127018, Москва, ул. Сущёвский вал, д. 18
Телефон: +7 (495) 995 4820
Факс: +7 (495) 995 4820
https://www.CryptoPro.ru
E-mail: info@CryptoPro.ru

Средство Криптографической Защиты Информации

Криптопро HSM версия 2.0

Комплектация 3

Описание процедуры сборки

ЖТЯИ.00096-02 94 01

Листов 40

2020 г.

СОДЕРЖАНИЕ

1. ОБЩИЕ УКАЗАНИЯ

1.1. Для сборки ПАКМ «КриптоПро HSM» требуется комплект документации.

1.2. Требуется аппаратный модуль, соответствующий спецификации на оборудование. Спецификация на оборудование приведена в документе «ЖТЯИ 00096-02. КриптоПро HSM. Технические Условия».

1.3. Требуется дополнительное оборудование: дисковод CD-ROM с интерфейсом SATA и флоппи дисковод.

1.4. Требуется диск №1 из дистрибутива Альт Линукс СПТ 7.0.

1.5. Инсталляция программного обеспечения ПАКМ проводится со специального дистрибутивного диска при помощи инсталляционных скриптов, находящихся на этом же диске.

1.6. Состав дистрибутивного диска и тексты инсталляционных скриптов с комментариями приведены в Приложениях 4.1 и 4.2.

2. ПОДГОТОВКА К ПРОЦЕДУРЕ ИНСТАЛЛЯЦИИ

2.1. Подготовительные поцедуры

С ПАКМ необходимо снять крышку и отсоединить заднюю блокирующую планку.

Подсоединить CD-ROM дисковод и флоппи-дисковод.

Подсоединить монитор, клавиатуру, манипулятор «мышь» и кабели питания.

2.2. Настройка CMOS BIOS

Подробное описание настроек CMOS BIOS приведено в Приложении 4.3.

2.3. Инициализация электронного замка «Соболь».

Для инициализации ЭЗ «Соболь» необходимо иметь две «таблетки» (ключи электронного замка ПАКМ). Одна из них будет инициализирована как администраторская, другая – как пользовательская.

Первый этап инициализации.

1) Выключить питание ПАКМ.
2) Снять крышку ПАКМ.
3) Снять перемычки с платы ЭЗ «Соболь».
4) Включить питание ПАКМ.
5) Появится окно меню «Режим инициализации». Необходимо войти в подменю «Диагностика платы» и выбрать пункт «Выполнить все тесты». После этого будет выполнено тестирование Соболя, и если он исправен, появится сообщение об успешном сообщении трех тестов, а также строчка «Ошибка чтения идентификатора: устройство отсутствует в считывателе».
6) Далее отдельно проводится тестирование считывателя. Для этого в меню «Диагностика платы» выбирается пункт «Тест считывателя iButton». После этого появится надпись «Предъявите идентификатор и удерживайте его». Надо прислонить *администраторскую* «таблетку» (она должна быть чем-то выделена, например, надписью на ней), и удерживать ее, пока тест не завершится.
7) Если тестирование завершилось успешно, надо выйти из меню «Диагностика платы» в меню «Режим инициализации», и выбрать пункт «Инициализация платы».
8) Появится окно «Общие параметры системы».
9) Установить следующие значения общих параметров:

Автономный режим работы	- Да
Контроль целостности файлов и секторов	- Нет
Число попыток тестирования ДСЧ	- 3
Тестирование ДСЧ для пользователей	- Да
Показ статистики пользователю	- Нет
Минимальная длина пароля пользователя	- 0

Максимальный срок действия пароля (дней) - 0

Предельное число неудачных входов пользователя - 0

Ограничение времени на вход в систему - 0

Время ожидания сторожевого таймера (сек) - 80

Период тестирования сторожевого таймера (дней) - 0

Выйти из меню клавишей «Esc».

10) После завершения тестирования перейти в верхний уровень меню режима инициализации клавишей «Esc» и выбрать пункт «Инициализация платы».
11) В соответствии с инструкцией ЭЗ «Соболь» произвести процедуру инициализации ЭЗ, генерацию и запись на «таблетку-идентификатор» ключа Администратора ЭЗ.
12) Выключить питание ПАКМ установить на место перемычки ЭЗ «Соболь», включить питание.
13) Зайти в раздел меню «Список пользователей». Создать пользователя. Для этого нужно нажать клавишу Ins и ввести имя пользователя (например, user1), затем нужно войти в меню пользовательских настроек (клавиша Enter) и выставить следующие настройки:

Режим контроля целостности - Жесткий

Запрет загрузки с внешних носителей - Да

Ограничение срока действия пароля - Нет

Замена аутентификатора при смене пароля - Нет

Выйти из меню настройки «Соболя» двойным нажатием на клавишу «Esc».

3. ПРОЦЕДУРА ИНСТАЛЛЯЦИИ ПРОГРАММНО-АППАРАТНОГО КРИПТОГРАФИЧЕСКОГО МОДУЛЯ

Процедура инсталляции ПАКМ «КриптоПро HSM» проходит в четыре этапа:

- установка базовой операционной системы;
- установка обновлённого ядра операционной системы и его окружения;
- установка и конфигурирование программного обеспечения;
- включение процедуры контроля целостности ПО при помощи ЭЗ «Соболь».

3.1. Для проведения процедуры инсталляции «КриптоПро HSM» необходимо иметь:

- диск №1 из дистрибутива Альт Линукс СПТ 7.0;
- дистрибутивный CD-ROM «КРИПТОПРО HSM»;
- набор дискет с ключевым материалом (например, ДСДР);
- *администраторскую и пользовательскую* «таблетки» для электронного замка «Соболь».

3.2. Устройства CD-ROM и ГМД 3,5'' должны быть подключены. К ПАКМ также должны быть подключены монитор, клавиатура и манипулятор «мышь».

3.3. В дисковод CD-ROM должен быть вставлен первый диск Альт Линукс СПТ 7.0 и ПАКМ перезагружен.

3.4. В появившемся меню выбираем пункт «Installation» и нажимаем клавишу Enter.

3.5. На предложении выбрать язык установки - выбираем Английский и нажимаем ОК.

3.6. На странице "License agreement" ставим галочку "I accept" и нажимаем Next.

3.7. На странице "Volume management" выбираем пункт «Custom», ставим галочку перед пунктом "Clear all before applying profile" и нажимаем Next.

3.8. Разбиение дисков проводим согласно следующей схеме:

Устройство	Раздел	Размер	Файл. Сист.	Точка монтир.
hda	sda1	50 MB	Ext2/3	/boot
	sda2	(оставш.)	Ext2/3	/
hdb	sdb1	20 MB	Ext2/3	/home
	sdb2	120 MB	Ext2/3	/tmp
	sdb3	(оставш.)	Ext2/3	/var
hdd	sdd1	(оставш.)	Ext2/3	/backup

3.9. Процедура разбиения проводится следующим образом: выбирается устройство (диск), например, sda, нажимается кнопка “Create Partitions”. В появившемся окошке вводится нужный размер раздела (строка “Size”), например, 50 MB, нажимаем OK. В следующем окне выбираем файловую систему Ext2/3, нажимаем OK. Выбираем точку монтирования, нажимаем OK. Переходим далее к формированию следующего раздела, выбираем на этом диске неиспользуемое место (unusd) и снова нажимаем “Create partitions” и т.д., а в конце процедуры разбиения на разделы нажимаем кнопку Next.

Начинается процедура копирования файлов на диски.

3.10. После завершения процедуры копирования появляется страница Bootloader setup. На ней устанавливаем положение загрузчик в Master Boot Record первого диска (sda).

3.11. На следующей странице задаем пароль для администратора.

3.12. Далее, обязательно требуется завести какого-то пользователя. Заводим пользователя с именем user1 и задаём ему пароль.

3.13. На шаге, где предлагается выбрать дополнительные серверные пакеты для установки (Additional packages), никакие пакеты не выбираем, нажимаем Next.

3.14. Сетевые интерфейсы настраиваем согласно следующей схеме:

eth0, IP=192.168.26.2, mask=255.255.255.0 gateway=192.168.26.1

eth1, IP=192.168.27.2, mask=255.255.255.0

eth2, IP=192.168.28.2, mask=255.255.255.0

Если в системе используются только два оптических сетевых интерфейса, то вводятся только две первые строчки таблицы.

3.15. Сетевые интерфейсы настраиваются следующим образом. Все параметры всех сетевых интерфейсов устанавливаются на одной закладке ”IPinterfaces”. Выбирается интерфейс в строчке “Interface” (например, eth0), ставится «галочка» против пункта “Interface is enabled”, задаётся IP address, выбирается NetMask, задаётся Default gateway (если необходимо). Затем снова переходим к строке “Interface”, выбираем eth1 и задаем все параметры для него и т.д.

3.16. Когда параметры для всех сетевых интерфейсов заданы, переходим к вкладке “General Network Setting”. Здесь в строке “Hostname” задаём серийный номер HSM-а (например, HSM32-00081a) не более 12 символов, далее нажимаем кнопку Next. На следующем шаге выбираем часовой пояс (Time zone). Строчки Country: Britain (UK), zone: London.

3.17. На следующей странице нажимаем кнопку Finish и, когда компьютер пойдет на перезагрузку удаляем CD-ROM из дисковода, далее выключаем питание.

3.18. После включения питания и прохождения контроля загрузки через ЭЗ «Собль» никаких пунктов в загрузочном меню не выбираем, загрузка должна пройти по умолчанию.

3.19. Устанавливаем в CD-ROM дисковод дистрибутивный CD-ROM «КРИПТОПРО HSM» и выполняем команды:

```
mount /dev/cdrom /mnt/cdrom
cd /mnt/cdrom/stage2
./update.sh
```

Ждем, пока пройдет установка дополнительных пакетов. Она должна завершиться сообщением “Done”. Отмонтируем CD-ROM дисковод командой umount /mnt/cdrom и даем команду halt. Выключаем питание.

3.20. Включаем питание, ждем, когда пройдет загрузка, заходим в систему как root. Устанавливаем в дисковод дистрибутивный CD-ROM «КРИПТОПРО HSM», выполняем команды:

```
mount /dev/cdrom /mnt/cdrom
cd /mnt/cdrom/stage2
/mnt/cdrom/stage2/installhsm2r.sh
```

После этого начнется установка пакетов прикладного программного обеспечения.

3.21. В процессе установки программных пакетов будут заданы следующие вопросы:

- Отключить проверку ECC памяти в crontab?
- Удалить rpm-пакет из релизной версии?
- Выбрать тип используемой LCD-панели.
- Выбрать скорость работы COM-порта.
- Выбрать тип считывателя смарт-карт.
- Выбрать количество установленных оптических сетевых интерфейсов.
- Ввести подтверждение на использование ДСДР.
- Ввести имя устройства, используемого для чтения ДСДР.

3.22. После завершения установки пакетов и конфигурирования ПАКМ будет автоматически выполнена команда halt (с автоматическим выключением питания).

3.23. Включаем питание, для продолжения загрузки используем Администраторскую таблетку ЭЗ «Соболь». На этом шаге необходимо включить функцию обеспечения контроля целостности файлов при помощи ЭЗ «Соболь». Для этого в меню Администратора ЭЗ нужно выбрать пункт «Общие параметры системы», далее пункт «Контроль целостности файлов и секторов» и нажать «Enter», а затем «Esc». Теперь выбираем пункт «Расчет контрольных сумм», затем пункт «Загрузка операционной системы».

3.24. После того как пройдет загрузка нажимаем кнопку «Enter» на LCD-панели ПАКМ. На LCD-панели появится надпись Init HSM? Yes[x] No[x], при помощи кнопки Esc или стрелок нужно выбрать пункт No[x] и нажать Enter, ПАКМ выключится.

3.25. Отключить от ПАКМ все внешние устройства (CD-ROM дисковод, флоппи-дисковод, монитор, клавиатуру и манипулятор «мышь»).

3.26. Установить верхнюю крышку и закрыть защитной планкой интерфейсные разъемы на задней панели ПАКМ.

3.27. Включить питание ПАКМ и провести загрузку, используя пользовательскую таблетку ЭЗ «Соболь». Убедиться в работоспособности изделия.

4. ПРИЛОЖЕНИЯ

4.1. Состав дистрибутивного диска «Крипто-Про HSM».

Дистрибутивный диск имеет следующую структуру:

/.disk/

/isolinux/

/Metadata/

/license.txt

/license.txt.ru

/rescue

/stage1

/stage2

/stage2/conf/

/stage2/db/

/stage2/mysql

/stage2/post-install/

/stage2/RPMS/

/stage2/installhsm2r.sh,

Данный диск является загрузочным и может использоваться для загрузки с него при проведении сервисных работ изготовителем ПАКМ. Каталоги .disk. isolinux, Metsdata и файлы rescue, license.txt, license.txt.ru – являются системными, необходимыми для загрузки с данного диска.

Файл installhsm2r.sh представляет собой собственно инсталляционный скрипт. Содержание этого скрипта приводится ниже.

Директория /stage2/conf/ содержит конфигурационные файлы:

- files — содержит список файлов, целостность которых контролируется в системе;
- Info.plist — содержит актуальный список считывателей смарт карт;
- logrotate.conf — конфигурационный файл, задающий параметры ротации файлов системного журнала (журнала событий);
- pcscd — конфигурационный файл процесса, обслуживающего работу считывателя смарт-карт;

- rc.local — конфигурационный файл заключительного этапа загрузки системы;
- syslog.conf — конфигурационный файл процесса, ведущего системный журнал.

Директория /db/ содержит файлы user_db.sql и log_db.sql – это файлы конфигураций баз данных пользователей и журнала аудита.

Директория /stage2/post-install/ содержит следующие файлы:

- good-passwd — исполняемый файл для генерации скрытых паролей из случайных чисел в операционной системе;
- sblchk — файл шаблонов для расчета контрольных сумм файлов, целостность которых контролирует ЭЗ «Соболь».

Директория /stage2/RPMS/ содержит файлы rpm-пакетов ПО «Крипто-Про HSM». Список этих файлов:

ccid-1.0.1-5.x86_64.rpm

cprocsp-rdr-pcsc-64-5.0.0-4.hsm.rpm

cprocsp-pam_hsm-64-5.0.0-4.hsm.rpm

cprocsp-stunnel-64-5.0.0-4.hsm.rpm

cprocsp-compat-altlinux-64-1.0.0-1.noarch.rpm

cprocsp-fenixm-server-64-5.0.0-4.hsm.rpm

cprocsp-hsm_db-64-5.0.0-4.hsm.rpm

cprocsp-hsm_webadmin-64-5.0.0-4.hsm.rpm

dosfstools-2.11-alt4.x86_64.rpm

lsb-cprocsp-base-5.0.0-4.noarch.rpm

lsb-cprocsp-capilite-64-5.0.0-4.hsm.rpm

lsb-cprocsp-rdr-64-5.0.0-4.hsm.rpm

lsb-cprocsp-rdr-sobol-64-5.0.0-4.hsm.rpm

pcsc-lite-1.3.1-7.x86_64.rpm

pcsc-lite-doc-1.3.1-7.x86_64.rpm

pcsc-lite-libs-1.3.1-7.x86_64.rpm

sobol-2.6.27-sec-def-alt1.M40.1-1-1.x86_64.rpm

Директория /stage2/mysql/ содержит файлы, необходимые для автоматической установки и настройки сервиса mysql.

4.2. Текст инсталляционного скрипта installhsm2r.sh

```
#!/bin/bash
# version=HSMDB 01
# emulate stage2 dir to be CD root
case `dirname $0` in
*stage2*)
    mkdir -p /mnt/cdrom
    mount -o bind `dirname $0` /mnt/cdrom
    cd /mnt/cdrom
#    exec ./`basename $0`
    ;;
esac

IsDEBUG=false
BOOTasVFAT=true
CONFIGNETWORK=true

read -p "Is this Debug Installation (y/n)?"
if [ x$REPLY = xy ];then
    IsDEBUG=true
    read -p "Convert /boot partition to VFAT (y/n)?"
    if [ x$REPLY != xy ];then
      BOOTasVFAT=false
    fi
    read -p "Configure network interfaces by this script(y/n)?"
    if [ x$REPLY != xy ];then
      CONFIGNETWORK=false
    fi
fi


probe_hdsd()
{
      if dd if=/dev/hda of=/dev/null count=1 bs=1 1>/dev/null 2>&1
      then
        DISK=/dev/hda
        BOOTD=/dev/hda1
      elif dd if=/dev/sda of=/dev/null count=1 bs=1 1>/dev/null 2>&1
      then
        DISK=/dev/sda
          BOOTD=/dev/sda1
      else
        echo "No disk found!!!"
        exit 1
      fi
}
fw_init()
{
            iptables --flush
            iptables -P INPUT DROP
            iptables -P OUTPUT DROP
            iptables -P FORWARD DROP
            service iptables save
          chkconfig --levels 345 iptables on
}

copy_key_material()
{
    if [ $IsDEBUG = true ]; then
            #copy KPIM sequence from cdrom for debug purposes only
                cp -f /mnt/cdrom/conf/kis_1 /var/opt/cprocsp/dsrf/db1/
```

```
                cp -f /var/opt/cprocsp/dsrf/db1/kis_1
/var/opt/cprocsp/dsrf/db2/kis_1
                ls -l /var/opt/cprocsp/dsrf/db?/kis_1
    else
        read -p "Use KPIM instead of dsrfcopy(y/n)?"
        if [ x$REPLY == xy ]
        then
            make_KPIM
        else
          copy_DSRF
        fi
    fi
}

copy_DSRF()
{
      local dsrfdev="/dev/fd0"
        while true
        do
                read -p "Input DSRF device name($dsrfdev).:"
                if [ x$REPLY != x ]
                then
                        dsrfdev=$REPLY
                fi
                /opt/cprocsp/sbin/amd64/dsrfcopy $dsrfdev
                read -p "Add another DSRF to the pool(y/n)?"
                if [ x$REPLY != xy ]
                then
                        break
                fi
        done
}

make_KPIM()
{
      dd if=/dev/sobol of=/var/opt/cprocsp/dsrf/db1/kis_1 bs=36 count=100000
      cp -f /var/opt/cprocsp/dsrf/db1/kis_1 /var/opt/cprocsp/dsrf/db2/
      ls -l /var/opt/cprocsp/dsrf/db?/kis_1
}

make_SerialNumberPool()
{
      mkdir -p /var/opt/cprocsp/users/0.0
      local POOL=/var/opt/cprocsp/users/0.0/SerialNumberPool
if [ $IsDEBUG = true ]; then
        dd if=/mnt/cdrom/conf/kis_1 of=$POOL bs=10 count=1
else
        dd if=/dev/sobol of=$POOL bs=10 count=1
fi
}

replace_all()
{
      sed "s/$2/$3/g"<$1 >$1.new
      mv -f $1.new $1
}

change_hostname()
{
      old_hostname=`cat /etc/HOSTNAME`
      while true
      do
          read -p "Enter HSM Name:" new_hostname
          if ! echo "$new_hostname"|grep -q '^[[:alpha:]][[:alpha:][:digit:]_-]*$'
```

```
            then
              echo "Invalid HSM name:\"$new_hostname\""
              continue
            fi
            read -p "Your new HSM name is $new_hostname. Is it correct(y/n)?"
            [ x$REPLY = xy ] || [ x$REPLY = xY ] && break
        done
        export HOSTNAME=$new_hostname
        for f in /etc/HOSTNAME /etc/issue.net
        do
            replace_all $f $old_hostname $new_hostname
        done
        # add name "HSM" to /etc/hosts for mysql connection
        replace_all /etc/hosts $old_hostname "$new_hostname HSM"
}

make_config64_ini()
{
    local log_format=57
    local log_mask=1
if [ $IsDEBUG = true ]; then
      log_format=7f
        log_mask=f
fi
    CPCONFIG=/opt/cprocsp/sbin/amd64/cpconfig
    $CPCONFIG -ini '\config\random\bio_tui' -add string dll librdrrndmbio_tui.so
    $CPCONFIG -ini '\config\random\dsrf' -add string dll librdrdsrf.so
    $CPCONFIG -ini '\config\random\cspd' -add string dll librdrdsrf.so

    $CPCONFIG -ini '\config\keydevices\fat12' -add string dll librdrfat12.so
    $CPCONFIG -ini '\config\keydevices\fat12' -add long  group 1
    $CPCONFIG -ini '\config\keydevices\hdimage' -add string dll librdrfat12.so
    $CPCONFIG -ini '\config\keydevices\hsm' -add string dll librdrfat12.so

    $CPCONFIG -ini '\config\capilite\settings' -add long max_elemnts 100
    $CPCONFIG -ini '\config\capilite\settings' -add long fresh_time 3600

if [ $IsDEBUG = true ]; then
    $CPCONFIG -ini '\config\parameters' -add long  TesterPeriod 599
else
    $CPCONFIG -ini '\config\parameters' -add long  TesterPeriod 10
fi
# TODO! pkzi_build read from another config
    $CPCONFIG -ini '\config\parameters' -add long pkzi_build 7437
    $CPCONFIG -ini '\config\parameters' -add long max_rpc_session 2048
    $CPCONFIG -ini '\config\parameters' -add long tls_max_sessions 0

    $CPCONFIG -ini '\config\parameters' -add bool disableshortcuts true
    $CPCONFIG -ini '\config\parameters' -add string configencoding CP1251

    $CPCONFIG -ini '\config\provider' -add bool checkpublic true

    $CPCONFIG -ini '\config\parameters\srvthreadexception' -add long
disablecatchsignal 0
    $CPCONFIG -ini '\config\parameters\srvthreadexception' -add long maxexception
77
    $CPCONFIG -ini '\config\parameters\srvthreadexception' -add long sigstacksize
16384
    $CPCONFIG -ini '\config\parameters\srvthreadexception' -add long
prefetchthread 16
    $CPCONFIG -ini '\config\parameters\srvthreadexception' -add long
prefetchtimeout 10
```

```
    $CPCONFIG -loglevel cpcsp -mask $log_mask
    $CPCONFIG -loglevel cpcsp -format $log_format
    $CPCONFIG -loglevel capi10 -mask $log_mask
    $CPCONFIG -loglevel capi10 -format $log_format
    $CPCONFIG -loglevel cprdr -mask $log_mask
    $CPCONFIG -loglevel cprdr -format $log_format
    $CPCONFIG -loglevel cpext -mask $log_mask
    $CPCONFIG -loglevel cpext -format $log_format
    $CPCONFIG -loglevel capi20 -mask $log_mask
    $CPCONFIG -loglevel capi20 -format $log_format
    $CPCONFIG -loglevel libcspr -mask $log_mask
    $CPCONFIG -loglevel libcspr -format $log_format
    $CPCONFIG -loglevel cryptsrv -mask $log_mask
    $CPCONFIG -loglevel cryptsrv -format $log_format
    $CPCONFIG -loglevel kchansrv -mask $log_mask
    $CPCONFIG -loglevel kchansrv -format $log_format
    $CPCONFIG -loglevel fenixmsrv -mask $log_mask
    $CPCONFIG -loglevel fenixmsrv -format $log_format
    $CPCONFIG -loglevel libssp -mask $log_mask
    $CPCONFIG -loglevel libssp -format $log_format
    $CPCONFIG -loglevel cppkcs11 -mask $log_mask
    $CPCONFIG -loglevel cppkcs11 -format $log_format
    $CPCONFIG -loglevel cpdrv -mask $log_mask
    $CPCONFIG -loglevel cpdrv -format $log_format
    $CPCONFIG -loglevel dmntcs -mask $log_mask
    $CPCONFIG -loglevel dmntcs -format $log_format
    $CPCONFIG -loglevel ocsp -mask $log_mask
    $CPCONFIG -loglevel ocsp -format $log_format
    $CPCONFIG -loglevel tsp -mask $log_mask
    $CPCONFIG -loglevel tsp -format $log_format
    $CPCONFIG -loglevel cades -mask $log_mask
    $CPCONFIG -loglevel cades -format $log_format
    $CPCONFIG -loglevel pkivalidator -mask $log_mask
    $CPCONFIG -loglevel pkivalidator -format $log_format
    $CPCONFIG -loglevel pcsc -mask $log_mask
    $CPCONFIG -loglevel pcsc -format $log_format
    $CPCONFIG -loglevel hsmdb -mask $log_mask
    $CPCONFIG -loglevel hsmdb -format $log_format

    $CPCONFIG -ini '\config\oid\1.3.14.3.2.26!1' -add string name sha1
    $CPCONFIG -ini '\config\oid\1.3.14.3.2.26!1' -add long algid 32772

    $CPCONFIG -ini '\config\oid\1.3.14.3.2.18!1' -add string name sha
    $CPCONFIG -ini '\config\oid\1.3.14.3.2.18!1' -add long algid 32772

    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.1.5!4' -add string name sha1/RSA
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.1.5!4' -add long algid 32772
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.1.5!4' -add hex extrainfo
0024000000000000001000000

    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.1.1!3' -add string name RSA
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.1.1!3' -add long algid 9216
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.1.1!3' -add hex extrainfo
00000000

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.19!3' -add string name "ГОСТ Р 34.10-
2001"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.19!3' -add long algid 11811
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.19!3' -add hex extrainfo 00000000

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.21!2' -add string name "ГОСТ 28147-89"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.21!2' -add long algid 26142
```

```
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.3!4' -add string name "ГОСТ Р
34.11/34.10-2001"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.3!4' -add long algid 32798
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.3!4' -add hex extrainfo
232e0000040000004b000000

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.30.1!20' -add string name "ГОСТ Р
34.11-94, параметры по умолчанию"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.30.2!20' -add string name "ГОСТ Р
34.11-94, параметры хеша 1"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.30.3!20' -add string name "ГОСТ Р
34.11-94, параметры хеша 2"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.30.4!20' -add string name "ГОСТ Р
34.11-94, параметры хеша 3"

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.1!20' -add string name  "ГОСТ
28147-89, параметры по умолчанию"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.2!20' -add string name  "ГОСТ
28147-89, параметры шифрования 1"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.3!20' -add string name  "ГОСТ
28147-89, параметры шифрования 2"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.4!20' -add string name  "ГОСТ
28147-89, параметры шифрования 3"

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.3!20' -add string name "ГОСТ 28147-
89, параметры Оскар 1.1"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.6!20' -add string name "ГОСТ 28147-
89, параметры Оскар 1.0"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.7!20' -add string name "ГОСТ 28147-
89, параметры РИК 1"

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.12!20' -add string name "ГОСТ
28147-89, параметры шифрования ТС26 1"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.13!20' -add string name "ГОСТ
28147-89, параметры шифрования ТС26 2"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.14!20' -add string name "ГОСТ
28147-89, параметры шифрования ТС26 3"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.15!20' -add string name "ГОСТ
28147-89, параметры шифрования ТС26 4"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.16!20' -add string name "ГОСТ
28147-89, параметры шифрования ТС26 5"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.31.17!20' -add string name "ГОСТ
28147-89, параметры шифрования ТС26 6"

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.34.1!7' -add string name "Аудит TLS
трафика"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.34.2!7' -add string name
"Идентификация пользователя на Центре Регистрации"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.34.3!7' -add string name "Подпись
содержимого сервера Интернет"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.34.4!7' -add string name
"Администратор Центра Регистрации"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.34.5!7' -add string name "Оператор
Центра Регистрации"

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.35.1!20' -add string name "ГОСТ Р
34.10-2001, параметры по умолчанию"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.35.2!20' -add string name "ГОСТ Р
34.10-2001, параметры Оскар 2.x"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.35.3!20' -add string name "ГОСТ Р
34.10-2001, параметры подписи 1"

    $CPCONFIG -ini '\config\oid\1.2.643.2.2.36.0!20' -add string name "ГОСТ Р
34.10-2001, параметры обмена по умолчанию"
```

```
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.36.1!20' -add string name "ГОСТ Р
34.10-2001, параметры обмена 1"

   $CPCONFIG -ini '\config\oid\1.2.643.2.2.9!1' -add string name "ГОСТ Р 34.11-94"
   $CPCONFIG -ini '\config\oid\1.2.643.2.2.9!1' -add long algid 32798

   $CPCONFIG -ini '\config\oid\1.2.643.2.2.98!3' -add string name "ГОСТ Р 34.10-
2001 DH"
    $CPCONFIG -ini '\config\oid\1.2.643.2.2.98!3' -add long algid 43556

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.1.1!3' -add string name "ГОСТ Р
34.10-2012"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.1.1!3' -add long algid 11849

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.1.2!3' -add string name "ГОСТ Р
34.10-2012"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.1.2!3' -add long algid 11837

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.2.2!1' -add string name "ГОСТ Р
34.10-2012 256 bit"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.2.2!1' -add long algid 32801

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.2.3!1' -add string name "ГОСТ Р
34.10-2012 512 bit"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.2.3!1' -add long algid 32802

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.3.2!4' -add string name "ГОСТ Р
34.11-2012/34.10-2012 256 bit"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.3.2!4' -add long algid 32801
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.3.2!4' -add hex extrainfo
492e00000400000050000000

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.3.3!4' -add string name "ГОСТ Р
34.11-2012/34.10-2012 512 bit"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.3.3!4' -add long algid 32802
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.3.3!4' -add hex extrainfo
3d2e00000400000051000000

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.6.1!3' -add string name "ГОСТ Р
34.10-2012 DH 256 bit"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.6.1!3' -add long algid 43590

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.6.2!3' -add string name "ГОСТ Р
34.10-2012 DH 512 bit"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.1.6.2!3' -add long algid 43586

    $CPCONFIG -ini '\config\oid\1.2.643.7.1.2.1.1.1!20' -add string name "ГОСТ Р
34.10-2012 256 bit, параметры ТС26 A"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.2.1.1.2!20' -add string name "ГОСТ Р
34.10-2012 256 bit, параметры ТС26 B"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.2.1.2.1!20' -add string name "ГОСТ Р
34.10-2012 512 bit, параметры по умолчанию"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.2.1.2.2!20' -add string name "ГОСТ Р
34.10-2012 512 bit, параметры ТС25 B"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.2.1.2.3!20' -add string name "ГОСТ Р
34.10-2012 512 bit, параметры ТС25 C"
    $CPCONFIG -ini '\config\oid\1.2.643.7.1.2.1.2.4!20' -add string name "ГОСТ Р
34.10-2012 512 bit, параметры ТС25 D"

    $CPCONFIG -ini '\config\oid\2.5.4.3!5' -add string name CN
    $CPCONFIG -ini '\config\oid\2.5.4.7!5' -add string name L
    $CPCONFIG -ini '\config\oid\2.5.4.10!5' -add string name O
    $CPCONFIG -ini '\config\oid\2.5.4.11!5' -add string name OU
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.9.1!5' -add string name E
```

```
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.9.1!5' -add hex extrainfo
07000000

    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.9.1!5!a' -add string name Email
    $CPCONFIG -ini '\config\oid\1.2.840.113549.1.9.1!5!a' -add hex extrainfo
07000000

    $CPCONFIG -ini '\config\oid\2.5.4.6!5' -add string name C
    $CPCONFIG -ini '\config\oid\2.5.4.6!5' -add hex extrainfo 04000000

    $CPCONFIG -ini '\config\oid\2.5.4.8!5' -add string name S
    $CPCONFIG -ini '\config\oid\2.5.4.8!5!a' -add string name ST
    $CPCONFIG -ini '\config\oid\2.5.4.9!5' -add string name STREET
    $CPCONFIG -ini '\config\oid\2.5.4.12!5' -add string name T
    $CPCONFIG -ini '\config\oid\2.5.4.12!5!a' -add string name Title
    $CPCONFIG -ini '\config\oid\2.5.4.42!5' -add string name G
    $CPCONFIG -ini '\config\oid\2.5.4.42!5!a' -add string name GN
    $CPCONFIG -ini '\config\oid\2.5.4.42!5!b' -add string name GivenName
    $CPCONFIG -ini '\config\oid\2.5.4.43!5' -add string name I
    $CPCONFIG -ini '\config\oid\2.5.4.43!5!a' -add string name Initials
    $CPCONFIG -ini '\config\oid\2.5.4.4!5' -add string name  SN
    $CPCONFIG -ini '\config\oid\0.9.2342.19200300.100.1.25!5' -add string name DC
    $CPCONFIG -ini '\config\oid\0.9.2342.19200300.100.1.25!5' -add hex extrainfo
07000000

    $CPCONFIG -ini '\config\oid\2.5.4.13!5' -add string name Decription
    $CPCONFIG -ini '\config\oid\2.5.4.17!5' -add string name PostalCode
    $CPCONFIG -ini '\config\oid\2.5.4.18!5' -add string name POBOX
    $CPCONFIG -ini '\config\oid\2.5.4.20!5' -add string name Phone
    $CPCONFIG -ini '\config\oid\2.5.4.20!5' -add hex extrainfo 04000000

    $CPCONFIG -ini '\config\oid\2.5.4.24!5' -add string name  X21Address
    $CPCONFIG -ini '\config\oid\2.5.4.24!5' -add hex extrainfo 03000000

    $CPCONFIG -ini '\config\oid\1.3.6.1.5.5.7.3.2!7' -add string name "Client
authentication"
}

make_HSM_ini()
{
      change_hostname
      CPCONFIG=/opt/cprocsp/sbin/amd64/cpconfig
      $CPCONFIG -ini '\hsm' -add string HSMID $HOSTNAME
      $CPCONFIG -ini '\hsm' -add string CardReader "$CARD_READER"
      $CPCONFIG -ini '\hsm' -add long ProviderType 80
      $CPCONFIG -ini '\hsm' -add string Provider "Crypto-Pro GOST R 34.10-2012
Cryptographic Service Provider"
      $CPCONFIG -ini '\hsm' -add long CardProviderType 80
      $CPCONFIG -ini '\hsm' -add string CardProvider "Crypto-Pro GOST R 34.10-2012
Cryptographic Service Provider"
      $CPCONFIG -ini '\hsm' -add string SignatureAlgorithm "1.2.643.7.1.1.3.2"
      $CPCONFIG -ini '\hsm' -add string Reader "HDIMAGE"
      $CPCONFIG -ini '\hsm' -add string EncryptedReader "HSMDB"
      $CPCONFIG -ini '\hsm' -add long EnableWeb 0
      $CPCONFIG -ini '\hsm' -add long EnableEncryption 1
      $CPCONFIG -ini '\hsm' -add long EnableUpload 0
      $CPCONFIG -ini '\hsm' -add long EnableSecondNetInterface 1
      $CPCONFIG -ini '\hsm' -add long EnableGamma 1
      $CPCONFIG -ini '\hsm' -add long GammaRequired 1
      $CPCONFIG -ini '\hsm' -add long AutocleanAuditLog 1
      $CPCONFIG -ini '\hsm' -add long AuditLogMaxRecCount 500000
      $CPCONFIG -ini '\hsm' -add long AutocleanPollInterval 600
        $CPCONFIG -ini '\hsm' -add string LoggedEvents
"33333333333333333333300000330003333333"
```

```
        $CPCONFIG -ini '\hsm' -add long ClearPortion 100000
        $CPCONFIG -ini '\hsm' -add long CheckIntrusion 0
        $CPCONFIG -ini '\hsm' -add long BackupMySQLOnHalt 1
        $CPCONFIG -ini '\hsm' -add string AuditSynchronous "OFF"
      $CPCONFIG -ini '\hsm' -add long ChangeEncryptionKeyTransaction 0
      $CPCONFIG -ini '\hsm' -add long EnableK2 0
      $CPCONFIG -ini '\hsm' -add long EnableK2s 0
      $CPCONFIG -ini '\hsm' -add long EnableEventLog 0

      #save original HSMID not in database, that may be cloned
      cat <<-END>/etc/opt/cprocsp/hsm.ini
      HSMID="$HOSTNAME"
      END

      cat <<-END>/etc/sysconfig/network
      NETWORKING=yes
      HOSTNAME=$HOSTNAME
      DOMAINNAME=$HOSTNAME
      RESOLV_MODS=no
      END

      cat <<-END>/etc/sysconfig/rpcbind
      #RPCBIND_ARGS=
      CONTROL_ARGS="-l"
      END
}

MYSQLDIR=/usr/local
MYSQLBASE=${MYSQLDIR}/mysql
MYSQLDATA=/var/lib/mysql/cprocsp/db
MYSQLETC=/var/lib/mysql/cprocsp/etc
MYSQLINITIALBACKUP=/etc/opt/cprocsp/hsmdb_dump.sql
config=".my.cnf.$$"
command=".mysql.$$"
mysql_client=""
rootpass=""
echo_n=
echo_c=

set_echo_compat() {
    case `echo "testing\c"`,`echo -n testing` in
      *c*,-n*) echo_n=   echo_c=     ;;
      *c*,*)   echo_n=-n echo_c=     ;;
      *)       echo_n=   echo_c='\c' ;;
    esac
}

prepare() {
    touch $config $command
    chmod 600 $config $command
}

find_mysql_client()
{
  for n in ./bin/mysql mysql
  do
    $n --no-defaults --help > /dev/null 2>&1
    status=$?
    if test $status -eq 0
    then
      mysql_client=$n
      return
    fi
  done
```

```
  echo "Can't find a 'mysql' client in PATH or ./bin"
  exit 1
}

do_query() {
    echo "$1" >$command
    #sed 's,^,> ,' < $command  # Debugging
    $mysql_client --defaults-file=$config <$command
    return $?
}

# Simple escape mechanism (\-escape any ' and \), suitable for two contexts:
# - single-quoted SQL strings
# - single-quoted option values on the right hand side of = in my.cnf
#
# These two contexts don't handle escapes identically.  SQL strings allow
# quoting any character (\C => C, for any C), but my.cnf parsing allows
# quoting only \, ' or ".  For example, password='a\b' quotes a 3-character
# string in my.cnf, but a 2-character string in SQL.
#
# This simple escape works correctly in both places.
basic_single_escape () {
    # The quoting on this sed command is a bit complex.  Single-quoted strings
    # don't allow *any* escape mechanism, so they cannot contain a single
    # quote.  The string sed gets (as argv[1]) is:  s/\(['\]\)/\\\1/g
    #
    # Inside a character class, \ and ' are not special, so the ['\] character
    # class is balanced and contains two characters.
    echo "$1" | sed 's/\(['"'"'\]\)/\\\1/g'
}

make_config() {
    echo "# mysql_secure_installation config file" >$config
    echo "[mysql]" >>$config
    echo "user=root" >>$config
    esc_pass=`basic_single_escape "$rootpass"`
    echo "password='$esc_pass'" >>$config
    #sed 's,^,> ,' < $config  # Debugging
}

get_root_password() {
    status=1
    while [ $status -eq 1 ]; do
      stty -echo
      echo $echo_n "Enter current password for root (enter for none): $echo_c"
      read password
      echo
      stty echo
      if [ "x$password" = "x" ]; then
          hadpass=0
      else
          hadpass=1
      fi
      rootpass=$password
      make_config
      do_query ""
      status=$?
    done
    echo "OK, successfully used password, moving on..."
    echo
}

set_root_password() {
    stty -echo
```

```
    echo $echo_n "New password: $echo_c"
    read password1
    echo
    echo $echo_n "Re-enter new password: $echo_c"
    read password2
    echo
    stty echo

    if [ "$password1" != "$password2" ]; then
      echo "Sorry, passwords do not match."
      echo
      return 1
    fi

    if [ "$password1" = "" ]; then
      echo "Sorry, you can't use an empty password here."
      echo
      return 1
    fi

    esc_pass=`basic_single_escape "$password1"`
    do_query "UPDATE mysql.user SET Password=PASSWORD('$esc_pass') WHERE
User='root';"
    if [ $? -eq 0 ]; then
      echo "Password updated successfully!"
      echo "Reloading privilege tables.."
      reload_privilege_tables
      if [ $? -eq 1 ]; then
            clean_and_exit
      fi
      echo
      rootpass=$password1
      make_config
    else
      echo "Password update failed!"
      clean_and_exit
    fi

    return 0
}

remove_anonymous_users() {
    echo " - Removing anonymous user prvileges..."
    do_query "DELETE FROM mysql.user WHERE User='';"
    if [ $? -eq 0 ]; then
      echo " ... Success!"
    else
      echo " ... Failed!"
      clean_and_exit
    fi

    return 0
}

remove_remote_root() {
    echo " - Removing remote login for root..."
    do_query "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN
('localhost', '127.0.0.1', '::1');"
    if [ $? -eq 0 ]; then
      echo " ... Success!"
    else
      echo " ... Failed!"
    fi
}
```

```
remove_test_database() {
    echo " - Dropping test database..."
    do_query "DROP DATABASE test;"
    if [ $? -eq 0 ]; then
      echo " ... Success!"
    else
      echo " ... Failed!  Not critical, keep moving..."
    fi

    echo " - Removing privileges on test database..."
    do_query "DELETE FROM mysql.db WHERE Db='test' OR Db='test\\_%'"
    if [ $? -eq 0 ]; then
      echo " ... Success!"
    else
      echo " ... Failed!  Not critical, keep moving..."
    fi

    return 0
}

reload_privilege_tables() {
    echo " - Reload privileges..."
    do_query "FLUSH PRIVILEGES;"
    if [ $? -eq 0 ]; then
      echo " ... Success!"
      return 0
    else
      echo " ... Failed!"
      return 1
    fi
}

interrupt() {
    echo
    echo "Aborting!"
    echo
    cleanup
    stty echo
    exit 1
}

cleanup() {
    echo "Cleaning up..."
    rm -f $config $command
}

# Remove the files before exiting.
clean_and_exit() {
      cleanup
      exit 1
}


mysql_secure_installation() {

  prepare
  find_mysql_client
  set_echo_compat

#get_root_password


#
```

```
# Set the root password
#

#    status=1
#    while [ $status -eq 1 ]; do
#     set_root_password
#     status=$?
#    done
  echo

#
# Remove anonymous users
#
    remove_anonymous_users
  echo
#
# Disallow remote root login
#
    remove_remote_root
  echo
#
# Remove test database
#
    remove_test_database
  echo
#
# Reload privilege tables
#
    reload_privilege_tables
  echo
  cleanup
}

install_mysql()
{
    # extract mysql to MYSQLBASE
    tar -xzf  /mnt/cdrom/mysql/mysql-5.5.29-linux-x86_64.tar.gz -C $MYSQLDIR
    mv ${MYSQLDIR}/mysql-5.5.29-linux-x86_64 $MYSQLBASE

    # add user mysql
    useradd mysql

    # create mysql DATA directory
    mkdir -p $MYSQLDATA
    chown mysql /var/lib/mysql
    chown mysql /var/lib/mysql/cprocsp
    chown mysql $MYSQLDATA

    # Copy mysql config
    mkdir -p $MYSQLETC
    chmod 644 $MYSQLETC
    cp /mnt/cdrom/mysql/my.cnf $MYSQLETC
    ln -sf ${MYSQLETC}/my.cnf /etc/my.cnf

    # link mysqlclient lib
    if ! grep -q '${MYSQLBASE}/lib' /etc/ld.so.conf;then
      echo "${MYSQLBASE}/lib" > /tmp/ld_tmp.$$
      cat /tmp/ld_tmp.$$ /etc/ld.so.conf >/tmp/ld.so.conf.$$
      rm /tmp/ld_tmp.$$
      mv /tmp/ld.so.conf.$$ /etc/ld.so.conf
    fi
    /sbin/ldconfig ${MYSQLBASE}/lib

    # install system database
```

```
    pushd  $MYSQLBASE
      ${MYSQLBASE}/scripts/mysql_install_db --basedir=$MYSQLDASE --
datadir=$MYSQLDATA --user=mysql
    popd

    # patch mysqld to store error messages to syslog (/var/log/messages)
    sed 's/^other_args="\$\*\"\(.*\)$/other_args="$* --syslog"\1/' <
${MYSQLBASE}/support-files/mysql.server > ${MYSQLBASE}/support-
files/mysql.server.tmp
    mv -f ${MYSQLBASE}/support-files/mysql.server.tmp ${MYSQLBASE}/support-
files/mysql.server
    sed 's/chkconfig: 2345 64 36\(.*\)$/chkconfig: 2345 07 92\1/' <
${MYSQLBASE}/support-files/mysql.server > ${MYSQLBASE}/support-
files/mysql.server.tmp
    mv -f ${MYSQLBASE}/support-files/mysql.server.tmp ${MYSQLBASE}/support-
files/mysql.server
    chmod +x ${MYSQLBASE}/support-files/mysql.server
    ln -sf ${MYSQLBASE}/support-files/mysql.server /etc/init.d/mysqld

    # chkconfig mysqld
    chkconfig --add mysqld
    chkconfig --level 2345 mysqld on

    #start mysqld
    service mysqld start

    # create keys db
    ${MYSQLBASE}/bin/mysql < /opt/cprocsp/sbin/amd64/init_db.sql

    # create registry db
    ${MYSQLBASE}/bin/mysql < /opt/cprocsp/sbin/amd64/init_registry_db.sql

    # force security
    pushd ${MYSQLBASE}
      mysql_secure_installation
    popd
    # remove unused files
    rm -rf ${MYSQLBASE}/mysql-test
    rm -rf ${MYSQLBASE}/sql-bench
    rm -rf ${MYSQLBASE}/man
    rm -rf ${MYSQLBASE}/include
    rm -rf ${MYSQLBASE}/docs
    rm -rf ${MYSQLBASE}/data
    cleanup
if [ $IsDEBUG = true ]; then
    mkdir -p /root/bin
    cp -f /mnt/cdrom/mysql/scripts/* /root/bin/
fi
}


dump_initial_mysql_backup()
{
    mkdir -p /backup
    /opt/cprocsp/sbin/amd64/hsmdb_dump.sh $MYSQLINITIALBACKUP
}

patch_rpm()
{
    paxctl -cpermx /bin/rpm
    paxctl -cpermx /usr/lib/rpm/rpmk
    paxctl -cpermx /usr/lib/rpm/rpmi
    paxctl -cpermx /usr/lib/rpm/rpmu
    paxctl -cpermx /usr/lib/rpm/rpmd
```

```
    paxctl -cpermx /usr/lib/rpm/rpme
    paxctl -cpermx /usr/lib/rpm/rpmv
    paxctl -cpermx /usr/lib/rpm/rpmq
    paxctl -cpermx /usr/lib64/librpmbuild-4.0.4.1.so

}

patch_grub()
{
    # grub-install does not work under grsec kernel. Need to patch following
urilities:
    paxctl -cpermx /usr/sbin/grub-probe
    paxctl -cpermx /usr/sbin/grub-bios-setup
    paxctl -cpermx /usr/bin/grub-script-check
}

install_pkg()
{
      PKG_VER=5.0.*-4
      ARCH=hsm
      NOARCH=hsm
        service pcscd stop
        service auditd stop
      chkconfig  --level 2345 auditd off
        service smartd stop
      chkconfig  --level 2345 smartd off
        service alteratord stop
      chkconfig  --level 2345 alteratord off
        service anacron stop
      chkconfig  --level 2345 anacron off
        service ntpd stop
      chkconfig  --level 2345 ntpd off
        service plymouth stop
      chkconfig  --level 2345 plymouth off
        service postfix stop
      chkconfig  --level 2345 postfix off
      mkdir /var/lib/lock
      mkdir /var/lib/lock/subsys
      chkconfig  --level 2345 pcscd on
      rm -f /usr/lib64/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist
      cp -f /mnt/cdrom/conf/Info.plist /usr/lib64/pcsc/drivers/ifd-
ccid.bundle/Contents/
      chmod 644 /usr/lib64/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist
      service pcscd start

      echo "Installing compat-package ..."
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-compat-altlinux-64-1.0.0-1.noarch.rpm

      echo "Installing main packages ..."
      rpm -ivh /mnt/cdrom/RPMS/lsb-cprocsp-base-${PKG_VER}.${NOARCH}.rpm

      install_mysql

      rpm -ivh /mnt/cdrom/RPMS/lsb-cprocsp-rdr-64-${PKG_VER}.${ARCH}.rpm

      #after installing cpconfig we can use it to load config64 to registry
database
      make_config64_ini

      rpm -ivh /mnt/cdrom/RPMS/lsb-cprocsp-rdr-sobol-64-${PKG_VER}.${ARCH}.rpm
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-rdr-pcsc-64-${PKG_VER}.${ARCH}.rpm
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-hsm-db-64-${PKG_VER}.${ARCH}.rpm --nodeps
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-pam-hsm-64-${PKG_VER}.${ARCH}.rpm
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-fenixm-server-64-${PKG_VER}.${ARCH}.rpm
```

```
      rpm -ivh /mnt/cdrom/RPMS/lsb-cprocsp-capilite-64-${PKG_VER}.${ARCH}.rpm
      cp -f /etc/grsec/policy_hsm /etc/grsec/policy
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-stunnel-64-${PKG_VER}.${ARCH}.rpm
      rpm -ivh /mnt/cdrom/RPMS/cprocsp-hsm-webadmin-${PKG_VER}.${NOARCH}.rpm
      service cprocsp stop
      service pcscd restart
      /opt/cprocsp/sbin/amd64/cpconfig -ini '\config\apppath\libcurl.so' -delparam
      ln -s /usr/lib64/libgdbm.so.3 /usr/lib64/libgdbm.so.2
      /opt/cprocsp/sbin/amd64/cpconfig -ini '\config\apppath' -add string
libpcsclite.so /usr/lib64/libpcsclite.so
      mkdir -p /var/opt/cprocsp/users/999.999/stores
      touch /var/opt/cprocsp/users/999.999/local.ini
      touch /var/opt/cprocsp/users/999.999/stores/my.sto
      chown -R tlsusers /var/opt/cprocsp/users/999.999
      chmod -R 744 /var/opt/cprocsp/users/999.999

##    paxctl -cpermx /opt/cprocsp/sbin/amd64/stunnel_hsm
##    rm -f /opt/cprocsp/lib/hashes/cprocsp-stunnel-64

      read -p "Disable crontab for ECC checking (y/n)?"
      if [ x$REPLY = xy ];then
                crontab -l |fgrep -v 'ecc_check'| fgrep -v '# ' >
/tmp/crontab.tmp; crontab /tmp/crontab.tmp
      fi
      echo "Total CE: 0" > /tmp/ecc_report_simple.txt
      echo "Total UE: 0" >> /tmp/ecc_report_simple.txt


      echo "Initialising DB..."
      /usr/bin/sqlite3 /var/opt/cprocsp/log/log_db.dat < /mnt/cdrom/db/log_db.sql
      /usr/bin/sqlite3 /var/opt/cprocsp/log/log_db_SCHEMA.dat <
/mnt/cdrom/db/log_db.sql
      service cprocsp start
      echo "Setting the license..."
      /opt/cprocsp/sbin/amd64/cpconfig -license -set 4040L-L0000-02ACD-C6MXF-645ET

#       echo "Coping config files ..."
      echo "Make SerialNumberPool ..."
        make_SerialNumberPool

if [ $IsDEBUG = false ]; then
      awk '/^[^#].*:respawn:/ || /l7:7:wait:/ || /~~:S:wait/ || /^ca:/ { print "#"
$0; next} {print}' </etc/inittab >/etc/inittab.new
      mv /etc/inittab.new /etc/inittab
      rpm -e apt apt-repo apt-conf-branch libapt --nodeps
      rpm -e rpm --nodeps
else
      ## in DEBUG comment halting when integrity check error occurs
      sed 's/\(\/sbin\/halt\)/#\1/g' < /opt/cprocsp/sbin/amd64/lcdfunctions >
/tmp/lcdfunctions.new
      mv -f /tmp/lcdfunctions.new /opt/cprocsp/sbin/amd64/lcdfunctions
fi
}

configure_intrusion_detection()
{
    local RET;
    CASEOPENUTIL=/opt/cprocsp/bin/amd64/caseopenutil
    $CASEOPENUTIL
    status=$?
    if test $status -eq 0
    then
      echo "Chassis intrusion detection Supported!"
      read -p "Push and keep the caseopen sensor. Then press any Enter key"
```

```
      $CASEOPENUTIL
      read -p "Release the caseopen sensor and press Enter key"
      $CASEOPENUTIL
    else
      echo "ERROR!!! Failed to detect supported Chassis intrusion Chip!!!"
    fi
    CPCONFIG=/opt/cprocsp/sbin/amd64/cpconfig
      read -p "Enable Chassis Intrusion detection support? (y/n)?"
      if [ x$REPLY != xn ]
      then
            $CPCONFIG -ini '\hsm' -add long CheckIntrusion 1
            read -p "Close computer's case, then press Enter key"
            $CASEOPENUTIL reset
            $CASEOPENUTIL
      else
            $CPCONFIG -ini '\hsm' -add long CheckIntrusion 0
      fi
}

adjust_boot()
{
# P’ DEBUG PIPµCЪCЃPëPë PjPsP¶PµPj PSPµ PePsPSPIPµCЪC,PëC,CЊ PïP°CЪC,PëC†PëCЋ
/boot PI FAT
if [ $BOOTasVFAT = true ]; then
    tar -C /boot -czhpf /tmp/boot.tgz .
    # convert hardlinks -> to regular files
    mkdir -p /tmp/boot
    cd /tmp/boot
    tar -xpf ../boot.tgz
    for inode in `find . -type f -links +1|grep -v "^.$"|xargs ls -i|awk '{print
$1}'|sort -u`;do find . -inum $inode|(read first;read second;read third; rm -rf
$second;cp -pr $first $second; [ x$third = x ] || (rm -rf $third && cp -rf $first
$third));done;
    cd ..
    rm -f boot.tgz
    tar -C /tmp/boot -czhpf /tmp/boot.tgz .
    rm -rf /tmp/boot

    umount /boot
    mkfs.vfat $BOOTD
    mount -t vfat $BOOTD /boot
    tar -C /boot -xzhpf /tmp/boot.tgz
    rm -f /tmp/boot.tgz
    sfdisk --change-id $DISK 1 e
#    sed 's/\(prompt\)/#\1/' </etc/lilo.conf >/tmp/lilo.conf
#    mv /tmp/lilo.conf /etc/lilo.conf
#    /sbin/lilo
fi
    mv /opt/cprocsp/lib/hashes /boot
    ln -s /boot/hashes /opt/cprocsp/lib/hashes
    # all hashes from cryptopro packages will be included in files.chk, so
*cprocsp* hashes may be deleted
    rm -f /boot/hashes/*
    grub-install --recheck $DISK
    if [ $IsDEBUG = true ]; then
      echo After adjusting boot. Press any key to continue
      read
    fi

}

remove_unused()
{
    rm -f /opt/cprocsp/sbin/amd64/stunnel_fork
```

```
    rm -f /opt/cprocsp/sbin/amd64/stunnel_thread

#   remove unused utils .
    rm -f /opt/cprocsp/bin/amd64/inittst
    rm -f /opt/cprocsp/bin/amd64/certmgr
    rm -f /opt/cprocsp/bin/amd64/der2xer
    rm -f /opt/cprocsp/bin/amd64/cryptcp
    rm -f /opt/cprocsp/bin/amd64/provtestf
    rm -f /opt/cprocsp/bin/amd64/provtest
    rm -f /opt/cprocsp/sbin/amd64/mount_flash.sh
}
adjust_readonly()
{
# /etc/mtab for mount
        ln -sf /proc/mounts /etc/mtab
        mkdir -p /var/etc

      mv /etc/blkid.tab /var/etc
      mv /etc/blkid.tab.old /var/etc
      ln -sf /var/etc/blkid.tab /etc/blkid.tab
      ln -sf /var/etc/blkid.tab.old /etc/blkid.tab.old
      sed 's/PROMPT=yes/PROMPT=no/' </etc/sysconfig/init >/tmp/init
      mv /tmp/init /etc/sysconfig/init
# time
        mv /etc/adjtime /var/etc
        ln -sf /var/etc/adjtime /etc/adjtime

# CSP config files
        mv /etc/opt /var/etc/opt
      ln -sf /var/etc/opt /etc/opt

# 1. do not mount cdrom, floppies
# 2. mount /boot as vfat
# 3. mount / readonly
# 4. make /var/opt/cprocsp/tmp tmpfs
    if [ $BOOTasVFAT = true ]; then
      sed
's%^\(.*[[:blank:]]/[[:blank:]].*ext[23][[:blank:]]*\)defaults\(.*\)$%\1ro\2%;s%^\
(.*/boot.*\)$%'$BOOTD'  /boot vfat  ro,nodev,nosuid,noexec  1
      2%;s%\(.*/var[[:blank:]].*\)$%\1\ntmpfs\t\t/var/opt/cprocsp/tmp\ttmpfs\tnosu
id\t0\t0%;s%^\(/dev/fd.*\)%#\1%;s%^\(.*/media/cdrom.*\)$%#\1%' </etc/fstab
>/tmp/fstab
        mv /tmp/fstab /etc/fstab
    fi

# Do not remount / in rw mode when booting
      sed 's%\(action \"Remounting root filesystem in read/write mode:\"
\${[}]REMOUNT_ROOTFS_RW_COMMAND:-mount -n -o remount,rw /[}]\)%#\1%;s%\(\[ -L
/etc/mtab \] || >/etc/mtab\)%#\1%;s%\(rm -f /etc/mtab~ /etc/mtab~~\)%#\1%'
</etc/rc.d/rc.sysinit >/tmp/rc.sysinit
      echo >> /tmp/rc.sysinit
      echo /sbin/ifrename >> /tmp/rc.sysinit
      mv /tmp/rc.sysinit /etc/rc.d/rc.sysinit
      chmod 755 /etc/rc.d/rc.sysinit

# do not remove files that won't be created while readonly
      sed 's%\(rm -f /fastboot /fsckoptions /forcefsck /halt /poweroff\)%#\1%'
</etc/rc.d/scripts/cleanup >/tmp/cleanup
      mv /tmp/cleanup /etc/rc.d/scripts/cleanup
      chmod 755 /etc/rc.d/scripts/cleanup

      service pcscd start

        echo "Writing dependancies..."
```

```
        ldconfig
        depmod -a
}

sbl_checksums()
{
    if [ $BOOTasVFAT = true ]; then
        umount /boot
        mount -t msdos $BOOTD /boot
        mkdir /boot/sobol
        cat <<-'END'>/var/tmp/post-install/instchk
            #!/bin/bash
            PATH=$PATH:/var/tmp/post-install
            sblchk create -tdir c:\\sobol
            sblchk addsect -track 0 -sector 1 -head 0 -disk 128
            sblchk addsect -track 0 -sector 1 -head 1 -disk 128
      END

        find /boot -type f|sed "/^\/boot\/sobol/d;s/^\/boot/sblchk addfile -file
c:/;s/\//\\\\\\\\\/g" >>/var/tmp/post-install/instchk
        chmod +x /var/tmp/post-install/instchk
        pushd /boot/sobol
        /var/tmp/post-install/instchk
        popd
        umount /boot
        mount -t vfat $BOOTD /boot
      if [ $IsDEBUG = true ]; then
          echo After sbl_checksums. Press any key to continue
          read
        fi
    fi
}

csp_checksums()
{
        local VERIFY=/opt/cprocsp/bin/amd64/cpverify
        local LIST=/var/tmp/post-install/files
        local CHK=/boot/hashes/files.chk
        local file
        cat $LIST|\
        (
                while read file
                do
                        [ ! -f $file ] && echo "File $file does not exist" &&
continue
                        echo -n $file >>$CHK
                        echo -en "\033[;1H$file\033[0K"
                  echo -e "\t$($VERIFY -mk $file)" >>$CHK
            done
    )
}

debug_passwd()
{
    if [ $IsDEBUG = true ]; then
      gradm -P
      gradm -P admin
      gradm -P shutdown
      cp -f /mnt/cdrom/conf/rc.local /var/tmp/post-install/
    fi
}

random_passwd()
```

```
{
    if [ $IsDEBUG = false ]; then
      local GP=/var/tmp/post-install/good-passwd
      local HASH
      echo "Random user1 password..."
      $GP|passwd user1
      echo "Random root password..."
      $GP|passwd root
      echo "Random gradm password..."
      $GP|gradm -P
      $GP|gradm -P admin
      $GP|gradm -P shutdown
      cp -f /mnt/cdrom/conf/rc.local /etc/rc.d/
    fi
}

find_working()
{
            local i
            echo "Remove all cables and press Enter..">&2
            read
            echo "Insert cable into $1 and press Enter..." >&2
            read
            for ((i=0;i<10;i++))
            do
                HW=`ifconfig -a|awk '/^[^ \t]/ {hw=$5}  /UP.*RUNNING/ {print
tolower(hw);exit}'|sed -n 's/^\([:0-9a-f][:0-9a-f]*\)$/\1/p'`
                if test -n "$HW"
                then
                  echo $HW
                  break
                else
                  echo  "Retrying..." >&2
                  sleep 5
                fi
            done
}
make_iftab()
{
            rm /etc/net/iftab
            rm /etc/iftab
            service network stop
            ifrename -p
            service network start
            for ((i=0;i<5;i++))
            do
                        ifconfig eth$i 192.168.$[26+$i].2 netmask 255.255.255.0
            done
            for int in $*
            do
                        mac=`find_working $int`
                        if [ -z "$mac" ]
                        then
                                    echo "cannot make iftab..."
                                    exit 1
                        fi
                        echo "$int  mac $mac" >>/etc/iftab
                        echo "$int  mac $mac" >>/etc/net/iftab
            done
            for ((i=0;i<5;i++))
            do
                        ifdown eth$i
            done
            service network stop
```

```
            ifrename -p
            service network start
}

network_config()
{
    #turn on arp filtering
    echo "net.ipv4.conf.all.arp_filter = 1" >> /etc/net/sysctl.conf
      CPCONFIG=/opt/cprocsp/sbin/amd64/cpconfig
      read -p "Use 3 network interfaces (y/n)?"
      if [ x$REPLY == xy ]
      then
            [ $CONFIGNETWORK = true ] && make_iftab eth0 eth1 eth2
            $CPCONFIG -ini '\network' -add long count 3
      else
            [ $CONFIGNETWORK = true ] && make_iftab eth0 eth1
            $CPCONFIG -ini '\network' -add long count 2
      fi

      $CPCONFIG -ini '\network' -add string gateway "192.168.26.1"
      $CPCONFIG -ini '\network\eth0' -add string ip_address "192.168.26.2"
      $CPCONFIG -ini '\network\eth0' -add string net_mask "255.255.255.0"
      $CPCONFIG -ini '\network\eth0' -add string media "autosense"
      $CPCONFIG -ini '\network\eth0' -add string gateway "192.168.26.1"

      $CPCONFIG -ini '\network\eth1' -add string ip_address "192.168.27.2"
      $CPCONFIG -ini '\network\eth1' -add string net_mask "255.255.255.0"
      $CPCONFIG -ini '\network\eth1' -add string media "autosense"
      $CPCONFIG -ini '\network\eth1' -add string gateway "192.168.27.1"

      if [ x$REPLY == xy ]
      then
          $CPCONFIG -ini '\network\eth2' -add string ip_address "192.168.28.2"
          $CPCONFIG -ini '\network\eth2' -add string net_mask "255.255.255.0"
          $CPCONFIG -ini '\network\eth2' -add string media "autosense"
          $CPCONFIG -ini '\network\eth2' -add string gateway "192.168.28.1"
      fi

        $CPCONFIG -ini '\firewall\443\192_168_27_1__255_255_255_255__eth0' -add
string enable 1
        $CPCONFIG -ini '\firewall\1502\192_168_26_1__255_255_255_255__eth0' -add
string enable 1
        $CPCONFIG -ini '\firewall\1501\192_168_26_1__255_255_255_255__eth0' -add
string enable  1

        $CPCONFIG -ini '\replication' -add long server_id 1
        $CPCONFIG -ini '\replication' -add long server_type 0
        $CPCONFIG -ini '\replication' -add string master_host "000.000.000.000"

    if [ $CONFIGNETWORK = true ]; then
      mkdir -p /etc/net/ifaces/eth0
      mkdir -p /etc/net/ifaces/eth1
      echo '`/opt/cprocsp/sbin/amd64/read_parm eth0 | grep -v Error`'
>/etc/net/ifaces/eth0/ipv4address
      echo '`/opt/cprocsp/sbin/amd64/read_parm eth1 | grep -v Error`'
>/etc/net/ifaces/eth1/ipv4address
      echo 'default via `/opt/cprocsp/sbin/amd64/cpconfig -ini "/network/gateway"
-view | grep -v Error`' >/etc/net/ifaces/eth0/ipv4route
      echo 'default via `/opt/cprocsp/sbin/amd64/cpconfig -ini "/network/gateway"
-view | grep -v Error`' >/etc/net/ifaces/eth1/ipv4route
      if [ x$REPLY == xy ]
      then
            mkdir -p /etc/net/ifaces/eth2
```

```
            echo '`/opt/cprocsp/sbin/amd64/read_parm eth2 | grep -v Error`'
>/etc/net/ifaces/eth2/ipv4address
            echo 'default via `/opt/cprocsp/sbin/amd64/cpconfig -ini
"/network/gateway" -view | grep -v Error`' >/etc/net/ifaces/eth2/ipv4route
#           cp /etc/net/ifaces/eth0/options /etc/net/ifaces/eth2/options
      fi
    fi
      cat <<-END>/etc/sysconfig/network
      # When set to no, this may cause most daemons' initscripts skip starting.
      NETWORKING=yes

      # Used by hotplug/pcmcia/ifplugd scripts to detect current network config
      # subsystem.
      CONFMETHOD=etcnet

      # Used by rc.sysinit to setup system hostname at boot.
      HOSTNAME=`/opt/cprocsp/sbin/amd64/cpconfig -ini "/hsm/HSMID" -view | grep -v
Error`
      DOMAINNAME=HSM

      # This is used by ALTLinux ppp-common to decide if we want to install
      # nameserver lines into /etc/resolv.conf or not.
      RESOLV_MODS=no
      END
      cat /etc/init.d/iptables |sed
's%IPTABLES_FILTER=.*$%IPTABLES_FILTER=/opt/cprocsp/sbin/amd64/genfw%'
>/tmp/iptables.new
      mv /tmp/iptables.new /etc/init.d/iptables
      chmod 755 /etc/init.d/iptables
}


cron_config()
{
      cat <<-END>>/etc/crontab
2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58
* * * * root /usr/sbin/logrotate /etc/logrotate.conf
      END
      rm -f /etc/cron.d/osec
      rm -f /etc/cron.d/mdadm
      rm -f /etc/cron.d/sysstat
      rm -f /etc/cron.daily/logrotate
      rm -f /etc/cron.daily/000anacron
      rm -f /etc/cron.daily/etckeeper
      rm -f /etc/cron.daily/postfix
      rm -f /etc/cron.daily/smtpclean

      rm -f /etc/cron.monthly/000anacron
      rm -f /etc/cron.weekly/000anacron
      rm -f /etc/cron.weekly/auditd
}

clock_config()
{
      cat <<-END>/etc/sysconfig/clock
HWCLOCK_SET_TIME_AT_START=true
HWCLOCK_SET_AT_HALT=false
HWCLOCK_ADJUST=false
UTC=true
ZONE=UTC
      END
      rm -f /etc/localtime
      cp -f /mnt/cdrom/conf/UTC /etc/localtime
}
```

```
select_reader()
{
	local R1="Gemplus GemPC433 SL 00 00"
	local R2="Gemplus GemPC Twin 00 00"
	local R3="Gemalto GemPC Twin 00 00"
	local R4="OmniKey CardMan 3121 00 00"
	[ $IsDEBUG = true ] && local R5="HDIMAGE"
	local i REPLY R RF AD

	service pcscd restart
	sleep 3
	RF=`/opt/cprocsp/bin/amd64/list_pcsc|sed 's/available reader: //'`

	if [ $? != 0 ] ; then
	    service pcscd restart
	    sleep 3
	fi

	RF=`/opt/cprocsp/bin/amd64/list_pcsc|sed 's/available reader: //'`

	[ $? == 0 ] || RF=NONE

	for ((i=1;;i++))
	do
		eval R=\$R$i
		AD=
		[ -z "$R" ] && break
		[ "$R" == "$RF" ] && AD="(found)"

		eval echo \"$i\) \$R$i\ $AD\"
	done

	while true
	do
		read -p "Select reader:"
		let R=$REPLY
		if [ $R -ge 1 ] && [ $R -lt $i ]
		then
			eval R=\$R$R
			echo $R
			break
		fi
	done

	sleep 3
	/opt/cprocsp/sbin/amd64/cpconfig -hardware reader -del FLASH
	sleep 3
	/opt/cprocsp/sbin/amd64/cpconfig -hardware reader -add "$R"
	  CARD_READER=$R
}


LCD_INI=/etc/opt/cprocsp/lcdport.ini
select_lcd()
{
	local R1="Old LCD panel interface"
	local R2="SAPIC-E LCD panel interface"
	local R3="LCM16A2 LCD panel interface"
	local R4="LCM-2x16 LCD panel interface"
	local i REPLY R
	local mod_suffix=""
```

```
	for ((i=1;;i++))
	do
		eval R=\$R$i
		[ -z "$R" ] && break
		eval echo \"$i\) \$R$i\"
	done

	while true
	do
		read -p "Select LCD panel interface:"
		let R=$REPLY
		if [ $R -ge 1 ] && [ $R -lt $i ]
		then
			break
		fi
	done
	case $R in
	2) mod_suffix="se";;
	3) mod_suffix="lcm";;
	4) mod_suffix="lcm2x16";;
	esac
	/opt/cprocsp/sbin/amd64/cpconfig -ini '\config\apppath' -add string
libhsmlcddrv.so /opt/cprocsp/lib/amd64/libhsmlcddrv${mod_suffix}.so

}

select_lcd_port()
{
	local R1="COM port 1"
	local R2="COM port 2"
	local i REPLY R
	local mod_suffix=""


	for ((i=1;;i++))
	do
		eval R=\$R$i
		[ -z "$R" ] && break
		eval echo \"$i\) \$R$i\"
	done

	while true
	do
		read -p "Select port for LCD:"
		let R=$REPLY
		if [ $R -ge 1 ] && [ $R -lt $i ]
		then
			break
		fi
	done

	/opt/cprocsp/sbin/amd64/cpconfig -ini '\config\lcdport' -add long Number $R
	cat <<-END>>$LCD_INI
	Number = $R
	END
}

select_lcd_port_speed()
{
	local R1="600"
	local R2="1200"
	local AR2=" - LCD LCM16A2-002 panel"
	local R3="1800"
	local R4="2400"
```

```
	local AR4=" - old LCD panel"
	local R5="4800"
	local R6="9600"
	local AR6=" - LCD SAPIC-E panel and LCD LCM-2x16 panel"
	local i REPLY R item RES
	local mod_suffix=""


	for ((i=1;;i++))
	do
		eval R=\$R$i
		[ -z "$R" ] && break
		eval echo \"$i\) \$R$i\${AR$i}\"
	done

	while true
	do
		read -p "Select speed for LCD port:"
		let R=$REPLY
		if [ $R -ge 1 ] && [ $R -lt $i ]
		then
			break
		fi
	done

	eval RES=\$R$R
	for ((i=1;;i++))
	do
		eval item=\$R$i
		[ -z "${item}" ] && break
		if [ $item -eq $RES ]
		then
			/opt/cprocsp/sbin/amd64/cpconfig -ini '\config\lcdport' -add
long Speed ${item}
			cat <<-END>>$LCD_INI
			Speed = ${item}
			END
			break
		fi
	done
}

select_lcd_conf()
{

cat <<-END>$LCD_INI
[lcdport]
END
    echo "Selecting the LCD interface..."
    select_lcd
    echo "Selecting the port for LCD..."
    select_lcd_port
    echo "Selecting speed for lcd port..."
    select_lcd_port_speed
}

remove_std_def_kernel()
{
#check if security kernel is installed and loaded
    uname -a | grep sec-def
    if [ $? != 0 ];then
      echo Security kernel not installed!!!
    else
    # remove old STD-def kernel with its modules
```

```
#    remove std-def modules first
      rpm -qa | grep std-def | grep -v kernel-image | xargs rpm -e
#    then remove std-def kernel-image
      rpm -qa | grep std-def | xargs rpm -e
    fi
}
#-----------------------------------------------
echo "Patching rpm with pax ..."
patch_rpm

echo "Patching grub with pax ..."
patch_grub

echo "Removing std-def kernel ..."
remove_std_def_kernel

echo "Installing packages ..."
install_pkg
mkdir /var/tmp/post-install
cp -f /mnt/cdrom/post-install/good-passwd /var/tmp/post-install/
cp -f /mnt/cdrom/post-install/sblchk /var/tmp/post-install/
chmod +x /var/tmp/post-install/sblchk
cp -f /mnt/cdrom/conf/files /var/tmp/post-install
echo -n "probing disk: "
probe_hdsd
echo "device=$DISK, boot device=$BOOTD"

echo "Selecting the LCD configuration..."
select_lcd_conf

echo "Selecting the reader..."
select_reader

echo "Creating hsm.ini..."
make_HSM_ini

#echo "Stopping cprocsp and services"
#service cprocsp stop

echo "Configuring the network..."
network_config

echo "Initialising iptables..."
fw_init

echo "Copying DSRF..."
copy_key_material

echo "Reconfiguring syslog..."
service syslogd stop
mkdir /var/run/dev
ln -sf /var/run/dev /dev/log
rm -f /var/log/alert
rm -f /var/log/boot.log
rm -f /var/log/syslog/messages
rm -f /var/log/messages
rm -f /var/log/secure
rm -f /var/log/maillog
rm -f /var/log/spooler
cp -f /mnt/cdrom/conf/syslog.conf /etc/
cp -f /mnt/cdrom/conf/logrotate.conf /etc/
cat <<-END>/etc/sysconfig/syslogd
SYSLOGD_OPTIONS=
END
```

```
service syslogd start

echo "Configuring logrotate..."
cron_config

echo "Configuring clock..."
clock_config

echo "Making random passwords...."
random_passwd

#echo "Making debug passwords..."
#debug_passwd

echo "Preparing / partition to work in an ro mode..."
adjust_readonly

echo "Adjusting /boot"
adjust_boot

#mv /sbin/consoletype /sbin/consoletype.evil
#cp /tmp/post-install/consoletype /sbin

echo "Removing unused components..."
remove_unused

echo "Starting cprocsp and it's services..."
service cprocsp start

#echo "configure_intrusion_detection..."
configure_intrusion_detection

echo "Dumping initial mysql database..."
dump_initial_mysql_backup

echo "Making checksums for bin, lib, config files with cpverify"
csp_checksums
clear

echo "Making checksums for the \"Sobol\" lock"
sbl_checksums

echo "Removing installation files..."
rm -rf /var/tmp/post-install

echo "You may now remove the CD-ROM..."
cd /root
umount /mnt/cdrom


_ROOT_DEV_=$(df -k / 2>/dev/null | tail -1 | expand | sed -n 's#^\([^ ][^ ]*\) *.*
/$#\1#p')
if [ "${?}" = "0" -a "${_ROOT_DEV_}" != "" ] ; then
  mount -o remount,ro / ### "/" MUST be read-only mounted at this point!
  tune2fs -O ^has_journal "${_ROOT_DEV_}" && \
  echo "INFO: 'has_journal' flag was removed for (readonly) ROOTFS."

# !!! ATTENTION/WARNING !!!
# After 'tune2fs -O ^has_journal ... '
# 'mount -o remount,rw /' does not work!
# It's necessary to reboot OS to be able to remount root FS.
#
# It means, that all commands after 'tune2fs -O ^has_journal ...' and before
# halt/reboot/shutdown will NOT work properly if read-write access (to rootfs)
```

```
# is required!
#

  fsck -y -f "${_ROOT_DEV_}"
#  mount -o remount,rw / ### does NOT WORK!!! under AltLinux ### => we'll skip
remount-rw since 'halt' is "near" ###
else
  IsDEBUG=true
  echo "WARNING: can't find ROOTFS device. Try to remove 'has_journal' flag
mannualy."
fi


echo "THE END"

if [ $IsDEBUG = false ]; then
    echo "Hulting the system"
    exec halt -p
fi
```

4.3. Описание настроек безопасности и конфигурации CMOS BIOS

С учетом принятой модели нарушителя ИБ для нейтрализации деструктивных возможностей неоходимо обеспечить:

1. Установку перемычек на контакты 1-2 разъемов JCMOS1, JME1 системной платы.

2. Исключение подключений к разъемам RJ-45 (LAN1, LAN2), USB (USB0_1, USB4_5), выводимым на заднюю панель корпуса сервера.

3. Исключение доступа к разъемам USB10, LPC1, CN1, SPI_CN1, GPIO1 системной платы.

4. Отключение платы Alarm Board SAB-2000 от разъема SMBUS1 и от контактов SNMP разъема JFP1 системной платы и исключение подключения к ним.

5. Исключение подключений к используемым разъемам SATA системной платы и устранение данных подключений при наличии (по умолчанию используются разъемы SATA1, SATA3, sSATA3 — для штатных твердотельных накопителей, SATA0, SATA2, SATA4, sSATA0, sSATA1, sSATA2 — не используются).

6. До подключения АПМЗД должны быть выполнены следующие настройки параметров утилиты "Aptio Setup Utility":

- Выполнение команды "Optimized Defaults" (клавиша F3);
- Установка следующих значений параметров утилиты "Aptio Setup Utility":

Advanced\Serial Port Console Redirection\COM0\Console Redirection	—	Disabled
Advanced\Serial Port for Out-of-Band Management/Windows Emergency Management Services (EMS)\Console	—	Disabled

Redirection		
Advanced\CSM Configuration\CSM Support	—	Enabled
Advanced\Trusted Computing\Security Device Support	—	Disabled
IntelRCSetup\Processor Configuration\VMX	—	Disabled
IntelRCSetup\PCH Configuration\PCH sSATA Configuration\sSATA Controller	—	Enabled
IntelRCSetup\PCH Configuration\PCH sSA TA Configuration\Configure sSA TA as	—	AHCI
IntelRCSetup\PCH Configuration\PCH sSATA Configuration\sSATA Port 3\Port 3	—	Enabled
IntelRCSetup\PCH Configuration\Networking\LAN1 Controller	—	Disabled
IntelRCSetup\PCH Configuration\Networking\LAN2 Controller	—	Disabled
Security\Administrator Password	—	пароль длиной 20 символов
Boot\Setup Prompt Timeout	—	1
Boot\Boot Option #1	—	штатный твердотельный накопитель SSD sSATA P3: SQF-SDMS4-4G-J6C
Boot\Boot Option #(остальные)	—	Disabled (при наличии)
Boot\Hard Drive BBS Priorities\Boot Option #1	—	штатный твердотельный накопитель SSD sSATA P3: SQF-SDMS4-4G-J6C
Boot\Hard Drive BBS Priorities\Boot Option #(остальные)	—	Disabled (при наличии)
Boot\Network Device BBS Priorities\Boot Option #(все)	—	Disabled (при наличии)

- Выполнение команды "Exit\Save Changes and Exit" (клавиша F4);

7. Установку пароля "Administrator Password" с целью увеличения временного интервал, требующийся нарушителю для входа в утилиту "Aptio Setup Utility" и изменения её параметров, что позволяет установленному АПМДЗ блокировать возможность входа в утилиту. Сведения о пароле "Administrator Password" должны быть известны только администратору. При этом дополнительные требования к качеству пароля не предъявляются.

В случае выхода из строя батареи питания CMOS на системной плате, осуществляются замена батареи и повторное выполнение вышеуказанных действий в утилите "Aptio Setup Utility". Периодичность плановой замены батареи — один раз в 5 лет.

8. Установку АПМДЗ "ПАК "Соболь" в соответствии со специальными рекомендациями, приведенными в руководстве по администрированию, с учетом степени секретности обрабатываемых данных.

9. Исключение из состава штатной ОС, системного и прикладного ПО средств, реализующих:

- перепрограммирование микросхем с компонентами ПО BIOS;
- манипулирование значениями переменных в области энергонезависимого ОЗУ (NVRAM) BIOS системной платы;
- компрометацию сценария режима "восстановления после сна" (BootScript), содержащегося в таблицах ACPI в ОЗУ;
- восстановление полноценного функционирования контроллера Intel ME;
- взаимодействия с контроллером Intel ME.

10. Исключение возможности бесконтрольного изменения состава ПО сервера.

11. Разметку штатного твердотельного накопителя SSD, с которого осуществляется загрузка штатной ОС, в соответствии с форматом MBR.

12. Проверку отсутствия нештатных ОС на штатных твердотельных накопителях SSD, с которых не осуществляется загрузка штатной ОС, и исключение нештатных ОС при наличии.

13. Устранение возможности бесконтрольного вскрытия нарушителем корпуса системного блока с целью изменения состава аппаратных средств и/или ПО сервера.