УДК 004.056.55, 004.057.4 doi: 10.15827/0236-235X.142.493-502 2023. Т. 36. № 3. С. 493–502

Разработка протокола передачи данных на основе комбинированного алгоритма их шифрования

О.А. Бакаева
Д.А. Барабошкин

Ссылка для цитирования
Бакаева О.А., Барабошкин Д.А. Разработка протокола передачи данных на основе комбинированного алгоритма их шифрования // Программные продукты и системы. 2023. Т. 36. № 3. С. 493–502. doi: 10.15827/0236-235X.142.493-502
Информация о статье
Поступила в редакцию: 21.02.23 После доработки: 31.03.23 Принята к публикации: 12.04.2023

Аннотация. Передача информации с технической точки зрения невозможна без используемых в них протоколов передачи данных. Одним из основных требований, предъявляемых к протоколам такого рода, является защита данных. Самый надежный метод, обеспечивающий защиту передаваемой информации по различным каналам связи, – шифрование данных. В статье проанализированы стандартные алгоритмы шифрования: AES, RSA, протокол Диффи–Хеллмана и функция хеширования данных SHA256. Выявлены их некоторые особенности, не позволяющие в полном объеме обеспечить максимальную защиту данных при передаче. Поэтому комбинированный алгоритм шифрования данных, суть которого в применении существующих алгоритмов на разных этапах шифрования, позволит избежать проблем, возникающих при использовании одного алгоритма. Предметом данного исследования является функционирование стандартных алгоритмов шифрования: AES, RSA, протокола Диффи–Хеллмана и функции хеширования данных SHA256. Основной результат работы – протокол передачи данных, созданный на основе комбинированного алгоритма шифрования данных. Протокол включает в себя разработку структуры пакета, реализацию процессов ClientResolving и Handshake, а также различные типы структур Payload. В конце осуществляется выбор параметров алгоритмов шифрования Диффи–Хеллмана и AES. Такая последовательность разработки позволила сделать протокол передачи данных универсальным и эффективным. В статье продемонстрирована работа протокола, включающая два этапа: установка соединения и непосредственно передача данных. Практическая значимость исследования заключается в том, что созданный протокол поможет обеспечить полноту, конфиденциальность и безопасность передачи данных любого типа – текст, графика, аудиофайл.
Ключевые слова: протокол передачи данных, комбинированный алгоритм шифрования данных, алгоритм AES, алгоритм RSA, протокол Диффи–Хеллмана, функция хеширования данных SHA256, процесс Handshake

Современный человек постоянно использует различные средства для обмена сообщениями, файлами, новостями, решает важные задачи, связанные с профессиональной деятельностью. Огромное количество самых разных конфиденциальных данных пропускается через эти системы. Можно осуществлять аудио- и видеозвонки, отправлять графические файлы, делать покупки, получать справочную информацию, заказывать различные товары и продукты питания, оплачивать услуги и многое другое. Все это стало неотъемлемой частью повседневной жизни.

При этом необходимо защищать передаваемую клиентом информацию и данные, потеря, изменение или модификация которых может нанести ущерб как крупным компаниям, так и физическому лицу.

Целью данной работы является исследование алгоритмов шифрования и протоколов передачи данных.

Поставлены следующие задачи:

– провести анализ применения существующих протоколов передачи данных;

– разработать комбинированный алгоритм шифрования данных;

– разработать протокол передачи данных на основе комбинированного алгоритма шифрования;

– написать программную часть для реализации процессов, организующих работу протокола.

В нашей стране с конца XX века применялись мощные криптографические алгоритмы «Магма» и позже «Кузнечик». Последний достаточно часто упоминают как аналогию криптоалгоритма AES – национального алгоритма шифрования США.

В работе [1] проведен сравнительный анализ российского и американского стандартов шифрования, который доказывает их неуязвимость и надежность в случае атак злоумышленников. В качестве основного критерия сравнения использована криптографическая стойкость алгоритма.

Однако, если рассматривать симметричный алгоритм AES с точки зрения шифрования данных, можно выделить ряд недостатков [2]:

– для шифрования и дешифрования используется один и тот же ключ;

– симметричные алгоритмы часто считаются менее безопасными, так как асимметричный алгоритм опирается на закрытый ключ для дешифрования и отдельный открытый ключ для шифрования файлов;

– более длинные ключи обеспечивают более высокую криптостойкость, поэтому для шифрования может потребоваться больше времени;

– симметричные алгоритмы требуют применения безопасного способа передачи ключа желаемому получателю.

В [3] отмечаются преимущества другого криптоалгоритма – RSA: возможность передачи приватной информации по незащищенным каналам связи без предварительной передачи секретных ключей и обеспечение цифровой подписи финансовых документов. В основе надежности RSA лежит задача разложения чисел на множители (задача факторизации). Так как в настоящее время не существует эффективного метода решения данной проблемы, алгоритм считается надежным. На этой особенности и базируется безопасность его использования. Однако во многих публикациях [4, 5] также отмечается, что неверный выбор параметров шифра RSA может привести к уменьшению его криптостойкости. В некоторых случаях открытый и закрытый ключи могут полностью совпасть, и тогда абонент случайно опубликует секретный ключ, что сделает криптосистему уязвимой.

До недавнего времени алгоритм, предложенный Диффи и Хеллманом (Diffie–Hellman, DH), считался самым безопасным и понятным. Он позволял двум сторонам создавать общий секретный ключ на основе открытых параметров друг друга без необходимости обмена секретными параметрами. Степень защиты зависит от сложности нахождения заданных целых чисел по определенному модулю. Сейчас алгоритм DH имеет функцию аутентификации (для борьбы с такими угрозами, как атака «человек посередине»). Впоследствии алгоритм DH стал основой других, более совершенных алгоритмов, таких как протоколы Нидхема–Шредера и MQV. Развитие протокола DH привело к созданию средств для включения трех сторон в генерацию сеансового ключа. С течением времени появилась угроза со стороны квантового компьютера. В связи с этим остро встал вопрос разработки более устойчивых алгоритмов генерации ключей шифрования. Так, например, в [6] приводится альтернативный протокол DH, который заключается в использовании десятичной части трансцендентных чисел, а не больших конечных целых чисел.

Многие специалисты по защите данных особо выделяют функцию хеширования данных SHA256 [7]. Ее основные достоинства состоят в однонаправленности в отличие от симметричных алгоритмов шифрования. У алгоритма SHA-256 также существует 2^{256} возможных вариантов хеш-значений, поэтому в случае какой-либо атаки, если производить перебор всех комбинаций для поиска нужной, на взлом потребуется большое количество операций и времени.

Эти алгоритмы зарекомендовали себя как эффективные и справляющиеся со своим классом задач. Но при этом каждый из них имеет ряд недостатков с точки зрения безопасности передачи и хранения данных пользователя. Таким образом, не существует универсального лучшего криптоалгоритма. Следовательно, необходимы дальнейшие научные изыскания в этой области.

Ценность алгоритмов заключается в том, что они являются основой создания протоколов передачи данных. А непосредственно сами протоколы используются практически во всех сферах жизни человека, где передаются данные (политика, экономика, банковское дело и биржа, образование и др.).

Во многих работах [8, 9] подчеркивается, что в настоящий момент остаются вопросы к надежности любого протокола (даже с несколькими видами ключей для шифрования передаваемой информации и разнообразием методов аутентификации), подтвержденные большим количеством уязвимостей [10]. Это проявляется в следующем:

– UDP: пакеты данных могут прийти не в полном объеме;

– FTP: используется только для передачи файлов;

– AMF: подвержен атакам;

– RTMP и RTSP: ограниченный функционал (передача видео);

– XMPP: зависимость от сервера и версии клиента;

– MTproto: логин и пароль при подключении к прокси передается в открытом виде.

Перечисленными недостатками алгоритмов и протоколов обусловлены следующие основные проблемы безопасности передачи данных:

– надежность аутентификации;

– создание и хранение public и private ключей;

– защищенность данных для авторизации пользователей и переписки;

– неудовлетворительная защищенность данных вследствие использования одного алгоритма шифрования данных;

– уязвимость алгоритма при атаках злоумышленников;

– утечка данных;

– передача информации в неполном объеме из-за обрыва канала.

Идеального протокола с точки зрения обеспечения абсолютной безопасности и конфиденциальности обмена сообщениями не существует, следовательно, разработка собственного протокола передачи данных с сервером с открытым исходным кодом может решить некоторые проблемы, связанные с безопасностью и хранением информации [11].

Практическая значимость работы заключается в разработке собственного протокола передачи данных. Это поможет обеспечить полноту, конфиденциальность и безопасность процесса передачи.

Анализ алгоритмов шифрования данных

В основе протоколов передачи данных лежат известные стандартные алгоритмы шифрования: AES, RSA, протокол DH и функция хеширования данных SHA256 [12].

Приведем краткую характеристику этих алгоритмов.

AES (AdvancedEncryptionStandards) – симметричный алгоритм блочного шифрования, который принят в качестве стандарта шифрования правительством США. Может обрабатывать данные блоками размером 128 бит, используя ключи шифрования длиной 128, 192 и 256 бит. Блоки определяют ввод открытого текста и вывод зашифрованного текста [2, 13].

RSA – криптографический алгоритм с открытым ключом, основанный на вычислительной сложности задачи факторизации больших целых чисел. Является самым распространенным асимметричным алгоритмом шифрования в мире. В обобщенном варианте содержит четыре шага: генерация ключей, распределение ключей, шифрование, расшифровка.

Криптосистема RSA используется для передачи ключей к симметричным криптосистемам, формирования цифровой подписи важных документов, создания почтовых клиентов [14].

Протокол DH – криптографический протокол, позволяющий двум и более сторонам получать общий секретный ключ, используя незащищенный от прослушивания канал связи. Полученный ключ используется для генерации ключей симметричного шифрования и дальнейшего обмена данными [12, 15].

Алгоритм SHA-256 (SecureHashAlgorithm 256-bit) – однонаправленная функция для создания цифровых отпечатков фиксированной длины (256 бит) из входных данных размером до 2,31 эксабайт (2^{64} бит), являющаяся частным случаем алгоритма из семейства криптографических алгоритмов SHA-2. С учетом уровня развития современных вычислительных мощностей алгоритм SHA-256 – одна из самых безопасных функций хеширования.

Разработка комбинированного алгоритма передачи пакета данных

Как известно, существуют различные пакеты передачи данных. Каждый из них отличается структурой и содержанием.

Прежде всего для установки соединения между клиентами, желающими обменяться информацией, осуществляется процесс Server Initialization – клиент отправляет пакет на сервер, тем самым показывая, что он находится в сети.

Условимся, что initiator и responder – это два отдельных клиента, которые будут общаться между собой.

После того как клиент прошел инициализацию с сервером, начинается процесс ClientResolving – первоначальное соединение клиентов, с помощью которого проверяется возможность установки связи. Он состоит из следующих действий:

– initiator отправляет пакет INIT_RESOLVE_REQUEST;

– responder отправляет пакет INIT_RESOLVE_RESPONSE;

– соединение установлено.

После успешного завершения ClientResolving начинается процесс handshake. Он состоит из трех шагов.

Шаг 1. Обмен public RSA-ключами между initiator и responder (рис. 1). Идентичность проверяется сравнением хеша от присланного public RSA собеседника с sender_id. Если они не совпадают, handshake прерывается [16].

Получение хеша и проверка целостности получаемого пакета осуществляются с помощью алгоритма SHA-256.

В процессе handshake происходят обмен public RSA-ключами между initiator и responder,

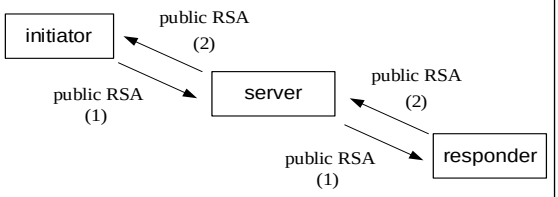

Рис. 1. Обмен RSA-ключами между инициатором и респондером

Fig. 1. RSA key exchange between an initiator and a respondent

генерация пар public и private DH keys на сторонах клиентов. Генерируется sharedkey с помощью пары private DH key и public DH key, создаются три криптоматериала. Они хешируются с помощью SHA-256, и от результатов хеша получаются пары encryption_key (AES key для initiator), decryption_key для responder и encryption_key для responder, decryption_key для initiator [17].

В процессе шифрования при отправке сообщения данные шифруются с помощью алгоритма AES.

Шаг 2. Обмен DH-ключами.

Данный алгоритм представляет собой последовательность действий [17]:

– на стороне инициатора генерируется пара public и private DH-ключей;

– public DH-ключ шифруется полученным public RSA собеседника;

– зашифрованный public DH-ключ отправляется собеседнику;

– сервер пересылает public DH-ключ собеседнику;

– на стороне респондера генерируется пара public и private DH-ключей;

– public DH-ключ шифруется полученным public RSA собеседника;

– зашифрованный public DH-ключ отправляется собеседнику, затем генерируется sharedkey с помощью private DH-ключа от респондера и public DH-ключа от инициатора;

– сервер пересылает public DH-ключ собеседнику.

Далее инициатор принимает сообщение сервера и генерирует sharedkey.

После предпоследней операции responder создает криптоматериал. К sharedkey добавляются байты 0, 1, 2:

1 – shared key | 0

2 – shared key | 1

3 – shared key | 2

Полученные криптоматериалы хешируются с помощью SHA-256, и от результатов хеша получаются:

1 – session_id;

2 – AES key (encryption_key для initiator, decryption_key для responder);

3 – AES key (encryption_key для responder, decryption_key для initiator).

После завершения шага 2 устанавливается защищенное соединение и начинается шаг 3 процесса Handshake, на котором между инициатором и респондером происходит обмен полученными session_id.

Шаг 3. Полученный session_id отправляется собеседнику. Собеседник отправляет свой полученный session_id. Происходит сравнение session_id собеседников. Если они одинаковые, соединение будет установлено. Если различные, соединение обрывается.

Для установки защищенного соединения были разработаны два пакета, представленные на рисунках 2 и 3 соответственно.

При отправке сообщения передаваемые данные шифруются с помощью алгоритма AES.

При делении содержимого пакета у всех его частей будет один и тот же message_id, различаться будет только part_id.

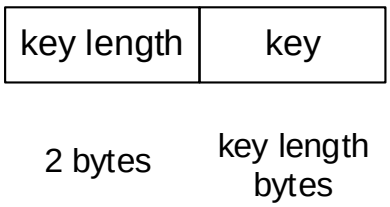

Рис. 2. Структура полезной нагрузки INIT_HANDSHAKE_REQUEST = 100

Fig. 2. INIT_HANDSHAKE_REQUEST payload structure

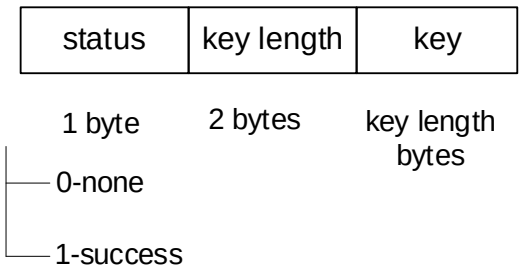

Рис. 3. Структура полезной нагрузки INIT_HANDSHAKE_RESPONSE = 101

Fig. 3. INIT_HANDSHAKE_RESPONSE payload structure

Алгоритм деления контента на части и отправки данных

Процесс составления пакетов.

1. Контент делится на части, то есть блоки (part_1, part_2, …, part_N-1, rest_part) размером 1 000 байт; если размер блока меньше, оставшаяся часть заполняется нулями (пример rest_part).

2. В начало блока добавляется поле block_size – его длина (длина полезной нагрузки).

3. Все блоки последовательно шифруются с помощью AES (encrypted_block).

4. Зашифрованная часть заполняется в PART_MESSAGE_PAYLOAD.

5. Пакет заполняется.

6. Полученный пакет отправляется на сервер, потом пересылается клиенту.

Проиллюстрируем процесс деления на примере отправляемого контента размером 13 337 байт, Part (batchsize) = 1 000 байт, Message_id в пределах одного содержимого будет один и тот же (рис. 4).

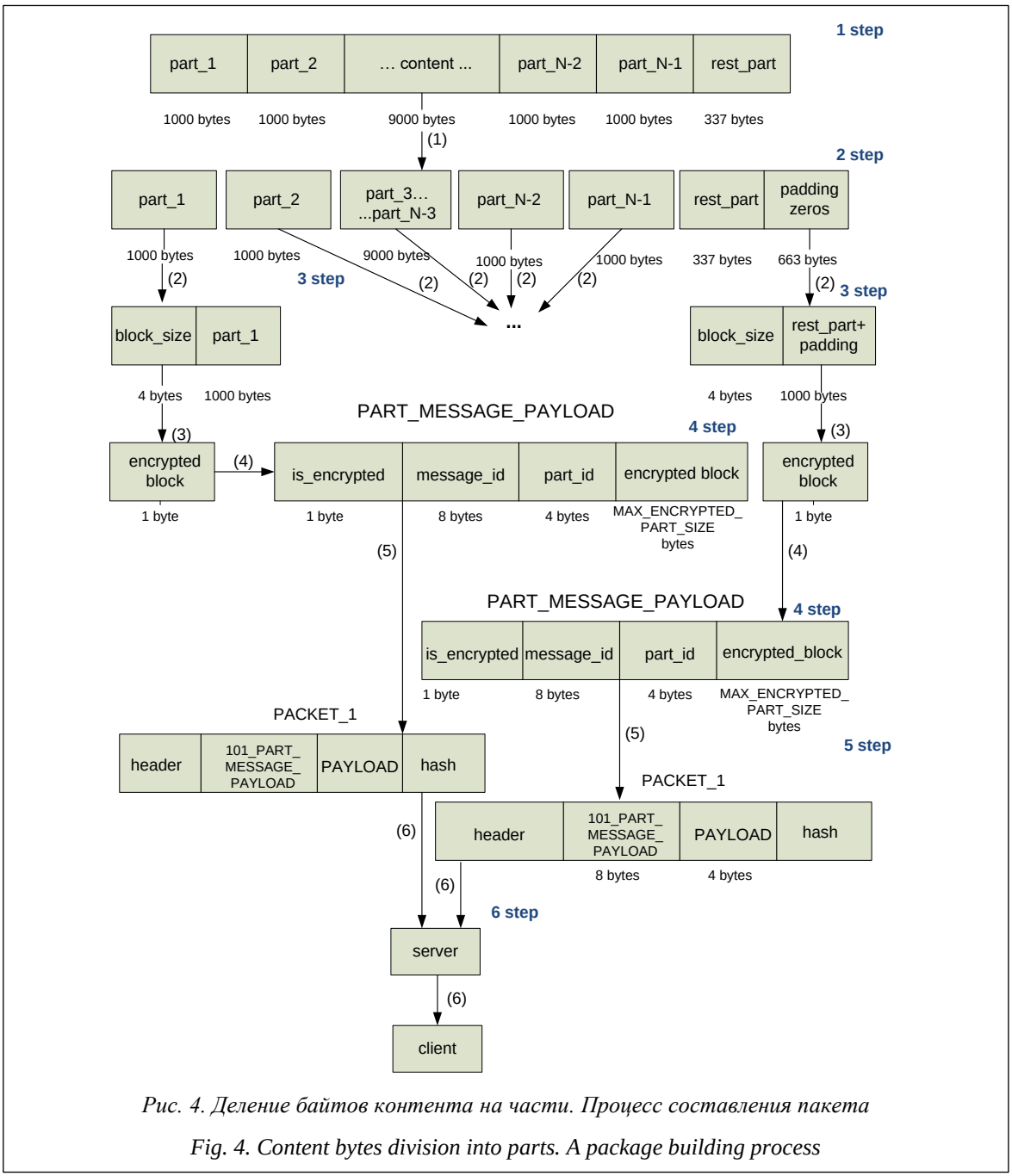

Рис. 4. Деление байтов контента на части. Процесс составления пакета

Fig. 4. Content bytes division into parts. A package building process

Алгоритм получения исходного контента

На стороне получателя по мере поступления зашифрованных пакетов происходит обратный процесс (рис. 5).

Алгоритм получения исходного контента.

1. Из пакета (PACKET_1, PACKET_2, …, PACKET_N) берется PAYLOAD.
2. Из блока PAYLOAD берется зашифрованная часть (encrypted_block) и расшифровывается.
3. Определяется количество байт полезной нагрузки, padding отделяется от полезной нагрузки, если она есть.
4. Из всех полученных блоков собирается исходное содержимое контента (порядок работы с блоками строго последователен).

Для осуществления процесса передачи сообщения были разработаны необходимые пакеты:

kFullMessage = 100

is_encrypted – **1 байт** (зашифровано ли сообщение): 0 – нет, 1 – зашифровано, остальные – discard

message_id – **8 байт** (id на отправку)

encrypted_block – **MAX_ENCRYPTED_PART_SIZE байт**

kPartMessage = 101

is_encrypted – **1 байт** (зашифровано ли сообщение): 0 – нет, 1 – зашифровано, остальные – discard

message_id – **8 байт** (id на отправку)

part_id – **4 байта** (при делении контента на части каждой части по порядку присваивается part_id)

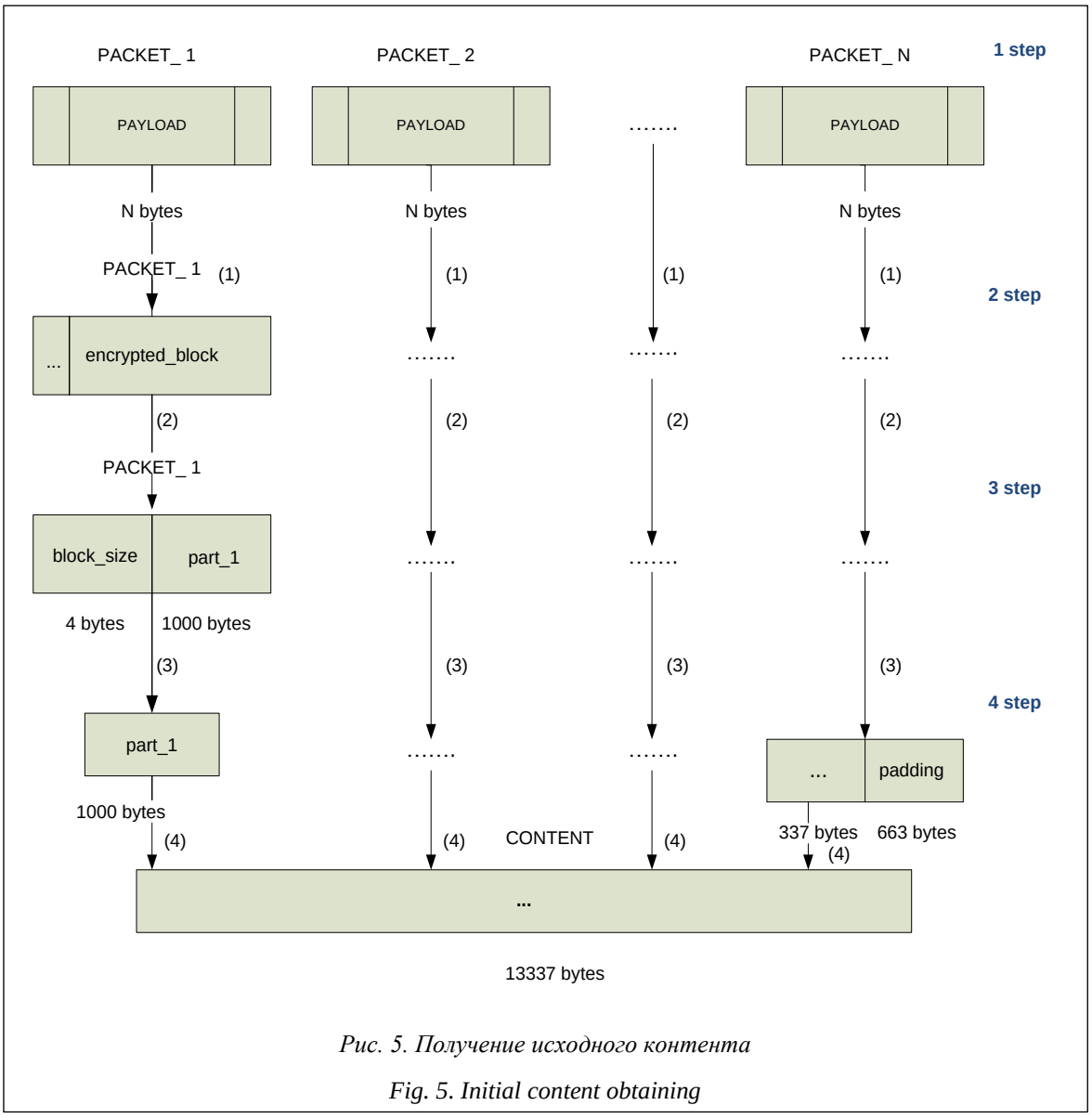

Рис. 5. Получение исходного контента

Fig. 5. Initial content obtaining

encrypted_block – **MAX_ENCRYPTED_PART_SIZE байт**

kPartMessageEnd = 102
message_id – **8 байт**

Комбинированный алгоритм основан на инициировании передачи RSA-ключей, а также DH-ключей, необходимых для осуществления процесса Handshake. Получение хеша и проверка целостности получаемого пакета осуществляются с помощью алгоритма SHA-256. В процессе Handshake используются асимметричный алгоритм шифрования RSA и криптографический протокол DH [17]. При отправке сообщения данные шифруются с помощью симметричного алгоритма блочного шифрования AES [18].

Выводы

Представленный протокол передачи данных имеет определенные преимущества:

– надежная авторизация благодаря созданию RSA-ключей при установке соединения (процесс Handshake);

– защищенность протокола, обусловленная тем, что сообщения шифруются с помощью алгоритма шифрования AES-256 в режиме CTR; изменение данных сообщения в процессе его отправки невозможно, так как в таком случае сообщение нельзя расшифровать на принимающей стороне из-за режима шифрования и проверки на хеш-сумму;

– повышение показателей безопасности передачи информации вследствие применения при создании комбинированного алгоритма шифрования на разных этапах алгоритмов DH, AES, RSA и функции хеширования SHA-256;

– невозможность попадания ключей в руки злоумышленников при атаках или попытках взлома сервера, поскольку ключи не хранятся на сервере (там находятся только хеши публичных ключей), private ключи находятся только у пользователей и никуда не передаются, а публичные ключи – это условно идентификационный номер пользователя.

В аналогичных протоколах данные для авторизации пользователя можно восстановить, следовательно, они хранятся на серверах. У разработанного протокола нет такого изъяна. Данные хранятся на сторонах пользователей. Владелец сервера не сможет получить данные авторизации пользователей и переписки. Сравнительный анализ протоколов передачи данных представлен в таблице.

Таким образом, разработанный алгоритм, включающий использование вышеупомянутых алгоритмов в указанной последовательности, позволяет обеспечить высокий уровень безопасности передачи данных и тем самым сохранить приватность пользователя.

Для разработки протокола были выбраны среда программирования VisualStudio и универсальный язык программирования C++.

Для реализации алгоритмов шифрования и создания комбинированного алгоритма использовалась бесплатная библиотека Crypto++ с открытым исходным кодом и криптографическими алгоритмами с возможностью гибкой настройки.

Демонстрация работы протокола передачи данных

После компиляции исходного кода запускаются exe-файлы – клиент и сервер протокола. При входе нужно или создать пару RSA-ключей, необходимых для процесса Handshake и генерации user_id, или использовать уже имеющиеся. Если вход осуществляется впервые, следует создать новую пару ключей. Соединение установлено. Созданы новые user_id пользователей (http://www.swsys.ru/uploaded/image/2023-3/2023-3-dop/2.jpg).

После успешной установки соединения у пользователей появятся папки self и other (http://www.swsys.ru/uploaded/image/2023-3/2023-3-dop/3.jpg). В них будут лежать RSA-ключи, необходимые для процесса Handshake (http://www.swsys.ru/uploaded/image/2023-3/2023-3-dop/4.jpg).

Из команд выбирается 1. Send text (t). Вводится сообщение, далее на клиентской и серверной сторонах отображается выполнение команды, осуществляется отправка зашифрованного текстового сообщения (http://www.swsys.ru/uploaded/image/2023-3/2023-3-dop/5.jpg).

Из команд выбирается 2. Sendfile (f). Отправляемый графический файл находится в папке files, далее на клиентской и серверной сторонах отображается выполнение команды, осуществляется отправка графического файла (http://www.swsys.ru/uploaded/image/2023-3/2023-3-dop/6.jpg). Полученный файл появится в корне папки, где находятся exe-файлы клиента и сервера.

Файл отправился. Полученный файл появился в корне папки, где находятся exe-файлы клиента и сервера (http://www.swsys.ru/uploaded/image/2023-3/2023-3-dop/30.jpg).

Сравнительный анализ протоколов передачи данных

A comparative analysis of data transfer protocols

Критерий	Стандартный протокол	Разрабатываемый протокол
Безопасность – защищенность данных (в целом)	Данные аутентификации и информация пользователей хранятся на сервере, что небезопасно	Сервер не хранит критически важные данные, что повышает безопасность
Безопасность – защищенность данных (на разных этапах)	Индивидуальна для каждого протокола. Уровень безопасности высокий, но протоколы не гарантируют 100-процентную защищенность данных (при атаках)	Структура генерации private и public ключей обеспечивает безопасность данных на этапе их создания. Организация процесса Handshake обеспечивает безопасность данных на этапе соединения пользователей
Использование собственного сервера	Частично используется в некоторых протоколах	Реализовано. Повышает безопасность хранения данных и сохраняет приватность пользователя
Использование алгоритмов шифрования внутри протокола	Использование одного алгоритма шифрования ненадежно и часто приводит к нарушению безопасности	Используется комбинированный алгоритм шифрования (DH, RSA, AES, SHA-256), который защищает от разных атак, в том числе и man in the middle
Случаи утечки информации	Были выявлены	При функциональном тестировании протокола не были выявлены (масштабное тестирование не проводилось)
Хранение данных для авторизации	Данные для авторизации хранятся на серверах	Данные для авторизации хранятся на сторонах пользователей
Типы передаваемых данных	Некоторые протоколы работают только с одним типом данных	Любые типы данных
Целостность	Имеется (через хеш)	Имеется (через хеш)

Как видно из демонстрации работы протокола передачи данных, соединение устанавливается успешно. Текстовый и графический файлы отправляются и получаются без потери информации, что свидетельствует о работоспособности полученного протокола.

Заключение

Многим пользователям важно знать, что их данные конфиденциальны и не передаются третьим лицам, а компаниям важна корпоративная защита внутренней информации. Не важно, какая информация передается – личная или служебная, она не должна потеряться, а тем более попасть в руки мошенников.

В основе созданного протокола лежит комбинированный алгоритм шифрования данных, суть которого в последовательном применении стандартных алгоритмов шифрования на разных этапах передачи. Представленный алгоритм позволяет обеспечить максимальный уровень безопасности передачи информации и тем самым сохранить приватность пользователя.

Собственный протокол передачи данных поможет сохранить конфиденциальную информацию и при этом быть уверенным в том, что она не передается третьим лицам и доставляется в полном объеме и исходном качестве.

Список литературы

1. Соболев М.А. Сравнительный анализ российского стандарта шифрования по ГОСТ Р 34.12–2015 и американского стандарта шифрования AES // Политехнический молодежный журнал. 2022. № 04. С. 1–13. doi: 10.18698/2541-8009-2022-4-785.

2. Архипова И.С. Криптографический алгоритм AES как средство защиты информации // Аллея науки. 2018. Т. 6. № 4. С. 83–87.

3. Алексеев А.П. Уязвимости алгоритма вычисления секретного ключа в криптосистеме RSA // Системы управления, связи и безопасности. 2015. № 3. С. 83–91. URL: https://www.elibrary.ru/download/elibrary_24252379_56555611.pdf (дата обращения: 10.02.2023).

4. Чиченин А.Д., Портнов Е.М. Недостатки использования алгоритма шифрования RSA // Интернаука. 2020. № 25-1. С. 11–12.

5. Квист Т.Д. Криптографический алгоритм RSA и его уязвимости при неправильном использовании // Научно-исследовательская работа обучающихся и молодых ученых: матер. 74-й Всерос. науч. конф. 2022. С. 530–533.

6. Коршев М.А. Альтернативный протокол Диффи–Хеллмана // Современные научные исследования и инновации. 2021. № 2. URL: http://web.snauka.ru/issues/2021/02/94553 (дата обращения: 28.02.2023).

7. Дремов И.С., Гирина А.Н. Использование алгоритма SHA-256 для хеширования данных // Тенденции развития науки и образования. 2022. Т. 86. № 1. С. 57–61. doi: 10.18411/trnio-06-2022-19.

8. Биджиева С.Х., Шебзухова К.В. Сетевые протоколы передачи данных: преимущества и недостатки // Тенденции развития науки и образования. 2022. Т. 86. № 1. С. 43–45. doi: 10.18411/trnio-06-2022-14.

9. Дементьев В.Е., Чулков А.А. Кибервоздействия на протоколы сетей передачи данных // Изв. ТулГУ. Технические науки. 2020. № 10. С. 245–254.

10. Лившиц И.И., Лонцих П.А. Формирование метрик для измерения результативности систем менеджмента информационной безопасности // Вестн. ИГТУ. 2016. Т. 112. № 5. С. 65–72. doi: 10.21285/1814-3520-2016-5-65-72.

11. Бакланова А.Р., Газизов А.Р. Применение защитных протоколов в системе передачи данных // Студенческий вестн. 2019. № 48-6. С. 29–31.

12. Барабошкин Д.А., Бакаева О.А. Анализ алгоритмов шифрования данных // За нами будущее: взгляд молодых ученых на инновационное развитие общества: сб. ст. науч. конф. 2022. Т. 2. С. 449–452.

13. Баймухамедов М.Ф., Жикеев А.А. Алгоритм шифрования AES как средство обеспечения информационной безопасности // Актуальные научные исследования в современном мире. 2019. № 9-1. С. 24–29.

14. Ибраимов Б.Э. Алгоритм шифрования RSA // TNS. Сб. избр. ст. по матер. науч. конф. ГНИИ «Нацразвитие»: матер. конф. 2019. С. 230–232.

15. Давтян А.В. Система открытого распределения ключей Диффи–Хеллмана // Лучшая исследовательская работа: сб. статей. 2021. С. 76–81.

16. Абелян В.З. Криптографический алгоритм RSA // IJASCSE. 2020. № 1. С. 4–10.

17. Барабошкин Д.А., Бакаева О.А. Разработка комбинированного алгоритма шифрования мультимедийных данных в процессе их передачи // Математическое моделирование, численные методы и комплексы программ: сб. тр. X Междунар. науч. молодежн. школы-семинара им. Е.В. Воскресенского. 2022. С. 27–31. URL: https://conf.svmo.ru/files/2022/papers/paper05.pdf (дата обращения: 28.02.2023).

18. Тищенко А.А., Лысов Д.А. Алгоритм шифрования AES, вопросы реализации и безопасности // Экономическая безопасность: правовые, экономические, экологические аспекты: сб. тр. 2016. С. 116–118.

Software & Systems doi: 10.15827/0236-235X.142.493-502 2023, vol. 36, no. 3, pp. 493–502

Developing a data transfer protocol based on a combined data encryption algorithm

Olga A. Bakaeva
Dmitry A. Baraboshkin

For citation
Bakaeva, O.A., Baraboshkin, D.A. (2023) ‘Developing a data transfer protocol based on a combined data encryption algorithm’, *Software & Systems*, 36(3), pp. 493–502 (in Russ.). doi: 10.15827/0236-235X.142.493-502
Article info
Received: 21.02.23 After revision: 31.03.23 Accepted: 12.04.2023

Abstract. From a technical point of view, information transfer is impossible without using data transfer protocols. One of the main requirements for such protocols is data protection. The most reliable method that ensures the protection of information transmitted over various communication channels is data encryption. The article analyzes standard encryption algorithms: AES, RSA, Diffie-Hellman protocol and SHA256 data hash function. It identifies some of their features that do not allow ensuring maximum data protection during their transfer completely. Therefore, the development of a combined data encryption algorithm, the essence of which is the use of existing algorithms at different stages of encryption, will help to avoid the problems that arise when using a single protocol. The research subject is functioning of standard encryption algorithms: AES, RSA, Diffie-Hellman protocol and SHA256 data hash function. The main result of the work is creating a data transfer protocol based on a combined data encryption algorithm. The protocol includes the development of the packet structure, the implementation of the Client Resolving and Handshake processes, as well as various types of Payload

structures. At the end, the parameters of the DH (Diffie-Hellman) and AES encryption algorithms are selected. This sequence of development made it possible to make this data transfer protocol universal and efficient. The article demonstrates the protocol operation that consists of two stages: establishing a connection and data transfer. The practical significance of the work is in the fact that the developed data transfer protocol will help to ensure the completeness, confidentiality and security of any type data transfer (text, graphics, audio file).
Keywords: data transfer protocol, combined data encryption algorithm, AES algorithm, RSA algorithm, Diffie-Hellman protocol, SHA256 data hashing function, Handshake process

Reference List

1. Sobolev, M.A. (2022) 'Comparative analysis of Russian GOST R 34.12-2015 encryption standard and American encryption standard AES', *Politechnical Student J.*, (04), pp. 1–13 (in Russ.). doi: 10.18698/2541-8009-2022-4-785.
2. Arkhipova, I.S. (2018) 'AES cryptographic algorithm as a means of information protection', *Alley of Science*, 6(4), pp. 83–87 (in Russ.).
3. Alekseev, A.P. (2015) 'Vulnerabilities algorithm for computing the secret key in the RSA cryptosystem', *Systems of Control, Communication and Security*, (3), pp. 83–91, available at: https://www.elibrary.ru/download/elibrary_24252379_56555611.pdf (accessed February 10, 2023) (in Russ.).
4. Chichenin, A.D., Portnov, E.M. (2020) 'Disadvantages of using the RSA encryption algorithm', *Interscience*, (25-1), pp. 11–12 (in Russ.).
5. Kvist, T.D. (2022) 'RSA cryptographic algorithm and its vulnerabilities when misusing', *Proc. Proc. 74th All-Russ. Sci. Conf. "Research Work of Students and Young Scientists"*, pp. 530–533 (in Russ.).
6. Korshev, M.A. (2021) 'Alternative Diffie–Hellman protocol', *Modern Sci. Researches and Innovations*, (2), available at: http://web.snauka.ru/issues/2021/02/94553 (accessed February 28, 2023) (in Russ.).
7. Dremov, I.S., Girina, A.N. (2022) 'Using the SHA-256 algorithm for hashing data', *Trends in the Development of Science and Education*, 86(1), pp. 57–61 (in Russ.). doi: 10.18411/trnio-06-2022-19.
8. Bidzhieva, S.Kh., Shebzukhova, K.V. (2022) 'Network data transfer protocols: Advantages and disadvantages', *Trends in the Development of Science and Education*, 86(1), pp. 43–45 (in Russ.). doi: 10.18411/trnio-06-2022-14.
9. Dementev, V.E., Chulkov, A.A. (2020) 'Cyber attacks on data network protocols', *Proc. of the TSU. Tech. Sci*., (10), pp. 245–254 (in Russ.).
10. Livshits, I.I., Lontsikh, P.A. (2016) 'Formation of metrics to measure information security management system efficiency', *Proc. of ISTU*, 112(5), pp. 65–72 (in Russ.). doi: 10.21285/1814-3520-2016-5-65-72.
11. Baklanova, A.R., Gazizov, A.R. (2019) 'Application of security protocols in the data transmission system', *Student Bull*., (48-6), pp. 29–31 (in Russ.).
12. Baraboshkin, D.A., Bakaeva, O.A. (2022) 'Analysis of data encryption algorithms', *Proc. Sci. Conf. The Future is Behind us: A View of Young Scientists on the Innovative Development of Society*, 2, pp. 449–452 (in Russ.).
13. Baimukhamedov, M.F., Zhikeev, A.A. (2019) 'AES encryption algorithm as a means of ensuring information security', *Actual Sci. Research in the Modern World*, (9-1), pp. 24–29 (in Russ.).
14. Ibraimov, B.E. (2019) 'RSA encryption algorithm', *Proc. TNS. Collection of Selected Articles Based on the Materials of Sci. Conf. of the GNII "National Development"*, pp. 230–232 (in Russ.).
15. Davtyan, A.V. (2021) 'Diffie–Hellman public key distribution system', *Proc. Best Research Paper*, pp. 76–81 (in Russ.).
16. Abelyan, V.Z. (2020) 'RSA cryptographic algorithm', *IJASCSE*, (1), pp. 4–10 (in Russ.).
17. Baraboshkin, D.A., Bakaeva, O.A. (2022) 'The development of a combined algorithm for encrypting multimediadata during transmission', *Proc. X Int. Sci. Youth School-seminar Math. Modeling, Numerical Methods and Software Packages*, pp. 27–31, available at: https://conf.svmo.ru/files/2022/papers/paper05.pdf (accessed February 28, 2023) (in Russ.).
18. Tishchenko, A.A., Lysov, D.A. (2016) 'AES encryption algorithm, implementation and security issues', *Proc. Int. Conf. Economic Security: Legal, Economic, Environmental Aspects*, pp. 116–118 (in Russ.).

Авторы
Бакаева Ольга Александровна [1], к.т.н.,
доцент кафедры систем автоматизированного проектирования, helga_rm@rambler.ru
Барабошкин Дмитрий Александрович [1],
магистр, torcktaaer@yandex.ru

[1] Национальный исследовательский Мордовский государственный университет им Н.П. Огарева,
г. Саранск, 430005, Россия

Authors
Olga A. Bakaeva[1], Ph.D. (Engineering),
Associate Professor,
helga_rm@rambler.ru
Dmitry A. Baraboshkin [1], Master of Science,
torcktaaer@yandex.ru

[1] National Research N.P. Ogarev Mordovian State University, Saransk,
430005, Russian Federation