КРИПТОКОМ

УТВЕРЖДЕН
СЕИУ.00009-01 32 - ЛУ

СРЕДСТВО КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ
МагПро КриптоПакет вер. 1.0

Руководство системного администратора

СЕИУ.00009-01 32
Листов 26

Литера О

2006

Аннотация

Настоящий документ содержит руководство системного администратора для работы с СКЗИ «МагПро КриптоПакет».

Содержание

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

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

1 НАЗНАЧЕНИЕ СКЗИ «МАГПРО КРИПТОПАКЕТ»

Основное назначение программного комплекса «МагПро КриптоПакет» — возможность использования российских криптоалгоритмов при работе с приложениями, рассчитанными на использование библиотеки OpenSSL.

СКЗИ «МагПро КриптоПакет» совместим на уровне исходных текстов с библиотекой OpenSSL версии 0.9.8. Бинарная совместимость с оригинальной OpenSSL не обеспечивается. Для работы с СКЗИ «МагПро КриптоПакет» приложения должны быть перекомпилированы.

СКЗИ «МагПро КриптоПакет» реализует следующие криптоалгоритмы:

- Алгоритм хэширования по ГОСТ Р 34.11-94
- Алгоритм выработки подписи по ГОСТ Р 34.10-94
- Алгоритм выработки подписи по ГОСТ Р 34.10-2001
- Алгоритм имитозащиты по ГОСТ 28147-89
- Алгоритм шифрования по ГОСТ 28147-89

Алгоритм ГОСТ Р 34.10-94 должен использоваться только для проверки ранее выработанных подписей.

СКЗИ «МагПро КриптоПакет» позволяет использование российских алгоритмов для следующих целей:

- Создание защищенных TCP-соединений с использованием протокола TLS;
- Обработка защищенных сообщений электронной почты в форматах S/MIME и PKCS#7;
- Работа с сертификатами ключей в формате X509 и заявками на сертификаты в формате PKCS#10;
- Проверка статуса сертификатов с использованием протокола OCSP.

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

2 УСЛОВИЯ РАБОТЫ СКЗИ «МАГПРО КРИПТОПАКЕТ»

2.1 Требования к конфигурации ПЭВМ

2.1.1 Аппаратные требования

Для работы СКЗИ «МагПро КриптоПакет» необходимы следующие условия:
для архитектуры x86 — процессор Pentium и выше
для архитектуры Sparc — процессор UltraSparc

2.1.2 Программные требования

СКЗИ «МагПро КриптоПакет» поставляется в собранном виде. Предназначен для работы в операционных системах:

Windows 2000/XP, Windows Server 2003, Windows Vista
Debian GNU/Linux 3.1, 4.0
AltLinux Master 2.4
Red Hat Enterprise Linux 4, 5
Fedora Core linux 5
SUSE Linux 9, 10
ASP Linux 9.2, 10, 11
Mandriva Linux 2006
Mandrake Linux 10.1
FreeBSD 4.x 5.x 6.x
Solaris Sparc 8, 9, 10
Solaris Intel 8, 9, 10

На платформах Linux, FreeBSD и Open Solaris 5.11 и выше для работы СКЗИ «МагПро КриптоПакет» требуется библиотека Tcl версии 8.4 и выше. Для платформы Win32 и Solaris 8–10 необходимые библиотеки Tcl включены в дистрибутив СКЗИ «МагПро КриптоПакет».

Для сборки приложений, использующих СКЗИ «МагПро КриптоПакет», требуется компилятор языка C. На всех платформах поддерживается компилятор gcc. Кроме того, допускается использование на отдельных платформах других компиляторов (например Microsoft Visual Studio на платформе Win32).

СКЗИ «МагПро КриптоПакет» для платформы Win32 предназначен для использования в native Win32 приложениях. Использование его в программах для среды cygwin не допускается, хотя возможно использование компилятора из среды cygwin в режиме сборки win32-приложений (`-mnocygwin`). Рекомендуемым компилятором для платформы Win32 является Mingw32.

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

3 УСТАНОВКА СКЗИ «МАГПРО КРИПТОПАКЕТ»

В зависимости от операционной системы СКЗИ «МагПро КриптоПакет» может поставляться как пакет (набор пакетов) для пакетного менеджера данной операционной системы, как архив в формате tar.gz или в виде установочного файла.

3.1 Установка под Win32

Запустить установочный файл. Во время установки будет предложено выбрать устанавливаемые модули из списка. В состав продукта входят следующие модули:

`OpenSSL runtime` — Динамические библиотеки модифицированной OpenSSL. Требуются на всех компьютерах, где используется продукт.

`Common Development Files` — Заголовочные файлы языка C, предназначенные для сборки приложений с использованием МагПро КриптоПакет.

`Mingw32 development libraries` — Библиотеки для сборки приложений с помощью компилятора Mingw32.

`MSVC development libraries` — Библиотеки для сборки приложений с помощью компилятора Microsoft Visual Studio.

`CA Certificates` — Сертификаты удостоверяющих центров, входящие в дистрибутив OpenSSL.

`Cryptocom GOST Support Module` — Динамическая библиотека, реализующая алгоритмы ГОСТ.

`Key management utilities` — Утилиты для создания и безопасного удаления закрытых ключей.

`Yarrow RNG` — Датчик случайных чисел YARROW.

3.2 Установка под Debian Linux

Установить дистрибутивный носитель в устройство чтения и выполнить команду `apt-cdrom add` с правами суперпользователя. Выполнить команду `aptitude update`.

Далее с помощью команды `aptitude install [имя пакета]` можно установить следующие пакеты:

`libssl-r0.9.8` — Содержит разделяемые библиотеки OpenSSL. Требуется на всех компьютерах, где будет использоваться СКЗИ «МагПро КриптоПакет». Если программное обеспечение, пересобранное для работы с СКЗИ «МагПро КриптоПакет», устанавливается с использованием пакетного менеджера, этот пакет будет установлен автоматически при разрешении зависимостей. Этот пакет может быть установлен одновременно с библиотеками OpenSSL, входящими в состав дистрибутива.

`cryptopack-utils` — Содержит программу создания долговременных закрытых ключей `mkkey` и команду удаления секретных ключей `wipekey`. Должен быть установлен на компьютерах, где предполагается работа с ключевой информацией.

`libssl-r-dev` — Содержит заголовочные файлы и статические библиотеки, необходимые для сборки приложений, использующих СКЗИ «МагПро КриптоПакет». Устанавливается на компьютерах, где предполагается производить сборку приложений. Конфликтует с пакетом libssl0.9.8-dev из дистрибутива Debian, т.е. одновременно на одной машине может быть установлен только один из этих пакетов.

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

`openssl-r` — Содержит утилиту `openssl`, предоставляющую доступ к большей части функциональности продукта из командной строки и скрипты `mkreq`, `installcadata`, `c_rehash`, упрощающие работу с сертификатами и заявками.

`libengine-cryptocom-openssl` — Содержит модуль поддержки алгоритмов ГОСТ.

`yarrowd` — Содержит демон ДСЧ YARROW. Устанавливается, если предполагается использование ДСЧ YARROW.

3.3 Установка в системах Linux, использующих пакетный менеджер RPM

С помощью команды `rpm -i` можно установить следующие пакеты:

`cryptopack-openssl` — Содержит библиотеки OpenSSL, утилиту openssl и скрипты `mkreq`, `installcadata`, `c_rehash`, упрощающие работу с сертификатами и заявками. Требуется на всех компьютерах, использующих СКЗИ «МагПро КриптоПакет». Ставится в отдельное дерево файловой системы `/opt/cryptopack`, может быть использован одновременно с пакетом `openssl`, входящим в состав дистрибутива.

`cryptopack-openssl-devel` — Содержит заголовочные файлы и статические библиотеки, необходимые для сборки приложений, использующих СКЗИ «МагПро КриптоПакет». Требуется при необходиомсти сборки приложений с использованием СКЗИ «МагПро КриптоПакет». Ставится в `/opt/cryptopack` и может быть установлен одновременно с входящим в дистрибутив пакетом openssl-devel. Однако одновременная установка этих двух пакетов не рекомендуется, так как при наличии установленного openssl-devel легко ошибиться и использовать при сборке приложения библиотеку openssl, не поддерживающую ГОСТ.

`cryptopack-engine` — Содержит сертифицированный модуль поддержки ГОСТ и утилиты управления секретными ключами. Должен быть установлен на всех машинах, где используется СКЗИ «МагПро КриптоПакет».

`yarrowd` — Содержит демон ДСЧ YARROW. Устанавливается, если предполагается использование ДСЧ YARROW.

3.4 Установка под FreeBSD

Порядок установки СКЗИ «МагПро КриптоПакет» под FreeBSD:

1. Установить из портов tcl84, libiconv, а в версиях FreeBSD 5 и выше также Perl.
2. В ОС FreeBSD 5 и 6 для функционирования пакета cryptopack-engine необходимо установить библиотеку ncurses из пакета cryptopack-ncurses, входящего в комплект поставки СКЗИ «МагПро КриптоПакет». Установку библиотеки выполнить с помощью команды `pkg_add`.
3. Установить собственно СКЗИ «МагПро КриптоПакет». В комплект поставки для FreeBSD входят следующие пакеты:
   `cryptopack-openssl` — Содержит как runtime, так и development файлы модифицированной OpenSSL. Устанавливается в отдельное поддерево файловой системы `/usr/local/cryptopack`.
   `cryptopack-engine` — Содержит сертифицированный модуль поддержки алгоритмов ГОСТ.
   `yarrowd` — Содержит демон генератора случайных числе YARROW.

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

4. Для того, чтобы иметь возможность в ОС FreeBSD собирать приложения с использованием СКЗИ «МагПро КриптоПакет», необходимо применить к дереву портов `ports.diff.gz`, входящий в комплект поставки.
   Применение патча к дереву портов:

   ```
   cd /usr/ports
   gzip -dc ports.diff.gz |patch -p0
   ```

   После применения этого патча появляется опция сборки портов `WITH_OPENSSL_GOST`. При сборке порта с этой опцией он использует библиотеки OpenSSL из состава СКЗИ «МагПро КриптоПакет». Для ряда приложений при этом автоматически применяются патчи, требуемые для полноценной поддержки алгоритмов ГОСТ.

Во FreeBSD версии 4.x клавиатурный ДСЧ не поддерживается и программа mkkey в состав пакета не входит.

3.5 Установка под Solaris

Порядок установки под Solaris:

1. Установить требуемые пакеты из состава операционной системы (и, возможно, пакеты, необходимые для их нормального функционирования):
   `SUNWeeuos` — поддержка русских локалей
   `SUNWuiu8` — (кроме Solaris 8) поддержка русских кодировок в системной функции iconv
   `SUNWpl5u` — (Solaris 8 и 9) интерпретатор perl
   `SUNWperl584core` — (Solaris 10) интерпретатор perl
   `SUNWzlib` — библиотека компрессии zlib
   `SUNWdoc` — утилиты для работы с документацией
   `SUNWgccruntime` — (Solaris 10) Разделяемые библиотеки компилятора GCC
2. Установить вспомогательные пакеты, поставляемые вместе с продуктом:
   `CCOMtcl` — интерпретатор Tcl. Поставляется для Solaris 8-10
   `CCOMgclib` — библиотеки из комплекта компилятора GCC. Поставляются для Solaris 8 и 9. В Solaris 10 и выше вместо этого необходимо поставить пакет SUNWgccruntime из состава дистрибутива ОС.
   `CCOMiconv` — библиотека portable iconv. Поставляется для Solaris 8. В Solaris 9 и выше вместо этого используется системная функция iconv.
3. Установить пакеты самого продукта:
   `CCOMopenssl` — Содержит библиотеки Openssl, как runtime, так и development файлы, утилита OpenSSL
   `CCCOMengine` — Содержит модуль поддержки алгоритиов ГОСТ, утилиты для работы с секретными ключами.
   `CCOMyarrowd` — Содержит демон ДСЧ YARROW.
   (В Solaris 8 в связи с ограничениями на длину имени пакета эти пакеты называются CCOMossl, CCOMcpack и CCOMyarr соответственно.)

При нарушении этого порядка установка может не завершиться успешно, так как из-за отсутствия пререквизитов будет невозможно выполнить действия периода установки.

Установка пакетов производится с помощью утилиты pkgadd с указанием параметра -d, зависящего от версии Solaris и архитектуры машины.

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

Пакеты для всех версий Solaris для архитектуры Sparc и пакеты Solaris 10 для архитектуры x86 поддерживают одновременно 32-битные и 64-битные версии библиотек. 64-битных версий утилит управления ключами и yarrowd для Solaris не поставляется.

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

4 КОНФИГУРИРОВАНИЕ СКЗИ «МАГПРО КРИПТОПАКЕТ»

Для корректного функционирования СКЗИ «МагПро КриптоПакет» требуется задание некоторых параметров конфигурации в конфигурационном файле СКЗИ «МагПро КриптоПакет».

Поддерживается несколько вариантов поиска конфигурационного файла:

1. Общесистемный конфигурационный файл. Его расположение зависит от операционной системы:

ОС	Расположение файла
Win32	`C:\openssl\openssl.cnf`
Debian Linux	`/etc/ssl-r/openssl.cnf`
Другие Linux	`/opt/cryptopack/openssl.cnf`
FreeBSD	`/usr/local/cryptopack/openssl.cnf`
Solaris	`/opt/cryptopack/openssl.cnf`

2. Конфигурационный файл текущей сессии. Имя этого файла задается переменной окружения `OPENSSL_CONF`
3. Конфигурационный файл текущей операции. Задается опцией `-config` тех команд утилиты `openssl`, которые поддерживают конфигурационные файлы для текущей операции, или соответствующими опциями других приложений.

При любой операции используется только один конфигурационный файл. Поэтому, если используется файл текущей сессии или текущей операции, в него должны быть перенесены из общесистемного файла настройки СКЗИ «МагПро КриптоПакет».

4.1 Конфигурирование поддеркжи алгоритмов ГОСТ

По умолчанию библиотеки libcrypto и libssl, входящие в состав СКЗИ «МагПро КриптоПакет», предоставляют только ту же самую функциональность, что и соответствующие библиотеки из состава OpenSSL 0.9.8.

Чтобы включить поддержку алгоритмов ГОСТ, необходимо описать в конфигурационном файле подключение модуля engine libcryptocom.

При установке соответствующего пакета автоматически выполняется подключение этого модуля в системном конфигурационном файле.

Для подключения необходимо добавить в конфигурационный файл следующую информацию:

1. До названия первой секции (первая строка [в квадратных скобках]) следует поместить команду `openssl_conf`, указывающую на секцию с глобальными параметрами конфигурации (по умолчанию этой секции не существует в файле, ее необходимо добавить):

   ```
   openssl_conf = openssl_def
   ```

2. Добавить в конфигурационный файл (например, в конец файла) секцию, указанную выше, и вставить в нее команду `engines`, указывающую на секцию со списком модулей, которые необходимо подгрузить:

```
[openssl_def]
engines = engine_section
```

3. Добавить в конфигурационный файл секцию `engines`, содержащую строку с ID МагПро Engine и название секции, описывающей его конфигурацию:

```
[engine_section]
cryptocom = cryptocom_section %(для Cryptocom Engine)
```

4. Добавить в конфигурационный файл секцию, описывающую конфигурацию библиотеки. Эта секция должна содержать по меньшей мере две строки — в одной указывается полный путь к модулю, во второй указывается его ID.

```
[cryptocom_section]
engine_id = cryptocom
default_algorithms = ALL
```

Кроме этого, в секцию конфигурации библиотеки libcryptocom может быть включена конфигурационная информация самой библиотеки.

Эта информация включает в себя три параметра:

`RNG` — Тип датчика случайных чисел. Допустимые значения: `PROGRAM`, `YARROW`, `ACCORD`, `SOBOL`.

`RNG_PARAMS` — Дополнительные параметры датчика случайных чисел. Для программного датчика этот параметр указывает на расположения файла начального заполнения программного ДСЧ, если его местоположение не совпадает с умолчательным.

Для датчика YARROW этот параметр указывает TCP-порт на сетевом loopback интерфейсе, через который осуществляется доступ к yarrowd. По умолчанию 7670.

При установке пакета значение параметра `RNG` в системном конфигурационом файле устанавливается в `YARROW`. Если вы не планируете использовать на данной машине датчик YARROW и не устанавливаете соответствующий пакет, необходимо отредактировать конфигурационный файл вручную.

`CRYPT_PARAMS` — Параметры алгоритма шифрования ГОСТ 28147-89. Значением опции является OID параметров алгоритма шифрования (см табл. 2), который будет использоваться для зашифрования документов. На работу TLS этот параметр не влияет, так как параметры шифрования жестко фиксированы в спецификации шифрсьютов TLS.

Эти параметры конфигурации могут быть также заданы в environment с помощью переменных с теми же именами и значениями. Значения, заданные в environment, имеют приоритет перед значениями в конфигурационном файле.

Некоторые приложения (например, apache/mod_ssl, stunnel, openvpn) не считывают конфигурационный файл libcrypto, а предоставляют собственные средства конфигурации, позволяющие загружать модули engine. При использовании этих приложений необходимо в их конфигурационном файле указать использование engine с идентификатором cryptocom, а параметры библиотеки libcryptocom передавать через environment.

4.2 Конфигурирование информации о владельце сертификатов

Сертификаты формата X.509 содержат информацию о владельце сертификата, в том числе название организации, подразделения, местонахождение и так далее.

Эта информация может задаваться явно при создании заявки на получение сертификата, но в системном конфигурационном файле могут быть указаны умолчательные значения, которые позволят при создании заявки на сертификат указывать только собственно имя владельца.

В начале файла `openssl.cnf` определяются следующие переменные

Имя	значение
C	страна, двухбуквенный код по ISO-630
L	местоположение (название населенного пункта)
O	название организации
OU	название подразделения

Если в значениях этих переменных используются русские буквы, они должны быть в кодировке UTF-8.

Значения переменных CN (имя владельца сертификата) и E (адрес электронной почты), заданные в конфигурационном файле, при создании заявок с помощью скрипта `mkreq` не используются никогда. Они могут быть использованы только при непосредственном вызове команды `openssl req` из командной строки.

Значение адреса электронной почты по умолчанию в Unix-системах формируется из имени текущего пользователя и доменного имени, содержащегося в файле `/etc/mailname`, а в Windows берется из настроек Microsoft Internet Account Manager (т.е. будет использован адрес первого найденного аккаунта Outlook или Outlook Express).

В случае отсутствия файла `/etc/mailname` или отсутствия настроенного аккаунта Outlook Express скрипт `mkreq` завершается с ошибкой, если адрес электронной почты не был указан явно.

4.3 Формат файла конфигурации библиотеки libcrypto

В библиотеке libcrypto приложения могут автоматически конфигурировать определенные аспекты библиотеки с использованием основного конфигурационного файла СКЗИ «МагПро КриптоПакет» или опционально - с использованием альтернативного конфигурационного файла. Утилита openssl включает эту функциональность: любая команда утилиты использует основной конфигурационный файл, если в этой команде не указана опция, использующая альтернативный конфигурационный файл.

Чтобы включить конфигурирование библиотеки, умолчательный раздел конфигурационного файла должен содержать соответствующую строку, указывающую на главный конфигурационный раздел. Имя, используемое утилитой openssl по умолчанию - openssl_conf. Другие приложения могут использовать альтернативные имена, например myapplication_conf.

Конфигурационный раздел должен состоять из набора пар "имя-значение которые содержат информацию о конфигурации конкретных модулей. Имя представляет имя конфигурационного модуля, значение же зависит от характера модуля; например, оно может представлять собой дальнейший конфигурационный раздел, содержащий информацию, специфичную для этого конфигурационного модуля. Например:

```
openssl_conf = openssl_init
```

```
[openssl_init]

oid_section = new_oids
engines = engine_section

[new_oids]

... Здесь новые OID ...

[engine_section]

... Здесь информация об engine ...
```

В этом примере указаны два конфигурационных модуля. Один для объектов ASN.1, другой для конфигурации ENGINE.

4.3.1 Конфигурационный модуль для объектов ASN.1

Имя этого модуля - oid_section. Значение этой величины указывает на раздел, содержащий пары «имя - значение» для OID: имя - короткое и длинное имена OID, значение - численная форма OID. Хотя некоторые из команд утилиты openssl уже имеют собственную функциональность раздела для объектов ASN.1, но не все. С использованием данного конфигурационного модуля все команды утилиты openssl и все приложения, вызывающие конфигурирование, могут видеть новые объекты. Например:

```
[new_oids]

some_new_oid = 1.2.3.4
some_other_oid = 1.2.3.5
```

Также возможно указать в качестве значения длинное имя, за которым следуют запятая и численная форма OID. Например:

```
shortName = some object long name, 1.2.3.4
```

4.3.2 Конфигурационный модуль ENGINE

Этот конфигурационный модуль ENGINE называется engines. Значение этой переменной указывает на раздел, содержащий дальнейшую конфигурационную информацию ENGINE.

Раздел, на который указывает переменная engines, содержит таблицу имен различных engine (хотя см. ниже engine_id) и дальнейших разделов, содержащих конфигурационную информацию, относящуюся к конкретным ENGINE.

Каждый раздел, относящийся к конкретному ENGINE, используется для установки умолчательных алгоритмов, загрузки динамической информации, выполнения инициализации и

отдачи управляющих команд. Какая именно операция выполняется, зависит от имени команды, которое указано в качестве имени в паре «имя - значение». Команды, поддерживаемые в настоящее время, перечислены ниже.

Например:

```
[engine_section]

# Конфигурация ENGINE с именем "foo"
foo = foo_section
# Конфигурация ENGINE с именем "bar"
bar = bar_section

[foo_section]
... Команды, относящиеся к foo ENGINE ...

[bar_section]
... Команды, относящиеся к "bar" ENGINE ...
```

Команда engine_id используется для явного указания имени ENGINE. Если эта команда используется, она должна быть первой. Например:

```
[engine_section]
# здесь должен описываться ENGINE "foo"
foo = foo_section

[foo_section]
# Для этого ENGINE используется имя "myfoo" вместо "foo"
engine_id = myfoo
```

Команда dynamic_path загружает и добавляет ENGINE, находящийся по указанному пути. Она эквивалентна отдаче ENGINE dynamic (ENGINE, отвечающему за загрузку других ENGINE) управляющих команд: SO_PATH с аргументом path, затем LIST_ADD со значением 2 и LOAD. Если требуется другая модель загрузки, ее можно реализовать, передавая нужный набор управляющих команд непосредственно ENGINE dynamic (с помощью соответствующих управляющих команд).

Команда init определяет, инициализировать ENGINE или нет. Если ее значение 0, ENGINE не будет инициализирован, если ее значение равно 1, делается попытка немедленно инициализировать ENGINE. Если команда init не присутствует, то будет попытка инициализировать ENGINE после того, как будут выполнены все остальные команды из данного раздела.

Команда default_algorithms устанавливает умолчательные алгоритмы, которые ENGINE будет предоставлять при использовании функций **ENGINE_set_default_string()**.

Если имя команды не соответствует ни одной из указанных выше команд, считается, что это управляющая команда, которая посылается ENGINE. Значение этой команды передается в качестве аргумента команды. Если значением является строка EMPTY, никаких аргументов не передается.

Например:

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

```
[engine_section]

# Конфигурация ENGINE "foo"
foo = foo_section

[foo_section]
# Загрузка engine из DSO
dynamic_path = /some/path/fooengine.so
# управляющая команда, специфичная для foo.
some_ctrl = some_value
# еще одна управляющая команда, не имеющая аргументов.
other_ctrl = EMPTY
# Предоставить все умолчательные алгоритмы
default_algorithms = ALL
```

4.3.3 Примечания

Если конфигурационный файл пытается подставить значение несуществующей переменной, возвращается ошибка и файл не загружается. Это может произойти при попытке подставить значение несуществующей переменной среды.

Это можно обойти, включив умолчательное значение в умолчательный раздел: тогда, если поиск среди переменных среды не удастся, будет использована умолчательная переменная. Чтобы это работало корректно, умолчательная величина должна быть в конфигурационном файле определена до подстановки. В разделе 4.3.4 указан пример того, как это делается.

Если одна и та же переменная в одном и том же разделе упоминается несколько раз, то все значения, кроме последнего, игнорируются. В определенных обстоятельствах, например в distinguished name сертификата, одно и то же поле может встречаться несколько раз. Это обычно обходится с помощью игнорирования всех символов до первой точки. Например:

```
1.OU="My first OU"
2.OU="My Second OU"
```

4.3.4 Примеры

Здесь приводится пример конфигурационного файла, использующего некоторые возможности, указанные выше.

```
# Это умолчательный раздел.

HOME=/temp
RANDFILE= ${ENV::HOME}/.rnd
configdir=$ENV::HOME/config

[ section_one ]
```

```
# Теперь мы в первом разделе.

# Кавычки позволяют включать пробелы в конце и начале значений.
any = " any variable name "

other = A string that can \
cover several lines \
by including \\ characters

message = Hello World\n

[ section_two ]

greeting = $section_one::message
```

Следующий пример показывает, как выполняется безопасная подстановка переменных среды.

Предположим, что вы хотите, чтобы переменная tmpfile указывала на временный файл. Каталог, в котором она находится, может быть определен переменной среды TEMP или TMP, но этим переменным может вообще не быть присвоено никакого значения. Если вы просто укажете имена этих переменных среды, а соответствующие переменные окажутся несуществующими, это приведет к ошибке при попытке загрузить файл конфигурации. С использованием умолчательного раздела можно учесть значения обеих перменных, причем в данном случае TEMP имеет приоритет, а если ни одна не указана, используется /tmp.

```
TMP=/tmp
# Вышеуказанная величина используется, если переменная среды TMP
не определена
TEMP=$ENV::TMP
# Вышеуказанная величина используется, если переменная TEMP не
определена
tmpfile=${ENV::TEMP}/tmp.filename
```

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

5 ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ДАТЧИКОВ СЛУЧАЙНЫХ ЧИСЕЛ

5.1 Программный ДСЧ

Програмный датчик может быть использован для выработки подписи под документами, выработки эфемерных ключей распределения ключей и сеансовых ключей шифрования, а также создания TLS-соединений.

Программный датчик нельзя использовать для создания долговременных закрытых ключей. Программа создания закрытых ключей mkkey в случае отсутствия других ДСЧ использует не программный, а клавиатурный ДСЧ.

Программный датчик при инициализации считывает файл начального заполнения и модифицирует его с тем, чтобы при следующем запуске начальное заполнение отличалось. Поэтому файл начального заполнения должен быть доступен для записи для процесса, использующего клавиатурный ДСЧ.

По умолчанию файл начального заполнения создается в каталоге `$HOME/.magprocryptopack/random_seed` на платформах unix и в `%APPDTA%\MagProCryptoPack\random_seed` на платформах Win32. Другое расположение может быть задано с помощью переменной среды `RNG_PARAMS` или с помощью соответствующего параметра конфигурационного файла `openssl.cnf`. В последнем случае при инициализации этого файла программой mkkey следует явно указать путь к файлу в командной строке.

Файл начального заполнения должен защищаться от несанкционированного доступа так же, как и хранящиеся в файловой системе закрытые ключи.

5.2 ДСЧ YARROW

YARROW — программный ДСЧ разработки ООО «Криптоком», который поставляется вместе с СКЗИ «МагПро КриптоПакет». ДСЧ YARROW может быть использован для работы с СКЗИ «МагПро КриптоПакет» на любом компьютере, при работе с ДСЧ YARROW не нужна ни установка дополнительных плат, ни клавиатурная инициализация ДСЧ. Инициализация ДСЧ YARROW занимает определенное время (до 40 сек.)

Для корректной работы ДСЧ YARROW при загрузке операционной системы должен быть запущен програмный компонент `yarrowd` (`yarrow.exe`). Этот компонент должен запускаться с минимальными правами, как правило, от имени специально созданного пользователя.

В большинстве систем необходимые для этого действия производятся автоматически при инсталляции СКЗИ «МагПро КриптоПакет».

ДСЧ YARROW имеет несколько командно-строчных параметров, которые могут быть полезны для системного администратора:

- port (7670) "TCP Port number"
- loglevel (notice) "Log level: debug, info, notice, warning or error"
- logfile (отсутствует) "Log file, empty for no logging"
- stopfile (отсутствует) "Stop-file-— файл, в который пишется порт, и по исчезновении которого программа завершается.

Следует учитывать, что ДСЧ YARROW работает только на интерфейсе 127.0.0.1. Порт по

умолчанию — 7670, возможно его конфигурирование с помощью соответствующего командно-строчного параметра.

В случае, если yarrowd запущен не на умолчательном порту, порт, доступ к которому должны осуществлять приложения, должен быть указан в параметре конфигурации `RNG_PARAMS`.

5.3 ДСЧ ACCORD

ДСЧ ACCORD — это аппаратный ДСЧ, входящий в состав устройства защиты от НСД «Аккорд». Использование этого ДСЧ поддерживается на платформах Linux и Windows. Для использования этого ДСЧ должны быть установлены драйвера платы «Аккорд» для соответствующей операционной системы, поставляемые производителем устройства.

5.4 ДСЧ SOBOL

ДСЧ SOBOL — это аппаратный ДСЧ, входящий в состав устройства защиты от НСД «Соболь». Использование этого ДСЧ поддерживается только на платформах Windows. Для использования этого ДСЧ должны быть установлены драйвера платы «Соболь» для соответствующей операционной системы, поставляемые производителем устройства.

6 КОНФИГУРИРОВАНИЕ ПРИЛОЖЕНИЙ

6.1 Список приложений, протестированных на совместимость с СКЗИ «МагПро КриптоПакет»

Ниже приводится список приложений, для которых гарантируется работа с СКЗИ «МагПро КриптоПакет»

- WWW-сервер Apache версий 1.3.x и 2.2.x
- TLS Stunnel
- Сервер каталогов OpenLDAP
- OpenVPN
- Почтовый сервер Postfix
- IMAP и POP3 сервер dovecot
- Web-браузер lynx
- Программа получения файлов wget
- Почтовый клиент mutt
- Средство разработки tcltls
- Модуль интерфейса с openssl для языка Ruby

Для работы с библиотекой libcryptocom необходимо соответствующее конфигурирование этих приложений. Подробно конфигурирование каждого приложения описано ниже.

6.2 Серверные приложения

6.2.1 www-сервер Apache

Apache (`http://httpd.apache.org`) — один из самых распространенных в мире www-серверов.

Для Apache существуют модули организации защищенных соединений. Наиболее распространенным из существующих является модуль `mod_ssl` (`http://www.modssl.org`).

В коде `mod_ssl` предполагается, что существует только два типа асимметричных ключей — RSA и DSA. Поэтому для реализации поддержки подгружаемых алгоритмов при работе с Apache требуется внести небольшие изменения в код `mod_ssl`, дважды заменив в коде явное указание константы `EVP_PKEY_DSA` на обращение к типу ключа из загруженного сертификата. Патч, выполняющий эту задачу, можно загрузить с сайта `http://www.cryptocom.ru`.

Кроме того, при сборке `apache 1.3.x +mod_ssl mod_ssl` следует конфигурировать с указанием ключа *–enable-rule=SSL_EXPERIMENTAL*, так как возможность указания используемого модуля до сих пор является экспериментальной возможностью, не включаемой по умолчанию.

Для подключения российских алгоритмов к Apache следует:

1. Создать закрытый ключ соответствующего алгоритма и получить сертификат на него.
2. Указать в файле конфигурации Apache (*httpd.conf*) директиву SSLCryptoDevice <имя engine>
3. Указать в том же файле с помощью директивы SetEnv тип и, если необходимо, параметры ДСЧ.

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

```
SSLCryptoDevice cryptocom
SetEnv PROGRAM
SetEnv RNG_PARAMS /etc/apache/cryptopack.rnd
```

6.2.2 Серверное приложение Stunnel

Stunnel (www.stunnel.org) — небольшое серверное приложение, позволяющее «обернуть» в TLS работу практически любой сетевой программы, использующей протокол TCP. Может быть использован для организации защищенного доступа к почтовым (IMAP4 и POP3) серверам.

Stunnel версии 4.x поддерживает указание динамически подгружаемых модулей в своем конфигурационном файле. Но в коде приложения имеется одно место, где явным образом предполагается исользование RSA. Поэтому, если преполагается использование только российских алгоритмов, `stunnel` следует собрать с опцией *–disable-rsa*. В силу исторических причин эта опция функционирует как «Разрешить все алгоритмы кроме RSA», а не как «Запретить использование RSA». Т.е при необходимости можно использовать собранный таким образом stunnel и с сертификатами RSA.

Следует задать в скрипте запуска stunnel переменные среды, определяющие тип ДСЧ.

6.2.3 Сервер каталогов OpenLDAP

Пакет OpenLDAP (`http://www.openldap.org`) объединяет в себе сервер, клиентскую библиотеку и ряд утилит для работы с протоколом LDAP.

В конфигурационном файле сервера Filenameslapd версии `openldap 2.2.23` не предусмотрено возможности явного указания динамически подключаемого модуля. Разработан патч, который обеспечивает считывание библиотекой `libldap` конфигурационного файла OpenSSL, что решает проблему подгрузки модулей как в серверных, так и клиентских приложениях, использующих библиотеку. Патч может быть загружен с сайта `http://www.cryptocom.ru`.

6.2.4 Виртуальная защищенная сеть OpenVPN

OpenVPN (`http://www.openvpn.org`) — законченное решениие для создания защищенных виртуальных сетей. Поддерживаются платформы Windows, Linux, FreeBSD, Solaris. Полноценная поддержка российских алгоритмов в OpenVPN требует модификации приложения, так как работа с алгоритмами симметричного шифрования и имитозащиты конфигурируется на уровне имитозащиты. Патч для версии 2.1rc4 может быть загружен с сайта `http://www.cryptocom.ru`.

Поддерживаются следующие возможности использования алгоритмов ГОСТ в openvpn

1. Использование алгоритма шифрования ГОСТ 28147-89 для шифрования траффика (`cipher gost89`)
2. Использование имитозащиты ГОСТ 28147-89 для контроля целостности траффика (`auth gost-mac`)
3. Использование HMAC на базе алгоритма хеширования ГОСТ Р 34.11-94 для контроля целостности траффика (`auth md_gost94`)
4. Использование гостовских шифрсьютов TLS при обмене ключами и аутентификации в TLS-режиме. Для этого в конфигурационном файле openvpn необходимо явным образом указать опцию `tlscipher` с соответствующим именем шифрсьюта и использовать сертификаты и секретные ключи ГОСТ.

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

6.2.5 Почтовый сервер Postfix

Postfix (`http://www.postfix.org`) — мощный и гибки OpenSource почтовый сервер. Использует TLS для создания защищенных каналов при аутентификации клиента, а также может производить аутентификацию клиента по клиентскому сертификату.

Патч, обеспечивающий считывание конфигурационного файла libcrypto можно получить с Web-сайта `http://www.cryptocom.ru`.

6.3 Клиентские программы

6.3.1 Web-браузер lynx

lynx — текстовый web-браузер. После пересборки lynx с использованием OpenSSL с нашими модификациями позволяет осуществлять доступ к сайтам, использующим российские алгоритмы в HTTPS. Для подключения библиотеки libcryptocom использует конфигурационный файл OpenSSL.

6.3.2 Программа получения файлов wget

wget — неинтерактивная программа получения файлов по протоколам HTTP/HTTPS/FTP/FTPS. Для использования библиотеки libcryptocom требуется добавить считывание конфигурационного файла OpenSSL.

6.3.3 Почтовый клиент mutt

mutt — почтовый клиент. Поддерживает работу с S/MIME посредством вызова внешней утилиты OpenSSL и использование TLS при работе с удаленными IMAP и POP3-серверами. Для использования библиотеки libcryptocom при работе с TLS требуется добавить считывание конфигурационного файла OpenSSL. Для использования российских криптоалгоритмов в S/MIME изменения исходных текстов не требуется. В конфигурационном файле необходимо указать

```
set smime_encrypt_with="cipher_alg gost89"
```

6.4 СУБД

6.4.1 PostgreSQL

Начиная с версии 8.3 PostgreSQL поддерживает работу с конфигурационным файлом libcrypto, а также использование аппаратных ключевых носителей при использовании клиентских сертификатов TLS.

Патч, обеспечивающий эту функциональность в более ранних версий PostgreSQL (8.1.x и 8.2.x) можно получить с сайта `http://www.cryptocom.ru`

6.5 Средства разработки

6.5.1 tcltls

tcltls — расширение Tcl, позволяющее создавать как клиенты, так и сервера различных TLS-протоколов на языке Tcl. Для поддержки библиотеки libcryptocom необходима

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

либо возможность чтения файла конфигурации OpenSSL, либо возможность явного указания библиотеки. Данная возможность реализуется патчем #1353033 в patch manager на `http://www.sourceforge.net/project/tls`.

6.5.2 Интерфейс к openssl языка Ruby

В состав дистрибутива интерпретатора Ruby входят модули обеспечивающие возможность работы с защищенными документами средствами библиотеки OpenSSL. Для обеспечения возможности проверки подписи и расшифрования документов ГОСТ с помощью СКЗИ «МагПро КриптоПакет» требуется в исходном тексте инициализации модуля раскомментировать строку загрузки конфигурационного файла libcrypto.

7 ПРИЛОЖЕНИЯ

7.1 Датчики случайных чисел для работы библиотеки libcryptocom

В таблице 1 приведен список датчиков случайных чисел, которые может использовать библиотека libcryptocom. Выбор датчика осуществляется установкой соответствующего значения переменной среды $RNG. Для некоторых датчиков необходима также установка значения переменной среды $RNG_PARAMS.

Значения переменной среды $RNG и необходимость установки значения переменной среды $RNG_PARAMS для каждого из возможных ДСЧ указаны в таблице 1.

Таблица 1
Датчики случайных чисел, с которыми работает библиотека libcryptocom

?	**Название датчика**	**Значение переменной RNG**	**Примечание**
1.	Программный	PROGRAM	Требует указания файла начального заполнения в переменной RNG_PARAMS. Не может быть использован для создания долговременных ключей.
2.	Системный	YARROW	Требует запущенного демона yarrowd. Готов к работе только через некоторое время после запуска демона.
3.	Клавиатурый	KEYBOARD	Требует взаимодействия с пользователем. Может быть использован только в специальных приложениях, написанных с учетом использования этого датчика.
4.	Аккорд	ACCORD	Аппаратный ДСЧ с платы «Аккорд». Поддерживается под Windows и Linux.
5.	Соболь	SOBOL	Аппаратный ДСЧ с платы «Соболь». Поддерживается под Windows и Linux.

7.2 Названия и параметры алгоритмов

В таблице 2 приведены названия и параметры алгоритмов, которые следует указывать в командной строке в тех случаях, когда команда или опция требуют указания алгоритма и его параметров (например, в качестве аргумента опции `-newkey` команды `openssl req` при создании ключей).

Работа с параметрами алгоритмов в варианте CryptoPro подробно описана в документе RFC 4357. Параметры XA и XB используются для обмена.

Таблица 2
Названия и параметры алгоритмов библиотеки libcryptocom

Алгоритм	**Название**	**Поддерживаемые параметры**
Подпись ГОСТ Р 34.10-94	gost94	A,B,C,D,XA,XB,XC
Хэш ГОСТ Р 34.11	md_gost94	
Подпись ГОСТ Р 34.10-2001	gost2001	A,B,C,XA,XB,test
Шифрование ГОСТ 28147-89		cp_cipher_param_a
		cp_cipher_param_b
		cp_cipher_param_c
		cp_cipher_param_d

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

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