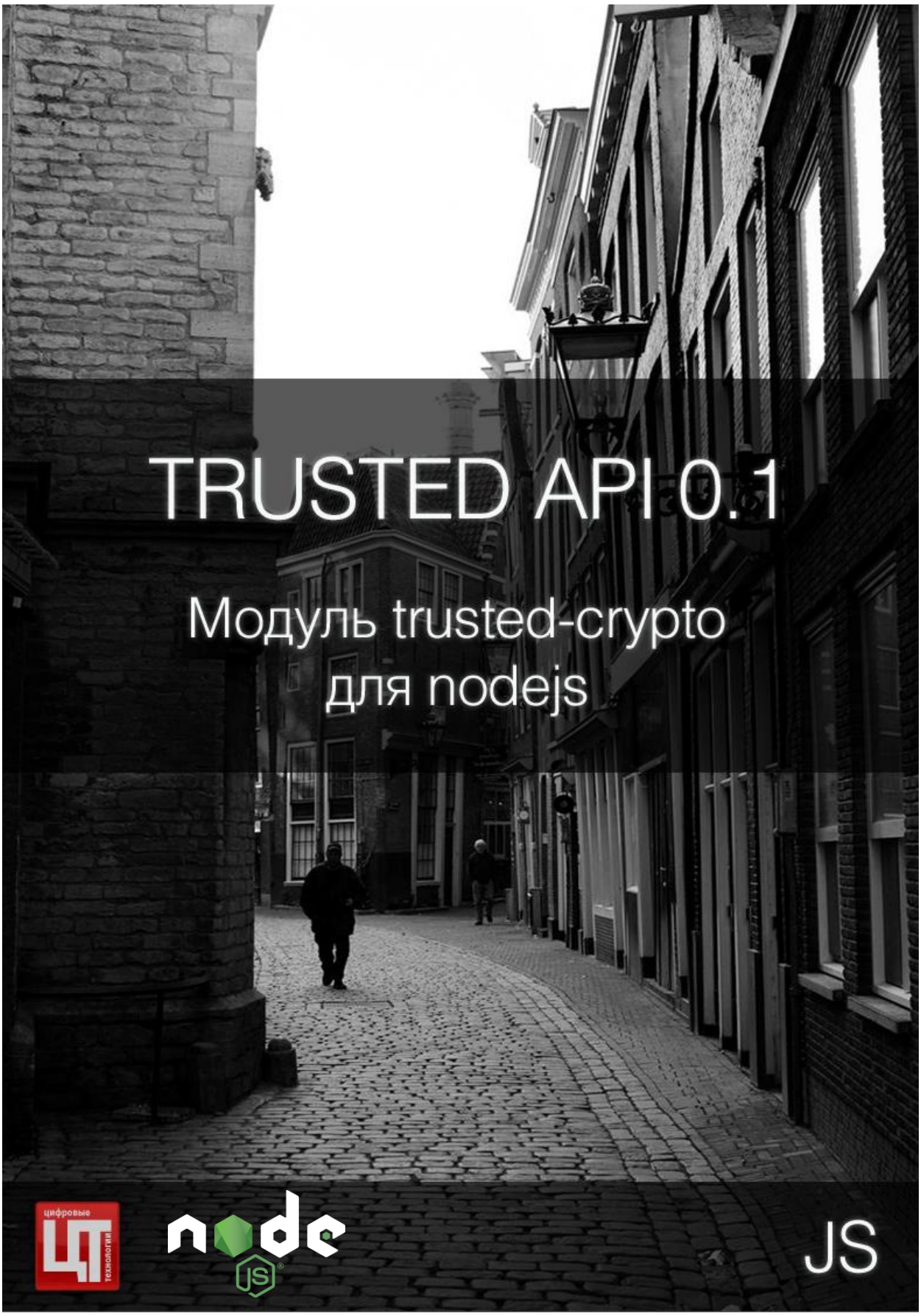
TRUSTED API 0.1
Модуль trusted-crypto
для nodejs
цифровые технологии
node
JS

Оглавление

Состав продукта

Продукт trusted-crypto представляет собой внешний модуль, который работает с приложением Nodejs и предоставляет расширенные возможности по работе с цифровыми сертификатами, подписью, шифрованием данных, работе с хранилищами ключей и сертификатов.

Порядок установки

Для сборки модуля необходимо сделать предварительную установку внешних модулей для Nodejs. Установка выполняется запуском последовательности консольных команд:
> npm install -g typescript
> npm install -g tsd
> npm install -g mocha

Сборка модуля при условии установки MS Visual Studio 2013 на Windows, g++ и библиотек разработки на Ubuntu и XCode на MacOS, выполняется с помощью консольной команды:
> npm install

Запуск тестов для проверки основных функций модуля trusted-crypto выполняется командой:
> npm test

Определения

CryptoMethod {
SYMMETRIC = 0,
ASSYMETRIC = 1,
}

DataFormat {
DER = 0,
PEM = 1,
}

PublicExponent {
RSA_3 = 0,
RSA_F4 = 1,
}

Cms

Класс SignedData

INTERFACES ----------

НАЗВАНИЕ	ПАРАМЕТРЫ	ОПИСАНИЕ
ISIGNEDDATACONTENT	type: SignedDataContentType data: string | Buffer	

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
CONTENT	ISignedDataContent	Возвращает или задает содержимое подписи
POLICIES	Array<string>	Возвращает или задает политики подписи

METHODS ----------

МЕТОД	ОПИСАНИЕ
ISDETACHED()	Возвращает true если подпись откреплennая
CERTIFICATES()	Возвращает коллекцию сертификатов
SIGNERS()	Возвращает коллекцию подписчиков
LOAD(STRING, DATAFORMAT)	Выполняет чтение подписи из файла
IMPORT(BUFFER, DATAFORMAT)	Выполняет чтение подписи из памяти
SAVE(STRING, DATAFORMAT)	Выполняет сохранение подписи в файл
EXPORT(DATAFORMAT)	Выполняет сохранение подписи в память
CREATESIGNER(CERTIFICATE, KEY)	Создает нового подписчика
VERIFY(CERTIFICATECOLLECTION)	Выполняет проверку подписи
SIGN()	Выполняет подпись данных

isDetached() : boolean

Возвращает true если подпись откреплennая

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("test.sig", trusted.DataFormat.PEM);
console.log(cms.isDetached()); // false
```

---

certificates(index: number): Certificate

Возвращает сертификат по индексу

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
```

```
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
var certificate = cms.certificates(0);
console.log(certificate.signatureAlgorithm); // sha256
```

---

certificates(): CertificateCollection

Возвращает коллекцию сертификатов CertificateCollection

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
var certificates = cms.certificates();
console.log(certificates.length); // 1
```

---

signers(index: number): Signer

Возвращает подписчика по индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("sigdoc.sig", trusted.DataFormat.PEM);
var signer = cms.signers(0);
console.log("Signer digest name:", signer.digestAlgorithm.name); // Signer digest name: sha1
```

---

signers(): SignerCollection

Возвращает коллекцию подписчиков

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("sigdoc.sig", trusted.DataFormat.PEM);
var signers = cms.signers();
for (var i = 0; i < signers.length; i++){
    var signer = signers.items(i);
    console.log("Signer digest name:", signer.digestAlgorithm.name); // Signer digest name:
sha1
```

---

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

Чтение подписи из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
```

---

static load(filename: string, format: DataFormat): SignedData

Чтение подписанных данных из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

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

Чтение подписи из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
BUFFER	buffer	Буфер
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

static import(buffer: Buffer, format: DataFormat): SignedData

Чтение подписанных данных из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
BUFFER	buffer	Буфер
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

export(format: DataFormat): Buffer

Сохранение подписи в память

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

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

Сохранение подписи в файл

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
cms.save("testsig1.sig", trusted.DataFormat.PEM);
```

---

createSigner(cert: Certificate, key: Key, digestName: string): Signer

Создание нового подписчика

ПАРАМЕТР	ТИП	ОПИСАНИЕ

CERT	Certificate	Сертификат подписчика
KEY	Key	Закрытый ключ сертификата подписчика

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert = trusted.pki.Certificate.load("cert1.crt", trusted.DataFormat.PEM);
var key = trusted.pki.Key.readPrivateKey("cert1.key", trusted.DataFormat.PEM, "");
var sd = new trusted.cms.SignedData();
var signer = sd.createSigner(cert, key);
```

---

verify(certs: CertificateCollection): boolean

Проверка подписи

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERTS	CertificateCollection	Коллекция дополнительных сертификатов

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
console.log(cms.verify()); //true
```

---

sign(): void

Создание подписи

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert = trusted.pki.Certificate.load("./test/cert.crt", trusted.DataFormat.PEM);
var key = trusted.pki.Key.readPrivateKey("./test/cert.key", trusted.DataFormat.PEM, "");
var sd = new trusted.cms.SignedData();
var signer = sd.createSigner(cert, key, "sha1");
  sd.content = {data: 'Hellow word'};
  sd.sign();
  sd.save("testsig.sig", trusted.DataFormat.PEM);
```

Класс Signer

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
CERTIFICATE	Certificate	Задает или возвращает сертификат подписчика
DIGESTALGORITHM	Algorithm	Возвращает хэш алгоритм проверки содержимого
SIGNERID	SignerId	Возвращает идентификатор подписчика

METHODS ----------

МЕТОД	ОПИСАНИЕ
SIGNEDATTRIBUTES(SIGNERATTRIBUTECOLLECTION)	Возвращает коллекцию подписанных атрибутов
UNSIGNEDATTRIBUTES(SIGNERATTRIBUTECOLLECTION)	Возвращает коллекцию неподписанных атрибутов
VERIFYCONTENT()	Возвращает результат проверки подписанного контента
VERIFY()	Возвращает результат проверки атрибутов подписи

signedAttributes(): SignerAttributeCollection

возвращает коллекцию подписанных атрибутов

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
var signer = cms.signers(0);
var signedAttributes = signer.signedAttributes();
console.log(signer.digestAlgorithm.name);// sha1
console.log(signer.signedAttributes.length);//1
```

---

signedAttributes(index: number): Attribute

возвращает атрибут из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

unsignedAttributes(): SignerAttributeCollection

возвращает коллекцию неподписанных атрибутов

unsignedAttributes(index: number): Attribute

возвращает атрибут из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
```

```
var signer = cms.signers(0);
var unsignedAttributes = signer.unsignedAttributes();
console.log(unsignedAttributes.length);//-1
```

---

verifyContent(v: ISignedDataContent): boolean

возвращает true, если подписанный контент не был изменен

ПАРАМЕТР	ТИП	ОПИСАНИЕ
v	any	Контент

verify(): boolean

возвращает результат проверки атрибутов подписи

Класс SignerCollection

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LENGTH	number	Возвращает размер коллекции

METHODS ----------

МЕТОД	ОПИСАНИЕ
ITEMS	Возвращает элемент из коллекции по заданному индексу

items(index: number): Signer

Возвращает элемент из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
var signers = cms.signers();
var signer = signers.items(0);
console.log(signers.length);// 1
console.log(signer.digestAlgorithm.name); //sha1
```

Класс SignerAtributeCollection

Представление коллекции атрибутов подписчика

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LENGTH	number	Возвращает размер коллекции

METHODS ----------

МЕТОД	ОПИСАНИЕ
ITEMS	Возвращает элемент из коллекции по заданному индексу
PUSH	Добавляет новый элемент в коллекцию
REMOVEAT	Удаляет элемент из коллекции по заданному индексу

push(attr: Attribute): void

добавляет новый элемент в коллекцию

ПАРАМЕТР	ТИП	ОПИСАНИЕ
ATTR	Attribute	Новый элемент коллекции

removeAt(index: number): void

удаляет элемент из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

items(index: number): Attribute

возвращает элемент коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
var signer = cms.signers(0);
var signedAttributes = signer.signedAttributes();
console.log(signedAttributes.length);//-1
```

Класс SignerId

Представление идентификационной информации подписчика

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
ISSUERNAME()	string	Возвращает полное имя эмитента
SERIALNUMBER()	string	Возвращает серийный номер
KEYID()	string	Возвращает идентификатор ключа

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cms = trusted.cms.SignedData.load("testsig.sig", trusted.DataFormat.PEM);
var signer = cms.signers(0);
var signerId = signer.signerId;
console.log(typeof signerId.issuerName); //string
console.log(typeof signerId.serialNumber); // string
console.log(typeof signerId.keyId); //string
```

Класс CmsRecipientInfo

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
ISSUERNAME	STRING	Возвращает полное имя издателя сертификата
SERIALNUMBER	STRING	Возвращает серийный номер сертификата

METHODS ----------

МЕТОД	ОПИСАНИЕ
KTRICERTCMP	Возвращает результат сравнения сертификатов по CMS_RecipientInfo структуре

ktriCertCmp(cert: pki.Certificate): number

возвращает результат сравнения сертификатов по CMS_RecipientInfo структуре

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cipher = new trusted.pki.Cipher();
var ris = cipher.getRecipientInfos("/encAssym.enc", trusted.DataFormat.PEM);
console.log(ris.length);
ri = ris.items(0);
console.log(ri.issuerName);
console.log(ri.serialNumber);
console.log(ri.ktriCertCmp(trusted.pki.Certificate.load("/cert1.crt",
trusted.DataFormat.PEM));
```

Класс CmsRecipientInfoCollection

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LENGTH	NUMBER	Возвращает размер коллекции получателей

METHODS ----------

МЕТОД	ОПИСАНИЕ
PUSH	Добавляет новый элемент в коллекцию
POP	Удаляет последний элемент из коллекции
REMOVEAT	Удаляет элемент по индексу из коллекции

push(ri: CmsRecipientInfo): void

добавляет новый элемент в коллекцию

pop(): void

удаляет последний элемент из коллекции

removeAt(index: number): void

удаляет элемент из коллекции по индексу

Pki

Определения

```
KeyUsageFlags {
  DigitalSignature = 128,
  NonRepudiation = 64,
  KeyEncipherment = 32,
  DataEncipherment = 16,
  KeyAgreement = 8,
  KeyCertSign = 4,
  CrlSign = 2,
  EncipherOnly = 1,
  DecipherOnly = 32768,
}
```

Класс Algorithm

Представление X509_ALGOR

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.ALGORITHM)	Конструктор с указанием алгоритма – через экземпляр объекта
CONSTRUCTOR(NAME: STRING)	Конструктор с указанием строкового названия алгоритма

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
NAME	string	Возвращает название алгоритма
TYPEID	Oid	Возвращает идентификатор алгоритма

METHODS ----------

МЕТОД	ОПИСАНИЕ
DUPLICATE	Возвращает копию алгоритма
ISDIGEST	Возвращает true если алгоритм предназначен для вычисления хэш

constructor()

Конструктор для создания экземпляра объекта по умолчанию.

constructor(handle: native.PKI.Algorithm)

Конструктор для создания экземпляра объекта на основе существующего экземпляра (Конструктор копированием.)

constructor(name: string)

Конструктор для создания экземпляра объекта по указанному имени алгоритма.

duplicate(): Algorithm

Возвращает копию алгоритма

isDigest(): boolean

Возвращает true если алгоритм предназначен для вычисления хэша.

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var alg = new trusted.pki.Algorithm('SHA');
console.log(alg.typeId.shortName); // SHA
console.log(alg.name);// sha
console.log(alg.duplicate().name); // sha
console.log(alg.isDigest()); //true
```

Класс Attribute

Представление X509_ATTR

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(HANDLE: NATIVE.PKI.ATTRIBUTE)	Конструктор с указанием атрибута – через экземпляр объекта

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
ASNTYPE	number	Задает ASN1 тип атрибута
TYPEID	Oid	Задает идентификатор атрибута

METHODS ----------

МЕТОД	ОПИСАНИЕ
DUPLICATE	Возвращает копию алгоритма
EXPORT	Возвращает атрибут в DER кодировке
VALUES	Возвращает коллекцию значений атрибута. Значения представляются в DER формате

dupicate(): Attribute

Возвращает копию атрибута

export(): Buffer

Возвращает атрибут в DER кодировке

values(): AttributeValueCollection

Возвращает коллекцию значений атрибута. Значения представляются в DER формате

values(index: number): Buffer

возвращает коллекцию значений атрибута. Значения представляются в DER формате

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

Класс AttributeValueCollection

Представление коллекции X509_ATTR

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(HANDLE: NATIVE.PKI.ATTRIBUTEVALUECOLLECTION)	Конструктор с указанием коллекции атрибутов – через экземпляр объекта

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LENGTH	number	Возвращает количество элементов в коллекции

METHODS ----------

МЕТОД	ОПИСАНИЕ
PUSH	Добавляет новый элемент в коллекцию
POP	Удаляет последний элемент из коллекции
REMOVEAT	Удаляет элемент коллекции по заданному индексу
ITEMS	Возвращает элемент из коллекции по заданному индексу

push(val: Buffer): void

добавляет новый элемент в коллекцию

ПАРАМЕТР	ТИП	ОПИСАНИЕ
VAL	Buffer	Новое значение коллекции

pop(): void

удаляет последний элемент из коллекции

removeAt(index: number): void

удаляет элемент коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

items(index: number): Buffer

возвращает элемент из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

Класс Certificate

Представление' X509' сертификата

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CERTIFICATE)	Конструктор с указанием сертификата – через экземпляр объекта

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
VERSION	number	Возвращает серийный номер сертификата
SERIALNUMBER	string	Возвращает серийный номер сертификата
TYPE	number	Возвращает тип сертификата
KEYUSAGE	number	Возвращает набор флагов KeyUsageFlags, задающих назначение ключа сертификата
ISSUERFRIENDLYNAME	string	Возвращает пользовательское имя издателя сертификата
ISSUERNAME	string	Возвращает полное имя издателя сертификата
SUBJECTFRIENDLYNAME	string	Возвращает пользовательское имя владельца сертификата
SUBJECTNAME	string	Возвращает полное имя владельца сертификата
NOTBEFORE	Date	Возвращает время с которого сертификат считается действительным
NOTAFTER	Date	Возвращает время до которого сертификат считается действительным
THUMBPRINT	string	Возвращает алгоритм подписи
SIGNATUREALGORITHM	string	Возвращает алгоритм подписи
ORGANIZATIONNAME	string	Возвращает название организации

METHODS ----------

МЕТОД	ОПИСАНИЕ
COMPARE	Сравнение сертификатов
EQUALS	Сравнение сертификатов
HASH	Вычисление значения хэша сертификата
DUPLICATE	Создает копию сертификата
LOAD	Чтение сертификата из файла
IMPORT	Чтение сертификата из памяти
EXPORT	Сохранение сертификата в память
SAVE	Сохранение сертификата в файл

compare(cert: Certificate): number

сравнение сертификатов

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат для сравнения

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert1 = trusted.pki.Certificate.load("test.crt");
var cert2 = trusted.pki.Certificate.load("test-ru.crt");
console.log (cert1.compare(cert2));  //  1
console.log(cert2.compare(cert1));   // -1
```

```
console.log(cert1.compare(cert1));    // 0
```

---

equals(cert: Certificate): boolean

сравнение сертификатов

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат для сравнения

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert1 = trusted.pki.Certificate.load("test.crt");
var cert2 = trusted.pki.Certificate.load("test-ru.crt");
console.log(cert1.equals(cert2));       //false
console.log(cert1.equals(cert1));       // true
```

---

hash(algorithm?: string): String

вычисление значения хэша сертификата

ПАРАМЕТР	ТИП	ОПИСАНИЕ
ALGORITHM	string	Имя хэш алгоритма. Опционально. По умолчанию sha1

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert1 = trusted.pki.Certificate.load("test.crt");
console.log(cert1.hash());
```

---

duplicate(): Certificate

Создает копию сертификата

ПАРАМЕТР	ТИП	ОПИСАНИЕ
ALGORITHM	string	Имя хэш алгоритма. Опционально. По умолчанию sha1

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert1 = trusted.pki.Certificate.load("test.crt");
var cert2 = cert1.duplicate();
console.log(cert1.thumbprint === cert2.thumbprint); // true
```

---

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

чтение сертификата из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert = trusted.pki.Certificate.load("test.crt");
console.log(cert.serialNumber);
```

---

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

чтение сертификата из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

import(buffer: Buffer, format?: DataFormat): void

чтение сертификата из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
BUFFER	Buffer	Буфер памяти
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

static import(buffer: Buffer, format?: DataFormat): Certificate

чтение сертификата из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
BUFFER	Buffer	Буфер памяти
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

export(format?: DataFormat): Buffer

сохранение сертификата в память

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

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

сохранение сертификата в файл

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cert = trusted.pki.Certificate.load("test.crt");
cert.save ('test_new.crt', trusted.DataFormat.PEM);
```

Класс CertificationRequest

Класс для создания ASN.1 структуры CertificationRequest, которая представлена как: CertificationRequest ::= SEQUENCE { certificationRequestInfo CertificationRequestInfo, signatureAlgorithm AlgorithmIdentifier{{ SignatureAlgorithms }}, signature BIT STRING }

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CERTIFICATIONREQUEST)	Конструктор копированием экземпляра объекта CertificationRequest

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
PEMSTRING	Buffer	Получение запроса на сертификат

METHODS ----------

МЕТОД	ОПИСАНИЕ
LOAD	Чтение запроса из файла
SIGN	Подпись запроса
VERIFY	Проверка запроса на сертификат

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

чтение запроса из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cr = new trusted.pki.CertificationRequest();
cr.load("test.csr");
console.log(cr.PEMString);
```

---

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

чтение запроса из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

sign(key: Key): void

подпись запроса

ПАРАМЕТР	ТИП	ОПИСАНИЕ

KEY	Key	Ключевая пара

verify(): boolean

проверка подписи запроса на сертификат

Класс CertificationRequestInfo

Класс для создания ASN.1 структуры CertificationRequestInfo, которая представлена как: CertificationRequestInfo ::= SEQUENCE { version INTEGER { v1(0) } (v1,...), subject Name, subjectPKInfo SubjectPublicKeyInfo{{ PKInfoAlgorithms }}, attributes [0] Attributes{{ CRIAttributes }} }

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CERTIFICATIONREQUESTINFO)	Конструктор копированием экземпляра объекта CertificationRequestInfo

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
SUBJECT	String	Возвращает заголовок
PUBKEY	Key	Возвращает публичный ключ
VERSION	number	Возвращает номер версии

Класс CertificationCollection

Представление коллекции `X509` сертификатов

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CERTIFICATECOLLECTION)	Конструктор копированием экземпляра объекта CertificateCollection

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LENGTH	number	Возвращает размер коллекции

METHODS ----------

МЕТОД	ОПИСАНИЕ
ITEMS	Возвращает элемент из коллекции по заданному индексу
PUSH	Добавляет новый элемент в коллекцию
POP	Удаляет последний элемент их коллекции
REMOVEAT	Удаляет элемент из коллекции по заданному индексу

items(index: number): Certificate

Возвращает элемент из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("test.crt"));
var cert = certs.items(0);
console.log(cert.version); //2
```

---

push(cert: Certificate): void

Добавляет новый элемент в коллекцию

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Элемент для добавления в коллекцию

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("test.crt"));
console.log(certs.length); //1
```

---

pop(): void

Удаляет последний элемент их коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("test.crt"));
certs.pop();
console.log(certs.length); //0
```

---

removeAt(index: number): void

Удаляет элемент из коллекции по заданному индексу

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var certs = new trusted.pki.CertificateCollection();
certs.push(trusted.pki.Certificate.load("test.crt"));
certs.push(trusted.pki.Certificate.load("test.crt"));
certs.removeAt(0);
console.log(certs.length); //1
```

Класс CertStore

Класс для организации стека хранилищ сертификатов и ключей. Организуется кэш (JSON) для формирования описателей объектов хранения.

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CERTSTORE)	Конструктор копированием экземпляра объекта CertStore

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LISTCERTSTORE	string	Возвращает список хранилищ сертификатов

METHODS ----------

МЕТОД	ОПИСАНИЕ
ADDCERTSTORE	Добавляет новое хранилище сертификатов в список хранилищ
REMOVECERTSTORE	Удаляет существующее хранилище сертификатов из списка
CREATECACHE	Создает кэш
ADDCACHESECTION	Создает кэш для указанного хранилища
GETPRVTYPEPRESENT	Проверяет присутствие провайдера в списке

addCertStore(pvdType: string, pvdURI: string): void

Добавление хранилища в стек

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PVDTYPE	string	Тип провайдера хранилища
PVDURI	string	Путь к месторасположению хранилища

removeCertStore(pvdType: string): void

Удаление выбранного хранилища из стека

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PVDTYPE	string	Тип провайдера хранилища

createCache(cacheURI: string): void

Создание кэша стека хранилищ (JSON)

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CACHEURI	string	Путь к файлу кэша

addCacheSection(cacheURI: string, pvdType: string): void

Добавление в кэш стека хранилищ (JSON) раздела для хранения описания по провайдеру указанного типа

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CACHEURI	string	Путь к файлу кэша
PVDTYPE	string	Тип провайдера хранилища

getPrvTypePresent(pvdType: string): boolean

Возврат true если в стеке содержится хранилище указанного типа, и false в противном случае.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PVDTYPE	string	Тип провайдера хранилища

Класс Chain

Класс для построения цепочки сертификатов

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию

METHODS ----------

МЕТОД	ОПИСАНИЕ
BUILDCHAIN	Выполняет построение цепочки сертификатов
VERIFYCHAIN	Возвращает статус проверки сертификата относительно цепочки

buildChain(cert: Certificate, certs: CertificateCollection): CertificateCollection

Возвращает коллекцию сертификатов CertificateCollection, участвующих при построении цепочки. В качестве параметра функции указывается сертификат cert, для которого производится построение цепочки, провайдер хранилища сертификатов (по умолчанию «pvdSystem» – системный провайдер). В перегруженной функции указывается сертификат cert, для которого производится построение цепочки, коллекция сертификатов certs относительно которой происходит построение цепочки.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат, для которого производится построение цепочки
CERTS	CertificateCollection	Коллекция сертификатов цепочки

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var  chain = new trusted.pki.Chain();
var certs = new trusted.pki.CertificateCollection();
var  cert1 = trusted.pki.Certificate.load("test.crt", trusted.DataFormat.DER);
    certs.push(cert1);
var  cert2 = trusted.pki.Certificate.load("cert.crt", trusted.DataFormat.PEM);
    certs.push(cert2);
cert3 = trusted.pki.Certificate.load("cert1.crt", trusted.DataFormat.PEM);
    certs.push(cert3);
var outchain = chain.buildChain(cert2, certs);
console.log(outchain.length);  //1
```

---

verifyChain(chain: CertificateCollection, crls: CrlCollection): boolean

Выполняет проверку относительно цепочки сертификатов. Результат возвращается в виде числового значения – 0 – цепочка не проверена, 1 – цепочка проверена. В случае возврата значения 0 в стеке ошибок error_stack содержится информация о причинах невозможности построения/проверки цепочки сертификатов.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CHAIN	CertificateCollection	Коллекция сертификатов цепочки
CRLS	CrlCollection	Коллекция списков отзыва сертификатов

Класс Cipher

Представление шифрования данных

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(CIPHERNAME: STRING)	Конструктор параметров в виде наименования алгоритма

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
CRYPTOMETHOD	CryptoMethod	Задает криптографический метод
RECIPIENTSCERTS	CertificateCollection	Задает коллекцию сертификатов получателей
PRIVKEY	Key	Задает закрытый ключ
RECIPIENTCERT	Certificate	Задает сертификат получателя
PASSWORD	string	Задает пароль для шифрования
DIGEST	string	Задает хеш алгоритм
RIV	Buffer	Задает вектор инициализации
IV	string	Возвращает вектор инициализации
RKEY	Buffer	Задает ключ, использующийся для шифрования или расшифрования
KEY	string	Возвращает ключ, использующийся для шифрования или расшифрования
RSALT	Buffer	Задает псевдослучайную вставку
SALT	string	Возвращает псевдослучайную вставку
ALGORITHM	String	Возвращает алгоритм
MODE	String	Возвращает режим шифрования
DGST	String	Возвращает хеш алгоритм

METHODS ----------

МЕТОД	ОПИСАНИЕ
ENCRYPT	Реализация шифрования данных
DECRYPT	Реализация расшифрования данных
GETRRECIPIENTINFOS	Возвращает информацию о получателях

encrypt(filenameSource: string, filenameEnc: string, format?: DataFormat): void

Реализация шифрования данных.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAMESOURCE	string	Исходное сообщение
FILENAMEENC	String	Зашифрованное сообщение
FORMAT	DataFormat	Формат представления. По умолчанию - DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
cipher = new trusted.pki.Cipher("aes256");
cipher.cryptoMethod = trusted.CryptoMethod.SYMMETRIC;
cipher.digest = "MD5";
cipher.password = "4321";
cipher.encrypt("test.txt", "encSym.enc");
```

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var cipher = new trusted.pki.Cipher("aes256");
var cert = new trusted.pki.CertificateCollection();
cert.push(trusted.pki.Certificate.load("cert.crt", trusted.DataFormat.PEM));
cipher.recipientsCerts = cert;
cipher.encrypt("test.txt", "encAssym.enc", trusted.DataFormat.PEM);
```

---

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

Реализация расшифрования данных.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAMEENC	string	Шифрованное сообщение
FILENAMEDEC	string	Расшифрованное сообщение
FORMAT	DataFormat	Формат представления. По умолчанию - DER

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var cipher = new trusted.pki.Cipher('aes256');
cipher.cryptoMethod = trusted.CryptoMethod.SYMMETRIC;
cipher.digest = "MD5";
cipher.password = "4321";
cipher.decrypt("encSym.enc", "decSym.txt", trusted.DataFormat.PEM);
```

Класс Crl

Представление X509_CRL списков отзыва сертификата

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CRL)	Конструктор копированием экземпляра объекта CRL

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
ENCODED	Buffer	Возвращает ASN.1 объект CRL в DER кодировке
SIGNATURE	Buffer	Возвращает значение подписи
VERSION	number	Возвращает версию
ISSUERNAME	string	Возвращает имя издателя
ISSUERFRIENDLYNAME	string	Возвращает пользовательское имя издателя сертификата
LASTUPDATE	Date	Возвращает дату последнего обновления
NEXTUPDATE	Date	Возвращает дату следующего обновления
THUMBPRINT	string	Возвращает отпечаток (SHA1)
SIGALGNAME	string	Возвращает имя алгоритма подписи CRL
SIGALGSHORTNAME	string	Возвращает короткое имя алгоритма подписи CRL
SIGALGOID	string	Возвращает OID алгоритма подписи CRL

METHODS ----------

МЕТОД	ОПИСАНИЕ
GETREVOKEDCERTIFICATECERT	возвращает список отозванных сертификатов с применением фильтра (отбор осуществляется сравнением с указанным сертификатов в качестве параметра)
GETREVOKEDCERTIFICATESERIAL	возвращает список отозванных сертификатов с применением фильтра (отбор осуществляется сравнением указанного серийного номера сертификата).
LOAD	чтение структуры из файла
IMPORT	чтение структуры из памяти
EXPORT	сохранение структуры в память
SAVE	сохранение структуры в файл
COMPARE	равнение crl
EQUALS	сравнение
HASH	возвращает хэш структуры по заданному алгоритму
DUPLICATE	создает копию элемента

getRevokedCertificateCert(cer: Certificate): native.PKI.RevokedCertificate

Функция возвращает список отозванных сертификатов с применением фильтра (отбор осуществляется сравнением с указанным сертификатов в качестве параметра).

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CER	Certificate	Сертификат для организации выборки

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crl = new trusted.pki.Crl();
var crl1 = trusted.pki.Crl.load("certcrl.crl");
var crl2 = crl1.getRevokedCertificateCert(trusted.pki.Certificate.load("test.crt"));
```

---

getRevokedCertificateSerial(serial: string): native.PKI.RevokedCertificate

Функция возвращает список отозванных сертификатов с применением фильтра (отбор осуществляется сравнением указанного серийного номера сертификата).

ПАРАМЕТР	ТИП	ОПИСАНИЕ
SERIAL	string	Серийный номер сертификата для организации выборки

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

чтение структуры из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
crl = new trusted.pki.Crl();
crl.load("certcrl.crl");
console.log(crl.sigAlgName);
```

---

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

чтение структуры из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

import(buffer: Buffer, format?: DataFormat): void

чтение структуры из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
BUFFER	Buffer	Буфер с данными в памяти
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

static import(buffer: Buffer, format?: DataFormat): Crl

чтение структуры из памяти

ПАРАМЕТР	ТИП	ОПИСАНИЕ
BUFFER	Buffer	Буфер с данными в памяти
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

export(format?: DataFormat): Buffer

сохранение структуры в память

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

save(filename: string, dataFormat?: DataFormat): void

сохранение структуры в файл

ПАРАМЕТР	ТИП	ОПИСАНИЕ

FILENAME	String	Путь к файлу
DATAFORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crl = new trusted.pki.Crl();
var crl1 = trusted.pki.Crl.load("certcrl.crl");
crl1.save ('certcrl2.crl');
```

---

compare(crl: Crl): number

сравнение crl

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CRL	Crl	Экземпляр объекта Crl

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crl = new trusted.pki.Crl();
var crl1 = trusted.pki.Crl.load("certcrl.crl");
var crl2 = trusted.pki.Crl.load("certcrl1.crl");
console.log(crl1.compare(crl2)); // 0
```

---

equals(crl: Crl): boolean

сравнение

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CRL	Crl	Экземпляр объекта Crl

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crl = new trusted.pki.Crl();
var crl1 = trusted.pki.Crl.load("certcrl.crl");
var crl2 = trusted.pki.Crl.load("certcrl1.crl");
console.log(crl1.equals(crl2)); // 0
```

---

hash(algorithm?: string): String

возвращает хэш структуры по заданному алгоритму

ПАРАМЕТР	ТИП	ОПИСАНИЕ
ALGORITHM	string	Наименование алгоритма

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var Crl = new trusted.pki.Crl
var crl1 = trusted.pki.Crl.load("certcrl.crl");
var hash = crl1.hash("sha1");
console.log(hash);
```

---

duplicate(): Crl

создает копию элемента

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var Crl = new trusted.pki.Crl();
var crl1 = trusted.pki.Crl.load("certcrl.crl");
var crl2 = crl1.duplicate();
console.log(crl1.equals(crl2));
```

Класс CrlCollection

Представление коллекции списков отзыва сертификатов

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.CRLCOLLECTION)	Конструктор копированием экземпляра объекта CrlCollection

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
LENGTH	number	Размер коллекции (число сертификатов)

METHODS ----------

МЕТОД	ОПИСАНИЕ
ITEMS	Возвращает элемент коллекции списков отзыва
PUSH	Добавляет элементы в коллекцию
POP	Удаляет последний элемент из коллекции
REMOVEAT	Удаляет элемент из коллекции

items(index: number): Crl

Возвращает элемент коллекции списков отзыва.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crls = new trusted.pki.CrlCollection();
crls.push(trusted.pki.Crl.load("certcrl.crl"));
var crl = crls.items(0);
console.log(crl.sigAlgName);
```

---

push(crl: Crl): void

Добавляет элементы в коллекцию.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CRL	Crl	Экземпляр объекта списка отзыва

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crls = new trusted.pki.CrlCollection();
crls.push(trusted.pki.Crl.load("certcrl.crl"));
console.log(crls.length); //1
```

---

pop(): void

Удаляет последний элемент из коллекции

removeAt(index: number): void

Удаляет элемент из коллекции.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
INDEX	number	Индекс элемента в коллекции

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var crls = new trusted.pki.CrlCollection();
crls.push(trusted.pki.Crl.load("certcrl.crl"));
crls.removeAt(0);
console.log(crls.length); //0
```

Класс Csr

Класс для создания запроса на сертификат.

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(NAME: STRING, KEY: KEY, DIGEST: STRING)	

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
ENCODED	Buffer	Получение сформированного запроса на сертификат в Hex представлении

METHODS ----------

МЕТОД	ОПИСАНИЕ
SAVE	Сохранение запроса на сертификат

save(filename: string, dataFormat?: DataFormat): void

Сохранение запроса на сертификат

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Полный путь к файлу
DATAFORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require("trusted-crypto");
var key = trusted.pki.Key.readPrivateKey("cert.key", trusted.DataFormat.PEM, "");
var csr = new trusted.pki.CSR("/C=US/O=Test/CN=example.com", key, "SHA1");
csr.save("test.csr");
```

Класс Key

Представление ключевой пары. В составе класса объявлена структура

KEYPAIR{KeyAlgorithm algorithm; EVP_PKEY pkey;}

enum KeySize{1024,2048,4096}

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.KEY)	Конструктор копированием экземпляра объекта Key

METHODS ----------

МЕТОД	ОПИСАНИЕ
GENERATE	Генерация ключевой пары
READPRIVATEKEY	Чтение приватного ключа из файла
WRITEPRIVATEKEY	Запись приватного ключа в файл
READPUBLICKEY	Чтение открытого ключа из файла
WRITEPUBLICKEY	Запись открытого ключа в файл
COMPARE	Сравнение ключей

generate(format: DataFormat, pubExp: PublicExponent, keySize: number, password: string): Key

Генерация ключевой пары

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER
PUBEXP	PublicExponent	
KEYSIZE	number	Размер ключа
PASSWORD	string	Пароль для доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var assert = require('assert');
var  key = new trusted.pki.Key();
var keyPair = key.generate(trusted.DataFormat.PEM, trusted.PublicExponent.RSA_F4, 1024);
assert.equal(keyPair === null, true, 'true');  // Ошибка assert true
```

---

readPrivateKey(filename: string, format: DataFormat, password: string): Key

Чтение приватного ключа из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER
PASSWORD	string	Пароль для доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var  key = new trusted.pki.Key();
```

```
var privateKey = key.readPrivateKey("privkey_s.key", trusted.DataFormat.PEM, "1234");
assert.equal(privateKey == null, true, 'true');//true
```

---

writePrivateKey(filename: string, format: DataFormat, password: string): any

Запись приватного ключа в файл

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER
PASSWORD	string	Пароль для доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var  key = new trusted.pki.Key();
var keyPair = key.generate(trusted.DataFormat.PEM, trusted.PublicExponent.RSA_F4, 1024);
keyPair.writePrivateKey("privkey_s.key", trusted.DataFormat.PEM, "1234");
```

---

readPublicKey(filename: string, format: DataFormat): Key

Чтение открытого ключа из файла

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var assert = require('assert');
var  key = new trusted.pki.Key();
publickey = key.readPublicKey("pubkey_s.key", trusted.DataFormat.PEM);
assert.equal(publickey == null, true, 'true'); // true
```

---

writePublicKey(filename: string, format: DataFormat): any

Запись открытого ключа в файл

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	путь к файлу
FORMAT	DataFormat	Формат данных. Опционально. По умолчанию DER

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var  key = new trusted.pki.Key();
var keyPair = key.generate(trusted.DataFormat.PEM, trusted.PublicExponent.RSA_F4, 1024);
keyPair.writePublicKey("pubkey_s.key", trusted.DataFormat.PEM);
```

---

compare(key: Key): number

Сравнение ключей

ПАРАМЕТР	ТИП	ОПИСАНИЕ
KEY	Key	Экземпляр объекта Key

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var  key = new trusted.pki.Key();
var  privateKey  =  key.readPrivateKey("privkey_s.key",  trusted.DataFormat.PEM,
"1234");
var  privateKey1  =  key.readPrivateKey("privkey_s.key",  trusted.DataFormat.PEM,
"1234");
console.log(privateKey.compare(privateKey1)); //1
```

Класс Oid

Представление объектного идентификатора (OID'а)

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(OID: STRING)	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.OID)	Конструктор копированием экземпляра объекта Key

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
VALUE	string	Возвращает текстовое представление значения OID
LONGNAME	string	Возвращает длинное имя OID
SHORTNAME	string	Возвращает короткое имя OID

---------- ПРИМЕР ----------

```
var oid = new trusted.Pki.Oid("2.5.4.3");
console.log(oid.longName);                    // commonName
console.log(oid.shortName);          // CN
console.log(oid.value);              // 2.5.4.3
```

Класс Pkcs12

Представление интерфейса PKCS12

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKI.PKCS12)	Конструктор копированием экземпляра объекта Pkcs12

METHODS ----------

МЕТОД	ОПИСАНИЕ
CERTIFICATE	Возвращает сертификат
KEY	Возвращает приватный ключ
CA	Возвращает цепочку сертификатов
LOAD	*Чтение pkcs12 из файла*
SAVE	*Сохранение pkcs12 в файл*
CREATE	*Создание структуры PKCS12*

certificate(password: string): Certificate

Возвращает сертификат.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PASSWORD	string	Пароль доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var p12 = new trusted.pki.Pkcs12();
  p12.load("test.pfx");
var cert = p12.certificate("password");
```

---

key(password: string): Key

Возвращает приватный ключ.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PASSWORD	string	Пароль доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var p12 = new trusted.pki.Pkcs12();
  p12.load("test.pfx");
var key = p12.key("password");
```

---

ca(password: string): CertificateCollection

Возвращает цепочку сертификатов.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PASSWORD	string	Пароль доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var p12 = new trusted.pki.Pkcs12();
  p12.load("test.pfx");
var ca = p12.ca("password");
console.log (ca.length);
```

---

load(filename: string): void

Чтение pkcs12 из файла.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var p12 = new trusted.pki.Pkcs12();
p12.load("test.pfx");
```

---

static load(filename: string): Pkcs12

Чтение сертификата из файла.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу

save(filename: string): void

Сохранение pkcs12 в файл.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
FILENAME	string	Путь к файлу

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var p12 = new trusted.pki.Pkcs12();
var   cert = trusted.pki.Certificate.load("./test/cert.crt", trusted.DataFormat.PEM);
var   key = trusted.pki.Key.readPrivateKey("./test/cert.key", trusted.DataFormat.PEM, "");
var   p12Res = p12.create(cert, key, null, "1", "test_name");
p12Res.save('test_pkcs12.pfx');
```

---

create(cert: Certificate, key: Key, ca: CertificateCollection, password: string, name: string): Pkcs12

Создание структуры PKCS12.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат
KEY	Key	Приватный ключ
CA	CertificateCollection	Цепочка сертификатов
PASSWORD	String	Пароль доступа к шифрованному контенту
NAME	string	Пользовательское имя

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var p12 = new trusted.pki.Pkcs12();
var   cert = trusted.pki.Certificate.load("./test/cert.crt", trusted.DataFormat.PEM);
var   key = trusted.pki.Key.readPrivateKey("./test/cert.key", trusted.DataFormat.PEM, "");
var   p12Res = p12.create(cert, key, null, "1", "test_name");
```

Класс Revocation

Представление Revocation-провайдера

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию

METHODS ----------

МЕТОД	ОПИСАНИЕ
GETCRLLOCAL	Ищет crl для сертификата в локальном хранилище
GETCRLDISTPOINTS	Возвращает массив из точек распространения crl
CHECKCRLTIME	Проверяет действительность времени crl
DOWNLOADCRL	Загружает crl

getCrlLocal(cert: Certificate, store: PkiStore): any

Ищет crl для сертификата в локальном хранилище

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	
STORE	PkiStore	

getCrlDistPoints(cert: Certificate): Array<string>

Возвращает массив из точек распространения crl

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var revocation = new trusted.pki.Revocation();
var cert = trusted.pki.Certificate.load("test.crt");
var array = revocation.getCrlDistPoints(cert);
console.log(array);
```

---

checkCrlTime(crl: Crl): boolean

Проверяет действительность времени crl

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CRL	Crl	CRL

downloadCRL(distPoints: Array<string>, pathForSave: string, done: Function): void

Загружает crl

ПАРАМЕТР	ТИП	ОПИСАНИЕ
DISTPOINTS	Array<string>	Точки распространения crl

PATHFORSAVE	String	Путь к файлу
DONE	Function	Функция обратного вызова

PkiStore

Класс CashJson

Представление кэша хранилищ объектов PKI

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(FILENAME: STRING)	Конструктор с параметром пути к файлу JSON

METHODS ----------

МЕТОД	ОПИСАНИЕ
EXPORT	Выполняет экспорт из файла кэша
IMPORT	Выполняет импорт в файл кэша

export(): native.PKISTORE.IPkiItem[]

Выполняет экспорт из файла кэша.

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
cashjson = new trusted.pkistore.CashJson('CertStore/cash.json');
var items = cashjson.export();
console.log(items.length);
```

---

import(items: native.PKISTORE.IPkiItem[]): void

Выполняет импорт в файл кэша.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
ITEMS	native.PKISTORE.IPkiItem[]	Массив описателей объектов PKI

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
cashjson = new trusted.pkistore.CashJson('CertStore/cash.json');
var items = cashjson.export();
cashjson.import(items);
```

Класс Filter

Представление фильтра для выборки объектов PKI

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
TYPES	string	Возвращает тип объекта
PROVIDERS	string	Возвращает провайдер хранилища в котором размещен объект
CATEGORYS	string	Возвращает категорию хранилища (MY, OTHER, TRUST)
HASH	String	Возвращает хэш объекта
SUBJECTNAME	string	Возвращает полное имя владельца сертификата
SUBJECTFRIENDLYNAME	string	Возвращает пользовательское имя владельца сертификата
ISSUERNAME	string	Возвращает полное имя издателя сертификата
ISSUERFRIENDLYNAME	String	Возвращает пользовательское имя издателя сертификата
SERIAL	string	Возвращает серийный номер сертификата

Класс PKIItem

Представление описателей PKI объектов

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
FORMAT	string	Задает формат представления объекта
TYPE	string	Задает тип объекта
PROVIDER	string	Задает провайдер хранилища в котором размещен объект
CATEGORY	string	Задает категорию хранилища (MY, OTHER, TRUST)
URI	string	Задает URI к физическому месторасположению объекта
HASH	string	Задает хэш объекта
SUBJECTNAME	string	Задает полное имя владельца сертификата
SUBJECTFRIENDLYNAME	string	Задает пользовательское имя владельца сертификата
ISSUERNAME	string	Задает полное имя издателя сертификата
ISSUERFRIENDLYNAME	string	Задает пользовательское имя издателя сертификата
SERIAL	string	Задает серийный номер сертификата
NOTBEFORE	String	Задает дату начала действия сертификата
NOTAFTER	string	Задает дату истечения срока действия сертификата
LASTUPDATE	string	Задает дату последнего обновления списка отзыва
NEXTUPDATE	string	Задает дату следующего выпуска списка отзыва
KEY	string	Задает хеш публичного ключа
ENCRYPTED	boolean	Задает зашифрован ли закрытый ключ
ORGANIZATIONNAME	string	Задает название организации
SIGNATUREALGORITHM	string	Задает алгоритм подписи

Класс PkiStore

Класс для организации стека хранилищ сертификатов и ключей. Создается кэш (JSON) для формирования описателей объектов хранения.

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(FOLDER: STRING)	Конструктор по умолчанию
CONSTRUCTOR(HANDLE: NATIVE.PKISTORE.PKISTORE)	Конструктор копированием экземпляра объекта PkiStore

PROPERTIES ----------

НАЗВАНИЕ	ТИП	ОПИСАНИЕ
CASH	CashJson	Возвращает структуру кэша (JSON)

METHODS ----------

МЕТОД	ОПИСАНИЕ
ADDPROVIDER	Добавляет новый провайдер хранилища
ADDCERT	Добавляет описатель сертификата
ADDCRL	Добавляет описатель списка отзыва
ADDKEY	Добавляет описатель ключа
ADDCSR	Добавляет описатель запроса на сертификат
FIND	Поиск объекта в кэше
FINDKEY	Поиск ключа в кэше
GETITEM	Извлечение объекта по его описателю

addProvider(provider: native.PKISTORE.Provider): void

Добавляет новый провайдер хранилища.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PROVIDER	native.PKISTORE.Provider	Провайдер хранилища

---------- ПРИМЕР ----------

```
var providerSystem = new trusted.pkistore.Provider_System('/CertStore');
store.addProvider(providerSystem.handle);
```

---

addCert(provider: native.PKISTORE.Provider, category: string, cert: Certificate): string

Добавляет описатель сертификата.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PROVIDER	native.PKISTORE.Provider	Провайдер хранилища
CATEGORY	string	Категория хранилища (MY, OTHER, TRUST)
CERT	Certificate	Сертификат для которого создается описатель

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('/CertStore');
var store = new trusted.pkistore.PkiStore("/CertStore/cash.json");
var cert = trusted.pki.Certificate.load("cert1.crt", trusted.DataFormat.PEM);
store.addCert(providerSystem.handle, "MY", cert);
```

```
providerSystem = new trusted.pkistore.Provider_System('/CertStore');
store.addProvider(providerSystem.handle);
var items = store.find();
store.cash.import(items);
```

---

addCrl(provider: native.PKISTORE.Provider, category: string, crl: Crl): string

Добавляет описатель списка отзыва.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PROVIDER	native.PKISTORE.Provider	Провайдер хранилища
CATEGORY	string	Категория хранилища (MY, OTHER, TRUST)
CRL	Crl	Список отзыва для которого создается описатель

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('/CertStore');
var store = new trusted.pkistore.PkiStore("CertStore/cash.json");
crl = trusted.pki.Crl.load("certcrl.crl");
store.addCrl(providerSystem.handle, "CRL", crl);
providerSystem = new trusted.pkistore.Provider_System('/CertStore');
store.addProvider(providerSystem.handle);
var items = store.find();
store.cash.import(items);
```

---

addKey(provider: native.PKISTORE.Provider, key: Key, password: string): string

Добавляет описатель ключа.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PROVIDER	native.PKISTORE.Provider	Провайдер хранилища
KEY	Key	Ключ для которого создается описатель
PASSWORD	string	Пароль для доступа к шифрованному контенту

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('CertStore');
var store = new trusted.pkistore.PkiStore("CertStore/cash.json");
var  key = trusted.pki.Key.readPrivateKey("cert.key", trusted.DataFormat.PEM, "");
store.addKey(providerSystem.handle, key, "password");
providerSystem = new trusted.pkistore.Provider_System('/CertStore');
store.addProvider(providerSystem.handle);
var items = store.find();
store.cash.import(items);
```

---

addCsr(provider: native.PKISTORE.Provider, category: string, csr: CertificationRequest): string

Добавляет описатель запроса на сертификат.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PROVIDER	native.PKISTORE.Provider	Провайдер хранилища
CATEGORY	string	Категория хранилища (MY, OTHER, TRUST)
CSR	CertificationRequest	Запрос на сертификат для которого создается описатель

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('/CertStore');
var store = new trusted.pkistore.PkiStore("/CertStore/cash.json");
var  csr = trusted.pki.CertificationRequest.load("test.csr", trusted.DataFormat.PEM, "");
store.addCsr(providerSystem.handle, "MY", csr);
providerSystem = new trusted.pkistore.Provider_System('/CertStore');
store.addProvider(providerSystem.handle);
var items = store.find();
store.cash.import(items);
```

---

find(ifilter?: native.PKISTORE.IFilter): native.PKISTORE.IPkiItem[]

Поиск объекта в кэше.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
IFILTER	native.PKISTORE.IFilter	Фильтра для выборки объектов PKI

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('CertStore');
var store = new trusted.pkistore.PkiStore("CertStore/cash.json");
store.addProvider(providerSystem.handle);
var items = store.find({type: ["CERTIFICATE"], category: ["MY"]});
console.log(items.length);
```

---

findKey(ifilter: native.PKISTORE.IFilter): native.PKISTORE.IPkiItem

Поиск ключа в кэше.

ПАРАМЕТР	ТИП	ОПИСАНИЕ
IFILTER	native.PKISTORE.IFilter	Фильтра для выборки объектов PKI

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('CertStore');
var store = new trusted.pkistore.PkiStore("CertStore/cash.json");
store.addProvider(providerSystem.handle);
var key = store.findKey({
        type: ["CERTIFICATE"],
        provider: ["SYSTEM"],
        category: ["MY"],
         hash: ['67cd1d796cfb42d00166737c6e16d596cf83695e']
        });
console.log(key.uri);
```

---

getItem(item: native.PKISTORE.IPkiItem): any

Извлечение объекта по его описателю.

ПАРАМЕТР	ТИП	ОПИСАНИЕ

ITEM	native.PKISTORE.IPkiItem	Описатель объекта

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System('CertStore');
var store = new trusted.pkistore.PkiStore("CertStore/cash.json");
store.addProvider(providerSystem.handle);
var certs = store.find({
        type: ["CERTIFICATE"],
        category: ["MY"]
    });
var item = certs[0];
var object = store.getItem(item);
console.log(object.type);
```

Класс Provider_System

Представление системного провайдера

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR(FOLDER: STRING)	Конструктор по умолчанию

METHODS ----------

МЕТОД	ОПИСАНИЕ
OBJECTTOPKIITEM	Возвращает PKI объект

objectToPkiItem(path: string): native.PKISTORE.IPkiItem

ПАРАМЕТР	ТИП	ОПИСАНИЕ
PATH	string	Задает URI к физическому месторасположению объекта

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var providerSystem = new trusted.pkistore.Provider_System(‘/CertStore’);
var store = new trusted.pkistore.PkiStore( "/CertStore/cash.json");
var uri = store.addCert(providerSystem.handle, "MY", cert);
var item = providerSystem.objectToPkiItem(uri);
```

Класс ProviderCryptopro

Поддержка провайдера КриптоПро CSP

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию

METHODS ----------

МЕТОД	ОПИСАНИЕ
GETKEY	Возвращает закрытый ключ сертификата из хранилища КриптоПро

getKey(cert: pki.Certificate)

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат, к которому привязывается закрытый ключ

---------- ПРИМЕР ----------

```
var trusted = require('trusted-crypto');
var cert = trusted.pki.Certificate.load('cert.crt', trusted.DataFormat.PEM);
var providerCryptopro = new trusted.pkistore.ProviderCryptopro();
var key = providerCryptopro.getKey(cert);
```

Класс ProviderMicrosoft

Поддержка провайдера Microsoft (только для Windows)

CONSTRUCTORS ----------

КОНСТРУКТОР	ОПИСАНИЕ
CONSTRUCTOR()	Конструктор по умолчанию

METHODS ----------

МЕТОД	ОПИСАНИЕ
GETKEY	Возвращает закрытый ключ сертификата из хранилища КриптоПро

getKey(cert: pki.Certificate)

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CERT	Certificate	Сертификат, к которому привязывается закрытый ключ

---------- ПРИМЕР ----------

```
var cert = trusted.pki.Certificate.load('cert.cer', trusted.DataFormat.PEM);
var providerMicrosoft = new trusted.pkistore.ProviderMicrosoft();
var key = providerMicrosoft.getKey(cert);
```
