

КРИПТОГРАФИЧЕСКИЙ ПРОТОКОЛ ОБМЕНА КЛЮЧАМИ «ДЕРЖАСЬ ЗА РУКИ»

Протокол – это порядок действий, предпринимаемых двумя или более сторонами, предназначенный для решения определенной задачи.

Криптографический протокол – это протокол, использующий криптографию.

Криптографические методы в протоколе применяются для предотвращения или обнаружении вредительства и мошенничества.

Условные обозначения участников протоколов:

- Алиса – Первый участник всех протоколов;
- Боб – Второй участник всех протоколов;
- Мэллори – Взломщик протоколов;
- Трент – Заслуживающий доверия посредник.

Протокол «держась за руки» позволяет предотвратить атаку «человек посередине».

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 [1].

Принцип его работы пошагово будет описан далее.

- 1) Алиса посылает Бобу свой открытый ключ;
- 2) Боб посылает Алисе свой открытый ключ;
- 3) Алиса зашифровывает своё сообщение открытым ключом Боба и отправляет Бобу половину зашифрованного сообщения;
- 4) Боб зашифровывает своё сообщение открытым ключом Алисы и отправляет ей половину зашифрованного сообщения;
- 5) Алиса отправляет Бобу вторую половину зашифрованного сообщения;

6) Боб складывает две части сообщения Алисы и расшифровывает его с помощью своего закрытого ключа, а затем отправляет Алисе вторую половину своего зашифрованного сообщения;

7) Алиса складывает две части сообщения Боба и расшифровывает его с помощью своего закрытого ключа.

Суть метода заключается в том, что половина зашифрованного сообщения не может быть дешифрована без второй половины. Боб не сможет прочитать ни одной части сообщения Алисы до этапа (6), а Алиса до этапа (7).

Но и Мелори, перехватив половину сообщения Алисы на этапе (3), не сможет расшифровать ее своим закрытым ключом и снова зашифровать открытым ключом Боба [2].

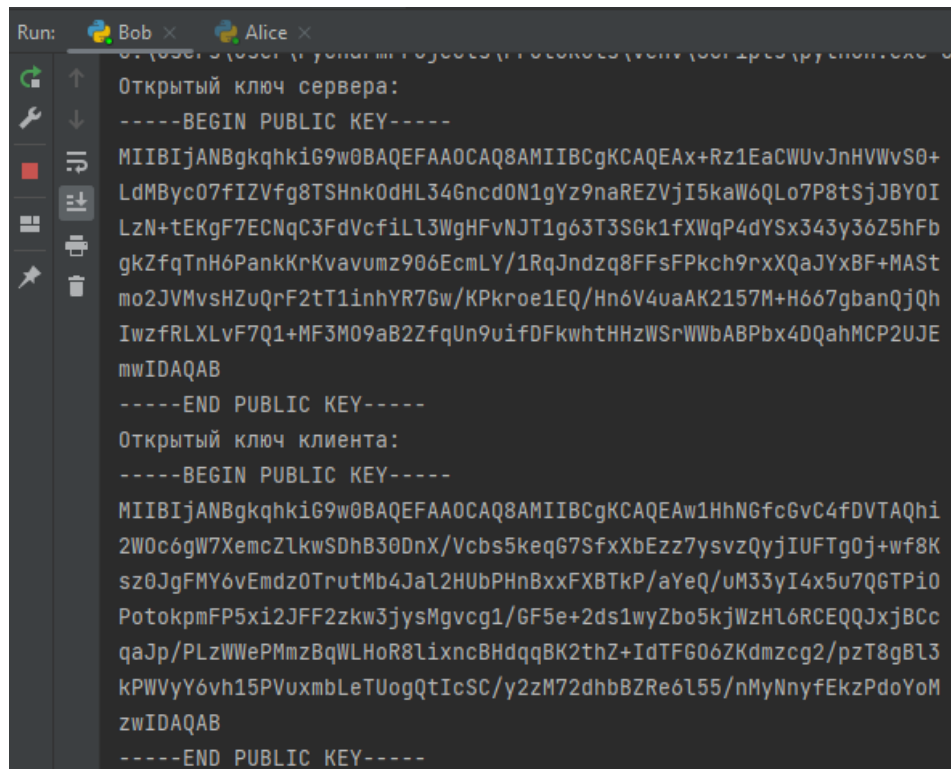
В качестве Алисы и Боба далее будем понимать взаимодействие клиента и сервера, где в качестве клиента выступает Алиса, а в качестве сервера соответственно – Боб.

Реализация протокола на Python

Взаимодействие между Алисой (клиентом) и Бобом (сервером) осуществлялось посредством сокетов. Сокет – это программный интерфейс для обеспечения информационного обмена между процессами.

На стороне Боба стоит понимать серверный сокет, который прослушивает определенный порт, а на стороне Алисы клиентский сокет, который подключается к серверу. После того как соединение было установлено происходил обмен данными.

В первую очередь программа сгенерировала и вывела открытый ключ на сервере, а после установки соединения с клиентом получила сгенерированный открытый ключ клиента и также вывела его (рисунок 2.1).



```
Run: Bob x Alice x
Открытый ключ сервера:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIBCgKCAQEAx+Rz1EaCWUvJnHVWvS0+
LdMByc07fIZVfg8TSHnk0dHL34Gncd0N1gYz9naREZVjI5kaW6QLo7P8tSjJBYOI
LzN+tEKgF7ECNqC3FdVcfiLL3WgHFvNJT1g63T3SGk1fXWqP4dYSx343y36Z5hFb
gkZfqTnH6PankKrKvavumz906EcmLY/1RqJndzq8FFsFPkch9rxXQaJYxBF+MASt
mo2JVMvsHZuQrF2tT1inhYR7Gw/KPkr0e1EQ/Hn6V4uaAK2157M+H667gbanQjQh
IwzfRLXLvF7Q1+MF3M09aB2ZfqUn9uifDFkwhthHzWSrWWbABPbx4DQahMCP2UJE
mwIDAQAB
-----END PUBLIC KEY-----
Открытый ключ клиента:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIBCgKCAQEAw1HhNGfc6vC4fDVTAQh1
2W0c6gW7XemcZLkwSDhB30DnX/Vcbs5keqG7SfxXbEzz7ysvzQyjiUFTg0j+wf8K
sz0JgFMY6vEmdz0TrutMb4JaL2HUbPHnBxxFXBTkP/aYeQ/uM33yI4x5u7Q6TPi0
PotokpmFP5xi2JFF2zkw3jysMgvcg1/GF5e+2ds1wyZbo5kjWzHL6RCEQQJxjBCc
qaJp/PLzWWePMmzBqWLHoR8lixncBHdqgBK2thZ+IdTFG06ZKdmzcg2/pzT8gB13
kPWVYy6vh15PVuxmbLeTUogQtIcSC/y2zM72dhbBZRe6L55/nMyNnyfEkzPdoYoM
zwIDAQAB
-----END PUBLIC KEY-----
```

Рисунок 2.1 – Обмен ключами сервера с клиентом

Клиент в свою очередь подключается к серверу через указанный порт, также генерирует свой открытый ключ и получает открытый ключ сервера (рисунок 2.2).

Для шифрования ключей использовался алгоритм RSA. RSA является наиболее распространенным и используемым алгоритмом с открытым

ключом. Его безопасность основана на сложности факторизации больших целых чисел. Алгоритм выдерживал атаки более 30 лет и поэтому считается достаточно безопасным для новых разработок. Криптографическая стойкость в первую очередь связана с длиной модуля RSA n . В 2017 году достаточной длиной считается 2048 бит.

```

Run: Bob x Alice x
Открытый ключ клиента:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAw1HhNGfcGvC4fDVTAQh1
2W0c6gW7XemcZlkwSDhB30DnX/Vcbs5keq67SfxXbEzz7ysvzQyjiUFTg0j+wf8K
sz0JgFMY6vEmdz0TrutMb4Ja12HUbPHnBxxFXBTKP/aYeQ/uM33yI4x5u7Q6TPi0
PotokpmFP5xi2JFF2zkW3jysMgvcg1/6F5e+2ds1wyZbo5kjWzHl6RCEQJxjBCc
qaJp/PLzWWePMmzBqWLHoR8LixncBHdqgBK2thZ+IdTFG06ZKdmzcg2/pzT8gB13
kPWVYy6vh15PVuxmbLeTUogQtIcSC/y2zm72dhhBZRe6l55/nMyNnyfEkzPdoYoM
zwIDAQAB
-----END PUBLIC KEY-----
Открытый ключ сервера:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAx+Rz1EaCWUvJnHVWvS0+
LdMByc07fIZVfg8TSHnk0dHL346ncd0N1gYz9naREZVjI5kaW6QLo7P8tSjJBYOI
LzN+tEKgF7ECNqC3FdVcfiL3WgHFvNJT1g63T3SGk1fXWqP4dYSx343y36Z5hFb
gkZfqTnH6PankKrKvavumz906EcmLY/1RqJndzq8FFsFPkch9rxXQaJYxBF+MAst
mo2JVMvsHZuQrF2tT1inhYR76w/KPkroe1EQ/Hn6V4uaAK2157M+H667gbanQjQh
IwzfRLXLvF7Q1+MF3M09aB2ZfqUn9uifDFkwhthHhWSrWWbABPbx4DQahMCP2UJE
mwIDAQAB
-----END PUBLIC KEY-----

```

Рисунок 2.2 – Обмена ключами клиента с сервером

Посимвольно сравнивая значения открытых ключей, можно заметить, что ключи сервера и клиента совпадают на рисунках 2.1 – 2.2, что говорит о том, что обмен ключами был произведен успешно.

Далее происходит обмен данными (сообщениями) между клиентом и сервером. Сообщение, как и было сказано ранее, на клиенте шифруется полученным ключом с сервера, а затем первая его часть отправляется в зашифрованном виде на сервер (рисунок 2.3).

```

Сообщение, отправляемое серверу:
Привет, Боб! Мне нужно сказать тебе что-то важное.
Первая часть сообщения клиента:
0JQVU,0,nd ~`10 0#0EK0?4G>n0:0-`jCVo0I00=w`c'n#)bpd?000f0! 1H

```

Рисунок 2.3 – Отправка части сообщения клиента на сервер

Далее на сервере также пользователь вводит сообщение, и оно зашифровывается. После этого сервер получает первую часть зашифрованного сообщения клиента. Далее сервер отправляет на клиент часть своего зашифрованного сообщения (рисунок 2.4).

```
Сообщение,отправляемое клиенту:
Алиса, ты самая лучшая!!!
Первая часть сообщения клиента:
0JQVU,0,nd ~`10 0#0EКС0?ц6>n0:0-`jCVo0I00=w`<n#)bpd?000f0! IH
Первая часть сообщения сервера:
Ер,w%Kxg_W>01
```

Рисунок 2.4 – Получение зашифрованной части сообщения с клиента

После этого клиент (Алиса) получает первую часть сообщения с сервера в зашифрованном виде и отправляет вторую часть своего сообщения в зашифрованном виде на сервер (рисунок 2.5).

```
Первая часть сообщения сервера:
Ер,w%Kxg_W>01
Вторая часть сообщения клиента:
A~kwv/$%00+0{><93г08]
```

Рисунок 2.5 – Получение зашифрованной части сообщения сервера

Далее сервер получает вторую часть сообщения клиента. Теперь он может расшифровать полностью сообщение Алисы и отправить ей вторую часть своего сообщения (рисунок 2.6).

```
Вторая часть сообщения клиента:
A~kwv/$%00+0{><93г08]
0V7Sb*+0f|onS0AË~0Qαт000AiYz81,~>00]
Вторая часть сообщения сервера:
BTњ0m0[)bT)2i0g_#00Hо1}0aNQ0HS8iũ2{h0o00Sw0)`k00!YP60*NEa_]qs6
Ответ от клиента:
Привет,Боб! Мне нужно сказать тебе что-то важное.
```

Рисунок 2.6 – Расшифровка сообщения от клиента на сервере

Далее клиент получает вторую часть зашифрованного сообщения от сервера и теперь он может также расшифровать все сообщение (рисунок 2.7).

```
Вторая часть сообщения сервера:  
BT50m[]bT)2ig_#00Hof}0aNQ0HS8i2{h0o00Sw0)`k00!YP60*NEa_]qs6  
Ответ от сервера:  
Алиса, ты самая лучшая!!!
```

Рисунок 2.7 – Расшифровка сообщения от сервера на клиенте

При расшифровке сообщений наблюдаются одинаковые, верные результаты, а это значит, что программа работает корректно.

Список использованных источников

- 1) Атака посредника [Электронный ресурс]: сайт ru.wikipedia.org. URL: https://ru.wikipedia.org/wiki/%D0%90%D1%82%D0%B0%D0%BA%D0%B0_%D0%BF%D0%BE%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D0%B8%D0%BA%D0%B0 (дата обращения: 13.09.2022);
- 2) Методы и задачи криптографической защиты информации: Учебное пособие [Электронный ресурс]: сайт window.edu.ru. URL: http://window.edu.ru/catalog/pdf2txt/904/58904/28771?p_page=8 (дата обращения: 15.09.2022).