

КРИПТОГРАФИЧЕСКИЙ ПРОТОКОЛ УДОСТОВЕРЕНИЯ ПОДЛИННОСТИ И ОБМЕНА КЛЮЧАМИ «YAHALOM»

Yahalom – это симметричный протокол аутентификации и безопасного обмена ключами, разработанный для использования в незащищенных сетях, таких как Интернет. Yahalom использует доверенного арбитра для распределения общего ключа между двумя людьми. Этот протокол можно рассматривать как улучшенную версию протокола Wide Mouth Frog (с дополнительной защитой от атаки типа «человек посередине», но менее безопасный, чем Needham –Schroeder protocol [1].

Условные обозначения участников протоколов:

- Алиса (А) – Первый участник всех протоколов;
- Боб (Б) – Второй участник всех протоколов;
- Трент (Т) – Заслуживающий доверия посредник.

Данный протокол «перекладывает» генерацию нового сессионного ключа на сторону доверенного центра, а также использует случайные числа для защиты от атак повтором. При симметричном шифровании, предполагается, что секретный ключ, который принадлежит клиенту, известен только ему и некоторой третьей доверенной стороне –серверу аутентификации (Тренту). В процессе сеанса протокола клиенты Алиса и Боб получают от сервера аутентификации Трента новый секретный сессионный ключ для шифрования взаимных сообщений в текущем сеансе связи. Реализация протокола [2].

Принцип его работы пошагово будет описан далее.

1) Первым сообщением Алиса инициирует сеанс, пересылая Бобу свой идентификатор А и некоторое случайное число R_A :

$$A \rightarrow B: A, R_A ;$$

2) Получив сообщение от Алисы, Боб объединяет идентификатор Алисы А, случайное число R_A Алисы и своё случайное число R_B и шифрует созданное сообщение общим с Трентом ключом. После добавления к этому

сообщению своего идентификатора В Боб отправляет полученное сообщение Тренту:

$$B \rightarrow T: B, E_B(A, R_A, R_B);$$

3) Трент расшифровывает сообщение Боба и создаёт два сообщения. Первое сообщение включает в себя идентификатор Боба В, сгенерированный Трентом сессионный ключ К, случайное число Алисы R_A и случайное число Боба R_B . Данное сообщение шифруется общим с Алисой ключом. Первое сообщение имеет вид:

$$\{E_A(B, K, R_A, R_B)\}.$$

Второе сообщение шифруется общим ключом для Трента и Боба и включает в себя идентификатор Алисы А и сгенерированный Трентом сессионный ключ К. Второе сообщение имеет вид:

$$\{E_B(A, K)\}.$$

Трент пересылает Алисе оба созданные сообщения:

$$T \rightarrow A: \{E_A(B, K, R_A, R_B), E_B(A, K)\}.$$

Алиса получает два сообщения от Трента и расшифровывает первое из них. Расшифровав сообщение, Алиса извлекает сессионный ключ К и убеждается, что случайное число переданное Трентом R_A совпадает со случайным числом, переданным Бобу на первом этапе. После этого Алиса отправляет два сообщения Бобу. Первое сообщение является полученное от Трента сообщением, зашифрованным общим ключом для Трента и Боба. Данное сообщение состоит из идентификатора Алисы А и сессионного ключа К.

Первое сообщение:

$$\{E_B(A, K)\}.$$

Второе сообщение является зашифрованным с помощью сгенерированного Трентом сессионного ключа случайного числа Боба:

$$\{E_K(R_B)\}$$

4) Оба сообщения Алиса отправляет Бобу:

$$A \rightarrow B: \{E_B(A, K), E_K(R_B)\}$$

Боб расшифровывает первое сообщение и извлекает сессионный ключ К. С помощью извлечённого сессионного ключа Боб расшифровывает второе сообщение и получает случайное число R_B . Боб сверяет полученное от Алисы число со случайным числом, отправленным на втором этапе. После описанных действий стороны могут использовать новый сессионный ключ К. Протокол Yahalom помимо генерации сессионного ключа обеспечивает аутентификацию сторон:

- Аутентификация Алисы перед Бобом происходит на четвёртом этапе (4), когда Боб может проверить возможность Алисы зашифровать известные только ей и Тренту случайное число R_A на ключе К.

- Аутентификация Боба перед Алисой происходит на третьем этапе (3), когда Трент показывает Алисе, что он получил случайное число R_A именно от Боба (поскольку в сообщении присутствует идентификатор Боба В).

Описанные шаги и действия сторон можно представить в виде рисунка для наглядности представления работы протокола (рисунок 1).

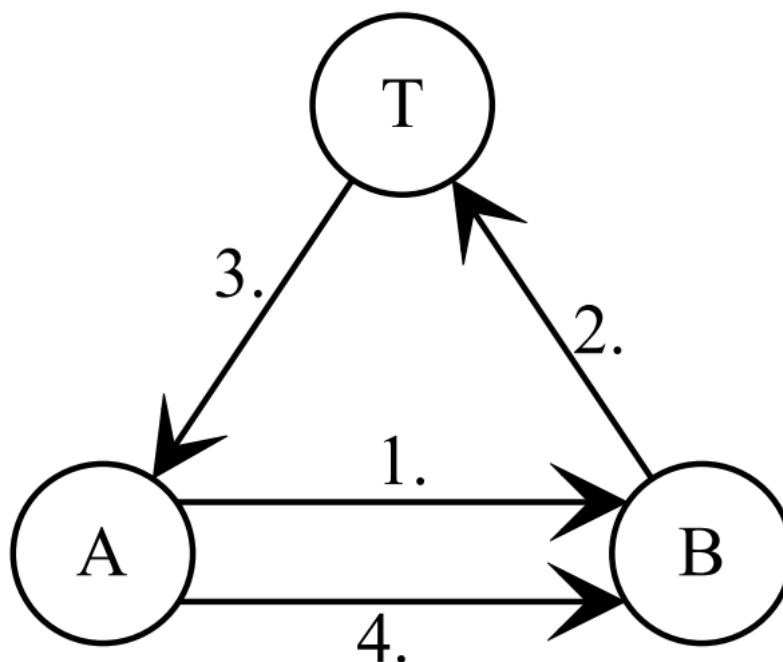
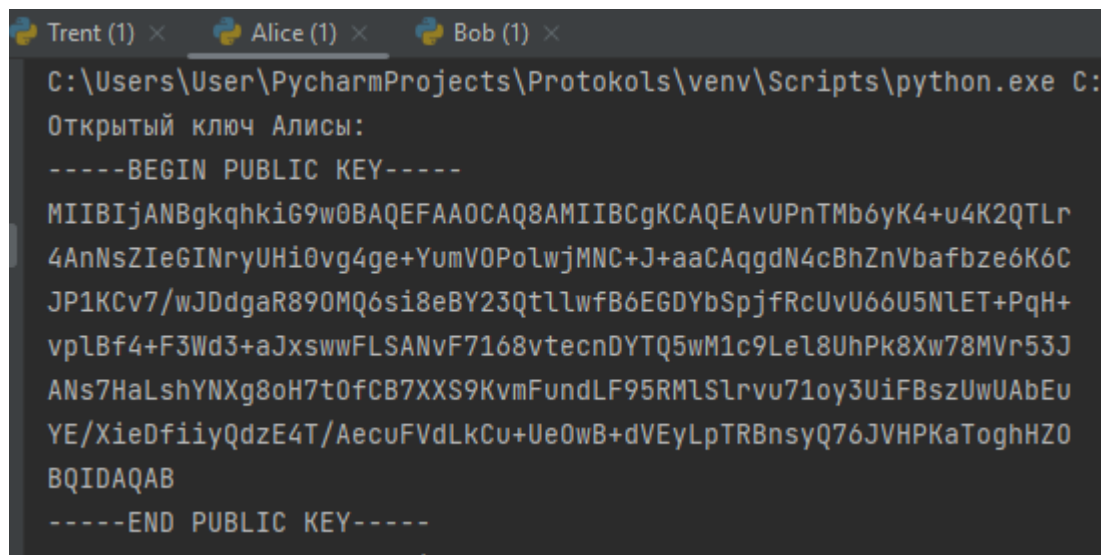


Рисунок 1 – Взаимодействие участников протокола Yahalom

Реализация протокола на Python

Во-первых, для того, чтобы организовать взаимодействие между Алисой, Бобом и Трентом были подключены сокеты. На стороне каждого участника происходило создание сокета TCP, связь сокета с хостом и портом. На стороне Трента произведено ожидание подключения клиентов (Алисы и Боба), в то время как Алиса подключается к Тренту и принимает запрос на подключение от Боба, Боб подключается к Тренту и к Алисе.

После организации подключения, начинается обмен открытыми ключами. На рисунке 1 представлен сгенерированный открытый ключ Алисы.



```
Trent (1) x Alice (1) x Bob (1) x
C:\Users\User\PycharmProjects\Protokols\venv\Scripts\python.exe C:
Открытый ключ Алисы:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuPnTMb6yK4+u4K2QTLr
4AnNsZIEGINryUHi0vg4ge+YumVOPoLwjMNC+J+aaCAqgdN4cBhZnVbafbze6K6C
JP1KCv7/wJDdgaR890MQ6si8eBY23QtllwFB6EGDYbSpjfRcUvU66U5NLET+PqH+
vp1Bf4+F3Wd3+aJxswWFLSANvF7168vtecdYDTQ5wM1c9Le18UhPk8Xw78MVR53J
ANs7HaLshYNXg8oH7t0fCB7XXS9KvmFundLF95RMLslrvu71oy3UiFBszUwUAbEu
YE/XieDfiiyQdzE4T/AecuFVdLkCu+Ue0wB+dVEyLpTRBnsyQ76JVHPKaToghHZ0
BQIDAQAB
-----END PUBLIC KEY-----
```

Рисунок 1 – Открытый ключ Алисы

Для шифрования ключей использовался асимметричный 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.

На рисунке 2 представлено получение открытых ключей на стороне Боба.

```
Открытый ключ Боба:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEA5NZcaC7x58Hl8JS0ckVI
gDocUaAg5pdbJIHbiNRYxEeCW5Hdxftc0odnLPZnNRZkzn8sq7a8nnblzWw3Jan
R4WxSCyJI6wAQw3CPNLiRy1pL5hSHgEJ1spuazYjcDaA0v2aLtayAwuQWgPDLLW
N0gRUyp7V9ypqZoNfnpFGpAVW0cM84LS1uA7uDmqnE4zmSRZ1cm43XyyI4BUaiAV
1Lku2DjiMFwQoiQCp+V9sYsjJBNNqZUh4SrMdsQhs0IPynHLWfcSbQEBuk3HTqG2
nTAWyqLXpsToqR09SXDvvIa8yTH8VD1n44Ns6AGECb4qqg1Gci3FQkehNPbUw1Qg
4wIDAQAB
-----END PUBLIC KEY-----
Открытый ключ Трента:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEA2y/ePfnMgpaHsc30edWG
v9NsgfxhK2jAgP4s3Smg4V8KKEto47tbawxzo4Z5uyYwYHXp3JMPj8WbUxBpwJTQ
yWp72quHpmqz0FpfZ7ZjpXadm6JHDTTniHbuUnFUWVKBySzMiSMXgf0/Uc0mcE6Q
roxXMwMFU+r09t7zZ80UvdaLkq9f8uvJrc8o1VEhgZyywU1/7+WfDddwRM8v5tFy
Jy5+wod4Iq/Xl2lyzNr2AleAIt/oRppvt0/UuYDhhsapzXXhLPjpf6e3mY1QUya0
y3d7TX6BT9Vxm2vS2W5fFCNhpyznhxMT3Icm3gcXlqTzZx3JFSH65eovh6/xiWx0
SwIDAQAB
-----END PUBLIC KEY-----
```

Рисунок 2 – Открытый ключ Боба

Так как для реализации протокола Тренту понадобятся оба открытых ключа, то он соответственно получает их от Алисы и Боба (рисунок 3).

```
Trent (1) x Alice (1) x Bob (1) x
C:\Users\User\PycharmProjects\ProtokoIs\venv\Scripts\python.exe C
Открытый ключ Трента:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2y/ePfnMgpaHsc30edWG
v9NsgfxhK2jAgP4s3Smg4V8KKEto47tbawxzo4Z5uyYwYHXp3JMPj8WbUxBpwJTQ
yWp72quHpmqz0FpfZ7ZjpXadm6JH0TTniHbuUnFUWVKBySzMiSMXgf0/Uc0mcE6Q
roxXMwMFU+r09t7zZ80UvdaLkq9f8uvJrc8o1VEhgZyywU1/7+WfddwRM8v5tFy
Jy5+wod4Iq/XL2lyzNr2AlEAt/oRppvt0/UuYDhhsapzXXhLPjpf6e3mY1QUya0
y3d7TX6BT9Vxm2vS2W5fFCNhpynzhxMT3Icm3gcXlqTzZx3JFSH65eovh6/xiWx0
SwIDAQAB
-----END PUBLIC KEY-----
Открытый ключ Алисы:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAUPnTmb6yK4+u4K2QTLr
4AnNsZIEGINryUH10vg4ge+YumVOPolwjMNC+J+aaCAqgdN4cBhZnVbafbz6K6C
JP1Kcv7/wJDdgaR890MQ6si8eBY23QtllwfB6EGDYbSpjRcUvU66U5NLET+PqH+
vplBf4+F3Wd3+aJxswWFLSANvF7168vtecdYTDQ5wM1c9Le18UhPk8Xw78MvR53J
ANs7HaLshYNXg8oH7t0fCB7XXS9KvmFundLF95RMLsLrvu71oy3U1FBszUwUABEu
YE/XieDfiyQdzE4T/AecuFVdLkCu+Ue0wB+dVEyLpTRBnsyQ76JVHPKaToghHZ0
BQIDAQAB
-----END PUBLIC KEY-----
Открытый ключ Боба:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5NZcaC7x58Hl8JS0ckVI
gDocUaAg5pdbJIHbiNRYxEeCWh5Hdxftc0odnLPZnNRZkzn8sq7a8nnblzWw3Jan
R4WxSCyJI6wAQw3CPNLiRy1pL5hSHgEJ1spuazYjcDaA0v2aLtaYAwuQqwgPDLLW
N0gRUyp7V9ypqZoNfnpFGpAVW0cM84LS1uA7u0mqnE4zmSRZ1cm43XyyI4BUaiAV
1Lku2DjiMFwQoiQCp+V9sYsjJBNNqZUh4SrMdsQhs0IPynHLWfcSbQEBuk3HTqG2
nTAWyqLXpsToqR09SX0vIa8yTH8VD1n44NsGAGECb4qqg16Ci3FQkehNPbUw1Qg
4wIDAQAB
-----END PUBLIC KEY-----
```

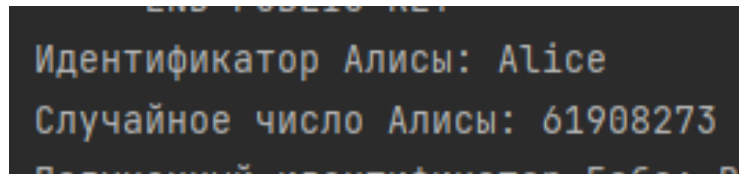
Рисунок 3 – Получение открытых ключей на стороне Трента

Можно заметить, что ключи от Алисы и Боба на рисунке 3 совпадают с теми, что представлены на рисунках 1 и 2, значит обмен ключами был произведен успешно.

Также стоит отметить, что Боб получает публичный ключ Трента только в рамках данной реализации. В описании сказано, что Боб и Трент должны обладать общим секретным ключом, им должны шифроваться данные от Боба. Но передача секретного ключа по каналу является по сути его компрометацией. Поэтому было принято решение шифровать данные от Боба публичным ключом Трента, его получение показано ниже.

Следующим шагом, после обмена ключами начинается работа самого протокола «Yahalom». Во-первых, программа на стороне Алисы создает

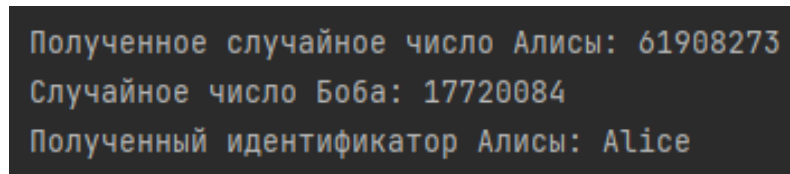
идентификатор Алисы и генерирует случайное число с помощью функции $\text{random}(\text{randint}(0, 72817281))$. После этого эти данные отправляются Бобу. На рисунке 4 представлено сгенерированное случайное число Алисы.



```
Идентификатор Алисы: Alice
Случайное число Алисы: 61908273
```

Рисунок 4 – Идентификатор и случайное число Алисы

Во-вторых, программа на стороне Боба принимает данные от Алисы и шифрует их вместе с сгенерированным числом Боба. Затем Боб посылает Тренту свой идентификатор и зашифрованное сообщение. На рисунке 5 представлены полученные данные от Алисы и случайное число Боба.

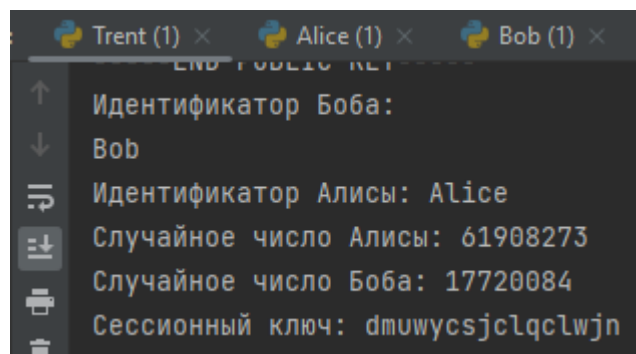


```
Полученное случайное число Алисы: 61908273
Случайное число Боба: 17720084
Полученный идентификатор Алисы: Alice
```

Рисунок 5 – Полученные данные от Алисы на стороне Боба

Если сравнить рисунки 4 и 5, то можно увидеть, что идентификатор и случайное число Алисы одинаковые, а это значит, что обмен данными между Алисой и Бобом произошел корректно.

В-третьих, Трент расшифровывает сообщение, вследствие чего получает идентификатор Боба, идентификатор Алисы, случайное число Боба и случайное число Алисы, а затем генерирует сессионный ключ (рисунок 6).

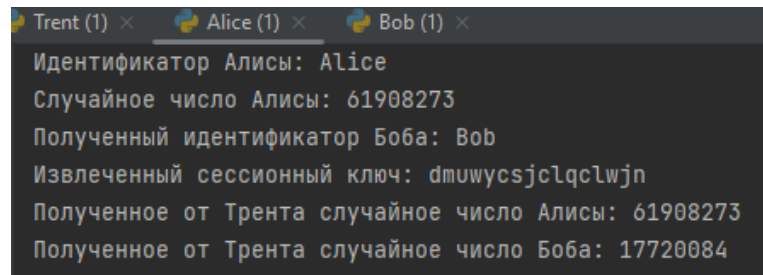


```
Trent (1) x Alice (1) x Bob (1) x
↑
Идентификатор Боба:
↓
Bob
⇅
Идентификатор Алисы: Alice
⇅
Случайное число Алисы: 61908273
⇅
Случайное число Боба: 17720084
⇅
Сессионный ключ: dmuwycsjclqclwjn
```

Рисунок 6 – Полученные данные от Боба на стороне Трента

Далее Трент создает два сообщения и оба пересылает Алисе: $T \rightarrow A: \{E_A(B, K, R_A, R_B), E_B(A, K)\}$.

Алиса в свою очередь расшифровывает первое сообщение, которое ей передал Трент. Следовательно, она получает идентификатор Боба, сессионный ключ, случайное число Боба и своё случайное число. Теперь можно убедиться, что случайное число переданное Трентом совпадает с тем числом, которое было передано от Алисы Бобу на первом этапе (рисунок 7).

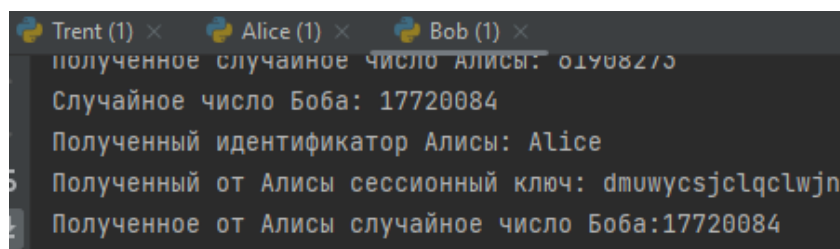


```
Trent (1) x Alice (1) x Bob (1) x
Идентификатор Алисы: Alice
Случайное число Алисы: 61908273
Полученный идентификатор Боба: Bob
Извлеченный сессионный ключ: dmuwycsjclqclwjn
Полученное от Трента случайное число Алисы: 61908273
Полученное от Трента случайное число Боба: 17720084
```

Рисунок 7 – Полученные данные от Трента на стороне Алисы

После этого программа Алисы шифрует еще одно сообщение на сессионном ключе для отправки Бобу, которое включает случайное число Боба. Далее Алиса отправляет оба сообщения.

И последним этапом Боб расшифровывает первое сообщение и получает идентификатор Алисы и сессионный ключ. С помощью сессионного ключа он расшифровывает второе сообщение и извлекает своё случайное число. На этом моменте он может сверить случайное число, которое было отправлено на втором этапе и число, которое пришло от Алисы, т.о. он аутентифицирует Алису (рисунок 8).



```
Trent (1) x Alice (1) x Bob (1) x
полученное случайное число Алисы: 61908273
Случайное число Боба: 17720084
Полученный идентификатор Алисы: Alice
Полученный от Алисы сессионный ключ: dmuwycsjclqclwjn
Полученное от Алисы случайное число Боба: 17720084
```

Рисунок 8 – Полученные данные от Алисы на стороне Боба

Можно заметить, что случайное число Боба, полученное от Алисы, совпадает с тем числом, которое было отправлено Тренту. Также на трех сторонах один и тот же сессионный ключ, что говорит о правильной, корректной работе протокола.

Список использованных источников

1) Яхалом (протокол) - Yahalom (protocol) [Электронный ресурс]: сайт wiki5.ru. URL: [https://wiki5.ru/wiki/Yahalom_\(protocol\)](https://wiki5.ru/wiki/Yahalom_(protocol)) (дата обращения: 27.09.2022);

2) Yahalom [Электронный ресурс]: сайт window.edu.ru. URL: <https://wiki4.ru/wiki/Yahalom> (дата обращения: 29.09.2022).