Клиентская JAVA-библиотека для сервера ЭП

11485466.72.21.12.111

Инструкция по установке и эксплуатации

Содержание

1 Введение

Настоящий документ содержит руководство по установке и эксплуатации программного изделия «Клиентская JAVA-библиотека для сервера ЭП». Руководство включает в себя справочную информацию по работе с библиотекой «Клиентская JAVA-библиотека для сервера ЭП».

2 Назначение и условия применения

2.1 Назначение системы

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

В программном изделии «Клиентская JAVA-библиотека для сервера ЭП» реализовано выполнение с помощью программного изделия «Сервер ОЭП» следующих основных функций:

- Формирование электронной подписи в соответствии с требованиями ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 .
- Проверка электронной подписи.
- Просмотр параметров сервера ЭП.

2.2 Условия применения системы

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

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

3 Установка программного изделия «Клиентская JAVA-библиотека для сервера ЭП»

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

4 Удаление программного изделия «Клиентская JAVA-библиотека для сервера ЭП»

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

5 Описание библиотеки

5.1 Пакеты библиотеки «Клиентская JAVA-библиотека для сервера ЭП»

В состав библиотеки «Клиентская JAVA-библиотека для сервера ЭП» входят два пакета:

- ru.infocrypt.crypto.enums
- ru.infocrypt.fpsu

5.2 Классы пакета ru.infocrypt.crypto.enums

В состав пакета ru.infocrypt.crypto.enums входит один класс – перечисление ProtocolType.

5.2.1 Перечисление ProtocolType

```
java.lang.Object
  java.lang.Enum<ProtocolType>
      ru.infocrypt.crypto.enums.ProtocolType
-----------------------------------------------------------------------
      public enum ProtocolType
      extends java.lang.Enum<ProtocolType>
```

Описание

Перечисление значений параметров открытого ключа по ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012.

Константы

Константа	Описание
TCP	public static final ProtocolType TCP
TLS	public static final ProtocolType TLS
TLS_GOST	public static final ProtocolType TLS_GOST
UDP	public static final ProtocolType UDP

Методы

Модификатор и тип	Метод и описание
static ProtocolType	valueOf(java.lang.String name) Возвращает константу перечисления данного типа с указанным именем
static ProtocolType[]	values() Возвращает массив констант в порядке, в котором они были указаны в перечислении данного типа

Метод valueOf

```
public static ProtocolType valueOf(java.lang.String name)
```

Описание:

Возвращает константу перечисления данного типа с указанным именем. Строка должна точно соответствовать идентификатору константы, указанному в перечислении данного типа. (Лишние пробелы недопустимы.)

Параметры:

`name – возвращаемое имя константы перечисления.`

Возвращаемое значение:

константа перечисления данного типа с указанным именем

Исключения:
`java.lang.IllegalArgumentException – если перечисление данного типа не содержит константу с указанным именем`

`java.lang.NullPointerException – если аргумент равен null`

Метод values

```
public static ProtocolType[] values()
```

Описание:
Возвращает массив констант в порядке, в котором они были указаны в перечислении данного типа. Данный метод позволяет перебрать константы следующим образом:

```
for (ProtocolType c : ProtocolType.values())
    System.out.println(c);
```

Возвращаемое значение:

Массив констант в порядке, в котором они были указаны в перечислении данного типа.

5.3 Классы пакета ru.infocrypt.fpsu

В состав пакета ru.infocrypt.fpsu входят классы:

- DSInfo,
- FpsuKey,
- GKUZInfo,
- ServerDS.

5.3.1 Класс DSInfo

```
java.lang.Object

        ru.infocrypt.fpsu.DSInfo
-------------------------------------------------------------------------

    public class DSInfo
    extends java.lang.Object
```

Описание

Информация об ЭП, возвращаемой сервером ФПСУ-IP.

Конструкторы

DSInfo (byte[] serialized) – структуризация объекта ЭП.

```
public DSInfo(byte[] serialized)
       throws ru.infocrypt.fpsu.exception.FpsuException
```

Параметры:
`serialized - сериализованная ЭП с дополнительной информацией`

Исключения:
`ru.infocrypt.fpsu.exception.FpsuException - возможные исключения`

DSInfo (byte[] signature, java.lang.String keyIdent, ru.infocrypt.crypto.enums.PublicKeyParamSet paramSet) – инициализация структуры, содержащей ЭП, с использованием входных параметров.

```
public DSInfo(byte[] signature,
              java.lang.String keyIdent,
              ru.infocrypt.crypto.enums.PublicKeyParamSet paramSet)
```

Параметры:
`signature - ЭП`

`keyIdent - идентификатор ключа`

`paramSet - набор параметров открытого ключа`

Исключения:
`ru.infocrypt.fpsu.exception.FpsuException - возможные исключения`

Методы

Модификатор и тип	Метод и описание
java.lang.String	getKeyIdentifier() Получить идентификатор ключа ЭП
ru.infocrypt.crypto.enums.PublicKeyParamSet	getParamSet() Получить набор параметров открытого ключа
byte[]	getSignature() Вернуть ЭП в чистом виде

Метод getKeyIdentifier

```
public java.lang.String getKeyIdentifier()
```

Назначение:
Получение идентификатора ключа ЭП.

Возвращаемое значение:

строка

Метод getParamSet

```
public ru.infocrypt.crypto.enums.PublicKeyParamSet getParamSet()
```

Назначение:
Получение набора параметров открытого ключа.

Возвращаемое значение:
перечисление PublicKeyParamSet

Метод getSignature

```
public byte[] getSignature()
```

Назначение:
Получение ЭП в чистом виде.

Возвращаемое значение:
набор байтов

5.3.2 Класс FpsuKey

```
java.lang.Object

         ru.infocrypt.fpsu. FpsuKey
-----------------------------------------------------------------------

  public class FpsuKey

      extends java.lang.Object
```

Описание

Интерпретация ключа ФПСУ-IP.

Конструкторы

FpsuKey(short keyNo, java.lang.String keyIdent, ru.infocrypt.crypto.enums.PublicKeyParamSet paramSet) – инициализация структуры ключа заданными параметрами.

```
public FpsuKey(short keyNo,
         java.lang.String keyIdent,
         ru.infocrypt.crypto.enums.PublicKeyParamSet paramSet)
```

Параметры:
keyNo - номер

keyIdent - идентификатор

`paramSet - набор параметров открытого ключа`

Методы

Модификатор и тип	Метод и описание
java.lang.String	getKeyIdent() Получить идентификатор ключа
short	getKeyNumber() Получить номер ключа
ru.infocrypt.crypto.enums.PublicKeyParamSet	getParamSet() Получить набор параметров открытого ключа

Метод getKeyIdent

```
public java.lang.String getKeyIdent()
```

Назначение:
Получение идентификатора ключа.

Возвращаемое значение:
строка

Метод getKeyNumber

```
public short getKeyNumber()
```

Назначение:
Получение номера ключа.

Возвращаемое значение:
число

Метод getParamSet

```
public ru.infocrypt.crypto.enums.PublicKeyParamSet getParamSet()
```

Назначение:

Получение набора параметров открытого ключа.

Возвращаемое значение:

перечисление [PublicKeyParamSet]

5.3.3 Класс GKUZInfo

```
java.lang.Object

        ru.infocrypt.fpsu.GKUZInfo
-------------------------------------------------------------------------

     public abstract class GKUZInfo

     extends java.lang.Object
```

Описание

Информация о ГК и УЗ, полученным с сервера ЭП.

Конструкторы

GKUZInfo(byte[] serialized)

```
public GKUZInfo(byte[] serialized)
        throws ru.infocrypt.fpsu.exception.FpsuException
```

Исключения:

```
     ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

Методы

Модификатор и тип	Метод и описание
byte[]	getGK() Главный ключ
byte[]	getUZ() Узел замены

Метод getGK

```
public byte[] getGK()
```

Назначение:
Получение главного ключа.

Метод getUZ

```
public byte[] getUZ()
```

Назначение:
Получение узла замены.

5.3.4 Класс ServerDS

```
java.lang.Object

        ru.infocrypt.fpsu.ServerDS
-------------------------------------------------------------------------

     public abstract class ServerDS

     extends java.lang.Object
```

Описание

Работа с сервером электронной подписи на базе ФПСУ-IP.

Конструкторы

ServerDS(java.lang.String addr, int port, int timeout) – создание соединения с сервером ЭП (ФПСУ-IP).

```
public ServerDS(java.lang.String addr,
          int port,
          int timeout)
   throws ru.infocrypt.fpsu.exception.FpsuException
```

Параметры:

```
     addr - адрес сервера ЭП

     port - порт сервера ЭП

     timeout - таймаут сервера ЭП
```

Исключения:

```
     ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

ServerDS(java.lang.String addr, int port, int timeout, ProtocolType protocolType, ru.infocrypt.fpsu.TLSContext tlsContext) – создание соединения с сервером ЭП (ФПСУ-IP).

```
public ServerDS(java.lang.String addr,
          int port,
          int timeout,
          ProtocolType protocolType,
          ru.infocrypt.fpsu.TLSContext tlsContext)
   throws ru.infocrypt.fpsu.exception.FpsuException
```

Параметры:

```
     addr - адрес сервера ЭП

     port - порт сервера ЭП

     timeout - таймаут сервера ЭП

     protocolType - протокол, используемый для связи
```

```
      tlsContext - контекст TLS сессии
```

Исключения:

```
      ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

ServerDS(java.lang.String addr, int port, int timeout, ProtocolType protocolType, ru.infocrypt.fpsu.TLSContext tlsContext, ru.infocrypt.profiler.Profiler profiler) – создание соединения с сервером ЭП (ФПСУ-IP).

```
public ServerDS(java.lang.String addr,
          int port,
          int timeout,
          ProtocolType protocolType,
          ru.infocrypt.fpsu.TLSContext tlsContext,
          ru.infocrypt.profiler.Profiler profiler)
   throws ru.infocrypt.fpsu.exception.FpsuException
```

Параметры:

```
      addr - адрес сервера ЭП

      port - порт сервера ЭП

      timeout - таймаут сервера ЭП

      protocolType - протокол, используемый для связи

      tlsContext - контекст TLS сессии

      profiler - профайлер (опционально) Profiler
```

Исключения:

```
      ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

Методы

Модификатор и тип	Метод и описание
void	check(java.lang.String user_id, int pk_no, byte[] data, byte[] sign) Проверить ЭП
ru.infocrypt.fpsu.FPSUSoftwareVersion	checkVersion() Получить версию ПО ФПСУ-IP
byte[]	createSesKey(ru.infocrypt.crypto.enums.PublicKeyParamSet paramSet, byte[] mac, byte[] encryptedKey, byte[] ephemeralPublicKey, byte[] ukm, short key_no) Создать сессионный ключ расшифрования CMS для БиКрипт
int	findPk(java.lang.String pk_base_id, java.lang.String user_id) Поиск в базе открытых ключей на сервере ключа пользователя user_id и возвращение его номера

Модификатор и тип	Метод и описание
FpsuKey	getFpsuKey(short key_no) Получить класс-описатель ключа ФПСУ
java.lang.String	getKeyIdentifier(short key_no) Получить идентификатор ключа по его номеру
byte[]	getPublicKey(FpsuKey key) Получить из ФПСУ открытый ключ RSA
int	getVersion() Версия сервера ФПСУ-IP
int	isPkBaseExists(java.lang.String pk_base_id) Существует ли БОК с указанным идентификатором
ru.infocrypt.crypto.EncryptionKey	loadEncryptionKey(int keyNumber) Загрузить ключ шифрования
ru.infocrypt.crypto.EncryptionKey	loadEncryptionKey(int keyNumber, byte[] salt) Загрузить ключ шифрования с замешиванием
GKUZInfo	loadGKUZ() Получить ГК и УЗ
java.util.List<byte[]>	multiTest(FpsuKey shortKey, FpsuKey longKey, short numberTests, byte shortDigests, byte longDigests) Произвести тестирование ФПСУ
DSInfo	sign(FpsuKey key, byte[] data) Метод формирования ЭП на сервере ФПСУ
DSInfo	sign(short key_no, byte[] data) Метод формирования ЭП на сервере ФПСУ
DSInfo	signDigest(FpsuKey key, byte[] digest, ru.infocrypt.crypto.enums.DigestParamSet paramSet) Метод формирования ЭП на сервере ФПСУ
DSInfo	signDigest(short key_no, byte[] digest) Метод формирования ЭП на сервере ФПСУ

Метод check

```
public void check(java.lang.String user_id,
                  int pk_no,
                  byte[] data,
                  byte[] sign)
           throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Проверка ЭП.

Исключения:
`ru.infocrypt.fpsu.exception.FpsuException - возможные исключения`

Метод checkVersion

```
public ru.infocrypt.fpsu.FPSUSoftwareVersion checkVersion()
          throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Получение версии ПО ФПСУ-IP.

Возвращаемое значение:
`структура [FPSUSoftwareVersion]`

Исключения:
`ru.infocrypt.fpsu.exception.FpsuException - возможные исключения`

Метод createSesKey

```
public byte[] createSesKey(
                ru.infocrypt.crypto.enums.PublicKeyParamSet paramSet,
                byte[] mac,
                byte[] encryptedKey,
                byte[] ephemeralPublicKey,
                byte[] ukm,
                short key_no)
          throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Создание сессионного ключа расшифрования CMS для БиКрипт.

Параметры:
`paramSet - набор параметров открытого ключа`

`mac - МАК`

`encryptedKey - зашифрованный сессионный ключ`

`ephemeralPublicKey - открытый ключ`

`ukm - УКМ`

`key_no - номер ключа`

Возвращаемое значение:

`сессионный ключ для Бикрипта (64 байта)`

Исключения:
`ru.infocrypt.fpsu.exception.FpsuException - возможные исключения`

Метод findPk

```
public int findPk(java.lang.String pk_base_id,
                  java.lang.String user_id)
           throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Поиск в базе открытых ключей на сервере ключа пользователя user_id и возвращение его номера.

Параметры:
pk_base_id - идентификатор БОК

user_id - идентификатор ключа

Возвращаемое значение:

номер ключа

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод getFpsuKey

```
public FpsuKey getFpsuKey(short key_no)
                   throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Получение класса-описателя ключа ФПСУ.

Параметры:
key_no - номер ключа

Возвращаемое значение:

структура FpsuKey

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод getKeyIdentifier

```
public FpsuKey getKeyIdentifier(short key_no)
                     throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Получение идентификатора ключа по его номеру.

Параметры:
key_no - номер ключа

Возвращаемое значение:

идентификатор

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод getPublicKey

```
public byte[] getPublicKey(FpsuKey key)
          throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Получение из ФПСУ открытого ключа.

Параметры:

key - закрытый ключ

Возвращаемое значение:

открытый ключ в виде набора байтов

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод checkVersion

```
public int getVersion()
```

Назначение:
Получение версии сервера ФПСУ-IP.

Возвращаемое значение:
числовая интерпретация версии

Метод isPkBaseExists

```
public int isPkBaseExists(java.lang.String pk_base_id)
          throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Проверка, существует ли БОК с указанным идентификатором.

Параметры:

pk_base_id - идентификатор БОК

Возвращаемое значение:

количество ключей в базе

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод loadEncryptionKey

```
public ru.infocrypt.crypto.EncryptionKey loadEncryptionKey(int keyNumber)
                      throws ru.infocrypt.fpsu.exception.FpsuException,
                             ru.infocrypt.crypto.exception.BicryptException
```

Назначение:
Загрузка ключа шифрования.

Параметры:

keyNumber - номер секретного ключа шифрования на сервере

Возвращаемое значение:

структура EncryptionKey

Исключения:

ru.infocrypt.fpsu.exception.FpsuException – FpsuException возможные исключения

ru.infocrypt.crypto.exception.BicryptException – BicryptException возможные исключения

Метод loadEncryptionKey

```
public ru.infocrypt.crypto.EncryptionKey loadEncryptionKey(
                    int keyNumber,
                    byte[] salt)
              throws ru.infocrypt.fpsu.exception.FpsuException,
                    ru.infocrypt.crypto.exception.BicryptException
```

Назначение:
Загрузка ключа шифрования.

Параметры:

keyNumber - номер секретного ключа шифрования на сервере

Возвращаемое значение:

структура EncryptionKey

Исключения:

ru.infocrypt.fpsu.exception.FpsuException – FpsuException возможные исключения

ru.infocrypt.crypto.exception.BicryptException – BicryptException возможные исключения

Метод loadGKUZ

```
public GKUZInfo loadGKUZ()
                  throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Получение ГК и УЗ.

Возвращаемое значение:
структура с данными главного ключа и узлов замены

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод multiTest

```
public java.util.List<byte[]> multiTest(FpsuKey shortKey,
                                        FpsuKey longKey,
                                        short numberTests,
                                        byte shortDigests,
                                        byte longDigests)
                        throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Тестирование ФПСУ.

Параметры:
shortKey - тестовый ключ (короткий, 32 байта)

longKey - тестовый ключ (длинный, 64 байта)

numberTests - количество производимых тестов

shortDigests - список хеш-данных для тестов короткого ключа

longDigests - список хеш-данных для тестов длинного ключа

Возвращаемое значение:

список тестовых ЭП

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод multiTest

```
public java.util.List<byte[]> multiTest(FpsuKey shortKey,
                                        FpsuKey longKey,
                                        short numberTests,
                                        byte shortDigests,
                                        byte longDigests)
                        throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Тестирование ФПСУ.

Параметры:
shortKey - тестовый ключ (короткий, 32 байта)

longKey - тестовый ключ (длинный, 64 байта)

numberTests - количество производимых тестов

shortDigests - список хеш-данных для тестов короткого ключа

```
longDigests - список хеш-данных для тестов длинного ключа
```

Возвращаемое значение:

```
список тестовых ЭП
```

Исключения:

```
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

Метод sign

```
public DSInfo sign(sign(FpsuKey key,
                  byte[] data)
            throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Формирование ЭП на сервере ФПСУ.

Параметры:

```
key - ключ ФПСУ, которым осуществляется подпись
[ru.infocrypt.fpsu.FpsuKey]

data - данные на подпись
```

Возвращаемое значение:

```
структура DSInfo с информацией об ЭП
```

Исключения:

```
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

Метод sign

```
public DSInfo sign(short key_no,
                  byte[] data)
            throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Формирование ЭП на сервере ФПСУ.

Параметры:

```
key_no - номер ключа ФПСУ, которым осуществляется подпись

data - данные на подпись
```

Возвращаемое значение:

```
структура DSInfo с информацией об ЭП
```

Исключения:

```
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения
```

Метод signDigest

```
public DSInfo signDigest(FpsuKey key,
                  byte[] digest,
                  ru.infocrypt.crypto.enums.DigestParamSet paramSet)
            throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Формирование ЭП на сервере ФПСУ.

Параметры:

key - ключ ФПСУ, которым осуществляется подпись

digest - хэш данных на подпись

paramSet - набор параметров расчета хеша [DigestParamSet]

Возвращаемое значение:

структура DSInfo с информацией об ЭП

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения

Метод signDigest

```
public DSInfo signDigest(short key_no,
                  byte[] digest)
            throws ru.infocrypt.fpsu.exception.FpsuException
```

Назначение:
Формирование ЭП на сервере ФПСУ.

Параметры:

key_no - номер ключа ФПСУ, которым осуществляется подпись

digest - хэш данных на подпись

Возвращаемое значение:

структура DSInfo с информацией об ЭП

Исключения:
ru.infocrypt.fpsu.exception.FpsuException - возможные исключения