Лабораторная работа №6
КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

1. Описание задания

Реализовать протокол идентификации Feige-Fiat-Shamir.
Требования к работе:

- Разработка двух независимых модулей-участников протокола.
- Реализация должна позволять попытки ложной аутентификации.
- Количество раундов протокола должно быть параметром схемы.
- Отчет должен содержать описание протокола с указанием особенностей реализации, расчеты вероятности ложной идентификации, примеры работы программы.

Примечания:

- Допускается демонстрация работы алгоритма на маленьких числах (например, типа int).
- Ограничений на язык программирования нет.

2. Описание алгоритма

Протокол Фейга — Фиата — Шамира — протокол идентификации с нулевым разглашением, обобщение более раннего протокола Фиата — Шамира, разработанный Уриэлем Фейге , Амосом Фиато и Ади Шамиром в 1986 году.

Протокол позволяет одному участнику (доказывающему A) доказать другому участнику (проверяющему B), что он обладает секретной информацией, не раскрывая ни единого бита этой информации.

Безопасность протокола основана на сложности извлечения квадратного корня по модулю достаточно большого составного числа n, факторизация которого неизвестна.

Схема идентификации:

A доказывает своё знание секрета **s** стороне **B** в течение **t** раундов, не раскрывая при этом ни одного бита самого секрета.

Выбор параметров системы:

Доверенный центр **T** публикует большое число **n=p*q**, где **p** и **q** — простые числа, которые держатся в секрете. Также выбираются целые числа **k** и **t** - параметры безопасности.

Генерация секретов для участников:

Каждый участник выбирает k случайных целых чисел s_1,s_2, ..., s_k,

$1 \leq s_i \leq n-1$ и k случайных бит b_1, b_2, ..., b_k .

Затем вычисляет $v_i = (-1)^{b_i} * (s_i^2)^{-1} \; mod \; n, 1 \leq i \leq k.$

Участник идентифицирует себя окружающим с помощью значений (v_1, v_2, ..., v_k; n), которые выступают в качестве его открытого ключа, в то время как секретный ключ s = (s_1, s_2 , ..., s_k) известен только самому участнику.

Действия протокола в рамках одного раунда:

1. **A** выбирает случайное целое число $r, 1 \leq r \leq n-1$ и случайный бит b; вычисляет: $x = (-1)^b * r^2 \; mod \; n$ и отправляет x стороне **B**.
2. **B** отправляет **A** случайный k-битный вектор (e_1, e_2 , ..., e_k), где $e_i = 0$ или $e_i = 1$.
3. **A** вычисляет и отправляет **B**:

$$y = r * \prod_{i=1}^{k} {s_i}^{e_i} \; mod \; n$$

4. **B** вычисляет: $z = y^2 * \prod_{i=1}^{k} {v_i}^{e_i} \; mod \; n$ и проверяет, что $z = \pm x \; mod \; n$.

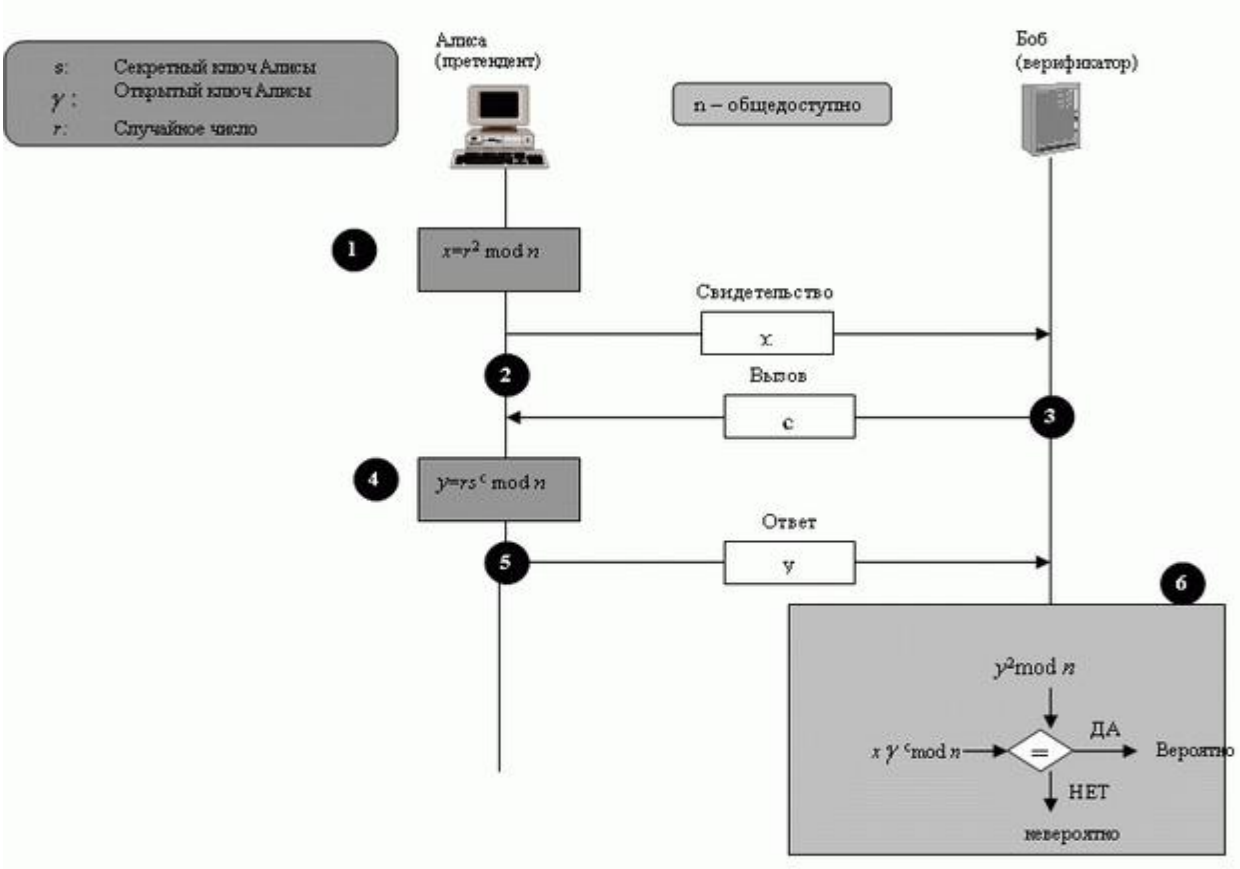

Рисунок 1. Протокол Фиата-Шамира.

Расчет вероятности ложной идентификации:

В своих работах Фейге, Фиат и Шамир показали, как параллельная схема может повысить количество аккредитаций на раунд и уменьшить объем взаимодействия между Пегги и Виктором.

Пегги и Виктор повторяют этот протокол t раз, пока Виктор не убедится, что Пегги знает числа (s_1, s_2 , ... , s_k). Вероятность, что Пегги удастся обмануть Виктора t раз, равна $\frac{1}{2kt}$. Авторы рекомендуют использовать вероятность мошенничества $\frac{1}{20kt}$ и предлагают значения k = 5 и t = 4.

3. Пример работы

```
апр. 07, 2022 2:00:02 AM org.suai.protocol.Application main
INFO: Enter the number of rounds:
1
апр. 07, 2022 2:00:05 AM org.suai.protocol.Application main
INFO: Enter the number k:
2
апр. 07, 2022 2:00:09 AM org.suai.protocol.Application authentication
INFO: n: 42388387743599721101388553649015057886931637954 9
апр. 07, 2022 2:00:09 AM org.suai.protocol.Application authentication
INFO: authentication successful
r: 362746493309031301
y: 191644103321460796002471712142616723341942311340
z: 131585018407999090425455130997752601
x: 42388387743586562599547753739972512373831862694 8
-x: 131585018407999090425455130997752601
e: [1, 1]
апр. 07, 2022 2:00:09 AM org.suai.protocol.Application falseAuthentication
INFO: n: 66010683550878899633942788616627638557461184839 9
апр. 07, 2022 2:00:09 AM org.suai.protocol.Application falseAuthentication
INFO: authentication successful
Был использован вектор e: [0, 0] - попробуем его угадать.
На данной итерации выбрали вектор e: [0, 0]
z: 397124367152513458731242831628005112660814716951
x: 397124367152513458731242831628005112660814716951
-x: 262982468356275537608185054538271272913797131448
```

Рисунок 2. Пример работы программы №1.

```
INFO: Enter the number of rounds:
2
апр. 07, 2022 2:03:54 AM org.suai.protocol.Application main
INFO: Enter the number k:
1
апр. 07, 2022 2:03:55 AM org.suai.protocol.Application authentication
INFO: n: 13057184368415107796704480950108501390261135930 11
апр. 07, 2022 2:03:55 AM org.suai.protocol.Application authentication
INFO: authentication successful
r: 774495733127050673
y: 774495733127050673
z: 599843640632007697242698491509752929
x: 13057184368409109360298160873136074405346038400 82
-x: 59984364063200769724269849150975292 9
e: [0]
r: 600726115118336909
y: 600726115118336909
z: 360871865385169368223360694031674281
x: 360871865385169368223360694031674281
-x: 13057184368411499078050629256426267783320819187 30
e: [0]
апр. 07, 2022 2:03:55 AM org.suai.protocol.Application falseAuthentication
INFO: n: 75999806612597873077629864880994921975214192634 7
апр. 07, 2022 2:03:55 AM org.suai.protocol.Application falseAuthentication
INFO: authentication successful
Был использован вектор e: [0] - попробуем его угадать.
На данной итерации выбрали вектор e: [0]
z: 735885981477678549660906472749760300189212546996
x: 735885981477678549660906472749760300189212546996
-x: 241120846483001811153921760601889195629293793 51
```

Рисунок 3. Пример работы программы №2.

```
апр. 07, 2022 2:05:31 AM org.suai.protocol.Application main
INFO: Enter the number of rounds:
2
апр. 07, 2022 2:05:33 AM org.suai.protocol.Application main
INFO: Enter the number k:
3
апр. 07, 2022 2:05:36 AM org.suai.protocol.Application authentication
INFO: n: 45438993542685764986169192631059280644917886667 7
апр. 07, 2022 2:05:36 AM org.suai.protocol.Application authentication
INFO: authentication successful
r: 996707146385186307
y: 17412598243807552743615493944808081499444721431 7
z: 99342513565530120560486135100298249
x: 99342513565530120560486135100298249
-x: 45438993542586422472603662510498793831407856842 8
e: [1, 0, 1]
r: 91097613659877357
y: 59896638428958443867277377546945288 1
z: 45438993542684935108647740203696647064089756122 8
x: 45438993542684935108647740203696647064089756122 8
-x: 82987752145242736263358082813054 49
e: [0, 1, 0]
апр. 07, 2022 2:05:36 AM org.suai.protocol.Application falseAuthentication
INFO: n: 69341728561190477954766552846836088684550069135 7
апр. 07, 2022 2:05:36 AM org.suai.protocol.Application falseAuthentication
INFO: authentication successful
Был использован вектор e: [1, 1, 1] - попробуем его угадать.
На данной итерации выбрали вектор e: [0, 0, 0]
z: 52214486754155746092197836839567542031940059413 7
x: 58115683030085176088308344167291338054436469626 9
-x: 11226045531105301866458208679544750630113599508 8
```

Рисунок 4. Пример работы программы №3.

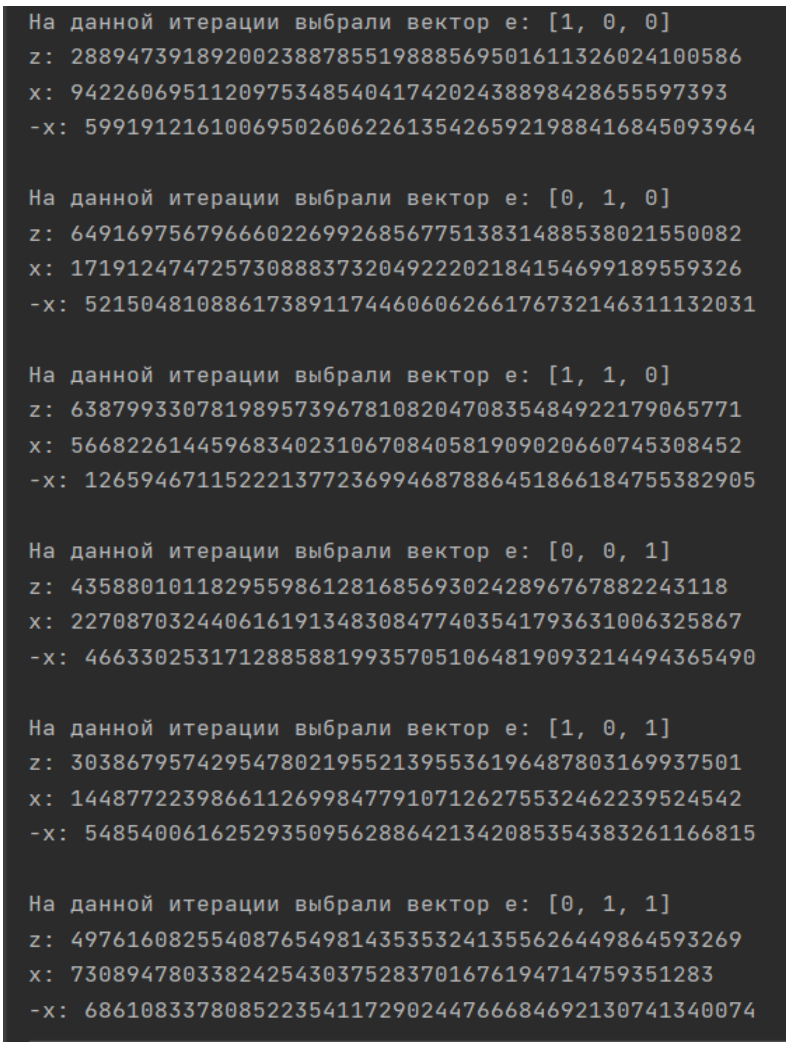

Рисунок 5. Пример работы программы №3.

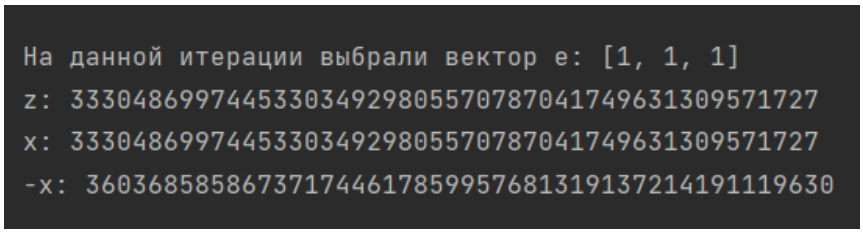

Рисунок 6. Пример работы программы №3.

4. Источники кода

1) **https://github.com/Frilock/feige-fiat-shamir-protocol**
2) **http://www.dialektika.com/PDF/978-5-9908462-4-1/part.pdf**

5. Вывод

Если **A** и **B** следуют протоколу, то **B** всегда принимает доказательства.

$$z = y^2 * \prod_{i=1}^{k} {v_i}^{e_i} = r^2 \prod_{i=1}^{k} ({s_i}^{2e_i} {v_i}^{e_i}) = \pm r^2 = \pm x \; mod \; n$$

Только одна из миллиона попыток нечестного пользователя обмануть проверяющего может увенчаться успехом.

Протокол доказывает, что **A** обладает своим секретным ключом, не раскрывая никакого знания о нём, когда $k = O(\log \log n)$ и $t = O(\log n)$.