

АО «СИГНАЛ-КОМ»

ПРОГРАММНЫЙ КОМПЛЕКС  
SIGNAL-COM AUTHKEY  
СЕРВЕР ЭЛЕКТРОННОЙ ПОДПИСИ  
Версия 1.0  
Руководство системного программиста

ШКНР.00072-01 32 01

Листов 15

---

## АННОТАЦИЯ

Средство имитозащиты с аутентификацией - программный комплекс «Signal-COM AuthKey» предназначено для обеспечения целостности и аутентификации данных, а также для аутентификации источника данных с использованием симметричных криптографических алгоритмов.

Настоящий документ содержит руководство системного программиста сервера электронной подписи (далее «Signal-COM AuthKey Server»), компонента программного комплекса «Signal-COM AuthKey», предназначенного для обслуживания ключей электронной подписи и для проверки электронной подписи.

---

## СОДЕРЖАНИЕ

Аннотация .....	2
Содержание .....	3
1. Общие сведения о программе.....	4
1.1. Назначение программы .....	4
1.2. Список сокращений.....	4
1.3. Термины и определения.....	4
1.4. Характеристики программы .....	4
2. Структура программы .....	6
3. Настройка программы.....	7
3.1. Подготовка окружения.....	7
3.1.1. Подготовка работы с токенами .....	7
3.2. Конфигурирование приложения .....	7
3.3. Настройка базы данных .....	9
3.4. Настройка TLS .....	9
3.5. Методы аутентификации клиента .....	10
3.6. Датчик случайных чисел.....	10
3.7. Настройка документации.....	11
3.8. Настройка Spring Boot Actuator.....	11
3.9. Запуск программы .....	11
3.10. Требования по безопасности .....	11
4. Проверка программы.....	13
5. Сообщения системному программисту .....	14
Литература .....	15

## **1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ**

### **1.1. Назначение программы**

Программный комплекс «Signal-COM AuthKey» предназначен для обеспечения целостности и аутентификации данных, а также для аутентификации источника данных с использованием симметричных криптографических алгоритмов.

Программный комплекс «Signal-COM AuthKey» может использоваться в информационных системах, предназначенных для хранения и обработки персональных данных, конфиденциальной, служебной, коммерческой и другой информации, не содержащей сведений, составляющих государственную тайну, а также для обмена такой информацией и обеспечения юридической значимости электронных документов.

Сервер ЭП «Signal-COM AuthKey Server» входит в состав программного комплекса «Signal-COM AuthKey» и предназначен для выполнения следующих функций:

- ведение реестра идентификаторов клиентов;
- ведение реестра идентификаторов ключей ЭП клиентов;
- генерация симметричного ключа ЭП клиента;
- выдача по запросу ключа ЭП клиента, с возможностью парольной защиты ключа;
- выдача по запросу информации о ключе ЭП клиента (дата создания, дата последнего использования, статус и т.п.);
- проверка ЭП документа с использованием ключа ЭП клиента.

«Signal-COM AuthKey Server» реализован в виде веб-сервиса и предоставляет программный интерфейс по протоколу REST.

### **1.2. Список сокращений**

В настоящем руководстве используются следующие сокращения:

- ПИН - персональный идентификационный номер;
- ПК – программный комплекс;
- ПО – программное обеспечение;
- ЭП – электронная подпись;
- API - прикладной программный интерфейс;
- REST - Representational State Transfer;
- RFC – Request for Comments;
- TLS - Transport Layer Security.

### **1.3. Термины и определения**

В настоящем руководстве используются следующие термины:

- веб-сервис – реализация интерфейса взаимодействия между различными приложениями по протоколу REST;
- ключ электронной подписи – уникальная последовательность символов, предназначенная для создания электронной подписи;
- электронная подпись – информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию.

### **1.4. Характеристики программы**

«Signal-COM AuthKey Server» реализован в виде веб-сервиса и предоставляет программный интерфейс по протоколу REST.

В качестве алгоритмов, обеспечивающих целостность и аутентификацию данных, а также аутентификацию источника данных используются алгоритмы HMAC\_GOSTR3411\_2012\_256 и HMAC\_GOSTR3411\_2012\_512, реализованные согласно RFC 2104 (HMAC: Keyed-Hashing for Message Authentication) и Рекомендациям по стандартизации Р 50.1.113 2016 «Информационная

---

технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования».

Для генерации ключей ЭП клиентов используется алгоритм диверсификации KDF\_TREE\_GOSTR3411\_2012\_256 в соответствии с Рекомендациями по стандартизации Р 50.1.113 2016 «Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования». Для генерации исходного ключа диверсификации используется криптографически стойкий ДСЧ из состава СКЗИ. Предусмотрена опциональная возможность хранения ключа диверсификации в разделённом по пороговой схеме виде.

Алгоритмы HMAC\_GOSTR3411\_2012\_256 и KDF\_TREE\_GOSTR3411\_2012\_256 основаны на использовании хэш-функции ГОСТ Р 34.11-2012 с длиной выходного значения 256 бит. Алгоритм HMAC\_GOSTR3411\_2012\_512 основан на использовании хэш-функции ГОСТ Р 34.11-2012 с длиной выходного значения 512 бит.

Для вычисления отпечатка ключа ЭП используется хэш-функция ГОСТ Р 34.11-2012 с длиной выходного значения 256 бит.

Парольное шифрование ключей ЭП клиентов реализовано с использованием алгоритма ГОСТ Р 34.12-2015 (алгоритм блочного шифрования «Кузнечик») согласно Рекомендациям по стандартизации Р 1323565.1.040–2022 «Информационная технология. Криптографическая защита информации. Парольная защита ключевой информации».

## **2. СТРУКТУРА ПРОГРАММЫ**

«Signal-COM AuthKey Server» состоит из следующих компонентов:

- auth-key-api-1.0.0.jar – архив, содержащий приложение и необходимые библиотеки;
- application.properties – файл тестовой конфигурации;
- native – каталог с модулями tokio для работы с токенами (см. п. 3.1.1);
- pse – каталог тестового ключевого контейнера для инициализации ДСЧ (см. п. 3.6).

### 3. НАСТРОЙКА ПРОГРАММЫ

#### 3.1. Подготовка окружения

Для развёртывания программы «Signal-COM AuthKey Server» необходимо предварительно установить:

- JRE/JDK 1.8 (рекомендуется использовать самую новую на момент установки версию);
- СУБД PostgreSQL (рекомендуемая версия - 9.6.24 и выше).

##### 3.1.1. Подготовка работы с токенами

Ключи диверсификации «Signal-COM AuthKey Server» могут храниться на токенах - отчуждаемых носителях, защищённых ПИН.

Взаимодействие с токенами осуществляется с помощью интерфейса PKCS #11.

Рекомендуется использовать токены, СКЗИ которых сертифицированы по требованиям ФСБ России к СКЗИ по классу KC1, KC2 или KC3.

Перед началом работы с токенами необходимо установить в соответствующий каталог (зависит от ОС) модуль интерфейса PKCS #11. Для каждого типа устройства используется собственный интерфейсный модуль, который (отдельно или в составе дистрибутива) можно загрузить с Web-сервера производителя устройства.

Если при подключении устройства к считывателю (например, USB-порту) на Windows появляется окно «Обнаружено новое устройство», необходимо выбрать подключение к Windows Update для поиска и установки драйвера USB CCID компании Microsoft.

Для работы с токенами необходимо также поместить модуль libtokio.so (Linux) или tokio.dll (Windows) в каталог, описанный в переменной окружения LD\_LIBRARY\_PATH (Linux) или PATH (Windows).

#### 3.2. Конфигурирование приложения

Конфигурирование приложения «Signal-COM AuthKey Server» осуществляется путём редактирования файла конфигурации application.properties.

В составе дистрибутива приложения «Signal-COM AuthKey Server» имеется тестовый файл конфигурации.

Для задания пути к файлу конфигурации необходимо использовать параметр командной строки --spring.config.location (см. п. 3.9).

Описание параметров файла конфигурации приведено в Таблица 1.

Таблица 1 Параметры файла конфигурации

Имя	Описание
spring.datasource.url	URL-адрес источника данных (DataSource).
spring.datasource.username	Имя пользователя для доступа к источнику данных.
spring.datasource.password	Пароль для доступа к источнику данных.
spring.jpa.properties.hibernate.dialect	Диалект языка SQL ORM-библиотеки Hibernate. Для PostgreSQL должен быть: org.hibernate.dialect.PostgreSQLDialect.
spring.jpa.hibernate.ddl-auto	Параметр инициализация базы данных с помощью библиотеки Hibernate. Необходимо установить значение validate.

Имя	Описание
server.servlet.context-path	Имя контекста в URL-адресе для доступа к API сервера.
server.port	Номер порта сервера.
logging.file.path	Путь к каталогу журнала событий.
ru.signalcom.auth.key.api.allowed-origin-patterns	<p>Шаблон для разрешенных источников запросов CORS. Примеры:</p> <p><a href="https://*.domain1.com">https://*.domain1.com</a> - домены, заканчивающиеся на domain1.com;</p> <p><a href="https://*.domain1.com:8080">https://*.domain1.com:8080</a> - домены, заканчивающиеся на domain1.com на порту 8080 или порту 8081;</p> <p><a href="https://*.domain1.com">https://*.domain1.com:[*]</a> - домены, заканчивающиеся на domain1.com на любом порту, включая порт по умолчанию;</p> <p><a href="https://a1.com">https://a1.com</a>,<a href="https://a2.com">https://a2.com</a> - список шаблонов, разделенных запятыми;</p> <p>* - разрешено всем.</p>
logging.level.ru.signalcom.auth.key.api	Уровень ведения журнала событий: ERROR, WARN, INFO, DEBUG, или TRACE.
server.ssl.enabled	Используется ли протокол TLS для доступа к серверу. Для штатного использования сервера значение данного параметра должно быть true.
server.ssl.key-store	Путь к файлу хранилища ключей сервера.
server.ssl.key-store-password	Пароль к хранилищу ключей сервера.
server.ssl.key-alias	Псевдоним ключа в хранилище ключей сервера, используемого для аутентификации сервера.
server.ssl.key-password	Пароль к ключу, используемому для аутентификации сервера.
server.ssl.client-auth	<p>Режим аутентификация клиента, используемый в протоколе TLS (см. п. 3.5):</p> <p>need - при использовании метода аутентификации клиента по сертификату X.509;</p> <p>none - при использовании метода аутентификации клиента basic.</p>
server.ssl.trust-store	<p>Путь к файлу хранилища доверенных сертификатов.</p> <p>Хранилище доверенных сертификатов используется в случае, если задан режим аутентификации клиента по сертификату (см. параметр server.ssl.client-auth).</p>
server.ssl.trust-store-password	Пароль к хранилищу доверенных сертификатов.
ru.signalcom.auth.key.api.client-subject	Имя владельца в сертификате клиента в формате RFC 2253.
ru.signalcom.auth.key.api.client-name	Имя клиента при использовании basic-аутентификации.



Имя	Описание
ru.signalcom.auth.key.api.client-password	Пароль клиента при использовании basic-аутентификации.
ru.signalcom.auth.key.api.login-attempts	Максимальное количество неудачных попыток логина (по умолчанию 10).
ru.signalcom.auth.key.api.blocking-duration	Время блокирования логина после достижения порога неудачных попыток в секундах (по умолчанию 30).
ru.signalcom.auth.key.api.remote-addr-header	Заголовок для извлечения локального адреса клиента, например, «X-Forwarded-For» (по умолчанию не используется).
ru.signalcom.auth.key.api.trusted-addr-index	Индекс доверенного адреса при использовании параметра ru.signalcom.auth.key.api.remote-addr-header (по умолчанию 1).
ru.signalcom.auth.key.api.reset-attempts	Сбрасывать ли количество неудачных попыток логина по истечении заданного времени (по умолчанию true).
ru.signalcom.auth.key.api.attempts-period	Период хранения количества неудачных попыток в секундах при использовании параметра ru.signalcom.auth.key.api.reset-attempts (по умолчанию 24 часа).
ru.signalcom.auth.key.api.client-number	Максимальное количество клиентов.
ru.signalcom.auth.key.api.pse-path	Путь к каталогу ключевого контейнера СКЗИ для инициализации ДСЧ (см. п. 3.6).
springdoc.api-docs.enabled	Разрешить ли использование конечной точки OpenAPI (по умолчанию /v3/api-docs). По умолчанию – true.
springdoc.swagger-ui.enabled	Разрешить ли использование конечной точки Swagger UI (по умолчанию /swagger-ui.html). По умолчанию – true.
spring.profiles.include	Расширение файла конфигурации application.properties для настройки Flyway.  Для новой базы данных установить значение new-db, для уже имеющейся existing-db.

### 3.3. Настройка базы данных

Для настройки базы данных «Signal-COM AuthKey Server» необходимо выполнить следующие действия:

- создать пользователя-владельца новой базы данных PostgreSQL;
- создать новую базу с указанием владельца;
- задать параметры базы данных в файле конфигурации (см. Таблица 1).

Пример:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/base_name
spring.datasource.username=some-name
spring.datasource.password=some-password
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=validate
```

### 3.4. Настройка TLS

При взаимодействии с «Signal-COM AuthKey Server» требуется обязательное использование протокола TLS и аутентификация сервера по сертификату X.509.

---

Для создания ключей сервера и хранилища ключей необходимо использовать стороннее ПО, например:

- приложение keytool из состава JRE/JDK;
- приложение Admin-PKI [4].

Для настройки протокола TLS на сервере необходимо задать в файле конфигурации следующие параметры:

- server.ssl.enabled
- server.ssl.key-store
- server.ssl.key-store-password
- server.ssl.key-alias
- server.ssl.key-password

Пример:

```
server.ssl.enabled=true
server.ssl.key-store=/opt/auth-key-api/keys/keystore.jks
server.ssl.key-store-password=some-password
server.ssl.key-alias=some-alias
server.ssl.key-password=one-more-password
```

### 3.5. Методы аутентификации клиента

В «Signal-COM AuthKey Server» может использоваться один из следующих методов аутентификации клиента:

- по сертификату X.509;
- basic-аутентификация.

Для настройки метода аутентификации клиента по сертификату требуется дополнительно задать в файле конфигурации следующие параметры:

- server.ssl.client-auth
- server.ssl.trust-store
- server.ssl.trust-store-password
- ru.signalcom.auth.key.api.client-subject

Пример:

```
server.ssl.client-auth=need
server.ssl.trust-store=/opt/auth-key-api/keys/truststore.jks
server.ssl.trust-store-password=some-password
ru.signalcom.auth.key.api.client-subject=CN=authkey,C=RU
```

Для настройки метода аутентификации клиента basic необходимо задать в файле конфигурации следующие параметры:

- server.ssl.client-auth
- ru.signalcom.auth.key.api.client-name
- ru.signalcom.auth.key.api.client-password

Пример:

```
server.ssl.client-auth=none
ru.signalcom.auth.key.api.client-name=some-name
ru.signalcom.auth.key.api.client-password=some-password
```

### 3.6. Датчик случайных чисел

Для генерации ключей в «Signal-COM AuthKey Server» используется криптографически стойкий ДСЧ из состава СКЗИ.

Инициализация ДСЧ производится при запуске приложения от вектора состояния, хранящегося в ключевом контейнере формата СКЗИ.

Для успешной инициализации ДСЧ, перед первым запуском сервера необходимо выполнить следующие действия:

- 
- создать ключевой контейнер СКЗИ с помощью ПО для генерации ключей, например, приложения Admin-PKI [4];
  - задать путь к каталогу этого ключевого контейнера в файле конфигурации сервера (см. п. 3.2).

Пример:

```
ru.signalcom.auth.key.api.pse-path=/opt/auth-key-api/pse
```

### 3.7. Настройка документации

По умолчанию «Signal-COM AuthKey Server» предоставляет конечные точки для протокола OpenAPI и графического интерфейса Swagger.

Для запрета конечной точки графического интерфейса Swagger используется параметр `springdoc.swagger-ui.enabled`:

Пример:

```
springdoc.swagger-ui.enabled=false
```

Для запрета конечной точки протокола OpenAPI используется параметр `springdoc.api-docs.enabled`:

Пример:

```
springdoc.api-docs.enabled=false
```

### 3.8. Настройка Spring Boot Actuator

«Signal-COM AuthKey Server» предоставляет встроенное средство контроля работоспособности сервера Spring Boot Actuator.

По умолчанию по протоколам HTTP и JMX предоставляется доступ только к одной точке `health (/actuator/health)`.

Для предоставления доступа к дополнительным точкам необходимо задавать соответствующие параметры в файле конфигурации (см. п. 3.2).

Пример:

```
management.endpoints.web.exposure.include=info  
management.endpoints.jmx.exposure.include=beans
```

Для получения дополнительной информации смотрите официальную документацию Spring Boot Actuator.

### 3.9. Запуск программы

Запуск программы «Signal-COM AuthKey Server» осуществляется из командной строки, пример:

```
$JAVA -jar /opt/auth-key-api/auth-key-api-1.0.0.jar --  
spring.config.location=file:/opt/auth-key-api/
```

где:

`$JAVA` – полный путь к исполняемому файлу java;

`--spring.config.location` – каталог, где размещается файл конфигурации `application.properties` (слэш в конце обязателен).

Для запуска программы «Signal-COM AuthKey Server» в качестве системной службы используйте соответствующие документированные возможности и рекомендации операционной системы, на которой развернута программа.

### 3.10. Требования по безопасности

Каналы взаимодействия «Signal-COM AuthKey Server» со смежными системами должны быть защищены от активного несанкционированного вторжения, включая атаку MITM.

Доступ к «Signal-COM AuthKey Server» должен быть ограничен (средствами сервера приложений, использованием VPN и т. д.).

---

Защите от несанкционированного изменения подлежит файл конфигурации «Signal-COM AuthKey Server» (см. п. 3.2).

---

#### **4. ПРОВЕРКА ПРОГРАММЫ**

Проверка работоспособности «Signal-COM AuthKey Server» осуществляется с помощью конечных точек Spring Boot Actuator (см. п. 3.8), а также путём анализа записей в журнале событий (см. п. 5).

---

## **5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ**

«Signal-COM AuthKey Server» направляет сообщения в журнал событий, для настройки которого используется файл конфигурации (см. п. 3.2).

---

## ЛИТЕРАТУРА

1. Housley, R., Cryptographic Message Syntax, RFC 5652, September 2009.
2. Федеральный закон № 63-ФЗ от 06.04.2011 «Об электронной подписи».
3. Signal-COM AuthKey Server. Руководство программиста. ШКНР.00072-01 33 01. АО «СИГНАЛ-КОМ», 2024.
4. Admin-PKI v5.3. Руководство пользователя. ШКНР.00057-01 34 01. ЗАО «Сигнал-КОМ», 2019.