МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ компьютерной безопасности и криптографии

**Протокол Ньюмана-Стабблбайна**

ОТЧЁТ

ПО ДИСЦИПЛИНЕ

«КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ»

студента 5 курса 531 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Норикова Павла Сергеевича

|  |  |  |
| --- | --- | --- |
| Преподаватель  аспирант | \_\_\_\_\_\_\_\_\_\_ | Р. А. Фарахутдинов |
|  | подпись, дата |  |

Саратов 2023

**ВВЕДЕНИЕ**

Цель работы – изучение и реализация протокола обмена ключами Ньюмана-Стабблбайна.

**1 Теория**

Криптографический протокол Ньюмана-Стабблбайна для удостоверения подписи и обмена ключами был впервые опубликован в 1993 году. Протокол является модификацией протокола Yahalom и разработан в Массачусетском технологическом институте (MIT) Клифордом Ньюманом и Стюартом Стабблбайном.

Данный протокол является усовершенствованной версией протокола Yahalom. Особенностью протокола является отсутствие необходимости синхронизации часов у сторон, а также возможность повторной аутентификации без использования промежуточной стороны.

* 1. **Описание алгоритма**

1. . Алиса отправляет Бобу сообщение, содержащее идентификатор Алисы и некоторое случайное число Алисы.
2. . Боб объединяет идентификатор Алисы, ее случайное число и метку времени, шифрует сообщение общим с Трентом ключом и посылает его Тренту, добавив свой идентификатор и случайное число Боба.
3. . Трент генерирует сеансовый ключ K , а затем создает два сообщения. Первое включает идентификатор Боба, случайное число Алисы, случайный сеансовый ключ, метку времени и шифруется общим для Трента и Алисы ключом. Второе состоит из идентификатора Алисы, сеансового ключа, метки времени и шифруется общим для Трента и Боба ключом. Трент добавляет к ним случайное число Боба и отправляет Алисе.
4. . Алиса извлекает K и убеждается, что R A совпадает с тем, что было послано на этапе 1. Алиса отправляет Бобу два сообщения. Первое - это второе сообщение от Трента, зашифрованное ключом Боба. Второе - это случайное число Боба, зашифрованное сеансовым ключом.

Боб расшифровывает сообщение своим ключом и убеждается, что значения T B и R B не изменились.

Если оба случайных числа и метка времени совпадают, то Алиса и Боб убеждаются в подлинности друг друга и получают секретный ключ. Нет необходимости синхронизировать часы, так как метка времени определяется только по часам Боба и только Боб проверяет созданную им метку времени.

* 1. **Проверка подлинности**

1. . Алиса отправляет Бобу сообщение, присланное Трентом на этапе 3 и новое случайное число.
2. . Боб отправляет Алисе свое новое случайное число и случайное число Алисы, шифруя их сеансовым ключом.
3. . Алиса отправляет Бобу его новое случайное число, зашифрованное сеансовым ключом.

**2 Практическая реализация**

**2.1 Описание программы**

Функции *encr* и *decr* осуществляют шифрование и расшифрование соответственно.

**2.2 Тестирование программы**

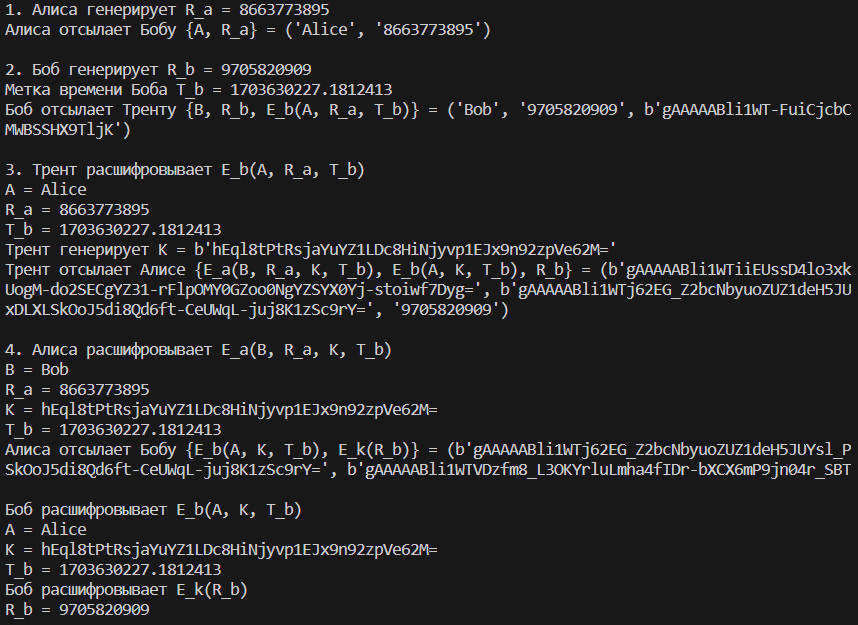


Рисунок 1 - Работа протокола

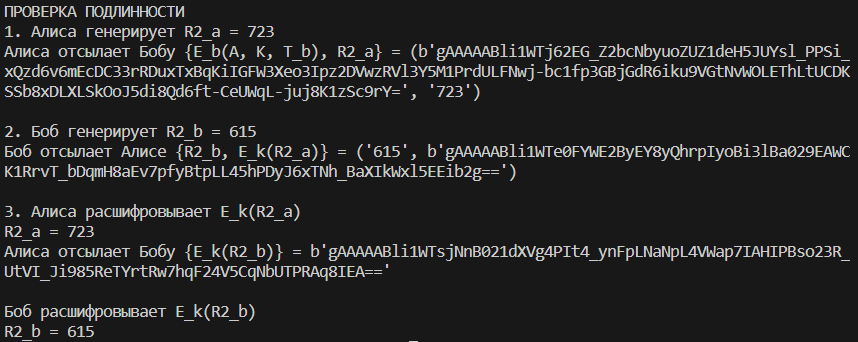


Рисунок 2 – Проверка подлинности

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

from cryptography.fernet import Fernet

import time

import random

def encr(key, x):

data = ''

for i in x:

if type(i) == bytes:

data += str(i, 'utf-8')

else:

data += str(i)

data += '\n'

data = data[:-1]

cipher\_suite = Fernet(key)

data = bytes(data, 'utf-8')

enc = cipher\_suite.encrypt(data)

return enc

def decr(key, enc):

cipher\_suite = Fernet(key)

dec = cipher\_suite.decrypt(enc)

dec = str(dec, 'utf-8')

res = dec.split('\n')

return res

########1########

A\_rand = random.randint(1, 10000000000)

A\_key = Fernet.generate\_key()

print('1. Алиса генерирует R\_a =', A\_rand)

s1 = ("Alice", str(A\_rand))

print("Алиса отсылает Бобу {A, R\_a} =", s1, '\n')

########2########

B\_rand = random.randint(1, 10000000000)

B\_key = Fernet.generate\_key()

B\_time = time.time()

print('2. Боб генерирует R\_b =', B\_rand)

print('Метка времени Боба T\_b =', B\_time)

s2 = ("Bob", str(B\_rand), encr(B\_key, (s1[0], s1[1], B\_time)))

print("Боб отсылает Тренту {B, R\_b, E\_b(A, R\_a, T\_b)} =", s2, '\n')

########3########

K = Fernet.generate\_key()

dec\_s2 = decr(B\_key, s2[2])

print('3. Трент расшифровывает E\_b(A, R\_a, T\_b)')

print('A =', dec\_s2[0])

print('R\_a =', dec\_s2[1])

print('T\_b =', dec\_s2[2])

print('Трент генерирует K =', K)

s3\_1 = encr(A\_key, (s2[0], dec\_s2[1], K, dec\_s2[2]))

s3\_2 = encr(B\_key, (dec\_s2[0], K, dec\_s2[2]))

s3 = (s3\_1, s3\_2, s2[1])

print("Трент отсылает Алисе {E\_a(B, R\_a, K, T\_b), E\_b(A, K, T\_b), R\_b} =", s3, '\n')

########4########

dec\_s3\_1 = decr(A\_key, s3[0])

print('4. Алиса расшифровывает E\_a(B, R\_a, K, T\_b)')

print('B =', dec\_s3\_1[0])

print('R\_a =', dec\_s3\_1[1])

print('K =', dec\_s3\_1[2])

print('T\_b =', dec\_s3\_1[3])

s4 = (s3[1], encr(dec\_s3\_1[2], {s3[2]}))

print("Алиса отсылает Бобу {E\_b(A, K, T\_b), E\_k(R\_b)} =", s4, '\n')

dec\_s4\_1 = decr(B\_key, s4[0])

dec\_s4\_2 = decr(bytes(dec\_s4\_1[1], 'utf-8'), s4[1])

print('Боб расшифровывает E\_b(A, K, T\_b)')

print('A =', dec\_s4\_1[0])

print('K =', dec\_s4\_1[1])

print('T\_b =', dec\_s4\_1[2])

print('Боб расшифровывает E\_k(R\_b)')

print('R\_b =', dec\_s4\_2[0], '\n', '\n')

########ПРОВЕРКА ПОДЛИННОСТИ########

########1########

print('ПРОВЕРКА ПОДЛИННОСТИ')

A\_rand2 = random.randint(1, 1000)

print('1. Алиса генерирует R2\_a =', A\_rand2)

ss1 = (s3\_2, str(A\_rand2))

print("Алиса отсылает Бобу {E\_b(A, K, T\_b), R2\_a} =", ss1, '\n')

########2########

B\_rand2 = random.randint(1, 1000)

print('2. Боб генерирует R2\_b =', B\_rand2)

ss2 = (str(B\_rand2), encr(dec\_s4\_1[1], {ss1[1]}))

print("Боб отсылает Алисе {R2\_b, E\_k(R2\_a)} =", ss2, '\n')

########3########

dec\_ss2 = decr(dec\_s3\_1[2], ss2[1])

print('3. Алиса расшифровывает E\_k(R2\_a)')

print('R2\_a =', dec\_ss2[0])

ss3 = encr(dec\_s4\_1[1], {ss2[0]})

print("Алиса отсылает Бобу {E\_k(R2\_b)} =", ss3, '\n')

dec\_ss3 = decr(bytes(dec\_s4\_1[1], 'utf-8'), ss3)

print('Боб расшифровывает E\_k(R2\_b)')

print('R2\_b =', dec\_ss3[0])