|  |  |  |
| --- | --- | --- |
|  |  | утверждаю |
|  |  | Директор научно-производственного республиканского унитарного предприятия «Научно-исследовательский институт  технической защиты информации» |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н.Горбач |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 |

**КОМПЛЕКС ПРОГРАММНЫХ СРЕДСТВ ПРИКЛАДНОЙ СИСТЕМЫ**

**Руководство системного программиста**

**ЛИСТ УТВЕРЖДЕНИЯ**

**BY.БФИД.10246-01 32 01-ЛУ**

|  |  |  |
| --- | --- | --- |
|  |  | Представители  предприятия-разработчика |
|  |  | Заместитель начальника отдела – начальник сектора |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.И.Шуманский |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 |
|  |  | Инженер-программист |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.С.Гацко |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 |
|  |  |  |
|  |  | Инженер 1 категории |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Е.Варюшина |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 |
|  |  |  |
|  |  | Нормоконтролер |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.И.Арнатович |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 |
|  |  |  |

2020

|  |
| --- |
| Утвержден |
| BY.БФИД.10246-01 32 01-ЛУ |

**КОМПЛЕКС ПРОГРАММНЫХ СРЕДСТВ ПРИКЛАДНОЙ СИСТЕМЫ**

**Руководство системного программиста**

**BY.БФИД.10246-01 32 01**

**Листов 67**

2020

**Аннотация**

Настоящий документ предоставляет сведения для проверки, обеспечения функционирования и настройки комплекса программных средств прикладной системы (далее – КПСИС) на условия конкретного применения и содержит следующие разделы:

– общие сведения о программе;

– структура программы;

– настройка программы;

– программный интерфейс сервисов;

– проверка программы;

– сообщения системному программисту;

– перечень сокращений.

**СОДЕРЖАНИЕ**

[1. Общие сведения о программе 4](#_Toc77940480)

[2. Структура программы 6](#_Toc77940481)

[3. Настройка программы 7](#_Toc77940482)

[3.1. Установка и настройка модуля терминала; сервиса выработки и проверки ЭЦП; сервиса предварительного шифрования; сервиса генерации ПСЧП; сервиса контроля целостности; сервиса генерации личного ключа и выпуска запроса на СОК 7](#_Toc77940483)

[3.2. Порядок получения личного ключа и СОК 8](#_Toc77940484)

[3.3. Установка TLS-сервера, функционирующего под управлением ОС семейства Linux 10](#_Toc77940485)

[3.4. Настройка TLS-сервера, функционирующего под управлением ОС семейства Linux 10](#_Toc77940486)

[3.5. Регистрация ПС в ЕС ИФЮЛ, получение client\_id 10](#_Toc77940487)

[3.6. Настройка сервиса предварительного шифрования и сервиса выработки ЭЦП 11](#_Toc77940488)

[3.7. Настройка сервиса проверки ЭЦП 12](#_Toc77940489)

[3.8. Порядок запуска терминала на СИ 12](#_Toc77940490)

[3.9. Выполнение аутентификации с помощью модуля поддержки OIDC 12](#_Toc77940491)

[3.10. Перечень доступных к запросу атрибутов конечного пользователя 15](#_Toc77940492)

[3.11. Организационные меры по защите частичных секретов 15](#_Toc77940493)

[3.12. Настройка среды 16](#_Toc77940494)

[3.13 Сервисы доступные администратору 16](#_Toc77940495)

[4. Программный интерфейс сервисов 18](#_Toc77940496)

[4.1. Описание входных и выходных параметров модуля поддержки OIDC 18](#_Toc77940497)

[4.2. Программный интерфейс сервисов 23](#_Toc77940498)

[4.3. Состав конфигурационных файлов сервисов 35](#_Toc77940499)

[4.4. Взаимодействие с терминалом 49](#_Toc77940519)

[5. Проверка программы 61](#_Toc77940520)

[6. Сообщения системному программисту 63](#_Toc77940521)

[Перечень сокращений 66](#_Toc77940522)

1. Общие сведения о программе

1.1. КПСИС предназначен для обеспечения возможности предоставления информационным системам различных государственных органов и иных организаций Республики Беларусь сервиса идентификации/аутентификации посредством Единой системы идентификации физических и юридических лиц (далее – ЕС ИФЮЛ) BY.БФИД.10243-01.

1.2. КПСИС обеспечивает:

– быструю интеграцию информационных систем различных государственных органов и иных организаций Республики Беларусь в Белорусскую интегрированную сервисно-расчетную систему;

– защиту передаваемых персональных данных;

– выработку/проверку электронной цифровой подписи (ЭЦП);

– организацию защищенного соединения;

– сохранение прикладной системой (ПС) билета аутентификации и проверка новых билетов аутентификации, поступивших в течение последних 60 минут.

1.3. КПСИС функционирует на сервере со следующими характеристиками:

– объем накопителя на жестких магнитных дисках «горячей» замены – не менее 4 ТБ;

– объем оперативной памяти с полной буферизацией – не менее 64 ГБ;

– тактовая частота процессора – не менее 2,1 ГГц.

1.4. КПСИС функционирует на сервере со следующими программными средствами:

– операционная система (ОС) – CentOS 7 версии ядра 3.10;

– программа «Docker»;

– программа «Docker-compose»;

– ЕС ИФЮЛ BY.БФИД.10243-01;

– клиентская программа (КП) BY.БФИД.10244-01.

1.5. Для прохождения процедуры аутентификации на персональной электронной вычислительной машине пользователя должны быть установлены программы и устройства в следующих сочетаниях:

* КП, идентификационная карта (КТА), соответствующая требованиям СТБ 34.101.79-2019, универсальный считыватель КТА СУ-1 ТУ BY 100299757.167-2018;
* КП, программа криптопровайдера (NTCrypto БФИД.10186-01 или Avest CSP BIGN РБ.ЮСКИ.12005-02 «AvPKISetup2.exe»), средство ЭЦП;
* КП, средство ЭЦП, соответствующее требованиям СТБ 34.101.79-2019.

1.6. Основные функции, выполняемые администратором КПСИС: установка и настройка КПСИС.

1.7. КПСИС для функционирования в режиме инициализации и в штатном режиме работы поддерживает роль:

– «администраторы» – роль, исполняемая авторизованным пользователем ОС, наделенным правами устанавливать, настраивать и запускать КПСИС.

Полномочные пользователи (администратор) КПСИС должны иметь достаточный уровень знаний и подготовки для корректного использования и обслуживания КПСИС, постоянно повышать и подтверждать свой профессиональный уровень.

Полномочные пользователи (администратор) КПСИС не должны совершать преднамеренных несанкционированных действий, приводящих к нарушению и (или) снижению безопасности.

Администратор должен гарантировать, что в среде функционирования КПСИС обеспечивается постоянный контроль, обнаружение и дезактивация вредоносного ПО.

1.8. При возникновении подозрения на наличие вредоносного ПО (нетипичная работа программных средств, появление графических и звуковых эффектов, искажений данных, пропадание файлов, частое появление сообщений о системных ошибках и т.п.) пользователь (клиент) самостоятельно или вместе с администратором КПСИС должен провести внеочередной контроль на отсутствие вредоносного ПО.

1.9. В случае обнаружения при проведении проверки зараженных вредоносным ПО файлов пользователь (клиент) обязан:

– приостановить работу;

– немедленно поставить в известность о факте обнаружения зараженных вредоносным ПО файлов администратора КПСИС, владельца зараженных файлов;

– совместно с владельцем зараженных вредоносным ПО файлов провести анализ возможности дальнейшего их использования;

– провести лечение или уничтожение зараженных файлов;

– в случае обнаружения нового вредоносного ПО, не поддающегося лечению применяемыми средствами защиты от вредоносного ПО, направить зараженный вредоносным ПО файл на машинном носителе информации поставщику средств защиты от вредоносного ПО для дальнейшей передачи его в организацию, с которой заключен договор на техническую поддержку средств защиты от воздействия вредоносного ПО;

– по факту обнаружения зараженных вредоносным ПО файлов составить докладную записку администратору КПСИС, в которой необходимо указать источник (отправителя, владельца и т. д.) зараженного файла, тип зараженного файла, характер содержащейся в файле информации и выполненные мероприятия по обезвреживанию вредоносного ПО.

2. Структура программы

2.1. КПСИС состоит из следующих компонентов:

– модуль поддержки OpenID Connect (OIDC);

– программный TLS-сервер;

– модуль терминала;

– сервисы выработки и проверки ЭЦП;

– сервис предварительного шифрования;

– сервис генерации псевдослучайной числовой последовательности;

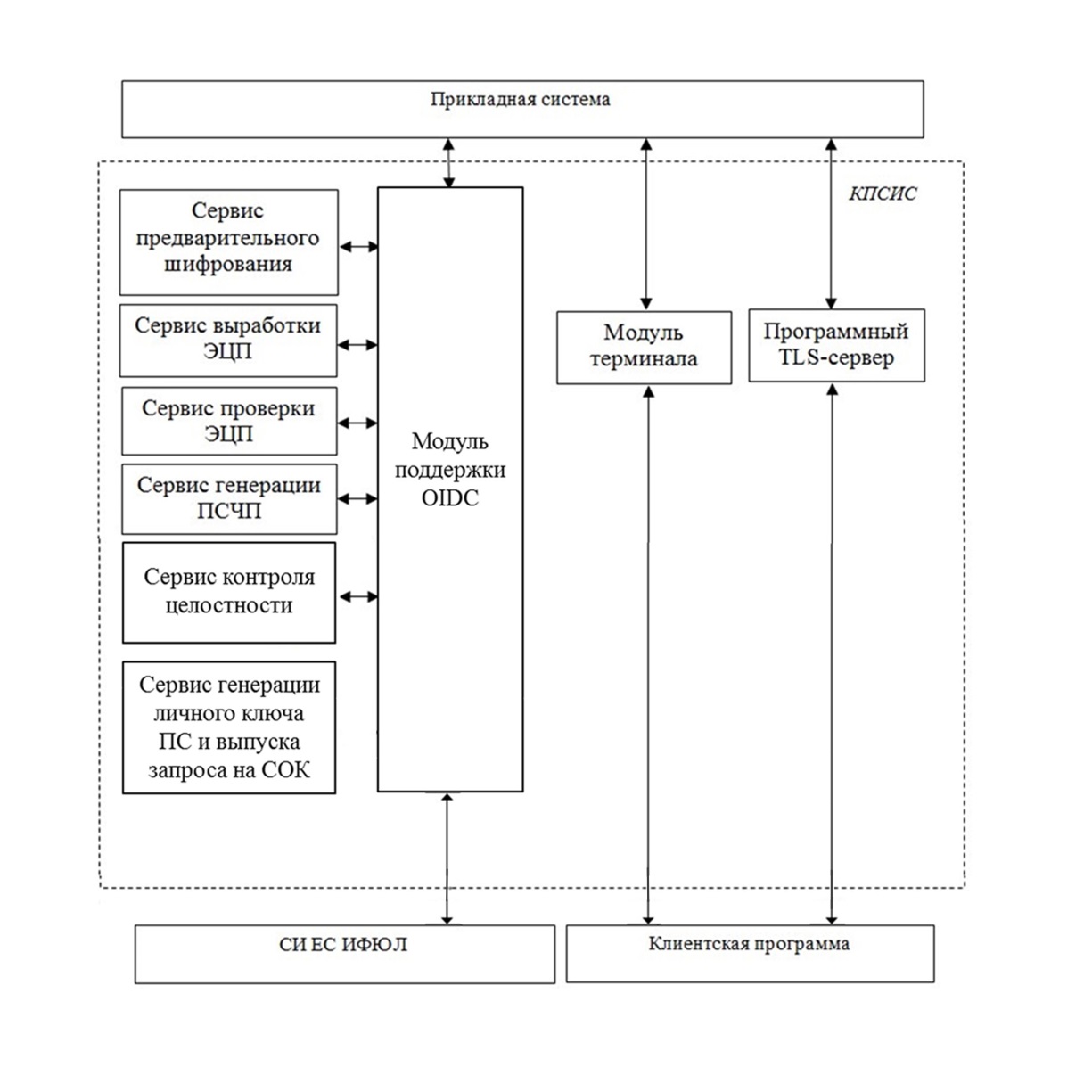
– сервис контроля целостности;

– сервис генерации личного ключа и выпуска запроса на сертификат открытого ключа (СОК).

Модуль терминала, сервисы выработки и проверки ЭЦП, сервис предварительного шифрования, сервис генерации псевдослучайной числовой последовательности, сервис контроля целостности, сервис генерации личного ключа и выпуска запроса на СОК содержатся в Docker-контейнерах. Описание установки и настройки приведено в разделе 3.

Подробное описание о логической структуре и функционировании компонентов приведено в описании программы на КПСИС BY.БФИД.10246-01 13 01.

2.2. Структура КПСИС приведена на рис. 1.



Сервис генерации псевдослучайной числовой последовательности

Рис. 1

3. НАСТРОЙКА ПРОГРАММЫ

3.1. Установка и настройка модуля терминала; сервиса выработки и проверки ЭЦП; сервиса предварительного шифрования; сервиса генерации ПСЧП; сервиса контроля целостности; сервиса генерации личного ключа и выпуска запроса на СОК

3.1.1. Порядок установки и настройки включает в себя следующие действия:

* выполнить установку КПСИС;
* сгенерировать личный ключ КПСИС;
* издать запрос на выпуск СОК КПСИС;
* направить запрос на выпуск СОК КПСИС в республиканское унитарное предприятие «Национальный центр электронных услуг» (НЦЭУ);
* получить СОК КПСИС;
* направить заявку на регистрацию ПС оператору ЕС ИФЮЛ (к заявке необходимо приложить СОК КПСИС);
* зарегистрировать терминал в Центре управления терминалов инфраструктуры открытых ключей облегченных сертификатов, получить enrollpwd;
* получить от оператора ЕС ИФЮЛ: clientID, корневой (https://store.nces.by/esiful/root.crt) и промежуточный (https://store.nces.by/esiful/intermediate.crt) СОК домена «ЕСИФЮЛ», а также СОК сервера идентификации (СИ);
* выполнить настройку компонентов КПСИС (поместить личный ключ и СОК в каталог с сервисами предварительного шифрования и выработки ЭЦП);
* выполнить интеграцию модуля поддержки OIDC в ПС.

3.1.2. Установить на сервер следующие программы:

– «Docker» (<https://docs.docker.com/install/>);

– «Docker-compose» (<https://docs.docker.com/compose/install/>).

Для установки на сервер, функционирующий под управлением ОС семейства Linux, программы «Docker», ввести в терминале команды:

# sudo yum install -y yum-utils \

device-mapper-persistent-data \

lvm2

# sudo yum-config-manager \

--add-repo \

<https://download.docker.com/linux/centos/docker-ce.repo>

# sudo yum install docker-ce docker-ce-cli containerd.io.

Для установки на сервер, функционирующий под управлением ОС семейства Linux, программы «Docker-compose», ввести в терминале команды:

# sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

# sudo chmod +x /usr/local/bin/docker-compose.

3.1.3. Распаковать архив «esiful-kpsis.zip», перейти в каталог с проектом КПСИС, введя в терминале команду:

cd esiful-kpsis,

где esiful-kpsis – наименование каталога с проектом КПСИС. Наименование может быть другим.

3.1.4. Изменить файл «client/config/production.json». В значении ключа «СОК» указать СОК СИ, а в значение «nest\_port» – порт, на котором будет запущен КПСИС. В значении ключа «key» указать промежуточный и корневой сертификат домена «ЕСИФЮЛ».

Пример:

"nest\_port": "3000",

"COK":"<значение СОК в формате BASE64>",

"key": "-----BEGIN CERTIFICATE-----\n<значение промежуточного СОК в формате BASE64>\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----\n<значение корневого СОК в формате BASE64>\n-----END CERTIFICATE-----\n".

3.1.5. Ввести в терминале команды:

# sudo chmod +x install\_docker\_images.sh

# sudo ./install\_docker\_images.sh.

3.1.6. Для запуска Docker-контейнеров ввести в терминале команду:

# docker-compose up.

3.1.7. Для остановки Docker-контейнеров нажать сочетание клавиш:

Ctrl + C.

3.1.8. Для запуска Docker-контейнеров в фоновом режиме ввести в терминале команду:

# docker-compose start.

3.1.9. Для остановки Docker-контейнеров фонового режима ввести в терминале команду:

# docker-compose stop.

3.2. Порядок получения личного ключа и СОК

3.2.1. Для генерации личного ключа необходимо направить http-запрос на сервис генерации личного ключа и выпуска запроса на СОК по адресу сервиса вида «http://host:port/api/v1/gencertreq». Адрес размещения сервиса генерации личного ключа КПСИС и выпуска запроса на СОК указывается при настройке КПСИС. Для получения сведений о том, на каком именно порте запущен сервис, необходимо ввести в терминале команду, находясь в каталоге «./esiful-kpsis»:

# docker-compose ps.

3.2.2. Запрос:

Endpoint: http://host:port/api/v1/gencertreq.

Method: POST.

Content-Type: application/json

Parameters:

Описание параметров запроса приведено в таблице 30.

Состав параметра «cert\_req\_param.cert\_info» для сервиса приведен в таблице 31.

3.2.3. Сервис возвращает следующий ответ:

Content-Type: application/json

Parameters.

Описание параметров ответа приведено в таблице 32.

3.2.4. Пример запроса:

{

"key\_param": {

"security\_level": 128,

"iteration\_count": 10000,

"users\_num": 5,

"treshold\_num": 3,

"passwords": [

"12345678",

"87654321",

"12345678",

"87654321",

"12345678"

]

},

"cert\_req\_param": {

"type": "service\_and\_tls",

"cert\_info": {

"common\_name": "\*.example.org",

"description": "Какой-то сервис",

"organization": "ООО Организация",

"state": "Минский район",

"locality": "Минск",

"street": "ул. Улица, 322, 212212",

"email": "org@example.com",

"subject\_alt\_name": ["\*.example.org", "www.example.org"]

}

},

"save\_to\_file": true

}

3.2.5. Пример ответа:

{ "cert\_request": "<запрос на СОК в формате BASE64>",

"error": "0",

"key": "<личный ключ в ключевом контейнере в формате BASE64>",

"secret\_shards": [

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 1 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 2 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 3 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 4 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 1 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----" ] }

3.2.6. Сформированный запрос на СОК КПСИС (cert\_req.req), личный ключ (key.pem) и частичные секреты (shard\*.pem) будут находиться в каталоге «esiful-kpsis/data/keygensrvc/».

3.2.7. Файл запроса на СОК КПСИС «cert\_req.req» необходимо направить в НЦЭУ для получения СОК.

3.2.8. Полученный СОК КПСИС, личный ключ в ключевом контейнере, зашифрованные частичные секреты и пароли от них необходимо использовать для запуска сервисов предварительного шифрования, проверки ЭЦП.

3.3. Установка TLS-сервера, функционирующего под управлением ОС семейства Linux

Запустить файл «kpsistls\_server\_install-1.0-1.x86\_64.rpm», находясь в каталоге   
«./esiful-kpsis/kpsis-tls-module» на сервере, функционирующем под управлением ОС семейства Linux, введя в терминале команду:

# sudo rpm -ivh kpsistls\_server\_install-1.0-1.x86\_64.rpm.

3.4. Настройка TLS-сервера, функционирующего под управлением ОС семейства Linux

3.4.1. Скорректировать конфигурационный файл «kpsistls.conf», который находится в каталоге «/opt/kpsistls/kpsistls.conf», следующим образом:

– в директиве «ssl\_certificate» указать путь к СОК КПСИС (СОК должен быть в формате PEM в кодировке Base64);

– в директиве «ssl\_certificate\_key» оставить значение «engine:ntctwe:config»;

– в директиве «ssl\_crl» указать путь к актуальным спискам отзыва СОК в формате PEM. Для перевода файла «ruc.crl» в формат PEM применить скрипт   
«./esiful-kpsis/convert\_CRL\_to\_pem.sh», указав в качестве аргумента путь к файлу «ruc.crl»;

– cкопировать файл «./esiful-kpsis/kpsis-tls-module/trusted.pem» в каталог «/opt/kpsistls/certs/»;

– в директиве «ssl\_client\_certificate» указать путь к доверенным корневым СОК;

– в поддирективе «listen» директивы «server» указать порт, который будет прослушивать TLS-сервер;

– в поддирективе «proxy\_pass» директивы «server» указать IP-адрес и порт конечного сервиса.

3.4.2. Скорректировать конфигурационный файл «ntctwe.conf», который находится в каталоге «/opt/kpsistls/sbin/ntctwe.conf», следующим образом:

– в поддирективе «id» директивы «key» указать путь к файлу ключа TLS-сервера;

– в поддирективе «shard» директивы «secret» указать пути к файлам частичного секрета   
TLS-сервера (рис. 2).

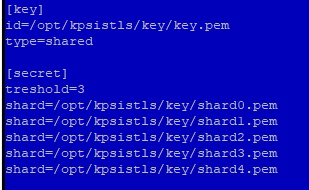


Рис. 2

3.5. Регистрация ПС в ЕС ИФЮЛ, получение client\_id

3.5.1. Заполнить регистрационную форму для регистрации ПС в ЕС ИФЮЛ в соответствии с рис. 3.

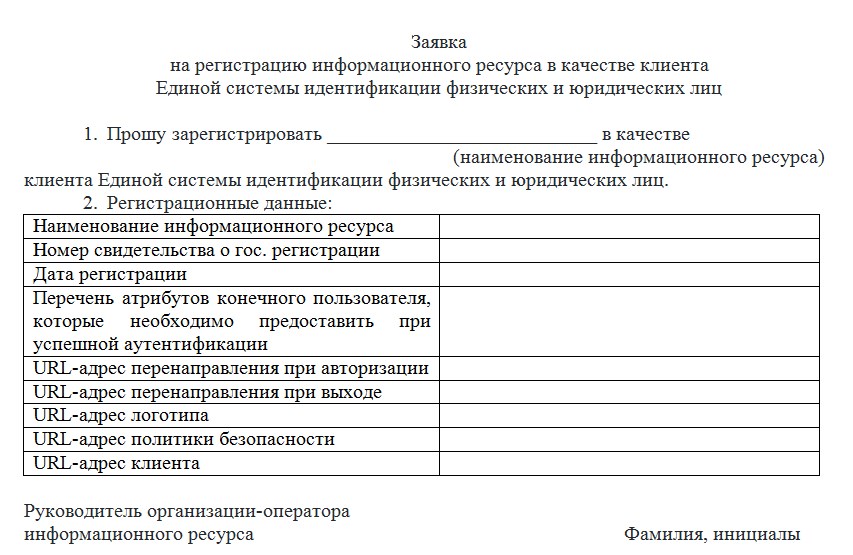


Рис. 3

3.5.2. Направить регистрационную форму оператору ЕС ИФЮЛ любым удобным способом. К регистрационной форме приложить СОК КПСИС.

3.5.3. После завершения процедуры регистрации ПС должна получить от оператора   
ЕС ИФЮЛ client\_id.

3.6. Настройка сервиса предварительного шифрования и сервиса выработки ЭЦП

3.6.1. Поместить СОК КПСИС, полученный в НЦЭУ, в каталог «./esiful-kpsis/data/keys». СОК должен иметь кодировку в формате base64 и имя «cert.pem».

3.6.2. Поместить личный ключ и частичные секреты в каталог «./esiful-kpsis/data/keys/». Ключ должен иметь название «key.pem», частичные секреты должны иметь следующие названия: «shard0.pem», «shard1.pem», «shard2.pem», «shard3.pem», «shard4.pem».

3.6.3. Указать в конфигурационном файле «./esiful-kpsis/docker-compose.yml» для сервисов **esiful-kpsis-signsrvc** и **esiful-kpsis-encrypt** пароли от контейнеров с частичными секретами в одну строку через пробел, сначала пароли от четных частичных секретов, затем пароль от нечетных (ключ command) частичных секретов (рис. 4).

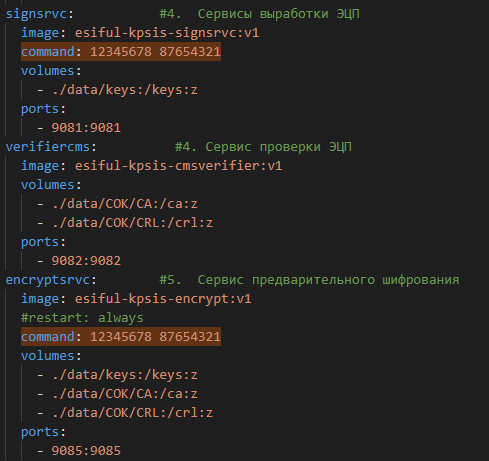


Рис. 4

3.7. Настройка сервиса проверки ЭЦП

3.7.1. Поместить список отозванных сертификатов (СОС) в каталог   
«./esiful-kpsis/data/COK/CRL/».

3.7.2. Поместить корневые СОК в каталог «./esiful-kpsis/data/COK/СА».

3.7.3. Перезапустить КПСИС.

3.8. Порядок запуска терминала на СИ

3.8.1. Направить оператору инфраструктуры открытых ключей облегченных сертификатов заявку для предварительной регистрации терминала. Для выпуска облегченного сертификата получить от оператора пароль: enrollpwd.

3.8.2. Открыть файл «docker-compose.yml», указать значения паролей для частичных секретов: command: password\_1 password\_2 enrollpwd.

3.8.3. Запустить Docker-контейнер с терминалом, для этого в командной строке ввести следующую команду:

docker-compose start.

3.9. Выполнение аутентификации с помощью модуля поддержки OIDC

3.9.1. Порядок прохождения аутентификации включает следующие последовательные шаги:

* создание файла с настройками авторизации;
* формирование и отправка запроса авторизации на КП;
* формирование и отправка запроса авторизации, перенаправление на ЕС ИФЮЛ;
* формирование и отправка запроса для обмена code на билет аутентификации, билет доступа, билет обновления, обмен билета доступа на данные о пользователе («Userinfo»);
* создание защищенного TLS-соединения;
* выход из ЕС ИФЮЛ (завершение сессии).

3.9.2. Создание файла с настройками аутентификации.

Файл в формате JSON, содержащий настройки аутентификации, должен иметь следующий вид:

{ "settings": {

"authority": "https://esiful.nces.by:8081",

"client\_id": "5c0SbSd9IiDIlhmF7q\_V0pq-uGp8z6y8dzWHQIisFIM",

"grant\_type": "authorization\_code",

"response\_type": "code",

"scope" : "openid profile offline\_access",

"post\_logout\_redirect\_uri": "http://localhost:8080",

"redirect\_uri": "http://localhost:8080",

"prompt": "consent",

"state": "niitzi.by" },

"kpsis": "http://192.168.0.165:3000"},

где

* authority (тип данных: строка) – URL-адрес провайдера ЕС ИФЮЛ;
* client\_id (тип данных: строка) – идентификатор клиентского приложения, зарегистрированный у провайдера ЕС ИФЮЛ;
* grant\_type (тип данных: строка, по умолчанию «authorization\_code») – тип потока;
* response\_type (тип данных: строка, по умолчанию «code») – тип ответа, требуемый от провайдера ЕС ИФЮЛ;
* scope (тип данных: строка, по умолчанию «openid») – список данных, запрашиваемых у провайдера ЕС ИФЮЛ. При необходимости ПС также могут запрашивать дополнительные данные: attribute\_certificate (информация об атрибутном сертификате), residence\_place (информация о прописке), director (список субъектов хозяйствования, в которых физическое лицо значится руководителем\ликвидатором);
* redirect\_uri (тип данных: строка) – URI перенаправления клиентского приложения для получения ответа от провайдера ЕС ИФЮЛ;
* post\_logout\_redirect\_uri (тип данных: строка) – URI перенаправления клиентского приложения после выхода из провайдера ЕС ИФЮЛ;
* prompt – обязательный параметр для получения билета обновления. Значение по умолчанию «consent»;
* state – значение, используемое для поддержания состояния между запросом и обратным вызовом, будет возвращено после успешной аутентификации;
* kpsis – ip-адрес КПСИС.

3.9.3. После нажатия на кнопку «Войти в ЕС ИФЮЛ» с Front-end ПС необходимо отправить http-запрос на Backend-часть ПС для начала аутентификации.

Пример запроса на JavaScript:

const { data: { signed\_data\_to\_check\_in\_cp } } = await axios.get("/log-in")

3.9.4. На Backend-части отправить http-запрос на модуль поддержки OIDC КПСИС для формирования ссылки аутентификации.

Пример запроса на JavaScript:

сonst { data: { enveloped\_and\_signed\_auth\_url, code\_verifier, signed\_data\_to\_check\_in\_cp } } = await axios.post(`${config.get('kpsis')}/api/v2/login`, { settings: config.get('settings'), })

После получения параметров ответа от КПСИС, параметр «code\_verifier» необходимо сохранить в cookie, а параметр «enveloped\_and\_signed\_auth\_url» в кэш-значении или базе данных.

Пример на JavaScript сохранения параметра «code\_verifier»:

res.cookie('code\_verifier', code\_verifier, { httpOnly: true });

Пример на JavaScript сохранения параметра «code\_verifier» в коллекции по типу ключ-значение, используя объект «Map», который служит ключом в хранилище:

const sessionStore = new Map()

sessionStore.set(code\_verifier, enveloped\_and\_signed\_auth\_url) // code\_verifier

Далее необходимо отправить ответ на Front-end-часть.

Пример ответа:

res.status(200).json({signed\_data\_to\_check\_in\_cp })

3.9.5. На Front-end части необходимо отправить запрос на КП.

Пример запроса:

const {data } = await axios.post(`http://127.0.0.1:8084/select\_auth`, {

data: signed\_data\_to\_check\_in\_cp

});

После успешного ответа от КП перенаправить пользователя на Backend-часть для перенаправления в ЕС ИФЮЛ.

Пример перенаправления:

const form = document.createElement("form");

form.action = "/redirect-to-esiful";

form.method = "GET";

form.style.display = "none";

document.body.append(form);

form.submit();

Пример обработчика запроса на Backend-части и перенаправления в ЕС ИФЮЛ:

const code\_verifier = req.cookies.code\_verifier

const signed\_data\_to\_check\_in\_cp = sessionStore.get(code\_verifier) // пример получения ссылки на аутентификацию из объекта Map. Создание объекта Map в пункте 3.9.4

sessionStore.delete(code\_verifier) //очистка объекта Map

res.writeHead(302, { 'Location': signed\_data\_to\_check\_in\_cp });

return res.end(); // перенаправление в ЕС ИФЮЛ

3.9.6. После успешной аутентификации в ЕС ИФЮЛ пользователь будет перенаправлен на callback-страницу с query-параметром «data». Необходимо выделить параметр «data» и отправить его на Backend-часть ПС.

Пример http-запроса на JavaScript:

const {

data: { id\_token, userinfo, access\_token, refresh\_token },

} = await axios.get("/login-callback", {

headers: {

params: location.search.split("?data=")[1],

}, });

3.9.7. После получения запроса на Backend-части ПС необходимо отправить запрос на модуль поддержки OIDC для получения билета аутентификации, билета доступа, билета обновления и профайла из 4.1.5.

Пример запроса на JavaScript:

const {params} = req.headers

const { cookies: { code\_verifier } } = req;

const { data: {userinfo, id\_token, access\_token, refresh\_token} } = await axios.post(`${config.get ('kpsis')}/api/v2/login\_callback`, { params, code\_verifier, settings: config.get('settings') })

Если в дальнейшем планируется использовать токены для получения информации от ЕС ИФЮЛ, необходимо их сохранить.

3.9.8. Для того чтобы завершить сессию в ЕС ИФЮЛ, необходимо отправить http-запрос в модуль поддержки OIDC из 4.1.6.

Пример запроса на JavaScript:

const { data: {logout\_url} } = await axios.post(`${config.get('kpsis')}/api/v2/logout`, {

id\_token, authority: config.get('settings').authority,

post\_logout\_redirect\_uri: config.get('settings').post\_logout\_redirect\_uri })

Далее необходимо отправить ответ на Front-end часть ПС со статусом 302 и параметром ‘Location’: logout\_url для перенаправления на СИ ЕС ИФЮЛ.

Пример ответа:

res.writeHead(302, { 'Location': logout\_url });

res.end();

3.9.9. Для установки защищенного TLS-соединения необходимо отправить http-запрос из 4.1.7.

Пример запроса:

const {data: {error}} = await axios.post('http://127.0.0.1:8084/tls\_init', {

"ip\_port": "192.168.0.165:8443" })

Если в ответе от КП параметр «error» будет равен «0», то необходимо перенаправить пользователя по адресу [http://127.0.0.1:8085](http://127.0.0.1:8085/).

Пример:

if (error === "0") { location.href = http://127.0.0.1:8085 }

3.10. Перечень доступных к запросу атрибутов конечного пользователя

Перечень доступных к запросу атрибутов конечного пользователя:

– openid – обязательный параметр для идентификации протокола OpenID Connect;

– profile – данные, составляющие профиль конечного пользователя. Для физического лица profile содержит: идентификационный номер, фамилию, имя, отчество;

– offline\_access – параметр для получения билета обновления (Refresh Token). Билет обновления необходим для получения нового билета доступа (когда срок его действия истечет). Билет доступа (Access Token) необходим для получения данных аутентифицированного пользователя.

3.11. Организационные меры по защите частичных секретов

Защита частичных секретов предполагает реализацию следующих организационных мер:

* частичные секреты размещены на различных устройствах (флеш-карты, съемные носители, разные серверы) таким образом, что на одном устройстве всегда размещено не более 2 частичных секретов;
* путь к частичным секретам указан в конфигурационном файле TLS-сервера и конфигурационных файлах сервиса выработки ЭЦП и сервиса предварительного шифрования.

3.12. Настройка среды

3.12.1. Пароли состоят не менее чем из 8 символов, обязательно содержат буквы в верхнем регистре, буквы в нижнем регистре, цифры и специальные символы (знаки пунктуации, скобки, знаки арифметических операций).

3.12.2. Учетная запись пользователя блокируется после 30 попыток ввода неверного пароля. Количество попыток ввода неверного пароля допускается уменьшить.

3.12.3. Как минимум одному пользователю ОС разрешена установка программ, включая сервис генерации личного ключа и выпуска запроса на СОК.

3.12.4. Рекомендуется предусмотреть защиту от принудительной смены пароля при наличии доступа к ОС (пароль на загрузчик ОС, пароль на BIOS сервера).

3.12.5. Для обеспечения безопасного функционирования КПСИС используются организационные меры – организация контроля доступа к вычислительной платформе (серверу), на которой функционирует КПСИС).

3.12.6. Файлы программ хранятся в специальном каталоге ОС, в Docker-контейнерах. Группе пользователей ОС запрещается изменять эти файлы (за исключением файлов аудита и настроек). Изменение этого каталога возможно при переустановке КПСИС, при этом файл протоколирования не очищается.

3.12.7. Защита сеансовых объектов и аутентификационных данных выполняется ядром ОС. Настройка не требуется.

3.12.8. При начальном запуске каждый модуль КПСИС проводит контроль целостности своих системных объектов.

3.12.9. КПСИС функционирует в автоматическом режиме, в связи с этим все ошибки обрабатываются по HTTP, при поступлении некорректных запросов возвращаются коды ошибок.

3.12.10. Соединение между КПСИС и ПС осуществляется по доверенному каналу.

Примечание – Доверенный канал – средство взаимодействия между функциональными возможностями безопасности объекта оценки и удаленным доверенным продуктом информационных технологий, обеспечивающее необходимую для этого степень уверенности (пункт 3.1.18, СТБ 34.101.1-2014 (ISO/IEC 15408-1:2009)).

3.13 Сервисы доступные администратору

*Вывод номера версии –* отображает номер версии модуля КПСИС в формате «старший номер, младший номер, номер сборки».

*Инициализация –* запускает сервис самотестирования, проверяет наличие конфигурационных файлов модуля КПСИС.

*Инициализация Модуля терминала –* запускает сервис самотестирования, проверяет наличие конфигурационных файлов Терминала, при необходимости генерирует ключевую пару, формирует заявку на облегченный СОК, направляет в Центр Управления Терминалами, который выпускает облегченный СОК и присылает его Терминалу.

*Самотестирование –* выполняет тестирование криптоалгоритмов, проверку целостности программных файлов, тестирование работоспособности ГСЧ.

*Получение личного ключа –* расшифровывает личный ключ, загружает сеансовую копию ключа в оперативную память.

*Шифрование данных –* выполняется проверка действительности СОК получателя, генерируется ключ шифрования данных, генерируется синхропосылка, выполняется зашифрование данных, создается токен ключа шифрования, создание CMS структуры.

*Расшифрование данных –* вычисляется имитовставка, выполняется расшифрование данных.

*Взаимная аутентификация KTA и Терминала –* выполняет первый шаг и второй терминальные шаги протокола BAUTH, формирование общего ключа.

*Защищенное соединение –* устанавливает защищенное соединение Терминал – КТА в соответствии с СТБ 34.101.79-2019 (8.5).

*Защита команды –* выполняет шифрование команд, передаваемые от Терминала на КТА.

*Снятие защиты с ответа –* выполняет расшифрование ответа, полученного от КТА.

*Защищенное соединение с клиентом –* реализует установление защищенного TLS-соединения между КП и КПСИС.

*Выработка ЭЦП –* выполняет выработку ЭЦП.

*Сервис генерации псевдослучайных чисел –* генерирует псевдослучайные числа в режиме HMAC по алгоритму Alg.brng-hmac-hbelt.

*Генерация ключевой пары, запрос на СОК и шифрование –* генерирует ключевую пару (уровень стойкости 128) по алгоритму Alg.bign-genkeypair, формирует запрос на СОК.

*Контроль целостности –* вычисляет хэш-значение для входных данных (если указана кодировка входных данных, то декодирование входных данных с учетом кодировки).

*Проверка ЭЦП –* выполняет проверку ЭЦП.

*Удаление ключей и объектов –* выполняет уничтожение объектов и/или их сеансовых копий.

4. Программный интерфейс сервисов

4.1. Описание входных и выходных параметров модуля поддержки OIDC

4.1.1. Входные и выходные данные модуля поддержки OIDC и КП приведены в таблице 1.

Таблица 1

| HTTP-метод | Метод веб-сервиса | Описание метода веб-сервиса |
| --- | --- | --- |
| POST | /api/v2/login | Формирование ссылки для аутентификации в ЕС ИФЮЛ |
| POST | /select\_auth | Первый шаг для начала протокола аутентификации |
| POST | /api/v2/login\_callback | Получение билета аутентификации, билета доступа, билета обновления и профайла пользователя |
| POST | /api/v2/logout | Выход из ЕС ИФЮЛ |
| POST | /tls\_init | Установка защищенного TLS-соединения |

4.1.2. Входные параметры запроса для метода /api/v2/login приведены в таблице 2.

Таблица 2

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| settings.authority | JSON | string | URL-адрес СИ ЕС ИФЮЛ |
| settings.client\_id | JSON | string | Уникальный идентификатор клиентского приложения, зарегистрированного в СИ ЕС ИФЮЛ |
| settings.grant\_type | JSON | string | Тип потока авторизации. Значение по умолчанию «authorization\_code» |
| settings.response\_type | JSON | string | Тип ответа, требуемый от СИ ЕС ИФЮЛ. Значение по умолчанию «code» |
| settings.scope | JSON | string | Список данных, которые запрашиваются у СИ ЕС ИФЮЛ. Значение по умолчанию «opened» |
| settings.redirect\_uri | JSON | string | URI перенаправления клиентского приложения для получения ответа от провайдера ЕС ИФЮЛ |
| settings.post\_logout\_redirect\_uri | JSON | string | URI перенаправления клиентского приложения после выхода из СИ ЕС ИФЮЛ |
| settings.prompt | JSON | string | Обязательный параметр для получения билета обновления. Значение по умолчанию «consent» |
| settings.state | JSON | string | Значение, используемое для поддержания состояния между запросом и обратным вызовом, будет возвращено после успешной аутентификации |

Пример запроса:

POST /api/v2/login

HTTP/1.1

Accept: application/json

Content-Type: application/json

{"settings": {

"authority": "https://esiful.nces.by:8081",

"client\_id": "5c0SbSd9IiDIlhmF7q\_V0pq-uGp8z6y8dzWHQIisFIM",

"grant\_type": "authorization\_code",

"response\_type": "code",

"scope" : "openid profile offline\_access",

"redirect\_uri": "http://localhost:8080",

"post\_logout\_redirect\_uri": "http://localhost:8080",

"prompt": "consent",

"state": "niitzi.by"}}

4.1.3. Выходные параметры ответа для метода /api/v2/login приведены в таблице 3.

Таблица 3

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| enveloped\_and\_signed\_auth\_url | JSON | string | Ccылка для перехода в СИ ЕС ИФЮЛ |
| code\_verifier | JSON | string | Случайно сгенерированная последовательность, которую необходимо сохранить и после успешной аутентификации предъявить в /api/v1/login\_callback |
| signed\_data\_to\_check\_in\_cp | JSON | string | Параметр, необходимый для передачи в КП для протоколов Bauth и Sauth |

Пример ответа:

HTTP/1.1

Content-Type: application/json

{

enveloped\_and\_signed\_auth\_url: 'https://esiful.nces.by:8081/authz?settings=&client\_id=NWMwU2JTZDlJaURJbGhtRjdxX1YwcHEtdUdwOHo2eThkeldIUUlpc0ZJTQ==',

code\_verifier:'Hduyqezr3v99sVuW5GDTn96ExMlHzKT1Y8oWyGPchZ~Yu3URIUR',

signed\_data\_to\_check\_in\_cp: ''

}

4.1.4. Входные и выходные параметры метода /select\_auth приведены соответственно   
в таблицах 4, 5.

Вызов метода: http://127.0.0.1:8084/select\_auth.

Таблица 4

| Наименование | Тип данных | Наличие обязательно | Описание |
| --- | --- | --- | --- |
| Data | Base64 | Да | Параметр signed\_data\_to\_check\_in\_cp из 4.1.3 |

Таблица 5

| Наименование | Тип данных | Описание |
| --- | --- | --- |
| step 0 | string | Параметр, что проверка ЭЦП, сохранение Scope\_ПС, выделения и сохранения базовой ЭЦП прошла успешно |

Пример запроса:

POST /select\_auth HTTP/1.1

Host: 127.0.0.1:8084

Content-Type: application/json

{

"data":"<Signed\_ПС(Scope\_ПС, Enveloped\_СИ(Req\_ПС))>"

}

Ответ успешный:

Код HTTP 200 OK

{

"step 0": "OK"

}

4.1.5. Входные параметры запроса для метода /api/v2/login\_callback приведены   
в таблице 6.

Таблица 6

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| params | JSON | string | Query параметр data, который был получен в URL-строке после успешной аутентификации |
| code\_verifier | JSON | string | Описание параметра приведено  в таблице 3 |
| settings | JSON | object | Описание настроек приведено в таблице 2 |

Пример запроса:

POST /api/v2/login\_callback HTTP/1.1

Host: 192.168.0.165:3000

Content-Type: application/json

{"params": "<парметры авторизации>",

"code\_verifier": "drVg7tl0WinWXSZAtfpaBzymMCeCTT7Nn67rAXiwsqSpI6ZG0wA",

"settings": {

"authority": "https://esiful.nces.by:8081",

"client\_id": "5c0SbSd9IiDIlhmF7q\_V0pq-uGp8z6y8dzWHQIisFIM",

"grant\_type": "authorization\_code",

"response\_type": "code",

"scope": "openid profile offline\_access",

"redirect\_uri": "http://localhost:8080",

"post\_logout\_redirect\_uri": "http://localhost:8080",

"prompt": "consent",

"state": "niitzi.by" }}

Выходные параметры ответа для метода /api/v2/callback приведены в таблице 7.

Таблица 7

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| Userinfo | JSON | string | Профайл пользователя |
| id\_token | JSON | string | Билет аутентификации |
| access\_token | JSON | string | Билет доступа |
| refresh\_token | JSON | string | Билет обновления |

Пример ответа:

HTTP/1.1

Content-Type: application/json

{id\_token: '<значение билета аутентификации в формате BASE64',

userinfo: {

sub: 'MTAxMDEwMU0xMDFQQjE=',

name: 'Валерий Иванович',

surname: 'Иванов',

serialNumber: '1010101M101PB1' },

access\_token: '<значение билета доступа в формате BASE64>',

refresh\_token: '<значение билета обновления в формате BASE64' }

4.1.6. Входные параметры запроса для метода /api/v2/logout приведены в таблице 8.

Таблица 8

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| id\_token | JSON | string | Билет аутентификации |
| authority | JSON | string | URL-адрес СИ ЕС ИФЮЛ |
| post\_logout\_redirect\_uri | JSON | object | URI перенаправления клиентского приложения после выхода из СИ ЕС ИФЮЛ |

Пример запроса /api/v2/logout:

HTTP/1.1

Content-Type: application/json

{"id\_token": "<значение билета аутентификации в формате BASE64",

"authority": "https://esiful.nces.by:8081",

"post\_logout\_redirect\_uri": "http://localhost:8080" }

Выходные параметры ответа для метода /api/v2/logout приведены в таблице 9.

Таблица 9

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| logout\_url | JSON | string | Ccылка, которая позволяет завершить сеанс СИ ЕС ИФЮЛ |

Пример ответа:

HTTP/1.1

Content-Type: application/json

{"logout\_url": "https://esiful.nces.by:8081/session/end?id\_token\_hint= <значение билета аутентификации в кодировке BASE64> &post\_logout\_redirect\_uri=http://localhost:8080"}

4.1.7. Входные параметры запроса для метода /tls-init приведены в таблице 10.

Таблица 10

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| Ip\_port | JSON | string | IP-адрес и порт TLS-сервера |

Пример запроса:

POST /tls\_init HTTP/1.1

Host: 127.0.0.1:8084

Content-Type: application/json

{ "ip\_port": "192.168.0.165:8443"}

Выходные параметры ответа для метода /api/v2/logout приведены в таблице 11.

Таблица 11

| Наименование | Тип параметра | Тип данных | Описание |
| --- | --- | --- | --- |
| error | JSON | string | Сообщение об ошибке. По умолчанию 0, если ошибок нет |

4.2. Программный интерфейс сервисов

4.2.1. Сервис контроля целостности.

4.2.1.1. Вычисление хэш-значения.

Запрос:

\*\*Адрес\*\*: \*http://localhost:port/api/v1/hash\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 12.

Таблица 12

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| Data | string (Base64) | Данные для вычисления хэш-значения |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 13.

Таблица 13

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| error | string | Код ошибки |
| hash | string (Base64) | Хэш-значение |

Пример запроса:

{

"data": "sZS6yAoI9Ts2bQCOWEpd5IUE+p0btsesJS5ywgL9zg1b49YSF7lhgf5nhq1xa4kL"

}

```

Пример ответа:

{

"error": "0",

"hash": "nQLuRG+2op/lyYLUsTr50+kIYbxM7yfPMGv7CxdKFUo="

}

```

4.2.1.2. Вычисление хэш-значения с дополнительными параметрами.

Запрос:

\*\*Адрес\*\*: \*http://localhost:port/api/v1/hashext\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 14.

Таблица 14

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| data | string (Base64 или Base64URL) | Данные для вычисления хэш-значения |
| input\_encoding | string | Кодировка входных данных:  standart – Base64;  url – Base64URL |
| output\_encoding | string | Кодировка выходных данных:  standart – Base64;  url – Base64URL |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 15.

Таблица 15

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| error | string | Код ошибки |
| hash | string (Base64 или Base64URL) | Хэш-значение |

Пример запроса:

{

"data": "VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4=",

"input\_encoding": "standart",

"output\_encoding": "url"

}

Пример ответа:

{

"error": "0",

"hash": "sCef9CEA6CQmQjdIhCHUfGK3QVVAI6SZWxhw\_U2bv7Q"

}

```

4.2.2. Сервис генерации псевдослучайной числовой последовательности.

4.2.2.1. Выработка случайной последовательности.

Запрос:

\*\*Method\*\*: \*POST\*

\*\*Endpoint\*\*: \*http://host:port/api/v1/genrand\*

\*\*Content-type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 16.

Таблица 16

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| size | integer | Какой длины (в байтах) необходимо сгенерировать последовательность (не больше 4096) |
| range | String | Набор символов, может быть:  full – любой символ от 0х00 до 0хff;  limited – только [a-Z][0-9] "-" "." "\_" "~" |

Ответ:

\*\*Content-type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 17.

Таблица 17

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| Random | String (Base64) | Случайная последовательность |

4.2.3. Сервис проверки ЭЦП.

4.2.3.1. Проверка ЭЦП CMS.

Запрос:

\*\*Method\*\*: \*POST\*

\*\*Endpoint\*\*: \*http://localhost:port/api/v1/verify\*

\*\*Content-type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 18.

Таблица 18

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| Cms | String (Base64) | Значение CMS |
| sampleCert | String (Base64) | Значение эталонного сертификата (опциональный параметр) |

Ответ:

\*\*Content-type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 19.

Таблица 19

| Параметр | Тип | Значение |
| --- | --- | --- |
| cert | String (Base64) | СОК |
| content | String (Base64) | Подписанные данные из CMS |
| certStatus | String | Статус СОК, может быть:  OK – действительный;  INVALID – недействительный;  UNKNOWN – статус не удалось проверить;  WRONG – сертификат не совпадает с эталонным |
| cmsStatus | String | Статус подписи CMS, может быть:  OK – действительная;  INVALID – недействительная;  UNKNOWN – статус не удалось проверить |
| error | String | Код ошибки, 0 – в случае успеха |

4.2.4. Сервис выработки ЭЦП.

4.2.4.1. Выработка ЭЦП и создание CMS-структуры согласно СТБ 34.101.23-2012.

Запрос:

\*\*Method\*\*: \*POST\*

\*\*Endpoint\*\*: \*host:port/api/v1/signcms\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 20.

Таблица 20

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| isDetached | boolean | Признак включения данных в структуру CMS |
| includeCert | boolean | Признак включения СОК структуру CMS |
| Data | string (Base64) | Данные для подписи |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 21.

Таблица 21

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| cms | string (Base64) | Значение подписи |
| error | int | Код ошибки |

Пример запроса:

{

"isDetached": false,

"includeCert": true,

"data": "VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVyIHRoZSBsYXp5IGRvZy4="

}

```

Пример ответа:

```c

{

"cms": "",

"error": "0"

}

```

4.2.4.2. Получение СОК сервиса.

Запрос:

\*\*Method\*\*: \*GET\*

\*\*Endpoint\*\*: \*http://host:port/api/v1/cert\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*: отсутствуют

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 22.

Таблица 22

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| cert | string (Base64) | Значение СОК |
| error | int | Код ошибки |

Пример ответа:

{

"cert": "",

"error": "0"

}

```

4.2.4.3. Получение id ключа сервиса.

Запрос:

\*\*Method\*\*: \*GET\*

\*\*Endpoint\*\*: \*http://host:port/api/v1/keyid\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*: отсутствуют

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 23.

Таблица 23

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| key\_id | string (Base64) | Значение id ключа |
| Error | int | Код ошибки |

Пример ответа:

{

"error": "0",

"key\_id": "irgdFR3ngmv2hFOK6HkAMvgocJw="

}

```

4.2.5. Сервис предварительного шифрования.

4.2.5.1. Зашифрование данных и создание CMS-структуры согласно СТБ 34.101.23-2012.

Запрос:

\*\*Адрес\*\*: \*127.0.0.1:8080/api/v1/encryptcms\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 24.

Таблица 24

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| data | string (Base64) | Данные для зашифрования |
| cert | string (Base64) | СОК |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 25.

Таблица 25

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| cms | string (Base64) | Зашифрованные данные |
| key | string (Base64) | Токен ключа |
| mac | string (Base64) | Имитовставка |
| error | int | Код ошибки |

Пример запроса:

{

"data": "eKBuIl+ZATsLUmMRGh/8DA98DzKELXRGKHGkNtjntIAqpfkxYS8mZLje2dc=",

"cert": "="

}

```

Пример ответа:

{

"cms": "",

"error": "0",

"key": "6KUWDNSuSdGoSlfCV7StaUBO6MorgZOEUx6z8g4v+917PVX2SLdGqSOy2ISES0uSsfNcDMaMExvYk6PQG+byUND6faOlsL+sIPlAUDfxrjI=",

"mac": "DCBXysTB6wU="

}

```

4.2.5.2. Расшифрование данных.

Запрос:

\*\*Адрес\*\*: \*127.0.0.1:8080/api/v1/decryptcms\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 26.

Таблица 26

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| Data | string (Base64) | Зашифрованные данные в формате CMS |
| Key | string (Base64) | Токен ключа |
| Mac | string (Base64) | Имитовставка |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Параметры\*\*:

Параметры приведены в таблице 27.

Таблица 27

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| cms | string (Base64) | Расшифрованные данные |
| error | int | Код ошибки |

Пример запроса:

{ "data": "",

"key": "u3fUCbDpzx+iOGwQ+YNc1n7l0+lV9Vo3bOreKqVEZ45jc0OESaC0hvUgejgSQYaot4drBpQe827ymepxvc3jyGaBixUFhaicltKresA6cMs=",

"mac": "5jiil775Bco="

}

```

Пример ответа:

{

"cms": "kNMqWpryt7u/ndky+VhJwg98DzKELXRGKHGkNtjntIAqpfkxYS8mZLje2dc=",

"error": "0"

}

```

4.2.6. Сервис генерации личного ключа и выпуска запроса на СОК.

4.2.6.1. Сервис генерации ключей с разделенным секретом и запросов на СОК.

Генерация ключа с разделенным секретом.

Запрос:

\*\*Endpoint\*\*: \*http://host:port/api/v1/genkey\*

\*\*Method\*\*: \*POST\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 28.

Таблица 28

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| key\_param | JSON Object | Содержит параметры для генерации ключа, состоит из следующих объектов: key\_param.security\_level, key\_param.iteration\_count, key\_param.users\_num, key\_param.treshold\_num, key\_param.passwords |
| key\_param.security\_level | int | Уровень стойкости. Принимает значение «128» |
| key\_param.iteration\_count | int | Количество итераций для алгоритма построения ключа по паролю PBKDF2 (не меньше 10000) |
| key\_param.users\_num | int | Количество участников для разделения секрета (обязательно 5) |
| key\_param.treshold\_num | int | Минимальное число частичных секретов, необходимых для восстановления секрета (обязательно 3) |
| key\_param.passwords | JSON Array | Массив паролей (пароль должен содержать минимум 8 символов), на которых будут зашифрованы частичные секреты.  Число паролей должно совпадать с числом участников. Выбрать один пароль для четных секретов и один пароль для нечетных секретов. Расположить их в следующем виде: четный, нечетный, четный, нечетный, четный |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 29.

Таблица 29

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| error | String | Код ошибки |
| key | String | Зашифрованный личный ключ в PEM формате |
| secret\_shards | JSON Array of String | Массив частичных секретов в PEM формате |

Пример запроса:

```sh

{

"key\_param": {

"security\_level": 128,

"iteration\_count": 10000,

"users\_num": 5,

"treshold\_num": 3,

"passwords": [

"12345678",

"87654321",

"12345678",

"87654321",

"12345678"

]

}

}

```

Пример ответа:

```sh

{

"error": "0",

"key": "-----BEGIN ENCRYPTED PRIVATE KEY-----MIGdMEgGCSqGSIb3DQEFDTA7MCoGCSqGSIb3DQEFDDAdBAj/tcaD3Q1CjQICJxAwDQYJKnAAAgAiZS8MBQAwDQYJKnAAAgAiZR9JBQAEUaab5vVMXgOzeux+j1snw/pweNSLm+6VbQGufzrMM6LsaAHadaQ8llIsSNUmVlUnUYOUpojoS7fQgRQin08lScCdFAb+ktO/Cz8xT087ckoDJw==-----END ENCRYPTED PRIVATE KEY-----",

"secret\_shards": [

"-----BEGIN ENCRYPTED PRIVATE KEY-----MIGdMEgGCSqGSIb3DQEFDTA7MCoGCSqGSIb3DQEFDDAdBAiuinoFxpfpJgICJxAwDQYJKnAAAgAiZS8MBQAwDQYJKnAAAgAiZR9JBQAEUWc/nUK8W8CGMVqrRyXOd+C6vFh6eDWT3Py0XCM3po8ZwPFI06OUm2KA7jzCfESGRuwYk1l2BKjpsTcz7VKdWMatU4BtvRVA4GAebjiNfNKlgA==-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----MIGdMEgGCSqGSIb3DQEFDTA7MCoGCSqGSIb3DQEFDDAdBAj9KCPWz2rEtAICJxAwDQYJKnAAAgAiZS8MBQAwDQYJKnAAAgAiZR9JBQAEUX2Ik53o6O+jYKlJkf8iBkWh9KK5AN9hWUKQEyQCR077t8aiEwnCka9zQ2SgTvejWL0WFQkZgqinFBkloWKIgcG0tcBfPrZxx1UAzBPwFVxKew==-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----MIGdMEgGCSqGSIb3DQEFDTA7MCoGCSqGSIb3DQEFDDAdBAj3mMu2ru9TlgICJxAwDQYJKnAAAgAiZS8MBQAwDQYJKnAAAgAiZR9JBQAEUTpKN8xt/RxHKGqnHUPh6c3GEGhAwD6TcTXLmPYK1Bfy2xc2TmOeX+GlT4/sE8qD2uYkWxiBZ7nHvBS27FI0JI4C/riBtxXvnNb9Md9uLS0nPA==-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----MIGdMEgGCSqGSIb3DQEFDTA7MCoGCSqGSIb3DQEFDDAdBAi9GFcvx7tsFQICJxAwDQYJKnAAAgAiZS8MBQAwDQYJKnAAAgAiZR9JBQAEUbc05e1bhqmsujSNFT7DjFUWTcoxAUXh8/TmjJb5PD2NIH+qnvJKJwNkuTBHSqcZPwITQw+ONc3JD90VRr6GIUxyl4cPuz1z8l6Rqy08Vj4DSw==-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----MIGdMEgGCSqGSIb3DQEFDTA7MCoGCSqGSIb3DQEFDDAdBAitCfcNJLbzrwICJxAwDQYJKnAAAgAiZS8MBQAwDQYJKnAAAgAiZR9JBQAEUbGmo1ZTltBJHThLrJ4lJ5+7uWctFgNn1OquB8MEkSu5iIQJcDaQ7zPzneRGNMvv2mUOIz0EKql2OfZssLF6lGZV2wLbrTNwXaS7FUC6sE81RQ==-----END ENCRYPTED PRIVATE KEY-----"

]

}

```

Генерация ключа с разделенным секретом и генерация запроса на СОК.

Запрос:

\*\*Endpoint\*\*: \*http://host:port/api/v1/gencertreq\*

\*\*Method\*\*: \*POST\*

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 30.

Таблица 30

| Параметр | Тип параметра | Значение параметра |
| --- | --- | --- |
| key\_param | JSON Object | Содержит параметры для генерации ключа, состоит из следующих объектов: key\_param.security\_level, key\_param.iteration\_count, key\_param.users\_num, key\_param.treshold\_num, key\_param.passwords, cert\_req\_param, cert\_req\_param.type, cert\_req\_param.cert\_info |
| key\_param.security\_level | int | Уровень стойкости. Принимает значение «128» |
| key\_param.iteration\_count | int | Количество итераций для алгоритма построения ключа по паролю PBKDF2 (не меньше 10000) |

*Окончание таблицы 30*

| Параметр | Тип параметра | Значение параметра |
| --- | --- | --- |
| key\_param.users\_num | int | Количество участников для разделения секрета  (обязательно 5) |
| key\_param.treshold\_num | int | Минимальное число частичных секретов, необходимых для восстановления секрета (обязательно 3) |
| key\_param.passwords | JSON Array | Массив паролей (пароль должен содержать минимум 8 символов), на которых будут зашифрованы частичные секреты.  Число паролей должно совпадать с числом участников. Выбрать один пароль для четных секретов и один пароль для нечетных секретов. Расположить их в следующем виде: четный, нечетный, четный, нечетный, четный |
| cert\_req\_param | JSON Object | Содержит параметры для запроса на СОК |
| cert\_req\_param.type | string | service\_and\_tls |
| cert\_req\_param.cert\_info | JSON Object | Информация о субъекте СОК, состав зависит от типа СОК |

Состав параметра «cert\_req\_param.cert\_info» для сервиса приведен в таблице 31.

Таблица 31

| Параметр | Тип параметра | Значение параметра |
| --- | --- | --- |
| cert\_req\_param.cert\_info.organization | string | Название организации |
| cert\_req\_param.cert\_info.common\_name | string | DNS-имя, IP-адрес сервера,  ID-сервера, устройства, процесса |
| cert\_req\_param.cert\_info.description | string | Общее наименование сервера, устройства, процесса |
| cert\_req\_param.cert\_info.state | string | Область нахождения организации-владельца сервера, устройства, процесса |
| cert\_req\_param.cert\_info.locality | string | Населенный пункт нахождения организации-владельца сервера, устройства, процесса |
| cert\_req\_param.cert\_info.street | string | Юридический адрес нахождения организации-владельца сервера, устройства, процесса |
| cert\_req\_param.cert\_info.email | string | Адрес электронной почты организации-владельца сервера, устройства, процесса |
| cert\_req\_param.cert\_info.subject\_alt\_name | array of String | Все DNS-имена TLS-сервера, например, [«www.my.tls.by», «\*.tls.by», «tls.by»] |

Ответ:

\*\*Content-Type\*\*: \*application/json\*

\*\*Parameters\*\*:

Параметры приведены в таблице 32.

Таблица 32

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Значение |
| error | String | Код ошибки |
| key | String | Зашифрованный личный ключ в PEM формате |
| secret\_shards | JSON Array of String | Массив частичных секретов в PEM формате |
| cert\_request | String | Запрос на СОК в Base64 кодировке |

Пример запроса:

{

"key\_param": {

"security\_level": 128,

"iteration\_count": 10000,

"users\_num": 5,

"treshold\_num": 3,

"passwords": [

"12345678",

"87654321",

"12345678",

"87654321",

"12345678"

]

},

"cert\_req\_param": {

"type": "service\_and\_tls",

"cert\_info": {

"common\_name": "\*.example.org",

"description": "Какой-то сервис",

"organization": "ООО Организация",

"state": "Минский район",

"locality": "Минск",

"street": "ул. Улица, 322, 212212",

"email": "org@example.com",

"subject\_alt\_name": ["\*.example.org", "www.example.org"]

}

},

"save\_to\_file": true

}

Пример ответа:

{ "cert\_request": "<запрос на СОК в формате BASE64>",

"error": "0",

"key": "<личный ключ в ключевом контейнере в формате BASE64>",

"secret\_shards": [

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 1 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 2 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 3 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 4 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----",

"-----BEGIN ENCRYPTED PRIVATE KEY-----

<значение частичного секрета 1 в формате BASE64>

-----END ENCRYPTED PRIVATE KEY-----" ] }

4.3. Состав конфигурационных файлов сервисов

4.3.1. Конфигурационный файл модуля поддержки OIDC.

Структура файла «production.json»:

«nest\_port»: «3000»,

«COK»:»<значение СОК в формате BASE64>»,

«key»: «-----BEGIN CERTIFICATE-----\n<значение промежуточного СОК в формате BASE64>\n-----END CERTIFICATE-----\n-----BEGIN CERTIFICATE-----\n<значение корневого СОК в формате BASE64>\n-----END CERTIFICATE-----\n».

Состав и описание компонентов файла «production.json» приведены в таблице 33.

Таблица 33

|  |  |
| --- | --- |
| Элемент | Описание |
| nest\_port | Порт, на котором будет запущен КПСИС |
| COK | Значение СОК СИ в формате BASE64 |
| key | Значение промежуточного СОК в формате BASE64 и значение корневого СОК в формате BASE64 |

4.3.2. Конфигурационные файлы программного TLS-сервера.

4.3.2.1. Файл «kpsistls.conf».

Структура файла «kpsistls.conf» представлена на риc. 5.

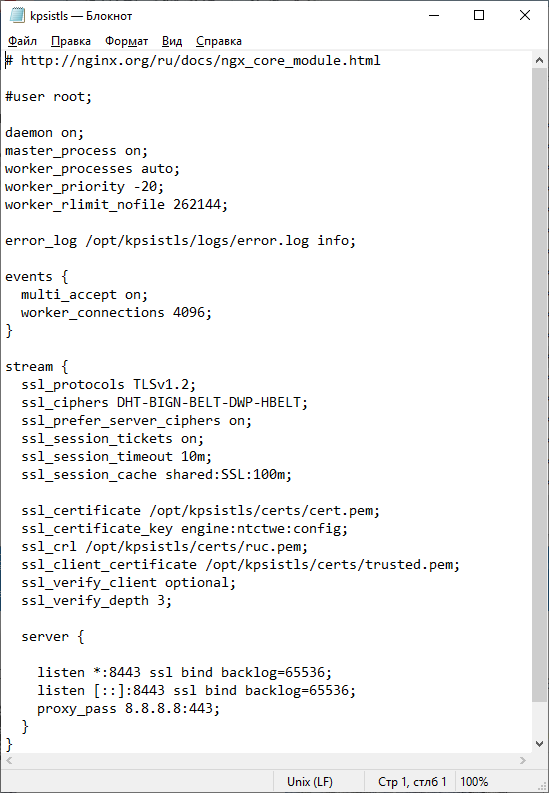


Рис. 5

Состав и описание компонентов файла «kpsistls.conf» приведены в таблице 34.

Таблица 34

| Элемент | Описание |
| --- | --- |
| user root | |
| daemon | Определяет, будет ли запущен режим демона |
| master\_process | Определяет, будут ли запускаться рабочие процессы |
| worker\_processes | Задает число рабочих процессов |
| worker\_priority | Задает приоритет планирования рабочих процессов |
| worker\_rlimit\_nofile | Изменяет ограничение на максимальное число открытых файлов для рабочих процессов |
| error\_log | Путь для записи лог-файлов (/opt/kpsistls/logs/error.log) |
| events | |
| multi\_accept | Если «on» – рабочий процесс за один раз будет принимать только одно новое соединение. В противном случае – все новые соединения |
| worker\_connections | Задает максимальное число соединений, которые одновременно может открыть рабочий процесс |
| stream | |
| ssl\_protocols | Разрешает указанные протоколы |
| ssl\_ciphers | Определяет разрешенные шифры |
| ssl\_prefer\_server\_ciphers | При использовании протоколов SSLv3 и TLS ставит в приоритет серверные шифры перед клиентскими шифрами |
| ssl\_session\_tickets | Разрешает или запрещает возобновление сессий |
| ssl\_session\_timeout | Задает время, в течение которого клиент может повторно использовать параметры сессии |
| ssl\_session\_cache | Задает тип и размеры кэш-значений для хранения параметров сессий |
| ssl\_certificate | Путь к СОК для данного виртуального сервера (/opt/kpsistls/certs/cert.pem) |
| ssl\_certificate\_key | Файл с секретом для данного виртуального сервера |
| ssl\_crl | Указывает файл с отозванными сертификатами (CRL) в формате PEM, используемыми для проверки клиентских СОК (/opt/kpsistls/certs/ruc.pem) |
| ssl\_client\_certificate | Путь к СОК клиента (/opt/kpsistls/certs/trusted.pem) |
| ssl\_verify\_client | Разрешает проверку клиентских СОК |
| ssl\_verify\_depth | Устанавливает глубину проверки в цепочке клиентских СОК |
| server | |
| listen | Задает адрес и порт для IP или путь для UNIX-сокета, на которые сервер будет принимать запросы |
| proxy\_pass | Задает протокол и адрес проксируемого сервера |

4.3.2.2. Файл «ntctwe.conf».

Структура файла «ntctwe.conf» представлена на риc. 6.

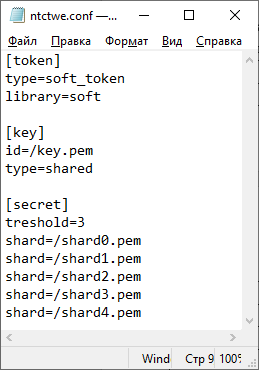


Рис. 6

Состав и описание компонентов файла «ntctwe.conf» приведены в таблице 35.

Таблица 35

|  |  |
| --- | --- |
| Элемент | Описание |
| [token] | |
| type | Тип токена (soft\_token – software token – программный токен) |
| library | Библиотека |
| [key] | |
| id | Идентификатор ключа |
| type | Тип ключа (shared – разделенный секрет) |
| [secret] | |
| treshold | Минимальное количество частичных секретов, необходимых для восстановления секрета |
| shard | Путь к частичному секрету |

4.3.3. Конфигурационный файл модуля терминала.

4.3.3.1. Файл «service.conf».

Структура файла «service.conf» представлена на риc. 7.

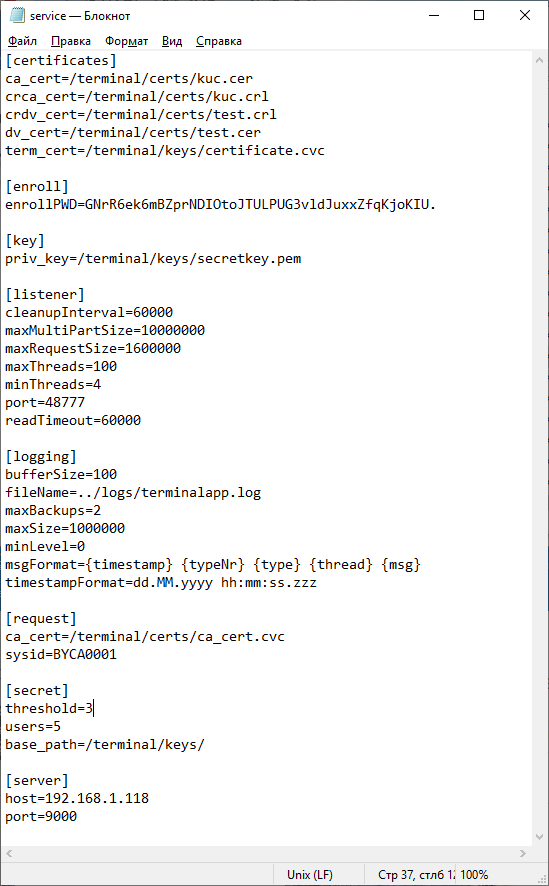


Рис. 7

Состав и описание компонентов файла «service.conf» приведены в таблице 36.

Таблица 36

|  |  |
| --- | --- |
| Элемент | Описание |
| [certificates] | |
| ca\_cert | Корневой СОК (/terminal/certs/kuc.cer) |
| crca\_cert | СОС корневые (/terminal/certs/kuc.crl) |
| crdv\_cert | СОС промежуточные (/terminal/certs/test.crl) |
| dv\_cert | Промежуточный СОК (/terminal/certs/test.cer) |
| term\_cert | СОК терминала (/terminal/keys/certificate.cvc) |
| [enroll] | |
| enrollPWD | Билет процесса enroll |
| [key] | |
| priv\_key | Личный ключ (/terminal/keys/secretkey.pem) |
| [listener] | |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| maxMultiPartSize | Размер принимаемого частями запроса |
| maxRequestSize | Максимальная длина запроса |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |
| port | Порт |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| [logging] | |
| bufferSize | Размер буфера для записи лог-файлов (вначале в буфер, потом в файл) |
| fileName | Путь для записи лог-файлов (../logs/terminalapp.log) |
| maxBackups | Максимально допустимое количество лог-файлов |
| maxSize | Максимальный размер лог-файла, если текущий лог-файл превышает размер, начинается запись в новый лог-файл |
| minLevel | Уровень логирования |
| msgFormat | Формат сообщения ({timestamp} {typeNr} {type} {thread} {msg}) |
| timestampFormat | Формат даты и времени (dd.MM.yyyy hh:mm:ss.zzz, где dd – день в диапазоне от 01 до 31, MM – месяц в диапазоне от 01 до 12, yyyy – год в виде четырехзначного числа, hh – час в 24-часовом формате от 00 до 23, mm – минуты в диапазоне от 00 до 59, ss – секунды в диапазоне от 00 до 59, zzz – указанное со знаком смещение часового пояса локальной ОС от времени UTC, измеренное в часах и минутах) |
| [request] | |
| ca\_cert | Корневой СОК (/terminal/certs/ca\_cert.cvc) |
| sysid | Идентификатор Удостоверяющего центра |

*Окончание таблицы 36*

|  |  |
| --- | --- |
| Элемент | Описание |
| [secret] | |
| threshold | Минимальное число частичных секретов, достаточное для восстановления секрета |
| users | Количество частичных секретов |
| base\_path | Путь к частичным секретам (/terminal/keys/) |
| [server] | |
| host | Адрес сервера |
| port | Порт |

4.3.4. Конфигурационные файлы сервиса выработки ЭЦП.

4.3.4.1. Файл «recover.ini».

Структура файла «recover.ini» представлена на риc. 8.

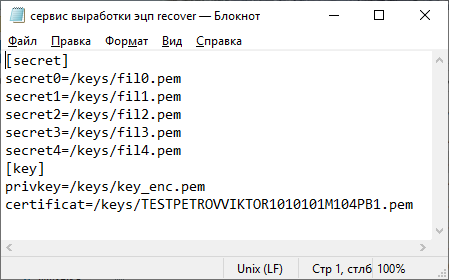


Рис. 8

Состав и описание компонентов файла «recove.ini» приведены в таблице 37.

Таблица 37

|  |  |
| --- | --- |
| Элемент | Описание |
| [secret] | |
| secret | Путь к секрету |
| [key] | |
| privkey | Путь к личному ключу (/keys/key\_enc.pem) |
| certificat | Путь к СОК (/keys/TESTPETROVVIKTOR1010101M104PB1.pem) |

4.3.4.2. Файл «web-service-cms.ini».

Структура файла «web-service-cms.ini» представлена на риc. 9.

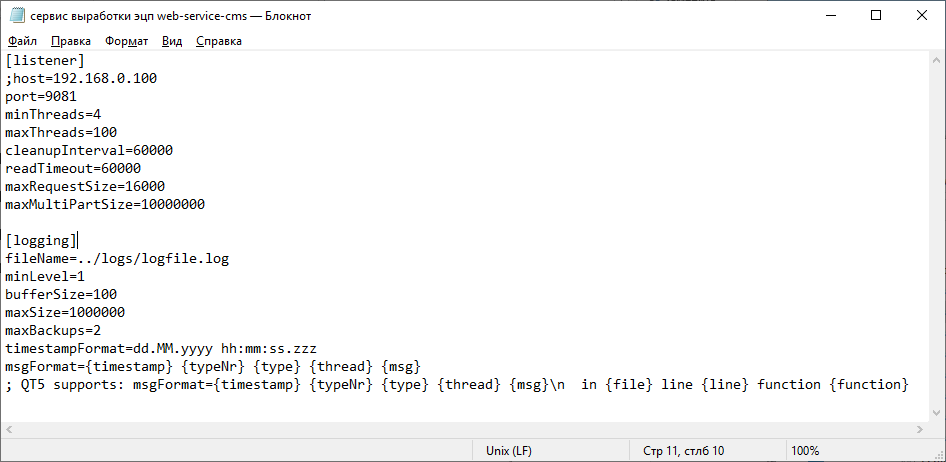


Рис. 9

Состав и описание компонентов файла «web-service-cms.ini» приведены в таблице 38.

Таблица 38

|  |  |
| --- | --- |
| Элемент | Описание |
| [listener] | |
| host | Адрес, на который приходят запросы |
| port | Номер порта |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| maxRequestSize | Максимальная длина запроса |
| maxMultiPartSize | Размер принимаемого частями запроса |
| [logging] | |
| fileName | Путь для записи лог-файлов (../logs/logfile.log) |
| minLevel | Уровень логирования |
| bufferSize | Размер буфера для записи лог-файла (вначале в буфер, потом в файл) |
| maxSize | Максимальный размер лог-файла. Если текущий лог-файл превышает размер, начинается запись в новый лог-файл |
| maxBackups | Максимально допустимое количество лог-файлов |
| timestampFormat | Формат даты и времени (dd.MM.yyyy hh:mm:ss.zzz, где dd – день в диапазоне от 01 до 31, MM – месяц в диапазоне от 01 до 12, yyyy – год в виде четырехзначного числа, hh – час в 24-часовом формате от 00 до 23, mm – минуты в диапазоне от 00 до 59, ss – секунды в диапазоне от 00 до 59, zzz – указанное со знаком смещение часового пояса локальной ОС от времени UTC, измеренное в часах и минутах) |
| msgFormat | Формат сообщения – {timestamp} {typeNr} {type} {thread} {msg}; QT5 supports: msgFormat={timestamp} {typeNr} {type} {thread} {msg}\n in {file} line {line} function {function} |

4.3.5. Конфигурационный файл сервиса проверки ЭЦП.

4.3.5.1. Файл «recover.conf».

Структура файла «recover.conf» представлена на риc. 10.

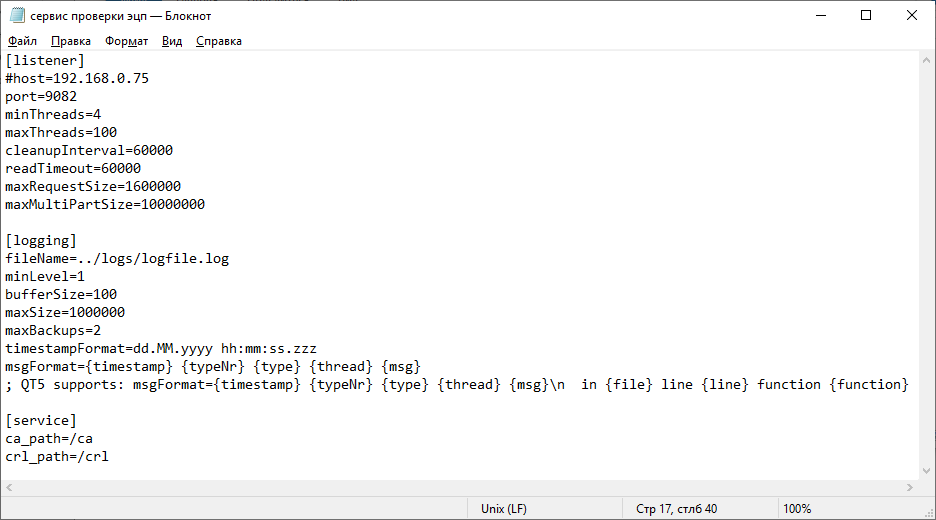


Рис. 10

Состав и описание компонентов файла «recover.conf» приведены в таблице 39.

Таблица 39

|  |  |
| --- | --- |
| Элемент | Описание |
| [listener] | |
| host | Адрес, на который приходят запросы |
| port | Номер порта |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| maxRequestSize | Максимальная длина запроса |
| maxMultiPartSiz | Размер принимаемого частями запроса |
| [logging] | |
| fileName | Путь для записи лог-файлов (../logs/logfile.log) |
| minLevel | Уровень логирования |
| bufferSize | Размер буфера для записи лог-файлов (вначале в буфер, потом в файл) |
| maxSize | Максимальный размер лог-файла. Если текущий лог-файл превышает размер, начинается запись в новый лог-файл |
| maxBackups | Максимально допустимое количество лог-файлов |
| timestampFormat | Формат даты и времени (dd.MM.yyyy hh:mm:ss.zzz, где dd – день в диапазоне от 01 до 31, MM – месяц в диапазоне от 01 до 12, yyyy – год в виде четырехзначного числа, hh – час в 24-часовом формате от 00 до 23, mm – минуты в диапазоне от 00 до 59, ss – секунды в диапазоне от 00 до 59, |

*Окончание таблицы 39*

|  |  |
| --- | --- |
| Элемент | Описание |
|  | zzz – указанное со знаком смещение часового пояса локальной ОС от времени UTC, измеренное в часах и минутах) |
| msgFormat | Формат сообщения – {timestamp} {typeNr} {type} {thread} {msg}; QT5 supports: msgFormat={timestamp} {typeNr} {type} {thread} {msg}\n in {file} line {line} function {function} |
| [service] | |
| ca\_path | Путь к СА-файлам |
| crl\_path | Путь к CRL-файлам |

4.3.6. Конфигурационные файлы сервиса предварительного шифрования.

4.3.6.1. Файл «recover.ini».

Структура файла «recover.ini» представлена на риc. 11.

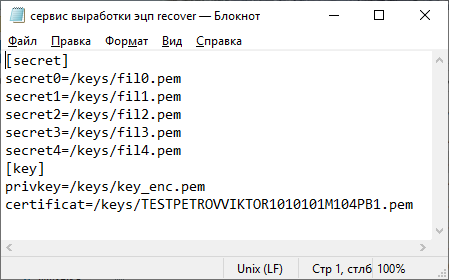


Рис. 11

Состав и описание компонентов файла «recover.ini» приведены в таблице 40.

Таблица 40

|  |  |
| --- | --- |
| Элемент | Описание |
| [secret] | |
| secret | Путь к секрету |
| [key] | |
| privkey | Путь к личному ключу (/keys/key\_enc.pem) |
| certificat | Путь к СОК (/keys/TESTPETROVVIKTOR1010101M104PB1.pem) |

4.3.6.2. Файл «web\_service\_cms\_encrypt.ini».

Структура файла «web\_service\_cms\_encrypt.ini» представлена на риc. 12.

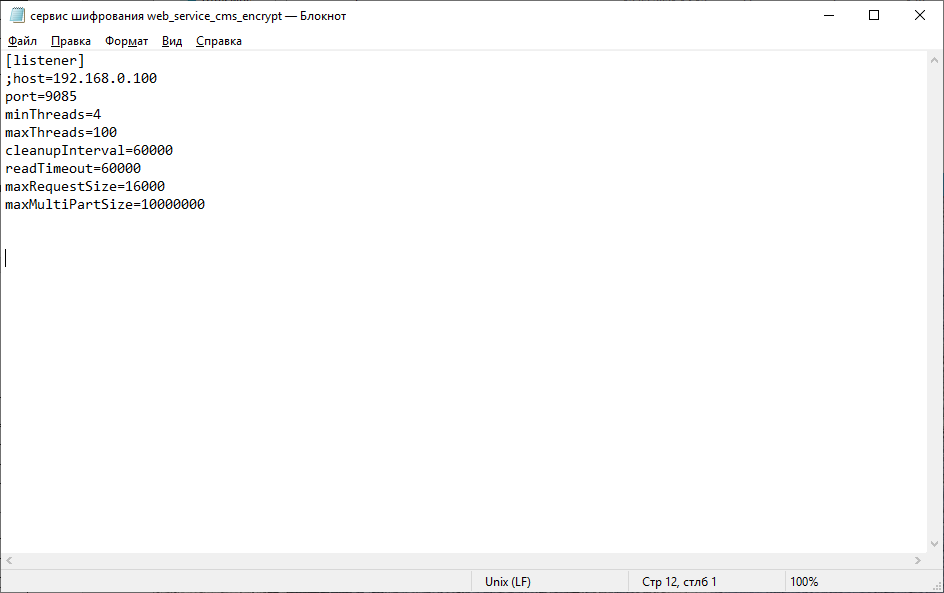


Рис. 12

Состав и описание компонентов файла «web\_service\_cms\_encrypt.ini» приведены   
в таблице 41.

Таблица 41

|  |  |
| --- | --- |
| Элемент | Описание |
| [listener] | |
| host | Адрес, на который приходят запросы |
| port | Номер порта |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| maxRequestSize | Максимальная длина запроса |
| maxMultiPartSiz | Размер принимаемого частями запроса |

4.3.7. Конфигурационный файл сервиса генерации псевдослучайной числовой последовательности.

4.3.7.1. Файл «recover.conf».

Структура файла «recover.conf» представлена на риc. 13.

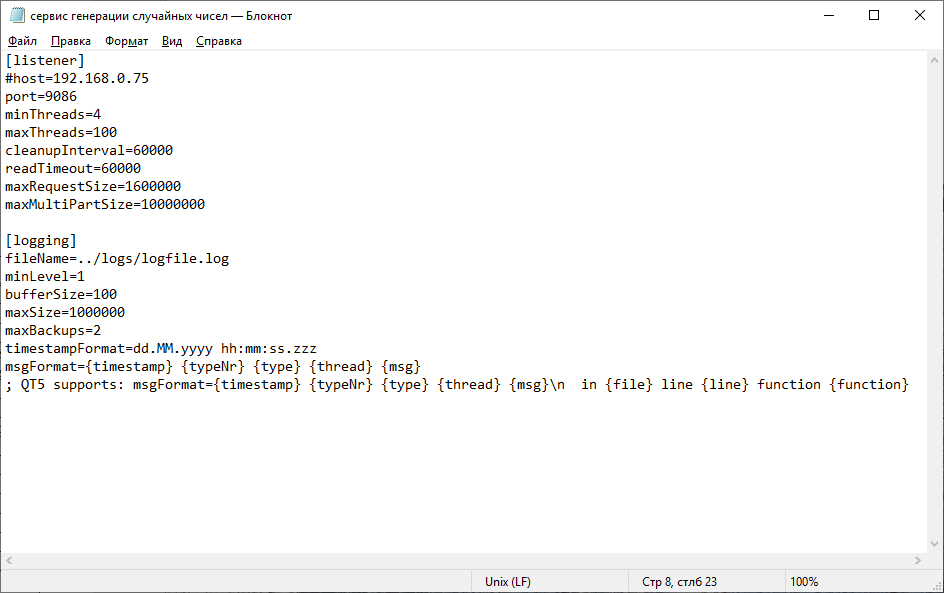


Рис. 13

Состав и описание компонентов файла «recover.conf» приведены в таблице 42.

Таблица 42

|  |  |
| --- | --- |
| Элемент | Описание |
| [listener] | |
| host | Адрес, на который приходят запросы |
| port | Номер порта |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |

*Окончание таблицы 42*

|  |  |
| --- | --- |
| Элемент | Описание |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| maxRequestSize | Максимальная длина запроса |
| maxMultiPartSiz | Размер принимаемого частями запроса |
| [logging] | |
| fileName | Путь для записи лог-файлов (../logs/logfile.log) |
| minLevel | Уровень логирования |
| bufferSize | Размер буфера для записи лог-файлов (вначале в буфер, потом в файл) |
| maxSize | Максимальный размер лог-файла. Если текущий лог-файл превышает размер, начинается запись в новый лог-файл |
| maxBackups | Максимально допустимое количество лог-файлов |
| timestampFormat | Формат даты и времени (dd.MM.yyyy hh:mm:ss.zzz, где dd – день в диапазоне от 01 до 31, MM – месяц в диапазоне от 01 до 12, yyyy – год в виде четырехзначного числа, hh – час в 24-часовом формате от 00 до 23, mm – минуты в диапазоне от 00 до 59, ss – секунды в диапазоне от 00 до 59, zzz – указанное со знаком смещение часового пояса локальной ОС от времени UTC, измеренное в часах и минутах) |
| msgFormat | Формат сообщения – {timestamp} {typeNr} {type} {thread} {msg}; QT5 supports: msgFormat={timestamp} {typeNr} {type} {thread} {msg}\n in {file} line {line} function {function} |

4.3.8. Конфигурационный файл сервиса контроля целостности.

4.3.8.1. Файл «recover.conf».

Структура файла «recover.conf» представлена на риc. 14.

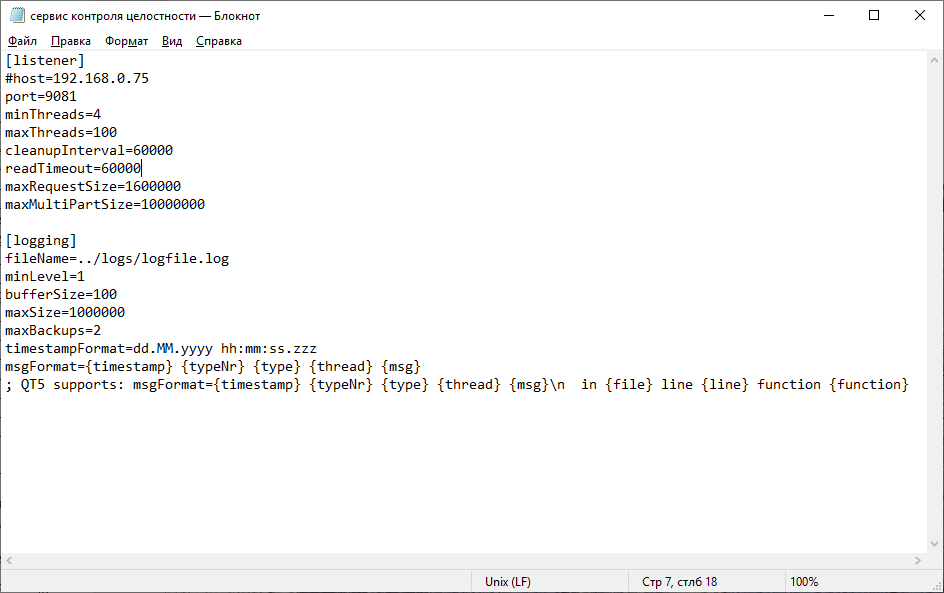


Рис. 14

Состав и описание компонентов файла «recover.conf» приведены в таблице 43.

Таблица 43

|  |  |
| --- | --- |
| Элемент | Описание |
| [listener] | |
| host | Адрес, на который приходят запросы |
| port | Номер порта |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| maxRequestSize | Максимальная длина запроса |
| maxMultiPartSiz | Размер принимаемого частями запроса |
| [logging] | |
| fileName | Путь для записи лог-файлов (../logs/logfile.log) |
| minLevel | Уровень логирования |
| bufferSize | Размер буфера для записи лог-файлов (вначале в буфер, потом в файл) |
| maxSize | Максимальный размер лог-файла. Если текущий лог-файл превышает размер, начинается запись в новый лог-файл |
| maxBackups | Максимально допустимое количество лог-файлов |
| timestampFormat | Формат даты и времени (dd.MM.yyyy hh:mm:ss.zzz, где dd – день в диапазоне от 01 до 31, MM – месяц в диапазоне от 01 до 12, yyyy – год в виде четырехзначного числа, hh – час в 24-часовом формате от 00 до 23, mm – минуты в диапазоне от 00 до 59, ss – секунды в диапазоне от 00 до 59, zzz – указанное со знаком смещение часового пояса локальной ОС от времени UTC, измеренное в часах и минутах) |
| msgFormat | Формат сообщения – {timestamp} {typeNr} {type} {thread} {msg}; QT5 supports: msgFormat={timestamp} {typeNr} {type} {thread} {msg}\n in {file} line {line} function {function} |

4.3.9. Конфигурационный файл сервиса генерации личного ключа и выпуска запроса на СОК.

4.3.9.1. Файл «recover.conf».

Структура файла «recover.conf» представлена на риc. 15.

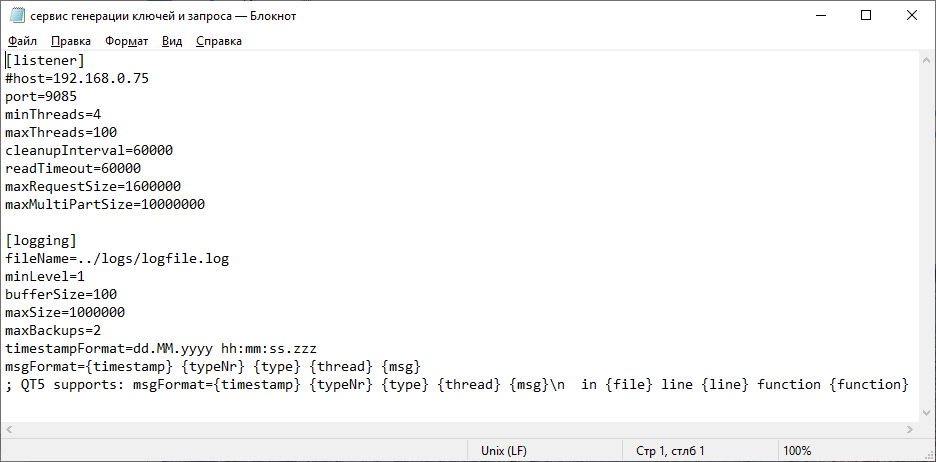


Рис. 15

Состав и описание компонентов файла «recover.conf» приведены в таблице 44.

Таблица 44

|  |  |
| --- | --- |
| Элемент | Описание |
| [listener] | |
| host | Адрес, на который приходят запросы |
| port | Номер порта |
| minThreads | Минимальное количество потоков, поддерживаемое после отключения всех остальных потоков в течение времени cleanupInterval |
| maxThreads | Максимальное количество одновременных рабочих потоков |
| cleanupInterval | Время, по истечении которого неиспользуемые потоки удаляются |
| readTimeout | Время, в течение которого операция чтения блокирует ожидание данных |
| maxRequestSize | Максимальная длина запроса |
| maxMultiPartSiz | Размер принимаемого частями запроса |
| [logging] | |
| fileName | Путь для записи лог-файлов (../logs/logfile.log) |
| minLevel | Уровень логирования |
| bufferSize | Размер буфера для записи лог-файлов (вначале в буфер, потом в файл) |
| maxSize | Максимальный размер лог-файла. Если текущий лог-файл превышает размер, начинается запись в новый лог-файл |
| maxBackups | Максимально допустимое количество лог-файлов |
| timestampFormat | Формат даты и времени (dd.MM.yyyy hh:mm:ss.zzz, где dd – день в диапазоне от 01 до 31, MM – месяц в диапазоне от 01 до 12, yyyy – год в виде четырехзначного числа, hh – час в 24-часовом формате от 00 до 23, mm – минуты в диапазоне от 00 до 59, ss – секунды в диапазоне от 00 до 59, zzz – указанное со знаком смещение часового пояса локальной ОС от времени UTC, измеренное в часах и минутах) |
| msgFormat | Формат сообщения – {timestamp} {typeNr} {type} {thread} {msg}; QT5 supports: msgFormat={timestamp} {typeNr} {type} {thread} {msg}\n in {file} line {line} function {function} |

4.4. Взаимодействие с терминалом

4.4.1. Настройка терминала приведена в 3.8 данного документа.

4.4.2. Интеграция терминала с информационной системой.

4.4.2.1. Выполнение протокола BAUTH. Описание методов терминала и КП приведено   
в таблице 45.

Таблица 45

|  |  |  |
| --- | --- | --- |
| HTTP-метод | Метод  веб-сервиса | Описание метода веб-сервиса |
| Методы терминала | | |
| POST | /api/v1/bauth\_init | Инициализация выполнения протокола BAUTH |
| POST | /api/v1/bauth\_process | Шаги выполнения протокола BAUTH |
| POST | /api/v1/bauth\_logout | Завершение выполнения протокола BAUTH |
| Методы КП | | |
| POST | /api/v1/terminal\_proxy\_bauth\_prefetch | Подготовительная команда |
| POST | /api/v1/terminal\_proxy\_bauth\_init | Инициализация выполнения протокола BAUTH |
| POST | /api/v1/terminal\_proxy\_bauth | Шаги выполнения протокола BAUTH |

4.4.2.2. Входные параметры метода «/api/v1/bauth\_init» (инициализация выполнения протокола BAUTH; терминал) приведены в таблице 46.

Таблица 46

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| so\_certificate | Сертификат объекта безопасности идентификационной карты (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.3. Выходные параметры метода «/api/v1/bauth\_init» (инициализация выполнения протокола BAUTH; терминал) приведены в таблице 47.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 47

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных |
| err | Код ошибки | JSON | String |
| hreq | Уникальный хэш-идентификатор. Требуется его запомнить и отсылать на терминал вместе с каждым последующим вызовом в сессии | JSON | Base64 String |

*Окончание таблицы 47*

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных |
| terminal\_certificate | Сертификат терминала | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | Base64 String |

4.4.2.4. Входные параметры метода «/api/v1/bauth\_process» (шаги выполнения протокола BAUTH; терминал) приведены в таблице 48.

Таблица 48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |
| card\_response | Ответ от КП | JSON | BASE64\_STRING | Да |

4.4.2.5. Выходные параметры метода «/api/v1/bauth\_process» (шаги выполнения протокола BAUTH; терминал) приведены в таблице 49.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 49

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| is\_bauth\_established | Показывает, установлен ли протокол BAUTH | JSON | Bool |

4.4.2.6. Входные параметры метода «/api/v1/bauth\_logout» (завершение выполнения протокола BAUTH; терминал) приведены в таблице 50.

Таблица 50

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |

4.4.2.7. Выходных параметров метода «/api/v1/bauth\_logout» (завершение выполнения протокола BAUTH; терминал) нет.

4.4.2.8. Входные параметры метода «/api/v1/terminal\_proxy\_bauth\_prefetch» (подготовительная команда; КП) приведены в таблице 51.

Таблица 51

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| cms | Подписанные данные, полученные от СИ | JSON | BASE64\_STRING | Да |

4.4.2.9. Выходные параметры метода «/api/v1/terminal\_proxy\_bauth\_prefetch» (подготовительная команда; КП) приведены в таблице 52.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 52

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| so\_certificate | Сертификат объекта безопасности идентификационной карты (КТА) | JSON | Base64 String |
| cert\_id | ID сертификата | JSON | Base64 String |

4.4.2.10. Входные параметры метода «/api/v1/terminal\_proxy\_bauth\_init» (инициализация выполнения протокола BAUTH; КП) приведены в таблице 53.

Таблица 53

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| terminal\_certificate | Сертификат терминала | JSON | BASE64\_STRING | Да |
| cmd\_to\_card | Команда для отправки  на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.11. Выходные параметры метода «/api/v1/terminal\_proxy\_bauth\_init» (инициализация выполнения протокола BAUTH; КП) приведены в таблице 54.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 54

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | Base64 String |

4.4.2.12. Входные параметры метода «/api/v1/terminal\_proxy\_bauth» (шаги выполнения протокола BAUTH; КП) приведены в таблице 55.

Таблица 55

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.13. Выходные параметры метода «/api/v1/terminal\_proxy\_bauth» (шаги выполнения протокола BAUTH; КП) приведены в таблице 56.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 56

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | Base64 String |

4.4.2.14. Алгоритм выполнения протокола BAUTH:

1. IS -> CP (/terminal\_proxy\_bauth\_prefetch);
2. IS -> T (/bauth\_init);
3. IS -> CP (/terminal\_proxy\_bauth\_init);
4. IS -> T (/bauth\_process);
5. IS -> CP (/terminal\_proxy\_bauth),

где IS – СИ или ПС; T – терминал; CP – КП; символ «->» – отправка запроса на контроллер, указанный в скобках, и возвращение ответа, содержащего статус «200». Значение параметра «err» равно «0» в случае успешного выполнения запроса. В случае неуспешного выполнения запроса ответ будет содержать статус: «4хх», «5хх»; либо значение параметра «err» будет содержать код ошибки.

4) и 5) повторять до тех пор, пока параметр «is\_bauth\_established» в ответе от терминала   
не будет равен «true».

4.4.2.15. Чтение групп данных. Описание методов терминала и КП приведено в таблице 57.

Таблица 57

|  |  |  |
| --- | --- | --- |
| HTTP-метод | Метод  веб-сервиса | Описание метода веб-сервиса |
| Методы терминала | | |
| POST | /api/v1/read\_dg\_init | Инициализация чтения данных |
| POST | /api/v1/read\_dg | Чтение группы данных |
| POST | /api/v1/request\_dg | Получение прочитанных групп данных |
| Методы КП | | |
| POST | /api/v1/terminal\_proxy\_command | Проксирование команды терминала  на ID-карту |

4.4.2.16. Входные параметры метода «/api/v1/read\_dg\_init» (инициализация чтения данных; терминал) приведены в таблице 58.

Таблица 58

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |
| data\_groups\_to\_read | Список групп данных, которые необходимо считать с ID-карты. Возможные значения:  dg1, dg2, dg3, dg4, dg5 | JSON | JSON Array of String | Да |

4.4.2.17. Выходные параметры метода «/api/v1/read\_dg\_init» (инициализация чтения данных; терминал) приведены в таблице 59.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 59

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | Base64 String |

4.4.2.18. Входные параметры метода «/api/v1/read\_dg» (чтение группы данных; терминал) приведены в таблице 60.

Таблица 60

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.19. Выходные параметры метода «/api/v1/read\_dg» (чтение группы данных; терминал) приведены в таблице 61.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 61

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| is\_last\_dg\_readed | Показывает, прочитана ли последняя группа данных | JSON | Bool |

4.4.2.20. Входные параметры метода «/api/v1/request\_dg» (получение прочитанных групп данных; терминал) приведены в таблице 62.

Таблица 62

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |

4.4.2.21. Выходные параметры метода «/api/v1/request\_dg» (получение прочитанных групп данных; терминал) приведены в таблице 63.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 63

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| personal\_data | Прочитанные группы данных | JSON | JSON Object |

Состав параметра «personal\_data» указан в таблице 64.

Таблица 64

| Элемент | Тип элемента | Описание |
| --- | --- | --- |
| BE:\_Family\_name | String | Фамилия на белорусском языке |
| BE:\_Given\_name | String | Имя на белорусском языке |
| BE:\_Middle\_name | String | Отчество на белорусском языке |
| LA:\_Family\_name | String | Фамилия на латинице |
| LA:\_Given\_name | String | Имя на латинице |
| RU:\_Family\_name | String | Фамилия на русском языке |
| RU:\_Given\_name | String | Имя на русском языке |
| RU:\_Middle\_name | String | Отчество на русском языке |
| RU\_Place\_of\_birth | String | Место рождения на русском языке |
| BE\_Place\_of\_birth | String | Место рождения на белорусском языке |
| Citizenship | String | Гражданство |
| Date\_of\_expiry | String | Дата окончания действия идентификационной карты (КТА) |
| Date\_of\_issuance | String | Дата выдачи идентификационной карты (КТА) |
| ID | String | Идентификационный номер |
| Issuance\_board | String | Орган, выдавший идентификационную карту (КТА) |
| Issuing\_State | String | Место выдачи идентификационной карты (КТА) |
| Serial\_Number | String | Серийный номер идентификационной карты (КТА) |
| Sex | String | Пол |
| Type | String | Тип параметра |
| birth\_date | String | Дата рождения |

4.4.2.22. Входные параметры метода «/api/v1/terminal\_proxy\_command» (проксирование команды терминала на ID-карту; КП) приведены в таблице 65.

Таблица 65

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.23. Выходные параметры метода «/api/v1/terminal\_proxy\_command» (проксирование команды терминала на ID-карту; КП) приведены в таблице 66.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 66

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | Base64 String |

4.4.2.24. Алгоритм чтения групп данных:

1. IS -> T (/read\_dg\_init);
2. IS -> CP (/terminal\_proxy\_command);
3. IS -> T (/read\_dg);
4. IS -> CP (/terminal\_proxy\_command);
5. 3) и 4) повторять до тех пор, пока параметр (is\_last\_dg\_readed) в ответе терминала не будет равен true;
6. IS -> T (/request\_dg),

где IS – СИ или ПС; T – терминал; CP – КП; символ «->» – отправка запроса на контроллер, указанный в скобках, и возвращение ответа, содержащего статус «200». Значение параметра «err» равно «0» в случае успешного выполнения запроса. В случае неуспешного выполнения запроса ответ будет содержать статус: «4хх», «5хх»; либо значение параметра «err» будет содержать код ошибки.

4.4.2.25. Выработка ЭЦП. Описание методов терминала и КП приведено в таблице 67.

Таблица 67

|  |  |  |
| --- | --- | --- |
| HTTP-метод | Метод  веб-сервиса | Описание метода веб-сервиса |
| Методы терминала | | |
| POST | /api/v1/sign\_init | Инициализация ЭЦП |
| POST | /api/v1/sign\_select\_app | Выбор прикладной программы eSign |
| POST | /api/v1/sign\_data | ЭЦП данных |
| POST | /api/v1/sign\_result | Получение результата ЭЦП |
| Методы КП | | |
| POST | /api/v1/terminal\_proxy\_sign\_init | Инициализация ЭЦП |
| POST | /api/v1/terminal\_proxy\_command | ЭЦП |

4.4.2.26. Входные параметры метода «/api/v1/sign\_init» (инициализация ЭЦП; терминал) приведены в таблице 68.

Таблица 68

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный хэш-идентификатор | JSON | BASE64\_STRING | Да |

4.4.2.27. Выходные параметры метода «/api/v1/sign\_init» (инициализация ЭЦП; терминал) приведены в таблице 69.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 69

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | Base64 String |

4.4.2.28. Входные параметры метода «/api/v1/sign\_select\_app» (выбор прикладной программы eSign; терминал) приведены в таблице 70.

Таблица 70

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.29. Выходные параметры метода «/api/v1/sign\_select\_app» (выбор прикладной программы eSign; терминал) приведены в таблице 71.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 71

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | Base64 String |

4.4.2.30. Входные параметры метода «/api/v1/sign\_data» (ЭЦП данных; терминал) приведены в таблице 72.

Таблица 72

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | BASE64\_STRING | Да |
| data\_to\_sign | Данные, которые необходимо подписать | JSON | BASE64\_STRING | Да |

4.4.2.31. Выходные параметры метода «/api/v1/sign\_data» (ЭЦП данных; терминал) приведены в таблице 73.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 73

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| header\_cmd\_to\_card | Заголовок команды для отправки на идентификационную карту (КТА) | JSON | Base64 String |
| cmd\_to\_card | Команда для отправки  на идентификационную карту (КТА) | JSON | Base64 String |

4.4.2.32. Входные параметры метода «/api/v1/sign\_result» (получение результата ЭЦП; терминал) приведены в таблице 74.

Таблица 74

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| hreq | Уникальный  хэш-идентификатор | JSON | BASE64\_STRING | Да |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.33. Выходные параметры метода «/api/v1/sign\_result» (получение результата ЭЦП; терминал) приведены в таблице 75.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 75

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| signature | Значение ЭЦП | JSON | Base64 String |

4.4.2.34. Входные параметры метода «/api/v1/terminal\_proxy\_sign\_init» (инициализация ЭЦП; КП) приведены в таблице 76.

Таблица 76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| header\_cmd\_to\_card | Заголовок команды  для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.35. Выходные параметры метода «/api/v1/terminal\_proxy\_sign\_init» (инициализация ЭЦП; КП) приведены в таблице 77.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 77

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | Base64 String |

4.4.2.36. Входные параметры метода «/api/v1/terminal\_proxy\_command» (ЭЦП; КП) приведены в таблице 78.

Таблица 78

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Описание | Тип параметра | Тип данных | Наличие обязательно |
| header\_cmd\_to\_card | Заголовок команды  для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |
| cmd\_to\_card | Команда для отправки на идентификационную карту (КТА) | JSON | BASE64\_STRING | Да |

4.4.2.37. Выходные параметры метода «/api/v1/terminal\_proxy\_command» (ЭЦП; КП) приведены в таблице 79.

Содержимое ответа о результатах обработки сообщения располагается в теле HTTP-ответа в JSON-виде.

Таблица 79

| Параметр | Описание | Тип параметра | Тип данных |
| --- | --- | --- | --- |
| err | Код ошибки | JSON | String |
| card\_response | Ответ от идентификационной карты (КТА) | JSON | Base64 String |

4.4.2.38. Алгоритм выработки ЭЦП:

1. IS -> T (/sign\_init);
2. IS -> CP (/terminal\_proxy\_sign\_init);
3. IS -> T (/sign\_select\_app);
4. IS -> CP (/terminal\_proxy\_command);
5. IS -> T (/sign\_data);
6. IS -> CP (/terminal\_proxy\_command);
7. IS -> T (/sign\_result),

где IS – СИ или ПС; T – терминал; CP – КП; символ «->» – отправка запроса на контроллер, указанный в скобках, и возвращение ответа, содержащего статус «200». Значение параметра «err» равно «0» в случае успешного выполнения запроса. В случае неуспешного выполнения запроса ответ будет содержать статус: «4хх», «5хх»; либо значение параметра «err» будет содержать код ошибки.

5. Проверка программы

5.1. Проверка работоспособности КПСИС включает в себя следующие действия:

* проверка запуска контейнеров программы «Docker-compose»;
* проверка возможности «слушать» определенный порт контейнерами программы «Docker-compose»;
* проверка возможности «слушать» определенный порт самим КПСИС;
* отправка тестового HTTP — запроса для формирования ссылки на авторизацию.

5.2. Запуск контейнеров программы «Docker-compose» завершен успешно, если выполнение команд запуска программы в терминале дает результат, представленный на рис. 16.

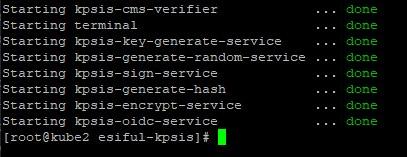


Рис. 16

5.3. Возможность «слушать» определенный порт контейнерами программы   
«Docker-compose» подтверждается наличием соответствующей строчки в терминале при выполнении команд запуска программы (рис. 17).

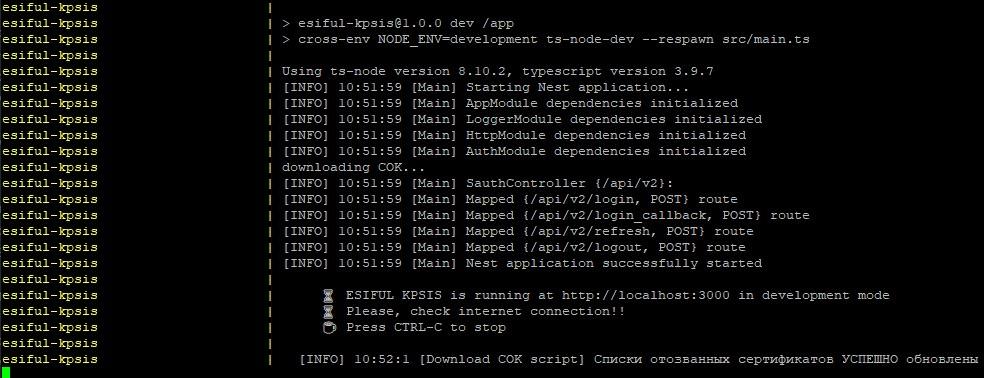


Рис. 17

5.4. Отправка тестового HTTP-запроса для формирования ссылки на авторизацию

Запрос:

POST /api/v2/login

HTTP/1.1

Host: localhost:3000

Content-Type: application/json

{

"settings": {

"response\_type": "code",

"grant\_type": "authorization\_code",

"redirect\_uri": "http://localhost:8080",

"client\_id": "5c0SbSd9IiDIlhmF7q\_V0pq-uGp8z6y8dzWHQIisFIM",

"post\_logout\_redirect\_uri": "http://localhost:8080",

"scope": "openid profile offline\_access",

"prompt": "consent",

"authority": "https://esiful.nces.by:8081"

}

}

Ответ:

{

enveloped\_and\_signed\_auth\_url: 'http://192.168.0.177:8081/authz?settings=MIIMBwYJKoZIhvcNAQcCoIIL+DCCC/……iAq09b61tHT4KgAzjJY0QdzdHM7Q+pl8sT+e5l/LlbiZcc3Ty1BQajFFgcfbNe8=&client\_id=NWMwU2JTZDlJaURJbGhtRjdxX1YwcHEtdUdwOHo2eThkeldIUUlpc0ZJTQ==',

code\_verifier: 'DjHWsx4HAlWFSnfA6CSoim8YpH-PqJfd006zdl~NOnz7jlbHDlf9F4UNA-9i5-Ppq2Qd5qvRKN.~uVm04ZZl638lHY4.KVTuUylfbZdfXNp8cjQSkfpS9HSD4U5rPgEo',

signed\_data\_to\_check\_in\_cp: 'MIIMBwYJKoZIhvcNAQcCoIIL+DCCC/……AEPgQyBDAALAAgADIANTEdMBsGCSqGSIb3DQEJARYOcmNhQHBraS5nb3YuYnkCDEDljxE+u0gAAA0mwjANBgkqcAACACJlH1EFAKBpMBgGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTIxMDMxNTEwMzExNVowLwYJKoZIhvcNAQkEMSIEIGen61FMT77HDeQ52wJz9uffw4L5Kh6tgFh6E3kKcR5mMA0GCSpwAAIAImUtDAUABDCIiAq09b61tHT4KgAzjJY0QdzdHM7Q+pl8sT+e5l/LlbiZcc3Ty1BQajFFgcfbNe8='

}

5.5. Успешное выполнение описанных в настоящем разделе действий свидетельствует о работоспособности КПСИС.

6. Сообщения системному программисту

6.1. Тексты сообщений, выдаваемых в ходе выполнения настройки, проверки КПСИС, а также в ходе выполнения КПСИС, описание их содержания и действий, которые необходимо предпринять по этим сообщениям, приведены в таблице 80.

Таблица 80

| Текст сообщения | Причина сообщения | Действие системного программиста |
| --- | --- | --- |
| Ошибки сервиса выработки ЭЦП | | |
| NO\_MEMORY =1 | Недостаточно памяти | Перезапустить сервис |
| SIGN\_SRVC\_CONF\_NOT\_FOUND=2 | Не найден конфигурационный файл сервиса ЭЦП | Проверить корректность местонахождения файла настроек сервиса ЭЦП |
| SIG\_SVC\_FAILED\_TO\_LOAD\_KEY=3 | Не удалось загрузить личный ключ | Проверить корректность настроек для восстановления секрета |
| SIG\_SVC\_FAILED\_TO\_LOAD\_CERT=4 | Не удалось загрузить СОК | Проверить корректность пути к файлу с СОК |
| OPERATION\_CMS\_FAILED=5 | Не удалось сформировать ЭЦП | Проверить корректность настроек для формирования ЭЦП |
| 400 «Bad request» | В запросе указаны неверные входные параметры | Указать верные входные параметры |
| 404 «Not found» | Неправильно задан URL | Проверьте корректность URL |
| Ошибки сервиса проверки ЭЦП | | |
| 400 «Bad request» | В запросе указаны неверные входные параметры | Указать верные входные параметры |
| 500 «Server internal error» | Внутренняя ошибка сервера | Перезапустить сервис |
| NO\_MEMORY = 3 | Не достаточно оперативной памяти | Освободить оперативную память, остановить лишние процессы |
| ADD\_CERT\_TO\_STORE\_FAILED =6 | Отсутствуют корневые СОК | Убедиться, что корневые СОК существуют и к ним правильно прописан путь в конфигурационном файле |
| ADD\_CRL\_TO\_STORE\_FAILED =7 | Отсутствуют СОС | Убедиться, что СОС существуют и к ним правильно прописан путь в конфигурационном файле |

*Продолжение таблицы 80*

| Текст сообщения | Причина сообщения | Действие системного программиста |
| --- | --- | --- |
| Ошибки сервиса предварительного шифрования | | |
| 400 «Bad request» | В запросе указаны неверные входные параметры | Указать верные входные параметры |
| 404 «Not found» | Неправильно задан URL | Проверьте корректность URL |
| NO\_MEMORY=1 | Недостаточно памяти | Перезапустить сервис |
| SIGN\_SRVC\_CONF\_NOT\_FOUND=2 | Не найден конфигурационный файл сервиса предварительного шифрования | Проверить корректность местонахождения файла настроек сервиса |
| DEC\_SVC\_FAILED\_TO\_LOAD\_KEY=3 | Не удалось загрузить личный ключ | Проверить корректность настроек для восстановления секрета |
| CO\_ENCRYPT\_CMS\_FAILED=4 | Не удалось выполнить функцию шифрования данных | Проверить корректность настроек в конфигурационном файле сервиса предварительного шифрования |
| CO\_PARSE\_CERT\_FAILED=5 | Данные, переданные в поле СОК, не являются СОК | Сформировать запрос правильно |
| CO\_GENERATE\_RANDOM=6 | Ошибка генерации псевдослучайного числа | Перезапустить сервис |
| CO\_DECRYPT\_CMS\_FAILED=7 | Ошибка расшифрования данных | Перезапустить сервис |
| NOT\_COMPARE\_MAC=8 | Нарушена целостность переданного зашифрованного сообщения | Отклонить данные |
| DEC\_SVC\_FAILED\_TO\_LOAD\_CERT=9 | Не удалось загрузить СОК | Проверить корректность пути к файлу с СОК |
| CO\_FAILED\_TO\_GET\_MAC=100 | Ошибка расчета имитовставки | Перезапустить сервис |
| CO\_FAILED\_TO\_KEY\_TRANSPORT=101 | Ошибка создания токена ключа | Перезапустить сервис |
| CO\_FAILED\_FROM\_KEY\_TRANSPORT=102 | Ошибка извлечения ключа из токена. Переданный сертификат не соответствует личному ключу пользователя | Отклонить данные |

*Окончание таблицы 80*

| Текст сообщения | Причина сообщения | Действие системного программиста |
| --- | --- | --- |
| CO\_PARSE\_CMS\_FAILED=103 | Формат переданных данных не соответствует структуре CMS | Отклонить данные |
| Ошибки сервиса генерации псевдослучайной числовой последовательности, сервиса контроля целостности и сервиса генерации личного ключа и выпуска запроса на СОК | | |
| 400 «Bad request» | В запросе указаны неверные входные параметры | Указать верные входные параметры |
| 500 «Server internal error» | Внутренняя ошибка сервера | Перезапустить сервис |

Перечень сокращений

В настоящем документе приняты следующие сокращения:

|  |  |
| --- | --- |
| ЕС ИФЮЛ | – Единая система идентификации физических и юридических лиц; |
| КП | – клиентская программа; |
| КПСИС | – комплекс программных средств прикладной системы; |
| НЦЭУ | – Республиканское унитарное предприятие «Национальный центр электронных услуг»; |
| ОС | – операционная система; |
| ПС | – прикладная система; |
| СИ | – сервер идентификации; |
| СОК | – сертификат открытого ключа; |
| СОС | – список отозванных сертификатов; |
| ЭЦП | – электронная цифровая подпись; |
| OIDC | – OpenID Connect. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов (страниц) | | | | Всего листов (стра-ниц) в доку-менте | Номер  документа | Входящий номер сопроводи-тельного документа  и дата | Подпись | Дата |
| изменен-ных | заменен-ных | новых | аннули-рованных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |