СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Р.Р. Фаткиева

Основы построения защищенных компьютерных сетей

Протоколы идентификации, аутентификации и авторизации

СПбГЭТУ «ЛЭТИ», 2021 г.

7 ПРОТОКОЛЫ ИДЕНТИФИКАЦИИ, АУТЕНТИФИКАЦИИ И АВТОРИЗАЦИИ

7.1 Методы аутентификации

При первичной регистрации в сети пользователю присваивается идентификатор и аутентфикатор, хранящейся в базе паролей. К наиболее часто используемым методам аутентификации относится ***аутентификация по многоразовым паролям***. При входе пользователь вводит идентификатор и аутентификатор, на основании которого вычисляются в случае необходимости различные преобразования для передачи их по сети. В дальнейшем определяется является ли учетная запись локальной, тогда она сравнивается, хранящейся в базе паролей. В случае, если запись не является локальной, то вызывается программа проверки подлинности на удаленном компьютере-сервере (**Ошибка! Источник ссылки не найден.**8). При успешной аутентификации создается процесс- оболочка, которая приписывает всем процессам пользователя права доступа.

Достоинство: простота и привычность.

Недостатки:

1. Угадывание "методом грубой силы"(используя, скажем, словарь)

2. Копирование файла пароля и подбор программированием полного перебора (предполагается, что алгоритм шифрования известен)

Противодействие:

- управление сроком действия паролей, по формуле:

$$T = \frac{A^S \cdot E/R}{2},$$

где E-число символов в пароле; R-скорость передачи пароля; количество возможных перестановок символов в пароле; S-длина пароля;

- управление длиной пароля, не менее 8 символов;
- ограничение доступа к файлу паролей;
- ограничение числа неудачных попыток входа в систему (это затруднит применение "метода грубой силы");
- использование программных генераторов паролей (такая программа, основываясь на несложных правилах, может порождать только благозвучные и, следовательно, запоминающиеся пароли).

Комбинированные методы для подтверждения подлинности субъекта помимо пароля требуют введение дополнительных предметов (мобильных телефонов, смарт-карт, токенов) или атрибутов (криптографических сертификатов).

Методы, основанные на информации о субъекте в своей работе, содержат информацию, однозначно идентифицирующую объект, например поведенческие паттерны или биометрические характеристики. Основные характеристики методов идентификации представлены в **Ошибка! Источник ссылки не найден.**.

Таблица7.1

Способ аутентификации	*Основное применение*	*Протоколы*	*Характеристики*
По многоразовым паролям	Аутентификация пользователей (web, ОС, сетевые службы)	HTTP, Forms	Мощность множества, длина
По одноразовым паролям	Аутентификация пользователей (Web-сайты, OTP-токены)	Forms	Мощность множества, длина Длина счетчика
По сертификатам	Аутентификация пользователей в безопасных приложениях; аутентификация сервисов	SSL/TLS	Мощность множества, длина Длина счетчика Вид криптографического протокола
По ключам доступа	Аутентификация сервисов и приложений	-	Характеристики ключа
По токенам	Делегированная аутентификация пользователей; делегированная авторизация приложений	SAML, WS-Federation, OAuth, OpenID Connec	Характеристики ключа

7.2 Одноразовые пароли

Одноразовый пароль – это динамическая аутентификационная информация, генерируемая различными способами для однократного использования. К подходам по построению протоколов аутентификации с одноразовыми паролями можно отнести:

Разделяемые списки одноразовых паролей (пользователь и система имеют согласованный список паролей, который проходит проверку при очередном сеансе).

Последовательно обновляемые одноразовые пароли (пользователь и система имеют согласованный список паролей, генерируемый на зашифрованном на ключе, полученном из предыдущего пароля)

Последовательности одноразовых паролей, основанные на однонаправленных функциях (пользователь и система имеют согласованную однонаправленную функцию, на основании которой формируется пароль).

Одним из наиболее распространенных протоколов аутентификации на основе одноразовых паролей является стандартизованный в Интернете протокол S/Key (RFC 1760, RFC 1938).

Система S/KEY (Алгоритм Лампорта)

1. Пусть односторонняя функция f функция известна и пользователю, и серверу аутентификации. Пользователь знает секретный ключ S, а также число одноразовых паролей- n.
2. Функция f применяется к ключу S n раз, после чего результат сохраняется на сервере.
3. При первичной регистрации сервер присылает пользователю число n-1
4. Пользователь применяет функцию f к секретному ключу S (n-1) раз и отправляет результат по сети на сервер аутентификации.
5. Сервер применяет функцию f к полученному от пользователя значению и сравнивает результат с ранее сохраненной величиной. В случае совпадения подлинность пользователя считается установленной, сервер запоминает новое значение (присланное пользователем) и уменьшает на единицу счетчик (n).
6. При обнулении счетчика n сервер и пользователь изменяют секретный ключ S.

7.3 Классификация и характеристики протоколов аутентификации

Применение открытых каналов передачи данных требует разработки методов, позволяющих одной или нескольким сторонам взаимодействия осуществлять проверку подлинности. В связи с этим в настоящее время сформировано несколько *видов аутентификации сторон:*

- полная анонимность;
- односторонняя проверка подлинности;
- многосторонняя аутентификация;
- многофакторная аутентификация.

При построении системы защиты, основанной на методах аутентификации возникает необходимость классифицировать протоколы аутентификации по уровню обеспечиваемой безопасности или возможности противостоять определенному классу атак, согласно х*арактеристикам протоколов аутентификации [1]:*

- вычислительная эффективность – количество операций, необходимых для выполнения протокола;
- коммуникационная эффективность –количество сообщений и их длину, необходимую для осуществления аутентификации;
- количество сторон, участвующих во взаимодействии;
- наличие третьей стороны – примером третьей стороны может служить доверенный сервер распределения симметричных ключей или сервер, реализующий дерево сертификатов открытых ключей
- способы хранения секрета –способ хранения критичной ключевой информации.

7.4 Протокол PAP

Двусторонний протокол обмена подтверждениями, предназначенный для совместного использования с протоколом PPP.

К задачам протокола относят:

-управление качеством линии (PPP отключает связь, если количество ошибок превысит заданное значение);

-обеспечение аутентификации сторон.

Для осуществления аутентификации (**Ошибка! Источник ссылки не найден.**) осуществляется:

1. Формирование аутентификационных данных (идентификатор и пароль хоста) и передача их по сети.

2. Сравнение полученных по сети аутентификационных данных, с данных, хранящимися в базе данных проверяющей стороны.

3. Передача подтверждения, в случае успешной проверки аутентификационных данных.

4. Обмен информации.

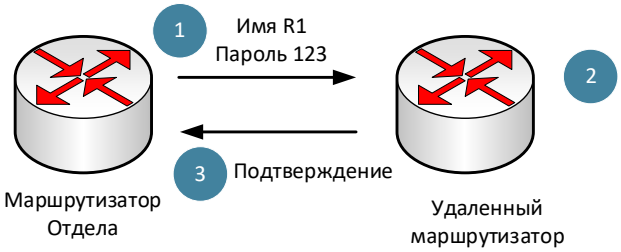


Рис. 7.1

К недостатку протокола относят передачу незашифрованных аутентификационных данных по сети, в связи с этим использование протокола в открытых сетях крайне небезопасно.

Однако использование протокола возможно в ЛВС, например при настройке работы маршрутизатора.

Стандарт ISO / IEC 9798-2

Стандарт определяет механизмы аутентификации [2-7] пользователей с использованием аутентифицированных алгоритмов шифрования, с применением дополнительных функцией.

Односторонняя аутентификация, основанная на метке времени, позволяет сформировать хэш пароля пользователя с учетом присланной от проверяющей стороны метки времени. В этом случае необходимо, чтобы разница между метками времени отправителя и получателя укладывалась в определенный интервал времени – окно принятия (acceptance window) (**Ошибка! Источник ссылки не найден.**2).

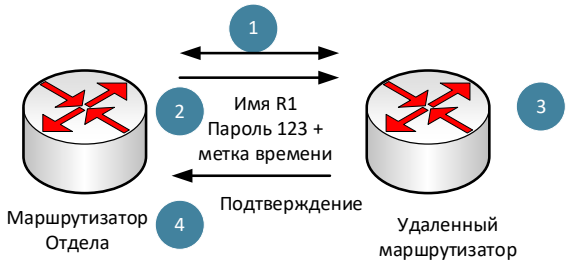


Рис. 7.2

Алгоритм реализации:

1. Клиент устанавливает соединение.

2. Клиент получает метку времени с системного таймера проверяющей стороны и криптографически складывает с аутентификационными параметрами.
3. Проверяющая сторона получает текущее время со своих локальных часов и сравнивает с величиной метки времени, полученной от клиента, в случае совпадения клиент допускается в систему.

Односторонняя аутентификация с использованием случайных чисел.

Алгоритм реализации:

1. Клиент отправляет запрос соединение (**Ошибка! Источник ссылки не найден.**3).
2. Клиент получает от проверяющей стороны случайное число, сгенерированное с помощью генератора псевдослучайных чисел. С помощью алгоритма шифрования над полученным случайным числом и (если необходимо) идентификатором и аутентификатором осуществляется криптографическое преобразование, с передачей этих данных по сети.
3. Проверяющая сторона расшифровывает полученный шифртекст и проверяет структуру запроса. Если она верна, соединение устанавливается.

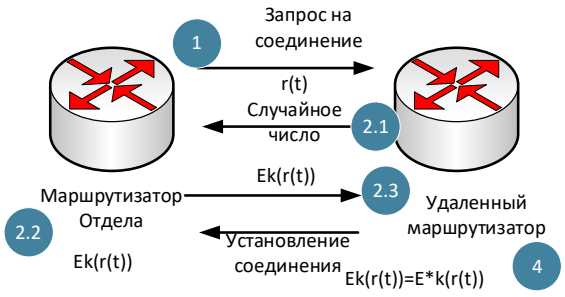


Рис. 7.3

Взаимная аутентификация с использованием случайных чисел. В этом случае одна и более сторон осуществляют проверку друг друга. Протокол допускает замену шифра на хеш-функцию с ключом (стандарте ISO/IEC 9798-4). Для повышения стойкости протокола к передаваемым сообщениям можно добавить метки времени (**Ошибка! Источник ссылки не найден.**4).

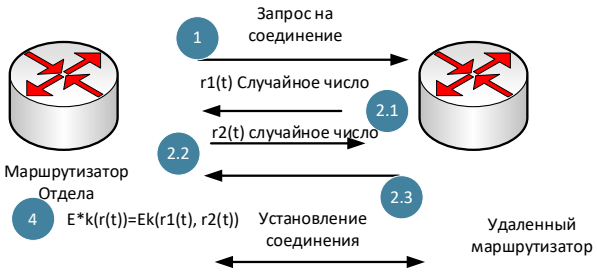


Рис. 7.4

7.5 Протокол CHAP

Основан на шифровании с помощью односторонней хэш-функции и по существу является односторонним, т. е. не сопровождается обратным преобразованием — расшифровыванием на приемной стороне. Обе стороны (отправитель и получатель) используют одну и ту же процедуру одностороннего шифрования (**Ошибка! Источник ссылки не найден.**5).

Рис. 7.5

Алгоритм реализации:

1. Установление PPP-соединения между клиентом и принимающей стороной.
2. Проверяющая сторона отправляет клиенту пакет CHAP, имеющий тип Challenge (вызов), который содержит в себе ключ и случайное число.
3. Узел на основе полученного ключа, случайного числа и пароля, вычисляет хеш с помощью алгоритма хеширования и отправляет пакет CHAP, содержащий в себе вычисленный хеш.
4. Проверяющая сторона сравнивает полученное значение хеша со своим расчётом ожидаемого значения хеша.

5. Если значения совпадают, то проверка подлинности считается успешной. При отличающихся значениях происходит разрыв соединения.
6. Через различные промежутки времени проверяющая сторона посылает новый запрос узлу, и шаги 1-3 повторяются.

CHAP обеспечивает защиту от использования чужих паролей за счет пошаговых изменений идентификатора и применения переменной величины запроса. Повторяющиеся запросы предназначены для ограничения времени, в течение которого система теоретически остается подверженной любой отдельной хакерской атаке. Частоту и количество неудачных попыток входа в систему контролирует идентификатор.

7.6 Аутентификация в протоколе HTTP

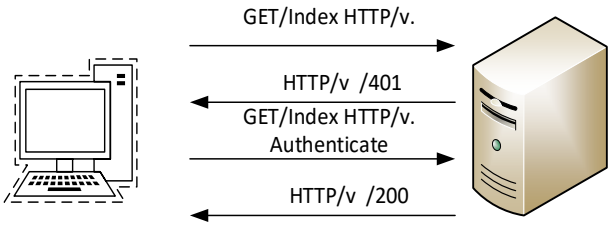
Схемы аутентификации:

При обращении к www-серверу, содержащему открытую информацию аутентификация не используется, однако, протокол http предоставляет возможность использование 2 схем аутентификации: basic и digest (Рис. 7.6).

Схема Basic выглядит следующим образом:

1. Браузер клиента обращается к серверу с запросом на некоторый документ.
2. Так как для доступа к ресурсам сервера требуется аутентификация в поле запроса– Authorization, которые браузер клиента не предоставил, то сервером генерируется сообщение кодом статусной строки 401 “Unauthorized” и заголовком ответа WWW-Authenticate: Basic realm= «Secret Database», в котором указывается метод аутентификации Basic и ресурс realm.
3. Получив ответ сервера, браузер генерирует окно с приглашением ввода имени и пароля пользователя. После ввода которых браузер вновь обращается к серверу, в заголовке запроса которого указывается схема аутентификации, преобразованные по алгоритму Base 64 имя и пароль, разделенные двоеточием.
4. При получении запроса сервер извлекает из заголовка имя и пароль и сверяет их по своей базе данных. При положительной аутентификации запрос передается на обслуживание. При

отрицательной трехкратно повторяется п.3, после чего в окне браузера клиента отображается содержимое ответа сервера.

5. 

6. Рис. 7.6

Недостаток данной схемы заключается в том, что протокол http-протокол без сохранения состояния, поэтому при обращении к следующему документу в этой же сессии, сервер посылает запрос на новую аутентификацию. Однако браузер в этом случае, не беспокоя пользователя самостоятельно подставит уже введенные пользователем данные, и только в случае, если эти данные не пройдут аутентификацию на сервере, то браузер сгенерирует окно для ввода имени и пароля. Однако данная схема облегчает работу как пользователя (не требует постоянное введение пароля), так и работу нарушителя, прослушивающего сеть (так как пароль передается при каждом новом запросе).

Для устранения данного недостатка используют схему Digest:

1. Браузер клиента обращается к серверу с запросом на некоторый документ.
2. Так как для доступа к ресурсам сервера требуется аутентификация в поле запроса- Authorization, которые браузер клиента не предоставил, то сервером генерируется сообщение кодом статусной строки 401 “Unauthorized” и заголовком ответа:

Заголовок ответа	Описание
WWW-Authenticate: Digest realm= «документ»,	метод аутентификации Digest и ресурс realm
Domain= “URI1;URI2”,	Список документов, внутри которых будут действительны запрашиваемые имя и пароль
nonce=”значение1”	Случайное число, генерируемое сервером и используемое при хешировании пароля
[opaque=”значение2”]	Идентификатор сессии

Stale={true|false},	Установленное поле true указывает браузеру о неудачной аутентификации
algorithm={“MD5” “MD5-sess”}	Используемый алгоритм хэширования
Qpo={“auth”| “auth-int”}	

3. Получив ответ сервера, браузер генерирует окно с приглашением ввода имени и пароля пользователя. После ввода которых, браузер генерирует:

3.1 заголовок запроса, в котором указываются следующие поля:

Заголовок запроса	**Описание**
Authorization: Digest username= «Имя пользователя»,	Метод аутентификации Digest имя пользователя
realm=”документ”	ресурс
nonce=”значение1”	Случайное число, присланное сервером и используемое при хешировании пароля
uri=”URI”	Используется при хешировании пароля и должно совпадать с запрашиваемым ресурсом
algorithm={“MD5” “MD5-sess”}	Используемый алгоритм хэширования
qpo={“auth”| “auth-int”}	«Метод защиты»
[opaque=”значение2”]	Идентификатор сессии
cnonce==”значение3”	Число, генерируемое браузером и используемое при хэшировании пароля
nc= счетчик	Указывает количество одинаковых значений поля nonce
Response=”хеш пароля”	

3.2 хеш пароля, вычисление которого производится по алгоритму MD5/

4. Заполненные заголовки запроса посылаются серверу.

5. При получении запроса сервер извлекает из заголовка данные и сверяет их по своей базе данных, содержащей значения хешированных паролей. При положительной аутентификации запроса сервер включает в ответ заголовок Authentication-Info:

Заголовок ответа	**Описание**

Authorization-Info: nextnonce=”значение”	Указание нового значение nonce, которое должно использоваться для вычисления дайджеста при последующем запросе
qpo={“auth”| “auth-int”}	«Метод защиты» Если qpo=auth, то A2=:URI Если qpo=auth-int, то A2 URI:MD5 (данные ответа)
cnonce==”значение2”	Число, генерируемое браузером и используемое при хэшировании пароля
nc= счетчик	Указывает количество одинаковых значений поля nonce
rsauth=”дайджест”	Сервер вычисляет дайджест ответа, по аналогии с дайджестом клиента

и передает запрос на обслуживание.

Аутентификация с использованием форм.

Алгоритм реализации:

1. Инициация клиентом запроса на получение формы для ввода аутентификационных данных (**Ошибка! Источник ссылки не найден.**7).
2. В HTML-форму, вводиться username/password и отправить их на сервер через HTTP POST для аутентификации. В случае успеха веб-приложение создает session token, который обычно помещается в browser cookies.
3. При последующих веб-запросах session token автоматически передается на сервер и позволяет приложению получить информацию о текущем пользователе для авторизации запроса.

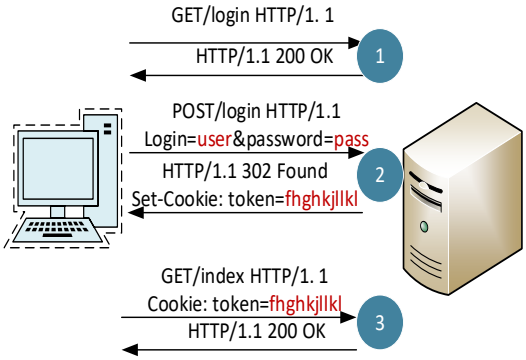


Рис. 7.7

7.7 Аутентификация с использованием NAS

Применение различных вариантов аутентификации с использованием инфраструктуры сети (**Ошибка! Источник ссылки не найден.**8), позволяет

подключать дополнительные устройства для управления доступом при передаче сетевого трафика. В этом случае возможно использование серверов **NAS** (Network Access Server) или **BRAS** (Broadband Remote Access Server), т.е. устройств, отвечающих за маршрутизацию внутри сервисной сети и предоставление доступа пользователям, согласно их учетным записям, к конкретным широкополосным сервисам Triple Play (Интернет, IP-телефония, IP-телевидение) посредством одного физического подключения (например BRAS находится в ядре сети провайдера и агрегирует пользовательские подключения из сети уровня доступа, с обеспечением перевода «серых» IP-адЫресов в белые).

Такая схема позволяет не только осуществлять аутентификацию пользователей на элемента NAS/BRAS, но и уменьшать циркуляцию трафика внутри сети провайдера за счет использования дополнительных сервисов, представленных на:

- NMS- **система управления элементами сети**, программное обеспечение предназначенное для управления и контроля как отдельных элементов, так и групп однотипных элементов;
- AAA –сервер аутентификации, авторизации, управление доступом. Здесь хранятся пароли, параметры подключения пользователя. Это сервис необходим для работы **BRAS.**
- Service Platform – система представления дополнительных сервисов со стороны поставщика услуг (игровые, файловые)

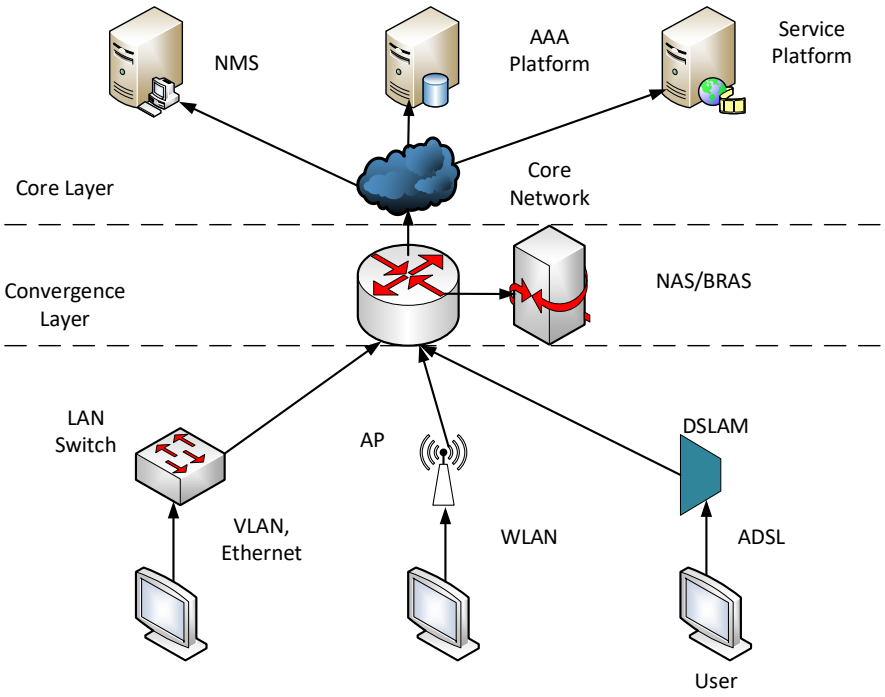


Рис. 7.8

К основным задачам BRAS относят:

- аутентификацию пользователей и устройств;
- выделение IP адреса;
- управление ресурсами;
- обеспечение безопасного соединения.

Рассмотрим управляющие элементы, входящие в структуру BRAS (**Ошибка! Источник ссылки не найден.**9). Основным сервисом является менеджер соединений. Сообщения от пользователя передается на модуль идентификатора пользователя, который отвечает за принятие пакетов пользователя и принимает решение что необходимо с ним сделать (отбросить, передать на вышестоящий уровень). Затем пакет передается на рассмотрение менеджеру AAA, а в случае необходимости с запросом к серверу AAA для аутентификации. Если проверка прошла удачно, то полномочия пользователя проверяются с использованием политики безопасности. Успешность прохождения всех этапов позволяет перейти к динамическому выделению IP адрес в соответствии с рассмотренной ранее политикой безопасности, формируемой через менеджера контроля доступа и учета.

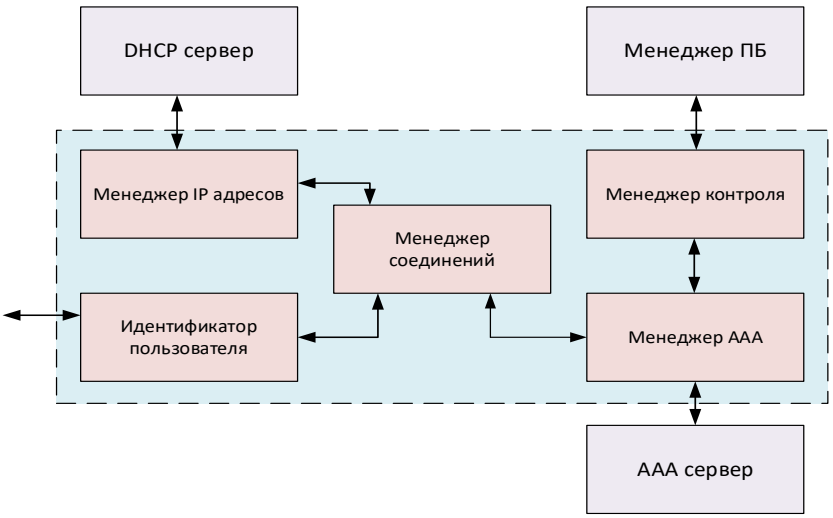


Рис. 7.9

7.8 Протокол DHCP

DHCP (Dynamic Host Configuration Protocol— протокол динамической настройки узла) — прикладной протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

Виды распределения IP адресов:

- автоматическое выделение IP (на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона);
- динамическое выделение из пула адресов;
- ручное выделение (осуществляется администратором, но настройки хранятся централизовано. При ручном выделении IP адреса для проверки выполняется ARP запрос, который осуществляет проверку есть использования IP адреса. Если имеются совпадение, то возникает необходимость переконфигурации выделяемого IP- адреса).

Из пула динамические адреса выделяются по мере необходимости. К особенностям выделения IP адресов можно отнести то, что DHCP-сервер не может самостоятельно отозвать IP адрес.

Алгоритм реализации:

1. Клиент выполняет широковещательный запрос по всей физической сети с целью обнаружить доступные DHCP-серверы (**Ошибка!**

Источник ссылки не найден.10). Он отправляет широковещательное сообщение типа DHCPDISCOVER.

2. DHCP -сервер определяет требуемую конфигурацию клиента в соответствии с указанными сетевым администратором настройками и отправляет ему ответ DHCPOFFER, в котором предлагает конфигурацию.
3. Если в сети находится несколько DHCP -серверов, то клиент может получить несколько различных предложений, обычно выбирается первый из полученных ответов.
4. Выбрав одну из конфигураций, предложенных DHCP-серверами, клиент отправляет конкретный запрос DHCPREQUEST на получение IP.
5. В случае, если адрес свободен сервер либо подтверждает выделение адреса сообщением DHCPACK, либо нет (например, в процессе согласования произошло выделение другому хосту, или пролонгация времени аренды истекло).
6. Для прекращения аренды IP-адреса отправляется сообщение DHCPRELEASE для его освобождения.

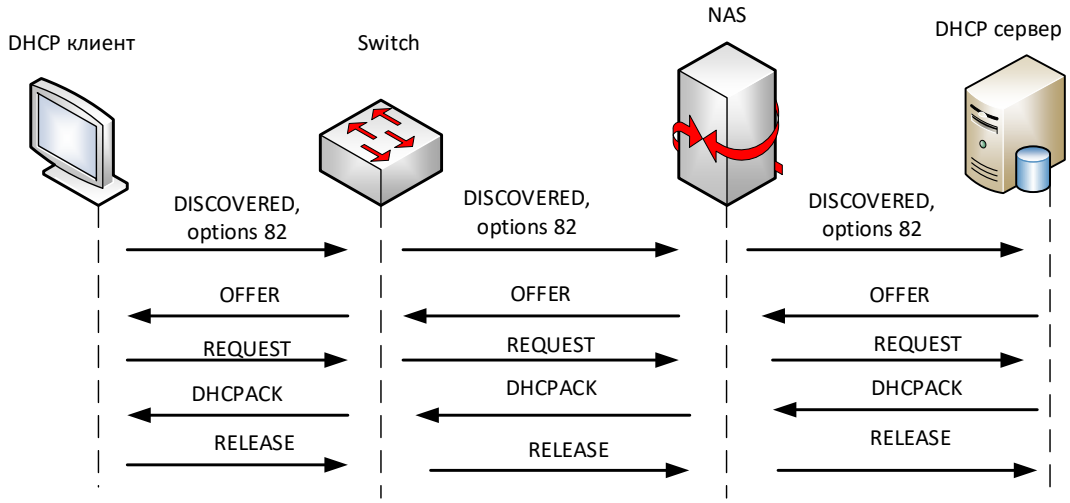


Рис. 7.10

Помимо IP-адреса, DHCP также может сообщать клиенту дополнительные опции, необходимые для нормальной работы в сети. Так для получения адреса может использоваться опция 82, с разными полями, которые включают устройства, стоящие на пути к получению адреса (имя порта, номер платы, номера VLANов). Использование опций позволят понять физически из какого места происходит запрос на получение адреса.

Например, если на определённой порт, который не занят приходит запрос о выделении IP адреса, то это говорит, что к сети подключились незаконно (и данное сообщение отбрасыается).

7.9 Протокол RADIUS

RADIUS (Remote Authentication in Dial-In User Service) — протокол для реализации аутентификации, авторизации и сбора сведений об использованных ресурсах, разработанный для передачи сведений между центральной платформой и оборудованием.

Протокол Radius позволяет хранить в одном месте базу данных пользователей и унифицировать информацию о них. Необходимость использования отдельного сервера аутентификации может возникнуть, потому что количество NAS(BRAS) в сети может быть значительным, и это приводит к тому, что сеть становиться немасштабируемой. Например, если в сети много мобильных пользователей, которые часто аутентифицируются на разных элементах BRAS, целесообразно выделить для этого отдельный сервер AAA. В качестве Radius клиента выступает устройство, которое контролирует доступ к NAS, и которое будет запрашивать у сервера есть ли такой пользователь и правильный пароль он прислал, поскольку сам хост прямого доступа к Radius серверу не имеет (Рис. 7.11 а). В протоколе предусмотрена двухсторонняя аутентификация и гибкий механизм аутентификации PAP либо CHAP. Для передачи сообщений используется протокол UDP, при этом проблемы доставки решаются дублированием UDP пакетов поскольку использование протокола TCP увеличивает длительность аутентификации при значительном количестве пользователей.

Назначение:

- создание и хранение учётных записей пользователей;
- ручная и автоматическая блокировка учётной записи абонента по достижении заданного критерия или лимита;
- сбор и анализ статистической информации о сессиях пользователя и всей обслуживаемой;
- мониторинг;
- создание, печать и отправка счетов к оплате.

Алгоритм реализации:

1. Весь процесс инициируется RADIUS-клиентом. Аутентификация осуществляется по протоколу PAP или CHAP. В этом случае NAS работает как прокси сервер (Рис. 7.11 b).
2. RADIUS-клиент отправляет RADIUS-серверу через NAS запрос доступа AccessRequest.
3. Получив сообщение сервер, в зависимости от используемого протокола аутентификации (PAP или CHAP) формирует один из видов ответа:0

1) Access-Accept — доступ получен, можно начинать использование ресурсов. Пакет, несущий данный ответ, также может содержать дополнительную информацию: IP, выданный пользователю, допустимую продолжительность сессии, максимальный объем передаваемого трафика и т.д.;

2) Access-Challenge — требуется ввод дополнительных данных (PIN, дополнительного пароля). Использование этого ответа позволяет проводить процедуры сложной аутентификации в рамках защищенного сетевого туннеля, установленного напрямую между пользовательской и серверной машинами (для избежания «оседания» данных на сервере доступа);

3) Access-Reject — доступ запрещен из-за неверно указанных пользовательских данных или отсутствия пользователя в базе. Правильность аутентификационных данных может проверяться с помощью разных схем аутентификации: PAP, EAP или CHAP.

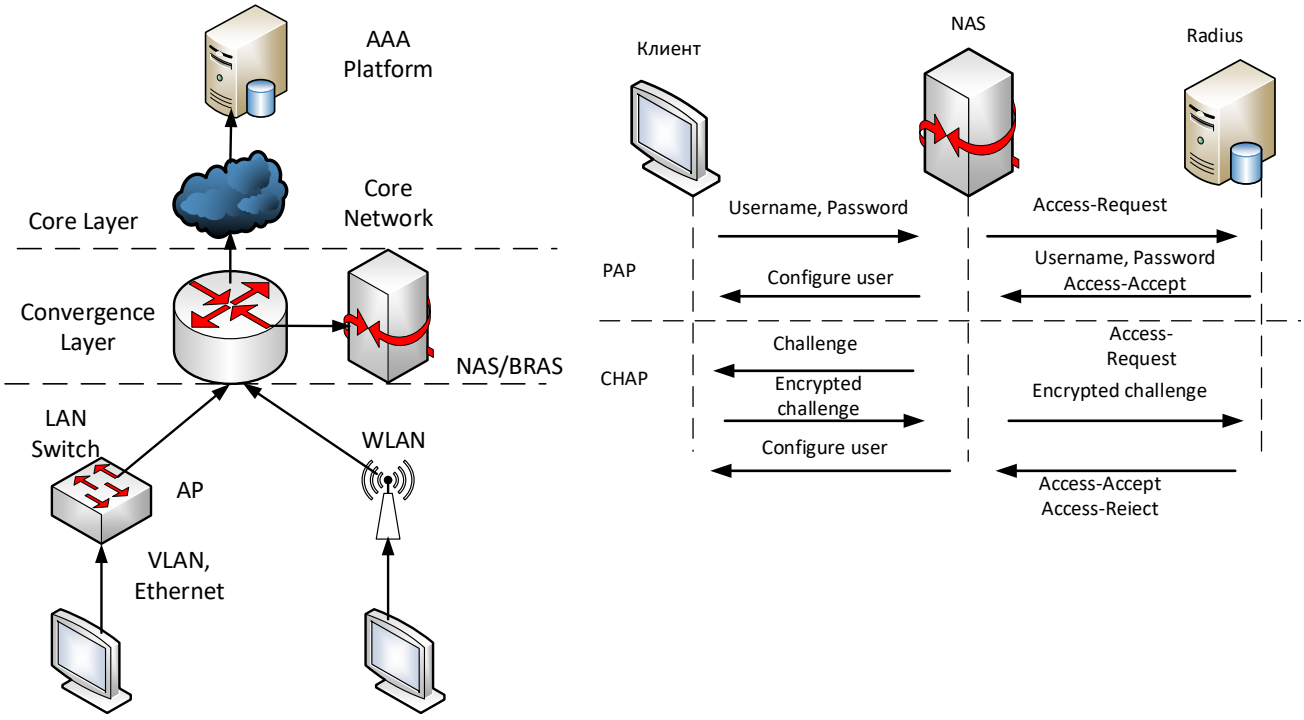


Рис. 7.11

Протокол позволяет осуществлять онлайн-учёт ресурсов абонента, в том числе с уведомлением о начале и конце сессии со стороны обслуживаемой системы. Для этого, после прохождения фазы аутентификации предусмотрены 3 стадии учета (Рис. 7.12):

1. На стадии Start отправка сообщения Accounting Request со стороны клиента инициирует начало учета трафика и времени нахождения клиента в сети, с подтверждением записи в базу данных сообщением Accounting Response. Таких пар сообщений может быть несколько.
2. На стадии Interim производится периодический учет потребляемого ресурса. Если клиент по каким-либо причинам прекратил использование ресурса, учет временно приостанавливается. Например, если пользователь ушел и не закрыл сессию.
3. Сообщение User Request на стадии Stop формирует завершение сеанса, на которое приходить подтверждение о закрытии сессии и остановки учета.

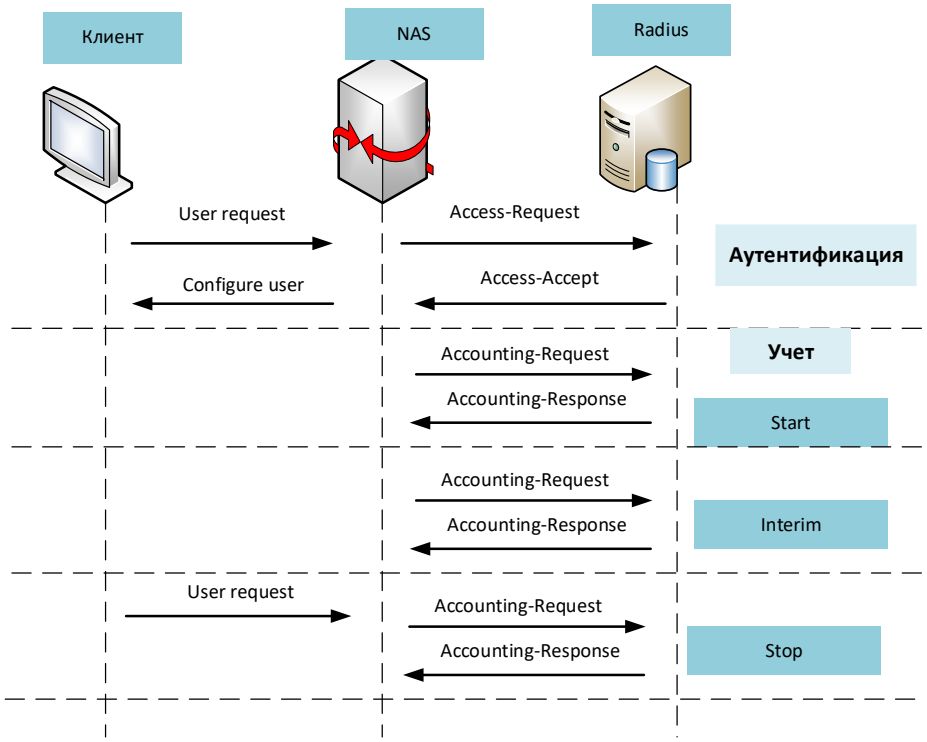


Рис. 7.12

7.10 Протокол Diameter

Протокол Diameter обеспечивает функциональность AAA, но использует TCP вместо UDP. Технология передачи данных может применяться в мобильных сетях, например в роуминге, поскольку обеспечивает для данного вида сетей отказоустойчивость. Для обеспечения безопасности возможно использование протокола IPSec (Рис. 7.13, а). Для использование протокола предусмотрено использование приложений Diameter – стека дополнительных протоколов, на основе базового протокола Diameter, которые реализуют необходимый в данной сети функционал.

Примеры приложений Diameter:

Приложение Diameter Mobile IPv4 (MobileIP, RFC 4004).

Приложение Diameter-сервера доступа к сети (NASREQ, RFC 7155).

Приложение Diameter Credit-Control (DCCA, RFC 8506]).

Алгоритм реализации:

1. Обмен данными между двумя одноранговыми узлами начинается с установления транспортного соединения (TCP или SCTP).
2. Клиент отправляет сообщение Capabilities-Exchange-Request (CER) другому одноранговому узлу, который отвечает ответом Capabilities-Exchange-Answer (CEA).

3. В случае необходимости осуществляется аутентификация сторон с использованием протоколов аутентификации (PAP, CHAP, EAP или TLS).
4. После завершения фазы аутентификации процесс переходит к фазе обмена сообщениями на уровне приложений.

Характеристики:

1. Отказоустойчивость.
2. Использование протоколов IPSec для обеспечения безопасности.
3. Технология построения сети позволяет работать с пользователями, которые перемещаются из одной сети в другую и при этом продолжать использовать один и тот же IP-адрес.
4. Автоматическое обнаружение соседей.

Аутентификация:

- PAP, CHAP, EAP

Авторизация:

- Перенаправление трафика, использование безопасных прокси, ретрансляция (relay), посредничество (broker).
- Согласование состояния.
- Отключение агентов по собственной инициативе.

Учет:

- Отчетность, мониторинг событий.

Diameter имеет одноранговую архитектуру, и каждый хост, реализующий протокол Diameter, может выступать либо клиентом, либо сервером, в зависимости от сетевой инфраструктуры. При установлении соединения между клиентами и серверами в качестве шлюзов передачи данных могут использоваться агенты, коорые являются промежуточными узлами и выполняют функции управления трафиком. Например, агенты могут агрегировать сообщения от устройств на одной площадке, выполнять роль балансировщика нагрузки, модифицировать пакеты, выступать в роли шлюзов безопасности при переходе из доверенной сети в публичную.

Функционально агенты делятся на (Рис. 7.13):

- **Relay agent** для перенаправления сообщения соответствующему адресату в зависимости от информации, содержащейся в сообщении. Объединяет запросы от различных областей (или регионов и устраняет

обременительную настройку серверов при каждом изменении Diameter-сервера.

- **Proxy** может изменить содержимое сообщения и предоставляет дополнительные службы (контроль доступ к определенным сервисам).
- **Redirect agent** выступает в роли централизованного репозитория конфигураций
- **Translation Agents** преобразование сообщения из одного AAA-протокола в другой (преобразование из RADIUS в Diameter).

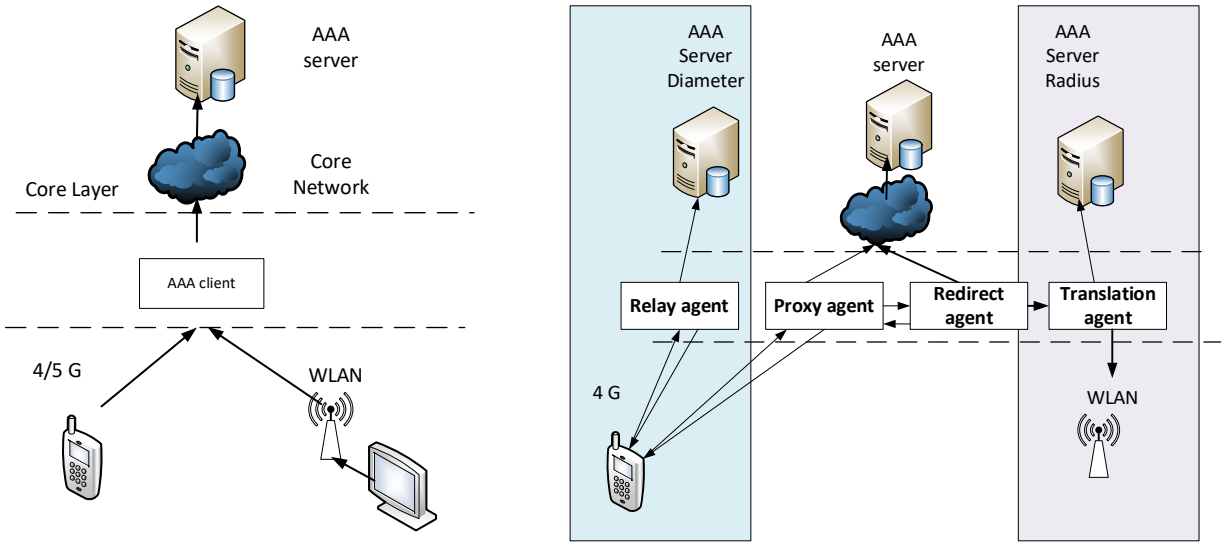


Рис. 7.13

7.11 Протокол Kerberos

Протокол Kerberos — сетевой протокол аутентификации, предлагает механизм взаимной аутентификации посредством которого может быть проведена авторизация без необходимости повторного ввода пароля или предоставления других удостоверяющих данных.

Задачи:

- взаимная аутентификация;
- обмен в незащищенных каналах связи с использованием ассиметричной криптографии.

Содержит два логических компонента:

-сервер аутентификации (CA);

-сервер выдачи билетов (TGS — Ticket Granting Server).

Эти компоненты могут быть единой программой, которая запускается на центре распределения ключей (ЦРК — содержит базу данных логинов/паролей для пользователей и сервисов, использующих Kerberos, Рис. 7.14).

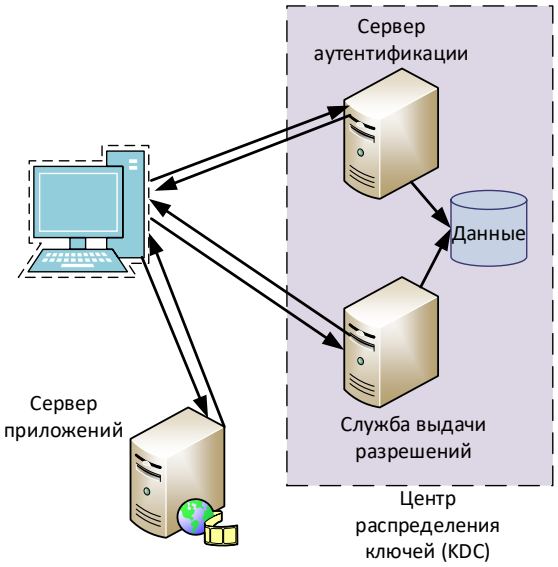


Рис. 7.14

Алгоритм работы:

1. Пользователь с использованием имени и пароля выполняет вход в систему на клиентском хосте (Рис. 7.15).
2. Клиентская машина выполняет над паролем одностороннюю функцию (обычно хэш), и результат становится секретным ключом клиента/пользователя.
3. Клиент отсылает запрос (AS_REQ) на CA для получения аутентификационных верительных данных и последующего их предоставления TGS серверу (впоследствии он будет их использовать для получения билетов без дополнительных запросов на применение секретного ключа пользователя)
4. Если политика ЦРК требует предварительной аутентификации, то пользователь получает сообщение KRB_ERROR, в ответ на которое посылает повторный запрос, но с уже данными для установления подлинности (5).

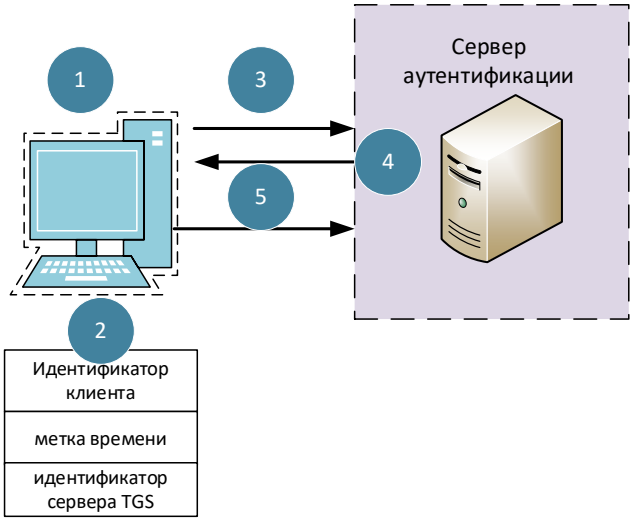


Рис. 7.15

7 Если клиент аутентифицирован CA проверяет, есть ли такой клиент в базе (Рис. 7.16).

8 CA отправляет сообщение (AS_REP).

9 Если же нет, то клиент получает новое сообщение, говорящее о произошедшей ошибке.

10 Получив сообщение, клиент расшифровывает свою часть для получения Сессионного Ключа Клиент/TGS. Этот сессионный ключ используется для дальнейшего обмена с сервером TGS. (Важно: Клиент не может расшифровать TGT, так как оно зашифровано секретным ключом TGS) В этот момент у пользователя достаточно данных, чтобы авторизоваться на TGS.

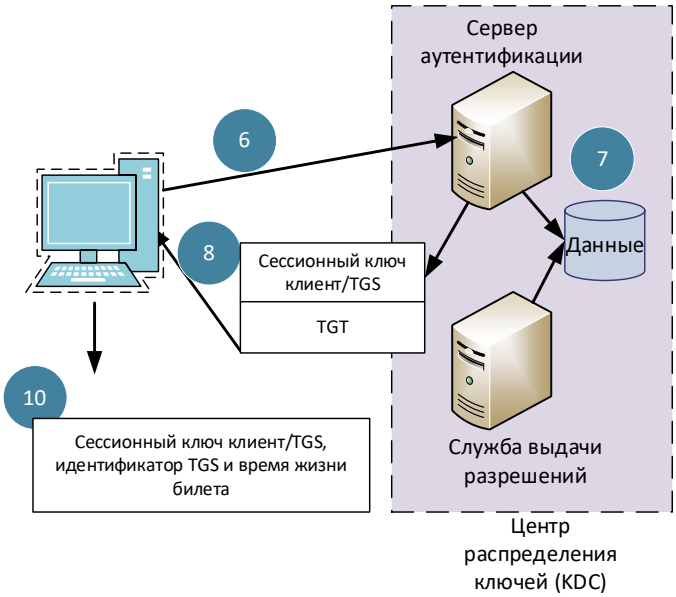


Рис. 7.16

11. Для запроса сервиса клиент формирует запрос на TGS (TGS_REQ, Рис. 7.17).
12. TGS извлекает из него TGT и расшифровывает его, используя секретный ключ TGS. Это дает ему Сессионный Ключ Клиент/TGS. Им он расшифровывает аутентификатор.
13. Затем он генерирует сессионный ключ клиент/сервис и посылает ответ (TGS_REP) включающий:
    - Билет сервиса (который содержит ID клиента, сетевой адрес клиента, метку времени ЦРК, время действия билета и Сессионный Ключ клиент/сервис) зашифрованный секретным ключом сервиса.
    - Сессионный ключ клиент/сервис, идентификатор сервиса и время жизни билета, зашифрованные на Сессионном Ключе Client/TGS.

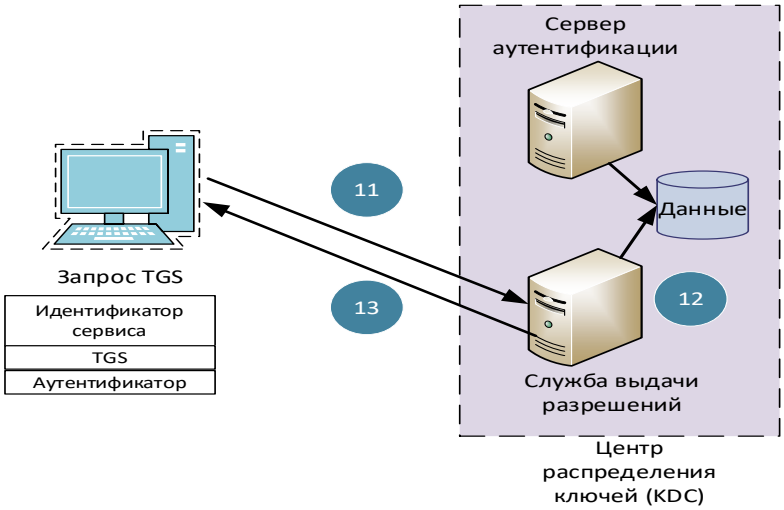


Рис. 7.17

14. После получения TGS_REP посылает сообщение (рис. 7.18), содержащее:
    - зашифрованный билет сервиса, полученный ранее.
    - новый аутентификатор, зашифрованный на сессионном ключе клиент/сервис, и включающий ID клиента и метку времени.
15. Сервер расшифровывает билет используя свой секретный ключ и получает сессионный ключ клиента. Используя новый ключ, он расшифровывает аутентификатор и посылает клиенту:

-метку времени, указанную клиентом + 1, зашифрованную на сессионном ключе клиент/сервис;

16. Клиент проверяет, действительно ли метка времени корректно обновлена. Если это так, то клиент может доверять серверу и может начать посылать запросы на сервер.
17. Сервер предоставляет клиенту требуемый сервис.

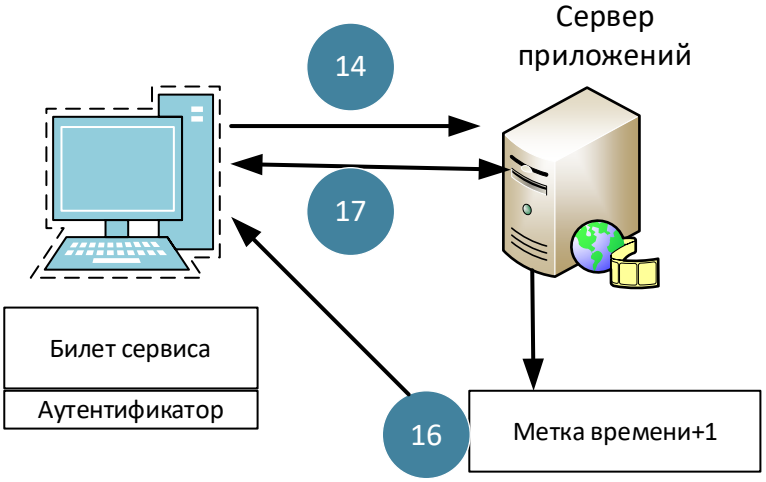


Рис. 7.18

СПИСОК ЛИТЕРАТУРЫ

1. Протоколы идентификации и аутентификации. Электронный ресурс https://bit.nmu.org.ua/ua/student/metod/cryptology/лекция22.pdf.
2. Стандарт Международной организации по стандартизации и Международной электротехнической комиссии ISO / IEC 9798 - Information technology - Security techniques - Entity authentication mechanisms, состоящий из пяти частей:
3. ISO / IEC 9798-1 - «General Model»;
4. ISO / EEC 9798-2 - «Mechanisms using symmetric encipherment algorithms»;
5. ISO / EEC 9798-3 - «Entity authentication using a public-key algorithm»;
6. ISO / IEC 9798-4 - «Mechanisms using a cryptographic check function»;
7. ISO / IEC 9798-5 - «Mechanisms using zero knowledge techniques».