А.М. Голиков

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Учебное пособие

для специалитета: 10.05.02 - Информационная безопасность телекоммуникационных систем (Безопасность телекоммуникационных систем информационного взаимодействия)

Курс лекций, компьютерный практикум, задание на самостоятельную работу

Томск

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Томский государственный университет систем управления и радиоэлектроники

А.М. ГОЛИКОВ

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Учебное пособие

для специалитета: 10.05.02 - Информационная безопасность телекоммуникационных систем (Безопасность телекоммуникационных систем информационного взаимодействия)

Курс лекций, компьютерный практикум, задание на самостоятельную работу

2016

УДК 621.39(075.8)

ББК 32.973(я73)

Г 60

Голиков А.М.

Криптографические методы защиты информации. Учебное пособие для специалитета: 10.05.02 Информационная безопасность телекоммуникационных систем. Курс лекций, компьютерный практикум, задание на самостоятельную работу / А.М.Голиков. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2016. – 97 с.: ил. — (Учебная литература для вузов)

Учебное пособие предназначено для подготовки специалистов по направлению 10.05.02 Информационная безопасность телекоммуникационных систем. Представляет собой изложение криптографических методов защиты информации. Учебное пособие содержит курс лекций, компьютерный практикум и задания на самостоятельную работу. Рассмотрены криптографические протоколы в сетях передачи данных и вопросы защиты информации в современных системах связи.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....5

1. КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ 6

1.1. Теория шифров с открытым ключом криптографические протоколы в сетях передачи данных.....6

1.2. Компьютерный практикум для шифров с открытым ключом.....44

1.3. Задания на самостоятельную работу по криптографическим протоколам в сетях передачи данных.....55

2. МЕТОДЫ ШИФРОВАНИЯ В СОВРЕМЕННЫХ СИСТЕМАХ СВЯЗИ 73

2.1. Безопасность GSM сетей.....73

2.2. Криптографическая защита беспроводных сетей стандартов LTE.....76

ЗАКЛЮЧЕНИЕ.....96

ЛИТЕРАТУРА.....97

ВВЕДЕНИЕ

Разные люди понимают под шифрованием разные вещи. Дети играют в игрушечные шифры и секретные языки. Это, однако, не имеет ничего общего с настоящей криптографией. Настоящая криптография (strong cryptography) должна обеспечивать такой уровень секретности, чтобы вы имели возможность надежно защитить критическую информацию от расшифровки крупными организациями — такими как мафия, транснациональные корпорации и крупные государства. Настоящая криптография в прошлом использовалась лишь в военных целях. Однако сейчас, со становлением информационного общества, она становится центральным инструментом для обеспечения конфиденциальности.

По мере образования информационного общества, крупным государствам становятся доступны Технологические средства тотального надзора за миллионами людей. Поэтому криптография становится одним из основных инструментов обеспечивающих конфиденциальность, доверие, авторизацию, электронные платежи, корпоративную безопасность и бесчисленное множество других важных вещей.

Криптография не является более придумкой военных, с которой не стоит связываться. Настала пора снять с криптографии покровы таинственности и использовать все ее возможности на пользу современному обществу. Широкое распространение криптографии является одним из немногих способов защитить человека от ситуации, когда он вдруг обнаруживает, что живет в тоталитарном государстве, которое может контролировать каждый его шаг.

1. КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ

1.1. Теория шифров с открытым ключом. Криптографические протоколы в сетях передачи данных

Безопасность сети передачи данных на транспортном уровне SSL и TLS [1-8]

Безопасность транспортного уровня обеспечивает услуги безопасности "из конца в конец" для приложений, которые используют протоколы транспортного уровня, такие как TCP. Основные идеи предназначены для того, чтобы обеспечить услуги безопасности на сети Интернет. Например, когда в сети имеются интерактивно работающие онлайн(online)-магазины, то желательны следующие услуги безопасности:

1. Клиент должен убедиться, что сервер принадлежит фактическому продавцу, а не самозванцу. Клиент не хочет сообщать самозванцу номер кредитной карточки (установление подлинности объекта).

2. Клиент и продавец должны быть убеждены, что содержание сообщения не изменено в течение передачи (целостность сообщения).

3. Клиент и продавец должны быть убеждены, что самозванец не перехватит чувствительную информацию, такую как номер кредитной карточки (конфиденциальность).

Сегодня применяются в основном два протокола обеспечения безопасности на транспортном уровне: *Протокол "Уровень безопасных розеток" (SSL - Secure Socket*

Layer) и *Протокол Безопасности Транспортного уровня (TLS - Transport Layer Security)*.

Мы сначала обсудим SSL, затем TLS, а потом их сравним и покажем их отличия друг от друга.

Одна из целей этих протоколов состоит в том, чтобы обеспечить сервер и клиента

услугами установления подлинности, конфиденциальности и целостности данных. Прикладной уровень программ клиент-сервер (client-server), таких как Язык передачи гипертекста (HTTP), который использует услуги TCP, может инкапсулировать свои

данные в пакеты SSL. Если сервер и клиент согласованы с функционирующими

программами SSL (или TLS), то клиент может использовать URL https: // ... вместо

http:// ... , для того чтобы разрешить сообщениям HTTP инкапсулироваться в пакеты

SSL (или TLS). Например, номера кредитной карточки могут быть безопасно переданы

через Интернет для онлайн-покупателей.

Протокол SSL

SSL (англ. secure sockets layer — уровень защищённых сокетов) — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений. Протокол широко использовался для обмена мгновенными сообщениями и передачи голоса через IP (англ. Voice over IP — VoIP), в таких приложениях, как электронная почта, Интернет-факс и др. В настоящее

время известно, что протокол не является безопасным. SSL должен быть исключен из работы в пользу TLS (см. CVE-2014-3566).

SSL изначально разработан компанией Netscape Communications для добавления протокола HTTPS в свой веб-браузер Netscape Navigator. Впоследствии, на основании протокола SSL 3.0 был разработан и принят стандарт RFC, получивший имя TLS.

Протокол SSL обеспечивает защищенный обмен данных за счет двух следующих элементов:

- Аутентификация
- Шифрование

SSL использует асимметричную криптографию для аутентификации ключей обмена, симметричный шифр для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

Протокол SSL предоставляет "безопасный канал", который имеет три основных свойства:

1. Канал является частным. Шифрование используется для всех сообщений после простого диалога, который служит для определения секретного ключа.

2. Канал аутентифицирован. Серверная сторона диалога всегда аутентифицируется, а клиентская делает это опционно.

3. Канал надежен. Транспортировка сообщений включает в себя проверку целостности.

Преимуществом SSL является то, что он независим от прикладного протокола. Протоколы приложений (HTTP, FTP, TELNET и т.д.) могут работать поверх протокола SSL совершенно прозрачно, т е SSL может согласовывать алгоритм шифрования и ключ сессии, а также аутентифицировать сервер до того, как приложение примет или передаст первый байт сообщения.

Принцип работы

SSL использует среду с несколькими слоями, что обеспечивает безопасность обмена информацией. Конфиденциальность общения присутствует за счет того, что безопасное соединение открыто только целевым пользователям.

Многослойная среда

Протокол SSL размещается между двумя протоколами: протоколом, который использует программа-клиент (HTTP, FTP, LDAP, TELNET etc) и транспортным протоколом TCP/IP. SSL защищает данные выступая в роли фильтра для обеих сторон и передает их далее на транспортный уровень. Работу протокола можно разделить на два уровня:

1. Слой протокола подтверждения подключения (Handshake Protocol Layer)
2. Слой протокола записи

Первый слой, в свою очередь, состоит из трех подпротоколов:

1. Протокол подтверждения подключения (Handshake Protocol)
2. Протокол изменения параметров шифра (Cipher Spec Protocol)
3. Предупредительный протокол (Alert Protocol)

Протокол подтверждения подключения используется для согласования данных сессии между клиентом и сервером. К данным сессии относятся:

- Идентификационный номер сессии
- Сертификаты обеих сторон
- Параметры алгоритма шифрования
- Алгоритм сжатия информации
- "Общий секрет" применен для создания ключей; открытый ключ

Протокол подтверждения подключения производит цепочку обмена данными, что в свою очередь начинает аутентификацию сторон и согласовывает шифрование, хэширование и сжатие. Следующий

этап - аутентификация участников, которая осуществляется также протоколом подтверждения подключения.

Протокол изменения параметров шифра используется для изменения данных ключа (keyingmaterial) - информации, которая используется для создания ключей шифрования. Протокол состоит всего из одного сообщения, в котором сервер говорит, что отправитель хочет изменить набор ключей.

Предупредительный протокол содержит сообщение, которое показывает сторонам изменение статуса или сообщает о возможной ошибке. Обычно предупреждение отсылается тогда, когда подключение закрыто и получено неправильное сообщение, сообщение невозможно расшифровать или пользователь отменяет операцию.

Цифровые сертификаты

Протокол SSL использует сертификаты для проверки соединения. Сертификаты расположены на безопасном сервере и используются для шифрования данных и идентификации Web-сайта. Способы получения SSL-сертификата:

1. Использовать сертификат, выданный CA
2. Использовать самоподписанный сертификат
3. Использовать "пустой" сертификат

Самоподписанный сертификат - сертификат, созданный самим пользователем - в этом случае издатель сертификата совпадает с владельцем сертификата. “Пустой” сертификат - сертификат, содержащий фиктивную информацию, используемую в качестве временной для настройки SSL и проверки его функциональности в данной среде.

Механизмы образования ключа для текущей сессии в SSL/TLS

Для обмена подлинными и конфиденциальными сообщениями клиенту и серверу нужны шесть криптографических объектов секретности (четыре ключа и два вектора инициализации). Однако чтобы создать их, между этими

двумя сторонами должен быть установлен один предварительный главный секретный код (pre-master secret). SSL определяет шесть методов обмена ключами, чтобы установить этот предварительный объект секретности: NULL, RSA, анонимный Диффи-Хеллман (Diffie-Hellman), кратковременный Диффи-Хеллман, фиксированный Диффи-Хеллман и Fortezza

RSA

В этом методе предварительный главный секретный код - 48-байтовое случайное

число, созданное клиентом, зашифрованное открытым ключом RSA-сервера и передаваемое серверу. Сервер должен передать свой сертификат шифрования/дешифрования RSA.

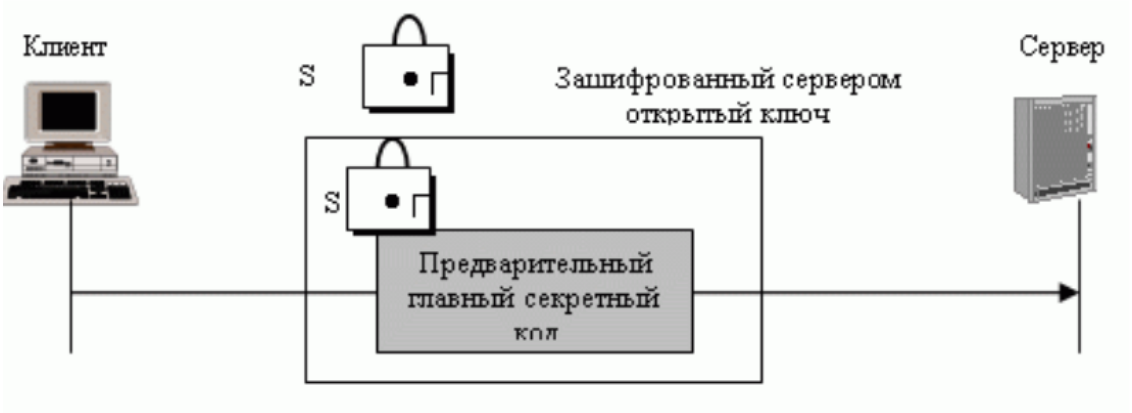


Рис. 1.1 – RSA – смена ключа; открытый ключ сервера

Анонимный протокол Диффи-Хеллмана

Это самый простой и наи**б**олее ненадежный метод. Предварительный главный секретный код устанавливают между клиентом и сервером, используя протокол Диффи-Хеллмана. При этом передают половину ключа в исходном тексте - это называется анонимным протоколом Диффи-Хеллмана, потому что ни одна сторона не известна другой. Самый серьезный недостаток этого метода - возможность атаки "посредника".

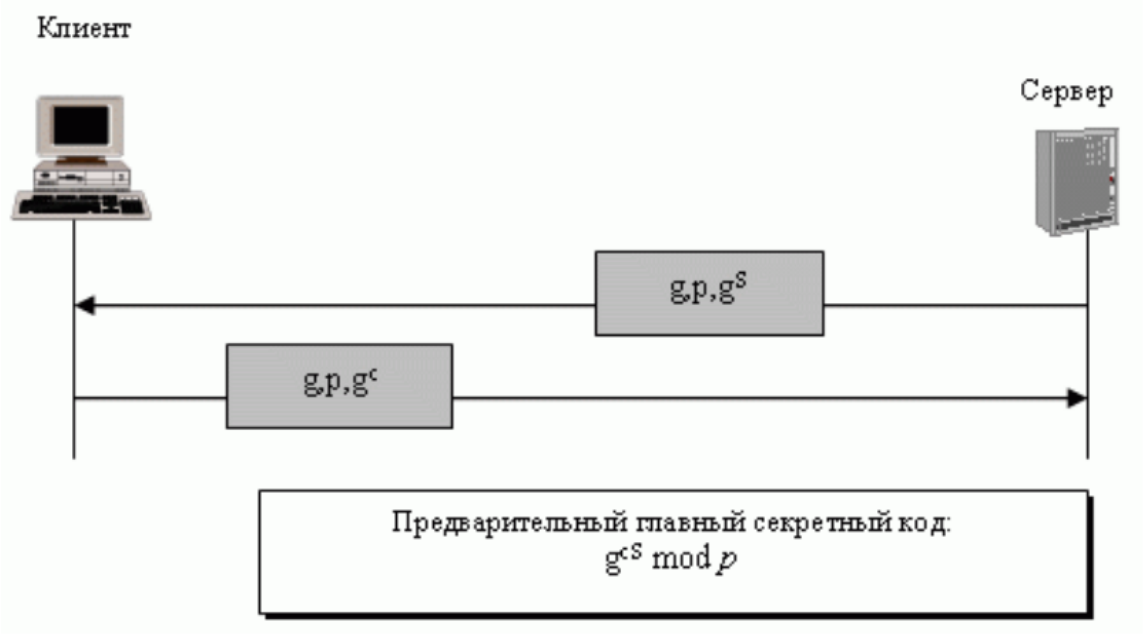


Рис. 1.2. Анонимный протокол Диффи-Хеллмана смены ключей

Кратковременный метод Диффи-Хеллмана

Чтобы сорвать атаку "посредника", может быть использована кратковременная смена ключей методом Диффи-Хеллмана. Каждая сторона передает ключ Диффи-Хеллмана, подписанный своим секретным ключом. На приемной стороне должны проверить подпись, используя открытый ключ передатчика. Обмен открытыми ключами для проверки использует либо RSA-, либо DSS-сертификат цифровой подписи.

Рис. 1.3. Кратковременный протокол Диффи-Хеллмана смены ключей

Фиксированный метод Диффи-Хеллмана

Другое решение - фиксированный метод Диффи-Хеллмана. Все объекты в группе могут подготовить фиксированные параметры (g и p). Затем каждый объект может создать фиксированную половину ключа (gx). Для дополнительной безопасности каждая отдельная половина ключа Диффи-Хеллмана вставляется в сертификат, проверенный центром сертификации (CA). Другими словами, две стороны отдельно не обмениваются полуключами; CA передает полуключи в специальном сертификате RSA или DSS. Когда клиент должен вычислить *предварительный главный секретный код*, он использует свой собственный фиксированный полуключ и полуключ сервера, полученный в сертификате. Сервер делает то же самое, но в обратном порядке. Обратите внимание, что в этом методе не передаются сообщения смены ключей, а происходит только обмен сертификатами.

Алгоритмы шифрования/дешифрования

Есть несколько возможностей выбора алгоритма шифрования/дешифрования. Мы можем разделить алгоритмы на 6 групп, как это показано на рис. 1.4. Все протоколы блока используют 8-байтовый вектор инициализации (IV), кроме Fortezza, который применяет 20 байтов IV.

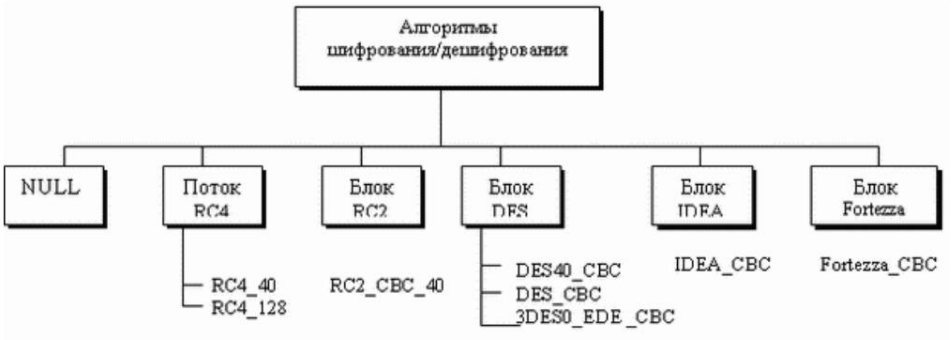


Рис. 1.4. Алгоритмы шифрования/дешифрования

NULL

NULL - категория, которая просто определяет отсутствие алгоритма шифрации/дешифрации.

Поток RC

В режиме потока RC определены два потока алгоритма: RC4-40 (ключ на 40 битов) и

RC4-128 (ключ на 128 битов).

Блок RS

В режиме блока RC определен один алгоритм: RC2_CBC_40 (ключ на 40 битов). CBC (Cipher Block Chaining) - сцепление шифрованных блоков.

DES

Все алгоритмы DES определены в режиме блока. DES40_CBC использует ключ на 40 битов. Стандартные DES определены как DES_CBC. 3DES_EDE_CBC используют ключ на 168 битов.

IDEA

В режиме блока IDEA определен один алгоритм - IDEA_CBC, с ключом на 128 битов.

Fortezza

В режиме блока Fortezza определен один алгоритм - FORTEZZA_CBC, с ключом на 96 бит.

2.1.5 Алгоритмы хэширования

SSL использует алгоритмы хэширования, чтобы обеспечить целостность сообщения

(установление подлинности сообщения). Имеются хэш-функции, показанные на рис. 1.5.

Рис. 1.5. Алгоритмы хэширования

Null (Пустой указатель)

Две стороны могут отказаться использовать алгоритм хэширования. В этом случае

сообщение не заверено.

MD5

Две стороны могут выбрать MD5 как алгоритм хэширования. В этом случае используется алгоритм хэширования MD5 - 128-битовый.

SHA-1

Две стороны могут выбрать SHA как алгоритм хэширования. В этом случае используется алгоритм хэширования SHA-1 на 160 битов.

Алгоритмы сжатия

Как мы уже говорили, сжатие является дополнительной услугой в SSLv3. Для SSLv3 не определен алгоритм сжатия. Поэтому заданным по умолчанию методом сжатия служит NULL. Однако система может использовать любой алгоритм сжатия по выбору сторон.

Генерирование криптографических параметров

Чтобы обеспечить целостность и конфиденциальность сообщения, в SSL необходимо иметь: шесть криптографических объектов секретности, четыре ключа и два инициализирующих вектора (IV). Клиенту нужно: один ключ для передачи сообщения установления подлинности (HMAC - HASH-BASED MESSAGE AUTHENTICATION CODE), один ключ для шифрования и один

IV для шифрования блока. Сервер нуждается в том же самом. SSL требует, чтобы ключи для одного направления отличались от ключей для другого направления. Если будет атака в одном направлении, она не затронет другое направление. Для генерации параметров используют следующую процедуру:

1. Клиент и сервер обмениваются двумя случайными числами, одно из которых создано клиентом, а другое - сервером.

2. Клиент и сервер обмениваются одним предварительным главным секретным кодом.

3. Создается 48-байтовый главный секретный код (master secret) из предварительного главного секретного кода (pre-master secret), с применением хэш-функций (SHA-1 и MD5), как это показано на рис. 1.6.

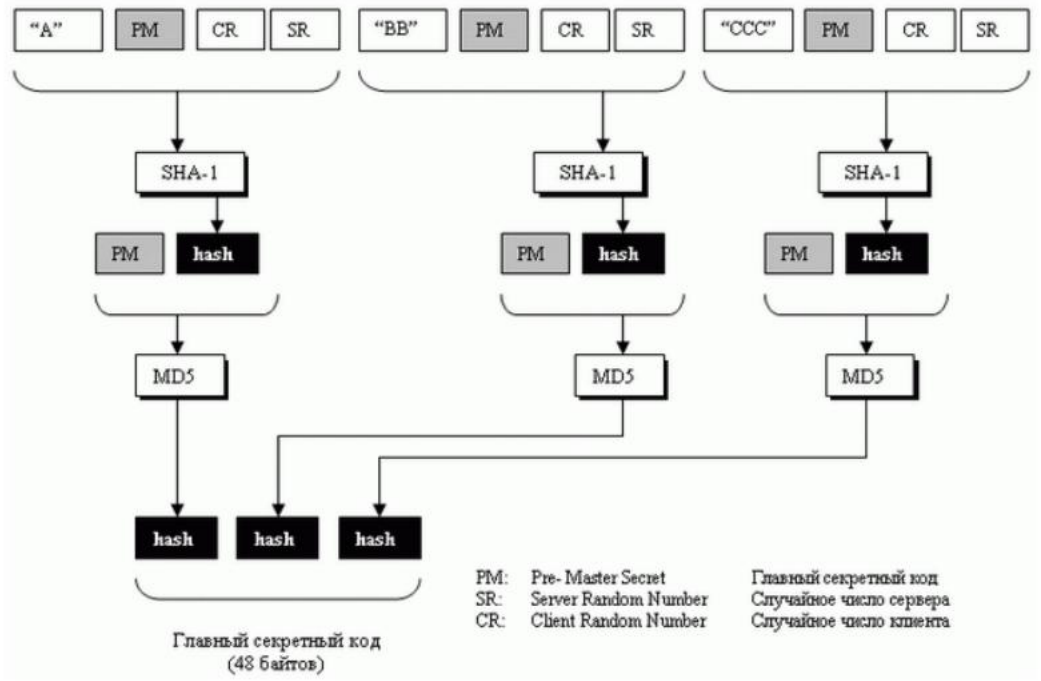


Рис. 1.6. Вычисление главного секретного кода из предварительного главного секретного кода

4. Главный секретный код используется для того, чтобы создать материал для ключей (key material), который имеет переменную длину. Для этого применяют то же самое множество хэш-функций, что и в предыдущем случае, и подставляют спереди различные константы, как это показано на

рис. 1.7. Алгоритм повторяется, пока не получится материал для ключа адекватного размера.

Длина блока материала для ключей зависит от выбранного набора шифра и размера ключей, необходимых для этого набора.

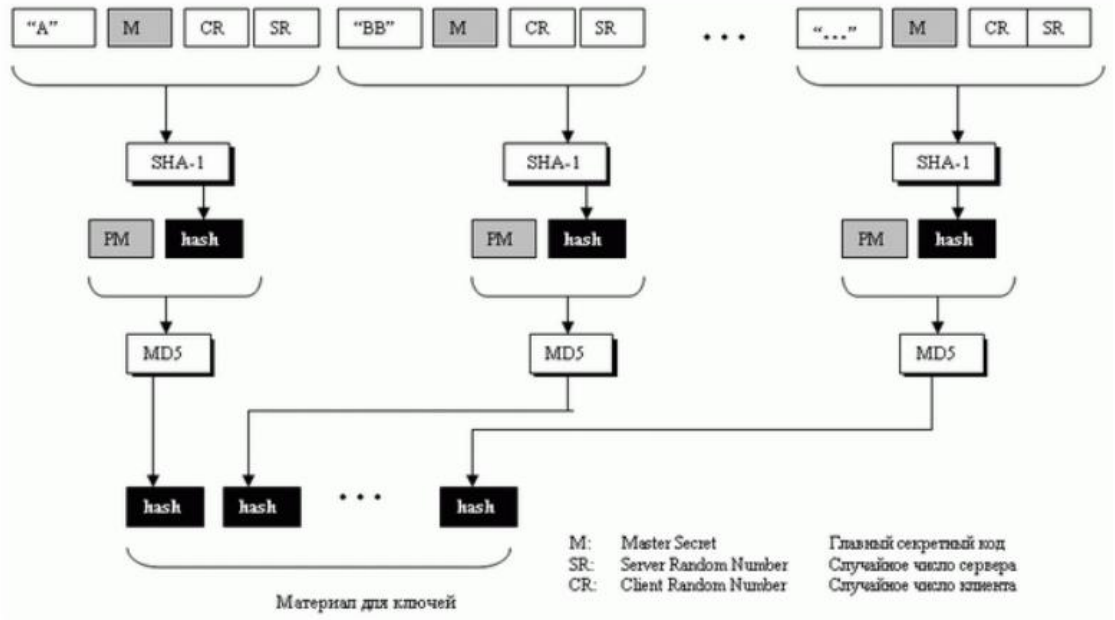


Рис. 1.7. Вычисление материала для ключей из главного секретного кода

5. Из материала для ключей извлекаются шесть различных ключей, как показано на рис. 1.8.

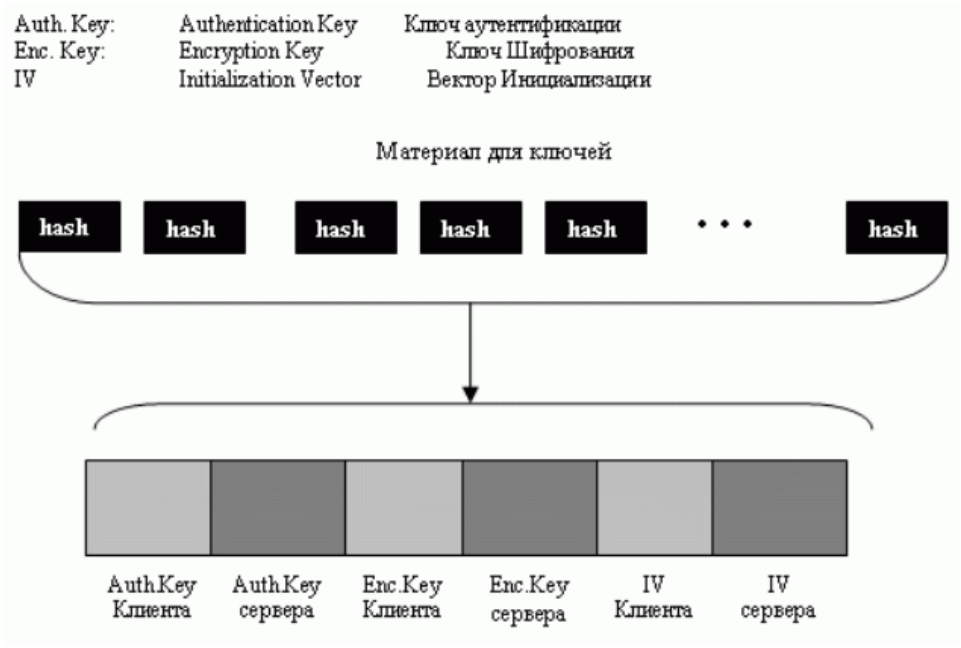


Рис. 1.8. Извлечение криптографических секретных кодов из материала

Сеансы и соединение

SSL отличает соединение от сеанса. Давайте рассмотрим эти два термина. Сеанс – связь между клиентом и сервером. После того как сеанс установлен, эти две стороны имеют общую информацию, такую как идентификатор сеанса, сертификат, подтверждающий подлинность каждого из них (в случае необходимости), метод сжатия(если необходимо), набор шифров и главный секретный код. Эта информация используется для того, чтобы создать ключи для сообщения, содержащего шифр установления подлинности.

Для двух объектов, чтобы начать обмен данными, установление сеанса необходимо, но не достаточно; они должны создать между собой соединение. Эти два объекта обмениваются двумя случайными числами и создают, используя главный секретный код, ключи и параметры, необходимые для того, чтобы обмениваться сообщениями, включая установление подлинности и секретность.

Сеанс может состоять из многих соединений. Соединение между двумя сторонами может быть закончено и восстановлено в пределах одного и того же сеанса. Когда соединение закончено, эти две стороны могут также закончить сеанс, но это необязательно. Сеанс может быть приостановлен и продолжен снова.

Чтобы создавать новый сеанс, эти две стороны должны пройти процесс переговоров. Чтобы возобновлять старый сеанс и создавать только новое соединение, эти две стороны могут пропустить часть переговоров, что уменьшает время вхождения в связь. Не надо создавать главный секретный код, когда сеанс продолжается.

Разделение сеанса от соединения предотвращает высокую стоимость создания главного секретного кода. Если мы разрешаем приостановления и продолжения сеанса, процесс вычисления главного секретного кода может быть устранен. рис. 1.9 иллюстрирует идею сеанса и соединения в этом сеансе.

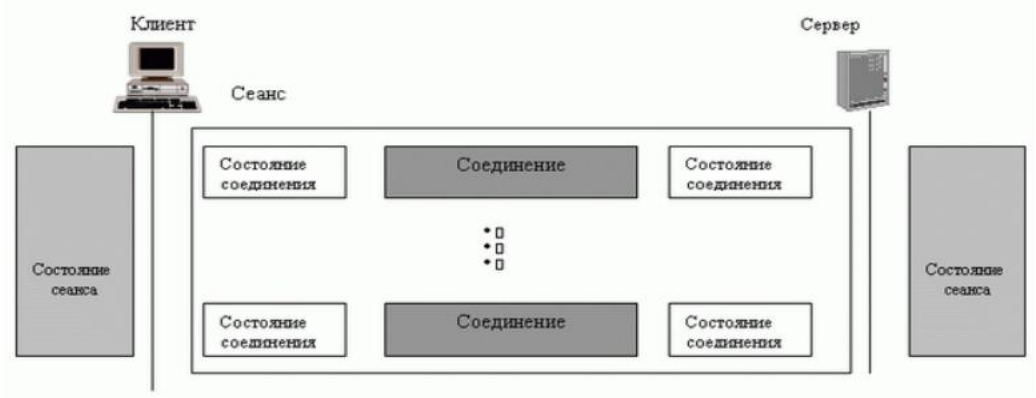


Рис. 1.9. Сеанс и соединение

В сеансе одна сторона играет роль клиента и другая - роль сервера. При соединении обе стороны имеют равные роли, они равны по уровню.

Четыре протокола

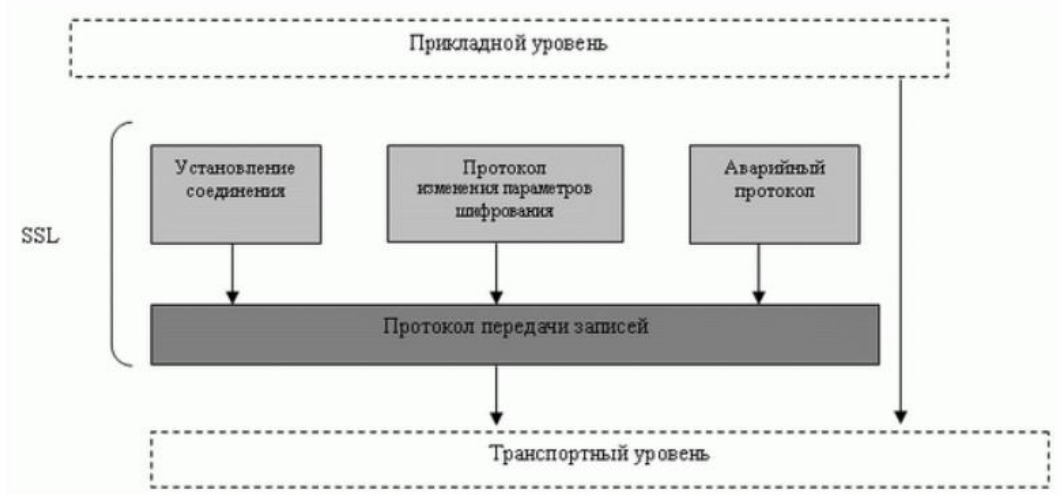


Рис. 1.10. Четыре протокола SSL

Протокол передачи записей - переносящий информацию. Он переносит на транспортный уровень сообщения от трех других протоколов, а также данные, поступающие от прикладного уровня. Сообщения из протокола записей - это полезная нагрузка для транспортного уровня, обычно TCP. Протокол установления соединения обеспечивает параметры безопасности для Протокола записей. Он устанавливает набор шифров и задает ключи и параметры безопасности.

Он также подтверждает, если необходимо, подлинность сервера клиенту и подлинность клиента серверу. Протокол изменения параметров шифрования

используется, чтобы передавать сигналы для подготовки к криптографической безопасности. Аварийный протокол нужен, чтобы известить о ситуациях, отклоняющихся от нормы.

Протокол TLS

Безопасность транспортного уровня (TLS - Transport Layer Security) - протокол IEFT, стандартная версия протокола SSL. Эти два протокола очень похожи, но имеют небольшие отличия. Вместо того чтобы описывать TLS полностью, в этой секции мы только отметим отличия между протоколами TLS и SSL.

Генерация криптографической секретности

Генерация криптографической секретности в TLS более сложная, чем в SSL. TLS сначала определяет две функции: функцию расширения данных и псевдослучайную функцию.

Функция расширения данных

Функция расширения данных использует заранее заданный код аутентификации на основе хэширования (HMAC-HASH-BASED MESSAGE AUTHENTICATION CODE), или MD5, или SHA-1 для того, чтобы расширить информацию засекречивания. Эту функцию можно рассматривать как функцию, содержащую множество секций, где каждая секция

создает одно значение хэширования. Расширенная секретность – последовательное соединение значений хэширования. Каждая секция использует два HMAC, информацию засекречивания и начальное число. Функция расширения данных - это формирование цепочки в виде многих секций. Однако чтобы сделать следующую секцию зависимой от предыдущей, второе начальное число – фактически выход первого HMAC предыдущей секции, как это показано на рис. 1.11.

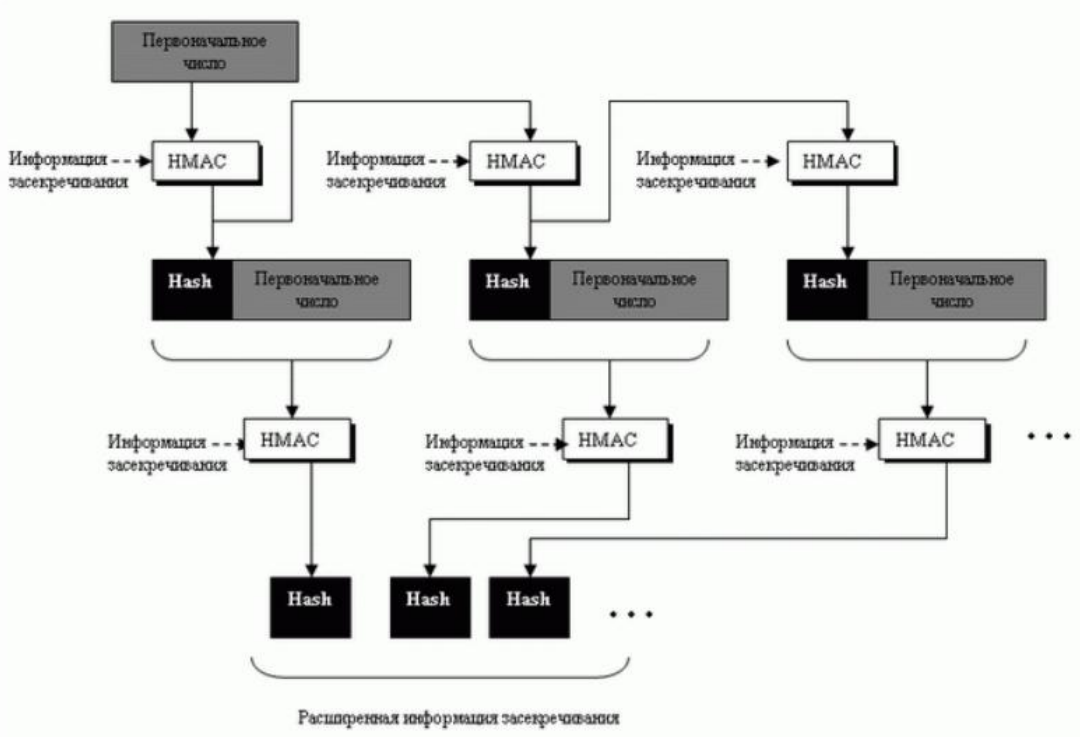


Рис. 1.11. Функция расширения данных

Псевдослучайная функция

TLS определяет псевдослучайную функцию (PRF - PseudoRandom Function), чтобы получить комбинацию двух функций расширения данных: одна из них использует MD5 и другая - SHA-1. На PRF поступает три части информации: секретный код, метка и начальное число.

Метка и начальное число связаны и служат начальным числом для каждой функции расширения данных. Информация засекречивания разделена на две части; каждая часть используется как информация засекречивания для каждой функции расширения данных. Выходы двух функций расширения данных складывают по модулю два, чтобы создать конечную расширенную информацию засекречивания. Обратите внимание, что поскольку хэш создается MD5 и SHA-1, он имеет различные размеры, поэтому должны быть созданы дополнительные секции функций на базе MD5, чтобы сделать два вывода с одинаковым размером. рис. 4.12 показывает идею применения PRF.

Рис. 1.12. PRF

Главный секретный код

TLS использует функцию PRF, чтобы создать главный секретный код от предварительного главного секретного кода. Это можно сделать, используя предварительный главный секретный код как информацию засекречивания, строку "главный секретный код" - как метку и последовательное соединение информации (конкатенацию) случайного числа клиента и случайное число сервера - как начальное число. Обратите внимание, что метка - фактически код ASCII строки "главного секретного кода". Другими словами, метка определяет выход для создания главного секретного кода. рис. 1.13 иллюстрирует идею.

Рис. 1.13. Генерация главного секретного ключа

Материал для ключей

TLS использует функцию PRF, чтобы создать материал для ключей от главного секретного кода. На сей раз информация засекречивания содержит: главный секретный код; метку - это строка "расширение ключа"; и начальное число – конкатенацию случайного числа сервера и случайного числа клиента, как это показано на рис. 1.14.

Рис. 1.14. Генерация материала для ключа

Протоколы

Аварийный протокол в TLS поддерживает все аварийные сигналы, определенные в SSL за исключением NoCertificate. TLS также добавляет к списку SSL некоторые новые.

Протокол установления соединения TLS вносит некоторые изменения в протокол установления соединения. Были специально изменены детали сообщения CertificateVerify и сообщения Finished.

Единственное изменение в *протоколе передачи записей* – использование HMAC, формируемы с помощью MAC, чтобы подписать сообщение.

4.2. Безопасность сети ПД на сетевом уровне IP SEC

IPSec является неотъемлемой частью IPv6 - Интернет-протокола следующего поколения, и расширением существующие версии Интернет-протокола IPv4. IPSec определён в RFC с 2401 по 2412.

Практически все механизмы сетевой безопасности могут быть реализованы на третьем уровне эталонной модели ISO/OSI в соответствии с рисунком 1.15. Кроме того, IP-уровень можно считать оптимальным для размещения защитных средств, поскольку при этом достигается удачный компромисс между защищенностью, эффективностью функционирования и прозрачностью для приложений.

Уровни TCP/IP	Уровни ISO/OSI
4. Прикладных программ	7. Прикладных программ 6. Представление данных
3. Транспортный	5. Сеансовый 4. Транспортный
2. Межсетевой	3. Сетевой
1. Доступа к сети	2. Канальный 1. Физический

Рис. 1.15. Модель OSI/ISO

Стандартизованными механизмами IP-безопасности могут (и должны) пользоваться протоколы более высоких уровней и, в частности, управляющие протоколы, протоколы конфигурирования и маршрутизации.

Средства безопасности для IP описываются семейством спецификаций IPSec, разработанных рабочей группой IP Security.

Протоколы IPSec обеспечивают управление доступом, целостность вне соединения, аутентификацию источника данных, защиту от воспроизведения, конфиденциальность и частичную защиту от анализа трафика.

Основополагающими понятиями IPSec являются:

- - аутентификационный заголовок (AH);
- - безопасное сокрытие данных (ESP);
- - режимы работы: туннельный и транспортный;
- - контексты (ассоциации) безопасности (SA);
- - управление ключами (IKE);

Основные составляющие архитектуры и их особенности

Архитектура средств безопасности для IP-уровня специфицирована в документе Security Architecture for the Internet Protocol. Ее основные составляющие представлены в соответствии с рисунком 1.2. Это, прежде всего протоколы обеспечения аутентичности (протокол аутентифицирующего заголовка - Authentication Header, AH) и конфиденциальности (протокол инкапсулирующей защиты содержимого - Encapsulating Security payload, ESP), а также механизмы управления криптографическими ключами. На более низком архитектурном уровне располагаются конкретные алгоритмы шифрования, контроля целостности и аутентичности. Наконец, роль фундамента выполняет так называемый домен интерпретации (Domain of Interpretation, DOI), являющийся, по сути, базой данных, хранящей сведения об алгоритмах, их параметрах, протокольных идентификаторах.

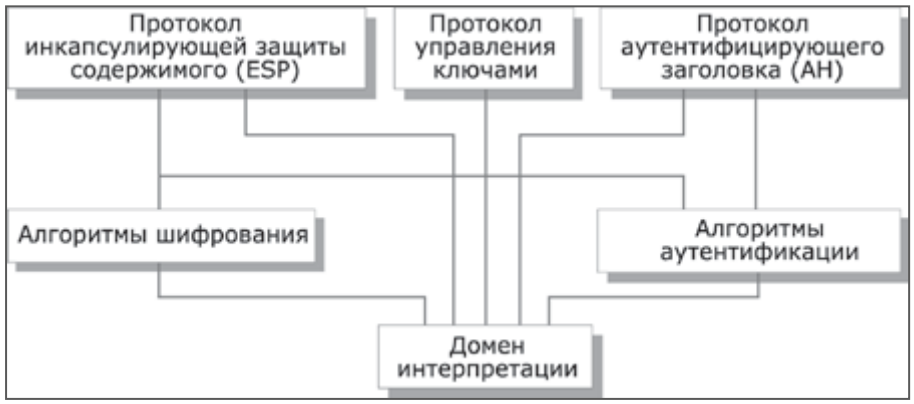


Рис. 1.16. Основные элементы архитектуры средств безопасности IP-уровня

Деление на уровни важно для всех аспектов информационных технологий. Там же, где участвует еще и криптография, важность возрастает вдвойне, поскольку приходится считаться не только с чисто техническими факторами, но и с особенностями законодательства различных стран, с ограничениями на экспорт и/или импорт криптосредств.

IPSec поддерживает две формы целостности: целостность соединения и частичную целостность последовательности. Целостность соединения является сервисом безопасности, который определяет модификацию конкретной IP датаграммы, безотносительно последовательности датаграмм в потоке трафика. Частичная целостность последовательности является anti-reply сервисом, с помощью которого определяется получение дубликатов IP датаграм.

Эти сервисы как раз и реализуются с использованием двух протоколов обеспечения безопасного трафика, Authentication Header (AH) и Encapsulating Security Payload (ESP), и с помощью процедур и протоколов управления криптографическим ключом. Множество применяемых IPSec протоколов и метод их использования определяются требованиями безопасности.

Когда данные механизмы установлены корректно, они не мешают пользователям, хостам и другим компонентам Internet, которые не применяют данные механизмы безопасности для защиты своего трафика. Протоколы обеспечения аутентичности и конфиденциальности в IPSec не зависят от конкретных криптографических алгоритмов. (Более того, само деление на аутентичность и конфиденциальность предоставляет и разработчикам, и пользователям дополнительную степень свободы в ситуации, когда к криптографическим относят только шифровальные средства.) В каждой стране могут применяться свои алгоритмы, соответствующие национальным стандартам, но для этого, как минимум, нужно позаботиться об их регистрации в домене интерпретации. Это означает возможность выбора различного набора алгоритмов без воздействия на другие части реализации. Например, различные группы пользователей могут выбрать при необходимости различные наборы алгоритмов.

Определен стандартный набор алгоритмов по умолчанию для обеспечения интероперабельности. Использование этих алгоритмов совместно с защитой трафика на основе IPSec и протоколами управления

ключа позволяет обеспечить высокую степень криптографической безопасности.

Алгоритмическая независимость протоколов имеет и оборотную сторону, состоящую в необходимости предварительного согласования набора применяемых алгоритмов и их параметров, поддерживаемых общающимися сторонами. Иными словами, стороны должны выработать общий контекст безопасности (Security Association, SA) и затем использовать такие его элементы, как алгоритмы и их ключи. SA подробно рассматривается далее. За формирование контекстов безопасности в IPSec отвечает особое семейство протоколов ISAKMP, которое рассматривается также в отдельном разделе.

Безопасность, обеспечиваемая IPSec, зависит от многих факторов операционного окружения, в котором IPSec выполняется. Например, от безопасности ОС, источника случайных чисел, плохих протоколов управления системой.

Размещение и функционирование IPSec

IPSec выполняется на хосте или шлюзе безопасности, обеспечивая защиту IP- трафика. Термин «шлюз безопасности» используется для обозначения промежуточной системы, которая реализует IPSec-протоколы. Защита основана на требованиях, определенных в Базе Данных Политики Безопасности (Security Policy Database- SPD), определяемой и поддерживаемой системным администратором. Пакеты обрабатываются одним из трех способов на основании соответствия информации заголовка IP или транспортного уровня записям в SPD. Каждый пакет либо отбрасывается сервисом безопасности IPSec, либо пропускается без изменения, либо обрабатывается сервисом IPSec на основе применения определенной политики.

IPSec обеспечивает сервисы безопасности на IP-уровне, выбирая нужные протоколы безопасности, определяя алгоритмы, используемые сервисами, и

предоставляя все криптографические ключи требуемым сервисам. IPSec может использоваться для защиты одного или нескольких путей между парой хостов, между парой шлюзов безопасности или между шлюзом безопасности и хостом.

IPSec использует два протокола для обеспечения безопасности трафика - Authentication Header (AH) и Encapsulating Security Payload (ESP). Хотя бы один из этих сервисов должен быть задействован при использовании ESP.

Эти протоколы могут применяться как по отдельности так и в комбинации с друг другом для обеспечения необходимого набора сервисов безопасности в IPv4 и IPv6. Каждый протокол поддерживает два режима использования: режим транспорта и режим туннелирования. В транспортном режиме протоколы обеспечивают защиту главным образом для протоколов более высокого уровня; в режиме туннелирования протоколы применяются для скрытия IP-заголовков исходных пакетов. Разница между двумя режимами рассматривается дальше.

IPSec позволяет системному администратору управлять детализацией, с которой предоставляется сервис безопасности. Например, можно создать единственный зашифрованный туннель между двумя безопасными шлюзами, или для каждого TCP соединения может быть создан зашифрованный туннель между парой хостов. IPSec позволяет указывать следующие параметры:

а) какие сервисы используются, и в какой комбинации;

б) необходимый уровень детализации применяемой защиты;

в) алгоритмы, используемые для обеспечения безопасности на основе криптографии.

Существует несколько способов реализации IPSec на хосте или в соединении с роутером или firewall (для создания безопасного шлюза). Несколько общих примеров:

а) интеграция IPSec в конкретную реализацию IP, что требует доступа к исходному коду IP и применимо как к хостам, так и к шлюзам безопасности;

б) bump-in-the-stack (BITS) реализации, где IPSec действует внизу существующей реализации стека протоколов IP, между обычным IP и локальными сетевыми драйверами; доступа к исходному коду стека IP в данном контексте не требуется, что делает такой подход пригодным для встраивания в существующие системы, и реализации на хостах;

в) использование внешнего криптопроцессора (обычно в военных и в некоторых коммерческих системах), как правило, это является Bump-in-the-stack (BITS) реализацией, используется как на хостах, так и на шлюзах, обычно BITS-устройства являются IP-адресуемыми;

Транспортный режим работы

В этом варианте механизмы безопасности применяются только для протоколов, начиная с транспортного (TCP) уровня и выше, оставляя данные самого сетевого уровня (заголовок IP) без дополнительной защиты. Места размещения дополнительной информации, вставляемой протоколами в пакет, представлены в соответствии с рисунком 1.17.

Рис. 1.17. Транспортный режим

Туннельный режим работы

Этот режим интересен тем, что обеспечивает защиту также и данных сетевого уровня путем добавления нового IP-заголовка. После определения

ассоциаций безопасности (например, между двумя шлюзами) истинные адреса хостов отправления и назначения (и другие служебные поля) полностью защищаются от модификаций для AH или вообще скрываются для ESP, а в новый заголовок выставляются адреса и другие данные для шлюзов (отправления/получения). В соответствии с рисунком 1.18 видны преимущества и недостатки обоих протоколов. ESP обеспечивает сокрытие данных, но не полную аутентификацию всего пакета. AH полностью аутентифицирует, но не скрывает данные. В этом причина того, что для обеспечения высокого уровня безопасности, применение протоколов совмещается.

Рис. 1.18. Туннельный режим

Контексты безопасности и управление ключами

Формирование контекстов безопасности в IPSec разделено на две фазы. Сначала создается управляющий контекст, назначение которого - предоставить доверенный канал, т. е. аутентифицированный, защищенный канал для выработки (в рамках второй фазы) протокольных контекстов и, в частности, для формирования криптографических ключей, используемых протоколами AH и ESP.

В принципе, для функционирования механизмов IPSec необходимы только протокольные контексты; управляющий играет вспомогательную роль. Более того, явное выделение двух фаз утяжеляет и усложняет формирование ключей, если рассматривать последнее как однократное действие. Тем не менее, из архитектурных соображений управляющие контексты не только могут, но и должны существовать, поскольку обслуживают все протокольные уровни стека TCP/IP, концентрируя в одном месте необходимую функциональность. Первая фаза начинается в ситуации, когда взаимодействующие стороны не имеют общих секретов (общих ключей) и не уверены в аутентичности друг друга. Если с самого начала не создать доверенный канал, то для выполнения каждого управляющего действия с ключами (их модификация, выдача диагностических сообщений и т.п.) в каждом протоколе (AH, ESP, TLS и т.д.) этот канал придется формировать заново.

Общие вопросы формирования контекстов безопасности и управления ключами освещаются в спецификации «Контексты безопасности и управление ключами в Internet» (Internet Security Association and Key Management Protocol, ISAKMP). Здесь вводятся две фазы выработки протокольных ключей, определяются виды управляющих информационных обменов и используемые форматы заголовков и данных. Иными словами строится протокольно-независимый каркас.

Существует несколько способов формирования управляющего контекста. Они различаются двумя показателями:

- используемым механизмом выработки общего секретного ключа;

- степенью защиты идентификаторов общающихся сторон;

В простейшем случае секретные ключи задаются заранее (ручной метод распределения ключей). Для небольших сетей такой подход вполне работоспособен, но он не является масштабируемым. Последнее свойство может быть обеспечено при автоматической выработке и распределении секретных ключей в рамках протоколов, основанных на протоколе Диффи-

Хеллмана. Пример тому - «Протокол для обмена ключами в Internet « (The Internet Key Exchange, IKE).

Протокол Ди́ффи-Хе́ллмана (англ. Diffie-Hellman, DH) — криптографический протокол, позволяющий двум и более сторонам получить общий секретный ключ, используя незащищенный от прослушивания канал связи. Полученный ключ используется для шифрования дальнейшего обмена с помощью алгоритмов симметричного шифрования. Алгоритм был впервые опубликован Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом в 1976 году[4, 5].

При формировании управляющего контекста идентификаторы общающихся сторон (например, IP-адреса) могут передаваться в открытом виде или шифроваться. Поскольку ISAKMP предусматривает функционирование в режиме клиент/сервер (т. е. ISAKMP-сервер может формировать контекст для клиента), сокрытие идентификаторов в определенной степени повышает защищенность от пассивного прослушивания сети.

Последовательность передаваемых сообщений, позволяющих сформировать управляющий контекст и обеспечивающих защиту идентификаторов, выглядит в соответствии с рисунком 1.19.

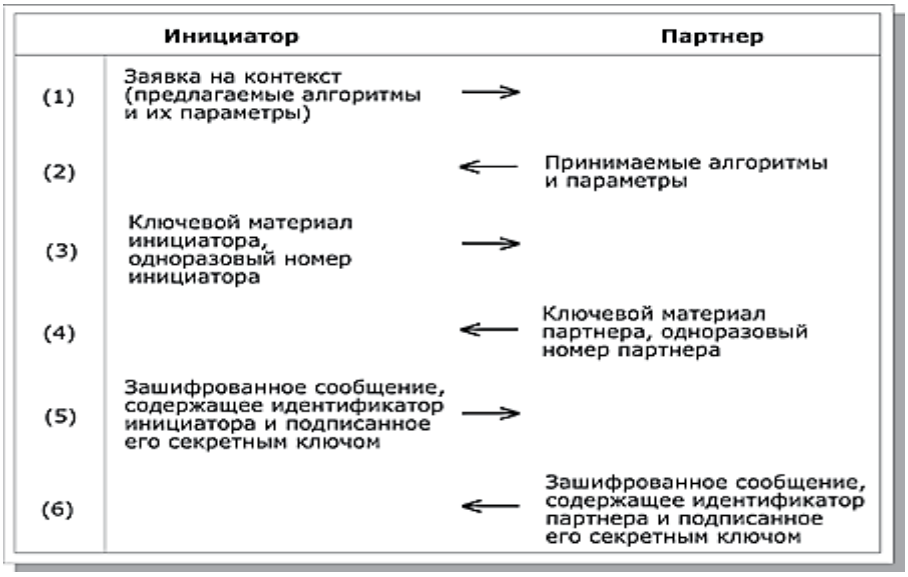


Рис. 1.19. Формирование управляющего контекста

В первом сообщении (1) инициатор направляет предложения по набору защитных алгоритмов и конкретных механизмов их реализации. Предложения упорядочиваются по степени предпочтительности (для инициатора). В ответном сообщении (2) партнер информирует о сделанном выборе - какие алгоритмы и механизмы его устраивают. Для каждого класса защитных средств (генерация ключей, аутентификация, шифрование) выбирается только один элемент.

В сообщениях (3) и (4) инициатор и партнер отправляют свои части ключевого материала, необходимые для выработки общего секретного ключа (опускаются детали, специфичные для алгоритма Диффи-Хеллмана). Одноразовые номера (nonce) представляют собой псевдослучайные величины, служащие для защиты от воспроизведения сообщений.

Посредством сообщений (5) и (6) происходит обмен идентификационной информацией, подписанной (с целью аутентификации) секретным ключом отправителя и зашифрованной выработанным на предыдущих шагах общим секретным ключом. Для аутентификации предполагается использование сертификатов открытых ключей. В число подписываемых данных входят одноразовые номера.

В представленном виде протокол формирования управляющего контекста защищает от атак, производимых нелегальным посредником, а также от нелегального перехвата соединений. Для защиты от атак на доступность, для которых характерно прежде всего навязывание интенсивных вычислений, присущих криптографии с открытым ключом, применяются так называемые идентифицирующие цепочки (cookies). Эти цепочки, формируемые инициатором и его партнером с использованием текущего времени (для защиты от воспроизведения), на самом деле присутствуют во всех ISAKMP-сообщениях и в совокупности идентифицируют управляющий контекст (в первом сообщении, по понятным причинам, фигурирует только цепочка инициатора). Согласно спецификациям, заголовок ISAKMP-сообщения имеет вид в соответствии с рисунком 1.20.

Если злоумышленник пытается «завалить» кого-либо запросами на создание управляющего контекста, подделывая при этом свой IP-адрес, то в сообщении (3) он не сможет предъявить идентифицирующую цепочку партнера, поэтому до выработки общего секретного ключа и, тем более, электронной подписи и полномасштабной проверки аутентичности дело попросту не дойдет.

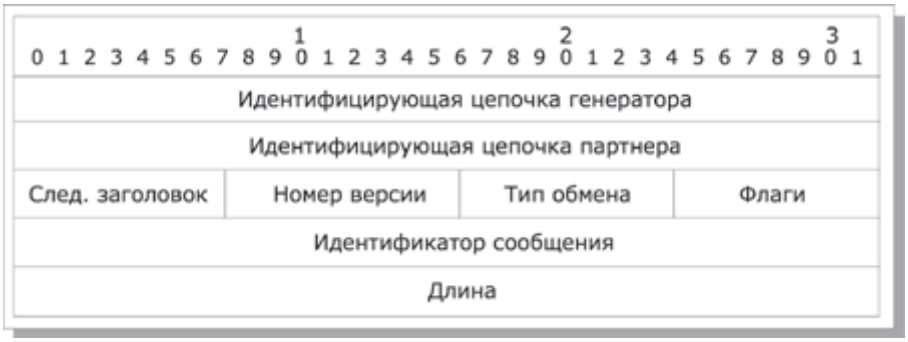


Рис. 1.20. Формат заголовка ISAKMP-сообщения

Управляющие контексты являются двунаправленными в том смысле, что любая из общающихся сторон может инициировать с их помощью выработку новых протокольных контекстов или иные действия. Для передачи ISAKMP-сообщений используется любой протокол, однако в качестве стандартного принят UDP с номером порта 500.

Протокольные контексты и политика безопасности

Системы, реализующие IPSec, должны поддерживать две базы данных:

- базу данных политики безопасности (Security Policy Database, SPD);

- базу данных протокольных контекстов безопасности (Security Association Database, SAD);

Все IP-пакеты (входящие и исходящие) сопоставляются с упорядоченным набором правил политики безопасности. При сопоставлении используется фигурирующий в каждом правиле селектор - совокупность анализируемых полей сетевого уровня и более высоких протокольных уровней. Первое подходящее правило определяет дальнейшую судьбу пакета:

- пакет может быть ликвидирован;

- пакет может быть обработан без участия средств IPSec;
- пакет должен быть обработан средствами IPSec с учетом набора протокольных контекстов, ассоциированных с правилом.

Таким образом, системы, реализующие IPSec, функционируют как межсетевые экраны, фильтруя и преобразуя потоки данных на основе предварительно заданной политики безопасности.

Далее рассматриваются контексты и политика безопасности, а также порядок обработки сетевых пакетов.

Протокольный контекст безопасности в IPSec - это однонаправленное соединение (от источника к получателю), предоставляющее обслуживаемым потокам данных набор защитных сервисов в рамках какого-то одного протокола (AH или ESP). В случае симметричного взаимодействия партнерам придется организовать два контекста (по одному в каждом направлении). Если используются и AH, и ESP, потребуется четыре контекста.

Элементы базы данных протокольных контекстов содержат следующие поля (в каждом конкретном случае некоторые значения полей будут пустыми):

- используемый в протоколе AH алгоритм аутентификации, его ключи и т.п.;
- используемый в протоколе ESP алгоритм шифрования, его ключи, начальный вектор и т.п.;
- используемый в протоколе ESP алгоритм аутентификации, его ключи и т.п.;
- время жизни контекста;
- режим работы IPSec: транспортный или туннельный;
- максимальный размер пакетов;
- группа полей (счетчик, окно, флаги) для защиты от воспроизведения пакетов.

Пользователями протокольных контекстов, как правило, являются прикладные процессы. Вообще говоря, между двумя узлами сети может существовать произвольное число протокольных контекстов, так как число приложений в узлах произвольно. В качестве пользователей управляющих контекстов обычно выступают узлы сети (поскольку в этих контекстах желательно сосредоточить общую функциональность, необходимую сервисам безопасности всех протокольных уровней эталонной модели для управления криптографическими ключами).

Управляющие контексты - двусторонние, т. е. любой из партнеров может инициировать новый ключевой обмен. Пара узлов может одновременно поддерживать несколько активных управляющих контекстов, если имеются приложения с существенно разными криптографическими требованиями. Например, допустима выработка части ключей на основе предварительно распределенного материала, в то время как другая часть порождается по алгоритму Диффи-Хеллмана.

Протокольный контекст для IPSec идентифицируется целевым IP-адресом, протоколом (AH или ESP), а также дополнительной величиной - индексом параметров безопасности (Security Parameter Index, SPI). Последняя величина необходима, поскольку могут существовать несколько контекстов с одинаковыми IP-адресами и протоколами. Далее показано, как используются индексы SPI при обработке входящих пакетов.

IPSec обязывает поддерживать ручное и автоматическое управление контекстами безопасности и криптографическими ключами. В первом случае все системы заранее снабжаются ключевым материалом и иными данными, необходимыми для защищенного взаимодействия с другими системами. Во втором - материал и данные вырабатываются динамически, на основе определенного протокола - IKE, поддержка которого обязательна.

Протокольный контекст создается на базе управляющего с использованием ключевого материала и средств аутентификации и шифрования последнего. В простейшем случае, когда протокольные ключи

генерируются на основе существующих, последовательность передаваемых сообщений выглядит в соответствии с рисунком 1.21.

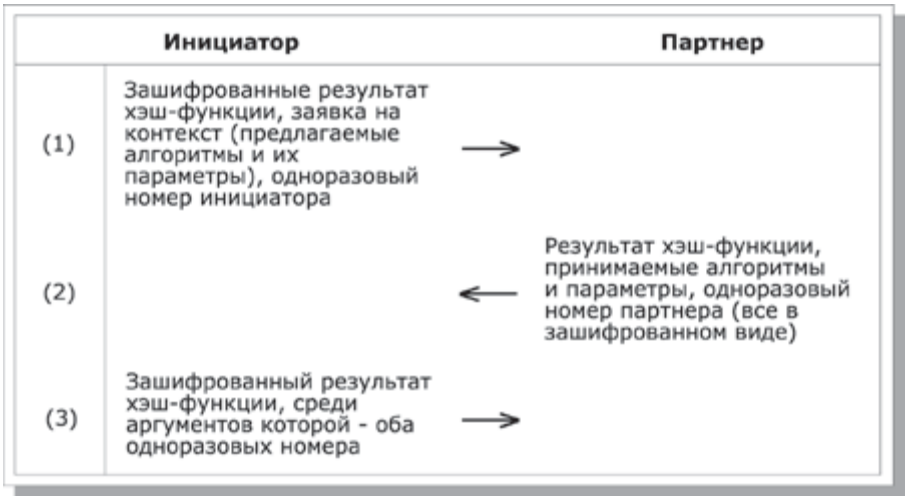


Рис. 1.21. Формирование протокольного контекста

Аутентификационный заголовок

Аутентификационный заголовок (англ. authentication header) AH предназначен для обеспечения аутентификации отправителя, контроля целостности данных и опционально для предотвращения повторной посылки (англ. replay) пакета - при условии, что принимающая сторона настроена производить проверку последовательного номера пакета. Поля IP-пакета, которые изменяются в пути следования, не подлежат контролю целостности. AH защищает данные протоколов более высоких уровней и те поля IP-заголовков, которые не меняются на маршруте доставки или меняются предсказуемым образом (число «непредсказуемых» полей невелико - это prio. (Traffic Class), Flow Label и Hop Limit. Предсказуемо меняется целевой адрес при наличии дополнительного заголовка исходящей маршрутизации.).

Формат заголовка AH показан в соответствии с рисунком 1.22. поле «Следующий заголовок» (Next header) идентифицирует тип следующих значимых данных за аутентификационным заголовком; значения поля

берутся из предопределенного множества установленных IANA (Internet Assigned Numbers Authority) номеров.

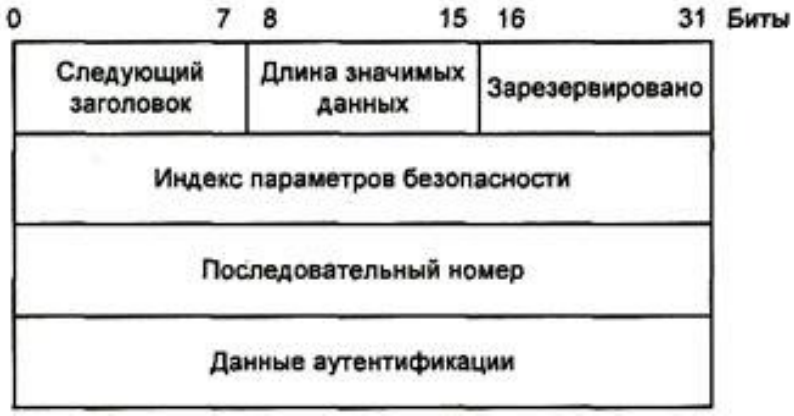


Рис. 1.22. Формат заголовка AH

Поле «Длина значимих данных» (Payload length) определяет длину самого AH в 32-битовых словах минус 2 слова, поскольку для заголовков, расширяемых для IPv6, установлено требование сокращения длины заголовка на 64 бита;

Зарезервированное поле должно быть нулевым;

Поле «Индекс параметров безопасности» (Security Parameters Index - SPI) - это значение, которое в совокупности с адресом назначения и самим протоколом (в данном случае AH) однозначно определяет ассоциацию безопасности (Security Association - SA) для данной датаграммы в виде 32-битного номера; номера с 1 по 255 зарезервированы IANA; SPI, равный нулю, означает, что SA не установлена; ассоциация безопасности - это набор параметров (версия алгоритмов шифрования и аутентификации, схема обмена ключами и т. п.), определяющих, каким образом будет обеспечиваться защита данных;

Поле «Последовательный номер» (Sequence number) - зто монотонно возрастающий от 0 (при установлении SA) номер пакета. Он используется для возможности контроля получателем ситуации повторной пересылки пакетов;

Поле «Данные аутентификации» (Authentication data) содержат значение контроля целостности (Integrity check value - ICV), рассчитанное по всем данным, которые не изменяются в пути следования пакета или предсказуемы на момент достижения им получателя. Значение ICV рассчитывается в зависимости от алгоритма, определенного в SA, например код аутентификации сообщения (Message Authentication Code - MAC) с ключом симметричного или асимметричного алгоритма или хэшфункции;

Безопасное сокрытие существенных данных

Протокол инкапсулирующей защиты содержимого (Encapsulating Security payload, ESP) предоставляет три вида сервисов безопасности:

1. обеспечение конфиденциальности (шифрование содержимого IP-пакетов, а также частичная защита от анализа трафика путем применения туннельного режима);
2. обеспечение целостности IP-пакетов и аутентификации источника данных;
3. обеспечение защиты от воспроизведения IP-пакетов;

Функциональность ESP шире, чем у AH (добавляется шифрование); ESP не обязательно предоставляет все сервисы, но либо конфиденциальность, либо аутентификация должны быть задействованы. Формат заголовка ESP выглядит в соответствии с рисунком 1.23. Это не столько заголовок, сколько обертка (инкапсулирующая оболочка) для зашифрованного содержимого. Например, ссылку на следующий заголовок нельзя выносить в начало, в незашифрованную часть, так как она лишится конфиденциальности.

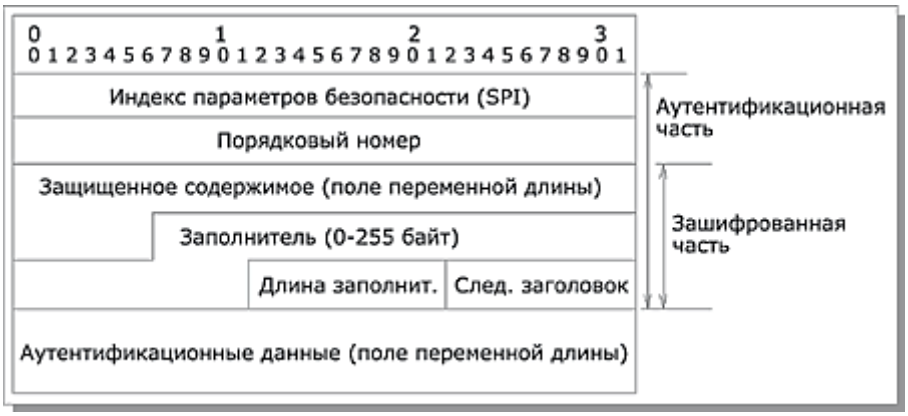


Рис. 1.23. Формат заголовка ESP

Поля "Индекс параметров безопасности (SPI)", "Порядковый номер" и "Аутентификационные данные" (последнее присутствует только при включенной аутентификации) имеют тот же смысл, что и для AH. ESP аутентифицирует лишь зашифрованную часть пакета (плюс два первых поля заголовка).

Применение протокола ESP к исходящим пакетам можно представлять себе следующим образом. Пусть остаток пакета та его часть, которая помещается после предполагаемого места вставки заголовка ESP. При этом не важно, какой режим используется - транспортный или туннельный. Шаги протокола таковы:

1. - остаток пакета копируется в буфер;

2. - к остатку приписываются дополняющие байты, их число и номер (тип) первого заголовка остатка, с тем чтобы номер был прижат к границе 32-битного слова, а размер буфера удовлетворял требованиям алгоритма шифрования;

3. - текущее содержимое буфера шифруется;

4. - в начало буфера приписываются поля "Индекс параметров безопасности (SPI)" и "Порядковый номер" с соответствующими значениями;

5. - пополненное содержимое буфера аутентифицируется, в его конец помещается поле "Аутентификационные данные";

6. - в новый пакет переписываются начальные заголовки старого пакета и конечное содержимое буфера;

Если в ESP включены и шифрование, и аутентификация, то аутентифицируется зашифрованный пакет. Для входящих пакетов действия выполняются в обратном порядке, т. е. сначала производится аутентификация. Это позволяет не тратить ресурсы на расшифровку поддельных пакетов, что в какой-то степени защищает от атак на доступность.

Протокол обмена ключами – IKE

Поскольку основным механизмом обеспечения безопасности данных протокола являются криптографические методы, участники защищенного соединения должны наладить обмен соответствующими криптографическими ключами. Обеспечить настройку процесса такого обмена можно вручную и автоматически. Первый способ допустим для небольшого количества достаточно статичных систем, а в общем случае это производится автоматически.

Для автоматического обмена ключами по умолчанию используется Протокол управления ключами в Интернете (Internet Key Management Protocol - IKMP), иначе называемый Обмен ключами в Интернете (Internet Key Exchange - IKE). Дополнительно или альтернативно могут быть применены другие протоколы, такие как Kerberos или SKIP.

IKE совмещает в себе три основных направления (отдельных протокола):

- ISAKMP (Internet Security Association and Key Management Protocol) - протокол ассоциаций безопасности и управления ключами в интернете; это общее описание (framework) для обеспечения аутентификации и обмена ключей без указания конкретных прикладных алгоритмов;

- Oakley (Oakley key determination protocol) - протокол определения ключей Окли; он описывает последовательности обмена ключами - моды (mode) и описывает предоставляемые ими функции;
- SKEMI (Secure Key Exchange Mechanism for Internet) - механизм безопасного обмена ключами в Интернете; он описывает многофункциональные технологии, предоставляющие анонимность, неотрекаемость (аппелируемость) и быстрое обновление ключей;

IKE содержит две фазы согласования ключей. В первой фазе происходит создание защищенного канала, во второй - согласование и обмен ключами, установление SA. Первая фаза использует один из двух режимов: основной (англ. Main Mode) или агрессивный (англ. Aggressive Mode). Различие между ними в уровне защищенности и скорости работы. Основной режим, более медленный, защищает всю информацию, передаваемую между узлами. Агрессивный режим для ускорения работы оставляет ряд параметров открытыми и уязвимыми для прослушивания, его рекомендуется использовать только в случае, когда критическим вопросом является скорость работы. Во второй фазе используется быстрый режим (англ. Quick Mode), названный так потому, что не производит аутентификации узлов, считая, что это было сделано в первой фазе. Эта фаза обеспечивает обмен ключами, с помощью которых происходит шифрование данных.

Расширенный обзор безопасных ассоциаций

Понятие "безопасные ассоциации" (Security Association - SA) является фундаментальным в IPSec.

SA есть симплексное (однонаправленное) логическое соединение, создаваемое для обеспечения безопасности. Весь трафик, передаваемый по SA, некоторым образом обрабатывается в целях обеспечения безопасности. И AH, и ESP используют в своей работе SAs. Одной из основных функций IKE

является установление SA. Далее приводятся различные аспекты управления SA, определим требуемые характеристики управления политикой SA, обработку трафика и технологии управления SA.

SA есть совокупность параметров соединения, которые дают возможность сервисам обеспечивать безопасный трафик. SA определяет использование AH или ESP. Если к потоку трафика применяются оба протокола, AH и ESP, то создаются две SAs. При двунаправленном соединении между двумя хостами или между двумя шлюзами безопасности требуется два SA (по одному на каждое направление).

SA однозначно определяется тройкой, состоящей из Security Parameter Index (SPI), IP Destination Address (адресом назначения) и идентификатора протокола безопасности (AH или ESP). В принципе адрес назначения может быть единственным адресом, широковещательным (broadcast) адресом или групповым (multicast) адресом. Однако механизм управления SA в настоящее время определяется только для единственной SA. Следовательно, SAs будут описаны в контексте point-to-point соединения, даже если концепция также применяется в случае point-to-multipoint.

Определены два режима SA: режим транспорта и режим туннелирования. Транспортный режим SA обеспечивает безопасную связь между двумя хостами. В IPv4 заголовок протокола безопасности транспортного режима появляется сразу после IP заголовка и всех опций и перед любыми протоколами более высокого уровня (TCP или UDP). В случае ESP транспортный режим SA обеспечивает сервисы безопасности только для протоколов более высокого уровня, но не для IP-заголовка. В случае AH защита также распространяется на отдельные части IP-заголовка.

Другим режимом SA является режим туннелирования. Если хотя бы одним из концов соединения является шлюз безопасности, то SA обязательно должна выполняться в туннелирующем режиме. SA между двумя шлюзами безопасности всегда находится в туннелирующем режиме, так же, как и SA между хостом и шлюзом безопасности. Заметим, что когда трафик

предназначен для шлюза безопасности, например, в случае SNMP-команд, шлюз безопасности рассматривается как хост, и допустим транспортный режим. Два хоста могут при желании так же устанавливать туннелирующий режим.

В туннелирующем режиме SA существует "внешний" IP-заголовок, который определяет пункт назначения IPSec, и "внутренний" IP-заголовок, который определяет конечный пункт назначения для пакета. Заголовок протокола безопасности расположен после внешнего IP-заголовка и перед внутренним IP-заголовком. Если AH используется в туннелирующем режиме, части внешнего IP заголовка являются защищенными, как и весь туннелируемый IP-пакет, т.е. все внутренние заголовки защищены, как и все протоколы более высокого уровня. Если применяется ESP, зашита обеспечивается только для туннелируемого пакета, а не для внешнего IP-заголовка.

1.2. Компьютерный практикум по сетевым протоколам

Программная реализация

Для создания защищенного соединения необходимо получить сертификат или создать его самому. Создать SSL сертификат самому можно средствами свободно распространяемого пакета IIS6 Resource Kit Tools. Может потребоваться установка служб. Для этого в Панели управления необходимо выбрать раздел «Программы и компоненты» и в открывшемся окне в меню слева нажать на ссылку «Включение или отключение компонентов Windows». Далее в открывшемся окне нужно включить компонент «Службы IIS».

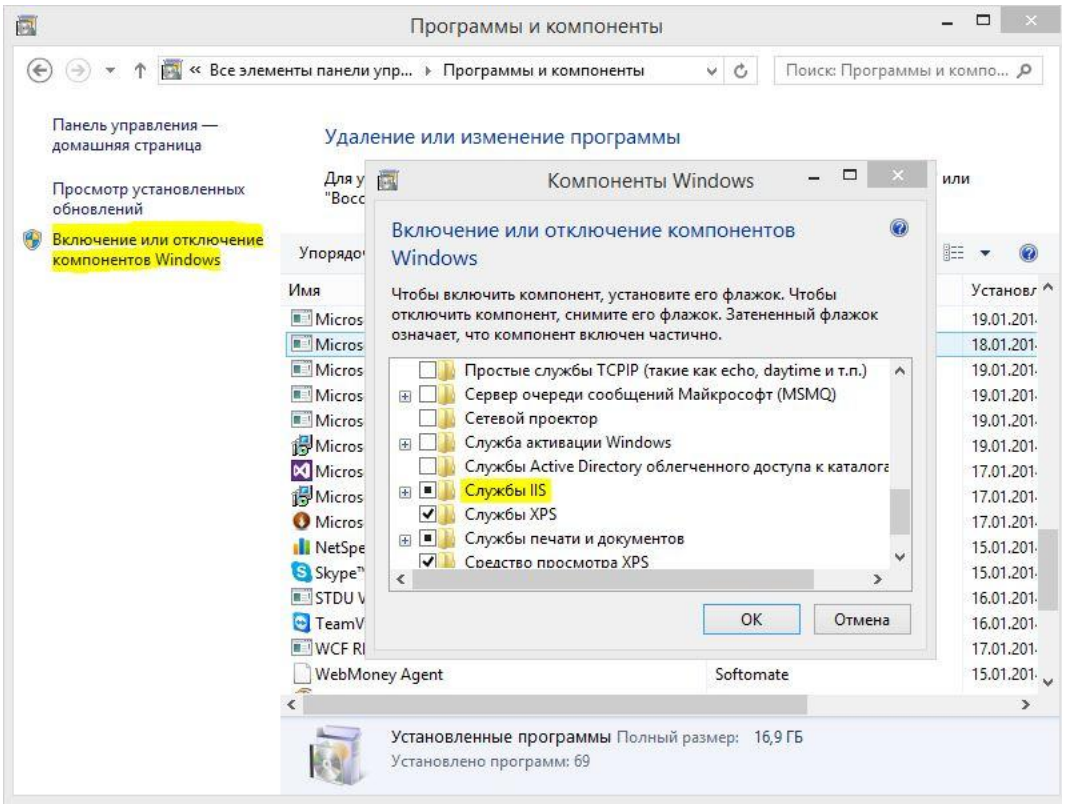


Рис. 1.24. Активация службы ISS

Выбираем службу ISS

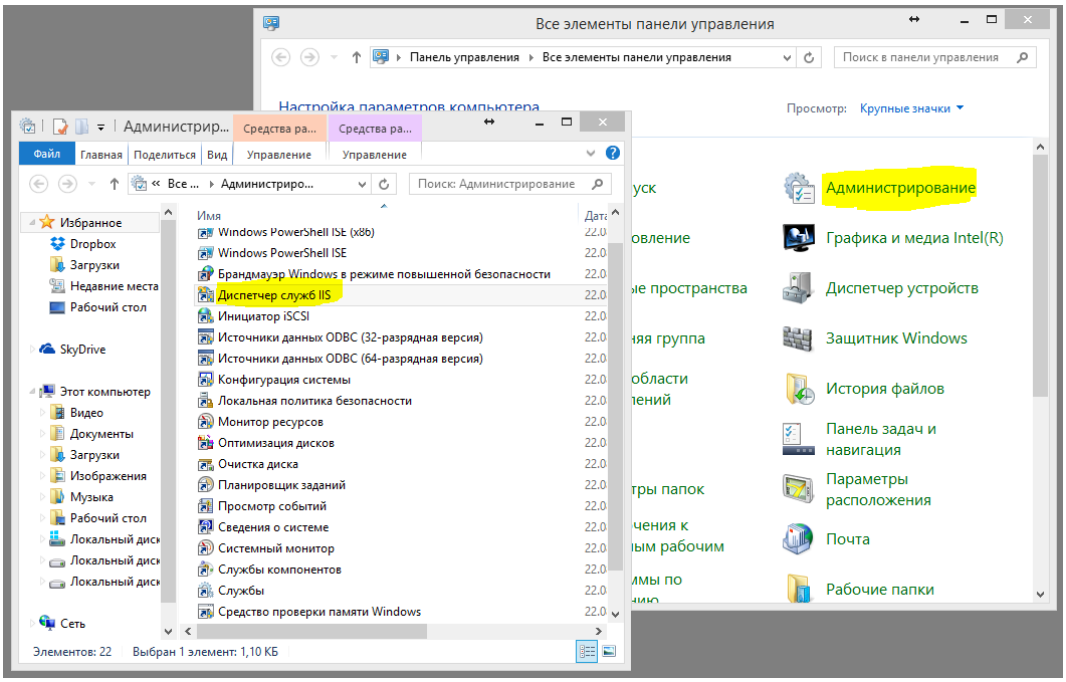


Рис. 1.25. Выбор службы ISS

В открывшемся окне жмем на «Сертификаты сервера»

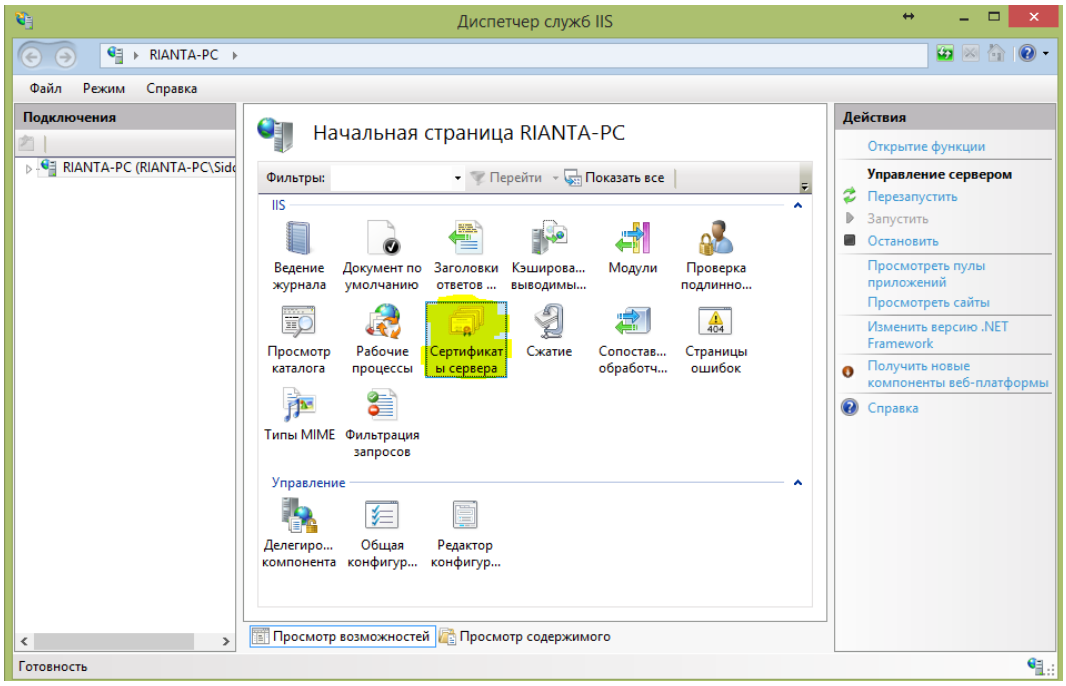


Рис. 1.26. Создание сертификата

Создаем самозаверенный сертификат

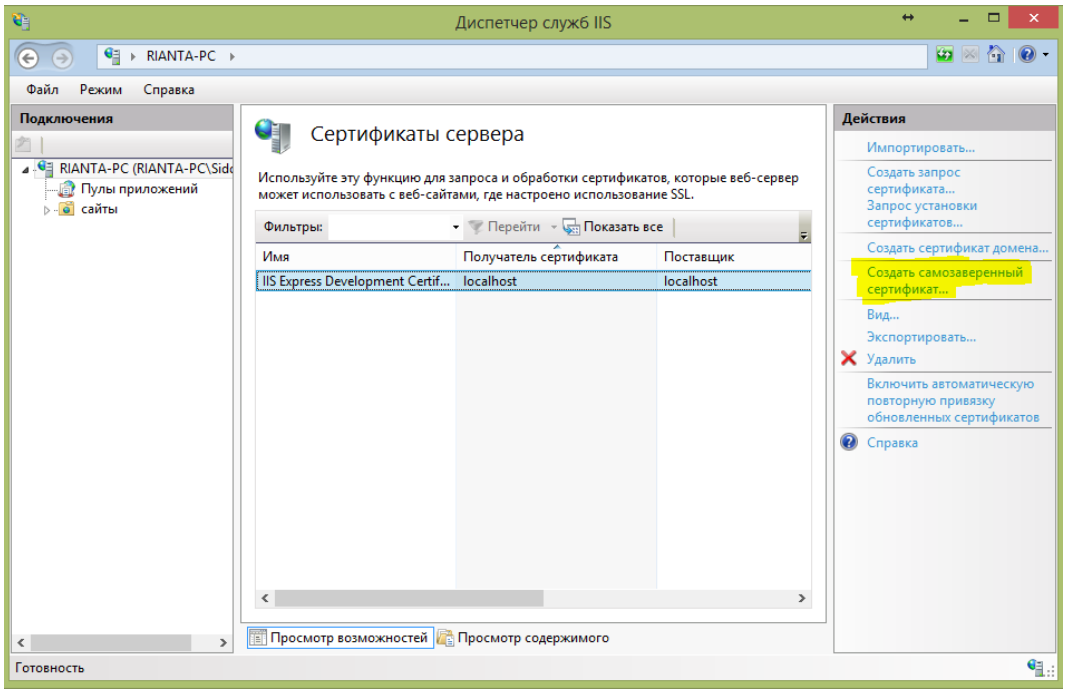


Рис. 1.27. Создание самозаверенного сертификата

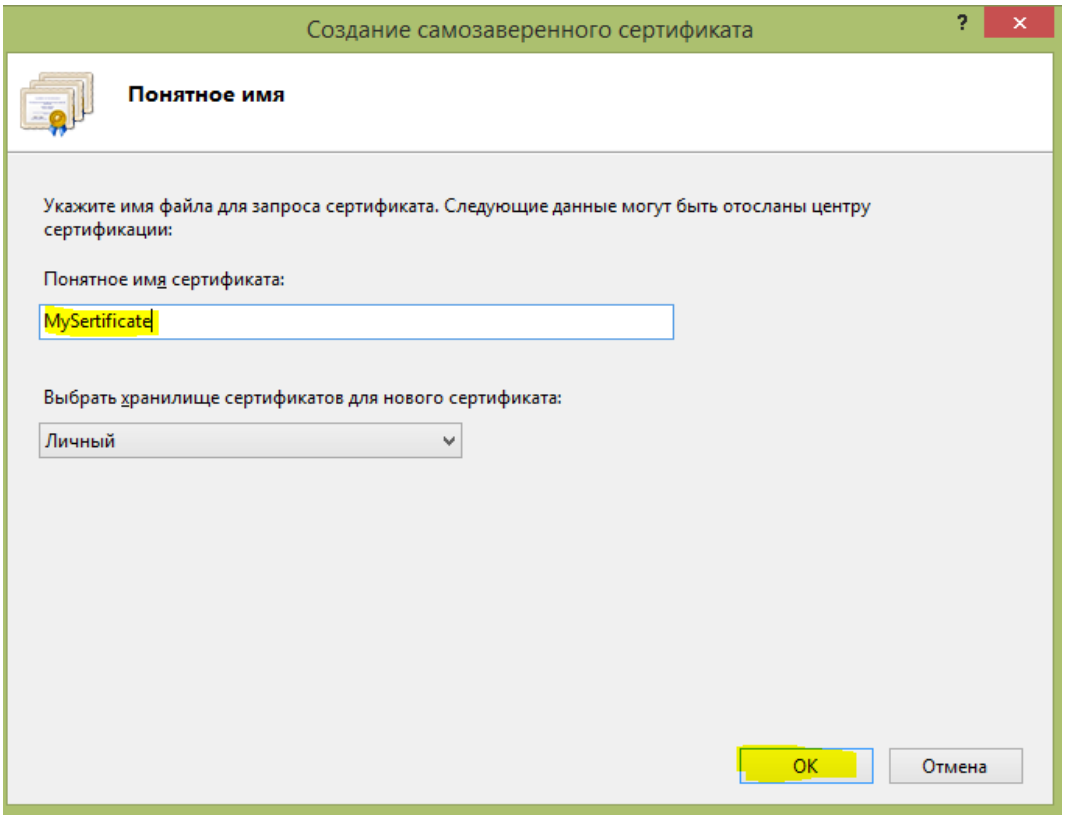


Рис. 1.28. Ввод имени сертификата

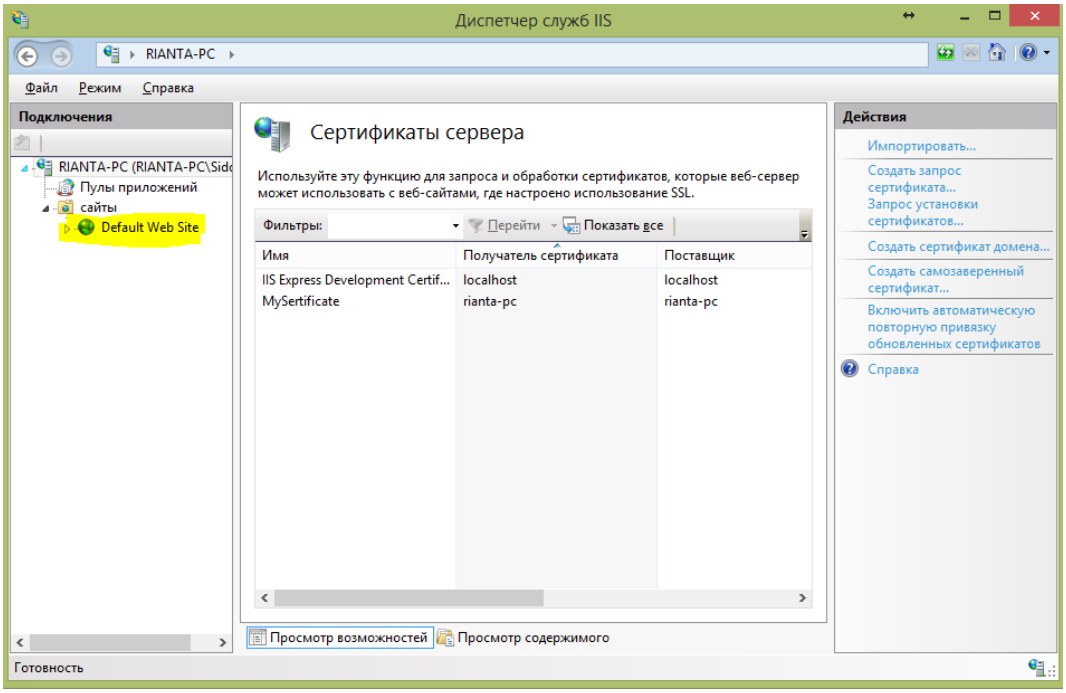


Рис. 1.29. Отображение сертификата

Привязываем сертификат к нужному IP

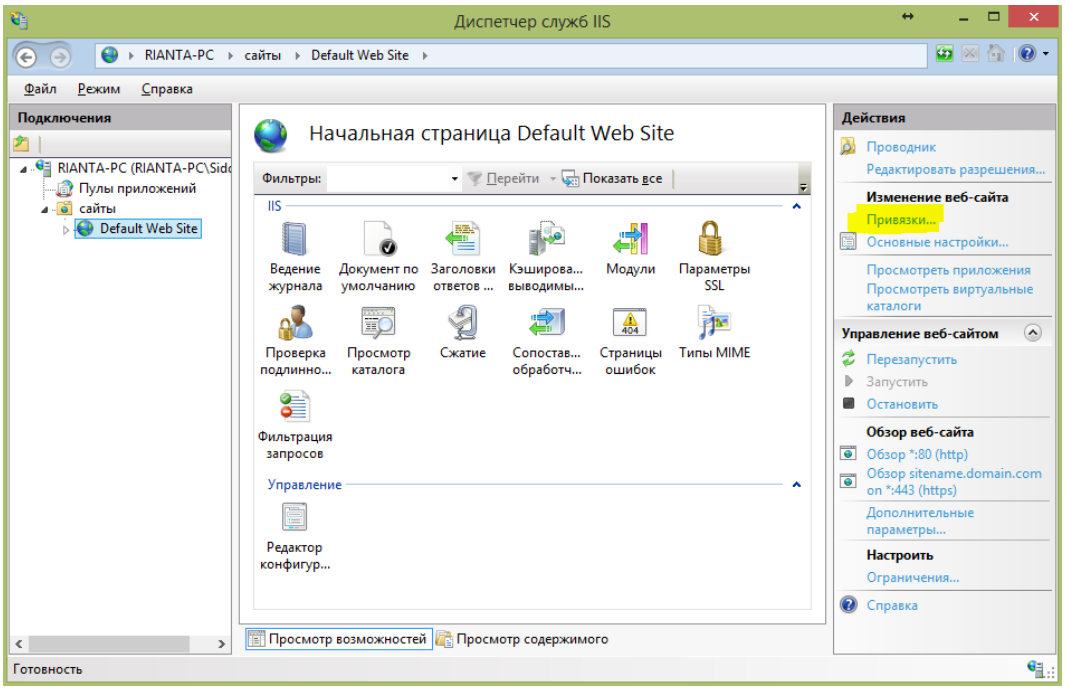


Рис. 1.30. Привязка сертификата к сайту (компьютеру)

Затем с помощью программы THEGREENBOW устанавливаем защищенное SSL – соединение.

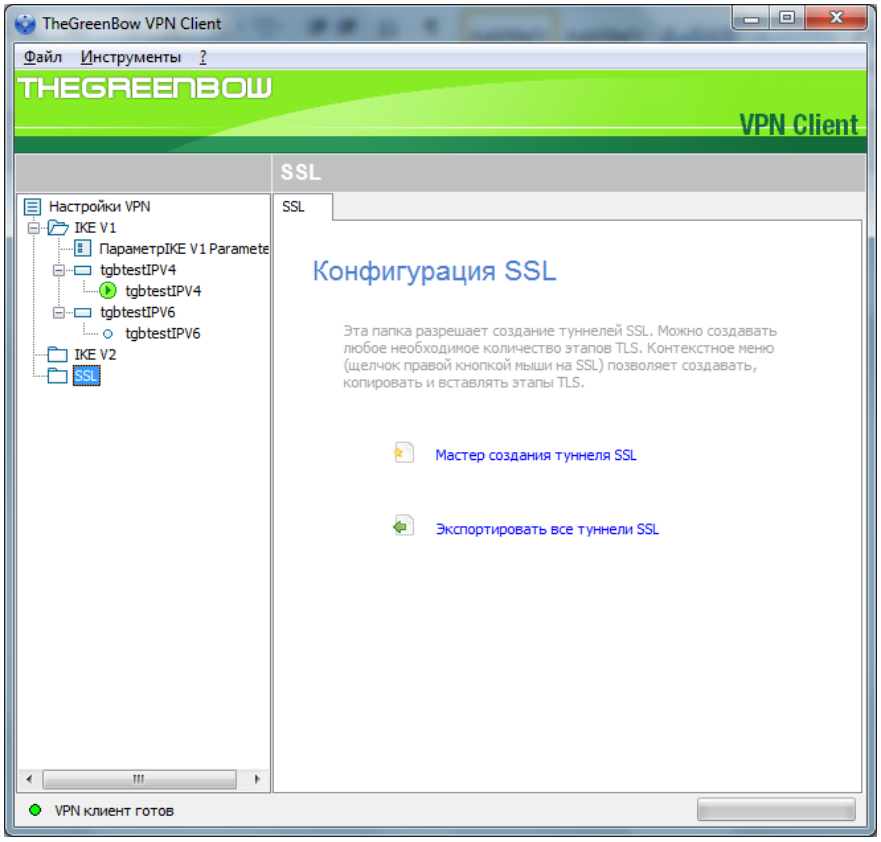


Рис. 1.31. Окно раздела SSL

Указываем путь к сертификату

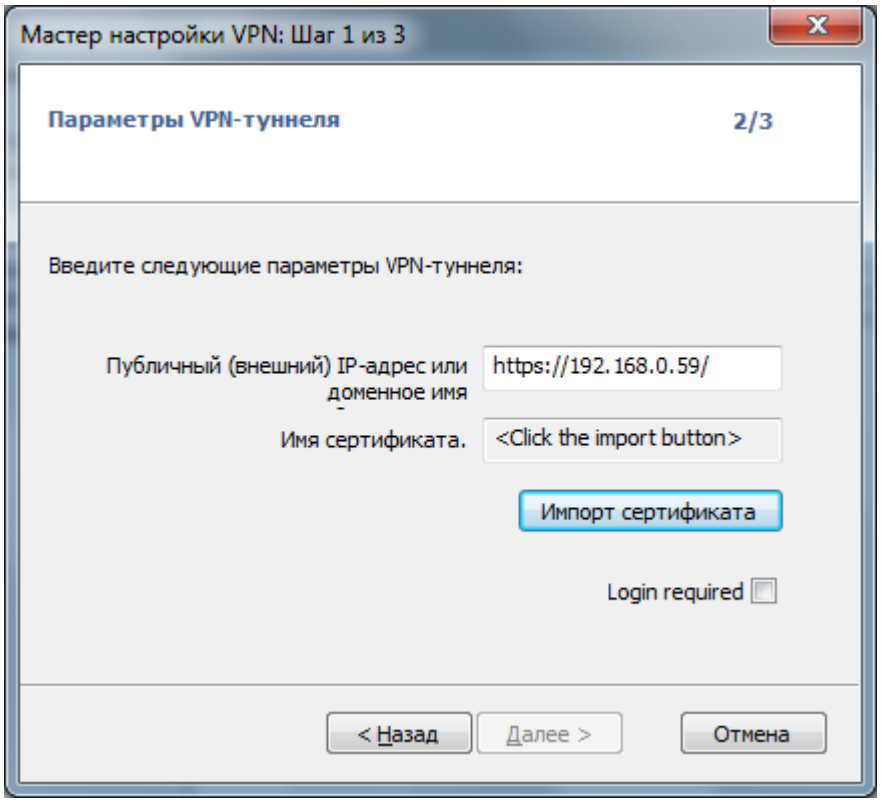


Рис. 1.32. Окно мастера настройки VPN-туннеля

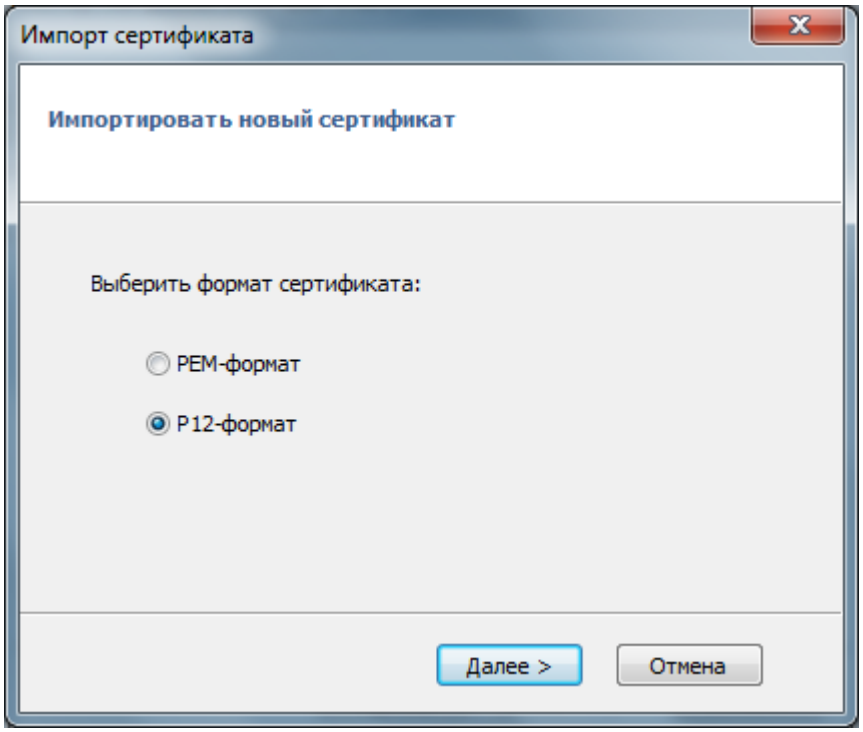


Рис. 1.33. Импорт сертификата

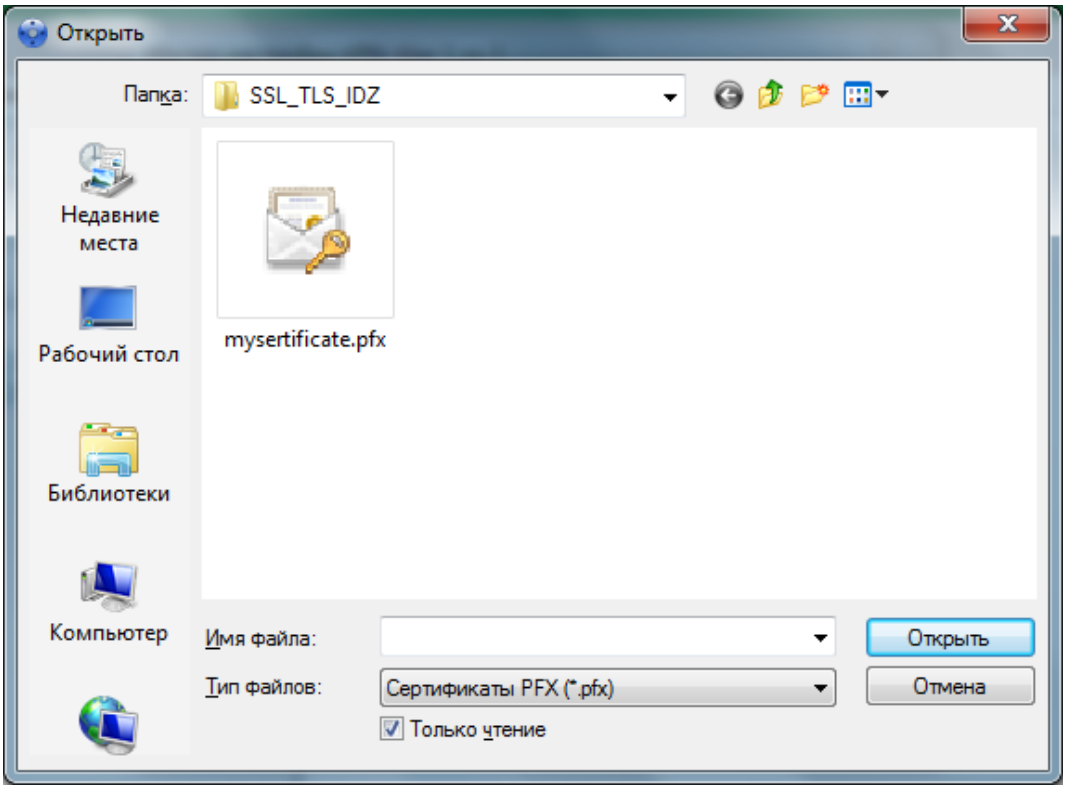


Рис. 1.34. Выбор созданного сертификата

Вводим пароль для доступа к сертификату

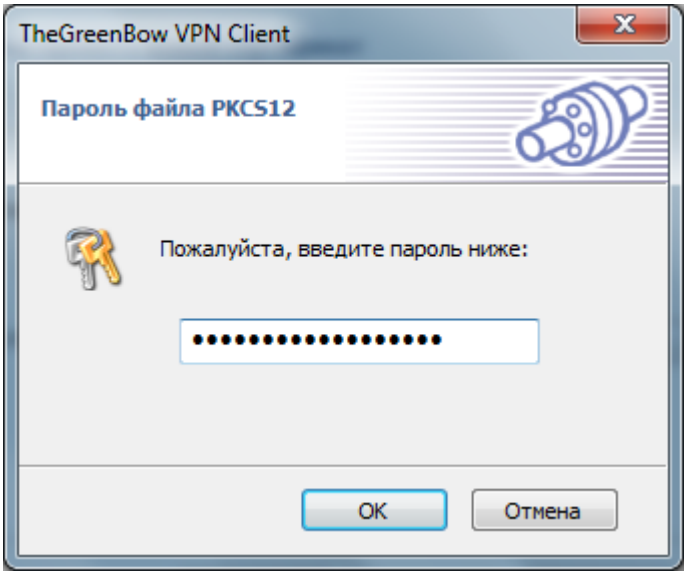


Рис. 1.35. Окно ввода пароля для доступа к сертификату

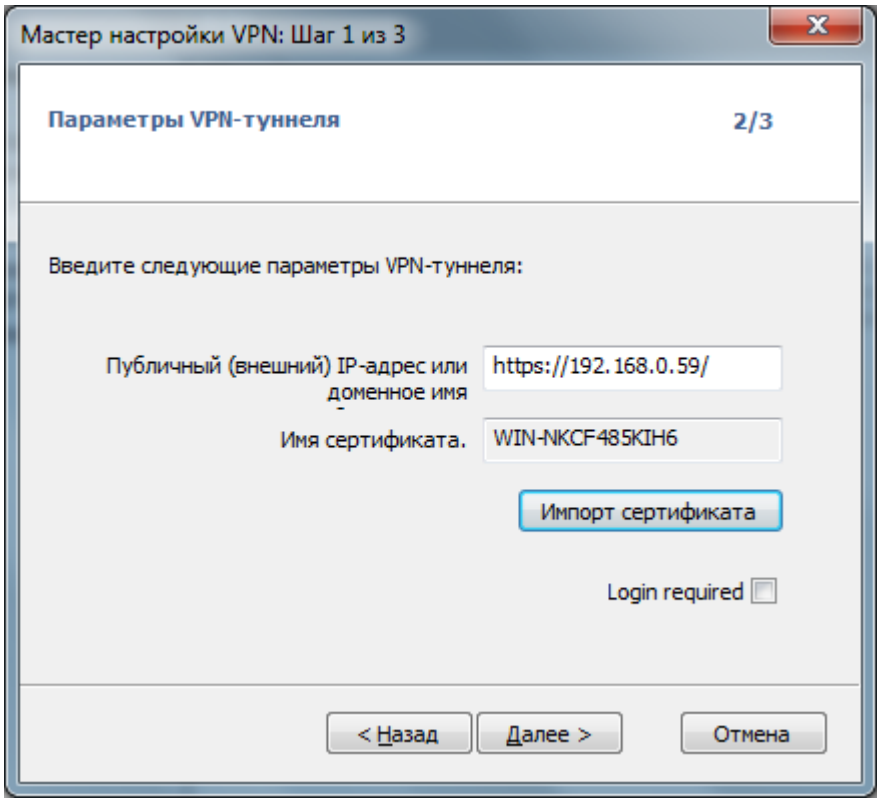


Рис. 1.36. Окно мастера настройки VPN-туннеля

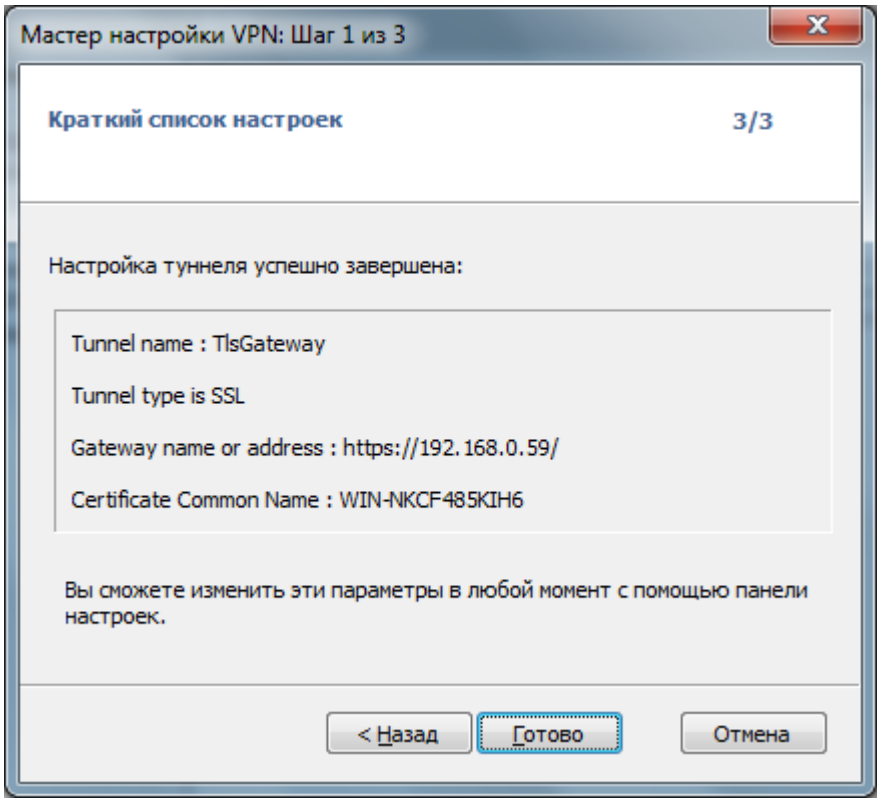


Рис. 1.37. Подтверждение об успешно созданном туннеле

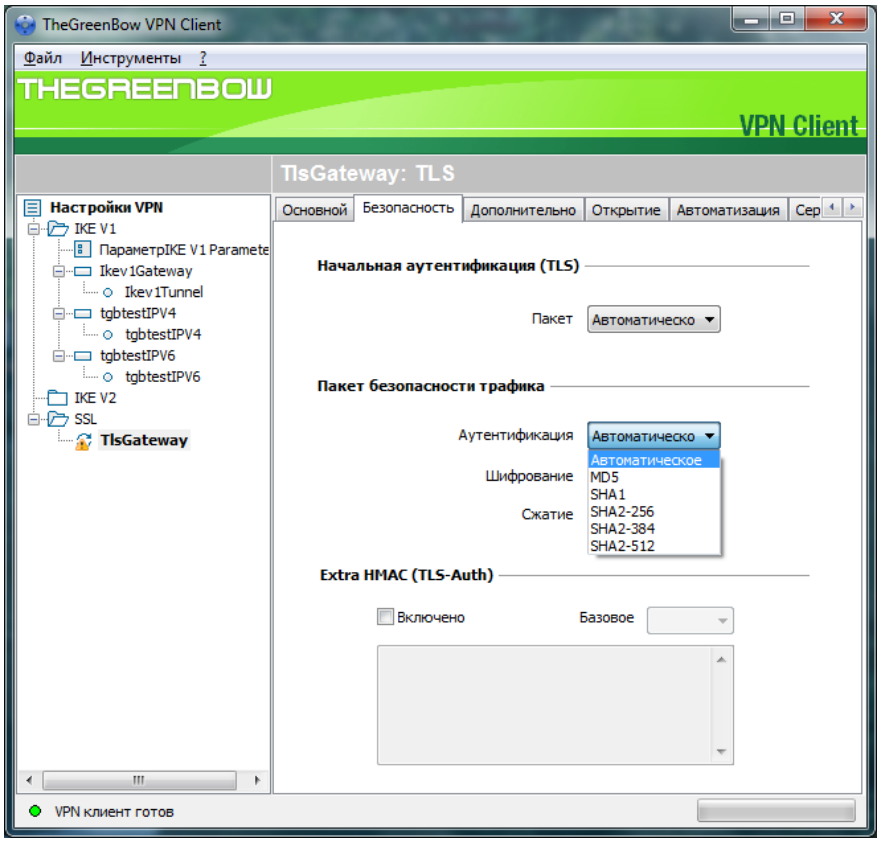


Рис. 1.38. Параметры аутентификации трафика

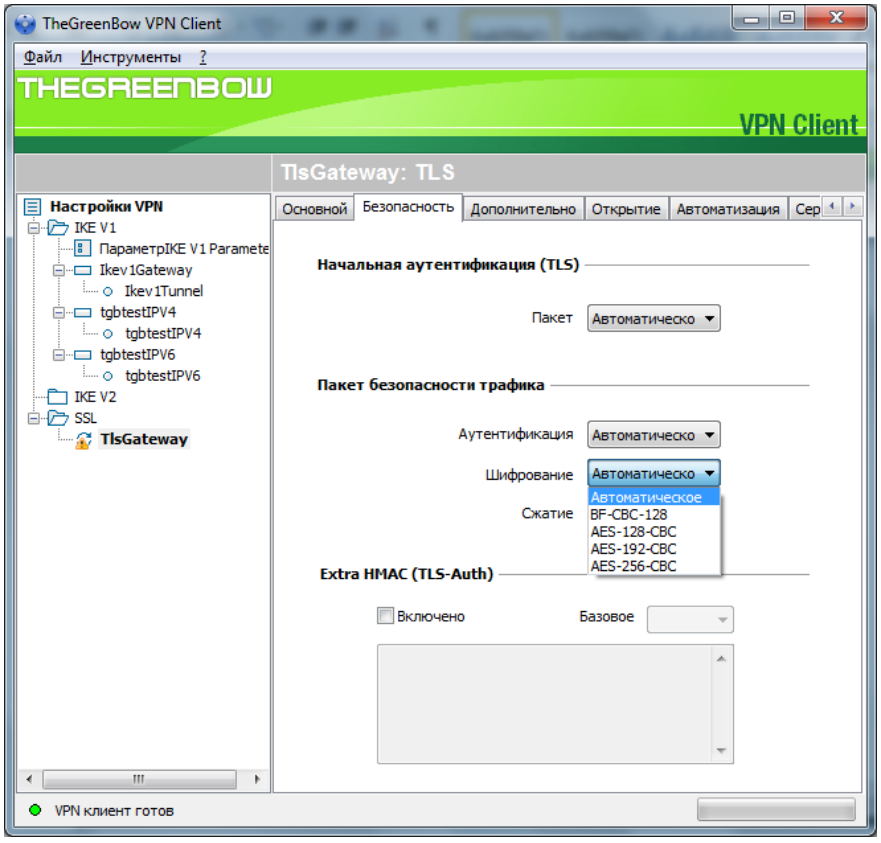


Рис. 1.39. Параметры шифрования

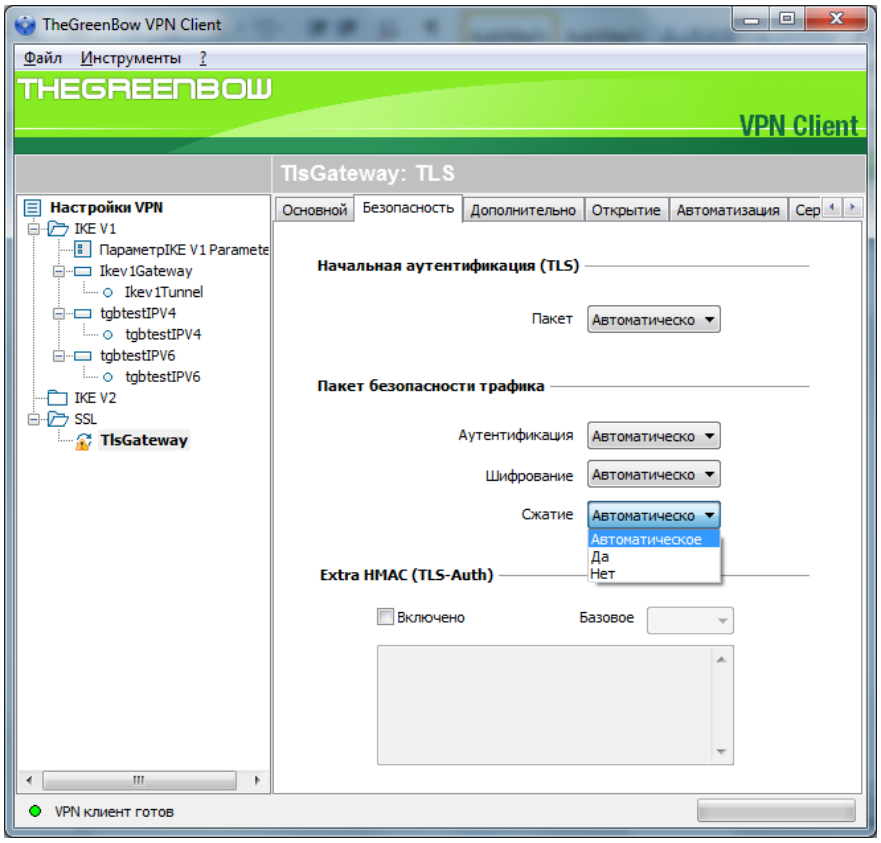


Рис. 1.40. Параметры сжатия

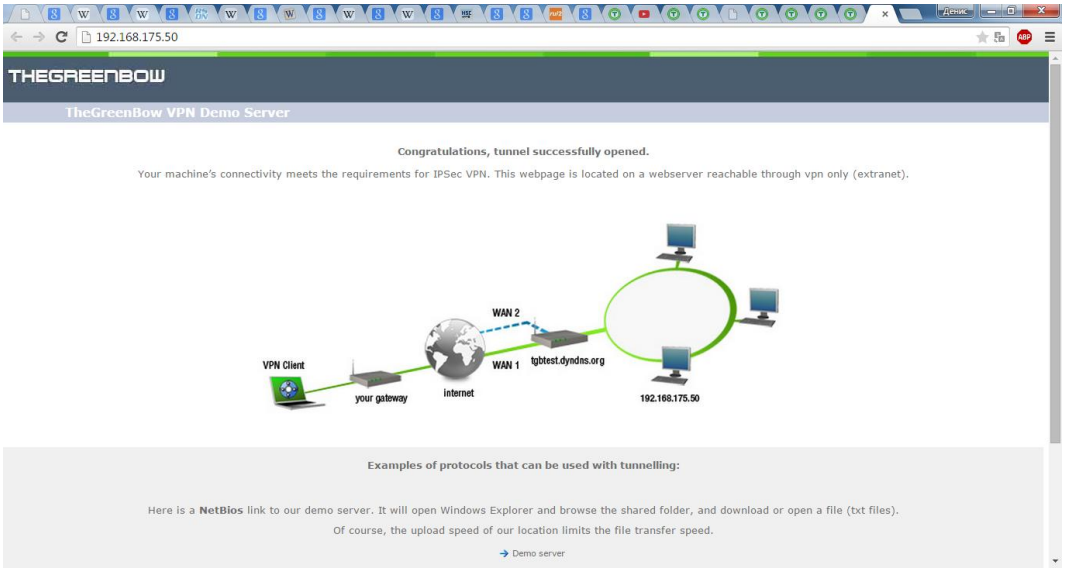


Рис. 1.41. Окно в браузере об успешном соединении

Выводы

Протокол безопасности транспортного уровня обеспечивает услуги безопасности

из конца в конец для приложений, которые пользуются протоколами

транспортного уровня, такими как, например, TCP. На сегодняшний день преобладает применение двух протоколов: *протокол "Уровень Безопасных Розеток" (SSL - Secure Sockets Layer)* и *протокол "Безопасность Транспортного уровня"* (TLS - Transport Layer Security).

- SSL или TLS обеспечивают такие услуги, как фрагментация, сжатие, целостность

сообщения, конфиденциальность и создание кадра данных, полученных от прикладного уровня. Как правило, SSL (или TLS) может получить прикладные

данные от любого протокола прикладного уровня, но работает протокол обычно с *HTTP*.

- Комбинация алгоритмов смены ключей, хэширования и алгоритм шифрования

определяют *набор шифров* для каждого сеанса.

- Для того чтобы обмениваться заверенными и конфиденциальными сообщениями,

клиенту и серверу необходимо иметь шесть единиц криптографической секретности (четыре ключа и два вектора инициализации).

- В SSL (или TLS) отличают *подключение* и сеанс. В сеансе одна сторона играет

роль клиента, а другая - роль сервера; при *подключении* обе стороны играют одинаковые роли, на равном подуровне.

- SSL (или TLS) определяет четыре протокола на двух уровнях: *протокол установления соединения*, *протокол изменения параметров шифрования, аварийный протокол* и *протокол передачи записей*. *Протокол установления соединения* использует

несколько сообщений, чтобы договориться о *наборе шифров*, подтвердить

подлинность сервера для клиента и клиента для сервера, если это необходимо, и обмениваться информацией для организации криптографической секретности.

Протокол изменения параметров шифрования определяет процесс перемещения информации между состоянием ожидания и активным состоянием. *Аварийный протокол* передает извещения об ошибках и ситуациях, отклоняющихся от нормальных.

Протокол передачи записей доставляет сообщения от верхнего уровня (протокол установления соединения, аварийный протокол, ChangeCipherSpec-протокол) или прикладного уровня.

1.3. Задание на самостоятельную работу по криптографическим методам защиты информации

PGP кодирование и шифрование с открытым ключом

PGP (англ. Pretty Good Privacy) — компьютерная программа, также библиотека функций, позволяющая выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации, представленной в электронном виде, в том числе прозрачное шифрование данных на запоминающих устройствах, например, на жёстком диске.

Первоначально разработана Филиппом Циммерманном в 1991 году.

Общие сведения

PGP имеет множество реализаций, совместимых между собой и рядом других программ (GnuPG, FileCrypt и др.) благодаря стандарту OpenPGP (RFC 4880), но имеющих разный набор функциональных возможностей. Существуют реализации PGP для всех наиболее распространённых операционных систем. Кроме свободно распространяемых реализаций есть еще и коммерческие.

Совместимость

Так как PGP развивается, некоторые системы позволяют создавать зашифрованные сообщения с использованием новых возможностей, которые отсутствуют в старых системах. Отправитель и получатель должны знать возможности друг друга или, по крайней мере, согласовать настройки PGP.

Защищённость

В 1996 году криптограф Брюс Шнайер охарактеризовал раннюю версию PGP как «ближайшую к криптосистемам военного уровня». На данный момент не известно ни одного способа взлома данных, зашифрованных PGP, при помощи полного перебора или уязвимости криптоалгоритма. Ранние версии PGP обладали теоретическими уязвимостями, поэтому рекомендуется пользоваться современными версиями.

Криптографическая стойкость PGP основана на предположении, что используемые алгоритмы устойчивы к криптоанализу на современном оборудовании. Например, в PGP первых версий для шифрования ключей сессии использовался алгоритм RSA, основанный на односторонней функции (факторизация). В PGP версии 2 дополнительно можно использовать алгоритм IDEA. В последующем были добавлены дополнительные алгоритмы шифрования. Ни у одного используемого алгоритма нет известных уязвимостей.

В 2010 году группе учёных из Швейцарии, Японии, Франции, Нидерландов, Германии и США удалось декодировать данные, зашифрованные по алгоритму RSA при помощи ключа длиной 768 бит. Нахождение простых сомножителей осуществлялось общим методом решета числового поля. На первый шаг (выбор пары полиномов степени 6 и 1) было потрачено около полугода вычислений на 80 процессорах, что составило около 3 % времени, потраченного на главный этап алгоритма (просеивание), который выполнялся на сотнях компьютеров в течение почти двух лет. Если интерполировать это время на работу одного процессора AMD Opteron 2.2ГГц с 2Гб памяти, то получилось бы порядка 1500 лет. Обработка данных после просеивания для следующего ресурсоёмкого шага (линейной алгебры)

потребовала несколько недель на малом количестве процессоров. Заключительный шаг после нахождения нетривиальных решений ОСЛУ занял не более 12 часов.

Решение ОСЛУ проводилось с помощью метода Видемана на нескольких раздельных кластерах и длилось чуть менее 4 месяцев. При этом размер разреженной матрицы составил 192 796 550x192 795 550 при наличии 27 795 115 920 ненулевых элементов. Для хранения матрицы на жёстком диске понадобилось около 105 гигабайт. В то же время понадобилось около 5 терабайт сжатых данных для построения данной матрицы.

В итоге группе удалось вычислить 232-цифровой ключ, открывающий доступ к зашифрованным данным.

Исследователи уверены, что с использованием их метода факторизации взломать 1024-битный RSA-ключ будет возможно в течение следующего десятилетия.

По словам исследователей, после их работы в качестве надежной системы шифрования можно рассматривать только RSA-ключи длиной 1024 бита и более. Причём от шифрования ключом длиной в 1024 бит стоит отказаться в ближайшие три-четыре года.

Зная разложение модуля на произведение двух простых чисел, противник может легко найти секретную экспоненту и тем самым взломать RSA. Однако на сегодняшний день самый быстрый алгоритм факторизации — решето обобщённого числового поля (General Number Field Sieve), скорость которого для k-битного целого числа составляет $\exp((c+o(1))k^{\frac{1}{3}}\log^{\frac{2}{3}}k)$ для некоторого $c<2$, не позволяет разложить большое целое за приемлемое время.

Механизм работы PGP

Шифрование PGP осуществляется последовательно хешированием, сжатием данных, шифрованием с симметричным ключом, и, наконец, шифрованием с открытым ключом, причём каждый этап может

осуществляться одним из нескольких поддерживаемых алгоритмов. Симметричное шифрование производится с использованием одного из семи симметричных алгоритмов (AES, CAST5, 3DES, IDEA, Twofish, Blowfish, Camellia) на сеансовом ключе. Сеансовый ключ генерируется с использованием криптографически стойкого генератора псевдослучайных чисел. Сеансовый ключ зашифровывается открытым ключом получателя с использованием алгоритмов RSAили Elgamal (в зависимости от типа ключа получателя). Каждый открытый ключ соответствует имени пользователя или адресу электронной почты. Первая версия системы называлась Сеть Доверия и противопоставлялась системе X.509, использовавшей иерархический подход, основанной на удостоверяющих центрах, добавленный в PGP позже. Современные версии PGP включают оба способа.

Ключи

Пользователь PGP создаёт ключевую пару: открытый и закрытый ключ. При генерации ключей задаются их владелец (имя и адрес электронной почты), тип ключа, длина ключа и срок его действия. Открытый ключ используется для шифрования и проверки цифровой подписи. Закрытый ключ - для декодирования и создания цифровой подписи.

PGP поддерживает три типа ключей RSA v4, RSA legacy (v3) и Diffie-Hellman/DSS (Elgamal в терминологии GnuPG).

Для ключей RSA legacy длина ключа может составлять от 1024 до 2048 бит, а для Diffie-Hellman/DSS и RSA — от 1024 до 4096. Ключи RSA legacy содержат одну ключевую пару, а ключи Diffie-Hellman/DSS и RSA могут содержать один главный ключ и дополнительные ключи для шифрования. При этом ключ электронной подписи в ключах Diffie-Hellman/DSS всегда имеет размер 1024. Срок действия для каждого из типов ключей может быть определён как неограниченный или до конкретной даты. Для защиты ключевого контейнера используется секретная фраза.

Цифровая подпись

PGP поддерживает аутентификацию и проверку целостности посредством цифровой подписи. По умолчанию она используется совместно с шифрованием, но также может быть применена и к открытому тексту. Отправитель использует PGP для создания подписи алгоритмом RSA или DSA. При этом сначала создаётся хеш открытого текста (также известный как дайджест), затем — цифровая подпись хеша при помощи закрытого ключа отправителя. Для формирования хеша могут использоваться алгоритмы MD5, SHA-1, RIPEMD-160, SHA-256, SHA-384, SHA-512. В новых версиях PGP поддержка MD5 осуществляется для сохранения совместимости с ранними версиями. Для подписи используются алгоритмы RSA или DSA (в зависимости от типа ключа).

Сжатие данных

В целях уменьшения объёма сообщений и файлов и, возможно, для затруднения криптоанализа PGP производит сжатие данных перед шифрованием. Сжатие производится по одному из алгоритмов ZIP, ZLIB, BZIP2. Для сжатых, коротких и слабосжимаемых файлов сжатие не выполняется.

Сеть доверия

Как при шифровании сообщений, так и при проверке цифровой подписи, необходимо, чтобы принятый получателем открытый ключ действительно принадлежал отправителю. При простом скачивании открытого ключа он может быть подменён. С первых версий PGP поддерживает сертификаты открытых ключей, с помощью которых подмены (или случайные ошибки передачи) легко распознаются. Однако недостаточно просто создать сертификат, защищённый от модификации, так как при этом гарантируется лишь целостность сертификата после его создания. Пользователи также должны каким-нибудь способом проверить, что открытый ключ в сертификате действительно принадлежит отправителю. С первых версий продукты PGP включают в себя внутреннюю схему проверки сертификатов, названную сеть доверия. Заданная пара «имя пользователя —

открытый ключ» может быть подписана третьим лицом, удостоверяющим соответствие ключа и владельца. В таких подписях может быть несколько вложенных уровней доверия. Хотя многие программы читают и пишут эту информацию, очень немногие учитывают этот уровень сертификата, принимая решение о принятии или отклонении сертификата.

Протокол сети доверия был впервые описан Циммерманном в 1992 году в руководстве PGP версии 2.0: «С течением времени вы будете накапливать ключи других людей, которых вы можете назвать доверенными рекомендателями. Кто-нибудь ещё может выбрать своих доверительных рекомендателей. И все будут постепенно накапливать и распространять со своими ключами набор заверенных подписей других людей, ожидая, что любой получатель доверяет по крайней мере одной или двум подписям. Это позволяет создать децентрализованную устойчивую к сбоям сеть всех открытых ключей.»

Механизм сети доверия обладает преимуществами над централизованной инфраструктурой управления открытыми ключами, например, используемой в S/MIME, но не получил повсеместного применения. Пользователи хотели проверять корректность сертификатов вручную или не проверять вовсе.

Сертификаты

В последних спецификациях OpenPGP доверенные подписи могут использоваться для поддержки создания центров сертификации. Доверенность сертификата означает, что ключ действительно принадлежит указанному владельцу и может использоваться для подписи сертификатов одним уровнем ниже. Сертификат уровня 0 означает обычную подпись. Уровень 1 означает, что при помощи подписанного ключа можно создавать сертификаты уровня 0. При помощи сертификата уровня 2 можно создавать сертификаты уровня 1. Уровень 2 практически идентичен степени доверия, с которой полагаются пользователи на списки доверенных сертификатов, встроенные в браузеры.

Все версии PGP включают в себя способ отмены сертификата. Это необходимо, если требуется сохранять безопасность связи при потере или компрометации закрытого ключа. Отмена сертификата похожа на списки отзыва сертификатов в централизованной инфраструктуре открытых ключей. Современные версии PGP также поддерживают сроки истечения сертификатов.

Проблема корректного определения принадлежности открытого ключа владельцу характерна для всех криптографических систем с асимметричным шифрованием. У неё не существует достаточно хороших решений. Оригинальная схема PGP позволяет решить пользователю, использовать ли схему проверки сертификатов, в то время как большинство других инфраструктур открытых ключей требуют проверки каждого сертификата.

История

В 1991 году Филипп Циммерман создал первую версию PGP. Первая версия включала в себя симметричный алгоритм шифрования BassOmatic, созданный самим Циммерманом. Циммерман участвовал в движении против ядерной энергии и создал PGP для защищённого использования BBS и хранения файлов и сообщений. Для некоммерческого использования не требовалось лицензии, со всеми копиями распространялся весь исходный код. PGP распространилась в Usenet, а затем и в Интернете.

Уголовное расследование

Вскоре после выпуска PGP стала использоваться за пределами США, и в 1993 году правительство США начало расследование против Циммермана по подозрению в нарушении экспортного законодательства, которое регулирует распространение криптографических систем с длиной ключа более 40 бит. В PGP использовались ключи длиной 128 бит и более.

Циммерман остроумно обошёл ограничения законодательства США. Он опубликовал исходный код в книге, изданной MIT Press. Код можно было сосканировать, распознать и скомпилировать. Экспорт книг не может быть запрещён, так как защищён первой поправкой к Конституции США.

OpenPGP

PGP Inc. была обеспокоена по поводу патентов. В компании был создан внутренний стандарт Unencumbered PGP («необременённый PGP»), не использующий алгоритмы, имеющие проблемы с лицензиями. Так как PGP широко использовалась во всём мире, многие хотели создавать собственное ПО, совместимое с PGP 5. В 1997 году PGP Inc. предложила IETF стандарт, названный OpenPGP. В IETF был созданы стандарты RFC 2440 (1998 год) и RFC 4880 (2007 год).

В 1999 году силами Фонда свободного программного обеспечения была создана свободная реализация OpenPGP под названием GNU Privacy Guard (GnuPG).

Поглощение Network Associates

В декабре 1997 года PGP Inc. была поглощена Network Associates Inc (ныне McAfee). NAI продолжила экспорт посредством печати исходных текстов. В составе NAI команда PGP разработала средства для шифрования дисков, брандмауэр, средства для обнаружения вторжений и IPsec VPN. После легализации экспорта криптографического ПО в 2000 году NAI прекратила публикацию исходных текстов, несмотря на возражения команды PGP.

В 2001 году Циммерман покинул NAI, NAI объявила о продаже PGP и остановке разработки PGP. В 2002 году NAI прекратила поддержку всех продуктов PGP PGP E-Business Server (исходной консольной версии PGP).

Современное состояние

В 2002 году несколько бывших разработчиков PGP основали PGP Corporation и выкупили PGP (кроме консольной версии). В 2003 году PGP Corporation разработала новый серверный продукт, PGP Universal.

В 2010-м году Symantec Corp. выкупил PGP за 300 млн. долларов.

Криптографические приложения PGP Corporation

PGP изначально разрабатывалась для шифрования электронной почты на стороне клиента, но с 2002 года включает также шифрование жёстких дисков

переносных компьютеров, файлов и директорий, сессий программ мгновенного обмена сообщениям, пакетной передачи файлов, защиту файлов и директорий в сетевых хранилищах, а в современных версиях — ещё и шифрование HTTP-запросов и ответов на стороне сервера (mod openpgp) и клиента (Enigform).

Клиентские программы объединены в семейство PGP Desktop (включает в себя PGP Desktop EMail, PGP Whole Disk Encryption и PGP NetShare).

PGP Universal Server позволяет из командной строки централизованно администрировать клиенты на основе PGP Desktop.

В 2010 году права на приложение были приобретены компаний Symantec за 300 млн. долларов.

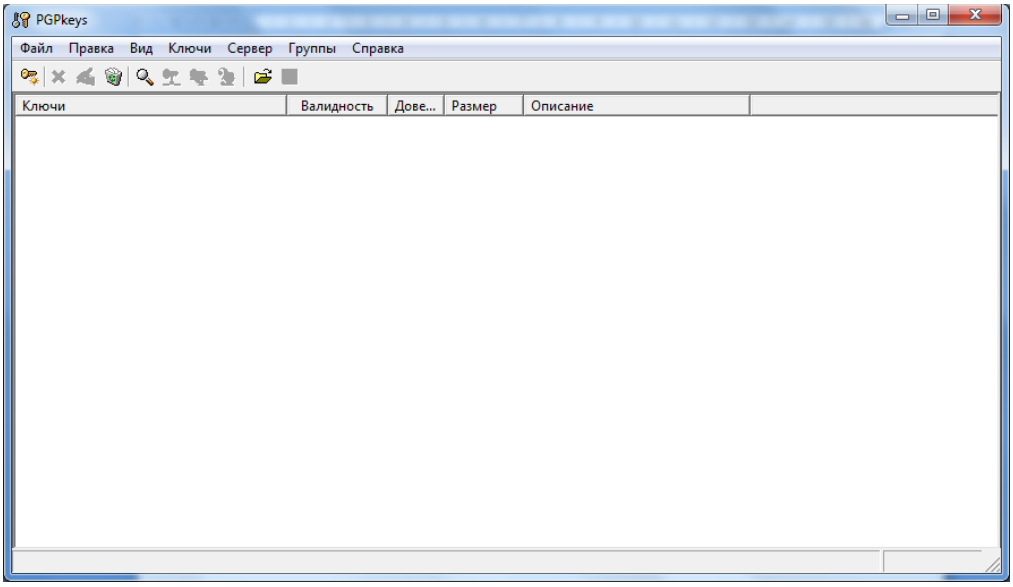
Правовые аспекты использования в России

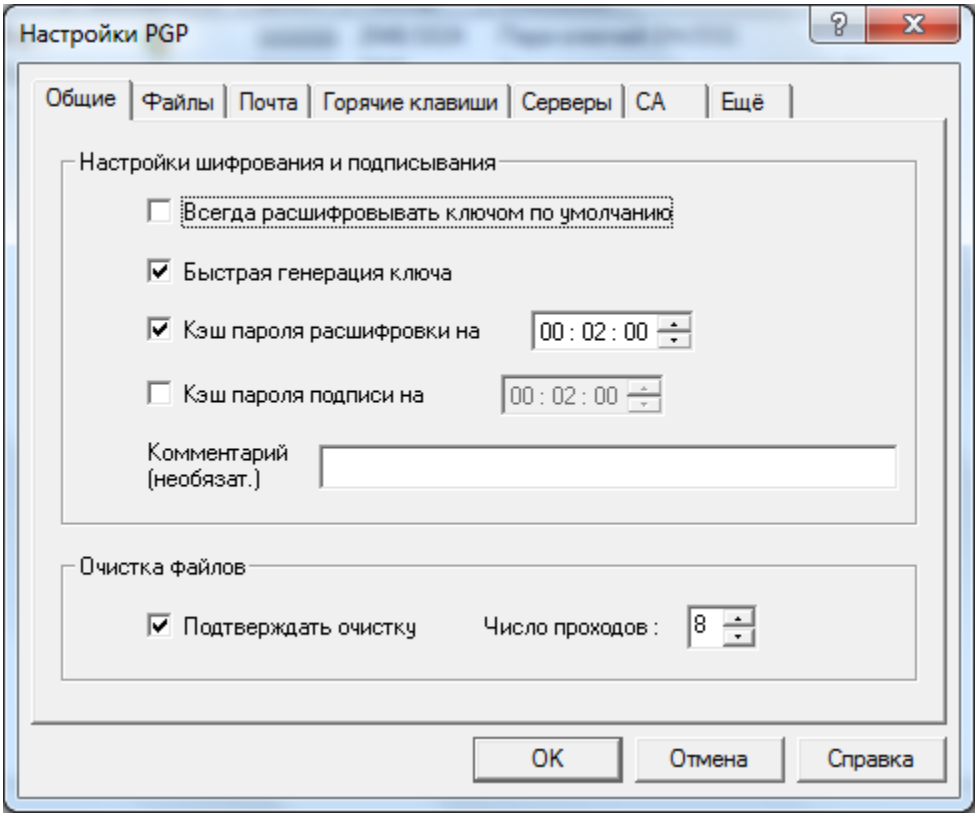
На сегодняшний день прямых законодательных запретов на использование PGP в России нет. Законодательно ограничивается использование криптографии только в государственных и муниципальных учреждениях. ФСБ предписывает всем государственным структурам использовать только сертифицированные средства криптографии. Физические лица и компании сами устанавливают какая информация является для них коммерческой тайной, методы хранения и передачи такой информации Закон «Об информации, информационных технологиях и защите информации» также указывает, что способ защиты информации, представляющей тайну, для негосударственных структур определяется оператором. Информационный ресурс Helpdesk24 в статье «Правомерность использования криптографических средств защиты информации» приводит выдержки из федеральных законов, поясняющие данный вопрос. Также авторы проекта «openPGP в России» утверждают, что не существует законов, запрещающих использование PGP . Указ от 3 апреля 1995 г. N 334 «О мерах по соблюдению законности в области разработки, производства, реализации и эксплуатации шифровальных средств» отменен. Электронная подпись, генерируемая с помощью PGP и её несертифицированных аналогов, имеет

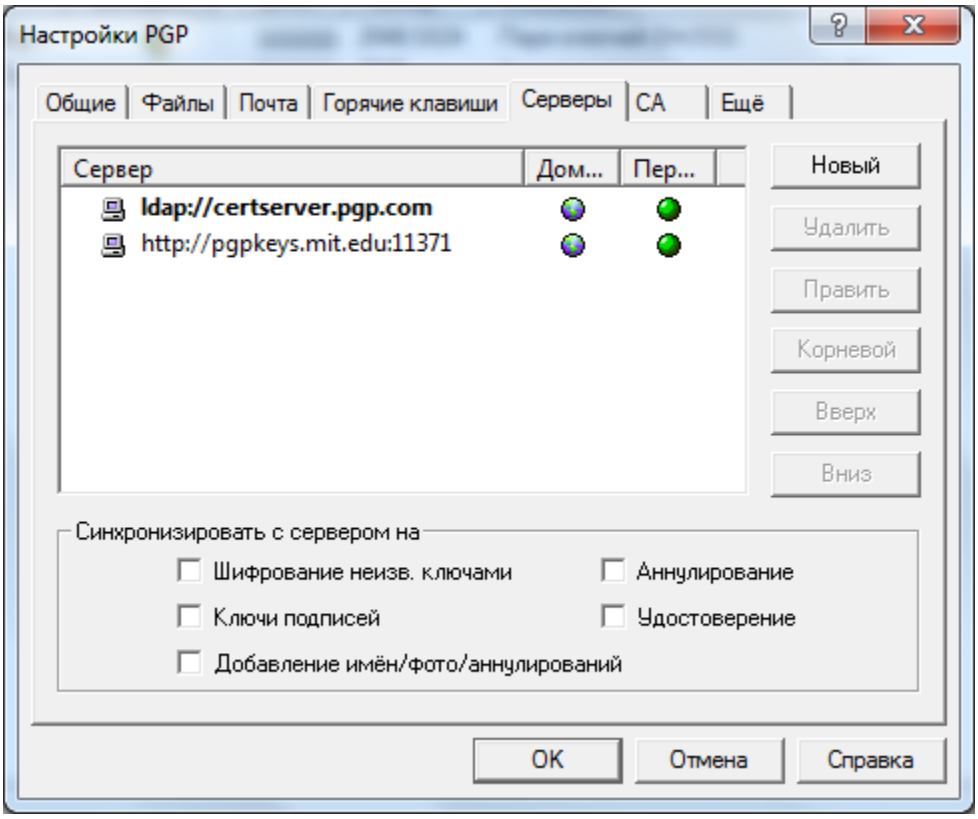
юридическую силу в Российской Федерации, т.к. согласно пункту 3 статьи 5 63-ФЗ "Об электронной подписи" попадает под определение усиленной неквалифицированной электронной подписи. Согласно пункту 2 статьи 6 этого ФЗ для признания такой ЭП необходимо соглашение между участниками электронного взаимодействия.

2.ХОД РАБОТЫ

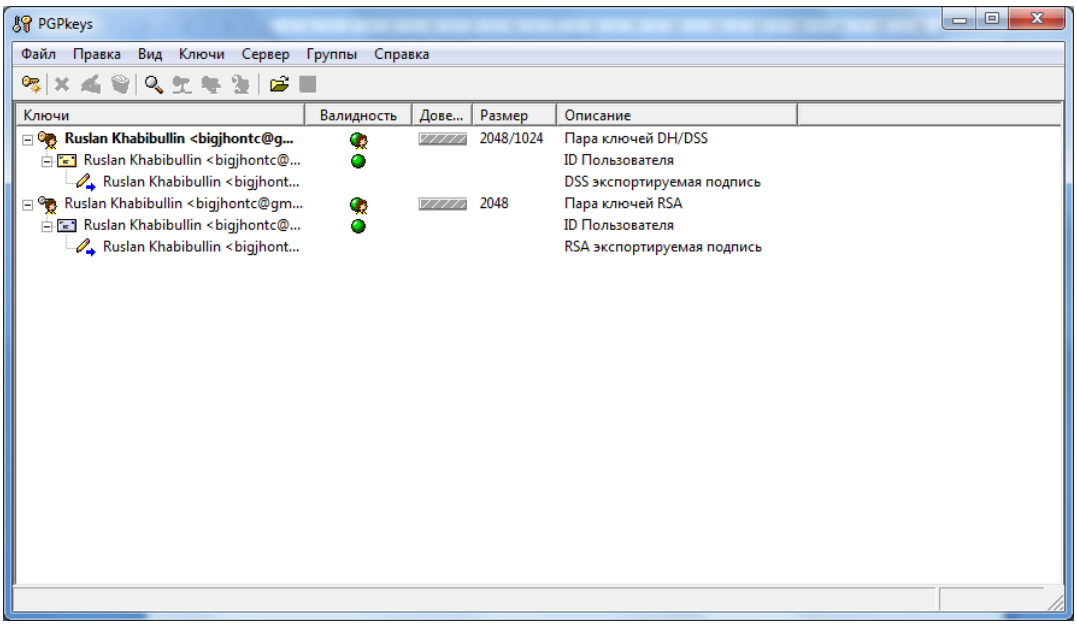
1. Изучите вкладки окна “Настройки PGP”.



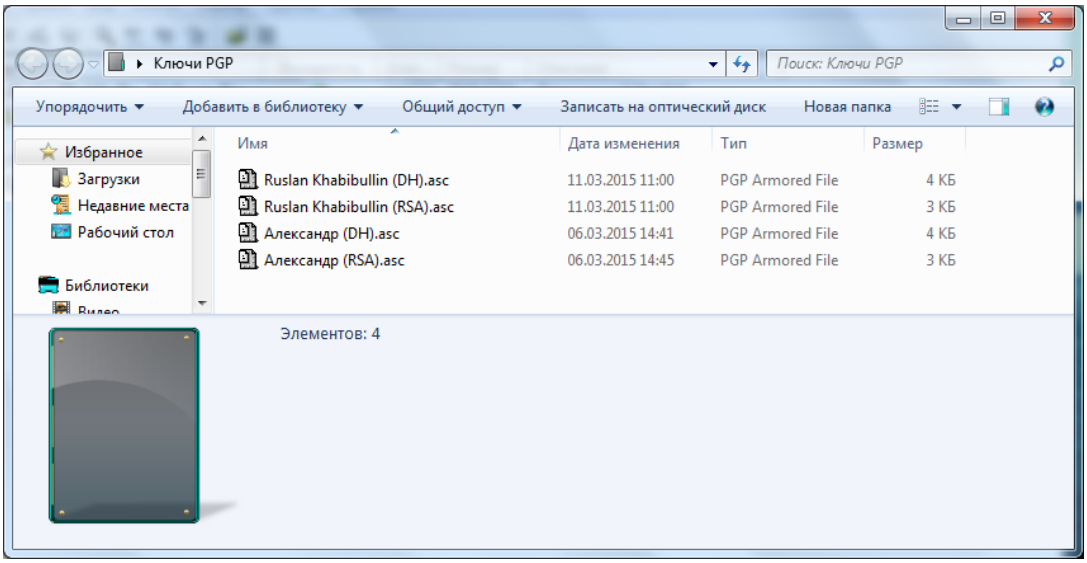




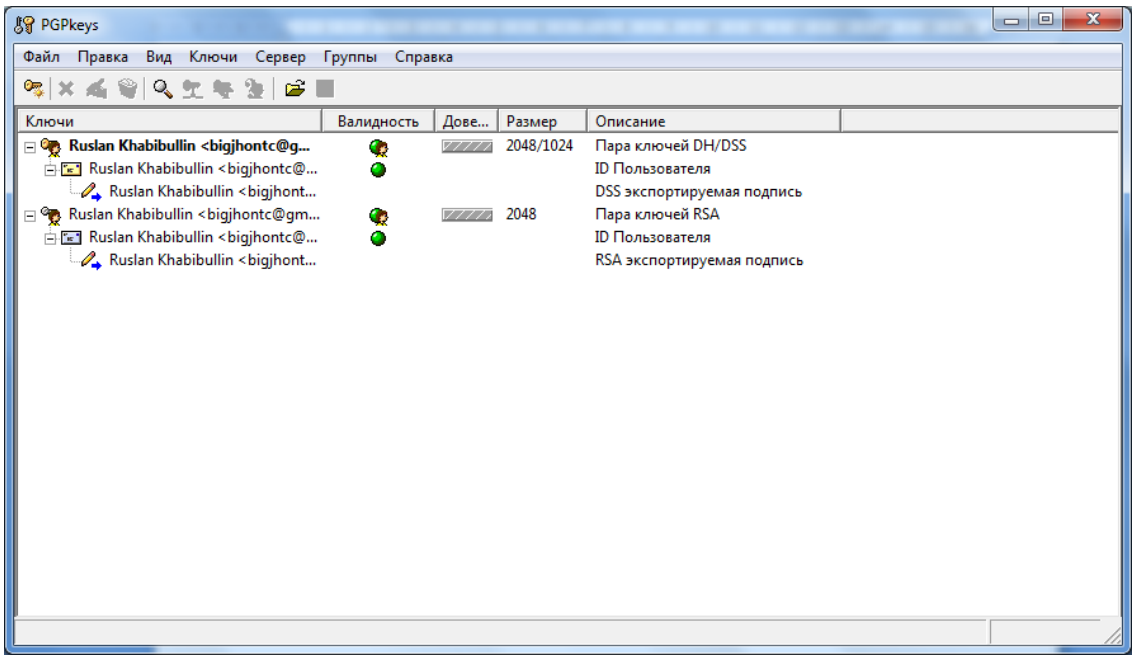
2. Создайте с помощью менеджера PGPkeys ключи шифрования двух типов: RSA и DH/DSS.



3. Сохраните полученные ключи (открытые и секретные) в отдельный файл.

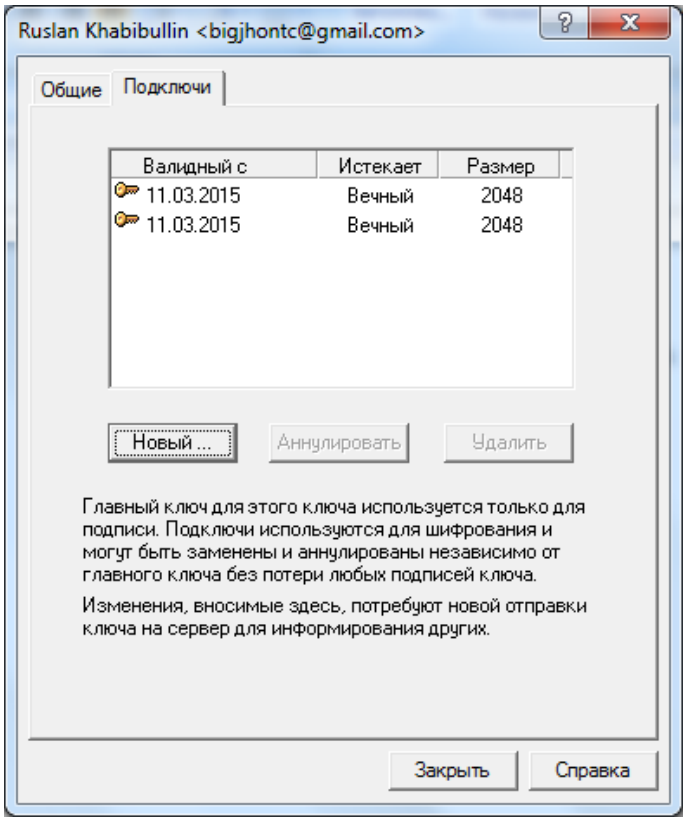


4. Назначьте ключ DH/DSS используемым по умолчанию.

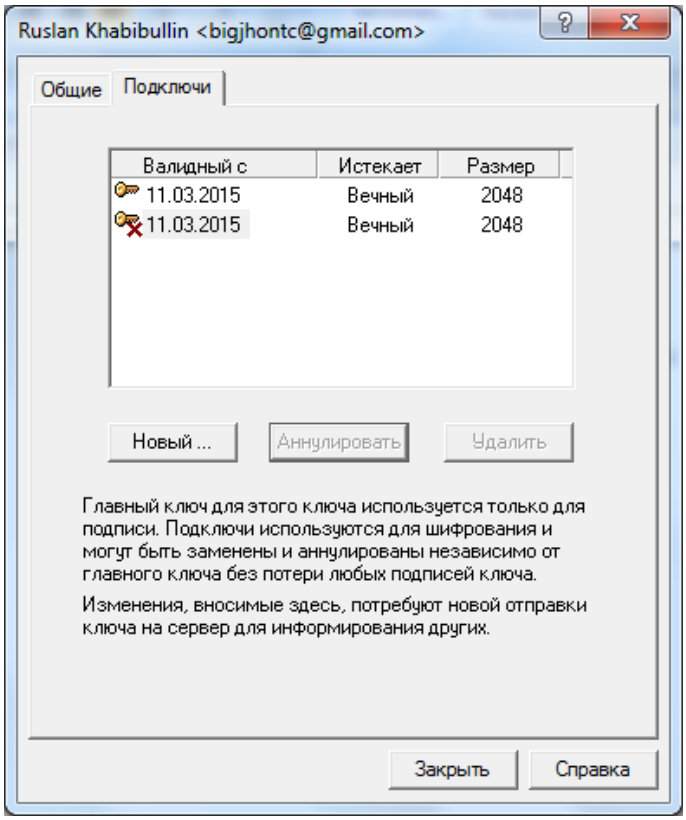


5. Изучите свойства ключевой пары DH/DSS.

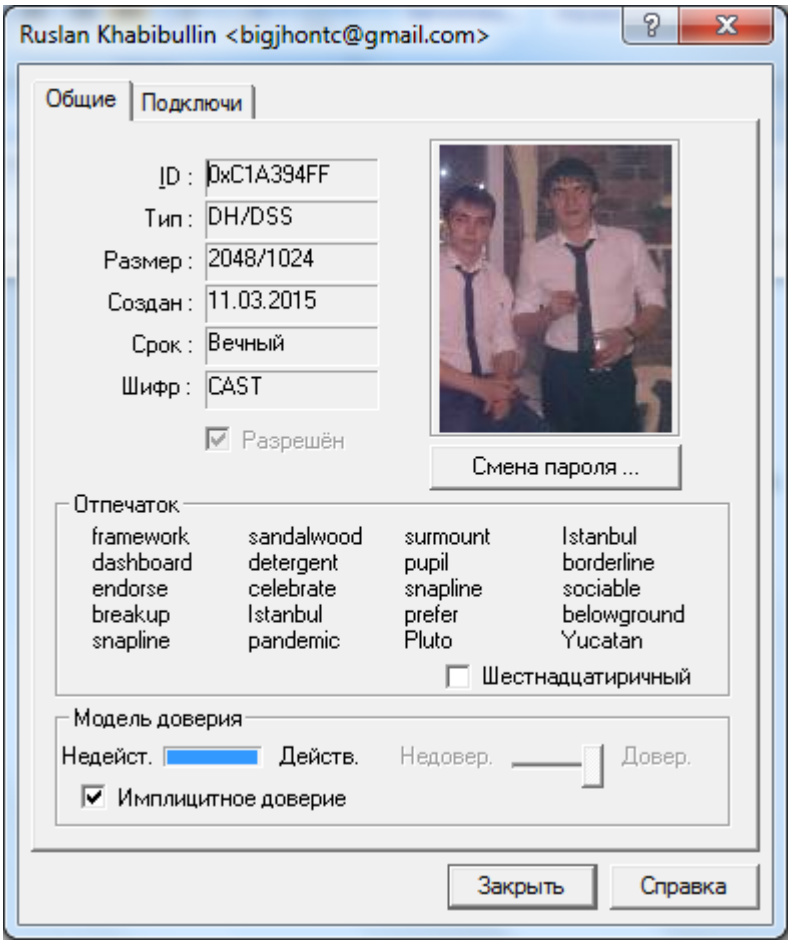
Создание подключа



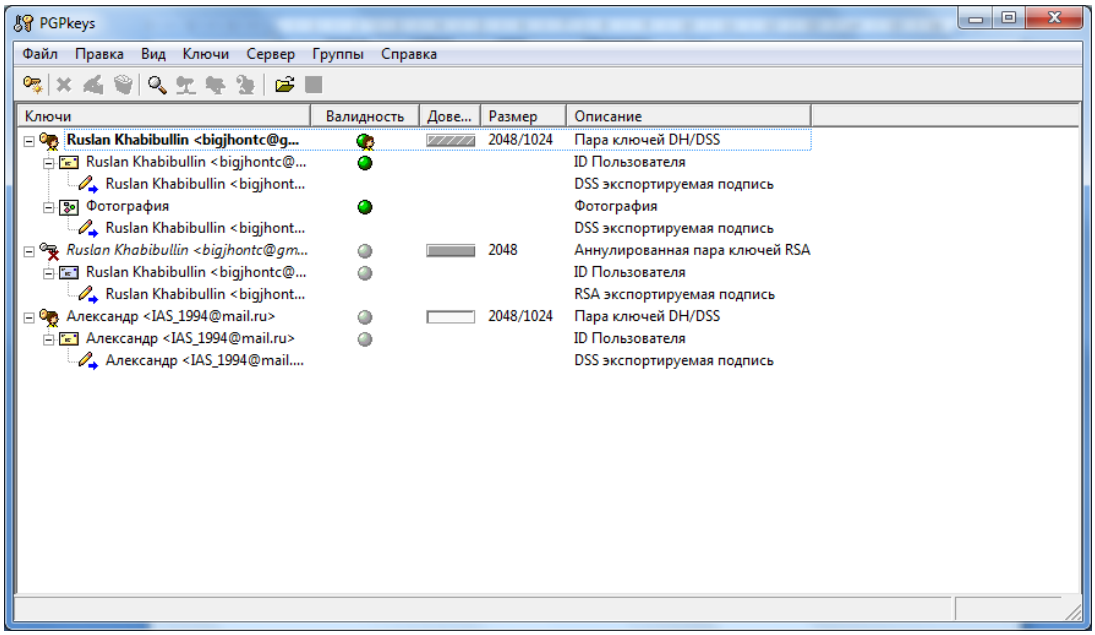
Аннулирование подключа



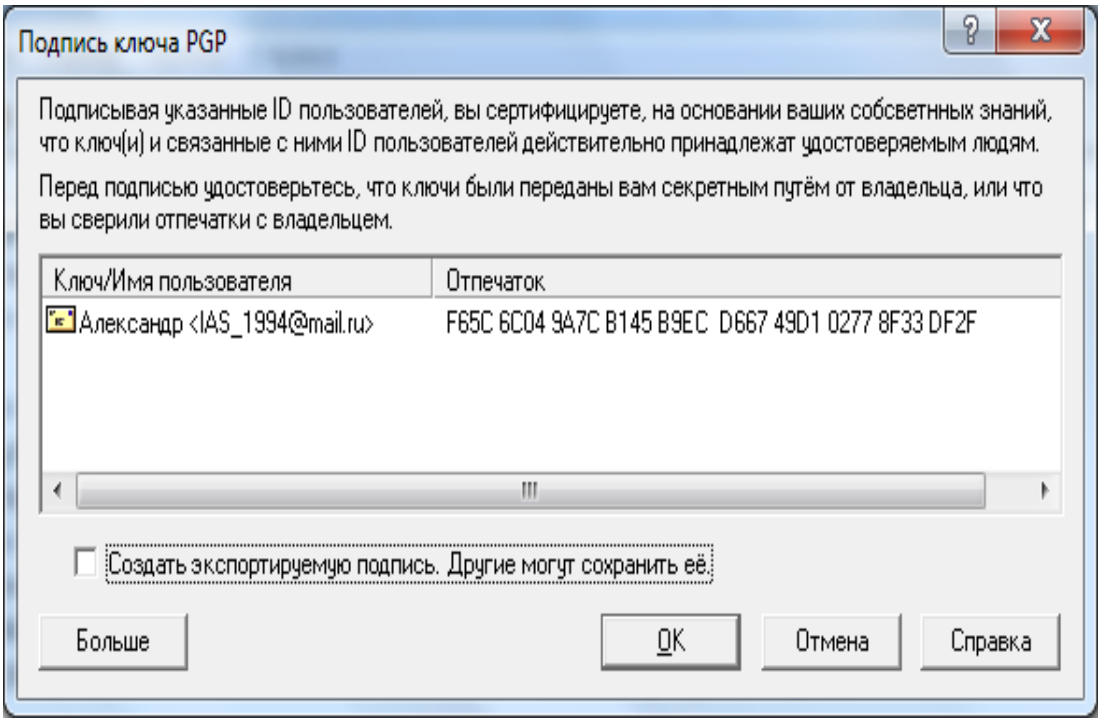
Добавление фотографии



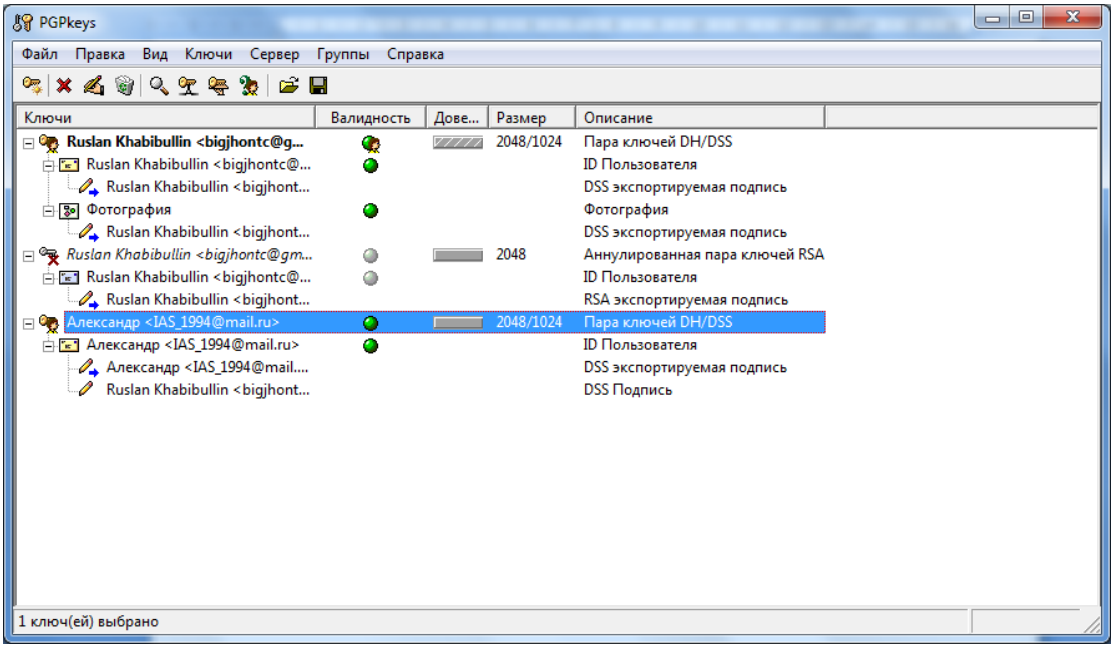
6. Обменяйтесь с другим студентом открытыми ключами DH/DSS.



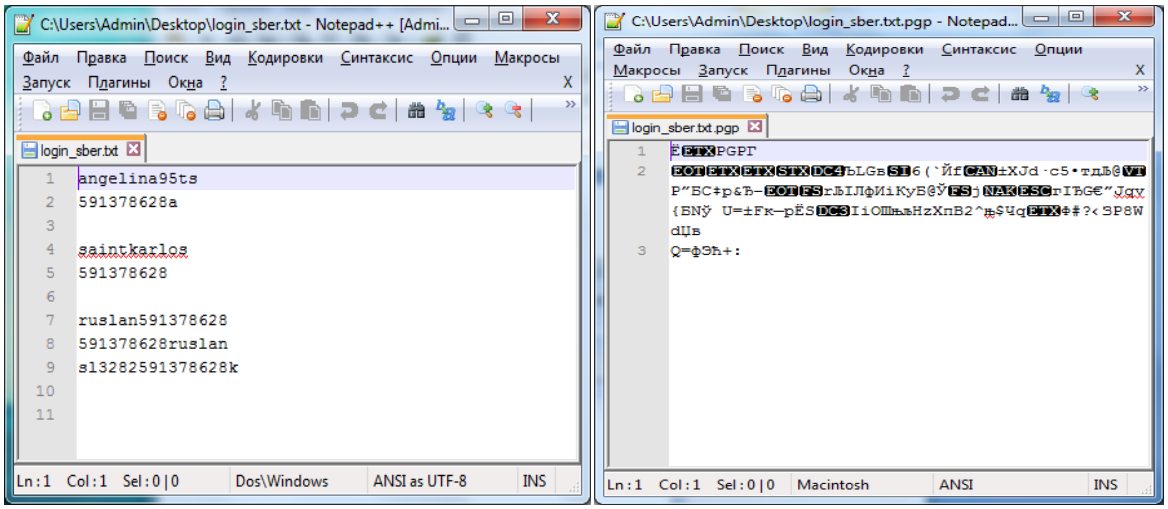
7. Установите подлинность полученного ключа с помощью его отпечатка.

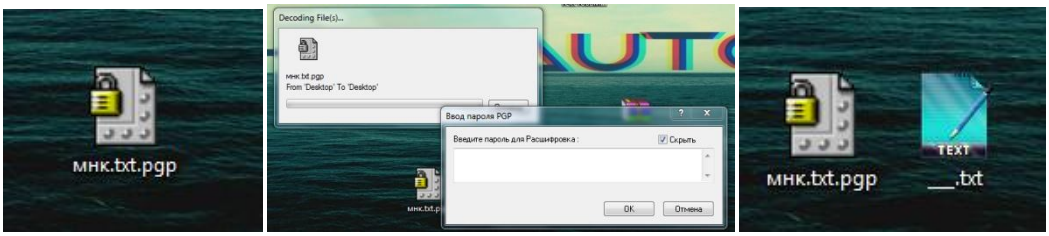


8. Установите степень доверия к владельцу полученного ключа на максимальный уровень.

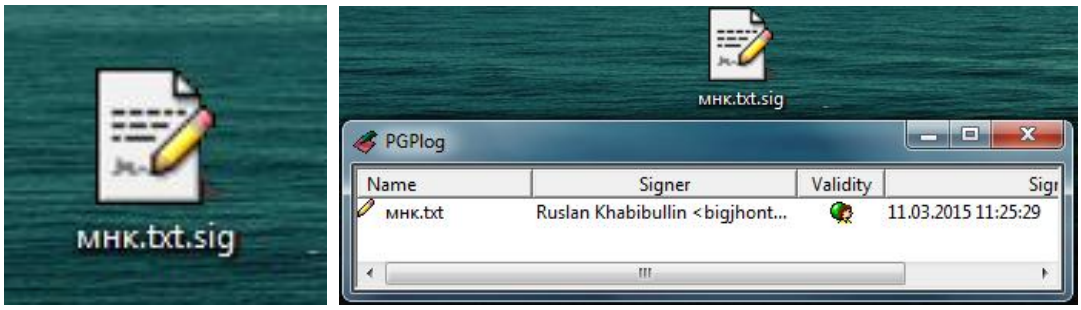


9. Зашифруйте произвольное сообщение/файл и обменяйтесь полученным результатом с другим студентом. Расшифруйте полученное сообщение.

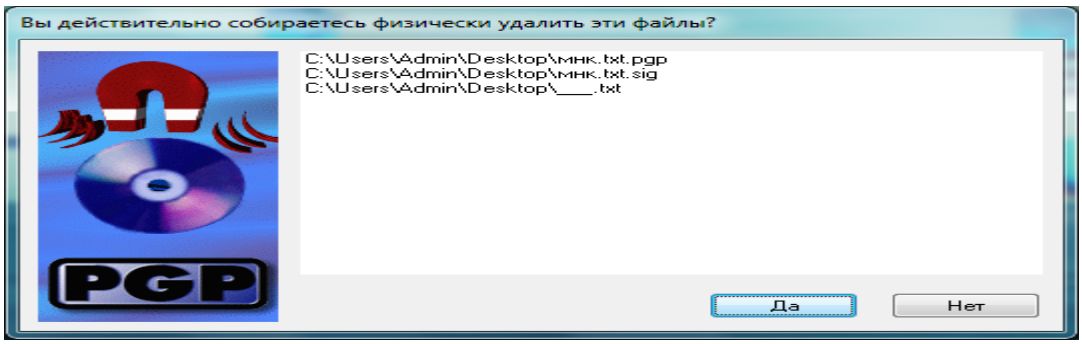




10. Подпишите произвольное сообщение/файл и обменяйтесь полученным результатом с другим студентом. Проверьте достоверность источников полученного сообщения.



11. Уничтожьте все ненужные файлы, используя утилиту PGP Wipe.



Контрольные вопросы.

1. Криптографическая система с открытым ключом (или асимметричное шифрование, асимметричный шифр) — система шифрования и/или электронной подписи (ЭП), при которой открытый ключ передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу и используется для проверки ЭП и для шифрования сообщения. Для генерации ЭП и для расшифровки сообщения используется закрытый ключ. Криптографические системы с открытым ключом в настоящее время широко применяются в различных сетевых протоколах, в частности, в протоколах TLS и его предшественнике SSL (лежащих в основе HTTPS), в SSH. Также используется в PGP, S/MIME.

2. Электронная подпись (ЭП), Электронная цифровая подпись (ЭЦП) — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий установить отсутствие искажения информации в электронном документе с момента формирования подписи и проверить принадлежность подписи владельцу сертификата ключа подписи.

3. Аннулирование – после аннулирования открытого ключа PGP синхронизирует его с сервером, дабы в дальнейшем ваши корреспонденты не могли его применять. **Дезактивация –** временное отключение неиспользуемого ключа или ключевой пары.

4. Отпечаток ключа — серия знаков, которой сопровождается каждый ключ. Сам по себе он не секретный, но уникальный. Если отпечаток того ключа, который вы получили (например) по электронной почте, и отпечаток, которым поделился ваш друг в Skype, совпадут, значит, у вас в руках правильный, настоящий ключ.

5. Имплицитное доверие – полное доверие зарезервировано для ключевых пар, расположенных на локальном ключе. Если одна часть

ключевой пары находится на вашей связке, PGP предполагает, что вы владелец пары ключей и что Вы без проблем сможете доверять себе.

П Е Р Е Ч Е Н Ь

заданий на самостоятельную работу

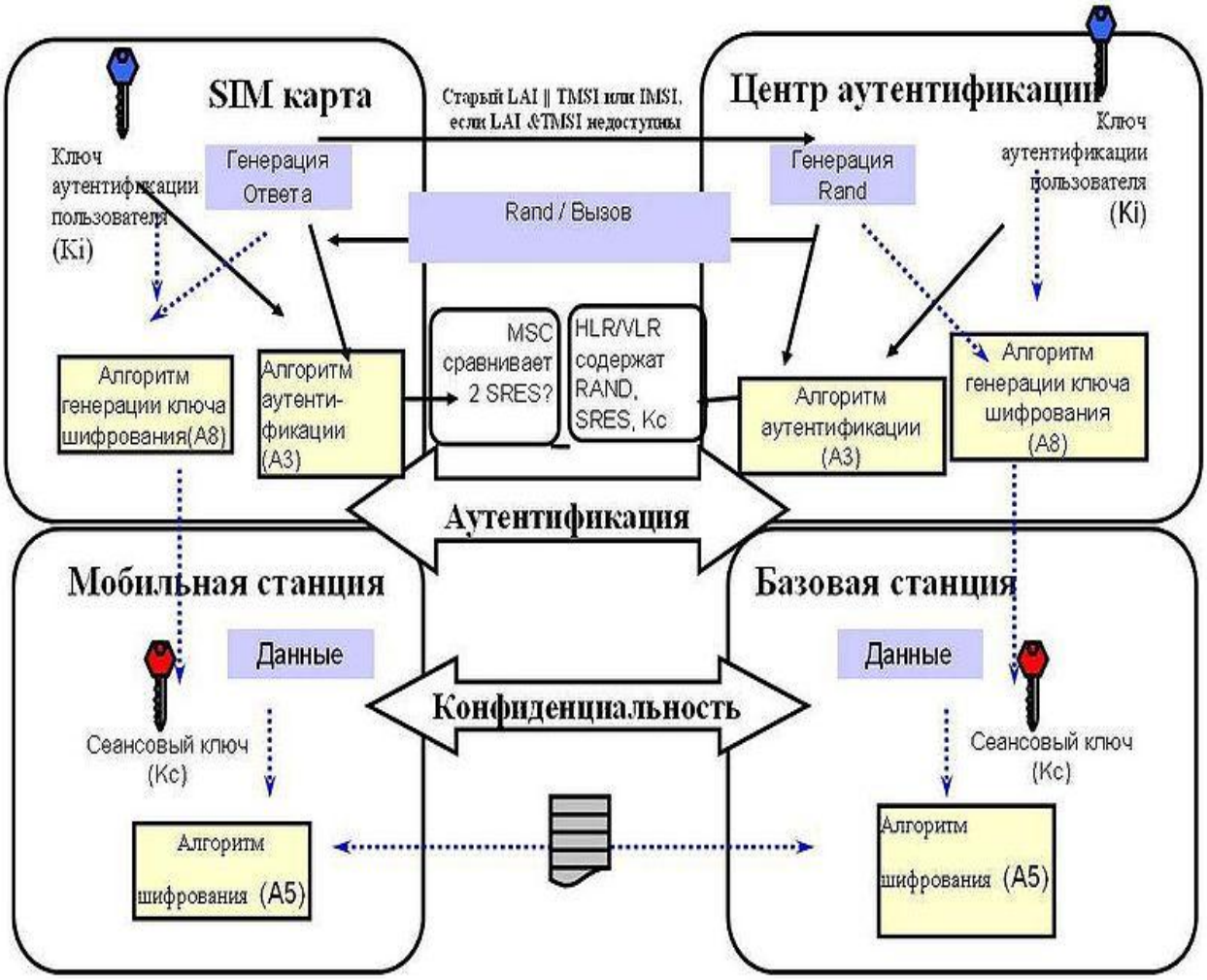
по курсу **«Криптографические методы защиты информации»**

№ п/п	Тема задания
1.	Разработка программного комплекса для исследования алгоритма ассиметричного шифрования Эль-Гамаля
2.	Разработка программного комплекса для исследования средств сжатия информации на базе вейвлет-фрактальных преобразований
3.	Российский алгоритм функции хэширования ГОСТ Р 34.11-94 и его программная реализация
4.	Методы оценки качества алгоритмов поточного шифрования и программная реализация статистических тестов НИСТ
5.	Алгоритм шифрования данных DES и его программная реализация
6.	Разработка программного комплекса защищенных платежных систем интернет коммерции на основе аппаратно-программного комплекса QNAP TS-259-PRO+
7.	Алгоритм ассиметричного шифрования Диффи-Хеллмана и его программная реализация
8.	Алгоритм шифрования данных AES и его программная реализация
9.	Разработка программного комплекса для исследования методов аналогового скремблирования на базе LabView
10.	Разработка системы обеспечения защищенного маршрутизируемого взаимодействия ViPNet OFFICE
11.	Разработка программного комплекса защищенного сайта интернет коммерции на основе аппаратно-программного комплекса QNAP TS-259-PRO+
12.	Новый российский стандарт ЭЦП ГОСТ Р 34.10-2001 и его программная реализация
13.	Алгоритм цифровой подписи RSA и его программная реализация
14.	Разработка программного комплекса для исследования ЭЦП в системе электронного документооборота (СЭД) ELMA
15.	Разработка программного комплекса для исследования защищенной системы спутниковой связи на базе Systemview 6.0 (SystemVue)
16.	Разработка программного комплекса защищенной системы хранения данных на базе программного обеспечения с открытым исходным кодом
17.	Разработка программного комплекса для создания криптовалюты Биткойн
18.	Разработка системы обеспечения защищенного маршрутизируемого взаимодействия ViPNet CUSTOM

2. МЕТОДЫ ШИФРОВАНИЕ В СОВРЕМЕННЫХ СИСТЕМАХ СВЯЗИ

2.1. **Безопасность GSM сетей**

Прежде чем приступить к описанию алгоритма шифрования используемого в GSM сетях рассмотрим каким образом происходит аутентификация пользователя и формирования ключа шифрования. Для этого воспользуемся картинкой.



На данном рисунке схематично представлены следующие шаги:

1. Телефон оператора подключается к сети.
2. Для подтверждения своей подлинности телефон посылает специальный идентификационный код, называемый TMSI(Temporary Mobile Subscriber Identity).

3. Центр Аутентификации(ЦА) генерирует 128-битное случайное число RAND и посылает его на Мобильную Станцию(МС).

4. МС зашифровывает полученное число RAND, используя свой секретный ключ K_i и алгоритм аутентификации A3.

5. МС берет первые 32 бита из последовательности, полученной на предыдущем шаге(назовем их SRES(signed response)) и отправляет их обратно на ЦА.

6. ЦА проделывает ту же операцию и получает 32 битную последовательность XRES(expected response).

7. После чего ЦА сравнивает SRES и XRES. В случае, если оба значения равны, телефон считается аутентифицированным.

8. МС и ЦА вычисляют сессионный ключ шифрования, используя секретный ключ K_i и алгоритм формирования ключа A8 $K_c=A8_{ki}(RAND)$

9. Говоря об алгоритмах аутентификации A3 и алгоритме формирования ключа A8, следует отметить что на практике большинство сотовых операторов используют для этих целей один алгоритм, называемый COMP128(он имеет множество модификаций COMP128-1, COMP128-2, COMP128-3). COMP128 представляет собой обыкновенную хэш-функцию, на входе которая принимает 128-битную последовательность и на выходе возвращает 96-битную.

Как всегда в криптографии, попытка сэкономить время разработчикам обернулась полным провалом. Безопасность GSM сетей изначально основывалась на принципе «безопасность за счёт неизвестности». И когда в 1998 году алгоритм был вскрыт группой исследователей состоящих из Marc Briceno, Ian Goldberg и David Wagner обранужилась одна занятная особенность: последние 10 бит секретного ключа K_i всегда равнялись нулю. Используя это любопытное свойство, а так же уязвимость COMP128 к «атаке дней рождений» Marc Briceno, Ian Goldberg и David Wagner смогли извлечь

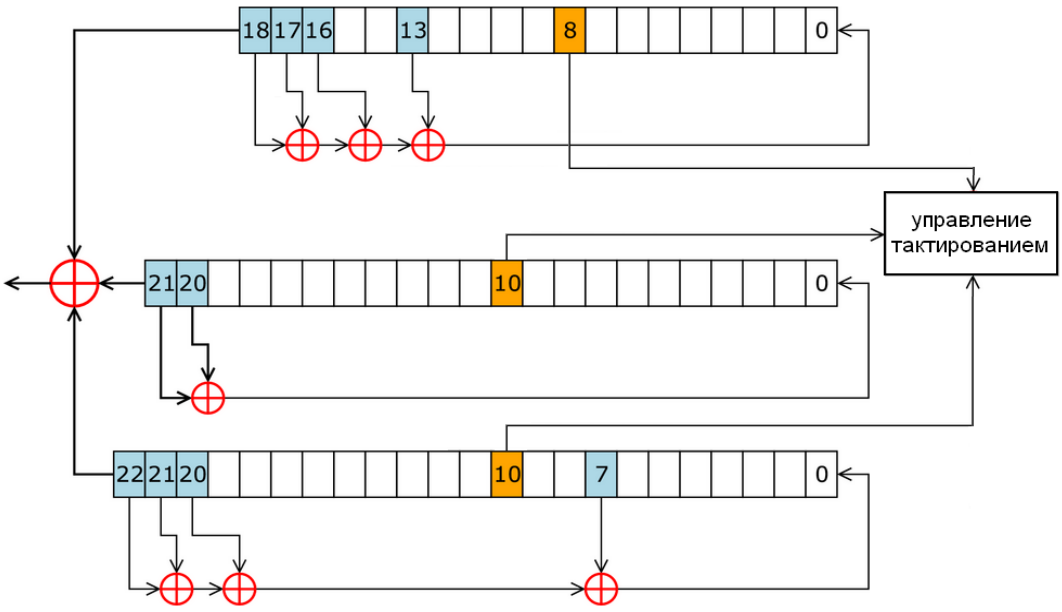
секретный ключ K_i из SIM-карты. Результатом этого исследования стал повсеместный отказ от алгоритма COMP128 и его замена на более надежные модификации COMP128-2 и COMP128-3, технические детали которых держатся в тайне.

Алгоритм шифрования A5/1

В качестве алгоритма шифрования в GSM используются алгоритмы из семейства A5. На сегодняшний день их всего 3:

- **A5/1** — поточный шифр, наиболее распространенный на сегодня.
- **A5/2**-вариант предыдущего алгоритма «для бедных». Очень похож на своего «старшего брата», но изначально задумывался, как сильно ослабленная версия A5/1. В настоящее время не используется
- A5/3-блочный шифр. Разработан в 2002 году с целью заменить устаревший A5/1. Однако в настоящее время используется только в 3GPP сетях. У алгоритма найден ряд уязвимостей, но о практических атаках речи пока не идет.

Рассмотрим подробнее алгоритм A5/1



Внутреннее состояние шифра A5/1 состоит из трех линейных регистров сдвига с обратной связью R1, R2, R3, длиной 19, 22 и 23 бита соответственно(всего 64 бита).

Сдвиг в регистрах R1, R2, R3 происходит только при выполнении определенного условия. Каждый регистр содержит " бит управления тактированием". В R1 это 8-й бит, а в R2 и R3 — 10-й. На каждом шаге сдвигаются только те регистры у которых значение бита синхронизации равно большинству значений синхронизирующих битов всех трех регистров.

На сегодняшний день известно большое количество успешных атак на GSM шифрование и все они относятся к атакам типа known-plaintext, т.е. для восстановления ключа атакующему помимо зашифрованных фреймов необходимо знать так же незашифрованные данные, которые соответствуют этим фреймам. На первый взгляд такое требование может показаться фантастическим, однако из-за специфики стандарта GSM, в котором помимо голосового трафика передаются различные системные сообщения, такого рода атаки из разряда теоретических переходят в разряд практических.

Системные сообщения GSM содержат повторяющиеся данные и могут использоваться злоумышленником. В частности метод, предложенный Karsten Nohl в 2010 году основан как раз таки на поиске такого рода данных в шифротексте и простом переборе различных вариантов ключей, хранящихся в радужных таблицах, до тех пор пока не будет найден ключ, порождающий нужный шифротекст для известного заранее системного сообщения.

2.2. Криптографическая защита беспроводных сетей стандартов LTE

Стандарт сетей LTE [9] – стандарт беспроводной высокоскоростной передачи данных для мобильных телефонов и других терминалов, работающих с данными. Он основан на GSM/EDGE и UMTS/HSPA сетевых технологиях, увеличивая пропускную способность и скорость за счёт использования другого радиоинтерфейса вместе с улучшением ядра сети.

На рисунке 2.1 представлена структура сети стандарта LTE.

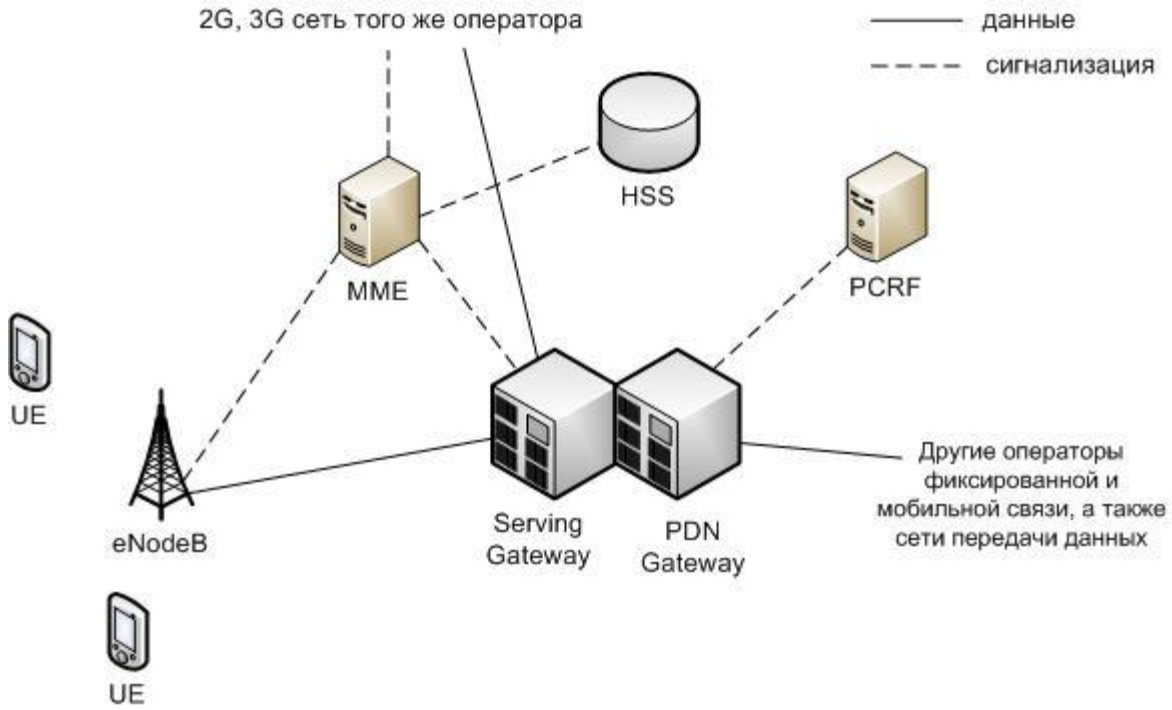


Рис. 2.1. Структура сети стандарта LTE

Из этой схемы видно, что структура сети сильно отличается от сетей стандартов 2G и 3G. Существенные изменения претерпела и подсистема базовых станций, и подсистема коммутации. Изменена технология передачи данных между оборудованием пользователя и базовой станцией. Также подверглись изменению и протоколы передачи данных между сетевыми элементами. Вся информация (голос, данные) передается в виде пакетов. Таким образом, уже нет разделения на части обрабатывающие либо только голосовую информацию, либо только пакетные данные.

Можно выделить следующие основные элементы сети стандарта LTE:

- **Serving SAE Gateway** или просто **Serving Gateway** *(SGW)* – обслуживающий шлюз сети LTE. Предназначен для обработки и маршрутизации пакетных данных поступающих из/в подсистему базовых станций. SGW имеет прямое соединение с сетями второго и третьего поколений того же оператора, что упрощает передачу соединения в /из них по причинам ухудшения зоны

покрытия, перегрузок и т.п. В SGW нет функции коммутации каналов для голосовых соединений, т.к. в LTE вся информация, включая голос коммутируется и передается с помощью пакетов.

- **Public Data Network SAE Gateway** или просто **PDN Gateway** *(PGW)* – шлюз к сетям передачи данных других операторов для сети LTE. Основная задача PGW заключается в маршрутизации трафика сети LTE к другим сетям передачи данных, таких как Интернет, а также сетям GSM, UMTS.
- **Mobility Management Entity** *(MME)* – узел управления мобильностью сети сотовой связи стандарта LTE. Предназначен для обработки сигнализации, преимущественно связанной с управлением мобильностью абонентов в сети.
- **Home Subscriber Server** *(HSS)* – сервер абонентских данных сети сотовой связи стандарта LTE. Представляет собой большую базу данных и предназначен для хранения данных об абонентах. Кроме того, HSS генерирует данные, необходимые для осуществления процедур шифрования, аутентификации и т.п. Сеть LTE может включать один или несколько HSS. Количество HSS зависит от географической структуры сети и числа абонентов.
- **Policy and Charging Rules Function** *(PCRF)* – элемент сети сотовой связи стандарта LTE, отвечающий за управление начислением платы за оказанные услуги связи, а также за качество соединений в соответствии с заданными конкретному абоненту характеристиками.

Для того чтобы данные могли быть транспортированы через интерфейс радио LTE, используются различные «каналы». Они используются для того, чтобы выделять различные типы данных и позволить им транспортироваться через сеть доступа более эффективно. Использование нескольких каналов

обеспечивает интерфейс более высокого уровня в рамках протокола LTE и включают более чёткую и определенную сегрегацию данных.

Есть три категории, в которые могут быть сгруппированы различные каналы передачи данных:

Логические каналы – предоставляет услуги среднего уровня управления доступом MAC *(Medium Access Control)* в пределах структуры протокола LTE. Логические каналы по типу передаваемой информации делятся на логические каналы управления и логические каналы трафика. Логические каналы управления используются для передачи различных сигнальных и информационных сообщений. По логическим каналам трафика передают пользовательские данные.

Транспортные каналы — транспортные каналы физического уровня предлагают передачу информации в MAC и выше. Информацию логических каналов после обработки на RLC/MAC уровнях размещают в транспортных каналах для дальнейшей передачи по радиоинтерфейсу в физических каналах. Транспортный канал определяет как и с какими характеристиками происходит передача информации по радиоинтерфейсу. Информационные сообщения на транспортном уровне разбивают на транспортные блоки. В каждом временном интервале передачи (Transmission Time Interval, TTI) по радиоинтерфейсу передают хотя бы один транспортный блок. При использовании технологии MIMO возможна передача до четырех блоков в одном TTI.

Физические каналы – это каналы передачи, которые переносят пользовательские данные и управляющие сообщения. Они изменяются между восходящим и нисходящим потоками, поскольку каждый из них имеет различные требования и действует по-своему.

Существующие методы и стандарты защиты беспроводных сетей LTE

Безопасность в сетях LTE заключается в нескольких видах:

- Защита абонентов.

- Защита передаваемых сообщений.
- Шифрование сообщений.
- Аутентификация и абонента, и сети.

Защита абонента заключается в том, что в процессе обслуживания его скрывают временными идентификаторами.

Для закрытия данных в сетях LTE используется потоковое шифрование методом наложения на открытую информацию псевдослучайной последовательности (ПСП) с помощью оператора XOR (исключающее или). В этих сетях для обеспечения безопасности внутри сети применяется принцип туннелирования соединений. Шифрации можно подвергать пакеты S1 и X2 при помощи IPsec ESP, а также подвергаются шифрации сигнальные сообщения этих интерфейсов.

В момент подключения или активизации абонентского оборудования (UE) в сети, сеть запускает процедуру аутентификации и соглашения о ключах AKA (Authentication and Key Agreement). Целью этой процедуры является взаимная аутентификация абонента и сети и выработка промежуточного ключа K_{ASME}. Работа механизма AKA занимает доли секунды, которые необходимы для выработки ключа в приложении USIM и для установления соединения с Центром регистрации (HSS). Вследствие этого, для достижения скорости передачи данных сетей LTE необходимо добавить функцию обновления ключевой информации без инициализации механизма AKA. Для решения этой проблемы в сетях LTE предлагается использовать иерархическую ключевую инфраструктуру. Здесь также, как и в сетях 3G, приложение USIM и Центр аутентификации (AuC) осуществляет предварительное распределение ключей. Когда механизм AKA инициализируется для осуществления двусторонней аутентификации пользователя и сети, генерируются ключ шифрования CK и ключ общей защиты, которые затем передаются из ПО USIM в Мобильное оборудование (ME) и из Центра аутентификации в Центр регистрации (HSS). ME и HSS, используя ключевую пару (CK;IK) и ID используемой сети, вырабатывает

ключ K_{ASME}. Установив зависимость ключа от ID сети, Центр регистрации гарантирует возможность использования ключа только в рамках этой сети. Далее K_{ASME} передается из Центра регистрации в устройство мобильного управления (MME) текущей сети, где он используется в качестве мастер-ключа. На основании K_{ASME} вырабатывается ключ $K_{nas\text{-}enc}$, который необходим для шифрования данных протокола NAS между мобильным устройством (UE) и MME, и $K_{nas\text{-}int}$, необходимый для защиты целостности. Когда UE подключается к сети, MME генерирует ключ KeNB и передает его базовым станциям. В свою очередь, из ключа KeNB вырабатывается ключ Kup-enc, используемый для шифрования пользовательских данных протокола U-Plane, ключ Krrc-enc для протокола RRC (Radio Resource Control - протокол взаимодействия между Мобильными устройствами и базовыми станциями) и ключ Krrc-int, предназначенный для защиты целостности.

Алгоритм аутентификации и генерации ключа представлен на рис 2.2

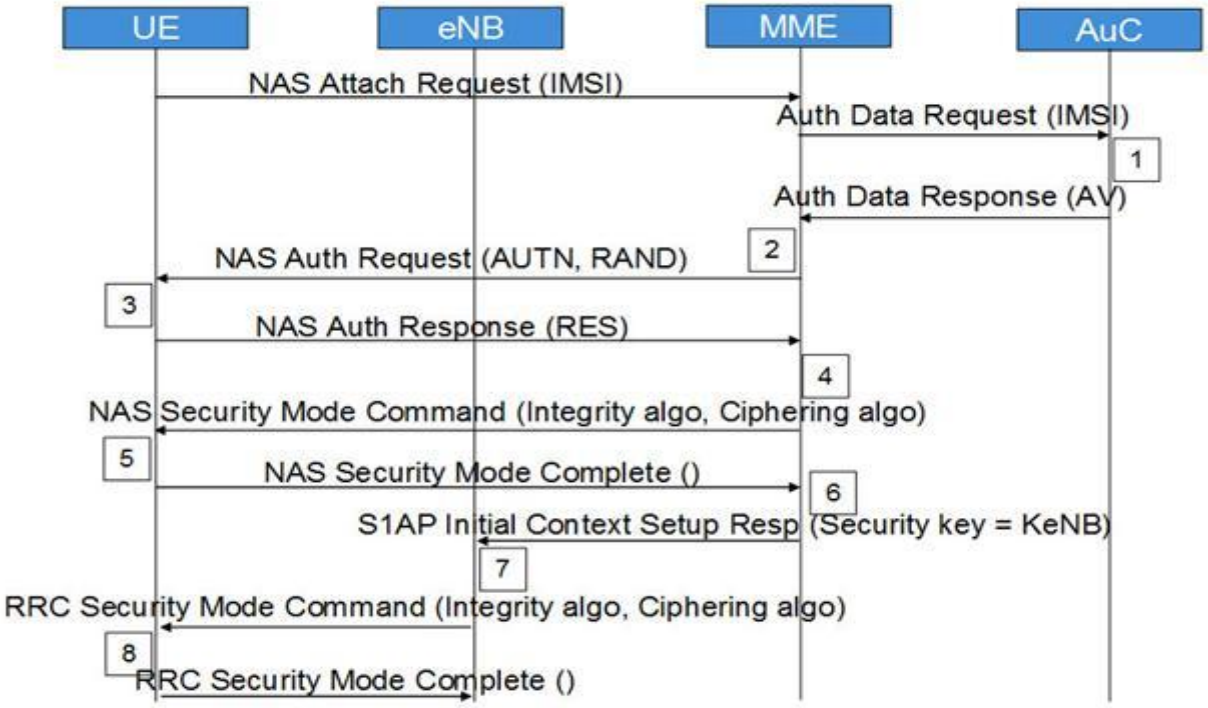


Рис. 2.2. Алгоритм аутентификации и генерации ключа

Здесь:

Шаг 1. Запрос о подключении к сети от мобильной станции (UE). MME запрашивает аутентификационные данные, относящиеся к конкретному IMSI, отправляя Authentication Data Request. AuC/HSS выбирает PSK, относящийся к конкретному IMSI и вычисляет аутентификационные данные по PSK. AuC/HSS отправляет обратно AV с Authentication Data Response.

Шаг 2. MME получает IK, CK, XRES, RAND и AUTH из AV. MME отправляет AUTH и RAND при помощи Authentication Request к UE.

Шаг 3. UE аутентифицирует NW, проверяя полученный AUTH. После чего вычисляет IK, CK, RES, XMAC из своего ключа защиты, AMF, (OP), AUTH и RAND. Она отправляет RES с Authentication response.

Шаг 4. После получения RES, MME сравнивает его с XRES и если они совпадают, то аутентификация прошла успешно, в противном случае, MME отправляет сбой аутентификации (Authentication failure) к UE. MME сбрасывает счетчик DL NAS. Рассчитывает KASME, KeNB, Knas-int, Knas-enc. Отправляет NAS команду режима безопасности (алгоритм целостности, алгоритм шифрования, NAS набор ключей ID, функцию безопасности UE) с целостностью охраняемых, но не зашифрованных, используя Knas-inc.

Шаг 5. После получения NAS команды режима безопасности, UE вычисляет KASME, KeNB, Knas-int, Knas-enc. UE отправляет NAS режима безопасности выполнен с целостностью, защищенных и зашифрованных.

Шаг 6. После получения NAS команды режима безопасности от UE, MME отправляет KeNB в eNB с S1AP первоначальная установка начального контекста (ключ защиты).

Шаг 7. После получения KeNB, eNB вычисляет Krrc-int, Krrc-enc, Kup-enc. Затем оно отправляет RRC ключ защиты команду с AS целостностью алгоритма и AS шифрующий алгоритм.

Шаг 8. После получения RRC команды ключа защиты UE вычисляет Krrc-int, Krrc-enc, Kup-enc. UE отправляет RRC выполненный ключ шифрования на eNB.

После всех описанных действий, все NAS и AS сообщения будут надежно защищены и зашифрованы, в отличие от пользовательских данных, которые будут только шифроваться.

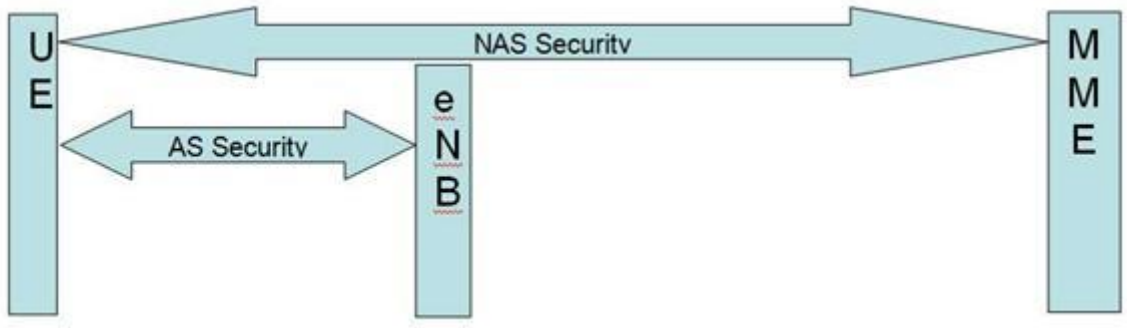


Рис. 2.3. Слои безопасности

Архитектура безопасности LTE определяет механизм безопасности и для уровня NAS и для уровня AS.

Безопасность NAS (слоя без доступа): Выполнена для NAS сообщений и принадлежит области UE и MME.

В этом случае необходима при передаче сообщений NAS между UE и MME – целостность, защищенная и зашифрованная с дополнительным заголовком безопасности NAS.

Безопасность AS (слоя с доступом): Выполнена для RRC и плоскости пользовательских данных, принадлежащих области UE и eNB. Уровень PDCP на сторонах UE и eNB отвечает за шифрование и защиту целостности.

RRC сообщения защищены целостностью и зашифрованы, однако данные U-Plane только зашифрованы.

Для генерации векторов аутентификации используется криптографический алгоритм с помощью однонаправленных функций (f1, f2, f3, f4, f5) когда прямой результат получается путем простых вычислений, а обратный результат не может быть получен обратным путем, то есть не существует эффективного алгоритма получения обратного результата. Для этого алгоритма используется случайное 128 битное случайное число RAND, мастер-ключ K абонента, также 128 бит и порядковый номер процедуры SQN

(Sequence Number). Счетчик SQN меняет свое значение при каждой генерации вектора аутентификации. Похожий счетчик SQN работает и в USIM. Такой метод позволяет генерировать каждый раз новый вектор аутентификации, не повторяя предыдущий уже использованный вектор аутентификации.

Помимо этих трех исходных величин: SQN, RAND и K в алгоритме f1 участвует поле управления аутентификацией Authentication Management Field (AMF), а в алгоритмах f2 – f5 исходные параметры – RAND и K, что и продемонстрировано на рис. 2.3, 2.4. На выходах соответствующих функций получают Message Authentication Code (MAC) - 64 бита; XRES – eXpected Response, результат работы алгоритма аутентификации <32 – 128 бит>; ключ шифрации CK, генерируемый с использованием входящих (K,RAND)->f3->CK; ключ целостности IK, сгенерированный с использованием входящего (K,RAND)->f4->IK; и промежуточный ключ Anonymity Key (AK), генерируемый с помощью (K,RAND)->f5->AK - 64 бита.

При обслуживании абонента сетью E-UTRAN ключи CK и IK в открытом виде в ядро сети не передают. В этом случае HSS генерирует K_{ASME} с помощью алгоритма KDF (Key Derivation Function), для которого исходными параметрами являются CK и IK, а также идентификатор обслуживающей сети и SQNÅAK. Вектор аутентификации содержит RAND, XRES, AUTN и K_{ASME}, на основе которого происходит генерация ключей шифрации и целостности, используемых в соответствующих алгоритмах.

Когда мобильная станция получает из ядра сети три параметра (RAND, AUTN и KSI_{ASME}, где KSI – Key Set Identifier, индикатор установленного ключа, однозначно связанный с K_{ASME} в мобильной станции).

После чего используя RAND и AUTN, USIM на основе алгоритмов безопасности, тождественных хранящимся в HSS, производит вычисление XMAC, RES, CK и IK.

Затем в ответе RES UE передает в MME вычисленное RES, которое должно совпасть с XRES, полученным из HSS. Так сеть аутентифицирует

абонента. Вычислив XMAC, UE сравнивает его с MAC, полученным ею в AUTN. При успешной аутентификации абонентом сети (MAC = XMAC) UE сообщает об этом в ответе RES. Если аутентификация сети не удалась (MAC ≠ XMAC), то UE направляет в MME ответ CAUSE, где указывает причину неудачи аутентификации.

При успешном завершении предыдущего этапа MME, eNB и UE производят генерацию ключей, используемых для шифрации и проверки целостности получаемых сообщений. В E-UTRAN имеется иерархия ключей, которая приведена на рис. 2.5.

Векторы аутентификации (рис. 2.3, 2.4):

Ключи IK и CK генерируются и в центре аутентификации, и в USIM;

Ключ AK генерируется только в центре аутентификации;

Ответ XRES генерируется только в центре аутентификации, а RES генерируется в USIM;

Код MAC генерируется только в центре аутентификации, а соответствующий ему параметр XMAC генерируется в USIM;

Маркер AUTH генерируется только в центре аутентификации.

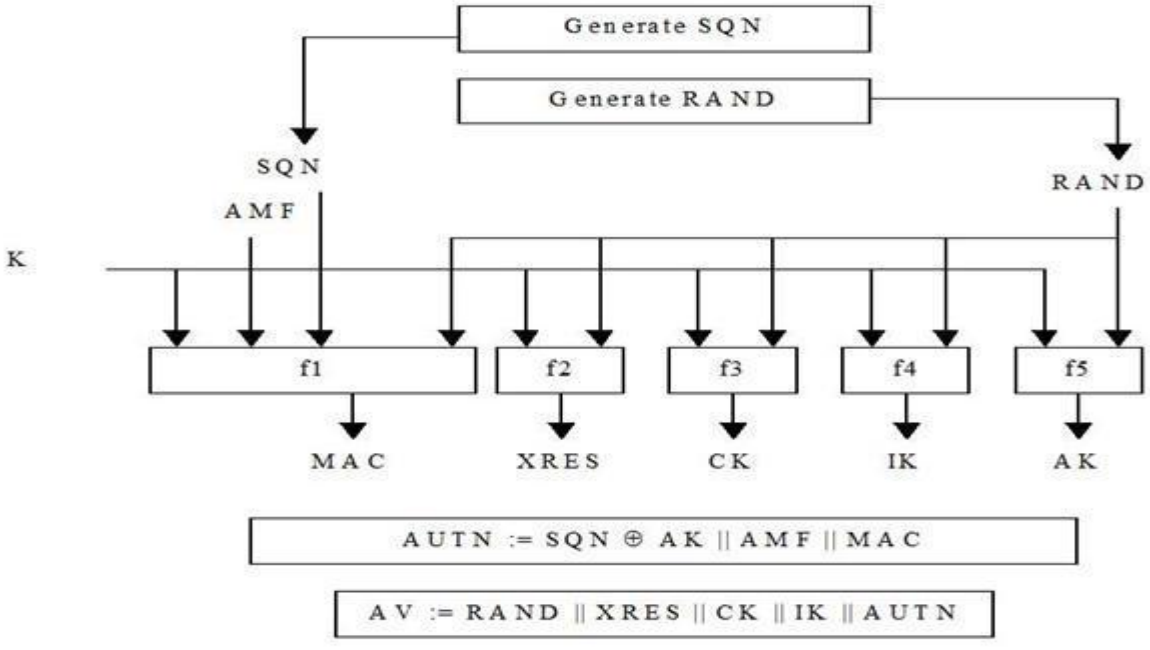


Рис. 2.4. Создание векторов на передающей стороне

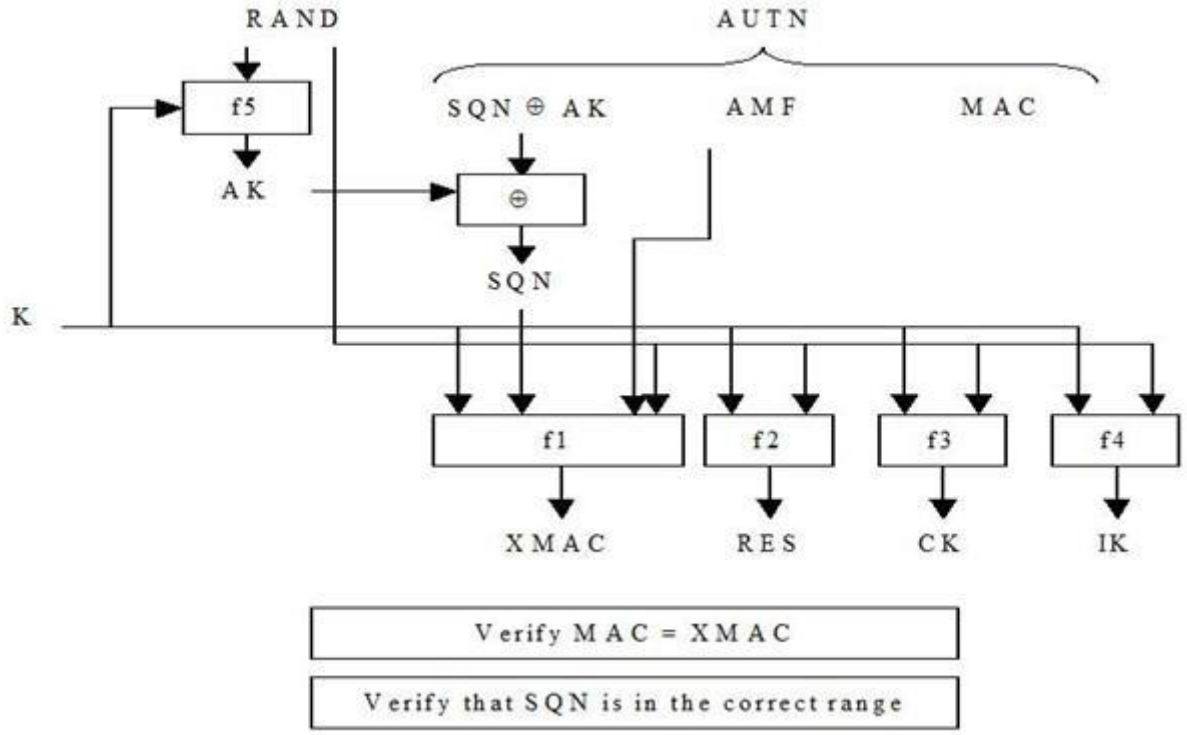


Рис. 2.5. Преобразование векторов на принимаемой стороне

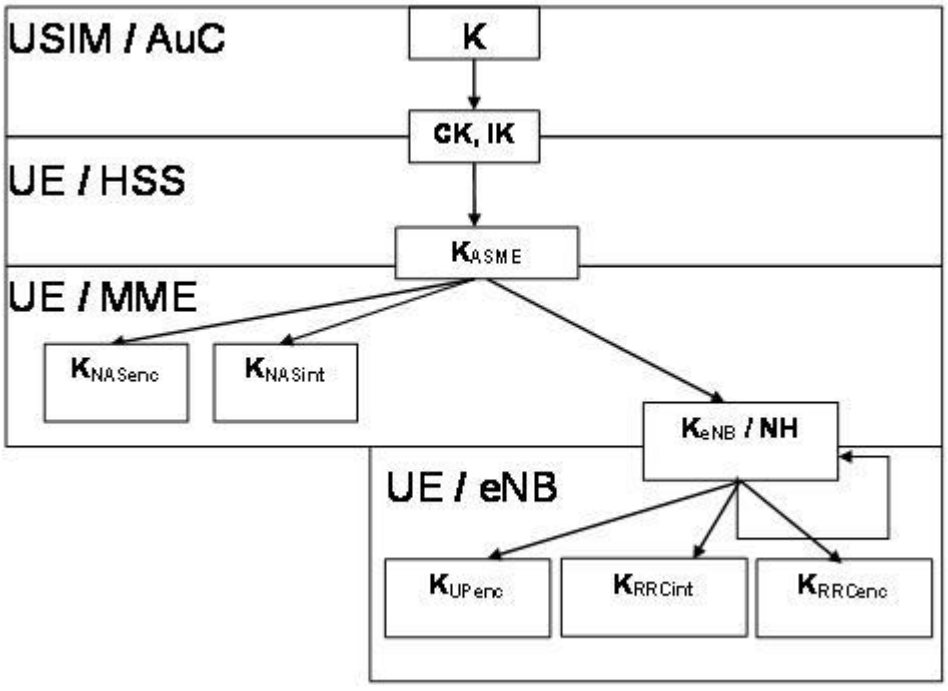


Рис. 2.6. Иерархия ключей в E-UTRAN

Исходным ключом для всей цепочки является K_{ASME} (256 бит). При передаче в радиоканале защиту обеспечивают для сигнального трафика (Control Plane) и для пользовательских пакетов (User Plane). При этом все

сообщения сигнализации разделяют на сквозные сигнальные сообщения между UE и MME протоколов MM и SM (NAS – Non Access Stratum) и сигнальные сообщения между eNB протокола RRC (AS – Access Stratum). Для шифрации и защиты целостности можно использовать разные базовые алгоритмы:

- UEA2 (UMTS Encryption Algorithm 2) и UIA2 (UMTS Integrity Algorithm 2);
- разработанные для стандартов 3G,AES (Advanced Encryption Standard).

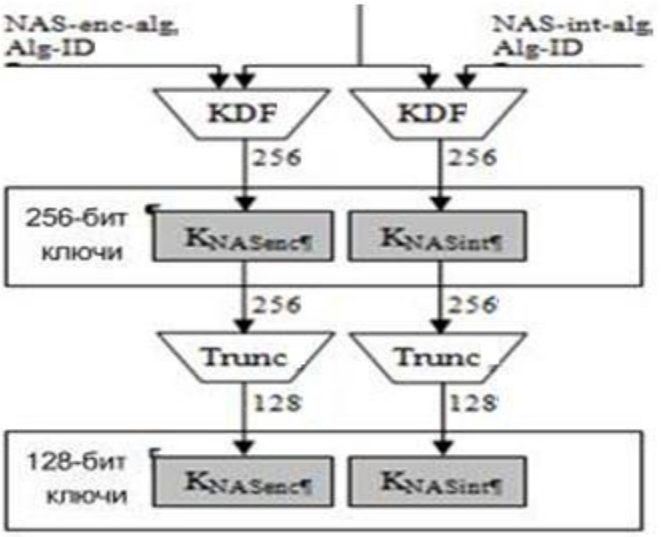


Рис. 2.7. Генерирование ключей шифрации и целостности для NAS сигнализации

Сигнальные сообщения протокола RRC (AS) также шифруют и обеспечивают их целостность. Пакеты трафика только шифруют. Эти операции производят в обслуживающей eNB и UE. Схема получения ключей шифрации и целостности (рис. 7) для AS и UP трафика отличается от предыдущего случая тем, что исходным параметром здесь служит вторичный промежуточный ключ KeNB (256 бит). Этот ключ генерируют, также используя KDF, где входными параметрами являются: KASME, счетчик сигнальных сообщений NAS вверх, прежнее значение KeNB, идентификатор

соты и номер частотного канала в направлении вверх. Следовательно, при каждой периодической локализации UE происходит изменение KeNB.

Также KeNB меняется и при хэндовере; при этом в алгоритме генерации нового KeNB можно использовать дополнительный параметр NH (Next Hop), фактически счетчик числа базовых станций, по цепочке обслуживающих абонента. Все реализуемые процедуры безопасности в сети E-UTRAN продемонстрированы на рис. 2.8.

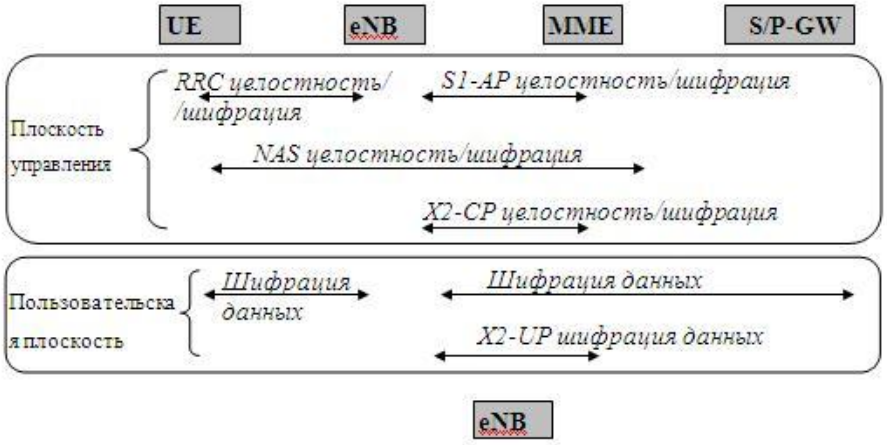


Рис. 2.8. Реализуемые процедуры безопасности в сети E-UTRAN

Алгоритм шифрации и дешифрации сообщений представлен на рис. 2.9.

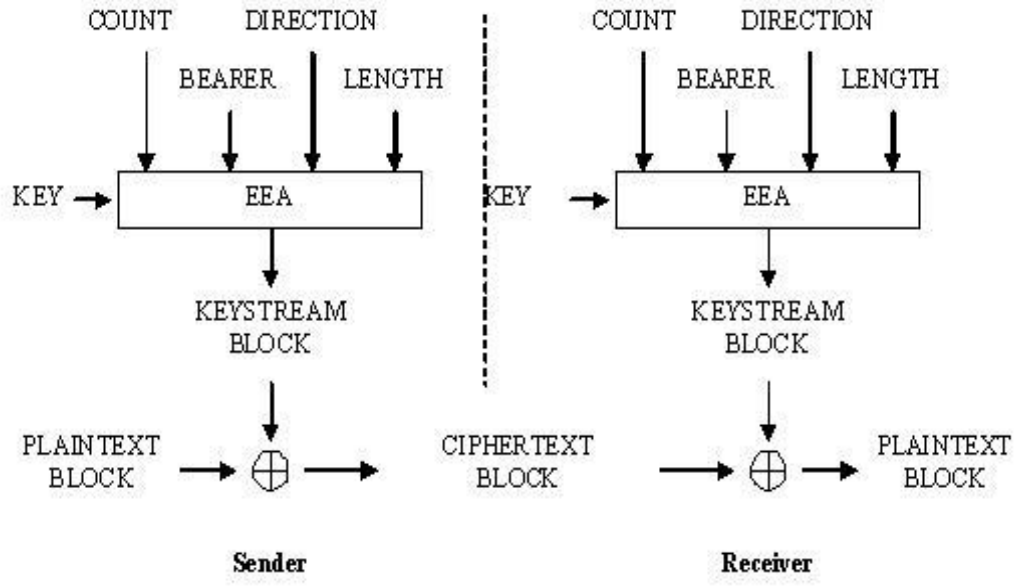


Рис. 2.9. Алгоритм шифрации в E-UTRAN

Исходными параметрами в этом алгоритме являются шифрующий ключ KEY (128 бит), счетчик пакетов (блоков) COUNT (32 бита), идентификатор сквозного канала BEARER (5 бит), указатель направления передачи DIRECTION (1 бит) и длина шифрующего ключа LENGTH. В соответствии с выбранным алгоритмом шифрации EEA (EPS Encryption Algorithm) вырабатывается шифрующее число KEYSTREAM BLOCK, которое при передаче складывают по модулю два с шифруемым исходным текстом блока PLAINTEXT BLOCK. При дешифрации на приемном конце повторно совершают эту же операцию.

Процедура защиты целостности сообщения состоит в генерации “хвоста“ MAC (Message Authentication Code) (32 бита), присоединяемого к передаваемому пакету. Алгоритм генерации MAC и проверки целостности полученного пакета путем сравнения XMAC с MAC (они должны совпасть) отображен на рис. 5.10.

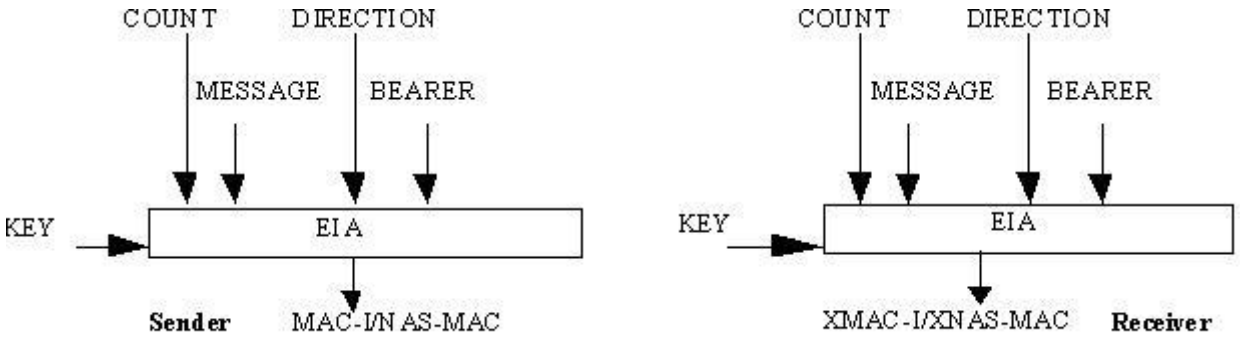


Рис. 2.10. Алгоритм проверки целостности E-UTRAN

В алгоритме EIA (EPS Integrity Algorithm) использован ключ целостности KEY (128 бит), счетчик сообщений COUNT (32 бита), идентификатор сквозного канала BEARER (5 бит), указатель направления передачи DIRECTION (1 бит) и само сообщение MESSAGE.

Моделирование технологии LTE в среде MATLAB с использованием встроенного пакета LTE System Toolbox

LTE System Toolbox™ предоставляет соответствующие стандарту функции и приложения для проектирования, моделирования и проверки

коммуникационных систем стандартов LTE и LTE-Advanced. Данный инструмент ускоряет разработку LTE-алгоритмов и физического уровня (PHY), предоставляет эталонный образец для проверки и тестирования на соответ- ствие стандарту, а также позволяет генерировать тестовые сигналы. С помощью LTE System Toolbox можно производить настройку, моделирование, измерения и анализ канала связи. Также можно создавать и повторно использовать сценарии тестов для подтверждения того, что проекты, прототипы и разработки соответствуют стандарту LTE.

Основные особенности:

• Модели, соответствующие стандартам LTE и LTE-Advanced (Release 8, 9, 10 и 11). • Функции обработки на канальном уровне, поддержка от 1 до 10 режимов передачи по нисходящему каналу, образцы проектов, в том числе CoMP.

• Тестовые модели (E-TM), эталонный измерительный канал (RMC) для LTE, LTE-A, а также генератор UMTS-сигналов.

• Интерактивные инструменты для проверки на соответствие стандарту и BER тестов.

• Передача и приём сигналов при помощи радиоустройств для тестирования систем в реаль- ном эфире.

• Выделение системных и контрольных параметров из принятого сигнала, в том числе cell ID, MIB и SIB1. • Оценка канала связи.

Сквозное моделирование LTE

System Toolbox даёт возможность моделировать и имитировать физический уровень стандарта LTE. Моделирование системы на канальном уровне позволяет добиться требуемых значений характеристик системы, в том числе пропускной способности и BER, а также опреде- лить конкретные реализации системы на основе производимых измерений.

LTE System Toolbox также позволяет улучшить планирование системы, облегчая моделирование канального уровня, которое предоставляет

некоторые параметры, необходимые для проектирования базовых станций с заданной геометрией и характеристиками распространения сигнала.

Набор поддерживаемых функций для моделирования режимов передачи и приёма, а также канала связи, включает в себя:

• режимы FDD и TDD на несущих частотах; • все полосы передачи LTE-сигналов от 1,4 до 20 МГц, LTE-A до 100 МГц с агрегацией несущей; • различные типы LTE-сигналов, включая нисходящие и восходящие опорные сигналы и сигналы синхронизации;

• физические LTE-каналы, в том числе каналы управления и каналы общего доступа; • готовую процедуру обработки нисходящего канала, в том числе формирование нисходящего общего канала и канала управления, все возможные MIMO-режимы и генерацию OFDM-сигналов;

• готовую процедуру обработки восходящего канала, в том числе формирование восходящих общего канала и канала управления, SU-MIMO и MU-MIMO режимы и генерацию SC-FDMA сигналов;

• адаптацию к каналу связи, включая схемы выбора типов модуляции и кодирования (MCS) в соответствии с оценкой качества канала связи (CQI), индикатора ранга (RI) и индикации матрицы прекодера (PMI);

• возможности и примеры построения LTE-Advanced, в том числе приём и передача с несколькими eNB (CoMP) и с агрегацией несущей;

• модели распространения LTE-сигналов, в том числе модель пешехода (EPA), модель автомобиля (EVA), модель типичный городской застройки (ETU), модель распространения в движении, а также модели MIMO-каналов в скоростном поезде. LTE System Toolbox даёт возможность создавать тесты, измеряющие пропускную способность PDSCH-канала в соответствии с указанными в стандарте LTE (TS 36.101) условиями испытаний. Структуры данных в LTE System Toolbox позволяют удобно отображать все параметры системы. Функции данного инструмента отражают любые возможные комбинации режимов работы передатчиков, моделей каналов и приемников. Используя этот инструмент для тестирования на соответствие стандарту и

BLER-тестирования, вы можете измерять характеристики системы и сравнивать их с указанными в спецификации к стандарту

На рисунке 2.11 изображена структурная схема программного комплекса.

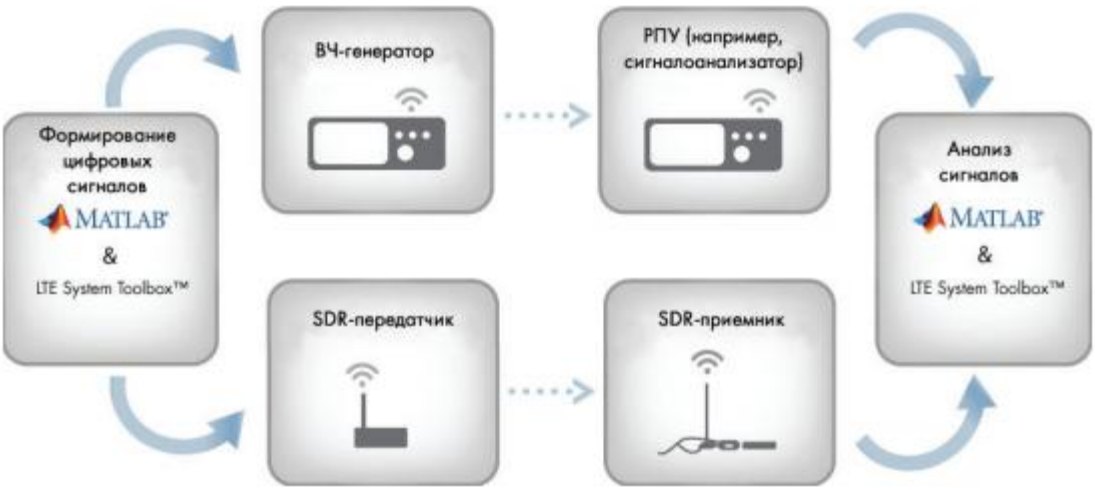


Рис. 2.11. Структурная схема программного комплекса

Для генерации тестового сигнала от базовой станции к абоненту используется генератор LTE-Downlink E-TM Generator.

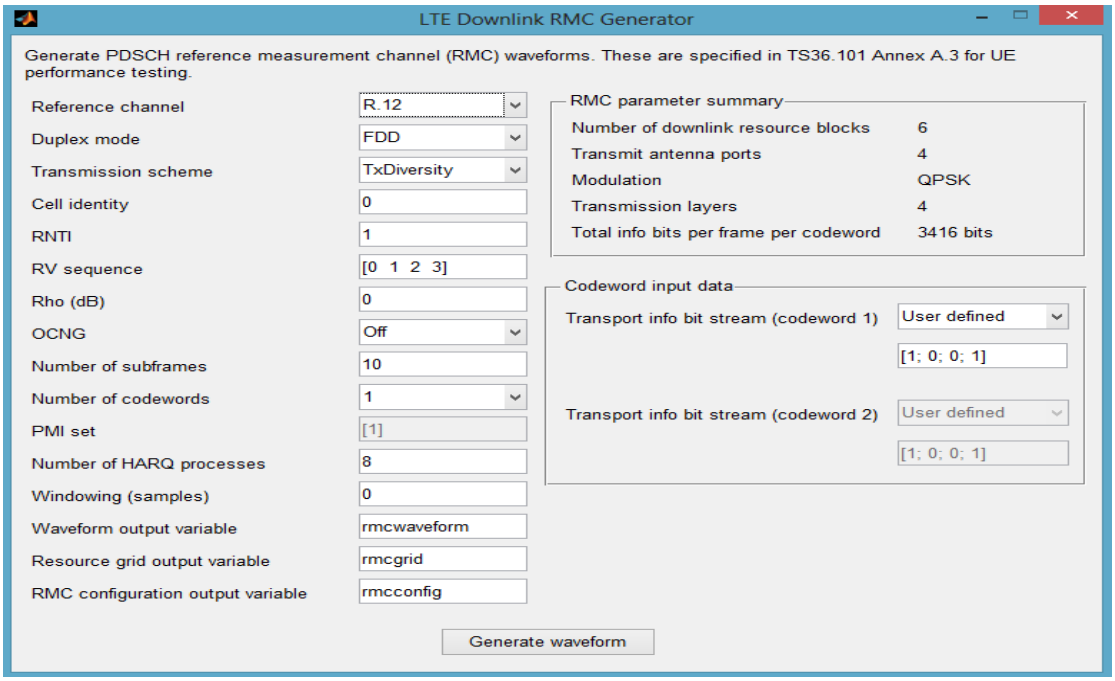


Рис. 2.12. Окно генератора от базовой станции к абоненту

В данном окне задаются параметры генерируемого сигнала на базовой станции, такие как: Количество каналов, вид модуляции, количество обслуживаемых абонентов в секунду, число кодовых слов, вектор инициализации.

На рисунке 2.13 показан вид сгенерированного сигнала, а на рисунке 2.14 трех мерный спектр полученного сигнала.

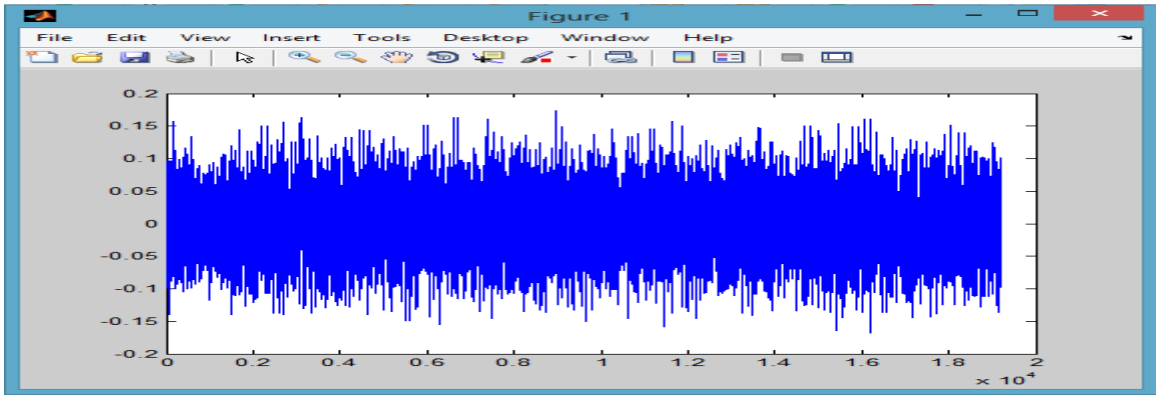


Рисунок 2.13. Сгенерированный сигнал станцией

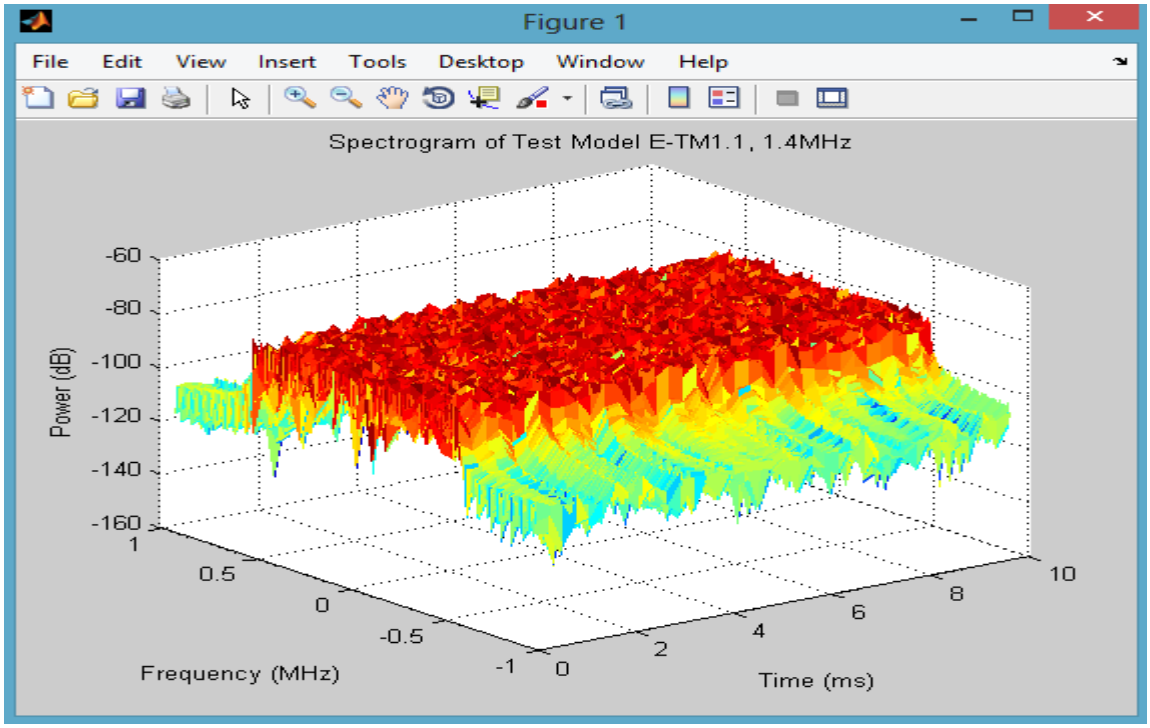


Рис. 5.14. Спектр сгенерированного сигнала станцией в течении 10 секунд

Для генерации тестового сигнала от абонента к базовой станции используется генератор LTE-Uplink RMS Generator.

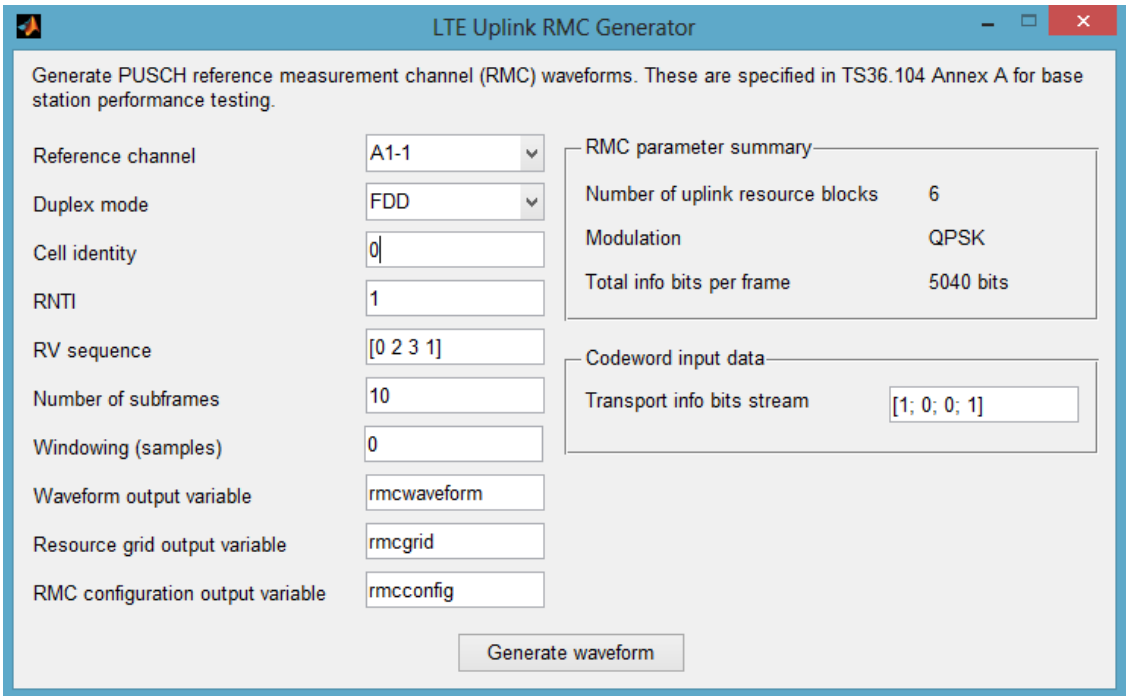


Рис. 2.15. Окно генератора LTE-Uplink RMS Generator

В окне на рисунке 5.16 производятся настройки параметров генерирования от абонента к базовой станции, такие как: Количество каналов для передачи потока пакетов, тип модуляции, вектор инициализации, количество передаваемых пакетов в секунду.

Сигнал, генерируемый генератором LTE-Uplink RMS Generator, представлен на рисунке 2.16.

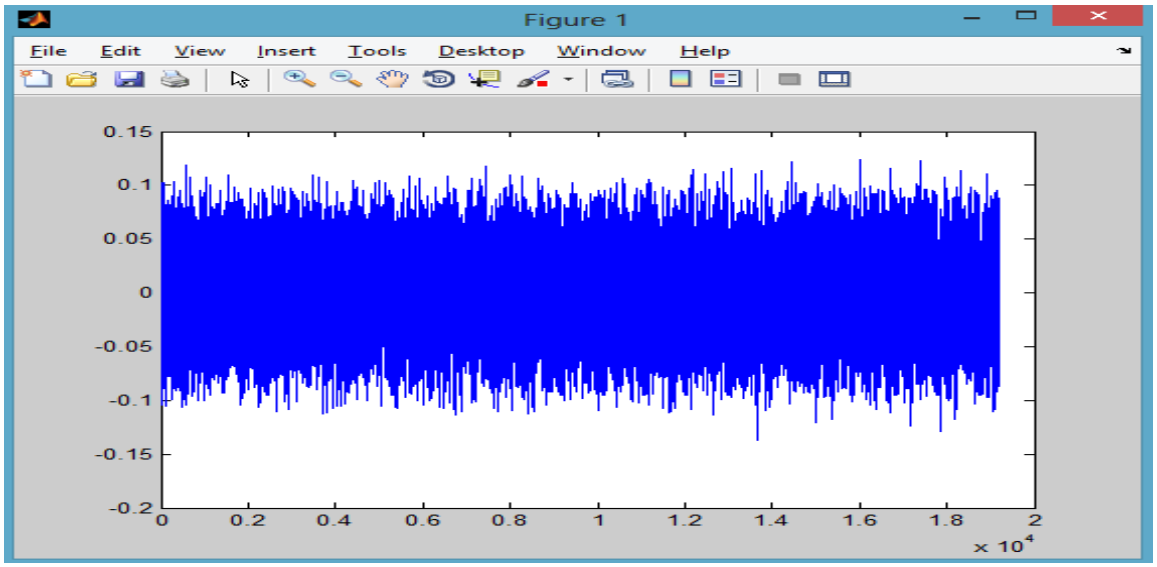


Рис. 2.16. Сигнал генерируемый генератором LTE-Uplink RMS Generator

Для вычисления потерь и пропускной способности системы передачи можно использовать блок LTE PDSCH Conformance Testing.

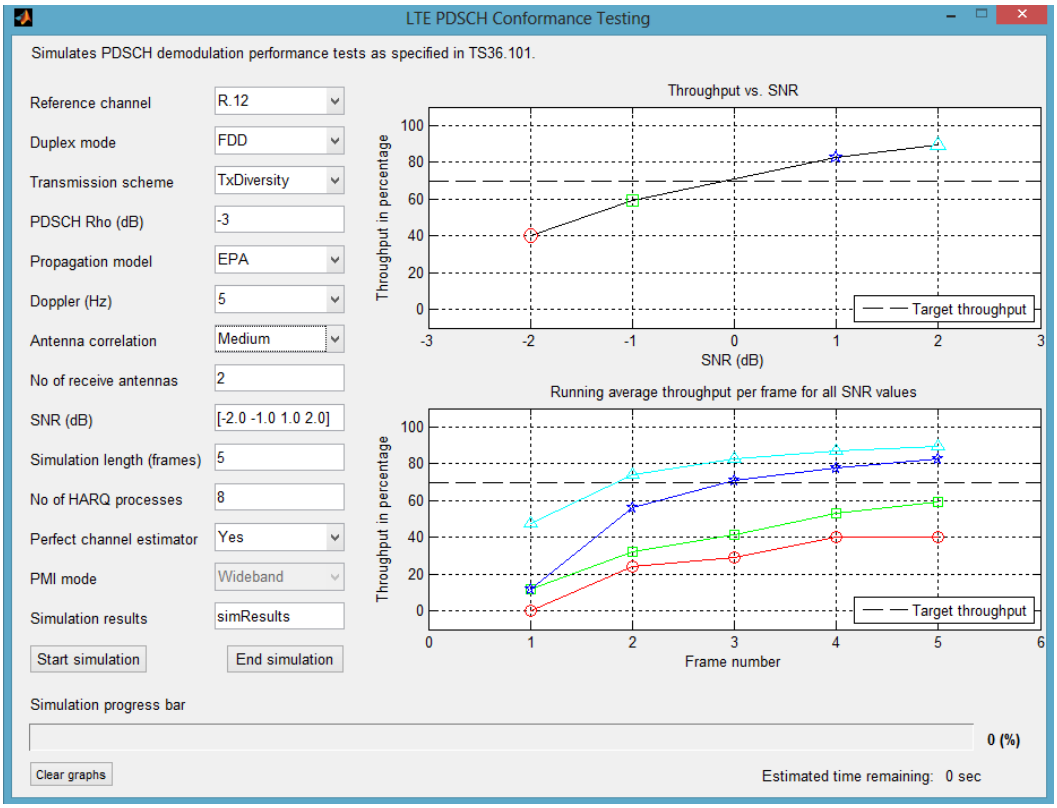


Рис. 2.17. Окно LTE PDSCH Conformance Testing

В данном окне можно произвести настройку параметров линии передачи, таких как: количество каналов, тип модуляции, доплеровскую частоту, уровень шума, настройки антенны и другие. А так же наглядно пронаблюдать изменение количества потерь в линии передачи и пропускную способность системы.

В результате проделанной работы были изучены основные понятия беспроводных сетей LTE, физическая структура построения беспроводной сети LTE. Изучена существующие методы и средства защиты беспроводных сетей LTE. Построена программная структурная схема беспроводной сети LTE среде Matlab с использование встроенного пакета LTE System Toolbox и проведены исследования основных узлов сети LTE.

ЗАКЛЮЧЕНИЕ

Представлены криптографические протоколы в сетях передачи данных и компьютерный практикум для исследования протоколов SSL и TSL. Рассмотрено шифрование в современных системах связи стандартов GSM и LTE и компьютерный практикум для исследования стандарта LTE в MATLAB.

ЛИТЕРАТУРА

1. Б. Шнайер «Прикладная криптография. 2-е издание. Протоколы, алгоритмы и исходные тексты на языке С». - М.: Изд-во "Триумф", 2002. - 816 с.

2. Романец Ю.В. Защита информации в компьютерных системах и сетях / Под ред. В.Ф. Шаньгина. - 2-е изд., перераб. и доп. - М.: Радио и связь, 2001. - 376 с.

3. RSA Laboratories // http://www.rsa.com/rsalabs/node.asp?id=2009

4. Diffie, D. New directions in Cryptography / D. Diffie, M.Hellman // IEEE Transactions on information theory. November. 1976.

5. Rivest, R. A Method for obtaining digital signatures and public keyCryptosystems / R. Rivest, A. Shamir, L. Adleman // Communications of the ACM. February. 1978.

6. Столингс, В. Криптография и защита сетей: принципы и практика / В. Столингс ; пер. с англ. . – 2-е изд. – М. : Изд. дом «Вильямс», 2001. – 672 с.

7. Коблиц, Н. Курс теории чисел и криптографии / Н. Коблиц ; пер. с англ. М. А.Михайловой и В. Е. Тараканова ; под. ред. А. М. Зубкова. – М. : Науч. изд-во ТВП, 2001. – 254 с.

8. Безопасность на транспортном уровне: SSL и TLS | Лекция | НОУ ИНТУИТ [Электронный ресурс]. – Режим доступа http://www.intuit.ru/studies/courses/553/409/lecture/9387 (дата обращение 13.04.2015).

9.http://matlab.ru/products/LTE-System-Toolbox/lte-system-toolbox_rus_web.pdf (дата запроса 05.04.2016)