Модификации протокола Д-Х

Оглавление

[Введение 3](#_Toc480560352)

[Протокол Дииффи-Хеллмана 4](#_Toc480560353)

[Алгоритм 4](#_Toc480560354)

[Пример 5](#_Toc480560355)

[Алгоритм Диффи-Хеллмана с тремя и более участниками 6](#_Toc480560356)

[Безопасность протокола 7](#_Toc480560357)

[Мan-in-middle 7](#_Toc480560358)

[Криптографические системы на эллиптических кривых 8](#_Toc480560359)

[Разделение сообщений 10](#_Toc480560360)

[ЭЦП в качестве защиты 11](#_Toc480560361)

[Заключение 13](#_Toc480560362)

[Источники 14](#_Toc480560363)

Введение

Передача ключа по открытым каналам была большой проблемой криптографии XX века. Но эту проблему удалось решить после появления алгоритма Диффи-Хеллмана. Данный алгоритм позволил дать ответ на главный вопрос: «Как при обмене зашифрованными посланиями уйти от необходимости передачи секретного кода расшифровки, который, как правило, не меньше самого послания?» Открытое распространение ключей Диффи-Хеллмана позволяет паре пользователей системы выработать общий секретный ключ, не обмениваясь секретными данными.

Схема открытого распределения ключей, предложенная Диффи и Хеллманом, произвела настоящую революцию в мире шифрования, так как снимала основную проблему классической криптографии — проблему распределения ключей.

Протокол Дииффи-Хеллмана

Протокол Дииффи-Хеллмана (англ. Diffie-Hellman, DH) — криптографический протокол, позволяющий двум и более сторонам получить общий секретный ключ, используя незащищенный от прослушивания канал связи. Полученный ключ используется для шифрования дальнейшего обмена с помощью алгоритмов симметричного шифрования.

Алгоритм был впервые опубликован Уитфилдом ДиффиМартином Хеллманом в 1976 году.

В протоколе Диффи-Хеллмана две стороны создают симметричный ключ сеанса. Перед установлением симметричного ключа эти две стороны должны выбрать два числа p и g. Первое число, p, является большим простым числом порядка 300 десятичных цифр (1024 бита). Второе число, g, служит генератором порядка p - 1 в группе <Zp\*, x >. Эти два числа (группа и генератор) не должны быть конфиденциальными. Их можно передать через Internet. Они могут быть общедоступны.

Алгоритм

1. Алиса выбирает большое случайное число x, такое, что 0 < x < p - 1, и вычисляет R1 = gx mod p.
2. Боб выбирает другое большое случайное число y, такое, что 0 < y < p -1, и вычисляет R2 = gy mod p.
3. Алиса передает Бобу R1. Обратите внимание, что Алиса не передает значение x ; она передает только R1.
4. Боб передает Алисе R2. Снова обратите внимание, что Боб не передает значение y, он передает только R2.
5. Алиса вычисляет K = (R2)x mod p.
6. Боб также вычисляет K = (R1)y mod p.

*K = (gx mod p)y mod p = (gymod p)x mod p = gxy mod p* (1)

Боб вычисляет K = (R1)y mod p = (gx mod p)y mod p = gxy mod p,   
Алиса вычисляет K = (R2)x mod p = (gy mod p)x mod p = gxy mod p

и получает то же самое значение без Боба, знающего значение x. А Боб получил это значение без Алисы, знающей значение y.

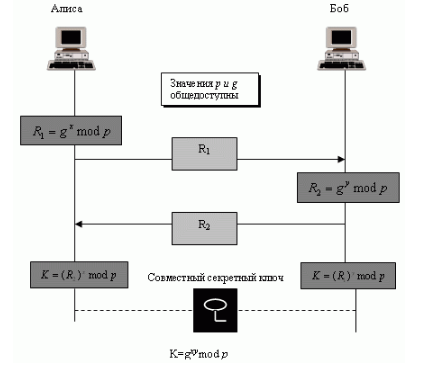


Рис.1

Число р должно удовлетворять условиям:

* p является случайным простым числом
* (p-1)/2 также должно быть случайным простым числом (для повышения безопасности)
* g является первообразным корнем по модулю p (также является простым числом)

Пример:

Приведем тривиальный пример, чтобы ясно понять процедуру. Наш пример использует маленькие числа, но заметим, что в реальной ситуации применяются очень большие числа. Предположим, что g = 7 и p = 23. Тогда процедура содержит следующие шаги.

1. Алиса выбирает x = 3 и вычисляет R1 = 73 mod 23 = 21.
2. Боб выбирает y = 6 и вычисляет R2 = 76 mod 23 = 4.
3. Алиса передает число 21 Бобу.
4. Боб передает число 4 Алисе.
5. Алиса вычисляет симметричный ключ K = 43 mod 23 = 18.
6. Боб вычисляет симметричный ключ K = 216 mod 23 = 18.

Значение K одно и то же и для Алисы, и для Боба: gxy mod p = 718 mod 23 = 18.

Алгоритм Диффи-Хеллмана с тремя и более участниками

Использование алгоритма Диффи-Хеллмана не ограничивается двумя участниками. Он может быть применен на неограниченное количество пользователей. Рассмотрим ситуацию, когда Алиса, Боб и Кэрол вместе генерируют исходный ключ. В данном случае последовательность действий будет следующая:

(Все вычисления производятся по модулю p)

1. Стороны договариваются о параметрах алгоритма p и g
2. Стороны, Алиса, Боб и Кэрол генерируют свои ключи — a, b и c соответственно.
3. Алиса вычисляет ga и посылает его Бобу
4. Боб вычисляет (ga)b = gab и посылает его Кэрол
5. Кэрол вычисляет (gab)c = gabc и получает тем самым общий секретный ключ
6. Боб вычисляет gb и посылает его Кэрол
7. Кэрол вычисляет (gb)c = gbc и посылает его Алисе
8. Алиса вычисляет (gbc)a = gbca = gabc — общий секретный ключ
9. Кэрол вычисляет gc и посылает его Алисе
10. Алиса вычисляет (gc)a = gca и посылает его Бобу
11. Боб вычисляет (gca)b = gcab = gabc и также получает общий секретный ключ

Если кто-то будет прослушивать канал связи, то он сможет увидеть ga, gb, gc, gab, gac, и gbc, но при этом не сможет воспроизвести gabc используя любые комбинации этих чисел.

Для того чтобы данный алгоритм был эффективно применен для большой группы людей, необходимо соблюдение двух основных принципов:

* Передача ключа должна начинаться с «пустого» ключа g. Весь секрет состоит в повышении текущего значения показателя каждого участника один раз;
* Любое промежуточное значение может быть раскрыто публично, но окончательное значение представляет из себя секретный ключ, который никогда не должен быть публично раскрыт. Таким образом, каждый пользователь получает свою копию тайного ключа и передает его последующему.

Безопасность протокола

Стойкость разделенного ключа в протоколе Диффи–Хеллмана обеспечивается вычислением значения K = gab modp по заданным числам ga и gb. Эта задача называется вычислительной проблемой Диффи–Хеллмана (CDH problem —Diffie-Hellman problem).

Проблема Диффи–Хеллмана опирается на сложность дискретного логарифмирования. (Дискретное логарифмирование аналогично обычному логарифмированию в поле действительных чисел. Однако в отличие от последней задачи, в которой решение является приближенным, задача о вычислении дискретного логарифма имеет точное решение.)

Проблема Диффи–Хеллмана и задача дискретного логарифмирования считаются неразрешимыми в конечной абелевой группе большого порядка, например, в подгруппе конечного поля, имеющей большой простой порядок, или в группе точек эллиптической кривой, определенной над конечным полем. Таким образом, протокол обмена ключами Диффи–Хеллмана правильно работает в этих группах.

Следует заметить, что данный алгоритм, как и все алгоритмы асимметричного шифрования, уязвим для атак типа "man-in-the-middle" ("человек в середине"). Если противник имеет возможность не только перехватывать сообщения, но и заменять их другими, он может перехватить открытые ключи участников, создать свою пару открытого и закрытого ключа и послать каждому из участников свой открытый ключ. После этого каждый участник вычислит ключ, который будет общим с противником, а не с другим участником.

Мan-in-middle

Протокол Диффи-Хеллмана отлично противостоит пассивному нападению, но в случае реализации атаки «человек посередине» он не устоит. В самом деле, в протоколе ни Алиса, ни Боб не могут достоверно определить, кем является их собеседник, поэтому вполне возможно представить следующую ситуацию, при которой Боб и Алиса установили связь с Меллори, который Алисе выдает себя за Боба, а Бобу представляется Алисой. И тогда вместо протокола Диффи-Хеллмана получаем, что-то похожее на следующее:



Рис.2

То есть Меллори получает один ключ общий с Алисой (которая считает, что это Боб) и один ключ общий с Бобом (который считает, что это Алиса). А, следовательно, он может получать от Алисы любое сообщение для Боба, расшифровать его ключом, прочитать, зашифровать ключом и передать Бобу. Таким образом, подлог может оставаться незамеченным очень долгое время.

Атака на протокол обмена ключами Диффи–Хеллмана вполне реальна, поскольку этот протокол не предусматривает проверки аутентичности источника сообщений. Для того чтобы ключи были согласованы только между Алисой и Бобом, обе стороны должны быть уверены друг в друге.

Решениями данной проблемы являются, в частности, использование криптосистем на эллиптических кривых или использование ЭЦП.

Криптографические системы на эллиптических кривых

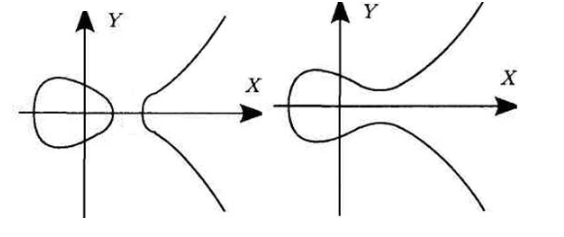
В 1985 году американские ученые Н. Коблиц (Neal Koblitz) и В. Миллер (Victor Miller) предложили использовать для криптосистем с открытым ключом теорию эллиптических кривых. Дальнейшие исследования подтвердили наличие подходящих свойств у этих математических функций и привели к созданию реальных криптографических систем, использующих математический аппарат эллиптических кривых. С 1998 года использование эллиптических кривых для решения криптографических задач, таких, как цифровая подпись, было закреплено в стандартах США ANSI X9.62 и FIPS 186-2, а в 2001 году аналогичный стандарт, ГОСТ Р34.10-2001, был принят и в России.

Основное достоинство криптосистем на эллиптических кривых состоит в том, что по сравнению с другими асимметричными криптосистемами, рассмотренными нами ранее, они обеспечивают существенно более высокую криптостойкость при равных затратах на обработку и вычисления. Это объясняется тем, что вычисление обратных функций на эллиптических кривых значительно сложнее, чем, например, вычисление дискретных логарифмов (алгоритмы Диффи-Хеллмана и Эль-Гамаля) или решение задачи факторизации (алгоритм RSA). В результате тот уровень стойкости, который достигается, скажем, в RSA при использовании 1024-битовых модулей, в системах на эллиптических кривых реализуется при размере модуля 160 бит, что обеспечивает более простую как программную, так и аппаратную реализацию.

В криптографии используются эллиптические кривые на плоскости, определяемые уравнениями вида:

*Y2= X3+ аХ + b mod р* (2)

где р – некоторое большое простое число, а a и b – константы. График эллиптической кривой при разных значениях параметров а и b имеет вид:

  
Рис.3 (Варианты графиков эллиптических кривых)

Принцип использования эллиптических кривых следующий. Для группы пользователей выбирается общая эллиптическая кривая Е и некоторая точка G на ней. Закрытым ключом пользователя выступает некоторое целое число с, а открытым – точка D на кривой Е, полученная в результате специального преобразования композиции с использованием числа с. Параметры кривой и список открытых ключей абонентов, как и обычно, передаются всем пользователям сети. Открытые и закрытые ключи пользователей используются для выполнения операций шифрования и расшифрования в зависимости от назначения алгоритма.

С помощью эллиптических кривых могут быть реализованы многие известные протоколы с открытым ключом. Любая криптосистема, основанная на дискретном логарифмировании, легко может быть перенесена на эллиптические кривые. Например, можно заменить математические операции вида у = gхmod р на операции математического аппарата эллиптических кривых (операции вычисления композиции точек) в алгоритмах формирования ключа Диффи-Хеллмана. В результате получатся те же алгоритмы, но с другими математическими операциями.

Несмотря на сложность математического аппарата эллиптических кривых, существуют эффективные вычислительные методы, позволяющие достаточно быстро реализовывать необходимые расчеты. За счет использования модуля меньшей длины операции генерации ключей и шифрования выполняются быстрее, чем, скажем, в алгоритме RSA или классическом алгоритме Диффи-Хеллмана. Криптографические методы на эллиптических кривых считаются перспективными и, закрепленные в различных стандартах, находят применение в современных системах защиты информации.

Разделение сообщений

На практике разработано несколько способов предотвращения атаки "man-in-the-middle". Один из способов заключается в разделении каждого зашифрованного сообщения на две части, каждая из которых бесполезна без другой. Части сообщения пересылаются по очереди и не могут быть расшифрованы по отдельности. Вот как может выглядеть этот протокол для обмена сообщениями между двумя пользователями А и Б:

1. Пользователи А и Б обмениваются открытыми ключами.
2. Пользователь А шифрует свое сообщение открытым ключом пользователя Б и пересылает половину зашифрованного сообщения пользователю Б.
3. Пользователь Б шифрует свое сообщение открытым ключом пользователя А и пересылает половину зашифрованного сообщения пользователю А.
4. Пользователь А пересылает вторую половину зашифрованного сообщения пользователю Б.
5. Пользователь Б соединяет обе полученные половины сообщения от пользователя А и расшифровывает его своим закрытым ключом. Затем посылает вторую половину своего зашифрованного сообщения пользователю А.
6. Пользователь А складывает полученные от пользователя Б половины сообщения и расшифровывает его своим закрытым ключом.

Этот усовершенствованный протокол не позволит злоумышленнику читать или изменять корреспонденцию пользователей А и Б. Нарушитель, как и раньше, может подменить открытые ключи абонентов, а также перехватить передаваемые между ними данные. Однако, получив на шаге 2 протокола в свое распоряжение первую половину зашифрованного сообщения от А к Б, он не сможет расшифровать ее своим закрытым ключом и снова зашифровать открытым ключом абонента Б. Абоненты А и Б тоже не смогут прочитать сообщения до окончания протокола (шагов 5 и 6), но в этом нет ничего плохого, так как в результате они получат корректную корреспонденцию. Для осуществления протокола процесс разделения сообщения на две части может производиться разными способами, например, каждый нечетный байт помещается в первое сообщение, а каждый четный – во второе или как-то иначе.

Атаки "человек-в-середине" можно избежать и другими способами, например, добавляя к передаваемым открытым ключам цифровые подписи специального удостоверяющего центра.

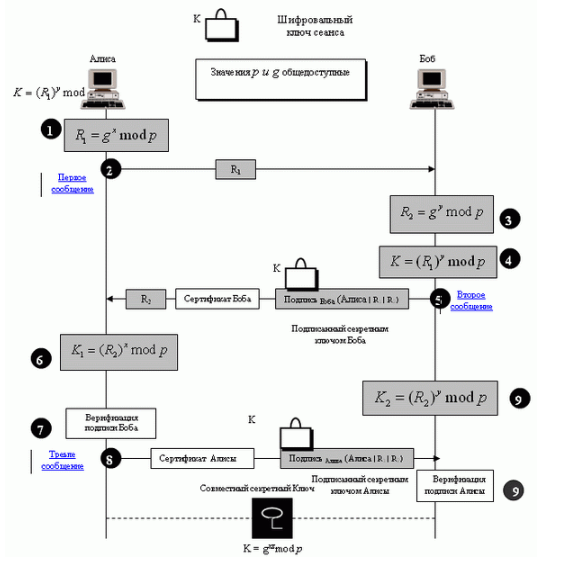
ЭЦП в качестве защиты

Как бороться с такой уязвимостью? Самый логичный и простой ответ: нужна взаимная аутентификация. И тут на помощь приходит ЭЦП.

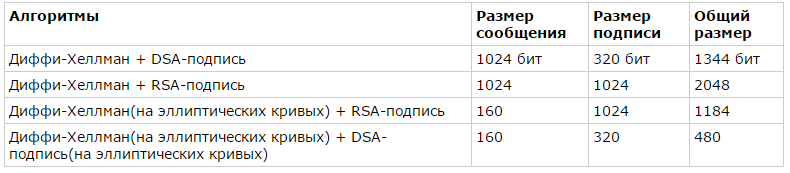
Если у Боба имеется открытый ключ Алисы, и он уверен на сто процентов, что это действительно ключ Алисы, то тогда для защиты от атаки «человек посередине» Алисе достаточно подписать своим закрытым ключом число на шаге 1. Теперь Меллори может сколько угодно пытаться выдать себя за Алису, но подделать ее подпись он не сможет, а Боб догадается, что что-то не так.

Вариант того, как выглядит алгоритм при использовании ЭЦП:

1. После вычисления R1 Алиса передает R1 Бобу
2. После вычисления R2 и ключа сеанса Боб конкатенирует ID Алисы, R1 и R2. Затем он подписывает результат своим секретным ключом. Боб теперь передает R2, подпись и собственное свидетельство общедоступного ключа Алисе. Подпись зашифрована ключом сеанса.
3. После вычисления ключа сеанса, если подпись Боба проверена, Алиса связывает ID Боба, R1 и R2. Затем она подписывает результат своим собственным секретным ключом и передает это Бобу. Подпись зашифрована ключом сеанса
4. Если подпись Алисы проверена, Боб сохраняет ключ сеанса

  
Рис.4

Казалось бы, решение найдено, протокол доработан, уязвимость устранена. Однако в данном случае есть одно «но»: чрезмерное увеличение размера сообщений за счет добавления подписи. Наглядно этот эффект демонстрирует следующая таблица:



(DSA (англ. Digital Signature Algorithm - алгоритм цифровой подписи) — криптографический алгоритм с использованием открытого ключа для создания электронной подписи, но не для шифрования (в отличие от RSA и схемы Эль-Гамаля). Подпись создается секретно, но может быть публично проверена. Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность. Алгоритм основан на вычислительной сложности взятия логарифмов в конечных полях.)

Заключение

Считается, что алгоритм Диффи-Хеллмана до сих пор превзойти никто не может. Собственно, именно он послужил основой для возникновения таких известных систем защиты в области шифрования данных, как AES128 и AES256.

Для безопасности протокола важным является выбор параметров. Многие реализации используют небольшое количество популярных наборов параметров алгоритма. В 2016 была представлена работа, показавшая возможность по подготовке специальных конечных полей для алгоритма Диффи — Хеллмана (DH). Выбранное исследователями простое число p специального вида (размером 1024 бита) выглядит обычным для пользователей, но упрощает на несколько порядков сложность вычислений по методу SNFS для решения задачи дискретного логарифмирования. Для борьбы с атакой предлагается увеличить размер модуля до 2048 бит.

Однако вообще, в чистом виде алгоритм Диффи-Хеллмана уязвим для модификации данных в канале связи, в том числе для атаки «Человек посередине», поэтому схемы с его использованием применяют дополнительные методы односторонней или двусторонней аутентификации, т.е. системы электронно-цифровых подписей.

Источники

* <http://fb.ru/article/225952/algoritm-diffi-hellmana-naznachenie>
* <https://habrahabr.ru/post/100950/>
* <https://ru.wikipedia.org/wiki/Протокол_Диффи_—_Хеллмана>
* <http://www.intuit.ru/studies/courses/691/547/lecture/12391?page=4>
* <http://www.intuit.ru/studies/courses/553/409/lecture/9383?page=4>
* <http://www.securitylab.ru/analytics/478912.php>
* <http://www.williamspublishing.com/PDF/5-8459-0847-7/part.pdf>