

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО  
ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 51

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

М.Н. Исаева

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

по курсу: Криптографические методы защиты информации

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

5912

07.04.2022

подпись, дата

Д.В. Гольденберг

инициалы, фамилия

Санкт-Петербург 2022

# 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  $s_1, s_2, \dots, s_k$ ,

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

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

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

## Действия протокола в рамках одного раунда:

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

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

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

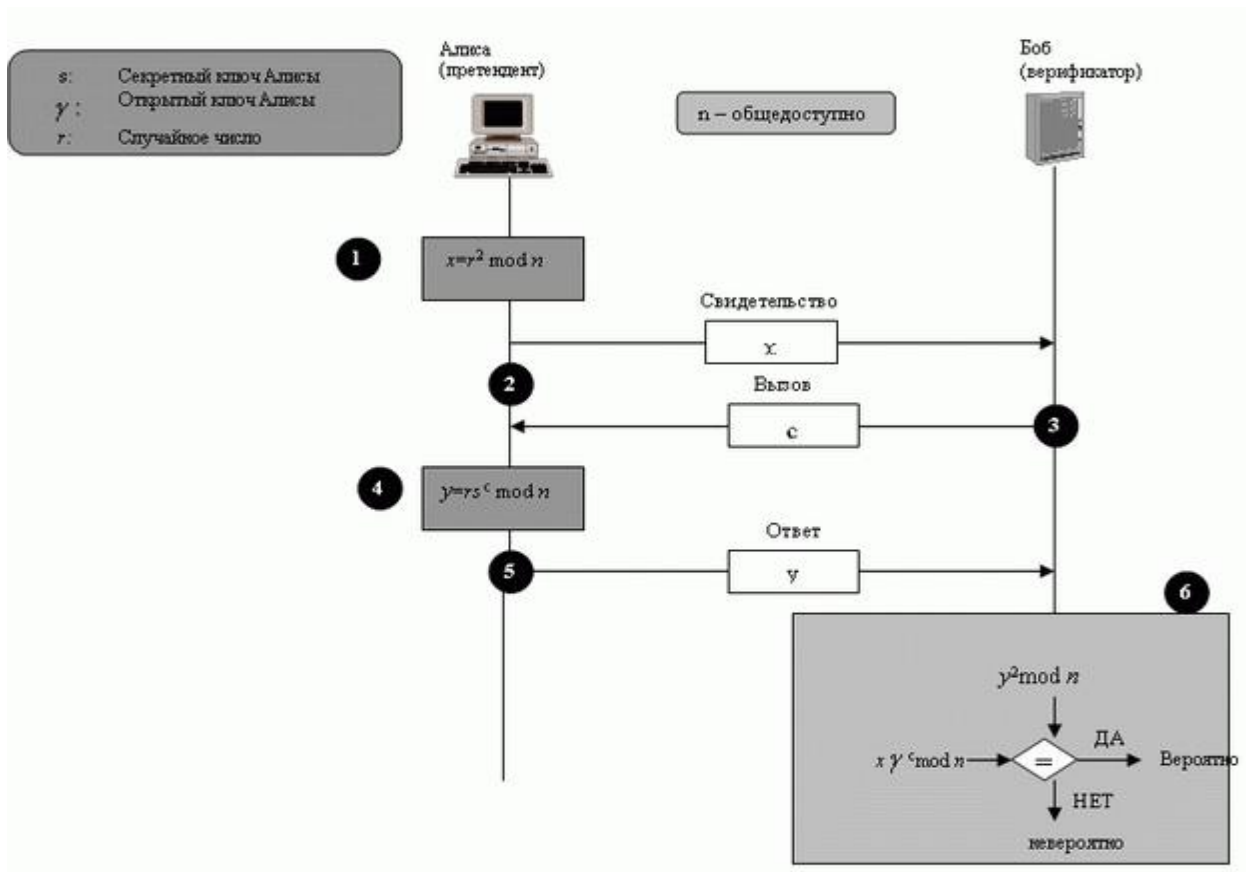


Рисунок 1. Протокол Фиата-Шамира.

### 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  $(s_1, s_2, \dots, s_k)$ . Вероятность, что Пегги удастся обмануть Виктора  $t$  раз, равна  $\frac{1}{2^{kt}}$ . Авторы рекомендуют использовать вероятность мошенничества  $\frac{1}{20kt}$  и предлагают значения  $k = 5$  и  $t = 4$ .

### 3. Пример работы

```
anp. 07, 2022 2:00:02 AM org.suai.protocol.Application main
INFO: Enter the number of rounds:
1
anp. 07, 2022 2:00:05 AM org.suai.protocol.Application main
INFO: Enter the number k:
2
anp. 07, 2022 2:00:09 AM org.suai.protocol.Application authentication
INFO: n: 423883877435997211013885536490150578869316379549
anp. 07, 2022 2:00:09 AM org.suai.protocol.Application authentication
INFO: authentication successful
r: 362746493309031301
y: 191644103321460796002471712142616723341942311340
z: 131585018407999090425455130997752601
x: 423883877435865625995477537399725123738318626948
-x: 131585018407999090425455130997752601
e: [1, 1]
anp. 07, 2022 2:00:09 AM org.suai.protocol.Application falseAuthentication
INFO: n: 660106835508788996339427886166276385574611848399
anp. 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
anp. 07, 2022 2:03:54 AM org.suai.protocol.Application main
INFO: Enter the number k:
1
anp. 07, 2022 2:03:55 AM org.suai.protocol.Application authentication
INFO: n: 1305718436841510779670448095010850139026113593011
anp. 07, 2022 2:03:55 AM org.suai.protocol.Application authentication
INFO: authentication successful
r: 774495733127050673
y: 774495733127050673
z: 599843640632007697242698491509752929
x: 1305718436840910936029816087313607440534603840082
-x: 599843640632007697242698491509752929
e: [0]
r: 600726115118336909
y: 600726115118336909
z: 360871865385169368223360694031674281
x: 360871865385169368223360694031674281
-x: 1305718436841149907805062925642626778332081918730
e: [0]
anp. 07, 2022 2:03:55 AM org.suai.protocol.Application falseAuthentication
INFO: n: 759998066125978730776298648809949219752141926347
anp. 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: 24112084648300181115392176060188919562929379351

```

*Рисунок 3. Пример работы программы №2.*

```

anp. 07, 2022 2:05:31 AM org.suai.protocol.Application main
INFO: Enter the number of rounds:
2
anp. 07, 2022 2:05:33 AM org.suai.protocol.Application main
INFO: Enter the number k:
1
anp. 07, 2022 2:05:36 AM org.suai.protocol.Application authentication
INFO: n: 454389935426857649861691926310592806449178866677
anp. 07, 2022 2:05:36 AM org.suai.protocol.Application authentication
INFO: authentication successful
r: 996707146385186307
y: 174125982438075527436154939448080814994447214317
z: 993425135655301205604868135100298249
x: 993425135655301205604868135100298249
-x: 454389935425864224726036625104987938314078568428
e: [1, 0, 1]
r: 91097613659877357
y: 598966384289584438672773775469452881
z: 454389935426849351086477402036966470640897561228
x: 454389935426849351086477402036966470640897561228
-x: 8298775214524273626335808281305449
e: [0, 1, 0]
anp. 07, 2022 2:05:36 AM org.suai.protocol.Application falseAuthentication
INFO: n: 693417285611904779547665528468360886845500691357
anp. 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: 522144867541557460921978368395675420319400594137
x: 581156830300851760883083441672913380544364696269
-x: 112260455311053018664582086795447506301135995088

```

*Рисунок 4. Пример работы программы №3.*

```

На данной итерации выбрали вектор e: [1, 0, 0]
z: 288947391892002388785519888569501611326024100586
x: 94226069511209753485404174202438898428655597393
-x: 599191216100695026062261354265921988416845093964

На данной итерации выбрали вектор e: [0, 1, 0]
z: 649169756796660226992685677513831488538021550082
x: 171912474725730888373204922202184154699189559326
-x: 521504810886173891174460606266176732146311132031

На данной итерации выбрали вектор e: [1, 1, 0]
z: 638799330781989573967810820470835484922179065771
x: 566822614459683402310670840581909020660745308452
-x: 126594671152221377236994687886451866184755382905

На данной итерации выбрали вектор e: [0, 0, 1]
z: 43588010118295598612816856930242896767882243118
x: 227087032440616191348308477403541793631006325867
-x: 466330253171288588199357051064819093214494365490

На данной итерации выбрали вектор e: [1, 0, 1]
z: 303867957429547802195521395536196487803169937501
x: 144877223986611269984779107126275532462239524542
-x: 548540061625293509562886421342085354383261166815

На данной итерации выбрали вектор e: [0, 1, 1]
z: 497616082554087654981435353241355626449864593269
x: 7308947803382425430375283701676194714759351283
-x: 686108337808522354117290244766684692130741340074

```

*Рисунок 5. Пример работы программы №3.*

```

На данной итерации выбрали вектор e: [1, 1, 1]
z: 333048699744533034929805570787041749631309571727
x: 333048699744533034929805570787041749631309571727
-x: 360368585867371744617859957681319137214191119630

```

*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.

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

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  $t = O(\log n)$ .