

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Б1.В.ДВ.03.02 Математические основы криптографии**

**Специальность** 10.05.03 Информационная безопасность автоматизированных систем

**Специализация** Информационная безопасность автоматизированных систем критически важных объектов

**Форма обучения** очная

## **СОДЕРЖАНИЕ**

### **1. Конспект лекций**

**Лекция № 1** Основные понятия и определения. История развития криптографии. Законодательные и правовые основы защиты компьютерной информации и информационных технологий.

**Лекция № 2** Энтропия, теоретическая и практическая стойкость, вычислительная стойкость.

**Лекция № 3** Блочные и поточные шифры. Шифры DES, режимы работы DES, AES, ГОСТ 28147-89.

**Лекция № 4** Поточные шифры: РСЛОС, RC4, шифр Рона.

**Лекция № 5** Общая схема функционирования систем с открытыми ключами.

**Лекция № 6** Криптосистема RSA и ее модификации. Криптосистема Эль Гамала. Криптосистема Рабина. Электронная цифровая подпись.

**Лекция № 7** Характеристика протоколов идентификации и аутентификации, идентификация на основе пароля. Взаимная проверка подлинности пользователей. Идентификация с нулевой передачей знаний. Схемы обязательств. Системы электронного голосования. Цифровые сертификаты: системы перераспределения доверия, неявные сертификаты.

**Лекция № 8** Тесты на простоту и факторизация. Надежность криптосистем. Элементы криптоанализа.

### **2. Методические указания по выполнению лабораторных работ**

**Лабораторная работа № 1** Поточные системы шифрования (РСЛОС, RC4, Рона)

**Лабораторная работа № 2** Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)

**Лабораторная работа № 3** Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)

**Лабораторная работа № 4** Асимметричные криптосистемы (*RSA*, El Gamal, Рабина)

**Лабораторная работа № 5-6** Программная реализация асимметричных криптосистем (*RSA*, El Gamal, Рабина)

**Лабораторная работа № 7-8** Исследование тестов на простоту и алгоритмы факторизации.

### **3. Методические указания по проведению практических занятий**

**Практическое занятие № 1** Поточные системы шифрования (РСЛОС, RC4, Рона)

**Практическое занятие № 2** Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)

**Практическое занятие № 3** Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)

**Практическое занятие № 4** Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)

**Практическое занятие № 5** Асимметричные криптосистемы (*RSA*, El Gamal, Рабина)  
Формирование асимметричных криптосистем

**Практическое занятие № 6** Асимметричные криптосистемы (*RSA*, El Gamal, Рабина)  
Формирование асимметричных криптосистем *RSA*

**Практическое занятие № 7** Асимметричные криптосистемы (*RSA*, El Gamal, Рабина)  
Формирование асимметричных криптосистем Рабина

**Практическое занятие № 8** Асимметричные криптосистемы (*RSA*, El Gamal, Рабина)  
Формирование асимметричных криптосистем El Gamal

# СОДЕРЖАНИЕ

## 1. КОНСПЕКТ ЛЕКЦИЙ

### 1. 1 Лекция № 1 (2 часа).

Тема: «Основные понятия и определения. История развития криптографии. Законодательные и правовые основы защиты компьютерной информации и информационных технологий.»

#### 1.1.1 Вопросы лекции:

1. Основные понятия и определения.

2. История развития криптографии. Классификация криптографических систем.

#### 1.1.2 Краткое содержание вопросов:

1. Основные понятия и определения.

Одно из основных понятий криптографии - (sifr) - «ноль», «ничто», «пустота», откуда фр. chiffré - «цифра»; арабы первыми стали заменять буквы на цифры с целью защиты исходного текста).

Под **шифром** понимается совокупность методов и способов обратимого преобразования информации с целью ее защиты от несанкционированного доступа (обеспечения конфиденциальности информации).

**Составными элементами шифра** являются:

- алфавиты для записи исходных сообщений (защищаемой информации, открытого текста) и шифрованных сообщений (шифртекстов, шифрограмм, криптограмм);
- алгоритмы криптографического преобразования (зашифрования и дешифрования);
- множество ключей.

**Азбука** или **алфавит** - форма письменности, основанная на стандартном наборе знаков, один или набор которых соответствуют фонемам<sup>1</sup> языка. В общем случае **алфавит для записи исходных сообщений** и **алфавит для записи шифрованных сообщений** могут отличаться.

**Алгоритм криптографического преобразования** — набор правил (инструкций), определяющих содержание и порядок операций по шифрованию и дешифрованию информации.

**Шифрование (зашифрование)** — процесс применения шифра к защищаемой информации, т.е. преобразование исходного сообщения в зашифрованное.

**Дешифрование** — процесс, обратный шифрованию, т. е. преобразование шифрованного сообщения в исходное.

**Ключ** — переменный параметр шифра, обеспечивающий выбор одного преобразования из совокупности всевозможных для данного алгоритма и сообщения. В общем случае, **ключ** — это минимально необходимая информация (за исключением сообщения, алфавитов и алгоритма), необходимая для шифрования и дешифрования сообщений.

**Криптография** (греч. κρυπτός — скрытый и γράφω — пишу, рисую) — наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства, а также невозможности отказа от авторства) информации.

**Криптоанализ** (греч. κρυπτός — скрытый и ἀνάλυση — разложение, расчленение) — наука, занимающаяся вопросами оценки сильных и слабых сторон методов шифрования,

а также разработкой методов, позволяющих взламывать криптосистемы.

## 2. История развития криптографии. Классификация криптографических систем.

История криптографии насчитывает около 4 тысяч лет. В качестве основного критерия периодизации криптографии возможно использовать технологические характеристики используемых методов шифрования.

Первый период (приблизительно с 3-го тысячелетия до н. э.) характеризуется господством моноалфавитных шифров (основной принцип — замена алфавита исходного текста другим алфавитом через замену букв другими буквами или символами). Второй период (хронологические рамки — с IX века на Ближнем Востоке и с XV века в Европе (Леон Баттиста Альберти) — до начала XX века) ознаменовался введением в обиход полиалфавитных шифров. Третий период (с начала и до середины XX века) характеризуется внедрением электромеханических устройств в работу шифровальщиков. При этом продолжалось использование полиалфавитных шифров.

Четвёртый период — с середины до 70-х годов XX века — период перехода к математической криптографии. В работе Шеннона появляются строгие математические определения количества информации, передачи данных, энтропии, функций шифрования. Обязательным этапом создания шифра считается изучение его уязвимости к различным известным атакам — линейному и дифференциальному криптоанализам. Однако, до 1975 года криптография оставалась «классической», или же, более корректно, криптографией с секретным ключом.

Современный период развития криптографии (с конца 1970-х годов по настоящее время) отличается зарождением и развитием нового направления — криптография с открытым ключом. Её появление знаменуется не только новыми техническими возможностями, но и сравнительно широким распространением криптографии для использования частными лицами (в предыдущие эпохи использование криптографии было исключительной прерогативой государства). Правовое регулирование использования криптографии частными лицами в разных странах сильно различается — от разрешения до полного запрета.

Современная криптография образует отдельное научное направление на стыке математики и информатики — работы в этой области публикуются в научных журналах, организуются регулярные конференции. Практическое применение криптографии стало неотъемлемой частью жизни современного общества — её используют в таких отраслях как электронная коммерция, электронный документооборот (включая цифровые подписи), телекоммуникации и других.

### 1. 2 Лекция № 2 (2 часа).

Тема: «Энтропия, теоретическая и практическая стойкость, вычислительная стойкость.»

#### 1.2.1 Вопросы лекции:

1. Энтропия, теоретическая и практическая стойкость, вычислительная стойкость.
2. Теоретико-информационная стойкость.

#### 1.2.2 Краткое содержание вопросов:

1. Энтропия, теоретическая и практическая стойкость, вычислительная стойкость.

Два подхода к определению стойкости криптографической системы Рассмотрим условия, которым должна удовлетворять криптосистема для надежной защиты информации. Стойкость зашифрованной информации (криптографическая стойкость, или просто стойкость) зависит от возможности несанкционированного чтения данных.

Существует два типа стойкости: теоретическая (математическая) и практическая. Эти концепции были предложены в классической работе Шеннона (Shannon, 1949). Термин "практическая стойкость" не означает, что определение не является

математически строгим. Стойкость обоих типов стойкости в следующем. Теоретическая стойкость основана на факте, что криптосистема моделируется некоторым формальным объектом, и для этой модели формулируются определенные условия невозможности раскола криптосистемы посторонним лицом. Обычно полагается, что доступная злоумышленнику информация должна быть недостаточной для определения открытого текста, даже если информация о криптосистеме несекретна. В качестве меры практической стойкости мы принимаем работу, т.е. число операций или временную сложность определения открытой информации посторонним лицом, либо средние значения этих характеристик над множеством всех открытых текстов. В этом случае цель состоит в получении максимальной сложности задачи несанкционированного дешифрования. Следует отметить, что существует два подхода к построению практически стойких шифров. В первом случае строится криптосистема, и затем показывается, что ее раскол является сложной задачей. Во втором случае выбирается некоторая сложная математическая задача, и затем строится соответствующая криптосистема, чей раскол эквивалентен решению этой задачи.

## 2. Теоретико-информационная стойкость

Вид стойкости теоретической, определяемый с точки зрения математической теории информации. С. т.-и. криптосистемы обычно характеризуется количеством информации (в смысле К. Шеннона) относительно неизвестного противнику и/или нарушителю элемента криптосистемы, содержащимся в перехваченном тексте шифрованном или других доступных данных и вычисленным в рамках той или иной вероятностной модели. Говорят также, что с. т.-и. криптосистемы характеризует ее способность противостоять атакам со стороны противника и/или нарушителя, располагающего неограниченными вычислительными ресурсами. Криптография занимается исследованием методов защиты информации и анализом их эффективности. Криптография представляет собой совокупность методов преобразования данных, направленных на то, чтобы сделать эти данные бесполезными для противника. Такие преобразования позволяют решить две главные проблемы защиты данных: проблему конфиденциальности (лишение противника возможности извлечь информацию из канала связи) и проблему целостности (лишение противника возможности изменить сообщение так, чтобы изменился его смысл, или ввести ложную информацию в канал связи).

Проблемы конфиденциальности и целостности информации тесно связаны между собой, поэтому методы решения одной из них часто применимы для решения другой. Криптография дает возможность преобразовать информацию таким образом, что ее прочтение (восстановление) возможно только при знании ключа. Под криптографическим ключом подразумевается некоторая секретная информация, известная законному собственнику (пользователю) информации и неизвестная нарушителю.

### 1. 3 Лекция № 3 (2 часа).

Тема: «Блочные и поточные шифры. Шифры DES, режимы работы DES, AES, ГОСТ 28147-89.»

#### 1.3.1 Вопросы лекции:

1. Блочные шифры.
2. Поточные шифры.

#### 1.3.2 Краткое содержание вопросов:

1. Блочные шифры.

**Блочный шифр** — разновидность симметричного шифра. Особенностью блочного шифра является обработка блока нескольких байт за одну итерацию (как правило 8 или 16). Блочные криптосистемы разбивают текст сообщения на отдельные блоки и затем осуществляют преобразование этих блоков с использованием ключа.

Преобразование должно использовать следующие принципы:

- Рассеивание (diffusion) — то есть изменение любого знака открытого текста или ключа влияет на большое число знаков шифротекста, что скрывает статистические свойства открытого текста;
- Перемешивание (confusion) — использование преобразований, затрудняющих получение статистических зависимостей между шифротекстом и открытым текстом.

К достоинствам блочных шифров относят похожесть процедур шифрования и расшифрования, которые, как правило, отличаются лишь порядком действий. Это упрощает создание устройств шифрования, так как позволяет использовать одни и те же блоки в цепях шифрования и дешифрования.

#### **Основная идея**

Блочный шифр состоит из двух взаимосвязанных алгоритмов: алгоритм шифрования  $E$  и алгоритм расшифрования  $E^{-1}$ . Входными данными служат блок размером  $n$  бит и  $k$ -битный ключ. На выходе получается  $n$ -битный зашифрованный блок. Для любого фиксированного ключа функция расшифрования является обратной к функции

шифрования  $E_K^{-1}(E_K(M)) = M$  для любого блока  $M$  и ключа  $K$ .

Для любого ключа  $K$ ,  $E_K$  является биективной функцией (перестановкой) на множестве  $n$ -битных блоков.

Размер блока  $n$  — это фиксированный параметр блочного шифра, обычно равный 64 или 128 битам, хотя некоторые шифры допускают несколько различных значений. Длина 64 бита была приемлема до середины 90-х годов, затем использовалась длина 128 бит, что примерно соответствует размеру машинного слова и позволяет эффективную реализацию на большинстве распространённых вычислительных платформах. Различные схемы шифрования позволяют зашифровывать открытый текст произвольной длины. Каждая имеет определённые характеристики: вероятность ошибки, простота доступа, уязвимость к атакам. Типичными размера ключа являются 40, 56, 64, 80, 128, 192 и 256 бит. В 2006 г. 80-битный ключ способен был предотвратить атаку грубой силой.

## **2. Поточные шифры.**

**Поточный шифр** — это симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста. Поточный шифр реализует другой подход к симметричному шифрованию, нежели блочные шифры.

#### **Классификация поточных шифров**

Допустим, например, что в режиме гаммирования для поточных шифров при передаче по каналу связи произошло искажение одного знака шифротекста. Очевидно, что в этом случае все знаки, принятые без искажения, будут расшифрованы правильно. Произойдёт потеря лишь одного знака текста. А теперь представим, что один из знаков шифротекста при передаче по каналу связи был потерян. Это приведёт к неправильному расшифрованию всего текста, следующего за потерянным знаком. Практически во всех каналах передачи данных для поточных систем шифрования присутствуют помехи. Поэтому для предотвращения потери информации решают проблему синхронизации шифрования и расшифрования текста. По способу решения этой проблемы шифрсистемы подразделяются на синхронные и системы с самосинхронизацией.

### **1. Синхронные поточные шифры**

Определение:

*Синхронные поточные шифры (СПШ)* — шифры, в которых поток ключей генерируется независимо от открытого текста и шифротекста.

При шифровании генератор потока ключей выдаёт биты потока ключей, которые идентичны битам потока ключей при дешифровании. Потеря знака шифротекста приведёт к нарушению синхронизации между этими двумя генераторами и невозможности расшифрования оставшейся части сообщения. Очевидно, что в этой ситуации отправитель и получатель должны повторно синхронизоваться для продолжения работы.

Обычно синхронизация производится вставкой в передаваемое сообщение специальных маркеров. В результате этого пропущенный при передаче знак приводит к неверному расшифрованию лишь до тех пор, пока не будет принят один из маркеров.

Заметим, что выполняться синхронизация должна так, чтобы ни одна часть потока ключей не была повторена. Поэтому переводить генератор в более раннее состояние не имеет смысла.

#### Плюсы СПШ:

- отсутствие эффекта распространения ошибок (только искажённый бит будет расшифрован неверно);
- предохраняют от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены.

#### Минусы СПШ:

- уязвимы к изменению отдельных бит шифрованного текста. Если злоумышленнику известен открытый текст, он может изменить эти биты так, чтобы они расшифровывались, как ему надо.

## 2. **Самосинхронизирующиеся поточные шифры**

#### Определение:

*Самосинхронизирующиеся поточные шифры (асинхронные поточные шифры (АПШ))* — шифры, в которых поток ключей создаётся функцией ключа и фиксированного числа знаков шифротекста.

Итак, внутреннее состояние генератора потока ключей является функцией предыдущих  $N$  битов шифротекста. Поэтому расшифрующий генератор потока ключей, приняв  $N$  битов, автоматически синхронизируется с шифрующим генератором.

Реализация этого режима происходит следующим образом: каждое сообщение начинается случайным заголовком длиной  $N$  битов; заголовок шифруется, передаётся и расшифровывается; расшифровка является неправильной, зато после этих  $N$  бит оба генератора будут синхронизованы.

#### Плюсы АПШ:

- Размешивание статистики открытого текста. Так как каждый знак открытого текста влияет на следующий шифротекст, статистические свойства открытого текста распространяются на весь шифротекст. Следовательно, АПШ может быть более устойчивым к атакам на основе избыточности открытого текста, чем СПШ.

#### Минусы АПШ:

- распространение ошибки (каждому неправильному биту шифротекста соответствуют  $N$  ошибок в открытом тексте);
- чувствительны к вскрытию повторной передачей.

### **Основные отличия поточных шифров от блочных**

Большинство существующих шифров с секретным ключом однозначно могут быть отнесены либо к поточным, либо к блочным шифрам. Но теоретическая граница между ними является довольно размытой. Например, используются алгоритмы блочного шифрования в режиме поточного шифрования (пример: для алгоритма DES режимы CFB



и OFB). Рассмотрим основные различия между поточными и блочными шифрами не только в аспектах их безопасности и удобства, но и с точки зрения их изучения в мире:

- важнейшим достоинством поточных шифров перед блочными является высокая скорость шифрования, соизмеримая со скоростью поступления входной информации; поэтому, обеспечивается шифрование практически в реальном масштабе времени вне зависимости от объема и разрядности потока преобразуемых данных.
- в синхронных поточных шифрах (в отличие от блочных) отсутствует эффект размножения ошибок, то есть число искаженных элементов в расшифрованной последовательности равно числу искаженных элементов зашифрованной последовательности, пришедшей из канала связи.
- структура поточного ключа может иметь уязвимые места, которые дают возможность криптоаналитику получить дополнительную информацию о ключе (например, при малом периоде ключа криптоаналитик может использовать найденные части поточного ключа для дешифрования последующего закрытого текста).
- ПШ в отличие от БШ часто могут быть атакованы при помощи линейной алгебры (так как выходы отдельных регистров сдвига с обратной линейной связью могут иметь корреляцию с гаммой). Также для взлома поточных шифров весьма успешно применяется линейный и дифференциальный анализ.

Теперь о положении в мире:

- в большинстве работ по анализу и взлому блочных шифров рассматриваются алгоритмы шифрования, основанные на стандарте DES; для поточных же шифров нет выделенного направления изучения; методы взлома ПШ весьма разнообразны.
- для поточных шифров установлен набор требований, являющихся критериями надёжности (большие периоды выходных последовательностей, постулаты Голомба, нелинейность); для БШ таких чётких критериев нет.
- исследованием и разработкой поточных шифров в основном занимаются европейские криптографические центры, блочных – американские.
- исследование поточных шифров происходит более динамично, чем блочных; в последнее время не было сделано никаких заметных открытий в сфере DES-алгоритмов, в то время как в области поточных шифров случилось множество успехов и неудач (некоторые схемы, казавшиеся стойкими, при дальнейшем исследовании не оправдали надежд изобретателей).

#### 1. 4 Лекция № 4 (2 часа).

Тема: «Поточные шифры: РСЛОС, RC4, шифр Рона»

##### 1.4.1 Вопросы лекции:

1. Характеристика шифров.
2. Режимы работы шифров.

##### 1.4.2 Краткое содержание вопросов:

###### 1. Характеристика шифров. Режимы работы шифров.

Большинство реальных поточных шифров основано на *регистрах сдвига с обратной связью*. Регистр сдвига применяют для генерации ключевой последовательности. Регистр сдвига с обратной связью состоит из двух частей: *регистра сдвига* и *функции обратной связи*.

Регистр сдвига представляет собой последовательность битов. (Количество битов определяется длиной сдвигового регистра. Если длина равна  $n$  битам, то регистр называется  $n$ -битовым регистром сдвига.) Всякий раз, когда нужно извлечь бит, все биты

регистра сдвига сдвигаются вправо на 1 позицию. Новый крайний левый бит является функцией всех остальных битов регистра.

На выходе регистра оказывается один, обычно младший значащий бит. Периодом регистра называется длина получаемой последовательности до начала ее повторения. Простейшим видом регистра сдвига с обратной связью является *регистр сдвига с линейной обратной связью (РСЛОС)*. Обратная связь представляет собой XOR некоторых битов регистра; эти биты называются *отводной последовательностью*.

РСЛОС ( $n$ -битовый) может находиться в одном из  $2^n - 1$  внутренних состояний. Это означает, что теоретически такой регистр может генерировать псевдослучайную последовательность с периодом  $2^n - 1$  битов. (Число внутренних состояний и период равны  $2^n - 1$ , потому что заполнение РСЛОС нулями приведет к тому, что сдвиговый регистр будет выдавать бесконечную последовательность нулей, что абсолютно бесполезно.) Только при определенных отводных последовательностях РСЛОС циклически пройдет через все  $2^n - 1$  внутренних состояний. Такие РСЛОС имеют максимальный период. Получившийся результат называется *М-последовательностью*. Для того чтобы конкретный РСЛОС имел максимальный период, многочлен, ассоциированный с отводной последовательностью, должен быть примитивным по модулю 2 — то есть не раскладываться на произведение двоичных многочленов меньшей степени.

Например, многочлен  $x^{32} + x^7 + x^5 + x^3 + x^2 + x + 1$  примитивен по модулю 2. Рассмотрим этот многочлен в терминах РСЛОС с максимальным периодом. Степень многочлена задает длину РСЛОС. Свободный член многочлена всегда равен 1, и его можно опустить. Степени формальной переменной многочлена, за исключением 0-й, задают отводную последовательность, отсчитываемую от левого края сдвигового регистра. То есть члены многочлена с меньшей степенью соответствуют позициям, расположенным ближе к правому краю регистра. Тогда для взятого 32-битового сдвигового регистра новый бит генерируется с помощью XOR тридцать второго, седьмого, пятого, третьего, второго и первого битов; получающийся РСЛОС будет иметь максимальную длину, циклически проходя до повторения через  $2^{32} - 1$  различных значений.

Сами по себе РСЛОС являются хорошими генераторами псевдослучайных последовательностей, но они обладают некоторыми нежелательными неслучайными свойствами. Для РСЛОС длины  $n$  внутреннее состояние представляет собой предыдущие  $n$  выходных битов генератора. Даже если схема обратной связи хранится в секрете, она может быть определена по  $2n$  выходным битам генератора с помощью алгоритма Берлекэмпа-Мэсси.

Кроме того, большие случайные числа, генерируемые с использованием идущих подряд бит этой последовательности, сильно коррелированы и для некоторых типов приложений вовсе не являются случайными. Несмотря на это, РСЛОС часто используются при разработке алгоритмов шифрования.

RC4 — это поточный шифр с переменным размером ключа, разработанный в 1987 г. Ривестом (R. Rivest) для RSA Data Security, Inc. Алгоритм работает в режиме OFB: поток ключей не зависит от открытого текста. Используется S-блок размером  $8 \times 8$ :  $S_0, S_1, S_2, \dots, S_{255}$ . Элементы представляют собой перестановку чисел от 0 до 255, а перестановка является функцией ключа переменной длины. В алгоритме применяются два счетчика,  $i$  и  $j$ , с нулевыми начальными значениями. Для генерации случайного байта выполняются следующие вычисления:

- $i = (i + 1) \bmod 256$ ;
- $j = (j + S_i) \bmod 256$ .
- Поменять местами  $S_i$  и  $S_j$ .
- $t = (S_i + S_j) \bmod 256$ ;
- $K = S_t$ .

$K$  используется в операции XOR с открытым текстом для получения шифротекста или в операции XOR с шифротекстом для получения открытого текста. Шифрование

выполняется примерно в 10 раз быстрее, чем в DES. Также несложна и инициализация S-блока. Сначала S-блок заполняется по правилу:  $S_0 = 0, S_1 = 1, \dots, S_{255} = 255$ . После этого ключ записывается в массив:  $K_0, K_1, \dots, K_{255}$ . Затем при начальном значении  $j = 0$  в цикле выполняются следующие вычисления:

- for  $i = 0$  to 255 do  $j = (j + S_i + K_i) \bmod 256$
- Поменять местами  $S_i$  и  $S_j$ .

Компания RSA Data Security, Inc. утверждает, что алгоритм устойчив к дифференциальному и линейному криптоанализу и что он в высокой степени нелинеен. S-блок медленно изменяется при использовании:  $i$  и  $j$  обеспечивают случайное изменение каждого элемента. RC4 входит в десятки коммерческих продуктов, включая Lotus Notes, AOCE компании Apple Computer и Oracle Secure SQL. Этот алгоритм также является частью спецификации стандарта Сотовой цифровой пакетной передачи данных CDPD (Cellular Digital Packet Data).

## 1. 5 Лекция № 5 (2 часа).

Тема: «Общая схема функционирования систем с открытыми ключами.»

### 1.5.1 Вопросы лекции:

1. Понятие открытого ключа.
2. Схема функционирования систем с открытым ключом.

### 1.5.2 Краткое содержание вопросов:

#### 1. Понятие открытого ключа.

**Ключ** это секретная информация, используемая криптографическим алгоритмом при шифровании/расшифровке сообщений, постановке и проверке цифровой подписи, вычислении кодов аутентичности (MAC). При использовании одного и того же алгоритма результат шифрования зависит от ключа. Для современных алгоритмов сильной криптографии утрата ключа приводит к практической невозможности расшифровать информацию.

Согласно принципу Керхгоффса, надёжность криптографической системы должна определяться сокрытием секретных ключей, но не сокрытием используемых алгоритмов или их особенностей.

**Открытый ключ** — ключ, который может быть опубликован и используется для проверки подлинности подписанного документа, а также для предупреждения мошенничества со стороны заверяющего лица в виде отказа его от подписи документа.

Открытый ключ подписи вычисляется, как значение некоторой функции от закрытого ключа, но знание открытого ключа не дает возможности определить закрытый ключ.

Идея криптографии с открытым ключом очень тесно связана с идеей односторонних функций, то есть таких функций  $f(x)$ , что по известному  $x$  довольно просто найти значение  $f(x)$ , тогда как определение  $x$  из  $f(x)$  невозможно за разумный срок.

Но сама односторонняя функция бесполезна в применении: ею можно зашифровать сообщение, но расшифровать нельзя. Поэтому криптография с открытым ключом использует односторонние функции с лазейкой. Лазейка — это некий секрет, который помогает расшифровать. То есть существует такой  $y$ , что зная  $f(x)$  и  $y$ , можно вычислить  $x$ . К примеру, если разобрать часы на множество составных частей, то очень сложно собрать вновь работающие часы. Но если есть инструкция по сборке (лазейка), то можно легко решить эту проблему.

Понять идеи и методы криптографии с открытым ключом помогает следующий пример — хранение паролей в компьютере. Каждый пользователь в сети имеет свой пароль. При входе он указывает имя и вводит секретный пароль. Но если хранить пароль на диске компьютера, то кто-нибудь его может считать (особенно легко это сделать администратору этого компьютера) и получить доступ к секретной информации. Для

решения задачи используется односторонняя функция. При создании секретного пароля в компьютере сохраняется не сам пароль, а результат вычисления функции от этого пароля и имени пользователя. Например, пользователь Алиса придумала пароль «Гладиолус».

При сохранении этих данных вычисляется результат функции  $f$  (АЛИСА\_ГЛАДИОЛУС), пусть результатом будет строка РОМАШКА, которая и будет сохранена в системе.

Когда Алиса вводит «секретный» пароль, компьютер проверяет, даёт или нет функция, применяемая к АЛИСА\_ГЛАДИОЛУС, правильный результат РОМАШКА, хранящийся на диске компьютера. Стоит изменить хотя бы одну букву в имени или в пароле, и результат функции будет совершенно другим. «Секретный» пароль не хранится в компьютере ни в каком виде. Файл паролей может быть теперь просмотрен другими пользователями без потери секретности, так как функция практически необратимая.

## 2. Схема функционирования систем с открытым ключом.

Пусть  $K$  — пространство ключей, а  $e$  и  $d$  — ключи шифрования и расшифрования соответственно.  $E_e$  — функция шифрования для произвольного ключа  $e \in K$ , такая что:

$$E_e(m) = c$$

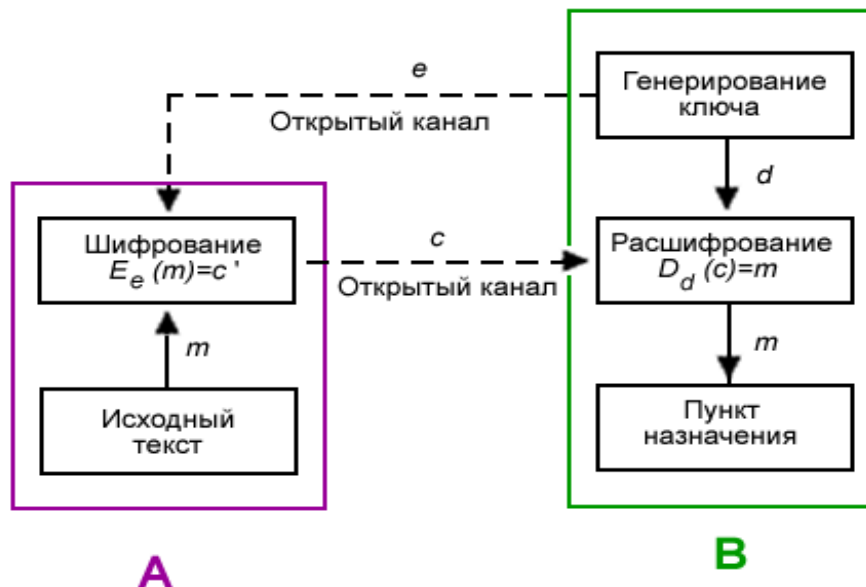
Здесь  $c \in C$ , где  $C$  — пространство шифротекстов, а  $m \in M$ , где  $M$  — пространство сообщений.

$D_d$  — функция расшифрования, с помощью которой можно найти исходное сообщение  $m$ , зная шифротекст  $c$  :

$$D_d(c) = m$$

$\{E_e: e \in K\}$  — набор шифрования, а  $\{D_d: d \in K\}$  — соответствующий набор для расшифрования. Каждая пара  $(E, D)$  имеет свойство: зная  $E_e$ , невозможно решить уравнение  $E_e(m) = c$ , то есть для данного произвольного шифротекста  $c \in C$ , невозможно найти сообщение  $m \in M$ . Это значит, что по данному  $e$  невозможно определить соответствующий ключ расшифрования  $d$ .  $E_e$  является односторонней функцией, а  $d$  — лазейкой.

Ниже показана схема передачи информации лицом А лицу В. Они могут быть как физическими лицами, так и организациями и так далее. Но для более лёгкого восприятия принято участников передачи отождествлять с людьми, чаще всего именуемыми Алиса и Боб. Участника, который стремится перехватить и расшифровать сообщения Алисы и Боба, чаще всего называют Евой.



1. Боб выбирает пару  $(e, d)$  и шлёт ключ шифрования  $e$  (открытый ключ) Алисе по открытому каналу, а ключ расшифрования  $d$  (закрытый ключ) защищён и секретен (он не должен передаваться по открытому каналу).
2. Чтобы послать сообщение  $m$  Бобу, Алиса применяет функцию шифрования, определённую открытым ключом  $e$ :  $E_e(m) = c$ ,  $c$  — полученный шифротекст.
3. Боб расшифровывает шифротекст  $c$ , применяя обратное преобразование  $D_d$ , однозначно определённое значением  $d$ .

## 1. 6 Лекция № 6 (2 часа).

Тема: «Криптосистема RSA и ее модификации. Криптосистема Эль Гамала.

Криптосистема Рабина. Электронная цифровая подпись. »

### 1.6.1 Вопросы лекции:

1. Криптосистема RSA и ее модификации.
2. Криптосистема Эль Гамала.
3. Криптосистема Рабина.

### 1.6.2 Краткое содержание вопросов:

1. Криптосистема RSA и ее модификации.

**RSA** (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и других.

Опубликованная в ноябре 1976 года статья Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» (англ. *New Directions in Cryptography*) перевернула представление о криптографических системах, заложив основы криптографии с открытым ключом. Разработанный впоследствии алгоритм Диффи — Хеллмана позволял двум сторонам получить общий секретный ключ, используя незащищенный канал связи. Однако этот алгоритм не решал проблему аутентификации.

Без дополнительных средств пользователи не могли быть уверены, с кем именно они сгенерировали общий секретный ключ.

Изучив эту статью, трое учёных Рональд Ривест, Ади Шамир и Леонард Адлеман из Массачусетского технологического института (MIT) приступили к поискам математической функции, которая бы позволяла реализовать сформулированную Уитфилдом Диффи и Мартином

Хеллманом модель криптографической системы с открытым ключом. После работы над более чем 40 возможными вариантами им удалось найти алгоритм, основанный на различии в том, насколько легко находить большие простые числа и насколько сложно раскладывать на множители произведение двух больших простых чисел, получивший впоследствии название RSA. Система была названа по первым буквам фамилий её создателей.

В 1982 году Ривест, Шамир и Адлеман организовали компанию RSA Data Security (англ.) (в настоящий момент — подразделение ЕМС). В 1989 году RSA, вместе с симметричным шифром DES, упоминается в RFC 1115, тем самым начиная использование алгоритма в зарождающейся сети Internet, а в 1990 году использовать алгоритм начинает министерство обороны США.

## 2. Криптосистема Эль Гамаля.

**Схема Эль-Гамаля (Elgamal)** — криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи в США(DSA) и России (ГОСТ Р 34.10-94).

Схема была предложена Тахером Эль-Гамалем в 1985 году.<sup>[1]</sup> Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

Генерация ключей:

1. Генерируется случайное простое число  $P$ .
2. Выбирается целое число  $g$  — первообразный корень  $P$ .
3. Выбирается случайное целое число  $x$  такое, что  $1 < x < p$ .
4. Вычисляется  $y = g^x \bmod p$ .
5. Открытым ключом является тройка  $(p, g, y)$ , закрытым ключом — число  $x$ .

Шифросистема Эль-Гамаля является фактически одним из способов выработки открытых ключей Диффи — Хеллмана. Шифрование по схеме Эль-Гамаля не следует путать с алгоритмом цифровой подписи по схеме Эль-Гамаля.

### Шифрование

Сообщение  $M$  должно быть меньше числа  $P$ . Сообщение шифруется следующим образом:

1. Выбирается сессионный ключ — случайное целое число  $k$  такое, что  $1 < k < p - 1$
2. Вычисляются числа  $a = g^k \bmod p$  и  $b = y^k M \bmod p$ .
3. Пара чисел  $(a, b)$  является шифротекстом.

Нетрудно видеть, что длина шифротекста в схеме Эль-Гамаля длиннее исходного сообщения  $M$  вдвое.

## Расшифрование

Зная закрытый ключ  $x$ , исходное сообщение можно вычислить из шифротекста  $(a, b)$  по формуле:

$$M = b(a^x)^{-1} \bmod p.$$

При этом нетрудно проверить, что

$$(a^x)^{-1} \equiv g^{-kx} \pmod{p}$$

и поэтому

$$b(a^x)^{-1} \equiv (y^k M) g^{-kx} \equiv (g^{kx} M) g^{-kx} \equiv M \pmod{p}.$$

Для практических вычислений больше подходит следующая формула:

$$M = b(a^x)^{-1} \bmod p = b \cdot a^{(p-1-x)} \bmod p$$

### 3. Криптосистема Рабина.

**Криптосистема Рабина** — криптографическая система с открытым ключом, безопасность которой обеспечивается сложностью поиска квадратных корней составного числа. Безопасность системы, как и безопасность метода RSA, обусловлена сложностью разложения на множители больших чисел. Зашифрованное сообщение можно расшифровать 4 способами. Недостатком системы является необходимость выбора истинного сообщения из 4-х возможных.

В январе 1979 года Майкл О. Рабин опубликовал описание своей системы. Было доказано, что восстановление исходного текста из зашифрованного столь же трудно, как факторизация больших чисел. Система Рабина стала первой асимметричной криптосистемой, для которой было выполнено такое доказательство. Сложность восстановления связана с трудностью извлечения квадратного корня по модулю составного числа  $N = p \cdot q$ . Задача факторизации и задача по извлечению квадратного корня эквивалентны, то есть:

1. зная простые делители числа  $N$  можно извлекать квадратные корни по модулю  $N$ ;
2. умея извлекать квадратные корни по модулю  $N$ , можно разложить  $N$  на простые множители.

Система Рабина, как и любая асимметричная криптосистема, использует открытый и закрытый ключи. Открытый ключ используется для шифрования сообщений и может быть опубликован для всеобщего обозрения. Закрытый ключ необходим для расшифровки и должен быть известен только получателям зашифрованных сообщений.

Процесс генерации ключей следующий:

- выбираются два случайных числа  $p$  и  $q$  с учётом следующих требований:
- числа должны быть большими (см. разрядность);
- числа должны быть простыми;
- должно выполняться условие:  $p \equiv q \equiv 3 \pmod{4}$ .

Выполнение этих требований сильно ускоряет процедуру извлечения корней по модулю  $p$  и  $q$ ;

- вычисляется число  $n = p \cdot q$ ;
- число  $n$  — открытый ключ; числа  $p$  и  $q$  — закрытый.

Пример. Пусть  $p = 7$  и  $q = 11$ . Тогда  $n = p \cdot q = 7 \cdot 11 = 77$ . Число  $n = 77$  — открытый ключ, а числа  $p = 7$  и  $q = 11$  — закрытый. Получатель сообщает отправителям число 77.

Отправители шифруют сообщение, используя число 77, и отправляют получателю.

Получатель расшифровывает сообщение с помощью чисел 7 и 11. Приведённые ключи плохи для практического использования, так как число 77 легко раскладывается на простые множители (7 и 11).

Исходное сообщение  $m$  (текст) шифруется с помощью открытого ключа — числа  $n$  по следующей формуле:

$$c = m^2 \bmod n.$$

Благодаря использованию умножения по модулю скорость шифрования системы Рабина больше, чем скорость шифрования по методу RSA, даже если в последнем случае выбрать небольшое значение экспоненты.

Пример (продолжение). Пусть исходным текстом является  $m = 20$ . Тогда зашифрованным текстом будет:  $c = m^2 \bmod n = 20^2 \bmod 77 = 400 \bmod 77 = 15$

Для расшифровки сообщения необходим закрытый ключ — числа  $p$  и  $q$ . Процесс расшифровки выглядит следующим образом:

- сначала, используя алгоритм Эвклида, из уравнения  $y_p \cdot p + y_q \cdot q = 1$  находят числа  $y_p$  и  $y_q$ ;

- далее, используя китайскую теорему об остатках, вычисляют четыре числа:

$$r = (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \bmod n$$

$$-r = n - r$$

$$s = (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \bmod n$$

$$-s = n - s$$

Одно из этих чисел является истинным открытым текстом  $m$ .

Пример (окончание). В результате расшифровки получаем:  $m \in \{64, 20, 13, 57\}$ .

Видим, что один из корней является исходным текстом  $m$ .

## 1. 7 Лекция № 7 (2 часа).

Тема: «Характеристика протоколов идентификации и аутентификации, идентификация на основе пароля. Взаимная проверка подлинности пользователей. Идентификация с нулевой передачей знаний. Схемы обязательств. Системы электронного голосования. Цифровые сертификаты: системы перераспределения доверия, неявные сертификаты.»

### 1.7.1 Вопросы лекции:

1. Характеристика протоколов идентификации и аутентификации, идентификация на основе пароля.

2. Взаимная проверка подлинности пользователей.

3. Идентификация с нулевой передачей знаний.

4. Схемы обязательств.

5. Системы электронного голосования.

6. Цифровые сертификаты: системы перераспределения доверия, неявные сертификаты.

### 1.7.2 Краткое содержание вопросов:

1. Характеристика протоколов идентификации и аутентификации, идентификация на основе пароля.

**Идентификация** - присвоение пользователям идентификаторов (уникальных имен или меток) под которыми система "знает" пользователя. Кроме идентификации пользователей, может проводиться *идентификация* групп пользователей, ресурсов ИС и т.д. *Идентификация* нужна и для других системных задач, например, для ведения журналов событий. В большинстве случаев *идентификация* сопровождается аутентификацией. **Аутентификация** - установление подлинности - проверка принадлежности пользователю предъявленного им идентификатора. Например, в начале сеанса работы в ИС *пользователь* вводит имя и *пароль*. На основании этих данных



система проводит идентификацию (*по имени пользователя*) и аутентификацию (сопоставляя *имя пользователя* и введенный *пароль*).

Система идентификации и аутентификации является одним из ключевых элементов инфраструктуры защиты от несанкционированного доступа (НСД) любой информационной системы. В соответствии с рассмотренной ранее моделью многоуровневой защиты, *аутентификация* пользователя компьютера относится к уровню защиты узлов.

Обычно выделяют 3 группы методов аутентификации.

1. Аутентификация по наличию у пользователя уникального объекта заданного типа. Иногда этот класс методов аутентификации называют по-английски "I have" ("у меня есть"). В качестве примера можно привести аутентификацию с помощью смарт-карт или электронных USB-ключей.
2. Аутентификация, основанная на том, что пользователю известна некоторая конфиденциальная информация - "I know" ("я знаю"). Например, аутентификация по паролю. Более подробно парольные системы рассматриваются далее в этом разделе.
3. Аутентификация пользователя по его собственным уникальным характеристикам - "I am" ("я есть"). Эти методы также называются биометрическими.

Нередко используются комбинированные схемы аутентификации, объединяющие методы разных классов. Например, двухфакторная *аутентификация* - *пользователь* предъявляет системе смарт-карту и вводит пин-код для ее активации.

Наиболее распространенными на данный момент являются парольные *системы аутентификации*. У пользователя есть *идентификатор* и *пароль*, т.е.

секретная *информация*, известная только пользователю (и возможно - системе), которая используется для прохождения аутентификации.

В зависимости от реализации системы, *пароль* может быть одноразовым или многоразовым. Операционные системы, как правило, проводят аутентификацию с использованием многоразовых паролей. Совокупность идентификатора, пароля и, возможно, дополнительной информации, служащей для описания пользователя составляют **учетную запись пользователя**.

Если нарушитель узнал *пароль* легального пользователя, то он может, например, войти в систему под его учетной записью и получить *доступ* к конфиденциальным данным.

Поэтому безопасности паролей следует уделять особой внимание.

Как отмечалось при рассмотрении стандарта *ISO 17799*, рекомендуется, чтобы пользователи системы подписывали документ о сохранении конфиденциальности пароля.

Но нарушитель также может попытаться подобрать *пароль*, угадать его, перехватить и т.д.

Рассмотрим некоторые рекомендации *по* администрированию парольной системы, позволяющие снизить *вероятность* реализации подобных угроз.

1. Задание минимальной длины используемых в системе паролей. Это усложняет атаку путем подбора паролей. Как правило, рекомендуют устанавливать минимальную длину в 6-8 символов.
2. Установка требования использовать в пароле разные группы символов - большие и маленькие буквы, цифры, специальные символы. Это также усложняет подбор.
3. Периодическая проверка *администраторами безопасности* качества используемых паролей путем *имитации атак*, таких как подбор паролей "по словарю" (т.е. проверка на использование в качестве пароля слов естественного языка и простых комбинаций символов, таких как "1234").
4. Установка максимального и минимального сроков жизни пароля, использование механизма принудительной смены старых паролей.

5. Ограничение числа неудачных попыток ввода пароля (блокирование учетной записи после заданного числа неудачных попыток войти в систему).
6. Ведение журнала истории паролей, чтобы пользователи, после принудительной смены пароля, не могли вновь выбрать себе старый, возможно скомпрометированный пароль.

## 2. Взаимная проверка подлинности пользователей.

Обычно стороны, вступающие в информационный обмен, нуждаются во взаимной проверке подлинности (аутентификации) друг друга. Этот процесс взаимной аутентификации выполняют в начале сеанса связи.

Для проверки подлинности применяют следующие способы:

- механизм запроса-ответа;
- механизм отметки времени ("временной штампель").

Механизм запроса-ответа состоит в следующем. Если пользователь А хочет быть уверенным, что сообщения, получаемые им от пользователя В, не являются ложными, он включает в посылаемое для В сообщение непредсказуемый элемент - запрос Х (например, некоторое случайное число). При ответе пользователь В должен выполнить над этим элементом некоторую операцию (например, вычислить некоторую функцию  $f(X)$ ). Это невозможно осуществить заранее, так как пользователю В неизвестно, какое случайное число Х придет в запросе. Получив ответ с результатом действий В, пользователь А может быть уверен, что В - подлинный. Недостаток этого метода - возможность установления закономерности между запросом и ответом.

Механизм отметки времени подразумевает регистрацию времени для каждого сообщения. В этом случае каждый пользователь сети может определить, насколько "устарело" пришедшее сообщение, и решить не принимать его, поскольку оно может быть ложным. В обоих случаях для защиты механизма контроля следует применять шифрование, чтобы быть уверенным, что ответ послан не злоумышленником.

При использовании отметок времени возникает проблема допустимого временного интервала задержки для подтверждения подлинности сеанса. Ведь сообщение с "временным штампелем" в принципе не может быть передано мгновенно. Кроме того, компьютерные часы получателя и отправителя не могут быть абсолютно синхронизированы. Какое запаздывание "штампеля" является подозрительным?

Для взаимной проверки подлинности обычно используют процедуру рукопожатия. Эта процедура базируется на указанных выше механизмах контроля и заключается во взаимной проверке ключей, используемых сторонами. Иначе говоря, стороны признают друг друга законными партнерами, если докажут друг другу, что обладают правильными ключами. Процедуру рукопожатия обычно применяют в компьютерных сетях при организации сеанса связи между пользователями, пользователем и хост-компьютером, между хост-компьютерами и т.д. Рассмотрим в качестве примера процедуру рукопожатия для двух пользователей А и В. (Это допущение не влияет на общность рассмотрения. Такая же процедура используется, когда вступающие в связь стороны не являются пользователями). Пусть применяется симметричная криптосистема. Пользователи А и В разделяют один и тот же секретный ключ  $K_{AB}$ .

## 3. Идентификация с нулевой передачей знаний.

Широкое распространение смарт-карт (интеллектуальных карт) для разнообразных коммерческих, гражданских и военных применений (кредитные карты, карты социального страхования, карты доступа в охраняемые помещения, компьютерные пароли и ключи и т.д.) потребовало обеспечения безопасности идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении смарт-карты оперативно обнаружить обман и отказать обманщику в допуске, ответе и обслуживании.

Для безопасного использования смарт-карт разработаны протоколы идентификации с нулевой передачей знаний. Секретный ключ владельца карты становится неотъемлемым признаком его личности. Доказательство знания этого секретного ключа с нулевой передачей этого знания служит доказательством подлинности личности владельца карты.

Схему идентификации с нулевой передачей знаний предложили в 1986 г. У.Фейге, А.Фиат и А.Шамир. Она является наиболее известным доказательством идентичности с нулевой передачей конфиденциальной информации.

Рассмотрим сначала упрощенный вариант схемы идентификации с нулевой передачей знаний для более четкого выявления ее основной концепции.

Но прежде всего определимся с терминологией.

Пусть  $a, b, d, g \in \mathbf{Z}$  (множество целых чисел),  $n \in \mathbf{N}$  (множество натуральных чисел, т.е. положительных целых чисел).

**Определение 1.** Число  $a$  *сравнимо* с  $b$  по модулю  $n$ , если  $a$  и  $b$  при делении на  $n$  дают одинаковые остатки, т.е.  $a \bmod n = b \bmod n$ .

Принятое обозначение:  $a \in b \pmod{n}$ .

**Определение 2.** Число  $d$  называют *обратным* к  $a$  по модулю  $n$ , если произведение  $a*d$  при делении на  $n$  дает в остатке 1, т.е.  $a*d \bmod n = 1$ .

Принятое обозначение:  $b = a^{-1} \pmod{n}$ .

Примите к сведению, что целое число  $a^{-1} \pmod{n}$  существует тогда и только тогда, когда  $a$  является взаимно простым с  $n$ , т.е. имеет с модулем  $n$  наибольший общий делитель, равный единице.

Тех, кому интересно, почему это действительно так, отсылаю к расширенному алгоритму Евклида. В Интернете вы найдете не один сайт, посвященный этой теме.

**Определение 3.** Число  $g$  называют *квадратным корнем* из  $a$  по модулю  $n$ , если произведение  $g*g$  при делении на  $n$  дает в остатке  $a$ , т.е.  $g*g \bmod n = a$ .

Принятое обозначение:  $g = \text{sqrt}(a) \pmod{n}$ .

#### 4. Схемы обязательств.

В криптографии, **схема обязательств** или **битовая схема**

**обязательств** (англ. *Commitment scheme*) — это метод, позволяющий пользователю подтверждать какое-либо значение, которое не разглашается, то есть в случае разглашения этого значения благодаря этой схеме будет известно, что пользователь знал его на момент выдачи обязательства и что оно не изменилось. Работу данной схемы можно представить как две стадии:

- «Commit» — посылку закрытой на ключ коробки (обязательство),
- «Reveal» — более поздняя отправка ключа от коробки (значение).

Одна из популярных реализаций схемы: Pedersen Commitment Scheme (основана на задаче дискретного логарифмирования)

#### 5. Системы электронного голосования.

**Электронное голосование** — термин, определяющий различные виды голосования, охватывающий как электронные средства голосования (электронная демократия), так и технические электронные средства подсчёта голосов. Разновидностью электронного голосования являются Интернет-выборы.

Технологии электронного голосования могут включать в себя перфокарты, системы оптического сканирования и специализированные терминалы для голосования. Они также могут включать передачу избирательных бюллетеней и голосов по телефону, частным компьютерным сетям или через Интернет.

Технология электронного голосования позволяет ускорить процесс подсчёта голосов, а также упростить голосование людям с ограниченными возможностями. Но в настоящее время ведутся споры о том, что электронное голосование может быть подвержено нарушениям (большим, чем при традиционных системах голосования).

#### Устройства подсчёта голосов

Системы электронного подсчёта голосов применяются на выборах с 1960-х годов, с тех пор, как появились перфокарты.

Более новая система оптического сканирования может считывать с бюллетеня отметку, поставленную избирателем.

Системы прямой записи голосов, накапливающие голоса на одном устройстве, используются повсеместно в Бразилии, также достаточно широко распространены в Индии, Нидерландах, Венесуэле и США.

Системы Интернет-голосования завоевали популярность и используются в правительственных выборах и референдумах в Великобритании, Эстонии и Швейцарии, а также муниципальных выборах в Канаде и партийных выборах в США и Франции.

### **Устройства заполнения**

Существуют системы, включающие в себя и устройство заполнения электронного бюллетеня (сенсорный экран, либо сканер штрих-кода). Также они, зачастую, оснащены дополнительным вспомогательным устройством для распечатки бумажной копии бюллетеня либо квитанции о голосовании. Хранение и подсчёт голосов при этом происходит на отдельном устройстве.

6. Цифровые сертификаты: системы перераспределения доверия, неявные сертификаты.

**Цифровой сертификат** — выпущенный удостоверяющим центром электронный или печатный документ, подтверждающий принадлежность владельцу открытого ключа или каких-либо атрибутов.

### **Сертификат открытого ключа**

Сертификат открытого ключа удостоверяет принадлежность открытого ключа некоторому субъекту, например, пользователю. Сертификат открытого ключа содержит имя субъекта, открытый ключ, имя удостоверяющего центра, политику использования соответствующего удостоверяемому открытому ключу закрытого ключа и другие параметры, заверенные подписью удостоверяющего центра.

Сертификат открытого ключа используется для идентификации субъекта и уточнения операций, которые субъекту разрешается совершать с использованием закрытого ключа, соответствующего открытому ключу, удостоверяемому данным сертификатом.

Формат сертификата открытого ключа X.509 v3 описан в RFC 5280

### **Сертификат атрибутов**

Структура сертификата атрибутов аналогична структуре сертификата открытого ключа. Отличие же заключается в том, что сертификат атрибутов удостоверяет не открытый ключ субъекта, а какие-либо его атрибуты — принадлежность к какой-либо группе, роль, полномочия и т. п. Сертификат атрибутов применяется для авторизации субъекта. Формат сертификата атрибутов описан в RFC 5755.

### **Классификация сертификатов**

VeriSign предложила следующую концепцию классификации цифровых сертификатов :

- Class 1 — индивидуальные, для идентификации электронной почты;
- Class 2 — для организаций;
- Class 3 — для серверов и программного обеспечения;
- Class 4 — для онлайн-бизнеса и транзакций между компаниями;
- Class 5 — для частных компаний или правительственной безопасности.

Тема: «Тесты на простоту и факторизация. Надежность криптосистем. Элементы криптоанализа.»

### 1.8.1 Вопросы лекции:

1. Тесты на простоту: пробное деление, тест Ферма, тест Миллера-Рабина.

2. Алгоритмы факторизации: пробное деление, гладкие числа, (P-1)-метод Полларда, разность квадратов, современные методы факторизации.

### 1.8.2 Краткое содержание вопросов:

1. Тесты на простоту: пробное деление, тест Ферма, тест Миллера-Рабина.

Вопрос определения того, является ли натуральное число  $N$  простым, известен как проблема простоты.

**Тестом простоты** (или проверкой простоты) называется алгоритм, который, приняв на входе число  $N$ , позволяет либо не подтвердить предположение о составности числа, либо точно утверждать его простоту. Во втором случае он называется истинным тестом простоты. Таким образом, тест простоты представляет собой только гипотезу о том, что если алгоритм не подтвердил предположение о составности числа  $N$ , то это число может являться простым с определенной вероятностью. Это определение подразумевает меньшую уверенность в соответствии результата проверки истинному положению вещей, нежели истинное испытание на простоту, которое дает математически подтвержденный результат.

**Тест простоты Ферма** в теории чисел — это тест простоты натурального числа  $n$ , основанный на малой теореме Ферма.

Если  $n$  — простое число, то оно удовлетворяет сравнению  $a^{n-1} \equiv 1 \pmod{n}$  для любого  $a$ , которое не делится на  $n$ .

Выполнение сравнения  $a^{n-1} \equiv 1 \pmod{n}$  является необходимым, но не достаточным признаком простоты числа. То есть, если найдётся хотя бы одно  $a$ , для которого  $a^{n-1} \not\equiv 1 \pmod{n}$ , то число  $n$  — составное; в противном случае ничего сказать нельзя, хотя шансы на то, что число является простым, увеличиваются. Если для составного числа  $n$  выполняется сравнение  $a^{n-1} \equiv 1 \pmod{n}$ , то число  $n$  называют **псевдопростым по основанию  $a$** . При проверке числа на простоту тестом Ферма выбирают несколько чисел  $a$ . Чем больше количество  $a$ , для которых  $a^{n-1} \equiv 1 \pmod{n}$ , тем больше шансы, что число  $n$  простое. Однако существуют составные числа, для которых сравнение  $a^{n-1} \equiv 1 \pmod{n}$  выполняется для всех  $a$ , взаимно простых с  $n$  — это числа Кармайкла. Чисел Кармайкла — бесконечное множество, наименьшее число Кармайкла — 561. Тем не менее, тест Ферма довольно эффективен для обнаружения составных чисел.

**Тест Миллера — Рабина** — вероятностный полиномиальный тест простоты. Тест Миллера — Рабина, наряду с тестом Ферма и тестом Соловея — Штрассена, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера — Рабина часто используется в криптографии для получения больших случайных простых чисел.

2. Алгоритмы факторизации: пробное деление, гладкие числа, (P-1)-метод Полларда, разность квадратов, современные методы факторизации.

натурального числа называется его разложение в произведение простых множителей. Существование и единственность (с точностью до порядка следования множителей) такого разложения следует из основной теоремы арифметики.

В отличие от задачи распознавания простоты числа, факторизация предположительно является вычислительно сложной задачей. В настоящее время неизвестно, существует

ли эффективный не квантовый алгоритм факторизации целых чисел. Однако доказательства того, что не существует решения этой задачи за полиномиальное время также нет.

Предположение о том, что для больших чисел задача факторизации является вычислительно сложной лежит в основе широко используемых алгоритмов (например, RSA). Множество областей математики и информатики находят применение в решении этой задачи. Среди них: эллиптические кривые, алгебраическая теория чисел и квантовые вычисления.

Одним из ключевых моментов в развитии факторизации целых чисел было создание алгоритма RSA, что возобновило интерес ученых в данном направлении, так как имело практическое применение в области шифрования. Этот алгоритм был предложен в 1977 году тремя учеными Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом из Массачусетского Технологического Института и назван по первым буквам фамилий авторов методом RSA. Он основан на идее криптографии с открытым ключом и для взлома системы необходимо выполнить разложение числа на простые сомножители. На момент публикации алгоритма RSA были известны методы, которые позволяли факторизовать числа, состоящие не более чем из 25—30 цифр, а наиболее известным и применяемым все ещё оставался метод Ферма. Метод RSA позволяет факторизовывать числа из 100 и более десятичных знаков. Создатели, в свою очередь, пообещали за факторизацию числа из 129 десятичных знаков символические сто долларов США<sup>[2]</sup>. На популярность задачи факторизации также повлияла публикация в 1977 году в журнале Scientific American Мартина Гарднера «Новый алгоритм шифрования, для взлома которого потребуется миллионы лет». Столь громкое название было воспринято в качестве вызова всему математическому сообществу. В результате этой гонки было предложено несколько новых и нестандартных идей факторизации<sup>[2]</sup>. Эпопея с разложением 129-значного числа завершилась в 1994 году, когда коллектив под руководством А. Ленстры, используя 1600 компьютеров, подготовил за 220 дней систему линейных уравнений, содержащую более полумиллиона неизвестных. Решение этой системы суперкомпьютером заняло два дня. Несмотря на то, что в то время уже были известны методы решета числового поля, данный результат был получен с помощью алгоритма квадратичного решета.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

### **2.1 Лабораторная работа №1 ( 2 часа).**

**Тема:** «Поточные системы шифрования (РСЛОС, RC4, Рона)»

**2.1.1 Цель работы:** Исследование систем РСЛОС, Рона, RC4

**2.1.2 Задачи работы:**

1. Изучить потоковые алгоритмы шифрования.

**2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Вычислительная машина.

**2.1.4 Описание (ход) работы:**

1. Ознакомьтесь с теоретическими основами шифрования данных в учебном пособии «Криптография», а также в учебно-методическом пособии к выполнению лабораторного практикума по дисциплине «Криптография».

2. Получите вариант задания у преподавателя.
3. Напишите программу согласно варианту задания.
4. Отладьте разработанную программу и покажите результаты работы программы преподавателю.
5. Составьте отчет по лабораторной работе

## **2.2 Лабораторная работа № 2 ( 2 часа).**

**Тема:** «Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)»

**2.2.1 Цель работы:** Исследование поточных систем шифрования.

**2.2.2 Задачи работы:**

1. Изучить потоковые алгоритмы шифрования.

**2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Электронно-вычислительная машина.

**2.2.4 Описание (ход) работы:**

1. Ознакомьтесь с теоретическими основами шифрования данных в учебном пособии «Криптография», а также в учебно-методическом пособии к выполнению лабораторного практикума по дисциплине «Криптография».
2. Получите вариант задания у преподавателя.
3. Напишите программу согласно варианту задания.
4. Отладьте разработанную программу и покажите результаты работы программы преподавателю.
5. Составьте отчет по лабораторной работе

## **2.3 Лабораторная работа № 3 ( 2 часа).**

**Тема:** «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)»

**2.3.1 Цель работы:** Исследование схем распределения ключей по схеме Диффи-Хеллмана.

**2.3.2 Задачи работы:**

1. Освоить алгоритм обмена ключами

**2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Электронно-вычислительная машина.

**2.3.4 Описание (ход) работы:**

**Задание .**

Открытый элемент  $P$  задан в табл. 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы  $i$ , вычислить ключ обмена для участника с номером  $35 - i$  по алгоритму Диффи – Хеллмана

### Пример выполнения Задания3. Алгоритм Диффи-Хеллмана

Открытый элемент  $P$  задан в таблице 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы  $i$ , вычислить ключ обмена для участника с номером  $35 - i$  по алгоритму Диффи-Хеллмана.

Вариант № 15, группа 2091 (№ 1)

Номер в списке группы  $i = 15$

Номер группы  $k = 1$

$P = 37$

Открытый элемент  $P = 37$ . Найти примитивный элемент поля. Секретный ключ каждого участника  $i = 15$ , вычислить ключ обмена для участника с номером  $35 - 15 = 20$ .

$GF(37) = \langle 0, 1, 2, 3, \dots, 35, 36, 37 \rangle$

Найдем примитивный элемент поля  $GF(37)$ .

Требуется найти такое число, принадлежащее интервалу  $[2, 37]$ , которое при возведении в 37-ю степень по модулю 37 будет давать в результате единицу. Если же единица будет получена раньше, чем при возведении в 36-ю степень, результаты возведения в степень начнут повторяться, и через выбранный элемент не удастся представить все элементы поля. Исходя из таких соображений, получаем несложный алгоритм нахождения примитивных элементов поля  $GF(37)$ .

Алгоритм нахождения примитивных элементов поля  $GF(37)$ .

```
int _tmain(int argc, _TCHAR* argv[])
{
    long int mem, i, j, num, deg, modul, res;
    unsigned char mas[38];
    deg = 0; modul = 37; mem = 2;
    while (mem < 36)
    {
        num = mem; deg = 0;
        while (deg != 36)
        {
            res = 1; deg = 0;
            for (i = 0; i < 36; i++) mas[i] = 0;
            do
            {
                res = res*num; res = res % modul;
                deg++; mas[deg] = res;
            }
            while(res != 1);
            num++;
        }
        std::cout << num-1 << std::endl;
```

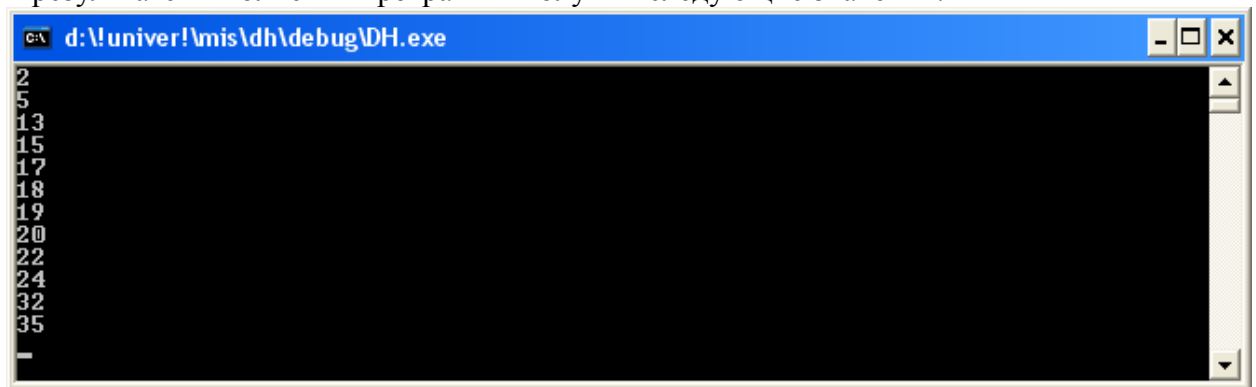


```

    for (i = 0; i < 36; i++)
    {
        for (j = 0; j < 36; j++)
        {
            if ((i != j) && (mas[j] == mas[i]))
                std::cout << "Wrong" << std::endl;
        }
        mem = num;
    }
}

```

В результате выполнения программы получим следующие значения:



2, 5, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35

Выберем значение  $\alpha = 2$ .

Секретный ключ участника А:  $X_A = 15$ .

Секретный ключ участника В:  $Y_B = 35 - 15 = 20$ .

Открытый ключ участника А:  $K_A = \alpha^{X_A} = 2^{15} \bmod 37 = 23$

Открытый ключ участника В:  $K_B = \alpha^{Y_B} = 2^{20} \bmod 37 = 33$

$K_A = 23$

$K_B = 33$

Обменный ключ участника А:  $K = K_B^{X_A} \bmod \alpha = 33^{15} \bmod 37 = 26$

Обменный ключ участника В:  $K = K_A^{Y_B} \bmod \alpha = 23^{20} \bmod 37 = 26$

Значения обменного ключа для А и В совпадают.

**Обменный ключ:  $K = 26$**

## **2.4 Лабораторная работа № 4 ( 2 часа).**

**Тема:** «Асимметричные криптосистемы (RSA, El Gamal, Рабина)»

**2.4.1 Цель работы:** Исследование формирования асимметричных криптосистем.

**2.4.2 Задачи работы:**

1. Исследование формирования асимметричных криптосистем.
2. Исследование формирования асимметричных криптосистем RSA.
3. Формирование асимметричных криптосистем Рабина.
4. Формирование асимметричных криптосистем El Gamal.

**2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Электронно-вычислительная машина.

**2.4.4 Описание (ход) работы:**

1. Основные принципы асимметричной криптографии Долгое время шифрование как способ преобразования сообщения в форму, недоступную для восприятия неавторизованным пользователям, существовало только в форме симметричной криптографии, когда и отправитель, и получатель должны знать секретный ключ, используемый для шифрации/дешифрации сообщений. Симметричное шифрование имеет недостатки, которые ограничивают возможности его применения в ряде конкретных случаев. В частности, зачастую невозможно организовать секретный канал для обмена ключами шифрования между участниками взаимодействия. Еще одним недостатком симметричных шифров является необходимость хранения большого количества ключей. Для того чтобы в вычислительной сети могли конфиденциально попарно взаимодействовать  $N$  участников, необходимо наличие в системе  $N(N-1)/2$  ключей. Эти недостатки можно устранить, используя алгоритмы асимметричного шифрования. Например, для асимметричной системы достаточно иметь  $2 \cdot N$  пар открытый/закрытый ключ, чтобы можно было организовать секретный канал между каждой парой участников. Асимметричная система шифрования работает по схеме, представленной на рис. 1. Отличительной особенностью асимметричных алгоритмов является наличие пары ключей шифрования: открытого (публичного)  $k_{от}$ , который передается второй стороне по незащищенному каналу связи и поэтому может быть известен криптоаналитику, а также закрытого (частного)  $k_{зак}$ , который известен лишь одному человеку (получателю сообщения) и держится в секрете [1]. Пара ключей обладает тем свойством, что сообщение, зашифрованное на одном из ключей, может быть расшифровано только на другом ключе. Фактически это означает, что секретным каналом передачи информации на схеме рис. 1 является направление “А-В”, поскольку сообщение, зашифрованное на открытом ключе отправителем, может дешифровать своим закрытым ключом только получатель. Если необходимо организовать двунаправленный безопасный обмен сообщениями, необходимо использовать две пары ключей и пользователь В должен шифровать информацию с использованием открытого ключа пользователя А.

Асимметричная система решает указанные выше проблемы симметричного шифрования – здесь нет необходимости передавать секретный ключ на противоположную сторону, публичный ключ можно передавать по открытому каналу связи. Единственное требование к каналу распространению открытого ключа – он должен быть аутентичным, т.е. всякий, получивший этот ключ, должен иметь возможность убедиться в его принадлежности лицу, заявленному как владелец соответствующей ключевой пары. На практике это требование реализуется путем сертификации открытого ключа – третья сторона, которой доверяют обе стороны взаимодействия, заверяет открытый ключ и выдает на него сертификат, подписанный электронной цифровой подписью. Открытый ключ в этом случае распространяется вместе с сертификатом, что дает возможность всегда удостовериться в его принадлежности. К асимметричным относятся такие алгоритмы шифрования как RSA, El Gamal, Рабина, Мессии-Омуры.

## 2. Исследование формирования асимметричных криптосистем RSA.

### Определение открытого « $e$ » и секретного « $d$ » ключей

Выбор двух взаимно простых больших чисел  $p$  и  $q$

Определение их произведения:  $n = p * q$

Определение функции Эйлера:  $\varphi(n) = (p-1)(q-1)$

Выбор открытого ключа  $e$  с учётом условий:

$$1 < e \leq \varphi(n), \quad \text{НОД}(e, \varphi(n)) = 1$$

Определение секретного ключа  $d$ , удовлетворяющего условию

$$e * d \equiv 1 \pmod{\varphi(n)}, \quad \text{где } d < n$$

### 3.2. Алгоритм шифрования сообщения $M$ (действия отправителя)

Разбивает исходный текст сообщения на блоки  $M_1, M_2, \dots, M_n$

$$(M_i = 0, 1, 2, \dots, n)$$

Шифрует текст сообщения в виде последовательности блоков:

$$C_i = M_i^e \pmod{n}$$

Отправляет получателю криптограмму:  $C_1, C_2, \dots, C_n$

Получатель расшифровывает криптограмму с помощью секретного ключа  $d$  по формуле:  $M_i = C_i^d \pmod{n}$

Процедуру шифрования данных рассмотрим на следующем примере (для простоты и удобства расчётов в данном примере использованы числа малой разрядности):

Выбираем два простых числа  $p$  и  $q$ ,  $p = 3$ ,  $q = 11$ ;

Определяем их произведение (модуль)  $n = p * q = 33$ ;

Вычисляем значение функции Эйлера  $\varphi(n) = (p-1)(q-1)$

$$\varphi(n) = 2 * 10 = 20$$

Выбираем случайным образом открытый ключ с учётом выполнения условий  $1 < e \leq \varphi(n)$ ,  $\text{НОД}(e, \varphi(n)) = 1$ ,  $e = 7$ ;

Вычисляем значение секретного ключа  $d$ , удовлетворяющего условию

$$e * d \equiv 1 \pmod{\varphi(n)}, \quad 7 * d \equiv 1 \pmod{20}; \quad d = 3;$$

Отправляем получателю пару чисел ( $n = 33$ ,  $e = 7$ );

Представляем шифруемое сообщение  $M$  как последовательность целых чисел **312**.

Разбиваем исходное сообщение на блоки  $M_1 = 3$ ,  $M_2 = 1$ ,  $M_3 = 2$ ;

Шифруем текст сообщения, представленный в виде последовательности блоков:  $C_i = M_i^e \pmod{n}$

$$C_1 = 3^7 \pmod{33} = 2187 \pmod{33} = 9,$$

$$C_2 = 1^7 \pmod{33} = 1 \pmod{33} = 1,$$

$$C_3 = 2^7 \pmod{33} = 128 \pmod{33} = 29.$$

Отправляем криптограмму  $C_1 = 9$ ,  $C_2 = 1$ ,  $C_3 = 29$ .

Получатель расшифровывает криптограмму с помощью секретного ключа  $d$  по формуле:

$$M_i = C_i^d \pmod{n}$$

$$M_1 = 9^3 \pmod{33} = 729 \pmod{33} = 3,$$

$$M_2 = 1^3 \pmod{33} = 1 \pmod{33} = 1,$$

$$M_3 = 29^3 \pmod{33} = 24389 \pmod{33} = 2.$$

Полученная последовательность чисел **312** представляет собой исходное сообщение  $M$ .

3. Формирование асимметричных криптосистем Рабина.

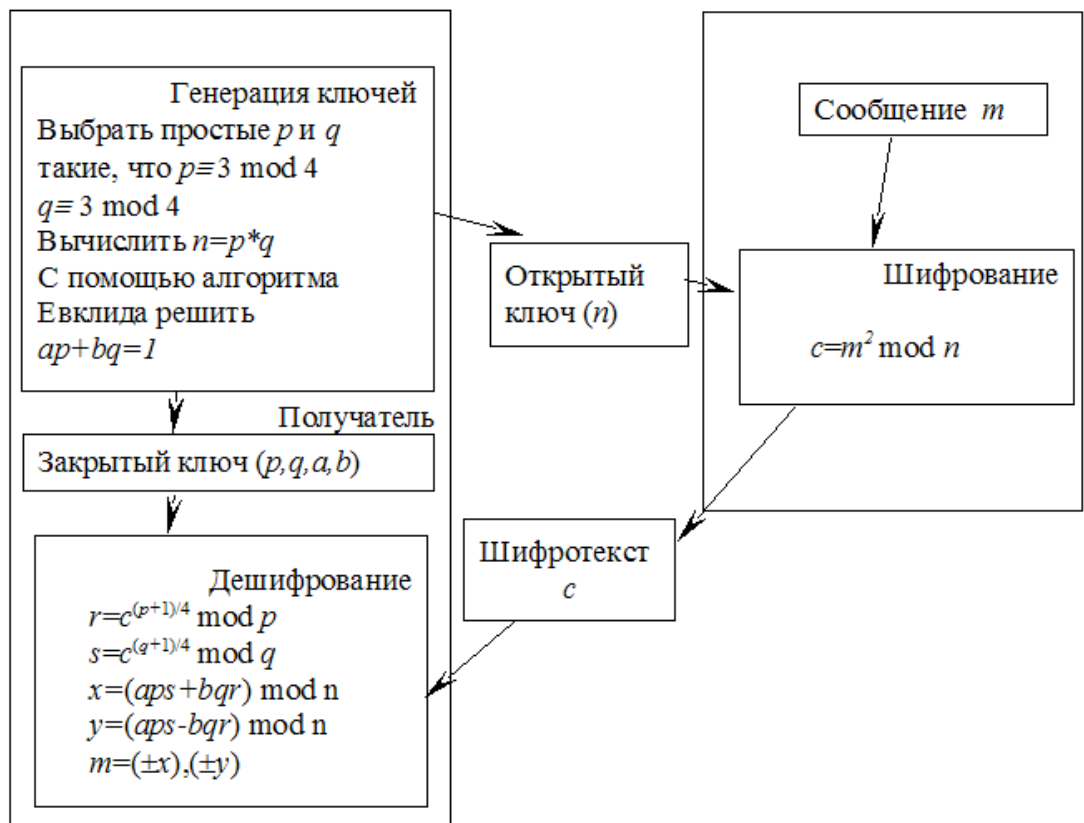


Рис.2.16. Схема шифрования алгоритма Рабина

#### 4. Формирование асимметричных криптосистем El Gamal.

##### Методика выполнения работы

Задание на выполнение лабораторной работы выдаётся преподавателем после прохождения студентами собеседования по основам криптографической защиты информации.

Порядок выполнения работы соответствует приведённой ниже криптосистеме шифрования данных по схеме Эль-Гамала.

##### Схема алгоритма шифрования данных Эль-Гамала

##### Определение открытого "y" и секретного "x" ключей

Выбор двух взаимно простых больших чисел  $p$  и  $q$ ,  $q < p$

Выбор значения секретного ключа  $x$ ,  $x < p$

. Определение значения открытого ключа  $y$  из выражения:

$$y = q^x \pmod{p}$$

##### Алгоритм шифрования сообщения $M$

Выбор случайного числа  $a$ , удовлетворяющего условию:

$$0 \leq k < p-1 \text{ и } \text{НОД}(k, p-1) = 1$$

Определение значения  $a$  из выражения:  $a = q^k \pmod{p}$

Определение значения  $b$  из выражения:  $b = y^k M \pmod{p}$

Криптограмма  $C$ , состоящая из  $a$  и  $b$ , отправляется получателю

Получатель расшифровывает криптограмму с помощью выражения:

$$M a^x = b \pmod{p}$$

Процедуру шифрования данных рассмотрим на следующее примере:

(для удобства расчётов в данном примере использованы числа малой разрядности):

Выбираем два взаимно простых числа  $p = 11$  и  $q = 2$ ;

Выбираем значение секретного ключа  $x$ , ( $x < p$ ),  $x = 8$ ;

Вычисляем значение открытого ключа  $y$  из выражения

$$y = q^x \pmod{p} = 2^8 \pmod{11} = 256 \pmod{11} = 3$$

Выбираем значение открытого сообщения  $M=5$ ;

Выбираем случайное число  $k=9$ ;  $\text{НОД}(9, 10) = 1$ ;

Определяем значение  $a$  из выражения:

$$a = q^k \pmod{p} = 2^9 \pmod{11} = 512 \pmod{11} = 6;$$

Определяем значение  $b$  из выражения:

$$b = y^k M \pmod{p} = 3^9 * 5 \pmod{11} = 98415 \pmod{11} = 9.$$

Таким образом, получаем зашифрованное сообщение как  $(a, b) = (6, 9)$  и отправляем получателю.

Получатель расшифровывает данный шифротекст, используя секретный ключ  $x$  и решая следующее сравнение:

$$M * a \equiv b \pmod{p} = 5 * 6^8 \equiv 9 \pmod{11} = 8398080 \equiv 9 \pmod{11}$$

Вычисленное значение сообщения  $M=5$  представляет собой заданное исходное сообщение.

## 2.5 Лабораторная работа №5-6 ( 4 часа).

**Тема:** «Программная реализация асимметричных криптосистем (RSA, El Gamal, Рабина)»

**2.5.1 Цель работы:** Освоить программную реализацию асимметричных криптосистем (RSA, El Gamal, Рабина)

### 2.5.2 Задачи работы:

1. Программная реализация асимметричного алгоритма шифрования данных RSA.

### 2.5.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Электронно-вычислительная машина.

### 2.5.4 Описание (ход) работы:

На данный момент асимметричное шифрование на основе открытого ключа RSA (расшифровывается, как Rivest, Shamir and Aldeman - создатели алгоритма) использует большинство продуктов на рынке информационной безопасности.

Его криптостойкость основывается на сложности разложения на множители больших чисел, а именно - на исключительной трудности задачи определить секретный ключ на основании открытого, так как для этого потребуется решить задачу о существовании делителей целого числа. Наиболее криптостойкие системы используют 1024-битовые и большие числа.

Рассмотрим алгоритм RSA с практической точки зрения.

Для начала необходимо сгенерировать открытый и секретные ключи:

- Возьмем два больших простых числа  $p$  and  $q$ .
- Определим  $n$ , как результат умножения  $p$  on  $q$  ( $n = p * q$ ).
- Выберем случайное число, которое назовем  $d$ . Это число должно быть взаимно простым (не иметь ни одного общего делителя, кроме 1) с результатом умножения  $(p-1)*(q-1)$ .
- Определим такое число  $e$ , для которого является истинным следующее соотношение  $(e*d) \pmod{((p-1)*(q-1))} = 1$ .
- Назовем открытым ключом числа  $e$  и  $n$ , а секретным -  $d$  и  $n$ .
- Для того, чтобы зашифровать данные по открытому ключу  $\{e, n\}$ , необходимо следующее:

- разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа  $M(i)=0,1,2,\dots, n-1$  (т.е. только до  $n-1$ ).
- зашифровать текст, рассматриваемый как последовательность чисел  $M(i)$  по формуле  $C(i)=(M(i)^e)\bmod n$ .

Чтобы расшифровать эти данные, используя секретный ключ  $\{d,n\}$ , необходимо выполнить следующие вычисления:  $M(i) = (C(i)^d) \bmod n$ . В результате будет получено множество чисел  $M(i)$ , которые представляют собой исходный текст.

Следующий пример наглядно демонстрирует алгоритм шифрования RSA:

Зашифруем и расшифруем сообщение "CAB" по алгоритму RSA. Для простоты возьмем небольшие числа - это сократит наши расчеты.

- Выберем  $p=3$  and  $q=11$ .
- Определим  $n= 3*11=33$ .
- Найдем  $(p-1)*(q-1)=20$ . Следовательно,  $d$  будет равно, например, 3: ( $d=3$ ).
- Выберем число  $e$  по следующей формуле:  $(e*3) \bmod 20=1$ . Значит  $e$  будет равно, например, 7: ( $e=7$ ).
- Представим шифруемое сообщение как последовательность чисел в диапазоне от 0 до 32 (незабудьте, что кончается на  $n-1$ ). Буква  $A=1$ ,  $B=2$ ,  $C=3$ .

Теперь зашифруем сообщение, используя открытый ключ  $\{7,33\}$

$$C1 = (3^7) \bmod 33 = 2187 \bmod 33 = 9;$$

$$C2 = (1^7) \bmod 33 = 1 \bmod 33 = 1;$$

$$C3 = (2^7) \bmod 33 = 128 \bmod 33 = 29;$$

Теперь расшифруем данные, используя закрытый ключ  $\{3,33\}$ .

$$M1=(9^3)\bmod33=729\bmod33=3(C);$$

$$M2=(1^3)\bmod33=1\bmod33=1(A);$$

$$M3=(29^3)\bmod33=24389\bmod33=2(B);$$

Данные расшифрованы!

Задание.

1. Создать программную реализацию алгоритма RSA.
2. Исследовать зависимость времени шифрования и дешифрования файлов от размера файла и длины ключа, результаты представить в графическом или табличном виде.
3. Сформировать и представить преподавателю отчет по результатам выполнения лабораторной работы.

## 2.6 Лабораторная работа № 7-8 ( 4 часа).

**Тема:** «Исследование тестов на простоту и алгоритмы факторизации»

**2.6.1 Цель работы:** Провести исследование тестов на простоту и алгоритмы факторизации

**2.6.2 Задачи работы:** Провести исследование тестов на простоту и алгоритмы факторизации

**2.6.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Электронно-вычислительная машина.

**2.6.4 Описание (ход) работы:**

1. Рабочее задание

1.1. Написать программу на языке "UBASIC", реализующую алгоритм факторизации модуля шифрования  $n$  из примера, приведенного ниже.

1.2. Найти разложение  $n=p*q$ .

- 1.3. Восстановить секретный ключ  $d$ .  
1.4. Разложить на множители числа  $p-1$  и  $q-1$ , убедиться, что в их разложении имеются большие делители.

## 2. Выполнение задания

2.1. Предположим, что Вы имеете открытый ключ схемы RSA  $(e, n)$ , где  $n=525169521992627614583344195951527749$ , а в качестве ключа зашифрования  $e$  используете числа по вариантам, определенным преподавателем:

- 1 вариант -  $e=131$
- 2 вариант -  $e=133$
- 3 вариант -  $e=137$
- 4 вариант -  $e=139$
- 5 вариант -  $e=149$
- 6 вариант -  $e=151$
- 7 вариант -  $e=157$
- 8 вариант -  $e=161$

2.2. Составьте программу на языке "UBASIC", реализующую алгоритм факторизации модуля шифрования  $n$  по формуле (3). При реализации алгоритма используйте встроенные функции языка:

- |                   |   |
|-------------------|---|
| $Y=MODINV(X,P)$   | - вычисляет обратное к $X$ по модулю $P$ ;              |
| $Y=MODPOW(A,X,P)$ | - вычисляет $Y=A^{**}X \bmod P$ ;                       |
| $Y=GCD(A,B)$      | - вычисляет наибольший общий делитель чисел $A$ и $B$ . |

- 2.3. По разложению  $n=pq$  восстановите секретный ключ расшифрования  $d$ .  
2.4. С помощью программы ECMX.UB разложите на множители числа  $p-1$  и  $q-1$ .  
2.5. Напишите отчет.

## 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

### 3.1 Практическое занятие № 1 ( 2 часа).

**Тема:** «Поточные системы шифрования (РСЛОС, RC4, Рона)»

#### 3.1.1 Задание для работы:

- 1. Система РСЛОС.
- 2. Система Рона.
- 3. Система RC4.

#### 3.1.2 Краткое описание проводимого занятия:

Пусть известно, что исходное сообщение представляло собой двоично-десятичное число, то есть число, каждая тетрада (четыре бита) которого получена при переводе десятичной цифры  $0...9$  в двоичный вид. Перехвачено 24 бита зашифрованного сообщения  $Y$ , то есть шесть тетрад  $Y_1, Y_2, Y_3, Y_4, Y_5, Y_6$ , а именно значение 1100 1101 1110 1111 0000 0001. Известно, что ключ шифрования состоял из четырех бит, которые тоже представляют собой однозначное десятичное число, то есть одно и то же значение  $0 \leq K \leq 9$  использовалось для шифрования каждых четырех бит исходного сообщения. Таким образом, шифрование числа  $X_1, X_2, X_3, X_4, X_5, X_6$  ключом  $K$  можно представить в виде системы уравнений:



$$X_1 \oplus K = 1100 \quad X_2 \oplus K = 1101 \quad X_3 \oplus K = 1110 \quad X_4 \oplus K = 1111 \quad X_5 \oplus K = 0000 \quad X_6 \oplus K = 0001$$

Исходя из условия, что  $X_i$  принимает десятичные значения от 0 до 9, для поиска неизвестного  $K$  определим все возможные значения  $X_1'$ ; и  $K$ , сумма которых по модулю 2 приводит к результату 1100:

$$K = 0000 \quad 0001 \quad 0010 \quad 0011 \quad 0100 \quad 0101 \quad 0110 \quad 0111 \quad 1000 \quad 1001$$

$$Y_1 = 1100 \quad 1100 \quad 1100 \quad 1100 \quad 1100 \quad 1100 \quad 1100 \quad 1100 \quad 1100 \quad 1100$$

$$X_1' = 1100 \quad 1101 \quad 1110 \quad 1111 \quad 1000 \quad 1001 \quad 1010 \quad 1011 \quad 0100 \quad 0101$$

Так как исходное значение состояло из цифр от 0 до 9, то можно исключить из рассмотрения значения ключа 0000, 0001, 0010, 0011, 0110, 0111, так при сложении с ними получаются значения большие 9 в десятичном эквиваленте. Такие значения не могли присутствовать в открытом тексте. Таким образом, первый этап анализа уже позволил сократить количество возможных ключей с десяти до четырех.

Для дальнейшего поиска неизвестного  $K$  определим все возможные значения  $X_2'$ ; и оставшихся вариантов ключа, сумма которых по модулю 2 приводит к результату  $Y_2 = 1101$ :

$$K = 0100 \quad 0101 \quad 1000 \quad 1001$$

$$Y_2 = 1101 \quad 1101 \quad 1101 \quad 1101$$

$$X_2' = 1001 \quad 1000 \quad 0101 \quad 0100$$

Видно, что этот этап не позволил отбросить ни одного из оставшихся вариантов ключа. Попробуем это сделать, используя  $Y_3 = 1110$ :

$$K = 0100 \quad 0101 \quad 1000 \quad 1001$$

$$Y_3 = 1110 \quad 1110 \quad 1110 \quad 1110$$

$$X_2' = 1010 \quad 1011 \quad 0110 \quad 0111$$

После проведения этого этапа становится ясно, что ключом не могли быть значения 0100 и 0101. Остается два возможных значения ключа:  $1000_{(2)} = 8_{(10)}$  и  $1001_{(2)} = 9_{(10)}$ .

Дальнейший анализ по данной методике в данном случае, к сожалению, не позволит однозначно указать, какой же из двух полученных вариантов ключа использовался при шифровании. Однако можно считать успехом уже то, что пространство возможных ключей снизилось с десяти до двух. Остается попробовать каждый из двух найденных ключей для дешифровки сообщений и проанализировать смысл полученных вскрытых текстов.

В реальных случаях, когда исходное сообщение составлено не только из одних цифр, но и из других символов, использование статистического анализа позволяет быстро и точно восстановить ключ и исходные сообщения при короткой длине ключа, закрывающего поток секретных данных.

### 3.1.3 Результаты и выводы:

На практике освоены поточные системы шифрования (РСЛОС, RC4, Рона)

## 3.2 Практическое занятие № 2 ( 2 часа).

**Тема:** «Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)»

### 3.2.1 Задание для работы:

1. Программная реализация поточных систем шифрования.

### 3.2.2 Краткое описание проводимого занятия:

В блочном шифре из двух одинаковых блоков открытого текста получаются одинаковые блоки шифрованного текста. Избежать этого позволяют поточные шифры, в которых шифрующее преобразование “элемента” открытого текста меняется от одного

элемента к другому. Так в DES в режиме сцепления блоков фактически происходит преобразование блочного шифра в поточный, что облегчает обнаружение искажений блоков и затрудняет попытки имитации и подмены. Специалисты, однако, используют термин поточный шифр только в том случае, когда “элементы” открытого текста очень малы (одна буква или один бит). Обычно аппаратные реализации поточных шифров быстрее и проще, чем блочных. Поточковые шифры пригодны для шифрования непрерывных потоков данных, например, в сетях передачи данных. Самые популярные сейчас поточные шифры можно назвать двоичными аддитивными. В таких шифрах  $k$ -битовый секретный ключ  $Z$  используется только для управления генератором ключевого потока, порождающего двоичную последовательность  $011, \dots, N - z z$ , называемую ключевым потоком, где  $N \gg k$ . Шифр текст образуется путем сложения по модулю 2 битов открытого текста и битов ключевого потока: Шифрование  $i$   $Y = X \oplus z, i = 1, 0, \dots, N - 1$  Дешифрование  $i$   $X = Y \oplus z, i = 1, 0, \dots, N - 1$  Шифрование и дешифрование выполняются одинаковыми устройствами. Аддитивный поточный шифр похож на двоичный шифр с ключом однократного применения. Если  $Z_i = z, i = 1, 0, \dots, k, k = N$ , т. е. секретный ключ используется как ключевая последовательность, то аддитивный поточный шифр есть шифр с ключом однократного применения. В практических поточных шифрах длина  $N$  шифртекста много больше длины  $k$  секретного ключа, а ключевая последовательность является псевдослучайной и имеет некоторый период. Стойкость системы целиком зависит от внутренней структуры генератора ключевой последовательности. Если генератор выдает последовательность с небольшим периодом, то стойкость системы будет невелика. Если бы генератор выдавал бесконечную последовательность битов, в которой каждый бит порождался независимо и с вероятностью  $1/2$  принимал значения 0 или 1, то мы получили бы совершенно стойкий шифр.

### 3.2.3 Результаты и выводы:

На практике освоить программную реализацию поточных систем шифрования (РСЛОС, RC4, Рона).

## 3.3 Практическое занятие № 3-4 ( 4 часа).

**Тема:** «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)»

### 3.3.1 Задание для работы:

1. Схема распределения ключей, основанных на эллиптических кривых.

### 3.3.2 Краткое описание проводимого занятия:

Открытый элемент  $P$  задан в табл. 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы  $i$ , вычислить ключ обмена для участника с номером  $35 - i$  по алгоритму Диффи – Хэллмана.

Пример выполнения Задания3. Алгоритм Диффи-Хеллмана

Открытый элемент  $P$  задан в таблице 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы  $i$ , вычислить ключ обмена для участника с номером  $35 - i$  по алгоритму Диффи-Хэллмана.

Вариант № 15, группа 2091 (№ 1)

Номер в списке группы                       $i = 15$   
Номер группы                                       $k = 1$   
 **$P = 37$**

Открытый элемент  $P = 37$ . Найти примитивный элемент поля. Секретный ключ каждого участника  $i = 15$ , вычислить ключ обмена для участника с номером  $35 - 15 = 20$ .

$GF(37) = \langle 0, 1, 2, 3, \dots, 35, 36, 37 \rangle$

Найдем примитивный элемент поля  $GF(37)$ .

Требуется найти такое число, принадлежащее интервалу  $[2, 37]$ , которое при возведении в 37-ю степень по модулю 37 будет давать в результате единицу. Если же единица будет получена раньше, чем при возведении в 36-ю степень, результаты возведения в степень начнут повторяться, и через выбранный элемент не удастся представить все элементы поля. Исходя из таких соображений, получаем несложный алгоритм нахождения примитивных элементов поля  $GF(37)$ .

Алгоритм нахождения примитивных элементов поля  $GF(37)$ .

```
int _tmain(int argc, _TCHAR* argv[])
{
    long int mem, i, j, num, deg, modul, res;
    unsigned char mas[38];
    deg = 0; modul = 37; mem = 2;
    while (mem < 36)
    {
        num = mem; deg = 0;
        while (deg != 36)
        {
            res = 1; deg = 0;
            for (i = 0; i < 36; i++) mas[i] = 0;
            do
            {
                res = res*num; res = res % modul;
                deg++; mas[deg] = res;
            }
            while(res != 1);
            num++;
        }
        std::cout << num-1 << std::endl;
        for (i = 0; i < 36; i++)
        {
            for (j = 0; j < 36; j++)
            {
                if ((i != j) && (mas[j] == mas[i]))
                    std::cout << "Wrong" << std::endl;
            }
            mem = num;
        }
    }
}
```

В результате выполнения программы получим следующие значения:

2, 5, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35

Выберем значение  $\alpha = 2$ .

Секретный ключ участника A:  $X_A = 15$ .

Секретный ключ участника В:  $Y_B = 35 - 15 = 20$ .

Открытый ключ участника А:  $K_A = \alpha^{X_A} = \alpha^{15} = 2^{15} \bmod 37 = 23$

Открытый ключ участника В:  $K_B = \alpha^{Y_B} = \alpha^{20} = 2^{20} \bmod 37 = 33$

$$K_A = 23$$

$$K_B = 33$$

Обменный ключ участника А:  $K = K_B^{X_A} \bmod \alpha = 33^{15} \bmod 37 = 26$

Обменный ключ участника В:  $K = K_A^{Y_B} \bmod \alpha = 23^{20} \bmod 37 = 26$

Значения обменного ключа для А и В совпадают.

**Обменный ключ:  $K = 26$**

### 3.3.3 Результаты и выводы:

На практике решить задачу распределения ключей на основе алгоритма Диффи — Хеллмана.

### 3.4 Практическое занятие № 5-8 ( 8 часов).

**Тема:** «Асимметричные криптосистемы (RSA, El Gamal, Рабина)»

#### 3.4.1 Задание для работы:

1. Формирование асимметричных криптосистем.
2. Формирование асимметричных криптосистем RSA
3. Формирование асимметричных криптосистем Рабина .
4. Формирование асимметричных криптосистем El Gamal.

#### 3.4.2 Краткое описание проводимого занятия:

1. Выбираем простые числа (небольшие, чтобы упростить вычисления):  $p=3, q=11$
2. Вычисляем модуль  $n=p*q=3*11=33$
3. Вычисляем функцию Эйлера от модуля  $n$ :  $\varphi(N)=(p-1)*(q-1)=2*10=20$
4. Выбираем открытую экспоненту  $e=7$
5. Определяем закрытую экспоненту  $d$ :  $d*e=1 \pmod{\varphi(N)}$   
Будем шифровать сообщение RSA, пусть букве А соответствует цифра 1, В - 2, С - 3 и т.д (Подобное соответствие вносим для простоты), тогда :  
 $R=18; S=19; A=1$ ; Открытый ключ :  $(e,n)=(7,33)$   
 $C1=(187) \bmod 33=6$   
 $C2=(197) \bmod 33=13$   
 $C3=(17) \bmod 33=1$

$C(\text{"RSA"})=6131$

### **3.4.3 Результаты и выводы:**

На практике рассмотреть пример шифрование с помощью алгоритма RSA