ESMART® TOKEN ГОСТ

Описание ESMART Token ГОСТ

Содержание

1. Введение 3
1.1 Инфраструктура открытых ключей 3
1.2 Аппаратно-программный комплекс 4
2. ESMART Token 5
2.1 Смарт-карта ESMART Token 5
2.2 USB-ключ ESMART Token 5
2.3 Оснащение бесконтактным интерфейсом 6
2.4 Основные характеристики 7
2.5 Преимущества ESMART Token 8
3. Архитектура 9
3.1 Windows 9
3.2 Linux 10
4. Проверка целостности ПО 11
4.1 Определение серийного номера и проверка целостности с помощью java-утилиты 11
4.2 Определение серийного номера с помощью приложения ESMART PKI Client 12
4.3 Возможные проблемы 13
5. ПИН-код 14
6. Возможности использования ESMART Token ГОСТ 15
6.1 ESMART Token CSP 15
6.2 ESMART Token PKCS#11 15
6.3 ESMART Token Java API 15
7. Приложение ESMART PKI Client 16
7.1 Интерфейс приложения 17
8. Работа с ESMART Token 18
8.1 Задачи администратора 18
8.2 Задачи пользователя 18
Приложение. Состав дистрибутива 20

1. Введение

Программно-аппаратный комплекс ESMART Token предназначен для повышения уровня информационной безопасности организаций, использующих пароли. ESMART Token позволяет производить двухфакторную авторизацию с использованием смарт-карт или USB-ключей.

Недостатки решений на основе паролей:

- *Пользователи задают слишком простые пароли, которые можно угадать перебором;*
- *Число попыток ввода пароля не всегда ограничено;*
- *Пользователи забывают сложные пароли;*
- *Однофакторная авторизация по паролю не достаточно защищена;*
- *Компрометация паролей может быть не заметна в течение длительного периода времени, а утеря карты или ключа обнаруживается практически сразу.*

Комплекс ESMART Token позволяет использовать все преимущества двухфакторной авторизации: предъявление физического носителя (карты или USB-ключа) и ввод ПИН-кода, защищенного от взлома методом перебора.

Новый ключевой носитель ESMART Token ГОСТ на базе отечественной микросхемы MIK51SC72D позволяет расширить возможности использования смарт-карт за счет биометрической аутентификации по отпечатку пальца.

Ключевой носитель ESMART Token ГОСТ с аппаратной поддержкой российских криптографических алгоритмов соответствует требованиям ФСБ России (Сертификат №СФ/124-2000) к шифровальным (криптографическим) средствам класса КС3 и может использоваться в качестве средства криптографической защиты информации (СКЗИ), не представляющей государственную тайну.

Сертификация СКЗИ распространяется только на использование отечественных криптографических алгоритмов: ГОСТ 28147-89, ГОСТ Р 34.10-2001, ГОСТ Р 34.11-94, вычисление и проверка ЭЦП ГОСТ Р34.10-2012 в 256-битном режиме, вычисление хэш-функции ГОСТ Р34.11-2012 в 256-битном режиме.

*При эксплуатации следует соблюдать **Правила пользования СКЗИ**. Правила пользования конкретным СКЗИ можно получить, отправив запрос на esmart@isbc.ru с указанием серийного номера чипа СКЗИ, даты приобретения и наименования юридического лица покупателя.*

1.1 Инфраструктура открытых ключей

Обеспечение информационной безопасности предприятия может быть основано на Инфраструктуре открытых ключей PKI (англ. Public Key Infrastructure). Инфраструктура открытых ключей подразумевает использование ключевой пары: открытого и закрытого ключа, математически связанных друг с другом. Основополагающим принципом PKI является то, что закрытый ключ известен только его владельцу и должен быть надежно защищен.

В качестве носителей рекомендуется использовать не обычные флеш-накопители, а криптографические устройства в форме карты или USB-ключа. Преимущество хранения закрытых ключей на криптографической карте состоит в том, что генерация ключевой пары и операции, требующие предоставления закрытого ключа, выполняются криптографическим процессором карты или USB-ключа.

Помимо ключевой пары на карте или USB-ключе хранится сертификат пользователя. Сертификат формата X.509 является фиксированным набором сведений о пользователе и ключевой паре.

ESMART Token ГОСТ легко встраивается в существующую инфраструктуру (PKI) предприятия.

1.2 Аппаратно-программный комплекс

Аппаратная часть:

- *Смарт-карты ESMART Token ГОСТ;*
- *рекомендуемые считыватели без биометрии Advanced Card Systems Ltd.:*
  - *ACR38U, ACR38K, ACR128, ACR1281;*
- *Рекомендуемые биометрические считыватели[1]:*
  - *AET65, Futronic FS82;*
- *USB-ключ ESMART Token ГОСТ;*

Программная часть:

- *Пользовательское приложение ESMART BIO PKI Client;*
- *ESMART Token CSP;*
- *ESMART Token PKCS11;*
- *Инструменты разработчика:*
  - *JNI-wrapper для Java с примерами;*
  - *Примеры для C++;*
- *Plug-in для КриптоПро CSP;*
- *Драйверы оборудования.*

[1] *Сканеры отпечатков файлов AET65 и Futronic82 не взаимозаменяемы. Отпечаток пальца, записанный на считывателе одной модели, может быть прочитан только на считывателе той же модели*

2. ESMART Token

Принцип работы смарт-карты и USB-ключа ESMART Token ГОСТ одинаков, следует выбирать наиболее удобный и подходящий вариант. Для офисного использования прекрасно подходит сочетание небольшого настольного считывателя и смарт-карты, которая помещается в кошелек. Носитель в виде USB-ключа фактически совмещает в себе миниатюрный считыватель и чип в одном корпусе, USB-ключ можно порекомендовать тем, кто часто бывает в разъездах.

Возможность, цены, условия и сроки поставки карт и USB-ключей ESMART Token ГОСТ с нестандартным дизайном зависят от объема заказа и оговариваются отдельно.

2.1 Смарт-карта ESMART Token

Модель ESMART Token в виде смарт-карты размера ID-1.

Лицевая сторона:

Опционально смарт-карта ESMART Token ГОСТ может поставляться со встроенной RFID-меткой[2] для использования в системах контроля и управления доступом (СКУД).

Для работы со смарт-картой ESMART Token ГОСТ требуется наличие считывателя контактных смарт-карт и свободный порт USB или RS-232 для подключения считывателя. На ПК пользователя должны быть установлены драйвера для выбранной модели считывателя и ПО для работы с картой посредством CSP (только Windows) и PKCS#11 (Windows, Linux, Mac OS).

Для выполнения основных операций с ESMART Token разработана программа ESMART PKI Client с графическим интерфейсом. См. раздел ***Приложение ESMART PKI Client****.*

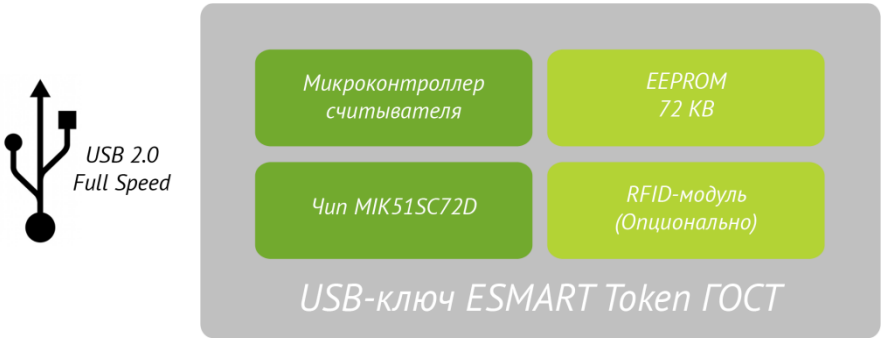
2.2 USB-ключ ESMART Token

USB-ключ ESMART Token представляет собой комбинацию считывателя и чипа смарт-карты в виде миниатюрного устройства. Устройство подключается к ПК напрямую, отдельный считыватель смарт-карт не требуется. На каждый ПК должен быть установлен драйвер устройства для соответствующей операционной системы.

Ключ ESMART Token может быть оснащен встроенной RFID-меткой[3].

[2] *Цены и условия поставки ESMART Token с RFID-меткой оговариваются отдельно*

[3] *Цены и условия поставки ESMART Token с Flash-памятью и/или RFID-меткой оговариваются отдельно*



2.3 Оснащение бесконтактным интерфейсом

Смарт-карты ESMART® Token ГОСТ могут оснащаться бесконтактным интерфейсом для интеграции в СКУД и системы управления логическим доступом

ISO 18000-2 (125 kHz)	*EM 4102*	*HID ISOProx II*	*HID Indala*	*Atmel T5577*
ISO 14443. ISO 15693	*NXP Mifare Classic*	*NXP Mifare Plus*	*NXP Mifare DESFire*	*NXP iCode*

2.4 Основные характеристики

Форм-фактор	Смарт-карта ID-1	USB-ключ
Подключение	Требуется считыватель	USB Plug&Play
Интерфейс	ISO 7816-2 ISO 14443	USB 2.0 Full Speed
Размер	ID-1 (85,6 × 53,98 мм) ID-000 (25,0 × 15,00 мм)	53,5 × 15,7 × 7,8 мм
Объем защищенной памяти EEPROM	72 Кбайт	
Функциональность операционной системы	Виртуальная машина Java Card 3.0 Classic. Соответствие стандарту управления приложениями Global Platform 2.2 Верификация пользователя по отпечатку пальца (технология Match-on-Card) Платежное приложение EMV (MasterCard M/Chip 4.1).	
Аппаратно реализованные алгоритмы	**Алгоритмы вычисления/проверки электронной подписи:** ГОСТ Р34.10-2001, ГОСТ Р 34.10-2012, RSA-1024, RSA-2048, ECDSA-256 **Алгоритмы шифрования:** ГОСТ 28147-89, DES, Triple DES, AES-128, AES-192, AES-256 **Алгоритмы хеширования:** ГОСТ Р 34.11–1994, ГОСТ Р 34.11–2012 SHA-1, SHA-256 Выработка сессионных ключей по схеме VKO GOST R 34.10-2001 (RFC4357); Аппаратный генератор случайных чисел	
Время вычисления/проверки ЭП ГОСТ	90 мс / 173 мс	
Поддерживаемые интерфейсы и стандарты	PKCS#11 версии 2.30, Microsoft CryptoAPI PC/SC, Microsoft CCID Сертификаты X.509 v3, SSL v3, IPSec/IKE ISO 7816 части 1, 2, 3, 4, 8, 9 ISO 14443 части 1, 2, 3, 4	
Поддерживаемые операционные системы	Microsoft Windows XP/2003/Vista/2008/7/8 (32/64-бит), MacOS X, Linux	
Возможность встраивания RFID-метки	EM-Marine, Mifare, ICODE, HID Prox, HID Indala, HID iClass	

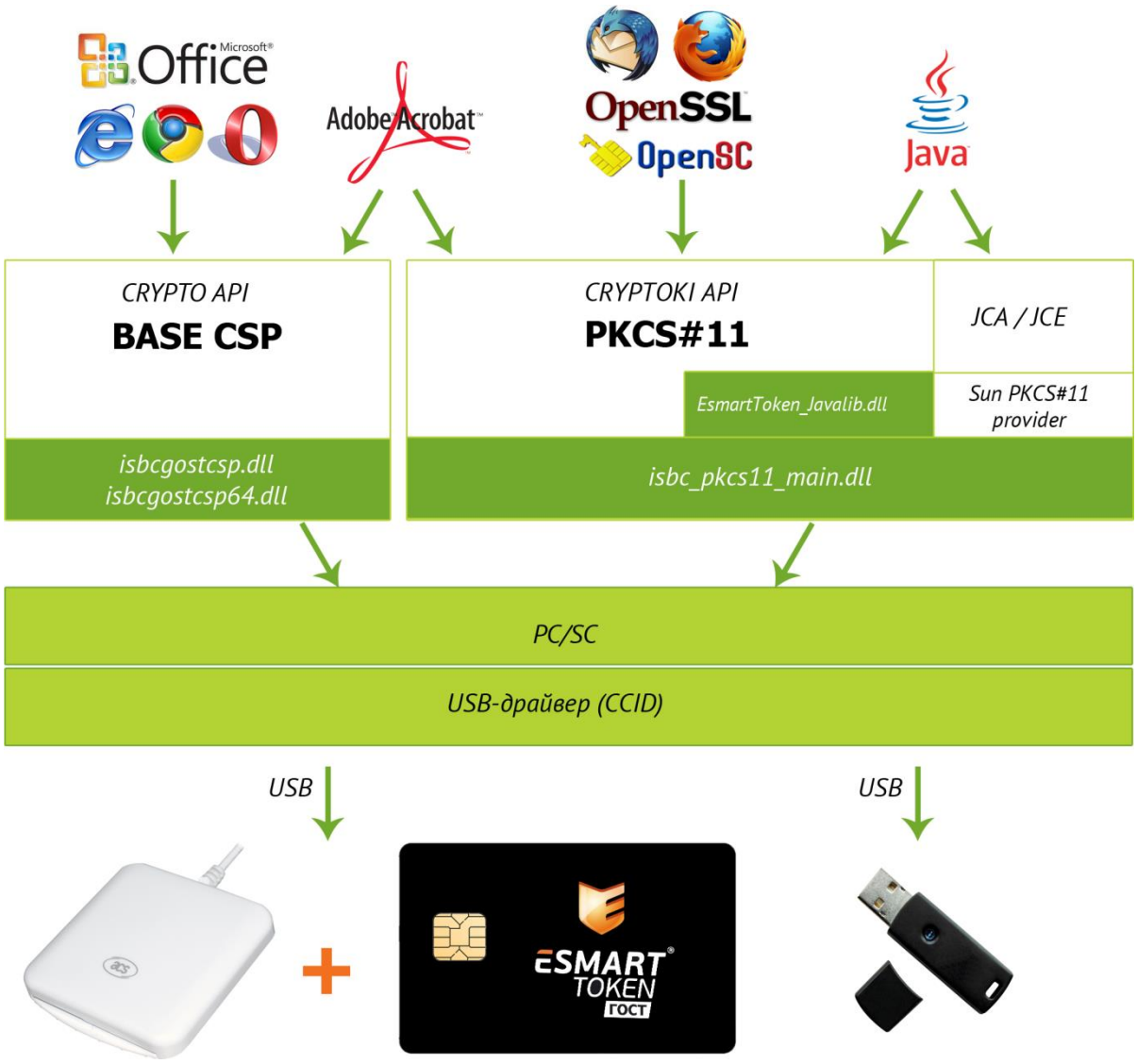
Спецификация может быть изменена без предварительного уведомления

2.5 Преимущества ESMART Token

- *Российские криптографические алгоритмы ГОСТ реализованы нативно в ОС смарт-карты в маске чипа (не используются Java апплеты для реализации ГОСТ);*
- *Аппаратная поддержка биометрической аутентификации по отпечаткам пальцев Match-on-Card на уровне СКЗИ. Поддерживаются как протяжные, так и оптические сканнеры отпечатков;*
- *Дуальный интерфейс (контактный ISO7816 и бесконтактный ISO14443);*
- *Поддержка алгоритмов RSA с ключом до 2048 байт;*
- *Встроенная виртуальная машина Java (полностью совместимая со стандартами Java Card 3.0 Classic и Global Platform 2.2) с поддержкой ГОСТ;*
- *Наличие сертифицированного апплета M/Chip (MasterCard) позволяет использовать смарт-карту одновременно как средство ЭП/аутентификации, так и как банковскую карту;*
- *Разработка и производство микросхемы MIK51 (Микрон) в России гарантируют отсутствие аппаратных «закладок»;*
- *Защищенный контроллер обеспечивает защиту от атак вида SPA, DPA, DFA. Наличие датчиков вскрытия, света, импульсной помехи, напряжения, температуры, а также аппаратный контроль целостности и возможность быстрого стирания EEPROM надежно защищают ключевую информацию от попыток вскрытия чипа и несанкционированного доступа;*
- *Два форм-фактора: карта и USB-ключ;*
- *Поддержка Windows, Linux и Mac OS;*
- *Большой объем памяти для хранения нескольких пар ключей и сертификатов;*
- *Возможность записи и безопасного хранения произвольных данных;*
- *Пользовательское приложение с графическим интерфейсом ESMART PKI Client;*
- *Встроенная RFID-метка (опционально).*

3. Архитектура

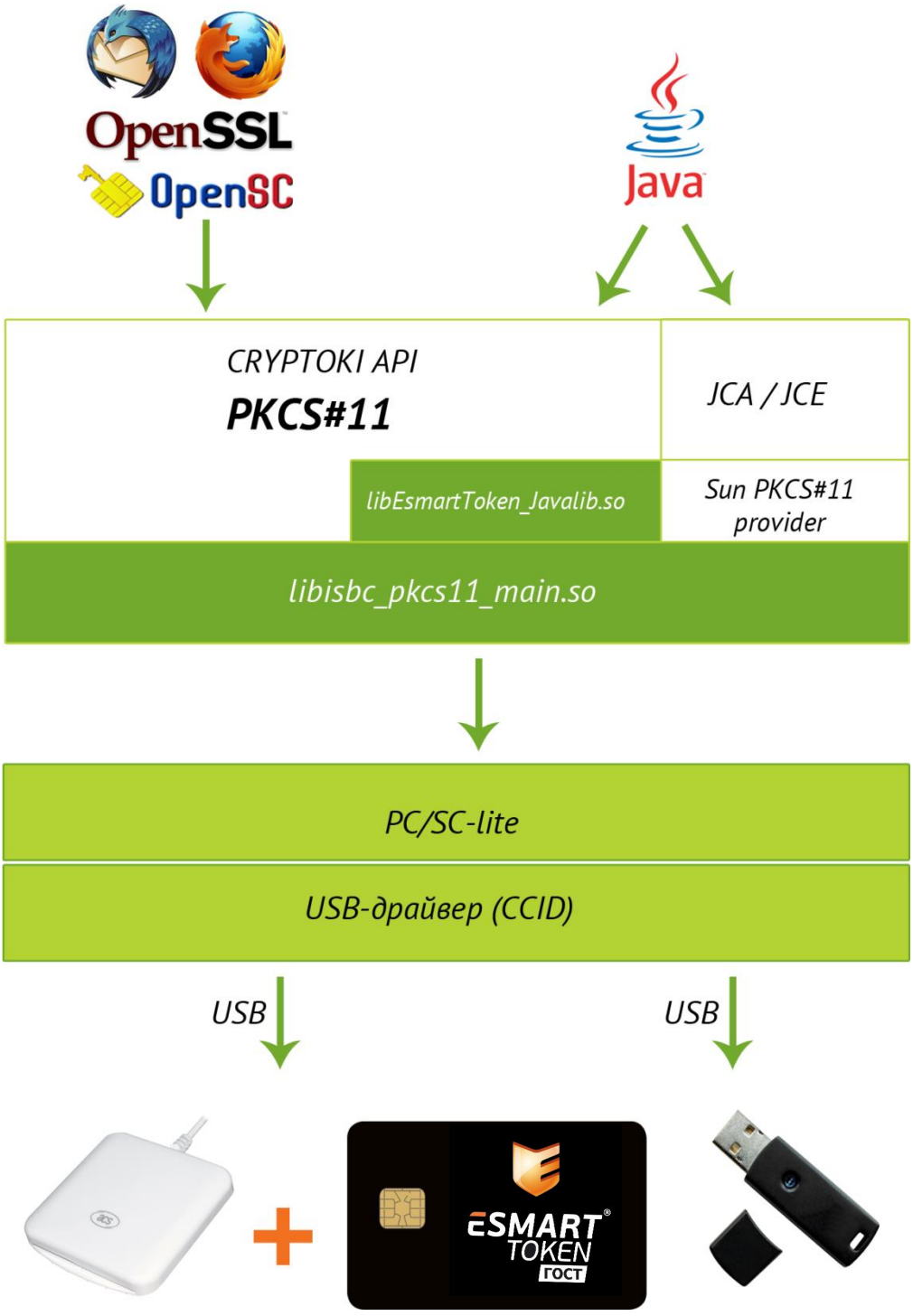
3.1 Windows



Примечания:

1) *JCA/JCE – см. ESMART Token Java API;*
2) *Темно-зеленым цветом выделены библиотеки, входящие в состав комплекса ESMART Token;*
3) *Base CSP для Windows XP устанавливается отдельно, См. раздел ESMART Token CSP.*

3.2 Linux



Примечания:

1) *JCA/JCE – см. ESMART Token Java API;*
2) *Темно-зеленым цветом выделены библиотеки, входящие в состав комплекса ESMART Token.*

4. Проверка целостности ПО

СКЗИ имеет встроенную систему проверки целостности внутреннего ПО носителя методом проверки контрольной суммы. Для каждого носителя необходимо получить транспортный ключ (последовательность символов).

Получить транспортный ключ можно, отправив запрос на адрес электронной почты esmart@isbc.ru с указанием серийного номера чипа СКЗИ, даты приобретения и наименования юридического лица.

Частота проверки целостности определяется Правилами пользования СКЗИ.

Серийный номер чипа можно узнать с помощью бесплатного программного обеспечения: утилиты для проверки целостности или приложения ESMART PKI Client.

4.1 Определение серийного номера и проверка целостности с помощью java-утилиты

Для проверки целостности применяется кросс-платформенная утилита на базе языка Java.
Для работы утилиты требуется установленная среда выполнения Java Runtime Environment 6 или выше. Скачать бесплатный пакет Oracle Java Runtime Environment для операционных систем Windows, Linux и Mac OS X можно на официальном сайте java.com.

*Скачать утилиту можно на странице **Загрузки** на сайте esmart.ru в разделе*
Модули поддержки ESMART Token ГОСТ › Утилита проверки контрольной суммы.

Команда проверки целостности одинакова для Windows, Linux и Mac OS X, отличается только путь к файлу esmart-token-gost-checker-1.0.jar. Для проверки введите в терминале

```
java -jar esmart-token-gost-checker-1.0.jar
```

Если путь содержит пробелы, его необходимо поместить в кавычки, например:

```
java -jar "c:\\some path\esmart-token-gost-checker-1.0.jar"
```

***Пример успешного выполнения проверки на ОС Windows** (с комментариями на русском языке):*

```
user>java -jar esmart-token-gost-checker-1.0.jar
The checker allows to verify the integrity of CKZI ESMART Token GOST.
You have to know the unique transport key which you can obtain by
sending UID of CKZI to esmart@isbc.ru
```

Утилита позволяет проверить целостность СКЗИ ESMART Token GOST. Для проверки необходимо знать уникальный транспортный ключ, который можно получить, отправив серийный номер СКЗИ (UID) на адрес электронной почты esmart@isbc.ru

```
ACS CCID USB Reader 0: ESMART Token GOST found. UID: 346066FD04189104
```

*Утилита обнаружила один считыватель с карточкой ESMART Token GOST с серийным номером (UID) **346066FD04189104.***

```
Working with ACS CCID USB Reader 0
Please provide transport key for UID 346066FD04189104:
```

Выполняется операция со считывателем ACS CCID USB Reader 0. Введите транспортный ключ для СКЗИ с серийным номером 346066FD04189104.

```
D1FB3E9A53738515
```

*На запрос вводим транспортный ключ, в данном случае последовательность **D1FB3E9A53738515.***
(Данный транспортный ключ приведен исключительно для примера, он будет не верен для любого СКЗИ ESMART Token GOST).

```
Send > 8020000008D1FB3E9A53738515
Recv > 9000
Transport key is accepted.
```

В ответ на отправленные команды, получен ответ:

Транспортный ключ принят.

```
Send >
800C02002053ED6D4E4C23B272F5AFE7FD0B75DC5BF88577B4E33E1C608A80E3382FC1C99B
Recv > 207775E09000
INTEGRITY CHECKING PASSED.
End program. Thank you!
```

В ответ на отправленные команды проверки целостности, получен ответ:

ПРОВЕРКА ЦЕЛОСТНОСТИ ПРОЙДЕНА УСПЕШНО.

Завершение программы. Спасибо!

Неверный транспортный ключ

Если был введен неверный транспортный ключ, появится соответствующее сообщение и кол-во оставшихся попыток:

```
Invalid key. 7 attempts left.
```

Неверный ключ. Осталось 7 попыток.

***Внимание!** Если 8 раз подряд ввести неверный транспортный ключ, СКЗИ ESMART Token GOST будет заблокировано **без возможности восстановления**.*

СКЗИ не прошел проверку целостности

Следующее сообщение появляется, если СКЗИ не прошел проверку целостности ПО:

```
INTEGRITY CHECKING NOT PASSED. You must stop using this product!
```

ПРОВЕРКА ЦЕЛОСТНОСТИ НЕ ПРОЙДЕНА. Данный продукт нельзя использовать!

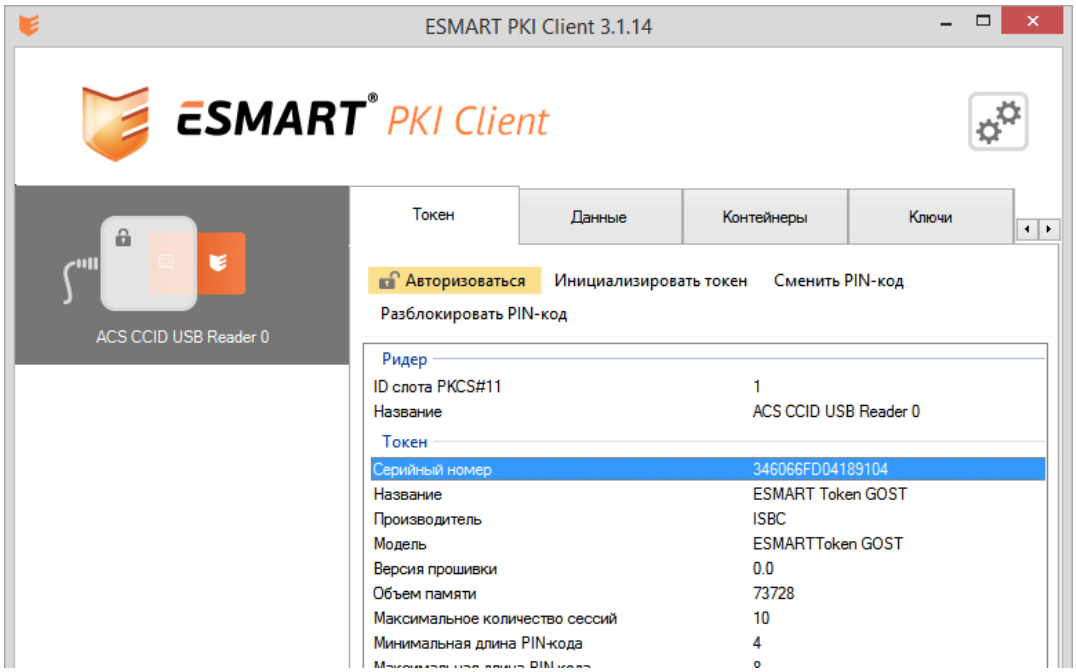
4.2 Определение серийного номера с помощью приложения ESMART PKI Client

Данный способ также позволяет узнать серийный номер носителя с помощью приложения с графическим интерфейсом. Данный способ может быть удобнее, например, для составления списка серийных номеров, который будет отправлен на адрес esmart@isbc.ru для запроса транспортных ключей.

Бесплатное приложение ESMART PKI Client позволяет выполнять ряд операций со смарт-картами и USB-ключами ESMART Token и ESMART Token GOST, в том числе получить справочную информацию о носителе. Для данной операции ввод ПИН-кода не требуется.

*Установка и основы работа с приложением ESMART PKI Client описана в документе **ESMART PKI Client — Руководство администратора**. Руководства и дистрибутив приложения ESMART PKI Client можно скачать на сайте esmart.ru в разделе **Загрузки**.*

*На рисунке показано окно приложения ESMART PKI Client. Синим цветом выделена строка, в которой отображается серийный номер чипа СКЗИ. Чтобы скопировать номер в буфер обмена, нажмите правую кнопку мыши, когда указатель будет над строкой **Серийный номер**. В появившемся меню выберите пункт **Копировать значение**.*



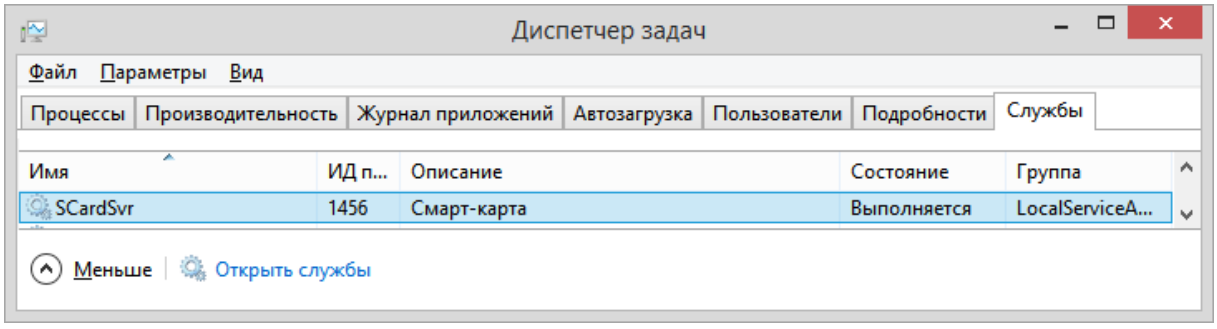
4.3 Возможные проблемы

Остановлена служба смарт-карт

Если на ПК на ОС Windows никогда не использовались смарт-карты, служба работы со смарт-картами может быть отключена. В этом случае в приложении ESMART PKI Client не отображаются считыватели, а утилита выдаст следующую ошибку:

```
java.lang.NullPointerException
        at com.isbc.utils.EsmartTokenGostChecker.main(EsmartTokenGostChecker.java:76)
```

Запустите службу вручную с помощью диспетчера задач или панели управления службами. Название службы ***Смарт-карта (SCardSRV)****.*



Неверное кол-во символов в транспортном ключе

Если введено неверное кол-во символов транспортного ключа, появится следующее сообщение об ошибке.

```
java.lang.IllegalArgumentException: Invalid APDU: length=12, b1=8
        at javax.smartcardio.CommandAPDU.parse(Unknown Source)
        at javax.smartcardio.CommandAPDU.<init>(Unknown Source)
        at
com.isbc.utils.EsmartTokenGostChecker.main(EsmartTokenGostChecker.java:89)
```

5. ПИН-код

ПИН-код предназначен для безопасного доступа к информации, хранящейся на карте или USB-ключе. ПИН-код задается ASCII-символами и должен иметь длину от 4 до 8 знаков. Не рекомендуется использовать слабые ПИН-коды (например 1234) и ПИН-коды по умолчанию[4].

ESMART Token использует следующие типы ПИН-кода:

- ***ПИН-код пользователя (User PIN)**, используется для смены ПИН-кода пользователя, авторизации на карте для получения доступа к ключам, генерации объектов, операциях подписания и шифрования из пользовательских программ.*
- ***ПИН-код администратора (SO[5] PIN)**, используется для инициализации карты, для очистки карты, для смены ПИН-кода пользователя.*

Два ПИН-кода позволяют обеспечить более высокую степень безопасности за счет разделения полномочий. Рекомендуется сменить все ПИН-коды сразу после получения карты.

Смарт-карты и USB-ключи защищены от взлома методом перебора ПИН-кода. ПИН-код блокируется после 10 неудачных попыток набора подряд. Сменить ПИН-код пользователя, когда карта заблокирована, можно только при помощи ПИН-кода администратора. Карту также можно повторно инициализировать, но в этом случае все сгенерированные ключевые пары, сертификаты и данные будут потеряны. Повторная инициализация может использоваться, например, если ключевая пара и сертификат записываются на карту из файла PKCS#12 или PFX.

Внимание! Количество неверных попыток набора ПИН-кода администратора (SO PIN) задается при инициализации карты. По умолчанию количество попыток 10. Если неверно набрать ПИН-код администратора указанное количество раз, карта блокируется без возможности восстановления или повторной инициализации.

[4] *ПИН-код пользователя по умолчанию «12345678», ПИН-код администратора по умолчанию «12345678»*

[5] *Security Officer - сотрудник по вопросам информационной безопасности*

6. Возможности использования ESMART Token ГОСТ

6.1 ESMART Token CSP

CSP[6] – криптопровайдер, т.е. модуль, позволяющий производить криптографические операции в ОС Windows через CryptoAPI. Приложения не работают непосредственно с криптопровайдером, они вызывают функции CryptoAPI из библиотек Advapi32.dll и Crypt32.dll.

Для работы в Windows XP требуется установка пакета Microsoft Base CSP http://www.microsoft.com/en-us/download/details.aspx?id=4670

Модуль предназначен для авторизации в Windows в домене и работы ESMART Token ГОСТ с приложениями, включая:

- *Программы пакета MS Office;*
- *Программы пакета Open Office;*
- *Почтовый клиент MS Outlook;*
- *Adobe Acrobat;*
- *Internet Explorer.*

*Подробно работа с приложениями описана в руководствах **ESMART Token – CSP** и **ESMART Token – Настройка пользовательских приложений в Windows**.*

6.2 ESMART Token PKCS#11

PKCS#11 – открытый кроссплатформенный стандарт взаимодействия операционной системы и смарт-карты. В отличие от CSP, сертификаты не хранятся в стандартном хранилище Windows. Стандарт PKCS#11 является основным средством работы со смарт-картами в Linux и Mac OS.

Модуль предназначен для работы ESMART Token с приложениями, включая:

- *Mozilla Firefox;*
- *Mozilla Thunderbird;*
- *Adobe Acrobat;*
- *OpenVPN;*
- *OpenSSL.*

*Подробно работа с приложениями описана в руководствах **ESMART Token – PKCS11** и **ESMART Token – Настройка пользовательских приложений в Windows**.*

6.3 ESMART Token Java API

Java API предназначено для вызова некоторых функций библиотеки PKCS#11 из Java. Вызовы функций библиотеки PKCS#11 осуществляются через технологию JNI (Java Native Interface). Возможно использование библиотеки в JavaScript с использованием технологии Java Applet.

Для использования ESMART Token в инфраструктурах, поддерживающих Java Cryptography Architecture (JCA) и Java Cryptography Extension (JCE), можно использовать Java wrapper над PKCS#11 от компании Oracle. Подробнее: http://docs.oracle.com/javase/7/docs/technotes/guides/security/p11guide.html.

*Подробно работа с Java API и демонстрационным апплетом описана в руководстве **ESMART Token – PKCS11 Java**.*

---

[6] *Cryptographic service provider*

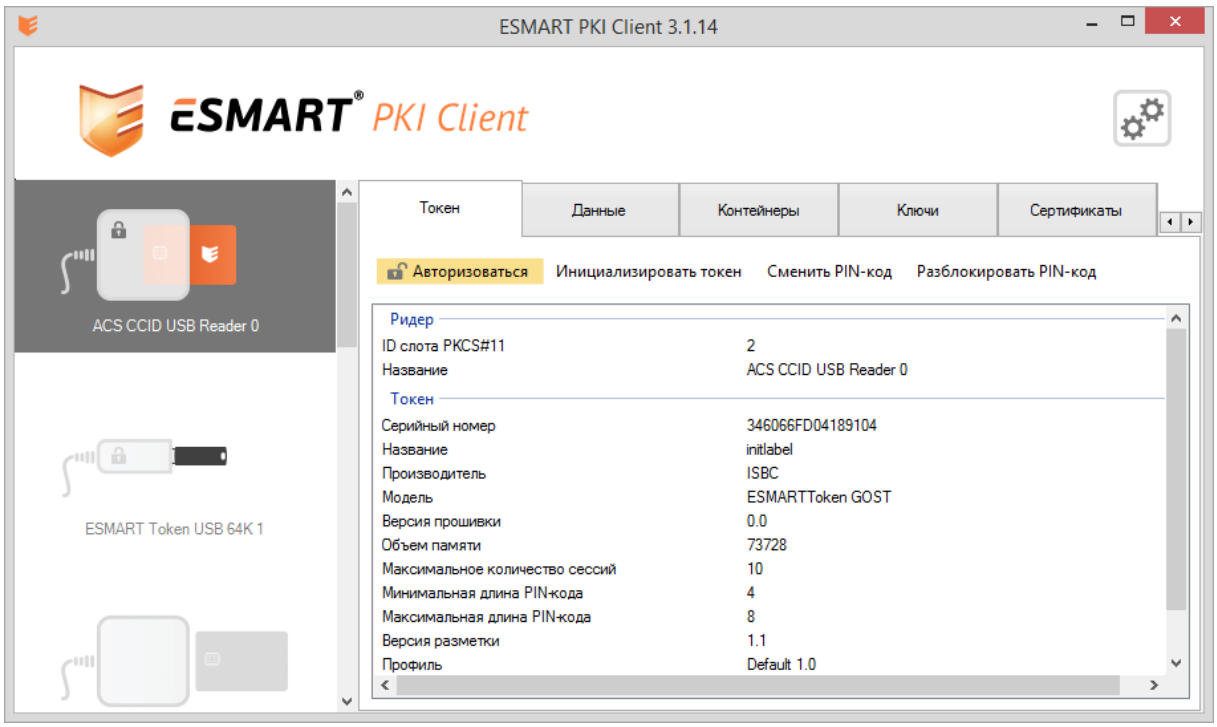
7. Приложение ESMART PKI Client

Программа настройки смарт-карт ESMART Token с удобным графическим интерфейсом позволяет выполнить все операции с картой:

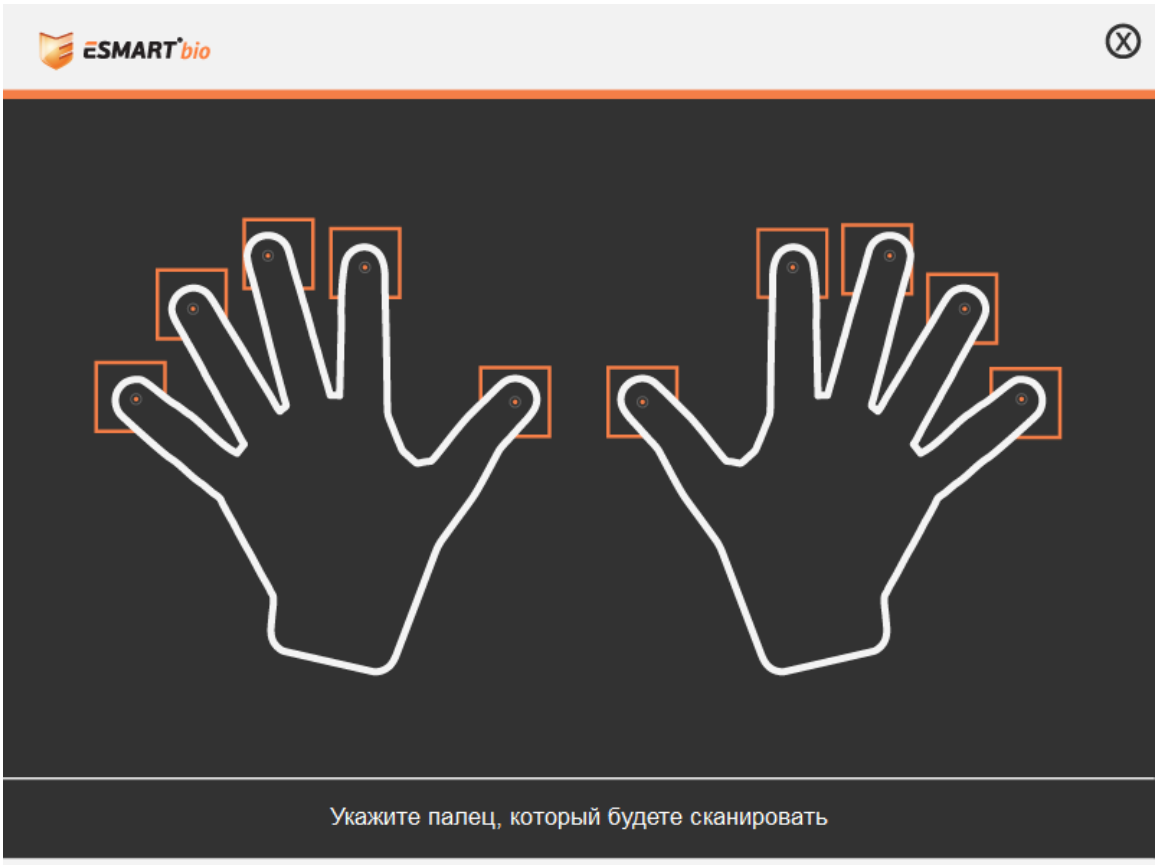
- *Просмотр информации о ESMART Token;*
- *Инициализация карты или USB-ключа ESMART Token;*
- *Смена и разблокировка ПИН-кода пользователя и ПИН-кода администратора;*
- *Просмотр объектов на ESMART Token;*
- *Генерация ключей RSA и ГОСТ;*
- *Создание запроса на сертификат;*
- *Запись сертификата на ESMART Token;*
- *Запись произвольных данных на ESMART Token;*
- *Подпись файла и проверка подписи в формате PKCS#7 (отделяемая и неотделяемая);*
- *Импорт файлов PKCS#12 или PFX на ESMART Token;*
- *Поддерживается работа сразу с несколькими считывателями смарт-карт и/или USB-ключами.*

7.1 Интерфейс приложения

Пример главного окна приложения:



Пример окна биометрической аутентификации – выбор пальца для сканирования:



8. Работа с ESMART Token

8.1 Задачи администратора

В задачи администратора могут входить:

- *Установка драйверов оборудования;*
- *Установка ПО для работы с ESMART Token;*
- *Инициализация ESMART Token;*
- *Генерация ключей и создание запроса на сертификат;*
- *Запись сертификата на ESMART Token;*
- *Разблокировка ПИН-кода;*
- *Настройка приложений для использования с ESMART Token.*

8.2 Задачи пользователя

В задачи пользователя могут входить:

- *Генерация ключей;*
- *Выписка запроса на сертификат;*
- *Запись сертификата на ESMART Token;*
- *Смена ПИН-кода;*
- *Запись произвольных данных, например, логинов и паролей;*
- *Использование в приложениях.*

Приложение. Состав дистрибутива

- *Documentation:*
  - *Руководства администратора;*
  - *Руководства пользователя;*
  - *Руководства разработчика;*
- *SDK:*
  - *Средства для использования ESMART Token в Java; Примеры на Java и JavaScript;*
  - *Примеры на C++;*
- *Windows:*
  - *ESMART PKI Client;*
  - *Пакет драйверов оборудования;*
  - *Библиотеки для установки вручную;*
  - *Утилита OpenSC с плагинами для OpenSSL;*
  - *Плагин для КриптоПро CSP.*