Dell™ PowerVault™ Encryption Key Manager

Руководство пользователя

Dell™ PowerVault™ Encryption Key Manager

Руководство пользователя

Содержание

Рисунки

Таблицы

Предисловие

Об этой книге

Данное руководство содержит сведения и инструкции, необходимые для установки и использования Dell™ Encryption Key Manager. В нем рассмотрены понятия и процедуры, относящиеся к:

- накопителям на магнитной ленте LTO 4 и LTO 5 с возможностью шифрования
- криптографическим ключам;
- цифровым сертификатам.

Для кого предназначена эта книга

Эта книга предназначена для администраторов систем хранения данных и систем обеспечения безопасности, ответственных за защиту и резервное копирование важнейших данных, а также для лиц, участвующих в настройке и обслуживании серверов Encryption Key Manager в операционной среде. Предполагается, что читатель имеет практические знания в области устройств и сетей хранения данных.

Условные обозначения и термины, используемые в этой книге

В книге используются следующие типографские обозначения:

Таблица 1. Типографские обозначения, используемые в данной книге

Обозначение	Применение
полужирный шрифт	Слова или символы, выделенные **полужирным шрифтом**, обозначают системные элементы, которые необходимо использовать в их буквальном виде, например, имена команд, файлов, флагов, путей и выбранных пунктов меню.
`моноширинный шрифт`	Примеры, вводимый пользователем текст и информация, отображаемая на экране системой, выделены `моноширинным шрифтом`.
курсив	Выделенные *курсивом* слова или символы представляют собой значения переменных, которые вводятся пользователем.
[элемент]	Обозначает необязательные элементы.
{элемент}	Содержит список, из которого необходимо выбрать один элемент описания формата или синтаксиса.
|	Вертикальная черта разделяет элементы в списке возможных вариантов.
<Клавиша>	Обозначает клавиши, которые нажимает пользователь.

Предупреждающее примечание

Предупреждающие примечания обращают внимание пользователя на возможное повреждение программы, устройства, системы или данных. Предупреждающие примечания могут сопровождаться восклицательным знаком, но этот символ не является обязательным. Примеры предупреждающих примечаний приведены ниже.

Внимание: В случае выполнения данной операции с использованием винтоверта возможно разрушение магнитного слоя ленты.

Связанные публикации

Дополнительные сведения можно найти в следующих публикациях:

- *Начало работы с библиотеками магнитных лент Dell™ PowerVault™ TL2000 и TL4000*. В этом руководстве приведены сведения об установке.
- *Dell™ PowerVault™ TL2000 Tape Library and TL4000 Tape Library SCSI Reference*. Здесь рассмотрены поддерживаемые команды SCSI, а также протокол, управляющий интерфейсом SCSI.

Сведения о Linux

Сведения о Red Hat

По следующему URL-адресу представлена информация об операционных системах Linux® Red Hat:

- **http://www.redhat.com**

Сведения о SuSE

По следующему URL-адресу представлена информация об операционных системах Linux SuSE:

- **http://www.suse.com**

Сведения о Microsoft Windows

По следующему URL-адресу можно найти информацию об операционных системах Microsoft® Windows®:

- **http://www.microsoft.com**

Интерактивная поддержка

На Web-сайте **http://support.dell.com** можно ознакомиться со следующей публикацией:

Краткое руководство пользователя компонента Dell Encryption Key Manager. В данном руководстве приведены сведения о настройке базовой конфигурации.

Посетите Web-сайт **http://www.dell.com**, чтобы ознакомиться со следующей публикацией:

Документ *Library Managed Encryption for Tape (Шифрование, управляемое библиотекой, для ленточных устройств)* описывает лучшие практические методы работы в области шифрования данных на ленточных накопителях LTO.

Прочтите это прежде всего

Обращение в Dell

Заказчики в США могут позвонить по тел. 800-WWW-DELL (800-999-3355).

Примечание: Если у вас нет активного подключения к Интернету, контактную информацию можно найти в счете-фактуре, на упаковочном листе, в счете или в каталоге продукции Dell.

Dell предлагает на выбор несколько программ интерактивной и телефонной поддержки и обслуживания. Возможность воспользоваться ими зависит от страны и продукта. Некоторые услуги могут быть недоступны в вашем регионе. Чтобы связаться с Dell по вопросам сбыта, технической поддержки или обслуживания клиентов, выполните следующие действия.

1. Посетите Web-сайт **http://support.dell.com**.
2. Проверьте правильность выбора страны или региона в раскрывающемся списке **Выберите страну/регион** в нижней части страницы.
3. Щелкните по ссылке **Контакты** в левой части страницы.
4. Выберите интересующий вас вид поддержки или обслуживания.
5. Выберите наиболее удобный способ связи с представителями Dell.

Глава 1. Общие сведения о шифровании на магнитной ленте

В условиях жестокой рыночной конкуренции данные являются одним из наиболее ценных ресурсов. Защита данных, контроль доступа к ним, обеспечение их достоверности и доступности - первоочередные задачи в современном мире, обеспокоенном вопросами безопасности. Шифрование данных - это средство, удовлетворяющее многие из этих потребностей. Dell Encryption Key Manager (далее упоминаемый как Encryption Key Manager) упрощает задачи шифрования данных.

Накопители LTO 4 и LTO 5 способны шифровать данные во время их записи на любые кассеты LTO 4 и LTO 5. Эта новая возможность позволяет хранить данные с более высокой степенью защиты без повышения непроизводительных затрат и падения производительности, обычно связанных с шифрованием, выполняемым на сервере или за счет выделенного устройства.

Решение по шифрованию данных на магнитной ленте состоит из трех основных компонентов.

Накопитель на магнитной ленте с возможностью шифрования
Все ленточные накопители LTO 4 и LTO 5 необходимо активировать через интерфейс библиотеки.

Дополнительную информацию о накопителях на магнитной ленте см. в разделе “Требования к оборудованию и программному обеспечению” на стр. 2-2.

Управление ключами шифрования
Шифрование включает использование ключей нескольких типов на разных иерархических уровнях. Генерация, поддержка, контроль и передача этих ключей зависят от рабочей среды, в которой устанавливается накопитель на магнитной ленте с поддержкой шифрования. Некоторые приложения могут управлять ключами. Для сред, не имеющих подобного рода приложений, или сред, где желательно применять шифрование, не зависящее от приложений, Dell Encryption Key Manager выполняет все необходимые задачи по управлению ключами. Эти задачи более подробно описаны в разделе “Управление шифрованием” на стр. 1-3.

Политика шифрования
Это метод, используемый для реализации шифрования. В него входят правила, определяющие, какие именно тома подлежат шифрованию, и механизм выбора ключей. Способ и источник установления этих правил зависят от рабочей среды. Дополнительные сведения см. в разделе “Управление шифрованием” на стр. 1-3.

Компоненты

Encryption Key Manager является составной частью среды Java и использует для шифрования компоненты Java Security. (Дополнительные сведения по компонентам Java Security см. в других публикациях по данной теме.) Приложение Encryption Key Manager состоит из трех основных компонентов, которые используются для управления его поведением. Эти компоненты перечислены ниже.

Хранилище ключей Java Security
Хранилище ключей определяется как часть расширения Java Cryptography Extension (JCE) и элемент компонентов Java Security, которые, в свою очередь, являются частью среды выполнения Java. Хранилище ключей содержит

сертификаты и ключи (или указатели на сертификаты и ключи), используемые приложением Encryption Key Manager при выполнении операций шифрования. Поддерживаются хранилища ключей Java нескольких типов, с различными рабочими параметрами, отвечающими широкому ряду требований. Их характеристики подробно рассматриваются в разделе “Замечания о хранилище ключей” на стр. 2-4.

Важность сохранения данных хранилища ключей несомненна. Без доступа к хранилищу ключей расшифровать зашифрованные данные на магнитной ленте не удастся. Внимательно прочтите нижеприведенные разделы, чтобы ознакомиться с существующими методами защиты данных хранилища ключей.

Файлы конфигурации

Файлы конфигурации позволяют настраивать поведение Encryption Key Manager в соответствии с потребностями предприятия. Настройки поведения подробно описаны в данном документе: сначала в разделе Глава 2, “Планирование среды Encryption Key Manager”, на стр. 2-1, затем в разделе Глава 4, “Конфигурирование диспетчера ключей шифрования (Encryption Key Manager, EKM)”, на стр. 4-1, а также в приложении B, содержащем описание полного набора параметров настройки.

Таблица ленточных накопителей

Таблица ленточных накопителей используется приложением Encryption Key Manager для отслеживания поддерживаемых им ленточных устройств. Таблица ленточных накопителей представляет собой нередактируемый двоичный файл, расположение которого указывается в файле конфигурации. При необходимости место его расположения можно изменять.

Файл KeyGroups.xml

Данный защищенный паролем файл содержит имена всех групп ключей шифрования и псевдонимы ключей шифрования, связанных с каждой группой ключей.

Рисунок 1-1. Четыре основных компонента Encryption Key Manager

Управление шифрованием

Dell Encryption Key Manager - это программа на языке Java™, обеспечивающая накопителям на магнитной ленте с возможностью шифрования создание, защиту, хранение и обслуживание ключей шифрования, при помощи которых данные шифруются при записи на ленточные носители (в формате лент и кассет) и расшифровываются при чтении с этих носителей. Encryption Key Manager работает в операционных системах Linux (SLES и RHEL) и Windows. Программа предназначена для выполнения в фоновом режиме в качестве общего ресурса, внедренного в нескольких точках предприятия. Клиент с интерфейсом командной строки предоставляет требуемый набор команд для настройки приложения Encryption Key Manager в пользовательской среде и мониторинга его операций. Многие функции настройки и мониторинга доступны также с помощью графического интерфейса пользователя (GUI) Dell Encryption Key Manager. Encryption Key Manager использует одно или несколько хранилищ ключей для хранения ключей и сертификатов (или указателей на ключи и сертификаты), необходимых для выполнения всех задач шифрования. Подробные сведения см. в разделе “Замечания о хранилище ключей” на стр. 2-4.

ВАЖНАЯ ИНФОРМАЦИЯ О КОНФИГУРАЦИИ ХОСТ-СЕРВЕРА Encryption Key Manager: Для минимизации риска потери данных на компьютерах с установленными программами Dell Encryption Key Manager рекомендуется использовать память ECC. Encryption Key Manager формирует запросы на генерацию ключей шифрования и их передачу накопителям на магнитной ленте LTO 4 и LTO 5. Encryption Key Manager при обработке ключа хранит его данные в системной памяти в свернутом (зашифрованном) виде. Следует заметить, что данные ключа необходимо передать соответствующему накопителю на магнитной ленте без ошибок, чтобы записанные на кассету данные можно было восстановить (расшифровать). Если по какой-либо причине данные ключа были повреждены из-за ошибок в разрядах системной памяти, но использовались для записи данных на кассету, то записанные на эту кассету данные будут недоступны для восстановления (последующей расшифровки). Существуют различные средства защиты, предотвращающие появление подобных ошибок данных. Однако если компьютер, на котором установлено приложение Encryption Key Manager, не использует память с коррекцией ошибок Error Correction Code (ECC), существует вероятность повреждения хранимых в памяти данных и, в результате, их потери. Вероятность появления таких ошибок достаточно мала, однако на компьютерах, на которых установлены критически важные приложения (например, Encryption Key Manager), рекомендуется всегда использовать память ECC.

Encryption Key Manager работает в фоновом режиме, ожидая запросы на генерацию или получение ключа, отправленные по каналу TCP/IP, связывающему его с ленточной библиотекой. Когда накопитель на магнитной ленте записывает зашифрованные данные, он сначала запрашивает ключ шифрования у Encryption Key Manager. Получив запрос, Encryption Key Manager выполняет следующие задачи.

Encryption Key Manager выбирает в хранилище ключей существующий ключ AES и свертывает его для безопасной пересылки на накопитель на магнитной ленте, где он развертывается и используется для шифрования данных, записываемых на ленту.

Когда ленточный накопитель LTO 4 или LTO 5 считывает зашифрованные данные с ленты, Encryption Key Manager находит в хранилище ключей нужный ключ с учетом информации, содержащейся в идентификаторе ключа на ленте, и посылает его накопителю на магнитной ленте в свернутом для безопасной пересылки виде.

Существует два метода управления шифрованием. Они различаются по местонахождению механизма политики шифрования, по устройству, осуществляющему управление ключами для решения по шифрованию данных, и по способу связи приложения Encryption Key Manager с накопителем. Следует выбирать метод, наиболее оптимальный для конкретной рабочей среды. Управление ключами и механизм политики шифрования могут выполняться на одном из следующих двух уровней рабочей среды.

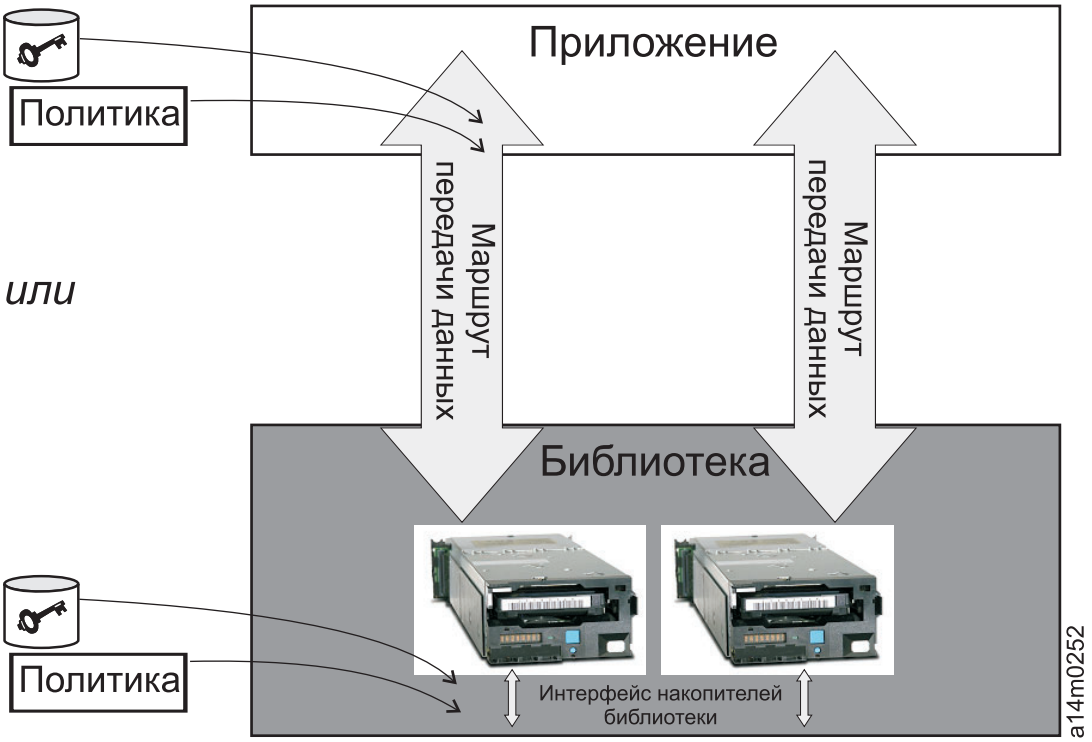

Рисунок 1-2. Два возможных уровня для управления ключами и механизма политики шифрования.

Уровень приложений

Отдельное приложение, помимо диспетчера ключей, инициирует передачу данных для системы хранения данных на магнитной ленте. Подробные сведения о поддерживаемых приложениях см. в разделе “Управляемое приложением шифрование данных на магнитной ленте”.

Уровень библиотеки

Устройство, в котором помещается система хранения данных на магнитной ленте - например, библиотеки Dell PowerVault TL2000/TL4000 и ML6000. Современная библиотека магнитных лент включает в себя внутренний интерфейс для каждого накопителя на магнитной ленте, входящего в ее состав.

Управляемое приложением шифрование данных на магнитной ленте

Этот метод лучше всего подходит для операционных сред, в которых уже выполняется приложение, способное создавать политики и ключи шифрования и управлять ими. Политики, задающие область использования шифрования, определяются с использованием интерфейса приложения. Политики и ключи передаются по пути прохождения данных между уровнем приложений и накопителями на магнитной ленте с поддержкой шифрования. Шифрование - это результат взаимодействия приложения и накопителя на магнитной ленте с возможностью шифрования, оно не требует изменений на уровне системы и библиотеки. Поскольку приложение управляет ключами шифрования, тома, записанные и зашифрованные под управлением приложения, доступны для чтения только с использованием метода шифрования, управляемого приложением - тем же самым приложением, с помощью которого они были записаны.

Encryption Key Manager не требуется и не используется при шифровании данных на магнитной ленте, управляемом приложением.

Для управления шифрованием могут использоваться следующие минимальные версии приложений:

- CommVault Galaxy 7.0 SP1;
- Symantec Backup Exec 12.

Шифрование данных на магнитной ленте, управляемое приложением, поддерживается накопителями на магнитной ленте LTO 4 и LTO 5 в следующих моделях библиотек:

- Библиотека магнитных лент Dell™ PowerVault™ TL2000
- Библиотека магнитных лент Dell™ PowerVault™ TL4000
- Библиотека магнитных лент Dell™ PowerVault™ ML6000

Сведения об управлении политиками и ключами шифрования см. в документации к вашему приложению для резервного копирования магнитных лент.

Управляемое библиотекой шифрование данных на магнитной ленте

Данный метод используется для ленточных накопителей LTO 4 и LTO 5 в следующих библиотеках: Библиотека магнитных лент Dell™ PowerVault™ TL2000, Библиотека магнитных лент Dell™ PowerVault™ TL4000 или Библиотека магнитных лент Dell™ PowerVault™ ML6000. Encryption Key Manager, приложение Java, выполняемое на подключенном к библиотеке хосте, отвечает за создание ключей и управление ими. Управление политикой и ключи передаются через интерфейс взаимодействия библиотеки и накопителя, поэтому шифрование прозрачно для приложений.

О ключах шифрования

Ключ шифрования - это произвольная строка битов, созданная специально для кодирования и декодирования данных. Ключи шифрования создаются с помощью алгоритмов, гарантирующих уникальность и непрогнозируемость каждого ключа. Чем длиннее ключ, созданный таким способом, тем сложнее взломать код шифрования. В методах шифрования IBM и T10 для шифрования данных используется 256-разрядный алгоритм стандарта AES. 256-разрядный стандарт AES - это стандарт шифрования, признанный и рекомендуемый к использованию правительством США, который допускает три различных длины ключа. 256-разрядные ключи - самый длинные из поддерживаемых стандартом AES.

Encryption Key Manager поддерживает два типа алгоритмов шифрования: симметричные и асимметричные алгоритмы. При симметричном шифровании, или шифровании с помощью засекреченного ключа, один и тот же ключ используется как для шифрования, так и для расшифровки. Шифрование с помощью симметричного ключа обычно используется для эффективного шифрования больших объемов данных. 256-разрядные ключи AES являются симметричными. При асимметричном, или открытом/секретном шифровании, используется пара ключей. Данные, зашифрованные с помощью одного ключа, могут быть расшифрованы только с помощью другого ключа из пары "открытый ключ/секретный ключ". При создании пары асимметричных ключей открытый ключ используется для шифрования, а секретный - для расшифровки.

Encryption Key Manager использует как симметричные, так и асимметричные ключи: симметричное шифрование - для скоростного шифрования данных пользователя или хоста, а асимметричное шифрование (которое всегда медленнее симметричного) - для защиты симметричного ключа.

Ключи шифрования для Encryption Key Manager можно создавать с помощью утилит, таких как keytool. Ответственность за создание ключей AES и способ их передачи на ленточный накопитель зависит от метода управления шифрованием. Однако понимание различия в использовании ключей шифрования в Encryption Key Manager и в других приложениях может быть полезным.

Обработка ключей шифрования в Dell Encryption Key Manager

При шифровании данных на магнитной ленте, управляемом библиотекой, незашифрованные данные отправляются на накопитель на магнитной ленте LTO 4 или LTO 5 и преобразуются в шифрованный текст с помощью предварительно созданного симметричного ключа шифрования данных (DK) из хранилища ключей, доступного для Encryption Key Manager, а затем записываются на магнитную ленту. В Encryption Key Manager предварительно созданный ключ выбирается циклически. Ключи используются повторно на кассетах с несколькими лентами в том случае, если предварительно созданных ключей шифрования данных недостаточно. С помощью Encryption Key Manager ключ отправляется на накопитель на магнитной ленте LTO 4 или LTO 5 в зашифрованном или свернутом виде. Накопители LTO 4 и LTO 5 разворачивают этот ключ и с его помощью выполняют шифрование или расшифровку данных. Однако на самих кассетах с магнитной лентой LTO 4 или LTO 5 зашифрованный ключ не хранится. После записи шифрованного тома ключ должен быть доступен на основании псевдонима или метки ключа для Encryption Key Manager, чтобы можно было осуществить чтение этого тома. Рисунок рис. 1-3 на стр. 1-8 иллюстрирует данный процесс.

Кроме того, Dell Encryption Key Manager позволяет упорядочить симметричные ключи для шифрования LTO и создать различные группы ключей. Возможно группирование ключей по типу шифруемых данных, типу пользователей, имеющих доступ к этим ключам, или по иным существенным параметрам. Дополнительные сведения см. в разделе “Создание групп ключей и управление ими” на стр. 3-16.

Обработка ключей шифрования другими приложениями

При шифровании данных на магнитной ленте, управляемом приложением, незашифрованные данные отправляются на ленточные накопители LTO 4 и LTO 5 и преобразуются в шифрованный текст с помощью симметричного ключа шифрования данных (DK), предоставленного приложением, а затем записываются на магнитную ленту. Ключ шифрования данных не хранится на кассете с магнитной лентой. После записи шифрованного тома ключ должен быть размещен в доступном для приложения месте - например, в серверной базе данных, - чтобы том можно было прочесть.

На ленточных накопителях LTO 4 и LTO 5 управляемое приложением шифрование может выполняться с помощью таких приложений, как Yosemite (для библиотек магнитных лент Dell PowerVault TL2000 и TL4000), CommVault и Symantec Backup Exec.

В свою очередь, приложения, использующие набор команд T10, могут использовать для выполнения шифрования накопители на магнитной ленте LTO 4 и LTO 5. Набор команд T10 использует симметричные 256-разрядные ключи стандарта AES, предоставляемые приложением. T10 может использовать несколько уникальных ключей на одной кассете с магнитной лентой и даже записывать зашифрованные данные и открытые данные на одну и ту же кассету с магнитной лентой. При шифровании кассеты с магнитной лентой приложение выбирает или создает ключ, используя метод, определяемый приложением, и отправляет его на накопитель на магнитной ленте. Этот ключ **не** скрывается асимметричным открытым ключом и **не**

хранится на кассете с магнитной лентой. После того как шифрованные данные записаны, для их последующего прочтения ключ должен быть размещен в доступном для приложения месте.

Процесс шифрования данных на магнитной ленте, управляемый приложением и управляемый библиотекой, показан на рис. 1-3.

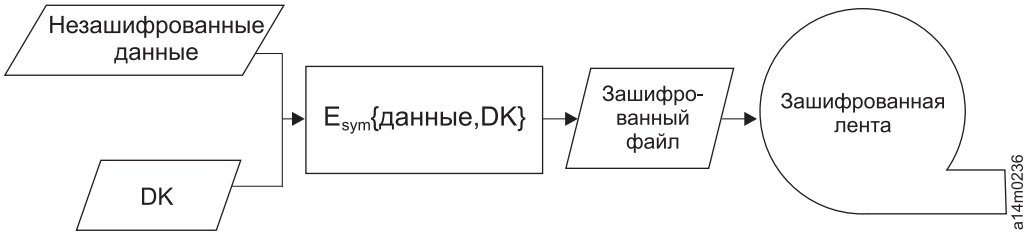

Рисунок 1-3. Шифрование с помощью симметричных ключей шифрования. Шифрование, управляемое библиотекой и управляемое приложением, на накопителях на магнитной ленте LTO 4 и LTO 5.

Краткая сводка

Число ключей шифрования, используемых для каждого тома, зависит от накопителя на магнитной ленте, стандарта шифрования и метода управления шифрованием. При прозрачном шифровании LTO 4 и LTO 5 (то есть при управляемом библиотекой шифровании с помощью Encryption Key Manager) уникальность ключей зависит от доступности для Encryption Key Manager достаточного числа предварительно созданных ключей.

Таблица 1-1. Краткие сведения о ключах шифрования

	Используемые ключи	
Метод управления шифрованием	**Шифрование IBM**	**Шифрование T10**
Шифрование, управляемое библиотекой	1 DK на каждую кассету	Неприменимо
Шифрование, управляемое приложением	Несколько ключей DK на каждую кассету	Несколько ключей DK на каждую кассету
DK = симметричный 256-разрядный ключ шифрования данных стандарта AES		

Глава 2. Планирование среды Encryption Key Manager

Этот раздел содержит сведения, помогающие определить конфигурацию Encryption Key Manager, лучшую с точки зрения потребностей предприятия. При разработке стратегии шифрования следует учитывать множество факторов.

Общее представление о настройке шифрования

Использование функций шифрования ленточного накопителя возможно при условии выполнения определенных требований к аппаратному и программному обеспечению. Следующие контрольные списки должны помочь выполнить данные требования.

Задачи настройки Encryption Key Manager

Перед шифрованием данных на магнитной ленте приложение Encryption Key Manager следует настроить и запустить, чтобы обеспечить возможность его взаимодействия с накопителями на магнитной ленте, поддерживающими шифрование. Запуск Encryption Key Manager не требуется при установке накопителей на магнитной ленте, но для выполнения шифрования это необходимо.

- Определите, какая системная платформа(-ы) будет использоваться в качестве сервера(-ов) Encryption Key Manager.
- При необходимости обновите операционную систему сервера. (См. раздел “Требования к оборудованию и программному обеспечению” на стр. 2-2.)
- Установите файлы неограниченной политики Java. (См. раздел “Требования к оборудованию и программному обеспечению” на стр. 2-2.)
- Обновите файл JAR для Encryption Key Manager. (См. раздел “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1.)
- Создайте ключи, сертификаты и группы ключей.

  “Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя” на стр. 3-5

  “Создание групп ключей и управление ими” на стр. 3-16
- Шаги, описанные ниже, не обязательны, если вы выполняете процедуру, приведенную в разделе “Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя” на стр. 3-5. Выполнение этих действий позволит воспользоваться преимуществами дополнительных параметров конфигурации.
  - При необходимости импортируйте ключи и сертификаты. (См. раздел “Импорт ключей шифрования данных с помощью команды keytool -importseckey ” на стр. 3-14.)
  - Определите файл свойств конфигурации. (См. раздел Глава 4, “Конфигурирование диспетчера ключей шифрования (Encryption Key Manager, EKM)”, на стр. 4-1.)
  - Определите накопители на магнитной ленте для Encryption Key Manager или задайте значение "включено" для свойства конфигурации **drive.acceptUnknownDrives**. (См. инструкции по явному определению накопителей в описании команды “adddrive” на стр. 5-8 или см. раздел “Автоматическое обновление таблицы ленточных накопителей” на стр. 4-1.)
  - Запустите сервер Encryption Key Manager. (См. раздел “Запуск, обновление и остановка сервера диспетчера ключей” на стр. 5-1.)

– Запустите клиент интерфейса командной строки. (См. раздел “Клиент интерфейса командной строки” на стр. 5-5.)

Планирование управляемого библиотекой шифрования данных на магнитной ленте

Для выполнения шифрования необходимы:

- Ленточные накопители LTO 4 и LTO 5 с поддержкой шифрования
- Хранилище ключей
- Dell Encryption Key Manager

Задачи управляемого библиотекой шифрования данных на магнитной ленте

1. Установите ленточные накопители LTO 4 и LTO 5 и подключите к ним кабели.
   - Обновите встроенное ПО библиотеки (TL2000, TL4000, ML6000, если необходимо). Посетите Web-сайт **http://support.dell.com**.
     – Библиотека магнитных лент Dell™ PowerVault™ TL2000: минимальная требуемая версия встроенного ПО - 5.xx.
     – Библиотека магнитных лент Dell™ PowerVault™ TL4000: минимальная требуемая версия встроенного ПО - 5.xx.
     – Семейство библиотек магнитных лент Dell™ PowerVault™ ML6000: минимальная требуемая версия встроенного ПО - 415G.xxx.
   - При необходимости обновите встроенное ПО накопителя на магнитной ленте. Минимальная требуемая версия встроенного ПО - 77B5.
2. Активируйте на ленточных накопителях LTO 4 и LTO 5 и библиотеках функцию управляемого библиотекой шифрования данных на магнитной ленте (подробные сведения см. в документации к ленточной библиотеке Dell).
   - Добавьте IP-адреса сервера Encryption Key Manager.
3. С помощью функций диагностики библиотеки проверьте пути и конфигурацию шифрования Encryption Key Manager (дополнительные сведения см. в документации к ленточной библиотеке Dell).

Требования к оборудованию и программному обеспечению

Примечание: Для каждой из следующих платформ Encryption Key Manager поддерживается только в среде выполнения Java Runtime Environment (JRE) версии IBM.

Компоненты решения для Linux

Операционные системы

- RHEL 4
- RHEL 5
- SLES 9
- SLES 10
- SLES 11

Encryption Key Manager (на платформе Linux)

Таблица 2-1. Минимальные требования к программному обеспечению для запуска в системе Linux

Система	Набор разработки программного обеспечения IBM	Доступно по адресу:
64-разрядный процессор AMD/Opteron/EM64T	Java 6.0 SR5	**http://support.dell.com**
32-разрядный процессор, совместимый с Intel®		

Ленточные библиотеки

Убедитесь, что на ленточных библиотеках Dell PowerVault TL2000, TL4000 и ML6000 установлена последняя версия встроенного ПО. Обновления встроенного ПО можно загрузить с сайта **http://support.dell.com**.

Ленточный накопитель

Убедитесь, что на ленточных накопителях LTO 4 и LTO 5 установлена последняя версия встроенного ПО. Обновления встроенного ПО можно загрузить с сайта **http://support.dell.com**.

Компоненты решения для Windows

Операционные системы

Windows Server 2003, 2008 и 2008 R2

Dell Encryption Key Manager

Минимальное требование - версия Encryption Key Manager не ниже 2.1, с датой сборки не ранее 14.09.2007, а также одна из следующих сред выполнения IBM Runtime Environment.

Таблица 2-2. Минимальные требования к программному обеспечению для запуска в Windows

Операционная система	Среда выполнения IBM Runtime Environment
Windows 2003	• 64-разрядная среда IBM® Runtime Environment для Windows с архитектурой AMD64/EM64T, Java 2 Technology Edition версии 5.0 SR5 • 32-разрядная среда IBM Runtime Environment для Windows, Java 2 Technology Edition версии 5.0 SR5
Windows 2008 и 2008 R2	64-разрядная среда IBM Runtime Environment для Windows с архитектурой AMD64/EM64T, Java 2 Technology Edition версии 6.0 SR5

Ленточные библиотеки

Убедитесь, что последняя версия встроенного ПО установлена на следующих ленточных библиотеках: Библиотека магнитных лент Dell™ PowerVault™ TL2000, Библиотека магнитных лент Dell™ PowerVault™ TL4000 и Библиотека магнитных лент

Dell™ PowerVault™ ML6000. Обновления встроенного ПО можно загрузить с сайта **http://support.dell.com**.

Ленточный накопитель

Убедитесь, что на ленточных накопителях LTO 4 и LTO 5 установлена последняя версия встроенного ПО. Обновления встроенного ПО можно загрузить с сайта **http://support.dell.com**.

Замечания о хранилище ключей

Невозможно переоценить важность обеспечения сохранности данных хранилища ключей. Без доступа к хранилищу ключей не удастся расшифровать зашифрованные данные на магнитной ленте. Внимательно прочитайте следующие разделы, чтобы ознакомиться с методами защиты хранилища ключей.

Хранилище ключей JCEKS

EKM поддерживает хранилища ключей формата JCEKS.

JCEKS (файловая подсистема Unix System Services) – это файловое хранилище ключей, которое поддерживается на всех платформах, на которых выполняется EKM. Это упрощает копирование содержимого такого хранилища ключей при выполнении резервного копирования и восстановления, а также позволяет поддерживать две синхронизированных копии EKM для переключения в случае отказа. JCEKS обеспечивает безопасность за счет защиты содержимого хранилища ключей паролем, а также обладает относительно высокой производительностью. Возможно использование таких методов копирования файлов, как передача по протоколу FTP.

Ключи шифрования и ленточные накопители LTO 4 и LTO 5

Dell Encryption Key Manager и поддерживаемые этим продуктом накопители на магнитной ленте используют для шифрования данных симметричные 256-разрядные ключи AES. В данном разделе содержатся необходимые сведения об этих ключах и сертификатах.

При выполнении операций шифрования на накопителях LTO 4 или LTO 5 для кассет LTO приложение Encryption Key Manager использует только симметричные 256-разрядные ключи AES.

Когда накопитель LTO 4 или LTO 5 запрашивает ключ, Encryption Key Manager использует псевдоним, указанный для этого накопителя на магнитной ленте. Если для данного накопителя на магнитной ленте псевдоним не был задан, используется псевдоним из группы ключей, списка псевдонимов ключей или диапазона значений псевдонимов ключей, заданного свойством конфигурации symmetricKeySet. Если псевдоним накопителя на магнитной ленте отсутствует, псевдонимы выбираются из других групп по очереди, чтобы обеспечить равномерное распределение ключей.

Выбранный псевдоним связывается с симметричным ключом шифрования данных (DK), предварительно загруженным в хранилище ключей. Encryption Key Manager посылает этот ключ DK, свернутый с помощью другого ключа, который может быть дешифрован накопителем на магнитной ленте, на накопитель LTO 4 или LTO 5 для шифрования данных. Ключ DK не передается по протоколу TCP/IP в незашифрованном виде. Выбранный псевдоним также преобразуется в объект под названием "идентификатор ключа шифрования данных" (Data Key identifier, DKi),

который записывается на магнитную ленту с зашифрованными данными. После этого Encryption Key Manager может использовать DKi для идентификации ключа DK, необходимого для расшифровки данных при чтении с магнитной ленты LTO 4 или LTO 5.

В подразделах, посвященных командам **adddrive** и **moddrive**, раздела “Команды CLI” на стр. 5-8 объясняется, как задать псевдоним для накопителя на магнитной ленте. См. раздел “Создание ключей и псевдонимов для шифрования на накопителях LTO 4 и LTO 5” на стр. 3-10, содержащий сведения об импортировании и экспортировании ключей и о задании псевдонимов по умолчанию в свойстве конфигурации symmetricKeySet. В разделе “Создание групп ключей и управление ими” на стр. 3-16 рассказывается о том, как определить группу ключей и заполнить ее псевдонимами из хранилища ключей.

На рис. 2-1 представлен процесс обработки ключей при записи с шифрованием.

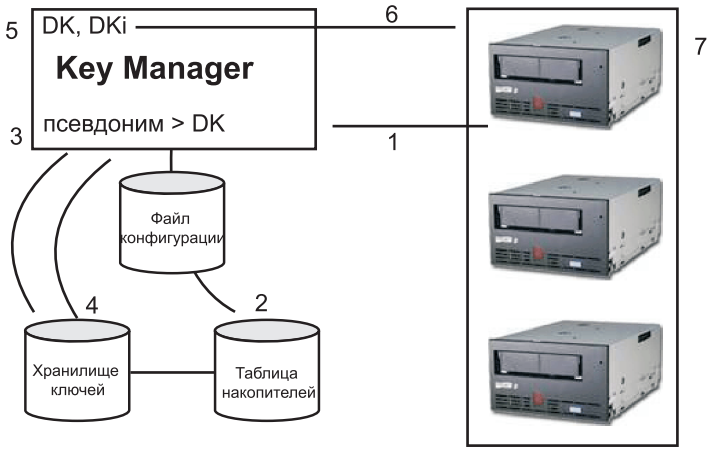

Рисунок 2-1. Запрос накопителя на магнитной ленте LTO 4 или LTO 5 на операцию записи с шифрованием

1. Накопитель на магнитной ленте запрашивает ключ для шифрования магнитной ленты.
2. Encryption Key Manager проверяет ленточное устройство по таблице накопителей.
3. Если ни в запросе, ни в таблице накопителей псевдоним не указан, Encryption Key Manager выбирает его из группы псевдонимов или группы ключей, заданной свойством keyAliasList.
4. Encryption Key Manager извлекает из хранилища ключей соответствующий ключ DK.
5. Encryption Key Manager преобразует псевдоним в DKi и свертывает ключ DK с помощью ключа, который накопитель может дешифровать.
6. Encryption Key Manager посылает ключ DK и DKi накопителю на магнитной ленте.
7. Накопитель на магнитной ленте расшифровывает ключ DK и записывает зашифрованные данные и DKi на ленту.

На рис. 2-2 на стр. 2-6 представлен процесс обработки ключей при чтении зашифрованных данных.

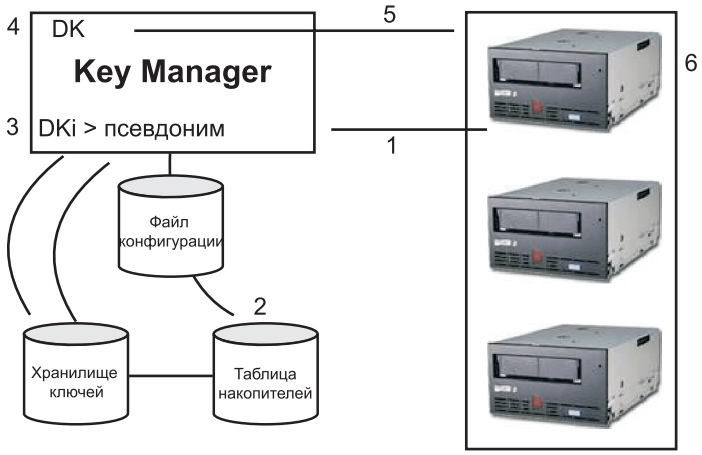

Рисунок 2-2. Запрос накопителя на магнитной ленте LTO 4 или LTO 5 на операцию чтения с расшифровкой

1. Накопитель на магнитной ленте получает запрос на чтение и посылает DKi приложению Encryption Key Manager.
2. Encryption Key Manager проверяет ленточное устройство по таблице накопителей.
3. Encryption Key Manager преобразует DKi в псевдоним и извлекает из хранилища ключей соответствующий ключ DK.
4. Encryption Key Manager свертывает ключ DK с помощью ключа, который накопитель может дешифровать.
5. Encryption Key Manager посылает свернутый ключ DK накопителю на магнитной ленте.
6. Накопитель на магнитной ленте расшифровывает ключ DK и использует его для расшифровки данных.

Резервное копирование данных хранилища ключей

Примечание: Вследствие чрезвычайной важности ключей, находящихся в хранилище, обязательно сохраняйте резервные копии данных хранилища на незашифрованном устройстве. Это позволит при необходимости восстановить данные хранилища ключей для чтения магнитных лент, которые были зашифрованы с помощью сертификатов, связанных с соответствующим накопителем на магнитной ленте или библиотекой магнитных лент. Отсутствие должного резервного копирования хранилища ключей приведет к безвозвратной потере любого доступа к зашифрованным данным.

Существует множество способов резервного копирования хранилища ключей. Каждый тип хранилища обладает своими уникальными характеристиками. Приведенные ниже рекомендации подходят для всех типов хранилищ ключей.

- Храните копию всех сертификатов, загруженных в хранилище ключей (как правило, файл формата PKCS12).
- Используйте возможности резервного копирования, существующие в системе (например, RACF), для создания резервной копии данных хранилища ключей (будьте внимательны: не шифруйте эту копию с использованием накопителей на магнитной ленте с поддержкой шифрования, поскольку ее расшифровка для восстановления будет невозможна).
- Поддерживайте основной и вспомогательный экземпляры Encryption Key Manager и копию хранилища ключей (для резервного копирования, а также обеспечения

отказоустойчивости за счет избыточности). Для дополнительного резервирования создайте резервную копию основной и вспомогательной копий.

- При использовании хранилища ключей JCEKS просто скопируйте файл хранилища ключей и храните обычную копию (без шифрования) в защищенном каталоге, например, в сейфе (будьте внимательны: не шифруйте эту копию с использованием накопителей на магнитной ленте с поддержкой шифрования, поскольку ее расшифровка для восстановления будет невозможна).

Необходимо создавать резервную копию хранилища ключей, по меньшей мере, каждый раз, когда в него вносятся изменения. Encryption Key Manager не изменяет данные хранилища ключей. Внести изменения в данные хранилища ключей может только пользователь, поэтому не забывайте сохранять резервную копию хранилища ключей после каждого изменения.

Резервное копирование файлов с помощью графического интерфейса пользователя

1. Если графический интерфейс пользователя еще не открыт, откройте его.

   В Windows
   Перейдите в каталог c:\ekm\gui и щелкните кнопкой мыши по файлу **LaunchEKMGui.bat**

   На платформе Linux
   Перейдите в каталог /var/ekm/gui и запустите файл `. ./LaunchEKMGui.sh`

2. В области навигации в левой части графического интерфейса приложения Encryption Key Manager выберите команду **Backup Critical Files** (Создать резервные копии важных файлов).
3. В появившемся диалоговом окне выберите месторасположение резервной копии данных (рис. 2-3 на стр. 2-8).

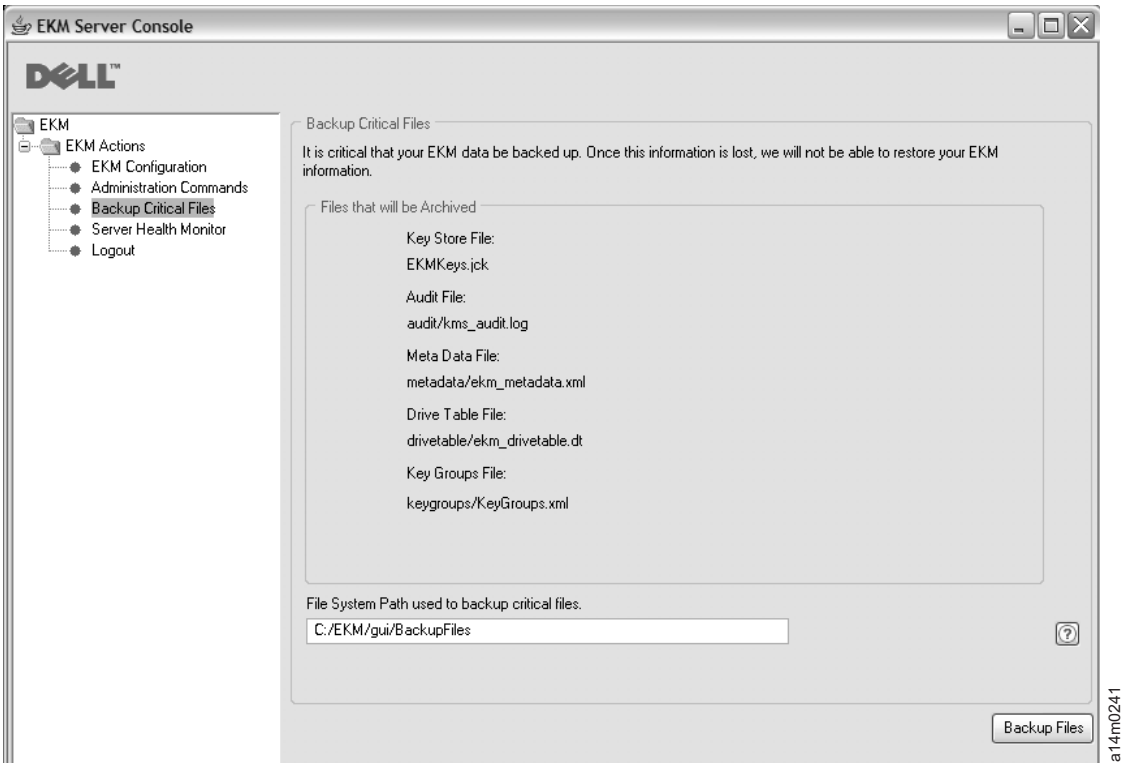

Рисунок 2-3. Окно Backup Critical Files (Создание резервных копий важных файлов)

4. Нажмите кнопку **Backup Files** (Создать резервную копию файлов).
5. Появится сообщение с информацией о результатах.

Использование нескольких диспетчеров ключей для обеспечения избыточности

Решение Encryption Key Manager предусматривает работу с накопителями на магнитной ленте и с библиотеками с обеспечением избыточности и, следовательно, высокой доступности, поэтому для обслуживания накопителей и библиотек можно использовать несколько диспетчеров ключей. Более того, диспетчеры ключей необязательно должны находиться в тех же системах, что и ленточные накопители и библиотеки. Максимальное число диспетчеров ключей зависит от используемой библиотеки и прокси-сервера. Единственное требование состоит в том, что они должны быть доступны для ленточных накопителей с помощью подключения по протоколу TCP/IP.

Это позволяет иметь два экземпляра Encryption Key Manager, которые являются зеркальными образами друг друга с встроенными средствами резервного копирования важнейших данных о хранилищах ключей, а также с возможностью переключения, если один из диспетчеров ключей становится недоступным. При настройке устройства (или прокси-сервера) его можно связать с двумя диспетчерами ключей. Если один диспетчер ключей по какой-либо причине становится недоступным, то устройство (или библиотека) будет использовать другой диспетчер ключей.

Кроме того, существует возможность поддерживать два экземпляра Encryption Key Manager в синхронизированном состоянии. Очень важно использовать эту функцию, когда необходимо, для обеспечения как резервного копирования важных данных, так

и возможности переключения в случае отказа, что позволяет избежать простоев при эксплуатации ленточных систем. См. раздел “Синхронизация данных между двумя серверами диспетчера ключей” на стр. 4-2.

Примечание: Синхронизация не затрагивает хранилища ключей. Их следует копировать вручную.

Конфигурации сервера Encryption Key Manager

Encryption Key Manager можно установить на один или несколько серверов. В следующих примерах показаны конфигурации диспетчера для одного и двух серверов, но ваша библиотека может допускать установку на большее количество серверов.

Конфигурация с одним сервером

Конфигурация с одним сервером, показанная на рис. 2-4, - это простейшая конфигурация Encryption Key Manager. Однако не рекомендуется использовать подобную конфигурацию, поскольку она не обеспечивает избыточности. В такой конфигурации все ленточные накопители зависят от одного сервера диспетчера ключей, не обеспечивающего возможности восстановления. В случае отказа сервера хранилище ключей, файл конфигурации, файл KeyGroups.xml и таблица накопителей станут недоступны и зашифрованную ленту невозможно будет прочитать. В конфигурации с одним сервером необходимо обеспечить хранение резервных копий хранилища ключей, файла конфигурации, файла KeyGroups.xml и таблицы накопителей в надежном месте, отдельно от Encryption Key Manager, чтобы его работу можно было восстановить на дополнительном сервере в случае потери копий на основном сервере.

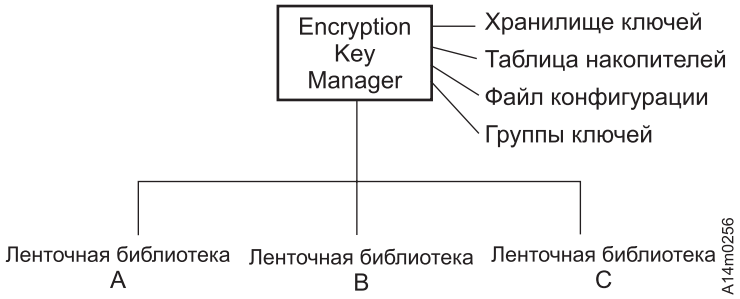

Рисунок 2-4. Конфигурация с одним сервером

Двухсерверные конфигурации

Рекомендуется использовать двухсерверную конфигурацию. При данной конфигурации Encryption Key Manager происходит автоматическое переключение на дополнительный диспетчер ключей, если основной становится по какой-то причине недоступен.

Примечание: Когда для обработки запросов от одного набора накопителей на магнитной ленте используются различные серверы Encryption Key Manager, сведения в соответствующих хранилищах ключей ДОЛЖНЫ быть одинаковыми. Это требуется для того, чтобы сведения, необходимые для поддержки запросов от накопителей на магнитной ленте, были всегда доступны независимо от используемого сервера диспетчера ключей.

Идентичные конфигурации: В средах с двумя серверами Encryption Key Manager с идентичными конфигурациями, как у представленных на рис. 2-5 на стр. 2-10,

происходит автоматическое переключение на дополнительный диспетчер ключей при отказе основного. В такой конфигурации серверы диспетчеров ключей обязательно должны быть синхронизированы. Обновления в файле конфигурации и таблице накопителей на одном сервере диспетчера ключей можно продублировать на другом автоматически с помощью команды **sync**, однако обновления в одном хранилище ключей необходимо скопировать в другое подходящим для используемого хранилища ключей способом. Хранилища ключей и XML-файл групп ключей следует копировать вручную. Дополнительные сведения см. в разделе “Синхронизация данных между двумя серверами диспетчера ключей” на стр. 4-2.

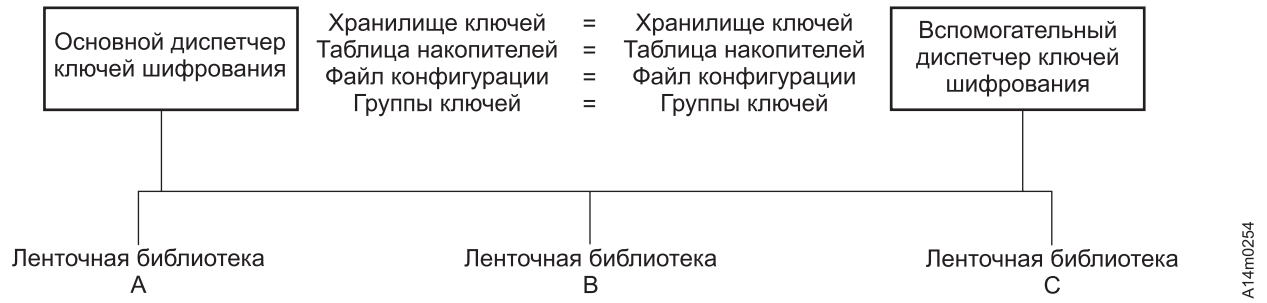

Рисунок 2-5. Два сервера с общей конфигурацией

Разные конфигурации: Два сервера Encryption Key Manager могут иметь общее хранилище ключей и таблицу накопителей, но разные файлы конфигурации и разные наборы групп ключей, определенных в XML-файлах серверов. Единственное требование состоит в том, что серверы должны использовать для обслуживания общих ленточных накопителей одни и те же ключи. Это позволяет каждому серверу диспетчера ключей иметь собственный набор свойств. В конфигурации такого типа, показанной на рис. 2-6, необходимо синхронизировать только таблицы накопителей серверов EKM. (Дополнительные сведения см. в разделе “Синхронизация данных между двумя серверами диспетчера ключей” на стр. 4-2.) Убедитесь, что значение параметра `sync.type = drivetab` задано верно (не используйте значения config или all), чтобы не допустить перезаписи файлов конфигурации.

Примечание: Частично общая конфигурация для серверов невозможна.

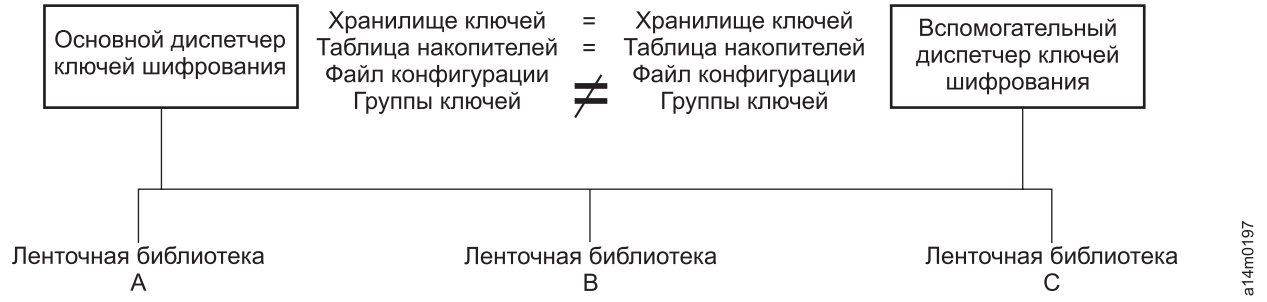

Рисунок 2-6. Два сервера с разными конфигурациями, обращающиеся к одним устройствам

Замечания о резервной площадке

Если планируется использовать резервную площадку, Encryption Key Manager предоставляет ряд возможностей, позволяющих этой резервной площадке записывать зашифрованные данные на ленты и считывать данные с лент. Эти возможности описаны ниже.

- Создайте копию Encryption Key Manager на резервной площадке.

Настройте копию Encryption Key Manager на резервной площадке с использованием тех же данных, что и в локальном приложении Encryption Key Manager (файл конфигурации, таблица ленточных накопителей, XML-файл групп ключей и хранилище ключей). Этот диспетчер ключей, таким образом, сможет брать на себя управление и считывать и записывать зашифрованные данные на магнитных лентах вместо одного из существующих производственных диспетчеров ключей.

- Создайте резервную копию трех файлов данных Encryption Key Manager, которые при необходимости можно будет использовать для восстановления.

  Создание текущей копии четырех объектов данных, необходимых для работы Encryption Key Manager (файл конфигурации, таблица ленточных накопителей, XML-файл групп ключей и хранилище ключей), позволит в любое время запустить резервный диспетчер в системе восстановления после сбоев. (Следует помнить, что нельзя использовать Encryption Key Manager для шифрования копий этих файлов, так как без работающего диспетчера ключей расшифровать их не удастся.) Если на резервной площадке используются ленточные накопители, отличные от размещенных в основном вычислительном центре, то файл конфигурации и таблица ленточных накопителей должны содержать правильные данные для резервной площадки.

Замечания об общем доступе к зашифрованным лентам за пределами организации

Примечание: Важно убедиться в действительности любого сертификата, полученного от бизнес-партнера, проверив цепочку доверия такого сертификата вплоть до центра сертификации (Certificate Authority, CA), который изначально выдал его. Если вы доверяете центру сертификации, то можно доверять такому сертификату. Действительность сертификата также может быть подтверждена, если он надежно охранялся при перемещении. Если ни один из этих способов не позволяет подтвердить действительность сертификата, это повышает вероятность атак типа “man-in-the-middle”.

Общий доступ к лентам LTO 4 и LTO 5

Чтобы обеспечить общий доступ к зашифрованным данным на лентах LTO 4 или LTO 5, необходимо предоставить другой организации копию симметричного ключа, с помощью которого зашифрованы данные на ленте. Это позволит этой организации считать данные с ленты. Чтобы передать симметричный ключ, другая организация должна предоставить вам свой открытый ключ. Этот открытый ключ будет использоваться для свертывания симметричного ключа при его экспорте из хранилища ключей Encryption Key Manager с помощью средства keytool (см. раздел “Экспорт ключей шифрования данных с помощью команды keytool -exportseckey ” на стр. 3-14). Когда другая организация импортирует симметричный ключ в свое хранилище ключей Encryption Key Manager, он будет развернут с помощью соответствующего секретного ключа (см. раздел “Импорт ключей шифрования данных с помощью команды keytool -importseckey ” на стр. 3-14). Это гарантирует, что симметричный ключ остается защищенным при перемещении, поскольку расшифровать его может только владелец секретного ключа. Имея симметричный ключ, использовавшийся для шифрования данных, в своем хранилище ключей Encryption Key Manager, другая организация сможет читать данные на ленте.

Замечания о федеральном стандарте обработки информации (Federal Information Processing Standards, FIPS) 140-2

Федеральный стандарт обработки информации FIPS 140-2 имеет сегодня большое значение, поскольку правительство США требует от всех своих поставщиков криптографических систем наличия сертификата соответствия этому стандарту. Этот стандарт также был принят в растущем негосударственном секторе криптографии. Сертификация криптографических средств третьими лицами в соответствии с правительственными стандартами приобрела большое значение в современном мире, озабоченном проблемами безопасности.

Программное обеспечение Encryption Key Manager само не предоставляет средств шифрования и поэтому не подлежит сертификации по стандарту FIPS 140-2. Однако в Encryption Key Manager используются криптографические средства IBM JVM компонента IBM Java Cryptographic Extension, что позволяет выбирать и использовать поставщика криптографических услуг IBMJCEFIPS, который имеет сертификат FIPS 140-2 уровня 1. При задании в файле свойств конфигурации значения **on** для параметра **fips** в Encryption Key Manager для всех криптографических функций будет использоваться поставщик IBMJCEFIPS.

Сведения о сертификации аппаратных или программных криптографических средств по стандарту FIPS 140-2 см. в документации поставщика.

Глава 3. Установка диспетчера ключей шифрования и хранилищ ключей

Encryption Key Manager поставляется в комплекте с установкой виртуальной машины Java компании IBM (IBM Java Virtual Machine, JVM) и требует наличия пакета для разработки ПО компании IBM (IBM Software Developer Kit) для ОС Linux и IBM Runtime Environment для ОС Windows (см. “Требования к оборудованию и программному обеспечению” на стр. 2-2). Выполните описанную ниже процедуру в соответствии с используемой операционной системой.

- “Установка Encryption Key Manager в ОС Linux”
- “Установка Encryption Key Manager в Windows” на стр. 3-2

Если вы не уверены, что используется последняя версия Encryption Key Manager, выполните приведенные в разделе “Загрузка последней версии ISO-образа диспетчера ключей” инструкции, чтобы проверить наличие более новой версии. Рекомендуется получить последнюю версию Encryption Key Manager, которая может не входить в установленную у вас версию Java. Для получения дополнительной информации посетите Web-сайт **http://support.dell.com**.

ВАЖНАЯ ИНФОРМАЦИЯ О КОНФИГУРАЦИИ ХОСТ-СЕРВЕРА Encryption Key Manager: Для минимизации риска потери данных на компьютерах с установленными программами Dell Encryption Key Manager рекомендуется использовать память ECC. Encryption Key Manager формирует запросы на генерацию ключей шифрования и их передачу накопителям на магнитной ленте LTO 4 и LTO 5. Encryption Key Manager при обработке ключа хранит его данные в системной памяти в свернутом (зашифрованном) виде. Следует заметить, что данные ключа необходимо передать соответствующему накопителю на магнитной ленте без ошибок, чтобы записанные на кассету данные можно было восстановить (расшифровать). Если по какой-либо причине данные ключа были повреждены из-за ошибок в разрядах системной памяти, но использовались для записи данных на кассету, то записанные на эту кассету данные будут недоступны для восстановления (последующей расшифровки). Существуют различные средства защиты, предотвращающие появление подобных ошибок данных. Однако если компьютер, на котором установлено приложение Encryption Key Manager, не использует память с коррекцией ошибок Error Correction Code (ECC), существует вероятность повреждения хранимых в памяти данных и, в результате, их потери. Вероятность появления таких ошибок достаточно мала, однако на компьютерах, на которых установлены критически важные приложения (например, Encryption Key Manager) всегда рекомендуется использовать память ECC.

Загрузка последней версии ISO-образа диспетчера ключей

Для загрузки последней версии ISO-образа Dell перейдите на Web-сайт http://support.dell.com.

Установка Encryption Key Manager в ОС Linux

Установка Encryption Key Manager в ОС Linux с компакт-диска

1. Вставьте компакт-диск Dell Encryption Key Manager и введите команду `Install_Linux` в корневом каталоге компакт-диска.

При установке все содержимое (документация, файлы GUI-интерфейса, файлы свойств конфигурации), соответствующее операционной системе, копируется с компакт-диска на жесткий диск компьютера. Во время установки система проверяется на наличие правильной версии среды IBM Java Runtime Environment. Если такая версия не обнаружена, она автоматически устанавливается.

По окончании установки запускается графический пользовательский интерфейс (GUI).

Установка пакета разработки ПО в ОС Linux вручную

Если установка производится не с компакт-диска, выполните следующие действия:

1. Загрузите с сайта **http://support.dell.com** правильную версию среды выполнения для Java, соответствующую операционной системе:
   - Java 6 SR 5 (32-битовая) или более поздняя версия
   - Java 6 SR 5 (64-битовая) или более поздняя версия
2. Поместите файл Java linux rpm в рабочий каталог:

   ```
   mordor:~ #/tape/Encryption/java/1.6.0# pwd
   /tape/Encryption/java/1.6.0
   mordor:~ #/tape/Encryption/java/1.6.0# ls
   ibm-java-i386-jre-6.0-5.0.i386.rpm
   ```
3. Установите пакет rpm:

   ```
   mordor:~ #rpm -ivh -nodeps ibm-java-i386-jre-6.0-5.0.i386.rpm
   ```

   При выполнении следующей команды файлы будут размещены в каталоге **/opt/ibm/java-i386-60/**:

   ```
   mordor:~ #/opt/ibm/java-i386-60/jre # ls
   .systemPrefs  bin  javaws  lib
   ```
4. Отредактируйте (или при необходимости создайте) файл **/eatc/profile.local**, содержащий каталоги JAVA_HOME, CLASSPATH и bin для установленной версии Java. Добавьте следующие три строки:

   ```
   JAVA_HOME=/opt/ibm/java-i386-60/jre
   CLASSPATH=/opt/ibm/java-i386-60/jre/lib
   PATH=$JAVA_HOME:opt/ibm/java-i386-60/jre/bin/:$PATH
   ```
5. Выйдите с хост-сервера и войдите опять, чтобы новые записи в **/etc/profile.local** вступили в действие, или введите следующие команды экспорта с помощью командной строки:

   ```
   mordor:~ # export JAVA_HOME=/opt/ibm/java-i386-60/jre
   mordor:~ # export CLASSPATH=/opt/ibm/java-i386-60/jre/lib
   mordor:~ # export PATH=/opt/ibm/java-i386-60/jre/bin/:$PATH
   ```
6. После вторичного входа в систему введите команду **java -version**. Вы должны увидеть на экране следующее:

   ```
   mordor:~ # java -version
   java version "1.6.0"
   Java(TM) SE Runtime Environment (build pmz60sr5-20090529(SR5))
   IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Linux x86-32 jvmxi3260-20090519_35743
   (JIT enabled)
   ...
   mordor:~ # which java
   /opt/ibm/java-i386-60/jre/bin/java
   ```

Установка Encryption Key Manager в Windows

1. Вставьте компакт-диск Dell Encryption Key Manager.

   При установке все содержимое (документация, файлы GUI-интерфейса, файлы свойств конфигурации), соответствующее операционной системе, копируется с компакт-диска на жесткий диск компьютера. Во время установки система

проверяется на наличие правильной версии среды IBM Java Runtime Environment. Если такая версия не обнаружена, она автоматически устанавливается.

По окончании установки запускается графический пользовательский интерфейс (GUI).

2. Когда откроется мастер InstallShield, нажмите **"Далее"**.
3. Прочитайте лицензионное соглашение и нажмите **"Да"**.
4. Когда откроется окно выбора каталога для установки (рис. 3-1), выберите каталог и запомните его путь. Информация о пути к каталогу Java понадобится для запуска Encryption Key Manager.
   Нажмите **"Далее"**.

Рисунок 3-1. Окно выбора каталога для установки

5. Откроется окно с вопросом, желаете ли вы использовать данную среду выполнения Java Runtime Environment в качестве JVM по умолчанию (рис. 3-2 на стр. 3-4).

Рисунок 3-2. Сделать эту версию JVM версией по умолчанию

Нажмите **"Нет"**.

6. Откроется окно Start Copying Files (Начать копирование файлов) (рис. 3-3). Обязательно запомните целевой каталог.
   Нажмите **"Далее"**.

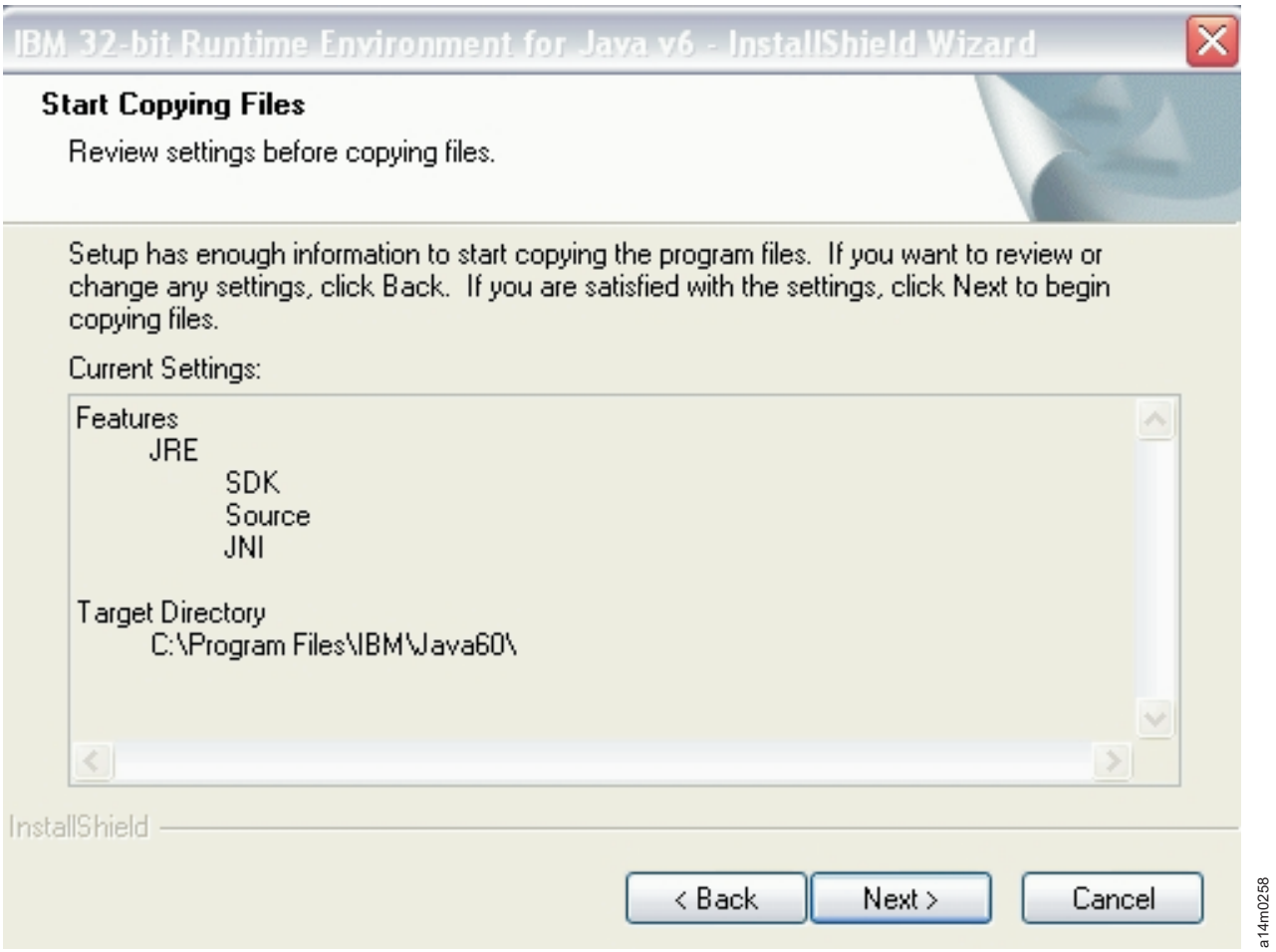

Рисунок 3-3. Окно начала копирования файлов

7. Окно состояния показывает ход выполнения процесса установки.
8. Откроется окно регистрации браузера. Выберите браузер, который будет использоваться с Encryption Key Manager. Нажмите **"Далее"**.
9. Когда откроется окно завершения установки с помощью мастера InstallShield, нажмите **"Готово"**.

   После установки можно открыть командное окно, чтобы запросить номер установленной версии Java:

```
C:\WinEKM>C:\"Program Files"\IBM\Java60\jre\bin\java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pwi3260sr5-20090529_04(SR5))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Windows Server 2003 x86-32 j9vmwi3223-20090
519_35743 (JIT enabled, AOT enabled)
...
```

10. Измените переменную PATH следующим образом (необходимо для Encryption Key Manager 2.1, но необязательно для версий с датой сборки 05.03.2007 и более ранних).

    Если вы планируете вызывать Java SDK из командного окна, можно настроить переменную PATH, чтобы иметь возможность запускать исполняемые файлы Java JRE (java.exe) из любого каталога без необходимости вводить полный путь команды. Если не установить переменную PATH, при запуске исполняемого файла потребуется каждый раз задавать полный путь, например:

    ```
    C:>\Program Files\IBM\Java60\jre\bin\java ...
    ```

    Чтобы использовать переменную PATH на постоянной основе (обязательно для Encryption Key Manager 2.1), добавьте в нее полный путь к каталогу java bin. Обычно полный путь выглядит следующим образом:

    ```
    C:\Program Files\IBM\Java60\jre\bin
    ```

    Чтобы задать постоянное значение PATH в ОС Microsoft Windows 2003, 2008 и 2008 R2, выполните следующие действия.

    Примечание: Установка переменной PATH из командной строки невозможна.

    a. В меню Пуск выберите пункт **Настройка**, а затем **Панель управления**.
    b. Дважды щелкните по значку **Система**.
    c. Перейдите на вкладку **Дополнительно**.
    d. Нажмите кнопку **Переменные среды**.
    e. Выберите в списке системных переменных переменную Path и нажмите кнопку **Изменить**.
    f. Добавьте в начало переменной Path путь к IBM JVM.

       Каталог установки по умолчанию — C:\PROGRA~1\IBM\Java60\jre\bin.

       ВАЖНО: добавьте в конец пути точку с запятой (;), чтобы отделить его от других каталогов в списке путей.
    g. Нажмите кнопку **OK**.

Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя

До запуска Encryption Key Manager следует создать как минимум одно хранилище ключей и хотя бы один самозаверяющий сертификат. Для создания файла свойств конфигурации, ключей, сертификатов и хранилища ключей Encryption Key Manager можно использовать графический интерфейс пользователя (GUI) сервера Dell Encryption Key Manager. В результате будет также создан простой файл свойств конфигурации для интерфейса командной строки.

1. Если графический интерфейс пользователя еще не открыт, откройте его.

   В Windows
   перейдите в каталог c:\ekm\gui и щелкните кнопкой мыши по файлу **LaunchEKMGui.bat**

   На платформах Linux
   перейдите в каталог /var/ekm/gui и запустите файл `../LaunchEKMGui.sh`

2. Выберите **EKM Configuration** (Конфигурация EKM) в навигационном меню с левой стороны окна GUI-интерфейса.
3. На странице “EKM Server Configuration (Настройка сервера EKM)” (рис. 3-4) введите данные во все обязательные поля (отмечены "звездочкой" (*)). Некоторые поля будут заполнены автоматически для вашего удобства. Для получения описания какого-либо поля щелкните по знаку вопроса справа от него. Нажмите **Next**.

Примечание: После задания пароля для хранилища ключей **не изменяйте его** до тех пор, пока не будет нарушена безопасность хранилища. Для предотвращения потенциального нарушения безопасности пароли скрываются. Чтобы изменить пароль хранилища ключей, необходимо изменить каждый пароль в этом хранилище отдельно с помощью команды **keytool**. См. раздел “Изменение паролей хранилищ ключей” на стр. 3-13.

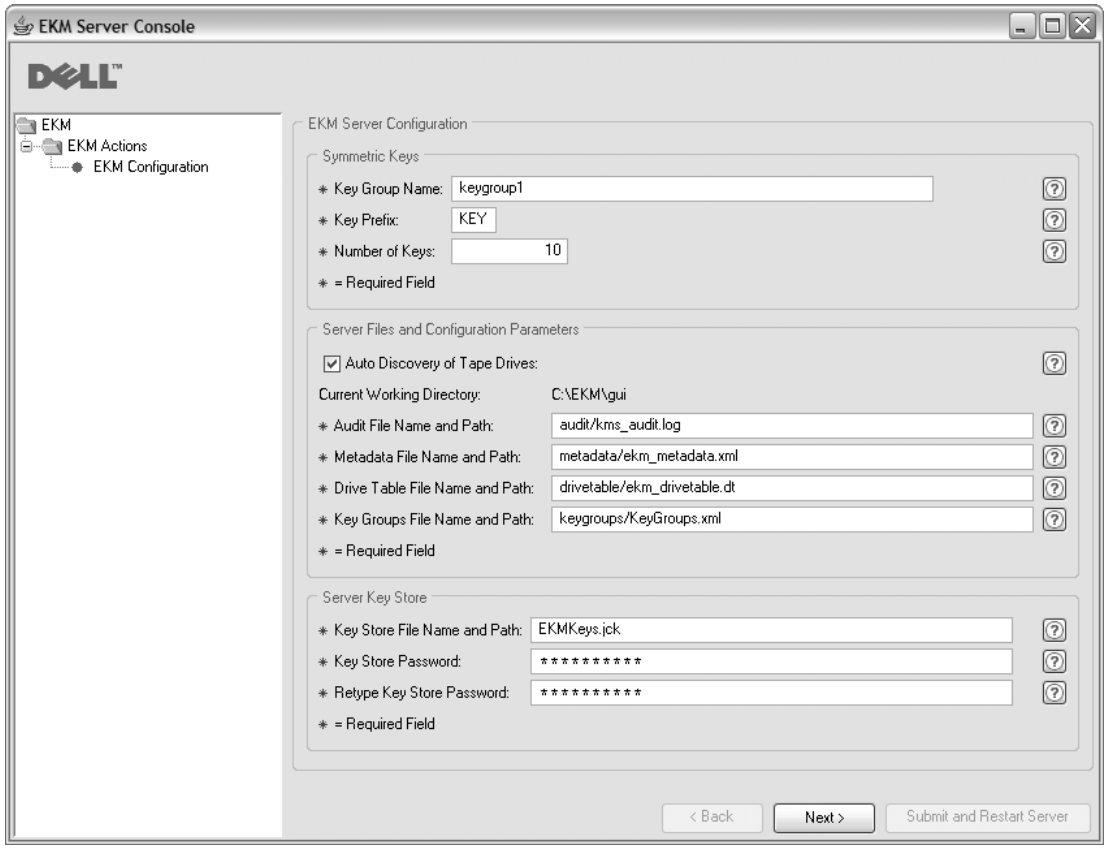

Рисунок 3-4. Страница EKM Server Configuration (Настройка сервера EKM)

Хотя число ключей, которое можно сгенерировать для хранилища ключей Dell Encryption Key Manager, не ограничено, с ростом числа ключей в запросе увеличивается время их генерации. Приложению Encryption Key Manager нужно 15 секунд для генерации 10 ключей и более 30 минут для генерации 10 000 ключей. Следует учитывать, что число ключей ограничивается ресурсами хост-сервера (объемом памяти на сервере). Во время работы приложение Encryption Key Manager размещает список хранилищ ключей в системной памяти, чтобы обеспечить быстрый доступ к ключам, когда библиотека отправляет от накопителя запрос о ключе.

Примечание: Если во время генерации ключа работа графического пользовательского интерфейса Encryption Key Manager была прервана, требуется повторная установка Encryption Key Manager.

Если процесс генерации ключа приложением Encryption Key Manager остановлен до его завершения, файл хранилища ключей будет поврежден. Для восстановления в этом случае выполните следующие действия:

- Если была прервана первоначальная установка Encryption Key Manager, перейдите в каталог, в который производилась установка (например, x:\ekm). Удалите каталог и повторно запустите установку.
- Если работа приложения Encryption Key Manager была прервана при добавлении новой группы ключей, остановите сервер Encryption Key Manager и восстановите файл хранилища ключей при помощи последней резервной копии хранилища ключей (этот файл находится в каталоге x:\ekm\gui\backupfiles). Следует учесть, что имя файла резервной копии содержит метку даты и времени его создания (например, 2007_11_19_16_38_31_EKMKeys.jck). После копирования файла в каталог x:\ekm\gui метку даты и времени из имени файла необходимо удалить. Перезапустите сервер Encryption Key Manager и повторите ранее прерванный процесс добавления группы ключей.

4. На странице “EKM Server Certificate Configuration (Конфигурация сертификатов сервера EKM)” (рис. 3-5 на стр. 3-8) введите псевдоним хранилища ключей и любые дополнительные данные по желанию. Щелкните по **Submit and Restart Server** (Применить и перезапустить сервер).

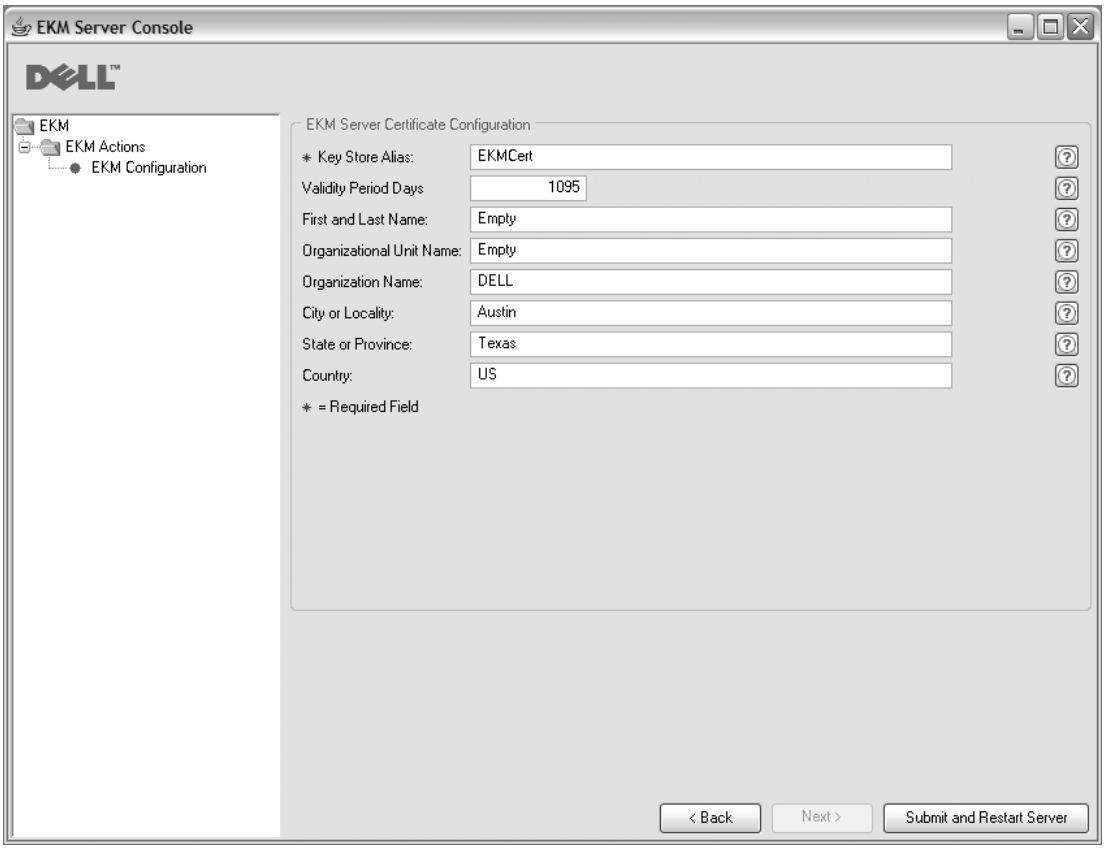

Рисунок 3-5. Страница EKM Server Certificate Configuration (Настройка сертификата сервера EKM)

5. Откроется окно “Backup Critical Files (Создание резервных копий важных файлов)” (рис. 3-6 на стр. 3-9), приглашающее создать резервные копии файлов данных Encryption Key Manager.

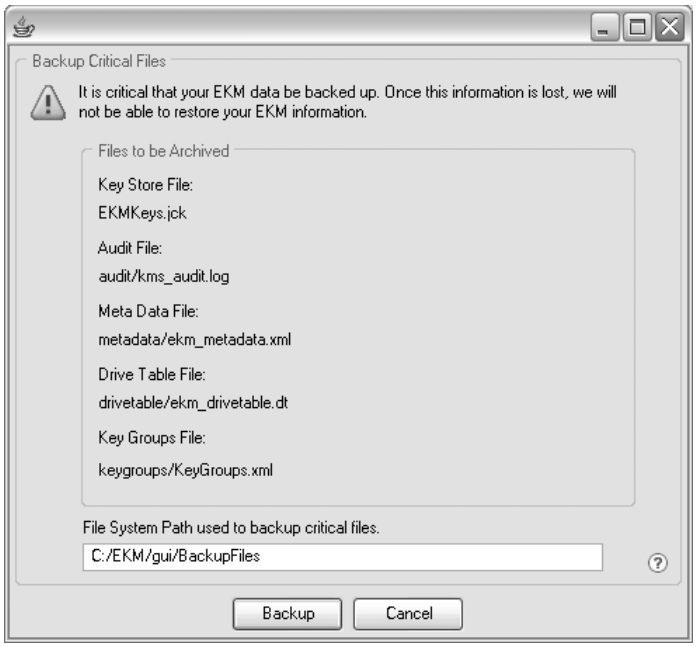

Рисунок 3-6. Окно Backup Critical Files (Создание резервных копий важных файлов)

Проверьте введенный путь и нажмите кнопку **Backup (Создать резервную копию)**. Сервер Dell Encryption Key Manager запускается в фоновом режиме.

Приложение Encryption Key Manager создает набор файлов резервных копий каждый раз, когда после изменения конфигурации сервера Encryption Key Manager нажимается кнопка **OK** либо в окне “Backup Critical Files (Создание резервных копий важных файлов)” нажимается кнопка **Backup (Создать резервную копию)**. Файлы, перечисленные в списке *Файлы для архивирования*, сохраняются в каталоге c:/ekm/gui/BackupFiles. В начало имени файла добавляются дата и время. Например, если резервное копирование набора файлов было выполнено 26 ноября 2007 года в 2 часа 58 минут и 46 секунд после полудня, все имена его файлов будут начинаться с метки даты и времени “2007_11_26_14_58_46_ИмяФайла. Файлы резервных копий не перезаписываются.

6. Выберите **Server Health Monitor** (Монитор работоспособности сервера) в навигаторе графического пользовательского интерфейса и убедитесь, что сервер Encryption Key Manager работает.

Инструкции по добавлению ключей к существующему хранилищу ключей приведены в разделе “Определение групп ключей и создание ключей с помощью графического интерфейса пользователя” на стр. 3-16.

Как определить правильный IP-адрес хоста:

Ограничения текущего графического пользовательского интерфейса Encryption Key Manager могут не позволить отобразить IP-адрес хоста в мониторе работоспособности сервера:

- Если хост настроен на IPV6-адреса, приложение Encryption Key Manager не сможет отобразить IP-адрес.
- Если приложение Encryption Key Manager установлено в системе Linux, оно отображает адрес localhost, а не фактический активный порт IP.

1. Чтобы получить фактический IP-адрес системы хоста, определите адрес порта IP при помощи конфигурации сети.

- В ОС Windows откройте командное окно и введите `ipconfig`.
- В Linux введите `isconfig`.

Как идентифицировать порт SSL в EKM

1. Запустите сервер Encryption Key Manager из командной строки.
   - В Windows перейдите в каталог `c:\ekm` и запустите файл **startServer.bat**
   - На платформах Linux перейдите в каталог `/var/ekm` и введите `startServer.sh`
   - Дополнительные сведения см. в разделе “Запуск, обновление и остановка сервера диспетчера ключей” на стр. 5-1.
2. Запустите CLI-клиент из командной строки.
   - В Windows перейдите в каталог `c:\ekm` и запустите файл **startClient.bat**
   - На платформах Linux перейдите в каталог `/var/ekm` и введите `startClient.sh`
   - Дополнительные сведения см. в разделе “Клиент интерфейса командной строки” на стр. 5-5.
3. Войдите в систему клиента CLI на сервере Encryption Key Manager при помощи следующей команды:

   `login -ekmuser` *`userID`* `-ekmpassword` *`password`*

   где *userID* = EKMAdmin и *password* = changeME (Это пароль по умолчанию. Если пароль по умолчанию ранее был изменен, используйте новый пароль.)

   После успешного входа в систему отображается сообщение `User successfully logged in (Пользователь успешно вошел в систему)`.
4. Идентифицируйте порт SSL, выполнив следующую команду:

   `status`

   Отображаемый результат должен быть подобен следующему: `server is running. TCP port: 3801, SSL port: 443`.

   Запомните номер порта, настроенного на протокол SSL, и убедитесь, что именно этот порт использован для настройки ваших параметров шифрования, управляемого библиотекой.
5. Выйдите из системы при помощи командной строки. Введите следующую команду:

   `exit`

   Закройте командное окно.

Создание ключей и псевдонимов для шифрования на накопителях LTO 4 и LTO 5

Графический интерфейс пользователя сервера Dell Encryption Key Manager обеспечивает наиболее простой способ создания симметричных ключей шифрования (см. раздел “Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя” на стр. 3-5). Для создания симметричных ключей шифрования можно также использовать средство Keytool. Средство Keytool специально предназначено для импорта и экспорта ключей между различными хранилищами ключей. Подробные сведения см. в разделах “Импорт ключей шифрования данных с помощью команды keytool -importseckey ” на стр. 3-14 и “Экспорт ключей шифрования данных с помощью команды keytool -exportseckey ” на стр. 3-14.

Keytool — это утилита для управления ключами, сертификатами и псевдонимами. С его помощью можно создавать, импортировать и экспортировать ключи шифрования и сохранять их в хранилище ключей.

Доступ к каждому ключу шифрования данных в хранилище ключей можно получить с помощью уникального псевдонима. Псевдоним - это строка символов, например, `123456tape`. При работе с хранилищами ключей формата JCEKS строка `123456Tape` эквивалентна строке `123456tape` и позволяет получить доступ к той же записи в хранилище ключей. При использовании команды **keytool -genseckey** для создания ключа шифрования данных в этой же команде указывается соответствующий псевдоним. Псевдоним помогает найти правильный ключ в правильной группе ключей и хранилище ключей, который будет использоваться во время считывания зашифрованных данных с ленты LTO 4 и LTO 5 и записи на нее.

Примечание: Индивидуальные псевдонимы и диапазоны псевдонимов должны быть уникальными. Это обеспечивается при создании ключей в заданном экземпляре хранилища ключей или Encryption Key Manager. Тем не менее, при использовании нескольких приложений Encryption Key Manager или хранилищ ключей необходимо придерживаться соглашения об именах. Это поможет обеспечить уникальность имен в нескольких экземплярах в случае, если потребуется копировать ключи из одного экземпляра в другой, сохраняя уникальность ссылок.

После создания ключей и псевдонимов необходимо обновить значение свойства symmetricKeySet в файле KeyManagerConfig.properties, чтобы задать новый псевдоним, диапазон псевдонимов, идентификатор группы ключей, имя файла, в котором хранятся симметричные ключи, и имя файла, в котором определены группы ключей. (Дополнительные сведения см. в разделе “Создание групп ключей и управление ими” на стр. 3-16.) Действительными будут признаны только те ключи, которые определены в свойстве symmetricKeySet. (Они будут проверены на предмет действительности псевдонима и допустимости размера и алгоритма симметричного ключа). Если в этом свойстве будет указан недействительный ключ, диспетчер ключей не запустится и будет создан протокол аудита.

Средство Keytool можно также использовать для экспорта ключей шифрования данных в другие хранилища ключей и импорта из них. Ниже приведены общие сведения о данных операциях. Чтобы вывести на экран все параметры, связанные с диспетчером ключей и упомянутые ниже, используйте команду **keytool -ekmhelp**.

Редактирование файлов свойств конфигурации

Чтобы изменить файл KeyManagerConfig.properties или ClientKeyManagerConfig.properties, выполните следующие действия:

1. Остановите сервер Encryption Key Manager.
2. При помощи текстового редактора по вашему выбору откройте файл KeyManagerConfig.properties, чтобы изменить конфигурацию сервера, или файл ClientKeyManagerConfig.properties для изменения конфигурации клиента. Нельзя редактировать файл для компьютера с ОС Linux средствами Windows из-за добавляемого конечного символа строки ^M. При использовании Windows редактируйте файл с помощью текстового редактора gvim/vim.
3. Измените значения свойств в соответствии с указаниями, приведенными в данном документе.
4. Сохраните файл.
5. Повторно запустите сервер Encryption Key Manager.

Если Keytool не используется

Если для создания ключей и псевдонимов не используется keytool или графический интерфейс, то создание диапазонов ключей, совместимых с Encryption Key Manager, невозможно. Для создания отдельных ключей, совместимых с Encryption Key Manager, необходимо указывать псевдонимы с использованием одного из следующих форматов:

- не более 12 печатных символов (например, `abcdefghijk`);
- 3 печатных символа, за которыми следуют два нуля, затем 16 шестнадцатеричных символов (например, `ABC00000000000000001`). Всего должен быть ровно 21 символ.

Создание ключей шифрования данных и псевдонимов с помощью команды keytool -genseckey

Примечание: Перед первым использованием команды **keytool** во время любого сеанса необходимо задать правильную среду с помощью сценария `updatePath`.

В Windows
перейдите в каталог c:\ekm и щелкните по файлу **updatePath.bat**

На платформах Linux
перейдите в каталог /var/ekm и запустите файл `. ./updatePath.sh`

Утилита Keytool создает псевдонимы и симметричные ключи, предназначенные для шифрования данных на накопителях LTO 4 и LTO 5 с использованием магнитных лент LTO 4 и LTO 5. Используйте команду **keytool -genseckey** для создания одного или нескольких засекреченных ключей и сохранения их в указанном хранилище ключей. В команде **keytool -genseckey** используются следующие параметры:

```
-genseckey      [-v] [-protected]
                 [-alias <псевдоним> | aliasrange <диапазон_псевдонимов>]
                 [-keypass <пароль_ключа>]
                 [-keyalg <алгоритм_ключа>] [-keysize <размер_ключа>]
                 [-keystore <хранилище_ключей>] [-storepass <пароль_хранилища>]
                 [-storetype <тип_хранилища>] [-providerName <имя>]
                 [-providerClass <класс_провайдера> [-providerArg <аргумент>] ...
                 [-providerPath <список_путей>]
```

При создании ключей, которые Encryption Key Manager использует для шифрования данных накопителей на магнитной ленте LTO 4 и LTO 5, особенно важны следующие параметры:

-alias
Укажите значение переменной *псевдоним* для одного ключа шифрования данных. Можно использовать до 12 печатных символов (например, `abcfrg` или `key123tape`).

-aliasrange
При создании нескольких ключей шифрования данных значение переменной *диапазон_псевдонимов* должно состоять из буквенного префикса (3 символа) и следующих за ним верхней и нижней границ для ряда 16-символьных (шестнадцатиричных) строк с начальными нулями, которые вставляются автоматически, образуя псевдонимы длиной 21 символ. Например, задание значения `key1-a` приведет к созданию последовательности псевдонимов от `KEY000000000000000001` до `KEY00000000000000000A`. Если переменной

диапазон_псевдонимов присвоено значение `xyz01-FF`, будут созданы псевдонимы от `XYZ000000000000000001` до `XYZ0000000000000000FF`, что, в свою очередь, приведет к созданию 255 симметричных ключей.

-keypass
Задайте пароль для защиты ключа шифрования данных. Этот пароль **должен совпадать** с паролем хранилища ключей. Если пароль не указан, появится приглашение на ввод пароля. Если при появлении приглашения нажать клавишу **Enter**, ключу будет присвоен то же пароль, который используется для хранилища ключей. Значение переменной *пароль_ключа* должно состоять из не менее чем шести символов.

Примечание: После задания пароля для хранилища ключей **не изменяйте его** до тех пор, пока не будет нарушена безопасность хранилища. См. раздел “Изменение паролей хранилищ ключей”.

-keyalg
Задайте алгоритм, который будет использоваться при создании ключа шифрования данных. В качестве значения этого параметра необходимо указать значение `AES`.

-keysize
Задайте размер создаваемого ключа. Для размера ключа необходимо указать значение `256`.

Примеры допустимых псевдонимов, которые могут быть связаны с симметричными ключами:

```
abc000000000000000001
abc00a0120fa000000001
```

Примеры псевдонимов, которые будут отклонены диспетчером ключей:

```
abcefghij1234567 ? недопустимая длина
abcg0000000000000001 ? длина префикса превышает 3 символа
```

Если псевдоним уже существует в хранилище, средство Keytool генерирует исключительную ситуацию и прекращает работу.

Изменение паролей хранилищ ключей

Примечание: После задания пароля для хранилища ключей **не изменяйте его** до тех пор, пока не будет нарушена безопасность хранилища. Для предотвращения потенциального нарушения безопасности пароли скрываются. Чтобы изменить пароль хранилища ключей, необходимо изменить пароль каждого ключа в этом хранилище отдельно с помощью следующей команды **keytool**.

Чтобы изменить пароль хранилища данных, введите:

```
keytool -keypasswd  -keypass старый_пароль -new новый_пароль -alias псевдоним
        -keystore имя_хранилища_ключей -storetype тип_хранилища_ключей
```

Для изменения пароля хранилища ключей в файле KeyManagerConfig.properties необходимо изменить значение каждого свойства конфигурации сервера, в котором он указан, используя один из следующих методов.

- Полностью удалите скрытый пароль и разрешите приложению Encryption Key Manager при его следующем запуске выдать приглашение на ввод нового пароля.
- Удалите скрытый пароль полностью и введите новый пароль в открытом виде. При следующем запуске EKM пароль будет скрыт.

Импорт ключей шифрования данных с помощью команды keytool -importseckey

Для импорта засекреченного ключа или набора засекреченных ключей из файла импорта используйте команду keytool -importseckey. В команде **keytool -importseckey** можно использовать следующие параметры:

-importseckey [-v]
[-keyalias <псевдоним_ключа>] [-keypass <пароль_ключа>]
[-keystore <хранилище_ключей>] [-storepass <пароль_хранилища>]
[-storetype <тип_хранилища>] [-providerName <имя>]
[-importfile <файл_импорта>] [-providerClass <класс_провайдера>]
[providerArg <аргумент>]

При импорте ключей шифрования данных, которые Encryption Key Manager использует для шифрования данных на ленточных накопителях LTO 4 и LTO 5, особенно важны следующие параметры:

-keyalias
Задайте псевдоним секретного ключа, находящегося в хранилище, который используется для расшифровки всех ключей шифрования данных в файле импорта, указанном переменной *файл_импорта*.

-importfile
Укажите файл, содержащий ключи шифрования данных, которые необходимо импортировать.

Экспорт ключей шифрования данных с помощью команды keytool -exportseckey

Для экспорта засекреченного ключа или набора засекреченных ключей в файл экспорта используйте команду keytool -exportseckey. В команде **keytool -exportseckey** можно использовать следующие параметры:

-exportseckey [-v]
[-alias <псевдоним> | aliasrange <диапазон_псевдонимов>]
[-keyalias <псевдоним_ключа>]
[-keystore <хранилище_ключей>] [-storepass <пароль_хранилища>]
[-storetype <тип_хранилища>] [-providerName <имя>]
[-exportfile <файл_экспорта>] [-providerClass <класс_провайдера>]
[providerArg <аргумент>]

При экспорте ключей шифрования данных, которые Encryption Key Manager использует для шифрования данных на ленточных накопителях LTO 4 и LTO 5, особенно важны следующие параметры:

-alias
Укажите значение переменной *псевдоним* для одного ключа шифрования данных. Можно использовать до 12 печатных символов (например, `abcfrg` или `key123tape`).

-aliasrange
При экспорте нескольких ключей шифрования данных значение переменной *диапазон_псевдонимов* должно состоять из буквенного префикса (3 символа) и следующих за ним верхней и нижней границ для ряда 16-символьных (шестнадцатиричных) строк с начальными нулями, которые вставляются автоматически, образуя псевдонимы длиной 21 символ. Например, задание значения `key1-a` приведет к созданию последовательности псевдонимов от

KEY000000000000000001 до KEY00000000000000000A. Если переменной *диапазон_псевдонимов* присвоено значение xyz01-FF, будут созданы псевдонимы от XYZ000000000000000001 до XYZ0000000000000000FF.

-exportfile
Задайте файл, в котором будут храниться ключи шифрования данных после экспорта.

-keyalias
Задайте псевдоним открытого ключа, находящегося в хранилище, который используется для шифрования всех ключей шифрования данных. Убедитесь, что хранилище ключей, в которое будут импортированы симметричные ключи (ключи шифрования данных), содержит соответствующий секретный ключ.

Пример создания псевдонима и симметричного ключа для шифрования на накопителях LTO 4 и LTO 5 при использовании хранилища ключей JCEKS

Запустите средство **KeyTool** с параметром -aliasrange.

Обратите внимание, что в качестве алгоритма ключа (параметр -keyalg) необходимо указать значение AES, а в качестве размера ключа (параметр –keysize) - значение 256, как показано ниже:

```
/bin/keytool -genseckey -v -aliasrange AES01-FF -keyalg AES -keysize 256
-keypass пароль -storetype jceks -keystore путь/имя_файла.jceks
```

Вызов средства KeyTool с использованием этих команд приводит к созданию 255 последовательных псевдонимов в диапазоне от AES000000000000000001 до AES0000000000000000FF и связанных с ними 256-разрядных симметричных AES-ключей. Каждую команду можно выполнить столько раз, сколько необходимо для создания полного набора последовательных и автономных псевдонимов ключей и обеспечения надежной работы диспетчера ключей. Например, для создания дополнительного псевдонима и симметричного ключа для LTO 4 и LTO 5

```
/bin/keytool -genseckey -v -alias abcfrg -keyalg AES -keysize 256
-keypass пароль -storetype jceks -keystore путь/имя_файла.jceks
```

В результате выполнения этой команды в заданное хранилище ключей, которое уже содержит 255 псевдонимов, созданных в результате выполнения указанной выше команды, будет добавлен автономный псевдоним abcfrg. При этом файл jceks, указанный в параметре –keystore, будет содержать 256 симметричных ключей.

Обновите значение свойства symmetricKeySet в файле KeyManagerConfig.properties, чтобы добавить имя файла, в котором сохранены симметричные ключи, и следующую строку, обеспечивая тем самым соответствие всем диапазонам псевдонимов, указанным выше. Обратите внимание, что при указании недействительного псевдонима Encryption Key Manager может не запуститься. Кроме того, сбой проверки достоверности может произойти при указании неправильного размера ключа (для алгоритма AES размер ключа ДОЛЖЕН быть равен 256) или неправильного алгоритма, не соответствующего платформе. Параметр -keyalg должен иметь значение AES, а параметр -keysize - значение 256. Имя файла, указанное в свойстве **config.keystore.file**, должно совпадать с именем файла, указанным в переменной <имя_файла> параметра –keystore при вызове средства KeyTool:

```
symmetricKeySet = AES01-FF,abcfrg
config.keystore.file = <имя_файла>.jceks
```

Действительными будут признаны только те ключи, которые определены в свойстве symmetricKeySet. Они будут проверены на предмет действительности псевдонима и допустимости размера и алгоритма симметричного ключа. Если в этом свойстве будет указан недействительный ключ, Encryption Key Manager не запустится и будет создан протокол аудита.

Создание групп ключей и управление ими

Encryption Key Manager позволяет группировать симметричные ключи для шифрования LTO 4 и LTO 5. Таким образом, можно упорядочить ключи в соответствии с типами данных, для шифрования которых они используются, в соответствии с именами пользователей, которые имеют доступ к этим ключам, или в соответствии с любыми другими значимыми характеристиками. После создания группы ключей ее можно связать с заданным накопителем на магнитной ленте с помощью ключевого слова -symrec в команде **adddrive**. Синтаксис команды см. в разделе “adddrive” на стр. 5-8.

Чтобы создать группу ключей, необходимо определить ее в файле KeyGroups.xml. В случае выполнения процедуры, описанной в разделе “Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя” на стр. 3-5, расположение этого файла указывается на странице конфигурации EKM. Если файл конфигурации создается вручную, расположение файла KeyGroups.xml указывается в файле свойств конфигурации следующим образом:

```
config.keygroup.xml.file = FILE:KeyGroups.xml
```

Если значение этого параметра не задано, то по умолчанию используется файл KeyGroups.xml, расположенный в рабочем каталоге, из которого запускается Encryption Key Manager. Если этот файл не существует, будет создан пустой файл KeyGroups.xml. При последующих запусках сервера Encryption Key Manager в журнале **native_stderr.log** может появиться следующее сообщение: `[Fatal Error] :-1:-1: Premature end of file.` Это ошибка синтаксического анализа пустого файла KeyGroups.xml, которая не препятствует запуску сервера Encryption Key Manager, если сервер Encryption Key Manager не настроен на использование групп ключей.

Группы ключей создаются с помощью графического интерфейса пользователя сервера Dell Encryption Key Manager или с помощью следующих команд клиента CLI (их синтаксис см. в разделе “Команды CLI” на стр. 5-8).

Определение групп ключей и создание ключей с помощью графического интерфейса пользователя

С помощью графического интерфейса пользователя можно выполнять все задачи, связанные с управлением группами ключей. Этот интерфейс можно также использовать для создания дополнительных ключей.

Примечание: Если при выполнении любого из описанных ниже действий нажать кнопку **Submit Changes** (Применить изменения), откроется диалоговое окно создания резервной копии (рис. 3-6 на стр. 3-9) с напоминанием о необходимости резервного копирования файлов данных Encryption Key Manager. Укажите каталог для сохранения резервных копий данных. Нажмите кнопку **Submit** (Применить). Проверьте путь к каталогу для сохранения резервных копий и нажмите кнопку **OK**.

Чтобы создать группу ключей и добавить в нее ключи или чтобы добавить ключи в существующую группу ключей, выполните следующие действия:

1. Если графический интерфейс пользователя еще не открыт, откройте его.

   В Windows
   перейдите в каталог c:\ekm\gui и щелкните кнопкой мыши по файлу **LaunchEKMGui.bat**

   На платформах Linux
   перейдите в каталог /var/ekm/gui и запустите файл `../LaunchEKMGui.sh`
2. Выберите **Administration Commands** (Команды администрирования) в навигационном меню с левой стороны окна GUI-интерфейса.
3. В нижней части окна выберите команду **Create a Group of Keys** (Создать группу ключей) (рис. 3-7).

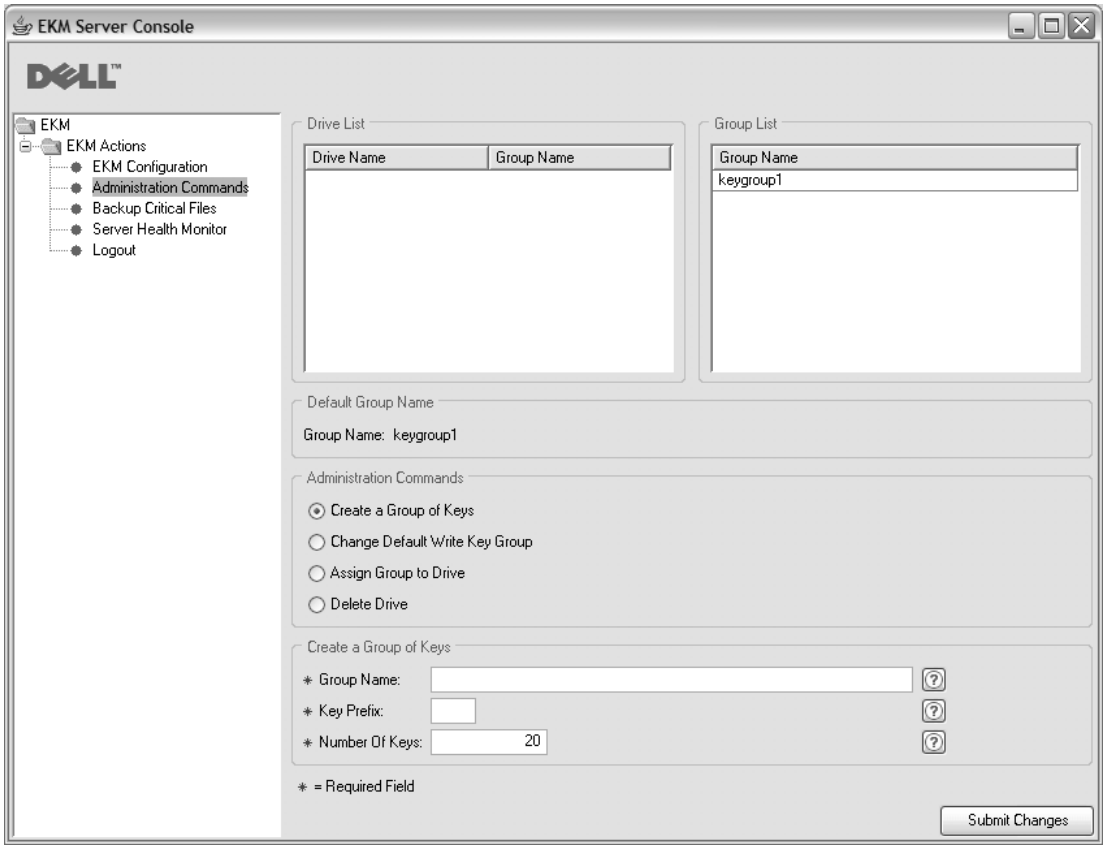

Рисунок 3-7. Create a Group of Keys (Создание группы ключей)

4. Введите имя группы ключей, префикс, который будет использоваться для псевдонимов ключей, а также количество ключей в группе. Нажмите кнопку **Submit Changes** (Применить изменения).

Чтобы изменить группу ключей, заданную по умолчанию, выполните следующие действия.

1. Выберите **Administration Commands** (Команды администрирования) в навигационном меню с левой стороны окна GUI-интерфейса.
2. В нижней части окна выберите команду **Change Default Write Key Group** (Изменить группу ключей, заданную по умолчанию) (рис. 3-8 на стр. 3-18).

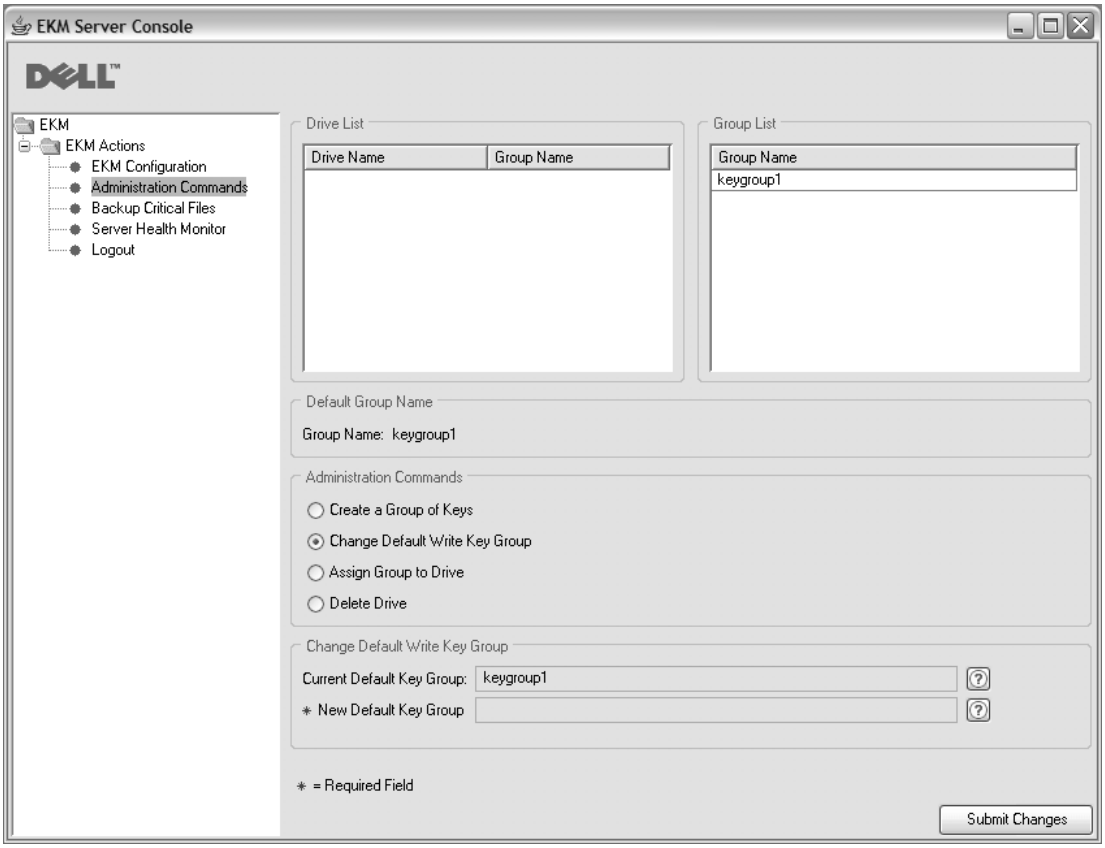

Рисунок 3-8. Change Default Write Key Group (Изменение группы ключей, заданной по умолчанию)

3. В правой части окна в области Group List (Список групп) выберите новую группу ключей, которая будет использоваться в качестве группы ключей по умолчанию.
4. В нижней части окна проверьте текущую группу ключей, выбранную по умолчанию, а также новую группу ключей и нажмите кнопку **Submit Changes** (Применить изменения).

Чтобы связать определенную группу ключей с заданным накопителем на магнитной ленте, выполните следующие действия.

1. Выберите **Administration Commands** (Команды администрирования) в навигационном меню с левой стороны окна GUI-интерфейса.
2. В нижней части окна выберите команду **Assign a Group to Drive** (Связать группу ключей с накопителем) (рис. 3-9 на стр. 3-19).

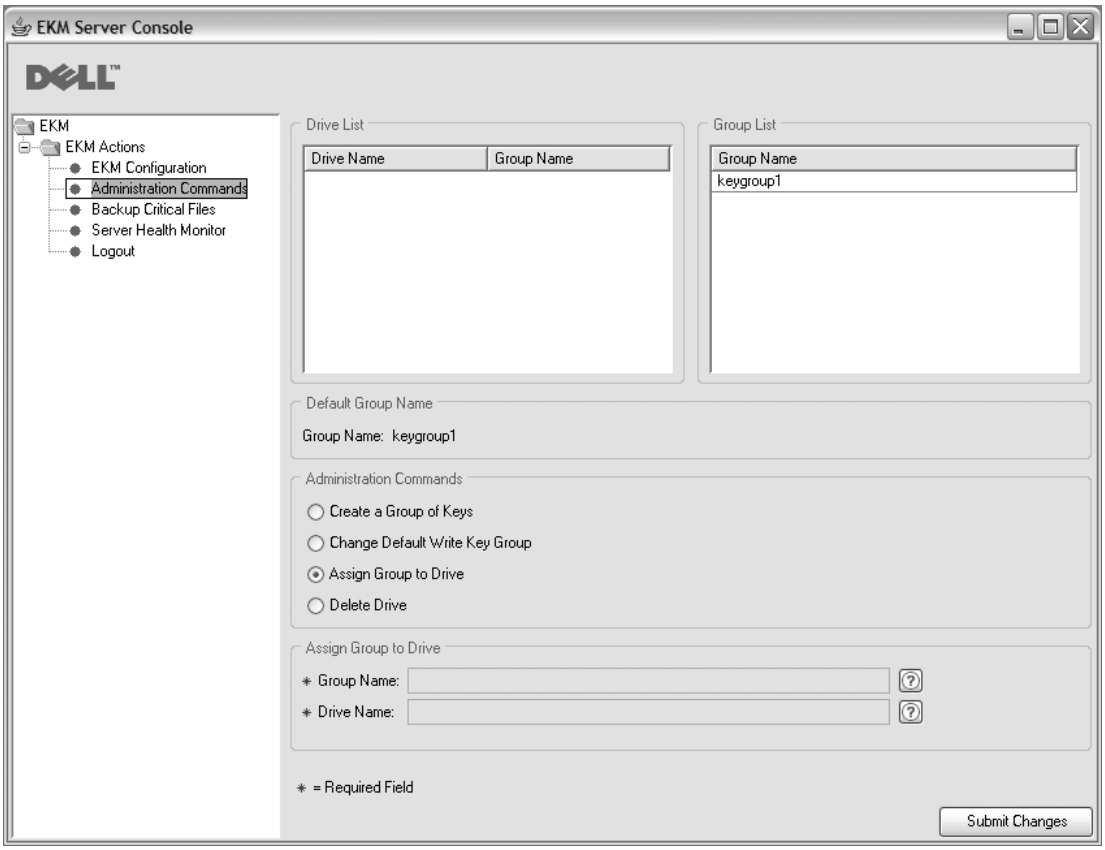

Рисунок 3-9. Assign Group to Drive (Связывание группы ключей с накопителем)

3. В списке Drive List (Список накопителей) выберите накопитель на магнитной ленте.
4. В списке Group List (Список групп) выберите группу ключей.
5. В нижней части окна проверьте данные о выбранном накопителе и группе ключей и нажмите кнопку **Submit Changes** (Применить изменения).

Чтобы удалить накопитель на магнитной ленте из таблицы накопителей, выполните следующие действия:

1. Выберите **Administration Commands** (Команды администрирования) в навигационном меню с левой стороны окна GUI-интерфейса.
2. В нижней части окна выберите команду **Delete Drive** (Удалить накопитель) (рис. 3-10 на стр. 3-20).

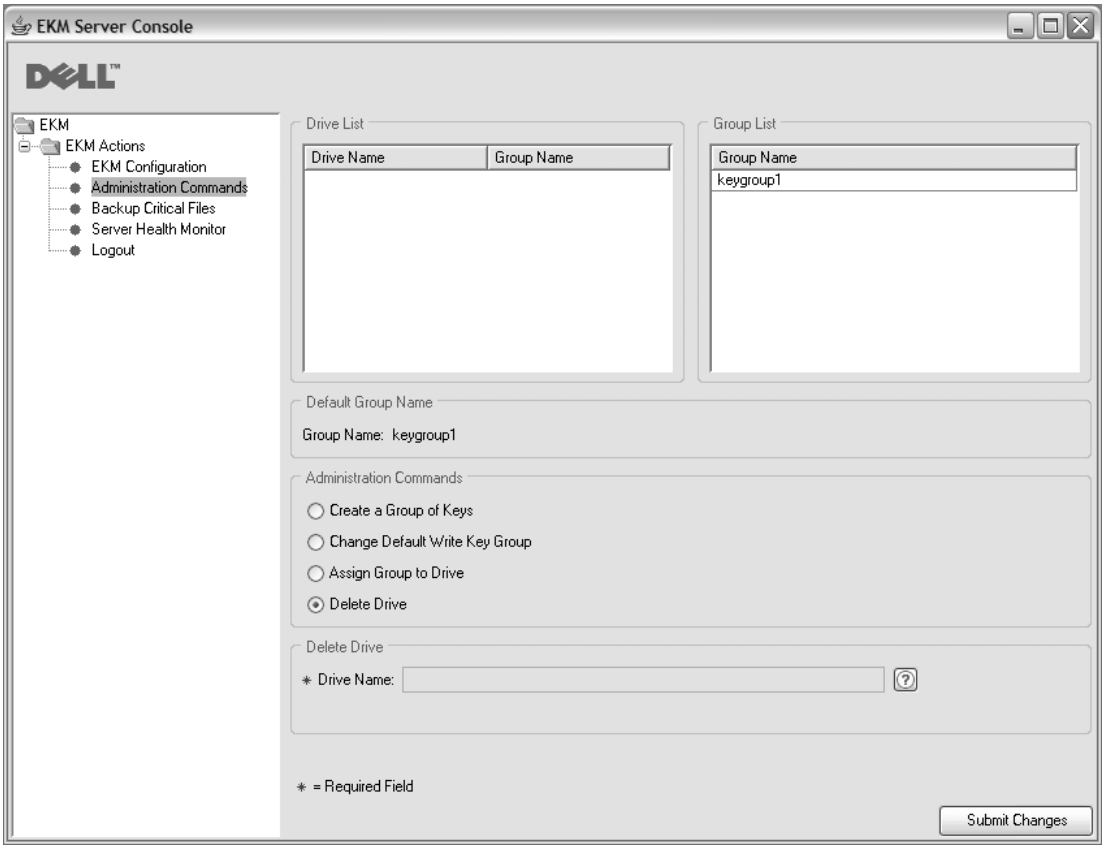

Рисунок 3-10. Удаление накопителя

3. В списке Drive List (Список накопителей) выберите накопитель на магнитной ленте.
4. В нижней части окна проверьте имя накопителя и нажмите кнопку **Submit Changes** (Применить изменения).

Использование команд CLI для определения групп ключей

Приложение Encryption Key Manager позволяет объединять наборы ключей в группы.

После установки и настройки приложения Encryption Key Manager (когда созданы ключи и хранилища ключей) и запуска сервера Encryption Key Manager подключитесь к нему при помощи клиента и выполните следующие действия:

1. Выполните команду **createkeygroup**.

   С помощью этой команды можно создать первый объект группы ключей в файле KeyGroups.xml. Эту команду следует запускать только один раз.

   Синтаксис: **createkeygroup -password** *пароль*

   -password
   Пароль, который используется для шифрования пароля хранилища ключей в файле KeyGroups.xml для последующего восстановления. Хранилище ключей шифрует ключ группы ключей, который, в свою очередь, шифрует пароль каждого псевдонима группы ключей. Поэтому все ключи в файле KeyGroups.xml являются защищенными.

   Пример: createkeygroup -password a75xynrd
2. Выполните команду **addkeygroup**.

С помощью этой команды можно создать экземпляр группы ключей с уникальным идентификатором группы в файле KeyGroups.xml.

Синтаксис: **addkeygroup -groupID** *имя_группы*

-groupID
Уникальное имя *имя_группы*, используемое для идентификации группы в файле KeyGroups.xml.

Пример: addkeygroup -groupID группа_ключей_1

3. Выполните команду **addkeygroupalias**.

   С помощью этой команды можно создать новый псевдоним для уже существующего в хранилище ключа, чтобы добавить его к заданному идентификатору группы ключей.

   Синтаксис: **addkeygroupalias -alias** *псевдоним* **-groupID** *имя_группы*

   -alias
   Новый *псевдоним* для ключа. Необходимо указать полное имя ключа; например, для ключа Key00 необходимо указать полное имя key000000000000000000.

   -groupID
   Уникальное имя *имя_группы*, используемое для идентификации группы в файле KeyGroups.xml.

   Пример: addkeygroupalias -alias key000000000000000000 -groupID группа_ключей_1

   Примечание: При использовании этой команды CLI можно одновременно добавить только один ключ. Ее необходимо выполнять для каждого ключа, который должен быть добавлен в группу ключей.

4. Свяжите группу ключей с новым или существующим накопителем на магнитной ленте.

   a. Выполните команду **moddrive** для связывания группы ключей с существующим накопителем на магнитной ленте.

      С помощью этой команды можно изменить информацию о накопителе в таблице накопителей.

      Синтаксис: **moddrive -drivename** *имя_накопителя* **-symrec** *псевдоним*

      -drivename
      Переменная *имя_накопителя* содержит серийный номер накопителя на магнитной ленте.

      -symrec
      Определяет *псевдоним* (симметричного ключа) или имя группы ключей накопителя на магнитной ленте.

      Пример: moddrive -drivename 000123456789 -symrec группа_ключей_1

   b. Выполните команду **adddrive** для добавления накопителя на магнитной ленте в таблицу накопителей и связывания его с группой ключей.

      С помощью этой команды можно добавить накопитель и связать его с заданной группой ключей.

      Синтаксис: **adddrive -drivename** *имя_накопителя* **-symrec** *псевдоним*

      -drivename
      Переменная *имя_накопителя* должна содержать 12-значный серийный номер накопителя, который необходимо добавить.

      Примечание: Перед 10-значным серийным номером нужно добавить два нуля (0), чтобы его длина составила 12 знаков.

-symrec
Определяет *псевдоним* (симметричного ключа) или идентификатор группы ключей для накопителя на магнитной ленте.

Пример: adddrive -drivename 000123456789 -symrec группа_ключей_1

Если у накопителя на магнитной ленте нет псевдонима, для задания группы ключей, используемой по умолчанию, свойству symmetrickeySet в файле свойств конфигурации необходимо присвоить значение идентификатора группы ключей, которая будет использована. Например,

```
symmetricKeySet = группа_ключей_1
```

Идентификатор группы ключей должен совпадать с существующим идентификатором группы, указанным в файле KeyGroups.xml. Если идентификаторы не совпадают, сервер Encryption Key Manager не запустится. Приложение Encryption Key Manager отслеживает использование ключей группы. Если указан действительный идентификатор группы, приложение Encryption Key Manager запоминает последний использовавшийся ключ, а затем выбирает случайный ключ из заданной группы ключей.

Копирование ключей из одной группы в другую

Выполните команду **addaliastogroup**.

С помощью этой команды можно скопировать псевдоним из существующей (исходной) в новую (целевую) группу ключей.

Синтаксис: **addaliastogroup -aliasID** *псевдоним* **-sourceGroupID** *имя_группы* **-targetGroupID** *имя_группы*

-aliasID
Псевдоним ключа, который нужно добавить.

-sourceGroupID
Уникальное имя *имя_группы*, используемое для идентификации группы, из которой необходимо скопировать псевдоним.

-targetGroupID
Уникальное имя *имя_группы*, используемое для идентификации группы, в которую необходимо добавить псевдоним.

Пример: addaliastogroup -aliasID псевдоним -sourceGroupID группа_ключей_1 -targetGroupID группа_ключей_2

Примечание: Ключ доступен в обеих группах ключей.

Глава 4. Конфигурирование диспетчера ключей шифрования (Encryption Key Manager, EKM)

Конфигурирование Encryption Key Manager с помощью графического интерфейса пользователя

Создать файл свойств конфигурации проще всего с помощью графического интерфейса пользователя Dell Encryption Key Manager, следуя процедуре, описанной в разделе “Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя” на стр. 3-5. Если вы уже выполнили описанные действия, то файл конфигурации создан и дополнительного конфигурирования не требуется. В случае, если есть необходимость в использовании дополнительных возможностей конфигурирования Encryption Key Manager, следующая информация может оказаться полезной.

Стратегии конфигурирования

Некоторые параметры конфигурации, содержащиеся в файле KeyManagerConfig.properties, создают ярлыки, об отдельных свойствах которых вам следует знать.

Автоматическое обновление таблицы ленточных накопителей

В файле конфигурации Encryption Key Manager предусмотрена переменная (drive.acceptUnknownDrives), которая, если ей присвоено значение `true`, обеспечивает автоматическое заполнение таблицы ленточных накопителей информацией о новом ленточном накопителе при его подключении к Dell Encryption Key Manager. Таким образом, исчезает необходимость в использовании команды **adddrive** для каждого накопителя на магнитной ленте или библиотеки магнитных лент. В этом режиме не потребуется вводить серийный номер каждого устройства, состоящий из 10 цифр, с помощью клиентских команд CLI. При добавлении новых накопителей происходит проверка идентификационной информации накопителя на магнитной ленте, включающая в себя стандартный криптографический обмен открытыми и секретными ключами. После завершения этой проверки новое устройство может считывать информацию с существующих накопителей на магнитной ленте с использованием хранящихся в них идентификаторов ключей (предполагается, что в заданном хранилище ключей найдена информация о соответствующем ключе).

Примечание: Чтобы убедиться в том, что информация о накопителях сохранена в соответствующей таблице, после автоматического добавления накопителей следует обновить сервер Encryption Key Manager с помощью графического интерфейса пользователя или команды “refresh” на стр. 5-14.

При использовании накопителей LTO 4 и LTO 5 можно задать стандартный пул симметричных ключей (symmetricKeySet), которые будут использоваться для шифрования на вновь добавленных устройствах. Другими словами, при подключении устройства его можно полностью настроить при помощи Encryption Key Manager с учетом данных соответствующих ключей. Если вы не хотите делать этого при

добавлении устройства в таблицу накопителей, это можно сделать с помощью команды **moddrive** после добавления накопителя на магнитной ленте в таблицу накопителей.

Помимо того, что администратор освобождается от необходимости ввода 10-значного серийного номера для каждого накопителя на магнитной ленте, обслуживаемого приложением Encryption Key Manager, при этом обеспечивается стандартная среда для больших системных конфигураций.

Необходимо обратить внимание, что подобное удобство достигается ценой сниженного уровня безопасности. Поскольку устройства добавляются автоматически и могут быть связаны с псевдонимом сертификата (есть возможность записать данные на накопитель на магнитной ленте с заданным псевдонимом сертификата), дополнительная проверка безопасности, которую администратор выполняет при добавлении устройств вручную, не происходит. Важно оценить преимущества и недостатки такого решения, чтобы определить, оправдан ли риск нарушения безопасности, связанный с автоматическим добавлением информации о накопителе на магнитной ленте в таблицу накопителей и неявным предоставлением новому устройству доступа к информации о сертификатах.

Примечание: Свойство drive.acceptUnknownDrives по умолчанию имеет значение `false`. Это означает, что Encryption Key Manager не будет автоматически добавлять новые накопители в таблицу накопителей. Выберите необходимый режим и внесите соответствующие изменения в конфигурацию. Дополнительные сведения см. в приложении B.

Синхронизация данных между двумя серверами диспетчера ключей

Таблицу накопителей и файл свойств конфигурации двух серверов Encryption Key Manager можно синхронизовать. Это можно сделать вручную с помощью команды CLI-клиента **sync** или автоматически, задав четыре свойства в файле KeyManagerConfig.properties.

Примечания

Ни один из методов синхронизации не применяется к хранилищу ключей и XML-файлу групп ключей. Их необходимо копировать вручную.

Функция автоматической синхронизации будет включена только в том случае, если в свойстве sync.ipaddress файла KeyManagerConfig.properties указан допустимый IP-адрес. См. раздел “Автоматическая синхронизация” на стр. 4-3.

Синхронизация вручную

При синхронизации вручную выполняется команда CLI клиента **sync**. Синтаксис следующий:

sync {**-all** | **-config** | **-drivetab**} **-ipaddr** *ip_адрес* :*ssl_порт* [**-merge** | **-rewrite**]

При выполнении этой команды свойства файла конфигурации или сведения таблицы накопителей (или и то, и другое) отправляются с исходного (отправляющего) сервера на сервер назначения (принимающий сервер), задаваемый параметром **–ipaddr**. Принимающий сервер Encryption Key Manager должен быть в рабочем состоянии.

Обязательные поля

-all
Отправка файла свойств конфигурации и сведений таблицы накопителей на сервер, заданный параметром **-ipaddr**.

-config
Отправка на сервер, заданный параметром **-ipaddr**, только файла свойств конфигурации.

-drivetab
Отправка на сервер, заданный параметром **-ipaddr**, только сведений таблицы накопителей.

-ipaddr
Параметр *ip_адрес:ssl_порт* задает адрес и SSL-порт принимающего сервера. Значение параметра *ssl_порт* должно совпадать со значением параметра "TransportListener.ssl.port" в файле KeyManagerConfig.properties принимающего сервера.

Необязательные поля

-merge
Объединение (добавление) новых данных таблицы накопителей с текущими данными на принимающем сервере. (Файл конфигурации всегда перезаписывается.) Это поведение по умолчанию.

-rewrite
Замена текущих данных на принимающем сервере новыми данными.

Автоматическая синхронизация

Таблица накопителей и файл свойств могут автоматически отправляться с основного сервера диспетчера ключей на дополнительный. Для выполнения синхронизации данных дополнительный сервер должен быть запущен. Чтобы автоматически синхронизовать данные дополнительного сервера с данными основного, в файле KeyManagerConfig.properties основного сервера необходимо задать следующие четыре параметра. Файл свойств дополнительного, или принимающего, сервера изменять не требуется.

sync.ipaddress
Указывается адрес и SSL-порт принимающего сервера. Например:

```
sync.ipaddress = backupekm.server.ibm.com:1443
```

Если данное свойство не указано или указано неверно, автоматическая синхронизация будет отключена.

sync.action
Объединение или перезапись существующих данных на принимающем сервере. Допустимы значения **-merge** (по умолчанию) и **rewrite**. Синхронизация свойств конфигурации всегда приводит к перезаписи.

sync.timeinhours
Частота отправки данных. Значение указывается в целых числах (в часах). Начало временного интервала совпадает с моментом запуска сервера, то есть синхронизация будет выполняться после указанного количества часов с момента запуска сервера. Значение по умолчанию - 24.

sync.type
Тип отправляемых данных. Допустимые значения - **drivetab** (по умолчанию), **config** и **all**.

Основы конфигурирования

Примечание: Если вы последовали указаниям, содержащимся в разделе “Создание файла конфигурации, сертификатов и хранилища ключей с помощью графического интерфейса пользователя” на стр. 3-5, то базовая конфигурация уже создана и выполнение описанных ниже действий не требуется. Приведенные ниже способы настройки без использования графического интерфейса могут быть полезны, если вы хотите воспользоваться дополнительными возможностями конфигурации.

Примечание для пользователей ОС Windows: В ОС Windows не могут обрабатываться команды, содержащие пробелы в пути к каталогу. Возможно, при вводе команд необходимо будет указывать краткое имя, которое система сгенерировала для таких путей, например, `progra~1` вместо `Program Files`. Чтобы увидеть список коротких имен каталогов, используйте команду **dir /x**.

Данная процедура содержит описание минимального набора шагов для конфигурирования Encryption Key Manager. В приложении A приведены примеры файлов свойств конфигурации сервера. Полный список всех свойств конфигурации, как для сервера, так и для клиента, см. в приложении B.

1. Используйте средство **keytool** для управления хранилищами ключей JCEKS. При создании хранилища ключей обратите внимание на путь и имя файла, а также на имена, присвоенные сертификатам и ключам. Эта информация будет использоваться при выполнении последующих шагов.
2. Если хранилища ключей не существует, создайте его. Добавьте или импортируйте в новое хранилище ключей сертификаты и ключи, которые будут использоваться для накопителей на магнитной ленте. (См. раздел “Создание ключей и псевдонимов для шифрования на накопителях LTO 4 и LTO 5” на стр. 3-10.) Запомните имена, присвоенные сертификатам и ключам. Эта информация будет использоваться при выполнении последующих шагов.
3. Создайте группы ключей и заполните их псевдонимами ключей. См. раздел “Создание групп ключей и управление ими” на стр. 3-16.
4. При помощи текстового редактора по вашему выбору откройте файл **KeyManagerConfig.properties** и укажите следующие свойства. Следует помнить, что в настоящее время конфигурация сервера удовлетворяет довольно жестким требованиям. Нельзя редактировать файл для компьютера с ОС Linux средствами Windows из-за добавляемого конечного символа строки ^M. При использовании Windows редактируйте файл с помощью текстового редактора gvim/vim.

   Примечание для пользователей ОС Windows: В пакете для разработки ПО на языке Java используются прямые, а не обратные косые черты даже при работе в ОС Windows. При задании путей в файле **KeyManagerConfig.properties** следует использовать только прямые косые черты. При задании полного пути к файлу в командном

окне используйте обратные косые черты, как это обычно делается в ОС Windows.

a. **Audit.Handler.File.Directory** – задайте каталог для хранения журналов аудита.

b. **Audit.metadata.file.name** – задайте полный путь и имя XML-файла с метаданными.

c. **Config.drivetable.file.url** – задайте местонахождение данных о накопителях, известных приложению Encryption Key Manager. Существование этого файла не является обязательным условием для запуска сервера или клиента интерфейса командной строки. Если файл не существует, он будет создан в процессе завершения работы сервера Encryption Key Manager.

d. **TransportListener.ssl.keystore.name** – задайте путь и имя файла хранилища ключей, созданного при выполнении шага 1.

e. **TransportListener.ssl.truststore.name** – задайте путь и имя файла хранилища ключей, созданного при выполнении шага 1.

f. **Admin.ssl.keystore.name** – задайте путь и имя файла хранилища ключей, созданного при выполнении шага 1.

g. **Admin.ssl.truststore.name** – задайте путь и имя файла хранилища ключей, созданного при выполнении шага 1.

h. **config.keystore.file** – задайте путь и имя файла хранилища ключей, созданного при выполнении шага 1.

i. **drive.acceptUnknownDrives** - задайте значение `true` или `false`. Значение true разрешает автоматическое добавление в таблицу накопителей на магнитных лентах новых накопителей, подключающихся к Encryption Key Manager. По умолчанию задано значение false.

5. Можно добавить следующие необязательные записи, определяющие пароли. Если эти записи отсутствуют в файле **KeyManagerConfig.properties**, при запуске сервера Encryption Key Manager предложит ввести пароль хранилища ключей.

   a. **Admin.ssl.keystore.password** – задайте пароль для хранилища ключей, созданного при выполнении шага 1.

   b. **config.keystore.password** – задайте пароль для хранилища ключей, созданного при выполнении шага 1.

   c. **TransportListener.ssl.keystore.password** – задайте пароль для хранилища ключей, созданного при выполнении шага 1.

   При добавлении паролей в файл **KeyManagerConfig.properties** приложение Encryption Key Manager шифрует их для дополнительной защиты.

6. (Необязательный шаг.) Можно задать для свойства **Server.authMechanism** значение `LocalOS`, если аутентификация клиента CLI должна выполняться на основании реестра локальной операционной системы. Если значение свойства не задано (или ему присвоено значение `EKM`), по умолчанию пользователь клиентской программы с интерфейсом командной строки будет входить на сервер диспетчера ключей, используя EKMAdmin в качестве ID и changeME в качестве пароля. (Этот пароль можно изменить с помощью команды **chgpasswd**.)

   В случае присвоения свойству **Server.authMechanism** значения LocalOS на платформах Linux требуется дополнительная настройка. Для получения дополнительных сведений обратитесь к файлу readme, который можно найти по адресу **http://support.dell.com** или на носителе Dell Encryption Key Manager, входящем в комплект поставки приобретенного продукта. Раздел “Аутентификация пользователей клиента CLI” на стр. 5-5 содержит дополнительную информацию.

7. Сохраните изменения в файле **KeyManagerConfig.properties**.
8. Запустите сервер Encryption Key Manager. Чтобы запустить сервер, не используя графический интерфейс пользователя, выполните следующие действия.

   В Windows
   перейдите в каталог c:\ekm\ekmserver и выберите файл **startServer.bat**

   На платформах Linux
   перейдите в каталог /var/ekm/ekmserver и запустите файл `. ./startServer.sh`

   Дополнительные сведения см. в разделе “Запуск, обновление и остановка сервера диспетчера ключей” на стр. 5-1.
9. Запустите клиент интерфейса командной строки:

   В Windows
   перейдите в каталог c:\ekm\ekmclient и выберите файл **startClient.bat**

   На платформах Linux
   перейдите в каталог /var/ekm/ekmclient и запустите файл `. ./startClient.sh`

   Дополнительные сведения см. в разделе “Клиент интерфейса командной строки” на стр. 5-5.
10. Если при выполнении шага 4(i) было задано значение **drive.acceptUnknownDrives = false**, настройте накопитель, введя по приглашению # следующую команду:

    `adddrive -drivename` *`drive_name`* `-rec1` *`cert_name`* `-rec2` *`cert_name`*

    Например:

    ```
    # adddrive -drivename 000001365054 -rec1 key1c1 -rec2 key1c2
    ```

    а затем

    ```
    # listdrives -drivename 000001365054
    ```

    Возвращаемая информация

    ```
    Entry Key:  SerialNumber = 000001365054

    Entry Key:  AliasTwo = key1c2

    Entry Key:  AliasOne = key1c1
     Deleted : false
     Updated : true
     TimeStamp : Sun Jul 03 17:34:44 MST 2007
    ```
11. Введите команду **listdrives** по приглашению #, чтобы убедиться, что накопитель был успешно добавлен.

Глава 5. Администрирование Encryption Key Manager

Запуск, обновление и остановка сервера диспетчера ключей

Сервер Encryption Key Manager очень просто запускать и останавливать.

При обновлении сервер Encryption Key Manager создает дамп содержимого хранилища ключей, таблицы накопителей и данных конфигурации из памяти, а затем снова загружает их в память. Выполнение обновления рекомендуется после внесения каких-либо изменений в эти компоненты с помощью клиента CLI. Несмотря на то что изменения автоматически сохраняются при завершении работы сервера Encryption Key Manager, обновление сервера обеспечивает защиту внесенных изменений от потери в случае сбоя в системе или отключения питания.

Запуск сервера Encryption Key Manager с помощью графического интерфейса пользователя Dell Encryption Key Manager:

1. Если графический интерфейс пользователя еще не открыт, откройте его.

   В Windows
   перейдите в каталог c:\ekm\gui и щелкните кнопкой мыши по файлу **LaunchEKMGui.bat**

   На платформе Linux
   перейдите в каталог /var/ekm/gui и запустите файл `../LaunchEKMGui.sh`
2. Выберите **Server Health Monitor** (Монитор работоспособности сервера) в навигационном меню в левой части окна интерфейса.
3. На странице “Server Status (Состояние сервера)” (рис. 5-1) нажмите **Start Server** (Запустить сервер) или **Refresh Server** (Обновить сервер).

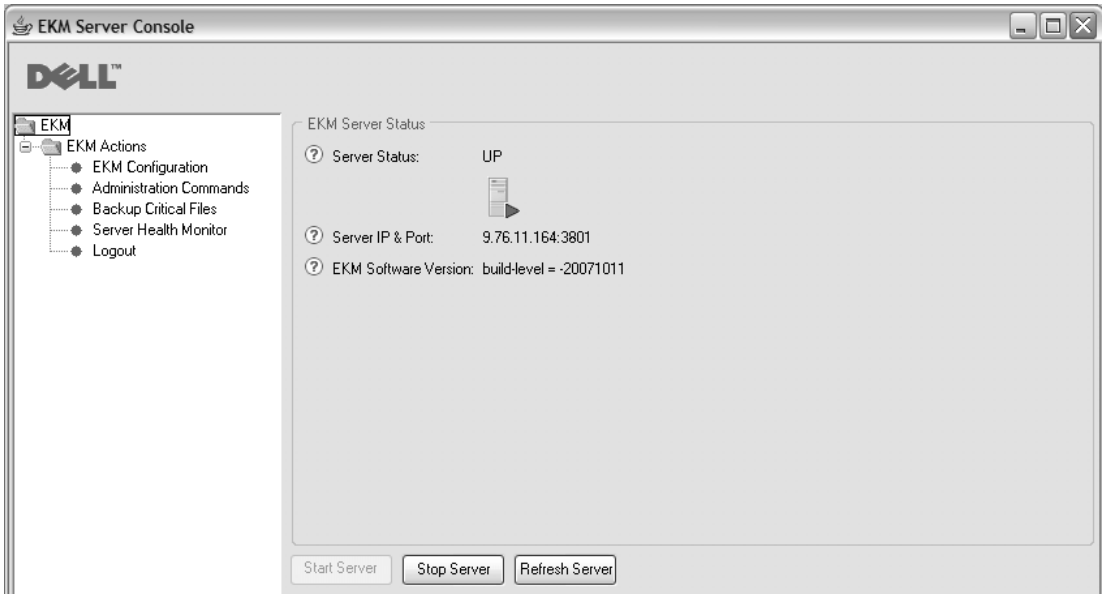

Рисунок 5-1. Server Status (Состояние сервера)

4. Изменение состояния сервера отражается в окне Server Status (Состояние сервера). См. раздел рис. 5-1.

5. Появляется окно входа (рис. 5-2).
   В качестве ID пользователя введите `EKMAdmin`. Исходный пароль - `changeME`. После

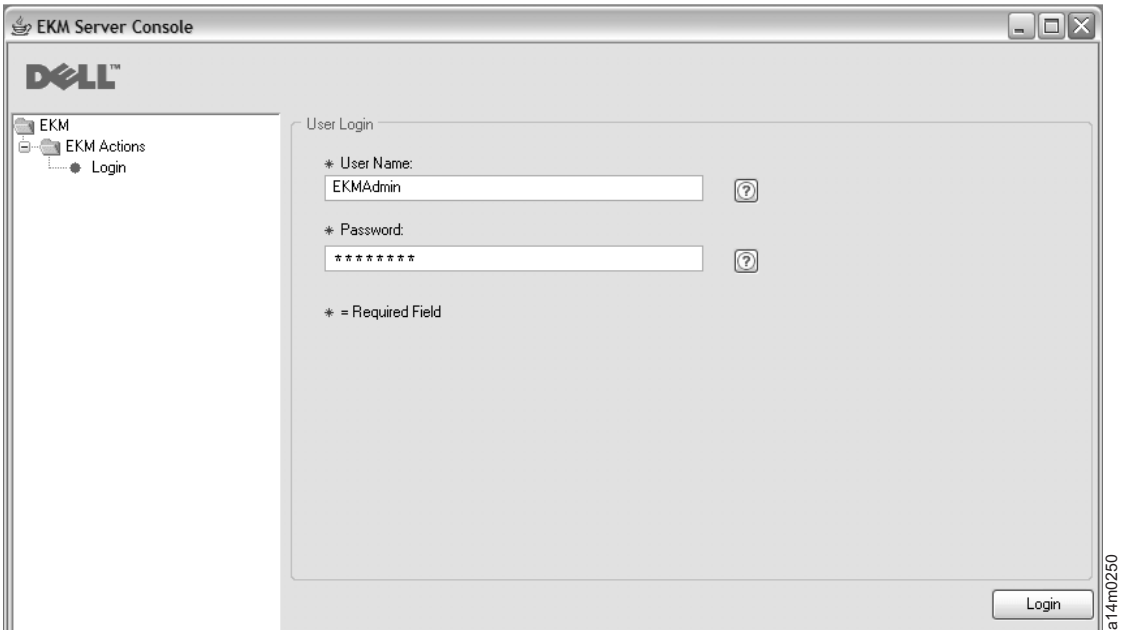

Рисунок 5-2. Login Window (Окно входа)

входа в систему пароль можно изменить с помощью команды **chgpasswd**. См. описание команды “chgpasswd” на стр. 5-9.

Примечание: Возможно, графический интерфейс пользователя Dell Encryption Key Manager не сможет отобразить IP-адрес хоста.

Отобразить в **мониторе работоспособности сервера** IP-адрес хоста Encryption Key Manager не позволяют два ограничения текущего графического интерфейса пользователя:

- Текущее приложение не распознает формат IPV6. Если хост настроен на IPV6-адреса, приложение Encryption Key Manager не сможет отобразить IP-адрес.
- Если приложение Encryption Key Manager установлено в системе Linux, оно отображает адрес localhost, а не фактический активный порт IP.

Чтобы получить фактический IP-адрес системы хоста, определите адрес порта IP при помощи конфигурации сети. В ОС Windows откройте командное окно и введите `ipconfig`. В Linux введите `isconfig`.

6. Щелкните **Login** (Вход).

На этой же странице Server Status можно остановить сервер.

Запуск сервера диспетчера ключей с помощью сценария

В Windows
перейдите в каталог c:\ekm\ekmserver и выберите файл **startServer.bat**

На платформе Linux
перейдите в каталог /var/ekm/ekmserver и запустите файл `. ./startServer.sh`

Чтобы остановить сервер, выполните команду **stopekm** одним из способов, описанных ниже в разделе “Клиент интерфейса командной строки” на стр. 5-5. Кроме того, можно отправить команду **sigterm** процессу диспетчера ключей. Это позволяет корректно завершить работу сервера и закрыть его. Не отправляйте процессу диспетчера ключей команду **sigkill**. **sigkill** не позволяет корректно завершить работу процесса. Например, в системах Linux выполните команду `kill -SIGTERM pid` или `kill -15 pid`.

Запуск и остановка сервера диспетчера ключей из командной строки

Чтобы запустить сервер Encryption Key Manager из любого командного окна или оболочки, введите команду:

```
java com.ibm.keymanager.EKMLaunch KeymanagerConfig.properties
```

При этом сервер Encryption Key Manager запускается в фоновом режиме. При корректном запуске Java-процесс Encryption Key Manager можно отобразить с помощью команды `ps -ef | grep java` (Linux) или с помощью диспетчера задач в Windows. При запуске в виде службы Windows процесс отображается как LaunchEKMService.

Чтобы остановить сервер, выполните команду **stopekm** одним из способов, описанных ниже в разделе “Клиент интерфейса командной строки” на стр. 5-5. Кроме того, можно отправить команду **sigterm** процессу диспетчера ключей. Это позволяет корректно завершить работу сервера и закрыть его. Не отправляйте процессу диспетчера ключей команду **sigkill**. **sigkill** не позволяет корректно завершить работу процесса. Например, в системах Linux выполните команду `kill -SIGTERM pid` или `kill -15 pid`.

В системах Windows при запуске Dell Encryption Key Manager в виде службы Windows процесс можно остановить из панели управления.

Установка сервера диспетчера ключей в виде службы Windows

Установка сервера Encryption Key Manager в виде службы на хост-сервере гарантирует запуск серверного приложения Encryption Key Manager после перезагрузки хост-сервера.

1. Распакуйте во временный каталог исполняемый файл LaunchEKMService.exe из выпуска, загруженного с Web-сайта технической поддержки Dell (**http://support.dell.com**).
2. Для правильной работы службы необходимо установить некоторые переменные среды:
   a. В меню "Пуск" выберите пункт **Панель управления**.
   b. Дважды щелкните пункт **Система**.
   c. Перейдите на вкладку **Дополнительно**.
   d. Нажмите кнопку **Переменные среды**.
   e. В списке системных переменных нажмите кнопку **Создать**.
   f. Укажите `JAVA_HOME` как имя переменной и каталог IBM JVM в качестве ее значения. Каталог установки по умолчанию — `C:\PROGRA~1\IBM\Java60`.
   g. Нажмите кнопку **OK**.
3. Измените переменную PATH в соответствии с этой процедурой.

Примечание: Установка переменной PATH из командной строки невозможна.

a. В меню "Пуск" выберите пункт **Панель управления**.

b. Дважды щелкните пункт **Система**.

c. Перейдите на вкладку **Дополнительно**.

d. Нажмите кнопку **Переменные среды**.

e. Выберите в списке переменных среды переменную **Path** и нажмите кнопку **Изменить**.

f. Добавьте в начало переменной Path путь к `IBM JVM`. Каталог установки по умолчанию — `C:\PROGRA~1\IBM\Java60\jre\bin`.

Примечание: В конец пути добавьте точку с запятой (;), чтобы отделить его от других каталогов списка путей.

g. Нажмите кнопку **OK**.

4. Убедитесь, что пути в файле свойств конфигурации сервера Encryption Key Manager полностью определены. Имя этого файла - KeyManagerConfig.properties, он находится в каталоге `C:\ekm\gui`. Все перечисленные ниже пути в файле необходимо проверить и изменить так, чтобы они были полностью определены (например, указывайте путь как `c:\ekm\gui\EKMKeys.jck`, а не в виде `gui\EKMKeys.jck`). В приведенных ниже примерах показано, как нужно изменить пути, если установка была выполнена по умолчанию.

   Это относится к свойствам и указывающим на них полностью определенным путям, когда при установке и для хранилищ ключей используются имена по умолчанию. Все эти записи можно найти в файле KeyManagerConfig.properties.

   config.keygroup.xml.file
   Путь необходимо изменить на: FILE:C:/ekm/gui/keygroups/KeyGroups.xml

   Admin.ssl.keystore.name
   Путь необходимо изменить на: C:/ekm/gui/EKMKeys.jck

   TransportListener.ssl.truststore.name
   Путь необходимо изменить на: C:/ekm/gui/EKMKeys.jck

   Audit.metadata.file.name
   Путь необходимо изменить на: C:/ekm/gui/metadata/ekm_metadata.xml

   Audit.handler.file.directory
   Путь необходимо изменить на: C:/ekm/gui/audit

   config.keystore.file
   Путь необходимо изменить на: C:/ekm/gui/EKMKeys.jck

   TransportListener.ssl.keystore.name
   Путь необходимо изменить на: C:/ekm/gui/EKMKeys.jck

   config.drivetable.file.url
   Путь необходимо изменить на: FILE:C:/ekm/gui/drivetable/ekm_drivetable.dt

   Admin.ssl.truststore.name
   Путь необходимо изменить на: C:/ekm/gui/EKMKeys.jck

5. Файл **LaunchEKMServices.exe** нужно запустить из командной строки. В Windows доступ к ней можно получить, если выбрать **Пуск > Программы > Стандартные > Командная строка**.

6. Из командной строки перейдите во временный каталог, куда был распакован файл **LaunchEKMService.exe**. Запустите файл **LaunchEKMService.exe**, используя как образец следующие параметры.

   LaunchEKMService {**-help** | **-i** *файл_конфигурации* | **-u**}

-help
Выводит сведения об использовании команды.

-i Устанавливает Encryption Key Manager в виде службы Windows. Для этого параметра необходимо указать как аргумент имя файла свойств конфигурации в виде полностью определенного пути. Путь и имя файла по умолчанию - `C:\ekm\gui\KeyManagerConfig.properties`.

-u Удаляет службу Windows для диспетчера ключей, если больше нет необходимости запускать его в виде службы. Следует заметить, что перед удалением службу EKMServer необходимо остановить. При выполнении этой команды возможно также появление следующего сообщения об ошибке: `Не удалось удалить EKMServer. Ошибка 0`. Тем не менее, служба все еще может быть удалена.

Чтобы установить Encryption Key Manager в виде службы Windows, выполните следующую команду:

`LaunchEKMService.exe -i` *`файл_конфигурации`*

7. После установки службы при помощи указанной выше команды EKMServer появится в панели управления службами, с помощью которой можно запускать и останавливать Encryption Key Manager.

Примечание: При первом использовании службы Windows ее необходимо запустить вручную с помощью панели управления.

Клиент интерфейса командной строки

После запуска сервера Encryption Key Manager им можно управлять из клиентского интерфейса локально или удаленно с помощью команд CLI. Чтобы подавать команды CLI, сначала необходимо запустить клиент CLI.

Аутентификация пользователей клиента CLI

Свойство Server.authMechanism в файле конфигурации позволяет указывать, какой механизм проверки подлинности будет использоваться для локальных и удаленных клиентов. Если значение этого свойства равно `EKM`, пользователь клиента CLI должен войти на сервер, используя ID EKMAdmin и пароль changeME. (Этот пароль можно изменить с помощью команды **chgpasswd**. См. описание команды “chgpasswd” на стр. 5-9.) По умолчанию свойство Server.authMechanism имеет значение `EKM`.

Если в файле KeyManagerConfig.properties свойству Server.authMechanism задано значение `LocalOS`, аутентификация клиента выполняется относительно реестра локальной операционной системы. Пользователь клиента CLI должен войти на сервер, используя ID и пароль для входа в операционную систему. Следует заметить, что вход в систему и передача команд серверу разрешены только при указании имени и пароля пользователя, учетная запись которого была использована при запуске сервера, кроме того, пользователь должен иметь полномочия суперпользователя (ROOT).

ВАЖНО: При изменении файла конфигурации Encryption Key Manager сервер Encryption Key Manager должен быть остановлен, а графический интерфейс пользователя - закрыт.

В Windows для аутентификации с помощью локальной операционной системы в файле KeyManagerConfig.properties установите Server.authMechanism=LocalOS, выполнив следующие действия:

1. Найдите файл KeyManagerConfig.properties (каталог `c:\ekm\gui`).

2. Откройте файл при помощи текстового редактора по вашему выбору (рекомендуется WordPad).
3. Найдите строку Server.authMechanism. Если эта строка отсутствует, добавьте ее в файл именно в таком формате: `Server.authMechanism=LocalOS`.
4. Сохраните файл.

Теперь ваш идентификатор пользователя и пароль для сервера Encryption Key Manager соответствуют учетной записи пользователя ОС. Следует заметить, что управлять сервером Encryption Key Manager могут только пользователи, которым разрешен вход в систему сервера и передача ему команд и которые имеют полномочия администратора.

При аутентификации с помощью локальной операционной системы на платформах Linux необходимы дополнительные действия:

1. Загрузите Dell Release R175158 (EKMServicesAndSamples) с **http://support.dell.com** и извлеките файлы в каталог по вашему выбору.
2. В структуре, полученной при загрузке, найдите каталог LocalOS.
3. Скопируйте файл libjaasauth.so из каталога JVM-JaasSetup для своей платформы в каталог *java_home*/jre/bin.
   - Для сред Linux на 32-разрядной платформе Intel: скопируйте файл LocalOS-setup/linux_ia32/libjaasauth.so в каталог *домашний_каталог_java*/jre/bin/. В случае 32-разрядного ядра Linux с JVM версии 1.6 на Intel *домашний_каталог_java* — это обычно *путь_установки_java*/IBMJava-i386-60.
   - Для сред Linux на 64-разрядной платформе AMD64: скопируйте файл LocalOS-setup/linux-x86_64/libjaasauth.so в каталог *домашний_каталог_java*/jre/bin/. В случае 64-разрядного ядра Linux с JVM версии 1.6 на AMD *домашний_каталог_java* — это обычно *путь_установки_java*/IBMJava-x86_64-60.

Для систем Windows этот файл необязателен.

По завершении установки можно запустить сервер Encryption Key Manager. Клиент Encryption Key Manager теперь сможет войти в систему, используя имя пользователя и пароль операционной системы. Следует заметить, что вход в систему и передача команд серверу разрешены только при указании имени пользователя, учетная запись которого была использована при запуске сервера, кроме того, пользователь должен иметь полномочия суперпользователя (ROOT).

Дополнительные сведения см. в файле readme, доступном на носителе с программным обеспечением Dell и для загрузки на **http://support.dell.com**.

Запуск клиента интерфейса командной строки

Примечание: Свойство TransportListener.ssl.port в файлах свойств сервера Encryption Key Manager и CLI-клиента Encryption Key Manager должно иметь одинаковое значение, в противном случае они не смогут взаимодействовать. При возникновении проблем см. раздел “Отладка для решения проблем связи между CLI-клиентом и сервером EKM” на стр. 6-2.

CLI-клиент Encryption Key Manager и сервер Encryption Key Manager для защиты своего взаимодействия используют протокол SSL. При использовании конфигурации JSSE по умолчанию, без проверки подлинности клиента, сертификаты в TransportListener.ssl.keystore на сервере Encryption Key Manager должны быть представлены в TransportListener.ssl.truststore. Благодаря этому клиент знает, что может доверять серверу. Если CLI-клиент Encryption Key Manager работает на том же

компьютере, что и сервер Encryption Key Manager, то можно использовать один и тот же файл свойств конфигурации. Благодаря этому CLI-клиент Encryption Key Manager может использовать ту же, что и сервер Encryption Key Manager, конфигурацию хранилищ ключей и доверенных хранилищ. Если они находятся на разных компьютерах или нужно, чтобы клиент использовал другие хранилища ключей, необходимо выполнить экспорт сертификатов из TransportListener.ssl.keystore, указанного в файле свойств конфигурации сервера Encryption Key Manager. Затем эти сертификаты необходимо импортировать в доверенное хранилище, указанное в TransportListener.ssl.truststore файла свойств CLI-клиента Encryption Key Manager.

Запустить клиент CLI и использовать команды CLI можно четырьмя способами. Независимо от способа необходимо указать имя файла конфигурации CLI. Подробности см. в приложении B.

Использование сценария

В Windows
перейдите в каталог c:\ekm\ekmclient и выберите файл **startClient.bat**

На платформе Linux
перейдите в каталог /var/ekm/ekmclient и запустите файл `. ./startClient.sh`

Интерактивный режим
Чтобы выполнять команды в интерактивном режиме из любого окна или оболочки, введите:

```
java com.ibm.keymanager.KMSAdminCmd CLIconfiglfile_name -i
```

Появится приглашение #. Перед выполнением каких-либо команд необходимо выполнить для клиента CLI вход на сервер диспетчера ключей с помощью следующей команды:

```
#login –ekmuser EKMAdmin –ekmpassword changeME
```

После успешного входа CLI-клиента на сервер диспетчера ключей можно выполнять любые команды CLI-интерфейса. По окончании работы завершите сеанс клиента CLI с помощью команды **quit** или **logout**. По умолчанию сервер Encryption Key Manager закрывает соединение, если клиент остается неактивным в течение десяти минут. Если после этого попытаться ввести команду, сеанс клиента будет закрыт. Чтобы увеличить тайм-аут соединения сервера и клиента Encryption Key Manager, измените параметр TransportListener.ssl.timeout в файле KeyManagerConfig.properties.

Использование командного файла
Чтобы отправить набор команд в файле на сервер диспетчера ключей, создайте файл с необходимыми командами, например, *clifile*. Первой в этом файле должна быть команда **login**, поскольку перед выполнением каких-либо команд необходимо выполнить вход клиента на сервер. Например, файл clifile может содержать следующий текст:

```
login -ekmuser EKMAdmin -ekmpassword changeME
listdrives
```

Чтобы выполнить этот командный файл, запустите клиент CLI:

```
java com.ibm.keymanager.admin.KMSAdminCmd CLIconfiglfile_name –filename clifile
```

По одной команде
Можно выполнять по одной команде, указывая идентификатор и пароль пользователя в CLI для каждой команды. В любом командном окне или в командной оболочке введите:

```
java com.ibm.keymanager.KMSAdminCmd ClientConfig.properties_name -listdrives
    -ekmuser EKMAdmin -ekmpassword changeME
```

(Этот пароль можно изменить с помощью команды **chgpasswd**.) Команда будет выполнена, а сеанс работы клиента завершится.

Команды CLI

Encryption Key Manager предоставляет набор команд, которые можно использовать для взаимодействия с сервером Encryption Key Manager через интерфейс командной строки клиента. Набор включает в себя следующие команды.

addaliastogroup

Копирование псевдонима из существующей (исходной) в новую (целевую) группу ключей. Эту команду можно использовать для добавления псевдонима, который существует в одной группе ключей, в другую группу ключей.

addaliastogroup -aliasID *псевдоним* **-sourceGroupID** *имя_группы* **-targetGroupID** *имя_группы*

-aliasID
Псевдоним ключа, который нужно добавить.

-sourceGroupID
Уникальное имя *имя_группы*, используемое для идентификации группы, из которой необходимо скопировать псевдоним.

-targetGroupID
Уникальное имя *имя_группы*, используемое для идентификации группы, в которую необходимо скопировать псевдоним.

Пример: addaliastogroup -aliasID псевдоним -sourceGroupID группа_ключей_1 -targetGroupID группа_ключей_2

adddrive

Добавление нового накопителя в таблицу накопителей диспетчера ключей. Сведения об автоматическом добавлении накопителей на магнитной ленте в таблицу накопителей см. в разделе “Автоматическое обновление таблицы ленточных накопителей” на стр. 4-1. Сведения о требованиях к псевдониму см. в разделах “Ключи шифрования и ленточные накопители LTO 4 и LTO 5” на стр. 2-4.

adddrive -drivename *имя_накопителя* [**-rec1** *псевдоним*] [**-rec2** *псевдоним*][**-symrec** *псевдоним*]

-drivename
Переменная *имя_накопителя* должна содержать 12-значный серийный номер накопителя, который необходимо добавить.

Примечание: Перед 10-значным серийным номером нужно добавить два нуля (0), чтобы его длина составила 12 знаков.

-rec1
Определяет *псевдоним* (или метку ключа) сертификата накопителя.

-rec2
Определяет второй *псевдоним* (или метку ключа) сертификата накопителя.

-symrec
Определяет *псевдоним* (симметричного ключа) или имя группы ключей накопителя на магнитной ленте.

Пример: adddrive -drivename 000123456789 -rec1 псевдоним_1 -rec2 псевдоним_2

addkeygroup

Создание экземпляра группы ключей с уникальным идентификатором группы в файле KeyGroups.xml.

addkeygroup -groupID *имя_группы*

-groupID
Уникальное имя *имя_группы*, используемое для идентификации группы в файле KeyGroups.xml.

Пример: addkeygroup -groupID группа_ключей_1

addkeygroupalias

Создание нового псевдонима для уже существующего в хранилище ключей псевдонима ключа для добавления к заданному идентификатору группы ключей.

addkeygroupalias -alias *псевдоним* **-groupID** *имя_группы*

-alias
Новый *псевдоним* для ключа.

-groupID
Уникальное имя *имя_группы*, используемое для идентификации группы в файле KeyGroups.xml.

Пример: addkeygroupalias -alias псевдоним -groupID группа_ключей_1

chgpasswd

Изменение пароля по умолчанию, заданного для пользователя клиента CLI (EKMAdmin).

chgpasswd -new *пароль*

-new
Новый *пароль*, заменяющий предыдущий.

Пример: chgpasswd -new ebw74jxr

createkeygroup

Создание первого объекта группы ключей в файле KeyGroups.xml. Команду следует запускать только один раз.

createkeygroup -password *пароль*

-password
Пароль, который используется для шифрования пароля хранилища ключей в файле KeyGroups.xml для последующего восстановления. Хранилище ключей

шифрует ключ группы ключей, который, в свою очередь, шифрует пароль каждого псевдонима группы ключей. Поэтому все ключи в файле KeyGroups.xml являются защищенными.

Пример: createkeygroup -password пароль

deletedrive

Удаление накопителя из таблицы накопителей диспетчера ключей. Эквивалентные команды - **deldrive** и **removedrive**.

deletedrive -drivename *имя_накопителя*

-drivename
Переменная *имя_накопителя* содержит серийный номер накопителя, который требуется удалить.

Пример: deletedrive -drivename 000123456789

delgroupalias

Удаление псевдонима ключа из группы ключей.

delgroupalias -groupID *имя_группы* **-alias** *псевдоним*

-groupID
Уникальное имя *имя_группы*, используемое для идентификации группы в файле KeyGroups.xml.

-alias
Переменная *псевдоним* содержит псевдоним ключа, который требуется удалить.

Пример: delgroupalias -groupID группа_ключей_1 -alias псевдоним

delkeygroup

Удаление всей группы ключей.

delkeygroup -groupID *имя_группы*

-groupID
Уникальное имя *имя_группы*, используемое для идентификации группы в файле KeyGroups.xml.

Пример: delkeygroup -groupID группа_ключей_1

exit

Завершение работы с клиентом CLI и остановка сервера Encryption Key Manager. Эквивалентная команда - **quit**.

Пример: exit

export

Экспорт таблицы накопителей или файла конфигурации сервера Encryption Key Manager в указанное URL-адресом место.

export {**-drivetab**|**-config**} **-url** *адрес*

-drivetab
Экспорт таблицы накопителей.

-config
Экспорт файла конфигурации сервера Encryption Key Manager.

-url
Переменная *адрес* определяет каталог, в который будет сохранен файл экспорта.

Пример: export -drivetab -url FILE:///keymanager/data/export.table

help

Вывод на экран названий и синтаксиса команд интерфейса командной строки. Эквивалентная команда - **?**.

help

import

Импорт таблицы накопителей или файла конфигурации из заданного местоположения.

import {**-merge**|**-rewrite**} {**-drivetab**|**-config**} **-url** *адрес*

-merge
Объединение новых данных с существующими.

-rewrite
Замена существующих данных новыми данными.

-drivetab
Импорт таблицы накопителей.

-config
Импорт файла конфигурации.

-url
Переменная *адрес* определяет каталог, из которого необходимо скопировать новые данные.

Пример: import -merge -drivetab -url FILE:///keymanager/data/export.table

list

Вывод списка сертификатов, находящихся в хранилище ключей, которое определяется свойством config.keystore.file.

list [**-cert** |**-key**|**-keysym**][**-alias** *псевдоним* **-verbose** |**-v**]

-cert
Вывод списка сертификатов, находящихся в указанном хранилище ключей.

-key
Вывод списка всех ключей, находящихся в указанном хранилище ключей.

-keysym
Вывод списка симметричных ключей, находящихся в указанном хранилище ключей.

-alias
Переменная *псевдоним* определяет конкретный сертификат, информацию о котором необходимо показать.

-verbose|-v
Вывод на экран дополнительной информации о сертификате.

Примеры:
Команда `list -v` выводит список всех сертификатов, находящихся в хранилище ключей.

Команда `list -alias mycert -v` показывает все данные для псевдонима mycert, если он существует в хранилище ключей, заданном свойством config.keystore.file.

listcerts

Вывод списка сертификатов, находящихся в хранилище ключей, которое определяется свойством config.keystore.file.

listcerts [**-alias** *псевдоним* **-verbose** |**-v**]

-alias
Переменная *псевдоним* определяет конкретный сертификат, информацию о котором необходимо показать.

-verbose|-v
Вывод на экран дополнительной информации о сертификате.

Пример: listcerts -alias псевдоним_1 -v

listconfig

Вывод списка свойств конфигурации сервера Encryption Key Manager, хранящихся в памяти. Список будет отображать актуальное содержимое файла KeyManagerConfig.properties, а также любые изменения, внесенные с помощью команды **modconfig**.

listconfig

listdrives

Вывод списка накопителей, содержащихся в таблице накопителей.

listdrives [**-drivename** *имя_накопителя*]

-drivename
Переменная *имя_накопителя* содержит серийный номер накопителя, информацию о котором требуется показать.

-verbose|-v
Вывод на экран дополнительной информации о накопителе (накопителях).

Пример: listdrives -drivename 000123456789

login

Вход в систему клиента CLI на сервере Encryption Key Manager.

login -ekmuser *ID_пользователя* **-ekmpassword** *пароль*

-ekmuser
В зависимости от используемого типа аутентификации в качестве значения переменной *ID_пользователя* укажите `EKMadmin` или localOS (см. раздел “Аутентификация пользователей клиента CLI” на стр. 5-5).

-ekmpassword
Действительный пароль, соответствующий идентификатору пользователя.

Пример: login -ekmuser EKMAdmin -ekmpassword changeME

logout

Завершение сеанса работы для текущего пользователя. Эквивалентная команда - **logoff**. Эти команды можно использовать только при активированном сеансе клиента.

Пример: logout

modconfig

Изменение свойства в файле свойств конфигурации сервера Encryption Key Manager (KeyManagerConfig.properties). Эквивалентная команда - **modifyconfig**.

modconfig {**-set | -unset**} **-property** *имя* **-value** *значение*

-set
Присвоение указанного значения указанному свойству.

-unset
Удаление указанного свойства.

-property
Переменная *имя* содержит имя требуемого свойства.

-value
Переменная *значение* определяет новое значение требуемого свойства при использовании ключа **-set**.

Пример: modconfig -set -property sync.timeinhours -value 24

moddrive

Изменение сведений о накопителе в таблице накопителей. Эквивалентная команда - **modifydrive**.

moddrive -drivename *имя_накопителя* {**-rec1** [*псевдоним*] | **-rec2** [*псевдоним*]| **-symrec** [*псевдоним*]}

-drivename
Переменная *имя_накопителя* содержит серийный номер накопителя на магнитной ленте.

-rec1
Определяет *псевдоним* (или метку ключа) сертификата накопителя.

-rec2
Определяет второй *псевдоним* (или метку ключа) сертификата накопителя.

-symrec
Определяет *псевдоним* (симметричного ключа) или имя группы ключей накопителя на магнитной ленте.

Пример: moddrive -drivename 000123456789 -rec1 новый_псевдоним_1

refresh

Сообщает приложению Encryption Key Manager о необходимости обновить значения в таблице отладки, аудита и накопителей с учетом актуальных параметров конфигурации.

Пример: refresh

refreshks

Обновление хранилища ключей. Используйте эту команду для перезагрузки указанного в свойстве **config.keystore.file** хранилища ключей, если в него были внесены изменения во время работы сервера Encryption Key Manager. Эта команда должна использоваться только в случае необходимости, поскольку ее выполнение может привести к снижению производительности.

Пример: refreshks

status

Вывод информации о состоянии сервера диспетчера ключей (запущен или остановлен).

Пример: status

stopekm

Останавливает сервер Encryption Key Manager.

Пример: stopekm

sync

Синхронизация свойств файла конфигурации либо информации в таблице накопителей (или и тех, и других данных), хранящихся на другом сервере Encryption Key Manager, с данными сервера диспетчера ключей, выполнившего команду.

Примечание: Ни один из способов синхронизации не предполагает синхронизации хранилища ключей или файла KeyGroups.xml. Эти объекты необходимо копировать вручную.

sync {**-all | -config | -drivetab**} **-ipaddr** *ip_адрес* :*ssl*:*порт* [**-merge | -rewrite**]

-all
Отправка файла свойств конфигурации и сведений таблицы накопителей на сервер Encryption Key Manager, заданный параметром **-ipaddr**.

-config
Отправка на сервер Encryption Key Manager, заданный параметром **-ipaddr**, только файла свойств конфигурации.

-drivetab
Отправка на сервер Encryption Key Manager, заданный параметром **-ipaddr**, только сведений таблицы накопителей.

-ipaddr
Параметр *ip_addr:ssl:port* задает адрес и SSL-порт принимающего сервера Encryption Key Manager. Значение *ssl:port* должно совпадать со значением свойства “TransportListener.ssl.port”, указанным в файле KeyManagerConfig.properties принимающего сервера.

-merge
Объединение новых данных таблицы накопителей с существующими. (Файл конфигурации всегда является перезаписываемым.) Этот ключ используется по умолчанию.

-rewrite
Замена существующих данных новыми данными.

Пример: sync -drivetab -ipaddr remoteekm.ibm.com:443 -merge

version

Отображает версию сервера Encryption Key Manager.

Пример: version

Глава 6. Выявление проблем

Отладку можно включить для одного, нескольких или всех компонентов Encryption Key Manager.

Важные файлы, которые следует проверить в случае проблем с сервером Encryption Key Manager

Если приложение Encryption Key Manager запустить не удается, для определения причины проблемы рекомендуется проверить следующие три файла.

- **native_stdout.log** и **native_stderr.log**
  - Поскольку сервер Encryption Key Manager работает как фоновый процесс, у него нет консоли для стандартного отображения информационных сообщений и сообщений об ошибках. Соответствующие сообщения регистрируются в этих двух файлах.
  - Если в файле свойств сервера Encryption Key Manager присутствует свойство **debug.output.file**, то эти два файла будут созданы в том же каталоге, что и журнал отладки.
  - Если в файле свойств сервера Encryption Key Manager свойство **debug.output.file** отсутствует, то эти два файла будут созданы в рабочем каталоге.
  - При каждом запуске сервера Encryption Key Manager эти два файла удаляются и создаются заново.
- **Журнал аудита**
  - Журнал аудита содержит записи, зарегистрированные в процессе работы Encryption Key Manager.
  - В **KeyManagerConfig.properties**, файле свойств конфигурации сервера Encryption Key Manager, размещение жунала определяют два свойства:
    - Audit.handler.file.directory – указывает каталог размещения журнала аудита
    - Audit.handler.file.name – указывает имя журнала аудита.
  - Дополнительные сведения об аудите см. в разделе Глава 7, “Протоколы аудита”, на стр. 7-1.

Записи в журнале о паролях хранилища ключей, длина которых превышает 127 символов

Когда приложение Encryption Key Manager установлено в качестве службы Windows и длина паролей хранилища ключей в файле KeyManagerConfig.properties составляет 128 символов или больше, Encryption Key Manager не запустится из-за невозможности запросить пароль допустимой длины. В таких случаях собственные журналы Encryption Key Manager будут содержать записи, подобные следующим:

native_stdout.log

```
Server initialized
Default keystore failed to load
```

native_stderr.log

```
at com.ibm.keymanager.KeyManagerException: Default keystore failed to load
at com.ibm.keymanager.keygroups.KeyGroupManager.loadDefaultKeyStore
   (KeyGroupManager.java:145)
at com.ibm.keymanager.keygroups.KeyGroupManager.init(KeyGroupManager.java:605)
```

```
at com.ibm.keymanager.EKMServer.c(EKMServer.java:243)
at com.ibm.keymanager.EKMServer.<init>(EKMServer.java:753)
at com.ibm.keymanager.EKMServer.a(EKMServer.java:716)
at com.ibm.keymanager.EKMServer.main(EKMServer.java:129)
```

Отладка для решения проблем связи между CLI-клиентом и сервером EKM

Связь между CLI-клиентом EKM и сервером EKM осуществляется с использованием портов, заданных свойством TransportListener.ssl.port в файлах свойств конфигурации клиента и сервера. Это соединение защищается с помощью протокола SSL.

Ниже приведен список причин невозможности подключения клиента к серверу EKM. Описываются действия, позволяющие определить причину проблемы и устранить ее.

- Сервер EKM не работает, поэтому клиенту не к чему подключаться.
  1. Введите команду **netstat –an** в командной строке и убедитесь, что выводятся порты, заданные свойствами TransportListener.ssl.port и TransportListener.tcp.port в файле свойств сервера EKM. Если эти порты не выводятся на экран, значит, сервер не работает.
- Свойство TransportListener.ssl.host в файле свойств CLI-клиента EKM не указывает на адрес, по которому находится сервер EKM.
  1. По умолчанию для свойства TransportListener.ssl.host в файле свойств CLI-клиента EKM задано значение `localhost`. Измените значение данного свойства, чтобы оно указывало на верный адрес.
- Сервер EKM и CLI-клиент EKM передают данные по разным портам.
  1. Проверьте свойства TransportListener.ssl.port в файлах свойств сервера EKM и CLI-клиента EKM, чтобы убедиться, что они имеют одинаковое значение.
- Сервер EKM и CLI-клиент EKM не могут обнаружить общий сертификат для установки защищенного соединения.
  1. Убедитесь, что хранилища ключей, указанные в свойствах TransportListener.ssl.keystore и TransportListener.ssl.truststore CLI-клиента, содержат те же сертификаты, что и хранилища ключей в свойствах Admin.ssl.keystore и Admin.ssl.truststore сервера.
  2. Убедитесь, что для параметра TransportListener.ssl.keystore.password в свойствах клиента задан верный пароль.
  3. Убедитесь, что ни один из сертификатов в данных хранилищах ключей не просрочен. JSSE не использует сертификаты с истекшим сроком действия для установки защищенных соединений.
- Файл свойств CLI-клиента EKM доступен только для чтения.
  1. Проверьте атрибуты файла или права доступа к файлу, чтобы убедиться, что у пользователя CLI-клиента EKM есть права на доступ к этому файлу и его изменение.
- В файле свойств сервера EKM задан параметр Server.authMechanism = LocalOS, но необходимый файл из пакета EKMServicesAndSamples не был установлен или был установлен в неправильном месте.
  1. Подробные сведения о проверке подлинности см. в файле readme, который входит в пакет EKMServiceAndSamples.

Отладка сервера диспетчера ключей

Большинство проблем, возникающих при работе диспетчера ключей, связано с конфигурацией и запуском сервера диспетчера ключей. Сведения о задании свойства отладки см. в приложении B, "Файл конфигурации по умолчанию".

Если Encryption Key Manager не запускается, проверьте брандмауэр.

Возможно, аппаратный или программный брандмауэр препятствует доступу Encryption Key Manager к порту.

Сервер EKM не был запущен. Не удается загрузить или найти файл конфигурации EKM.properties.

1. Эта ошибка возникает при запуске KMSAdminCmd или EKMLaunch без указания полного пути к файлу свойств **KeyManagerConfig.properties**, когда этот файл находится не в каталоге, заданном по умолчанию.

   В Windows путь по умолчанию - **C:/Program Files/IBM/KeyManagerServer/**.

   На платформах Linux путь по умолчанию - **/opt/ibm/KeyManagerServer/**.
2. Введите еще раз команду для запуска KMSAdminCmd, включив в нее полный путь к файлу **KeyManagerConfig.properties**. Дополнительные сведения см. в приложении B (“Файлы свойств конфигурации Encryption Key Manager”).

Сервер EKM не был запущен. В файле конфигурации следует задать имя XML-файла метаданных.

В файле конфигурации отсутствует запись Audit.metadata.file.name.

Чтобы решить эту проблему, добавьте свойство Audit.metadata.file.name в файл конфигурации **KeyManagerConfig.properties**.

Не удается запустить EKM.Mykeys. Система не может найти указанный файл.

1. Это сообщение об ошибке выводится, когда записи о хранилище ключей в файле **KeyManagerConfig.properties** не указывают на существующий файл.
2. Для решения этой проблемы убедитесь, что следующие записи в файле **KeyManagerConfig.properties** указывают на существующие файлы хранилища ключей:

   Admin.ssl.keystore.name

   TransportListener.ssl.truststore.name

   TransportListener.ssl.keystore.name

   Admin.ssl.truststore.name

   Дополнительные сведения см. в приложении B (“Файлы свойств конфигурации Encryption Key Manager”).

Не удается запустить EKM. Файл не существует = safkeyring://xxx/yyy

Причиной ошибки может быть указание неправильного поставщика в переменной IJO сценария оболочки среды Encryption Key Manager.

Используйте для хранилищ ключей JCECCARACFKS:

```
-Djava.protocol.handler.pkgs=com.ibm.crypto.hdwrCCA.provider
```

и для хранилищ ключей JCERACFKS:

```
-Djava.protocol.handler.pkgs=com.ibm.crypto.provider
```

Не удается запустить EKM. Совершена попытка несанкционированного доступа к хранилищу ключей или введен неверный пароль.

1. Эта ошибка появляется, если для одной или нескольких записей в файле свойств (см. приложение B, “Файлы свойств конфигурации Encryption Key Manager”) указаны неправильные значения:

   config.keystore.password (соответствует файлу config.keystore.file)

   admin.keystore.password (соответствует admin.keystore.name)

   transportListener.keystore.password (соответствует transportListener.keystore.name)
2. Это сообщение может также появиться при вводе неверного пароля, когда он запрашивается при запуске сервера.
3. Если в конфигурации не хранится ни одного пароля, приглашение будет выводиться до трех раз, если все три записи хранилищ ключей в файле свойств уникальны. Если все три записи в файле свойств совпадают, приглашение ввести пароль появляется один раз.

Не удается запустить EKM. Недопустимый формат хранилища ключей.

1. Эта ошибка может произойти, если в одной или нескольких записях хранилищ ключей в файле свойств задан неправильный тип хранилища ключей.
2. Если все записи хранилищ ключей в файле свойств указывают на один и тот же файл, Encryption Key Manager использует значение config.keystore.type в качестве типа хранилища ключей для всех хранилищ.
3. Если в файле свойств не указан тип для какого-либо хранилища ключей, Encryption Key Manager присваивает этому параметру значение `jceks`.

Не удается запустить сервер. Не запущен поток-приемник.

Эта ошибка может произойти по ряду причин.

1. Следующие две записи в файле **KeyManagerConfig.properties** указывают на один и тот же порт:

   TransportListener.ssl.port

   TransportListener.tcp.port

   Каждый транспортный приемник следует настроить на прием со своего собственного порта.
2. Одна из двух записей или обе записи настроены на порт, который уже используется другой службой, выполняемой в той же системе, что и сервер диспетчера ключей. Найдите порты, не занятые другими службами, и используйте их для настройки сервера диспетчера ключей.
3. В средах под управлением ОС Linux эта ошибка может произойти, если один или оба порта имеют номер ниже 1024 и пользователь, запускающий сервер диспетчера ключей, не обладает правами привилегированного пользователя. Измените записи транспортных приемников в файле **KeyManagerConfig.properties** для использования портов с номерами выше 1024.

“[Fatal Error] :-1:-1: Premature end of file.” - сообщение в файле native_stderr.log.

Это сообщение появляется, если Encryption Key Manager загружает пустой файл групп ключей. Данное сообщение создает синтаксический анализатор XML, не препятствуя запуску Encryption Key Manager, если последний не настроен на использование групп ключей, а файл, указанный в свойстве config.keygroup.xml.file в файле **KeyManagerConfig.properties** (файл свойств сервера Encryption Key Manager), поврежден.

Ошибка: не удается найти Secretkey в конфигурации хранилища ключей с псевдонимом alias:MyKey.

Запись symmetricKeySet в файле свойств содержит псевдоним ключа, не существующий в файле config.keystore.file.

Чтобы решить эту проблему, измените запись symmetricKeySet в файле конфигурации так, чтобы она содержала только псевдонимы из файла хранилища ключей, на который указывает запись config.keystore.file в файле **KeyManagerConfig.properties**, ИЛИ добавьте недостающий симметричный ключ в хранилище ключей. Дополнительные сведения см. в приложении B (“Файлы свойств конфигурации Encryption Key Manager”).

В symmetricKeySet отсутствуют симметричные ключи, накопители LTO не поддерживаются.

Это информационное сообщение. Сервер Encryption Key Manager сохранит возможность для запуска, но накопители LTO данным экземпляром сервера Encryption Key Manager поддерживаться не будут. Это не представляет собой проблемы, если в конфигурации отсутствуют накопители LTO, настроенные для работы с данным Encryption Key Manager.

Сообщения об ошибках Encryption Key Manager

В этом разделе описываются сообщения приложения Encryption Key Manager (EKM) об ошибках, которые содержат данные, относящиеся к накопителям. Обычно они называются кодами неисправности (fault symptom code, FSC). Таблица содержит номер ошибки, краткое описание ошибки и действия, которые необходимо выполнить для ее исправления. Сведения о задании свойства отладки см. в приложении B ("Файл конфигурации по умолчанию").

Таблица 6-1. Отчеты Encryption Key Manager об ошибках

Номер ошибки	Описание	Меры по устранению
EE02	Encryption Read Message Failure: DriverErrorNotifyParameterError: "Bad ASC & ASCQ received. ASC & ASCQ does not match with either of Key Creation/Key Translation/Key Aquisition operation."	Запрашиваемая накопителем на магнитной ленте операция не поддерживается. Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Проверьте версию встроенного ПО накопителя или прокси-сервера и при необходимости обновите его до последней версии. Включите управление отладкой на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE0F	Encryption logic error: Internal error: "Unexpected error. Internal programming error in EKM."	Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Проверьте версию встроенного ПО накопителя или прокси-сервера и при необходимости обновите его до последней версии. Включите управление отладкой на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
	Error: Hardware error from call CSNDDSV returnCode 12 reasonCode 0.	Если используется аппаратное шифрование, убедитесь, что запущено программное обеспечение ICSF.
EE23	Encryption Read Message Failure: Internal error: "Unexpected error....."	Сообщение, полученное от накопителя или прокси-сервера, не удалось проанализировать из-за ошибки общего характера. Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Включите отладку на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.

Таблица 6-1. Отчеты Encryption Key Manager об ошибках (продолжение)

Номер ошибки	Описание	Меры по устранению
EE25	Encryption Configuration Problem: Errors that are related to the drive table occurred.	Если в файле KeyManagerConfig.properties определен параметр config.drivetable.file.url, убедитесь, что его значение задано правильно. На сервере Encryption Key Manager выполните команду `listdrives -drivename <имя_накопителя>`, чтобы проверить правильность конфигурации накопителя (например, правильность серийного номера, псевдонима и сертификатов накопителя). Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Проверьте версию встроенного ПО накопителя или прокси-сервера и при необходимости обновите его до последней версии. Включите управление отладкой и повторите операцию. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE29	Encryption Read Message Failure: Invalid signature	Сообщение, полученное от накопителя или прокси-сервера, не соответствует своей подписи. Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Включите отладку на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE2B	Encryption Read Message Failure: Internal error: "Either no signature in DSK or signature in DSK can not be verified."	Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Проверьте версию встроенного ПО накопителя или прокси-сервера и при необходимости обновите его до последней версии. Включите управление отладкой на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.

Таблица 6-1. Отчеты Encryption Key Manager об ошибках (продолжение)

Номер ошибки	Описание	Меры по устранению
EE2C	Encryption Read Message Failure: QueryDSKParameterError: "Error parsing a QueryDSKMessage from a device. Unexpected dsk count or unexpected payload."	Запрашиваемая накопителем на магнитной ленте операция не поддерживается приложением Encryption Key Manager. Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Проверьте версию встроенного ПО накопителя или прокси-сервера и при необходимости обновите его до последней версии. Включите управление отладкой на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE2D	Encryption Read Message Failure: Invalid Message Type	Сообщение было получено приложением Encryption Key Manager вне очереди или не может быть обработано. Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Включите отладку на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE2E	Encryption Read Message Failure: Internal error: Invalid signature type	Тип подписи сообщения, полученного от накопителя или прокси-сервера, является недопустимым. Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Включите отладку на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE30	Prohibited request.	Запрошенная операция не поддерживается накопителем на магнитной ленте. Введите правильную команду, поддерживаемую целевым ленточным накопителем.

Таблица 6-1. Отчеты Encryption Key Manager об ошибках (продолжение)

Номер ошибки	Описание	Меры по устранению
EE31	Encryption Configuration Problem: Errors that are related to the keystore occurred.	Проверьте метки ключей, которые вы пытаетесь использовать или настроенные по умолчанию. Список сертификатов, доступных приложению Encryption Key Manager, можно получить с помощью команды listcerts. Если известно, что используются значения по умолчанию, на сервере Encryption Key Manager выполните команду listdrives -drivename *имя_накопителя*, чтобы проверить правильность конфигурации накопителя (например, правильность серийного номера накопителя, а также связанных с ним псевдонимов и меток ключей). Если интересующий вас накопитель не имеет связанных с ним псевдонимов или меток ключей, проверьте значения параметров default.drive.alias1 и default.drive.alias2. Если это не помогло или псевдоним либо метка ключа существует, проанализируйте журналы отладки и обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа в разделе “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.
EE32	Keystore-related problem.	Наиболее вероятная причин: либо данная лента была зашифрована на другом экземпляре Encryption Key Manager с другими ключами, либо ключ, использованный для ее шифрования, был переименован или удален из хранилища ключей. Выполните команду `list -keysym` и убедитесь, что псевдоним запроса находится в хранилище ключей.
EEE1	Encryption logic error: Internal error: "Unexpected error: EK/EEDK flags conflict with subpage."	Убедитесь, что используется последняя версия приложения Encryption Key Manager (информацию о последней версии см. в разделе “Загрузка последней версии ISO-образа диспетчера ключей” на стр. 3-1). Проверьте версию встроенного ПО накопителя или прокси-сервера и при необходимости обновите его до последней версии. Включите отладку на сервере диспетчера ключей. Попробуйте воссоздать неполадку и проанализировать информацию из журналов отладки. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.

Таблица 6-1. Отчеты Encryption Key Manager об ошибках (продолжение)

Номер ошибки	Описание	Меры по устранению
EF01	Encryption Configuration Problem: "Drive not configured."	Накопитель, который пытается взаимодействовать с приложением Encryption Key Manager, отсутствует в таблице накопителей. Если в файле KeyManagerConfig.properties определен параметр config.drivetable.file.url, убедитесь, что его значение задано правильно. Чтобы проверить наличие накопителя в списке, запустите команду `listdrives`. Если накопитель отсутствует в списке, используйте команду `adddrive`, чтобы добавить накопитель в список вручную и указать правильную информацию, или команду `modconfig`, чтобы присвоить свойству drive.acceptUnknownDrives значение true. Включите управление отладкой и повторите операцию. Если проблема возникнет снова, обратитесь к главе “Обращение в Dell” раздела “Прочтите это прежде всего” в начале данного документа за сведениями о получении помощи по техническим вопросам.

Сообщения

Следующие сообщения могут генерироваться в Encryption Key Manager и выводиться на консоль администратора.

Config File not Specified (Не указан файл конфигурации)

Текст

```
Не указан файл конфигурации: при запуске EKM не указан файл конфигурации
диспетчера ключей.
```

Описание

Команда KMSAdmin требует ввода имени файла конфигурации в качестве параметра командной строки.

Ответ системы

Программа останавливается.

Действие оператора

Повторите команду, указав имя файла конфигурации.

Failed to Add Drive (Не удалось добавить накопитель)

Текст

```
Не удалось добавить накопитель. Накопитель уже существует.
```

Описание

При выполнении команды **adddrive** возникла ошибка, поскольку данный накопитель уже включен в конфигурацию Encryption Key Manager и указан в таблице накопителей.

Действие оператора

Выполните команду **listdrives**, чтобы проверить, включен ли данный накопитель в конфигурацию Encryption Key Manager. Если накопитель уже существует, конфигурацию накопителя можно изменить с помощью команды **moddrive**. Для получения дополнительной информации используйте команду **help**.

Failed to Archive the Log File (Не удается поместить в архив файл журнала)

Текст

```
Не удается поместить в архив файл журнала.
```

Описание

Файл журнала не может быть переименован.

Действие оператора

Проверьте права доступа к файлу и наличие свободного места на данном накопителе.

Failed to Delete the Configuration (Не удается удалить конфигурацию)

Текст

```
Не удается выполнить команду "modconfig".
```

Описание

Не удается удалить конфигурацию Encryption Key Manager с помощью команды **modconfig**.

Действие оператора

Проверьте синтаксис указанной команды с помощью команды **help** и правильность введенных параметров. Для получения дополнительных сведений ознакомьтесь с журналами аудита.

Failed to Delete the Drive Entry (Не удается удалить запись о накопителе)

Текст

```
Не удается выполнить команду "deldrive".
```

Описание

Не удается удалить запись о накопителе из таблицы накопителей с помощью команды **deldrive**.

Действие оператора

Проверьте синтаксис указанной команды с помощью команды **help** и правильность введенных параметров. При помощи команды **listdrives** убедитесь, что накопитель включен в конфигурацию Encryption Key Manager. Для получения дополнительных сведений ознакомьтесь с журналами аудита.

Failed to Import (Не удается импортировать)

Текст

```
Не удается выполнить команду "import".
```

Описание

Не удается импортировать таблицу накопителей или файлы конфигурации.

Ответ системы

Сервер Encryption Key Manager не запускается.

Действие оператора

Убедитесь, что заданный URL-адрес существует и разрешает чтение. Проверьте синтаксис указанной команды с помощью команды **help**. Проверьте правильность введенных параметров и повторите попытку.

Failed to Modify the Configuration (Не удается изменить конфигурацию)

Текст

```
Не удается выполнить команду "modconfig".
```

Описание

Не удается изменить конфигурацию Encryption Key Manager с помощью команды **modconfig**.

Действие оператора

Проверьте синтаксис указанной команды с помощью команды **help** и правильность введенных параметров. Для получения дополнительных сведений ознакомьтесь с журналами аудита.

File Name Cannot be Null (Имя файла не может быть пустым)

Текст

```
Не указано имя файла журнала аудита.
```

Описание

В свойствах конфигурации Encryption Key Manager не было задано имя файла журнала аудита. Это обязательный параметр конфигурации.

Ответ системы

Программа останавливается.

Действие оператора

Убедитесь, что в файле свойств конфигурации EKM определено свойство `Audit.handler.file.name`, и попробуйте перезапустить Encryption Key Manager.

File Size Limit Cannot be a Negative Number (Максимальный размер файла не может быть отрицательным числом)

Текст

```
Максимальный размер файла журнала аудита не может быть отрицательным
числом.
```

Описание

Значение свойства `Audit.handler.file.size` в файле конфигурации Encryption Key Manager должно быть положительным числом.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Задайте допустимое число для свойства `Audit.handler.file.size` и попробуйте перезапустить Encryption Key Manager.

No Data to be Synchronized (Нет данных для синхронизации)

Текст

```
Не удается обнаружить данные, подлежащие синхронизации с помощью команды
sync.
```

Описание

При использовании команды sync не удается обнаружить данные, подлежащие синхронизации.

Действие оператора

С помощью команды *config.drivetable.file.url* убедитесь, что указанный файл конфигурации существует и что таблица накопителей в этом файле настроена правильно. Проверьте синтаксис с помощью команды **help** и повторите команду **sync**.

Invalid Input (Введены недопустимые данные)

Текст

```
Введены недопустимые в интерфейсе командной строки параметры.
```

Описание

Возможно, введенная команда содержит синтаксическую ошибку.

Действие оператора

Проверьте правильность вводимой команды. Проверьте синтаксис указанной команды с помощью команды **help**. Проверьте правильность введенных параметров и повторите попытку.

Invalid SSL Port Number in Configuration File (Недопустимый номер порта SSL в файле конфигурации)

Текст

```
В файле конфигурации задан недопустимый номер порта SSL.
```

Описание

Номер порта SSL, заданный в файле конфигурации, является недопустимым.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Задайте допустимый номер порта для свойства `TransportListener.ssl.port` в файле конфигурации, используемом при запуске Encryption Key Manager, и попробуйте выполнить повторный запуск.

Invalid TCP Port Number in Configuration File (Недопустимый номер порта TCP в файле конфигурации)

Текст

```
В файле конфигурации EKM задан недопустимый номер порта TCP.
```

Описание

Номер порта TCP, заданный в файле конфигурации, является недопустимым.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Задайте допустимый номер порта для свойства `TransportListener.tcp.port` в файле конфигурации, используемом при запуске Encryption Key Manager, и попробуйте выполнить повторный запуск. Значение по умолчанию для номера порта TCP - 3801.

Must Specify SSL Port Number in Configuration File (Необходимо указать номер порта SSL в файле конфигурации)

Текст

```
Номер порта SSL не указан в файле свойств.
```

Описание

В файле свойств конфигурации обязательно следует указать номер порта SSL. Он используется для связи серверов Encryption Key Manager в среде с несколькими серверами.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Укажите допустимый номер порта для свойства `TransportListener.ssl.port` и попробуйте перезапустить Encryption Key Manager.

Must Specify TCP Port Number in Configuration File (Необходимо указать номер порта TCP в файле конфигурации)

Текст

```
Номер порта TCP не указан в файле свойств.
```

Описание

В файле свойств конфигурации обязательно следует указать номер порта TCP. Он используется для связи накопителя и Encryption Key Manager.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Укажите допустимый номер порта для свойства `TransportListener.tcp.port` и попробуйте перезапустить Encryption Key Manager. Значение по умолчанию для номера порта TCP - 3801.

Server Failed to Start (Не удается запустить сервер)

Текст

```
Не удается запустить сервер EKM.
```

Описание

Не удается запустить сервер Encryption Key Manager из-за проблем с конфигурацией.

Действие оператора

Проверьте параметры в файле конфигурации. Для получения дополнительных сведений ознакомьтесь с журналами аудита.

Sync Failed (Сбой синхронизации)

Текст

```
Не удалось выполнить команду "sync".
```

Описание

Произошел сбой операции sync при синхронизации данных между двумя серверами Encryption Key Manager.

Действие оператора

Убедитесь, что для удаленного сервера Encryption Key Manager указан правильный IP-адрес и к нему можно получить доступ. Убедитесь в наличии файла конфигурации и в правильности данных, содержащихся в таблице накопителей. Проверьте синтаксис команды **sync** с помощью команды **help**. Проверьте журналы, чтобы получить дополнительные сведения.

The Specified Audit Log File is Read Only (Указанный файл журнала аудита доступен только для чтения)

Текст

```
Запись данных в файл журнала аудита невозможна.
```

Описание

Файл журнала аудита из конфигурации Encryption Key Manager, на который ссылается свойство `Audit.handler.file.name`, невозможно открыть для записи данных.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Проверьте разрешения, заданные для указанного файла аудита и каталога, и попытайтесь перезапустить Encryption Key Manager.

Unable to Load the Admin Keystore (Не удается загрузить хранилище ключей администратора)

Текст

```
Невозможно загрузить хранилище ключей для администратора.
```

Описание

Хранилище ключей администратора, заданное для Encryption Key Manager, загрузить невозможно. Хранилище ключей администратора используется при взаимодействии серверов Encryption Key Manager в среде с несколькими серверами.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Проверьте настройку файла конфигурации. Убедитесь, что свойства `admin.keystore.file`, `admin.keystore.provider` и `admin.keystore.type` в файле конфигурации Encryption Key Manager заданы правильно (см. приложение B) и что файл хранилища ключей существует, а также имеются права на чтение данного файла. Проверьте правильность пароля для хранилища ключей администратора, введенного с использованием свойства `admin.keystore.password` или из командной строки. Попробуйте повторно запустить Encryption Key Manager.

Unable to load the keystore (Не удается загрузить хранилище ключей)

Текст

```
Не удается загрузить хранилище ключей для EKM.
```

Описание

Хранилище ключей, указанное для Encryption Key Manager, загрузить невозможно.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Проверьте настройку файла конфигурации. Убедитесь, что свойства `config.keystore.file`, `config.keystore.provider` и `config.keystore.type` в файле конфигурации Encryption Key Manager заданы правильно и что файл хранилища ключей существует, а также имеются права на чтение данного файла. Проверьте правильность пароля, введенного для хранилища ключей Encryption Key Manager с использованием свойства `config.keystore.password` или из командной строки. Попробуйте выполнить повторный запуск.

Unable to Load the Transport Keystore (Не удается загрузить хранилище транспортных ключей)

Текст

```
Невозможно загрузить хранилище транспортных ключей.
```

Описание

Хранилище транспортных ключей, заданное для Encryption Key Manager, загрузить невозможно. Хранилище транспортных ключей используется при взаимодействии серверов Encryption Key Manager с клиентами в среде с несколькими серверами.

Ответ системы

Encryption Key Manager не запускается.

Действие оператора

Проверьте настройку файла конфигурации. Убедитесь, что свойства `transport.keystore.file`, `transport.keystore.provider` и `transport.keystore.type` в файле конфигурации Encryption Key Manager заданы правильно и что файл хранилища ключей существует, а также имеются права на чтение данного файла. Проверьте правильность пароля, введенного для хранилища транспортных ключей с использованием свойства `transport.keystore.password` или из командной строки. Попробуйте повторно запустить Encryption Key Manager.

Unsupported Action (Неподдерживаемое действие)

Текст

`Пользователь ввел с помощью интерфейса командной строки действие, которое не поддерживается для EKM.`

Описание

Действие, введенное для команды **sync**, не поддерживается или не распознается приложениемEncryption Key Manager. Допустимые действия - merge или rewrite.

Действие оператора

Проверьте синтаксис команды с помощью команды **help** и повторите попытку.

Глава 7. Протоколы аудита

Примечание: Форматы протокола аудита, описанные в этой главе, не являются программируемыми интерфейсами. Формат этих протоколов может меняться от версии к версии. Формат протокола описан в этой главе на случай, если потребуется какой-либо синтаксический анализ протоколов аудита.

Общие сведения об аудите

Если во время обработки запросов приложением Encryption Key Manager происходят контролируемые события, подсистема аудита записывает протоколы аудита в текстовом формате в набор последовательных файлов. Подсистема аудита записывает данные в файл (каталог и имя файла являются настраиваемыми параметрами). Кроме того, можно указать максимальный размер этих файлов. По мере записи протоколов аудита в файл и достижения максимально допустимого размера используемого файла этот файл закрывается и переименовывается в соответствии с текущей временной меткой. После этого открывается другой файл и протоколы аудита записываются во вновь созданный файл. Таким образом, общий журнал протоколов аудита разделен на несколько файлов определенного размера, порядок расположения которых определяется временной меткой. Эта метка отражает момент превышения заданного максимального размера файла аудита.

Чтобы предотвратить чрезмерное увеличение количества информации, хранящейся в общем файле аудита (охватывающем все созданные последовательные файлы), и превышение объема свободного места, доступного в файловой системе, подумайте о создании сценария или программы для мониторинга набора файлов, хранящихся в заданном контейнере/папке/каталоге аудита. По мере закрытия файлов и присвоения им имен в соответствии с временной меткой содержимое этих файлов должно быть скопировано и добавлено в определенный пользователем каталог долгосрочного хранения журналов, а затем удалено из исходного каталога. Будьте внимательны: не удаляйте и не изменяйте файл, в который записываются протоколы аудита во время работы приложения Encryption Key Manager (в имени такого файла нет метки времени).

Параметры конфигурации аудита

В файле конфигурации Encryption Key Manager для управления событиями, которые будут записаны в журнал аудита, а также для указания месторасположения и максимального размера файлов журнала аудита используются следующие параметры.

Audit.event.types

Синтаксис

Audit.event.types={*тип*[;*тип*]}

Использование

Используется для указания типа аудита и событий, которые должны регистрироваться в журнале аудита. Возможны следующие значения данного параметра конфигурации.

all	Все типы событий
authentication	События, связанные с аутентификацией
data_synchronization	События, происходящие во время синхронизации информации серверов Encryption Key Manager
runtime	События, происходящие в процессе обработки операций и запросов, отправленных приложению Encryption Key Manager
configuration_management	События, происходящие при внесении изменений в конфигурацию
resource_management	События, происходящие при изменении параметров ресурсов (накопителей на магнитной ленте) в Encryption Key Manager

Примеры

Пример указания значения данного параметра конфигурации:

```
Audit.event.types=all
```

Другой пример:

```
Audit.event.types=authentication;runtime;resource_management
```

Audit.event.outcome

Синтаксис

Audit.event.outcome={*результат*[;*результат*]}

Использование

С помощью этого параметра можно указать, какие события подлежат аудиту: произошедшие в результате успешно выполненных операций, неуспешных операций или в обоих случаях. Укажите значение **success**, чтобы регистрировать события, произошедшие в результате успешно выполненных операций. Укажите значение **failure**, чтобы регистрировать события, произошедшие в результате неуспешных операций.

Примеры

Пример указания значения данного параметра конфигурации:

```
Audit.event.outcome=failure
```

Для выполнения аудита как успешных, так и неуспешных случаев задайте следующее значение:

```
Audit.event.outcome=success;failure
```

Audit.eventQueue.max

Синтаксис

Audit.eventQueue.max=*число_событий*

Использование

С помощью этого параметра можно задать максимальное число объектов событий, которые могут содержаться в очереди памяти. Этот параметр является

необязательным, но рекомендуется его использовать. По умолчанию задано нулевое значение.

Пример

```
Audit.eventQueue.max=8
```

Audit.handler.file.directory

Синтаксис

Audit.handler.file.directory=*имя_каталога*

Использование

С помощью этого параметра можно указать каталог, в который должны сохраняться файлы протоколов аудита. Если каталог не существует, Encryption Key Manager попытается создать его. Если операция не будет выполнена успешно, Encryption Key Manager не запустится. Рекомендуется создать каталог перед запуском Encryption Key Manager. Следует также заметить, что пользователь, от имени которого запускается Encryption Key Manager, должен иметь права на запись в указанный каталог.

Примеры

Чтобы указать каталог **/var/ekm/ekm1/audit**, используйте следующий синтаксис:

```
Audit.handler.file.directory=/var/ekm/ekm1/audit
```

Audit.handler.file.size

Синтаксис

Audit.handler.file.size=*размер_в_килобайтах*

Использование

С помощью этого параметра можно задать предельный размер файла аудита, при достижении которого запись данных в текущий файл прекращается и создается новый файл аудита. Обратите внимание, что фактический размер конечного файла аудита может превысить заданное значение на несколько байт, поскольку закрытие текущего файла происходит при превышении предельного размера.

Примеры

Чтобы задать максимальный размер файла равным приблизительно 2 МБ, используйте следующий синтаксис:

```
Audit.handler.file.size=2000
```

Audit.handler.file.name

Синтаксис

Audit.handler.file.name=*имя_файла*

Использование

С помощью этого параметра можно указать имя базового файла, которое будет использоваться в качестве базового имени при создании файлов журнала аудита в заданном каталоге аудита. Следует заметить, что в данном параметре нужно

указывать только имя базового файла, но не полный путь к нему. Полное имя файла журнала аудита состоит из базового имени и добавленного к нему значения времени создания этого файла.

В качестве иллюстрации можно рассмотреть пример, в котором параметру Audit.handler.file.name присвоено значение **ekm.log**. Полное имя файла будет выглядеть приблизительно следующим образом: `ekm.log.2315003554`. Добавленную строку можно использовать для определения порядка, в котором создавались файлы журнала аудита: большие числовые значения соответствуют более новым файлам журнала аудита.

Примеры

Присвоение значения **ekm.log** базовому имени осуществляется следующим образом:

```
Audit.handler.file.name=ekm.log
```

Audit.handler.file.multithreads

Синтаксис

Audit.handler.file.multithreads={**yes**|**true**|**no**|**false**}

Использование

При выборе значения **true** для записи данных события в журнал аудита используется отдельный поток. Это позволяет не прерывать текущий поток выполнения (операции) для завершения процедуры записи данных в журнал аудита. По умолчанию используются несколько потоков.

Примеры

Присвоение значения **true** базовому имени осуществляется следующим образом:

```
Audit.handler.file.multithreads=true
```

Audit.handler.file.threadlifespan

Синтаксис

Audit.handler.file.threadlifespan=*время_в_секундах*

Использование

С помощью этого параметра можно указать максимальное желаемое время, которое потребуется потоку для записи данных в журнал аудита. Это значение используется во время процедуры очистки ресурсов, позволяя завершить выполнение текущих потоков до того, как они будут прерваны. Если в течение времени, заданного параметром threadlifespan, выполнение фонового потока не завершилось, то во время процедуры очистки ресурсов его выполнение будет прервано.

Примеры

Чтобы задать время, которое, как ожидается, потребуется потоку для записи данных в журнал аудита, равным 10 секундам, используйте следующий синтаксис:

```
Audit.handler.file.threadlifespan=10
```

Формат протокола аудита

Для всех протоколов аудита используется похожий формат вывода, описанный в этом разделе. Все протоколы аудита содержат общую информацию, включая временную метку и тип протокола, а также информацию, связанную с произошедшим событием аудита. Общий формат протоколов аудита приведен ниже:

```
Тип_протокола_аудита:[
  timestamp=временная_метка
  имя_атрибута=значение_атрибута
  ...
  ]
```

Для каждого протокола в файле существует несколько строк. Первая строка протокола, начиная с первого символа в строке, содержит тип протокола аудита, за которым следует двоеточие (:) и открывающая квадратная скобка ([). Последующие строки, относящиеся к тому же протоколу аудита, имеют отступ в два (2) пробела, что повышает удобочитаемость протоколов в журнале. Последняя строка каждого протокола аудита содержит закрывающую квадратную скобку (]) с отступом в два (2) пробела. Количество строк в каждом протоколе аудита может различаться в зависимости от типа протокола и информации о дополнительных атрибутах, приведенной в протоколе аудита.

Метка времени присваивается протоколу аудита с учетом показаний системных часов компьютера, на котором запущено приложение Encryption Key Manager. Если эти протоколы необходимо соотнести с событиями, происходящими в других системах, то для обеспечения приемлемой точности синхронизации часов различных систем, входящих в одну среду, необходимо использовать одно из средств синхронизации времени.

Точки аудита в Encryption Key Manager

В зависимости от конфигурации приложение Encryption Key Manager может записывать протоколы аудита для множества событий, которые происходят во время обработки запросов. В этом разделе описываются доступные для аудита наборы событий, а также категории конфигурации протоколов аудита, которые необходимо указать для того, чтобы разрешить запись протоколов в файлы аудита (см. Табл. 7-1).

Таблица 7-1. Типы протоколов аудита, которые Encryption Key Manager записывает в файлы аудита

Тип протокола аудита	**Тип аудита**	**Описание**
Аутентификация	`аутентификация`	Предназначен для регистрации событий, связанных с аутентификацией
Синхронизация данных	`data_synchronization`	Предназначен для регистрации событий, связанных с обработкой данных синхронизации
Время выполнения	`runtime`	Предназначен для регистрации различных важных событий, связанных с обработкой данных, которые происходят на сервере Encryption Key Manager во время обслуживания запросов
Управление ресурсами	`resource_management`	Предназначен для регистрации в Encryption Key Manager изменений, вносимых в конфигурацию ресурсов

Таблица 7-1. Типы протоколов аудита, которые Encryption Key Manager записывает в файлы аудита (продолжение)

Тип протокола аудита	Тип аудита	Описание
Управление конфигурацией	`configuration_management`	Предназначен для регистрации изменений, вносимых в конфигурацию сервера Encryption Key Manager

Атрибуты протокола аудита

В следующем списке приведены атрибуты, используемые в каждом из типов протокола аудита.

Событие Authentication (Аутентификация)

Такие протоколы имеют следующий формат:

```
Authentication event:[
  timestamp=временная_метка
  event source=источник
  outcome=результат
  event type=SECURITY_AUTHN
  message=сообщение
  authentication type=тип
  users=пользователи
  ]
```

Обратите внимание, что значение параметра `message` отображается только в случае наличия соответствующей информации.

Событие Data Synchronization (Синхронизация данных)

Такие протоколы имеют следующий формат:

```
Data synchronization event:
  timestamp=временная_метка
  event source=источник
  outcome=результат
  event type=SECURITY_DATA_SYNC
  message=сообщение
  action=событие
  resource=ресурс
  user=пользователь
  ]
```

Обратите внимание, что значения параметров `message` и `user` отображаются только в случае наличия соответствующей информации.

Событие Runtime (Время выполнения)

Такие протоколы имеют следующий формат:

```
Runtime event:
  timestamp=временная_метка
  event source=источник
  outcome=результат
  event type=SECURITY_RUNTIME
  message=сообщение
  resource=ресурс
  action=событие
  user=пользователь
  ]
```

Обратите внимание, что значения параметров `message` и `user` отображаются только в случае наличия соответствующей информации.

Событие Resource Management (Управление ресурсами)

Такие протоколы имеют следующий формат:

```
Resource management event:
  timestamp=временная_метка
  event source=источник
  outcome=результат
  event type=SECURITY_MGMT_RESOURCE
  message=сообщение
  action=событие
  user=пользователь
  resource=ресурс
  ]
```

Обратите внимание, что значение параметра `message` отображается только в случае наличия соответствующей информации.

Событие Configuration Management (Управление конфигурацией)

Такие протоколы имеют следующий формат:

```
Configuration management event:
  timestamp=временная_метка
  event source=источник
  outcome=результат
  event type=SECURITY_MGMT_CONFIG
  message=сообщение
  action=событие
  command type=тип
  user=пользователь
  ]
```

Обратите внимание, что значение параметра `message` отображается только в случае наличия соответствующей информации.

Проверенные события

Табл. 7-2 описывает события, в связи с которыми создаются протоколы аудита. В данной таблице приведены типы протоколов аудита, создаваемых в момент возникновения события.

Таблица 7-2. Соответствие типов протоколов аудита контролируемым событиям

Контролируемое событие	**Тип протокола аудита**
Проверка подлинности пользователя успешно пройдена	authentication
Сбой проверки подлинности пользователя	authentication
Данные успешно переданы другому EKM	data_synchronization
Ошибка передачи данных другому EKM	data_synchronization
Команда sync обработана	data_synchronization
Ошибка обработки команды sync	data_synchronization
Начата обработка командной строки	runtime
Получена команда exit	runtime

Таблица 7-2. Соответствие типов протоколов аудита контролируемым событиям (продолжение)

Контролируемое событие	**Тип протокола аудита**
Введена неизвестная команда	runtime
Получено сообщение от накопителя	runtime
Ошибка обработки сообщения от накопителя	runtime
Информация об ошибке в сообщении, полученном от накопителя	runtime
Ошибка внесения информации, полученной от накопителя, в таблицу накопителей	runtime
Ошибка получения информации из таблицы накопителей	runtime
Ошибка получения информации из хранилища ключей	runtime
Ошибка обработки сертификата из хранилища ключей	runtime
Ошибка поиска секретного ключа в хранилище ключей	runtime
Ошибка расчета криптографических величин	runtime
Обмен сообщениями успешно выполнен	runtime
Начат обмен сообщениями	runtime
Начата обработка командной строки	runtime
Обнаружена неполадка при использовании криптографических служб	runtime
Обнаружен новый накопитель	runtime
Ошибка настройки накопителя в соответствии с таблицей накопителей	runtime
Обработка сообщений от накопителя успешно начата	runtime
Получена и обработана команда stopekm	runtime
Накопитель удален из таблицы накопителей	resource_management
Ошибка удаления накопителя из таблицы накопителей	resource_management
Таблица накопителей успешно импортирована	resource_management
Ошибка импорта таблицы накопителей	resource_management
Таблица накопителей успешно экспортирована	resource_management
Ошибка экспорта таблицы накопителей	resource_management
Команда listcerts успешно выполнена	resource_management
Накопитель успешно добавлен в таблицу накопителей	resource_management
Ошибка добавления накопителя в таблицу накопителей	resource_management
Команда listdrives успешно выполнена	resource_management
Ошибка обработки команды listdrives	resource_management

Таблица 7-2. Соответствие типов протоколов аудита контролируемым событиям (продолжение)

Контролируемое событие	**Тип протокола аудита**
Таблица накопителей успешно изменена	resource_management
Ошибка изменения таблицы накопителей	resource_management
Хранилище ключей успешно открыто	resource_management
Ошибка открытия хранилища ключей	resource_management
Изменено свойство конфигурации	configuration_management
Ошибка изменения свойства конфигурации	configuration_management
Свойство конфигурации удалено	configuration_management
Ошибка удаления свойства конфигурации	configuration_management
Конфигурация успешно импортирована	configuration_management
Ошибка импорта конфигурации	configuration_management
Конфигурация успешно экспортирована	configuration_management
Ошибка экспорта конфигурации	configuration_management
Команда listconfig успешно выполнена	configuration_management

Глава 8. Использование метаданных

Настройки Encryption Key Manager должны предусматривать создание XML-файла, в который будет записываться важнейшая информация по мере шифрования данных и их записи на ленту. В этом файле можно по серийному номеру тома найти псевдоним или метку ключа, использованные при шифровании этого тома. И наоборот, по псевдониму можно запрашивать список всех томов, связанных с этой меткой ключа или псевдонимом.

Примечание: Если файл метаданных не настроен, Encryption Key Manager не запустится.

В процессе шифрования Encryption Key Manager собирает следующие данные:

- серийный номер накопителя;
- глобальное имя накопителя;
- дата создания;
- псевдоним ключа 1;
- псевдоним ключа 2;
- DKi (идентификатор ключа данных);
- VolSer (серийный номер тома).

Когда собранные данные превышают определенный объем, они записываются в файл XML. Предельный объем данных можно задать в файле свойств Encryption Key Manager (KeyManagerConfig.properties). Значение по умолчанию - 100 записей. Когда запись в файл произведена, в нем можно осуществлять поиск данных во время работы Encryption Key Manager. Чтобы файл не слишком разрастался, по достижении определенного максимального размера запись автоматически продолжается в новый файл. Предельный размер файла, по достижении которого создается новый файл, также задается в свойствах Encryption Key Manager и по умолчанию равен 1 МБ. Сохраняются только текущий и предыдущий файлы. В файле свойств конфигурации Encryption Key Manager необходимо настроить следующие значения.

Audit.metadata.file.name
Имя файла XML, в котором сохраняются метаданные. Это обязательный параметр.

Audit.metadata.file.size
Максимальный размер файла в килобайтах, по достижении которого запись в текущий файл прекращается. Это необязательный параметр. Значение по умолчанию - 1024 (1 МБ).

Audit.metadata.file.cachecount
Количество записей, хранящихся в кэш-памяти до того, как будет произведена запись в файл метаданных. Это необязательный параметр. Значение по умолчанию - 100.

Формат XML-файла

Файл содержит записи в следующем формате.

```
<KeyUsageEvent>
 <DriveSSN>FVTDRIVE0000</driveSSN>     – серийный номер накопителя
 <VolSer>TESTER</volSer>                – серийный номер тома
```

<DriveWWN>57574E414D453030</driveWWN> – глобальное имя накопителя
<keyAlias2>cert2</keyAlias2> – псевдоним ключа 1
<keyAlias1>cert1</keyAlias1> – псевдоним ключа 2
<dateTime>Tue Feb 20 09:18:07 CST 2007</dateTime> – дата создания
</KeyUsageEvent>

Примечание. Для накопителей LTO 4 и LTO 5 файл будет содержать только запись <keyAlias1></keyAlias1> и DKi (идентификатор ключа данных).

Запросы к XML-файлу метаданных

Для запросов к XML-файлу метаданных используется инструмент EKMDataParser. Он анализирует XML-файл с помощью технологии Document Object Model (DOM) и не может быть запущен из интерфейса командной строки Encryption Key Manager. Он запускается следующим образом:

java com.ibm.keymanager.tools.EKMDataParser -filename
полный_путь_к_файлу_метаданных {**-volser** *серийный_номер_тома* | **-keyalias** *псевдоним*}

metadata_path
Это путь к каталогу файла метаданных, заданный для свойства Audit.metadata.file.name в файле **KeyManagerConfig.properties**.

-filename
Имя файла является обязательным параметром; это должно быть имя XML-файла метаданных. Оно обычно соответствует имени, заданному для свойства Audit.metadata.file.name в файле **KeyManagerConfig.properties**.

-volser
Серийный номер тома кассеты, сведения о котором необходимо найти в файле XML. Следует указать либо **-volser**, либо **-keyalias**.

-keyalias
Метка ключа или псевдоним, которые необходимо найти в файле XML. Следует указать либо **-volser**, либо **-keyalias**.

Пример

Если свойство имени файла метаданных (Audit.metadata.file.name) в файле **KeyManagerConfig.properties** задано как `metadata`, а файл находится в локальном каталоге, в котором выполняется Encryption Key Manager, следующая команда отфильтрует (отобразит) только записи XML, имеющие отношение к серийному номеру тома (volser) 72448:

```
<jvm_path>/bin/java com.ibm.keymanager.tools.EKMDataParser -filename metadata
    -volser 72448
```

Выходные данные будут иметь следующий формат:

Таблица 8-1. Формат вывода для запроса метаданных

keyalias1	keyalias2	volSer	dateTime	driveSSN	dki
cert1	cert2	72448	Wed Mar 14 10:31:32 CDT 2007	FVTDRIVE0004	

Восстановление поврежденного файла метаданных

При неправильном завершении работы Encryption Key Manager или сбое компьютера, на котором выполняется Encryption Key Manager, возможно повреждение файла

метаданных Encryption Key Manager. Повреждение файла метаданных возможно также при неправильном его редактировании или изменении. Повреждение не будет выявлено до тех пор, пока EKMDataParser не попытается выполнить анализ файла метаданных. При этом возможен сбой в работе EKMDataParser и вывод сообщения об ошибке, примерно такого:

```
[Fatal Error] EKMData.xml:290:16: The end-tag for element type "KeyUsageEvent" must
 end with a '>' delimiter.
org.xml.sax.SAXParseException: The end-tag for element type "KeyUsageEvent" must
 end with a '>' delimiter.
at org.apache.xerces.parsers.DOMParser.parse(Unknown Source)
at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown Source)
at javax.xml.parsers.DocumentBuilder.parse(Unknown Source)
at com.ibm.keymanager.tools.EKMDataParser.a(EKMDataParser.java:136)
at com.ibm.keymanager.tools.EKMDataParser.a(EKMDataParser.java:26)
at com.ibm.keymanager.tools.EKMDataParser.main(EKMDataParser.java:93)
```

Появление такой ошибки означает отсутствие закрывающего тега в элементе XML. Файл метаданных Encryption Key Manager можно восстановить, а затем выполнить с помощью EKMDataParser его повторный анализ.

1. Создайте резервную копию файла метаданных Encryption Key Manager.
2. Отредактируйте файл метаданных Encryption Key Manager.
3. В XML каждый фрагмент данных или событие должны быть заключены между открывающим и соответствующим ему закрывающим тегом.
   - Несколько примеров открывающих тегов:
     - <KeyUsageEvent>
     - <driveSSN>
     - <keyAlias1>
   - Несколько примеров закрывающих тегов:
     - </KeyUsageEvent>
     - </driveSSN>
     - </keyAlias1>
4. Просмотрите файл и найдите непарные теги. В сообщении об ошибке EKMDataParser выводит список элементов, у которых отсутствует закрывающий тег. Это должно несколько облегчить поиск.
5. Если непарный тег найден, временно удалите событие или добавьте теги, необходимые для правильного оформления события.
   - Например, в приведенном ниже фрагменте файла метаданных Encryption Key Manager первый элемент KeyUsageEvent не имеет закрывающего тега:

```
<KeyUsageEvent>
<driveSSN>001310000109</driveSSN>
<volSer>      </volSer>
<driveWWN>5005076312418B07</driveWWN>
<keyAlias1>key0000000000000000F</keyAlias1>
<dki>6B65790000000000000000000F</dki>
<dateTime>Thu Aug 30 09:50:53 MDT 2007</dateTime>
<KeyUsageEvent>
<driveSSN>001310000100</driveSSN>
<volSer>      </volSer>
<driveWWN>5005076312418ABB</driveWWN>
<keyAlias1>key00000000000000000</keyAlias1>
<dki>6B6579000000000000000000</dki>
<dateTime>Thu Sep 06 16:49:39 MDT 2007</dateTime>
</KeyUsageEvent>
```

Добавление строки </KeyUsageEvent> между строками <dateTime>Thu Aug 30 09:50:53 MDT 2007</dateTime> и <KeyUsageEvent> позволит дополнить непарный первый элемент <KeyUsageEvent>.

После устранения ошибок в файле EKMDataParser сможет успешно выполнить анализ данных.

Приложение A. Примеры файлов

Пример сценария демона запуска

Внимание: Невозможно переоценить важность обеспечения сохранности данных хранилища ключей. Потеря доступа к хранилищу ключей приведет к невозможности расшифровки зашифрованных данных накопителей. Убедитесь в том, что вы сохранили информацию о хранилище ключей и паролях.

Платформы Linux

Ниже представлен пример сценария, который позволяет проверенным способом запускать приложение EKM в фоновом режиме. В результате выполнения этого сценария происходит запуск приложения EKM и задание пароля хранилища ключей (*пароль_хранилища_ключей*) внутри сценария. В этом случае пароль хранилища ключей не нужно указывать в файле конфигурации EKM (см. примечание ниже). Файл сценария должен содержать следующие строки:

```
java com.ibm.keymanager.KMSAdminCmd KeyManagerConfig.properties <<EOF
startekm
пароль_хранилища_ключей
status
EOF
```

Примечание: Если пароль хранилища ключей задается в приложении EKM посредством сценария (это означает, что файл конфигурации EKM не содержит пароля хранилища ключей), то при создании резервной копии данных EKM файлы (файл конфигурации, таблица накопителей и файл резервной копии хранилища ключей) не обязательно должны храниться в защищенном месте. Однако для сценария, содержащего пароль хранилища ключей, **необходимо** выбрать безопасный и отказоустойчивый метод хранения (например, хранить несколько копий в разных местах). Не забывайте, что пароль хранилища ключей является конфиденциальной информацией. Способы защищенного резервного копирования файла сценария аналогичны способам создания резервной копии файла конфигурации, содержащего пароль хранилища ключей. Однако резервные копии сценариев можно хранить и перемещать независимо от файлов резервной копии данных EKM, что обеспечивает дополнительную безопасность и секретность. Наконец, важно подчеркнуть, что в любом случае необходимо обеспечить безопасность и отказоустойчивость хранения пароля хранилища ключей (в сценарии или в файле конфигурации EKM) и гарантировать возможность его восстановления в любой ситуации. **Потеря всех копий пароля хранилища ключей может привести к потере всех ключей в хранилище, причем в этом случае возможность восстановления исключена**.

Примеры файлов конфигурации

Ниже приведен пример файла свойств EKM, в котором все записи, относящиеся к хранилищу ключей, указывают на одно и то же программное хранилище ключей.

```
Admin.ssl.keystore.name = /keymanager/testkeys
Admin.ssl.keystore.type = jceks
Admin.ssl.truststore.name = /keymanager/testkeys
Admin.ssl.truststore.type = jceks
Audit.event.outcome = success,failure
Audit.event.types = all
Audit.eventQueue.max = 0
Audit.handler.file.directory = /keymanager/audit
Audit.handler.file.name = kms_audit.log
Audit.handler.file.size = 10000
Audit.metadata.file.name = /keymanager/metafile.xml
config.drivetable.file.url = FILE:///keymanager/drivetable
config.keystore.file = /keymanager/testkeys
config.keystore.provider = IBMJCE
config.keystore.type = jceks
fips = Off
TransportListener.ssl.ciphersuites = JSSE_ALL
TransportListener.ssl.clientauthentication = 0
TransportListener.ssl.keystore.name = /keymanager/testkeys
TransportListener.ssl.keystore.type = jceks
TransportListener.ssl.port = 443
TransportListener.ssl.protocols = SSL_TLS
TransportListener.ssl.truststore.name = /keymanager/testkeys
TransportListener.ssl.truststore.type = jceks
TransportListener.tcp.port = 3801
```

Ниже приведен пример файла свойств EKM, в котором все записи, относящиеся к хранилищу ключей, указывают на разные программные хранилища ключей. Строки, выделенные полужирным шрифтом, отличаются от строк в первом примере.

```
Admin.ssl.keystore.name = /keymanager/adminkeys.jceks
Admin.ssl.keystore.type = jceks
Admin.ssl.truststore.name = /keymanager/admintrustkeys
Admin.ssl.truststore.type = jceks
Audit.event.outcome = success,failure
Audit.event.types = all
Audit.eventQueue.max = 0
Audit.handler.file.directory = /keymanager/audit
Audit.handler.file.name = kms_audit.log
Audit.handler.file.size = 10000
Audit.metadata.file.name = /keymanager/metafile.xml
config.drivetable.file.url = FILE:///keymanager/drivetable
config.keystore.file = /keymanager/drive.keys
config.keystore.provider = IBMJCE
config.keystore.type = jceks
fips = Off
TransportListener.ssl.ciphersuites = JSSE_ALL
TransportListener.ssl.clientauthentication = 0
TransportListener.ssl.keystore.name = /keymanager/sslkeys
TransportListener.ssl.keystore.type = jceks
TransportListener.ssl.port = 443
TransportListener.ssl.protocols = SSL_TLS
TransportListener.ssl.truststore.name = /keymanager/ssltrustkeys
TransportListener.ssl.truststore.type = jceks
TransportListener.tcp.port = 3801
```

Приложение B. Файлы свойств конфигурации Encryption Key Manager

Для работы Encryption Key Manager требуются два файла свойств конфигурации: один для сервера Encryption Key Manager и один для клиента CLI. Каждый из этих файлов обрабатывается и анализируется как файл загрузки Java.util.Properties, что налагает определенные ограничения на формат и спецификацию свойств.

- Свойства конфигурации записываются по одному на строку. Значение (или значения) определенного свойства занимает строку до конца.
- Значения свойств (например, пароли), содержащие пробелы, не нужно заключать в кавычки.
- Пароли хранилища ключей не должны быть длиннее 127 символов.
- Пробел, случайно добавленный в конец строки, может интерпретироваться как часть значения свойства.

Примеры файлов свойств конфигурации доступны для загрузки по адресу **http://support.dell.com** - см. файл EKMServicesandSamples.

Файл свойств конфигурации сервера Encryption Key Manager

Ниже приведен полный набор параметров файла конфигурации сервера Encryption Key Manager (KeyManagerConfig.properties). Порядок параметров свойств в файле не имеет значения. Файл может содержать комментарии. Чтобы добавить комментарий, используйте символ “#” в первом столбце строки.

Примечание: Изменения, внесенные в файл KeyManagerConfig.properties, могут быть потеряны при завершении работы. Поэтому перед внесением изменений в свойства конфигурации убедитесь, что сервер Encryption Key Manager остановлен. Чтобы остановить сервер Encryption Key Manager, из CLI-клиента выполните команду **stopekm**. Внесенные вами изменения вступят в силу после перезагрузки сервера Encryption Key Manager.

Admin.ssl.ciphersuites = *значение*

Определяет кодовые наборы, используемые для связи между серверами Encryption Key Manager. В кодовом наборе содержится описание криптографических алгоритмов и протоколов связи, которые используются протоколами Transport Layer Security (TLS) и Secure Sockets Layer (SSL) для передачи данных.

Обязательный параметр?
Дополнительный.

Значения Любые кодовые наборы, поддерживаемые IBMJSSE2.

Значение по умолчанию
JSSE_ALL

Admin.ssl.keystore.name = *значение*

Имя базы данных пар ключей и сертификатов, используемых клиентом Secure Socket Layer при выполнении операций, например, команды **sync** между серверами Encryption Key Manager. При выполнении синхронизации клиент Secure Sockets предоставляет серверу Secure Sockets сертификат именно из этого хранилища ключей.

Обязательный параметр?
Дополнительный. Используется только с командой **sync**. По умолчанию значение этого свойства равно значению свойства **config.keystore.file**.

Admin.ssl.keystore.password = пароль
Пароль для доступа к хранилищу ключей, заданному свойством Admin.ssl.keystore.name.

Обязательный параметр?
Дополнительный. Если значение этого свойства не задано, при запуске Encryption Key Manager может появиться соответствующее приглашение. Если значение этого свойства задано, в целях безопасности оно скрывается, а сама строка в файле свойств заменяется новой строкой - Admin.ssl.keystore.password.obfuscated.

Admin.ssl.keystore.type = *значение*
Тип используемого хранилища ключей.

Обязательный параметр?
Дополнительный.

Значение по умолчанию
jceks

Admin.ssl.protocols = *значение*
Протоколы безопасности.

Обязательный параметр?
Дополнительный.

Значения SSL_TLS | SSL | TLS

Значение по умолчанию
SSL_TLS

Admin.ssl.timeout = *значение*
Определяет время, в течение которого сокет ожидает выполнения команды read(), прежде чем генерировать исключительную ситуацию SocketTimeoutException.

Обязательный параметр?
Дополнительный.

Значения Указывается в минутах. Значение 0 отменяет время ожидания.

Значение по умолчанию
1

Admin.ssl.truststore.name = *значение*
Имя файла базы данных, используемого для проверки надежности сертификата сервера Secure Sockets, который предоставляется сервером клиенту Secure Sockets.

Обязательный параметр?
Дополнительный. Используется только с командой **sync**. По умолчанию значение этого свойства равно значению свойства **config.keystore.file**.

Admin.ssl.truststore.type = *значение*
Тип используемого хранилища ключей.

Обязательный параметр?
Дополнительный.

Значение по умолчанию
jceks

Audit.event.outcome = *значение*
Записываются только те события аудита, которые приводят к заданному результату.

Обязательный параметр?
Да.

Значения success | failure. Если указываются оба значения одновременно, их необходимо разделить запятой или точкой с запятой.

Значение по умолчанию
success

Audit.event.Queue.max = 0
Максимальное количество объектов событий в очереди памяти аудита, по достижении которого эти объекты записываются в файл.

Обязательный параметр?
Дополнительный. Рекомендуется использовать.

Значения 0-? (при значении 0 запись объектов в файл выполняется немедленно).

Значение по умолчанию
0

Audit.event.types = *значение*
Записываются только те события аудита, которые приводят к заданному результату.

Обязательный параметр?
Да.

Значения all | authentication | authorization | data synchronization | runtime | audit management | authorization terminate | configuration management | resource management | none. Если указываются несколько значений одновременно, их необходимо разделить запятой или точкой с запятой.

Значение по умолчанию
all

Audit.handler.file.directory = ../audit
Каталог, в котором будет расположен файл, заданный свойством Audit.handler.file.name.

Обязательный параметр?
Дополнительный. Рекомендуется использовать.

Audit.handler.file.multithreads = *значение*
Определяет, должен ли обработчик аудита выделять отдельные потоки на обработку протоколов аудита.

Обязательный параметр?
Дополнительный.

Значения true | false

Значение по умолчанию
true

Audit.handler.file.name = kms_audit.log
Имя файла, в который будут заноситься записи аудита.

Обязательный параметр?
Да.

Audit.handler.file.size = 100
Размер файла, заданного свойством Audit.Handler.file.name, по достижении которого начнется перезапись содержимого файла.

Обязательный параметр?
Дополнительный. Рекомендуется использовать.

Значения 0-? (указывается в килобайтах).

Значение по умолчанию
100

Audit.handler.file.threadlifespan = *значение*
Ограничивает время существования потока, обрабатывающего протоколы аудита. Используется только в том случае, если значение свойства audit.handler.file.multithreads равно true.

Обязательный параметр?
Дополнительный.

Значения Указывается в миллисекундах.

Значение по умолчанию
10000

Audit.metadata.file.cachecount = 100
Определяет количество записей, хранящихся в памяти, по достижении которого они записываются в файл метаданных.

Обязательный параметр?
Нет.

Значение по умолчанию
100

Audit.metadata.file.name = *значение*
Определяет имя XML-файла, в который записываются метаданные.

Обязательный параметр?
Да.

Audit.metadata.file.size = 1024
Определяет максимальный размер XML-файла метаданных (в килобайтах), по достижении которого закрывается существующий файл и открывается новый. Сохраняются только текущая и предыдущая версия файла.

Обязательный параметр?
Нет.

Значение по умолчанию
1024

config.drivetable.file.url = FILE:../filedrive.table
Файл, содержащий информацию о накопителе на магнитной ленте, такую как серийный номер, сертификаты и т.д.

Обязательный параметр?
Да.

config.keygroup.xml.file = *значение*
Определяет имя XML-файла, в котором хранятся отдельные псевдонимы, упорядоченные по группам ключей.

Обязательный параметр?
Дополнительный.

config.keystore.file = *значение*
Определяет используемое хранилище ключей.

Обязательный параметр?
Да.

config.keystore.password = пароль
Пароль для доступа к файлу, заданному свойством config.keystore.file. Если значение этого свойства задано, в целях безопасности оно скрывается, а сама строка в файле свойств заменяется новой строкой - config.keystore.password.obfuscated.

Обязательный параметр?
Дополнительный. Если значение этого свойства не задано, при запуске Encryption Key Manager может появиться соответствующее приглашение.

config.keystore.provider = IBMJCE

Обязательный параметр?
Дополнительный.

config.keystore.type = jceks

Обязательный параметр?
Дополнительный. Рекомендуется использовать.

Значение по умолчанию
jceks

debug = *значение*
Включает отладку для указанного компонента Encryption Key Manager.

Обязательный параметр?
Дополнительный.

Значения all | audit | server | drivetable | config | admin | transport | logic | keystore | console | none. При использовании нескольких значений они разделяются запятыми.

Значение по умолчанию
none

debug.output = *значение*
Перенаправляет результат отладки в заданное местоположение.

Обязательный параметр?
Дополнительный.

Значения simple_file | console (не рекомендуется).

debug.output.file = debug
Путь и имя файла, используемые при сохранении результата отладки.

Обязательный параметр?
Дополнительный. Необходимо использовать, если свойству debug.output присвоено значение simple_file. Путь к файлу должен существовать.

drive.acceptUnknownDrives = *значение*
Автоматическое добавление нового накопителя, который обменивается данными с Encryption Key Manager, в таблицу накопителей

Обязательный параметр?
Да.

Значения true | false

Значение по умолчанию
false

Примечание о безопасности - это значение в сочетании с допустимым значением свойства drive.default.alias1 позволяет добавлять готовые к работе накопители на магнитной ленте, которые обмениваются данными с Encryption Key Manager, без проверки администратором факта добавления. Дополнительные сведения см. в главе 4 в разделе “Автоматическое обновление таблицы ленточных накопителей”.

fips = *значение*
Федеральный стандарт обработки информации (Federal Information Processing Standard). Дополнительные сведения см. в главе 2 в разделе “Замечания о федеральном стандарте обработки информации (Federal Information Processing Standard) 140-2”.

Обязательный параметр?
Дополнительный.

Значения on | off

Значение по умолчанию
off

maximum.threads = 200
Максимальное количество потоков, которое может создать приложение Encryption Key Manager.

Обязательный параметр?
Дополнительный.

Server.authMechanism = *значение*
Определяет механизм аутентификации, который используется для локальных или удаленных клиентов. Если значение этого свойства равно `EKM`, пользователь клиента CLI должен войти на сервер, используя ID EKMAdmin и пароль changeME. (Этот пароль можно изменить с помощью команды chgpasswd.) Если значение этого свойства равно `LocalOS`, аутентификация клиента выполняется с использованием реестра локальной операционной системы. (Перед изменением файла KeyManagerConfig.properties убедитесь, что сервер Encryption Key Manager остановлен.) Пользователь клиента CLI должен входить на сервер с использованием ID и пароля для входа в операционную систему. При аутентификации с помощью локальной операционной системы на платформах Linux выполните следующие действия:

1. Загрузите архивный файл Dell Release R175158 (EKMServicesAndSamples) с **http://support.dell.com** и извлеките его файлы в произвольный каталог.

2. Извлеките содержимое файла EKMServiceAndSamples.jar (доступен на носителе с программным обеспечением Dell и для загрузки на Web-сайте **http://support.dell.com**) во временный каталог.
3. Скопируйте файл libjaasauth.so из каталога LocalOS-setup, соответствующего используемой платформе, в каталог *java_home*/jre/bin.
   - Для Linux в среде 32-разрядных процессоров Intel скопируйте файл LocalOS-setup/linux_ia32/libjaasauth.so в каталог *java_home*/jre/bin/. В случае 32-разрядного ядра Intel Linux под управлением JVM версии 1.4.2 *java_home* - это обычно *java_install_path*/IBMJava2-i386-142.
   - Для Linux в среде 64-разрядных процессоров AMD64 скопируйте файл LocalOS-setup/linux-x86_64/libjaasauth.so в каталог *java_home*/jre/bin/. В случае 64-разрядного ядра AMD Linux под управлением JVM версии 1.4.2 *java_home* - это обычно *java_install_path*/IBMJava2-amd64-142.

Этот файл не нужен для платформ Windows.

По завершении установки можно запустить сервер Encryption Key Manager. Клиент Encryption Key Manager теперь сможет войти в систему, используя имя пользователя и пароль операционной системы. Следует заметить, что вход в систему и передача команд серверу разрешены только пользователю, от имени которого запущен сервер, и, кроме того, имеющему права суперпользователя (ROOT).

Дополнительные сведения об установке см. в файле readme, который находится на носителе с продуктом Dell и доступен по адресу **http://support.dell.com**.

Обязательный параметр?
Дополнительный.

Значения EKM | LocalOS

Значение по умолчанию
EKM

Server.password = *значение*
Внутреннее свойство. Не редактируйте его значение.

symmetricKeySet = {идентификатор_группы | список_псевдонимов_ключей [, список_псевдонимов_ключей,]}
Определяет псевдонимы симметричных ключей и группы ключей, используемые с накопителями на магнитной ленте LTO 4 и LTO 5.

Обязательный параметр?
Дополнительный. Относится только к кассетам с магнитной лентой LTO 4 и LTO 5.

Значения

Для переменной *идентификатор_группы* можно указать одно значение, для переменной *список_псевдонимов_ключей* - одно или несколько значений.

Переменная *идентификатор_группы* задает имя группы ключей, расположенной первой в списке симметричных ключей и используемой по умолчанию, если для накопителя на магнитной ленте не задан псевдоним. Значение переменной *идентификатор_группы* должно совпадать с идентификатором группы ключей, существующим в файле KeyGroups.xml. Если значение не совпадает, генерируется исключительная ситуация KeyManagerException. Если

указано несколько значений переменной *идентификатор_группы*, генерируется исключительная ситуация KeyManagerException. Если задано действительное значение переменной *идентификатор_группы*, каждый раз при вызове функции getKey для получения списка симметричных ключей в файле KeyGroups.xml выполняется поиск ключа, который использовался последним, и происходит случайный выбор следующего ключа.Каждый экземпляр переменной *список_псевдонимов_ключей* содержит значение для переменной *псевдоним_ключа* или для переменной *диапазон_псевдонимов_ключей*.

Значение переменной *псевдоним_ключа* содержит имя или псевдоним (длиной до 12 символов) симметричного ключа в форме Бэкуса-Наура (BNF) или значение sequentialKeyID длиной 21 символ.

Значение переменной *диапазон_псевдонимов_ключей* (длиной до 18 символов) содержит значение sequentialKeyID и шестнадцатиричные числа, разделенные дефисом (-). Если указывается 18 символов, первые два символа должны иметь вид 00. Это значение должно быть указано в одной строке и не содержать символа возврата каретки с переводом строки.

Переменная *идентификатор_группы* задает имя группы псевдонимов.

Пример `symmetricKeySet = KMA0238ab34,KMB0000034acd2345678a,THZ001-FF` Указывает, что при предоставлении ключей накопителям на магнитной ленте LTO 4 и LTO 5 приложение Encryption Key Manager должно использовать псевдонимы KMA0238ab34, KMB0000034acd2345678a и диапазон псевдонимов от THZ000000000000000001 до THZ0000000000000000FF. Эти ключи должны существовать в хранилище ключей, заданном свойством **config.keystore.file** в файле свойств.

sync.action = *значение*

Определяет, каким образом обрабатываются данные во время автоматической синхронизации.

Обязательный параметр?
Дополнительный.

Значения rewrite | merge

Значение по умолчанию
merge

Примечание: Объединение информации о конфигурации равнозначно ее перезаписи.

sync.ipaddress = *ip_адрес:ssl*

Определяет IP-адрес и порт удаленного диспетчера Encryption Key Manager, с которым выполняется автоматическая синхронизация.

Обязательный параметр?
Дополнительный. Если значение этого свойства не указано или указано неправильно, функция синхронизации выключена.

Значения IP-адрес удаленного сервера:номер порта SSL

sync.timeinhours = *значение*

Определяет время ожидания (в часах) перед выполнением автоматической синхронизации с удаленным Encryption Key Manager.

Обязательный параметр?

Дополнительный.

Значения Указывается в часах.

Значение по умолчанию

24

sync.type = *значение*

Определяет тип данных, которые необходимо синхронизировать в автоматическом режиме.

Обязательный параметр?

Дополнительный.

Значения config | drivetab | all

Значение по умолчанию

drivetab

TransportListener.ssl.ciphersuites = JSSE_ALL

Кодовые наборы, используемые при передаче данных между серверами Encryption Key Manager. В кодовом наборе содержится описание криптографических алгоритмов и протоколов связи, которые используются протоколами Transport Layer Security (TLS) и Secure Sockets Layer (SSL) для передачи данных.

Обязательный параметр?

Дополнительный.

Значения Любые кодовые наборы, поддерживаемые IBMJSSE2.

TransportListener.ssl.clientauthentication = 0

Аутентификация SSL, необходимая для связи между серверами Encryption Key Manager.

Обязательный параметр?

Дополнительный.

Значения
0 - аутентификация клиента отсутствует (по умолчанию)
1 - может потребоваться аутентификация клиента на сервере
2 - необходима аутентификация клиента на сервере

TransportListener.ssl.keystore.name = *значение*

Имя базы данных, в которой сервер Encryption Key Manager хранит сертификат и секретные ключи для сервера Secure Sockets. Этот сертификат предоставляется клиенту Secure Sockets для аутентификации и проверки надежности. Это хранилище ключей также используется клиентом Encryption Key Manager для обмена данными с сервером Encryption Key Manager и взаимодействия с ним в качестве клиента Secure Sockets.

Обязательный параметр?

Да.

TransportListener.ssl.keystore.password = пароль

Пароль для доступа к хранилищу ключей, заданному свойством TransportListener.ssl.keystore.name. Если значение этого свойства задано, в

целях безопасности оно скрывается, а сама строка в файле свойств заменяется новой строкой - TransportListener.ssl.keystore.password.obfuscated.

Обязательный параметр?
Дополнительный.

TransportListener.ssl.keystore.type = jceks

Обязательный параметр?
Дополнительный. Рекомендуется использовать.

Значения JCEKS

TransportListener.ssl.port = *значение*
Порт, который прослушивается сервером Encryption Key Manager на предмет получения запросов от других серверов Encryption Key Manager или клиентов Encryption Key Manager CLI.

Обязательный параметр?
Да.

Значения Например, порт номер 443. Это значение должно соответствовать значению свойства TransportListener.ssl.port в файле свойств конфигурации клиента CLI.

TransportListener.ssl.protocols = SSL_TLS
Протоколы безопасности.

Обязательный параметр?
Дополнительный.

Значения SSL_TLS (по умолчанию) | SSL | TLS

TransportListener.ssl.timeout = 10
Определяет время, в течение которого сокет ожидает выполнения команды read(), прежде чем генерировать исключительную ситуацию SocketTimeoutException.

Обязательный параметр?
Дополнительный.

Значение Указывается в минутах.

Значение по умолчанию
1

TransportListener.ssl.truststore.name = *значение*
Имя базы данных, содержащей открытые ключи и подписанные сертификаты, которые используются для проверки идентификационной информации других клиентов и серверов. Если для свойства TransportListener.ssl.clientauthentication **не** задано значение по умолчанию, равное 0 (аутентификация клиента не выполняется), сервер Encryption Key Manager, выступая в роли сервера Secure Socket, должен выполнить аутентификацию клиента с помощью этого файла. Это доверенное хранилище также используется клиентом Encryption Key Manager для обмена данными с сервером Encryption Key Manager в роли клиента Secure Sockets.

Обязательный параметр?
Да.

TransportListener.ssl.truststore.type = jceks

Обязательный параметр?
Дополнительный. Рекомендуется использовать.

Значения JCEKS

TransportListener.tcp.port = *значение*

Порт, который прослушивается сервером Encryption Key Manager на предмет получения запросов от накопителей на магнитной ленте. Значение по умолчанию для номера порта TCP - 3801.

Обязательный параметр?

Да.

Значения Например, порт номер 10.

TransportListener.tcp.timeout = *значение*

Определяет время, в течение которого сокет ожидает выполнения команды read(), прежде чем генерировать исключительную ситуацию SocketTimeoutException.

Обязательный параметр?

Дополнительный.

Значения Указывается в минутах. Значение 0 отменяет время ожидания.

Значение по умолчанию

10

Файл свойств конфигурации клиента CLI

Файл ClientKeyManagerConfig.properties содержит неполный набор свойств из файла KeyManagerConfig.properties. В этот неполный набор входят следующие свойства.

TransportListener.ssl.ciphersuites = JSSE_ALL

Кодовые наборы, используемые при передаче данных между серверами Encryption Key Manager и клиентами CLI. В кодовом наборе содержится описание криптографических алгоритмов и протоколов связи, которые используются протоколами Transport Layer Security (TLS) и Secure Sockets Layer (SSL) для передачи данных.

Обязательный параметр?

Дополнительный.

Значения Это значение должно соответствовать значению свойства TransportListener.ssl.ciphersuites в файле свойств сервера Encryption Key Manager (KeyManagerConfig.properties).

TransportListener.ssl.host = *значение*

Определяет сервер Encryption Key Manager для клиента Encryption Key Manager CLI.

Обязательный параметр?

Дополнительный.

Значения IP-адрес или имя узла.

Значение по умолчанию

localhost

Примеры

```
TransportListener.ssl.host = 9.24.136.444
TransportListener.ssl.host = ekmsvr02
```

Примечание: Не используется в файле KeyManagerConfig.properties.

TransportListener.ssl.keystore.name = *значение*

Это хранилище ключей также используется клиентом Encryption Key Manager для обмена данными с сервером Encryption Key Manager и взаимодействия с ним в качестве клиента Secure Sockets.

Обязательный параметр?

Да.

TransportListener.ssl.keystore.type = jceks

Тип хранилища ключей.

Обязательный параметр?

Дополнительный. Рекомендуется использовать.

Значение по умолчанию

jceks

TransportListener.ssl.port = *значение*

Порт, используемый клиентом CLI для обмена данными с серверами Encryption Key Manager.

Обязательный параметр?

Да.

Значения Это значение должно соответствовать значению свойства TransportListener.ssl.port в файле свойств сервера Encryption Key Manager (KeyManagerConfig.properties).

TransportListener.ssl.protocols = SSL_TLS

Протоколы безопасности.

Обязательный параметр?

Дополнительный.

Значения Это значение должно соответствовать значению свойства TransportListener.ssl.protocols в файле свойств сервера Encryption Key Manager (KeyManagerConfig.properties).

TransportListener.ssl.truststore.name = *значение*

Имя базы данных, содержащей открытые ключи и подписанные сертификаты, которые используются для проверки идентификационной информации других клиентов и серверов.

Обязательный параметр?

Да.

TransportListener.ssl.truststore.type = jceks

Тип доверенного хранилища.

Обязательный параметр?

Дополнительный. Рекомендуется использовать.

Значение по умолчанию

jceks

Примеры файлов свойств конфигурации доступны для загрузки в файле EKMServicesAndSamples с **http://support.dell.com**.

Приложение C. Часто задаваемые вопросы

Возможно ли совместное использование программных средств управления ключами и шифрования, управляемого библиотекой?

Нет. При использовании шифрования, управляемого приложением, шифрование прозрачно на уровнях библиотеки. Аналогичным образом, при использовании шифрования, управляемого библиотекой, этот процесс прозрачен на всех остальных уровнях. Один способ шифрования исключает применение других способов. Для выполнения шифрования, управляемого библиотекой, не требуется какое-либо изменение приложений.

Требуется ли установка и выполнение программного обеспечения Encryption Key Manager на всех системах, которые могут генерировать запросы на шифрование или расшифровку магнитной ленты?

При использовании шифрования, управляемого библиотекой, НЕ требуется выполнение программного обеспечения Encryption Key Manager в системе, создающей запрос на запись данных на ленточный накопитель. Более того, НЕ требуется выполнение экземпляра Encryption Key Manager в каждой системе, из которой осуществляется доступ к ленточному накопителю с поддержкой шифрования данных.

Если в файле конфигурации указан параметр "drive.acceptUnknownDrives = True", нужно ли указывать параметр "config.drivetable.file.url = FILE:/filename"?

Параметр **config.drivetable.file.url** необходимо указывать всегда. Он определяет место хранения сведений о накопителях. Если задается значение параметра `drive.acceptUnknownDrives = True`, следует также присвоить переменным `drive.default.alias1` и `drive.default.alias2` правильные значения псевдонима сертификата/метки ключа.

Является ли FILE:/filename правильным синтаксисом для свойства `config.drivetable.file.url`? Значение FILE:///filename приведено в файле примера, а FILE:../ - в описании.

Эти примеры корректны. Адрес вводится в соответствии со спецификацией URL, а не со спецификацией структуры каталогов, как можно было бы предположить.

Какую косую черту (прямую или обратную) следует использовать при указании полных путей в файле KeyManagerConfig.properties для экземпляра Encryption Key Manager, работающего в среде Windows?

Поскольку файл KeyManagerConfig.properties является файлом свойств Java, при указании путей допустима только прямая косая черта, даже при работе в среде Windows. Использование обратной косой черты в файле KeyManagerConfig.properties приведет к возникновению ошибок.

Выполняет ли Encryption Key Manager проверку списка отозванных сертификатов (CRL)?

Нет, Encryption Key Manager не выполняет какую-либо проверку списка CRL.

Что происходит при истечении срока действия сертификата, используемого для шифрования данных на магнитных лентах? Сохранит ли Encryption Key Manager возможность чтения ранее зашифрованных лент?

Истечение срока действия сертификата не имеет значения для Encryption Key Manager. EKM по-прежнему будет признавать эти сертификаты и осуществлять чтение ранее зашифрованных лент. Однако сертификаты с истекшим сроком действия должны оставаться в хранилище ключей, чтобы обеспечить возможность чтения или дополнения ранее зашифрованных лент.

Требуется ли для Encryption Key Manager переименование сертификатов при их обновлении?

Настройки Encryption Key Manager по умолчанию подразумевают признание новых запросов ключа, у которых просрочены сертификаты. Экземпляр Encryption Key Manager, настроенный таким образом, не требует обновления сертификата. Если эта функция отключена, а данную пару "секретный ключ/сертификат" все еще нужно использовать в новых запросах ключа, то пользователю следует обновить сертификат. Обновляется только сам сертификат (сроки его действия), но не связанные с ним ключи.

Сохранят ли следующие версии Encryption Key Manager возможность чтения зашифрованных лент, созданных с использованием предыдущих версий этого программного обеспечения?

Да. Encryption Key Manager признает сертификаты независимо от версии.

Замечания

Товарные знаки

Товарные знаки, упоминающиеся в тексте - Dell, эмблема Dell и PowerVault, - являются товарными знаками Dell Inc. Microsoft и Windows - зарегистрированные товарные знаки Microsoft Corporation. Прочие товарные знаки и торговые наименования могут использоваться в этом документе для обозначения компаний и названий продуктов. Компания Dell Inc. отказывается от каких-либо прав на товарные знаки и торговые наименования помимо собственных.

Глоссарий

В данном глоссарии приведены определения терминов, сокращений и аббревиатур, используемых в данном руководстве и других связанных с ним публикациях.

AES. Стандарт AES (Advanced Encryption Standard). Блочный шифр, принятый в качестве стандарта шифрования правительством США.

DK. Ключ шифрования данных. Буквенно-цифровая строка, используемая для шифрования данных.

EEDK. Зашифрованный ключ шифрования данных. Ключ шифрования данных, зашифрованный с помощью ключа шифрования других ключей перед сохранением на магнитной ленте. См. "KEK".

KEK. Ключ шифрования других ключей. Буквенно-цифровой асимметричный ключ, используемый для шифрования ключей шифрования данных. См. "EEDK".

PKDS. Набор данных открытого ключа. Также называется набором данных криптографического ключа PKA.

RSA. Алгоритм Райвеста-Шамира-Адлемана. Асимметричная криптографическая система с открытым ключом, которая может использоваться для шифрования данных и проверки подлинности. Она была изобретена в 1977 году Роном Райвестом (Ron Rivest), Ади Шамиром (Adi Shamir) и Леонардом Адлеманом (Leonard Adleman). Надежность системы обеспечивается сложностью разложения на множители произведения двух больших простых чисел.

Метка ключа (key label). Уникальный идентификатор, используемый для сопоставления ключа EEDK и секретного ключа (KEK), необходимого для расшифровки защищенного симметричного ключа шифрования данных. В зависимости от используемого хранилища ключей может также называться псевдонимом или меткой сертификата.

Метка сертификата (certificate label). См. "метка ключа".

Открытый ключ (public key). Один из пары асимметричных ключей, обычно используемый для шифрования. Encryption Key Manager использует открытые ключи для свертывания (защиты) AES-ключей шифрования данных перед их сохранением на магнитной ленте.

Псевдоним (alias). См. "метка ключа".

Связка ключей (key ring). См. "хранилище ключей".

Секретный ключ (private key). Один из пары асимметричных ключей, обычно используемый для расшифровки. Encryption Key Manager использует секретные ключи для развертывания защищенных AES-ключей шифрования данных перед расшифровкой данных.

Сертификат (certificate). Цифровой документ, устанавливающий связь между открытым ключом и идентификационными данными владельца сертификата и обеспечивающий таким образом возможность проверки подлинности владельца сертификата.

Смена ключа (rekey). Процесс смены асимметричного ключа шифрования других ключей (KEK), который защищает ключ шифрования данных (DK), сохраненный на уже зашифрованной магнитной ленте, и таким образом позволяет различным объектам получать доступ к данным.

Хранилище ключей (keystore). База данных секретных ключей и связанных с ними цепочек цифровых сертификатов стандарта X.509, используемых для проверки подлинности соответствующих открытых ключей. В некоторых средах может также называться хранилищем сертификатов или связкой ключей.

Хранилище сертификатов (certificate store). См. "хранилище ключей".

Шифрование (encryption). Преобразование данных в шифр. Для шифрования и расшифровки данных требуется ключ. Шифрование обеспечивает защиту от лиц или программ, пытающихся получить доступ к данным без ключа.

Индекс

Р

С

Т

У

Ш