Закрытое акционерное общество «НТЦ КОНТАКТ»

УТВЕРЖДАЮ

Директор

ЗАО «НТЦ КОНТАКТ»

_____А.А.Тепляков

1 декабря 2021 г.

КОМПЛЕКС ПРОГРАММНО-АППАРАТНЫЙ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ «БАС»

Инструкция по управлению ключевой информацией

СЮИК.465634.001 ИС21

Содержание

1 Описание работы 4
2 Описание процесса управления ключевой информацией 5
3 Настройка генерации личного ключа и запроса на выпуск сертификата открытого ключа 6
4 Формирование запроса на получение сертификата открытого ключа 8
5 Экспорт запроса на получение сертификата открытого ключа 12
6 Импорт сертификата открытого ключа 13
7 Настройка программного обеспечения 14
8 Формирование повторного запроса на получение сертификата открытого ключа 15
9 Замена ключевой пары 19
10 Настройка восстановления личного ключа 24
11 Восстановление личного ключа 26
12 Уничтожение личного ключа (возврат к заводским настройкам) 27

Настоящая инструкция распространяется на «Комплекс программно-аппаратный криптографической защиты информации «БАС» СЮИК.465634.001 (далее – ПАК «БАС»), предназначенный для защиты информации, циркулирующей в каналах передачи данных.

Настоящая инструкция является расширением Руководства по эксплуатации ПАК «БАС» СЮИК.465634.001 РЭ и предназначена для облегчения работы администратора при настройке ПАК «БАС».

Настоящая инструкция предназначена для администратора, имеющего навыки работы с ОС Linux и сетевым администрированием. Для понимания принципов работы ПАК «БАС» администратор должен ознакомиться с документом «Комплекс программно-аппаратный криптографической защиты информации «БАС». Руководство по эксплуатации» СЮИК.465634.001 РЭ прежде, чем преступить к настройкам согласно данной инструкции.

1 Описание работы

ПАК «БАС» обеспечивает криптографическую защиту передаваемых данных посредством полной инкапсуляции IP-пакетов путем их шифрования с использованием криптографических алгоритмов на основе протоколов IPsec.

IPsec (англ. Internet Protocol Security) – это структура открытых стандартов для обеспечения конфиденциальных, безопасных соединений по интернет-протоколу (IP) сетей, за счет использования криптографических служб безопасности.

Специфика IPsec состоит в том, что он реализуется на сетевом уровне, дополняя его таким образом, чтобы для последующих уровней все происходило незаметно. Основная сложность состоит в том, что в процессе установки соединения двум участникам защищенного канала необходимо согласовать довольно большое количество различных параметров: они должны аутентифицировать друг друга, сгенерировать и обменяться ключами (причем через недоверенную среду), а также договориться, при помощи каких протоколов защищать данные.

Именно по этой причине IPsec и состоит из стека протоколов, обязанность которых обеспечить установку защищенного соединения, его работу и управление им. Весь процесс установки соединения включает две фазы.

В процессе первой фазы участники аутентифицируют друг друга и договариваются о параметрах установки специального соединения, предназначенного только для обмена информацией, о желаемых алгоритмах шифрования и прочих деталях будущего IPsec-туннеля.

На второй фазе уже доверяющие друг другу участники договариваются о том, как строить основной туннель для передачи непосредственно данных. Они предлагают друг другу варианты и, если приходят к согласию, поднимают основной туннель.

Для надежной аутентификации участники должны обладать неизвлекаемым личным ключом, а также парным ему открытым ключом в составе сертификата.

2 Описание процесса управления ключевой информацией

Для надежного управления личным ключом ПАК «БАС» имеет в своем составе «Комплекс программно-аппаратный защиты информации от несанкционированного доступа «Барьер – USB» СЮИК.467458.004 (далее – ПАК «Барьер – USB»).

Генерация ключевой информации в ПАК «БАС» осуществляется при помощи утилиты **RequestBuilder**, которая в процессе своей работы:

– генерирует личный ключ ПАК «БАС» с использованием случайности, полученной от физического источника ПАК «Барьер – USB»;

– помещает личный ключ в защищенное хранилище ПАК «Барьер – USB»;

– вычисляет открытый ключ;

– помещает открытый ключ вместе с данными о ПАК «БАС» в запрос на получение сертификата.

Дополнительно может быть сформирована резервная копия личного ключа для возможности его восстановления, в этом случае:

– копия личного ключа помещается в ключевой контейнер;

– ключевой контейнер защищается при помощи высокоэнтропийного ключа, сгенерированного с использованием случайности, полученной от физического источника ПАК «Барьер – USB»;

– высокоэнтропийный ключ разделяется на частичные секреты;

– частичные секреты помещаются в ключевые контейнеры;

– ключевые контейнеры защищаются при помощи ключа, сгенерированного с использованием пароля, заданного Администратором.

После генерации личного ключа ПАК «БАС» считается инсталлированным и при помощи ПАК «Барьер – USB» ведет непрерывный контроль вскрытия корпуса. При обнаружении вскрытия корпуса уничтожается личный ключ, хранящийся в защищенном хранилище ПАК «Барьер – USB». Устройство при этом переходит в режим блокировки, из которого может быть выведено только Администратором.

3 Настройка генерации личного ключа и запроса на выпуск сертификата открытого ключа

Настройки ПАК «БАС» по умолчанию позволяют выполнить генерацию личного ключа и сформировать запроса на получение сертификата открытого ключа. Однако у Администратора есть возможность выполнить дополнительные настройки, если есть такая необходимость.

Конфигурирование утилиты **RequestBuilder** осуществляется посредством редактирования настроечного файла. Настоечный файл представляет собой текстовый файл, разделённый на секции. Секции разделяются именем (секции), заключённым в прямоугольные скобки ([имя_секции]). Каждая секция может иметь одно или несколько полей вида «Параметр = Значение». Имена параметров предопределены для каждой секции.

Утилита обращается к настроечному файлу с именем **RequestBuilder.conf**, расположенному в директории «**/etc/support»**.

Секция, соответствующая модулю формирования заявки на выпуск сертификата открытого ключа, должна именоваться **RequestBuilder**. Данная секция включает следующие параметры:

– **BasType**. Параметр, устанавливающий тип устройства;

– **PersonalData**. Значением параметра должен быть путь к XML-документу, содержащему персональные данные, необходимые для формирования запроса на выпуск сертификата открытого ключа.

– **CardTemplate**. Значением параметра должен быть путь к RTF-документу, содержащему шаблон карточки открытого ключа;

– **OutputDirectory**. Значением параметра должен быть путь к директории, в которую необходимо сохранить сформированный запрос на выпуск сертификата, карточку открытого ключа, и, если необходимо, контейнер личного ключа;

– **ShareSecretsAmount**. Значением параметра должно быть число, определяющее количество частичных секретов, на которое должен быть разделен

ключ защиты контейнера с личным ключом, если необходимо. Значение должно быть не меньше 2 и не больше 16;

– **Threshold**. Значением параметра должно быть пороговое число частичных секретов, из которых должен быть восстановлен ключ защиты контейнера с личным ключом. Значение должно быть не меньше 2 и не больше значения параметра **ShareSecretsAmount**;

– **ShareSecretPath_N**. N – порядковый номер частичного секрета. Значением параметра должен быть путь к директории, в которую необходимо сохранить N-ый частичный секрет.

Пример заполнения секции RequestBuilder:

```
[RequestBuilder]
        BasType             = hardware
        PersonalData        = ./PersonalData.xml
        CardTemplate        = ./Templates/PublicKeyCardTemplate.rtf
        OutputDirectory     = ./IssueRequestsAndCards/
        ShareSecretsAmount  = 2
        Threshold           = 2
        ShareSecretPath_1   = ./IssueRequestsAndCards/
        ShareSecretPath_2   = ./IssueRequestsAndCards/
```

При использовании настроек по умолчанию высокоэнтропийный ключ, при создании резервной копии личного ключа, будет разделен на 2 частичных секрета. Для его восстановления также потребуется иметь 2 частичных секрета. Администратор может изменить количество частичных секретов и величину кворума.

4 Формирование запроса на получение сертификата открытого ключа

Для формирования запроса на получение сертификата открытого ключа необходимо воспользоваться утилитой **RequestBuilder**. Она выполнит самотестирование ПАК «БАС», сгенерирует личный ключ и сформирует запрос на получение сертификата открытого ключа.

```
server@server:~$ sudo RequestBuilder
[sudo] пароль для server:
> Контроль Целостности
Проверка целостности прошла успешно!
Результаты: /etc/support/IntegrityController.log
> Тестирование Библиотеки Криптопреобразований
    Зашифрование в режиме простой замены (|X| = 384) +
    Зашифрование в режиме простой замены (|X| = 376) +
    Расшифрование в режиме простой замены (|X| = 384) +
    Расшифрование в режиме простой замены (|X| = 288) +
    Зашифрование в режиме сцепления блоков (|X| = 384) +
    Зашифрование в режиме сцепления блоков (|X| = 288) +
    Расшифрование в режиме сцепления блоков (|X| = 384) +
    Расшифрование в режиме сцепления блоков (|X| = 288) +
    Зашифрование в режиме гаммирования с обратной связью +
    Расшифрование в режиме гаммирования с обратной связью +
    Шифрование в режиме счетчика +
    Выработка имитовставки (|X| = 104) +
    Выработка имитовставки (|X| = 384) +
    Установка защиты данных +
    Снятие защиты данных +
    Установка защиты ключа +
    Снятие защиты ключа +
    Хэширование (|X| = 104) +
    Хэширование (|X| = 256) +
    Хэширование (|X| = 384) +
    Преобразование ключа (m = 128) +
    Преобразование ключа (m = 192) +
    Преобразование ключа (m = 256) +
  Тестирование алгоритмов СТБ.34.101.31 выполнено.
    Генерация пары ключей +
    Выработка электронной цифровой подписи +
    Проверка электронной цифровой подписи +
    Создание токена ключа +
    Разбор токена ключа +
    Извлечение пары ключей +
    Выработка идентификационной электронной цифровой подписи +
    Проверка идентификационной электронной цифровой подписи +
  Тестирование алгоритмов СТБ 34.101.45 выполнено.
    Выработка имитовставки (алгоритм hmac-hbelt, keySize = 232) +
    Выработка имитовставки (алгоритм hmac-hbelt, keySize = 256) +
```

```
        Выработка имитовставки (алгоритм hmac-hbelt, keySize = 336) +
        Генерация псевдослучайных чисел (алгоритм brng-ctr-hbelt) +
        Генерация псевдослучайных чисел (алгоритм brng-hmac-hbelt) +
      Тестирование алгоритмов СТБ 34.101.47 выполнено.
        Разделение секрета (l = 128) +
        Восстановление секрета (l = 128), подмножество пользователей {1,2} +
        Восстановление секрета (l = 128), подмножество пользователей {1,3} +
        Восстановление секрета (l = 128), подмножество пользователей {1,4} +
        Восстановление секрета (l = 128), подмножество пользователей {1,5} +
        Восстановление секрета (l = 128), подмножество пользователей {2,3} +
        Восстановление секрета (l = 128), подмножество пользователей {2,4} +
        Восстановление секрета (l = 128), подмножество пользователей {2,5} +
        Восстановление секрета (l = 128), подмножество пользователей {3,4} +
        Восстановление секрета (l = 128), подмножество пользователей {3,5} +
        Восстановление секрета (l = 128), подмножество пользователей {4,5} +
        Восстановление секрета (l = 128), подмножество пользователей {1,3,5} +
        Восстановление секрета (l = 128), подмножество пользователей {2,3,4,5} +
        Восстановление секрета (l = 128), подмножество пользователей {1,2,3,4,5} +
      Тестирование алгоритмов СТБ 34.101.60 выполнено.
        Сеанс протокола BMQV +
        Сеанс протокола BSTS +
        Сеанс протокола BPACE +
        Сеанс протокола Диффи-Хеллмана +
      Тестирование алгоритмов СТБ 34.101.66 выполнено.
        Зашифрование в режиме простой замены +
        Расшифрование в режиме простой замены +
        Зашифрование в режиме гаммирования с обратной связью +
        Расшифрование в режиме гаммирования с обратной связью +
        Шифрование в режиме гаммирования +
        Выработка имитовставки +
      Тестирование алгоритмов ГОСТ 28147-89 выполнено.
    Самотестирование библиотеки криптографических преобразований завершено успешно.
    > Контроль Работоспособности ПАК "Барьер-USB"
    На защищённом хранилище не установлена парольная защита!
    Защищённое хранилище готово к работе.
    <13>Jan  1 12:00:00 basctl: Тестирование завершено успешно!
    Желаете отредактировать XML-файл с данными об устройстве? [/etc/support/PersonalData.xml]
(Y/N): y
```

Для формирования запроса на получение сертификата необходимо отредактировать XML-файл с данными об устройстве, указав в нем серийный номер устройства, название организации и адрес, где эксплуатируется ПАК «БАС».

Если ПАК «БАС» планируется для использования в качестве сервера для подключения программных клиентов обязательно должно быть заполнено поле SubjectAltName. Рекомендуется указать открытый IP-адрес ПАК «БАС».

```
<?xml version="1.0" encoding="UTF-8"?>
<PersonalData>
   <Subject>
      <CommonName OId="2.5.4.3" Description="Общее имя устройства (DNS-имя, IP-адрес сервера,
      ID сервера, устройства)">
         BAS00001
      </CommonName>
      <CountryName OId="2.5.4.6" Description="Код страны нахождения организации">
         BY
      </CountryName>
      <LocalityName OId="2.5.4.7" Description="Населённый пункт нахождения организации">
         г. Минск
      </LocalityName>
      <StateOrProvinceName OId="2.5.4.8" Description="Обл. и район нахождения орг.”>

      </StateOrProvinceName>
      <StreetAddress OId="2.5.4.9" Description="Улица, дом, корпус, офис">
         пер.Студенческий, д.17
      </StreetAddress>
      <OrganizationName OId="2.5.4.10" Description="Сокращенное название организации">
         ЗАО "НТЦ КОНТАКТ"
      </OrganizationName>
      <Description OId="2.5.4.13" Description="Описание субъекта">
         Комплекс программно-аппаратный криптографической защиты информации "БАС".Сервер защиты
      </Description >
      <OrganizationUnitName OId="2.5.4.11" Description="Подразделение организации">

      </OrganizationUnitName>
   </Subject>
   <ExtensionRequest OId="1.3.6.1.4.1.311.2.1.14" Description="Расширения сертификата">
      <!--
      <SubjectAltName OId="2.5.29.17" Description="Альтернативное имя устройства">
         <EMail> example@mail.by </EMail>
         <DNS> example.by </DNS>
         <URI> http://example.by </URI>
         <IP> 10.0.0.1 </IP>
      </SubjectAltName> -->

      <!-- For GosSUOK uncomment next -->
      <!--
      <CertificatePolicies OId="2.5.29.32" Description="Политики сертификата">
         1.2.112.1.2.1.1.1.3.2.2|1.2.112.0.2.0.34.101.78.2.70
      </CertificatePolicies> -->
   </ExtensionRequest>
</PersonalData>
```

Для сохранения изменений в файле, необходимо нажать сочетание клавиш **Ctrl+O**, и выйти из текстового редактора **nano**, нажав сочетание клавиш **Ctrl+X**.

Далее необходимо установить пароль для доступа к защищенному хранилищу, при необходимости резервирования личного ключа задать пароль к ключевому контейнеру частичного секрета.

```
Введите пароль доступа к защищённому хранилищу (8-24 символа): ********
Подтвердите пароль: ********
Сохранить личный ключ для возможности восстановления системы? (Y/N): y
Задайте пароль доступа к контейнеру личного ключа (8-24 символа): ********
Подтвердите пароль: ********
Запрос на получение сертификата открытого ключа успешно сохранен:
[/etc/support/IssueRequestsAndCards/CertificateIssueRequest_0123456789101112.der]
Карточка открытого ключа успешно сохранена:
[/etc/support/IssueRequestsAndCards/PublicKeyCard_0123456789101112.rtf]
Контейнер личного ключа успешно сохранён:
[/etc/support/IssueRequestsAndCards/PrivateKeyContainer.pkc]
Контейнер частичного секрета №1 успешно сохранён:
[/etc/support/IssueRequestsAndCards/ShareSecretContainer_1.ssc]
Контейнер частичного секрета №2 успешно сохранён:
[/etc/support/IssueRequestsAndCards/ShareSecretContainer_2.ssc]
server@server:~$
```

В результате выполнения утилиты **RequestBuilder**, в папке **/etc/support/IssueRequestsAndCards/** были сформированы:

– запрос на выпуск сертификата в соответствии с СТБ 34.101.17:

CertificateIssueRequest_0123456789101112.der

– карточка открытого ключа в соответствии с СТБ 34.101.49:

PublicKeyCard_0123456789101112.rtf

– контейнер личного ключа в соответствии с СТБ 34.101.78:

PrivateKeyContainer.pkc

– контейнеры частичных секретов в соответствии с СТБ 34.101.78:

ShareSecretContainer_1.ssc

ShareSecretContainer_2.ssc

Контейнеры частичных секретов должны быть экспортированы из ПАК «БАС» и распределены на разных носителях между Администраторами, а их копии в файловой системе ПАК «БАС» удалены при помощи команды **rm**.

Для выпуска сертификата открытого ключа необходимо экспортировать запрос на получение сертификата из ПАК «БАС» любым удобным способом и передать Администратору Удостоверяющего центра.

5 Экспорт запроса на получение сертификата открытого ключа

Для экспорта запроса на получение сертификата открытого ключа на съемный USB-носитель необходимо подключить носитель к ПАК «БАС».

При помощи команды **fdisk** необходимо определить имя, присвоенное съемному USB-носителю операционной системой ПАК «БАС».

```
server@server:~$ sudo fdisk -l
...
Device         Boot   Start  End          Sectors      Size   Id     Type
/dev/sdb1      *      2048   7935999      7933952      3,8 G  c      W95 FAT32 (LBA)
```

Операционная система ПАК «БАС» присвоила подключенному съемному USB-носителю объемом 4 Гбайт имя **/dev/sdb1**.

Для монтирования файловой системы съемного USB-носителя необходимо воспользоваться командой **mount**.

```
server@server:~$ sudo mount /dev/sdb1 /mnt
```

Для копирования файла запроса на получение сертификата на съемный USB-носитель необходимо воспользоваться командой **cp**.

```
server@server:~$ sudo cp /etc/support/IssueRequestsAndCards/
CertificateIssueRequest_0123456789101112.der /mnt/
```

Для того чтобы убедиться, что запрос на получение сертификата был успешно скопирован на съемный USB-носитель, необходимо воспользоваться командой **ls**.

```
server@server:~$ ls /mnt/
CertificateIssueRequest_0123456789101112.der
```

Для размонтирования файловой системы съемного USB-носителя необходимо воспользоваться командой **umount**.

```
server@server:~$ sudo umount /mnt
```

Извлечь съемный USB-носитель и передать Администратору Удостоверяющего центра для выпуска сертификата и записи его на носитель.

Если Удостоверяющий центр требует имени или расширения файла запроса на выпуск сертификата, отличного от сгенерированного ПАК «БАС», файл запроса может быть переименован при помощи стандартных средств ОС.

6 Импорт сертификата открытого ключа

Если файл сертификата был получен в формате p7b, необходимо выполнить экспорт в отдельные файлы сертификатов. Экспорт сертификатов из p7b-файла может быть выполнен при помощи Мастера экспорта сертификатов ОС Windows.

Для импорта сертификата открытого ключа со съемного USB-носителя необходимо подключить носитель к ПАК «БАС», определить имя, присвоенное съемному USB-носителю ОС, произвести монтирование файловой системы.

```
server@server:~$ sudo mount /dev/sdb1 /mnt
```

В файловой системе съемного USB-носителя находятся сертификат открытого ключа, корневой сертификат и список отозванных сертификатов.

Сертификаты открытого ключа необходимо импортировать в ПАК «БАС». Сертификат устройства в папку **/usr/local/etc/ipsec.d/certs/**, корневой сертификат, а также промежуточные (при их наличии), в **/usr/local/etc/ipsec.d/cacerts/**, файлы списка отозванных сертификатов (при их наличии), в **/usr/local/etc/ipsec.d/crl/**.

```
server@server:~$ sudo cp /mnt/Root.cer /usr/local/etc/ipsec.d/cacerts/
server@server:~$ sudo cp /mnt/cert00001.cer /usr/local/etc/ipsec.d/certs/
server@server:~$ sudo cp /mnt/RootCRL.crl /usr/local/etc/ipsec.d/crl/
```

ВНИМАНИЕ: Если сертификат ПАК «БАС» был выпущен подчиненным Удостоверяющим центром, то сертификаты всех центров цепочки доверия должны быть импортированы в папку **/usr/local/etc/ipsec.d/cacerts/**.

В процессе установки защищенного соединения ПАК «БАС» проверяют состояние сертификатов друг друга. Для этого используются списки отозванных сертификатов, которые ПАК «БАС» запрашивает автоматически из точки распространения, указанной в сертификате партнера (при ее доступности). При невозможности получить список из точки распространения, используются списки из папки **/usr/local/etc/ipsec.d/crl/**. В таком случае поддержка списков в актуальном состоянии возлагается на Администратора. При отсутствии списков в папке **/usr/local/etc/ipsec.d/crl/** и в точке распространения ПАК «БАС» не может проверить валиден ли сертификат, считает, что список пуст, а сертификат валиден.

7 Настройка программного обеспечения

Для того, чтобы ПАК «БАС» начал использовать импортированный в него сертификат открытого ключа в качестве пары к личному ключу из защищенного хранилища ПАК «Барьер – USB», необходимо заполнить параметр **leftcert** в настроечном файле **/usr/local/etc/ipsec.conf**, указав в нем имя файла сертификата открытого ключа из папки **/usr/local/etc/ipsec.d/certs/**.

```
server@server:~$ sudo nano /usr/local/etc/ipsec.conf
...
leftcert = cert00001.cer
...
```

Для применения настроек необходимо перезапустить IPsec соединение.

```
server@server:~$ sudo ipsec restart
Stopping strongSwanCont IPsec...
Starting strongSwanCont 5.8.4 IPsec [starter]...
```

Убедиться в том, что программное обеспечение ПАК «БАС» подгрузило сертификат открытого ключа и сопоставило его с личным ключом, можно при помощи команды **ipsec listcerts**.

```
server@server:~$ sudo ipsec listcerts

List of X.509 End Entity Certificates:
  subject:     "CN=BAS00001, C=BY, L=г.Минск, O=ЗАО "НТЦ Контакт", D=Комплекс программно-
аппаратный криптографической защиты информации "БАС". Сервер защиты"
  issuer:      "CN=УЦ для тестирования, C=BY, L=г.Минск, O=ЗАО 'НТЦ КОНТАКТ'"
  validity:    not before    Jan 1 00:00:00 2021, ok
               not after     Jan 1 00:00:00 2023, ok (expired in 365 days)
  serial:      01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  certificatePolicies:
               1.2.112.0.2.0.34.101.78.2.70
  authkeyId:   01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  sudjkeyId:   01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  pubkey:      BIGN 512 bits, has private key
  keyid:       01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  subjkey:     01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
```

О том, что программное обеспечение ПАК «БАС» верно подгрузило сертификат открытого ключа и сопоставило его с личным ключом, свидетельствует запись **pubkey: BIGN 512 bits, has private key**.

8 Формирование повторного запроса на получение сертификата открытого ключа

Сертификат открытого ключа имеет срок действия. В течении этого срока сертификат считается валидным и ПАК «БАС» может устанавливать защищенные соединения с другими ПАК «БАС». После истечения времени действия сертификата, ПАК «БАС», обладающий таким сертификатом, не сможет пройти аутентификацию у партнера и не сможет установить защищенное соединение. Поэтому необходимо поддерживать сертификат открытого ключа ПАК «БАС» в валидном состоянии.

Администратор должен следить за окончанием срока действия сертификата открытого ключа, чтобы не допускать обрыва защищенного соединения.

За некоторое время (зависит от регламента работы Удостоверяющего центра) до окончания срока действия сертификата следует выпустить повторный запрос на получение сертификата открытого ключа.

Для формирования повторного запроса на получение сертификата открытого ключа необходимо воспользоваться утилитой **RequestBuilder**. Она выполнит самотестирование ПАК «БАС», сгенерирует резервный личный ключ и сформирует запрос на получение сертификата открытого ключа.

```
server@server:~$ sudo RequestBuilder
[sudo] пароль для server:
> Контроль Целостности
Проверка целостности прошла успешно!
Результаты: /etc/support/IntegrityController.log
> Тестирование Библиотеки Криптопреобразований
    Зашифрование в режиме простой замены (|X| = 384) +
    Зашифрование в режиме простой замены (|X| = 376) +
    Расшифрование в режиме простой замены (|X| = 384) +
    Расшифрование в режиме простой замены (|X| = 288) +
    Зашифрование в режиме сцепления блоков (|X| = 384) +
    Зашифрование в режиме сцепления блоков (|X| = 288) +
    Расшифрование в режиме сцепления блоков (|X| = 384) +
    Расшифрование в режиме сцепления блоков (|X| = 288) +
    Зашифрование в режиме гаммирования с обратной связью +
    Расшифрование в режиме гаммирования с обратной связью +
    Шифрование в режиме счетчика +
    Выработка имитовставки (|X| = 104) +
```

```
  Выработка имитовставки (|X| = 384) +
  Установка защиты данных +
  Снятие защиты данных +
  Установка защиты ключа +
  Снятие защиты ключа +
  Хэширование (|X| = 104) +
  Хэширование (|X| = 256) +
  Хэширование (|X| = 384) +
  Преобразование ключа (m = 128) +
  Преобразование ключа (m = 192) +
  Преобразование ключа (m = 256) +
Тестирование алгоритмов СТБ.34.101.31 выполнено.
  Генерация пары ключей +
  Выработка электронной цифровой подписи +
  Проверка электронной цифровой подписи +
  Создание токена ключа +
  Разбор токена ключа +
  Извлечение пары ключей +
  Выработка идентификационной электронной цифровой подписи +
  Проверка идентификационной электронной цифровой подписи +
Тестирование алгоритмов СТБ 34.101.45 выполнено.
  Выработка имитовставки (алгоритм hmac-hbelt, keySize = 232) +
  Выработка имитовставки (алгоритм hmac-hbelt, keySize = 256) +
  Выработка имитовставки (алгоритм hmac-hbelt, keySize = 336) +
  Генерация псевдослучайных чисел (алгоритм brng-ctr-hbelt) +
  Генерация псевдослучайных чисел (алгоритм brng-hmac-hbelt) +
Тестирование алгоритмов СТБ 34.101.47 выполнено.
  Разделение секрета (l = 128) +
  Восстановление секрета (l = 128), подмножество пользователей {1,2} +
  Восстановление секрета (l = 128), подмножество пользователей {1,3} +
  Восстановление секрета (l = 128), подмножество пользователей {1,4} +
  Восстановление секрета (l = 128), подмножество пользователей {1,5} +
  Восстановление секрета (l = 128), подмножество пользователей {2,3} +
  Восстановление секрета (l = 128), подмножество пользователей {2,4} +
  Восстановление секрета (l = 128), подмножество пользователей {2,5} +
  Восстановление секрета (l = 128), подмножество пользователей {3,4} +
  Восстановление секрета (l = 128), подмножество пользователей {3,5} +
  Восстановление секрета (l = 128), подмножество пользователей {4,5} +
  Восстановление секрета (l = 128), подмножество пользователей {1,3,5} +
  Восстановление секрета (l = 128), подмножество пользователей {2,3,4,5} +
  Восстановление секрета (l = 128), подмножество пользователей {1,2,3,4,5} +
Тестирование алгоритмов СТБ 34.101.60 выполнено.
  Сеанс протокола BMQV +
  Сеанс протокола BSTS +
  Сеанс протокола BPACE +
  Сеанс протокола Диффи-Хеллмана +
Тестирование алгоритмов СТБ 34.101.66 выполнено.
  Зашифрование в режиме простой замены +
  Расшифрование в режиме простой замены +
  Зашифрование в режиме гаммирования с обратной связью +
```

```
        Расшифрование в режиме гаммирования с обратной связью +
        Шифрование в режиме гаммирования +
        Выработка имитовставки +
      Тестирование алгоритмов ГОСТ 28147-89 выполнено.
    Самотестирование библиотеки криптографических преобразований завершено успешно.
    > Контроль Работоспособности ПАК "Барьер-USB"
    Защищённое хранилище готово к работе.
    <13>Jan  1 12:00:00 basctl: Тестирование завершено успешно!
    Устройство инсталлировано. Желаете сгенерировать резервную ключевую пару и сформировать
запрос на получение сертификата? (Y/N): y
```

При необходимости можно отредактировать XML-файл с данными об устройстве, указав в нем серийный номер устройства, название организации и адрес, где эксплуатируется ПАК «БАС». Обычно это не требуется для выпуска повторного запроса, если данные устройства не менялись.

Для записи ключа в резервную область защищенного хранилища необходимо ввести пароль для получения доступа. При необходимости резервирования личного ключа задать пароль к ключевому контейнеру частичного секрета.

```
    Желаете отредактировать XML-файл с данными об устройстве? [/etc/support/PersonalData.xml]
(Y/N): n
    Действие отменено пользователем!
    Введите пароль доступа к защищённому хранилищу (8-24 символа): ********
    Подтвердите пароль: ********
    Сохранить личный ключ для возможности восстановления системы? (Y/N): y
    Задайте пароль доступа к контейнеру личного ключа (8-24 символа): ********
    Подтвердите пароль: ********
    Желаете удалить старые файлы запросов на получение сертификатов? (Y/N): y
    Запрос на получение сертификата открытого ключа успешно сохранен:
    [/etc/support/IssueRequestsAndCards/CertificateIssueRequest_0123456789101112.der]
    Карточка открытого ключа успешно сохранена:
    [/etc/support/IssueRequestsAndCards/PublicKeyCard_0123456789101112.rtf]
    Контейнер личного ключа успешно сохранён:
    [/etc/support/IssueRequestsAndCards/PrivateKeyContainer_reserve.pkc]
    Контейнер частичного секрета №1 успешно сохранён:
    [/etc/support/IssueRequestsAndCards/ShareSecretContainer_1_reserve.ssc]
    Контейнер частичного секрета №2 успешно сохранён:
    [/etc/support/IssueRequestsAndCards/ShareSecretContainer_2_reserve.ssc]
    server@server:~$
```

В результате выполнения утилиты **RequestBuilder**, в папке **/etc/support/IssueRequestsAndCards/** сформируются:

– повторный запрос на получение сертификата в соответствии с СТБ 34.101.17:

CertificateIssueRequest_0123456789101112.der

– карточка открытого ключа в соответствии с СТБ 34.101.49:

PublicKeyCard_0123456789101112.rtf

– контейнер резервного личного ключа в соответствии с СТБ 34.101.78:

PrivateKeyContainer_reserve.pkc

– контейнеры частичных секретов для восстановления резервного личного ключа в соответствии с СТБ 34.101.78:

ShareSecretContainer_1_reserve.ssc

ShareSecretContainer_2_reserve.ssc

Для выпуска сертификата открытого ключа необходимо экспортировать запрос на получение сертификата из ПАК «БАС» любым удобным способом и передать Администратору Удостоверяющего центра.

9 Замена ключевой пары

После того как новый сертификат открытого ключа был получен, необходимо выполнить импорт в ПАК «БАС»: сертификат устройства в папку **/usr/local/etc/ipsec.d/certs/**, корневой сертификат, а также промежуточные (при их наличии), в **/usr/local/etc/ipsec.d/cacerts/**, файлы списка отозванных сертификатов (при их наличии), в **/usr/local/etc/ipsec.d/crl/**.

Для замены личного ключа ПАК «БАС» необходимо воспользоваться утилитой **KeysReplacer**.

Утилита **KeysReplacer** выполнит самотестирование ПАК «БАС», очистит защищенное хранилище ПАК «Барьер – USB» и запишет резервный личный ключ в основную область защищенного хранилища. Если при этом в директориях, указанных в настроечном файле с именем **KeysReplacer.conf**, расположенном в директории «**/etc/support»**, будут найдены контейнеры личных ключей и частичных секретов, то основные будут уничтожены, а резервные переименованы, став основными.

Секция, соответствующая модулю смены ключей, в настроечном файле **KeysReplacer.conf** должна именоваться **KeysReplacer** и включать параметры:

– **PrivateKeysDirectory**. Значением параметра должен быть путь к директории контейнера основного личного ключа;

– **ShareSecretFileName_N**. N – порядковый номер основного частичного секрета. Значением параметра должен быть путь к директории, в которой хранится N-ый частичный секрет;

– **ReserveKeyword**. Значением параметра должно быть ключевое слово, которое содержится в названии файла контейнера резервного личного ключа.

Пример заполнения секции KeysReplacer:

```
[KeysReplacer]
        PrivateKeysDirectory = ./IssueRequestsAndCards/
        ShareSecretDirectory_1      = ./ShareSecret1/
        ShareSecretDirectory_2      = ./ ShareSecret2/
        ReserveKeyword              = _reserve
```

```
server@server:~$ sudo KeysReplacer
[sudo] пароль для server:
> Контроль Целостности
Проверка целостности прошла успешно!
Результаты: /etc/support/IntegrityController.log
> Тестирование Библиотеки Криптопреобразований
    Зашифрование в режиме простой замены (|X| = 384) +
    Зашифрование в режиме простой замены (|X| = 376) +
    Расшифрование в режиме простой замены (|X| = 384) +
    Расшифрование в режиме простой замены (|X| = 288) +
    Зашифрование в режиме сцепления блоков (|X| = 384) +
    Зашифрование в режиме сцепления блоков (|X| = 288) +
    Расшифрование в режиме сцепления блоков (|X| = 384) +
    Расшифрование в режиме сцепления блоков (|X| = 288) +
    Зашифрование в режиме гаммирования с обратной связью +
    Расшифрование в режиме гаммирования с обратной связью +
    Шифрование в режиме счетчика +
    Выработка имитовставки (|X| = 104) +
    Выработка имитовставки (|X| = 384) +
    Установка защиты данных +
    Снятие защиты данных +
    Установка защиты ключа +
    Снятие защиты ключа +
    Хэширование (|X| = 104) +
    Хэширование (|X| = 256) +
    Хэширование (|X| = 384) +
    Преобразование ключа (m = 128) +
    Преобразование ключа (m = 192) +
    Преобразование ключа (m = 256) +
  Тестирование алгоритмов СТБ.34.101.31 выполнено.
    Генерация пары ключей +
    Выработка электронной цифровой подписи +
    Проверка электронной цифровой подписи +
    Создание токена ключа +
    Разбор токена ключа +
    Извлечение пары ключей +
    Выработка идентификационной электронной цифровой подписи +
    Проверка идентификационной электронной цифровой подписи +
  Тестирование алгоритмов СТБ 34.101.45 выполнено.
    Выработка имитовставки (алгоритм hmac-hbelt, keySize = 232) +
    Выработка имитовставки (алгоритм hmac-hbelt, keySize = 256) +
    Выработка имитовставки (алгоритм hmac-hbelt, keySize = 336) +
    Генерация псевдослучайных чисел (алгоритм brng-ctr-hbelt) +
    Генерация псевдослучайных чисел (алгоритм brng-hmac-hbelt) +
  Тестирование алгоритмов СТБ 34.101.47 выполнено.
    Разделение секрета (l = 128) +
    Восстановление секрета (l = 128), подмножество пользователей {1,2} +
    Восстановление секрета (l = 128), подмножество пользователей {1,3} +
    Восстановление секрета (l = 128), подмножество пользователей {1,4} +
    Восстановление секрета (l = 128), подмножество пользователей {1,5} +
```

```
        Восстановление секрета (l = 128), подмножество пользователей {2,3} +
        Восстановление секрета (l = 128), подмножество пользователей {2,4} +
        Восстановление секрета (l = 128), подмножество пользователей {2,5} +
        Восстановление секрета (l = 128), подмножество пользователей {3,4} +
        Восстановление секрета (l = 128), подмножество пользователей {3,5} +
        Восстановление секрета (l = 128), подмножество пользователей {4,5} +
        Восстановление секрета (l = 128), подмножество пользователей {1,3,5} +
        Восстановление секрета (l = 128), подмножество пользователей {2,3,4,5} +
        Восстановление секрета (l = 128), подмножество пользователей {1,2,3,4,5} +
      Тестирование алгоритмов СТБ 34.101.60 выполнено.
        Сеанс протокола BMQV +
        Сеанс протокола BSTS +
        Сеанс протокола BPACE +
        Сеанс протокола Диффи-Хеллмана +
      Тестирование алгоритмов СТБ 34.101.66 выполнено.
        Зашифрование в режиме простой замены +
        Расшифрование в режиме простой замены +
        Зашифрование в режиме гаммирования с обратной связью +
        Расшифрование в режиме гаммирования с обратной связью +
        Шифрование в режиме гаммирования +
        Выработка имитовставки +
      Тестирование алгоритмов ГОСТ 28147-89 выполнено.
    Самотестирование библиотеки криптографических преобразований завершено успешно.
    > Контроль Работоспособности ПАК "Барьер-USB"
    Защищённое хранилище готово к работе.
    <13>Jan  1 12:00:00 basctl: Тестирование завершено успешно!
    Выполнение данной программы приведет к уничтожению текущего личного ключа и установке резервного в качестве основного!
    Вы уверены, что хотите продолжить? (Y/N): y
    Смена личного ключа должна сопровождаться заменой сертификата открытого ключа.
    Поместите сертификаты Удостоверяющих центров в директорию /usr/local/etc/ipsec.d/cacerts/.
    Поместите сертификат устройства в директорию /usr/local/etc/ipsec.d/certs/.
    Вы уверены, что хотите продолжить? (Y/N): y
    Введите пароль доступа к защищённому хранилищу (8-24 символа): ********
    Подтвердите пароль: ********
    Контейнер личного ключа:
    [/etc/support/IssueRequestsAndCards/PrivateKeyContainer.pkc]
    Контейнер частичного секрета:
    [/etc/support/IssueRequestsAndCards/ShareSecretContainer_1.ssc]
    Контейнер частичного секрета:
    [/etc/support/IssueRequestsAndCards/ShareSecretContainer_2.ssc]
    Смена личного ключа выполнена успешно.
    Замените имя сертификата устройства в файле ipsec.conf (при необходимости).
    Для начала работы с новыми ключами необходимо перезагрузить устройство либо выполнить команду «ipsec restart».
    server@server:~$
```

После того, как новый личный ключ установлен в качестве основного, а новые сертификаты разложены в нужные папки, необходимо выполнить настройки для того, чтобы ПАК «БАС» начал использовать новую ключевую пару при аутентификации в IPsec-соединении. Для этого необходимо заполнить параметр **leftcert** в настроечном файле **/usr/local/etc/ipsec.conf**, указав в нем имя нового файла сертификата открытого ключа (если оно изменилось) из папки **/usr/local/etc/ipsec.d/certs/**. Если новый сертификат имеет такое же имя, как и первоначальный, то при копировании в папку он заменит первоначальный, и изменение настроек не понадобится.

```
server@server:~$ sudo nano /usr/local/etc/ipsec.conf
...
leftcert = cert00001.cer
...
```

Для применения настроек необходимо перезапустить IPsec соединение.

```
server@server:~$ sudo ipsec restart
Stopping strongSwanCont IPsec...
Starting strongSwanCont 5.8.4 IPsec [starter]...
```

Убедиться в том, что программное обеспечение ПАК «БАС» подгрузило сертификат открытого ключа и сопоставило его с личным ключом, можно при помощи команды **ipsec listcerts**.

```
server@server:~$ sudo ipsec listcerts

List of X.509 End Entity Certificates:
  subject:     "CN=BAS00001, C=BY, L=г.Минск, O=ЗАО "НТЦ Контакт", D=Комплекс программно-
аппаратный криптографической защиты информации "БАС". Сервер защиты"
  issuer:      "CN=УЦ для тестирования, C=BY, L=г.Минск, O=ЗАО 'НТЦ КОНТАКТ'"
  validity:    not before    Jan 1 00:00:00 2021, ok
               not after     Jan 1 00:00:00 2023, ok (expired in 365 days)
  serial:      01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  certificatePolicies:
               1.2.112.0.2.0.34.101.78.2.70
  authkeyId:   01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  sudjkeyId:   01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  pubkey:      BIGN 512 bits, has private key
  keyid:       01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
  subjkey:     01:23:45:67:89:ab:cd:ef:00:01:23:45:67:89:ab:cd:ef:00
```

О том, что программное обеспечение ПАК «БАС» верно подгрузило сертификат открытого ключа и сопоставило его с личным ключом, свидетельствует запись **pubkey: BIGN 512 bits, has private key**.

О том, что сертификат новый, также должны свидетельствовать записи срока действия сертификата. Он должен начинаться в момент выпуска его Удостоверяющим центром и заканчиваться через продолжительное время (обычно от 1 до 3 лет).

10 Настройка восстановления личного ключа

ПАК «БАС» при помощи ПАК «Барьер – USB» ведет непрерывный контроль вскрытия корпуса. При обнаружении вскрытия корпуса уничтожается личный ключ, хранящийся в защищенном хранилище ПАК «Барьер – USB». Устройство при этом переходит в режим блокировки.

После обнаружения вскрытия корпуса личный ключ ПАК «БАС» считается скомпрометированным и не пригодным для дальнейшего использования. В таком случае необходимо воспользоваться процедурой возврата ПАК «БАС» к заводским настройкам, а затем заново сформировать ключевую информацию.

Однако, если вскрытие корпуса санкционированное (например, Администратором при обслуживании, или имеются другие причины, позволяющие Администратору принять решение, что личный ключ ПАК «БАС» не был скомпрометирован), то он может быть восстановлен из контейнера.

Для восстановления личного ключа требуется:

– контейнеры основного и резервного (при его наличии) личного ключа:

PrivateKeyContainer.pkc

PrivateKeyContainer_reserve.pkc

– основные и резервные (при его наличии) контейнеры частичных секретов в количестве, указанном в параметре **Threshold** файла с **RequestBuilder.conf**, при формировании запроса на выпуск сертификата открытого ключа:

ShareSecretContainer_1.ssc

ShareSecretContainer_2.ssc

ShareSecretContainer_1_reserve.ssc

ShareSecretContainer_2_reserve.ssc

Для этого Администраторы, владеющие частичными секретами, должны собраться в нужном количестве и импортировать свои частичные секреты в файловую систему ПАК «БАС».

Настройки по умолчания ПАК «БАС» позволяют выполнить процедуру восстановления личного ключа, предполагая, что необходимые для этого файлы

находятся в тех же местах и в том количестве, что заданы в настройках по умолчанию для генерации личного ключа. Однако у Администратора есть возможность выполнить дополнительные настройки.

Конфигурирование утилиты **PrivateKeyRecovery** осуществляется посредством редактирования настроечного файла. Настоечный файл представляет собой текстовый файл, разделённый на секции. Секции разделяются именем (секции), заключённым в прямоугольные скобки ([имя_секции]). Каждая секция может иметь одно или несколько полей вида «Параметр = Значение». Имена параметров предопределены для каждой секции.

По умолчанию приложение обращается к настроечному файлу с именем **PrivateKeyRecovery.conf**, расположенному в директории «**/etc/support**».

Секция, соответствующая модулю восстановления личного ключа, должна именоваться **PrivateKeyRecovery**. Данная секция включает следующие параметры:

– **PrivateKeyFileName**. Значением параметра должен быть путь к файлу контейнера основного личного ключа;

– **ShareSecretFileName_N**. N – порядковый номер основного частичного секрета. Значением параметра должен быть путь к файлу N-го частичного секрета;

– **PrivateKeyFileNameReserve**. Опциональный параметр. Значением параметра должен быть путь к файлу контейнера резервного личного ключа;

– **ShareSecretFileNameReserve_N**. Опциональный параметр. N – порядковый номер резервного частичного секрета. Значением параметра должен быть путь к файлу N-го частичного секрета.

Пример заполнения секции PrivateKeyRecovery:

```
[PrivateKeyRecovery]
PrivateKeyFileName          = /etc/support/IssueRequestsAndCards/PrivateKeyContainer.pkc
ShareSecretFileName_1       = ./IssueRequestsAndCards/ShareSecretContainer_1.ssc
ShareSecretFileName_2       = ./IssueRequestsAndCards/ShareSecretContainer_2.ssc
PrivateKeyFileNameReserve   = ./IssueRequestsAndCards/PrivateKeyContainer_reserve.pkc
ShareSecretFileNameReserve_1 = ./IssueRequestsAndCards/ShareSecretContainer_1_reserve.ssc
ShareSecretFileNameReserve_2 = ./IssueRequestsAndCards/ShareSecretContainer_2_reserve.ssc
```

11 Восстановление личного ключа

Для восстановления личного ключа ПАК «БАС» необходимо воспользоваться утилитой **PrivateKeyRecovery**.

Перед выполнением процедуры восстановления личного ключа необходимо:

– импортировать частичные секреты в ПАК «БАС» в количестве, указанном в параметре **Threshold** файла с **RequestBuilder.conf**, при формировании запроса на выпуск сертификата открытого ключа;

– выполнить настройки (при необходимости) утилиты **PrivateKeyRecovery**, указав пути к файлам контейнеров основного и резервного (при его наличии) личного ключа, а также частичных секретов, в фале **PrivateKeyRecovery.conf**, расположенному в директории «**/etc/support»**.

Утилита **PrivateKeyRecovery** обработает файлы контейнеров личного ключа и частичных секретов, пути к которым указаны в настроечном файле, и выполнит восстановление личного ключа в защищенное хранилище ПАК «Барьер – USB».

```
server@server:~$ sudo PrivateKeyRecovery
[sudo] пароль для server:
Сброс ПАК «Барьер-USB»:
Введите пароль доступа к защищённому хранилищу (8-24 символа): ********
Подтвердите пароль: ********
Операция завершена успешно.
Установка нового пароля ПАК «Барьер-USB»:
Введите пароль доступа к защищённому хранилищу (8-24 символа): ********
Подтвердите пароль: ********
Введите пароль к контейнеру личного ключа (8-24 символа): ********
Введите пароль к контейнеру резервного личного ключа (8-24 символа): ********
Личный ключ восстановлен из контейнера личного ключа:
[/etc/support/IssueRequestsAndCards/PrivateKeyContainer.pkc]
Резервный личный ключ восстановлен из контейнера личного ключа:
[/etc/support/IssueRequestsAndCards/PrivateKeyContainer_reserve.pkc]
Состояние защиты ПАК «Барьер-USB»: защита установлена.
server@server:~$
```

После успешного восстановления личного ключа в защищенное хранилище ПАК «Барьер – USB» файлы контейнеров частичных секретов должны быть удалены из файловой системы ПАК «Барьер – USB» при помощи команды **rm**, при этом остаться на носителях, распределенных между Администраторами.

12 Уничтожение личного ключа (возврат к заводским настройкам)

Для уничтожения личного ключа ПАК «БАС» необходимо воспользоваться утилитой **Uninstaller**.

Утилита **Uninstaller**, выполнит очистку защищенного хранилища ПАК «Барьер – USB» и снимет признак инсталляции ПАК «БАС» (выполнит возврат к заводским настройкам).

```
server@server:~$ sudo Uninstaller
[sudo] пароль для server:
Выполнение данной программы приведет к откату устройства к заводским настройкам!
Вы уверены, что хотите продолжить? (Y/N):
y
Введите пароль доступа к защищённому хранилищу (8-24 символа):
********
Подтвердите пароль:
********
Очистить хранилище сертификатов? (Y/N):
y
Файл
[/etc/support/IssueRequestsAndCards/CertificateIssueRequest_0123456789101112.der]
---> Удален
Файл [/etc/support/IssueRequestsAndCards/PublicKeyCard_0123456789101112.rtf]
---> Удален!
Файл [/etc/support/IssueRequestsAndCards/PrivateKeyContainer.pkc] ---> Удален!
Файл [/etc/support/IssueRequestsAndCards/ShareSecretContainer_1.ssc] ---> Удален!
Файл [/etc/support/IssueRequestsAndCards/ShareSecretContainer_2.ssc] ---> Удален!
Защищенное хранилище успешно очищено!
Хранилище сертификатов успешно очищено!
Очистка завершена! Рекомендуется перезагрузить устройство.
server@server:~$
```

После возврата ПАК «БАС» к заводским настройкам Администраторам необходимо уничтожить свои частичные секреты, связанные с данным устройством, хранящиеся на носителях вне ПАК «БАС».