**Описание протокола в целом**

Модель “KRB” основывается частично на Needham-Schroeder-прото­коле доверительной 3-сторонней аутентификации и его модификациях, предложенных Denning и Sacco.

KRB определяет основные средства проверки подлинности участни­ков сеанса связи (рабочей станции пользователя или сетевого сервера) в открытых (не защищенных) сетях. Это достигается без доверия к процеду­рам аутентификации, проводимым операционной системой (ОС) IP-узла, без доверия адресам IP-узлов, без необходимой физической безопасности всех IP-узлов сети, и в предположении, что пакеты, “путешествующие” по сети, могут быть кем-то прочитаны, модифицированы (преобразованы) и могут вставляться по чьему-то желанию. KRB позволяет проводить идентификацию “объектов” в таких условиях, высту­пая в роли доверительной службы аутентификации, используя процедуры трехстороннего сеанса связи и обусловленные заранее криптографические методы защиты информации, то есть коллективный секретный ключ.

Процедура аутентификации происходит следующим образом: клиент передает запрос на сервер аутентификации (СА), требуя от него “удостове­рить его личность” (верительные документы — далее ВД) для нужного сервера. СА от­вечает на это требование сообщением, в котором содержится зашифрован­ный ключ клиента. Требование клиента “удостоверить его личность” со­держит (состав ВД):

* “билет” для сервера;
* временный криптоключ (называемый “сеансовым” ключом).

Клиент передает билет (который содержит клиентский идентифика­тор и сеансовый ключ, и это все зашифровано с использованием ключа сервера) на сервер. Сеансовый ключ (действующий в настоящее время ме­жду клиентом и сервером) используется для аутентификации клиента и может дополнительно использоваться для аутентификации сервера. Он может также использоваться в процедуре шифрования в последующем ин­формационном обмене между двумя участниками сеанса, либо использо­ваться в специальной сессии распределения ключей при изменении ключа с целью дальнейшего применения последнего в процедуре шифрования в дальнейшем информационном обмене.

В практической реализации имеют место один или несколько СА, работающих с физически защищенными IP-узлами. СА поддерживают БД об участниках процесса аутентификации (то есть о пользователях и серве­рах) и их секретных ключах. Библиотеки кодов обеспечивают процедуры шифрования и функционирования KRB-протокола. С целью учета проце­дур аутентификации при информационном обмене, типичная сетевая реа­лизация дополняет KRB-библиотеку один или двумя кодовыми вызовами, которые используются для передачи необходимых сообщений при успеш­ном завершении аутентификации.

KRB-протокол включает в себя несколько субпротоколов (или про­цедур обмена). Существуют два метода, с помощью которых клиент может запросить KRB-сервер для получения разрешения (верительных докумен­тов) на доступ в сеть. Во-первых, клиент передает на СА не зашифрован­ный запрос на билет для запрашиваемого сервера. В ответ передается сообщение, зашифрованное с использованием секретного ключа клиента. Обычно этот запрос на получение “билета, запрашивающего билет-разре­шение” (ББР), который позже может использоваться “сервером выдачи би­летов-разрешений” (СБР). Во втором методе, клиент передает запрос на СБР. Клиент передает ББР на СБР так, как если бы он был соединен с лю­бым другим прикладным сервером, который бы затребовал верительные документы в соответствии с KRB-протоколом. Ответ зашифровывается с использованием сеансового ключа, изъятого из ББР.

Еще одно применение, ВД могут быть ис­пользованы для:

* определения подлинности участников информационного об­мена;
* обеспечения целостности сообщений, которыми обмениваются участники сессии;
* защиты конфиденциальности сообщений.

Конкретный прикладной процесс или служба свободны в выборе формы обеспечения безопасности в соответствии со своими потребно­стями.

Для определения подлинности участников информационного об­мена, клиент передает билет на сервер. В связи с тем, что билет передается в “чистом виде” (не шифруется) и может быть перехвачен и повторно ис­пользован нарушителем (части билета зашифровываются, но эта процедура шифрования не влияет на ответ), то передается дополнительная информа­ция с целью того, чтобы подтвердить принадлежность данного сообщения участнику сессии, который выдал билет. Эта информация (вызываемый ау­тентификатор) зашифровывается с помощью сеансового ключа, а также включает отметку времени. Эта отметка времени подтверждает, что сооб­щение сформировано недавно и не является повторным. Шифрование ау­тентификатора с использованием сеансового ключа подтверждает, что по­следний был получен с помощью специальной процедуры генерации сеан­сового ключа. Таким образом, подлинность клиента гарантируется тем, что не было ни одного отказа на запрос участника сессии и что сервер знает сеансовый ключ (так как он никогда не передается в открытом виде по сети).

Целостность сообщений, с помощью которых обмениваются участ­ники сессии, также может быть гарантирована, благодаря использованию сеансового ключа (передавая его в билете и включая в состав верительных документов). Обеспечение целостности сообщений позволяет также защититься, как и от ложных ответов нарушителя, так и от преобразований в по­следовательности сообщений. Это достигается за счет вычисления и дальнейшей передачи проверочной последовательности (часто именуемой хэш-функцией или функция вычисления остатка от деления) в сообщении клиента, которое содержит сеансовый ключ. Конфиденциальность и целостность сообщений, которыми обмениваются участники сессии, может быть обеспечена с помощью процедуры шифрования данных, предназначенных для передачи, с помощью сеансового ключа, передаваемого в билете и содержащегося в верительных документах.

Упомянутая выше процедура аутентификации (на основе обмена сообщениями) требует того, чтобы база данных (БД) KRB-протокола (KRB-БД) была доступна только в режиме чтения данных. Иногда, однако, необходимо корректировать и наращивать эту БД, в связи с тем, что появля­ются новые пользователи или меняются ключи участников информацион­ного обмена. Эти процедуры могут проводиться на основании специаль­ного протокола между клиентом и третьим KRB-сервером, называемым административным KRB-сервером (АДМКС). Однако административный протокол в этом документе на рассматривается. Также существует специальный протокол эксплуатации (функционирования) нескольких копий KRB-БД.

**Описание запросов, выполняемых в процессе алгоритма**

**Формирование сообщения KRB\_AS\_REQ**

Клиент в начальном запросе определяет совокупность дополнительных функций, среди которых могут быть:

* проведение процедур предварительной аутентификации;
* запрос билета (БВОСС, уполномочивающего или для доставки данных аутентификации);
* запрос на ратификацию БПОЗД или на присвоение БПОЗД статуса потерянного билета;
* запрос на ратификацию другого билета, если представленный БПОЗД не удовлетворяет условиям ратификации (в следствие конструктивных ограничений).
* Клиент формирует сообщение KRB\_AS\_REQ и направляет его в ЦРК.

**Прием сообщения KRB\_AS\_REQ**

Если все прошло корректно, то в результате обработки сообщения KRB\_AS\_REQ будет сформирован билет для клиента, который он представит серверу.

**Формирование сообщения KRB\_AS\_REP**

СА анализирует имена участников сессии (под которыми выступают клиент и сервер), представленных в запросе и в своей БД, извлекая из последней их соответствующие ключи. Если необходимо, то СА посылает запрос на предварительную аутентификацию, и если эта процедура выявила ошибку, то СА возвращает сообщение об ошибке, в котором установлен код ошибки “KDC\_ERR\_PREAUTH\_FAILED”. Если сервер не может обеспечить запрашиваемый тип шифрования, то он возвращает сообщение об ошибке (“KDC\_ERR\_ETYPE\_NOSUPP”). В противном случае он вырабатывает “случайный” сеансовый ключ. (Под термином “случайный” понимается то, что этот ключ имеет ничтожно малую вероятность своего повторения, а как последующий сеансовый ключ он также не будет повторен, так как при его генерации должен быть известен предшествующий ключ. Это можно обеспечить только с помощью генератора псевдослучайной последовательности, основанного на криптографических принципах. Но более предпочтителен генератор действительно случайных чисел, основанный на случайном физическом явлении.)

Если запрашиваемое начальное время отсутствует или устарело, то тогда время начала действия билета устанавливается по текущему времени СА. Если же это время показывает, что оно еще только наступит в будущем и при этом не указана дополнительная функция “POSTDATED”, то сервер возвращает сообщение об ошибке (“KDC\_ERR\_CANNOT\_POSTDATE”). В противном случае, запрашиваемое начальное время проверяется по времени локальной зоны (администратор может принимать решение о запрете определенных типов и групп БПОЗД) и если оно допустимо, то устанавливается (запрашиваемое начальное время) в БПОЗД, а также в нем устанавливается флаг “INVALID”. Этот билет, в свою очередь, должен быть ратифицирован (еще до начала своего применения) путем отправки его в ЦРК, но только после того, как наступило время начала действия билета.

Время окончания ЖЦБ (то есть время, до которого билет считается действительным) будет определяться на основании как минимум следующим данных:

* в сообщении KRB\_AS\_REQ запрашивается окончание времени ЖЦБ (“endtime”);
* начальное время билета складывается с максимальным временем ЖЦБ, разрешенным для конкретного клиента, участвующего в сессии (БД СА включает запись для каждого участника сессии, в которой есть поле “максимальное время ЖЦБ” — “maximum ticket lifetime”);
* начальное время билета складывается с максимальным временем ЖЦБ, разрешенным для конкретного сервера, участвующего в сессии;
* начальное время билета складывается с максимальным временем ЖЦБ, устанавливаемым в соответствии со стратегией безопасности в конкретной локальной зоне.

Если запрашиваемое время окончания действия билета за вычетом начального времени (как определено выше) — меньше чем минимально определенное время ЖЦБ, тогда сервер будет возвращать сообщение об ошибке (“KDC\_ERR\_NEVER\_VALID”). Если запрашиваемое время окончания действия билета превышает время, которое было определено выше, и если при этом запрашивается дополнительная функция “RENEWABLE-OK”, то тогда в новом билете устанавливается флаг “RENEWABLE”, а также — время окончания действия БВОСС (“renew-till”), аналогичное тому, которое было бы в запросе на дополнительную функцию “RENEWABLE”. Если была затребована дополнительная функция “RENEWABLE” (или была установлена дополнительная функция “RENEWABLE-OK”) и при этом был выдан БВОСС, то значение в поле “renew-till” будет определяться на основании как минимум следующим данных:

* это значение было в запросе;
* начальное время билета в совокупности с минимальным из двух максимальных восстановленных временных интервалов ЖЦБ, связанных с записью в БД для конкретного участника сессии;
* начальное время билета в совокупности с максимальным восстановленным временным интервалом ЖЦБ, определяемым в соответствии со стратегией безопасности в конкретной локальной зоне.

Поле “флаги” в новом билете будет содержать следующие дополнительный функции (либо вследствие запроса, либо вследствие стратегии безопасности в конкретной локальной зоне): “FORWARDABLE”, “MAY-POSTDATE”, “POSTDATED”, “PROXIABLE”, “RENEWABLE”. Если новый билет является БПОЗД (и его начальное время будет установлено позже или еще не наступило), то в нем также будет установлен флаг “INVALID”.

Если все, о чем говорилось выше прошло успешно, то сервер формирует сообщение KRB\_AS\_REP, копируя (в поле “caddr”) адреса из запроса и размещая (в поле “padata”) любые запрашиваемые данные предварительной аутентификации, зашифровывает часть текста (подлежащую шифрованию) с помощью ключа клиента, используя запрашиваемый метод шифрования, и после этого передает его клиенту.

**Формирование сообщения KRB\_ERROR**

При ведении информационного обмена существует вероятность появления ошибок. Если одна их них имеет место, то СА реагирует на нее путем передачи клиенту сообщения об ошибке (KRB\_ERROR), в котором поля “error-code” и “e-text” имеют конкретные значения.

**Прием сообщения KRB\_AS\_REP**

Если в ответ на свой запрос клиент получил сообщение KRB\_AS\_REP, то после этого он определяет, что параметры в полях “cname” и “crealm”, содержащихся в открытой части текста, соответствуют тем, которые были в запросе. Если имеют место поля “padata”, то они могут быть использованы для получения истинного секретного ключа для расшифрования сообщения. Клиент расшифровывает зашифрованную часть ответа, используя для этого свой секретный ключ, и определяет, что текущее время в зашифрованной части текста соответствует текущему времени, посланному им в запросе (с цель определения ответов на свои запросы). Далее он определяет, что поля “sname” и “srealm” в ответе соответствуют тем, что были в запросе, и что поле “host address” — также корректно. После этого клиент сохраняет билет, сеансовый ключ, начальное время и время ЖЦБ, а также другую информацию для последующего использования. Поле “key-expiration” в зашифрованной части ответа также проверяется клиентом на предмет критичности времени окончания жизненного цикла ключа (в этом случае программа клиента может в дальнейшем “сделать напоминание” о необходимости соответствующей процедуры, например, обмена паролями).

Корректное расшифрование сообщения KRB\_AS\_REP не является достаточным для установления подлинности пользователя; пользователь и нарушитель могут “скооперироваться” для формирования сообщения KRB\_AS\_REP, которое будет правильно расшифровано, но на самом деле оно поступило на из истинного ЦРК. Если IP-узел желает определить подлинность пользователя, он должен запросить у него представить ВД приложения, которые могут быть проверены с использованием секретного ключа, хранящегося в безопасных условиях. Если эти ВД могут быть проверены, то такая проверка гарантирует подлинность пользователя.

**Прием сообщения KRB\_ERROR**

Если в ответ поступило сообщение KRB\_ERROR, то тогда клиент воспринимает его как ошибку и проводит специальные прикладные проверки и тесты, необходимые для восстановления штатного режима.

**Организация обмена сообщениями между клиентом и сервером при аутентификации**

Основные типы сообщений и направления передачи:

|  |  |
| --- | --- |
| Направление передачи сообщения | Тип сообщения |
| ➀ Клиент 🠲 Прикладной сервер | KRB\_AP\_REQ |
| ➁ (доп. функция) Прикладной сервер 🠲 Клиент | KRB\_AP\_REP или KRB\_ERROR |

Обмен сообщениями при аутентификации между клиентом и сервером используется сетевыми прикладными процессами для аутентификации клиента сервером и наоборот. Клиент должен быть в готов получить ВД для сервера, используя для этого обмен сообщениями с СЛА и СБР.

**Сообщение KRB\_AP\_REQ**

Это сообщение включает аутентификационную информацию, которая будет частью первого сообщения при аутентификационном обмене. Она включает билет, аутентификатор и некоторую дополнительную учетную информацию. Билет как таковой не является достаточным для аутентификации клиента (пользователя), так как, во-первых, билеты доставляются через сеть в открытом виде (билеты состоят из открытой и зашифрованной частей, причем открытая часть передается как единое целое, которое может копироваться из одного сообщения и передаваться в ответном без какой либо криптографической обработки). Во-вторых, аутентификатор используется для предотвращения недействительных ответных билетов путем доказательства серверу, что клиент знает сеансовый ключ из билета, и таким образом получает право на его использование. Сообщение KRB\_AP\_REQ направляется куда бы то ни было в качестве “аутентификационного заголовка”.

**Формирование сообщения KRB\_AP\_REQ**

Когда клиент желает начать процедуру аутентификации на сервере, то вначале он приобретает билет и сеансовый ключ для запрашиваемой службы (он хранит ВД, полученные в результате обмена сообщениями либо с СЛА, либо с СБР). Клиент может повторно использовать любые билеты до истечения их срока ЖЦБ. После этого он формирует новый аутентификатор, используя системное время, свое имя и (не обязательно) проверочную сумму, начальный последовательный номер (если используются сообщения KRB\_SAFE или KRB\_PRIV) и/или сеансовый субключ, который используется при формировании (по определенным правилам) сеансового ключа, являющегося уникальным для текущей сессии. Аутентификаторы не могут использоваться повторно, если же они переданы повторно, то они могут уничтожаться сервером при приеме. (Заметим, что это могут делать прикладные службы, использующие сети передачи данных на основе ненадежных каналов связи, в которых не предусмотрено исправление ошибок, а только применяется повторная передача сообщений. В таких случаях для каждой повторной передачи обязательно должен формироваться новый аутентификатор.) Если был включен порядковый номер, то он должен быть случайным и не повторять другие номера в переданных ранее сообщениях.

Клиент может потребовать обоюдной аутентификации или использования сеансового ключа, указанного в билете. Для этого он передает запрос KRB\_AP\_REQ, в поле “ap-options” которого устанавливается флаг “appropriate”.

Аутентификатор шифруется с помощью сеансового ключа и объединяется с билетом для формирования сообщения KRB\_AP\_REQ, которое затем передается оконечному серверу вместе с другой дополнительной специальной информацией для прикладной службы.

**Прием сообщения KRB\_AP\_REQ**

Аутентификация основана на текущем суточном времени сервера (часы должны свободно синхронизироваться), аутентификаторе и билете. При приеме сообщения вероятны ошибки. При их обнаружении сервер предполагает послать клиенту ответ, в котором будет содержаться сообщение об ошибке KRB\_ERROR. Это сообщение может пройти процедуру обрамления в соответствии с логической характеристикой протокола прикладной службы, если данное сообщение не может передаваться этой службой непосредственно.

Алгоритм проверки аутентификационной информации следующий. Если получено сообщение не являющееся KRB\_AP\_REQ, то сервер отвечает сообщение об ошибке (“KRB\_AP\_ERR\_MSG\_TYPE”). Если версия ключа, указанная в билете сообщения KRB\_AP\_REQ, которую может использовать сервер, не единственная (это указывает на наличие старого ключа, а сервер не может использовать копию старого ключа), то он возвращает сообщение об ошибке (“KRB\_AP\_ERR\_BADKEYVER”). Если в поле “ap-options” установлен флаг “USE-SESSION-KEY”, это указывает серверу на то, что билет зашифрован с помощью сеансового ключа, изъятого из ББР сервера, который предпочтительнее чем его секретный ключ. Так как сервер имеет возможность зарегистрироваться в нескольких зонах с различными ключами, поле “srealm” из открытой части билета в сообщении KRB\_AP\_REQ используется для указания секретного ключа, который будет применен сервером для расшифрования этого билета. Если сервер не обладает необходимым ключом для расшифрования билета, то он возвращает сообщение об ошибке (“KRB\_ AP\_ERR\_NOKEY”).

Билет расшифровывается с использованием версии ключа сервера, указанной в билете. Если при расшифровании выявлена подделка билета (каждая криптосистема должна гарантировать выявление модифицированного шифртекста), то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_BAD\_ INTEGRITY”). (В данном случае высока вероятность того, что зашифрования и расшифрования использовались разные ключи.)

Аутентификатор расшифровывается с помощью сеансового ключа, извлеченного из расшифрованного билета. Если расшифрование показало наличие подделки, то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_ BAD\_INTEGRITY”). Имя и зона нахождения клиента, указанные в билете, сравниваются с соответствующими полями в аутентификаторе. Если они не совпадают, то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_ BADMATCH”). (В данном случае они могут не совпадать еще и потому, что для расшифрования аутентификатора использовался неверный сеансовый ключ.) После этого, адреса содержащиеся в билете, сравниваются с адресами клиента, хранящимися в ОС сервера. Если они не совпадут или затребованные сервером адреса не были представлены в билете, то сервер возвращает сообщение об ошибке (“KRB\_AP\_ERR\_BADADDR”).

Если локальное время сервера и клиентское время, указанное в аутентификаторе, различаются более чем на допустимую величину (то есть на пять минут), то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_SKEW“). Если последовательность полей: “имя сервера”, “имя клиента”, “время” и “микросекунды”, — содержащаяся в аутентификаторе совпадают с любой другой ранее наблюдавшейся последовательностью, то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_REPEAT“). (Необходимо заметить, что приведенный здесь отказ является ограничением для аутентификаторов от одного и того же участника сессии к одному и тому же серверу. Другой клиент, соединяясь с тем же сервером, не может иметь аутентификатор, которому будет дан отказ, если даже в нем поля “время” и “микросекунды” случайно совпадут с такими же полями аутентификатора другого клиента.) Сервер обязан запоминать любой аутентификатор в пределах допустимого временнóго интервала для того, чтобы повторная атака нарушителя была гарантировано отражена. Если сервер “потерял” запись о любом аутентификаторе, представленного в пределах допустимого временнóго интервала, то он обязан удалять все запросы пока этот интервал не пройдет. Это гарантирует то, что любые потерянные или повторяемые аутентификаторы будут теряться за пределами допустимого периода времени и не смогут в дальнейшем успешно использоваться повторно. (Если так не поступать, то нарушитель с высокой вероятностью сможет записать билет и аутентификатор, передаваемые по сети серверу, затем, замаскировавшись под клиентский узел, будет выступать от имени клиента и повторять (повторно передавать) их для разрушения процедуры аутентификации.) Если в аутентификаторе представлен порядковый номер, то сервер сохраняет его для последующего использования при обработке сообщений KRB\_SAFE и/или KRB\_PRIV. Если представлен субключ, то сервер либо его сохраняет для дальнейшего применения, либо использует его для генерации альтернативного субключа, чтобы в дальнейшем возвратить его в сообщении KRB\_AP\_REP.

Сервер определяет “возраст” билета: из локального времени (сервера) вычитается начальное время билета, содержащееся внутри него. Если начальное время больше чем текущее время и эта разность превышает допустимый временной интервал или в билете установлен флаг “INVALID”, то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_TKT\_NYV”). В противном случае, если текущее время больше чем окончание времени ЖЦБ и эта разность больше чем допустимый временной интервал, то возвращается сообщение об ошибке (“KRB\_AP\_ERR\_TKT\_EXPIRED”).

Если все проверки прошли успешно (ошибки не обнаружены) и сервер убеждается в том, что клиент обладает ВД участника сессии, указанными в билете, то процедура аутентификации прошла успешно, и клиент получил доступ на сервер.

**Формирование сообщения KRB\_AP\_REP**

Обычно запрос клиента будет включать как аутентификационную информацию, так и его начальные требования (в этом же сообщении), а серверу нет необходимости возвращать подробный ответ на KRB\_AP\_REQ. Однако, если выполняются процедуры обоюдной аутентификации (не только аутентификация клиента на сервере, но и сервера на узле клиента), то в поле “ap-options” сообщения KRB\_AP\_REQ будет установлена функция “MUTUAL-REQUIRED”, а сервер в ответ будет передавать запрашиваемое сообщение KRB\_AP\_REP. Как и в случае сообщения об ошибке этот ответ может подвергаться процедуре обрамления в соответствии с логической характеристикой протокола прикладной службы, если его “открытая форма” недоступна этому протоколу. Поля “отметка времени” и “микросекунды”, имеющиеся в ответе, должны быть такими же как и аналогичные поля в запросе клиента (размещаемые в аутентификаторе). Если в сообщении имеется порядковый номер, то он также выбирается случайным образом (так как это было рассмотрено ранее для аутентификатора). Если сервер желает согласовать другой субключ, то последний включается в сообщение. Сообщение KRB\_AP\_REP зашифровывается с помощью сеансового ключа, извлеченного из билета.

**Прием сообщения KRB\_AP\_REP**

Если клиенту на его запрос возвращено сообщение KRB\_AP\_REP, то он использует для его расшифрования сеансовый ключ, извлеченный из ВД, приобретенных для сервера. (Заметим, что для зашифрования сообщения KRB\_AP\_REP сеансовый субключ не используется, даже если он был представлен в аутентификаторе.) Далее он проверяет поля “отметка времени” и “микросекунды” на их соответствие тем, которые передавались на сервер в аутентификаторе. Если они совпадают, то клиент убеждается в том, что сервер подлинный.

**Использование криптографического ключа**

После того как завершен обмен сообщениями KRB\_AP\_REQ/ KRB\_AP\_REP, сервер и клиент совместно используют шифрключ, который может быть применен прикладными процессами. “Истинный” сеансовый ключ, используемый в сообщениях KRB\_PRIV, KRB\_SAFE или в других специальных прикладных процессах, может быть изменен с помощью прикладного процесса, который применяет субключи из сообщения KRB\_AP\_REP и аутентификатора. (Некоторые программные реализации KRB могут “высказывать пожелания” по обеспечению их специальными дополнительными программами, которые, во-первых, осуществляют выбор субключей на основе сеансовых ключей и случайных чисел, а во-вторых, управляют согласованием ключа, возвращаемого в KRB\_AP\_REP.) В отдельных случаях применение такого сеансового ключа будет подразумеваться в протоколе; в других случаях метод его применения должен выбираться из нескольких альтернатив. Каждый программист (разработчик прикладных программ) в праве сам определить, как использовать ключ в структуре протокола (то есть выбрать тип шифрования и расчета контрольных сумм; KRB не дает готовых решений по применению дополнительных функций).

Как и в односторонней, так и двухсторонней процедуре аутентификационного обмена участники сессии должны заботиться о том, чтобы не передавать друг другу “чувствительную к ошибкам и атакам” информацию без соответствующих гарантий. В частности, прикладные процессы, требующие обеспечения конфиденциальности и целостности сообщений, будут использовать для ответов сообщения KRB\_AP\_REP или KRB\_ERROR, направляемые сервером клиенту с целью обеспечения гарантированного подтверждения подлинности сторон (клиента и сервера). Если протокол прикладной службы требует обеспечения конфиденциальности своих сообщений, то он может использовать сообщение KRB\_PRIV. А для обеспечения целостности сообщений — KRB\_SAFE.

**Организация обмена сообщениями со СЛБР**

На рис.3 представлены основные типы сообщений и направления передачи, на которых они используются.

|  |  |
| --- | --- |
| Направление передачи сообщения | Тип сообщения |
| ➀ Клиент 🠲 Протокол KRB | KRB\_TGS\_REQ |
| ➁ Протокол KRB 🠲 Клиент | KRB\_TGS\_REP или KRB\_ERROR |

Обмен сообщениями СЛБР между клиентом и KRB-СБР инициируется пользователем, когда он “желает”:

* приобрести аутентификационные ВД для выбранного сервера (который, в свою очередь, может находиться в удаленной зоне);
* восстановить или ратифицировать имеющийся у него билет;
* приобрести уполномочивающий билет (для своего уполномоченного).

В первом случае клиент должен всегда приобретать билет для СЛБР, используя аутентификационный обмен со СЛА (ББР обычно приобретается тогда, когда клиент проходит первоначальную процедуру аутентификации при доступе в систему, то есть когда он регистрируется в ней). Формат сообщения для аутентификационного обмена со СЛБР, почти такой же, какой используется для сообщений СЛА. Основное отличие заключается в том, что процедуры зашифрования и расшифрования при аутентификационном обмене со СЛБР не применяют клиентский ключ. А вместо него используются сеансовые ключи, которые извлекаются из ББР или восстановленного билета, или субключи, извлекаемые из аутентификатора. Как и для всех прикладных серверов, просроченные билеты не доступны в СЛБР, и поэтому если однажды время жизненного цикла восстановленного билета или ББР истечет, то клиент обязан применить отдельную процедуру обмена сообщениями для приобретения действительных билетов.

Обмен сообщениями со СЛБР включает два сообщения:

* запрос клиента на СБР (KRB\_TGS\_REQ);
* ответ СБР на запрос клиента (KRB\_TGS\_REP или KRB\_ERROR).

Сообщение KRB\_TGS\_REQ включает информацию о клиенте, проходящим процедуру аутентификации, и запрос на ВД. Аутентификационная информация состоит из заголовка аутентификации (KRB\_AP\_REQ), который включает предварительно приобретенные клиентом ББР, восстановленный или недействительный билет. Если направляется ББР или билет от уполномоченного, то запрос может содержать (один или несколько):

* список сетевых адресов;
* набор типов данных для авторизации, размещаемых в билете с целью авторизации, осуществляемой прикладным сервером;
* дополнительные билеты (применение которых предполагается в дальнейшем).

Ответ СЛБР (KRB\_TGS\_REP) включает требуемые ВД, зашифрованный сеансовый ключ, который был указан в ББР и БВОСС, или (если он был представлен в запросе) сеансовый субключ, содержащийся в соответствующем поле аутентификатора (как часть аутентификационного заголовка в запросе). Сообщение KRB\_TGS\_REP включает также информацию, которая может использоваться для обработки ответов и для связи с другими сообщениями, на которые были получены эти ответы.

Сообщение KRB\_ERROR включает код ошибки и тест, который раскрывает сущность ошибки. Это сообщение не зашифровывается. Оно включает также информацию, которая может использоваться для связи с другими сообщениями, на которые был получен этот ответ. А отсутствие процедуры шифрования этого сообщения предотвращает возможность приема и обработки таких ответов нарушителем или подделки им данных сообщений.

**Формирование сообщения KRB\_TGS\_REQ**

Перед отправкой запроса в СЛБР клиент должен определить в какой зоне зарегистрирован прикладной сервер. (Замечание. Это можно осуществить несколькими способами. Необходимую зону можно узнать заблаговременно (так как имя зоны является частью идентификатора участника сессии) или сохранить имя зоны на сервере имен (“nameserver”). Через некоторое время, однако, эта информация извлекается из файла настройки. Если название зоны, которое будет использоваться, берется из сервера имен, то существует опасность обмана, так как этот сервер не прошел процедуру аутентификации. Угроза обмана может стать результатом либо использования зоны, которая себя скомпрометировала, либо попытки нарушителя скомпрометировать процедуру аутентификации прикладного сервера на узле клиента.) Если клиент не готов предоставить ББР для соответствующей зоны, то хотя бы один такой билет обязательно должен быть получен. Надо в первую очередь попытаться это сделать путем процедуры запроса ББР из локального KRB-СБР для конкретной зоны (используя сообщение KRB\_TGS\_REQ повторно). KRB-СБР может возвратить ББР для указанной зоны, из которой он в дальнейшем может поступить. В противном случае, KRB-СБР может возвратить ББР для зоны, которая является “закрытой” для запрашиваемой зоны (вследствие иерархической структуры построения зон). Тогда должна быть повторена эта процедура и с KRB-СБР, обслуживающим зону, которая определена в возвращаемом ББР. Если ни в одном, ни в другом случаях билет не был возвращен, то запрос обязательно должен быть передан повторно на СБР, обслуживающий зону, которая является высшей (старшей) в иерархии. Это запрос будет сам требовать ББР для старшей зоны, которая должна быть достижима за счет неоднократного обращения (передачи запроса) по всем направлениям.

После того как клиент приобрел ББР для необходимой зоны, он определяет какие KRB-СБР обслуживают эту зону и соединяется с одним из них. Перечень таких СБР может быть получен в файле настройки или от сетевой службы. (Пока секретные ключи, которыми обменивались зоны (серверы в этих зонах), хранились в секрете (безопасно), то от “ложного” KRB-СБР мог последовать только “отказ в обслуживании”.)

Как и при обмене со СЛА клиент может определить необходимое количество дополнительных функций в сообщении KRB\_TGS\_REQ. Пользователь формирует этот запрос, добавляя заголовок аутентификационный заголовок как элемент поля “padata” и включая такие же поля, которые используются в сообщении KRB\_AS\_REQ, вместе с несколькими дополнительными полями: поле “enc-authorization-data” для использования его прикладным сервером и дополнительные билеты необходимые для некоторых дополнительных функций.

При формировании аутентификационного заголовка клиент может выбрать сеансовый субключ, с помощью которого будет зашифрован ответ от KRB-СБР. (Если клиент выбирает сеансовый субключ, то он должен гарантировать случайный характер такого выбора. Случайность такого выбора будет обеспечиваться путем генерирования случайного числа и сложением его по модулю 2 с сеансовым ключом, изъятым из ББР.) Если субключ не определен, то тогда используется сеансовый ключ, представленный в ББР. Если в поле “enc-authorization-data” представлена информация, она должны быть зашифрована с помощью сеансового субключа. Если он представлен, то он содержится в аутентификаторе аутентификационного заголовка, а если не представлен, то его заменяет сеансовый ключ из ББР.

Когда сообщение сформировано, оно передается на KRB-СБР, обслуживающий выбранную зону.

**Прием сообщения KRB\_TGS\_REQ**

Сообщение KRB\_TGS\_REQ обрабатывается аналогично сообщению KRB\_AS\_REQ, но для него существует несколько обязательных дополнительных проверок. Во-первых, KRB-СБР должен определить какому серверу предназначен соответствующий билет, а также должен определить необходимый ключ для расшифрования этого билета. Если сообщение KRB\_TGS\_ REQ корректно, то оно будет использовано СЛБР, а для его расшифрования применяется содержащийся в нем ключ. Если ББР поступил из другой зоны, то в этом случае должен использоваться межзональный ключ. Если поступивший билет не является ББР для данной зоны, а предназначен для прикладного сервера в этой зоне, и в запросе установлены дополнительные функции “RENEW”, VALIDATE” или “PROXY”, а также сервер, для которого затребован билет, именуется также, как указано в принятом билете, то тогда ЦРК зашифрует билет в аутентификационном заголовке, используя ключ сервера, для которого он поступил. Если билет не может быть найден в поле “padata”, то возвращается сообщение об ошибке (“KDC\_ERR\_PADATA\_ TYPE\_NOSUPP”).

После расшифрования поступившего билета проверочная сумма, рассчитанная пользователем и находящаяся в аутентификаторе, должна сверяться с аналогичной, содержащейся в запросе, и если они не совпадают сообщение уничтожается и возвращается сообщение об ошибке “KRB\_AP\_ERR\_MODIFIED”. А если указанная проверочная сумма не соответствует вновь рассчитанной, или вообще не указана, то возвращается сообщение об ошибке “KRB\_AP\_ ERR\_ INAPP\_CKSUM”. Если метод расчета проверочной суммы не поддерживается сервером, то также возвращается сообщение об ошибке “KDC\_ ERR\_SUMTYPE\_NOSUPP”. Если в поле “authorization-data” представлена информация для процедуры авторизации, то она расшифровывается с помощью сеансового субключа, изъятого из аутентификатора.

Если в процессе расшифрования выявится ошибка в проверке на целостность данных, то возвращается сообщение об ошибке “KRB\_AP\_ERR\_ BAD\_INTEGRITY”.

**Формирование сообщения KRB\_TGS\_REP**

Сообщение KRB\_TGS\_REP очень похоже по своему формату на KRB\_AS\_REP (KRB\_KDC\_REP), но кодирование его некоторых полей отличается.

Ответ будет включать билет для запрашиваемого сервера. KRB-БД снова обращается к своей записи для этого сервера (эта запись содержит ключ и билет, который шифруется с использованием этого ключа). Если имеет место запрос на ББР для удаленной зоны и если нет необходимого межзонального ключа, то тогда KRB-СБР выбирает зону, с которой межзональный ключ согласован, но которая является “закрытой” для запрашиваемой зоны, и использует этот ключ вместо отсутствующего. Это единственный случай, когда ответ из ЦРК будет предназначаться другому серверу, а не тому, который запросил клиент.

По умолчании из ББР или восстановленного билета в новый билет будут копироваться следующие поля:

* адрес;
* имя клиента и зона;
* список транзитных зон;
* время начала процедуры аутентификации;
* время окончания ЖЦБ;
* данные авторизации.

Если поле для указания транзитных зон необходимо дополнить новой информацией, а такой тип информации не поддерживается сервером, то будет возвращено сообщение об ошибке (“KDC\_ERR\_TRTYPE\_NOSUPP”).

Если запрос время окончания билета, то тогда время окончания действия нового билета устанавливается как минимум из следующих источников:

1. из запроса;
2. время из ББР;
3. начальное время ББР в совокупности с минимальным значением из двух следующих:

* максимальное время ЖЦБ для прикладного сервера;
* максимальное время ЖЦБ для локальной зоны (максимальный ЖЦБ, затребованный участником сессии всегда определяется при выдаче ББР).

Если новый билет должен быть восстановлен, то окончание его времени ЖЦБ берется как минимум из двух следующих источников:

* 1. значение, указанное в поле “renew\_till” билета;
  2. начальное время нового билета в совокупности с временем жизненного цикла старого билета (“endtimestarttime”).

Если была запрошена дополнительная функция “FORWARDED”, то ответный билет будет содержать адреса, определенные клиента. Эта функция будет учитываться только тогда, когда в ББР установлен флаг “FORWARDABLE”. С дополнительной функцией “PROXY” ситуация аналогична: итоговый билет будет включать адреса, указанные клиентом, а учитываться она будет только в том случае, если установлен флаг “PROXIABLE”. Функция “PROXY” не будет учитываться в запросах на дополнительные ББР.

Если запрашиваемое начальное время отсутствует или просрочено, то тогда начальное время билета будет устанавливаться по текущему времени сервера, на котором клиент проходит процедуру аутентификации. Если еще время не наступило, а в ББР не определена функция “POSTDATED” или не установлен флаг “MAY-POSTDATE”, то тогда возвращается сообщение об ошибке (“KDC\_ERR\_CANNOT\_POSTDATE”). В противном случае, если ББР имеет установленный “MAY-POSTDATE”, то тогда итоговый билет будет БПОЗД и в нем будет запрос на начальное время, которое будет контролироваться администратором локальной зоны. Если это возможно, то начальное время билета устанавливается в соответствии с требуемым, но фиксируется флаг “INVALID”. БПОЗД обязательно должен быть ратифицирован (еще до начала его использования) путем предоставления его в ЦРК, но только после того как наступило начальное время. Однако, начальное время, время окончания и время ЖЦБ для вновь выданного БПОЗД не могут быть больше чем восстановленное время ЖЦБ ББР.

Если в запросе установлена дополнительная функция “ENC-TKT-IN-SKEY” и включен дополнительный билет, то тогда ЦРК будет расшифровывать этот билет с помощью ключа для сервера (получателя этого билета) и определит, что это ББР. Если имя запрашиваемого сервера на самом деле в запросе отсутствует, то тогда будет использоваться имя клиента, указанное в дополнительном билете. В противном случае, имя запрашиваемого сервера будет сравниваться с именем клиента в дополнительном билете и если они различны, то запрос уничтожается. Если запрос достиг цели, то сеансовый ключ (полученный вместо используемого ключа сервера) из дополнительного билета будет применяться для зашифрования нового билета, предназначенного для данного сервера. (Это делает достаточно легкой процедуру межабонентской аутентификации, которая использует сеансовые ключи из ББР вместо секретных ключей сервера, а это особенно важно в тех случаях, когда такие секретные ключи могут быть легко скомпрометированы.)

Имя сервера в билете, который представлен в ЦРК как часть аутентификационного заголовка в запросе, не является собственным именем СБР, зарегистрированного в зон ЦРК, то тогда запрос уничтожается.

Если запрашивается дополнительная функция “RENEW”, то тогда ЦРК определит, что в билете установлен флаг “RENEWABLE” и что начальное время ЖЦБ, указанное в поле “renew\_till”, еще не наступило. Если запрашивается дополнительная функция “VALIDATE” ЦРК будет проверять наличие в переданном сообщении время начала ЖЦБ и установленного флага “INVALID”. В случае если затребована дополнительная функция “PROXY” ЦРК будет проверять наличие в билете установленного флага “PROXIABLE”. Если все необходимые проверки прошли успешно, то ЦРК будет выдавать соответствующий новый билет.

Всякий раз когда на СБР поступает запрос, представленный(е) в нем билет(ы) проверяется на наличие его в списке “украденных” билетов, и если проверка дала положительный результат, то билет(ы) аннулируются. Этот список может быть дополнен данными о так называемых подозрительных билетах, в которых указывается перечень дат, когда такие билеты выдавались; и если поступил билет, в котором указано время начала аутентификации, попадающее в этот перечень, то данный билет уничтожается. Таким же образом украденный ББР или ПВОСС не могут быть использованы для приобретения дополнительных билетов (восстановленных или других) если они однажды были объявлены сворованными. Любой другой обычный билет, включенный в сообщение до того как он был объявлен украденным, тем не менее будет считаться действительным (так как для него не требуется взаимодействие с ЦРК), но только до тех пор пока не истечет время.

Шифруемая часть ответа в сообщении KRB\_TGS\_REP зашифровывается с помощью сеансового ключа, изъятого из аутентификатора (если он представлен), или с помощью сеансового ключа из ББР. Но она не зашифровывается с помощью секретного ключа клиента. Кроме того, поля “дата окончания действия ключа клиента” (“client's key's expiration date”) и “номер версии ключа” (“key version number”) пропускаются (на заполняются), хотя эти величины хранятся в KRB-БД совместно с другой информацией о клиенте, но они не нужны для ответа запрос, основанного на ББР.

**Прием сообщения KRB\_TGS\_REP**

Кода клиент принимает сообщение KRB\_TGS\_REP он обрабатывает его также, как и KRB\_AS\_REP, рассмотренное выше. Единственное отличие заключается в том, что зашифрованная часть ответа должна расшифровываться с использованием сеансового ключа, изъятого из ББР, что является предпочтительнее, чем секретный ключ клиента.

**Обмен сообщениями KRB\_SAFE**

Сообщение KRB\_SAFE может быть использовано клиентом по его запросу, если он желает иметь возможность обнаруживать модифицированные сообщения в данном информационном обмене. С этой целью в сообщения будет включаться специальная проверочная сумма (последовательность), полученная из данных пользователя, и некоторая контрольная информация. Проверочная последовательность вычисляется с использованием шифрключа (обычно это последний ключ, полученный из субключей, или сеансовый ключ, если субключ не был получен).

**Формирование сообщения KRB\_SAFE**

Когда прикладному процессу необходимо передать сообщение KRB\_SAFE, он выбирает определяет свои данные и соответствующую контрольную информацию и по ним вычисляет проверочную сумму. Проверочная сумма рассчитывается по определенному алгоритму, который реализует выбранную однонаправленную хэш-функцию (“RSA-MD5-DES”, “DES MAC” или др.), вычисляемую с использованием сеансового субключа или при его отсутствии с помощью сеансового ключа. В сообщении могут быть использованы различные алгоритмы в зависимости от типа проверочной последовательности. Бесключевые и неоднозначные проверочные последовательности не применяются.

Контрольная информация для сообщения KRB\_SAFE включает отметку времени и последовательный номер (“timestamp” и “sequence number”). Разработчик прикладных программ, использующих сообщение KRB\_SAFE, должен выбирать по крайней мере один из двух указанных выше механизмов. Этот выбор должен основываться на потребностях прикладного протокола.

Последовательные номера наиболее приемлемы, когда все передаваемые сообщения принимаются одним участником сессии. Если состояние соединения требует использования сеансового ключа, то использование последовательного номера не вызовет дополнительных проблем.

Если прикладной протокол допускает потерю сообщений без их повторной передачи, то использование временных меток является весьма нужным механизмом приема ответа. Применение отметок времени является также необходимой процедурой и для многопользовательских протоколов, при использовании которых все до одного клиентов обладают общим сеансовым субключом, а некоторые сообщения будут передаваться только для выделенной подгруппы пользователей.

После расчета проверочной суммы клиент передает необходимую информацию и эту сумму получателю в сообщении KRB\_SAFE.

**Прием сообщения KRB\_SAFE**

Когда прикладной процесс принимает сообщение KRB\_SAFE, он обрабатывает его следующим образом. Если обнаружена какая либо ошибка, то будет указываться код ошибки для использования его приложением.

Во-первых, сообщение проверяется на предмет соответствия содержащихся в нем версии протокола и полей номеру текущей версии и типу полей сообщения KRB\_SAFE. В случае их несоответствия формируется код ошибки “KRB\_AP\_ERR\_BADVERSION” или “KRB\_AP\_ERR\_MSG\_TYPE”. Приложение далее проверяет используемый алгоритм вычисления контрольной суммы. И если он является бесключевым и дает неоднозначный результат (проверочную последовательность), то тогда вырабатывается код ошибки “KRB\_AP\_ERR\_INAPP\_CKSUM”. Получатель сравнивает адрес передающей стороны в сообщении с адресом отправителя, хранящимся в памяти ОС, а также проверяет (если адрес получателя определен или получатель затребовал адрес), что один из адресов получателя соответствует адресу получателя в сообщении. Если в одной из проверок обнаружено несоответствие, то тогда формируется код ошибки “KRB\_AP\_ERR\_ BADADDR”. Затем проверяются поля “timestamp”, “usec” (микросекунды) и/или “sequence number”. Если поля “timestamp” и “usec” должны быть, но отсутствуют, или они не соответствуют текущему времени, то тогда выдается код ошибки “KRB\_AP\_ERR\_SKEW”. Если имя сервера вместе с именем клиента, поля “timestamp” и “usec”, изъятые из аутентификатора, совпадают с какими-либо аналогичными предшествующими записями, то тогда формируется код ошибки “KRB\_AP\_ERR\_REPEAT”. Если имеет место ошибочный последовательный номер или он не соответствует ожидаемому — код ошибки “KRB\_AP\_ERR\_BADORDER”. Если не представлены поля “timestamp”, “usec” или “sequence number” — код ошибки “KRB\_AP\_ERR\_ MODIFIED”.

И в заключении, если представленная в сообщении проверочная сумма не совпала с вновь вычисленной (по данным и контрольной информации) на приемной стороне — код ошибки “KRB\_AP\_ERR\_MODIFIED”.

Если же все проверки прошли успешно, то прикладной процесс убеждается в том, что сообщение было сформировано пользователем и не было подделано при передаче.

**Обмен сообщениями KRB\_PRIV**

Сообщение KRB\_PRIV может использоваться клиентами если они хотят обеспечить конфиденциальность своей информации и возможность выявления модифицированных сообщений. Это достигается путем зашифрования сообщений и введения дополнительной контрольной информации.

**Формирование сообщения KRB\_PRIV**

Когда прикладной процесс желает передать сообщение KRB\_PRIV, то он определяет собственные данные и необходимую контрольную информацию для дальнейшего зашифрования с помощью шифрключа (обычно это последний ключ, полученный через субключи, или сеансовый ключ, если первый не был получен). Клиент должен решить что ему применить: либо отметку времени, либо последовательный номер (либо оба механизма), — в качестве контрольной информации. После принятия решения данные и контрольная информация зашифровываются и клиент передает зашифрованный текст и необходимую обрамляющую информацию получателю.

**Прием сообщения KRB\_PRIV**

Когда прикладной процесс принимает сообщение KRB\_PRIV, он обрабатывает его следующим образом. Если обнаружена какая либо ошибка, то будет указываться код ошибки для использования его приложением.

Во-первых, сообщение проверяется на предмет соответствия содержащихся в нем версии протокола и полей номеру текущей версии и типу полей сообщения KRB\_PRIV. В случае их несоответствия формируется код ошибки “KRB\_AP\_ERR\_BADVERSION” или “KRB\_AP\_ERR\_MSG\_TYPE”. Затем приложение расшифровывает шифртекст и обрабатывает полученный открытый текст. Если во время процедуры расшифрования выявлена подделка данных, то выдается код ошибки “KRB\_AP\_ERR\_BAD\_INTEGRITY”. Получатель сравнивает адрес передающей стороны в сообщении с адресом отправителя, хранящимся в памяти ОС, а также проверяет (если адрес получателя определен или получатель затребовал адрес), что один из адресов получателя соответствует адресу получателя в сообщении. Если в одной из проверок обнаружено несоответствие, то тогда формируется код ошибки “KRB\_AP\_ ERR\_BADADDR”. После этого проверяются поля “timestamp”, “usec” и/или “sequence number”. Если поля “timestamp” и “usec” должны быть, но отсутствуют, или они не соответствуют текущему времени, то тогда выдается код ошибки “KRB\_AP\_ERR\_SKEW”. Если имя сервера вместе с именем клиента, поля “timestamp” и “usec”, изъятые из аутентификатора, совпадают с какими-либо аналогичными предшествующими записями, то тогда формируется код ошибки “KRB\_AP\_ERR\_REPEAT”. Если имеет место ошибочный последовательный номер или он не соответствует ожидаемому — код ошибки “KRB\_ AP\_ERR\_BADORDER”. Если не представлены поля “timestamp”, “usec” или “sequence number” — код ошибки “KRB\_AP\_ERR\_ MODIFIED”.

Если же все проверки прошли успешно, то прикладной процесс может предполагать, что сообщение было сформировано истинным клиентом и было защищено при передаче (то есть не было вторжения с целью прочтения содержания на основе дешифрования).

**Обмен сообщениями KRB\_CRED**

### Сообщение KRB\_CRED может использоваться клиентом для передачи ВД от одного узла к другому. Это достигается путем передачи билетов с зашифрованной информацией, включающей сеансовые ключи и другие необходимые данные, связанные с билетами.

**Формирование сообщения KRB\_CRED**

Когда прикладному процессу необходимо передать сообщение KRB\_ CRED, он, во-первых, получает ВД (используя обмен сообщениями со СЛБР) для передачи их удаленному узлу. Затем он формирует сообщение KRB\_ CRED, используя для этого билет или билеты, также заранее полученные, и размещая сеансовый ключ (необходимый для применения каждого билета) в поле “key”, входящее в последовательность “KrbCredInfo”, которая является шифруемой частью этого сообщения.

Другая информация, связанная каждым билетом и получаемая в течении всего периода обмена сообщениями со СЛБР, также входит в состав последовательности “KrbCredInfo”, которая является шифруемой частью сообщения KRB\_CRED. Текущее время и поля “s-address” и “raddress” (если в данном случае они были затребованы приложением) размещаются в шифруемой части сообщения KRB\_CRED, которая затем зашифровывается с помощью шифрключа, который был предварительно получен в результате аутентификационного обмена между клиентом и СЛБР (обычно это последний ключ, полученный из субключей, или сеансовый ключ, если субключ не был получен).

**Прием сообщения KRB\_CRED**

Когда прикладной процесс принимает сообщение KRB\_CRED, он обрабатывает его следующим образом. Если обнаружена какая либо ошибка, то будет указываться код ошибки для использования его приложением.

Сообщение проверяется на предмет соответствия содержащихся в нем версии протокола и полей номеру текущей версии и типу полей сообщения KRB\_CRED. В случае их несоответствия формируется код ошибки “KRB\_ AP\_ERR\_BADVERSION” или “KRB\_AP\_ERR\_MSG\_TYPE”. Затем приложение расшифровывает шифртекст и обрабатывает полученный открытый текст. Если во время процедуры расшифрования выявлена подделка данных, то выдается код ошибки “KRB\_AP\_ERR\_BAD\_INTEGRITY”.

Получатель сравнивает адрес передающей стороны в сообщении с адресом отправителя, хранящимся в памяти ОС, а также проверяет (если адрес получателя определен или получатель затребовал адрес), что один из адресов получателя соответствует адресу получателя в сообщении. Если в одной из проверок обнаружено несоответствие, то тогда формируется код ошибки “KRB\_AP\_ERR\_BADADDR”. После этого проверяются поля “timestamp”, “usec” (и поле “случайная величина” (“nonce”), если оно затребовано). Если поля “timestamp” и “usec” должны быть, но отсутствуют, или они не соответствуют текущему времени, то тогда выдается код ошибки “KRB\_AP\_ERR\_ SKEW”.

Если же все проверки прошли успешно, то прикладной процесс сохраняет в кэш-памяти каждый новый билет вместе с сеансовым ключом и другой информацией, размещенной в последовательности “KrbCredInfo”, которая является шифруемой частью сообщения KRB\_CRED.

**KRB-БД**

KRB-сервер должен иметь доступ к KRB-БД, содержащей идентификаторы и секретные ключи участников сессии, для проведения процедур аутентификации. (При реализации KRB-сервера целесообразно KRB-БД и собственно сервер размещать на разных ЭВМ. Вполне допустимо хранить данные об участниках сессии, например, на сервере сетевых имен, пока они хранятся на нам они защищены от несанкционированного доступа к ним и от модификации при неавторизованном проникновении. Однако, такие стратегии безопасности не рекомендуются они могут чрезвычайно усложнить управление системой и выявление угроз.)

**Содержание KRB-БД**

Содержание KRB-БД должно как минимум включать следующие поля:

* “name” — идентификатор участника сессии;
* “key” — секретный ключ участника сессии;
* “p\_kvno” — версия ключа участника сессии;
* “max\_life” — максимальное время ЖЦБ;
* “max\_renewable\_life” — максимальное общее время ЖЦБ для БВОСС.

Поле “name” представляет собой закодированный идентификатор участника сессии. Поле “key” содержит зашифрованный ключ, то есть секретный ключ участника сессии. (Ключ может быть зашифрован с помощью “мастер-ключа” протокола KRB еще до того как он будет размещен в KRB-БД. Это делается для того, чтобы защитить секретный ключ от возможной компрометации KRB-БД, если сам “мастер-ключ” не скомпрометирован. Если используется “мастер-ключ”, то добавляется дополнительное поле, в котором указывается версия “мастер-ключа”.) Поле “p\_kvno” представляет номер версии секретного ключа участника сессии. Поле “max\_life” содержит значение максимально допустимого времени жизненного цикла (“endtime – starttime”) любого билета, выдаваемого конкретному участнику сессии. Поле “max\_ renewable\_life” содержит значение максимально допустимого времени жизненного цикла для любого БВОСС, выдаваемого конкретному участнику сессии.

Один сервер может обеспечивать ЦРК-обслуживание нескольких зон, пока реализованная в нем KRB-БД поддерживает механизм распознавания записей идентификаторов участников сессий, которые отличаются только по именам зон.

Когда происходит обмен ключами с прикладным сервером, а сам обмен является стандартным (то есть не является результатом компрометации старого ключа), старый ключ будет сохраняться сервером до тех пор, пока все билеты, выданные с использованием этого ключа, не будут просрочены. Из этого следует, что существует вероятность одновременного действия нескольких активных ключей одного участника сессии. Шифртекст, зашифрованный с помощью ключа участника сессии, всегда имеет указатель версии ключа для того, чтобы помочь получателю найти необходимый ключ для расшифрования.

Когда для участника сессии активизировано несколько ключей, то для него отводится несколько записей в KRB-БД. Ключи и номера их версий будут различаться в записях (а прочие поля могут отличаться, а могут быть одинаковыми). Всякий раз, когда KRB выдает билет или отвечает на запрос начала процедуры аутентификации, то для шифрования будет использоваться самый “свежий” ключ (известный KRB-серверу). Это ключ будет иметь наивысший номер версии ключа.