Клиентская JAVA-библиотека для Криптографической подсистемы аппаратно-программного комплекса ФПСУ-IP "Сервер ОТП"

11485466.72.21.12.154

Инструкция по установке и эксплуатации

Содержание

1 Введение ... 3

2 Назначение и условия применения ... 3

2.1 Назначение системы ... 3
2.2 Условия применения системы ... 3

3 Установка программного изделия «JAVA-библиотека для Сервера ОТП» ... 4

4 Удаление программного изделия «JAVA-библиотека для Сервера ОТП» ... 4

5 Описание библиотеки ... 4

5.1 Пакеты библиотеки «JAVA-библиотека для Сервера ОТП» ... 4
5.2 Классы пакета ru.infocrypt.otp ... 4
5.2.1 Класс OTP ... 4
5.2.2 Класс OtpAnswer ... 9
5.2.3 Класс OtpKeyInfo ... 9
5.2.4 Класс TLSContext ... 12
5.3 Классы пакета ru.infocrypt.otp.exception ... 13
5.3.1 Класс FpsuException ... 13

1 Введение

Настоящий документ содержит руководство по установке и эксплуатации программного изделия «Клиентская JAVA-библиотека для Криптографической подсистемы аппаратно-программного комплекса ФПСУ-IP "Сервер ОТП"» (далее «JAVA-библиотека для Сервера ОТП). Руководство включает в себя справочную информацию по работе с библиотекой «JAVA-библиотека для Сервера ОТП».

2 Назначение и условия применения

2.1 Назначение системы

«JAVA-библиотека для Сервера ОТП» представляет собой библиотеку JAVA, которая предназначена для предоставления удобного мультиплатформенного программного интерфейса к программному изделию «Сервер ОТП» в составе ПАК ФПСУ-IP.

В программном изделии «JAVA-библиотека для Сервера ОТП» реализовано выполнение с помощью программного изделия «Сервер ОТП» следующих основных функций:

- Создание мастер-ключа.
- Создание облачного одноразового пароля и его шифрование с помощью мастер-ключа.
- Проверка облачного одноразового пароля.

2.2 Условия применения системы

«Клиентская JAVA-библиотека для Сервера ОТП» должна работать под управлением ОС, поддерживающих среду JVM версий 1.6, 1.7 и 1.8.

Для работы программного изделия «JAVA-библиотека для Сервера ОТП» необходим сетевой доступ к ПАК ФПСУ-IP, на котором установлено программное изделие «Сервер ОТП».

3 Установка программного изделия «JAVA-библиотека для Сервера ОТП»

Для того чтобы установить программное изделие «JAVA-библиотека для Сервера ОТП», следует скопировать содержимое дистрибутива «JAVA-библиотека для Сервера ОТП» на жёсткий диск компьютера.

4 Удаление программного изделия «JAVA-библиотека для Сервера ОТП»

Для того чтобы удалить программное изделие «JAVA-библиотека для Сервера ОТП», необходимо удалить с жёсткого диска компьютера ранее установленные файлы дистрибутива «JAVA-библиотека для Сервера ОТП».

5 Описание библиотеки

5.1 Пакеты библиотеки «JAVA-библиотека для Сервера ОТП»

В состав библиотеки «JAVA-библиотека для Сервера ОТП» входят два пакета:

- ru.infocrypt.otp,
- ru.infocrypt.otp.exception.

5.2 Классы пакета ru.infocrypt.otp

В состав пакета ru.infocrypt.otp входят классы:

- OTP,
- OtpAnswer,
- OtpKeyInfo,
- TLSContext.

5.2.1 Класс OTP

```
java.lang.Object
        ru.infocrypt.otp.OTP
-----------------------------------------------------------------------
    public class OTP
```

```
extends java.lang.Object
```

Описание

Облачные одноразовые пароли.

Конструкторы

OTP (java.lang.String addr, int port, int timeout) – инициализация соединения с сервером облачных одноразовых паролей.

```
public OTP(java.lang.String addr,
           int port,
           int timeout)
    throws FpsuException
```

Параметры:

```
addr - адрес сервера

port - порт

timeout - таймаут сессии с сервером
```

Исключения:

```
FpsuException - возможные исключения
```

OTP (java.lang.String addr, int port, int timeout, ru.infocrypt.otp.enums.ProtocolType protocolType, ru.infocrypt.otp.TLSContext tlsContext) – инициализация соединения с сервером облачных одноразовых паролей с указанием протокола связи.

```
public OTP(java.lang.String addr,
                int port,
                int timeout,
                ru.infocrypt.otp.enums.ProtocolType protocolType,
                TLSContext tlsContext)
          Throws FpsuException
```

Параметры:

```
addr - адрес сервера

port - порт

timeout - таймаут сессии с сервером

protocolType - протокол связи ProtocolType

tlsContext - контекст TLS (в случае, если значение protocolType -
TLS или TLS_GOST)
```

Исключения:

```
FpsuException - возможные исключения
```

Методы

Модификатор и тип	Метод и описание
int	activateMaster(int number) Активация существующего мастер-ключа
byte[]	calcOtp(OtpKeyInfo keyInfo, byte[] data) Вычисление ОТП
int	checkOtp(byte[] data, byte[] rsaCert, OtpKeyInfo keyInfo, byte[] otpCode) Проверка ОТП
OtpAnswer	generateOtpKey(ru.infocrypt.otp.enums.DigestParamSet paramSet, byte[] rsaCert) Создать ключевую пару ОТП
byte[]	getOtpKey(OtpKeyInfo keyInfo, byte[] rsaCert) Возвращение зашифрованного ОТП
int	makeNewMaster(int number, byte[] salt) Создание нового мастер ключа

Метод activateMaster

```
public int activateMaster(int number)
                  throws FpsuException
```

Назначение:
Активация существующего мастер-ключа.

Параметры:
`number - номер существующего мастер-ключа [0..255]`

Возвращаемое значение:
`результат работы (0 - без ошибок, иначе код ошибки)`

Исключения:
`FpsuException - возможные исключения`

Метод calcOtp

```
public byte[] calcOtp(OtpKeyInfo keyInfo,
                      byte[] data)
```

```
                throws FpsuException
```

Назначение:
Вычисление одноразового облачного пароля.

Параметры:
keyInfo - ключевая структура [OtpKeyInfo]

data - данные для которых необходимо вычислить ОТП

Возвращаемое значение:
Одноразовый облачный пароль в виде массива байтов

Исключения:
FpsuException - возможные исключения

Метод checkOtp

```
public int checkOtp(byte[] data,
                    byte[] rsaCert,
                    OtpKeyInfo keyInfo,
                    byte[] otpCode)
             throws FpsuException
```

Назначение:
Проверка одноразового облачного пароля.

Параметры:
data - данные на проверку ОТП

rsaCert - сертификат, на котором осуществлять проверку ОТП

keyInfo - ключевая структура [OtpKeyInfo]

otpCode - Одноразвый облачный пароль

Возвращаемое значение:
Одноразовый облачный пароль в виде массива байтов

Исключения:
FpsuException - возможные исключения

Метод generateOtpKey

```
public OtpAnswer generateOtpKey(ru.infocrypt.otp.enums.DigestParamSet
paramSet,
                                byte[] rsaCert)
                         throws FpsuException
```

Назначение:
Создание ключевой пары одноразового облачного пароля.

Параметры:

paramSet - набор параметров хеша

rsaCert - сертификат в виде массива байтов

Возвращаемое значение:
OtpAnswer структура OTP_CLOUD_STRUCT + зашифрованный ключ

Исключения:
FpsuException - возможные исключения

Метод getOtpKey

```
public byte[] getOtpKey(OtpKeyInfo keyInfo,
                        byte[] rsaCert)
                 throws FpsuException
```

Назначение:
Получение зашифрованного одноразового облачного пароля.

Параметры:
keyInfo - ключевая структура [OtpKeyInfo]

rsaCert - сертификат в виде массива байтов

Возвращаемое значение:
зашифрованный одноразовый облачный пароль в виде массива байтов

Исключения:
FpsuException - возможные исключения

Метод makeNewMaster

```
public int makeNewMaster(int number,
                         byte[] salt)
                  throws FpsuException
```

Назначение:
Создание нового мастер-ключа.

Параметры:
number - номер мастер-ключа [0..255]

salt – затравка мастер-ключа

Возвращаемое значение:
результат работы (0 - без ошибок, иначе код ошибки)

Исключения:
FpsuException - возможные исключения

Метод reencrMaster

```
public OtpKeyInfo reencrMaster(int newNumber,
                               OtpKeyInfo keyInfo)
                        throws FpsuException
```

Назначение:
Перешифрование мастер-ключа.

Параметры:
`newNumber - номер нового мастер-ключа [0..255]`

`keyInfo - ключевая структура [OtpKeyInfo]`

Возвращаемое значение:
`новая ключевая структура [OtpKeyInfo]`

Исключения:
`FpsuException - возможные исключения`

5.2.2 Класс OtpAnswer

```
java.lang.Object

    ru.infocrypt.otp.OtpAnswer
-----------------------------------------------------------------------
    public class OtpAnswer

    extends java.lang.Object
```

Методы

Модификатор и тип	Метод и описание
byte[]	getEncryptedKey()
OtpKeyInfo	getOtpKeyInfo()

Метод getEncryptedKey

```
public byte[] getEncryptedKey()
```

Метод getOtpKeyInfo

```
public OtpKeyInfo getOtpKeyInfo()
```

5.2.3 Класс OtpKeyInfo

```
java.lang.Object

    ru.infocrypt.otp.OtpKeyInfo
-----------------------------------------------------------------------
    public class OtpKeyInfo

    extends java.lang.Object
```

Описание

Структурированная информация об одноразовом облачном пароле.

Конструкторы

OtpKeyInfo(byte[] serialized) – восстановление объекта OtpKeyInfo из сериализованного набора байтов.

```
public OtpKeyInfo(byte[] serialized)
           throws FpsuException
```

Параметры:

```
serialized - сериализованный объект
```

Исключения:

```
FpsuException - возможные исключения
```

Методы

Модификатор и тип	Метод и описание
int	getCertificateSubjectLength()
ru.infocrypt.otp.enums.DigestParamSet	getDigestParamSet() Набор параметров функции расчета хеша
java.util.Calendar	getKeyBeginDate() Дата начала действия одноразового облачного пароля
java.util.Calendar	getKeyExpiredDate() Дата окончания действия одноразового облачного пароля
int	getMasterKeyNumber() Номер мастер-ключа
byte[]	getOtpKeyStructData()
int	getOtpKeyStructSize()
byte[]	serialize() Сериализовать объект OtpKeyInfo

Метод getCertificateSubjectLength

```
public int getCertificateSubjectLength()
```

Метод getDigestParamSet

```
public ru.infocrypt.otp.enums.DigestParamSet getDigestParamSet()
```

Назначение:
Получение набора параметров функции расчета хеша.

Возвращаемое значение:
перечисление DigestParamSet

Метод getKeyBeginDate

```
public java.util.Calendar getKeyBeginDate()
```

Назначение:
Получение даты начала действия одноразового облачного пароля.

Возвращаемое значение:
дата в формате Calendar

Метод getKeyExpiredDate

```
public java.util.Calendar getKeyExpiredDate()
```

Назначение:
Получение даты окончания действия одноразового облачного пароля.

Возвращаемое значение:
дата в формате Calendar

Метод getMasterKeyNumber

```
public int getMasterKeyNumber()
```

Назначение:
Получение номера мастер-ключа.

Возвращаемое значение:
номер

Метод getOtpKeyStructData

```
public byte[] getOtpKeyStructData()
```

Метод getOtpKeyStructSize

```
public int getOtpKeyStructSize()
```

Метод serialize

```
public byte[] serialize()
```

Назначение:
Сериализация объекта OtpKeyInfo.

Возвращаемое значение:
сериализованный в набор байтов объект

5.2.4 Класс TLSContext

```
java.lang.Object

   ru.infocrypt.otp.TLSContext
-----------------------------------------------------------------------
public class TLSContext

extends java.lang.Object
```

Описание

Контекст TLS сессии.

Конструкторы

TLSContext(boolean isClientAuthEnabled, java.lang.String certStorePath, java.lang.String certStorePassword, java.lang.String keyPathPKCS12, java.lang.String keyPassword) – создание контекста TLS сессии.

```
public TLSContext(boolean isClientAuthEnabled,
                  java.lang.String certStorePath,
                  java.lang.String certStorePassword,
                  java.lang.String keyPathPKCS12,
                  java.lang.String keyPassword)
```

Параметры:
isClientAuthEnabled - использовать двухфакторную аутентификацию TLS

certStorePath - хранилище сертификатов

certStorePassword - пароль хранилища сертификатов

keyPathPKCS12 - ключ в контейнере PKCS#12

keyPassword - пароль ключа

Методы

Модификатор и тип	Метод и описание
java.net.Socket	createTlsSocket(java.lang.String host, int port, int timeout) Установить TLS соединение с указанным в параметрах хостом

Метод createTlsSocket

```
public java.net.Socket createTlsSocket(java.lang.String host,
                                        int port,
                                        int timeout)
                                 throws java.lang.Exception
```

Назначение:
Установление TLS соединения с указанным в параметрах хостом.

Параметры:

host - адрес хоста

port - порт

timeout - таймаут соединения

Возвращаемое значение:

объект Socket. В дальнейшем через него осуществляется ввод и вывод данных

Исключения:

java.lang.Exception - возможные исключения

5.3 Классы пакета ru.infocrypt.otp.exception

5.3.1 Класс FpsuException

```
java.lang.Object
  java.lang.Throwable
        java.lang.Exception
                ru.infocrypt.otp.exception.FpsuException
-----------------------------------------------------------------------
```

public class **FpsuException**

extends java.lang.Exception

Описание

Возможные исключения, которые приложения могут перехватить.

Конструкторы

FpsuException(int err)

```
public FpsuException(int err)
```

FpsuException(int err, java.lang.Throwable inner)

```
public FpsuException(int err,
                     java.lang.Throwable inner)
```

FpsuException(ru.infocrypt.otp.enums.RetCode rc) – создание нового исключения с указанным кодом ошибки.

```
public FpsuException(ru.infocrypt.otp.enums.RetCode rc)
```

Параметры:

rc - объект FpsuRetCode

FpsuException(ru.infocrypt.otp.enums.RetCode rc, java.lang.Throwable inner) – создание нового исключения с указанными кодом ошибки и причиной её возникновения.

```
public FpsuException(ru.infocrypt.otp.enums.RetCode rc,
                     java.lang.Throwable inner)
```

Параметры:

rc - объект FpsuRetCode

inner - причина (сохраняется для последующего извлечения методом Throwable.initCause)

Методы

Модификатор и тип	Метод и описание
ru.infocrypt.otp.enums.RetCode	getError()
int	getErrorCode()

Метод getError

```
public ru.infocrypt.otp.enums.RetCode getError()
```

Метод getErrorCode

```
public ru.infocrypt.otp.enums.RetCode getErrorCode()
```