

УТВЕРЖДЕН  
СЕИУ.00009-01 34 01 - ЛУ

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

СРЕДСТВО КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ  
MagПро КриптоПакет вер. 1.0

**Утилита OpenSSL.  
Руководство оператора**

СЕИУ.00009-01 34 01  
Листов 79

Литера О

## Аннотация

Настоящий документ содержит руководство оператора по работе с утилитой openssl из комплекта СКЗИ «МагПро КриптоПакет» при использовании российских алгоритмов.

Авторские права на СКЗИ «МагПро КриптоПакет» принадлежат ООО «Криптоком».

В СКЗИ использован код OpenSSL, ©1998-2004, The OpenSSL Project.

«МагПро» является зарегистрированной торговой маркой ООО «Криптоком».

# Содержание

<b>1</b>	<b>УТИЛИТА OPENSSL</b>	<b>6</b>
1.1	Описание утилиты . . . . .	6
1.2	Формат вызова утилиты . . . . .	6
1.3	Описание команд . . . . .	6
1.4	Стандартные команды . . . . .	7
1.5	Ключи на аппаратных носителях . . . . .	7
1.6	Пароли как аргументы . . . . .	8
<b>2</b>	<b>ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ ГОСТ</b>	<b>9</b>
<b>3</b>	<b>КОМАНДА SA</b>	<b>10</b>
3.1	Описание . . . . .	10
3.2	Формат ввода команды . . . . .	10
3.3	Опции команды . . . . .	10
3.3.1	Опции обработки заявок и выпуска сертификатов . . . . .	10
3.3.2	Опции работы со списками отзыва сертификатов (CRL) . . . . .	12
3.3.3	Опции конфигурационного файла . . . . .	13
3.4	Формат раздела политики . . . . .	16
3.5	Формат SPKAC . . . . .	16
3.6	Примеры . . . . .	16
3.7	Файлы . . . . .	18
3.8	Переменные среды . . . . .	18
3.9	Ограничения . . . . .	18
3.10	Предупреждения . . . . .	18
<b>4</b>	<b>КОМАНДА CRL</b>	<b>20</b>
4.1	Описание команды . . . . .	20
4.2	Формат ввода команды . . . . .	20
4.3	Опции команды . . . . .	20
4.4	Примечания . . . . .	20
4.5	Примеры . . . . .	21
<b>5</b>	<b>КОМАНДА CRL2PKCS7</b>	<b>22</b>
5.1	Описание команды . . . . .	22
5.2	Формат ввода команды . . . . .	22
5.3	Опции команды . . . . .	22
5.4	Примеры . . . . .	22
5.5	Примечания . . . . .	23
<b>6</b>	<b>КОМАНДА DGST</b>	<b>24</b>
6.1	Описание команды . . . . .	24
6.2	Формат ввода команды . . . . .	24
6.3	Опции команды . . . . .	24

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

<b>7</b>	<b>КОМАНДА ENC</b>	<b>25</b>
7.1	Описание команды . . . . .	25
7.2	Формат вызова команды . . . . .	25
7.3	Опции команды . . . . .	25
7.4	Примечания . . . . .	26
7.4.1	Примеры . . . . .	26
<b>8</b>	<b>КОМАНДА OCSP</b>	<b>28</b>
8.1	Описание . . . . .	28
8.2	Формат ввода команды . . . . .	28
8.3	Опции команды . . . . .	28
8.3.1	Клиентские опции команды ocsp . . . . .	28
8.3.2	Серверные опции команды ocsp . . . . .	31
8.4	Проверка OCSP-ответов . . . . .	31
8.5	Примечания . . . . .	32
8.6	Примеры . . . . .	32
<b>9</b>	<b>КОМАНДА PKCS7</b>	<b>34</b>
9.1	Описание команды . . . . .	34
9.2	Формат ввода команды . . . . .	34
9.3	Опции команды . . . . .	34
9.4	Примеры . . . . .	34
9.5	Примечания . . . . .	34
9.6	Ограничения . . . . .	35
<b>10</b>	<b>КОМАНДА PKCS8</b>	<b>36</b>
10.1	Описание команды . . . . .	36
10.2	Формат ввода команды . . . . .	36
10.3	Опции команды . . . . .	36
10.4	Примечания . . . . .	37
10.5	Примеры . . . . .	38
<b>11</b>	<b>КОМАНДА REQ</b>	<b>39</b>
11.1	Описание команды . . . . .	39
11.2	Формат ввода команды . . . . .	39
11.3	Опции команды . . . . .	39
11.4	Формат конфигурационного файла . . . . .	42
11.5	Формат разделов конфигурационного файла distinguished name и attribute . . . . .	44
11.6	Примеры . . . . .	44
11.7	Примечания . . . . .	46
11.8	Диагностика . . . . .	47
11.9	Переменные среды . . . . .	47
<b>12</b>	<b>КОМАНДА SMIME</b>	<b>48</b>
12.1	Описание команды . . . . .	48
12.2	Формат ввода команды . . . . .	48
12.3	Опции команды . . . . .	48
12.4	Примечания . . . . .	51

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

12.5	Коды выхода . . . . .	51
12.6	Примеры . . . . .	51
<b>13</b>	<b>КОМАНДА S_CLIENT</b>	<b>53</b>
13.1	Описание команды . . . . .	53
13.2	Формат ввода команды . . . . .	53
13.3	Опции команды . . . . .	53
13.4	Команды, выводимые при установленном соединении . . . . .	55
13.5	Примечания . . . . .	56
<b>14</b>	<b>КОМАНДА S_SERVER</b>	<b>57</b>
14.1	Описание команды . . . . .	57
14.2	Формат ввода команды . . . . .	57
14.3	Опции команды . . . . .	57
14.4	Команды, используемые при установленном соединении . . . . .	60
14.5	Примечания . . . . .	60
<b>15</b>	<b>КОМАНДА VERIFY</b>	<b>61</b>
15.1	Описание команды . . . . .	61
15.2	Формат ввода команды . . . . .	61
15.3	Опции команды . . . . .	61
15.4	Операция проверки . . . . .	62
15.5	Диагностика . . . . .	63
<b>16</b>	<b>КОМАНДА X509</b>	<b>68</b>
16.1	Описание команды . . . . .	68
16.2	Формат ввода команды . . . . .	68
16.3	Описание опций . . . . .	68
16.3.1	Опции ввода, вывода и общего назначения . . . . .	68
16.3.2	Опции просмотра сертификатов . . . . .	69
16.3.3	Настройки доверия . . . . .	70
16.3.4	Опции подписания сертификатов . . . . .	71
16.3.5	Опции именованя . . . . .	72
16.3.6	Опции текста . . . . .	74
16.4	Примеры . . . . .	75
16.5	Примечания . . . . .	76
16.6	Расширения сертификатов . . . . .	76

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

# 1 УТИЛИТА OPENSSL

## 1.1 Описание утилиты

OpenSSL — криптографическая библиотека, реализующая сетевые протоколы Secure Sockets Layer (SSL v2/v3) и Transport Layer Security (TLS v1) и соответствующие криптографические стандарты, необходимые для работы с этими протоколами.

Программа openssl — командно-строчная утилита для использования различных криптографических функций криптобиблиотеки OpenSSL из командной оболочки. С ее помощью можно:

- Создавать сертификаты формата X.509, заявки на выдачу сертификатов и списки отзыва.
- Производить вычисление хэш-сумм.
- Производить зашифрование и расшифрование с помощью симметричных алгоритмов шифрования.
- Выполнять тестирование SSL/TLS серверов и клиентов.
- Работать с подписанными и зашифрованными почтовыми сообщениями формата S/MIME.

## 1.2 Формат вызова утилиты

openssl команда [опции команды] [аргументы команды] — общий формат вызова утилиты

openssl [list-standard-commands] — формат вызова команды, выводящий список стандартных команд

openssl по-XXX [необязательные опции] — проверка существования команды

## 1.3 Описание команд

Утилита openssl предоставляет широкий выбор команд (см. выше употребление понятия «команда» в формате вызова утилиты), многие из которых используются с различными опциями и аргументами (см. выше «опции команды» и «аргументы команды»).

Псевдокоманда list-standard-commands выводит список (по одной в строке) названий всех стандартных команд.

Псевдокоманда по-XXX проверяет, доступна ли указанная команда (вместо XXX указывается название команды). Если команды с указанным именем не существует, команда по-XXX возвращает 0 (успех) и выводит по-XXX; в противном случае она возвращает 1 и выводит XXX. В обоих случаях результат направляется в стандартный вывод и ничего не выводится в stderr. Дополнительные командно-строчные аргументы всегда игнорируются.

Команда по-XXX не может определить доступность псевдокоманд, таких как quit, а также самой команды по-XXX.

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

## 1.4 Стандартные команды

Команда	Описание
asn1parse	Разбор ASN.1-структур, понимаемых библиотекой
base64	Кодирование и декодирование кодировки base64
ca	Управление удостоверяющим центром
ciphers	Вывод списка доступных шифр-сьютов SSL/TLS
crl	Обработка списком отзыва сертификатов (CRL)
crl2pkcs7	Создание специального сообщения формата PKCS#7 из CRL и списка сертификатов
dgst	Вычисление хэш-суммы
enc	Шифрование с помощью симметричных алгоритмов шифрования.
errstr	Расшифровка кода ошибки
passwd	Генерация хэшированных паролей
pkcs12	Работа с форматом PKCS#12
pkcs7	Работа с форматом PKCS#7
rand	Генерация псевдослучайных байтов
req	Работа с заявкой на получение сертификата формата X.509 (CSR)
s_client	Реализация SSL/TLS-клиента общего назначения, который может установить прозрачное соединение с удаленным сервером, работающим по протоколу SSL/TLS. Клиент предназначен только для тестовых целей и предоставляет только простейший интерфейс с рудиментарной функциональностью, хотя внутри себя он использует практически всю функциональность библиотеки OpenSSL.
s_server	Реализация SSL/TLS-сервера общего назначения, с которым могут быть установлены соединения удаленными клиентами, работающими по протоколу SSL/TLS. Сервер предназначен только для тестовых целей и предоставляет только простейший интерфейс с рудиментарной функциональностью, хотя внутри себя он использует практически всю функциональность библиотеки OpenSSL. Он предоставляет как свой собственный, ориентированный на работу в командной строке протокол для тестирования функций SSL, так и простейший http-сервер.
s_time	Измеритель производительности SSL
sess_id	Анализ сохраненных SSL-сессий
smime	Обработка почтовых сообщений формата S/MIME
speed	Определение скорости работы алгоритма
verify	Проверка корректности сертификатов формата X.509
version	Вывод информации о версии библиотеки OpenSSL
x509	Работа с сертификатами формата X.509

## 1.5 Ключи на аппаратных носителях

Пользователь утилиты openssl может воспользоваться ключами, находящимися на аппаратных носителях «Аккорд» или «Вьюга», если используемая команда имеет опции key и keyform. В этом случае в качестве значения опции keyform указывается engine, в качестве значения опции engine указывается cryptocom, а значение опции key строится следующим образом.

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

Вместо ключевого файла указывается строка-идентификатор, имеющая следующий формат: *ТИП-УСТРОЙСТВА[=ID][:имя-контейнера].{X|S}*

Здесь ТИП-УСТРОЙСТВА может быть ACCORD или VJUGA, ID — специфичный для устройства аппаратный идентификатор, имя-контейнера - имя ключевого контейнера, заданного при его создании (обязательно должен быть указан либо ID, либо ключевой контейнер), X или S — идентификатор одного из двух ключей в контейнере. X — ключ обмена ключами, S — ключ подписи.

Попытка обращения к ключу с именем контейнера, отличным от того, который имеется на доступном в данный момент устройстве, приводит к ошибочному завершению операции.

## 1.6 Пароли как аргументы

Некоторые команды принимают пароли в качестве аргументов, как правило, используя для входного и выходного паролей соответственно опции `-passin` и `-passout`. Эти опции позволяют получать пароли из различных источников. Каждая из этих опций принимает один аргумент, формат которого показан ниже. Если аргумент не указан, а пароль запрашивается, пользователю предлагается ввести пароль: как правило, такой пароль вводится с текущего терминала без вывода на экран.

Формат аргумента	Описание
<code>pass:пароль</code>	Пароль указывается явным образом. Поскольку такой пароль видят утилиты (например утилита <code>ps</code> под Unix-подобные ОС), этот способ ввода пароля следует использовать только в том случае, если безопасность не важна.
<code>env:var</code>	Считать пароль из переменной среды <code>var</code> . Поскольку среда других процессов на некоторых платформах видна (например, <code>ps</code> под некоторыми Unix-подобными ОС), этот способ ввода пароля следует использовать с осторожностью.
<code>file:pathname</code>	Первая строка файла с именем <code>pathname</code> является паролем. Если одно и то же имя файла указывается в качестве аргумента опций <code>-passin</code> и <code>-passout</code> , то для входного пароля будет использована первая строка файла, а для выходного — вторая. Необязательно указывать именно на файл: можно, например, указывать на устройство или именованный канал.
<code>fd:number</code>	Прочитать пароль из открытого файла текущего процесса, заданного числовым дескриптором <code>number</code> . Это можно использовать, например, чтобы передать данные по неименованному каналу.
<code>stdin</code>	Прочитать пароль со стандартного ввода.

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

## 2 ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ ГОСТ

В большинстве случаев при использовании ключей СКЗИ «МагПро КриптоПакет» в приложениях явное указание алгоритма не нужно — используемый алгоритм автоматически определяется на основе используемого ключа.

Необходимо явным образом указывать алгоритмы при использовании команд:

DGST (см. раздел 6) — для использования алгоритма хэширования по ГОСТ Р 34.11-94 всегда необходимо указывать параметр `-md_gost94`;

ENC (см. раздел 7) — при симметричном шифровании всегда обязательно используется параметр `-gost89` для выбора алгоритма шифрования ГОСТ 28147-89;

OCSF (см. раздел 8) — для проверки статусов сертификатов при определении алгоритма дайджеста с помощью опции `-digest` необходимо указывать опцию `-md_gost94`;

REQ (см. раздел 11) — если использовать эту команду для создания ключей, необходимо явным образом указывать опцию `-gost2001`: с нужным набором параметров.

SMIME (см. раздел 12) — при использовании опции `encrypt` симметричного шифрования;

X509 (см. раздел 16) — при использовании алгоритма хэширования ГОСТ Р 34.11-94 при вычислении отпечатка (fingerprint) сертификата.

Следует иметь в виду, что после 31 декабря 2007 года алгоритм ГОСТ Р 34.10-94 должен использоваться только для проверки ранее выработанных подписей.

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

## 3 КОМАНДА СА

### 3.1 Описание

Команда са реализует простейший удостоверяющий центр. Ее можно использовать для подписывания заявок на сертификаты различным образом и для генерации списков отзыва сертификатов (CRL). Команда также поддерживает текстовую базу данных выпущенных сертификатов и их статуса.

### 3.2 Формат ввода команды

```
openssl ca [-verbose] [-config filename] [-name section] [-gencrl] [-revoke file] [-crl_reason reason] [-crl_hold instruction] [-crl_compromise time] [-crl_CA_compromise time] [-crl days days] [-crl hours hours] [-crl_exts section] [-startdate date] [-enddate date] [-days arg] [-md arg] [-policy arg] [-keyfile arg] [-key arg] [-passin arg] [-cert file] [-selfsign] [-in file] [-out file] [-notext] [-outdir dir] [-infile] [-spkac file] [-ss_cert file] [-preserveDN] [-noemailDN] [-batch] [-msie_hack] [-extensions section] [-extfile section] [-engine id] [-subj arg] [-utf8] [-multivalue-rdn]
```

### 3.3 Опции команды

Описание опций разбито на разделы по их целевому назначению.

#### 3.3.1 Опции обработки заявок и выпуска сертификатов

Опция	Описание
-config filename	Указывает, какой конфигурационный файл следует использовать.
-name section	Указывает, какой раздел конфигурационного файла использовать (имеет больший приоритет, нежели значение параметра default_ca в разделе ca файла конфигурации).
-in filename	Указывает входной файл, содержащий одну заявку на выдачу сертификата, которую следует подписать подписью удостоверяющего центра.
-ss_cert filename	Один самозаверенный сертификат, который следует подписать подписью удостоверяющего центра.
-spkac filename	Файл, содержащий один подписанный открытый ключ и challenge в формате Netscape и значения дополнительных полей, на основе которого следует выработать сертификат. См. раздел 3.5 для получения информации по требуемому формату.
-infile	Если эта опция присутствует, она должна быть указана последней, все последующие аргументы считаются именами файлов, содержащих заявки на сертификаты.
-out filename	Выходной файл для сертификатов. По умолчанию — стандартный вывод. Информация о сертификате также будет выведена в этот файл.
-outdir directory	Выходной каталог для сертификатов. Сертификат будет записан в файл с именем, состоящим из серийного номера в шестнадцатеричном виде, с расширением .pem.

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

Опция	Описание
-cert file	Имя файла сертификата удостоверяющего центра.
-keyfile filename	Имя файла, содержащего закрытый ключ, с помощью которого следует вырабатывать подписи под сертификатами.
-key password	Пароль для зашифрования закрытого ключа. Поскольку в некоторых операционных системах аргументы командной строки видны (например Unix-подобные ОС с утилитой ps), эту опцию следует использовать с осторожностью.
-selfsign	<p>Опция для перевыпуска сертификата удостоверяющего центра. Указывает, что выпускаемый сертификат следует подписать тем ключом, на котором были подписаны заявки (ключ определяется опцией -keyfile). Заявки, подписанные другим ключом, игнорируются. Если указаны опции -spkas, -ss_cert или -genctrl, опция -selfsign игнорируется.</p> <p>В результате использования опции -selfsign в базе данных сертификатов (см. configuration option database) появляется самоподписанный сертификат, использующий тот же счетчик серийных номеров, что и другие сертификаты, подписанные с помощью самоподписанного сертификата.</p>
-passin arg	Указывает, где содержится пароль для ключа. Для получения дополнительной информации по формату аргумента arg см. раздел 1.6.
-verbose	Выводит дополнительную информацию о выполняемых операциях
-notext	Отменяет вывод текстовой формы сертификата в выходной файл.
-startdate date	Позволяет явным образом указать дату вступления сертификата в действие. Формат даты ГГММДДЧЧММССЗ (как в структуре ASN.1 UTCTime; З — временная зона).
-enddate date	Позволяет явным образом указать дату окончания действия сертификата. Формат даты ГГММДДЧЧММССЗ (как в структуре ASN.1 UTCTime; З — временная зона).
-days arg	Срок действия сертификата в днях.
-md alg	Указание алгоритма хэширования. Для сертификатов, подписанных ключом с алгоритмом ГОСТ Р-34.10 необходимо использовать алгоритм хэширования md_gost94, каковой используется по умолчанию.
-policy arg	Эта опция определяет, какая «политика» удостоверяющего центра используется. Это раздел конфигурационного файла, который определяет, какие поля должны быть обязательными или соответствовать сертификату удостоверяющего центра. Для получения дополнительной информации см. раздел 3.4.
-msie_hack	Эта опция совместимости для поддержки работы удостоверяющего центра с очень старыми версиями Microsoft Internet Explorer. Так как использование российских криптоалгоритмов с этими старыми версиями IE невозможно, этой опцией пользоваться не следует.
-preserveDN	Обычно порядок полей в Distinguished Name сертификата совпадает с порядком полей в соответствующем разделе политики. Когда установлена эта опция, порядок полей такой же, как в заявке. Это делается в основном для совместимости со старыми версиями IE.

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

Опция	Описание
-noemailDN	Distinguished Name сертификата может содержать поле EMAIL, если такое поле присутствует в заявке, но считается хорошей политикой просто указывать адрес электронной почты в расширении сертификата altName. Когда установлена эта опция, поле EMAIL удаляется из тела сертификата и устанавливается только в присутствующих расширениях. Для достижения этого же результата можно также использовать ключевое слово email_in_dn в конфигурационном файле.
-batch	Эта опция устанавливает пакетный (неинтерактивный) режим работы. В этом режиме не задается никаких вопросов.
-extensions section	Указывает, что при выпуске сертификата необходимо добавить расширения, указанные в соответствующем разделе конфигурационного файла (опция по умолчанию для X509_extensions, если не используется опция -extfile). Если в момент выпуска сертификата в конфигурационном файле отсутствует раздел расширений, создается сертификат версии V1. Если раздел расширений присутствует (даже если он пуст), создается сертификат версии V3.
-extfile file	Прочитать расширения сертификата из дополнительного конфигурационного файла (с использованием умолчательного раздела, если не указана также опция -extensions).
-engine id	Указывает на загружаемый модуль engine (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.
-subj arg	Замещает содержание поля subject name, указанное в заявке. Формат аргумента arg должен быть /type0=value0/type1=value1/type2=..., символы могут быть экранированы знаком \ (обратный слэш), пробелы не опускаются.
-utf8	Эта опция указывает, что значения полей следует интерпретировать как строки в кодировке UTF8, по умолчанию они интерпретируются в кодировке ASCII. Это означает, что значения полей, считанные с терминала или из конфигурационного файла, должны быть корректными UTF-8 строками.
-multivalue-rdn	Эта опция указывает, что аргумент опции -subj необходимо интерпретировать с полной поддержкой многозначных RDN. Пример: /DC=org/DC=OpenSSL/DC=users/UID=123456+CN=John Doe Если опция -multi-rdn не используется, то значение поля UID будет 123456+CN=John Doe.

### 3.3.2 Опции работы со списками отзыва сертификатов (CRL)

Опция	Описание
-gencrl	Эта опция генерирует список отзыва сертификатов на основе информации из индексного файла.
-crl days num	Срок действия списка отзыва сертификата в днях. Этот срок от момента выпуска будет указан в поле nextUpdate списка отзыва сертификатов.

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

Опция	Описание
-crlhours num	Срок действия списка отзыва сертификатов в часах.
-revoke filename	Файл, содержащий сертификат, который следует отозвать.
-crl_reason reason	<p>Причина отзыва, где в качестве значения аргумента reason может быть:</p> <p><b>unspecified</b> (не указана)  <b>keyCompromise</b> (компрометация ключа владельца сертификата)  <b>CACompromise</b> (компрометация ключа удостоверяющего центра)  <b>affiliationChanged</b> (смена должности владельца сертификата)  <b>superseded</b> (замена сертификата)  <b>cessationOfOperation</b> (прекращение деятельности)  <b>certificateHold</b> (временный отзыв сертификата)  <b>removeFromCRL</b> (отмена отзыва ранее отозванного сертификата)</p> <p>Ключевые слова, указывающие причины отзыва, нечувствительны к регистру. Указание любой причины отзыва придаст сертификату версию V2. На практике причина removeFromCRL (отмена отзыва) не слишком широко употребляется, потому что она используется только в списках типа delta, которые сейчас не поддерживаются.</p>
-crl_hold instruction	Эта опция устанавливает причину отзыва в certificateHold (временный отзыв сертификата) и значение аргумента instruction в инструкцию к временному отзыву, которая должна быть OID. Хотя можно использовать любой OID, обычно используются только holdInstructionNone (использование которого не рекомендовано в RFC2459), holdInstructionCallIssuer или holdInstructionReject.
-crl_compromise time	Эта опция устанавливает причину отзыва в keyCompromise (компрометация ключа владельца сертификата), а время компрометации — в значение аргумента time. Значение аргумента time должно быть указано в формате GeneralizedTime, т.е. ГГГГММДДЧЧММССЗ (где З — временная зона).
-crl_CA_compromise time	То же самое, что и -crl_compromise, с той разницей, что причина отзыва устанавливается в значение CACompromise.
-crlxts section	Раздел конфигурационного файла, содержащий расширения списков отзыва сертификатов, которые следует включить. Если в конфигурационном файле нет такого раздела, создается список версии V1, если присутствует (даже пустой), создается список версии V2. Указанные расширения являются расширениями именно CRL, а не его отдельных входов (описаний сертификатов). Следует отметить, что некоторые программы (например Netscape) не умеют работать со списками отзыва сертификатов версии V2.

### 3.3.3 Опции конфигурационного файла

Раздел конфигурационного файла, содержащий опции для команды ca, находится следующим образом: если в командной строке указана опция -path, она указывает нужный раздел. В противном случае необходимый раздел должен быть указан в опции default\_ca раздела

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

са конфигурационного файла (или умолчательного раздела конфигурационного файла). Кроме default\_ca, непосредственно из раздела са считываются следующие опции: RANDFILE, preserve, msie\_hack. Кроме RANDFILE, это, вероятно, ошибка и может измениться в последующих версиях.

Многие опции конфигурационного файла идентичны опциям командной строки. Опции командной строки имеют приоритет над опциями конфигурационного файла. Если опция указана как обязательная, необходимо указать либо эту опцию в конфигурационном файле, либо ее аналог в командной строке (если такой аналог существует).

Опция	Описание
oid_file	Эта опция указывает на файл, содержащий дополнительные OID (OBJECT IDENTIFIERS). Каждая строка файла должна иметь следующий формат: OID в численном виде, пробел, короткое имя, пробел, длинное имя.
oid_section	Эта опция указывает на раздел конфигурационного файла, содержащий дополнительные OID. Каждая строка раздела должна иметь формат: короткое имя OID=численный вид OID. В случае использования этой опции короткое и длинное имена совпадают.
new_certs_dir	То же, что и опция командной строки -outdir. Указывает каталог, в который должны быть помещены новые сертификаты. Обязательна.
certificate	то же, что и опция командной строки -cert. Указывает файл, содержащий сертификат удостоверяющего центра. Обязательна.
private_key	то же, что и опция командной строки -keyfile. Указывает файл, содержащий закрытый ключ удостоверяющего центра. Обязательна.
RANDFILE	Файл, используемый для считывания и записи информации для инициализации генератора случайных чисел.
default_days	То же, что и опция командной строки -days. Срок действия сертификата в днях.
default_startdate	То же, что и опция командной строки -startdate. Дата начала действия сертификата. Если не установлена, используется текущее время.
default_enddate	То же, что и опция командной строки -enddate. Должна быть указана либо эта опция, либо опция default_days (или их командно-строчные эквиваленты).
default_crl_hours default_crl_days	То же, что и опции -crlhours и -crldays. Используются только в том случае, если не указано ни одной из соответствующих командно-строчных опций. Чтобы создать список отзыва сертификатов, необходимо указать хотя бы одну из этих четырех опций.
default_md	Опция обязательно должна быть указана, но используется только в том случае, если ключ удостоверяющего центра имеет алгоритм, позволяющий использовать разные алгоритмы хэширования (RSA). Значение опции представляет собой название алгоритма хэширования, используемого для подписи сертификатов (в противном случае оно может быть любым). Обязательная.

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

Опция	Описание
database	Используемый файл текстовой базы данных. Обязательная. Указанный файл должен существовать, хотя сначала он будет пустым.
unique_subject	Если указано значение этой опции yes, то описания сертификатов в базе данных должны иметь уникальные поля subject. Если указано значение no, несколько описаний актуальных сертификатов могут иметь одно и то же значение поля subject. Значение по умолчанию — yes для совместимости со старыми (до 0.9.8) версиями криптобиблиотеки OpenSSL. Однако для упрощения процедуры перевыпуска сертификата удостоверяющего центра рекомендуется использовать значение no, особенно в комбинации с опцией командной строки -selfsign.
serial	Текстовый файл, содержащий серийный номер для следующего сертификата в шестнадцатеричной форме. Обязательна. Указанный файл должен существовать и содержать корректный серийный номер.
crlnumber	Текстовый файл, содержащий номер для следующего списка отзыва сертификатов в шестнадцатеричной форме. Номер будет включен в списки отзывов сертификатов только в том случае, если этот файл существует. Если этот файл существует, он должен содержать корректный номер списка отзыва сертификатов.
x509_extensions	То же, что и опция командной строки -extensions.
crl_extensions	То же, что и опция командной строки -crlxts.
preserve	То же, что и опция командной строки -preserveDN
email_in_dn	То же, что и опция командной строки -noemailDN. Если вы хотите удалить поле EMAIL из distinguished name сертификата, просто установите значение этой опции в no. Если опция не указана, по умолчанию поле EMAIL в distinguished name сертификата позволено.
msie_hack	То же, что и опция командной строки -msie_hack
policy	То же, что и опция командной строки -policy. Обязательна. См. раздел 3.4 для получения дополнительной информации.
name_opt, cert_opt	<p>Эти опции определяют формат вывода информации о сертификате при запросе подтверждения подписывания сертификата от пользователя. Здесь могут использоваться все опции, поддерживаемые опциями -nameopt и -certopt утилиты x509 (см. раздел 16), кроме того, что опции no_signame и no_sigdump жестко установлены и не могут быть отключены (потому что подпись под сертификатом не может быть выведена, т.к. в этот момент сертификат еще не подписан).</p> <p>Для удобства можно придать обоим этим опциям значения ca_default для получения достойного результата.</p> <p>Если обе эти опции отсутствуют, используется формат из старых версий криптобиблиотеки OpenSSL. Использование старого формата не рекомендуется, потому что он выводит только те поля, которые указаны в разделе политики, неправильно работает со строковыми типами данных и не выводит расширения.</p>

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

Опция	Описание
copy_extensions	<p>Эта опция определяет, как следует работать с расширениями в заявках на выдачу сертификата. Если значение опции установлено в <code>popе</code>, или эта опция вообще не указана, расширения игнорируются и не переносятся в сертификат. Если значение этой опции установлено в <code>copy</code>, все расширения, имеющиеся в заявке, которых еще нет в сертификате, переносятся в сертификат. Если значение этой опции установлено в <code>copyall</code>, то все расширения из заявки переносятся в сертификат: если расширение уже присутствует в сертификате, оно сначала оттуда удаляется. Перед применением этой опции см. раздел 3.10.</p> <p>Главное применение этой опции — предоставить возможность переноса из заявки значений некоторых расширений, таких как <code>subjectAltName</code>.</p>

### 3.4 Формат раздела политики

Раздел политики состоит из набора переменных, соответствующих полям `distinguished name` сертификата. Если значение переменной установлено в «`match`», то значение поля должно соответствовать тому же полю в сертификате удостоверяющего центра. Если значение установлено в «`supplied`», поле должно присутствовать. Если значение установлено в «`optional`», поле может присутствовать. Все поля, не упомянутые в разделе политики, удаляются без предупреждения, если только не указана опция `-preserveDN`, но этот случай можно рассматривать как отклонение от стандартного образа действий.

### 3.5 Формат SPKAC

Входными данными для опции командной строки `-srkas` являются подписанный открытый ключ и `challenge` формата Netscape. Обычно эти данные создаются тэгом `KEYGEN` `html`-формы, создающей новый закрытый ключ. Однако возможно создать SPKAC с помощью утилиты `srkas`.

Файл, указываемый в качестве аргумента опции `-srkas`, должен содержать переменную SPKAC, значение которой установлено в значение SPKAC, а также требуемые компоненты DN в виде пар "имя-значение". Если вам нужно включить один и тот же компонент дважды, перед ним можно указать номер с точкой.

### 3.6 Примеры

**Примечание:** эти примеры предполагают, что структура каталогов удостоверяющего центра уже создана и соответствующие файлы уже существуют. Это обычно включает создание сертификата удостоверяющего центра и закрытого ключа с помощью утилиты `req`, файла номера серийного файла и пустого индексного файла, и размещение их в соответствующих каталогах.

Чтобы использовать нижеуказанный пример конфигурационного файла, необходимо создать каталоги `demoCA`, `demoCA/private` и `demoCA/newcerts`. Сертификат удостоверяющего центра следует скопировать в файл `demoCA/cacert.pem`, а его закрытый ключ — в файл `demoCA/private/cakey.pem`. Следует создать файл `demoCA/serial`, содержащий, например, «01», и пустой индексный файл `demoCA/index.txt`.

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

Подписание заявки на сертификат:

```
openssl ca -in req.pem -out newcert.pem
```

Подписание заявки на сертификат с использованием расширений удостоверяющего центра:

```
openssl ca -in req.pem -extensions v3_ca -out newcert.pem
```

Создание списка отзыва сертификатов:

```
openssl ca -gencrl -out crl.pem
```

Подписывание нескольких заявок:

```
openssl ca -infiles req1.pem req2.pem req3.pem
```

Сертификация SPKAC формата Netscape:

```
openssl ca -spkac spkac.txt
```

Образец файла SPKAC (строка SPKAC сокращена для наглядности):

```
SPKAC=MIGOMGAwXDANBqkqhkiG9w0BAQEFAANLADBIaKEAn7PDhCeV/xIxUg8V70YRxBK2A5
CN=Steve Test
emailAddress=steve@openssl.org
0.OU=OpenSSL Group
1.OU=Another Group
```

Образец конфигурационного файла с соответствующими разделами для команды ca:

```
[ ca ]
default_ca      = CA_default          # The default ca section

[ CA_default ]

dir             = ./demoCA           # top dir
database       = $dir/index.txt     # index file.
new_certs_dir  = $dir/newcerts      # new certs dir

certificate    = $dir/cacert.pem     # The CA cert
serial        = $dir/serial         # serial no file
private_key    = $dir/private/cakey.pem # CA private key
RANDFILE      = $dir/private/.rand  # random number file

default_days   = 365                # how long to certify for
default_crl_days= 30                # how long before next CRL
default_md     = md5                # md to use

policy        = policy_any          # default policy
email_in_dn   = no                  # Don't add the email into cert DN

name_opt      = ca_default          # Subject name display option
cert_opt      = ca_default          # Certificate display option
copy_extensions = none              # Don't copy extensions from request

[ policy_any ]
countryName   = supplied
stateOrProvinceName = optional
```

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

```

organizationName      = optional
organizationalUnitName = optional
commonName            = supplied
emailAddress          = optional
    
```

### 3.7 Файлы

**Примечание:** Расположение всех файлов может быть изменено либо в опциях времени компилирования, либо в конфигурационном файле, переменных среди или опциях командной строки. Указанные значения являются умолчательными.

```

/usr/local/ssl/lib/openssl.cnf - главный конфигурационный файл
./demoCA                      - головной каталог удостоверяющего центра (УЦ)
./demoCA/cacert.pem           - сертификат удостоверяющего центра
./demoCA/private/akey.pem     - Закрытый ключ удостоверяющего центра
./demoCA/serial               - Файл серийного номера УЦ
./demoCA/serial.old           - Бэкапный файл серийного номера УЦ
./demoCA/index.txt            - Файл текстовой базы данных УЦ
./demoCA/index.txt.old        - Бэкапный файл текстовой базы данных УЦ
./demoCA/certs                - файл для вывода сертификатов
./demoCA/.rnd                 - информация для инициализации генератора
                              случайных чисел УЦ
    
```

### 3.8 Переменные среды

Переменная среды `OPENSSL_CONF` отражает расположение главного конфигурационного файла. Опция командной строки `-config` имеет приоритет над этой переменной.

### 3.9 Ограничения

Индексный файл текстовой базы данных — критическая часть процесса, и если этот файл поврежден, его может оказаться трудно восстановить. Теоретически возможно восстановить индексный файл из всех выпущенных сертификатов и текущего списка отзыва сертификатов, но такой опции не существует.

Свойства версии V2 списков отзыва сертификатов, такие как дельта-списки, сейчас не поддерживаются.

Хотя одновременно может быть введено и обработано несколько заявок, можно включить только один SPKAC или самоподписанный сертификат.

### 3.10 Предупреждения

Команда `sa` прихотлива и иногда ведет себя недружественно по отношению к пользователю.

Утилита `sa` сначала была написана в качестве примера того, как работать с удостоверяющим центром. Она не планировалась как полнофункциональный удостоверяющий центр: однако некоторые используют ее в этом качестве.

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

Команда са по сути представляет собой однопользовательскую команду: ни на какие файлы не накладываются блокировки, и попытки запустить более одной команды са на одной и той же базе данных могут привести к непредсказуемым результатам.

Опцию `copy_extensions` следует использовать с осторожностью. Иначе может быть нарушена безопасность системы. Например, если заявка на сертификат содержит расширение `basicConstraints` с `CA:TRUE`, значение `copy_extensions` установлено в `copyall`, а пользователь этого не замечает, когда сертификат демонстрируется, это предоставит запрашивающему действительный сертификат удостоверяющего центра.

Этой ситуации можно избежать, установив опцию `copy_extensions` в `copy` и включив `basicConstraints` с `CA:FALSE` в конфигурационный файл. Тогда, если заявка будет содержать расширение `basicConstraints`, оно будет проигнорировано.

Также рекомендуется включать в конфигурационный файл значения других расширений, таких как `keyUsage`, чтобы предотвратить автоматический перенос значений этих расширений в сертификат из заявки.

Дополнительные ограничения можно включить в сам сертификат удостоверяющего центра. Например, если в сертификате удостоверяющего центра указано:

```
basicConstraints = CA:TRUE, pathlen:0
```

то даже если будет выпущен пользовательский сертификат с `CA:TRUE`, он будет недействителен.

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

## 4 КОМАНДА CRL

### 4.1 Описание команды

Команда `crl` обрабатывает файлы списков отзыва сертификатов в формате DER или PEM.

### 4.2 Формат ввода команды

```
openssl crl [-inform PEM|DER] [-outform PEM|DER][-text] [-in filename] [-out filename] [-noout] [-hash] [-issuer] [-lastupdate] [-nextupdate] [-CAfile file] [-CApath dir]
```

### 4.3 Опции команды

Опция	Описание
-inform DER PEM	Указывает входной формат. Формат DER — структура CRL в DER-кодировке. PEM (умолчание) — версия DER-формы в кодировке base64 с верхним и нижним ограничителями.
-outform DER PEM	Указывает выходной формат. Опция имеет те же значения, что и опция -inform.
-in filename	Указывает файл с входными данными. Если опция не указана, данные читаются со стандартного ввода.
-out filename	Указывает файл для записи выходных данных. Если опция не указана, данные выводятся в стандартный вывод.
-text	Вывести список отзыва сертификатов в текстовом виде.
-noout	Не выводить кодированную версию списка отзыва сертификатов.
-hash	Вывести хэш-сумму значения поля issuer. Эту опцию можно использовать для поиска списков отзыва сертификатов в каталоге по полю issuer name.
-issuer	Вывести имя выпускающего.
-lastupdate	вывести значение поля lastUpdate.
-nextupdate	вывести значение поля nextUpdate.
-CAfile file	Проверить подпись под списком отзыва сертификатов с поиском сертификата выпустившего удостоверяющего центра в файле
-CApath dir	Проверить подпись под списком отзыва сертификатов с поиском сертификата выпустившего удостоверяющего центра в каталоге. Этот каталог должен быть стандартным каталогом сертификатов: то есть с каждым сертификатом должен быть связан хэш значения соответствующего поля subject name (полученного с помощью утилиты x509 с опцией -hash).

### 4.4 Примечания

PEM-формат списка отзыва сертификатов использует следующие верхний и нижний ограничители:

```
-----BEGIN X509 CRL-----
-----END X509 CRL-----
```

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

## 4.5 Примеры

Преобразовать файл списка отзыва сертификатов из формата PEM в формат DER:

```
openssl crl -in crl.pem -outform DER -out crl.der
```

Output the text form of a DER encoded certificate:

```
openssl crl -in crl.der -text -noout
```

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

## 5 КОМАНДА CRL2PKCS7

### 5.1 Описание команды

Команда `crl2pkcs7` превращает один или более сертификатов и список отзыва сертификатов (необязателен) в вырожденную PKCS#7-структуру, содержащую только сертификаты (не содержащую сообщений).

### 5.2 Формат ввода команды

```
openssl crl2pkcs7 [-inform PEM|DER] [-outform PEM|DER] [-in filename] [-out filename] [-certfile filename] [-nocrl]
```

### 5.3 Опции команды

Опция	Описание
-inform DER PEM	Указывает входной формат списка отзыва сертификатов. Формат DER — структура CRL в DER-кодировке. PEM (умолчание) — версия DER-формы в кодировке base64 с верхним и нижним ограничителями.
-outform DER PEM	Указывает выходной формат PKCS#7-структуры. Формат DER — структура CRL в DER-кодировке. PEM (умолчание) — версия DER-формы в кодировке base64 с верхним и нижним ограничителями.
-in filename	Эта опция определяет входной файл, из которого следует считать список отзыва сертификатов. Если эта опция не указана, список отзыва сертификатов считывается со стандартного ввода.
-out filename	Эта опция определяет выходной файл, в который записывается полученная PKCS#7-структура. Если эта опция не указана, PKCS#7-структура выводится на стандартный вывод.
-certfile filename	Эта опция определяет файл, содержащий один или больше сертификатов в PEM-формате. Все сертификаты из этого файла будут включены в PKCS#7-структуру. Эта опция может быть указана несколько раз, если нужно считать сертификаты из нескольких файлов.
-nocrl	Как правило, в выходной файл включается список отзыва сертификатов. Если указана данная опция, список отзыва сертификатов не считывается из входных данных и не включается в выходной файл.

### 5.4 Примеры

Создать PKCS#7-структуру из сертификата и списка отзыва сертификатов:

```
openssl crl2pkcs7 -in crl.pem -certfile cert.pem -out p7.pem
```

Создать PKCS#7-структуру в DER-формате из нескольких разных сертификатов, список отзыва сертификатов не включать:

```
openssl crl2pkcs7 -nocrl -certfile newcert.pem -certfile demoCA/cacert.pem -outform DER -out p7.der
```

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

## 5.5 Примечания

Выходной файл представляет собой PKCS#7-структуру «signed data», не содержащую никаких подписей, содержащую только сертификаты и опциональный список отзыва сертификатов.

Эту утилиту можно использовать для отправки сертификатов и списков отзыва сертификатов в браузеры в качестве части процесса ввода сертификата в действие. Это включает отправку данных MIME-типа application/x-x509-user-cert.

Данные в PEM-формате без верхнего и нижнего ограничителей можно использовать для установки пользовательских сертификатов и сертификатов УЦ в Microsoft Internet Explorer с помощью Active-X элемента Xenroll.

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

## 6 КОМАНДА DGST

### 6.1 Описание команды

Команда позволяет вычислить хэш-сумму для предоставленного файла или файлов в шестнадцатеричном виде. Также может быть использована для формирования и подтверждения электронной цифровой подписи (ЭЦП). Опция `-md_gost94` в этой команде используется всегда при работе с алгоритмами ГОСТ.

### 6.2 Формат ввода команды

```
openssl dgst -md_gost94 [-c][-d] [-hex] [-binary] [-out filename] [-sign filename] [-passin arg][
-verify filename] [-prverify filename] [-signature filename] [file...]
```

### 6.3 Опции команды

Опция	Описание
-c	Вывести хэш-сумму в двух цифровых группах, разделенных вертикальной чертой, актуальна только если используется шестнадцатеричный формат вывода.
-d	Вывести отладочную информацию ВЮ.
-hex	Вывести хэш-сумму в виде шестнадцатеричного дампа. Это умолчательный вариант для «обыкновенной» хэш-суммы, в отличие от цифровой подписи.
-binary	Вывести хэш-сумму или подпись в бинарном виде.
-out filename	указать имя файла вывода, по умолчанию — стандартный вывод.
-md_gost94	Выбор алгоритма хэширования ГОСТ Р 34.11-94
-sign filename	Выработать цифровую подпись, используя закрытый ключ, содержащийся в указанном файле.
-passin arg	Место хранения пароля к закрытому ключу. За дополнительной информацией о формате аргумента см. раздел 1.6.
-verify filename	Проверить подпись с использованием открытого ключа, содержащегося в указанном файле. Результат — либо «Подпись корректна», либо «Подпись некорректна».
-prverify filename	Проверить подпись с использованием открытого ключа, содержащегося в указанном файле.
-signature filename	Указать файл, под которым необходимо проверить подпись.
-rand file(s)	Файл или файлы, содержащие случайные данные, используемые для инициации генератора случайных чисел. Несколько файлов можно указать через разделитель, который определяется операционной системой: ; для MS-Windows, , для OpenVMS и : для всех остальных.
file...	Файл или файлы, для которых нужно вычислить хэш-суммы. Если не указано ни одного файла, используется стандартный ввод.

**Примечание.** Опции выработки и проверки подписи следует применять только в том случае, если обрабатывается только один файл.

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

## 7 КОМАНДА ENC

### 7.1 Описание команды

Команда симметричного шифрования предоставляет возможность зашифрования и расшифрования данных с использованием различных блочных и потоковых алгоритмов и с использованием ключей, полученных из пароля или указанных явным образом. С помощью этой команды также может быть выполнено кодирование и декодирование в кодировку base64, как само по себе, так и вместе с зашифрованием и расшифрованием.

### 7.2 Формат вызова команды

```
openssl enc -ciphername [-in filename] [-out filename] [-pass arg] [-e][-d] [-a] [-A] [-gost89] [-k password] [-kfile filename] [-K key] [-iv IV] [-p][-P] [-bufsize number] [-nopad] [-debug]
```

### 7.3 Опции команды

Опция	Описание
-in filename	имя входного файла, по умолчанию стандартный ввод
-out filename	имя выходного файла, по умолчанию стандартный вывод
-pass arg	источник пароля. Для получения дополнительной информации о формате аргумента arg см. раздел 1.6.
-salt	использовать «соль» при формировании ключа из пароля. Эту опцию следует использовать <b>всегда</b> , за исключением тех случаев, когда требуется совместимость с предыдущими версиями криптобиблиотеки OpenSSL или SSLeay. Эта опция присутствует в версиях криптобиблиотеки OpenSSL начиная с версии 0.9.5.
-nosalt	Не использовать «соль» в процедурах формирования ключа из пароля. Эта опция является умолчательной для совместимости с предыдущими версиями криптобиблиотеки OpenSSL или SSLeay.
-e	Зашифровать входные данные: это опция по умолчанию.
-d	Расшифровать входные данные.
-a	base64-обработка данных. Это означает, что если происходит зашифрование, то после него выполняется также кодирование в кодировку base64. Если задано расшифрование, то перед расшифрованием выходные данные декодируются из кодировки base64.
-A	если задана опция -a, то результат base64-обработки не разбивается на строки. Эта опция некорректно работает с очень большими файлами.
-gost89	Выбор алгоритма шифрования ГОСТ 28147-89
-k password	Пароль, из которого следует сформировать ключ. Эта опция введена для совместимости с предыдущими версиями криптобиблиотеки OpenSSL. Имеет меньший приоритет, чем опция -pass.
-kfile filename	Прочитать пароль, из которого следует сформировать ключ, из первой строки файла с указанным именем. Эта опция введена для совместимости с предыдущими версиями криптобиблиотеки OpenSSL. Имеет меньший приоритет, чем опция -pass.

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

Опция	Описание
-S salt	Явным образом указанная «соль»: она должна быть представлена в виде строки, состоящей только из шестнадцатирчных цифр.
-K key	Явным образом указанный ключ шифрования: он должен быть представлен в виде строки, состоящей только из шестнадцатирчных цифр. Если указан только ключ, необходимо также указать вектор инициализации с помощью опции -iv. Если указаны и ключ и пароль, то в шифровании будут использованы ключ, указанный в опции -K, и вектор инициализации, генерированный из пароля. Вероятно, не большого смысла указывать и ключ, и пароль.
-iv IV	Явным образом указанный вектор инициализации: он должен быть представлен в виде строки, содержащей только шестнадцатирчные цифры. Если указан только ключ с помощью опции -K, необходимо указать вектор инициализации явным образом. Если с помощью одной из опций указан пароль, вектор инициализации генерируется из пароля.
-p	Вывести используемый ключ и вектор инициализации
-P	Вывести используемый ключ и вектор инициализации и немедленно завершить работу: не выполнять ни зашифрования, ни расшифрования.
-bufsize number	Установить размер буфера для I/O
-popad	отключить дополнение блоков до стандартной длины
-debug	отладить В/О, используемые для И/О.

## 7.4 Примечания

При необходимости запрашивается пароль для формирования ключа и вектора инициализации.

Опцию -salt следует использовать **всегда**, если ключ формируется из пароля, за исключением случаев, когда нужна совместимость с предыдущими версиями криптобиблиотеки OpenSSL.

При отсутствии опции -salt возможно проведение успешных атак методом подбора пароля и атака на данные, зашифрованные при помощи потокового алгоритма. Причиной этого является тот факт, что без «соли» на основе одного и того же пароля всегда формируется один и тот же ключ. При использовании «соли» первые восемь байт зашифрованных данных зарезервированы под «соль»: она генерируется случайным образом при зашифровании файла и считывается из зашифрованного файла при расшифровывании.

### 7.4.1 Примеры

Перевести бинарный файл в кодировку base64 без зашифровывания:

```
openssl base64 -in file.bin -out file.b64
```

Раскодировать тот же файл:

```
openssl base64 -d -in file.b64 -out file.bin
```

Зашифровать файл, используя алгоритм шифрования ГОСТ 28147-89:

```
openssl enc -gost89 -salt -in file.txt -out file.enc
```

Расшифровать файл, используя предоставленный пароль:

```
openssl enc -gost89 -d -salt -in file.enc -out file.txt -k
```

mypassword

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

Зашифровать файл, а потом перевести его в кодировку base64 (например, чтобы его можно было отправить по почте):

```
openssl enc -gost89 -a -salt -in file.txt -out file.enc
```

Перевести файл из кодировки base64 в обычную и расшифровать:

```
openssl enc -gost89 -d -salt -a -in file.enc -out file.txt
```

Расшифровать данные, используя указанный ключ (ключ сокращен для наглядности):

```
openssl enc -gost89 -d -in file.enc -out file.txt -K 0102030405...
```

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

## 8 КОМАНДА OSCP

### 8.1 Описание

OSCP (онлайн-протокол статусов сертификатов) позволяет приложениям определять (отозванное) состояние идентифицированного сертификата (RFC 2560).

Команда `oscp` выполняет многие обычные задачи OSCP. Ее можно использовать для вывода запросов и ответов на них, создавать и посылать запросы на OSCP-ответчик, а также в качестве небольшого `oscp`-сервера.

### 8.2 Формат ввода команды

```
openssl ocp [ -out file ] [ -issuer file ] [ -cert file ] [ -serial n ] [ -signer file ] [ -signkey file ] [ -sign_other
file ] [ -no_certs ] [ -req_text ] [ -resp_text ] [ -text ] [ -reqout file ] [ -respout file ] [ -reqin file ] [ -respin
file ] [ -nonce ] [ -no_nonce ] [ -url URL ] [ -host host:n ] [ -path ] [ -CApath dir ] [ -CAfile file ] [ -VAfile
file ] [ -validity_period n ] [ -status_age n ] [ -noverify ] [ -verify_other file ] [ -trust_other ] [ -no_intern ]
[ -no_signature_verify ] [ -no_cert_verify ] [ -no_chain ] [ -no_cert_checks ] [ -port num ] [ -index file ] [ -
CA file ] [ -rsigner file ] [ -rkey file ] [ -rother file ] [ -resp_no_certs ] [ -nmin n ] [ -ndays n ] [ -resp_key_id ]
[ -nrequest n ]
```

### 8.3 Опции команды

#### 8.3.1 Клиентские опции команды `oscp`

Опция	Описание
<code>-out filename</code>	Указывает имя выходного файла. По умолчанию выходные данные направляются на стандартный выход.
<code>-issuer filename</code>	Указывает файл сертификата удостоверяющего центра, выпустившего проверяемый сертификат. Эту опцию можно использовать несколько раз. Сертификат, указанный в качестве значения опции, должен быть в формате PEM.
<code>-cert filename</code>	Добавляет к формируемому запросу запрос статуса сертификата, содержащегося в файле <code>filename</code> . Информация об удостоверяющем центре берется из предыдущей опции <code>issuer</code> ; если ни одной такой опции не указано, выводится сообщение об ошибке.
<code>-serial num</code>	То же, что и опция <code>cert</code> , за исключением того, что к формируемому запросу добавляется запрос о статусе сертификата с серийным номером, указанным в качестве значения опции <code>num</code> . Серийный номер интерпретируется как десятичное целое число, если только он не начинается с 0x. Можно указывать также отрицательные целые числа с помощью знака «-» перед значением <code>num</code> .
<code>-signer filename, -signkey filename</code>	Подписывает <code>oscp</code> -запрос с использованием сертификата, указанного в опции <code>signer</code> , и закрытого ключа, указанного в опции <code>signkey</code> . Если опция <code>signkey</code> не указана, закрытый ключ считывается из того же файла, что и сертификат. Если не указана ни одна опция, <code>oscp</code> -запрос не подписывается.

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

Опция	Описание
-sign_other filename	Дополнительные сертификаты, которые следует включить в подписанный запрос.
-nonce, -no_nonce	Включает OCSP-расширение nonce в запрос или отключает добавление этого расширения. Как правило, если OCSP-запрос вводится с использованием опции respin, расширение nonce не включается; указывание опции nonce указывает на необходимость включения расширения nonce. Если OCSP-запрос создается (с использованием опции cert и serial), расширение nonce добавляется автоматически; указанием опции no_nonce добавление расширения nonce отменяется.
-req_text, -resp_text, -text	Выводит текстовую форму OCSP-запроса, ответа или и то и другое соответственно.
-reqout file, -respout file	Выводит запрос статуса сертификата или ответ в DER-кодировке в файл.
-reqin file, -respin file	Считывает OCSP-запрос или файл ответа из файла. Эти опции игнорируются, если другими опциями подразумевается создание OCSP-запроса или ответа (например, присутствуют опции serial, cert или host).
-url responder_url	Определяет URL ответчика. Могут указываться как URL, начинающиеся с HTTP, так и с HTTPS (SSL/TLS).
-host hostname:port, -path pathname	Если опция host присутствует, то OCSP-запрос отправляется на указанный port указанного hostname. path указывает http-путь или «/» по умолчанию.
-CAfile file, -CApath pathname	Файл, содержащий сертификаты доверенных удостоверяющих центров. Они используются для проверки подписи на OCSP-ответе.
-verify_other file	Файл, содержащий дополнительные сертификаты, среди которых следует искать сертификат, на котором подписан OCSP-ответ. Некоторые ответчики удаляют из ответа сертификат подписчика; эту опцию можно использовать, чтобы в таких случаях предоставить необходимый сертификат.
-trust_other	сертификаты, указанные в опции verify_certs, должны быть явным образом указаны как доверенные, и над ними не будет производиться никаких дополнительных проверок. Это полезно в тех случаях, когда полная цепочка сертификата ответчика не доступна или корневой сертификат не является доверенным.
-VAfile file	Файл, содержащий сертификаты ответчиков, явным образом указанные как доверенные. Эквивалент опций verify_certs и -trust_other.
-noverify	Не пытаться проверять подпись под OCSP-ответом или значения расширения nonce. Эта опция, как правило, используется только для отладки, поскольку она отключает любую проверку сертификата ответчика.
-no_intern	Игнорировать сертификаты, содержащиеся в OCSP-ответе, во время поиска сертификата, на котором подписан данный. При указании этой опции сертификат, на котором подписан данный, должен быть указан с помощью опции -verify_certs или -VAfile.

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

Опция	Описание
no_signature_verify	Не проверять подпись в OCSP-ответе. Поскольку эта опция позволяет принимать ответы с некорректными подписями, она, как правило, используется только в целях тестирования.
-no_cert_verify	Вообще не проверять сертификат, на котором подписан OCSP-ответ. Поскольку эта опция позволяет подписывать OCSP-ответ любым сертификатом, ее следует использовать только в целях тестирования.
-no_chain	Не использовать сертификаты в ответе в качестве дополнительных недоверенных сертификатов удостоверяющих центров.
-no_cert_checks	Не выполнять никаких дополнительных проверок сертификата, на котором подписан OCSP-ответ. Это значит — не проверять, имеет ли право указанный сертификат предоставлять необходимую информацию по статусу; в результате эту опцию следует использовать только в целях тестирования.
-validity_period nsec, -status_age age	<p>Эти опции определяют диапазон времени в секундах, который будет выдерживаться в OCSP-ответе. Каждый ответ по статусу сертификата включает значение поля notBefore и опционально — значение поля notAfter. Текущее время должно попадать между этими двумя величинами, но интервал между ними может быть всего несколько секунд. На практике часы OCSP-ответчика и клиентов могут не быть точно синхронизированы и проверка может не удасться. Чтобы этого избежать, можно использовать опцию -validity_period для определения приемлемого диапазона ошибок в секундах. Значение этой опции по умолчанию — 5 минут.</p> <p>Если в ответ не включено значение поля notAfter, это означает, что новая информация по статусу уже доступна. В этом случае проверяется возраст поля notBefore, чтобы убедиться, что он не старше чем age секунд. По умолчанию эта дополнительная проверка не проводится.</p>
-digest	Эта опция определяет алгоритм дайджеста для использования при генерации идентификатора сертификата удостоверяющего центра. По умолчанию используется алгоритм SHA1. Поэтому при использовании российских алгоритмов следует обязательно указывать опцию -md_gost94.

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

### 8.3.2 Серверные опции команды ocsp

Опция	Описание
-index indexfile	indexfile — текстовый индексный файл в формате команды са утилиты openssl (см. раздел 3), содержащий информацию о статусе сертификатов. Если указана опция index, команда ocsp работает в режиме ответчика (сервера), в противном случае — в режиме клиента. Запрос(ы), которые обрабатывает ответчик, могут либо определяться в командной строке (с помощью опций issuer и serial), либо считываться из файла (с помощью опции respin) или с помощью внешних ocsp-клиентов (если указаны порт или url). Если указана опция index, также должны присутствовать опции CA и rsigner.
-CA file	Сертификат удостоверяющего центра, соответствующий информации об отзывах в файле indexfile.
-rsigner file	Сертификат, на котором следует подписывать OCSP-ответы.
-rother file	Дополнительные сертификаты, которые следует включить в OCSP-ответ.
-resp_no_certs	Не включать никаких сертификатов в OCSP-ответ.
-resp_key_id	Идентифицировать сертификат подписчика с использованием ID ключа, по умолчанию — использовать значение поля subject. Использовать этот режим не рекомендуется, т.к. в RFC 2560 явно прописано использование нестандартного для России алгоритма хэширования.
-rkey file	Закрытый ключ для подписывания OCSP-ответов: если эта опция не присутствует, используется файл, указанный в качестве значения опции rsigner.
-port portnum	Порт, с которого следует считывать OCSP-запросы. Этот порт также может быть указан с помощью опции url.
-nrequest number	OCSP-сервер закончит работу по получении number запросов, по умолчанию — неограниченного количества.
-nmin minutes, -ndays days	Количество минут или дней, когда доступна свежая информация об отзывах: используется в поле nextUpdate. Если ни одна из этих опций не указана, поле nextUpdate опускается, что означает, что свежая информация об отзывах доступна немедленно.

## 8.4 Проверка OCSP-ответов

OCSP-ответы следуют правилам, указанным в RFC2560.

Изначально определяется местоположение сертификата OCSP-ответчика и проверяется подпись под OCSP-запросом с использованием открытого ключа из сертификата ответчика.

Затем на OCSP-ответчике происходит обычная проверка сертификата с построением цепочки сертификатов в процессе. Местонахождение доверенных сертификатов, используемых для построения цепочки, может быть определено с помощью опций CAfile и CApath, или они будут отыскиваться в стандартном каталоге сертификатов OpenSSL.

Если первичная проверка не удастся, процесс проверки OCSP прекращается с сообщением об ошибке.

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

В противном случае сертификат выпустившего СА в запросе сравнивается с сертификатом OCSP-ответчика: если они совпадают, проверка OCSP считается успешной.

В противном случае сертификат, выпустивший сертификат OCSP-ответчика, сравнивается с сертификатом выпускающего удостоверяющего центра в запросе. Если они совпадают и в сертификате OCSP-ответчика присутствует OCSPSigning extended key usage, проверка OCSP считается успешной.

В противном случае корневой сертификат удостоверяющего центра, выпустившего сертификат OCSP-ответчика, проверяется на предмет того, является ли он доверенным для подписывания OCSP. Если да, то проверка OCSP считается успешной.

Если ни одна из этих проверок не оказывается успешной, проверка OCSP не удастся.

По сути это означает, что если сертификат OCSP-ответчика является доверенным непосредственно у того удостоверяющего центра, о котором он передает информацию об отзывах (и корректно сконфигурирован), то проверка удастся.

Если OCSP-ответчик является «глобальным ответчиком», который может давать информацию о многих удостоверяющих центрах и обладает собственной цепочкой сертификатов, то его корневой сертификат может быть доверенным для OCSP-подписи. Например:

```
openssl x509 -in ocspCA.pem -addtrust OCSPSigning -out trustedCA.pem
```

Или же сертификат самого ответчика может быть явным образом объявлен доверенным с помощью опции -VAfile.

## 8.5 Примечания

Как уже было сказано, многие из проверочных опций предназначены для тестовых и отладочных целей. Как правило, нужно использовать только опции -CApath, -CAfile и (если ответчик — глобальный VA) -VAfile.

OCSP-сервер полезен только для тестовых и демонстрационных целей: на самом деле он не может использоваться в качестве полноценного OCSP-ответчика. Он содержит только очень простой обработчик HTTP-запросов и может обрабатывать только POST-форму OCSP-запросов. Он также обрабатывает запросы последовательно, что означает, что он не может отвечать на новые запросы, пока не обработает текущий. Текстовый формат индексного файла отзывов сертификатов также неэффективен для больших количеств данных по отзывам сертификатов.

Возможно запускать приложение ocsp в режиме ответчика через CGI-скрипт с использованием опций respin и respout.

## 8.6 Примеры

Создать OCSP-запрос и записать его в файл:

```
tt openssl ocsp -issuer issuer.pem -cert c1.pem -cert c2.pem -reqout req.der
```

Отправить запрос на OCSP-ответчик с URL-адресом <http://ocsp.myhost.com/>, сохранить ответ в файле и вывести его в текстовой форме

```
openssl ocsp -issuer issuer.pem -cert c1.pem -cert c2.pem -url http://ocsp.myhost.com/ -resp_text -respout resp.der
```

Прочитать OCSP-ответ и вывести в текстовой форме:

```
openssl ocsp -respin resp.der -text
```

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

OCSP-сервер на порте 8888 использует стандартную конфигурацию удостоверяющего центра (см. раздел 3.6) и отдельный сертификат ответчика. Все запросы и ответы выводятся в файл.

```
openssl ocsp -index demoCA/index.txt -port 8888 -rsigner rcert.pem
-CA demoCA/cacert.pem -text -out log.txt
```

Как выше, но закончить работу после обработки одного запроса:

```
openssl ocsp -index demoCA/index.txt -port 8888 -rsigner rcert.pem
-CA demoCA/cacert.pem -nrequest 1
```

Запросить информацию о статусе с использованием внутренне сгенерированного запроса:

```
openssl ocsp -index demoCA/index.txt -rsigner rcert.pem -CA
demoCA/cacert.pem -issuer demoCA/cacert.pem -serial 1
```

Запросить информацию о статусе с использованием запроса, прочитанного из файла, записать ответ в другой файл.

```
openssl ocsp -index demoCA/index.txt -rsigner rcert.pem -CA
demoCA/cacert.pem -reqin req.der -respout resp.der
```

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

## 9 КОМАНДА PKCS7

### 9.1 Описание команды

Команда `pkcs7` переводит файлы формата PKCS#7 в форматы DER или PEM.

### 9.2 Формат ввода команды

```
openssl pkcs7 [-inform PEM|DER] [-outform PEM|DER] [-in filename] [-out filename] [-print_certs] [-text] [-noout] [-engine id]
```

### 9.3 Опции команды

Опция	Описание
<code>-inform DER PEM</code>	Указывает входной формат. Формат DER — структура формата PKCS#7 версии 1.5 в DER-кодировке. PEM (умолчание) — версия DER-формы в кодировке base64 с верхним и нижним ограничителями.
<code>-outform DER PEM</code>	Указывает выходной формат. Опция имеет те же значения, что опция <code>-inform</code> .
<code>-in filename</code>	Указывает файл с входными данными. Если эта опция не указана, данные считываются со стандартного ввода.
<code>-out filename</code>	Указывает файл для записи выходных данных. По умолчанию данные направляются на стандартный вывод.
<code>-print_certs</code>	Выводит все сертификаты и списки отзыва сертификатов, содержащиеся в файле. Перед сертификатами выводятся значения полей <code>subject</code> и <code>issuer</code> в однострочном формате.
<code>-text</code>	Выводит информацию о сертификатах полностью, а не только значения полей <code>subject</code> и <code>issuer</code> .
<code>-noout</code>	Не выводит зашифрованную версию PKCS#7-структуры (или сертификаты, если указана опция <code>-print_certs</code> ).
<code>-engine id</code>	Указывает загружаемый модуль <code>engine</code> (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.

### 9.4 Примеры

Перевести PKCS#7-файл из формата PEM в формат DER:

```
openssl pkcs7 -in file.pem -outform DER -out file.der
```

Вывести все сертификаты, содержащиеся в файле:

```
openssl pkcs7 -in file.pem -print_certs -out certs.pem
```

### 9.5 Примечания

PEM-формат PKCS#7-структуры использует следующий вид верхнего и нижнего ограничителей:

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

```
-----BEGIN PKCS7-----  
-----END PKCS7-----
```

Для совместимости с некоторыми удостоверяющими центрами он также поддерживает следующий вид верхнего и нижнего ограничителей:

```
-----BEGIN CERTIFICATE-----  
-----END CERTIFICATE-----
```

## 9.6 Ограничения

Не существует опции, позволяющей вывести все поля PKCS#7-файла.

Эта команда поддерживает только версию 1.5 формата PKCS#7, описанную в RFC2315.

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

## 10 КОМАНДА PKCS8

### 10.1 Описание команды

Команда `pkcs8` работает с закрытыми ключами формата `PKCS#8`. Она может работать с незашифрованными ключами формата `PKCS#8 PrivateKeyInfo` и с зашифрованными ключами формата `EncryptedPrivateKeyInfo format` с различными алгоритмами формата `PKCS#5` (версии 1.5 и 2.0) и `PKCS#12`.

### 10.2 Формат ввода команды

```
openssl pkcs8 [-topk8] [-inform PEM|DER] [-outform PEM|DER] [-in file- name] [-passin arg]
[-out filename] [-passout arg] [-noiter] [-nocrypt] [-nooc] [-embed] [-nsdb] [-v2 alg] [-v1 alg]
[-engine id]
```

### 10.3 Опции команды

Опция	Описание
<code>-topk8</code>	Как правило, эта команда переводит закрытый ключ формата <code>PKCS#8</code> в закрытый ключ традиционного формата. Данная опция указывает на обратную ситуацию: команда считывает ключ традиционного формата и выводит ключ формата <code>PKCS#8</code> .
<code>-inform DER PEM</code>	Указывает входной формат. Если в качестве входных данных ожидается ключ формата <code>PKCS#8</code> , данная опция указывает на PEM- или DER-версию. В противном случае используется PEM- или DER-формат ключа традиционного формата.
<code>-outform DER PEM</code>	Указывает выходной формат. Опция имеет те же значения, что и опция <code>-inform</code> .
<code>-in filename</code>	Указывает входной файл, из которого следует считать ключ. Если эта опция не указана, ключ считывается со стандартного входа. Если ключ зашифрован, будет запрошена пассфразы.
<code>-passin arg</code>	Источник пароля для входного файла. Для получения дополнительной информации по формату аргумента <code>arg</code> см. раздел 1.6.
<code>-out filename</code>	Указывает выходной файл, в который следует записать ключ. Если эта опция не указывается, ключ выводится на стандартный вывод. Если указаны какие-либо опции зашифрования, будет запрошена пассфразы. Имя выходного файла не должно совпадать с именем входного файла.
<code>-passout arg</code>	Источник пароля для выходного файла. Для получения дополнительной информации по формату аргумента <code>arg</code> см. раздел 1.6.

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

Опция	Описание
-nocrypt	Как правило, ключи формата PKCS#8 являются структурами формата PKCS#8 EncryptedPrivateKeyInfo, используемыми соответствующим алгоритмом шифрования, основанного на пароле. Эта опция указывает, что на вводе или выходе должна быть незашифрованная структура формата PrivateKeyInfo. Эта опция полностью отменяет зашифрование закрытых ключей, и ее следует использовать только при крайней необходимости. Некоторые программы используют незашифрованные закрытые ключи.
-v2 alg	Эта опция дает возможность использования алгоритмов версии 2.0. формата PKCS#5. Как правило, закрытые ключи формата PKCS#8 зашифровываются на пароле при помощи алгоритма под названием pbeWithMD5AndDES-CBC, использующим 56-битное DES-зашифрование, но это был самый мощный алгоритм шифрования, который поддерживался версией 1.5 формата PKCS#8. Данная опция указывает на использование алгоритмов версии 2.0, которые могут использовать любой алгоритм зашифрования, такой, как 168-битный тройной DES или 128-битный RC2, но пока что не все программы поддерживают версию 2.0. Если вы работаете с закрытыми ключами только в рамках криптобиблиотеки OpenSSL, это не имеет значения. Аргумент arg — алгоритм зашифрования, который следует использовать, возможными значениями являются des, des3 и rc2. Рекомендуется использовать des3.
-v1 alg	Эта опция указывает, какой алгоритм версии 1.5 формата PKCS#5 или PKCS#12 следует использовать.
-engine id	Указывает загружаемый модуль engine (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.

## 10.4 Примечания

Зашифрованная форма PKCS#8-файлов в формате PEM использует следующий вид верхнего и нижнего ограничителей:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
-----END ENCRYPTED PRIVATE KEY-----
```

Незашифрованная форма использует:

```
-----BEGIN PRIVATE KEY-----
-----END PRIVATE KEY-----
```

Закрытые ключи, зашифрованные алгоритмами PKCS#5 версии 2.0 с высоким значением количества итераций более надежны, чем ключи, зашифрованные алгоритмами традиционных SSLеау-совместимых форматов. Поэтому если дополнительная безопасность считается важной, следует преобразовывать ключи.

Умолчательное зашифрование всего лишь 56-битное, потому что это зашифрование поддерживают самые современные реализации PKCS#8.

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

Некоторые программы могут использовать PKCS#12-алгоритмы зашифрования на пароле с закрытыми ключами формата PKCS#8: они обрабатываются автоматически, но не существует опции для их получения.

Можно записать DER-закодированные зашифрованные закрытые ключи в формате PKCS#8, потому что информация о зашифровании включена в уровень ASN1, в то время как традиционный формат включает ее на уровне PEM.

## 10.5 Примеры

Перевести закрытый ключ из традиционного формата в формат PKCS#5 v2.0 с помощью тройного алгоритма DES:

```
openssl pkcs8 -in key.pem -topk8 -v2 des3 -out enckey.pem
```

Перевести закрытый ключ в формат PKCS#8, используя алгоритм, совместимый с версией 1.5 формата PKCS#5:

```
openssl pkcs8 -in key.pem -topk8 -out enckey.pem
```

Перевести закрытый ключ в PKCS#8, используя алгоритм, совместимый с форматом PKCS#12 (3DES):

```
openssl pkcs8 -in key.pem -topk8 -out enckey.pem -v1 PBE-SHA1-3DES
```

Прочитать незашифрованный закрытый ключ формата PKCS#8 в DER-кодировке:

```
openssl pkcs8 -inform DER -nocrypt -in key.der -out key.pem
```

Перевести закрытый ключ из любого формата PKCS#8 в традиционный формат:

```
openssl pkcs8 -in pk8.pem -out key.pem
```

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

## 11 КОМАНДА REQ

### 11.1 Описание команды

Команда `req` в основном используется для создания и обработки заявок на сертификаты формата `PKCS#10`. Она также может создавать самоподписанные сертификаты, которые можно использовать, например, в качестве корневых сертификатов удостоверяющих центров.

**Внимание.** При использовании СКЗИ «МагПро КриптоПакет» команду `req` утилиты `openssl` можно использовать для создания ключей. Но следует иметь в виду, что эта команда записывает ключи только в `PKCS#8`-контейнеры. Для создания ключей, которые записываются в аппаратные устройства («Аккорд», «Соболь»), следует использовать программу `mkkey` из состава СКЗИ «МагПро КриптоПакет». Заявки на регистрацию ключей, созданных с помощью программы `mkkey`, создаются с помощью команды `req` утилиты `openssl`. Кроме того, создание ключей с помощью команды `req` возможно только при наличии установленного на компьютере ДСЧ `YARROW` или аппаратного ДСЧ: использование клавиатурного датчика в этом случае невозможно. Программа `mkkey` позволяет создавать ключи при помощи клавиатурного датчика.

### 11.2 Формат ввода команды

```
openssl req [-inform PEM|DER] [-outform PEM|DER] [-in filename] [-passin arg] [-out filename]
[-passout arg] [-text] [-pubkey] [-noout] [-verify] [-modulus] [-new] [-rand file(s)] [-newkey
rsa:bits] [-newkey dsa:file] [-nodes] [-key filename] [-keyform PEM|DER] [-keyout file-
name] [-[md5|sha1|md2|mdc2]] [-config filename] [-subj arg] [-multi-
value-rdn] [-x509] [-days n] [-
set_serial n] [-asn1-kludge] [-newhdr] [-extensions section] [-reqexts section] [-utf8] [-nameopt]
[-batch] [-verbose] [-engine id]
```

### 11.3 Опции команды

Опция	Описание
<code>-inform DER PEM</code>	Определяет входной формат. Опция <code>DER</code> использует ASN.1 DER-закодированную форму, совместимую с <code>PKCS#10</code> . Форма <code>PEM</code> — формат по умолчанию: она состоит из <code>DER</code> -формы, закодированной в <code>base64</code> , с дополнительными верхним и нижним ограничителями.
<code>-outform DER PEM</code>	Определяет выходной формат, опция имеет те же значения, что и опция <code>-inform</code> .
<code>-in filename</code>	Определяет входной файл, из которого следует считывать заявку. Если эта опция не указана, заявка считывается со стандартного входа. Заявка считывается только в том случае, если не указаны опции создания ( <code>-new</code> и <code>-newkey</code> ).
<code>-passin arg</code>	Источник пароля для входного файла. За дополнительной информацией о формате аргумента <code>arg</code> см. раздел 1.6.
<code>-out filename</code>	Указывает имя выходного файла для записи результатов выполнения команды. По умолчанию используется стандартный вывод.
<code>-passout arg</code>	Источник пароля для выходного файла. За дополнительной информацией о формате аргумента <code>arg</code> см. раздел 1.6.

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

Опция	Описание
-text	Выводит заявку в текстовом виде.
-pubkey	Выводит открытый ключ.
-noout	Эта опция предотвращает вывод закодированной версии заявки.
-modulus	Эта опция выводит значение модулюса открытого ключа, содержащегося в заявке.
-verify	Проверяет подпись под заявкой.
-new	Эта опция создает новую заявку на сертификат. Она запрашивает у пользователя значения соответствующих полей. Конкретные запрашиваемые поля, а также их максимальные и минимальные размеры определяются в конфигурационном файле, как и любые запрашиваемые расширения. Если опция -key не указана, данная опция создаст новый закрытый ключ RSA, используя информацию, содержащуюся в конфигурационном файле.
-rand file(s)	Указывает файл или файлы, содержащие случайные данные, которые используются для инициализации генератора случайных чисел. Несколько файлов можно указать через разделитель, который определяется операционной системой: ; для MS-Windows, , для OpenVMS и : для всех остальных.
-newkey arg	Эта опция создает новую заявку на сертификат и новый закрытый ключ. Аргумент arg имеет форму алгоритм:параметры. Для алгоритма ГОСТ Р 34.10-2001 поддерживаются следующие наборы параметров: для ключей подписи А, В, С; для ключей обмена ключами ХА, ХВ. Аргумент опции -newkey может выглядеть, например, так: gost2001:А.
-key filename	Указывает файл, из которого следует считать закрытый ключ.
-keyform PEM DER	Формат закрытого ключа, указанного в качестве аргумента опции -key. По умолчанию PEM.
-keyout filename	Указывает файл, в который следует записать созданный закрытый ключ. Если эта опция не указана, используется файл, указанный в конфигурационном файле.
-nodes	Если эта опция указана, то если создается закрытый ключ, он не шифруется.
-config filename	Это позволяет указать альтернативный конфигурационный файл. Данная опция имеет больший приоритет, чем имя файла, заданное при компиляции или имя, указанное в переменной среды OPENSSL_CONF.
-subj arg	Устанавливает значение поля subject для новой заявки или замещает значение этого поля при обработке заявки. Формат аргумента arg должен быть /type0=value0/type1=value1/type2=..., символы могут быть экранированы знаком \ (обратный слэш), пробелы не опускаются.

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

Опция	Описание
-multivalue-rdn	Эта опция указывает, что аргумент опции -subj необходимо интерпретировать с полной поддержкой многозначных RDN. Пример: /DC=org/DC=OpenSSL/DC=users/UID=123456+CN=John Doe Если опция -multi-rdn не используется, то значение поля UID будет 123456+CN=John Doe.
-x509	Эта опция создает самоподписанный сертификат вместо заявки. Это обычно используется для создания тестового сертификата или самоподписанного корневого сертификата удостоверяющего центра. Расширения, добавляемые в сертификат (если таковые есть) указываются в конфигурационном файле. Если не указано другого с помощью опции set_serial, в качестве серийного номера будет указан 0.
-days n	Если указана опция -x509, данная опция указывает срок действия сертификата в днях. По умолчанию 30 дней.
-set serial n	Серийный номер для выпускаемого самоподписанного сертификата. Может быть указан как десятичная величина, или как шестнадцатичная с префиксом 0x. Указывать отрицательные серийные номера можно, но не рекомендуется.
-extensions section -reqexts section	Эти опции указывают альтернативные секции для включения расширений в сертификат (если указана опция -x509) или в заявку. Это позволяет использовать несколько различных секций в одном и том же конфигурационном файле для создания заявок с различными целевыми назначениями.
-utf8	Эта опция указывает, что значения полей следует интерпретировать как строки в кодировке UTF8, по умолчанию они интерпретируются в кодировке ASCII. Это означает, что значения полей, считанные с терминала или из конфигурационного файла, должны быть корректными UTF-8 строками.
-nameopt option	Опция указывает, как должны выводиться значения полей subject и issuer. Аргументом может быть одна опция или несколько, разделенных запятыми. Для установки нескольких опций можно также несколько раз использовать свитч -nameopt. Для дополнительной информации см. раздел 16.
-asn1-kludge	По умолчанию команда req выводит заявки, не содержащие атрибутов, в корректном формате PKCS#10. Но некоторые удостоверяющие центры принимают только заявки, не содержащие атрибутов, в некорректном формате. Эта опция создает такой некорректный формат. Точнее атрибуты в заявке формата PKCS#10 определены как атрибут SET OF. Они не являются опциональными, поэтому, если атрибуты в заявке отсутствуют, они должны быть закодированы как пустой SET OF. Эта некорректная форма не включает такой пустой SET OF, а корректная форма включает. Следует отметить, что очень немногие удостоверяющие центры требуют использования данной опции.

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

Опция	Описание
-newhdr	Добавляет слово NEW в верхний и нижний ограничители в PEM-файле заявки. Это необходимо для некоторых программ (сертификационный сервер Netscape) и некоторых удостоверяющих центров.
-batch	Неинтерактивный режим.
-verbose	Вывести дополнительную информацию о производимых операциях.
-engine id	Указывает на загружаемый модуль engine (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.

## 11.4 Формат конфигурационного файла

Опции конфигурации указываются в разделе req конфигурационного файла. Как и в любом конфигурационном файле, если некая величина не указана в конкретном разделе (например, разделе req) то она ищется также в начальном непоименованном разделе или в разделе по умолчанию.

Доступные опции подробно описаны ниже.

Опция	Описание
input_password output_password	Пароли для входного файла закрытого ключа (если таковой присутствует) и для выходного файла закрытого ключа (если таковой создается). Опции командной строки passin и passout имеют больший приоритет, чем опции, указанные в конфигурационном файле.
default_bits	Эта опция определяет умолчательный размер ключа в битах. Если опция не указана, используется 512. Опция применяется, если в командной строке указана опция -new. Опция командной строки -newkey имеет больший приоритет.
default_keyfile	Это умолчательное имя для выходного файла закрытого ключа. Если эта опция не указана, ключ записывается в стандартный выход. Опция командной строки -keyout имеет больший приоритет.
oid_file	Эта опция указывает на файл, содержащий дополнительные OID (OBJECT IDENTIFIERS). Каждая строка файла должна иметь следующий формат: OID в численном виде, пробел, короткое имя, пробел, длинное имя.
oid_section	Эта опция указывает на раздел конфигурационного файла, содержащий дополнительные OID. Каждая строка раздела должна иметь формат: короткое имя OID=численный вид OID. В случае использования этой опции короткое и длинное имена совпадают.
RANDFILE	Файл, используемый для считывания и записи информации для инициализации генератора случайных чисел.
encrypt_key	Если эта опция установлена в 0, то создаваемый закрытый ключ не зашифровывается. Эта опция эквивалентна опции командной строки -nodes.

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

Опция	Описание
default_md	Опция обязательно должна быть указана, но используется только в том случае, если ключ удостоверяющего центра имеет алгоритм, позволяющий использовать разные алгоритмы хэширования (RSA). Значение опции представляет собой название алгоритма хэширования, используемого для подписи сертификатов (в противном случае оно может быть любым).
string_mask	Эта опция маскирует использование некоторых типов строк в некоторых полях. Существует несколько возможных значений данной опции. Значение default (оно же по умолчанию) использует PrintableStrings, T61Strings и BMPStrings. При указании значения pkix будут использоваться только PrintableStrings and BMPStrings в соответствии с PKIX рекомендацией в RFC2459. Если указывается значение utf8only, используются только UTF8Strings: это рекомендация PKIX в RFC2459 после 2003 г. Наконец, значение nombstr использует только PrintableStrings и T61Strings: некоторые программы встречаются затруднения с BMPStrings and UTF8Strings, особенно Netscape. Для поддержки кириллицы в полях сертификата необходимо указывать либо значение pkix (тогда формируемые заявки будут использовать тот же набор строковых типов, что и заявки, создаваемые Active-X элементом Xenroll в Windows), либо utf8only
req_extensions	Указывает раздел конфигурационного файла, содержащий список расширений, которые необходимо добавить в заявку на сертификат. Командно-строчный свитч -reqexts имеет больший приоритет.
x509_extensions	Указывает раздел конфигурационного файла, содержащий список расширений, которые необходимо добавить в сертификат, созданный с использованием опции -x509. Командно-строчный свитч -extensions имеет больший приоритет.
prompt	Если эта опция имеет значение no, отключается запрашивание полей сертификата и просто считывает значения полей прямо с конфигурационного файла. Кроме того, изменяется ожидаемый формат разделов distinguished_name и attributes.
utf8	Эта опция указывает, что значения полей следует интерпретировать как строки в кодировке UTF8, по умолчанию они интерпретируются в кодировке ASCII. Это означает, что значения полей, считанные с терминала или из конфигурационного файла, должны быть корректными UTF-8 строками.
attributes	Указывает раздел конфигурационного файла, содержащий атрибуты заявки: его формат совпадает с форматом distinguished_name. Как правило, они содержат типы challengePassword или unstructuredName. В настоящее время они игнорируются утилитами OpenSSL, выполняющими подписывание заявки, но некоторые удостоверяющие центры могут требовать их наличия.
distinguished_name	Указывает раздел конфигурационного файла, содержащий поля структуры distinguished name, которые следует запрашивать при создании сертификата или заявки. Формат описан в разделе 11.5.

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

## 11.5 Формат разделов конфигурационного файла distinguished name и attribute

Существуют два различных формата для разделов distinguished name и attribute. Если опция prompt установлена в значение по, эти разделы просто содержат наименования и значения полей, например

```
CN=Ivanov Ivan Ivanovich OU=Company emailAddress=someone@somewhere.org
```

Это позволяет внешним программам (например, программам с графическим интерфейсом) создавать файл-шаблон со всеми названиями и значениями полей и просто передавать этот файл команде req. Пример такого рода конфигурационного файла содержится в разделе .

Или же, если опция prompt не указана или не установлена в по, файл содержит информацию о запросах полей. Она состоит из строк вида:

```
fieldName="prompt"
fieldName_default="значение поля по умолчанию"
fieldName_min= 2
fieldName_max= 4
```

Здесь fieldName — наименование используемого поля, например commonName или CN. Строка символов "prompt" используется для запроса к пользователю ввести соответствующую информацию. Если пользователь ничего не вводит, используется умолчательное значение поля. Если и умолчательного значения не указано, поле опускается. Поле может быть опущено и в том случае, если величина по умолчанию присутствует, но пользователь просто введет символ '.'.

Количество введенных символов должно быть в пределах fieldName\_min and fieldName\_max: могут также быть дополнительные ограничения в зависимости от рассматриваемого поля (например, значение поля countryName может быть только двухбуквенным и соответствовать типу PrintableString).

Некоторые поля (например organizationName) могут использоваться в структуре DN больше одного раза. Это представляет собой проблему, потому что конфигурационные файлы не распознают одно и то же имя, встречающееся дважды. Чтобы избежать этой проблемы, если fieldName содержит несколько символов, за которыми следует точка, они будут проигнорированы. Поэтому, например, второе поле organizationName может быть введено как 1.organizationName.

Корректные разрешенные наименования полей могут быть любыми короткими или длинными именами OID. Они компилируются в OpenSSL и включают обычные величины, такие как commonName, countryName, localityName, organizationName, organizationUnitName, stateOrProvinceName. Дополнительно введены также emailAddress, name, surname, givenName initials и dnQualifier.

Дополнительные OID могут быть определены с помощью опций конфигурационного файла oid\_file и oid\_section. Любые дополнительные поля обрабатываются как строки типа DirectoryString.

## 11.6 Примеры

Просмотреть и проверить заявку на сертификат:

```
openssl req -in req.pem -text -verify -noout
```

Генерировать заявку на сертификат с явным указанием ключа:

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

```
openssl req -new -key key.pem -out req.pem
```

То же самое, но с генерацией ключа:

```
openssl req -newkey gost2001:A -keyout key.pem -out req.pem
```

Создать самоподписанный корневой сертификат:

```
openssl req -x509 -newkey gost2001:A -keyout key.pem -out req.pem
```

Пример файла, который указывается в опции `oid_file`:

```
1.2.3.4      shortName      A longer Name
1.2.3.6      otherName      Other longer Name
```

Пример раздела конфигурационного файла, который указывается в опции `oid_section` с использованием переменного расширения:

```
testoid1=1.2.3.5
testoid2=${testoid1}.6
```

Образец конфигурационного файла, обеспечивающего вывод запросов значений полей:

```
[ req ]
default_bits          = 1024
default_keyfile       = privkey.pem
distinguished_name    = req_distinguished_name
attributes            = req_attributes
x509_extensions       = v3_ca

dirstring_type = nobmp

[ req_distinguished_name ]
countryName           = Country Name (2 letter code)
countryName_default   = AU
countryName_min       = 2
countryName_max       = 2

localityName          = Locality Name (eg, city)

organizationalUnitName = Organizational Unit Name (eg, section)

commonName            = Common Name (eg, YOUR name)
commonName_max        = 64

emailAddress          = Email Address
emailAddress_max      = 40

[ req_attributes ]
challengePassword     = A challenge password
challengePassword_min = 4
challengePassword_max = 20

[ v3_ca ]

subjectKeyIdentifier=hash
```

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

```
authorityKeyIdentifier=keyid:always,issuer:always
basicConstraints = CA:true
```

Образец конфигурационного файла с указанными значениями полей:

```
RANDFILE                = $ENV::HOME/.rnd

[ req ]
default_bits             = 1024
default_keyfile          = keyfile.pem
distinguished_name      = req_distinguished_name
attributes               = req_attributes
prompt                  = no
output_password         = mypass

[ req_distinguished_name ]
C                        = GB
ST                       = Test State or Province
L                        = Test Locality
O                        = Organization Name
OU                       = Organizational Unit Name
CN                       = Common Name
emailAddress             = test@email.address

[ req_attributes ]
challengePassword       = A challenge password
```

## 11.7 Примечания

Как правило, верхний и нижний ограничители в формате PEM выглядят как:

```
-----BEGIN CERTIFICATE REQUEST-----
-----END CERTIFICATE REQUEST-----
```

Некоторым программам (например некоторым версиям сертификационного сервера Netscape) необходим другой вид ограничителей:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
-----END NEW CERTIFICATE REQUEST-----
```

Такие ограничители создаются при использовании опции `-newhdr`, но в обратную сторону они совместимы. Обе формы при вводе принимаются прозрачно.

Заявки на сертификаты, создаваемые в Microsoft IE Active-X элементом Xenroll, включают в себя добавленные расширения, в том числе расширение `KeyUsage`, которое определяет тип ключа (только подпись или общего назначения) и все дополнительные `OID`, введенные скриптом в расширении `extendedKeyUsage`.

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

## 11.8 Диагностика

Часто выводятся следующие сообщения:

```
Using configuration from /some/path/openssl.cnf
Unable to load config info
```

Через некоторое время выводится:

```
unable to find 'distinguished_name' in config
problems making Certificate Request
```

Первое сообщение — ключевое: не обнаружен конфигурационный файл! Некоторые операции (такие как просмотр заявки на сертификат) не требуют конфигурационного файла, поэтому его использование необязательно. Но создание сертификатов или заявок требует конфигурационного файла. Это можно считать ошибкой.

Еще одно озадачивающее сообщение:

```
Attributes:
a0:00
```

Это сообщение выводится, когда никаких attributes не указано, а заявка включает корректную пустую структуру SET OF (DER-форма которой выглядит как 0xa0 0x00). Если вы видите только:

```
Attributes:
```

Значит, структура SET OF отсутствует и кодировка технически некорректна (но допустима). Для получения дополнительной информации см. описание опции `-asn1-kludge`.

## 11.9 Переменные среды

Переменная `OPENSSL_CONF`, если она определена, позволяет определить расположение дополнительного конфигурационного файла. Опция командной строки `-config` имеет больший приоритет. Для совместимости переменная среды `SSLEAY_CONF` может использоваться для той же цели, но ее использование не рекомендуется.

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

## 12 КОМАНДА SMIME

### 12.1 Описание команды

Команда `smime` обрабатывает почтовые сообщения типа S/MIME. Она может зашифровать, расшифровывать, подписывать и проверять такие сообщения.

### 12.2 Формат ввода команды

```
openssl smime [-encrypt] [-decrypt] [-sign] [-verify] [-pk7out] [-gost89] [-in file] [-certfile file]
[-signer file] [-recip file] [-inform SMIME|PEM|DER] [-passin arg] [-inkey file] [-out file] [-outform SMIME|PEM|DER]
[-content file] [-to addr] [-from ad] [-subject s] [-text] [-rand file(s)] [cert.pem]...
```

### 12.3 Опции команды

Существует пять операционных опций, которые устанавливают тип производимой операции. Значение остальных опций варьируется в зависимости от типа операции.

Опция	Описание
-encrypt	Зашифровывает почту для указанных сертификатов получателей. Входным файлом является незашифрованное сообщение. Выходной файл — зашифрованное почтовое сообщение в формате MIME.
-decrypt	Расшифровать почту с использованием указанного сертификата и закрытого ключа. В качестве входного файла ожидается зашифрованное почтовое сообщение в формате MIME. В выходной файл записывается расшифрованное сообщение.
-sign	подписывает почту с использованием указанного сертификата и закрытого ключа. Входным файлом является сообщение, которое необходимо подписать. В выходной файл записывается подписанное сообщение в формате MIME.
-verify	Проверяет подписанную почту. Ожидает подписанное почтовое сообщение в качестве входного файла и выводит подписанные данные. Поддерживаются как незашифрованные, так и зашифрованные подписанные файлы.
-pk7out	Считывает входное сообщение и записывает в выходной файл PKCS#7-структуру в PEM-формате.
-in filename	Входное сообщение, которое нужно подписать или зашифровать, или сообщение в формате MIME, которое нужно расшифровать или под которым нужно проверить подпись.
-inform SMIME PEM DER	Определяет входной формат PKCS#7-структуры. По умолчанию — SMIME, для считывания сообщений в формате S/MIME. Указание на форматы PEM или DER заставляет ожидать в качестве входного файла PKCS#7-структуры в соответствующем формате. Сейчас эта опция влияет только на входной формат PKCS#7-структуры, если никакой PKCS#7-структуры не вводится (например если указаны опции -encrypt или -sign), эта опция игнорируется.

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

Опция	Описание
-content filename	Определяет файл, содержащий отдельное (неподписанное) сообщение, используется только вместе с опцией -verify. Имеет смысл только в том случае, если PKCS#7-структура использует форму отдельной подписи, в которую не включено содержание сообщения. Эта опция замещает содержанием указанного файла содержание любого входного сообщения в формате S/MIME; она использует multipart/signed MIME content type.
-text	Эта опция добавляет MIME-заголовки простого текста (text/plain) в указанное сообщение при зашифровании или подписывании. При расшифровании или проверке подписи она удаляет эти заголовки: если зашифрованное или проверяемое сообщение не является сообщением MIME-типа text/plain, выводится сообщение об ошибке.
-add	Добавляет вторую подпись к уже подписанному сообщению. Входными данными должно быть подписанное PKCS#7-сообщение в формате PEM или DER. Добавление дополнительных подписей к сообщениям в формате S/MIME не поддерживается.
-CAfile file	Файл, содержащий доверенный сертификат удостоверяющего центра. Данная опция используется только с опцией -verify.
-CApath dir	Каталог, содержащий доверенные сертификаты удостоверяющих центров. Используется только с опцией -verify. Этот каталог должен быть стандартным каталогом сертификатов, то есть с каждым сертификатом должна быть связана хэш-сумма поля subject name.
-gost89	Используемый алгоритм зашифрования. Используется только с опцией -encrypt. При шифровании с помощью алгоритмов ГОСТ данная опция является обязательной.
-nointern	При проверке подписи, как правило, сертификат отправителя ищется среди сертификатов, включенных в сообщение (если таковые есть). Если указана данная опция, используются только сертификаты, указанные в опции -certfile. Сертификаты, включенные в сообщение, могут использоваться в качестве недоверенных сертификатов удостоверяющих центров.
-noverify	Не проверять сертификат отправителя подписанного сообщения.
-nochain	Не выполнять проверку цепочки доверия сертификатов отправителя, то есть не использовать сертификаты, включенные в подписанное сообщение, в качестве недоверенных сертификатов удостоверяющего центра.
-nosigs	Не пытаться проверять подписи под сообщением.
-nocerts	При подписывании сообщения сертификат отправителя, как правило, включается в сообщение. При указании данной опции сертификат отправителя в сообщение не включается. Это уменьшает размер подписанного сообщения, но получатель должен иметь на своем компьютере копию сертификата отправителя (например, переданную с помощью опции -certfile).

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

Опция	Описание
-binary	Как правило, входное сообщение переводится в «канонический» формат, использующий CR и LF в качестве концов строк, как требует спецификация S/MIME. При указании данной опции перевода в такой формат не производится. Это полезно при передаче бинарных данных, которые передаются не в MIME-формате.
-nodetach	«Непрозрачное» подписание сообщения: эта форма более устойчива при почтовой передаче, но ее не смогут прочитать почтовые программы, не поддерживающие формат S/MIME. Если эта опция не указана, выполняется «прозрачное» подписание с использованием типа MIME multipart/signed.
-certfile file	Позволяет указать дополнительные сертификаты. При подписывании сообщения эти сертификаты будут включены в сообщение. При проверке сообщения среди этих сертификатов будет искаться сертификат отправителя. Сертификаты должны быть в PEM-формате.
-signer file	Сертификат отправителя при подписи сообщения. При проверке сообщения сертификаты отправителя будут записаны в этот файл, если проверка была успешной.
-recip file	Сертификат получателя при расшифровании сообщения. Этот сертификат должен принадлежать одному из получателей сообщения, иначе выводится сообщение об ошибке.
-inkey file	Закрытый ключ, который следует использовать при подписывании или расшифровании сообщения. Закрытый ключ должен соответствовать сертификату. Если эта опция не указана, закрытый ключ должен быть включен в файл сертификата, указанный в опции -recip или -signer.
-passin arg	Источник пароля для закрытого ключа. Для получения дополнительной информации о формате аргумента arg см. раздел 1.6.
-rand file(s)	Файл или файлы, содержащие случайные данные, используемые для инициации генератора случайных чисел. Несколько файлов можно указать через разделитель, который определяется операционной системой: ; для MS-Windows, , для OpenVMS и : для всех остальных.
cert.pem...	Один или больше сертификатов получателей. Используется при зашифровании сообщения.
-to, -from, -subject	Соответствующие заголовки почтовых сообщений. Они включаются снаружи подписанной части сообщения, чтобы их можно было включить вручную. Многие почтовые клиенты, работающие с форматом S/MIME, проверяют, совпадает ли почтовый адрес, указанный в сертификате отправителя, с почтовым адресом отправителя.
-policy имя	Включает политику проверки сертификатов с указанным именем
-purpose имя	Требует чтобы сертификат имел указанную область применения. Имена областей применения, определенные в утилите openssl sslclient, sslserver, nssserver, smimeencrypt, smimesign, crlsign, any.
-ignore_critical	Игнорировать при проверке сертификата неизвестные расширения X509v3, помеченные как критичные.
-crl_check	Выполнять проверку на наличие сертификата подписи в соответствующем списке отзыва

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

Опция	Описание
-crl_check_all	Выполнять проверку на наличие соответствующем списке отзыва всех сертификатов из цепочки доверия
-policy_check	Включает проверку соответствия сертификатов политике указанной в сертификате удостоверяющего центра
-explicit_policy	Требовать явного указания политики
-x509_strict	Строгая проверка соответствия сертификатов формату x509
-policy_print	выводить информацию о политике удостоверяющего центра

## 12.4 Примечания

Заголовки MIME-сообщения при отправке не должны отделяться от остального содержания пустыми строками. Некоторые почтовые программы автоматически добавляют такие пустые строки. Направить почту непосредственно в программу sendmail — один из способов получить корректный формат.

Подписываемое и зашифрованное сообщение должно включать необходимые MIME-заголовки, иначе многие почтовые клиенты не смогут его корректно воспроизвести (или вообще не смогут). Вы можете использовать опцию -text для автоматического добавления заголовков.

«Подписанное и зашифрованное» сообщение — сообщение, сначала подписанное, затем зашифрованное. Такое сообщение можно получить, зашифровав уже подписанное сообщение (см. раздел 12.6).

Оригинальная версия не поддерживает возможность создания нескольких подписей под одним почтовым сообщением в формате S/MIME, но может проверять корректность сообщений с несколькими подписями.

Функциональность создания второй и последующих подписей (опция -add) добавлена в МагПро КриптоПакет.

Некоторые почтовые клиенты не в состоянии обрабатывать сообщения, содержащие более одной подписи. Поэтому применять этот режим работы следует только в условиях, когда всё используемое программное обеспечение поддерживает этот режим.

Опции -encrypt и -decrypt отражают обычное использование соответствующих функций в почтовых клиентах. Строго говоря, эти опции работают с разновидностью enveloped data формата PKCS#7. PKCS#7 encrypted data используется для других целей.

## 12.5 Коды выхода

- 1 Операция выполнена полностью успешно.
- 2 Ошибка при обработке опций команды.
- 3 Один из входных файлов не прочитан.
- 4 Ошибка при создании PKCS#7-файла или считывании MIME-сообщения.
- 5 При проверке сообщение признано корректным, но произошла ошибка при записи одного из сертификатов отправителя.

## 12.6 Примеры

Создать «прозрачно» подписанное сообщение:

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

```
openssl smime -sign -in message.txt -text -out mail.msg -signer
mycert.pem
```

Создать «непрозрачно» подписанное сообщение:

```
openssl smime -sign -in message.txt -text -out mail.msg -nodetach
-signer mycert.pem
```

Создать подписанное сообщение, включить несколько дополнительных сертификатов и считать закрытый ключ из другого файла:

```
openssl smime -sign -in in.txt -text -out mail.msg -signer
mycert.pem -inkey mykey.pem -certfile mycerts.pem
```

Отправить подписанное сообщение в Unix-подобными ОС прямо в программу sendmail, включая заголовки:

```
openssl smime -sign -in in.txt -text -signer mycert.pem -from
steve@openssl.org -to someone@somewhere -subject ''Signed message''
| sendmail someone@somewhere
```

Проверить сообщение и в случае успешной проверки сохранить сертификат отправителя в файле:

```
openssl smime -verify -in mail.msg -signer user.pem -out
signedtext.txt
```

Отправить зашифрованное сообщение, используя алгоритм gost89:

```
openssl smime -encrypt -in in.txt -from steve@openssl.org -to
someone@somewhere -subject <<Encrypted message>> -gost89 user.pem -out
mail.msg
```

Подписать и зашифровать сообщение:

```
openssl smime -sign -in ml.txt -signer my.pem -text | openssl smime
-encrypt -out mail.msg -from steve@openssl.org -to someone@somewhere
-subject ''Signed and Encrypted message'' -gost89 user.pem
```

**Примечание:** команда зашифрования не включает опцию -text, потому что зашифровываемое сообщение уже включает MIME-заголовки.

Расшифровать сообщение:

```
openssl smime -decrypt -in mail.msg -recip mycert.pem -inkey key.pem
```

Выходными данными из подписывающей программы Netscape является PKCS#7-структура в формате с отдельной подписью. Вы можете использовать эту программу, чтобы проверить такую подпись, разбив на строки структуру в кодировке base64, окружив ее ограничителями:

```
-----BEGIN PKCS7-----
-----END PKCS7-----
```

и воспользовавшись командой

```
openssl smime -verify -inform PEM -in signature.pem -content
content.txt
```

Вы также можете декодировать подпись из кодировки base64 и воспользоваться командой

```
openssl smime -verify -inform DER -in signature.der -content
content.txt
```

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

## 13 КОМАНДА S\_CLIENT

### 13.1 Описание команды

Команда `s_client` реализует SSL/TLS-клиент общего назначения, устанавливающий соединение с отдаленным SSL/TLS-сервером. Это очень полезный диагностический инструмент для SSL-серверов.

### 13.2 Формат ввода команды

```
openssl s_client [-connect host:port] [-verify depth] [-cert filename] [-certform DER|PEM] [-key filename] [-keyform DER|PEM] [-pass arg] [-CApath directory] [-CAfile filename] [-reconnect] [-pause] [-showcerts] [-debug] [-msg] [-nbio_test] [-state] [-nbio] [-crlf] [-ign_eof] [-quiet] [-ssl2] [-ssl3] [-tls1] [-no_ssl2] [-no_ssl3] [-no_tls1] [-bugs] [-cipher cipherlist] [-starttls protocol] [-engine id] [-rand file(s)]
```

### 13.3 Опции команды

Опция	Описание
<code>-connect host:port</code>	указывает адрес сервера, с которым нужно установить соединение, и опционально порт. Если не указано другое, делается попытка установить связь с локальным хостом, порт 4433.
<code>-host имя</code>	указывает адрес сервера с которым надо установить соединение. Устаревшая опция, рекомендуется использовать <code>-connect</code>
<code>-port число</code>	Указывает номер порта, с которым нужно установить соединение. Устаревшая опция, рекомендуется использовать <code>-connect</code>
<code>-cert certname</code>	Указывает сертификат, который следует использовать, если таковой запрашивается сервером. По умолчанию сертификат не используется.
<code>-certform format</code>	Формат используемого сертификата: DER или PEM. По умолчанию PEM.
<code>-crl_check</code>	Включает проверку наличия сертификата сервера в списке отзыва.
<code>-crl_check_all</code>	Включает проверку всех сертификатов в цепочке доверия по соответствующим спискам отзывов
<code>-key keyfile</code>	Закрытый ключ, который следует использовать. Если не указан, будет использоваться файл сертификата.
<code>-keyform format</code>	Формат закрытого ключа: DER или PEM. По умолчанию PEM.
<code>-pass arg</code>	Источник пароля для закрытого ключа. Для получения дополнительной информации о формате аргумента <code>arg</code> см. раздел 1.6.
<code>-verify depth</code>	Используемая глубина верификации. Опция определяет максимальную длину цепочки сертификатов и включает проверку серверного сертификата. В настоящее время операция проверки продолжается и после появления сообщений об ошибках, чтобы диагностировать все проблемы в цепочке сертификатов. В качестве побочного эффекта соединение не обрывается в случае, если серверный сертификат будет признан некорректным.

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

Опция	Описание
-CApath directory	Каталог, который следует использовать для проверки серверного сертификата. Этот каталог должен быть в «хэш-формате», см. раздел 15 для получения дополнительной информации. Сертификаты из этого каталога также используются для построения цепочки для проверки клиентского сертификата.
-CAfile file	Файл, содержащий доверенные сертификаты, которые следует использовать при аутентификации сервера и во время попыток построить цепочку клиентских сертификатов.
-reconnect	Указывает, что необходимо связываться с одним и тем же сервером 5 раз с одним и тем же сессионным ID. Эту опцию можно использовать при тестировании кэширования сессий.
-pause	Устанавливает односекундную паузу между каждым вызовом функций read и write.
-showcerts	Вывести всю цепочку сертификатов, присланную сервером. Как правило, выводится только сам сертификат сервера.
-prexit	Вывести информацию о сессии при завершении работы программы. При использовании этой опции попытка вывода информации о сессии предпринимается всегда, даже если установить соединение не удастся. Эта опция полезна, потому что используемый шифр-сьют может быть пересогласован или соединение может быть не установлено из-за того, что требуется сертификат клиента или таковой запрашивается после попытки обратиться к определенному URL. Примечание: результат работы этой опции не всегда является точным, потому что соединение, возможно, так и не удастся установить.
-state	Выводит состояния SSL-сессии.
-debug	Вывести подробную отладочную информацию, включающую шестнадцатеричный дамп всего трафика.
-msg	Вывести все сообщения протокола с шестнадцатеричным дампом.
-nbio_test	Тестирует неблокирующий ввод-вывод
-nbio	Включает неблокирующий ввод-вывод
-crlf	Эта опция переводит символ конца строки с терминала в CR+LF, как требуют некоторые серверы.
-ign_eof	Подавляет закрытие соединения при обнаружении конца файла стандартного ввода.
-quiet	Подавляет вывод сессионной информации и информации о сертификате. Как следствие, отключает действие опции — -ign_eof.

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

Опция	Описание
-ssl2, -ssl3, -tls1, -no_ssl2, -no_ssl3, -no_tls1	Эти опции отключают использование определенных SSL- и TLS-протоколов. По умолчанию начальное «рукопожатие» использует метод, который должен быть совместимым со всеми серверами и позволять им использовать протоколы SSL v3, SSL v2 или TLS как следует. К сожалению, существует и используется множество старых или сломанных серверов, которые не могут выполнять эту процедуру и с которыми поэтому не удастся установить соединение. Некоторые серверы работают только если отключен протокол TLS с помощью опции -no_tls, другие поддерживают только вторую версию протокола SSL, и им нужна опция -ssl2.
-bugs	В распространенных реализациях SSL и TLS есть несколько известных ошибок. Указание этой опции включает разнообразные методы их обхода.
-cipher cipherlist	Эта опция позволяет модифицировать список допустимых шифр-сьютов, отправляемый клиентом. Хотя сервер определяет, какой шифр-сьют следует использовать, он обязан использовать первый поддерживаемый шифр-сьют из отправленного клиентом списка. Для получения дополнительной информации см. руководство по команде ciphers.
-starttls protocol	Отправляет протоколно-специфичное сообщение (сообщения) для переключения в TLS для коммуникации. Аргумент protocol - ключевое слово для соответствующего протокола. В настоящее время поддерживаются только ключевые слова smtp, pop3, imap, and ftp.
-engine id	Указывает загружаемый модуль engine (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.
-rand file(s)	Файл или файлы, содержащие случайные данные, используемые для инициации генератора случайных чисел. Несколько файлов можно указать через разделитель, который определяется операционной системой: ; для MS-Windows, , для OpenVMS и : для всех остальных.
-mtu число	Устанавливает максимальный размер пакета протокола TCP в указанное значение и запрещает операцию согласования размера пакета с промежуточными узлами.
-serverpref	Задаёт использование шифр-сьюта предпочитаемого сервером (только SSLv2)

### 13.4 Команды, выводимые при установленном соединении

Если установлено соединение с SSL-сервером, то выводятся все данные, полученные с сервера, и все нажатия на клавиши будут переданы на сервер. При интерактивном использовании (что означает, что не были указаны ни опция -quiet, ни опция -ign\_eof) то сессия будет пере-согласована, если строка начинается с R, а если строка начинается с Q, или достигнут конец файла, соединение будет закрыто.

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

## 13.5 Примечания

Команда `s_client` может применяться для отладки SSL-сервером. Для установления соединения с SSL HTTP-сервером обычно используется команда

`openssl s_client -connect servername:443` (протокол `https` использует порт 443). Если соединение установлено успешно, можно дать `http`-команду, например `"GET /"` — запрос веб-страницы.

Если «рукопожатие» неудачно, этому может быть несколько возможных причин, если ничего очевидного, вроде отсутствия клиентского сертификата, можно попробовать использовать опции `-bugs`, `-ssl2`, `-ssl3`, `-tls1`, `-no_ssl2`, `-no_ssl3`, `-no_tls1` в том случае, если ошибка на сервере. Особенно вы можете поиграть с этими опциями перед отправкой баг-репорта в список рассылки `OpenSSL`.

Часто встречающаяся проблема при попытке заставить работать клиентские сертификаты — веб-клиент жалуется, что у него нет сертификатов, или выдает пустой список для выбора. Это, как правило, происходит потому, что сервер не отправляет название удостоверяющего центра клиента в своем списке «приемлемых удостоверяющих центров», когда он запрашивает сертификат. При использовании программы `s_client` можно просмотреть и проверить список приемлемых удостоверяющих центров. Но некоторые серверы запрашивают клиентскую аутентификацию только после запроса определенного URL. Чтобы получить список в этом случае, необходимо воспользоваться опцией `-rgetit` и отправить `http`-запрос соответствующей страницы.

Если сертификат указан в командной строке с использованием опции `-cert`, он не будет использоваться, если только сервер специально не запросит клиентский сертификат. Таким образом, простое включение клиентского сертификата в командную строку не является гарантией, что сертификат работает.

Если возникают трудности при просмотре серверного сертификата, следует воспользоваться опцией `-showcerts`, чтобы просмотреть всю цепочку сертификатов.

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

## 14 КОМАНДА S\_SERVER

### 14.1 Описание команды

Команда `s_server` реализует SSL/TLS-сервер общего назначения, который отвечает на запросы на установление соединения на определенном порте с использованием протокола SSL/TLS.

### 14.2 Формат ввода команды

```
openssl s_server [-accept port] [-context id] [-verify depth] [-Verify depth] [-cert filename] [-certform DER|PEM] [-key keyfile] [-keyform DER|PEM] [-pass arg] [-dcert filename] [-dcertform DER|PEM] [-dkey keyfile] [-dkeyform DER|PEM] [-dpass arg] [-dhparam filename] [-nbio] [-nbio_test] [-crlf] [-debug] [-msg] [-state] [-CApath directory] [-CAfile filename] [-nocert] [-cipher cipherlist] [-quiet] [-no_tmp_rsa] [-ssl2] [-ssl3] [-tls1] [-no_ssl2] [-no_ssl3] [-no_tls1] [-no_dhe] [-bugs] [-hack] [-www] [-WWW] [-HTTP] [-engine id] [-id_pre- fix arg] [-rand file(s)]
```

### 14.3 Опции команды

Опция	Описание
-accept port	ТСР-порт, который следует прослушивать в ожидании запросов на соединения. Если не указан, используется порт 4433.
-context id	устанавливает идентификатор контекста SSL. В качестве значения может быть любая строка. Если эта опция не указана, используется значение по умолчанию.
-cert certname	Сертификат, который следует использовать. Большинство серверных шифр-сьютов требуют использования сертификатов, но некоторые требуют сертификат с определенным видом открытого ключа. Если не указано, используется имя файла <code>server.pem</code> .
-certform format	Формат используемого сертификата: DER или PEM. По умолчанию PEM.
-key keyfile	Закрытый ключ, который следует использовать. Если эта опция не указана, используется файл сертификата.
-keyform format	Формат используемого закрытого ключа: DER или PEM. По умолчанию PEM.
-pass arg	Источник пароля закрытого ключа. Для получения дополнительной информации о формате аргумента <code>arg</code> см. раздел 1.6.
-dcert filename, -dkey keyname	Указывает дополнительный сертификат и закрытый ключ, которые ведут себя так же, как сертификат и закрытый ключ, указанные в опциях <code>-cert</code> и <code>-key</code> , за исключением того, что если эти опции не указаны, никаких умолчательных дополнительных сертификата и ключа не используется. Как указано выше, некоторые шифр-сьюты требуют сертификат, содержащий ключ определенного типа. Поэтому если сервер уже работает с сертификатом другого типа, ему необходим дополнительный сертификат для установления соединения.

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

Опция	Описание
-dcertform format, -dkeyform format, -dpassarg	Формат соответственно дополнительных сертификата, закрытого ключа и пассфразы.
-nocert	Если указана эта опция, никакой сертификат не используется. Это обуславливает возможность использования только анонимных шифр-сьютов (сейчас только анонимных ДН).
-dhparam filename	Указывает на файл параметров ДН, который следует использовать. Эфемерные ДН-шифр-сьюты создают ключи, использующие набор ДН-параметров. Если не указано противного, делается попытка загрузить параметры из файла сертификата сервера. Если это не удастся, то будет использован статический набор параметров, жестко встроенный в код команды s_server.
-no_dhe	Если установлена эта опция, никаких ДН-параметров загружено не будет, что практически отключает эфемерные ДН-шифр-сьюты.
-verify depth, -Verify depth	Используемая глубина верификации. Опция определяет максимальную длину цепочки сертификатов клиента и заставляет сервер запросить клиентский сертификат. Опция -verify запрашивает сертификат, но клиент не обязан его отправлять, в то время как при указании опции -Verify клиент обязан предоставить сертификат, или произойдет ошибка.
-CApath directory	Каталог, который следует использовать для проверки клиентского сертификата. Этот каталог должен быть в «хэш-формате», см. раздел 15 для получения дополнительной информации. Сертификаты из этого каталога также используются для построения цепочки для проверки серверного сертификата.
-CAfile file	Файл, содержащий доверенные сертификаты, которые следует использовать при аутентификации клиента и во время попыток построить цепочку серверных сертификатов. Этот список также используется в списке приемлемых клиентских сертификатов удостоверяющих центров, который передается клиенту при запросе сертификата.
-state	Выводит состояния SSL-сессии.
-debug	Вывести подробную отладочную информацию, включающую шестнадцатеричный дамп всего трафика.
-msg	Вывести все сообщения протокола с шестнадцатеричным дампом.
-nbio_test	Тестирует неблокирующий ввод-вывод
-nbio	Включает неблокирующий ввод-вывод
-crlf	Эта опция переводит символ конца строки с терминала в CR+LF, как требуют некоторые серверы.
-ign_eof	Подавляет закрытие соединения при обнаружении конца файла на стандартном вводе.
-quiet	Подавляет вывод сессионной информации и информации о сертификате. Как следствие, отключает действие опции — -ign_eof.

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

Опция	Описание
-ssl2, -ssl3, -tls1, -no_ssl2, -no_ssl3, -no_tls1	Эти опции отключают использование определенных SSL- и TLS-протоколов. По умолчанию начальное «рукопожатие» использует метод, который должен быть совместимым со всеми серверами и позволять им использовать протоколы SSL v3, SSL v2 или TLS как следует.
-bugs	В распространенных реализациях SSL и TLS есть несколько известных ошибок. Указание этой опции включает разнообразные методы их обхода.
-hack	Эта опция включает дальнейшие методы обхода ошибок в некоторых ранних реализациях SSL фирмы Netscape.
-cipher cipherlist	Эта опция позволяет модифицировать список допустимых шифр-сьютов, используемый сервером. Когда клиент отправляет список поддерживаемых шифр-сьютов, используется первый шифр-сьют из списка, включенный в соответствующий список сервера. Поскольку клиент указывает шифр-сьюты в порядке предпочтения, порядок шифр-сьютов сервера неважен. Для получения дальнейшей информации см. команду ciphers.
-www	Отправляет статусное сообщение клиенту при установлении соединения. Это сообщение включает большое количество информации об используемых шифр-сьютах и различных параметрах сессии. Выводится в HTML-формате, так что эта опция, как правило, используется с веб-браузерами.
-WWW	Эмулирует простой веб-сервер. Страницы будут загружаться относительно текущего каталога, например если запрашивается URL <code>https://myhost/page.html</code> , будет загружен файл <code>./page.html</code> .
-HTTP	Эмулирует простой веб-сервер. Страницы будут загружаться относительно текущего каталога, например если запрашивается URL <code>https://myhost/page.html</code> , будет загружен файл <code>./page.html</code> . Предполагается, что загруженные файлы содержат полный и корректный HTML-ответ (строки, являющиеся частью строк и заголовков HTTP-ответа, должны заканчиваться CRLF).
-starttls protocol	Отправляет протокольно-специфичное сообщение (сообщения) для переключения в TLS для коммуникации. Аргумент protocol - ключевое слово для соответствующего протокола. В настоящее время поддерживаются только ключевые слова smtp, pop3, imap, and ftp.
-engine id	Указывает загружаемый модуль engine (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.
-id_prefix arg	Создает идентификаторы SSL/TLS сессий, начинающиеся с arg. Это в основном полезно для тестирования любого SSL/TLS-кода (например прокси), который желает иметь дело со множеством серверов, когда каждый из них может генерировать уникальный набор (range) сессионных идентификаторов (например, с определенным префиксом).

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

Опция	Описание
-rand file(s)	Файл или файлы, содержащие случайные данные, используемые для инициации генератора случайных чисел. Несколько файлов можно указать через разделитель, который определяется операционной системой: ; для MS-Windows, , для OpenVMS и : для всех остальных.

#### 14.4 Команды, используемые при установленном соединении

Если установлен запрос на соединение с SSL-клиентом и не использована опция `-www` или `-WWW`, то, как правило, выводятся все данные, полученные от клиента, и все нажатия клавиш будут переданы клиенту.

Также распознаются определенные однобуквенные команды, выполняющие специальные операции. Они перечислены ниже:

- q** Завершить текущее SSL-соединение, но принимать новые соединения.
- Q** Завершить текущее SSL-соединение и закончить работу.
- r** Пересогласовать SSL-сессию.
- R** Пересогласовать SSL-сессию и запросить клиентский сертификат.
- P** Отправить некоторый открытый текст по underlying TCP-соединению: это должно заставить клиента прервать соединение из-за нарушения протокола.
- S** Вывести информацию о статусе кэша сессии.

#### 14.5 Примечания

Команду `s_server` можно использовать для отладки SSL-клиентов. Чтобы принять запросы на соединения от веб-браузеров, можно, например, использовать команду

```
openssl s_server -accept 443 -www
```

Хотя указание пустого списка сертификатов удостоверяющих центров при запросе клиентского сертификата, строго говоря, являются нарушением протокола, большинство SSL-клиентов интерпретируют это как то, что приемлемым является сертификат любого удостоверяющего центра. Это полезно для отладочных целей.

Параметры сессии можно вывести с помощью команды `sess_id`.

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

## 15 КОМАНДА VERIFY

### 15.1 Описание команды

Команда `verify` проверяет цепочки сертификатов.

### 15.2 Формат ввода команды

```
openssl verify [-CApath directory] [-CAfile file] [-purpose purpose] [-untrusted file] [-help] [-issuer_checks] [-verbose] [-] [certificates]
```

### 15.3 Опции команды

Опция	Описание
<code>-CApath directory</code>	Каталог доверенных сертификатов. Сертификаты должны иметь имена формата: <code>hash.0</code> или иметь символические ссылки на них в таком же формате (здесь « <code>hash</code> » — хэшированное значение поля <code>subject name</code> ; см. раздел 16.) Под Unix-подобными ОС скрипт <code>c_rehash</code> автоматически создает символические ссылки на каталог сертификатов.
<code>-CAfile file</code>	Файл доверенных сертификатов. Файл должен содержать больше одного сертификата в PEM-формате, конкатенированных вместе.
<code>-untrusted file</code>	Файл недоверенных сертификатов. Этот файл должен содержать больше одного сертификата
<code>-purpose purpose</code>	Назначение сертификата. Без этой опции никакой цепочечной проверки не выполняется. В настоящее время возможны следующие значения этой опции: <code>sslserver</code> , <code>nssslserver</code> , <code>smimesign</code> , <code>smimeencrypt</code> . Для получения дополнительной информации см. раздел 15.4.
<code>-help</code>	Выводит сообщение об использовании.
<code>-verbose</code>	Выводит дополнительную информацию о выполняемых операциях.
<code>-issuer_checks</code>	Выводит диагностику, связанную с поиском сертификата, на котором заверен обрабатываемый сертификат. Эта диагностика показывает, почему каждый из рассмотренных сертификатов отвергнут. Однако само по себе присутствие сообщений об отказе не означает, что что-то не так — во время обычного процесса проверки может произойти несколько отказов.
<code>-</code>	Отмечает последнюю опцию. Все аргументы, следующие за этой опцией, считаются файлами сертификатов. Это полезно, если имя первого файла сертификатов начинается с <code>"-</code> ".
<code>certificates</code>	Один или больше сертификатов, которые необходимо проверить. Если не указано ни одного имени сертификата, делается попытка считать сертификат со стандартного входа. Все сертификаты должны быть в формате PEM.
<code>-policy имя</code>	Включает политику проверки сертификатов с указанным именем
<code>-purpose имя</code>	Требует чтобы сертификат имел указанную область применения Имена областей применения, определенные в утилите <code>openssl</code> <code>sslclient</code> , <code>sslserver</code> , <code>nssslserver</code> , <code>smimeencrypt</code> , <code>smimesign</code> , <code>crlsign</code> , <code>any</code> .

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

Опция	Описание
-ignore_critical	Игнорировать при проверке сертификата неизвестные расширения X509v3, помеченные как критичные.
-crl_check	Выполнять проверку на наличие сертификата подписи в соответствующем списке отзыва
-crl_check_all	Выполнять проверку на наличие соответствующем списке отзыва всех сертификатов из цепочки доверия
-policy_check	Включает проверку соответствия сертификатов политике указанной в сертификате удостоверяющего центра
-explicit_policy	Требовать явного указания политики
-x509_strict	Строгая проверка соответствия сертификатов формату x509
-policy_print	выводить информацию о политике удостоверяющего центра

## 15.4 Операция проверки

Команда `verify` использует те же функции, что и `internal SSL and S/MIME verification`, таким образом это описание применяется также и к упомянутым операциям.

Существует одно принципиальное отличие операций проверки, которые выполняет команда `verify`, от всех остальных операций проверки: всегда, когда это возможно, делается попытка продолжить работу, в то время как обычно операция проверки прерывается при первой же ошибке. Это позволяет определить все проблемы в цепочке сертификатов.

Операция проверки состоит из ряда отдельных шагов.

Сначала строится цепочка сертификатов от указанного сертификата до корневого сертификата удостоверяющего центра. Если нельзя построить цепочку полностью, это является ошибкой. Цепочка строится с помощью поиска сертификата, на котором подписан рассматриваемый сертификат. Если обнаружен самоподписанный сертификат, он считается корневым сертификатом удостоверяющего центра.

Сам процесс «поиск сертификата, на котором подписан данный сертификат» включает ряд шагов. В версиях OpenSSL до 0.9.5a первый сертификат, поле `subject name` которого соответствовало полю `issuer` рассматриваемого сертификата, считался искомым сертификатом. В OpenSSL версии 0.9.6. и позднее все сертификаты, поле `subject name` которого соответствует полю `issuer` рассматриваемого сертификата, подвергаются дальнейшему исследованию. Идентификационные компоненты ключа рассматриваемого сертификата (если таковые есть) должны совпадать с соответствующими компонентами искомого сертификата, кроме того, расширение `keyUsage` искомого сертификата (если таковое есть) должно позволять подписывать сертификаты.

Искомый сертификат прежде всего ищется в списке недоверенных сертификатов, и если он там не найден, дальнейший поиск ведется в списке доверенных сертификатов. Корневой сертификат УЦ всегда ищется в списке доверенных сертификатов; если сертификат, который надо проверить, является корневым сертификатом, то в списке доверенных сертификатов необходимо найти его точную копию.

Второй шаг — проверка расширений каждого недоверенного сертификата на соответствие указанному назначению. Если опция `-rigrose` не указана, такая проверка не проводится. Указанный или «листовой» (`leaf`) сертификат должен включать расширения, совместимые с указанным назначением, и все остальные сертификаты также должны быть действительными сертификатами удостоверяющих центров. Какие именно расширения требуется, детально описано в разделе 16.6.

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

Третий шаг — проверить установки доверия корневому сертификату удостоверяющего центра. Корневой сертификат должен иметь установленное доверие для данного назначения. Для совместимости с предыдущими версиями SSLеау и OpenSSL сертификат без установок доверия считается доверенным для всех назначений.

Последний шаг — проверка сроков действия сертификатов из цепочки. Срок действия сертификатов проверяется исходя из текущего системного времени и дат notBefore и notAfter в сертификате. В это же время проверяются подписи под сертификатами.

Если все шаги выполняются успешно, сертификат считается корректным. Если какой-то из шагов выполнить не удастся, сертификат некорректен.

## 15.5 Диагностика

Если проверку сертификата выполнить не удастся, генерируются сообщения, которые могут озадачить пользователя. Общий вид сообщения об ошибке:

```
server.pem: /C=AU/ST=Queensland/O=CryptSoft Pty Ltd/CN=Test CA (1024 bit)
error 24 at 1 depth lookup:invalid CA certificate
```

Первая строка содержит название проверяемого сертификата, за которым следует содержание поля subject name этого сертификата. Вторая строка содержит номер и глубину ошибки. Глубина — это номер сертификата в цепочке, вызвавшего ошибку, причем сам проверяемый сертификат имеет глубину 0, сертификат, на котором подписан проверяемый - глубину 1 и так далее. Заканчивается строка выводом текстовой версии номера ошибки.

Ниже приведен длинный список кодов ошибок и соответствующих сообщений. Список также включает наименование кода, определенное в заголовочном файле x509\_vfy.h. Некоторые из кодов ошибки определены, но никогда не возникают; они описаны как «неиспользуемые».

Номер ошибки	Код ошибки	Расшифровка	Пояснения
0	X509_V_OK	ok	Операция выполнена успешно
2	X509_V_ERR_UNABLE_TO_GET_ISSUER_CERT	unable to get issuer certificate	Не удастся найти сертификат, на котором подписан данный: это происходит, если отсутствует сертификат, на котором подписан данный.
3	X509_V_ERR_UNABLE_TO_GET_CRL	unable to get certificate CRL	Не обнаружен список отзыва сертификатов. Не используется.
5	X509_V_ERR_UNABLE_TO_DECRYPT_CRL_SIGNATURE	unable to decrypt CRL's signature	Подпись под списком отзыва сертификатов не удастся расшифровать. Не используется.
6	X509_V_ERR_UNABLE_TO_DECODE_ISSUER_PUBLIC_KEY	unable to decode issuer public key	Открытый ключ в SubjectPublicKeyInfo сертификата не найден.

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

Номер ошибки	Код ошибки	Расшифровка	Пояснения
7	X509_V_ERR_CERT_SIGNATURE_FAILURE	certificate signature failure	Подпись под сертификатом некорректна
8	X509_V_ERR_CRL_SIGNATURE_FAILURE	CRL signature failure	Подпись под списком отзыва сертификатов некорректна. Не используется
9	X509_V_ERR_CERT_NOT_YET_VALID	certificate is not yet valid	Сертификат еще не вступил в действие: дата notBefore еще не наступила.
10	X509_V_ERR_CERT_HAS_EXPIRED	certificate has expired	Срок действия сертификата закончился: дата notAfter уже прошла.
11	X509_V_ERR_CRL_NOT_YET_VALID	CRL is not yet valid	Срок действия списка отзвов сертификатов еще не начался. Не используется
12	X509_V_ERR_CRL_HAS_EXPIRED	CRL has expired	Срок действия списка отзыва сертификатов закончился. Не используется
13	X509_V_ERR_ERROR_IN_CERT_NOT_BEFORE_FIELD	format error in certificate's notBefore field	Поле сертификата notBefore содержит некорректное время
14	X509_V_ERR_ERROR_IN_CERT_NOT_AFTER_FIELD	format error in certificate's notAfter field	Поле сертификата notAfter содержит некорректное время
15	X509_V_ERR_ERROR_IN_CRL_LAST_UPDATE_FIELD	format error in CRL's lastUpdate field	Поле списка отзыва сертификатов lastUpdate содержит некорректное время. Не используется
16	X509_V_ERR_ERROR_IN_CRL_NEXT_UPDATE_FIELD	format error in CRL's nextUpdate field	Поле списка отзыва сертификатов nextUpdate содержит некорректное время. Не используется
17	X509_V_ERR_OUT_OF_MEM	out of memory	Произошла ошибка при распределении памяти. Этого никогда не должно происходить.
18	X509_V_ERR_DEPTH_ZERO_SELF_SIGNED_CERT	self signed certificate	Проверяемый сертификат является самоподписанным и не обнаружен в списке доверенных сертификатов.
19	X509_V_ERR_SELF_SIGNED_CERT_IN_CHAIN	self signed certificate in certificate chain	Может быть составлена цепочка из недоверенных сертификатов, но корневой сертификат на данном компьютере не обнаружен.

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

Номер ошибки	Код ошибки	Расшифровка	Пояснения
20	X509_V_ERR_UNABLE_TO_GET_ISSUER_CERT_LOCALLY	unable to get local issuer certificate	Сертификат, на котором подписан сертификат, найденный на данном компьютере, не обнаружен. Это обычно означает, что список доверенных сертификатов не полон.
21	X509_V_ERR_UNABLE_TO_VERIFY_LEAF_SIGNATURE	unable to verify the first certificate	Ни одна подпись не может быть проверена, потому что цепочка содержит только один сертификат, не являющийся самоподписанным.
22	X509_V_ERR_CERT_CHAIN_TOO_LONG	certificate chain too long	Длина цепочки сертификатов превосходит указанную максимальную глубину. Неиспользуемая.
23	X509_V_ERR_CERT_REVOKED	certificate revoked	Сертификат был отозван. Не используется
24	X509_V_ERR_INVALID_CA	invalid CA certificate	Сертификат удостоверяющего центра недействителен. Либо это не сертификат удостоверяющего центра, либо его расширения не соответствуют указанному назначению.
25	X509_V_ERR_PATH_LENGTH_EXCEEDED	path length constraint exceeded	Величина параметра the basicConstraints pathlength превышена.
26	X509_V_ERR_INVALID_PURPOSE	unsupported certificate purpose	Указанный сертификат не может быть использован по указанному назначению.
27	X509_V_ERR_CERT_UNTRUSTED	certificate not trusted	Корневой сертификат удостоверяющего центра не отмечен как доверенный для указанного назначения.
28	X509_V_ERR_CERT_REJECTED	certificate rejected	Корневой сертификат отмечен как отвергнутый для указанного назначения.

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

Номер ошибки	Код ошибки	Расшифровка	Пояснения
29	X509_V_ERR_SUBJECT_ISSUER_MISMATCH	subject issuer mismatch	Сертификат, на котором предположительно был подписан проверяемый сертификат, был отвергнут, потому что значение поля subject name не соответствовало значению поля issuer name проверяемого сертификата. Выводится только в том случае, если указана опция -issuer_checks.
30	X509_V_ERR_AKID_SKID_MISMATCH	authority and subject key identifier mismatch	Сертификат, на котором предположительно был подписан проверяемый сертификат, был отвергнут, потому что идентификатор subject key присутствует и не соответствует authority key identifier проверяемого сертификата. Выводится только в том случае, если указана опция -issuer_checks.
31	X509_V_ERR_AKID_ISSUER_SERIAL_MISMATCH	authority and issuer serial number mismatch	Сертификат, на котором предположительно был подписан проверяемый сертификат, был отвергнут, потому что значения полей issuer name и serial number присутствуют и не соответствуют полю authority key identifier of the current certificate. Выводится только в том случае, если указана опция -issuer_checks.
32	X509_V_ERR_KEYUSAGE_NO_CERTSIGN	key usage does not include certificate signing	Сертификат, на котором предположительно был подписан проверяемый сертификат, был отвергнут, потому что его расширение keyUsage не позволяет подписывать сертификаты.

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

<b>Номер ошиб- ки</b>	<b>Код ошибки</b>	<b>Расшифровка</b>	<b>Пояснения</b>
50	X509_V_ERR_APPLICATION_ VERIFICATION	application verification failure	Ошибка, специфичная для приложения. Не использует- ся.

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

## 16 КОМАНДА X509

### 16.1 Описание команды

Команда `x509` — многоцелевая утилита для работы с сертификатами. Ее можно использовать для вывода информации о сертификате, преобразования сертификатов в различные формы, подписывания заявок на сертификаты в качестве «мини-УЦ» и редактирования настроек доверия сертификата.

### 16.2 Формат ввода команды

```
openssl x509 [-inform DER|PEM|NET] [-outform DER|PEM|NET] [-keyform DER|PEM] [-CAform DER|PEM] [-CAkeyform DER|PEM] [-in filename] [-out filename] [-serial] [-hash] [-subject_hash] [-issuer_hash] [-subject] [-issuer] [-nameopt option] [-email] [-startdate] [-enddate] [-purpose] [-dates] [-modulus] [-fingerprint] [-alias] [-noout] [-trustout] [-clrtrust] [-clrreject] [-addtrust arg] [-addreject arg] [-setalias arg] [-days arg] [-set_serial n] [-signkey filename] [-x509toreq] [-req] [-CA filename] [-CAkey filename] [-CAcreateserial] [-CAserial filename] [-text] [-C] [-md2|-md5|-sha1|-mdc2|-md_gost94] [-clrext] [-extfile filename] [-extensions section] [-engine id]
```

### 16.3 Описание опций

Поскольку у этой команды много опций, их описание разбито на несколько разделов.

#### 16.3.1 Опции ввода, вывода и общего назначения

Опция	Описание
<code>-inform DER PEM NET</code>	Определяет входной формат. Как правило, команда ожидает сертификат формата X509, но это может измениться, если присутствуют другие опции, например <code>-req</code> . Формат <code>DER</code> — DER-кодировка сертификата, а формат <code>PEM</code> — DER-форма в кодировке <code>base64</code> с добавленными верхним и нижним ограничителями. Опция <code>NET</code> — непрозрачный формат сервера Netscape, сейчас не рекомендованный к применению
<code>-outform DER PEM NET</code>	Определяет выходной формат. Опция имеет те же значения, что и опция <code>-inform</code> .
<code>-in filename</code>	Определяет имя входного файла, из которого следует считать сертификат. Если эта опция не указана, сертификат считывается со стандартного входа.
<code>-out filename</code>	Определяет имя выходного файла. Если эта опция не указана, выходные данные выводятся на стандартный вывод.
<code>-md5</code>	Использовать алгоритм хэширования MD5 при вычислении отпечатка ( <code>fingerprint</code> ) сертификата или при подписании сертификата с помощью опции <code>-signkey</code> (только при использовании алгоритмов подписи, для которых не специфицировано использование определенного алгоритма хэширования)

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

Опция	Описание
-sha1	Использовать алгоритм хэширования SHA1 при вычислении отпечатка (fingerprint) сертификата или при подписании сертификата с помощью опции -signkey (только при использовании алгоритмов подписи, для которых не специфицировано использование определенного алгоритма хэширования)
-md_gost94	Использовать алгоритм хэширования ГОСТ Р 34.11-94 при вычислении отпечатка (fingerprint) сертификата.
-engine id	Указывает на загружаемый модуль engine (по его уникальной идентификационной строке) и вызывает попытку использовать реализацию криптографических алгоритмов из этого модуля.

### 16.3.2 Опции просмотра сертификатов

**Примечание:** опции -alias и -purpose также являются опциями просмотра, но описываются в разделе 16.3.3.

Опция	Описание
-text	Выводит сертификат для просмотра в текстовом виде. Полный вывод содержит открытый ключ, алгоритмы подписи, содержание полей subject name и issuer name, серийный номер, все присутствующие расширения и все настройки доверия.
-certopt option	Регулирует формат вывода при использовании опции -text. Аргумент option может быть одной опцией или множеством опций, разделенных запятыми. Для установки значения различных опций свитч -certopt также может использоваться несколько раз.
-noout	Эта опция предотвращает вывод закодированной версии заявки.
-modulus	Эта опция выводит значение модуля открытого ключа, содержащегося в сертификате.
-serial	Выводит серийный номер сертификата
-subject_hash	Выводит «хэш» значения поля subject name сертификата. Используется в OpenSSL для создания указателя, позволяющего искать сертификаты в каталоге по значению поля subject name.
-issuer_hash	Выводит «хэш» значения поля issuer name сертификата.
-hash	синоним опции -hash для обеспечения обратной совместимости.
-subject	Выводит значение поля subject name.
-issuer	Выводит значение поля issuer name.
-nameopt option	Опция указывает, как должны выводиться значения полей subject и issuer. Аргументом может быть одна опция или несколько, разделенных запятыми. Для установки нескольких опций можно также несколько раз использовать свитч -nameopt. Для получения дополнительной информации см. раздел 16.3.5.
-email	Выводит адрес(а) электронной почты, если таковые указаны.
-startdate	Выводит дату начала срока действия сертификата notBefore.
-enddate	Выводит дату окончания срока действия сертификата notAfter.
-dates	Выводит даты начала и окончания срока действия сертификата.
-fingerprint	Выводит хэш-сумму DER-закодированной версии всего сертификата (см. раздел 16.5).

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

Опция	Описание
-C	Эта опция выводит сертификат в виде файла на языке C.

### 16.3.3 Настройки доверия

Пожалуйста, учтите, что эти опции в настоящее время являются экспериментальными и могут измениться.

Доверенный сертификат — это обычный сертификат, к которому добавлена некоторая дополнительная информация, такая как разрешенные или запрещенные назначения сертификата и его «алиас».

Обычно при подтверждении сертификата хотя бы один сертификат в цепочке должен быть «доверенным». По умолчанию доверенный сертификат должен храниться на локальном компьютере и представлять собой корневой сертификат удостоверяющего центра. Любая цепочка сертификатов, заканчивающаяся этим сертификатом, при таких условиях годится для любых целей.

В настоящее время настройки доверия используются только для корневых сертификатов удостоверяющих центров. Они предоставляют более гибкий контроль над назначениями корневого сертификата удостоверяющего центра. Например, такой сертификат может быть доверенным для использования SSL-клиентом, но не SSL-сервером.

См. раздел 15 для получения дополнительной информации о значении настроек доверия.

Будущие версии OpenSSL будут распознавать настройки доверия любых сертификатов, не только сертификатов удостоверяющего центра.

Опция	Описание
-trustout	Эта опция заставляет утилиту x509 создать доверенный сертификат в качестве выходных данных. В качестве входных данных может быть как обычный сертификат, так и доверенный, но по умолчанию выходными данными является обычный сертификат, и все настройки доверия отбрасываются. С использованием этой опции создается доверенный сертификат. Доверенный сертификат создается автоматически, если модифицируются какие-либо настройки доверия.
-setalias arg	Устанавливает алиас сертификата. Это позволяет ссылаться на сертификат по алиасу, например «сертификат Иванова».
-alias	Выводит алиас сертификата, если таковой существует.
-clrtrust	Очищает все дозволенные или доверенные назначения сертификата.
-clrreject	Очищает все запрещенные или отвергнутые назначения сертификата.
-addtrust arg	Добавляет доверенное назначение сертификата. Здесь может быть использовано любое наименование объекта, но в настоящее время используется только clientAuth (использование с SSL-клиентом), serverAuth (использование с SSL-сервером) и emailProtection (электронная почта формата S/MIME). Другие OpenSSL-приложения могут определять дополнительные назначения.
-addreject arg	Добавляет запрещенное назначение сертификата. Опция принимает те же значения, что и опция -addtrust.

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

Опция	Описание
-purpose	Эта опция выполняет тестирование расширений сертификатов и выводит результаты. Для получения дополнительной информации см. раздел 16.6.

#### 16.3.4 Опции подписания сертификатов

Команда x509 может использоваться для подписания сертификатов и заявок; таким образом, она может вести себя как «мини-УЦ».

Опция	Описание
-signkey filename	Эта опция создает самоподписанный файл при помощи указанного закрытого ключа. Если входной файл является сертификатом, опция устанавливает значение поля issuer name равным значению поля subject name (т.е. делает его самоподписанным), заменяет открытый ключ на указанный и меняет даты начала и конца срока действия. Дата начала действия устанавливается в текущее время, а дата конца действия устанавливается во время, определенное опцией -days. Все расширения сертификата сохраняются, если не указана опция -clrext. Если входной файл является заявкой на сертификат, то создается самоподписанный сертификат, использующий указанный закрытый ключ и значение поля subject name из заявки.
-clrext	Удаляет все расширения из сертификата. Эта опция используется, когда сертификат создается из другого сертификата (например с помощью опции -signkey или -CA). Если эта опция не указана, все расширения сохраняются.
-keyform PEM DER	Указывает формат (DER или PEM) файла закрытого ключа, используемого в опции -signkey.
-days arg	Указывает срок действия сертификата. По умолчанию 30 дней.
-x509toreq	превращает сертификат в заявку. Опция -signkey используется для передачи требуемого закрытого ключа.
-req	По умолчанию в качестве входных данных ожидается сертификат. Если указана данная опция, вместо сертификата ожидается заявка на сертификат.
-set_serial n	Указывает используемый серийный номер. Эта опция может использоваться с опциями -signkey или -CA. Если используется вместе с опцией -CA, то файл серийного номера (определенный опциями -CAserial или -CAcreateserial) не используется. Серийный номер может быть десятичным или шестнадцатиричным (с префиксом 0x). Указывать отрицательные серийные номера можно, но не рекомендуется.

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

Опция	Описание
-CA filename	Указывает сертификат удостоверяющего центра, который следует использовать для подписывания. Когда указывается эта опция, команда x509 работает как «мини-УЦ». При указании этой опции входной файл подписывается на указанном сертификате удостоверяющего центра, то есть поле issuer name входного файла устанавливается в значение поля subject name указанного сертификата УЦ, формируется цифровая подпись под входным файлом с использованием закрытого ключа, соответствующего указанному сертификату УЦ. Эта опция, как правило, указывается вместе с опцией -req. Без указания опции -req входным файлом является сертификат, который необходимо сделать самоподписанным.
-CAkey filename	Указывает закрытый ключ, соответствующий сертификату УЦ, на котором следует подписывать входной сертификат. Если эта опция не указана, считается, что закрытый ключ включен в файл сертификата УЦ.
-CAserial filename	Устанавливает, какой файл серийного номера УЦ следует использовать. Если указывается опция -CA при подписывании сертификата, она использует серийный номер, указанный в файле. Этот файл состоит из одной строки, содержащей четное количество шестнадцатиричных цифр. После каждого применения этой опции серийный номер увеличивается на единицу и снова записывается в этот файл. Умолчательное значение имени файла состоит из имени файла сертификата УЦ (взятого без расширения) с расширением .srl. Например, если файл сертификата УЦ называется mysacert.pem, ожидается существующий файл серийного номера mysacert.srl.
-CAcreateserial	При указании этой опции, если файл серийного номера УЦ не существует, он создается. Файл будет содержать серийный номер 02, а подписанный сертификат получит серийный номер 1. Как правило, если указана опция -CA и файла серийного номера не существует, это является ошибкой.
-extfile filename	Файл, содержащий расширения сертификатов, которые следует использовать. Если не указан, к сертификату не добавляется никаких расширений.
-extensions section	раздел конфигурационного файла, из которого следует добавлять расширения в сертификаты. Если эта опция не указана, расширения должны либо содержаться в непоименованном (умолчательном) разделе, либо умолчательный раздел должен содержать переменную "extensions", содержащую наименование соответствующего раздела.

### 16.3.5 Опции именования

Командно-строчный свитч nameopt определяет, как выводятся поля subject name и issuer name. Если ни одного свитча nameopt не указано, используется умолчательный «однострочный» формат, совместимый с предыдущими версиями OpenSSL. Каждая опция подробно описана внизу, перед каждой опцией может стоять «-» для ее отключения. Как правило, исполь-

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

зуются только первые четыре.

Опция	Описание
compat	Использовать старый формат. Это эквивалентно отсутствию указания опций именованя.
RFC2253	Выводит имена, совместимые с определенным в RFC2253 эквивалентом опций esc_2253, esc_ctrl, esc_msb, utf8, dump_nostr, dump_unknown, dump_der, sep_comma_plus, dn_rev и sname.
oneline	Однорочный формат, более читабельный, чем RFC2253. Эквивалентно определению опций esc_2253, esc_ctrl, esc_msb, utf8, dump_nostr, dump_der, use_quote, sep_comma_plus_space, space_eq и sname.
multiline	Многочрочный формат. Эквивалентен определению опций esc_ctrl, esc_msb, sep_multiline, space_eq, lname и align.
esc_2253	Экранировать «специальные» символы, требуемые RFC2253 в поле, то есть ,+<>;. Кроме того, # экранируется в начале строки, а пробел — в начале или в конце строки.
esc_ctrl	Экранировать контрольные символы, т.е. символы с ASCII-значениями, меньшими 0x20 (пробел), и символ удаления (0x7f). Они экранируются с использованием определенной в RFC2253 нотации \XX notation (где XX — две шестнадцатеричных цифры, представляющие значение символа).
esc_msb	Экранировать символы с ненулевым старшим (most significant) битом, то есть символы, ASCII-значения которых превосходят 127. Эту опцию не рекомендуется использовать при работе с сертификатами, содержащими кириллицу
use_quote	Экранирует некоторые символы, окружая всю строку символов символами без указания этой опции все экранирование выполняется с помощью символа \.
utf8	Сначала перевести все строки в UTF-формат. Это требуется в RFC2253. Если вам повезло и у вас есть UTF-8-совместимый терминал, то использование этой опции (без установки esc_msb) может привести к корректному выводу многобайтных (международных) символов. Если эта опция не присутствует, многобайтные символы, превосходящие 0xff, будут представлены в формате \UXXXX для шестнадцатитбитных и \WXXXXXXXXX для тридцатидвухбитных. Кроме того, если эта опция отключена, любые UTF-8-строки сначала будут переведены в свою символьную форму.
no_type	Эта опция вообще не пытается интерпретировать многобайтные символы. Их содержательные октеты просто дампируются так, как если бы один октет содержал один символ. Это полезно для диагностических целей, но приведет к весьма странно выглядящим выходным данным.
show_type	Показывает тип символьной строки ASN.1. Тип указывается до содержания поля. Например "BMPSTRING: Hello World".

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

Опция	Описание
dump_der	Когда эта опция установлена, любые поля, которые нуждаются в шестнадцатичном дампировании, будут дампированы с использованием DER-кодировки. Если опция не установлена, будут выведены только октеты содержания. Обе опции используют описанный в RFC2253 формат #XXXX...
dump_nostr	Дампировать типы несимвольных строк (например OCTET STRING). Если эта опция не указана, типы несимвольных строк будут выводиться так, как будто каждый октет содержания представляет один символ.
dump_all	Дампировать все поля. Эта опция при использовании с dump_der позволяет однозначно определить DER-кодирование структуры.
dump_unknown	Дампировать все поля, чьи OID не распознаются OpenSSL.
sep_comma_plus, sep_comma_plus_space, sep_semi_plus_space, sep_multiline	Эти опции определяют разделители полей. Первый символ - разделитель для RDN и второй — для многократных AVA (многократные AVA встречаются очень редко, и их использование не рекомендуется). Опции, заканчивающиеся на слово space, дополнительно помещают пробел после разделителя для лучшей читабельности. Опция sep_multiline использует символ LF для разделителя RDN и окруженный пробелами + в качестве разделителя AVA. Кроме того, она обуславливает помещение в начале каждого поля отступа в четыре символа.
dn_rev	Изменить порядок полей в DN на противоположный. Это требование RFC2253. В качестве дополнительного эффекта эта опция также обращает порядок многократных AVA, но это позволительно.
nofname, sname, lname, oid	Эти опции влияют на формат вывода названия поля. Опция nofname вообще не выводит соответствующее поле. Опция sname использует форму «короткого имени» (например CN вместо commonName). Опция lname использует полное наименование. Опция oid представляет OID в численной форме и полезна для диагностических целей.
align	Выравнивает значения полей для более читабельного вывода. Может использоваться только вместе с опцией sep_multiline.
space_eq	Окружает пробелами символ «=», следующий за именем поля.

### 16.3.6 Опции текста

Можно регулировать не только формат вывода имен, но и набор выводимых полей, используя опции certopt, если присутствует опция text. По умолчанию выводятся все поля.

Опция	Описание
compatible	использовать старый формат. Это эквивалентно отсутствию указания опций вывода.
no_header	Не выводить информацию о заголовках, т.е. о строках Certificate и Data.
no_version	Не выводить номер версии
no_serial	Не выводить серийный номер
no_signame	Не выводить используемый алгоритм подписи

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

Опция	Описание
no_validity	Не выводить срок действия, т.е. поля notBefore и notAfter.
no_subject	Не выводить поле subject name.
no_issuer	Не выводить поле issuer name.
no_pubkey	Не выводить открытый ключ.
no_sigdump	Не выводить шестнадцатеричный дампы подписи под сертификатом.
no_aux	Не выводить настройки доверия сертификата.
no_extensions	Не выводить расширения версии 3 формата X509.
ext_default	Сохранить умолчательное поведение расширений; попытаться вывести неподдерживаемые расширения сертификатов.
ext_error	Вывести сообщение об ошибке для неподдерживаемых расширений сертификатов.
ext_parse	Вывести неподдерживаемые расширения в виде ASN.1-структуры.
ext_dump	Вывести шестнадцатеричный дампы неподдерживаемых расширений.
ca_default	Значение, используемое командой ca, эквивалентно набору опций no_issuer, no_pubkey, no_header, no_version, no_sigdump и no_signame.

## 16.4 Примеры

Вывести содержание сертификата на экран:

```
openssl x509 -in cert.pem -noout -text
```

Вывести серийный номер сертификата:

```
openssl x509 -in cert.pem -noout -serial
```

Вывести поле subject name сертификата:

```
openssl x509 -in cert.pem -noout -subject
```

Вывести поле subject name сертификата в RFC2253-формате:

```
openssl x509 -in cert.pem -noout -subject -nameopt RFC2253
```

Вывести поле subject name сертификата в однострочном формате на терминале, поддерживающем UTF-8:

```
openssl x509 -in cert.pem -noout -subject -nameopt oneline,-esc_msb
```

Вывести MD5-отпечаток сертификата:

```
openssl x509 -in cert.pem -noout -fingerprint
```

Вывести SHA1-отпечаток сертификата:

```
openssl x509 -sha1 -in cert.pem -noout -fingerprint
```

Перевести сертификат из PEM-формата в DER-формат:

```
openssl x509 -in cert.pem -inform PEM -out cert.der -outform DER
```

Получить из сертификата заявку:

```
openssl x509 -x509toreq -in cert.pem -out req.pem -signkey key.pem
```

Превратить заявку на сертификат в самоподписанный сертификат, используя расширения для сертификата УЦ:

```
openssl x509 -req -in careq.pem -extfile openssl.cnf -extensions v3_ca -signkey key.pem -out cacert.pem
```

Подписать заявку на сертификат, используя сертификат УЦ из предыдущего примера, и добавить расширения пользовательского сертификата:

```
openssl x509 -req -in req.pem -extfile openssl.cnf -extensions v3_usr -CA cacert.pem -CAkey key.pem -CAcreateserial
```

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

Установить для сертификата доверие для использования с SSL-клиентом и установить для него алиас Steve's Class 1 CA:

```
openssl x509 -in cert.pem -addtrust clientAuth -setalias "Steve's
Class 1 CA" out trust.pem
```

## 16.5 Примечания

PEM-формат использует следующий вид ограничителей:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

Кроме того, обрабатываются файлы, содержащие следующий вид ограничителей:

```
-----BEGIN X509 CERTIFICATE-----
-----END X509 CERTIFICATE-----
```

Вид ограничителей для доверенных сертификатов:

```
-----BEGIN TRUSTED CERTIFICATE-----
-----END TRUSTED CERTIFICATE-----
```

Перевод в формат UTF-8, используемый с опциями именованя, предполагает, что в переменных типа T61Strings используется кодировка ISO8859-1. Это не так, но так работают Netscape и Microsoft IE, а также множество сертификатов. Поэтому хотя это предположение и некорректно, его использование позволяет корректно вывести большинство сертификатов.

Опция `-fingerprint` выводит хэш-сумму DER-закодированного сертификата. Эта хэш-сумма обычно называется «отпечатком пальца». Вследствие природы хэш-сумм «отпечаток пальца» уникален для каждого сертификата, и два сертификата с одним и тем же «отпечатком» могут считаться одним и тем же сертификатом.

Netscape использует алгоритм хэширования MD5, а Microsoft IE использует SHA1.

Опция `-email` просматривает поле `subject name` и расширение `subject alternative name`. Выводятся только уникальные почтовые адреса: опция не выводит один и тот же почтовый адрес дважды.

## 16.6 Расширения сертификатов

Опция `-purpose` проверяет расширения сертификатов и определяет, для чего сертификат может быть использован. Выполняемые проверки довольно сложны и включают различные способы обращения с некорректными сертификатами и программами.

Тот же код используется при проверке недоверенных сертификатов в цепочках, поэтому этот раздел полезен в том случае, если цепочка признана некорректной при использовании команды `-verify`.

Флаг расширения `basicConstraints` сертификата удостоверяющего центра используется для определения того, может ли сертификат использоваться как сертификат удостоверяющего центра. Если установлен флаг сертификата удостоверяющего центра `true`, то это сертификат УЦ, если этот флаг `false`, то это не сертификат УЦ. Все сертификаты УЦ должны иметь значение этого флага `true`.

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

Если сертификат является сертификатом версии V1 (то есть не имеет расширений) и он самоподписан, он считается сертификатом УЦ, но выводится предупреждение об этом. Это способ обхода проблемы с корневыми сертификатами Verisign, которые являются самоподписанными сертификатами версии V1.

Если присутствует расширение keyUsage, на возможные назначения сертификата накладываются дополнительные ограничения. Сертификат УЦ должен иметь установленные бит keyCertSign, если в нем присутствует данное расширение.

Расширение extended key usage накладывает дальнейшие ограничения на возможные назначения сертификатов. Если это расширение присутствует (критичное или нет), ключ может использоваться только для указанных назначений.

Ниже приводится полное описание каждой проверки. Приведенные выше комментарии о basicConstraints, keyUsage и сертификатов версии 1 применимы ко всем сертификатам удостоверяющих центров.

Назначение	Требования
SSL Client	Расширение extended key usage должно отсутствовать или включать OID web client authentication. Расширение keyUsage должно отсутствовать или иметь установленный бит digitalSignature. Тип сертификата Netscape должен отсутствовать или иметь установленный бит SSL client.
SSL Client CA	Расширение extended key usage должно отсутствовать или включать OID web client authentication. Тип сертификата Netscape должен отсутствовать или иметь установленный бит SSL CA, это используется для обхода ситуации, если отсутствует расширение basicConstraints.
SSL Server	Расширение extended key usage должно отсутствовать или включать OID web server authentication и/или один из SGC OID. Расширение keyUsage должно отсутствовать или иметь установленный бит digitalSignature или keyEncipherment (или оба). Тип сертификата Netscape должен отсутствовать или иметь установленный бит SSL server.
SSL Server CA	Расширение extended key usage должно отсутствовать или включать OID web server authentication и/или один из SGC OID. Тип сертификата Netscape должен отсутствовать или иметь установленный бит SSL CA, это используется для обхода ситуации, если отсутствует расширение basicConstraints.
Netscape SSL Server	Чтобы Netscape SSL-клиент мог соединиться с SSL-сервером, сертификат должен иметь установленный бит keyEncipherment если присутствует расширение keyUsage. Это не всегда корректно, потому что некоторые шифр-сыюты используют этот ключ для создания цифровой подписи. Все остальное так же, как для обычного SSL-сервера.
Common S/MIME Client Tests	Расширение extended key usage должно отсутствовать или включать OID email protection. Тип сертификата Netscape должен отсутствовать или иметь установленный бит S/MIME. Если бит S/MIME не установлен в типе сертификата Netscape, в качестве альтернативы допустим установленный бит SSL client, но выводится предупреждение; это связано с тем, что некоторые сертификаты Verisign не устанавливают бит S/MIME.

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

Назначение	Требования
S/MIME Signing	В дополнение к обычным проверкам для S/MIME-клиента должен быть установлен бит digitalSignature, если присутствует расширение keyUsage.
S/MIME Encryption	В дополнение к обычным проверкам для S/MIME-клиента должен быть установлен бит keyEncipherment, если присутствует расширение keyUsage.
S/MIME CA	Расширение extended key usage должно отсутствовать или включать OID email protection. Тип сертификата Netscape должен отсутствовать или иметь установленный бит S/MIME CA, это используется для обхода ситуации, если отсутствует расширение basicConstraints.
CRL Signing	Расширение keyUsage должно отсутствовать или иметь установленный бит CRL signing.
CRL Signing CA	Проводятся обычные проверки сертификата удостоверяющего центра. Но в этом случае должно присутствовать расширение basicConstraints.

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

