КРИПТОКОМ

УТВЕРЖДЕН
СЕИУ.00009-01 34 02 - ЛУ

СРЕДСТВО КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ
МагПро КриптоПакет вер. 1.0

Программа создания закрытых ключей mkkey.
Руководство оператора

СЕИУ.00009-01 34 02
Листов 17

Литера О

2006

Инв.№ подл.	Подп. и дата	Взам. инв.№	Инв. № дубл.	Подп. и дата

Аннотация

Настоящий документ содержит руководство оператора для работы с программой mkkey из состава СКЗИ «МагПро КриптоПакет».

Содержание

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

1 НАЗНАЧЕНИЕ ПРОГРАММЫ

Программа mkkey из комплекта СКЗИ «МагПро КриптоПакет» выполняет создание закрытых ключей для использования в программном комплексе СКЗИ «МагПро КриптоПакет», в том числе создание ключей в аппаратных контейнерах, с использованием криптоядра МагПро.

Программа не формирует заявки на получение сертификата. Для формирования заявок необходимо использовать команду req утилиты openssl.

Программа также может создавать файл начального заполнения датчика случайных чисел, необходимый в случае использования программного ДСЧ.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

2 УСЛОВИЯ РАБОТЫ ПРОГРАММЫ

Программа mkkey предназначена для работы в ОС Windows 2000/XP/2003 Server/Vista и Unix-подобных системах Linux/FreeBSD/Solaris.

Программа mkkey может использовать как программные, так и аппаратные датчики случайных чисел.

К программным ДСЧ относятся клавиатурный (KEYBOARD) и YARROW.

Аппаратные ДСЧ входят в состав изделий «Аккорд» (ACCORD) и «Соболь» (SOBOL). Использование ДСЧ «Аккорд» и «Соболь» возможно в том случае, если соответствующие изделия заранее установлены в компьютере.

Набор возможных ДСЧ зависит от операционной системы, в которой работает программа. Выбор ДСЧ определяет тип созданного ключевого контейнера (контейнеров).

Таблица 1. Поддерживаемые ДСЧ и ключевые контейнеры:

Тип ДСЧ	**Операционные системы**	**Тип контейнера**
ACCORD	Windows, Linux	Touch Memory на устройстве ACCORD
SOBOL	Windows	Touch Memory на устройстве SOBOL
YARROW	Windows, Linux, FreeBSD, Solaris	Файлы PKCS#8
KEYBOARD	Windows, Linux, FreeBSD, Solaris	Файлы PKCS#8

Для инициализации клавиатурного ДСЧ, запроса пароля к ключевому контейнеру и сообщения пользователю о необходимых действиях с аппаратными устройствами программа использует полноэкранный текстовый интерфейс.

Задание режимов работы (например, параметры алгоритмов и набор генерируемых ключей) выполняется с помощью опций командной строки.

Ключевые контейнеры на аппаратных устройствах создаются и защищаются средствами криптоядра «МагПро».

Ключевые контейнеры в PKCS#8-файлах шифруются алгоритмом ГОСТ 28147-89 в режиме гаммирования с обратной связью, ключом, полученным из пароля с помощью алгоритма PBKDF PKCS-5 v.2. При выводе ключа из пароля используется HMAC ГОСТ Р 34.11-94.

В составе СКЗИ «МагПро КриптоПакет» для FreeBSD 4.x программа mkkey не поставляется, так как на этой платформе не поддерживаются никакие аппаратные контейнеры, а также не работает клавиатурный ДСЧ.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

3 ЗАПУСК ПРОГРАММЫ

3.1 Формат запуска программы

Формат запуска программы:

```
>mkkey [опции] [имя-контейнера]
```

Указание опций и имен контейнера является необязательным для запуска.

По умолчанию (если не указаны ни опции, ни имя контейнера) программа создает два ключа — подписи и обмена с алгоритмом ГОСТ Р 34.10-2001 с параметрами cryptocom и запрашивает пароль для защиты ключевого контейнера не менее чем из 6-ти символов; для создания ключей используется клавиатурный датчик; PEM-encoded PKCS#8 контейнер(ы) выводятся на стандартный вывод.

3.2 Опции

Таблица 2. Поддерживаемые опции

Опция	**Наименование опции**	**Описание опции и возможные значения**	**Примечание**	Раздел описания
-a	алгоритм создаваемого закрытого ключа	gost2001, gost94	Умолчание — gost2001. После 31 декабря 2007 года ключи подписи и шифрования для алгоритма gost94 создаваться не должны.	4.1
-d [device-id]	идентификатор устройства			4.8
-e	Использовать существующий аппаратный контейнер	Необходимо указать имя контейнера. Если в устройстве обнаружен контейнер с соответствующим именем, программа записывает ключ в него. Если в устройстве обнаружен контейнер с несоответствующим именем, программа запрашивает другой аппаратный контейнер.	Если эта опция не указана, программа создает новый ключевой контейнер.	4.8
-f	перезапись существующего контейнера	Если на аппаратном устройстве обнаружен существующий контейнер, перезаписать его.	Если эта опция не указана, то при обнаружении существующего контейнера на указанном устройстве программа завершается с ошибкой.	4.8

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

Опция	Наименование опции	Описание опции и возможные значения	Примечание	Раздел описания
-n	не запрашивать пароль	При указании этой опции программа не запрашивает пароль, и создаются незашифрованные контейнеры.	По умолчанию программа требует пароль не короче 6 символов.	4.6
-r [имя-файла]	имя файла начального заполнения ДСЧ	Опция предоставляет возможность указать файл, содержащий начальное заполнение инициализируемого клавиатурного/программного ДСЧ	Игнорируется, если используется аппаратный ДСЧ.	4.3
-t	тип ДСЧ, устройства и типа контейнера	Определяет используемый ДСЧ, тип контейнера и устройство, в которое будут записаны ключевые контейнеры. Допустимые значения: KEYBOARD, YARROW — создаются PKCS#8-контейнеры на диске ACCORD, SOBOL — создается аппаратный ключевой контейнер на соответствующем устройстве	Умолчание — значение переменной среды RNG или, если она не установлена, YARROW	4.2, 4.7, 4.8
-s	набор параметров ключа подписи	Допустимые значения в зависимости от алгоритма: gost2001 — A, B, C gost94 — A, B, C, D	Выбор параметров следует уточнить в удостоверяющем центре	4.4
-x	набор параметров ключа обмена	Допустимые значения в зависимости от алгоритма: gost2001 — XA, XB gost94 — XA, XB, XC	Выбор параметров следует уточнить в удостоверяющем центре	4.4

Если при запуске программы указывается неизвестная опция, программа выводит формат запуска и список опций и заканчивает работу.

3.3 Имя контейнера

Если имя контейнера не указано, на аппаратных устройствах создается безымянный контейнер, а PEM-encoded PKCS#8 контейнер(ы) выводятся на стандартный вывод. Программа позволяет переназначить стандартный вывод без ущерба для функционирования полноэкранного интерфейса.

Если имя контейнера указано, и создаются два ключа в PKCS#8-контейнерах (каждый из которых должен быть помещен в отдельный контейнер), то созданные файлы будут называться `[имя контейнера]_sign.key` и `[имя контейнера]_xchg.key`. Если создается только один ключ, то он будет помещен в файл `[имя контейнера].key`.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

4 ВЫПОЛНЕНИЕ ПРОГРАММЫ

4.1 Выбор алгоритма

Программа mkkey предоставляет возможность создания ключей подписи и обмена для алгоритмов:

- по ГОСТ Р 34.10-94 (gost94)
- по ГОСТ Р 34.10-2001 (gost2001)

Для выбора алгоритма необходимо при запуске программы указать опцию -a с кратким обозначением алгоритма в качестве значения этой опции.

Пример:
`mkkey -agost94`
В этом случае будут созданы закрытые ключи для алгоритма gost94.

Перед созданием ключей следует уточнить в удостоверяющем центре, какой из алгоритмов следует выбирать.

По умолчанию создаются ключи для алгоритма gost2001.

После 31 декабря 2007 года ключи подписи и шифрования для алгоритма gost94 создаваться не должны.

4.2 Выбор ДСЧ

Выбор ДСЧ осуществляется с помощью параметра -t.

В качестве значения параметра указывается, какой ДСЧ следует использовать при создании ключей: программный (KEYBOARD или YARROW) или аппаратный (ACCORD или SOBOL).

По умолчанию используется значение переменной среды RNG или, если эта переменная не определена, то YARROW.

Внимание. С выбором ДСЧ однозначно связан выбор устройства, в котором будет сохранен ключевой контейнер (см. п. 4.7 и 4.8.)

4.3 Выбор файла заполнения ДСЧ

При создании ключей с использованием клавиатурного ДСЧ или ДСЧ Yarrow программа выполняет инициализацию файла начального заполнения ДСЧ, что позволяет дальнейшую эксплуатацию OpenSSL с выбранным ДСЧ.

Если в момент запуска программы с клавиатурным ДСЧ определена переменная среды RNG_PARAMS, и при запуске не указана опция -r, то содержимое переменной RNG_PARAMS считается именем файла заполнения.

Если переменная RNG_PARAMS не определена, то создается файл с умолчательным именем:

В ОС Windows — `%APPDATA%\MagProSSL\random_seed`

В POSIX-системах — `$HOME/.magprossl/random_seed`

В случае использования клавиатурного ДСЧ или ДСЧ Yarrow также возможно явное указание имени файла заполнения. Имя файла заполнения указывается в качестве значения параметра -r. В этом случае содержимое переменной RNG_PARAMS игнорируется.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

При использовании аппаратных ДСЧ файл начального заполнения создается только при явном указании его имени при помощи параметра -r.

4.4 Состав набора ключей. Параметры ключей

Состав создаваемого набора ключей указывается с помощью опция -x и -s.

Если не указана ни одна из этих опций, создаются два ключа — подписи и обмена, с умолчательными параметрами (A и XA).

Если указана только одна из этих опций, создается только один ключ, соответствующий указанной опции.

Если указаны обе опции, создаются два ключа с указанными параметрами.

Какие параметры ключей необходимо указывать, следует уточнить в удостоверяющем центре.

4.5 Инициализация ДСЧ

После запуска программы происходит инициализация указанного ДСЧ.

В случае использования ДСЧ Yarrow или аппаратного ДСЧ эта операция происходит незаметно для пользователя, и программа сразу же переходит к запросу пароля (если не указана опция -n) и созданию ключей.

В случае использования клавиатурного датчика (явно указанного или по умолчанию) программа выводит полноэкранное изображение клавиатуры и предлагает нажимать указанные клавиши (с учетом клавиши SHIFT).

Внимание! При инициализации использование русских букв не допускается!

В нижней части окна выводится прогресс-бар, указывающий степень завершения процесса инициализации. При нажатии корректной клавиши степень завершения процесса увеличивается, при нажатии некорректной клавиши — уменьшается.

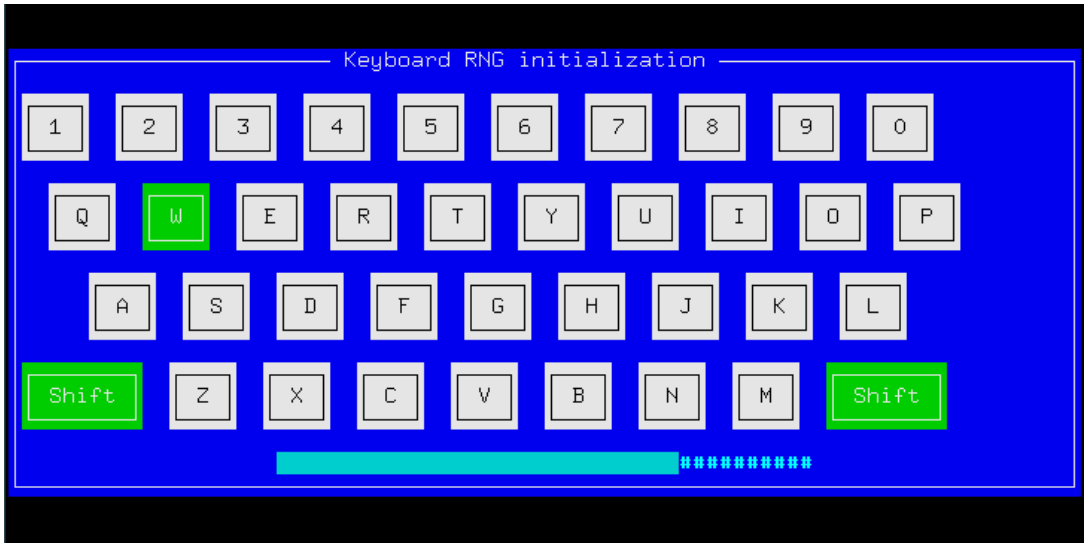

Рис. 1. Инициализация клавиатурного ДСЧ

Возможно использование черно-белого терминала.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

4.6 Ввод пароля

Для отказа от зашифрования ключевых контейнеров на пароле необходимо при запуске программы указать опцию -n.

Если опция -n при запуске программы не была указана, после завершения инициализации ДСЧ программа требует дважды ввести пароль для защиты ключевых контейнеров длиной от 6-ти символов до 32 символов (для перехода из поля в поле можно использовать клавишу TAB или ENTER):

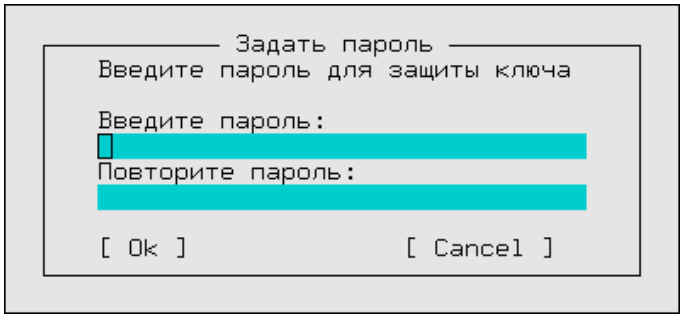

Рис. 2. Запрос ввода пароля

После ввода пароля необходимо выбрать OK, нажав клавиши TAB и ENTER или дважды ENTER. Для отмены ввода пароля (в этом случае ключи не создаются, выполнение программы завершается) необходимо выбрать CANCEL, нажав два раза клавишу TAB и один раз клавишу ENTER, или нажать клавишу ESC.

В пароле могут быть использованы как символы ASCII, так и русские буквы. Русские буквы интерпретируются как имеющие кодировку UTF-8, что соответствует рекомендациям PKCS#5. Следует обратить внимание, что OpenSSL и большая часть приложений, её использующих, не производит никаких преобразований кодировки пароля. Поэтому ключи, защищенные паролем, состоящим из русских букв, могут быть использованы с командно-строчной утилитой openssl и большинством приложений библиотеки OpenSSL только при запуске последних в локали UTF-8.

4.7 Создание PKCS#8-контейнеров на диске

PKCS#8-контейнеры (предназначенные для записи в файл на диске) создаются, если программе при запуске не задано никаких опций (по умолчанию) или задана опция -t со значениями KEYBOARD или YARROW.

По умолчанию (без указания опций и имени контейнера) программа создает два ключа — ключ подписи и ключ обмена, зашифровывает их на указанном пароле и выводит ключевые контейнеры в стандартный вывод (по умолчанию — на терминал).

Рис. 3. Вывод ключевых контейнеров на терминал

Программа позволяет переназначить стандартный вывод без ущерба для функционирования полноэкранного интерфейса.

При сохранении полученных ключевых контейнеров в файлы необходимо переносить в файл содержимое контейнера вместе со строками BEGIN ENCRYPTED PRIVATE KEY и END ENCRYPTED PRIVATE KEY.

Если имя контейнера указано, то:

- если создаются два ключа в PKCS#8-контейнерах (каждый из которых должен быть помещен в отдельный контейнер), то созданные файлы будут называться `[имя контейнера]_sign.key` и `[имя контейнера]_xchg.key`;
- если создается только один ключ, то он будет помещен в файл `[имя контейнера].key`.

4.8 Создание контейнеров на аппаратном носителе

4.8.1 Выбор аппаратного носителя

Ключевые контейнеры на аппаратных носителях создаются в тех случаях, если указана опция -t со значением ACCORD или SOBOL. Запись ключевых контейнеров производится на (или через) соответствующее устройство.

4.8.2 Сообщения и действия при необнаружении аппаратного устройства

Если указанное в опции -t аппаратное устройство не найдено, программа выводит сообщение об этом.

Если указано и не найдено устройство «Аккорд» или «Соболь», программа сообщает об ошибке и завершает работу. Это может означать, что указанное устройство либо неисправно, либо не установлено на компьютере.

Если указано устройство «Аккорд» или «Соболь», но не обнаружено устройство Touch Memory, программа предлагает подключить указанное устройство либо прервать операцию создания ключей с помощью клавиши ESC. В этом случае следует, не прерывая работу программы, прислонить устройство Touch Memory к считывателю. Программа самостоятельно обнаружит устройство, создаст на нем ключевой контейнер и завершит работу.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

4.8.3 Имена аппаратных контейнеров

Если имя контейнера не указано, на аппаратном устройстве будет создан безымянный ключевой контейнер, содержащий один или два ключа в зависимости от указанных опций (по умолчанию — два ключа).

При указании имени контейнера на аппаратном устройстве будет создан ключевой контейнер с указанием имени, содержащий один или два ключа в зависимости от указанных опций (по умолчанию — два ключа).

4.8.4 Запись в существующий аппаратный контейнер

Если ключевой контейнер на аппаратном устройстве содержит один закрытый ключ (подписи или обмена), программа mkkey дает возможность создать еще один закрытый ключ (соответственно обмена или подписи) и сохранить его в том же контейнере. Для этого следует указать опцию -e с указанием имени контейнера.

Если в аппаратном устройстве уже существует ключевой контейнер, то при попытке записать в это же устройство новый контейнер (т.е. при запуске программы без указания опции -e) программа сообщит об ошибке, и запись произведена не будет. Для перезаписи ключевого контейнера необходимо использовать опцию -f.

5 СОЗДАНИЕ ЗАЯВОК НА ПОЛУЧЕНИЕ СЕРТИФИКАТА

5.1 Краткие сведения о команде *req* утилиты openssl

Программа mkkey создает только закрытый ключ.

Для получения сертификата на соответствующий открытый ключ необходимо сформировать заявку на ключ с помощью команды *req* утилиты openssl.

Внимание. На каждый ключ создается отдельная заявка.

Команда *req* позволяет вводить информацию, которая должна содержаться в заявке, с клавиатуры, либо считывать её из конфигурационного файла.

Если в полях сертификата должны присутствовать русские буквы, вводить информацию с клавиатуры следует в кодировке utf-8.

Рекомендуется использовать считывание с конфигурационного файла.

Формат вызова команды *req* при использовании PKCS#8-контейнеров (содержащихся в файлах):

```
>openssl req -new -config файл-конфигурации  -key файл-pkcs8\
  -out имя-файла-заявки
```

Формат вызова команды *req* при использовании аппаратного контейнера:

```
 openssl req -new -config файл-конфигурации -key спецификация-ключа \
  -keyform ENGINE -engine cryptocom -out имя-файла-заявки
```

Спецификация ключа, хранящегося в аппаратном контейнере, имеет вид:

```
 <устройство>[:<имя-контейнера>][=идентификатор].(S|X)
```

После двоеточия указывается имя контейнера, после знака равенства указывается идентификатор устройства, S|X — указание, на какой ключ создается заявка (S — на ключ подписи, X — на ключ обмена).

Указание имени устройства и имени контейнера необязательно.

Более подробная информация по ключам команды *req* приведена в соответствующей странице руководства по программе openssl (см. документ «Средство криптографической защиты информации «МагПро КриптоПакет» вер. 1.0. Утилита openssl. Руководство оператора» СЕИУ.00009-01 34 01).

5.2 Формат конфигурационного файла команды openssl req

Конфигурационный файл команды openssl req состоит из пар имя=значение, каждая из которых размещается в отдельной строке. Файл разделен на секции, начинающиеся с заголовка, представляющего собой имя в квадратных скобках. То что находится в конфигурационном файле до первого заголовка, называется глобальной секцией.

По формату конфигурационный файл команды openssl req соответствует конфигурационному файлу СКЗИ «МагПро КриптоПакет», подробно описанному в документе «Средство криптографической защиты информации «МагПро КриптоПакет» вер. 1.0. Руководство системного администратора.» СЕИУ.00009-01 33.

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

Параметры заявки на сертификат, используемые командой *req*, размещаются в секции с именем `req`. Поля *distinguished name* и расширения *X509v3* описываются в отдельных секциях, имена которых указываются соответственно в переменных `distinguished_name` и `x509_extensions` секции `req`.

Если файл конфигурации в значениях каких-либо полей содержит русские буквы, такой файл должен быть в кодировке utf-8.

Пример файла конфигурации команды req для сертификата ключа электронной подписи:

```
openssl_conf = openssl_def
[openssl_def]
engines = engine_section
[engine_section]
cryptocom = cryptocom_section
[cryptocom_section]
dynamic_path = $ENV::ENGINE_DIR/cryptocom.so
engine_id = cryptocom
default_algorithms = ALL
[req]
; Неинтерактивный режим
prompt=no
; Название секции где описаны поля Distinguished Name заявки
distinguished_name = req_dn
; Название секции где описаны атрибуты заявки
attributes = req_attributes
; Название секции где описаны расширения
x509_extensions = req_ext
; Указывает что файл конфигурации содержит символы, отличные от ASCII (русские буквы
utf8 = yes
; Способ упаковки русских букв в заявке. Рекомендуемые значения
; pkix или utf8only
string_mask = pkix
[ req_dn ]
; Подразделение организации
OU=Группа тестирования
; Местонахождения
L=Москва
; Имя пользователя
CN=Алиса Селезнева
emailAddress=alice@cryptocom.ru
; Организация
O=ООО Криптоком
; Страна. Используется двухбуквенный код ISO.
C=RU
[ req_ext ]
; Данный ключ не может использоваться как ключ удостоверяющего центра
basicConstraints=CA:FALSE
; Области использования ключа
keyUsage = digitalSignature, nonRepudiation
; Расширенная информация об области использования ключа
extendedKeyUsage = emailProtection
; Альтернативное наименование владельца сертификата (в данном случае адрес Jabber)
```

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

```
subjectAltName=im:alice@jabber.cryptocom.ru
[ req_attributes ]
```

Сертификат для ключей обмена ключами для шифрования электронной почты должен иметь расширение *keyUsage*, содержащее значение `keyEncipherment`.

В качестве значения расширения *keyUsage* допускается использование конструкции `digitalSignature, nonRepudiation, keyEncipherment`.

У серверного сертификата для TLS-сервера поле *CommonName (CN)* должно быть равно имени этого сервера.

Серверный сертификат для TLS-сервера должен содержать расширение *extendedKeyUsage*, равное `serverAuth`.

Клиентский сертификат для аутентификации TLS-клиента должен содержать расширение *extendedKeyUsage*, равное `clientAuth`.

Допускается использование нескольких значений расширения *extendedKeyUsage*, например

```
extendedKeyUsage=clientAuth, emailProtection
```

Более подробную информацию можно получить в ман-страницах *req* и *config* документации на OpenSSL.

6 СООБЩЕНИЯ ОПЕРАТОРУ

Сообщение	**Причина**	**Действия оператора**
Usage: mkkey [options] [container] (+ список опций)	Указана некорректная опция	Перезапустить программу с корректными опциями
Доступ к устройству: аппаратное устройство не найдено	В качестве значения опции -t указано устройство «Аккорд» или «Соболь», не установленное на компьютере	Указать установленное на компьютере устройство или воспользоваться программным ДСЧ
	Устройство «Аккорд» или «Соболь», указанное в качестве значения опции -t, неисправно	Обратиться к администратору для ремонта или замены устройства или воспользоваться программным ДСЧ
Нет отклика от устройства. Вставьте ключевой носитель. Для прерывания операции нажмите Esc	При указании ACCORD или SOBOL в качестве значения опции -t не обнаружено устройство Touch Memory	Прислонить устройство Touch Memory к считывателю

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц) в докум.	№ документа	Входящий сопроводительного докум. и дата	Подпись	Дата
Изм.	измененных	замененных	новых	аннулированных					

Порядковый № изменения	Подпись лица, ответственного за изменение	Дата внесения изменения