Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

«Обзор защитных свойств криптографического протакола TLS 1.3»

Студент ИУ7-52Б (Группа) _____ (Подпись, дата) В. М. Короткая (И. О. Фамилия)

Руководитель _____ (Подпись, дата) К. А. Кивва (И. О. Фамилия)

2022 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Очередной этап технологической революции, происходящий в настоящее время в мире, влечет серьезные изменения в экономике, социальной структуре общества. Массовое применение новых технологических средств, на основе которых осуществляется информатизация, стирает геополитические границы, изменяет образ жизни миллионов людей. Вместе с тем, информационная сфера становиться не только одной из важнейших сфер международного сотрудничества, но и объектом соперничества.

Таким образом появляется возможность перехвата и подмены какой-либо информации в сети.

Для решения данной проблемы существует протокол TLS (Transport Layer Security) — криптографический протокол, обеспечивающий защищенную передачу данных в сети Интернет.

Целью данной работы является обзор методов обмена ключами TLS - рукопожатии.

Для достижения поставленной цели требуется решить следующие задачи:

- провести анализ предметной облости;
- определить основные термины, связанные с протоколом TLS;
- рассмотреть алгоритм Диффи — Хеллмана;
- рассмотреть алгоритм RSA;
- выделить типы возможных атак на данные алгоритмы.

1 Анализ предметной области

TLS даёт возможность клиент-серверным приложениям осуществлять связь в сети таким образом, что нельзя производить прослушивание пакетов и осуществить несанкционированный доступ.

1.1 История создания

Первые попытки создания сетевых сокетов принадлежат компании Netscape, они носили имя SSL (Secure Sockets Layer). TLS является приемником SSL (имя было сменено из-за юридических проблем с компанией Netscape).

Ниже представлена таблица с версиями протокола.

Таблица 1.1 – Протоколы TLS и SSL.

Протокол	**Дата публикации**	**Состояние**
SSL 1.0	—	—
SSL 2.0	**1995**	**Признан устаревшим в 2011 году**
SSL 3.0	**1996**	**Признан устаревшим в 2015 году**
TLS 1.0	**1999**	**Признан устаревшим в 2020 году**
TLS 1.1	**2006**	**Признан устаревшим в 2020 году**
TLS 1.2	**2008**	
TLS 1.3	**2018**	

1.2 Задачи, решающиеся в TLS

Протокол TLS предназначен для предоставления трёх услуг всем приложениям, работающим над ним, а именно:

- аутентификация – проверка авторства передаваемой информации;
- целостность – обнаружение подмены информации подделкой;
- конфиденциальность - сокрытие информации, передаваемой от одного компьютера к другому.

1.3 Описание процедуры

TLS предстовляет две фазы или два протокола.

Протокол рукопожатия (Handshaking Protocols), на этом шаге клиент и сервер будут:

- согласовать версию протокола,
- выбирать криптографический алгоритм или наборов шифров,
- аутентифицировать друг друга с помощью асимметричной криптографии,
- устанавливать общий секретный ключ, который будет использоваться для симметричного шифрования на следующей фазе.

Таким образом, основная цель рукопожатия — аутентификация и обмен ключами.

Протокол записи (Record Protocol), на этом шаге:

- все исходящие сообщения будут зашифрованы с помощью общего секретного ключа, установленного при рукопожатии,
- затем зашифрованные сообщения передаются другой стороне,
- их проверяют, чтобы увидеть, возникли ли какие-то изменения во время передачи или нет,
- если нет, то сообщения будут дешифрованы с использованием того же симметричного секретного ключа.

Таким образом, добивается как конфиденциальности, так и целостности в этом протоколе записи.

1.4 Обмен ключами

1.5 Вывод

На данный момент (01.01.2020) существуют две актуальные версии TLS: TLS 1.2 и TLS 1.3, остальные признаны устаревшими. Существование двух версий обосновывается тем, что старые машины не в силах поддерживать версию 1.3. Но далее в этой работе будет рассматриваться версия 1.3, так как является более актуальной.

2 Классификация существующих решений

2.1 Существующие решения

Симметричное шифрование производительнее, чем асимметричное, что делает его более подходящим для отправки данных по HTTPS-соединению. Точный метод генерации ключа зависит от выбранного шифронабора, два самых распространённых из них — RSA и Диффи — Хеллман.

2.1.1 Обмен ключами Diffie — Hellman, DH

Алгоритм Диффи — Хеллмана является одним из первых алгоритмов с открытым ключом, предложенным Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman) [1] Данный алгоритм позволил уменьшить требования к каналу связи для установления защищённого соединения без предварительного обмена ключами.

Алгоритм позволяет двум сторонам создать общий сеансовый ключ используя такой канал связи, который может прослушивать злоумышленник, но в предположении, что последний не может менять содержимое сообщений.

Ключ

Для того чтобы установить ключ, клиенту и серверу необходимо выполнить следующие действия.

1. Клиент генерирует число a, вычисляет число

$$A = g^a \bmod p \tag{2.1}$$

   и посылает его серверу.

2. Сервер генерирует число b, вычисляет число

$$B = g^b \bmod p \tag{2.2}$$

   и посылает его клиенту.

3. Клиент вычисляет значение

$$B^a \bmod p = g^{ab} \bmod p \tag{2.3}$$

.

4. Сервер вычисляет значение

$$A^b \bmod p = g^{ab} \bmod p \tag{2.4}$$

.

Заметим что обе стороны вычисляют одно и тоже значение

$$K = g^{ab} \bmod p. \tag{2.5}$$

Таким образом, числа p и q можно разослать всем участникам системы.

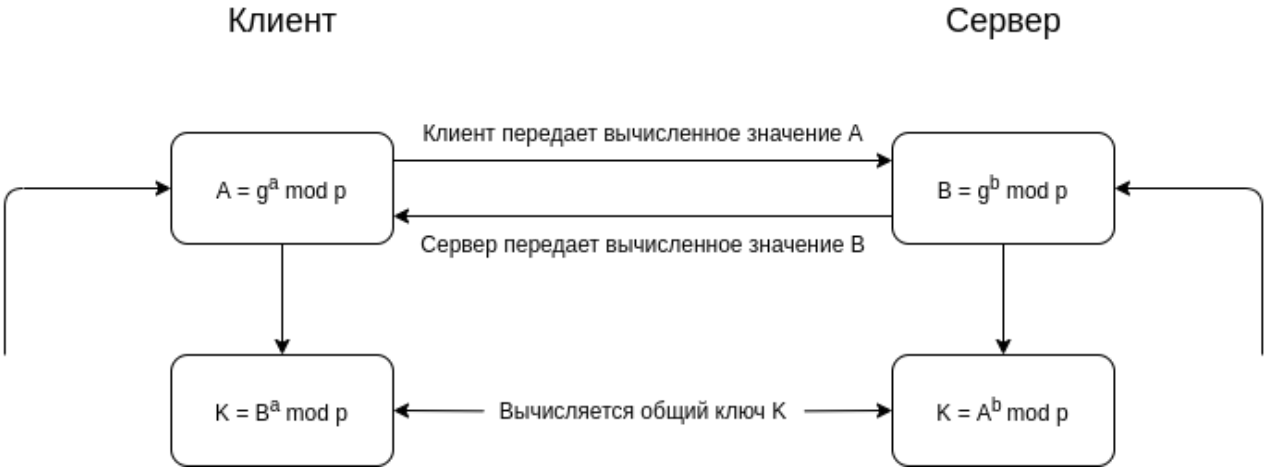

Рисунок 2.1 – Алгоритм шифрования сеансового ключа.

2.1.2 Обмен ключами RSA

Алгоритм RSA носит имя в честь своих создателей Рона Ривест (Ron Rivest), Ади Шамира (Adi Shamir) и Леонарда Адлемана (Leonard Adleman) из Массачусетского технологического института.

Называть это обменом ключами RSA на самом деле неправильно. На самом деле это RSA-шифрование. RSA использует асимметричное шифрование для создания ключа сеанса.

Ключ

Для того чтобы установить ключ, клиенту необходимо выполнить следующие действия.

1. Выбрать два различных случайных простых числа p и q, удовлетворяющих условию $\mid p \mid \approx \mid g \mid$.

2. Вычислить $N = pq$.

3. Вычислить
$$\phi(N) = (p-1)(q-1). \tag{2.6}$$

4. Выбрать случайное целое число $e < \phi(N)$ и найти целое число d такое что
$$de \equiv 1 \bmod \phi(N). \tag{2.7}$$

5. Использовать пару (N, e) в качестве параметров открытого ключа, тщательно уничтожить числа $p, g, \phi(N)$ и запомнить число d в качестве закрытого ключа.

Шифрование

Для того чтобы переслать клиенту секретное сообщение, имеющее длину $m < N$, сервер создает зашифрованный текст

$$c = m^e \pmod{N}$$

Расшифровка

Для того чтобы расшифровать зашифрованный текст c, клиент вычисляет формулу

$$m = c^d \pmod{N}$$

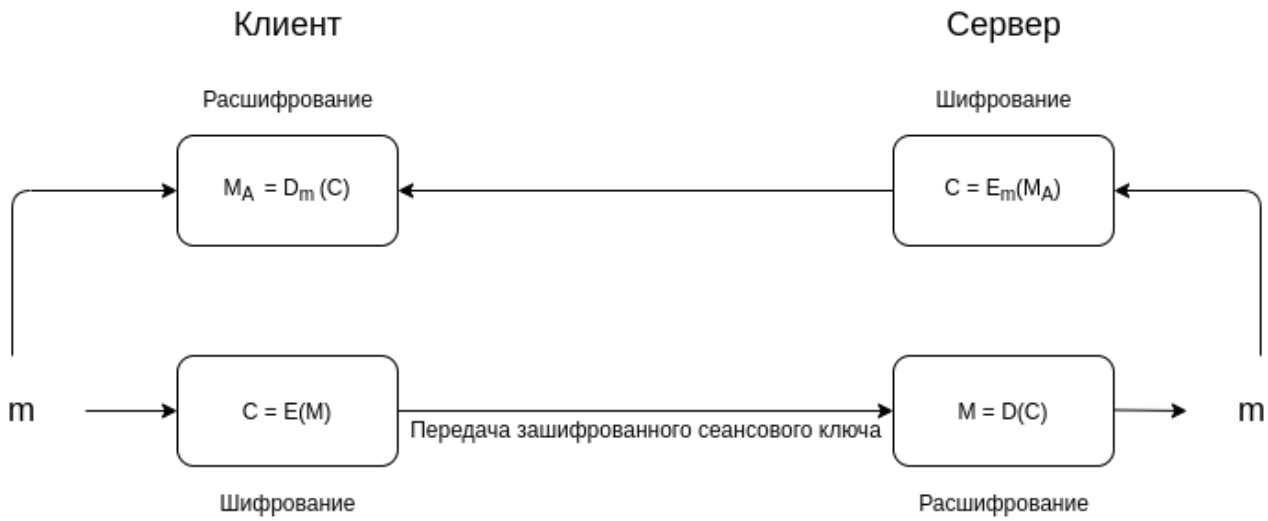

Рисунок 2.2 – Алгоритм шифрования сеансового ключа.

2.2 Критерий по колличеству ключей

Симметричное шифрование — это метод использования одних и тех же криптографических ключей как для шифрования открытого текста, так и для дешифрования зашифрованного текста.

Асимметричное шифрование — это метод использования пары ключей: открытого ключа, который широко распространен, и частного ключа, который известен только владельцу.

Рисунок 2.3 – Критерий по колличеству ключей

2.3 Активные атаки на криптосистемы

2.3.1 Атака "человек посередине"(man-in-the-middle attack — MITM)

Вид атаки, когда злоумышленник тайно ретранслирует и при необходимости изменяет связь между двумя сторонами, которые считают, что они непосредственно общаются друг с другом.

2.3.2 Атака на основе подобранного открытого текста (chosen-plaintext attack — CPA).

Атакующий выбирает исходные сообщения и передает их шифровальщику для получения зашифрованных текстов. Задача атакующего — взломать криптосистему, используя полученные пары открытых и зашифрованных текстов.

2.3.3 Атака на основе подобранного зашифрованного текста (chosen-ciphertext attack — CCA).

Атакующий выбирает зашифрованные сообщения и передает их на расшифровку для получения исходных сообщений. Цель атакующего — взломать криптосистему, используя полученные пары открытых и зашифрованных текстов. Атакующий достигает успеха, если он раскрывает ключ и способен в дальнейшем извлекать секретную информацию из зашифрованного текста, не прибегая к посторонней помощи.

2.3.4 Атака на основе адаптивно подобранного зашифрованного текста (adaptive chosen-ciphertext attack — CCA2).

Это — разновидностьатаки CCA, в которой услуги расшифровки доступны для всех зашифрованных текстов, за исключением заданного.

2.4 Вывод

RSA облегчает обмен ключами, позволяя клиенту шифровать общий секрет и отправлять его на сервер, где он используется для вычисления

Таблица 2.1 – Классификация по атакам на криптосистемы.

	Diffie — Hellman	**RSA**
MITM		
CPA		
CCA		
CCA2		

соответствующего сеансового ключа. Обмен ключами DH на самом деле вообще не требует обмена открытым ключом, скорее обе стороны создают ключ вместе.

ЗАКЛЮЧЕНИЕ

Так же были выполнены следующие задачи:

- провести анализ предметной облости;
- определить основные термины, связанные с протоколом TLS;
- рассмотреть алгоритм Диффи — Хеллмана;
- рассмотреть алгоритм RSA;
- выделить типы возможных атак на данные алгоритмы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Diffie-Hellman Key Agreement Method. — URL: https://datatracker.ietf.org/doc/html/rfc2631.
2. The Transport Layer Security (TLS) Protocol Version 1.3. — URL: https://datatracker.ietf.org/doc/html/rfc8446#section-7.4.1.
3. The Transport Layer Security (TLS) Protocol Version 1.2. — URL: https://datatracker.ietf.org/doc/html/rfc5246#section-4.7.
4. PKCS 1: RSA Cryptography Specifications Version 2.2. — URL: https://datatracker.ietf.org/doc/html/rfc8017.