**Санкт-Петербургский государственный электротехнический университет имени В.И.Ульянова «ЛЭТИ»**

***Реферат***

на тему

***"Протоколы передачи данных"***

Выполнил: студент гр. 3371

Константинова А.А

Преподаватель: Кондратьева Ю.Б.

Санкт-Петербург

2015г.

## Содержание:

## Введение…………………………………………………………………………………3

## Протоколы передачи данных…………………………………………………………3

1. **Протоколы безопасной передачи данных………………………………...................8**
2. **Протокол защиты передачи данных в телекомуникационных сетях – IPSec….9**
3. **Список использованной литературы………………………………………………13**

## Введение:

## Благодаря возникновению и развитию сетей передачи данных появился новый, высокоэффективный способ взаимодействия между людьми. Первоначально сети использовались главным образом для научных исследований, но затем они стали проникать буквально во все области человеческой деятельности. При этом большинство сетей существовало совершенно независимо друг от друга, решая конкретные задачи для конкретных групп пользователей. В соответствии с этими задачами выбирались те или иные сетевые технологии и аппаратное обеспечение. Построить универсальную физическую сеть мирового масштаба из однотипной аппаратуры просто невозможно, поскольку такая сеть не могла бы удовлетворять потребности всех ее потенциальных пользователей. Тогда возникла идея объединить множество физических сетей в единую глобальную сеть, в которой использовались бы как соединения на физическом уровне, так и новый набор специальных "соглашений" или протоколов. Эта технология, получившая название internet, должна была позволить компьютерам "общаться" друг с другом независимо от того, к какой сети и каким образом они подсоединены

***Протоколы передачи данных*** это набор соглашений, который определяет обмен данных междуразличными программами. Протоколы задают способы передачи сообщений и обработки ошибок в сети, а также позволяют разрабатывать стандарты, не привязанные к конкретной аппаратной платформе.

Основное, что отличает Internet от других сетей - это ее протоколы - TCP/IP. Вообще, термин TCP/IP обычно означает все, что связано с протоколами взаимодействия между компьютерами в Internet. Он охватывает целое семейство протоколов, прикладные программы, и даже саму сеть. TCP/IP - это технология межсетевого взаимодействия, технология Internet. Сеть, которая использует технологию Internet, называется "Internet". Если речь идет о глобальной сети, объединяющей множество сетей с технологией Internet, то ее называют Internet.

Свое название протокол TCP/IP получил от двух коммуникационных протоколов (или протоколов связи). Это Transmission Control Protocol (TCP) и Internet Protocol (IP). Несмотря на то, что в сети Internet используется большое число других протоколов, сеть Internet часто называют TCP/IP-сетью, так как эти два протокола, безусловно, являются важнейшими.

Как и во всякой другой сети в Internet существует 7 уровней взаимодействия между компьютерами: физический, логический, сетевой, транспортный, уровень сеансов связи, представительский и прикладной уровень. Соответственно каждому уровню взаимодействия соответствует набор протоколов (т.е. правил взаимодействия).

Протоколы физического уровня определяют вид и характеристики линий связи между компьютерами. В Internet используются практически все известные в настоящее время способы связи от простого провода (витая пара) до волоконно-оптических линий связи (ВОЛС).

Для каждого типа линий связи разработан соответствующий протокол логического уровня, занимающийся управлением передачей информации по каналу. К протоколам логического уровня для телефонных линий относятся протоколы SLIP (Serial Line Interface Protocol) и PPP (Point to Point Protocol). Для связи по кабелю локальной сети - это пакетные драйверы плат ЛВС.

Протоколы сетевого уровня отвечают за передачу данных между устройствами в разных сетях, то есть занимаются маршрутизацией пакетов в сети. К протоколам сетевого уровня принадлежат IP (Internet Protocol) и ARP (Address Resolution Protocol).

Протоколы транспортного уровня управляют передачей данных из одной программы в другую. К протоколам транспортного уровня принадлежат TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

Протоколы уровня сеансов связи отвечают за установку, поддержание и уничтожение соответствующих каналов. В Internet этим занимаются уже упомянутые TCP и UDP протоколы, а также протокол UUCP (Unix to Unix Copy Protocol).

Протоколы представительского уровня занимаются обслуживанием прикладных программ. К программам представительского уровня принадлежат программы, запускаемые, к примеру, на Unix-сервере, для предоставления различных услуг абонентам. К таким программам относятся: telnet-сервер, FTP-сервер, Gopher-сервер, NFS-сервер, NNTP (Net News Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP2 и POP3 (Post Office Protocol) и т.д.

К протоколам прикладного уровня относятся сетевые услуги и программы их предоставления.

Сеть Internet - это одна из реализаций технологии internet, которая объединяет около 10 млн. компьютеров по всему миру, которые взаимодействуют друг с другом с помощью стека протоколов TCP/IP.

Серия протоколов TCP/IP - яркий пример открытой системы в том смысле, что, в отличие от протоколов, используемых в коммуникационных системах разных поставщиков, все спецификации этого стека протоколов и многие из его реализаций общедоступны (предоставляются бесплатно или за символическую цену). Это позволяет любому разработчику создавать свое программное обеспечение, необходимое для взаимодействия по Internet.

Хотя рассматриваемый стек протоколов и называется TCP/IP, сами протоколы TCP и IP (**IP**, досл. «межсетевой протокол»)являются важнейшими, но не единственными представителями этого семейства. Каждый уровень коммуникаций обслуживается несколькими протоколами. Рассмотрим их более подробно.

**TCP** и **UDP** - протоколы транспортного уровня, организующие поток данных между конечными системами для приложений верхнего уровня. Эти протоколы значительно отличаются друг от друга.

**TCP (Transmission Control Protocol)** обеспечивает надежную передачу данных между двумя хостами. Он позволяет клиенту и серверу приложения устанавливать между собой логическое соединение и затем использовать его для передачи больших массивов данных, как если бы между ними существовало прямое физическое соединение. Протокол позволяет осуществлять дробление потока данных, подтверждать получение пакетов данных, задавать таймауты (которые позволяют подтвердить получение информации), организовывать повторную передачу в случае потери данных и т.д. Поскольку этот транспортный протокол реализует гарантированную доставку информации, использующие его приложения получают возможность игнорировать все детали такой передачи.

Протокол **UDP (User Datagram Protocol)** реализует гораздо более простой сервис передачи, обеспечивая подобно протоколам сетевого уровня, ненадежную доставку данных без установления логического соединения, но, в отличие от IP, - для прикладных систем на хост-компьютерах. Он просто посылает пакеты данных, дейтаграммы (datagrams), с одной машины на другую, но не предоставляет никаких гарантий их доставки. Все функции надежной передачи должны встраиваться в прикладную систему, использующую UDP. Протокол UDP имеет и некоторые преимущества перед TCP. Для установления логических соединений нужно время, и они требуют дополнительных системных ресурсов для поддержки на компьютере информации о состоянии соединения. UDP занимает системные ресурсы только в момент отправки или получения данных. Поэтому если распределенная система осуществляет непрерывный обмен данными между клиентом и сервером, связь с помощью транспортного уровня TCP окажется для нее более эффективной. Если же коммуникации между хост-компьютерами осуществляются редко, предпочтительней использовать протокол UDP.

Почему же существуют два транспортных протокола TCP и UDP, а не один из них? Дело в том, что они предоставляют разные услуги прикладным процессам. Большинство прикладных программ пользуются только одним из них. Программист выбирает тот протокол, который наилучшим образом соответствует его потребностям. Если нужна надежная доставка, то лучшим может быть TCP, если же нужна доставка датаграмм, то лучше может быть UDP. Если нужна эффективная доставка по длинному и ненадежному каналу передачи данных, то лучше может подойти протокол TCP, если же нужна эффективность на быстрых сетях с короткими соединениями, то лучшим может быть протокол UDP.

*Среди известных распределенных приложений, использующих TCP, - такие как Telnet, FTP и SMTP*. *Протоколом UDP пользуется, в частности, протокол сетевого управления SNMP.* Протоколы прикладного уровня ориентированы на конкретные прикладные задачи. Они определяют как процедуры по организации взаимодействия определенного типа между прикладными процессами, так и форму представления информации при таком взаимодействии.

Протокол **TELNET** позволяет обслуживающей машине рассматривать все удаленные терминалы как стандартные "сетевые виртуальные терминалы" строчного типа, работающие в коде ASCII, а также обеспечивает возможность согласования более сложных функций (например, локальный или удаленный эхо-контроль, страничный режим, высота и ширина экрана и т.д.) TELNET работает на базе протокола TCP. На прикладном уровне над TELNET находится либо программа поддержки реального терминала (на стороне пользователя), либо прикладной процесс в обсуживающей машине, к которому осуществляется доступ с терминала. Работа с TELNET походит на набор телефонного номера. Пользователь набирает на клавиатуре что-то вроде

telnet delta

и получает на экране приглашение на вход в машину delta. Протокол TELNET существует уже давно. Он хорошо опробован и широко распространен. Создано множество реализаций для самых разных операционных систем.

Протокол **FTP (File Transfer Protocol)** распространен также широко как TELNET. Он является одним из старейших протоколов семейства TCP/IP. Также как TELNET он пользуется транспортными услугами TCP. Существует множество реализаций для различных операционных систем, которые хорошо взаимодействуют между собой. Пользователь FTP может вызывать несколько команд, которые позволяют ему посмотреть каталог удаленной машины, перейти из одного каталога в другой, а также скопировать один или несколько файлов.

Протокол **SMTP (Simple Mail Transfer Protocol)** поддерживает передачу сообщений (электронной почты) между произвольными узлами сети internet. Имея механизмы промежуточного хранения почты и механизмы повышения надежности доставки, протокол SMTP допускает использование различных транспортных служб. Он может работать даже в сетях, не использующих протоколы семейства TCP/IP. Протокол SMTP обеспечивает как группирование сообщений в адрес одного получателя, так и размножение нескольких копий сообщения для передачи в разные адреса.

Сетевая файловая система **NFS (Network File System)** впервые была разработана компанией Sun Microsystems Inc. NFS использует транспортные услуги UDP и позволяет монтировать в единое целое файловые системы нескольких машин с ОС UNIX. Бездисковые рабочие станции получают доступ к дискам файл-сервера так, как будто это их локальные диски. NFS значительно увеличивает нагрузку на сеть. Если в сети используются медленные линии связи, то от NFS мало толку. Однако, если пропускная способность сети позволяет NFS нормально работать, то пользователи получают большие преимущества. Поскольку сервер и клиент NFS реализуются в ядре ОС, все обычные несетевые программы получают возможность работать с удаленными файлами, расположенными на подмонтированных NFS-дисках, точно также как с локальными файлами.

Протокол **SNMP (Simple Network Management Protocol)** работает на базе UDP и предназначен для использования сетевыми управляющими станциями. Он позволяет управляющим станциям собирать информацию о положении дел в сети internet. Протокол определяет формат данных, их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети.

TCP и UDP идентифицируют приложения по 16-битным номерам портов. Серверы приложений обычно имеют заранее известные номера портов. Например, в каждой реализации TCP/IP, которая поддерживает сервер FTP, этот протокол передачи файлов получает для своего сервера номер TCP-порта 21. Каждый Telnet-сервер имеет TCP-порт 23, а сервер протокола TFTP (Trivial File Transfer Protocol) - UDP-порт 69. Службам, которые могут поддерживаться любой реализацией TCP/IP, назначаются номера портов в диапазоне от 1 до 1023. Назначение номеров портов находится в ведении организации Internet Assigned Numbers Authority (IANA). Клиент приложения обычно "не интересуется" номером своего порта для транспортного уровня, который он использует. Ему лишь необходимо гарантировать, что этот номер уникален для данного хоста. Номера портов клиентов приложений принято называть краткосрочными (т.е. недолговечными), поскольку в общем случае клиенты существуют ровно столько времени, сколько работающий с ним пользователь нуждается в соответствующем сервере. (Серверы, напротив, находятся в рабочем состоянии все время, пока включен хост, на котором они работают.) В большинстве реализаций TCP/IP краткосрочным номерам портов выделен диапазон от 1024 до 5000.

**Internet Protocol (IP)** - основной протокол сетевого уровня, позволяющий реализовывать межсетевые соединения. Он используется обоими протоколами транспортного уровня. IP определяет базовую единицу передачи данных в internet, IP-дейтаграмму, указывая точный формат всей информации, проходящей по сети TCP/IP. Программное обеспечение IP выполняет функции маршрутизации, выбирая путь данных по паутине физических сетей. Для определения маршрута поддерживаются специальные таблицы; выбор осуществляется на основе адреса сети, к которой подключен компьютер-адресат. Протокол IP определяет маршрут отдельно для каждого пакета данных, не гарантируя надежной доставки в нужном порядке. Он задает непосредственное отображение данных на нижележащий физический уровень передачи и реализует тем самым высокоэффективную доставку пакетов.

Кроме IP, на сетевом уровне используются также протоколы ICMP и IGMP. **ICMP (Internet Control Message Protocol)** отвечает за обмен сообщениями об ошибках и другой важной информацией с сетевым уровнем на другом хосте или маршрутизаторе. **IGMP (Internet Group Management Protocol)** используется для отправки IP-дейтаграмм множеству хостов в сети.

На самом нижнем уровне - сетевого интерфейса - используются специальные протоколы разрешения адресов **ARP (Address Resolution Protocol)** и **RARP (Reverse Address Resolution Protocol)**. Эти протоколы применяются только в определенных типах физических сетей (Ethernet и Token Ring) для преобразования адресов сетевого уровня в адреса физической сети и обратно.

Еще в Интернете используется протокол **IPv4(*IPv4****- это четвертая версия протокола IP (Internet Protocol), которая на сегодняшний является основной и обслуживает большую часть сети Интернет. IPv4 протокол устанавливает правила функционирования компьютерных сетей по принципу обмена пакетами..*), созданный в 70-е годы прошлого столетия. Каждый IP-адрес в нем состоит из 32 бит и представлен в виде четырех чисел по 8 бит, разделенных точками. Такой подход позволяет получить более четырех миллиардов уникальных IP-адресов. На заре эры Интернета казалось, что этого более чем достаточно. А поэтому адреса целыми блоками выдавались напрямую организациям, среди которых преобладали научные учреждения и университеты.

Однако быстрый рост популярности Сети показал, что свободные идентификаторы закончатся гораздо быстрее, чем предполагалось изначально. Число устройств, способных подключаться к Интернету, многократно возросло. Эксперты рассчитали, что при нынешних темпах развития Интернета возможностей системы IPv4 хватит еще на 5 лет. По данным корпорации [ICANN](http://www.icann.org/) на октябрь 2007 года, оставалось всего лишь чуть более 650 тыс. свободных IP-адресов.

Пессимисты утверждали, что IPv4 исчерпает себя уже в 2009 году. Протокол обладает и рядом других существенных недостатков. В 32-битном пространстве достаточно сложно построить и упорядочить структуру адресов, что приводит к увеличению маршрутных таблиц и, следовательно, значительно усложняет маршрутизацию в Интернете.

Кроме того, в протоколе не предусмотрены механизмы информационной безопасности, например, отсутствует возможность шифрования данных.

Наконец, в IPv4 не поддерживается качество обслуживания, то есть информация о пропускной способности и задержках, которая необходима для работы некоторых сетевых приложений.

В 1992 году появилась новая технология, которая получила название **IPv6** или Internet Protocol version 6. В IPv6 длина IP-адреса расширена до 128 бит, поэтому число доступных идентификаторов увеличивается практически до бесконечности.

Таким образом, применение этой технологии позволяет снабдить каждое устройство, имеющее доступ в Интернет, уникальным IP-адресом. А это обеспечивает непосредственное взаимодействие всех устройств, подключенных к Сети. Такое взаимодействие даст возможность, например, управлять кондиционером, находящимся у вас дома, прямо из офиса. Помимо увеличения адресного пространства протокол обладает и другими преимуществами. Например, в IPv6 существует отдельный тип адресов "anycast address", который позволяет устройству (в терминологии адресации оно называется узлом), подключенному к Интернету, отправлять запрос любой группе серверов. Это дает возможность узлу определить сервер, находящийся к нему ближе других и далее взаимодействовать только с ним.

Кроме того, в новом протоколе был улучшен формат заголовка пакета данных. Ряд его полей, которые существовали в IPv4, не вошли в IPv6, часть из них стала необязательной, а некоторые были усовершенствованы. При этом в заголовке IPv6 появилось несколько новых полей. С их помощью можно задать хосту-отправителю приоритет для своих пакетов, а также обеспечить потоковую обработку, что значительно ускоряет маршрутизацию. В результате оптимизации заголовка число полей сократилось с 14 до 8, что позволяет существенно увеличить скорость обмена данными между устройствами. Стоит отметить, что протокол позволяет при необходимости добавлять в IP-заголовки новые поля.

**Протоколы безопасной передачи данных**

  Для передачи и защиты ценной информации в Интернете используются протоколы безопасной передачи данных, а именно SSL, SET, IP v.6.  
  Платежные системы являются наиболее критичной частью электронной коммерции и будущее их присутствия в сети во многом зависит от возможностей обеспечения информационной безопасности и других сервисных функций в Интернете. В платежных системах Интернета используются протоколы передачи данных SSL и SET.  
  Протокол SSL (Secure Socket Layer) был разработан американской компанией Netscape Communications как протокол, обеспечивающий защиту данных между сервисными протоколами (HTTP, NNTP, FTP и др.) и транспортными протоколами (TCP/IP) с помощью криптографии в соединениях «точка-точка».  
  Протокол SSL предназначен для решения традиционных задач обеспечения защиты информационного взаимодействия, которые в среде клиент-сервер интерпретируются следующим образом:  
  - пользователь и сервер должны быть взаимно уверены, что они обмениваются информацией не с подставными абонентами, а именно с теми, которые нужны, не ограничиваясь паролевой защитой;  
  - после установления соединения между сервером и клиентом весь информационный поток между ними должен быть защищен от несанкционированного доступа;  
  - при обмене информацией стороны должны быть уверены в отсутствии случайных или умышленных искажений при ее передаче.  
  Протокол SSL позволяет серверу и клиенту перед началом информационного взаимодействия аутентифицировать друг друга, согласовать алгоритм шифрования и сформировать общие криптографические ключи.  
  Если два пользователя хотят быть уверенными, что информацию, которой они обмениваются, не получит третий, то каждый из них, должен передать одну компоненту ключевой пары (а именно открытый ключ) другому и хранить другую компоненту (секретный ключ). Сообщения шифруются с помощью открытого, расшифровываются только с использованием секретного ключа.  
  Целостность и аутентификация сообщения обеспечиваются использованием электронной цифровой подписи.  
  В Интернете также используется протокол безопасных электронных транзакций SET (Security Electronics Transaction), предназначенный для организации электронной торговли через сеть, который основан на использовании цифровых сертификатов по стандарту Х.509.  
  SET обеспечивает кросс-аутентификацию счета держателя карточки, продавца и банка продавца для проверки готовности оплаты товара, целостность и секретность сообщения, шифрование ценных и уязвимых данных. Поэтому SET можно назвать стандартной технологией или системой протоколов выполнения безопасных платежей с использованием пластиковых карточек через Интернет.  
  SET позволяет потребителям и продавцам подтвердить подлинность всех участников сделки, происходящей в Интернете, с помощью криптографии, применяя, в том числе, и цифровые сертификаты.  
  SET обеспечивает следующие специальные требования защиты операций электронной коммерции:  
  - секретность данных оплаты и конфиденциальность информации заказа, переданной вместе с данными об оплате;  
  - сохранение целостности данных платежей (целостность обеспечивается при помощи цифровой подписи);  
  - специальную криптографию с открытым ключом для проведения аутентификации;  
  - аутентификацию держателя по кредитной карточке, которая обеспечивается применением цифровой подписи и сертификатов держателя карточек;  
  - аутентификацию продавца и его возможности принимать платежи по пластиковым карточкам с применением цифровой подписи и сертификатов продавца;  
  - подтверждение того, что банк продавца является действующей организацией, которая может принимать платежи по пластиковым карточкам через связь с процессинговой системой (это подтверждение обеспечивается с помощью цифровой подписи и сертификатов банка продавца);  
  - готовность оплаты транзакций в результате аутентификации сертификата с открытым ключом для всех сторон;  
  - безопасность передачи данных посредством преимущественного использования криптографии.  
  Наиболее распространенный зарубежный опыт решения вопросов ключевого управления электронного финансового документооборота основывается на использовании Public Key Infrastructure (PKI) - Инфраструктуры Открытых Ключей (ИОК), названной таким образом по используемому способу защиты электронных документов - криптографии с открытыми ключами.  
  PKI подразумевает использование цифровых сертификатов и развернутой сети центров сертификации, обеспечивающих выдачу и сопровождение цифровых сертификатов для всех участников электронного обмена документами. По своим функциям цифровые сертификаты аналогичны обычной печати, которой удостоверяют подпись на бумажных документах.  
  Цифровые сертификаты - это определенная последовательность битов, основанных на криптографии с открытым ключом. Они представляют совокупность персональных данных владельца и открытого ключа его электронной подписи (а при необходимости и шифрования), связанных в единое неизменяемое целое электронной подписью центра сертификации. Цифровой сертификат оформляется в виде файла или области памяти и может быть записан на дискету, смарт-карточку и любой другой носитель данных.

Таблица 1. Сравнительная характеристика протоколов SSL и SET



  Цифровые сертификаты содержат открытые криптографические ключи абонентов, заверенные электронной цифровой подписью центра сертификации и обеспечивают однозначную аутентификацию участников обмена. Центры сертификации обеспечивают надежное распространение и сопровождение ключевой информации.

# Протокол защиты передачи данных в телекомуникационных сетях - IPSec

**IPSec** – Internet Protocol Security, комплекс, включающий в себя целый набор протоколов, которые обеспечивают безопасную передачу данных через сеть Интернет. К ним относятся: протокол установки безопасного канала связи IKE, протокол безопасности ESP, протоколы шифрования DES, 3DES, AES, RSA, протоколы проверки подлинности данных MD5, SHA-1 и протокол обмена ключами DH.

#### Протокол установки безопасного канала связи:

**IKE** – Internet Key Exchange, протокол, обеспечивающий установку безопасной сессии с удаленным устройством. IKE позволяет удаленным устройствам «договориться» о том, какой протокол безопасности использовать при передаче данных, каким способом зашифровать данные, с помощью какого протокола проверять подлинность переданных данных и какой алгоритм использовать для обмена ключами. Для обмена данными IKE использует протокол UDP, порт 500.

#### Протоколы безопасности:

**AH** – Authentication Header, протокол, обеспечивающий только проверку подлинности данных. В настоящий момент считается устаревшим.  
**ESP** - Encapsulating Security Payload, протокол, обеспечивающий не только проверку подлинности данных, но и их зашифрованную передачу через сеть Интернет с помощью тех же алгоритмов, которые использует IKE для установки безопасного соединения. Для обмена данными ESP использует протокол IP, порт 50.

**Протоколы шифрования:**

**DES** - Data Encryption Standard, протокол шифрования, разработанный IBM. Длина ключа для шифрования данных составляет 56 бит. С ростом вычислительных мощностей, сегодня данный протокол не является эталоном безопасности. На его смену пришел 3DES.  
**3DES** – Triple Data Encryption Standard, протокол, пришедший на смену DES. Использует три ключа DES для шифрования данных. Следовательно, для его работы требуется больше вычислительных ресурсов процессора.  
**AES** - Advanced Encryption Standard, новейший протокол шифрования данных. Алгоритм данного протокола использует длину ключа, которая составляет 128, 192 или 256 бит. Это один из самых ресурсоемких протоколов.  
**RSA** - Rivest, Shamir и Adleman, крипографический алгоритм, в основе которого лежит асимметричная система обмена ключами. Чаще всего используется в криптографических приложениях, наиболее известные из которых: SSH, SSL, HTTPS и др. Длина ключа составляет 512, 768, 1024 бит и более. Алгоритм RSA с длиной ключа 1024 бит является наиболее оптимальным соотношением между криптостойкостью и затратами вычислительной мощности процессора.

#### Протоколы проверки подлинности данных:

**MD5** - Message Digest 5, протокол хэширования данных. Длина хэша составляет 128 бит.

Хэш – число фиксированной длины, полученное после математического анализа блока данных определенной длины. Используется для проверки подлинности данных. Отправляющее устройство вычисляет хэш данных и ставит полученное значение в конец пакета. Принимающее устройство получает пакет с данными, проводит ту же процедуру и сравнивает полученное значение хэш с тем, что пришло вместе с пакетом. Если значения совпадают, значит данные подлинные, если значения не совпадают, значит данные были искажены в процессе доставки.

**SHA-1** - Secure Hash Algorithm 1, более эффективный протокол хэширования данных чем MD5. Длина хэша составляет 160 бит.

#### Протокол обмена ключами:

**DH** - Diffie-Hellman криптографический протокол, который позволяет двум сторонам через небезопасную сеть (например, Интернет) сгенерировать общий секретный ключ, который в последствии будет использоваться для шифрования данных между этими сторонами.  
Суть действия данного протокола заключается в том, что у каждого устройства есть 2 ключа - открытый и закрытый. Любым из этих ключей можно зашифровать данные, а другим – расшифровать. Закрытый ключ известен только тому устройству, которому он принадлежит. Открытый – любому устройству пожелавшему обменяться криптографическими данными.

Процесс проходит следующим образом. Предположим в сети есть два устройства – устройство А и устройство B, а между ними находится незащищенная сеть. Устройству А необходимо обменяться конфиденциальными данными с устройством B. Для этого устройство А отправляет устройству B свой открытый ключ. С помощью открытого ключа устройства А, устройство B шифрует данные и отправляет обратно устройству А. Получив зашифрованные данные, устройство А использует свой закрытый ключ, чтобы расшифровать их.

Криптостойкость алгоритма определяется размером ключа. DH группы 1 имеет размер ключа, равный 768 бит. Размер ключа DH группы 2 равен 1024 бит. Чем выше группа, тем более криптоскойким становится алгоритм, и тем больше ресурсов процессора он потребляет.

### Протоколы шифрования / расшифрования

В основе протокола этого класса содержится некоторый симметричный или асимметричный алгоритм шифрования/расшифрования. Алгоритм шифрования выполняется на передаче отправителем сообщения, в результате чего сообщение преобразуется из открытой формы в шифрованную. Алгоритм расшифрования выполняется на приёме получателем, в результате чего сообщение преобразуется из шифрованной формы в открытую. Так обеспечивается свойство конфиденциальности.

Для обеспечения свойства целостности передаваемых сообщений симметричные алгоритмы шифрования / расшифрования, обычно, совмещаются с алгоритмами вычисления имитозащитной вставки (ИЗВ) на передаче и проверки ИЗВ на приёме, для чего используется ключ шифрования. При использовании асимметричных алгоритмов шифрования / расшифрования свойство целостности обеспечивается отдельно путем вычисления электронной цифровой подписи (ЭЦП) на передаче и проверки ЭЦП на приёме, чем обеспечиваются также свойства безотказности и аутентичности принятого сообщения.

### Протоколы электронной цифровой подписи (ЭЦП)

В основе протокола этого класса содержится некоторый алгоритм вычисления ЭЦП на передаче с помощью секретного ключа отправителя и проверки ЭЦП на приёме с помощью соответствующего открытого ключа, извлекаемого из открытого справочника, но защищенного от модификаций. В случае положительного результата проверки протокол, обычно, завершается операцией архивирования принятого сообщения, его ЭЦП и соответствующего открытого ключа. Операция архивирования может не выполняться, если ЭЦП используется только для обеспечения свойств целостности и аутентичности принятого сообщения, но не безотказности. В этом случае, после проверки, ЭЦП может быть уничтожена сразу или по прошествии ограниченного промежутка времени ожидания.

### Протоколы идентификации / аутентификации

В основе протокола идентификации содержится некоторый алгоритм проверки того факта, что идентифицируемый объект (пользователь, устройство, процесс, …), предъявивший некоторое имя (идентификатор), знает секретную информацию, известную только заявленному объекту, причем метод проверки является, конечно, косвенным, то есть без предъявления этой секретной информации.

Обычно с каждым именем (идентификатором) объекта связывается перечень его прав и полномочий в системе, записанный в защищенной базе данных. В этом случае протокол идентификации может быть расширен до протокола аутентификации, в котором идентифицированный объект проверяется на правомочность заказываемой услуги.

Если в протоколе идентификации используется ЭЦП, то роль секретной информации играет секретный ключ ЭЦП, а проверка ЭЦП осуществляется с помощью открытого ключа ЭЦП, знание которого не позволяет определить соответствующий секретный ключ, но позволяет убедиться в том, что он известен автору ЭЦП.

### Протоколы аутентифицированного распределения ключей

Протоколы этого класса совмещают аутентификацию пользователей с протоколом генерации и распределения ключей по каналу связи. Протокол имеет двух или трёх участников; третьим участником является центр генерации и распределения ключей (ЦГРК), называемый для краткости сервером S. Протокол состоит из трёх этапов, имеющих названия: генерация, регистрация и коммуникация. На этапе генерации сервер S генерирует числовые значения параметров системы, в том числе, свой секретный и открытый ключ. На этапе регистрации сервер S идентифицирует пользователей по документам (при личной явке или через уполномоченных лиц), для каждого объекта генерирует ключевую и/или идентификационную информацию и формирует маркер безопасности, содержащий необходимые системные константы и открытый ключ сервера S (при необходимости).

На этапе коммуникации реализуется собственно протокол аутентифицированного ключевого обмена, который завершается формированием общего сеансового ключа.

## Задачи

* Обеспечение различных режимов аутентификации
* Генерация, распределение и согласование криптографических ключей
* Защита взаимодействий участников
* Разделение ответственности между участниками

## Разновидности атак на протоколы

* Атаки, направленные против криптографических алгоритмов
* Атаки против криптографических методов, применяемых для реализации протоколов
* Атаки против самих протоколов (активные или пассивные)

## Требования к безопасности протокола

1. Аутентификация (нешироковещательная):
   * аутентификация субъекта
   * аутентификация сообщения
   * защита от повтора
2. Аутентификация при рассылке по многим адресам или при подключении к службе подписки/уведомления:
   * неявная (скрытая) аутентификация получателя
   * аутентификация источника
3. [Авторизация](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) (доверенной третьей стороной)
4. Свойства совместной генерации ключа:
   * аутентификация ключа
   * подтверждение правильности ключа
   * защищенность от чтения назад
   * формирование новых ключей
   * защищенная возможность договориться о параметрах безопасности
5. [Конфиденциальность](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%84%D0%B8%D0%B4%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C)
6. Анонимность:
   * защита идентификаторов от прослушивания (несвязываемость)
   * защита идентификаторов от других участников
7. Ограниченная защищенность от атак типа «отказ в обслуживании»
8. Инвариантность отправителя
9. Невозможность отказа от ранее совершенных действий:
   * подотчётность
   * доказательство источника
   * доказательство получателя
10. Безопасное временное свойство

**Список литературы.**

1. Крейг Хант, "Персональные компьютеры в сетях TCP/IP", "BHV-Kиев", 384 стр., 1997 г.

ISBN: 6773300192

2. Лоу Д., "Компьютерные сети для "чайников", "Диалектика", 256 стр., 1996 г.

ISBN 5-85225-030-9 1-56884-079-9

3. Нанс Б., "Компьютерные сети", "Бином", 400 стр., 1996 г.

ISBN 5-89350-024-5 1-56529-824-1

4. Стен Шатт, "Мир компьютерных сетей", "BHV-Kиeв", 288 стр., 1996 г.

ISBN 5-7733-0028-1 0-07-057063-9

5. Золотов С., "Протоколы Internet", "BHV-Санкт-Петербург", 304 стр., 1998 г.

ISBN 5-7791-0076-4