Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

КАФЕДРА № 51

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук		А.М.Буланов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Криптографические протоколы

по курсу: Криптографические методы защиты информации

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.№	5721		Ковалева А.Е.
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2020

Цель работы: Реализовать протокол совместной выработки сеансового ключа Diffie-Hellman.

Задание:

Разработать двух независимых модулей-участников протокола.
Реализовать атаку «человека по середине»

Описание протокола:

Протокол Ди́ффи — Хе́ллмана (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 чистом виде алгоритм Диффи — Хеллмана уязвим для модификации данных в канале связи, в том числе для атаки «Man-in-the-middle (человек посередине)», поэтому схемы с его использованием применяют дополнительные методы односторонней или двусторонней аутентификации.

Описание алгоритма

Предположим, существует два абонента: Алиса и Боб.

Обоим абонентам известны некоторые два числа g и p, которые не являются секретными и могут быть известны также другим заинтересованным лицам.

Для того, чтобы создать неизвестный более никому секретный ключ, оба абонента генерируют большие случайные числа: Алиса — число a, Боб — число b.

Затем Алиса вычисляет остаток от деления (1): $A = g^a \pmod{p}$ и пересылает его Бобу, а Боб вычисляет остаток от деления (2): $B = g^b \pmod{p}$ и передаёт Алисе.

Предполагается, что злоумышленник может получить оба этих значения, но не модифицировать их (то есть, у него нет возможности вмешаться в процесс передачи).

На втором этапе Алиса на основе имеющегося у неё a и полученного по сети B вычисляет значение (3): $B^a \pmod{p} = g^{ab} \pmod{p}$

Боб на основе имеющегося у него b и полученного по сети A вычисляет значение (4): $A^b \pmod{p} = g^{ab} \pmod{p}$

Как нетрудно видеть, у Алисы и Боба получилось одно и то же число (5): $K = g^{ab} \pmod{p}$

Его они и могут использовать в качестве секретного ключа, поскольку здесь злоумышленник встретится с практически неразрешимой (за разумное время) проблемой вычисления (3) или (4) по перехваченным $g^a \pmod{p}$ и $g^b \pmod{p}$, если числа p, a, b выбраны достаточно большими.

Работа алгоритма показана на рисунке 1:

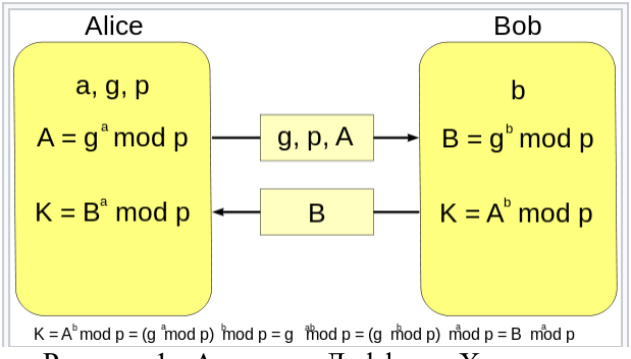

Рисунок 1 - Алгоритм Диффи — Хеллмана,
где K — итоговый общий секретный ключ

При работе алгоритма каждая сторона:

1. генерирует случайное натуральное число a — закрытый ключ
2. совместно с удалённой стороной устанавливает открытые параметры p и g (обычно значения p и g генерируются на одной стороне и передаются другой), где
   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)
3. вычисляет открытый ключ A, используя преобразование над закрытым ключом
   $$A = g^a \pmod{p}$$
4. обменивается открытыми ключами с удалённой стороной
5. вычисляет общий секретный ключ K, используя открытый ключ удаленной стороны B и свой закрытый ключ a
   $$K = B^a \pmod{p}$$
   K получается равным с обеих сторон, потому что:
   $$B^a \pmod{p} = (g^b \bmod p)^a \bmod p = g^{ab} \pmod{p} = (g^a \bmod p)^b \bmod p = A^b \pmod{p}$$

В практических реализациях для a и b используются числа порядка 10100 и p порядка 10300. Число g не обязано быть большим и обычно имеет значение в пределах первого десятка.

Криптографическая стойкость алгоритма

Криптографическая стойкость алгоритма Диффи — Хеллмана (то есть сложность вычисления $K = g^{ab} \pmod{p}$ по известным p, g, $A = g^a \pmod{p}$ и $B = g^b \pmod{p}$ основана на предполагаемой сложности задачи дискретного логарифмирования.

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 случай, при котором 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:

Шаг	Алиса	Меллори	Боб
1	$g^a \rightarrow$	g^a	
2	$g^n \leftarrow$	g^n	
	g^{an}	g^{an}	
3		$g^m \rightarrow$	g^m
4		$g^b \leftarrow$	g^b
		g^{mb}	g^{mb}

То есть Меллори получает один общий ключ с Алисой (которая считает, что это Боб) и один общий ключ 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 человек по середине

1 Описание атаки

Атака посредника, или атака «человек посередине» (англ. Man in the middle (MITM)) — вид атаки в криптографии и компьютерной безопасности, когда злоумышленник тайно ретранслирует и при необходимости изменяет связь между двумя сторонами, которые считают, что они непосредственно общаются друг с другом. Является методом компрометации канала связи, при котором взломщик, подключившись к каналу между контрагентами, осуществляет вмешательство в протокол передачи, удаляя или искажая информацию.

Одним из примеров атак типа «человек посередине» является активное прослушивание, при котором злоумышленник устанавливает независимые связи с жертвами и передаёт сообщения между ними. Тем самым он заставляет жертв поверить, что они разговаривают непосредственно друг с другом через частную связь, фактически же весь разговор управляется злоумышленником. Злоумышленник должен уметь перехватывать все передаваемые между двумя жертвами сообщения, а также вводить новые. В большинстве случаев это довольно просто, например, злоумышленник может вести себя как «человек посередине» в пределах диапазона приёма беспроводной точки доступа (Wi-Fi).

Данная атака направлена на обход взаимной аутентификации или отсутствие таковой и может увенчаться успехом только тогда, когда злоумышленник имеет возможность выдать себя за каждую конечную точку либо оставаться незамеченным в качестве промежуточного узла.

2 Принцип атаки

Атака обычно начинается с прослушивания канала связи и заканчивается тем, что криптоаналитик пытается подменить перехваченное сообщение, извлечь из него полезную информацию, перенаправить его на какой-нибудь внешний ресурс.

Предположим, объект A планирует передать объекту B некую информацию.

Объект C обладает знаниями о структуре и свойствах используемого метода передачи данных, а также о факте планируемой передачи собственно информации, которую C планирует перехватить.

Для совершения атаки C «представляется» объекту A как B, а объекту B — как A.

Объект A, ошибочно полагая, что он направляет информацию B, посылает её объекту C.

Объект C, получив информацию и совершив с ней некоторые действия (например, скопировав или модифицировав в своих целях), пересылает данные собственно получателю — B; объект B, в свою очередь, считает, что информация была получена им напрямую от A.

3 Пример атаки на алгоритмическом языке

Предположим, что Алиса хочет передать Бобу некоторую информацию. Мэлори хочет перехватить сообщение и, возможно, изменить его так, что Боб получит неверную информацию.

Мэлори начинает свою атаку с того, что устанавливает соединение с Бобом и Алисой, при этом они не могут догадаться о том, что кто-то третий присутствует в их канале связи. Все сообщения, которые посылают Боб и Алиса, приходят Мэлори.

Алиса просит у Боба его открытый ключ. Мэлори представляется Алисе Бобом и отправляет ей свой открытый ключ. Алиса, считая, что это ключ Боба, шифрует им сообщение и отправляет его Бобу.

Мэлори получает сообщение, расшифровывает, затем изменяет его, если нужно, шифрует его открытым ключом Боба и отправляет его ему. Боб получает сообщение и думает, что оно пришло от Алисы:

1. Алиса отправляет Бобу сообщение, которое перехватывает Мэлори:

   Алиса *«Привет, Боб, это Алиса. Пришли мне свой открытый ключ.»* → Мэлори Боб
2. Мэлори пересылает сообщение Бобу; Боб не может догадаться, что это сообщение не от Алисы:

   Алиса Мэлори *«Привет, Боб, это Алиса. Пришли мне свой открытый ключ.»* → Боб
3. Боб посылает свой ключ:

   Алиса Мэлори ← *[ключ Боба]* Боб
4. Мэлори подменяет ключ Боба своим и пересылает сообщение Алисе:

   Алиса ← *[ключ Мэлори]* Мэлори Боб
5. Алиса шифрует сообщение ключом Мэлори, считая, что это ключ Боба, и только он может расшифровать его:

   Алиса *«Встречаемся на автобусной остановке!» [зашифровано ключом Мэлори]* → Мэлори Боб
6. Мэлори расшифровывает сообщение, читает его, модифицирует его, шифрует ключом Боба и отправляет его:

   Алиса Мэлори *«Жди меня у входа в музей в 18:00.» [зашифровано ключом Боба]* → Боб

7. Боб считает, что это сообщение Алисы.

   Этот пример демонстрирует необходимость использования методов для подтверждения того, что обе стороны используют правильные открытые ключи, то есть что у стороны A открытый ключ стороны B, а у стороны B — открытый ключ стороны A. В противном случае, канал может быть подвержен атаке «человек посередине».

4 Атака на протокол Диффи-Хеллмана

Рассмотрим атаку на протокол выработки общего секрета Диффи-Хеллмана между сторонами ***A*** и ***B***. Допустим, криптоаналитик ***E*** имеет возможность не только перехватывать сообщения, но и подменять их своими, то есть осуществлять активную атаку: $A \longleftrightarrow E \longleftrightarrow B$

5 Перехват и подмена ключей

1. Сторона ***A*** отправляет сообщение стороне ***B***: $A \overset{E}{\rightarrow} B: \ g^x \mod p.$
2. Криптоаналитик ***E*** перехватывает сообщение стороны ***A*** и подменяет его, отправляя стороне ***B*** уже другое сообщение : $E \rightarrow B: \ g^z \mod p.$
3. Сторона ***B*** отправляет сообщение стороне ***A***: $A \overset{E}{\leftarrow} B: \ g^y \mod p.$
4. Криптоаналитик ***E*** перехватывает сообщение стороны ***B*** и подменяет его, отправляя стороне ***A*** какое-то своё сообщение : $A \leftarrow E: \ g^z \mod p.$
5. Результатом данных действий является образование двух каналов связи криптоаналитика ***E*** со сторонами ***A*** и ***B***, причем сторона ***A*** считает что общается со стороной ***B*** при помощи секретного ключа K_{AE} , а сторона ***B*** отправляет сообщения при помощи ключа K_{BE}.

   При этом стороны ***A*** и ***B*** не подозревают, что обмен сообщениями происходит не напрямую, а через криптоаналитика ***E***: $K_{AE} = g^{xz} \mod p,$ $K_{BE} = g^{yz} \mod p.$

6 Подмена сообщений

1. Сторона ***A*** отправляет сообщение m стороне ***B***, зашифрованное при помощи ключа K_{AE}: $A \overset{E}{\to} B : E_{K_{AE}}(m).$
2. Криптоаналитик ***E*** перехватывает это сообщение, расшифровывает ключом K_{AE}, при необходимости, изменяет его на m', зашифровывает ключом K_{BE} и отправляет стороне ***B***: K_{AE}: $E \to B : E_{K_{BE}}(m').$
3. Аналогичные действия криптоаналитик ***E*** предпринимает при передаче сообщений от ***B*** к ***A***.

Таким образом, криптоаналитик ***E*** получает возможность перехватывать и подменять все сообщения в канале связи.

При этом, если содержимое сообщений не позволяет выявить наличие в канале связи третьей стороны, то атака «человек посередине» считается успешной.

Описание программы:

В классе Alice хранятся такие переменные, как:

private BigInteger **p**;// *открытый параметр, простое число*
private BigInteger **g**;// *открытый параметр, первообразный корень по модулю p (простое число)*
private BigInteger **a**; *// закрытый ключ, случайное натуральное число*
private BigInteger **A**; *// открытый ключ Алисы*
private BigInteger **sessionKey**; *//общий сессионный ключ*

В классе Bob хранятся такие переменные, как:

private final BigInteger **p**; //*сгенерировано на стороне Алисы и передано Бобу*
private final BigInteger **g**; //*сгенерировано на стороне Алисы и передано Бобу*
private BigInteger **b**; *// закрытый ключ Боба*
private BigInteger **B**; *// открытый ключ Боба*
private BigInteger **sessionKey**; *//общий сессионный ключ*

Так как мы на стороне Алисы генерируем p и g, то в классе Alice содержится метод:

private void generation_p_g()//*генерирует числа p и g*

Помимо этого, есть такие методы как:

public BigInteger Public_Key() // *возвращает открытый ключ A*
public void generation_Session_Key(BigInteger B) //*генерирует общий ключ как* $K = B^a \pmod{p}$

public BigInteger SessionKey() //*возвращает общий ключ*

В классе Bob есть следующие методы:

public void SecretKey()// *генерирует закрытый ключ по формуле* $B = g^b \pmod{p}$

public BigInteger PublicKey()// *возвращает открытый ключ B*
public void SessionKey(BigInteger A) //*генерирует общий ключ по формуле:* $K = A^b \pmod{p}$

public BigInteger SessionKey() // *возвращает общий ключ*

В роли «человека по середине» выступает Меллори. Атака реализована в классе Mellory:

private final BigInteger **p**; //*сгенерировано на стороне Алисы и передано Бобу*
private final BigInteger **g**; //*сгенерировано на стороне Алисы и передано Бобу*
private BigInteger **m**; *// секретный ключ Меллори*
private BigInteger **M**; *// Публичный ключ Меллори*
private BigInteger **sessionKeyA**; *// сессионный ключ с Алисой*
private BigInteger **sessionKeyB**; *// сессионный ключ с Бобом*

public BigInteger PublicKey()//возвращает открытй ключ
public void generation_SessionKey_A(BigInteger A)//*генерирует общий ключ с Алисой*
public void generation_SessionKey_B(BigInteger B) //*генерирует общий ключ с Бобом*

В классе Main мы вызываем методы из предыдущих классов, генерируя сессионные ключи и реализуя атаку.

Пример работы

1. Генерация общего сессионного ключа

```
публичный ключ Алисы (A):
67460772448584551229629822943021274694332628343898608420286261747060694499l0
публичный ключ Боба (B):
819401186541841275911697309398230852769994518159219339222504679823513329693 75
общий сессионный ключ на стороне Алисы:
419326446933734276688098035142566930754506768146885551478419570903005227169 56
общий сессионный ключ на стороне Боба:
419326446933734276688098035142566930754506768146885551478419570903005227169 56
Ключи индентичны
```

2. Генерация общего сессионного ключа с вмешательством третьего лица
(атака «человек по середине»)

```
Открытый ключ A на стороне Алисы:
261022350020281327849637159101839844997385728211750514942920394139108688556 52
Открытый ключ A на стороне Боба:
129962875938622511964828464270736779818927178315250461797741659530739677404522
Открытый ключ M на стороне Меллори:
164546476111309520143627474951554640240818722336718642464536043638991491538703
общий сессионный ключ Алисы с 'Бобом' (Меллори):
136740747070126473129516833266245073027065790263821786196779022754271288827128
общий сессионный на стороне Меллори ключ с Алисой:
136740747070126473129516833266245073027065790263821786196779022754271288827128
общий сессионный ключ Боба с 'Алисой' (Меллори):
130722542033369027192358610032251119711766974680483593151104204448926135939334
общий сессионный на стороне Меллори ключ с Бобом:
130722542033369027192358610032251119711766974680483593151104204448926135939334
Ключи Алисы и Меллори индентичны
Ключи Боба и Меллори индентичны
```