УДК 004.057.4

АНАЛИЗ КРИПТОГРАФИЧЕСКИХ ПРОТОКОЛОВ УДАЛЕННОГО ДОСТУПА К ПК

К. Н. Ефименко[1], Д. В. Пауков[2]
[1]доцент кафедры прикладной математики, [2]магистрант гр. ПМКм-21
ГОУ ВПО «Донецкий национальный технический университет» (г. Донецк)
e-mail: KN_Efimenko@mail.ru, denshik591@gmail.com

Аннотация

Выполнен краткий сравнительный анализ основных характеристик, особенностей использования и преимуществ современных криптографических протоколов, обеспечивающих защищённую передачу данных между узлами в сети Интернет и при удаленном доступе к ПК. TLS/SSL – протокол для шифрования передаваемой информации и аутентификации. IPSec – набор протоколов для безопасного обмена информацией по сетевому протоколу IP. SSH – протокол для удаленного управления операционными системами и туннелирования. TCP-соединения. Описаны этапы установления соединения.

Введение

Сетевой протокол – это комплекс установок, благодаря которым определяется и регулируется процесс информационного обмена между компьютерами, подключенными к Интернету.

Функционирование сети основывается на работе сразу нескольких протоколов, располагаемых на разных уровнях. В настоящее время распространены две сетевых модели передачи данных.

TCP/IP – сетевая модель передачи данных, которая описывает способ передачи данных от источника информации к получателю. Модель предусматривает прохождение информации через четыре уровня, каждый из которых описывается протоколом передачи данных, на которых базируется Интернет.

OSI/ISO – сетевая модель стека сетевых протоколов, с помощью которой различные сетевые устройства взаимодействуют между собой. Модель предусматривает семь уровней взаимодействия систем. Каждый уровень выполняет определённые функции при таком взаимодействии.

Наиболее часто используемыми криптографическими протоколами, обеспечивающими защищённую передачу данных между узлами в сети Интернет, являются: TLS/SSL, IPSec и SSH-соединения. Каждый из перечисленных протоколов имеет свои особенности использования, достоинства и недостатки, но все они были приняты как Интернет-стандарт.

Целью данной работы является выполнение краткого анализа наиболее распространённых современных криптографических протоколов удаленного доступа к ПК, выбор и углубленное изучение которых должны помочь будущим специалистам в области прикладной математики стать востребованными на рынке труда.

Криптографические протоколы удаленого доступа

Проанализируем особенности использования и преимущества трех современных криптографических протоколов удаленного доступа.

1. Протокол TLS/SSL.

В настоящее время все более актуальным становится использование цифровых сертификатов. Особенностью данной технологии является использование протоколов TLS/SSL. На рынке IT-услуг появились компании, которые бесплатно предоставляют цифровые сертификаты всем желающим, чтобы гарантировать шифрование трафика между посещаемыми сайтами и браузером клиента.

Сетевые протоколы SSL и TLS являются криптографическими протоколами, обеспечивающими аутентификацию и защиту от несанкционированного доступа, нарушения целостности передаваемых данных. Протоколы SSL/TLS предназначены для исключения подмены идентификатора на клиентской или серверной стороне, раскрытия или искажения данных. Для этих целей используется надежный метод аутентификации, применяются шифрование канала связи и коды целостности сообщений.

Протокол TLS/SSL изначально был разработан компанией Netscape для защиты данных между сервисными и транспортными протоколами. Первая обнародованная версия была выпущена в 1995 году. Широко используется для VoIP-приложений, сервисов обмена мгновенными сообщениями. TLS/SSL представляет собой безопасный частный аутентифицированный ка-

нал. При транспортировке сообщений осуществляется проверка целостности с использованием MAC. Протокол TLS/SSL использует как симметричный, так и асимметричный ключи [1].

Протокол TLS/SSL обеспечивает решение двух задач – шифрование передаваемой информации и передача информации именно туда, куда требуется (аутентификация). Основное назначение протокола – предоставление надежного способа обмена данными между приложениями. Реализация SSL выполнена в виде многослойной среды, которая используется для безопасной передачи информации посредством незащищенных каналов связи.

Установка соединения обеспечивается в несколько этапов [2]:

1) Клиент устанавливает соединение с сервером и запрашивает защищенное подключение. Это может обеспечиваться либо установлением соединения на порт, который изначально предназначен для работы с SSL/TLS, например, 443, либо дополнительным запросом клиентом установки защищенного соединения после установки обычного.

2) При установке соединения клиент предоставляет список алгоритмов шифрования, которые он «знает». Сервер сверяет полученный список со списком алгоритмов, которые «знает» сам сервер, и выбирает наиболее надежный алгоритм, после чего сообщает клиенту, какой алгоритм использовать

3) Сервер отправляет клиенту свой цифровой сертификат, подписанный удостоверяющим центром, и открытый ключ сервера.

4) Клиент может связаться с сервером доверенного центра сертификации, который подписал сертификат сервера, и проверить, валиден ли сертификат сервера. Но может и не связываться. В операционной системе обычно уже установлены корневые сертификаты центров сертификации, с которыми сверяют подписи серверных сертификатов, например, браузеры.

5) Генерируется сеансовый ключ для защищенного соединения. Это делается следующим образом:

– клиент генерирует случайную цифровую последовательность;

– клиент шифрует ее открытым ключом сервера и посылает результат на сервер;

– сервер расшифровывает полученную последовательность при помощи закрытого ключа.

Учитывая, что алгоритм шифрования является асимметричным, расшифровать последовательность может только сервер. При использовании асимметричного шифрования используется два ключа – приватный и публичный. Публичным отправляемое сообщение шифруется, а приватным расшифровывается. Расшифровать сообщение, имея публичный, ключ нельзя.

6) Таким образом, устанавливается зашифрованное соединение. Данные, передаваемые по нему, шифруются и расшифровываются до тех пор, пока соединение не будет разорвано.

TLS/SSL имеет множество мер безопасности [3]:

– защита от понижения версии протокола к предыдущей (менее защищённой) версии или менее надёжному алгоритму шифрования;

– нумерация последовательных записей приложения и использование порядкового номера в коде аутентификации сообщения (MAC);

– использование ключа в идентификаторе сообщения (только владелец ключа может сгенерировать код аутентификации сообщения). Алгоритм вычисления кода аутентификации (HMAC), используемый во многих сессиях TLS, определён в RFC 2104;

– сообщение, которым заканчивается подтверждение связи («Finished»), используется для подтверждения аутентичности ранее переданных сообщений и, таким образом, выбранных параметров TLS-соединения.

В общем случае применение криптографии в протоколах TLS/SSL значительно снижает производительность приложений, зато обеспечивает надежную защиту передачи данных. Протоколы не требуют практически никаких настроек с клиентской стороны, считаются самыми распространенными протоколами защиты в сети интернет. Последними версиями протокола являются TLS 1.2, опубликованная в 2008 году и TLS 1.3 – 2018 года.

2. Протокол IPSec.

Протоколы прикладного уровня, поддерживающие шифрование (например, HTTPS) не могут охватить все возможные сценарии использования. Поэтому реализация шифрования на сетевом уровне обеспечивает большую гибкость.

IPSec (Internet Protocol Security) – набор (стек) протоколов для безопасного обмена информацией по сетевому протоколу IP. Позволяет осуществлять подтверждение подлинности (аутентификацию), проверку целостности и/или шифрование IP-пакетов. IPSec также включает в себя протоколы для защищённого обмена ключами в сети Интернет. IPSec является набором стандартов Интернета и своего рода «надстройкой» над IP-протоколом. В основном, применяется для организации VPN-соединений [3].

Базовой особенностью протокола является понятие SA (Security Association) – это набор параметров о том, как стороны будут в дальнейшем использовать те или иные свойства протоколов из состава IPSec.

Набор служб безопасности, которые может предоставлять IPSec, включает в себя контроль доступа, целостность без установления соединения, аутентификацию источника данных, отказ от повторных пакетов (форма частичной целостности последовательности), конфиденци-

альность (шифрование) и ограниченность конфиденциальности трафика. Поскольку эти услуги предоставляются на уровне IP, они могут использоваться любым протоколом более высокого уровня, например TCP, UDP и т.д. [4].

IPSec использует два протокола для обеспечения безопасности трафика:

– *AH* (Authentication Header) обеспечивает целостность без установления соединения, аутентификацию источника данных и дополнительную службу защиты от повтора. AH использует хэш-алгоритм для вычисления значения хэша, как для полезной нагрузки, так и для заголовка пакета, обеспечивая целостность пакета. Однако это вызывает очень специфическую проблему. AH не будет работать через NAT-устройство. NAT изменяет IP-заголовок пакета во время перевода, но значение хэша не изменяется. Таким образом, принимающее устройство будет полагать, что пакет был изменен при передаче и произойдет отклонение пакета;

– *ESP* (Encapsulating Security Payload) обеспечивает как полную, так и ограниченную конфиденциальность (шифрование) трафика, а также может обеспечивать подключение. Протокол обеспечивает целостность без установления соединения, аутентификацию источника данных и службу защиты от повтора. Таким образом, ESP выполняет шифрование и по своей сути более безопасен, чем AH. ESP вводит в пакет как дополнительный заголовок, так и трейлер. ESP также использует алгоритм хеширования для целостности данных. Однако хэш не включает IP-заголовок пакета, и, таким образом, ESP будет (обычно) работать через NAT-устройство.

Оба протокола AH и ESP являются транспортными средствами для контроля доступа на основе распределения криптографических ключей и управления потоками трафика по отношению к этим протоколам безопасности. Эти протоколы могут применяться отдельно или в сочетании друг с другом для обеспечения требуемого набора служб безопасности [4].

Каждый протокол поддерживает два режима: транспортный и режим туннеля. В транспортном режиме протоколы обеспечивают защиту в основном для протоколов верхнего уровня; в туннельном режиме протоколы применяются к туннелированным IP-пакетам. IPSec позволяет пользователю (или системному администратору) контролировать степень детализации, в которой предлагается служба безопасности. Например, можно создать один зашифрованный туннель для переноса всего трафика между двумя шлюзами безопасности или отдельный зашифрованный туннель, который может быть создан для каждого TCP-соединения между каждой парой хостов, взаимодействующих через эти шлюзы. Руководство IPSec должно включать средства для указания:

– какие службы безопасности использовать и в каких комбинациях;

– гранулярность, при которой должна применяться данная защита;

– алгоритмы, используемые для обеспечения криптографической безопасности [4].

Поскольку эти службы безопасности используют общие секретные значения (криптографические ключи), IPSec опирается на отдельный набор механизмов для размещения этих ключей (ключи используются для служб аутентификации/целостности и шифрования). Этот документ требует поддержки как ручного, так и автоматического распределения ключей. Он определяет конкретный подход на основе открытого ключа для автоматического управления ключами, но могут использоваться другие автоматизированные методы распространения ключей.

Каждый протокол IPSec (AH или ESP) может работать в двух режимах – транспортном и туннельном [4].

Транспортный режим – используется для шифрования поля данных IP пакета, содержащего протоколы транспортного уровня (TCP, UDP, ICMP). Исходные IP-заголовки остаются нетронутыми (рис. 1). Используется при обеспечении связи с одного устройства на другое одно устройство.

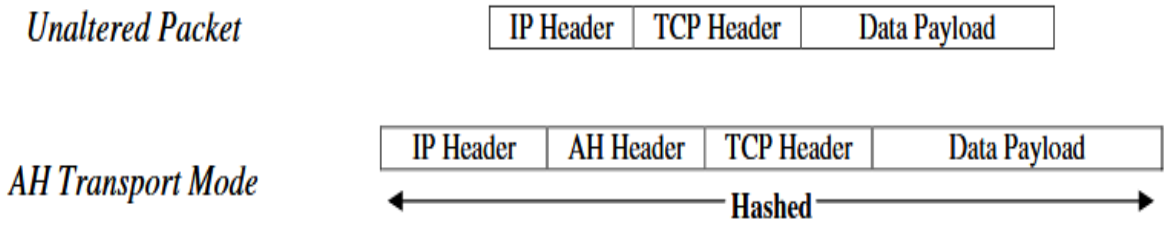

Рисунок 1 – Изменение IP-пакета протоколом AH

Туннельный режим – предполагает шифрование всего пакета, включая заголовок сетевого уровня. Во время транзита к пакету применяется временный IP-заголовок (рис. 2). Туннельный режим применяется в случае необходимости скрытия информационного обмена организации с внешним миром.

Взаимодействие двух узлов начинается с

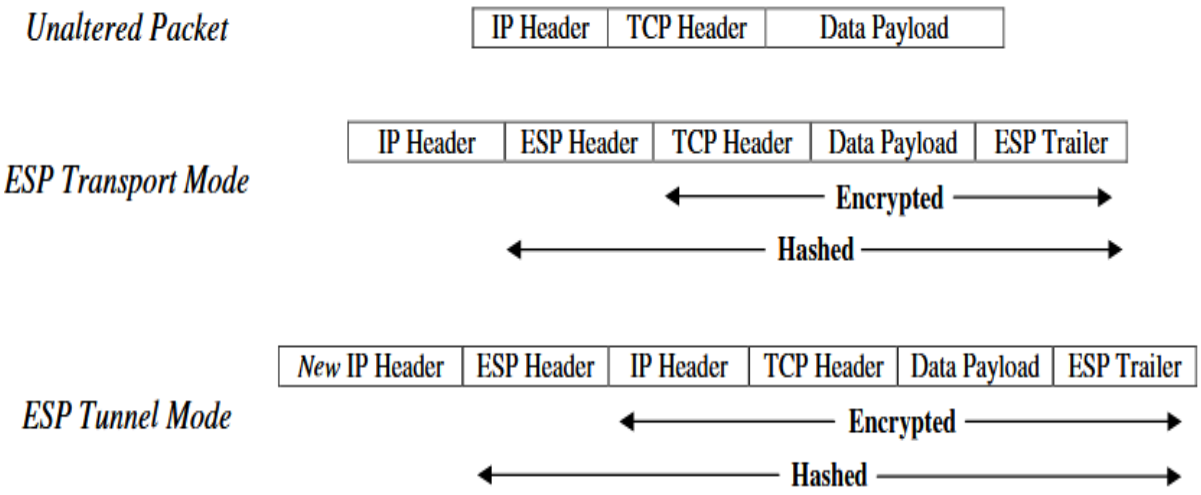

Рисунок 2 – Изменение IP-пакета протоколом ESP

установления SA. Точнее с двух согласований – для протокола AH и ESP причем в одну и в другую стороны. SA начинается с аутентификации и затем стороны согласовывают будущие параметры сессии [5]:

1) для протокола AH – используемый алгоритм аутентификации, ключи, время жизни ключей и другие параметры;

2) для протокола ESP – алгоритмы шифрования и аутентификации, ключи, параметры инициализации, время жизни ключей и другие параметры.

При этом стороны договариваются о туннельном или транспортном режиме работы IPSec.

Установка соединения проходит две фазы [5]:

На фазе 1 происходит установление SA первой фазы. В первой фазе стороны договариваются о методе идентификации, алгоритме шифрования, алгоритме хэширования и группе Диффи-Хеллмана. Эта фаза может пройти путем обмена тремя нешифрованными пакетами (агрессивный режим) или шестью нешифрованными пакетами – стандартный режим. Если все прошло успешно, то создается SA фазы 1 под названием IKE SA (IKE – протокол, связывающий все компоненты IPSec в работающее целое) и осуществляется переход ко второй фазе.

На фазе 2 стороны договариваются о политике и создаются сами ключи. Эта фаза, в отличие от первой полностью шифруется и она наступает только в случае успешного окончания первой фазы. В связи с тем, что трафик этой фазы полностью шифрован, становится сложно осуществлять поиск неполадок, однако если все прошло успешно, то создается SA фазы 2 под названием IPSec SA. В этот момент можно сказать, что туннель установлен.

Согласованные на двух фазах ключи должны работать оговоренное политикой время. Это означает, что сторонам возможно предстоит пережить процедуру смены ключей (rekeying), а иначе согласованные SA распадутся. Как было сказано выше, у сторон есть ключи в рамках процесса фазы 1 (IKE) и фазы 2 (IPsec). Процедуры их смены различны, как и таймеры, которые за это отвечают. Для того, чтобы не было перерыва связи в процессе смены ключей стороны сначала согласовывают параметры новой SA и лишь после этой успешной процедуры уничтожают старую SA [5].

Важным аспектом является способность IPsec устанавливать соединение заново. Для этого существуют настройки в ispec.conf, однако детали этих настроек могут отличаться от версии ПО.

3. Протокол SSH.

SSH (Secure Shell) – это протокол удаленного администрирования, разработанный для осуществления удаленного управления операционными системами и туннелирования TCP-соединения. Использование этого протокола допускает использование разных алгоритмов шифрования, что позволяет безопасно работать практически в любой незащищенной среде: работать с ПК через командную оболочку, передавать по шифрованному каналу любой тип данных [6]. Протокол относится к прикладному уровню сетевой модели передачи данных.

Протокол SSH организует безопасное соединение над небезопасными каналами передачи данных. Особенностью протокола является шифрование по выбранному алгоритму всего передаваемого трафика, в том числе и паролей.

Первый релиз протокола состоялся в 1995 г, а уже в 1996 г была представлена усовершенствованная его версия, которая и стала основой для дальнейшего развития продукта. Протокол SSH-1 имел некоторые ошибки в схеме обеспечения безопасности. Он был средством безопасной аутентификации. Протокол второй версии, то есть SSH-2, отвечает всем современным требованиям к шифрованию и обеспечивает

невозможность подмены трафика путем пересылки контрольных сумм [7].

SSH – это коммерческий продукт и предоставляется на платной основе. Присутствует и бесплатная версия – *OpenSSH*, которую использует большинство программистов. Обе версии схожи между собой командами. Релиз последнего обновления протокола OpenSSH 8.6 за апрель 2021 г. можно найти по ссылке http://www.openssh.com/txt/release-8.6.

Сегодня для всех сетевых ОС доступны SSH сервер и SSH клиент, а сам протокол SSH является одним из самых популярных решений для удаленного управления системами и передачи важной информации.

SSH-сервер обычно прослушивает соединения на TCP-порту 22. Для аутентификации сервера в SSH используется протокол аутентификации сторон на основе алгоритмов электронно-цифровой подписи RSA или DSA, но допускается также аутентификация при помощи и даже IP-адреса [3].

Аутентификация по паролю наиболее распространена. При каждом подключении подобно HTTPS вырабатывается общий секретный ключ для шифрования трафика.

При аутентификации по ключевой паре предварительно генерируется пара открытого и закрытого ключей для определённого пользователя. На машине, с которой требуется произвести подключение, хранится закрытый ключ, а на удалённой машине – открытый. Эти файлы не передаются при аутентификации, система лишь проверяет, что владелец открытого ключа также владеет и закрытым. При данном подходе, как правило, настраивается автоматический вход от имени конкретного пользователя в ОС.

Аутентификация по IP-адресу небезопасна, эту возможность чаще всего отключают.

Для создания общего сеансового ключа используется алгоритм Диффи-Хеллмана. Для шифрования передаваемых данных используется симметричное шифрование, алгоритмы AES, Blowfish или 3DES. Целостность передачи данных проверяется с помощью CRC32 в SSH1 или HMAC-MD5 в SSH2.

Для сжатия шифруемых данных может использоваться алгоритм LempelZiv (LZ77), который обеспечивает такой же уровень сжатия, что и архиватор ZIP. Сжатие SSH включается лишь по запросу клиента, и на практике используется редко [3].

Одной из основных особенностей протокола является туннелирование. SSH-туннель – это туннель, создаваемый посредством SSH-соединения и используемый для шифрования туннелированных данных. Используется для того, чтобы обезопасить передачу данных в Интернете (аналогичное назначение имеет IPsec). При пересылке через SSH-туннель незашифрованный трафик любого протокола шифруется на одном конце SSH-соединения и расшифровывается на другом.

Основным недостатком протокола SSH является то, что он не имеет средств защиты от действий злоумышленника, получившего root-доступ. Одной из мер предосторожности является запрет на удалённый root-доступ.

Один из наиболее популярных SSH-клиентов – программа Putty. Она используется для обеспечения удаленного доступа. Клиент FireSSH реализован как расширение браузера Firefox. Такой способ делает возможным использование программы в разных ОС. Для браузера Google Chrome есть другой официальный клиент – Secure Shell. Для использования в UNIX подходит программа OpenSSH. Для ее применения желателен опыт работы с командной оболочкой SSH. Для Windows подходит эмулятор терминалов Xshell, который позволяет отправлять команды нескольким серверам одновременно [6].

Алгоритм установления соединения в протоколе SSH можно разделить на три уровня, каждый из которых располагается над предыдущим: транспортный (открытие защищённого канала), аутентификация, подключение. В качестве предварительного уровня можно добавить уровень установки сетевого соединения, хотя официально этот уровень находится ниже SSH.

Выводы

При использовании SSL/TLS одним из основных методов является метод MITM (Man In The Middle – человек посередине). Этот метод основывается на использовании серверного сертификата и ключа на каком-то узле, который будет прослушивать трафик и расшифровывать информацию, которой обмениваются сервер и клиент. Для организации прослушивания можно использовать, например, программу sslsniff. Поэтому корневой сертификат и ключ обычно желательно хранить на машине, которая не подключена к сети, для подписания приносить запросы на подпись на флэшке, подписывать и так же уносить.

Использование IPSec протокола имеет как положительные стороны: высокая криптоустойчивость; возможность использования L2TP внутри IPsec для аутентификации по имени пользователя и паролю, так и отрицательные: сложен для настройки и поиска неисправностей; большие накладные расходы на передачу трафика в канале за счет заголовков.

Протокол IPSec используется, в основном, для организации VPN-туннелей. В этом случае протоколы ESP и AH работают в режиме туннелирования.

Использование SSH подключения имеет ряд преимуществ: безопасная работа на удаленном ПК с использованием командной оболочки;

использование разных алгоритмов шифрования (симметричного, асимметричного и хеширования); возможность безопасного использования любого сетевого протокола, что позволяет передавать по защищенному каналу файлы любого размера [6]. Таким образом, SSH – один из самых безопасных протоколов для реализации удаленного доступа к ПК, что делает его самым популярным вариантом для удаленного администрирования компьютеров и безопасной передачи данных.

Область применения протокола SSH практически не ограничена. Исходя из его основной функции – удаленного входа в операционную систему, протокол используют: системные администраторы для удаленной настройки компьютеров локальной сети; для настройки почтовых служб; для скрытого обмена внутри сети массивными файлами; для Интернет-игр.

Таким образом, без использования криптографических протоколов при удаленном доступе не возможна работа ни одной компании. Удаленный доступ требуется повсеместно как для сотрудников, так и для технического персонала.

Литература

1. Starlink [Электронный ресурс] / Режим доступа: https://www.starlink.ru/articles/ssl-tls. – Загл. с экрана.

2. Блог про Linux, Bash и другие информационные технологии [Электронный ресурс] / Режим доступа: https://mnorin.com/tls-ssl-neobhodimy-j-minimum-znanij.html. – Загл. с экрана.

3. Свободная энциклопедия Википедия [Электронный ресурс] / Режим доступа: https://ru.wikipedia.org. – Загл. с экрана.

4. Сайт lanmarket.ua [Электронный ресурс] / Режим доступа: https://lanmarket.ua/entsiklopediya/telekommunikatsionnye-tekhnologii/ipsec.html. – Загл. с экрана.

5. VPNKI [Электронный ресурс] / Режим доступа: https://ua.vpnki.ru/questions/ technologies/ ipsec. – Загл. с экрана.

6. Хостинг Freehost [Электронный ресурс] / Режим доступа: https://freehost.com.ua/faq/wiki/chto-takoe-ssh/. – Загл. с экрана.

7. Сила технологий [Электронный ресурс] / Режим доступа: https://ipipe.ru/info. – Загл. с экрана.

Ефименко К. Н., Пауков Д. В. Анализ криптографических протоколов удаленного доступа к ПК. *Выполнен краткий сравнительный анализ основных характеристик, особенностей использования и преимуществ современных криптографических протоколов, обеспечивающих защищённую передачу данных между узлами в сети Интернет и при удаленном доступе к ПК. TLS/SSL – протокол для шифрования передаваемой информации и аутентификации. IPSec – набор протоколов для безопасного обмена информацией по сетевому протоколу IP. SSH – протокол для удаленного управления операционными системами и туннелирования. TCP-соединения. Описаны этапы установления соединения.*

Ключевые слова: *анализ, криптографический протокол, TLS/SSL, IPSec, SSH.*

Efimenko K. N., Paukov D. V. Analysis of cryptographic protocols for remote PC access. *Performed a brief comparative analysis of the main characteristics, uses and advantages of modern cryptographic protocols that provide secure transmission of data between nodes on the Internet and with remote access to the PC. TLS/SSL – protocol for encrypting transmitted information and authentication. IPSec – a set of protocols for the secure exchange of information over a network protocol IP. SSH – protocol for remote control of operating systems and tunneling. The steps to establish a connection are described.*

Keywords: *analysis, cryptographic protocol, TLS/SSL, IPSec, SSH.*

Статья поступила в редакцию 20.11.2021
Рекомендована к публикации профессором Павлышом В. Н.