Модификации протокола Д-Х

Оглавление

[Введение 3](#_Toc480129578)

[Протокол Дииффи-Хеллмана 4](#_Toc480129579)

[Алгоритм 4](#_Toc480129580)

[Пример: 5](#_Toc480129581)

[Алгоритм Диффи-Хеллмана с тремя и более участниками 6](#_Toc480129582)

[Безопасность протокола 7](#_Toc480129583)

[Мan-in-middle 7](#_Toc480129584)

[ЭЦП в качестве защиты 8](#_Toc480129585)

[Заключение 10](#_Toc480129586)

[Источники 11](#_Toc480129587)

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.

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.

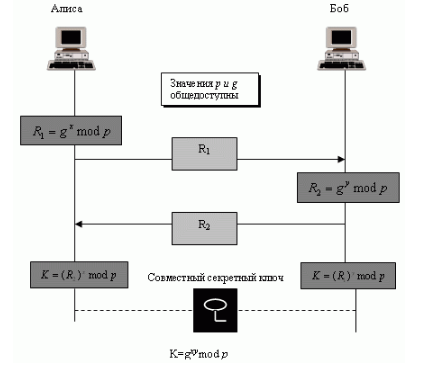


Рис.1

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

Криптографическая стойкость алгоритма Диффи-Хеллмана (то есть сложность вычисления K = gab modp по известным p, g, A = ga modp и B = gb modp), основана на предполагаемой сложности проблемы дискретного логарифмирования.

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.

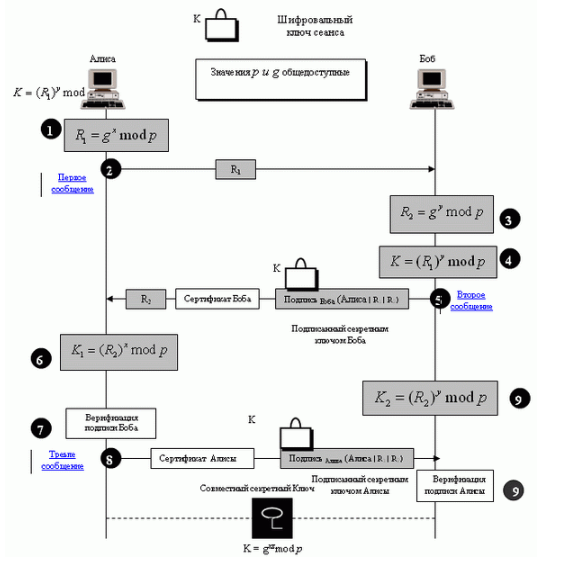
ЭЦП в качестве защиты

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

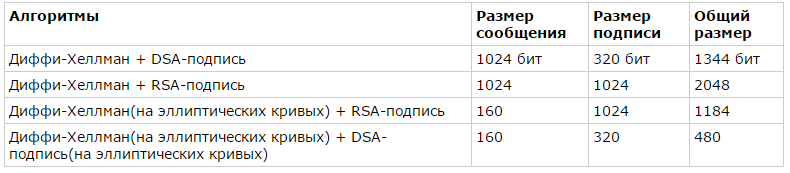
Если у Боба имеется открытый ключ Алисы, и он уверен на сто процентов, что это действительно ключ Алисы, то тогда для защиты от атаки «человек посередине» Алисе достаточно подписать своим закрытым ключом число на шаге 1. Теперь Меллори может сколько угодно пытаться выдать себя за Алису, но подделать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

Имеются следующие шаги:

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.3

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:



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://ru.wikipedia.org/wiki/Протокол_Диффи_—_Хеллмана>
* <http://www.intuit.ru/studies/courses/553/409/lecture/9383?page=4>
* <https://habrahabr.ru/post/100950/>