметр. Хэш алгоритм. Значение по умолчанию: sha1

2.4.16. Метод duplicate

Создает копию объекта сертификата

duplicate(): CRL

2.4.17. Метод export

Экспорт CRL в буфер

export(format: DataFormat = DEFAULT_DATA_FORMAT): Buffer

Параметр	Тип	Описание
format	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.DER

2.4.18. Метод save

Запись CRL в файл.

save(filename: string, format: DataFormat): void

Параметр	Тип	Описание
filename	string	Полный путь до файла
format	DataFormat	Тип кодировки (BASE64 | DER)

2.4.19. Примеры

Чтение CRL из файла:

```
const crl = new trusted.cms.CRL();
crl.load("./example.crl");
```

Импорт CRL из памяти:

```
const cert = new trusted.cms.CRL();
const data = fs.readFileSync("./export.crl");
cert.import(data);
```

Получение значений полей CRL:

```
const crl = new trusted.cms.CRL();
crl.load("./example.crl”);
crl.version;
crl.issuerFriendlyName;
```

```
crl.issuerName;
crl.lastUpdate;
crl.nextUpdate;

crl.serialNumber;
crl.thumbprint;
crl.signatureAlgorithm;
crl.signatureDigestAlgorithm;
crl.crlNumber;
```

2.5. Класс CrlCollection

Метод	Возвращаемый тип	Описание
items	CRL	Возвращает объект коллекции по его индексу
get length	number	Возвращает количество элементов коллекции
push	void	Добавляет элемент в коллекцию
pop	void	Удаляет один элемент коллекции
removeAt	void	Удаляет элемент коллекции по его индексу

2.5.1. Метод items

Возвращает объект коллекции по его индексу.

items(index: number): CRL

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

2.5.2. Метод length

Возвращает количество элементов коллекции

get length(): number

2.5.3. Метод push

Добавляет элемент в коллекцию

push(crl: CRL): void

Параметр	Тип	Описание
crl	CRL	Добавляемый СОС

2.5.4. Метод pop

Удаляет один элемент коллекции

pop(): void

2.5.5. Метод removeAt

Возвращает объект коллекции по его индексу.

removeAt(index: number): void

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

2.5.6. Примеры

Добавление CRL в коллекцию и получение количества элементов:

```
const crls = new trusted.pki.CrlCollection();
crls.push(trusted.pki.CRL.load("./crl1.crl"));
crls.push(trusted.pki.CRL.load("./crl2.crl"));
crls.length; // 2
```

Удаление элемента из коллекции:

```
const crls = new trusted.pki.CrlCollection();
crls.push(trusted.pki.CRL.load("./crl1.crl"));
crls.push(trusted.pki.CRL.load("./crl2.crl"));
crls.length; // 2

crls.pop();
crls.length; // 1
```

Получение элемента из коллекции:

```
const crls = new trusted.pki.CrlCollection ();
crls.push(trusted.pki.CRL.load("./crl1.crl"));
crls.push(trusted.pki.CRL.load("./crl2.crl"));
const crl = crls.items(0);
```

2.6. Класс Certificate

Метод	Возвращаемый тип	Описание
load	void	Чтение сертификата из файла
import	Certificate	Чтение сертификата из памяти
get version	number	Возвращает версию сертификата
get serialNumber	string	Возвращает серийный номер сертификата

set serialNumber	void	Задает серийный номер сертификата
get keyUsage	number	Возвращает поле использование ключа сертификата
get issuerFriendlyName	string	Возвращает CN имени издателя
get issuerName	string	Возвращает полное имя издателя
get subjectFriendlyName	string	Возвращает CN имени субъекта
get subjectName	string	Возвращает полное имя субъекта
get notBefore	Date	Возвращает дату начала действия сертификата
set notBefore	void	Задает дату начала действия сертификата
get notAfter	Date	Возвращает дату окончания действия сертификата
set notAfter	void	Задает дату окончания действия сертификата
get thumbprint	string	Возвращает SHA-1 отпечаток сертификата
get signatureAlgorithm	string	Возвращает алгоритм подписи сертификата
get signatureDigestAlgorithm	string	Возвращает хэш алгоритм подписи сертификата
get publicKeyAlgorithm	string	Возвращает алгоритм публичного ключа сертификата
get organizationName	string	Возвращает поле «Организация» субъекта сертификата
get OCSPUrls	string[]	Возвращает массив ссылок OCSP
get CAIssuersUrls	string[]	Возвращает массив точек распространения сертификатов УЦ
isSelfSigned	boolean	Проверка является ли сертификат самоподписанным
isCA	boolean	Проверка является ли сертификат сертификатом УЦ
sign	void	Подпись сертификата
compare	number	Сравнивает два сертификата
equals	boolean	Проверка являются ли два сертификата эквивалентными
hash	string	Возвращает вычисленный отпечаток сертификата
duplicate	Certificate	Создает копию объекта сертификата
export	Buffer	Экспорт сертификата в память (буфер)
save	void	Запись сертификата в файл

2.6.1. Метод load

Чтение сертификата из файла

static load(filename: string, format?: DataFormat): Certificate

load(filename: string, format?: DataFormat): void

Параметр	Тип	Описание
filename	string	Полный путь до файла
format?	DataFormat	Необязательный параметр. Тип кодировки

2.6.2. Метод import

Импорт сертификата из памяти

static import(buffer: Buffer, format: DataFormat = DEFAULT_DATA_FORMAT): Certificate

import(buffer: Buffer, format: DataFormat = DEFAULT_DATA_FORMAT): void

Параметр	Тип	Описание
buffer	Buffer	Буфер с контентом сертификата
format?	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.DER

2.6.3. Метод version

Возвращает версию сертификата

get version(): number

2.6.4. Метод serialNumber

Возвращает или задает серийный номер сертификата (getters/setters).

get serialNumber(): string

set serialNumber(serial: string)

Параметр	Тип	Описание
serial	string	Серийный номер

2.6.5. Метод keyUsage

Возвращает поле использование ключа сертификата

get keyUsage(): number

2.6.6. Метод issuerFriendlyName

Возвращает CN имени издателя

get issuerFriendlyName(): string

2.6.7. Метод issuerName

Возвращает полное издателя

get issuerName(): string

2.6.8. Метод subjectFriendlyName

Возвращает CN имени субъекта

get subjectFriendlyName(): string

2.6.9. Метод subjectName

Возвращает полное субъекта

get subjectName(): string

2.6.10. Метод notBefore

Возвращает или задает время начала действия сертификата (getters/setters).

get notBefore (): Data

set notBefore (offsetSec: number)

Параметр	Тип	Описание
offsetSec	number	Unix-время начала действия сертификата

2.6.11. Метод notAfter

Возвращает или задает время окончания действия сертификата (getters/setters).

get notBefore (): Data

set notBefore (offsetSec: number)

Параметр	Тип	Описание
offsetSec	number	Unix-время окончания действия сертификата

2.6.12. Метод thumbprint

Возвращает SHA-1 отпечаток сертификата

get thumbprint(): string

2.6.13. Метод signatureAlgorithm

Возвращает алгоритм подписи сертификата

get signatureAlgorithm(): string

2.6.14. Метод signatureDigestAlgorithm

Возвращает хэш алгоритм подписи сертификата

get signatureDigestAlgorithm(): string

2.6.15. Метод publicKeyAlgorithm

Возвращает алгоритм публичного ключа сертификата

get publicKeyAlgorithm(): string

2.6.16. Метод organizationName

Возвращает поле «Организация» субъекта сертификата

get organizationName(): string

2.6.17. Метод OCSPUrls

Возвращает массив ссылок OCSP

get OCSPUrls(): string[]

2.6.18. Метод CAIssuersUrls

Возвращает массив точек распространения сертификатов УЦ

get CAIssuersUrls(): string[]

2.6.19. Метод isSelfSigned

Проверка является ли сертификат самоподписанным. Для самоподписанного сертификата вернет true

get isSelfSigned(): boolean

2.6.20. Метод isCA

Проверка является ли сертификат сертификатом УЦ. Для сертификата УЦ вернет true

get isCA (): boolean

2.6.21. Метод sign

Подпись сертификата. Используется для генерации самоподписанных сертификатов

sign(): void

2.6.22. Метод compare

Сравнивает два сертификата. Для различных вернет 1 или -1. Для идентичных 0

compare(cert: Certificate): number

Параметр	Тип	Описание

cert	Certificate	Сертификат с которым требуется сравнить текущий

2.6.23. Метод equals

Проверка идентичности двух сертификатов.

equals (cert: Certificate): boolean

Параметр	Тип	Описание
cert	Certificate	Сертификат с которым требуется сравнить текущий

2.6.24. Метод hash

Возвращает вычисленный отпечаток сертификата. По умолчанию используется алгоритм SHA-1

hash(algorithm: string = "sha1"): string

Параметр	Тип	Описание
algorithm?	string	Необязательный параметр. Хэш алгоритм. Значение по умолчанию: sha1

2.6.25. Метод duplicate

Создает копию объекта сертификата

duplicate(): Certificate

2.6.26. Метод export

Экспорт сертификата в буфер

export(format: DataFormat = DEFAULT_DATA_FORMAT): Buffer

Параметр	Тип	Описание
format	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.DER

2.6.27. Метод save

Запись сертификата в файл.

save(filename: string, format: DataFormat): void

Параметр	Тип	Описание
filename	string	Полный путь до файла
format	DataFormat	Тип кодировки (BASE64 | DER)

2.6.28. Примеры

Чтение сертификата из файла:

```
const cert = new trusted.cms.Certificate();
cert.load("./example.cer", trusted.DataFormat.PEM);
```

Импорт сертификата из памяти:

```
const cert = new trusted.cms.Certificate();
const data = fs.readFileSync("./export.cer");
cert.import(data, trusted.DataFormat.PEM);
```

Получение значений полей сертификата:

```
const cert = new trusted.cms.Certificate();
cert.load("./example.cer", trusted.DataFormat.PEM);
cert.version;
cert.subjectFriendlyName;
cert.issuerFriendlyName;
cert.subjectName;
cert.issuerName;
cert.notAfter;
cert.notBefore;
cert.serialNumber;
cert.thumbprint;
cert.keyUsage;
cert.signatureAlgorithm;
cert.signatureDigestAlgorithm;
cert.publicKeyAlgorithm;
cert.organizationName;
cert.OCSPUrls;
cert.CAIssuersUrls;
cert.isSelfSigned;
cert.isCA;
```

Сохранение сертификата в файл:

```
const cert = new trusted.cms.Certificate();
cert.load("./example.cer", trusted.DataFormat.PEM);
cert.save("./out.cer", trusted.DataFormat.DER);
```

2.7. Класс CertificateCollection

Метод	Возвращаемый тип	Описание
items	Certificate	Возвращает объект коллекции по его индексу
get length	number	Возвращает количество элементов коллекции
push	void	Добавляет элемент в коллекцию
pop	void	Удаляет один элемент коллекции
removeAt	void	Удаляет элемент коллекции по его индексу

2.7.1. Метод items

Возвращает объект коллекции по его индексу.

items(index: number): Certificate

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

2.7.2. Метод length

Возвращает количество элементов коллекции

get length(): number

2.7.3. Метод push

Добавляет элемент в коллекцию

push(cert: Certificate): void

Параметр	Тип	Описание
cert	Certificate	Добавляемый сертификат

2.7.4. Метод pop

Удаляет один элемент коллекции

pop(): void

2.7.5. Метод removeAt

Возвращает объект коллекции по его индексу.

removeAt(index: number): void

Параметр	Тип	Описание
index	number	Индекс элемента в коллекции

2.7.6. Примеры

Добавление сертификатов в коллекцию и получение количества элементов:

```
const certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("./cert1.cer"));
certs.push(trusted.pki.Certificate.load("./cert2.cer"));
certs.length; // 2
```

Удаление элемента из коллекции:

```
const certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("./cert1.cer"));
certs.push(trusted.pki.Certificate.load("./cert2.cer"));
certs.length; // 2

certs.pop();
certs.length; // 1
```

Получение элемента из коллекции:

```
const certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("./cert1.cer"));
certs.push(trusted.pki.Certificate.load("./cert2.cer"));
const cert = certs.items(0);
```

2.8. Класс CertificationRequest

Метод	Возвращаемый тип	Описание
set subject	void	Задает имя субъекта
get version	number	Возвращает версию запроса
set version	void	Задает версию запроса
set extensions	void	Задает расширения
get containerName	string	Возвращает имя контейнера
set containerName	void	Задает имя контейнера закрытого ключа
get pubKeyAlgorithm	string	Возвращает алгоритм публичного ключа сертификата
set pubKeyAlgorithm	void	Задает алгоритм публичного ключа сертификата
get exportableFlag	boolean	Возвращает значение экспортируемости
set exportableFlag	void	Задает значение экспортируемости
get newKeysetFlag	boolean	Возвращает флаг, будет ли генерировать новый набор ключей
set newKeysetFlag	void	Задает флаг, будет ли генерироваться новый набор ключей
save	void	Запись запроса в файл

2.8.1. Метод subject

Задает имя субъекта

subject(x509name: string | native.PKI.INameField[])

Параметр	Тип	Описание
x509name	string | native.PKI.INameField[]	Имя субъекта

2.8.2. Метод version

Возвращает или задает версию запроса (getters/setters).

get version(): number

set version(version: number)

Параметр	Тип	Описание
version	number	Версия запроса

2.8.3. Метод extensions

Задает расширения.

set extensions(exts: pki.ExtensionCollection)

Параметр	Тип	Описание
exts	ExtensionCollection	Коллекция расширений

2.8.4. Метод containerName

Возвращает или задает имя контейнера закрытого ключа (getters/setters).

get containerName(): string

set containerName(name: string)

Параметр	Тип	Описание
name	string	Имя контейнера

2.8.5. Метод pubKeyAlgorithm

Возвращает или задает алгоритм публичного ключа ключа (getters/setters).

get pubKeyAlgorithm(): string

set pubKeyAlgorithm(alg: string)

Параметр	Тип	Описание
alg	string	Алгоритм публичного ключа

2.8.6. Метод exportableFlag

Определяет, будет ли ключ экспортируемый или возвращает этот флаг.

get exportableFlag(): boolean

set exportableFlag(flag: boolean)

Параметр	Тип	Описание
flag	boolean	Экспортируемость ключа

2.8.7. Метод newKeysetFlag

Определяет, будут ли генерироваться новый ключевой набор или возвращает этот флаг.

get newKeysetFlag(): boolean

set newKeysetFlag(flag: boolean)

Параметр	Тип	Описание

flag	boolean	Если true, то генерируется новый ключевой набор

2.8.8. Метод save

Запись запроса на сертификат в файл.

save(filename: string, dataFormat: DataFormat = DEFAULT_DATA_FORMAT): void

Параметр	Тип	Описание
filename	string	Путь до файла, куда будет записан запрос
dataFormat?	DataFormat	Необязательный параметр. Тип кодировки. Значение по умолчанию: DataFormat.PEM

2.8.9. Примеры

Генерация запроса на сертификат с формированием нового ключевого набора:

```
const certReq = new trusted.pki.CertificationRequest();

const exts = new trusted.pki.ExtensionCollection();
const oid = new trusted.pki.Oid("keyUsage");
const ext = new trusted.pki.Extension(oid, "critical,digitalSignature");
exts.push(ext);
const atrs = [

      { type: "C", value: "RU" },
      { type: "CN", value: "Иван Иванов 2001" },
      { type: "L", value: "Yoshkar-Ola" },
      { type: "S", value: "Mari El" },
      { type: "O", value: "Test Org" },
      { type: "1.2.643.100.3", value: "12295279882" },
      { type: "1.2.643.3.131.1.1", value: "002465363366" }
];
certReq.subject = atrs;

certReq.version = 2;
certReq.extensions = exts;
certReq.exportableFlag = true;
certReq.pubKeyAlgorithm = "gost2012-256";
certReq.containerName = "containerName2012_256";
certReq.save("./certreq.req", trusted.DataFormat.PEM);
```

2.9. Класс Cipher

Метод	Возвращаемый тип	Описание
set ProvAlgorithm	void	Задает тип используемого провайдера
set recipientsCerts	void	Задает сертификаты получателей
encrypt | encryptAsync	void	Шифрование файла
decrypt | decryptAsync	void	Расшифрование файла
getRecipientInfos	cms.CertInfoCollection	Возвращает сведения и получателях зашифрованного сообщения

2.9.1. Метод ProvAlgorithm

Задает тип используемого провайдера. Не используется т.к. провайдер определяем по алгоритму поличного ключа сертификата.

set ProvAlgorithm(name: string)

Параметр	Тип	Описание
name	string	Допустимые имена провайдера: gost2001, gost2012_256, gost2012_512

2.9.2. Метод recipientsCerts

Задает сертификаты получателей.

set recipientsCerts(certs: CertificateCollection)

Параметр	Тип	Описание
certs	CertificateCollection	Коллекция сертификатов получателей, в адрес кого шифруем

2.9.3. Метод encrypt

Шифрование файла.

encrypt(filenameSource: string, filenameEnc: string, alg: EncryptAlg = DEFAULT_ENC_ALG, format: DataFormat = DEFAULT_DATA_FORMAT): void

Асинхронный метод:

encrypt(filenameSource: string, filenameEnc: string, done: (msg: string) => void, alg: EncryptAlg = DEFAULT_ENC_ALG, format: DataFormat = DEFAULT_DATA_FORMAT): void

Параметр	Тип	Описание
filenameSource	string	Путь до файла, который шифруем
filenameEnc	string	Путь до файла, куда запишем зашифрованный файл

alg	EncryptAlg	Алгоритм шифрования. Значение по умолчанию: GOST_28147. Поддерживаются: GOST_28147, GOST_R3412_2015_M, GOST_R3412_2015_K
format	DataFormat	Кодировка. Значение по умолчанию: PEM

2.9.4. Метод decrypt

Расшифрование файла.

decrypt(filenameEnc: string, filenameDec: string, format?: DataFormat): void

Асинхронный метод:

decryptAsync(filenameEnc: string, filenameDec: string, format?: DataFormat, done: (msg: string) => void): void

Параметр	Тип	Описание
filenameEnc	string	Путь до зашифрованного файла
filenameDec	string	Путь до файла, куда запишем расшифрованный файл
format?	DataFormat	Необязательный параметр. Тип кодировки.

2.9.5. Примеры

Шифрование и расшифрование файла:

```
const cipher = new trusted.pki.Cipher();

const certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("./cert1.cer", trusted.DataFormat.PEM));
cipher.recipientsCerts = certs;
cipher.encrypt("./test.txt", "./encAssym2001.txt.enc",
trusted.EncryptAlg.GOST_28147, trusted.DataFormat.PEM);
const cipher2 = new trusted.pki.Cipher();

cipher2.decrypt("./encAssym2001.txt.enc", "./decAssym2001.txt",

trusted.DataFormat.PEM);
```

2.10. Класс OCSP

Является оберткой над Криптопро OCSP SDK, Класс CResponse

Метод	Возвращаемый тип	Описание
Export	Buffer	Возвращает OCSP-ответ
Verify	number	Проверяет подпись OCSP-ответа и отсутствие неизвестных критических расширений

VerifyCertificate	number	Проверяет сертификат
get RespStatus	CPRespStatus	Поле "Status" OCSP-ответа
get SignatureAlgorithmOid	string	Алгоритм подписи OCSP-ответа
get Certificates	CertificateCollection	Возвращает сертификаты, содержащиеся в OCSP-ответе
get ProducedAt	Date	Время подписи OCSP-ответа
get RespNumber	number	Количество ответов со статусами сертификатов
RespIndexByCert	number	Поиск информации о статусе сертификата в OCSP-ответе по самому сертификату
get OcspCert	Certificate	Возвращает сертификат, службы актуальных статусов
getOcspCert	Certificate	Возвращает сертификат, службы актуальных статусов
Status	CPCertStatus	Статус сертификата
RevTime	Date	Время отзыва сертификата
RevReason	CPCrlReason	Причина отзыва сертификата
ThisUpdate	Date	Поле "ThisUpdate"
NextUpdate	Date	Поле "NextUpdate"

2.11. Класс TSPRequest

Является оберткой над КриптоПро TSP SDK, Класс CRequest

Метод	Возвращаемый тип	Описание
AddData	void	Добавляет очередной блок данных для создания запроса
get CertReq	boolean	Возвращает флаг для включения сертификата службы штампов в штамп
set CertReq	void	Задает флаг для включения сертификата службы штампов в штамп
get Nonce	boolean	Возвращает флаг для включения в запрос поля "Nonce"
set Nonce	void	Задает флаг для включения в запрос поля "Nonce"
get PolicyId	string	Возвращает идентификатор политики службы штампов (UNICODE)
set PolicyId	void	Задает идентификатор политики службы штампов (UNICODE)
get HashAlgOid	string	Возвращает алгоритм хэширования данных
set HashAlgOid	void	Задает алгоритм хэширования данных

get DataHash	Buffer	Возвращает хэш данных, включаемый в запрос на штамп времени
set DataHash	void	Задает хэш данных, включаемый в запрос на штамп времени

2.12. Класс TSP

Является оберткой над КриптоПро TSP SDK, Класс CStamp

Метод	Возвращаемый тип	Описание
Export	Buffer	Возвращает штамп времени
get Certificates	CertificateCollection	Возвращает сертификаты, содержащиеся в штампе
get TSACertificate	Certificate	Возвращает сертификат службы штампов
Verify	number	Проверяет подпись штампа времени и отсутствие неизвестных критических расширений
VerifyCertificate	number	Проверяет сертификат
get FailInfo	number	Поле "FailInfo" ответа службы штампов времени
get Status	number	Поле "Status" ответа службы штампов времени
get StatusString	string	Поле "StatusString" ответа службы штампов времени
get DataHashAlgOID	string	Алгоритм хэширования данных
get DataHash	Buffer	Хэш данных, на которые был выдан штамп времени
get PolicyID	string	Идентификатор политики службы штампов, согласно которой штамп был выпущен
get SerialNumber	Buffer	Серийный номер штампа
get Time	Date	Время штампа
get Accuracy	number	Точность времени в штампе
get Ordering	boolean	Поле "Ordering" штампа времени
get HasNonce	boolean	Определяет наличие поля "Nonce" в штампе времени
get TsaName	string	Название службы штампов, выдавшей данный штамп времени в виде строки
get TsaNameBlob	Buffer	Название службы штампов, выдавшей данный штамп времени, в бинарном виде

2.13. Класс PKCS12

Метод	Возвращаемый тип	Описание
load	void	Чтение PFX из файла
save	void	Запись PFX в файл

2.13.1. Метод load

Чтение PFX из файла.

load(filename: string): void

static load(filename: string): PKCS12

Параметр	Тип	Описание
filename	string	Полный путь до файла pfx

2.13.2. Метод save

Запись PFX в файл.

save(filename: string): void

Параметр	Тип	Описание
filename	string	Полный путь до файла

2.13.3. Примеры

```
const pkcs12 = new trusted.pki.PKCS12();

pkcs12.load("./pfx2012-256.pfx");
pkcs12.save("./out2012-256.pfx");
```

3. Пространство имен utils

3.1. Класс Csp

Метод	Возвращаемый тип	Описание
isGost2001CSPAvailable	boolean	Проверка доступности провайдера реализующего ГОСТ 2001
isGost2012_256CSPAvailable	boolean	Проверка доступности провайдера реализующего ГОСТ 2012 256
isGost2012_512CSPAvailable	boolean	Проверка доступности провайдера реализующего ГОСТ 2012 512
checkCPCSPLicense	boolean	Проверка действительности лицензии КриптоПро CSP
getCPCSPLicense	string	Возвращает лицензию КриптоПро CSP
getCPCSPVersion	string	Возвращает версию КриптоПро CSP
getCPCSPVersionPKZI	string	Возвращает версию КриптоПро CSP PKZI
getCPCSPVersionSKZI	string	Возвращает версию КриптоПро CSP SKZI
getCPCSPSecurityLvl	string	Возвращает уровень КриптоПро CSP
enumProviders	object[]	Возвращает доступные провайдеры
enumContainers	IContainerName[]	Возвращает имена контейнеров закрытых ключей
getCertificateFromContainer	Certificate	Возвращает сертификат из контейнера закрытого ключа
installCertificateFromContainer	void	Установка сертификата из контейнера
installCertificateToContainer	void	Установка сертификата в контейнер
deleteContainer	void	Удаление контейнера
getContainerNameByCertificate	string	Возвращает имя контейнера по сертификату
hasPrivateKey	boolean	Проверка сертификата на наличие привязки с закрытым ключом
buildChain | buildChainAsync	CertificateCollection	Строит и возвращает цепочку для сертификата
verifyCertificateChain | verifyCertificateChainAsync	boolean	Проверка цепочки сертификатов
isHaveExportablePrivateKey	boolean	Поверка экспортируемости закрытого ключа
certToPkcs12	PKCS12	Возвращает pfx по сертификату
importPkcs12	void	Устанавливает pfx в хранилище

3.1.1. Метод enumContainers

Возвращает имена контейнеров закрытых ключей.

static enumContainers(type: null, provName = ""): IContainerName[]

Параметр	Тип	Описание
type?	number	Необязательный параметр. Тип провайдера (для ГОСТ: 75, 80, 81). Если не задан, то будут использованы все доступные в системе провайдеры
provName?	string	Необязательный параметр. Имя криптопровайдера

3.1.2. Метод getCertificateFromContainer

Возвращает сертификат из контейнера закрытого ключа.

static getCertificateFromContainer(contName: string, provType: number, provName: string = ""): Certificate

Параметр	Тип	Описание
contName	string	Имя контейнера
provType	number	Тип провайдера (для ГОСТ: 75, 80, 81)
provName?	string	Необязательный параметр. Имя криптопровайдера

3.1.3. Метод installCertificateFromContainer

Установка сертификата из контейнера в хранилище.

static installCertificateFromContainer(contName: string, provType: number, provName = ""): void

Параметр	Тип	Описание
contName	string	Имя контейнера
provType	number	Тип провайдера (для ГОСТ: 75, 80, 81)
provName?	string	Необязательный параметр. Имя криптопровайдера

3.1.4. Метод installCertificateToContainer

Установка сертификата в контейнер.

static installCertificateToContainer(cert: pki.Certificate, contName: string, provType: number, provName = ""): void

Параметр	Тип	Описание
cert	Certificate	Сертификат, который будет записан в контейнер
contName	string	Имя контейнера
provType	number	Тип провайдера (для ГОСТ: 75, 80, 81)

provName?	string	Необязательный параметр. Имя криптопровайдера

3.1.5. Метод deleteContainer

Удаление контейнера.

static deleteContainer(contName: string, provType: number, provName = ""): void

Параметр	Тип	Описание
contName	string	Имя контейнера
provType	number	Тип провайдера (для ГОСТ: 75, 80, 81)
provName?	string	Необязательный параметр. Имя криптопровайдера

3.1.6. Метод getContainerNameByCertificate

Возвращает имя контейнера по сертификату.

static getContainerNameByCertificate(cert: pki.Certificate, category: string = "MY"): string

Параметр	Тип	Описание
cert	Certificate	Сертификат
category?	string	Необязательный параметр. Категория в хранилище сертификатов. Значение по умолчанию «MY»

3.1.7. Метод hasPrivateKey

Проверка сертификата на наличие привязки с закрытым ключом.

hasPrivateKey(cert: pki.Certificate): Boolean

Параметр	Тип	Описание
cert	Certificate	Сертификат, который проверяем на наличие привязки

3.1.8. Метод buildChain

Строит и возвращает цепочку для сертификата.

static buildChain(cert: pki.Certificate): pki.CertificateCollection

Асинхронный метод: buildChainAsync(cert: pki.Certificate, done: (error: string, certs: pki.CertificateCollection) => void): void

Параметр	Тип	Описание

cert	Certificate	Сертификат, для которого строим цепочку

3.1.9. Метод verifyCertificateChain

Проверка цепочки сертификатов.

static verifyCertificateChain(cert: pki.Certificate): Boolean

Асинхронный метод: verifyCertificateChainAsync(cert: pki.Certificate, done: (error: string, result: boolean) => void): void

Параметр	Тип	Описание
cert	Certificate	Сертификат, для которого проверяем цепочку

3.1.10. Метод isHaveExportablePrivateKey

Поверка экспортируемости закрытого ключа привязанного к сертификату.

static isHaveExportablePrivateKey(cert: pki.Certificate): Boolean

Параметр	Тип	Описание
cert	Certificate	Сертификат

3.1.11. Метод certToPkcs12

Возвращает PKCS12 (pfx) по сертификату.

static certToPkcs12(cert: pki.Certificate, exportPrivateKey: boolean, password?: string): pki.PKCS12

Параметр	Тип	Описание
cert	Certificate	Сертификат, экпортируемый в pfx
exportPrivateKey	boolean	Флаг, определяющий будет ли экспортироваться в pfx закрытый ключ
password?	string	Необязательный параметр. Пароль, который используется для защиты pfx

3.1.12. Метод importPkcs12

Устанавливает pfx в хранилище.

static importPkcs12(p12: pki.PKCS12, password?: string): void

Параметр	Тип	Описание
p12	PKCS12	Импортируемый pfx

password?	string	Необязательный параметр. Пароль, который используется для чтения pfx (если он был защищен)

3.1.13. Примеры

Проверка сертификата на наличие привязки с закрытым ключом и установка сертификата из контейнера:

```
const cert = new trusted.pki.Certificate();
cert.load("./TrustedCrypto2012-512.cer");
const res = trusted.utils.Csp.hasPrivateKey(cert);
if (res) {
      const contName = trusted.utils.Csp.getContainerNameByCertificate(cert, "MY");
      trusted.utils.Csp.installCertificateFromContainer(containerName, 81,
      "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider");

}
```

Построение и проверка цепочки сертификатов:

```
let chain = new trusted.pki.CertificateCollection();
const cert = new trusted.pki.Certificate();
cert.load("./TrustedCrypto2012-256.cer");
chain = trusted.utils.Csp.buildChain(cert);
const chainStatus = trusted.utils.Csp.verifyCertificateChain(cert);
```

Формирование PKCS12 (pfx) по сертификату:

```
const cert = trusted.pki.Certificate.load("./TrustedCrypto2012-256.cer",
 trusted.DataFormat.DER);
const p12Res = trusted.utils.Csp.certToPkcs12(cert, true, "1");
```

3.2. Класс ConnectionSettings

Метод	Возвращаемый тип	Описание
get AuthType	number	Возвращает тип аутентификации службы
set AuthType	void	Задает тип аутентификации службы
get Address	string	Возвращает адрес службы
set Address	void	Задает адрес службы
get UserName	string	Возвращает имя пользователя для использования при аутентификации
set UserName	void	Задает имя пользователя для использования при аутентификации
get Password	string	Возвращает пароль для использования при аутентификации
set Password	void	Задает пароль для использования при аутентификации
get ClientCertificate	Certificate	Возвращает клиентский сертификат, для подключения к службе
set ClientCertificate	void	Задает клиентский сертификат, для подключения к службе
get ProxyAuthType	number	Возвращает тип аутентификации для прокси-сервера
set ProxyAuthType	void	Задает тип аутентификации для прокси-сервера
get ProxyAddress	string	Возвращает адрес прокси-сервера
set ProxyAddress	void	Задает адрес прокси-сервера
get ProxyUserName	string	Возвращает имя пользователя для использования прокси-сервера
set ProxyUserName	void	Задает имя пользователя для использования прокси-сервера
get ProxyPassword	string	Возвращает пароль для использования при аутентификации для прокси-сервера
set ProxyPassword	void	Задает пароль для использования при аутентификации для прокси-сервера

3.2.1. Метод AuthType

Возвращает или задает тип аутентификации службы (getters/setters).

get AuthType(): number

set AuthType(authType: number)

Параметр	Тип	Описание

authType	number	Тип аутентификации. Возможные значения из перечисления (enumAuthTypes): 1. atAnonymous - Анонимная аутентификация. 2. atBasic - Обычная аутентификация (пароль отправляется в текстовом формате). 3. atNTLM - Встроенная проверка подлинности Windows. 4. atDigest - Краткая проверка для серверов доменов Windows. 5. atNegotiate - Встроенная проверка подлинности Windows или Kerberos (в зависимости от ОС).

3.2.2. Метод Address

Возвращает или задает адрес службы (getters/setters).

get Address(): string

set Address(addr: string)

Параметр	Тип	Описание
addr	string	Адрес службы

3.2.3. Метод UserName

Возвращает или задает имя пользователя для использования при аутентификации (getters/setters).

get UserName(): string

set UserName(usrName: string)

Параметр	Тип	Описание
usrName	string	Имя пользователя

3.2.4. Метод Password

Возвращает или задает пароль для использования при аутентификации (getters/setters).

get Password(): string

set Password(passwd: string)

Параметр	Тип	Описание
passwd	string	Пароль

3.2.5. Метод ClientCertificate

Возвращает или задает клиентский сертификат, для подключения к службе (getters/setters).

get ClientCertificate(): pki.Certificate

set ClientCertificate(clntCert: pki.Certificate)

Параметр	Тип	Описание
clntCert	Certificate	Клиентский сертификат

3.2.6. Метод ProxyAuthType

Возвращает или задает тип аутентификации для прокси-сервера (getters/setters).

get ProxyAuthType(): number

set ProxyAuthType(authType: number)

Параметр	Тип	Описание
authType	number	Тип аутентификации (enumAuthTypes)

3.2.7. Метод ProxyAddress

Возвращает или задает адрес прокси-сервера (getters/setters).

get ProxyAddress(): string

set ProxyAddress(addr: string)

Параметр	Тип	Описание
addr	string	Адрес прокси

3.2.8. Метод ProxyUserName

Возвращает или задает имя пользователя для использования прокси-сервера (getters/setters).

get ProxyUserName(): string

set ProxyUserName(usrName: string)

Параметр	Тип	Описание
usrName	string	Имя пользователя прокси-сервера

3.2.9. Метод ProxyPassword

Возвращает или задает имя пользователя для использования при аутентификации для прокси-сервера (getters/setters).

get ProxyPassword(): string

set ProxyPassword(passwd: string)

Параметр	Тип	Описание
passwd	string	Пароль

3.3. Класс ModuleInfo

Метод	Возвращаемый тип	Описание
get version	string	Возвращает версию модуля
get name	string	Возвращает имя модуля
get cadesEnabled	boolean	Проверка доступности CAdES

3.4. Класс Tools

Метод	Возвращаемый тип	Описание
stringFromBase64	string	Декодирует строку из BASE64
stringToBase64	string	Кодирует строку в BASE64

4. Пространство имен pkistore

4.1. Класс Filter

Класс используется для фильтрации (поиска) объектов типа PkiItem.

Метод	Возвращаемый тип	Описание
set types	void	Задает тип объекта PkiItem
set providers	void	Задает тип провайдера
set categorys	void	Задает категорию (хранилище)
set hash	void	Задает отпечаток (SHA-1)
set subjectName	void	Задает имя субъекта
set subjectFriendlyName	void	Задает имя субъекта (CN)
set issuerName	void	Задает имя издателя
set issuerFriendlyName	void	Задает имя издателя (CN)
set serial	void	Задает серийный номер

4.2. Класс PkiStore

Метод	Возвращаемый тип	Описание
addProvider	void	Добавляет провайдер. В данной версии только ProviderCryptopro
find	IPkiItem[]	Поиск объектов в хранилище
getItem	Certificate | CRL	Возвращает объект по его описанию PkiItem
get certs	CertificateCollection	Возвращает все сертификаты их хранилища
addCert	void	Импорт сертификата в хранилище
addCrl	void	Импорт CRL в хранилище
deleteCert	void	Удаление сертификата из хранилища
deleteCrl	void	Удаление CRL из хранилища

4.2.1. Метод addProvider

Добавляет провайдер. В данной ревизии только ProviderCryptopro

addProvider(provider: Provider): void

Параметр	Тип	Описание
provider	Provider	Провайдер. В данной версии только ProviderCryptopro

4.2.2. Метод find

Поиск объектов в хранилище.

find(ifilter?: IFilter): IPkiItem[]

Параметр	Тип	Описание
filter?	IFilter	Необязательный параметр. Параметры фильтрации

4.2.3. Метод getItem

Возвращает объект по его описанию IPkiItem.

getItem(item: IPkiItem)

Параметр	Тип	Описание
item	IPkiItem	Описание объекта

4.2.5. Метод certs

Возвращает все сертификаты из хранилища.

get certs():CertificateCollection

4.2.6. Метод addCert

Импорт сертификата в хранилище.

addCert(provider: Provider, category: string, cert: pki.Certificate, contName?: string, provType?: number): void

Параметр	Тип	Описание
provider	Provider	Провайдер. В данной версии только ProviderCryptopro
category	string	Название хранилища
cert	Certificate	Импортируемый сертификат
contName?	string	Необязательный параметр. Имя контейнера
provType?	number	Необязательный параметр. Тип провайдера. Для ГОСТ 75, 80, 81

4.2.7. Метод addCrl

Импорт CRL в хранилище.

addCrl(provider: Provider, category: string, crl: pki.CRL): void

Параметр	Тип	Описание
provider	Provider	Провайдер. В данной версии только ProviderCryptopro
category	string	Название хранилища
crl	CRL	Импортируемый СОС (CRL)

4.2.8. Метод deleteCert

Удаление сертификата из хранилища.

deleteCert(provider: Provider, category: string, cert: Certificate): void

Параметр	Тип	Описание
provider	Provider	Провайдер. В данной версии только ProviderCryptopro
category	string	Название хранилища
cert	Certificate	Удаляемый сертификат

4.2.9. Метод deleteCrl

Удаление СОС (CRL) из хранилища.

deleteCrl(provider: Provider, category: string, crl: pki.CRL): void

Параметр	Тип	Описание
provider	Provider	Провайдер. В данной версии только ProviderCryptopro
category	string	Название хранилища
crl	CRL	Удаляемый СОС (CRL)

4.3. Класс ProviderCryptopro

Класс при инициализации перечитывает объекты их локального хранилища. Методы класса ProviderCryptopro используются из класса PkiStore. На уровень typescript вынесен один метод.

Метод	Возвращаемый тип	Описание
hasPrivateKey	void	Проверка сертификата на наличие привязки с закрытым ключом

4.4. Интерфейсы

4.4.1. IPkiKey

Свойство	Тип	Описание
`encrypted?`	boolean	Зашифрован ли ключ

4.4.2. IPkiCrl

Свойство	Тип	Описание
`authorityKeyid?`	string	Authority Key Identifier
`crlNumber?`	string	Номер CRL
`issuerName?`	string	Полное имя издателя
`issuerFriendlyName?`	string	CN имени издателя
`lastUpdate?`	string	Дата последнего обновления
`nextUpdate?`	string	Дата следующего обновления

4.4.3. IPkiRequest

Свойство	Тип	Описание
`subjectName?`	string	Полное имя субъекта
`subjectFriendlyName?`	string	CN имени субъекта
`key?`	string	Привязка с закрыты ключом

4.4.3. IPkiCertificate

Свойство	Тип	Описание
subjectName?	string	Полное имя субъекта
subjectFriendlyName?	string	CN имени субъекта
issuerName?	string	Полное имя издателя
issuerFriendlyName?	string	CN имени издателя
notAfter?	string	Дата окончания действия сертификата
notBefore?	string	Дата начала действия сертификата
serial?	string	Серийный номер сертификата
key?	string	Привязка с закрыты ключом
organizationName?	string	Организация
signatureAlgorithm?	string	Алгоритм подписи
signatureDigestAlgorithm?	string	Хэш алгоритм подписи
publicKeyAlgorithm?	string	Алгоритм публичного ключа

4.4.3. IPkiItem

extends IPkiCrl, IPkiCertificate, IPkiRequest, IPkiKey

Свойство	Тип	Описание
format	string	PEM | DER
type	string	CRL | CERTIFICATE | KEY | REQUEST
uri	string	Ссылка на объект. В данной версии не используется
provider	string	Имя провайдера
category	string	Категория (название хранилища)
hash	string	Хэш (SHA-1 отпечаток)

5. Пространство имен common

5.1. Класс Logger

Метод	Возвращаемый тип	Описание
start	void	Старт логгирования
stop	void	Завершение логгирования
clear	void	Очистка файла журнала (log)

5.1.1. Метод start

Старт логгирования операций и ошибок. Запись ведется в файл.

start(filename: string, level: LoggerLevel = DEFAULT_LOGGER_LEVEL): void

Параметр	Тип	Описание
filename	string	Путь до файла для записи журнала
Level?	LoggerLevel	Необязательный параметр. Уровень логгирования. Значение по умолчанию: LoggerLevel.ALL