

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению практических работ  
по дисциплине «Методы и средства криптографической защиты информации»  
для студентов направления подготовки  
10.03.01 Информационная безопасность  
Направленность (профиль)  
«Организация и технологии защиты информации (по отрасли или в сфере профессиональной  
деятельности)»

## Содержание

- Практическое занятие 1. Шифрование алгоритмом S-DES
- Практическое занятие 2. Расшифрование алгоритмом S-DES
- Практическое занятие 3. Блочный шифр. Ключевая система
- Практическое занятие 4. Шифрование алгоритмом S-AES
- Практическое занятие 5. Расшифрованное алгоритмом S-AES
- Практическое занятие 6. Шифрование и расшифрование алгоритмом S-Магма
- Практическое занятие 7. Шифрование на основе генератора Фибоначчи
- Практическое занятие 8. Расшифрование на основе генератора Фибоначчи
- Практическое занятие 9. Шифрование на основе генераторов много тактовых кодовых фильтров
- Практическое занятие 10. Арифметика целых чисел
- Практическое занятие 11. Модульная арифметика
- Практическое занятие 12. Алгебраические структуры
- Практическое занятие 13. Расширение полей Галуа  $GF(2^m)$
- Практическое занятие 14. Математические модели расширенных полей Галуа
- Практическое занятие 15. Алгоритм шифрования без передачи ключа
- Практическое занятие 16. Алгоритм шифрования Эль-Гамала
- Практическое занятие 17. Построение электронной цифровой подписи RSA
- Практическое занятие 18. Построение электронной цифровой подписи Эль-Гамала
- Практическое занятие 19. Ключевой обмен на основе алгоритма Диффи-Хеллмана
- Практическое занятие 20. Модифицированный метод ключевого обмена на основе алгоритма Диффи-Хеллмана

Практическое занятие 21. Схема разделения секрета Шамира

Практическое занятие 22. Протокол аутентификации Шнорра

Практическое занятие 23. Протокол аутентификации для систем электронных платежей.

Практическое занятие 24. Отечественная криптографическая хеш-функция.

Практическое занятие 25. Скрытый канал Эль-Гамала

(есть лишние)

Указания по технике безопасности .....

Список рекомендуемой литературы .....

## ВВЕДЕНИЕ

### 1. Цель и задачи освоения дисциплины

Дисциплина «Методы и средства криптографической защиты информации» относится к обязательной части образовательной программы направления подготовки 10.03.01 Информационная безопасность, направленность (профиль) «Организация и технологии защита информации (по отрасли или в сфере профессиональной деятельности)».

**Цель освоения дисциплины:** формирование у будущего бакалавра 10.03.01 «Информационная безопасность фундаментальных знаний основных положений теории криптографической защиты информации, принципов построения симметричных и асимметричных шифров, схем цифровой подписи и функции хеширования, инфраструктуры систем управления ключами, оценки криптостойкости, имитостойкости и помехоустойчивости шифров,, криптографических протоколов.

**Задачи освоения дисциплины:**

- изучить математические основы криптографических методов защиты информации;
- изучить основные алгоритмы симметричного и асимметричного шифрования данных;
- изучить основы организации структуры криптосистем.

### 2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

| Код, формулировка компетенции                                                                                                                                                                                                                                                                                             | Код, формулировка индикатора                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ОПК-6</b> Способен при решении профессиональных задач организовывать защиту информации ограниченного доступа в соответствии с нормативными правовыми актами, нормативными и методическими документами Федеральной службы безопасности Российской Федерации, Федеральной службы по техническому и экспортному контролю; | ИД-2 <sub>ОПК-6</sub> Эффективно применяет знания теории и методики курса при раскрытии компьютерных преступлений, при работе с конфиденциальной информацией и государственной тайной, при работе с организационно-правовым обеспечением, при работе с ЭП, при защите авторского и международного права, при проведении экспертизы преступлений в сфере компьютерной информации; применять нормативно-правовые акты при защите информации; оценивать состояние организационной защиты информации на объекте. | -осуществляет организацию защиты информации ограниченного доступа в соответствии с нормативными и методическими документами Федеральной службы безопасности Российской Федерации, Федеральной службы по техническому и экспортному контролю;<br>-выбирает мероприятия защиты информации ограниченного доступа в соответствии с нормативными и методическими документами Федеральной службы безопасности Российской Федерации, Федеральной службы по техническому и экспортному контролю; |

|                                                                                                                                                    |                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                    |                                                                                                                                                                             | <p>-участвует в организации защиты информации ограниченного доступа в соответствии с нормативными и методическими документами Федеральной службы безопасности Российской Федерации, Федеральной службы по техническому и экспортному контролю;</p>                                                                                                                                                               |
| <p><b>ОПК-9</b> Способен применять средства криптографической и технической защиты информации для решения задач профессиональной деятельности;</p> | <p>ИД-1<sub>ОПК-9</sub> Понимает базовые принципы построения средств криптографической и технической защиты информации для решения задач профессиональной деятельности.</p> | <p>- оперирует знаниями средств криптографической и технической защиты информации по техническим каналам;<br/>         - оценивает средства криптографической и технической защиты информации по техническим каналам;<br/>         - анализирует средства криптографической и технической защиты информации по техническим каналам;</p>                                                                          |
|                                                                                                                                                    | <p>ИД-2<sub>ОПК-9</sub> Использует средства криптографической и технической защиты информации для решения задач профессиональной деятельности.</p>                          | <p>- осуществляет выбор средств криптографической и технической защиты информации для решения задач профессиональной деятельности;<br/>         - выделяет средства криптографической и технической защиты информации для решения задач профессиональной деятельности;<br/>         - использует средства криптографической и технической защиты информации для решения задач профессиональной деятельности.</p> |
|                                                                                                                                                    | <p>ИД-3<sub>ОПК-9</sub> Выбирает оптимальный вариант использования навыков и методик применения средств криптографической и технической защиты информации</p>               | <p>- демонстрирует знания устройства и принципов работы криптографических средств защиты информации, применяемых в компьютерных системах и сетях;</p>                                                                                                                                                                                                                                                            |

|  |                                                 |                                                                                                                        |
|--|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
|  | для решения задач профессиональной деятельности | -устанавливает, настраивает и интегрирует криптографические средства защиты информации в компьютерных системах и сетях |
|--|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|

## ПЛАНЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

### Практическое занятие № 1

**Тема:** Шифрование алгоритмом S-DES

**Цель и содержание:**

1. Углубить знания, полученные на лекциях основам шифрования.
2. Изучить основные характеристики моделей шифров.

**Формируемые компетенции:** ОПК-6, ОПК-9.

### Теоретический материал

Оценка стойкости модели шифров - алгебраическая и вероятностная.

Пусть  $X$ ,  $K$ ,  $Y$  — конечные множества возможных открытых текстов, ключей и шифрованных текстов соответственно;  $E_k: X \rightarrow Y$  — правило зашифрования на ключе  $k$  принадлежит  $K$ . Множество  $\{E_k: k \text{ принадлежит } K\}$  обозначим через  $E$ , а множество  $\{E_k(x): x \text{ принадлежит } X\}$  — через  $E_k(X)$ . Пусть  $D_k: E_k(X) \rightarrow X$  — правило расшифрования на ключе  $k$  принадлежит  $K$ , и  $D$  — это множество  $\{D_k: k \text{ принадлежит } K\}$ . Здесь и далее мы будем предполагать, что если  $k$  принадлежит  $K$  представляется в виде  $k = (k_z, k_p)$ , где  $k_z$  — ключ зашифрования, а  $k_p$  — ключ расшифрования (причем  $k_z \neq k_p$ ), то  $E_k$  понимается как функция  $E_{k_z}$ , а  $D_k$  — как функция  $D_{k_p}$ .

**Определение.** Шифром (шифрсистемой) назовем совокупность  $\varepsilon_A = (X, K, Y, E, D)$  введенных множеств, для которых выполняются следующие свойства:

1. для любых  $x$  принадлежит  $X$  и  $k$  принадлежит  $K$  выполняется равенство  $D_k(E_k(x)) = x$ ;
2.  $Y = \bigcup_k E_k(X)$ .

Неформально, **шифр** — это совокупность множеств возможных открытых текстов (то, что шифруется), возможных ключей (то, с помощью чего шифруется), возможных шифртекстов (то, во что шифруется), правил зашифрования и правил расшифрования.

Отметим, что условие 1) отвечает требованию однозначности расшифрования. Условие 2) означает, что любой элемент  $y$  принадлежит  $Y$  может быть представлен в виде  $E_k(x)$  для подходящих элементов  $x$  принадлежит  $X$  и  $k$  принадлежит  $K$ . Отметим также, что в общем случае утверждение «для любых  $k$  принадлежит  $K$  и  $y$  принадлежит  $E_k(X)$  выполняется равенство  $E_k(D_k(y)) = y$ » является неверным. Иначе говоря, функция  $D_k$  постоянна, а  $E_k$  может меняться в процессе шифрования.

Легко проверить, что из условия 1) следует свойство инъекции функции  $E_k$ . Другими словами, если  $x_1, x_2$  принадлежит  $X$ , причем  $x_1 \neq x_2$ , то при любом  $k$  принадлежит  $K$  выполняется неравенство  $E_k(x_1) \neq E_k(x_2)$ .

По сути дела определение, приведенное выше, вводит математическую модель, отражающую основные свойства реальных шифров. В силу этого мы будем отождествлять реальный шифр с его моделью  $\varepsilon_A$ , которую будем называть *алгебраической моделью шифра*.

Введем теперь вероятностную модель шифра. Следуя К. Шеннону, определим априорные распределения вероятностей  $P(X)$ ,  $P(K)$  на множествах  $P$  и  $K$  соответственно. Тем самым для любого  $x$  принадлежит  $X$  определена вероятность  $p_x(x)$  принадлежит  $P(x)$  и для любого  $k$  принадлежит  $K$  — вероятность  $p_k(k)$  принадлежит  $P(K)$ , причем выполняются равенства:

$$\sum_x \text{принадлежит } X p_x(x) = 1, \quad \sum_k \text{принадлежит } K p_k(k) = 1.$$

В тех случаях, когда нам требуется знание распределений  $P(X)$ ,  $P(K)$ , мы будем пользоваться *вероятностной моделью*  $\varepsilon_B$ , состоящей из пяти множеств, связанных условиями 1) и 2) определения, и двух вероятностных распределений:

$$\varepsilon_B = (X, K, Y, E, D, P(X), P(K)).$$

Отметим, что вероятностные характеристики шифров используются лишь в криптоанализе, т. е. при вскрытии (или взломе) шифров.

## Типы криптостойких систем шифрования

### АБСОЛЮТНО СТОЙКИЕ СИСТЕМЫ

Доказательство существования абсолютно стойких алгоритмов шифрования было выполнено Клодом Шенноном и опубликовано в работе «Теория связи в секретных системах». Там же определены требования к такого рода системам:

- ключ генерируется для каждого сообщения (каждый ключ используется один раз)
- ключ статистически надёжен (то есть вероятности появления каждого из возможных символов равны, символы в ключевой последовательности независимы и случайны)
- длина ключа равна или больше длины сообщения
- исходный (открытый) текст обладает некоторой избыточностью (является критерием оценки правильности расшифровки)

Стойкость этих систем **не зависит** от того, какими вычислительными возможностями обладает криптоаналитик. Практическое применение систем, удовлетворяющих требованиям абсолютной стойкости, ограничено соображениями стоимости и удобства пользования.

Некоторыми аналитиками утверждается, что Шифр Вернама является одновременно абсолютно криптографически стойким и к тому же единственным шифром, который удовлетворяет этому условию.

### ДОСТАТОЧНО СТОЙКИЕ СИСТЕМЫ

В основном применяются **практически стойкие** или **вычислительно стойкие** системы. Стойкость этих систем **зависит** от того, какими вычислительными возможностями обладает криптоаналитик. Практическая стойкость таких систем базируется на теории сложности и оценивается исключительно на какой-то определенный момент времени и последовательно с двух позиций:

- вычислительная сложность полного перебора

- известные на данный момент слабости (уязвимости) и их влияние на вычислительную сложность.

В каждом конкретном случае могут существовать дополнительные критерии оценки стойкости.

Сложность алгоритма позволяет оценить, насколько быстро растет его трудоёмкость с увеличением объема входных данных. Под трудоёмкостью понимается количество элементарных операций, которые необходимо выполнить для решения задачи с помощью данного алгоритма. Обычно оценка сложности алгоритма представляется в виде  $O(f(N))$ , где  $O$  – функция сложности, а  $N$  – число обрабатываемых наблюдений или примеров. Наименее затратными являются алгоритмы, для которых функция сложности имеет вид  $f(N) = C$  и  $f(N) = C \cdot N$ , где  $C$  – константа. В первом случае вычислительные затраты не зависят от количества обрабатываемых данных, а во втором – линейно возрастают. Самыми затратными являются алгоритмы, сложность которых имеет степенную и факториальную зависимости от числа обрабатываемых наблюдений.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 2.

**Тема:** Расшифрование алгоритмом S-DES

**Цель и содержание:**

1. Углубить знания, по основам построения простейших шифров подстановки и перестановки.
2. Изучить основные характеристики алгоритма шифрования.

**Формируемые компетенции:** ОПК-6, ОПК-9.

**Теоретический материал:**

**Шифр подстановки** заменяет один символ другим. Если символы в исходном тексте — символы алфавита, мы заменяем одну букву другой. Например, мы можем заменить букву А буквой D, а букву Т — буквой Z. Если символы — цифры (от 0 до 9), мы можем заменить 3 на 7 и 2 на 6. Шифры подстановки могут быть разбиты на две категории: моноалфавитные или многоалфавитные шифры.

Шифр подстановки заменяет один символ другим.

Моноалфавитные шифры

Сначала обсудим шифры подстановки, называемые **моноалфавитными шифрами**. В такой подстановке буква (или символ) в исходном тексте всегда изменяется на одну и ту же самую букву (или символ) в зашифрованном тексте независимо от его позиции в тексте. Например, если алгоритм определяет, что буква А в исходном тексте меняется на букву D, то при этом каждая буква А изменяется на букву D. Другими словами, буквы в исходном тексте и зашифрованном тексте находятся в отношении один к одному.

В моноалфавитной подстановке отношения между буквой в исходном тексте и буквой в зашифрованном тексте — один к одному.

### Пример 2.1

Приведенный ниже пример показывает исходный текст и соответствующий ему зашифрованный текст. Мы используем строчные символы, чтобы показать исходный текст, и заглавные буквы (символы верхнего

регистра), чтобы получить зашифрованный текст. Шифр моноалфавитный, потому что оба l зашифрованы как O.

Исходный текст: hello    Зашифрованный текст: KHOOR

**Пример 2.2**

Приведенный ниже пример показывает исходный текст и соответствующий ему зашифрованный текст. Шифр не является моноалфавитным, потому что каждая буква l (эль) зашифрована различными символами. Первая буква l (эль) зашифрована как N; вторая — как Z.

Исходный текст: hello    Зашифрованный текст: ABNZF

*Аддитивный шифр*

Самый простой моноалфавитный шифр — **аддитивный шифр**, его иногда называют **шифром сдвига**, а иногда — **шифром Цезаря**, но термин аддитивный шифр лучше показывает его математический смысл. Предположим, что исходный текст состоит из маленьких букв (от a до z) и зашифрованный текст состоит из заглавных букв (от A до Z). Чтобы обеспечить применение математических операций к исходному и зашифрованному текстам, мы присвоим каждой букве числовое значение (для нижнего и верхнего регистра), как это показано на [рис. 2.8](#).

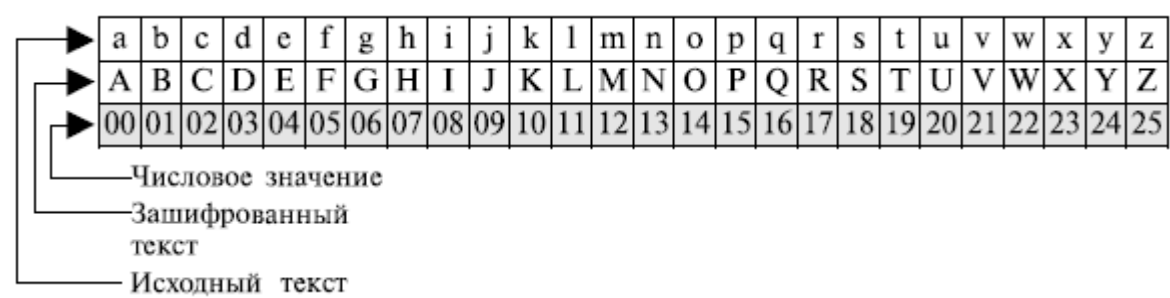


**Рис. 1** Представление букв исходного текста и зашифрованного текста в  $Z_{26}$

Каждому символу (нижний регистр или верхний регистр) сопоставлено целое число из  $Z_{26}$ . Ключ засекречивания между Алисой и Бобом — также целое число в  $Z_n$ . Алгоритм кодирования прибавляет ключ к символу исходного текста;

алгоритм дешифрования вычитает ключ из символа зашифрованного текста. Все операции проводятся в  $Z_{26}$ . показывает процесс шифрования и дешифрования.

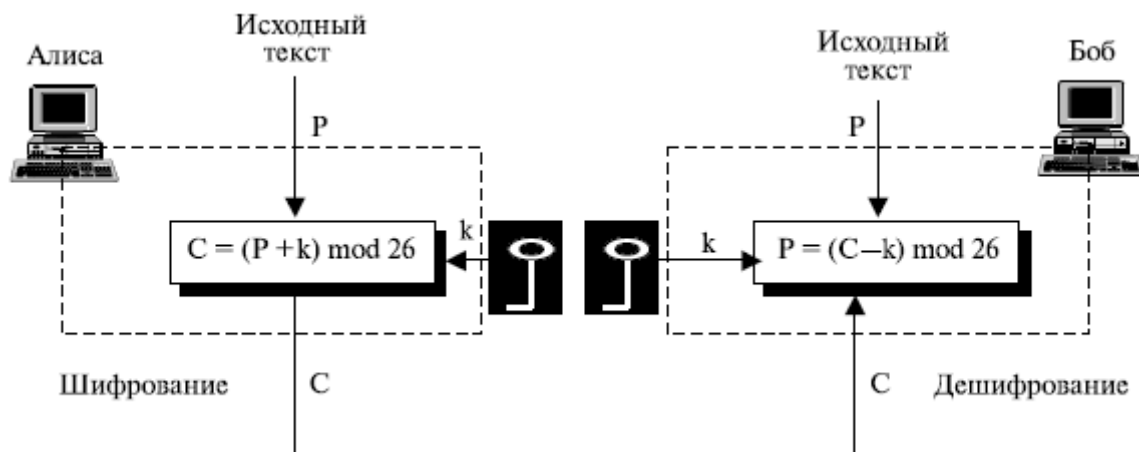


**Рис. 1.** Аддитивный шифр

Мы можем легко показать, что шифрование и дешифрование являются инверсными друг другу, потому что исходный текст, созданный Бобом ( $P_1$ ), тот же самый, что и тот, который передан Алисой ( $P$ ).

$$P_1 = (C - k) \bmod 26 = (P + k - k) \bmod 26 = P$$

Когда применяется аддитивный шифр, исходный текст, зашифрованный текст и ключ — целые числа в  $Z_{26}$ .

### Пример 2.3

Используйте аддитивный шифр с ключом = 15, чтобы зашифровать сообщение "hello".

### Решение

Мы применяем алгоритм кодирования к исходному тексту, буква за буквой:

|                  |   |    |            |                      |           |
|------------------|---|----|------------|----------------------|-----------|
| Исходный текст h | → | 07 | Шифрование | $(07 + 15) \bmod 26$ | ШифрТекст |
| 22               | → | W  |            |                      |           |
| Исходный текст e | → | 04 | Шифрование | $(04 + 15) \bmod 26$ | ШифрТекст |
| 19               | → | T  |            |                      |           |
| Исходный текст l | → | 11 | Шифрование | $(11 + 15) \bmod 26$ | ШифрТекст |
| 00               | → | A  |            |                      |           |

Исходный текст 1  $\rightarrow$  11      Шифрование  $(11 + 15) \bmod 26$  ШифрТекст  
00  $\rightarrow$  A

Исходный текст o  $\rightarrow$  14      Шифрование  $(14 + 15) \bmod 26$  ШифрТекст  
03  $\rightarrow$  D

Результат — "WTAAD". Обратите внимание, что шифр моноалфавитный, потому что два отображения одной и той же буквы исходного текста (l) зашифрованы как один и тот же символ (A).

### Пример 2.4

Используйте шифр сложения с ключом = 15, чтобы расшифровать сообщение "WTAAD".

### Решение

Мы применяем алгоритм дешифрования к исходному тексту буква за буквой:

Шифр. Текст W  $\rightarrow$  22      Шифрование  $(22 - 15) \bmod 26$  Исходный  
текст 07  $\rightarrow$  h

Шифр. Текст T  $\rightarrow$  19 Шифрование  $(19 - 15) \bmod 26$  Исходный текст 04  
 $\rightarrow$  e

Шифр. Текст A  $\rightarrow$  00 Шифрование  $(00 - 15) \bmod 26$  Исходный текст 11  
 $\rightarrow$  l

Шифр. Текст A  $\rightarrow$  00 Шифрование  $(00 - 15) \bmod 26$  Исходный текст 11  
 $\rightarrow$  l

Шифр. Текст D  $\rightarrow$  03 Шифрование  $(03 - 15) \bmod 26$  Исходный текст 14  
 $\rightarrow$  o

Результат — "hello". Обратите внимание, что операции проводятся по модулю 26 (см. лекции 2-3), отрицательный результат должен быть отображен в  $Z_{26}$  (например,  $-15$  становится 11).

### Шифр сдвига

Исторически аддитивные шифры назывались **шифрами сдвига** — по той причине, что алгоритм шифрования может интерпретироваться как "клавиша сдвига буквы вниз", а алгоритм дешифрования может интерпретироваться как "клавиши сдвига буквы вверх". Например, если ключ = 15, алгоритм

кодирования сдвигает букву на 15 букв вниз (к концу алфавита). Алгоритм дешифрования сдвигает букву на 15 букв вверх (к началу алфавита). Конечно, когда мы достигаем конца или начала алфавита, мы двигаемся по кольцу к началу (объявленные свойства операции по модулю 26).

### **Шифр Цезаря**

Юлий Цезарь использовал аддитивный шифр, чтобы связаться со своими чиновниками. По этой причине аддитивные шифры упоминаются иногда как **шифры Цезаря**. Цезарь для своей связи использовал цифру 3.

Аддитивные шифры упоминаются иногда как шифры сдвига или шифры Цезаря.

### **Криптоанализ**

Аддитивные шифры уязвимы к атакам только зашифрованного текста, когда используется исчерпывающий перебор ключей (атака грубой силы). Множество ключей аддитивного шифра очень мало — их только 26. Один из ключей, нулевой, является бесполезным (зашифрованный текст будет просто соответствовать исходному тексту). Следовательно, остается только 25 возможных ключей. Ева может легко начать атаку грубой силы зашифрованного текста.

### **Пример 2.5**

Ева перехватила зашифрованный текст "UVACLYFZLJBYL". Покажите, как она может взломать шифр, используя атаку грубой силы.

### **Решение**

Ева пробует раскрыть текст и последовательно перебирает ключи начиная с первого. С помощью ключа номер 7 она получает осмысленный текст "not very secure" (не очень безопасный).

Зашифрованный текст: UVACLYFZLJBYL

|       |                               |
|-------|-------------------------------|
| К = 1 | Исходный текст: tubkxeuykiaxk |
| К = 2 | Исходный текст: styajwdxjhzwj |
| К = 3 | Исходный текст: rsxzivewigyvi |
| К = 4 | Исходный текст: qrwyhubvhfhuh |

K = 5            Исходный текст: pqvxgtaugewtg

K = 6            Исходный текст: opuwfsztfdvst

K = 7            Исходный текст: notverysecure

#### Статистические атаки

Аддитивные шифры также могут быть объектами статистических атак. Это особенно реально, если противник перехватил длинный зашифрованный текст. Противник может воспользоваться знаниями о частоте употребления символов в конкретном языке. [Таблица 2.1](#) показывает частоту появления определенных букв для английского текста длиной в 100 символов.

Таблица 2.1. Частота появления букв в английском тексте

| Буква | Частота | Буква | Частота | Буква | Частота | Буква | Частота |
|-------|---------|-------|---------|-------|---------|-------|---------|
|-------|---------|-------|---------|-------|---------|-------|---------|

|   |      |   |     |   |     |   |      |
|---|------|---|-----|---|-----|---|------|
| E | 12,7 | H | 6,1 | W | 2,3 | K | 0,08 |
|---|------|---|-----|---|-----|---|------|

|   |     |   |     |   |     |   |      |
|---|-----|---|-----|---|-----|---|------|
| T | 9,1 | R | 6,0 | F | 2,2 | J | 0,02 |
|---|-----|---|-----|---|-----|---|------|

|   |     |   |     |   |     |   |      |
|---|-----|---|-----|---|-----|---|------|
| A | 8,2 | D | 4,3 | G | 2,0 | Q | 0,01 |
|---|-----|---|-----|---|-----|---|------|

|   |     |   |     |   |     |   |      |
|---|-----|---|-----|---|-----|---|------|
| O | 7,5 | L | 4,0 | Y | 1,9 | X | 0,01 |
|---|-----|---|-----|---|-----|---|------|

|   |     |   |     |   |     |   |      |
|---|-----|---|-----|---|-----|---|------|
| I | 7,0 | C | 2,8 | P | 1,5 | Z | 0,01 |
|---|-----|---|-----|---|-----|---|------|

|   |     |   |     |   |     |  |  |
|---|-----|---|-----|---|-----|--|--|
| N | 6,7 | U | 2,8 | B | 1,0 |  |  |
|---|-----|---|-----|---|-----|--|--|

|   |     |   |     |   |  |  |  |
|---|-----|---|-----|---|--|--|--|
| S | 6,3 | M | 2,4 | V |  |  |  |
|---|-----|---|-----|---|--|--|--|

Однако информации о частоте единственного символа недостаточно, и это затрудняет анализ шифрованного текста, основанный на анализе частоты появления букв. Весьма желательно знать частоту появления комбинаций символов. Мы должны знать частоту появления в зашифрованном тексте комбинаций с двумя или с тремя символами и сравнивать ее с частотой в языке, на котором написан исходный документ.

Наиболее употребляемые группы с двумя символами (диаграмма (diagrams)) и группы с тремя символами (триграмма (trigrams)) для английского текста показаны в [таблице 2.2](#).

Таблица 2.2. Группы диаграмм и триграмм , основанные на их частоте появления в английском языке

Диагра TH,HE,IN,ER,AN,RE,ED,ON,ES,ST,EN,AT,TO,NT,HA,ND,OU,EA,NG,  
мма AS,OR,TI,IS,ET,IT,AR,TE,SE,HI,OF  
Тригра THE,ING,AND,HER,ERE,ENT,THA,NTH,WAS,ETH,FOR,DTH  
мма

### Пример 2.6

Ева перехватила следующий зашифрованный текст. Используя статистическую атаку, найдите исходный текст.

XLILSYWIMWRS AJSVWEPIJSVJSYVQMPPMSRHSPP EVWMXMWASV  
X-LQSVILY-VVCFIJSVIXLIWIPP IVVIGIMZIWQSVISJJIVW

### Решение

Когда Ева составит таблицу частоты букв в этом зашифрованном тексте, она получит: I = 14, V = 13, S = 12, и так далее. Самый частый символ – I — имеет 14 появлений. Это показывает, что символ I в зашифрованном тексте, вероятно, соответствует символу e в исходном тексте. Тем самым, ключ = 2. Ева расшифровывает текст и получает

the house is now for sale for four million dollars it is worth more hurry before  
the seller receives more offers

дом теперь продается за четыре миллиона долларов, стоит поспешить,  
пока продавец не получил больше предложений

## Вопрос 2. Шифры перестановки

### Определение преобразования - перестановка

*Шифрование перестановкой* заключается в том, что символы открытого текста переставляются по определенному правилу в пределах некоторого блока этого текста. Рассмотрим перестановку, предназначенную для шифрования сообщения длиной  $n$  символов. Его можно представить с помощью таблицы

$$\begin{pmatrix} 1, 2, \dots, n \\ i_1, i_2, \dots, i_n \end{pmatrix},$$

где  $i_1$  - номер места шифртекста, на которое попадает первая буква открытого текста при выбранном преобразовании,  $i_2$  - номер места для второй буквы и т. д.

В верхней строке таблицы выписаны по порядку числа от 1 до  $n$ , а в нижней те же числа, но в произвольном порядке. Такая таблица называется **перестановкой степени  $n$** .

Зная перестановку, задающую преобразование, можно осуществить как шифрование, так и расшифрование текста. В этом случае, сама таблица перестановки служит ключом шифрования.

Число различных преобразований шифра перестановки, предназначенного для шифрования сообщений длины  $n$ , меньше либо равно  $n!$  ( $n$  факториал). Заметим, что в это число входит и вариант преобразования, оставляющий все символы на своих местах.

С увеличением числа  $n$  значение  $n!$  растет очень быстро. Для использования на практике такой шифр не удобен, так как при больших значениях  $n$  приходится работать с длинными таблицами. Поэтому широкое распространение получили шифры, использующие не саму таблицу перестановки, а некоторое правило, порождающее эту таблицу. Рассмотрим несколько примеров таких шифров.

### **Примеры шифрования методом перестановки**

**Шифр перестановки "скитала"**. Известно, что в V веке до нашей эры правители Спарты, наиболее воинственного из греческих государств, имели хорошо отработанную систему секретной военной связи и шифровали свои послания с помощью *скитала*, первого простейшего криптографического устройства, реализующего метод простой перестановки.

Шифрование выполнялось следующим образом. На стержень цилиндрической формы, который назывался скитала, наматывали спиралью (виток к витку) полоску пергамента и писали на ней вдоль стержня несколько строк текста сообщения. Затем снимали со стержня полоску пергамента с

написанным текстом. Буквы на этой полоске оказывались расположенными хаотично.



Шифр "Скитала"

Такой же результат можно получить, если буквы сообщения писать по кольцу не подряд, а через определенное число позиций до тех пор, пока не будет исчерпан весь текст. Сообщение **"НАСТУПАЙТЕ"** при размещении его по окружности стержня по три буквы дает шифртекст: **"НУТАПЕСА\_ТЙ"**.

Для расшифрования такого шифртекста нужно не только знать правило шифрования, но и обладать ключом в виде стержня определенного диаметра. Зная только вид шифра, но, не имея ключа, расшифровать сообщение было непросто.

**Шифрующие таблицы.** С начала эпохи Возрождения (конец XIV столетия) начала возрождаться и криптография. В разработанных шифрах перестановки того времени применяются шифрующие таблицы, которые, в сущности, задают правила перестановки букв в сообщении.

В качестве ключа в шифрующих таблицах используются:

- *размер таблицы;*
- *слово или фраза, задающие перестановку;*
- *особенности структуры таблицы.*

Одним из самых примитивных табличных шифров перестановки является **простая перестановка**, для которой ключом служит размер таблицы. Этот метод шифрования сходен с шифром скитала. Например, сообщение **"ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ"** записывается в таблицу поочередно по столбцам. Результат заполнения таблицы из 5 строк и 7 столбцов показан на рисунке.

После заполнения таблицы текстом сообщения по столбцам для формирования шифртекста считывают содержимое таблицы по строкам. Если

шифртекст записывать группами по пять букв, получается такое шифрованное сообщение: **"ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ"**.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| Т | Н | П | В | Е | Г | Л |
| Е | А | Р | А | Д | О | Н |
| Р | Т | И | Е | Ь | В | О |
| М | О | Б | Т | М | П | Ч |
| И | Р | Ы | С | О | О | Ь |

Заполнение шифрующей таблицы из 5 строк и 7 столбцов

Естественно, отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Следует заметить, что объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

### Практическое занятие № 3.

**Тема:** Блочный шифр. Ключевая система

**Цель:** Углубить знания, по основам многоалфавитного шифрования.

Исследовать основные характеристики алгоритма шифрования.

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### **Теоретическое обоснование**

Наиболее простыми являются шифры замены или подстановки, особенностью которых является замена символов (слов) открытого текста соответствующими символами, принадлежащими алфавиту шифротекста. Различают: одноалфавитную, многоалфавитную замену.

Вскрытие одноалфавитного шифра осуществляется на учете частоты появления отдельных букв или сочетаний (биграмм, триграмм, и т.д.) в языке. Примером многоалфавитного шифра замены является система Виженера. Шифрование осуществляется по таблице, представляющей собой квадратную матрицу размерности  $n \times n$ , где  $n$  - число символов алфавита (для русского алфавита - 32). Первая строка содержит все символы алфавита. Каждая последующая строка получается из предыдущей путем циклического сдвига вправо на один символ (или влево).

Выбирается ключ или ключевая фраза. После чего процесс зашифровывания осуществляется следующим образом:

1. Под каждой буквой исходного сообщения последовательно записываются буквы ключа (если ключ короче – его используют несколько раз).
2. Каждая буква шифротекста находится на пересечении столбца таблицы, определяемого буквой открытого текста, и строки, определяемой буквой ключа.

**Пример.** Пусть надо зашифровать текст – *А нам все равно.*

В качестве ключа используем слово – *КОЛОКОЛА.*

Тогда процесс зашифрования можно представить в следующем виде. Таблица Виженера для ключевого слова КОЛОКОЛА примет вид представленный в таблице 1.

**ТАБЛИЦА 1 – ТАБЛИЦА ВИЖЕНЕРА С КЛЮЧЕВЫМ СЛОВОМ  
КОЛОКОЛА**

|          |   |          |   |   |   |   |   |   |   |   |   |          |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |          |
|----------|---|----------|---|---|---|---|---|---|---|---|---|----------|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----------|
| А        | Б | В        | Г | Д | Е | Ж | З | И | Й | К | Л | М        | Н        | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | _        |
| <b>К</b> | Л | М        | Н | О | П | Р | С | Т | У | Ф | Х | Ц        | Ч        | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й        |
| О        | П | Р        | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ        | Ы        | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | <b>Н</b> |
| Л        | М | Н        | О | П | Р | С | Т | У | Ф | Х | Ц | Ч        | <b>Ш</b> | Щ | Ъ | Ы | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й | К        |
| <b>О</b> | П | Р        | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ        | Ы        | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н        |
| К        | Л | М        | Н | О | П | Р | С | Т | У | Ф | Х | <b>Ц</b> | Ч        | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й        |
| О        | П | Р        | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ        | Ы        | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | <b>Н</b> |
| Л        | М | <b>Н</b> | О | П | Р | С | Т | У | Ф | Х | Ц | Ч        | Ш        | Щ | Ъ | Ы | Ь | Э | Ю | Я | _ | А | Б | В | Г | Д | Е | Ж | З | И | Й | К        |
| А        | Б | В        | Г | Д | Е | Ж | З | И | Й | К | Л | М        | Н        | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | _        |

Буквы, выделенные в таблице, соответствуют символам шифротекста.

В результате процедуры шифрования получаем текст, представленный в таблице 2.

Таблица 2 – Пример шифрования текста с помощью таблицы Виженера

| Исходный текст      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| А                   | _ | Н | А | М | _ | В | С | Е | _ | Р | А | В | Н | О |
| Ключевое слово      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| К                   | О | Л | О | К | О | Л | А | К | О | Л | О | К | О | Л |
| Зашифрованный текст |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| К                   | Й | Ч | К | Ц | Й | М | Ы | П | Й | Ъ | К | М | Ч | Щ |

Расшифровка осуществляется следующим образом. Под буквами шифротекста последовательно записываются буквы ключа: в строке соответствующей очередной букве ключа, происходит поиск соответствующей буквы шифротекста. Находящаяся над ней в первой строке буква и будет соответствовать букве исходного текста.

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.
2. Провести исследование системы многоалфавитной замены.

Студенты делятся на две подгруппы. В первой подгруппе студенты выбирают ключевые слова, а так же получают текст, выданный преподавателем. Затем они строят таблицу и осуществляют шифрование текста. Студенты второй подгруппы, получив от студентов первой подгруппы зашифрованное сообщение и необходимый сдвиг, строят таблицу и осуществляют процесс дешифрования.

По окончании расшифрования студенты второй подгруппы приступают к процедуре зашифрования с использованием нового ключевого слова. Студенты первой подгруппы, получив ключевое слово и зашифрованный текст, приступают к его расшифрованию.

### **Вопросы для контроля**

1. Дайте определение шифра многоалфавитной замены. Какие шифры данного вида вы знаете.
2. Дайте определение шифра многоалфавитной замены.
3. Назовите основные достоинства и недостатки шифра многоалфавитной замены.

### **Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 4.

**Тема:** Шифрование алгоритмом S-AES.

**Цель и содержание:**

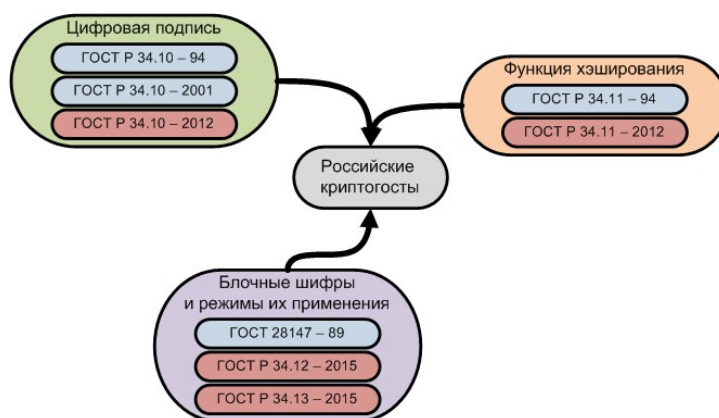
1. Углубить знания, полученные на лекциях по основам построения блочных шифров
2. Изучить алгоритм шифрования Магма.

**Формируемые компетенции:** ОПК-6, ОПК-9.

**Теоретическое обоснование**

Прежде чем, перейдем к изучению алгоритма шифрования «Магма» (ГОСТ 28147-89) рассмотрим три опоры, три слона, на которых держится отечественная криптография:

(слайд № ) по щелчку



В настоящее время отечественная криптография базируется на нескольких основных государственных стандартах:

1. ГОСТ 34.10 -2012 – (стандарт, описывающий процесс формирования и проверки электронной цифровой подписи);
2. ГОСТ 34.11-2012. (стандарт, описывающий отечественную функцию хэширования);
3. ГОСТ 34.12-2019. (стандарт блочного шифрования);  
ГОСТ 34.13- 2019. (режимы работы блочных шифров).

Стандарт отечественного шифрования ГОСТ Р 34.12-2019 «Информационная технология. Криптографическая защита информации. Блочные шифры» утвержден приказом от 19 июня 2019 года № 749-ст.

Стандарт вступил в действие с 1 января 2016 года. *(Шифр разработан Центром защиты информации и специальной связи ФСБ России с участием*

ОАО «Информационные технологии и коммуникационные системы» (ОАО «[ИнфоТеКС](#)»)).

Национальным стандартом Российской Федерации ГОСТ Р 34.12–2019 определены два базовых алгоритма шифрования:

– «Магма», представляющий собой алгоритм шифрования ГОСТ 28147-89 с открытыми блоками подстановки

–«Кузнечик», относится к шифрам SPN типа (шифр перестановки и подстановки).

Данные алгоритмы шифрования могут применяться в криптографических методах обработки и защиты информации, в том числе для обеспечения конфиденциальности, аутентичности и целостности информации при ее передаче, обработке и хранении в автоматизированных системах.

Прежде чем, мы перейдем к изучению принципов построения алгоритма шифрования «Магма», давайте проведем обсуждение.

Вопрос: «При разработке отечественного стандарта шифрования целесообразно:

1. использовать уже известные и проверенные решения, даже если он разработаны не в нашей стране;

2. разработать новые решения и технологии закрытия информации.

(Студенты в течение 1-2' обсуждают, предлагают и обосновывают свое решение). Одни – можно старое, другие – надо разрабатывать свое. Я напоминаю потом про правило Кирхгоффа – противник знает алгоритм шифрования – зачем мучиться, если есть хорошее криптографическое решение, но его необходимо доработать по своему (И это увидим, вместо перестановки – сдвиг, два сумматора, больше ключ, больше раундов)

Алгоритм блочного шифрования «Магма» представляет собой российский государственный стандарт шифрования данных ГОСТ 28147-89, но с открытыми блоками подстановки S.

Это отечественный алгоритм блочного шифрования, являющийся аналогом DES. В качестве базовой криптографической функции используется «Петля Фейстеля».

В таблице 1 приведены основные характеристики алгоритмов шифрования DES и «Магма».

|                                       | DES              | Магма            |
|---------------------------------------|------------------|------------------|
| Криптографические преобразования      | «Петля Фейстеля» | «Петля Фейстеля» |
| Размер входного открытого блока       | 64 бит           | 64 бит           |
| Размер выходного зашифрованного блока | 64 бит           | 64 бит           |

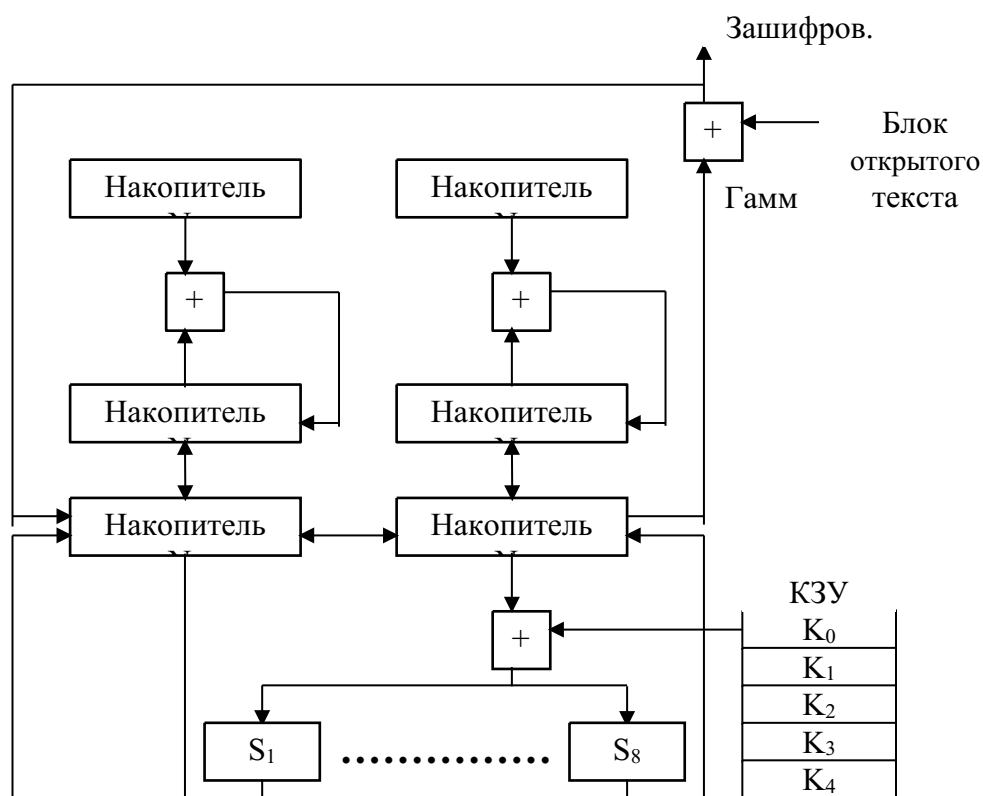
|                                                    |              |                                     |
|----------------------------------------------------|--------------|-------------------------------------|
| Размер ключа                                       | 56 бит       | 256 бит                             |
| Количество раундов                                 | 16           | 32                                  |
| Ключевая система                                   | есть         | нет                                 |
| Структура S блоков                                 | известна     | известна                            |
| Блоки перестановок                                 | есть         | нет                                 |
| Регистр сдвига                                     | нет          | есть                                |
| Количество сумматоров, используемых в одном раунде | 1 - по mod 2 | 1 - по mod 2<br>1 – по mod $2^{32}$ |

Разработчики сумели соединить в ГОСТе две важные характеристики:

- высокую криптографическую стойкость;
- возможность эффективного программного исполнения (за счёт использования узлов, реализуемых на современной аппаратно-программной платформе).

Алгоритм работает с 64-битными входными блоками, 64-битными выходными блоками и ключами длиной 256 бит. Использование 256-битного ключа позволяет не обращать внимания на возможность прямой атаки, поскольку мощность ключевого пространства равна  $2^{256}$ .

Структура алгоритма показана на рисунке 1.



## Рисунок 1 – Алгоритм шифрования Магма

Алгоритм может применяться в следующих режимах:

- простая замена;
- гаммирование;
- гаммирование с обратной связью;
- выработка имитовставки.

Для всех четырёх режимов применяется одно общее преобразование:

$$L_i = R_{i-1};$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), i = 1 \dots 31,$$

где L и R – соответственно левая и правая части 64-битного блока. Это преобразование характерно для всех раундов (всего их 32), кроме последнего.

Для него преобразование будет иметь вид:

$$R_i = R_{i-1};$$

$$L_i = L_{i-1} \oplus f(R_{i-1}, K_i), i = 32.$$

Для каждого раунда функция  $f$  остаётся неизменной. Перед началом преобразования блок открытого текста разбивается на две 32-битные половины, которые помещаются в 32-разрядные регистры  $N_2$  и  $N_1$ . Также перед началом зашифрования 256-битный ключ, разбитый на 8 частей по 32 бита каждая ( $K_0 - K_7$ ), помещается в ключевое запоминающее устройство (КЗУ). Далее содержимое регистра  $N_1$  суммируется по модулю  $2^{32}$  с очередным заполнением устройства, а ключевые последовательности выбираются из КЗУ в следующем порядке:

- в раундах с 1-го по 24-й -  $K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7$ ;

- в раундах с 25-го по 32-й -  $K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0$ .

Полученная битовая последовательность разбивается по 4 бита на 8 блоков ( $S_1 - S_8$ ). В них реализуется подстановка, имеющая, например, следующий вид:

|          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Двоичный |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| № входа  |   |   |   |   |   |   |   |   |   |   | 0 | 1 | 2 | 3 | 4 | 5 |
| Двоичный | 7 | 1 | 1 | 1 | 0 | 8 | 9 | 1 | 1 | 4 | 6 | 1 | 1 | 2 | 5 | 3 |
| № выхода |   | 0 |   | 3 |   |   |   | 5 |   | 4 |   | 2 |   | 1 |   |   |

Числовое представление битового входа однозначно определяет последовательный номер числа в подстановке и битовое представление выхода S-блока. Для каждого S-блока задаётся своя подстановка, которая является долговременным секретным ключом. Генерация подстановок в S-блоках является сложной математической задачей, от решения которой как раз и зависит стойкость этого алгоритма.

После S-блоков последовательность поступает в циклический регистр сдвига, который осуществляет смещение на определённое количество разрядов влево. Использование этого регистра вместо перестановок, реализуемых в DES, обусловлено тем, что данный регистр легко реализуется как программно (используя битовую операцию циклического сдвига), так и аппаратно. При этом эффект распространения влияния каждого бита блока открытого текста на каждый бит блока зашифрованного текста достигается за счёт прохождения преобразуемого блока через данный регистр 32 раза.

Можно задать вопрос – почему сдвиг идет на 11 разрядов

*Можно показать на примере*

Далее происходит поразрядное суммирование по модулю 2 содержимого регистра циклического сдвига с содержимым регистра  $N_2$ . Результат суммирования записывается в регистр  $N_1$ , а старое содержимое  $N_1$  записывается в  $N_2$ .

Все описанные операции составляют один раунд преобразования очередного блока открытого текста.

Для закрепления теоретических знаний об алгоритме проведем 2 раунда шифрования учебного ГОСТ. Он был разработан на кафедре ИБАС в 2017 году.

Выдаю раздаточный материал (4 стр.) всем участникам (студентам, преподавателям, комиссии) 1 раунд делаю на доске, поясняя, как выполняются операции.

Второй раунд студенты выполняют самостоятельно. Молодые

*преподаватели помогают студентам в этом задании (рефлексия занятия)*

*Задание на СР: выполнить шифрование с помощью учебного ГОСТ. Открытый текст и секретный ключ выбирают самостоятельно!!!*

Стойкость и безопасность ГОСТ 28147-89.

Данный алгоритм обладает высокими криптографическими свойствами, которые достигаются за счёт:

- большой длины ключа (256 бит); если учесть, что подстановки в S-шифраторах могут являться секретными, то общий объем секретной информации окажется равным 610 битам;

Чтобы понять, что 256 бит - достаточно большая длина ключа, можно привести следующие данные. Подсчитать сложность атаки методом тотального перебора достаточно просто. Если ключ имеет длину 64 бита, то суперкомпьютер, который может опробовать 1 млн ключей за 1 с, потратит более 5 тыс. лет на проверку всех возможных ключей. При увеличении длины ключа до 128 бит, этому же суперкомпьютеру понадобится  $10^{25}$  лет, чтобы перебрать все ключи. Вселенная существует всего-навсего  $10^{10}$  лет, поэтому можно сказать, что  $10^{25}$  – это достаточно большой запас надёжности для тех, кто пользуется 128-битными ключами.

- 32 раундов преобразований. Например, уже после 8 раундов единичное изменение входной последовательности отразится на изменении всех битов выхода (для DES это наступает после 5 раунда).

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.
2. Провести исследование алгоритма шифрования S-Магма.

Каждый студент получает индивидуальное задание по выработке ключа для алгоритма упрощенного S-Магма. Задание представлено в таблице 1.

Таблица 1 – Задание на исследование алгоритма шифрования S-Магма

| № п/п | Исходные данные | № п/п | Исходные данные |
|-------|-----------------|-------|-----------------|
| 1.    | 1100110101      | 15.   | 1111110101      |
| 2.    | 1010100101      | 16.   | 1100100101      |
| 3.    | 0001010101      | 17.   | 0111010101      |
| 4.    | 1010111101      | 18.   | 1110111101      |
| 5.    | 1011101011      | 19.   | 1001101011      |
| 6.    | 1100111101      | 20.   | 1100100001      |
| 7.    | 0011010101      | 21.   | 0100010101      |
| 8.    | 1100001111      | 22.   | 1101101111      |
| 9.    | 0100110101      | 23.   | 0111110101      |

|     |            |     |            |
|-----|------------|-----|------------|
| 10. | 1011010101 | 24. | 1011100101 |
| 11. | 0100001100 | 25. | 1111101100 |
| 12. | 0011001010 | 26. | 1111001010 |
| 13. | 1110101000 | 27. | 0100101001 |
| 14. | 1100011111 | 28. | 0001100110 |

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования систем по своему варианту и ответы на контрольные вопросы.

При защите студенты предоставляют программу с помощью которой провели исследования ключевой системы данного алгоритма шифрования.

### **Контрольные вопросы**

1. Поясните особенности шифрования с использованием S-Магма. Какие виды шифров вы можете отнести к этой группе.
2. Какие виды ключей используются в S-Магма.
3. Основные этапы получения ключей в алгоритме S-Магма
4. Операции, используемые в алгоритме шифрования S-Магма

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие №5

### РАСШИФРОВАННОЕ АЛГОРИТМОМ S-AES

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретическое обоснование

#### Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Провести исследование процесса шифрования .

Каждый студент получает индивидуальное задание по осуществлению процесса шифрования. Открытый текст приведен в таблице 1. В качестве ключа для алгоритма S-DES использовать результаты предыдущего занятия.

Таблица 1 – Задание на разработку ключей для S-DES

| № п/п | Исходные данные | № п/п | Исходные данные |
|-------|-----------------|-------|-----------------|
| 1.    | 1111110101      | 15.   | 1100110101      |
| 2.    | 1100100101      | 16.   | 1010100101      |
| 3.    | 0111010101      | 17.   | 0001010101      |
| 4.    | 1110111101      | 18.   | 1010111101      |
| 5.    | 1001101011      | 19.   | 1011101011      |
| 6.    | 1100100001      | 20.   | 1100111101      |
| 7.    | 0100010101      | 21.   | 0011010101      |
| 8.    | 1101101111      | 22.   | 1100001111      |
| 9.    | 0111110101      | 23.   | 0100110101      |
| 10.   | 1011100101      | 24.   | 1011010101      |
| 11.   | 1111101100      | 25.   | 0100001100      |
| 12.   | 1111001010      | 26.   | 0011001010      |
| 13.   | 0100101001      | 27.   | 1110101000      |
| 14.   | 0001100110      | 28.   | 1100011111      |

#### Содержание отчета и его форма

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования систем по своему варианту и ответы на контрольные вопросы.

#### Вопросы для защиты работы

1. Поясните особенности процесса шифрования в S-DES.

2. Сколько раундов используется в S-DES.
3. Основные операции шифрования с помощью алгоритма S-DES.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Практическое занятие №6**

### **ШИФРОВАНИЕ И РАСШИФРОВАНИЕ АЛГОРИТМОМ S-МАГМА**

#### **Цель и содержание:**

1. Углубить знания, по основам построения алгоритма шифрования .
2. Исследовать вопросы расшифрования принятого сообщения.

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### **Теоретическое обоснование**

##### **Вопросы для защиты работы**

1. Поясните особенности процесса дешифрования в S-DES.
2. Сколько раундов в S-DES для осуществления дешифрования.
3. Проведите сравнение алгоритма DES и S-DES по этапу расширения передаваемого сообщения.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Практическое занятие № 7**

### **ШИФРОВАНИЕ НА ОСНОВЕ ГЕНЕРАТОРА ФИБОНАЧЧИ**

#### **Цель и содержание:**

1. Углубить знания, по основам поточного шифрования .
2. Исследовать вопросы получения синхронного ПСП.

1. **Формируемые компетенции:** ОПК-6, ОПК-9.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 8

### РАСШИФРОВАНИЕ НА ОСНОВЕ ГЕНЕРАТОРА ФИБОНАЧЧИ

#### Цель и содержание:

1. Углубить знания, по основам поточного шифрования .
2. Изучить правила построения генератора синхронного ПСП на основе многотактового фильтра.

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретическое обоснование

Построить эффективный криптоалгоритм можно, лишь отказавшись от абсолютной стойкости. Возникает задача разработки такого теоретически нестойкого шифра, для вскрытия которого противнику потребовалось бы выполнить такое число операций, которое неосуществимо на современных и ожидаемых в ближайшей перспективе вычислительных средствах за разумное время. В первую очередь представляет интерес схема, использующая ключ небольшой разрядности, который в дальнейшем выполняет функцию "зародыша", порождающего значительно более длинную ключевую последовательность.

Данный результат может быть достигнут при использовании *гаммирования*, схема которого показана на рисунке 1. Гаммированием называют процедуру наложения на входную информационную последовательность *гаммы* шифра, т. е. последовательности с выходов *генератора псевдослучайных кодов*. Последовательность называется *псевдослучайной*, если по своим статистическим свойствам она неотличима от истинно *случайной* последовательности, но в отличие от последней является детерминированной, т. е. знание алгоритма ее формирования дает возможность ее повторения необходимое число раз. Если символы входной информационной последовательности и гаммы представлены в двоичном виде, наложение чаще всего реализуется с помощью операции

поразрядного сложения по модулю 2. Надежность шифрования методом гаммирования определяется качеством генератора гаммы

Простейшие устройства синхронного и самосинхронизирующегося шифрования с использованием ГПК, реализованного на основе  $N$ -разрядного регистра сдвига с линейной обратной связью - *LFSR* (Linear Feedback Shift Register), называются *скремблерами*, а сам процесс преобразования – скремблированием.

Используя приложение выбираем производящий полином  $P(x)$  такой, чтобы его степень была равна трем, а вес полинома – 3. В качестве образующего полинома используем  $P(x) = x^4 + x + 1$ .

Количество ячеек памяти должно равняться степени порождающего полинома (т.е. четырем), количество сумматоров по модулю два определяется числом ненулевых коэффициентов перед степенью « $x$ » (т.е. один). Тогда схема кодирующего устройства примет следующий вид:

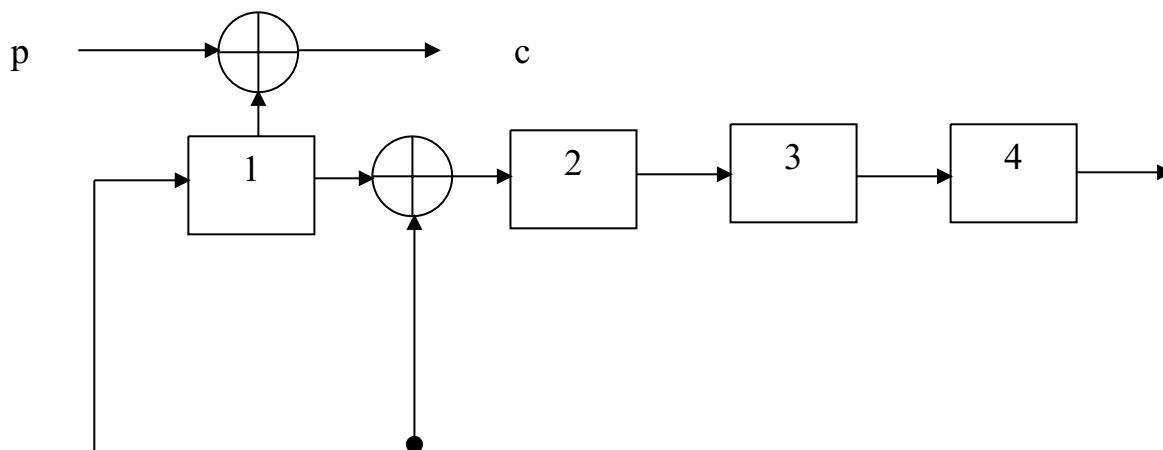


Рисунок 1 - Шифрующее устройство поточного кода.

Для иллюстрации работы кодирующего устройства воспользуемся соотношениями, показывающими процесс образования символов в ячейках:

Символ в 1 яч. = яч. 4\*

Символ в 2 яч. = яч. 4\* + яч. 1\*

Символ в 3 яч. = яч. 2\*

Символ в 4 яч. = яч. 3\*

В таблице 1 приведены результаты шифрования открытого кода в течение первых 7 тактов.

**Таблица 1.**

| Номер такта | Открытый текст | ЯЧЕЙКА № 1 | ЯЧЕЙКА № 2 | ЯЧЕЙКА № 3 | ЯЧЕЙКА № 3 | Закрывается текст |
|-------------|----------------|------------|------------|------------|------------|-------------------|
| 1           | 1              | 1          | 1          | 0          | 0          | 1                 |
| 2           | 1              | 1          | 0          | 1          | 1          | 1                 |
| 3           | 0              | 1          | 0          | 0          | 1          | 0                 |
| 4           | 1              | 1          | 1          | <u>0</u>   | 0          | 0                 |
| 5           | 0              | 0          | 0          | 1          | 0          | 0                 |
| 6           | 1              | 0          | 0          | 0          | 1          | 1                 |
| 7           | 1              | 1          | 0          | 0          | 0          | 0                 |

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.
2. Построить схему шифратора на основе МКФ и провести исследование системы поточного шифрования.

Используя данные представленные в таблице 4, произвести разработку шифратора ПСП и осуществить процедуры зашифрования и расшифрования.

**Таблица 4 – Исходные данные**

| М | A0 | A1 | A2 | A3 | A4 | A5 | A6 |
|---|----|----|----|----|----|----|----|
| 4 | 1  | 0  | 0  | 1  | 1  |    |    |
| 4 | 1  | 1  | 0  | 0  | 1  |    |    |
| 5 | 1  | 0  | 0  | 1  | 0  | 1  |    |
| 5 | 1  | 0  | 1  | 0  | 0  | 1  |    |
| 5 | 1  | 1  | 1  | 1  | 0  | 1  |    |
| 5 | 1  | 1  | 0  | 1  | 1  | 1  |    |
| 5 | 1  | 1  | 1  | 0  | 1  | 1  |    |
| 6 | 1  | 0  | 0  | 0  | 0  | 1  | 1  |

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная: 1-5*

*Дополнительная: 1-6*

**Практическое занятие № 9**  
**ШИФРОВАНИЕ НА ОСНОВЕ ГЕНЕРАТОРОВ МНОГО**  
**ТАКТОВЫХ КОДОВЫХ ФИЛЬТРОВ**

**Цель и содержание:**

1. Углубить знания, по основам поточного шифрования .
2. Изучить правила построения шифратора синхронного ПСП на основе многотактового фильтра.

**Формируемые компетенции**

1. Способность применять средства криптографической и технической защиты информации для решения задач профессиональной деятельности (ОПК-9)
2. Способность при решении профессиональных задач организовывать защиту информации ограниченного доступа в соответствии с нормативными правовыми актами, нормативными и методическими документами Федеральной службы безопасности Российской Федерации, Федеральной службы по техническому и экспортному контролю (ОПК-6)

**Теоретический материал**

В синхронных поточных шифрах гамма формируется независимо от входной последовательности, каждый элемент (бит, символ, байт и т. п.) которой таким образом шифруется независимо от других элементов. В синхронных поточных шифрах отсутствует эффект размножения ошибок, т. е. число искаженных элементов в расшифрованной последовательности равно числу искаженных элементов зашифрованной последовательности, пришедшей из канала связи.

Вставка или выпадение элемента зашифрованной последовательности недопустимы, так как из-за нарушения синхронизации это приведет к неправильному расшифрованию всех последующих элементов.

В таблице 2 показан пример поточного шифрования и расшифрования

двоичной последовательности *11100101010110* с использованием гаммы формируемой 4-разрядным *LFSR* при начальном состоянии **1001**.

Зашифрованная

последовательность имеет вид *01001010010010*.

Таблица 1 – Пример поточного шифрования и расшифрования двоичной последовательности, когда отсутствуют ошибки в принятой комбинации

| Процедура шифрования поточного двоичной последовательности |          |               |   |   |   | Процедура расшифрования, поточного когда отсутствуют ошибки в принятой комбинации |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|-----------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                  |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                          | <i>p</i> | Генератор ПСП |   |   |   |
| 0                                                          | 1        | 1             | 1 | 0 | 0 | 0                                                                                 | 1        | 1             | 1 | 0 | 0 |
| 1                                                          | 1        | 0             | 1 | 1 | 0 | 1                                                                                 | 1        | 0             | 1 | 1 | 0 |
| 0                                                          | 1        | 1             | 0 | 1 | 1 | 0                                                                                 | 1        | 1             | 0 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 0 | 1 | 0                                                                                 | 0        | 0             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 0 | 1 | 0 | 1                                                                                 | 0        | 1             | 0 | 1 | 0 |
| 0                                                          | 1        | 1             | 1 | 0 | 1 | 0                                                                                 | 1        | 1             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 1 | 1 | 0 | 1                                                                                 | 0        | 1             | 1 | 1 | 0 |
| 0                                                          | 1        | 1             | 1 | 1 | 1 | 0                                                                                 | 1        | 1             | 1 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 1 | 1 | 0                                                                                 | 0        | 0             | 1 | 1 | 1 |
| 1                                                          | 1        | 0             | 0 | 1 | 1 | 1                                                                                 | 1        | 0             | 0 | 1 | 1 |
| 0                                                          | 0        | 0             | 0 | 0 | 1 | 0                                                                                 | 0        | 0             | 0 | 0 | 1 |
| 0                                                          | 1        | 1             | 0 | 0 | 0 | 0                                                                                 | 1        | 1             | 0 | 0 | 0 |
| 1                                                          | 1        | 0             | 1 | 0 | 0 | 1                                                                                 | 1        | 0             | 1 | 0 | 0 |
| 0                                                          | 0        | 0             | 0 | 1 | 0 | 0                                                                                 | 0        | 0             | 0 | 1 | 0 |

При отсутствии искажений в канале связи после расшифрования с использованием той же гаммы получается исходная последовательность.

На рисунке 1 приведена структура дешифратора поточного шифра, использующего ПСП, реализованного на основе многотактового кодового фильтра. Количество ячеек памяти должно равняться степени порождающего полинома (т.е. четырем), количество сумматоров по модулю два определяется числом ненулевых коэффициентов перед степенью «х» (т.е. один). Тогда схема кодирующего устройства примет следующий вид:

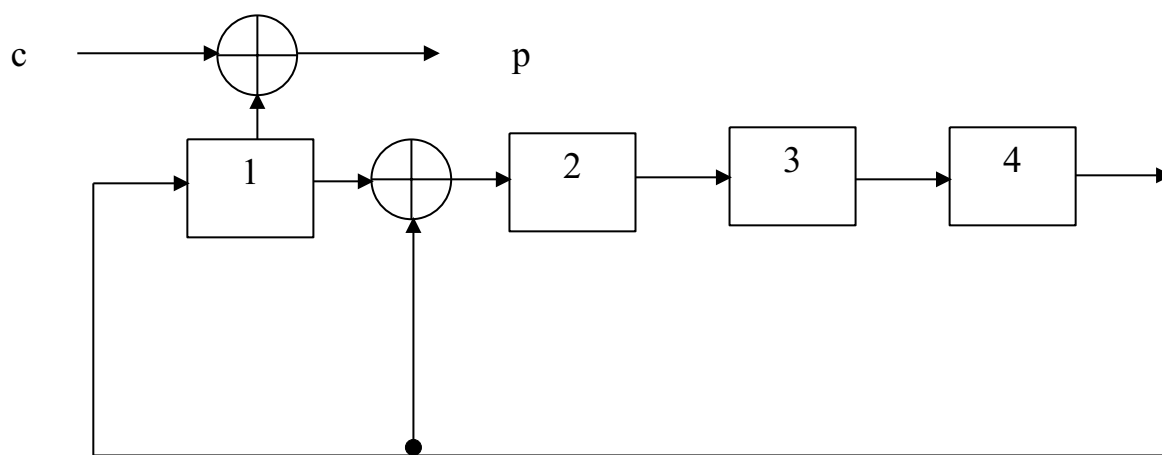


Рисунок 1 - Дешифратор поточного кода

В таблице 2 рассмотрена ситуация, когда при передаче зашифрованной последовательности был потерян четвертый бит, и вместо правильной последовательности к получателю пришла последовательность **0101010010010**.

Таблица 2 – Пример поточного шифрования и расшифрования двоичной последовательности, когда при передаче был потерян четвертый бит

| Процедура шифрования поточного двоичной последовательности |          |               |   |   |   | Процедура расшифрования, когда при передаче был потерян четвертый бит |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|-----------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                      |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                              | <i>p</i> | Генератор ПСП |   |   |   |
| 0                                                          | 1        | 1             | 1 | 0 | 0 | 0                                                                     | 1        | 1             | 1 | 0 | 0 |
| 1                                                          | 1        | 0             | 1 | 1 | 0 | 1                                                                     | 1        | 0             | 1 | 1 | 0 |
| 0                                                          | 1        | 1             | 0 | 1 | 1 | 0                                                                     | 1        | 1             | 0 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 0 | 1 | 1                                                                     | 1        | 0             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 0 | 1 | 0 | 0                                                                     | 1        | 1             | 0 | 1 | 0 |
| 0                                                          | 1        | 1             | 1 | 0 | 1 | 1                                                                     | 0        | 1             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 1 | 1 | 0 | 0                                                                     | 1        | 1             | 1 | 1 | 0 |
| 0                                                          | 1        | 1             | 1 | 1 | 1 | 0                                                                     | 1        | 1             | 1 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 1 | 1 | 1                                                                     | 1        | 0             | 1 | 1 | 1 |
| 1                                                          | 1        | 0             | 0 | 1 | 1 | 0                                                                     | 0        | 0             | 1 | 1 | 1 |
| 0                                                          | 0        | 0             | 0 | 0 | 1 | 0                                                                     | 0        | 0             | 0 | 0 | 1 |
| 0                                                          | 1        | 1             | 0 | 0 | 0 | 1                                                                     | 0        | 1             | 0 | 0 | 0 |
| 1                                                          | 1        | 0             | 1 | 0 | 0 | 0                                                                     | 0        | 1             | 0 | 0 | 0 |
| 0                                                          | 0        | 0             | 0 | 1 | 0 |                                                                       |          |               |   |   |   |

Видно, что после расшифрования всех битов, следующих после выпавшего, происходят искажения информации. В результате вместо битовой строки **0101010110** будет получена строка **1101110000**.

В таблице 4 рассмотрена ситуация, когда при передаче зашифрованной последовательности произошло искажение пятого (1 - 0) и восьмого (0 - 1) битов и вместо правильной последовательности к получателю пришла последовательность **01000011010010**. Видно, что после расшифрования вместо правильной строки будет получена строка **11101100010110** с искаженным пятым (0—1) и восьмым (1—0) битами.

Таблица 3 – Пример поточного шифрования и расшифрования двоичной последовательности, когда при передаче произошло искажение битов

| Процедура поточного шифрования двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда при передаче произошло искажение битов |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|---------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                        | <i>p</i> | Генератор ПСП |   |   |   |
| 0                                                          | 1        | 1             | 1 | 0 | 0 | 0                                                                               | 1        | 1             | 1 | 0 | 0 |
| 1                                                          | 1        | 0             | 1 | 1 | 0 | 1                                                                               | 1        | 0             | 1 | 1 | 0 |
| 0                                                          | 1        | 1             | 0 | 1 | 1 | 0                                                                               | 1        | 1             | 0 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 0 | 1 | 0                                                                               | 0        | 0             | 1 | 0 | 1 |
| 0                                                          | 0        | 1             | 0 | 1 | 0 | 0                                                                               | 1        | 1             | 0 | 1 | 0 |
| 0                                                          | 1        | 1             | 1 | 0 | 1 | 0                                                                               | 1        | 1             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 1 | 1 | 0 | 1                                                                               | 0        | 1             | 1 | 1 | 0 |
| 1                                                          | 1        | 1             | 1 | 1 | 1 | 1                                                                               | 0        | 1             | 1 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 1 | 1 | 0                                                                               | 0        | 0             | 1 | 1 | 1 |
| 1                                                          | 1        | 0             | 0 | 1 | 1 | 1                                                                               | 1        | 0             | 0 | 1 | 1 |
| 0                                                          | 0        | 0             | 0 | 0 | 1 | 0                                                                               | 0        | 0             | 0 | 0 | 1 |
| 0                                                          | 1        | 1             | 0 | 0 | 0 | 0                                                                               | 1        | 1             | 0 | 0 | 0 |
| 1                                                          | 1        | 0             | 1 | 0 | 0 | 1                                                                               | 1        | 0             | 1 | 0 | 0 |
| 0                                                          | 0        | 0             | 0 | 1 | 0 | 0                                                                               | 0        | 0             | 0 | 1 | 0 |

### Методика и порядок выполнения работы

3. Изучить теоретический материал работы.
4. Построить схему дешифратора на основе МКФ и провести исследование системы поточного шифрования.

Используя данные представленные в таблице 4, произвести разработку дешифратора поточного шифра на основе ПСП и осуществить процедуры зашифрования и расшифрования.

Таблица 4 – Исходные данные

| М | A0 | A1 | A2 | A3 | A4 | A5 | A6 |
|---|----|----|----|----|----|----|----|
| 4 | 1  | 0  | 0  | 1  | 1  |    |    |
| 4 | 1  | 1  | 0  | 0  | 1  |    |    |
| 5 | 1  | 0  | 0  | 1  | 0  | 1  |    |
| 5 | 1  | 0  | 1  | 0  | 0  | 1  |    |
| 5 | 1  | 1  | 1  | 1  | 0  | 1  |    |
| 5 | 1  | 1  | 0  | 1  | 1  | 1  |    |
| 5 | 1  | 1  | 1  | 0  | 1  | 1  |    |
| 6 | 1  | 0  | 0  | 0  | 0  | 1  | 1  |

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 10

### Арифметика целых чисел

#### Цель и содержание:

1. Углубить знания, по основам построения поточных систем шифровании.
2. Изучить алгоритм поточного шифрования S-A5

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретическое обоснование

Шифры потока более эффективны для обработки в реальном масштабе времени. Некоторые шифры потока использовались в различных протоколах в течение прошлых нескольких десятилетий. Мы рассмотрим : RC4 и A5/1.

В этом разделе мы вводим шифр потока, который использует линейный регистр сдвига, чтобы создать битовый поток: A5/1. A5/1 (член семейства шифров A5) используется в **Глобальной Системе Мобильной связи (GSM)**. Телефонная связь в GSM осуществляется как последовательность кадров на 228 битов, при этом каждый кадр длится 4,6 миллисекунды. A5/1 создает поток бит, исходя из ключа на 64 бита. Разрядные потоки собраны в буфере по 228 битов, чтобы складывать их по модулю два с кадром на 228 битов, как показано на рисунке 1.



Рисунок – 1 Общий вид A5/1

#### Шифратор алгоритма S-A5

В алгоритме S-A5 используются три LFSR, которые заданы порождающими полиномами  $P_1(x) = x^4 + x + 1$ ,  $P_2(x) = x^3 + x + 1$ .  $P_3(x) = x^3 + x^2 + 1$ .

Ключевые биты, с помощью которых будет происходить управление синхронизацией генераторов ПСП, выбраны 3 элемент в первом генераторе (обозначим **X**), 2 элемент во втором генераторе (обозначим **Y**) и 3 элемент в третьем генераторе (обозначим **Z**). Для формирования шифрующей гаммы используем старшие разряды генераторов. Схема шифратора S-A5 показана на рисунке 2.

**Мажоритарная функция.** Значение мажоритарной функции (majority) с параметрами (X, Y, Z) равно 1, если значение большинства битов — 1; если это — 0, то ее значение — 0. Например,

$$\text{majority}(1, 0, 1) = 1,$$

$$\text{majority}(0, 0, 1) = 0.$$

Значение мажоритарной функции определяется перед поступлением тактового импульса, согласно правила

$$F = (X \wedge Y) \vee (X \wedge Z) \vee (Z \wedge Y). \quad (1)$$

Составим соотношения и таблицы, описывающие работу генераторов. Отдельно выделим таблицу для формирования синхроимпульса сдвига генераторов ПСП. Данные таблицы заполняются потактово. После создадим и заполним таблицу формирования шифрующей гаммы и зашифрования открытого текста.

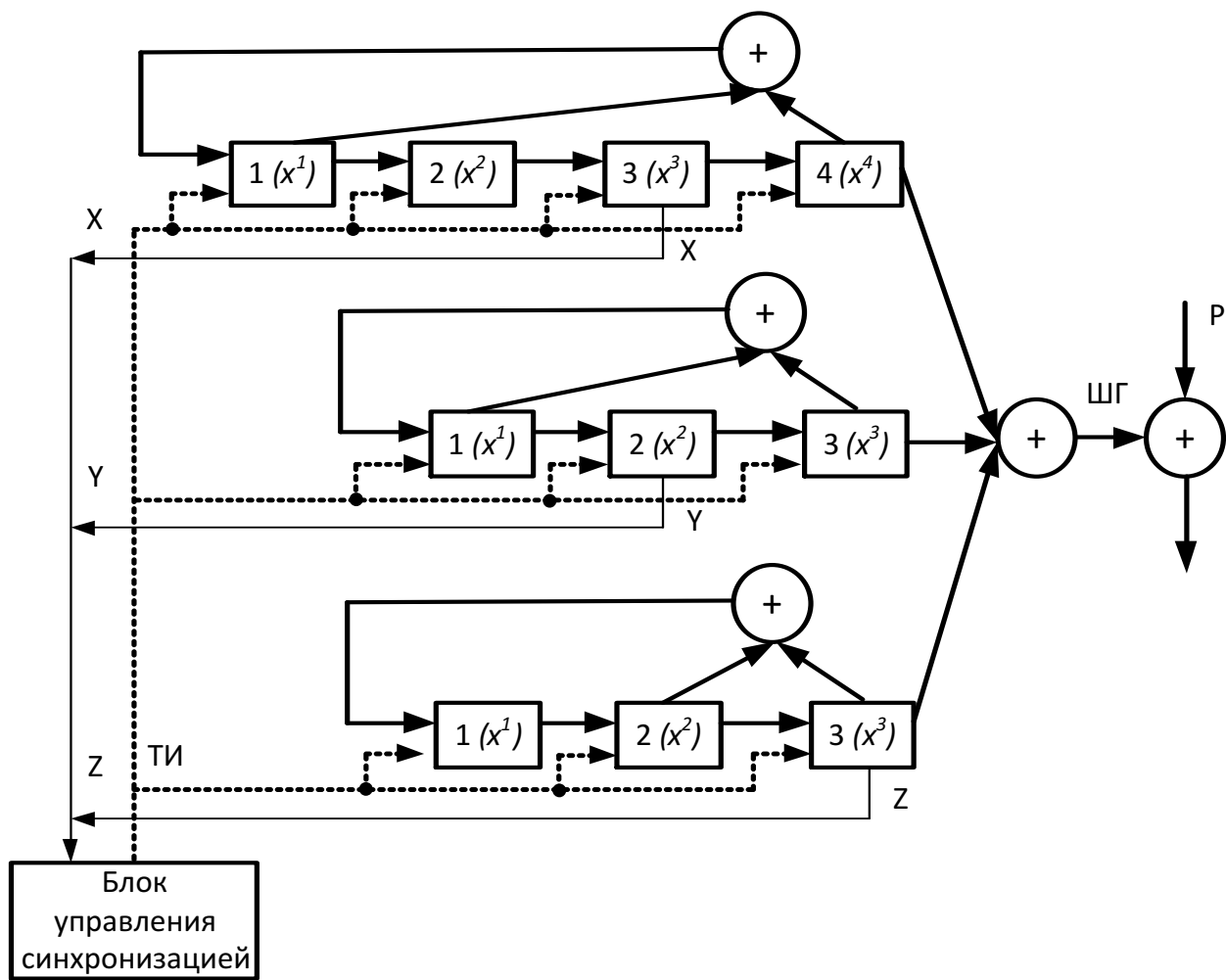


Рисунок 2 – Шифра тор алгоритма S-A5

Составим соотношения, которые описывают работу шифратора.

$$\text{Яч1} = \text{яч.1}^* + \text{яч4}^*$$

$$\text{Яч2} = \text{яч.1}^*$$

$$\text{Яч3} = \text{яч.2}^*$$

$$\text{Яч4} = \text{яч.3}^*$$

$$\text{Яч1} = \text{яч.1}^* + \text{яч3}^*$$

$$\text{Яч2} = \text{яч.1}^*$$

$$\text{Яч3} = \text{яч.2}^*$$

$$\text{Яч1} = \text{яч.2}^* + \text{яч3}^*$$

$$\text{Яч2} = \text{яч.1}^*$$

$$\text{Яч3} = \text{яч.2}^*$$

|   |   |   |      |   |   |   |       |   |   |   |   |       |   |  |  |   |  |
|---|---|---|------|---|---|---|-------|---|---|---|---|-------|---|--|--|---|--|
|   | * |   |      | * |   |   | *     |   |   | * |   |       | * |  |  | * |  |
| № | 1 | 2 | 3(X) | 4 | № | 1 | 2 (Y) | 3 | № | 1 | 2 | 3 (Z) |   |  |  |   |  |
| 0 | 1 | 0 | 0    | 1 | 0 | 1 | 1     | 0 | 0 | 1 | 1 | 0     |   |  |  |   |  |
| 1 | 0 | 1 | 0    | 0 | 1 | 1 | 1     | 0 | 1 | 1 | 1 | 1     |   |  |  |   |  |
| 2 | 0 | 1 | 0    | 0 | 2 | 1 | 1     | 1 | 2 | 0 | 1 | 1     |   |  |  |   |  |
| 3 | 0 | 1 | 0    | 0 | 3 | 0 | 1     | 1 | 3 | 0 | 0 | 1     |   |  |  |   |  |
| 4 | 0 | 1 | 0    | 0 | 4 | 1 | 0     | 1 | 4 | 1 | 0 | 0     |   |  |  |   |  |
| 5 | 0 | 0 | 1    | 0 | 5 | 0 | 1     | 0 | 5 | 0 | 1 | 0     |   |  |  |   |  |
| 6 | 0 | 0 | 0    | 1 | 6 | 0 | 0     | 1 | 6 | 0 | 1 | 0     |   |  |  |   |  |

Формирование синхроимпульса сдвига

Таблица 4

| № | $X \wedge Y$ | $X \wedge Z$ | $Z \wedge Y$ | $F=(X \wedge Y) \vee (X \wedge Z) \vee (X \wedge Z)$ |
|---|--------------|--------------|--------------|------------------------------------------------------|
| 1 | $0 * 1 = 0$  | $0 * 0 = 0$  | $0 * 1 = 0$  | $0 + 0 + 0 = 0$ (сдвиг нулей)                        |
| 2 | $0 * 1 = 0$  | $0 * 1 = 0$  | $1 * 1 = 1$  | $0 + 0 + 1 = 1$ (сдвиг 2 и 3)                        |
| 3 | $0 * 1 = 0$  | $0 * 1 = 0$  | $1 * 1 = 1$  | $0 + 0 + 1 = 1$ (сдвиг 2 и 3)                        |
| 4 | $0 * 1 = 0$  | $0 * 1 = 0$  | $1 * 1 = 1$  | $0 + 0 + 1 = 1$ (сдвиг 2 и 3)                        |
| 5 | $0 * 0 = 0$  | $0 * 0 = 0$  | $0 * 0 = 0$  | $0 + 0 + 0 = 0$ (сдвиг 1, 2 и 3)                     |
| 6 | $1 * 1 = 1$  | $1 * 0 = 0$  | $0 * 1 = 0$  | $1 + 0 + 0 = 1$ (сдвиг 1 и 2)                        |

Составим таблицу, показывающую процесс получения шифрующей гаммы для данного алгоритма, а также процедуру зашифрования. Шифрующая гамма получается путем суммирования по модулю два старших разрядов генераторов.  $P = 11001$

Таблица 5

| № | Яч4 (1) | Яч3(2) | Яч3(3) | ШГ = яч4 + яч3+яч.3 | P | C |
|---|---------|--------|--------|---------------------|---|---|
| 0 | 1       | 0      | 0      | -                   | - | - |
| 1 | 0       | 0      | 1      | 1                   | 1 | 0 |
| 2 | 0       | 1      | 1      | 0                   | 1 | 1 |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 3 | 0 | 1 | 1 | 0 | 0 | 0 |
| 4 | 0 | 1 | 0 | 1 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 1 | 0 | 0 | 1 | 1 |

В результате получен закрытый текст  $C = 010101$

### Задание

Разработать структуру шифратора S-A5 согласно варианта задания, приведенного в таблицах 6-8. Начальное заполнение выбирается произвольно студентом. Зашифровать 10 бит открытой информации.

Таблица 6. Первый генератор

| Вариант           | Полином                            |
|-------------------|------------------------------------|
| 1, 7, 13, 19, 25  | $P(x) = x^5 + x^3 + x^2 + x + 1$   |
| 2, 8, 14, 20, 26  | $P(x) = x^5 + x^4 + x^2 + x + 1$   |
| 3, 9, 15, 21, 27  | $P(x) = x^5 + x^4 + x^3 + x + 1$   |
| 4, 10, 16, 22, 28 | $P(x) = x^5 + x^4 + x^3 + x^2 + 1$ |
| 5, 11, 17, 23, 29 | $P(x) = x^5 + x^3 + 1$             |
| 6, 12, 18, 24, 30 | $P(x) = x^5 + x^2 + 1$             |

Таблица 6. Второй генератор

| Вариант | Полином                | Вариант | Полином                | Вариант | Полином                |
|---------|------------------------|---------|------------------------|---------|------------------------|
| 1       | $P(x) = x^4 + x^3 + 1$ | 11      | $P(x) = x^4 + x^3 + 1$ | 21      | $P(x) = x^4 + x^3 + 1$ |
| 2       | $P(x) = x^4 + x + 1$   | 12      | $P(x) = x^4 + x + 1$   | 22      | $P(x) = x^4 + x + 1$   |
| 3       | $P(x) = x^4 + x^3 + 1$ | 13      | $P(x) = x^4 + x^3 + 1$ | 23      | $P(x) = x^4 + x^3 + 1$ |
| 4       | $P(x) = x^4 + x + 1$   | 14      | $P(x) = x^4 + x + 1$   | 24      | $P(x) = x^4 + x + 1$   |
| 5       | $P(x) = x^4 + x^3 + 1$ | 15      | $P(x) = x^4 + x^3 + 1$ | 25      | $P(x) = x^4 + x^3 + 1$ |
| 6       | $P(x) = x^4 + x + 1$   | 16      | $P(x) = x^4 + x + 1$   | 26      | $P(x) = x^4 + x + 1$   |
| 7       | $P(x) = x^4 + x^3 + 1$ | 17      | $P(x) = x^4 + x^3 + 1$ | 27      | $P(x) = x^4 + x^3 + 1$ |
| 8       | $P(x) = x^4 + x + 1$   | 18      | $P(x) = x^4 + x + 1$   | 28      | $P(x) = x^4 + x + 1$   |
| 9       | $P(x) = x^4 + x^3 + 1$ | 19      | $P(x) = x^4 + x^3 + 1$ | 29      | $P(x) = x^4 + x^3 + 1$ |
| 10      | $P(x) = x^4 + x + 1$   | 20      | $P(x) = x^4 + x + 1$   | 30      | $P(x) = x^4 + x + 1$   |

Таблица 7. Третий генератор

| Вариант | Полином                | Вариант | Полином                | Вариант | Полином                |
|---------|------------------------|---------|------------------------|---------|------------------------|
| 1       | $P(x) = x^4 + x + 1$   | 11      | $P(x) = x^3 + x + 1$   | 21      | $P(x) = x^3 + x^2 + 1$ |
| 2       | $P(x) = x^3 + x + 1$   | 12      | $P(x) = x^4 + x^3 + 1$ | 22      | $P(x) = x^4 + x + 1$   |
| 3       | $P(x) = x^4 + x^3 + 1$ | 13      | $P(x) = x^3 + x^2 + 1$ | 23      | $P(x) = x^3 + x + 1$   |
| 4       | $P(x) = x^3 + x^2 + 1$ | 14      | $P(x) = x^4 + x + 1$   | 24      | $P(x) = x^4 + x^3 + 1$ |

|    |                        |    |                        |    |                        |
|----|------------------------|----|------------------------|----|------------------------|
| 5  | $P(x) = x^4 + x + 1$   | 15 | $P(x) = x^3 + x + 1$   | 25 | $P(x) = x^3 + x^2 + 1$ |
| 6  | $P(x) = x^3 + x + 1$   | 16 | $P(x) = x^4 + x^3 + 1$ | 26 | $P(x) = x^4 + x + 1$   |
| 7  | $P(x) = x^4 + x^3 + 1$ | 17 | $P(x) = x^3 + x^2 + 1$ | 27 | $P(x) = x^3 + x + 1$   |
| 8  | $P(x) = x^3 + x^2 + 1$ | 18 | $P(x) = x^4 + x + 1$   | 28 | $P(x) = x^4 + x^3 + 1$ |
| 9  | $P(x) = x^4 + x + 1$   | 19 | $P(x) = x^3 + x + 1$   | 29 | $P(x) = x^3 + x^2 + 1$ |
| 10 | $P(x) = x^3 + x + 1$   | 20 | $P(x) = x^4 + x^3 + 1$ | 30 | $P(x) = x^4 + x + 1$   |

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

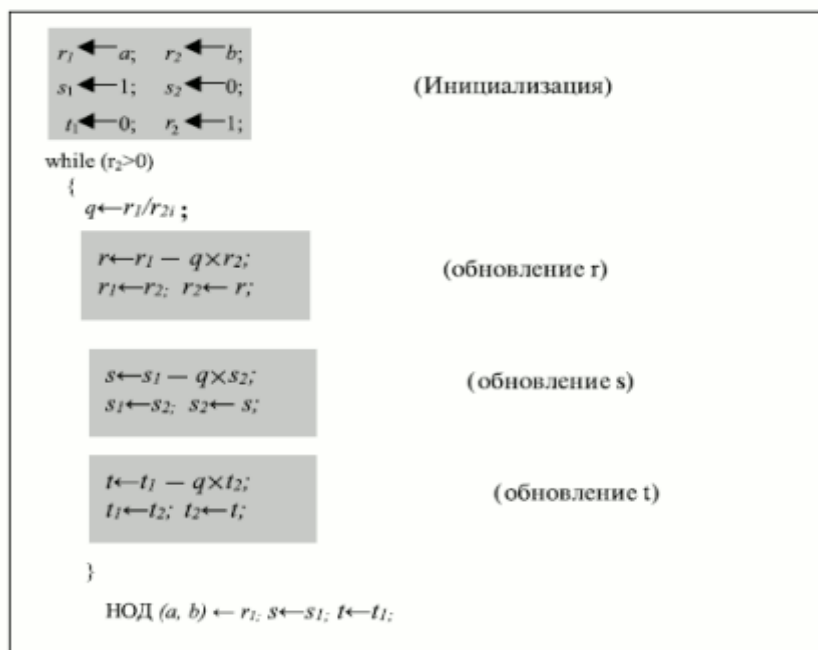
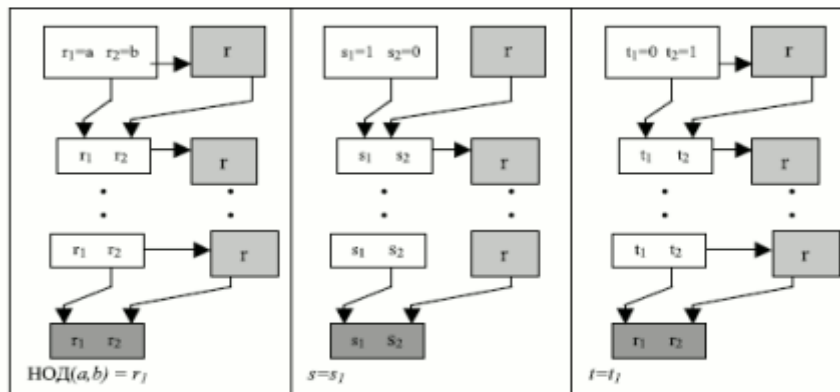
## 6 семестр

### Практическое занятие №11 МОДУЛЬНАЯ АРИФМЕТИКА

Формируемые компетенции: ОПК-6, ОПК-9.

**Расширенный алгоритм Евклида** может вычислить НОД ( $a, b$ ) и в то же самое время вычислить значения  $s$  и  $t$ . Алгоритм и процесс такого вычисления показан на рис. 1.

Здесь расширенный алгоритм Евклида использует те же самые шаги, что и простой алгоритм Евклида. Однако в каждом шаге мы применяем три группы вычислений вместо одной. Алгоритм использует три набора переменных:  $r$ ,  $s$  и  $t$ .



## Рисунок 1 - Расширенный алгоритм Евклида

На каждом шаге переменные  $r_1$ ,  $r_2$  и  $r$  используются так же, как в алгоритме Евклида. Переменным  $r_1$  и  $r_2$  присваиваются начальные значения  $a$  и  $b$  соответственно. Переменным  $s_1$  и  $s_2$  присваиваются начальные значения 1 и 0 соответственно. Переменным  $t_1$  и  $t_2$  присваиваются начальные значения 0 и 1, соответственно. Вычисления  $r$ ,  $s$  и  $t$  одинаковы, но с одним отличием. Хотя  $r$  — остаток от деления  $r_1$  на  $r_2$ , такого соответствия в других двух группах вычислений нет. Есть только одно частное,  $q$ , которое вычисляется как  $r_1/r_2$  и используется для других двух вычислений.

### Пример 1

Дано  $a = 161$  и  $b = 28$ , надо найти НОД ( $a$ ,  $b$ ) и значения  $s$  и  $t$ .

#### Решение

$$r = r_1 - q \times r_2 \quad s = s_1 - q s_2 \quad t = t_1 - q \times t_2$$

Для отображения алгоритма мы используем следующую таблицу:

| q | $r_1$ | $r_2$ | R  | $s_1$ | $s_2$ | s  | $t_1$ | $t_2$ | t   |
|---|-------|-------|----|-------|-------|----|-------|-------|-----|
| 5 | 161   | 28    | 21 | 1     | 0     | 1  | 0     | 1     | -5  |
| 1 | 28    | 21    | 7  | 0     | 1     | -1 | 1     | -5    | 6   |
| 3 | 21    | 7     | 0  | 1     | -1    | 4  | -5    | 6     | -23 |
|   | 7     | 0     |    | -1    | 4     |    | 6     | -23   |     |

Мы получаем НОД  $(161, 28) = 7$ ,  $s = -1$  и  $t = 6$ . Ответы могут быть проверены, как это показано ниже.

$$(-1) \times 161 + 6 \times 28 = 7$$

### Пример 2

Дано  $a = 17$  и  $b = 0$ , найти НОД ( $a$ ,  $b$ ) и значения  $s$  и  $t$ .

#### Решение

Для отображения алгоритма мы используем таблицу.

| q | r <sub>1</sub> | r <sub>2</sub> | R | s <sub>1</sub> | s <sub>2</sub> | s | t <sub>1</sub> | t <sub>2</sub> | t |
|---|----------------|----------------|---|----------------|----------------|---|----------------|----------------|---|
|   | 17             | 0              |   | 1              | 0              |   | 0              | 1              |   |

Обратите внимание, что нам не надо вычислять q, r и s. Первое значение r<sub>2</sub> соответствует условию завершения алгоритма. Мы получаем НОД (17, 0) = 17, s = 1 и t = 0. Это показывает, почему мы должны придавать начальные значения s<sub>1</sub> — 1 и t<sub>1</sub> — 0. Ответы могут быть проверены так, как это показано ниже:

$$(1 \times 17) + (0 \times 0) = 17$$

### Пример 3

Даны a = 0 и b = 45, найти НОД (a, b) и значения s и t.

Решение

Для отображения алгоритма мы используем следующую таблицу:

| q | r <sub>1</sub> | r <sub>2</sub> | R | s <sub>1</sub> | s <sub>2</sub> | s | t <sub>1</sub> | t <sub>2</sub> | t |
|---|----------------|----------------|---|----------------|----------------|---|----------------|----------------|---|
| 0 | 0              | 45             | 0 | 1              | 0              | 1 | 0              | 1              | 0 |
|   | 45             | 0              |   | 0              | 1              |   | 0              | 1              |   |

Мы получаем НОД (0,45) = 45, s = 0 и t = 1. Отсюда ясно, что мы должны инициализировать s<sub>2</sub> равным 0, а t<sub>2</sub> — равным 1. Ответ может быть проверен, как это показано ниже:

$$(0 \times 0) + (1 \times 45) = 45$$

### Линейные диофантовы уравнения

Хотя очень важное приложение расширенного алгоритма Евклида будет рассмотрено далее, здесь мы остановимся на другом приложении — "нахождение решения **линейных диофантовых уравнений** двух переменных", а именно, уравнения  $ax + by = c$ . Мы должны найти значения целых чисел для x и y, которые удовлетворяют этому уравнению. Этот тип уравнения либо не имеет решений, либо имеет бесконечное число решений. Пусть  $d = \text{НОД}(a, b)$ . Если  $d \nmid c$ , то уравнение не имеет решения. Если  $d \mid c$ , то мы имеем бесконечное число решений. Одно из них называется частным, остальные — общими.

**Линейное диофантово уравнение — это уравнение двух переменных:** $ax + by = c$ .

Частное решение

Если  $d|c$ , то можно найти частное решение вышеупомянутого уравнения, используя следующие шаги.

1. Преобразуем уравнение к  $a_1x + b_1y = c_1$ , разделив обе части уравнения на  $d$ . Это возможно, потому, что  $d$  делит  $a$ ,  $b$ , и  $c$  в соответствии с предположением.

2. Найти  $s$  и  $t$  в равенстве  $a_1s + b_1t = 1$ , используя расширенный алгоритм Евклида.

3. Частное решение может быть найдено:

Частное решение:  $x_0 = (c/d)s$  и  $y_0 = (c/d)t$

Общие решения

После нахождения частного решения общие решения могут быть найдены:

Общие решения:  $x = x_0 + k(b/d)$  и  $y = y_0 - k(a/d)$ , где  $k$  — целое число

### Пример 3

Найти частные и общие решения уравнения  $21x + 14y = 35$ .

#### Решение

Мы имеем  $d = \text{НОД}(21, 14) = 7$ . При  $7|35$  уравнение имеет бесконечное число решений. Мы можем разделить обе стороны уравнения на 7 и получим уравнение  $3x + 2y = 5$ . Используя расширенный алгоритм Евклида, мы находим  $s$  и  $t$ , такие, что  $3s + 2t = 1$ . Мы имеем  $s = 1$  и  $t = -1$ . Решения будут следующие:

Частное решение :  $x_0 = 5 \times 1 = 5$  и  $y_0 = 5 \times (-1) = -5$  тогда  $35/7 = 5$

Общие:  $x = 5 + k \times 2$   $y = -5 - k \times 3$  где  $k$  — целое

Поэтому решения будут следующие  $(5, -5), (7, -8), (9, -11)...$

Мы можем легко проверить, что каждое из этих решений удовлетворяет первоначальному уравнению.

### Пример 4

Рассмотрим очень интересное приложение решения диофантовых уравнений в реальной жизни. Мы хотим найти различные комбинации объектов,

имеющих различные значения. Например, мы хотим обменять денежный чек 100\$ на некоторое число банкнот 20\$ и несколько банкнот по 5\$. Имеется много вариантов, которые мы можем найти, решая соответствующее диофантово уравнение  $20x + 5y = 100$ . Обозначим  $d = \text{НОД}(20, 5) = 5$  и  $5|100$ . Уравнение имеет бесконечное число решений, но в этом случае приемлемы только несколько из них (только те ответы, в которых и  $x$  и  $y$  являются неотрицательными целыми числами). Мы делим обе части уравнения на 5, чтобы получить  $4x + y = 20$ , и решаем уравнение  $4s + t = 1$ . Мы можем найти  $s = 0$  и  $t = 1$ , используя расширенный алгоритм Эвклида. Частное решение:  $x_2 = 0 \times 20 = 0$  и  $y_0 = 1 \times 20 = 20$ . Общие решения с неотрицательными  $x$  и  $y$  —  $(0, 20), (1, 16), (2, 12), (3, 8), (4, 4), (5, 0)$ . Остальная часть решений неприемлема, потому что  $y$  становится отрицательным. Кассир в банке должен спросить, какую из вышеупомянутых комбинаций мы хотим. Первое число в скобках обозначает число банкнот по 20\$; второе число обозначает число банкнот по 5\$.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие №12

### АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ

#### Цель и содержание:

1. Углубить знания по основам алгебраических систем, используемых при шифровании.
2. Изучить правила выполнения модульной арифметики

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретическое обоснование

Вышеупомянутый бинарный оператор назван **оператором по модулю** и обозначается как **mod**. Второй вход (**n**) назван **модулем**. Вывод **r** назван **вычетом**. [Рисунок 2.9](#) показывает отношение деления по сравнению с оператором по модулю.

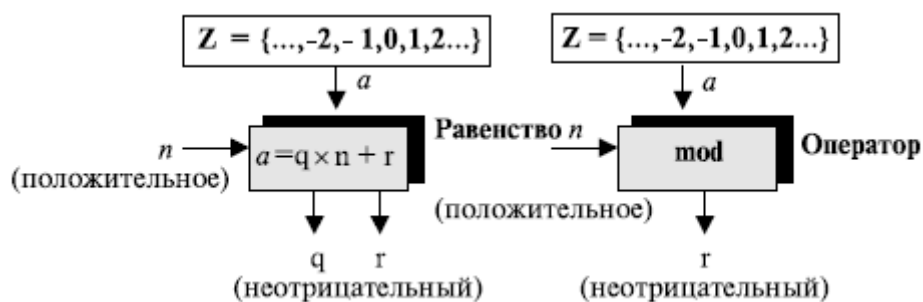


Рисунок 1 - Соотношение уравнения деления и оператора по модулю

Как показано на [рис. 1](#), оператор по модулю (**mod**) выбирает целое число (**a**) из множества  $Z$  и положительный модуль (**n**). Оператор определяет неотрицательный остаток (**r**).

Мы можем сказать, что

$$a \bmod n = r$$

#### Пример 1

Найти результат следующих операций:

- a.  $27 \bmod 5$
- b.  $36 \bmod 12$
- c.  $-18 \bmod 14$

d.  $-7 \bmod 10$

### Решение

Мы ищем вычет  $r$ . Мы можем разделить  $a$  на  $n$  и найти  $q$  и  $r$ . Далее можно игнорировать  $q$  и сохранить  $r$ .

а. Разделим 27 на 5 - результат:  $r = 2$ . Это означает, что  $27 \bmod 5 = 2$ .

б. Разделим 36 на 12 — результат:  $r = 0$ . Это означает, что  $36 \bmod 12 = 0$ .

в. Разделим  $(-18)$  на 14 — результат:  $r = -4$ . Однако мы должны прибавить модуль (14), чтобы сделать остаток неотрицательным. Мы имеем  $r = -4 + 14 = 10$ . Это означает, что  $-18 \bmod 14 = 10$ .

г. Разделим  $(-7)$  на 10 — результат:  $r = -7$ . После добавления модуля  $-7$  мы имеем  $r = 3$ . Это означает, что  $-7 \bmod 10 = 3$ .

Система вычетов:  $Z_n$

Результат операции по модулю  $n$  — всегда целое число между 0 и  $n - 1$ . Другими словами, результат  $a \bmod n$  — всегда неотрицательное целое число, меньшее, чем  $n$ . Мы можем сказать, что операция по модулю создает набор, который в модульной арифметике можно понимать как **систему наименьших вычетов по модулю  $n$** , или  $Z_n$ . Однако мы должны помнить, что хотя существует только одно множество целых чисел ( $Z$ ), мы имеем бесконечное число множеств вычетов ( $Z_n$ ), но лишь одно для каждого значения  $n$ . [Рисунок 2.](#) показывает множество  $Z_n$  и три множества  $Z_2$ ,  $Z_6$  и  $Z_{11}$ .

$$Z_n = \{0, 1, 2, 3, \dots, (n-1)\}$$
$$Z_2 = \{0, 1\} \quad Z_6 = \{0, 1, 2, 3, 4, 5\} \quad Z_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Рисунок 2 - Некоторые наборы  $Z_n$

### Сравнения

В криптографии мы часто используем понятие **сравнения** вместо равенства. Отображение  $Z$  в  $Z_n$  не отображаются "один в один". Бесконечные элементы множества  $Z$  могут быть отображены одним элементом  $Z_n$ . Например, результат  $2 \bmod 10 = 2$ ,  $12 \bmod 10 = 2$ ,  $22 \bmod 10 = 2$ , и так далее. В модульной

арифметике целые числа, подобные 2, 12, и 22, называются сравнимыми по модулю 10 ( $\text{mod } 10$ ). Для того чтобы указать, что два целых числа сравнимы, мы используем **оператор сравнения** ( $\equiv$ ). Мы добавляем  $\text{mod } n$  к правой стороне сравнения, чтобы определить значение модуля и сделать равенство правильным.

Например, мы пишем:

$$\begin{aligned} 2 &\equiv 12 \pmod{10} & 13 &\equiv 23 \pmod{10} & 34 &\equiv 4 \pmod{10} & -8 &\equiv 12 \pmod{10} \\ 3 &\equiv 8 \pmod{5} & 8 &\equiv 13 \pmod{5} & 23 &\equiv 3 \pmod{5} & -8 &\equiv 2 \pmod{5} \end{aligned}$$

[Рисунок 2.11](#) показывает принцип сравнения. Мы должны объяснить несколько положений.

а. Оператор сравнения напоминает оператор равенства, но между ними есть различия. Первое: оператор равенства отображает элемент  $Z$  самого на себя; оператор сравнения отображает элемент  $Z$  на элемент  $Z_n$ . Второе: оператор равенства показывает, что наборы слева и справа соответствуют друг другу "один в один", оператор сравнения — "многие — одному".

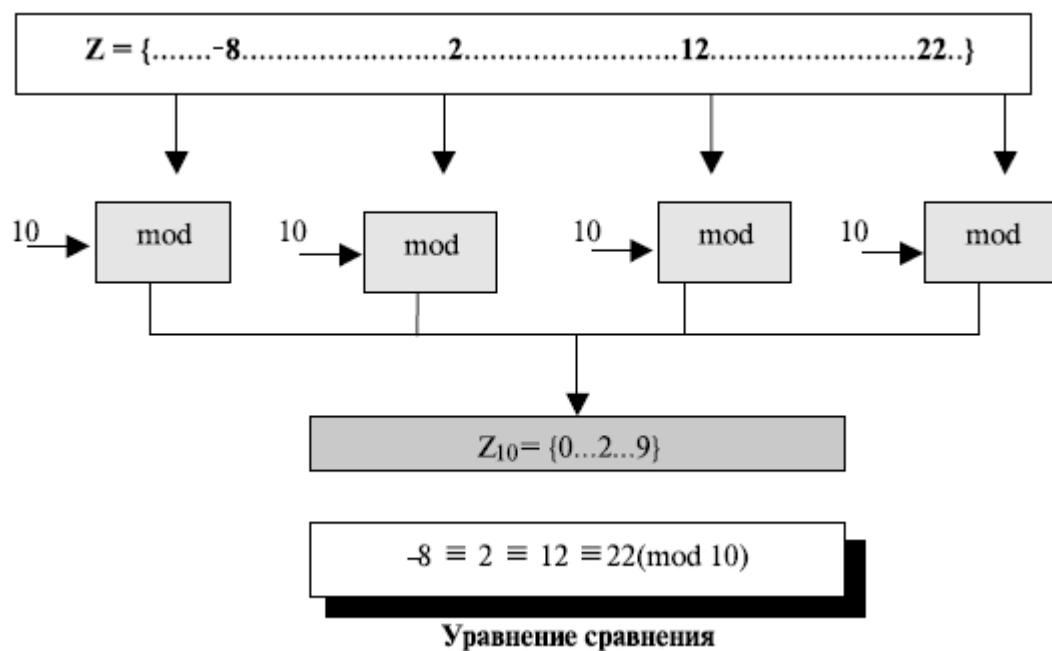


Рисунок 3 - Принцип сравнения

б. Обозначение  $(\text{mod } n)$ , которое мы вставляем с правой стороны оператора сравнения, обозначает признак множества ( $Z_n$ ). Мы должны добавить это

обозначение, чтобы показать, какой модуль используется в отображении. Символ, используемый здесь, не имеет того же самого значения, как бинарный оператор в уравнении деления. Другими словами, символ  $\text{mod}$  в выражении  $12 \text{ mod } 10$  — оператор; а сочетание  $(\text{mod } 10)$  в сравнении  $2 \equiv 12(\text{mod } 10)$  означает, что набор —  $Z_{10}$ .

### Система вычетов

Система вычетов  $[a]$ , или  $[a]_n$ , — множество целых чисел, сравнимых по модулю  $n$ . Другими словами, это набор всех целых чисел, таких, что  $x = a \pmod{n}$ . Например, если  $n = 5$ , мы имеем множество из пяти элементов  $[0]$ ,  $[1]$ ,  $[2]$ ,  $[3]$  и  $[4]$ , таких как это показано ниже:

$$[0] = \{ \dots, -15, 10, -5, 0, 5, 10, 15, \dots \}$$

$$[1] = \{ \dots, -14, -9, -4, 1, 6, 11, 16, \dots \}$$

$$[2] = \{ \dots, -13, -8, -3, 2, 7, 12, 17, \dots \}$$

$$[3] = \{ \dots, -12, -7, -2, 3, 8, 13, 18, \dots \}$$

$$[4] = \{ \dots, 11, -6, -1, 4, 9, 14, 19, \dots \}$$

Целые числа в наборе  $[0]$  все дают остаток 0 при делении на 5 (сравнимы по модулю 5). Целые числа в наборе  $[1]$  все дают остаток 1 при делении на 5 (сравнимы по модулю 5), и так далее. В каждом наборе есть один элемент, называемый наименьшим (неотрицательным) вычетом. В наборе  $[0]$  это элемент 0; в наборе  $[1]$  — 1, и так далее. Набор, который показывает все наименьшие вычеты:  $Z_5 = \{0, 1, 2, 3, 4\}$ . Другими словами, набор  $Z_n$  — набор всех **наименьших вычетов по модулю  $n$** .

### Круговая система обозначений

Понятие "сравнение" может быть лучше раскрыто при использовании круга в качестве модели. Так же, как мы применяем линию, чтобы показать распределение целых чисел в  $Z$ , мы можем использовать круг, чтобы показать распределение целых чисел в  $Z_n$ .

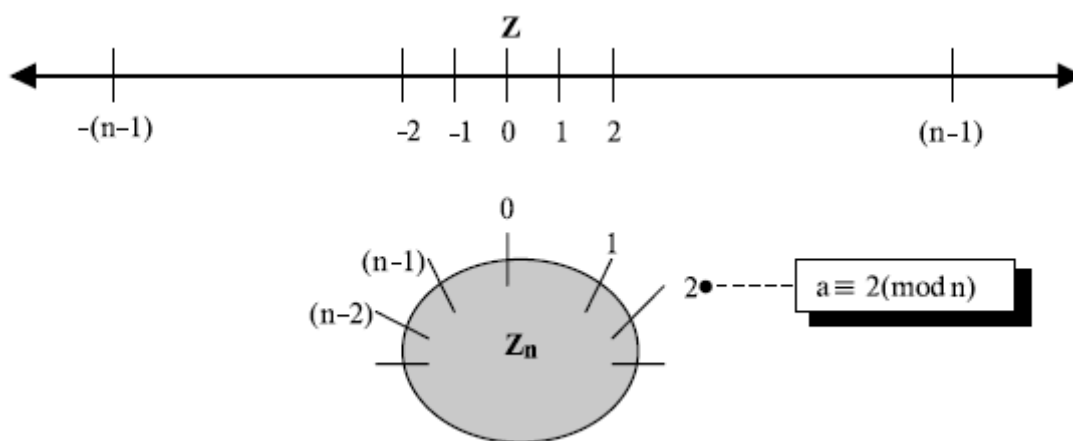


Рисунок 4 - Сравнение использования диаграмм для  $Z$  и  $Z_n$

[Рисунок 4](#) позволяет сравнить два этих подхода. Целые числа от  $0$  до  $n-1$  расположены равномерно вокруг круга. Все целые числа, сравнимые по модулю  $n$ , занимают одни и те же точки в круге. Положительные и отрицательные целые числа от  $Z$  отображаются в круге одним и тем же способом, соблюдая симметрию между ними.

### Пример 2

Мы пользуемся сравнением по модулю в нашей ежедневной жизни; например, мы применяем часы, чтобы измерить время. Наша система часов использует арифметику по модулю 12. Однако вместо  $0$  мы берем отсечку 12, так что наша система часов начинается с  $0$  (или 12) и идет до 11. Поскольку наши сутки делятся 24 часа, мы считаем по кругу два раза и обозначаем первое вращение как утро до полудня, а второе — как вечер после полудня.

### Вопрос 2 Операции в $Z_n$

Три бинарных операции (сложение, вычитание и умножение), которые мы обсуждали для  $Z$ , могут также быть определены для набора  $Z_n$ . Результат, возможно, должен быть отображен в  $Z_n$  с использованием операции по модулю, как это показано на [рис. 5](#).

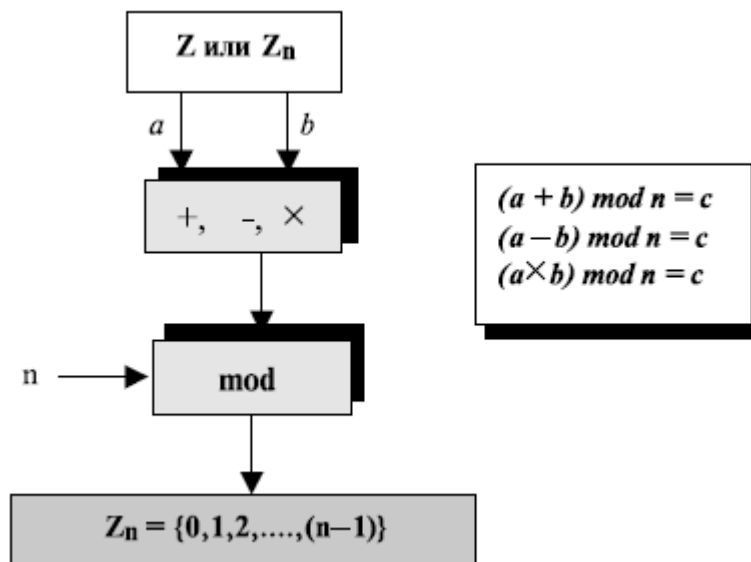


Рисунок 5 - Бинарные операции в  $Z_n$

Фактически применяются два набора операторов: первый набор — один из бинарных операторов  $(+, -, \times)$ ; второй — операторы по модулю. Мы должны использовать круглые скобки, чтобы подчеркнуть порядок работ. Как показано на [рис. 5](#), входы ( $a$  и  $b$ ) могут быть членами  $Z$  или  $Z_n$ .

### Пример 3

Выполните следующие операторы (поступающие от  $Z_n$ ):

- а. Сложение 7 и 14 в  $Z_{15}$
- б. Вычитание 11 из 7 в  $Z_{13}$
- в. Умножение 11 на 7 в  $Z_{20}$

### Решение

Ниже показаны два шага для каждой операции:

$$(14+7) \bmod 15 \rightarrow (21) \bmod 15 = 6$$

$$(7-11) \bmod 13 \rightarrow (-4) \bmod 13 = 9$$

$$(7 \times 11) \bmod 20 \rightarrow (77) \bmod 20 = 17$$

### Пример 4

Выполните следующие операции (поступающие от  $Z_n$ ):

- а. Сложение 17 и 27 в  $Z_{14}$
- б. Вычитание 43 из 12 в  $Z_{13}$

с. Умножение 123 на -10 в  $Z_{19}$

### Решение

Ниже показаны два шага для каждой операции:

$$(17 + 27) \bmod 14 \rightarrow (44) \bmod 14 = 2$$

$$(12 - 43) \bmod 13 \rightarrow (-31) \bmod 13 = 8$$

$$((123) \times (-10)) \bmod 19 \rightarrow (-1230) \bmod 19 = 5$$

### Свойства

Мы уже упоминали, что два входа для трех бинарных операторов в сравнении по модулю могут использовать данные из  $Z$  или  $Z_n$ . Следующие свойства позволяют нам сначала отображать два входа к  $Z_n$  (если они прибывают от  $Z$ ) перед выполнением этих трех бинарных операторов  $(+, -, \times)$ . Заинтересованные читатели могут найти доказательства для этих свойств в приложении Q.

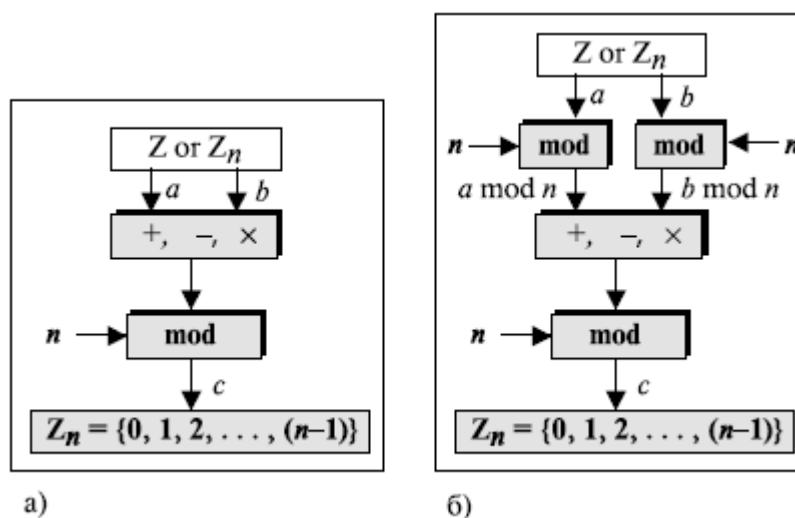


Рисунок 6 - Свойства оператора mod

**Первое свойство:**  $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$

**Второе свойство:**  $(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$

**Третье свойство:**  $(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

[Рисунок 2.14](#) показывает процесс до и после применения указанных выше свойств. Хотя по рисунку видно, что процесс с применением этих свойств более длинен, мы должны помнить, что в криптографии мы имеем дело с очень

большими целыми числами. Например, если мы умножаем очень большое целое число на другое очень большое целое число, которое настолько большое, что не может быть записано в компьютере, то применение вышеупомянутых свойств позволяет уменьшить первые два операнда прежде, чем начать умножение. Другими словами, перечисленные свойства позволяют нам работать с меньшими числами. Этот факт станет понятнее при обсуждении экспоненциальных операций в последующих лекциях.

### Пример 5

Следующие примеры показывают приложение вышеупомянутых свойств.

1.  $(1723345 + 2124945) \bmod 11 = (8 + 9) \bmod 11 = 6$
2.  $(1723345 - 2124945) \bmod 11 = (8 - 9) \bmod 11 = 10$
3.  $(1723345 \times 2124945) \bmod 11 = (8 \times 9) \bmod 11 = 6$

### Пример 6

В арифметике мы часто должны находить остаток от степеней числа 10 при делении на целое число. Например, мы должны найти  $10 \bmod 3$ ,  $10^2 \bmod 3$ ,  $10^3 \bmod 3$ , и так далее. Мы также должны найти  $10 \bmod 7$ ,  $10^2 \bmod 7$ ,  $10^3 \bmod 7$ , и так далее. Третье свойство модульных операторов, упомянутое выше, делает жизнь намного проще.

$$10^n \bmod x = (10 \bmod x)^n \quad \text{Применение третьего свойства } n \text{ раз.}$$

Мы имеем

$$10 \bmod 3 = 1 \rightarrow 10^n \bmod 3 = (10 \bmod 3)^n = 1$$

$$10 \bmod 9 = 1 \rightarrow 10^n \bmod 9 = (10 \bmod 9)^n = 1$$

$$10 \bmod 7 = 3 \rightarrow 10^n \bmod 7 = (10 \bmod 7)^n = 3^n \bmod 7$$

### Пример 7

Мы уже говорили, что в арифметике остаток от целого числа, разделенного на 3, такой же, как остаток от суммы деления его десятичных цифр. Другими словами, остаток от деления 6371 равен остатку от деления суммы его цифр (17), на 3. Мы можем доказать, что это утверждение использует свойства модульного оператора. Запишем целое число как сумму его цифр, умноженных на степени 10.

$$a = a_n 10^n + \dots + a_1 10^1 + a_0 10^0$$

$$\text{Например: } 6371 = 6 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 1 \times 10^0$$

Теперь мы можем применить модульную операцию к двум сторонам равенства и использовать результат предыдущего примера, где остаток  $10^n \bmod 3$  равен 1.

$$\begin{aligned} a \bmod 3 &= (a_n \times 10^n + \dots + a_1 \times 10^1 + a_0 \times 10^0) \bmod 3 \\ &= (a_n \times 10^n) \bmod 3 + \dots + (a_1 \times 10^1) \bmod 3 + (a_0 \times 10^0 \bmod 3) \bmod 3 \\ &= (a_n \bmod 3) \times (10^n \bmod 3) + \dots + (a_1 \bmod 3) \times (10^1 \bmod 3) + \\ &\quad (a_0 \bmod 3) \times (10^0 \bmod 3) \bmod 3 \\ &= (a_n \bmod 3) + \dots + (a_1 \bmod 3) + (a_0 \bmod 3) \bmod 3 \\ &= (a_n + \dots + a_1 + a_0) \bmod 3 \end{aligned}$$

### Инверсии

Когда мы работаем в модульной арифметике, нам часто нужно найти операцию, которая позволяет вычислить величину, обратную заданному числу. Мы обычно ищем **аддитивную инверсию** (оператор, обратный сложению) или **мультипликативную инверсию** (оператор, обратный умножению).

#### Аддитивная инверсия

В  $Z_2$  два числа  $a$  и  $b$  **аддитивно инверсны** друг другу, если  $b = n - a$ .

Например,

$$a + b \equiv 0 \pmod{n}$$

В  $Z_n$  аддитивная инверсия числу  $a$  может быть вычислена как  $b = n - a$ .

Например, аддитивная инверсия 4 в  $Z_{10}$  равна  $10 - 4 = 6$ .

**В модульной арифметике каждое целое число имеет аддитивную инверсию. Сумма целого числа и его аддитивной инверсии сравнима с 0 по модулю  $n$ .**

Обратите внимание, что в модульной арифметике каждое число имеет аддитивную инверсию, и эта инверсия уникальна; каждое число имеет одну и только одну аддитивную инверсию. Однако инверсия числа может быть непосредственно тем же самым числом.

### Пример 8

Найдите все взаимно обратные пары по сложению в  $Z_{10}$ .

### Решение

Даны шесть пар аддитивных инверсий —  $(0, 0)$ ,  $(1, 9)$ ,  $(2, 8)$ ,  $(3, 7)$ ,  $(4, 6)$  и  $(5, 5)$ . В этом списке 0 — инверсия самому себе; так же и 5. Обратите внимание: аддитивные инверсии сложения обратны друг другу; если 4 — аддитивная инверсия 6, тогда 6 — также аддитивная инверсия числу 4.

Мультипликативная инверсия

В  $Z_n$  два числа  $a$  и  $b$  мультипликативно инверсны друг другу, если

$$a \times b \equiv 1 \pmod{n}$$

Например, если модуль равен 10, то мультипликативная инверсия 3 есть 7.

Другими словами, мы имеем  $(3 \times 7) \bmod 10 \equiv 1$ .

**В модульной арифметике целое число может или не может иметь мультипликативную инверсию. Целое число и его мультипликативная инверсия сравнимы с 1 по модулю  $n$ .**

Может быть доказано, что  $a$  имеет мультипликативную инверсию в  $Z_n$ , если только  $\text{НОД}(n, a) = 1$ . В этом случае говорят, что  $a$  и  $n$  **взаимно простые**.

### Пример 9

Найти мультипликативную инверсию 8 в  $Z_{10}$ .

### Решение

Мультипликативная инверсия не существует, потому что  $\text{НОД}(10, 8) = 2 \neq 1$ . Другими словами, мы не можем найти число между 0 и 9, такое, что при умножении на 8 результат сравним с 1 по mod 10.

### Пример 10

Найти все мультипликативные инверсии в  $Z_{10}$ .

### Решение

Есть только три пары, удовлетворяющие условиям существования мультипликативной инверсии:  $(1, 1)$ ,  $(3, 7)$  и  $(9, 9)$ . Числа 0, 2, 4, 5, 6 и 8 не имеют мультипликативной инверсии.

Мы можем проверить, что

$$(1 \times 1) \bmod 10 = 1 \quad (3 \times 7) \bmod 10 = 1 \quad (9 \times 9) \bmod 10 = 1$$

### Пример 11

Найти все мультипликативные обратные пары в  $Z_{11}$ .

#### Решение

Мы имеем семь пар: (1, 1), (2, 6), (3, 4), (5, 9), (7, 8), (9, 9) и (10, 10). При переходе от  $Z_{10}$  к  $Z_{11}$  число пар увеличивается. При  $Z_{11}$  НОД (11, a) = 1 (взаимно простые) для всех значений a, кроме 0. Это означает, что все целые числа от 1 до 10 имеют мультипликативные инверсии.

**Целое число a в  $Z_n$  имеет мультипликативную инверсию тогда и только тогда, если НОД (n, a) = 1(mod n)**

Расширенный алгоритм Евклида, который мы обсуждали ранее в этой лекции, может найти мультипликативную инверсию b в  $Z_n$ , когда даны n и b и инверсия существует. Для этого нам надо заменить первое целое число a на n (модуль). Далее мы можем утверждать, что алгоритм может найти s и t, такие, что  $s \times n + b \times t = \text{НОД}(n, b)$ . Однако если мультипликативная инверсия b существует, НОД (n, b) должен быть 1. Так что уравнение будет иметь вид

$$(s \times n) + (b \times t) = 1$$

Теперь мы применяем операции по модулю к обеим сторонам уравнения. Другими словами, мы отображаем каждую сторону к  $Z_n$ . Тогда мы будем иметь

$$(s \times n + b \times t) \bmod n = 1 \bmod n$$

$$[(s \times n) \bmod n] + [(b \times t) \bmod n] = 1 \bmod n$$

$$0 + [(b \times t) \bmod n] = 1$$

$(b \times t) \bmod n = 1 \rightarrow$  Это означает, что t — это мультипликативная инверсия b в  $Z_n$

Обратите внимание, что  $[(s \times n) \bmod n]$  на третьей строке — 0, потому что, если мы делим  $(s \times n)n$ , частное — s, а остаток — 0.

**Расширенный алгоритм Евклида находит мультипликативные инверсии b в  $Z_n$ , когда даны n и b и НОД (n, b) = 1. Мультипликативная инверсия b — это значение t, отображенное в  $Z_n$ .**

[Рисунок 2.15](#) показывает, как мы находим мультипликативную инверсию числа, используя расширенный алгоритм Евклида.

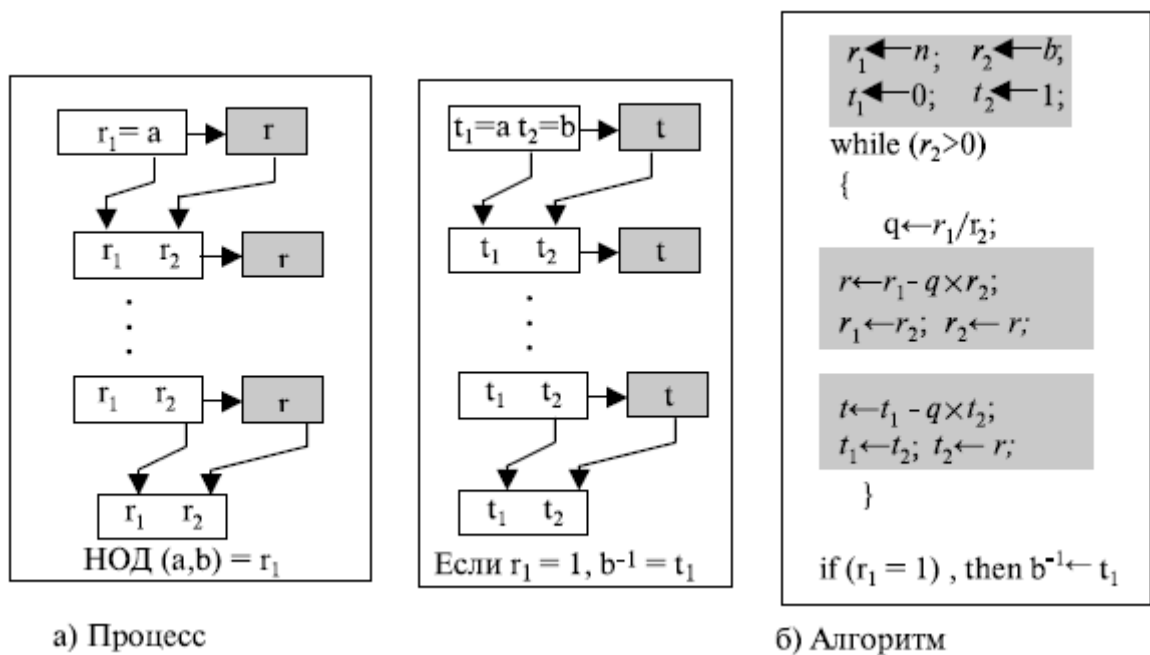


Рисунок 7 - Применение расширенного алгоритма Евклида для поиска мультипликативной инверсии

### Пример 12

Найти мультипликативную инверсию 11 в  $Z_{26}$ .

#### Решение

Мы используем таблицу, аналогичную одной из тех, которые мы уже применяли прежде при данных  $r_1 = 26$  и  $r_2 = 11$ . Нас интересует только значение  $t$ .

| q | r <sub>1</sub> | r <sub>2</sub> | r | t <sub>1</sub> | t <sub>2</sub> | t  |
|---|----------------|----------------|---|----------------|----------------|----|
| 2 | 26             | 11             | 4 | 0              | 1              | -2 |
| 2 | 11             | 4              | 3 | 1              | -2             | 5  |
| 1 | 4              | 3              | 1 | -2             | 5              | -7 |
| 3 | 3              | 1              | 0 | 5              | -7             | 26 |
|   | 1              | 0              |   | -7             | 26             |    |

НОД (26, 11) = 1, что означает, что мультипликативная инверсия 11 существует. Расширенный алгоритм Евклида дает  $t_1 = (-7)$ .

Мультипликативная инверсия равна  $(-7) \bmod 26 = 19$ . Другими словами, 11 и 19 — мультипликативная инверсия в  $Z_{19}$ . Мы можем видеть, что  $(11 \times 19) \bmod 26 = 209 \bmod 26 = 1$ .

### Пример 13

Найти мультипликативную инверсию 23 в  $Z_{100}$ .

### Решение

Мы используем таблицу, подобную той, которую применяли до этого при  $r_1 = 100$  и  $r_2 = 23$ . Нас интересует только значение  $t$ .

| q | r <sub>1</sub> | r <sub>2</sub> | r | t <sub>1</sub> | t <sub>2</sub> | t   |
|---|----------------|----------------|---|----------------|----------------|-----|
| 4 | 100            | 23             | 8 | 0              | 1              | -4  |
| 2 | 23             | 8              | 7 | 1              | -4             | 19  |
| 1 | 8              | 7              | 1 | -4             | 9              | -13 |
| 7 | 7              | 1              | 0 | 9              | -13            | 100 |
|   | 1              | 0              |   | 13             | 100            |     |

НОД (100, 23) — 1, что означает, что инверсия 23 существует. Расширенный Евклидов алгоритм дает  $t_1 = -13$ . Инверсия —  $(-13) \bmod 100 = 87$ . Другими словами, 13 и 87 — мультипликативные инверсии в  $Z_{100}$ . Мы можем видеть, что  $(23 \times 87) \bmod 100 = 2001 \bmod 100 = 1$ .

### Пример 14

Найти инверсию 12 в  $Z_{26}$ .

### Решение

Мы используем таблицу, подобную той, которую мы применяли раньше при  $r_1 = 26$  и  $r_2 = 12$ .

| q | r <sub>1</sub> | r <sub>2</sub> | r | t <sub>1</sub> | t <sub>2</sub> | t |
|---|----------------|----------------|---|----------------|----------------|---|
| 2 | 26             | 12             | 2 | 0              | 1              |   |
| 6 | 12             | 2              | 0 | 1              | -2             |   |
|   | 2              | 0              |   | -2             | 13             |   |

$\text{НОД}(26, 12) = 2 \neq 1$ , что означает отсутствие для числа 12 мультипликативной инверсии в  $Z_{26}$

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие №13

### РАСШИРЕНИЕ ПОЛЕЙ ГАЛУА $GF(2^m)$

#### Цель и содержание:

1. Углубить знания по основам построения алгебраических систем, используемых при шифровании.
2. Изучить правила получения алгебраических структур группа, кольцо и поле

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретический материал

##### Группы

Следующие лекции посвящены обсуждению современных симметрично-ключевых блочных шифров, которые выполняют операции с  $n$ -битовыми словами. Понимание и анализ этих шифров требуют некоторого знания разделов современной алгебры, называемых алгебраическими структурами. В начале этой лекции делается обзор алгебраических структур, а затем показывается, как выполнить сложение или умножение с  $n$ -битовыми словами.

В лекциях мы обсуждали некоторые множества чисел, таких как  $\mathbb{Z}$ ,  $\mathbb{Z}_n$ ,  $\mathbb{Z}_n^*$ ,  $\mathbb{Z}_p$  и  $\mathbb{Z}_p^*$ . Криптография требует, чтобы были заданы множества целых чисел, и операции, определенные для них. Комбинация множеств и операций, которые могут быть применены к элементам множества, называются **алгебраической структурой**. В этой лекции мы определим три общих алгебраических структуры: группы, кольца и поля ([рис. 11.1](#)).



Рисунок 1 - Общие алгебраические структуры

## Группы

Группа ( $G$ ) — набор элементов с бинарной операцией " $\bullet$ " обладает четырьмя свойствами (или удовлетворяет аксиомам), которые будут перечислены ниже.

**Коммутативная группа**, также называемая **абелева**, — группа, в которой оператор обладает теми же четырьмя свойствами для групп плюс дополнительным — коммутативностью. Эти пять свойств определены ниже

- **Замкнутость.** Если  $a$  и  $b$  — элементы  $G$ , то  $c = a \bullet b$  — также элемент  $G$ . Это означает, что результат применения операции с любыми двумя элементами множества есть элемент этого множества.
- **Ассоциативность.** Если  $a$ ,  $b$  и  $c$  — элементы  $G$ , то верно  $(a \bullet b) \bullet c = a \bullet (b \bullet c)$ . Другими словами, не имеет значения, в каком порядке мы применяем операцию более чем с двумя элементами.
- **Коммутативность.** Для всех  $a$  и  $b$  в  $G$  мы имеем  $a \bullet b = b \bullet a$ . Обратите внимание, что это свойство должно быть верно только для коммутативной группы.
- **Существование нейтрального элемента.** Для все элементов в  $G$  существует элемент  $e$ , который называется нейтральным элементом, такой, что  $e \bullet a = a \bullet e = a$ .
- **Существование инверсии.** Для каждого  $a$  в  $G$  существует элемент  $a'$ , называемый инверсией, такой, что  $a \bullet a' = a' \bullet a = e$ .

[Рисунок 11.2](#) иллюстрирует понятие группы.

## Приложение

Хотя группа включает единственный оператор, свойства, присущие каждой операции, позволяют использование пары операций, если они — инверсии друг друга. Например, если определенный выше оператор — сложение, то группа поддерживает и сложение, и вычитание, ибо вычитание и сложение — аддитивно инверсные операции. Это также верно для умножения и деления. Однако группа может поддерживать только сложение/вычитание или умножение/деление, но не оба сочетания операторов одновременно.

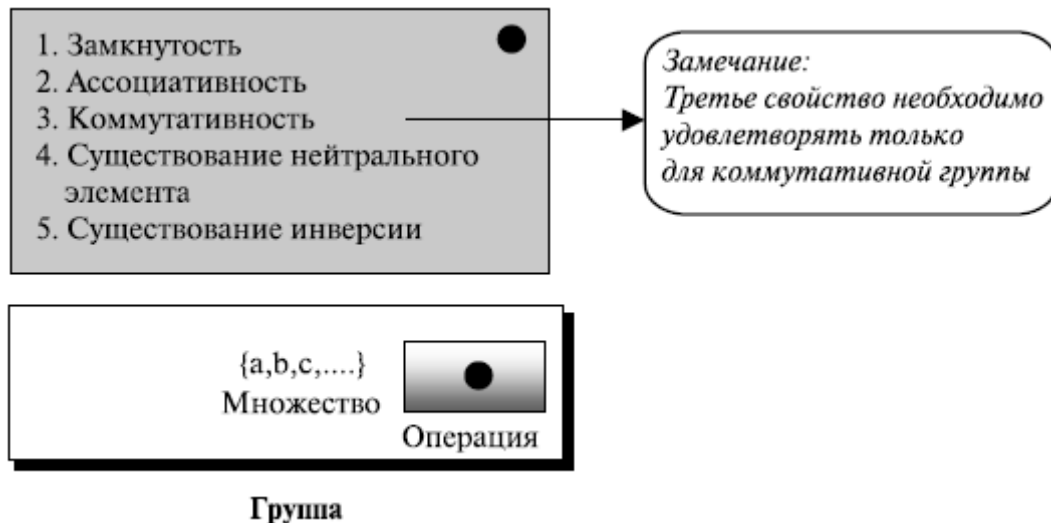


**Рис. 11.2.** Группа

### Пример 11.1

Множество целых чисел, входящих в вычет с оператором сложения,  $G = \langle \mathbb{Z}_n, + \rangle$ , является коммутативной группой. Мы можем выполнить сложение и вычитание на элементах этого множества, не выходя за его пределы.

Проверим эти свойства.

1. Замкнутость удовлетворяется. Результат сложения двух целых чисел в  $\mathbb{Z}_n$  — другое целое число в  $\mathbb{Z}_n$ .
2. Ассоциативность удовлетворяется. Результат  $4 + (3 + 2)$  тот же самый, что в случае  $(4 + 3) + 2$ .
3. Коммутативность удовлетворяется. Мы имеем  $3 + 5 = 5 + 3$ .
4. Нейтральный элемент — 0. Мы имеем  $3 + 0 = 0 + 3 = 3$ .
5. Каждый элемент имеет аддитивную инверсию. Инверсия элемента — его дополнение. Например, инверсия 3 — это  $-3$  ( $n - 3$  в  $\mathbb{Z}_n$ ), и инверсия  $-3$  — это 3. Инверсия позволяет нам выполнять вычитание на множестве.

### Пример 11.2

Множество  $\mathbb{Z}_n^*$  с оператором умножения  $G = \langle \mathbb{Z}_n^*, \cdot \rangle$ , является также абелевой группой. Мы можем выполнить умножение и деление на элементах этого множества, не выходя за его пределы. Это облегчает проверку первых трех

свойств. Нейтральный элемент равен 1. Каждый элемент имеет инверсию, которая может быть найдена согласно расширенному алгоритму Евклида.

### Пример 11.3

Хотя мы обычно представляем группу как множество чисел с обычными операторами, такими, как сложение или вычитание, определения группы позволяют нам определять любое множество объектов и операций, которые удовлетворяют вышеупомянутым свойствам. Определим множество  $G = \{a, b, c, d\}$ ,  $\bullet$  и операцию, показанную с помощью [таблицы 11.1](#).

Таблица 11.1. Таблица операции для примера 11.3

|          | <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> |
|----------|----------|----------|----------|----------|
| <b>a</b> | a        | b        | c        | d        |
| <b>b</b> | b        | c        | d        | a        |
| <b>c</b> | c        | d        | a        | b        |
| <b>d</b> | d        | a        | b        | c        |

Это — абелева группа. Все пять свойств удовлетворены.

1. Замкнутость удовлетворена. Применение оператора на любой паре элементов дает в результате другой элемент этого множества.

2. Ассоциативность также удовлетворена. Чтобы доказать это, мы должны проверить свойство для любой комбинации из трех элементов. Например,  $(a + b) + c = a + (b + c) = d$ .

3. Операция коммутативна. Мы имеем  $a + b = b + a$ .

4. Группа имеет нейтральный элемент, которым является a.

5. Каждый элемент имеет инверсию. Обратные пары могут быть найдены. В таблице они указаны теньевыми элементами в каждой строке. Пары — (a, a), (b, d), (c, c).

### Пример 11.4

В группе элементы в множестве не обязательно должны быть числами или объектами; они могут быть правилами, отображениями, функциями или действиями. Очень интересная группа — **группа подстановок**. Множество всех

перестановок и оператор является композицией: применения одной перестановки за другой. [Рисунок 11.3](#) показывает композиции двух перестановок, которые перемещают три входных сигнала, чтобы создать три выходных сигнала.

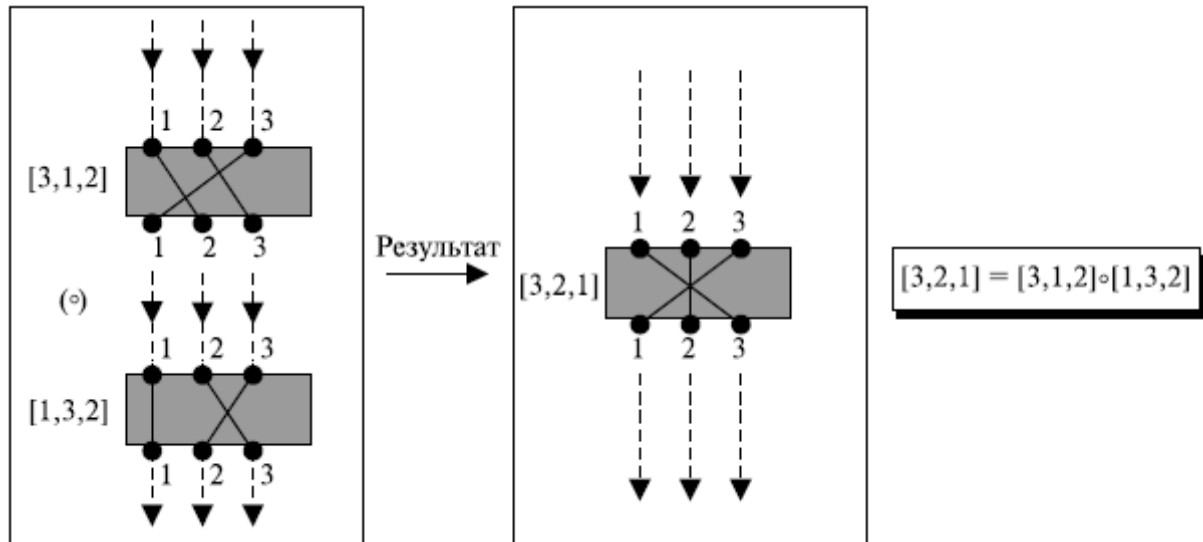


**Рис. 11.3.** Композиции перестановок (Пример 11.4)

Входные сигналы и выходные сигналы могут быть символами (лекции 2-3) или битами. Мы изобразили каждую перестановку прямоугольником, внутри которого показано, где исходящий входной сигнал и индекс (1,2,3) определяет выходной сигнал. Композиция состоит из двух перестановок одна за другой. При трех входных сигналах и трех выходных сигналах может быть  $3!$  или 6 различных перестановок. [Таблица 11.2](#) дает определение этого оператора. Первая строка — первая перестановка; первый столбец — вторая перестановка. Результат содержится на пересечении.

В этом случае удовлетворены только четыре свойства; поэтому группа — не абелева.

1. Замкнутость удовлетворена.
2. Ассоциативность также удовлетворена. Чтобы доказать это, мы должны проверить свойство для любой комбинации из трех элементов.
3. Свойство коммутативности не удовлетворено. Это может быть легко проверено, но мы оставим это для упражнения.

4. Множество имеет нейтральный элемент  $[1\ 2\ 3]$  (перестановка отсутствует). Эти элементы показаны другим цветом.

5. Каждый элемент имеет инверсию. Обратные пары могут быть найдены, если использовать нейтральные элементы.

Таблица 11.2. Таблица операции для группы перестановок

|             | $[1\ 2\ 3]$ | $[1\ 3\ 2]$ | $[2\ 1\ 3]$ | $[2\ 3\ 1]$ | $[3\ 1\ 2]$ | $[3\ 2\ 1]$ |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| $[1\ 2\ 3]$ | $[1\ 2\ 3]$ | $[1\ 3\ 2]$ | $[2\ 1\ 3]$ | $[2\ 3\ 1]$ | $[3\ 1\ 2]$ | $[3\ 2\ 1]$ |
| $[1\ 3\ 2]$ | $[1\ 3\ 2]$ | $[1\ 2\ 3]$ | $[2\ 3\ 1]$ | $[2\ 1\ 3]$ | $[3\ 2\ 1]$ | $[3\ 1\ 2]$ |
| $[2\ 1\ 3]$ | $[2\ 1\ 3]$ | $[3\ 1\ 2]$ | $[1\ 2\ 3]$ | $[3\ 2\ 1]$ | $[1\ 3\ 2]$ | $[2\ 3\ 1]$ |
| $[2\ 3\ 1]$ | $[2\ 3\ 1]$ | $[3\ 2\ 1]$ | $[1\ 3\ 2]$ | $[3\ 1\ 2]$ | $[1\ 2\ 3]$ | $[2\ 1\ 3]$ |
| $[3\ 1\ 2]$ | $[3\ 1\ 2]$ | $[2\ 1\ 3]$ | $[3\ 2\ 1]$ | $[1\ 2\ 3]$ | $[2\ 3\ 1]$ | $[1\ 3\ 2]$ |
| $[3\ 2\ 1]$ | $[3\ 2\ 1]$ | $[2\ 3\ 1]$ | $[3\ 1\ 2]$ | $[1\ 3\ 2]$ | $[2\ 1\ 3]$ | $[1\ 2\ 3]$ |

### Пример 11.5

В предыдущем примере мы показали, что множество перестановок с композицией операций — группа. Поэтому использование двух перестановок (одна за другой) не могут усилить безопасность шифра. Мы сможем всегда найти перестановку, которая может сделать ту же самую операцию, используя свойства замкнутости.

### Конечная группа

Группа называется **конечной группой**, если множество имеет конечное число элементов; иначе это — **бесконечная группа**.

### Порядок группы

**Порядок группы**,  $G$ , — это число элементов в группе. Если группа не конечна, ее порядок бесконечен; если конечна, порядок конечен.

### Подгруппы

Подмножество  $H$  группы  $G$  — **подгруппа**  $G$ , если само  $H$  — группа относительно операции на  $G$ . Другими словами, если  $G = \langle S, \bullet \rangle$  — группа, то  $H = \langle T, \bullet \rangle$  — группа для той же самой операции, и  $T$  — непустое подмножество  $S$ , то  $H$  — подгруппа  $G$ . Вышеупомянутое определение подразумевает, что:

1. если  $a$  и  $b$  — члены обеих групп, то  $c = a \cdot b$  — также элемент обеих групп;
2. для группы и подгруппы имеется один и тот же нейтральный элемент;
3. если этот элемент принадлежит обеим группам, инверсия  $a$  — также элемент обеих групп;
4. группа, полученная с помощью нейтрального элемента  $G$ ,  $H = \langle \{e\}, \cdot \rangle$ , является подгруппой  $G$ ;
5. каждая группа — подгруппа самой себя.

### Пример 11.6

Является ли группа  $H = \langle \mathbb{Z}_{10}, + \rangle$  подгруппой группы  $G = \langle \mathbb{Z}_{12}, + \rangle$ ?

Решение

Ответ — нет. Хотя  $H$  — подмножество  $G$ , операции, определенные для этих двух групп, различны. Операция  $H$  — сложение по модулю 10; операция в  $G$  — сложение по модулю 12.

Циклические подгруппы

Если подгруппа группы может быть сгенерирована, используя возведение в степень элемента, то такая подгруппа называется **циклической подгруппой**. Термин возведение в степень здесь означает многократное применение к элементу групповой операции:

$$a^n \rightarrow a \bullet a \bullet \dots \bullet a (\text{n раз})$$

Множество, полученное в результате этого процесса, обозначается в тексте как  $\langle a \rangle$ . Обратите внимание также, что  $a^0 = e$ .

### Пример 11.7

Из группы  $G = \langle \mathbb{Z}_6, + \rangle$  могут быть получены четыре циклических подгруппы. Это  $H_1 = \langle \{0\}, + \rangle$ ,  $H_2 = \langle \{0, 2, 4\}, + \rangle$ ,  $H_3 = \langle \{0, 3\}, + \rangle$  и  $H_4 = G$ . Заметим, что когда операция — сложение, то  $a^n$  означает умножение  $n$  на  $a$ . Заметим также, что во всех этих группах операция — это сложение по модулю 6. Ниже показано, как мы находим элементы этих циклических подгрупп.

а. Циклическая подгруппа, сгенерированная из 0, — это  $H_1$ , имеет только один элемент (нейтральный элемент).

$$0^0 \bmod 6 = 0 \text{ (остановка, далее процесс повторяется).}$$

б. Циклическая подгруппа, сгенерированная на основе 1, — это  $H_4$ , которая есть сама группа  $G$ .

$$1^0 \bmod 6 = 0$$

$$1^1 \bmod 6 = 1$$

$$1^2 \bmod 6 = (1 + 1) \bmod 6 = 2$$

$$1^3 \bmod 6 = (1 + 1 + 1) \bmod 6 = 3$$

$$1^4 \bmod 6 = (1 + 1 + 1 + 1) \bmod 6 = 4$$

$1^5 \bmod 6 = (1 + 1 + 1 + 1 + 1) \bmod 6 = 5$  (остановка, далее процесс повторяется)

в. Циклическая подгруппа, сгенерированная на основе 2, — это  $H_2$ , которая имеет три элемента: 0, 2, и 4.

$$2^0 \bmod 6 = 0$$

$$2^1 \bmod 6 = 2$$

$$2^2 \bmod 6 = (2 + 2) \bmod 6 = 4 \text{ (остановка, далее процесс повторяется)}$$

г. Циклическая подгруппа, сгенерированная на основе 3, — это  $H_3$ , которая имеет два элемента: 0 и 3.

$$3^0 \bmod 6 = 0$$

$$3^1 \bmod 6 = 3 \text{ (остановка, далее процесс повторяется)}$$

д. Циклическая подгруппа, сгенерированная на основе 4, —  $H_2$ ; это — не новая подгруппа.

$$4^0 \bmod 6 = 0$$

$$4^1 \bmod 6 = 4$$

$$4^2 \bmod 6 = (4 + 4) \bmod 6 = 2 \text{ (остановка, далее процесс повторяется)}$$

е. Циклическая подгруппа, сгенерированная на основе 5, — это  $H_4$ , она есть сама группа  $G$ .

$$5^0 \bmod 6 = 0$$

$$5^1 \bmod 6 = 5$$

$$5^2 \bmod 6 = 4$$

$$5^3 \bmod 6 = 3$$

$$5^4 \bmod 6 = 2$$

$$5^5 \bmod 6 = 1 \text{ (остановка, далее процесс повторяется)}$$

### Пример 11.8

Из группы  $G = \langle \mathbb{Z}_{10}, \times \rangle$  можно получить три циклических подгруппы.  $G$  имеет только четыре элемента: 1, 3, 7 и 9. Циклические подгруппы —  $H_1 = \{1\}, \times$ ,  $H_2 = \{1, 9\}, \times$  и  $H_3 = G$ . Ниже показано, как мы находим элементы этих подгрупп.

а. Циклическая подгруппа, сгенерированная на основе 1, — это  $H_1$ . Подгруппа имеет только один элемент, а именно — нейтральный.

$$1^0 \bmod 10 = 1 \text{ (остановка, далее процесс повторяется)}$$

б. Циклическая подгруппа, сгенерированная на основе 3, — это  $H_3$ , которая есть группа  $G$ .

$$3^0 \bmod 10 = 1$$

$$3^1 \bmod 10 = 3$$

$$3^2 \bmod 10 = 9$$

$$3^3 \bmod 10 = 7 \text{ (остановка, далее процесс повторяется)}$$

в. Циклическая подгруппа, сгенерированная на основе 7, — это  $H_3$ , которая есть группа  $G$ .

$$7^0 \bmod 10 = 1$$

$$7^1 \bmod 10 = 7$$

$$7^2 \bmod 10 = 9$$

$$7^3 \bmod 10 = 3 \text{ (остановка, далее процесс повторяется)}$$

г. Циклическая подгруппа, сгенерированная на основе 9, — это  $H_2$ . Подгруппа имеет только два элемента.

$$9^0 \bmod 10 = 1$$

$$9^1 \bmod 10 = 9 \text{ (остановка, далее процесс повторяется)}$$

## Циклические группы

**Циклическая группа** — группа, которая является собственной циклической подгруппой. В примере 11.7 группа  $G$  имеет циклическую подгруппу  $H_5 = G$ . Это означает, что группа  $G$  — циклическая группа. В этом случае элемент, который генерирует циклическую подгруппу, может также генерировать саму группу. Этот элемент далее именуется "генератор". Если  $g$  — генератор, элементы в конечной циклической группе могут быть записаны как

$$\{e, g, g^2, \dots, g^{n-1}\}, \text{ где } g^n = e.$$

Заметим, что циклическая группа может иметь много генераторов.

### Пример 11.9

а. Группа  $G = \langle \mathbb{Z}_6, + \rangle$  — циклическая группа с двумя генераторами,  $g = 1$  и  $g = 11$ .

б. Группа  $G = \langle \mathbb{Z}_{10}, \times \rangle$  — циклическая группа с двумя генераторами,  $g = 3$  и  $g = 7$ .

### Теорема Лагранжа

**Теорема Лагранжа** показывает отношение между порядком группы к порядку ее подгруппы. Предположим, что  $G$  — группа и  $H$  — подгруппа  $G$ . Если порядок  $G$  и  $H$  —  $|G|$  и  $|H|$ , соответственно, то согласно этой теореме  $|H|$  делит  $|G|$ . В примере 11.7  $|G| = 6$ . Порядок подгруппы —  $|H_1| = 1$ ,  $|H_2| = 3$ ,  $|H_3| = 2$  и  $|H_4| = 6$ . Очевидно, все эти порядки есть делители 6.

Теорема Лагранжа имеет очень интересное приложение. Когда дана группа  $G$  и ее порядок  $|G|$ , могут быть легко определены порядки потенциальных подгрупп, если могут быть найдены делители. Например, порядок группы  $G = \langle \mathbb{Z}_{17}, + \rangle$  — это  $|17|$ . Делители 17 есть 1 и 17. Это означает, что эта группа может иметь только две подгруппы — нейтральный элемент и  $H_2 = G$ .

### Порядок элемента

**Порядок элемента** в группе  $\text{ord}(a)$  (порядок  $(a)$ ) является наименьшим целым числом  $n$ , таким, что  $a^n = e$ . Иными словами: порядок элемента — порядок группы, которую он генерирует.

### Пример 11.10

a. В группе  $G = \langle \mathbb{Z}_6, + \rangle$ , порядки элементов: порядок  $\text{ord}(0) = 1$ , порядок  $\text{ord}(1) = 6$ , порядок  $\text{ord}(2) = 3$ , порядок  $\text{ord}(3) = 2$ , порядок  $\text{ord}(4) = 3$ , порядок  $\text{ord}(5) = 6$ .

b. В группе  $G = \langle \mathbb{Z}_{10}, * \rangle$ , порядки элементов: порядок  $\text{ord}(1) = 1$ , порядок  $\text{ord}(3) = 4$ , порядок  $\text{ord}(7) = 4$ , порядок  $\text{ord}(9) = 2$ .

## Вопрос 2. Кольцо

**Кольцо**, обозначенное как  $R = \{ \dots \}, \bullet, \perp$ , является алгебраической структурой с двумя операциями. Первая операция должна удовлетворять всем пяти свойствам, требуемым для абелевой группы. Вторая операция должна удовлетворять только первым двум свойствам абелевой группы. Кроме того, вторая операция должна быть распределена с помощью первой. **Дистрибутивность** означает, что для всех  $a, b$  и  $c$  элементов из  $R$  мы имеем  $a \perp (b \bullet c) = (a \perp b) \bullet (a \perp c)$  и  $(a \bullet b) \perp c = (a \perp c) \bullet (b \perp c)$ . Коммутативное кольцо — кольцо, в котором коммутативное свойство удовлетворено и для второй операции. [Рисунок 11.4](#) показывает кольцо и коммутативное кольцо.

## Дополнительное замечание

Кольцо включает две операции. Однако вторая операция может не соответствовать третьему и четвертому свойствам. Другими словами, первая операция — фактически операция пары операций, таких как сложение и вычитание; вторая операция может содержать единственную операцию, например умножение, но может не содержать деление.

## Пример 11.11

Множество  $\mathbb{Z}$  с двумя операциями — сложением и умножением — является коммутативным кольцом, которое обозначается  $R = \mathbb{Z}, +, \times$ . Сложение удовлетворяет всем пяти свойствам; умножение удовлетворяет только трем свойствам.



**Рис. 11.4.** Кольцо

Умножение дистрибутивно с помощью сложения. Например,  $5 \times (3 + 2) = (5 \times 3) + (5 \times 2) = 25$ . Хотя, мы можем выполнить на этом множестве сложение и вычитание и умножение, но не деление. Деление не может применяться в этой структуре, потому что оно приводит к элементу из другого множества. Результат деления 12 на 5 есть 2,4, и он не находится в заданном множестве.

### Вопрос 3. Поле

**Поле**, обозначенное  $F = \{...\}, \bullet, \perp$  — коммутативное кольцо, в котором вторая операция удовлетворяет всем пяти свойствам, определенным для первой операции, за исключением того, что нейтральный элемент первой операции (иногда называемый нулевой элемент) не имеет инверсии. [Рисунок 11.5](#) показывает поле.

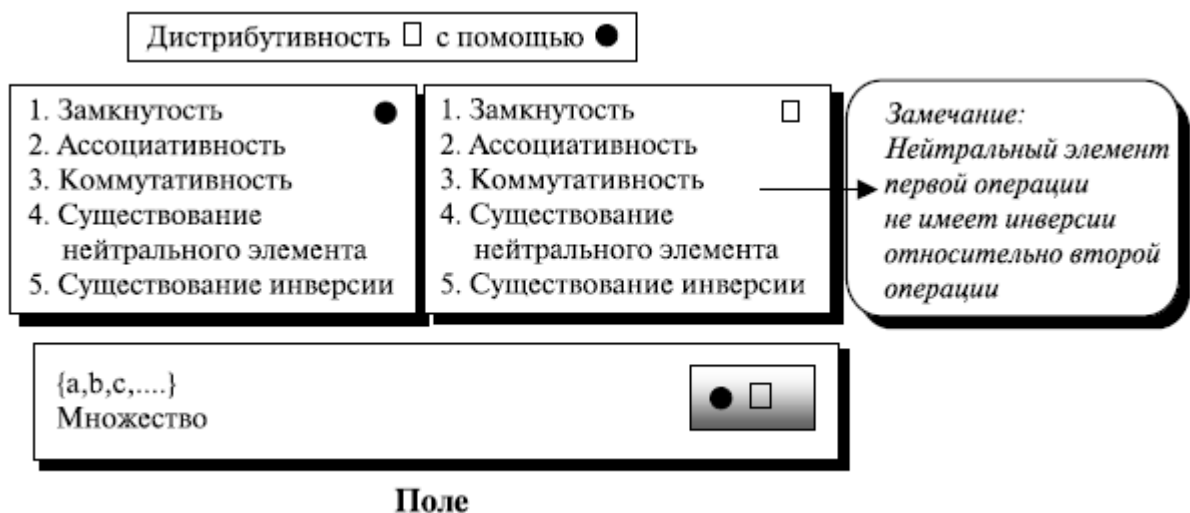


**Рис. 11.11. Поле**

Дополнительное замечание

Поле — структура, которая поддерживает две пары операций, используемые в математике: сложение/вычитание и умножение/деление. Есть одно исключение: не разрешено деление на нуль.

Конечные поля

Хотя общее определение касается полей бесконечного порядка, в криптографии используются экстенсивно только конечные поля. **Конечное поле** — поле с конечным числом элементов — является очень важной структурой в криптографии. Галуа показал что поля, чтобы быть конечными, должны иметь число элементов  $p^n$ , где  $p$  — простое, а  $n$  — положительное целое число. Конечные поля обычно называют **полями Галуа** и обозначают как  $GF(p^n)$ .

**Поле Галуа,  $GF(p^n)$ , — конечное поле с  $p^n$  элементами.**

Поля  $GF(p)$

Когда  $n = 1$ , мы имеем поле  $GF(p)$ . Это поле может быть множеством  $Z_p$ ,  $(0, 1, \dots, p-1)$  с двумя арифметическими операциями (сложение и умножение). Любой элемент в этом множестве имеет аддитивную инверсию, и элементы, отличные от нуля, имеют мультипликативную инверсию (мультипликативная инверсия для 0 отсутствует).

**Пример 11.12**

Очень общее поле в этой категории —  $GF(2)$  с множеством  $\{0,1\}$  и двумя операциями, сложением и умножением, как показано на [рисунке 11.6](#).

|                                                                                                                                                                                                                                            |     |   |   |           |   |   |          |     |   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---|---|-----------|---|---|----------|-----|---|
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <math>\{0,1\}</math> <div style="border: 1px solid black; padding: 2px; margin: 5px; display: inline-block;"> <math>+</math> <math>\times</math> </div> </div> |     |   |   |           |   |   |          |     |   |
|                                                                                                                                                                                                                                            | $+$ | 0 | 1 | $\times$  | 0 | 1 | $a$      | 0   | 0 |
| 0                                                                                                                                                                                                                                          | 0   | 1 | 0 | 0         | 0 | 0 | $-a$     | 1   | 1 |
| 1                                                                                                                                                                                                                                          | 1   | 0 | 1 | 1         | 0 | 1 | $a^{-1}$ | $-$ | 1 |
| Сложение                                                                                                                                                                                                                                   |     |   |   | Умножение |   |   | Инверсия |     |   |

**Рис. 11.6.** Поле  $GF(2)$

Есть несколько моментов, которые следует отметить в определении этого поля. Первый: множество имеет только два элемента, которые являются двоичными цифрами или битами (0 и 1). Второй: операция сложения — фактически ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR), операция, которую мы используем с двумя двоичными цифрами. Третий: операция умножения — AND, операция, которую мы используем с двумя двоичными цифрами. Четвертый: сложение и операции вычитания — те же самые (операция XOR). Пятый: умножение и операции деления — те же самые (ОПЕРАЦИЯ AND).

**Сложение/вычитание в  $GF(2)$  — операция ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR); умножение/деление — ОПЕРАЦИЯ И (AND).**

### Пример 11.13

Мы можем определить  $GF(5)$  на множестве  $Z_5$  (5 простое) с операторами сложения и умножения, показанными на [рис. 11.7](#).

Хотя мы можем использовать расширенный алгоритм Евклида, чтобы найти мультипликативные инверсии элементов в  $GF(5)$ , проще составить таблицу умножения и находить каждую пару, произведение которой равняется 1. Это (1, 1), (2, 3), (3, 2) и (4, 4). Заметим, что мы можем на этом множестве применить вычитание и умножение/деление (за исключением запрещенного деления на 0).



## Практическое занятие №14

### Математические модели расширенных полей Галуа

#### Цель и содержание:

1. Углубить знания по основам построения алгебраических систем, используемых при шифровании.
2. Изучить правила получения элементов расширения поле  $GF(2^n)$

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретическое обоснование

В криптографии мы часто должны использовать четыре операции (сложение, вычитание, умножение и деление). Другими словами, мы должны использовать поля. Однако когда мы работаем с компьютерами, положительные целые числа сохраняются в компьютере как  $n$ -битовые слова, в которых  $n$  является обычно 8, 16, 32, 64, и так далее. Это означает, что диапазон целых чисел — 0 до  $2^n - 1$ . Модуль —  $2^n$ . Так что возможны два варианта, если мы хотим использовать поле.

1. Мы можем задействовать  $GF(p)$  с множеством  $Z_p$ , где  $p$  — наибольшее простое число, меньшее, чем  $2^n$ . Но эта схема неэффективна, потому что мы не можем использовать целые числа от  $p$  до  $2^n - 1$ . Например, если  $n = 4$ , то наибольшее простое число, меньшее, чем  $2^4$ , — это 13. Это означает, что мы не можем использовать целые числа 13, 14 и 15. Если  $n = 8$ , наибольшее простое число, меньшее, чем  $2^8$ , — это 251, так что мы не можем использовать 251, 252, 253, 254 и 255.

2. Мы можем работать в  $GF(2^n)$  и использовать множество  $2^n$  элементов. Элементы в этом множестве —  $n$ -битовые слова. Например, если  $n = 3$ , множество равно:

$\{000, 001, 010, 100, 101, 110, 111\}$

Однако мы не можем интерпретировать каждый элемент как целое число от 0 до 7, потому что не могут быть применены обычные четыре операции

(модуль  $2^n$  — не простое число). Мы должны определить множество слов по 2 бита и две новых операции, которые удовлетворяют свойствам, определенным для поля.

### Пример 1

Определим  $GF(2^2)$  поле, в котором множество имеет четыре слова по 2 бита:  $\{00, 01, 10, 11\}$ . Мы можем переопределить сложение и умножение для этого поля таким образом, чтобы все свойства этих операций были удовлетворены, как это показано на рис. 4.8.

| $\oplus$ | 00 | 01 | 10 | 11 |
|----------|----|----|----|----|
| 00       | 00 | 01 | 10 | 11 |
| 01       | 01 | 00 | 11 | 10 |
| 10       | 10 | 11 | 00 | 01 |
| 11       | 11 | 10 | 01 | 00 |

Нейтральный элемент 00

| $\otimes$ | 00 | 01 | 10 | 11 |
|-----------|----|----|----|----|
| 00        | 00 | 00 | 00 | 00 |
| 01        | 00 | 01 | 10 | 11 |
| 10        | 00 | 10 | 11 | 01 |
| 11        | 00 | 11 | 01 | 10 |

Нейтральный элемент 01

Рисунок 1 - Пример поля  $GF(2^2)$  в степени 2)

Каждое слово — аддитивная инверсия себя. Каждое слово (кроме 00) имеет мультипликативную инверсию. Мультипликативные обратные пары —  $(01, 01)$  и  $(10, 11)$ . Сложение и умножение определено в терминах полиномиалов.

### Полиномы

Хотя мы можем непосредственно определить правила для операций сложения и умножения слов из двух бит, которые удовлетворяют свойства в  $GF(2^n)$ , проще работать с полиномиальным степени  $n - 1$  побитным представлением слов. **Полиномиальное** выражение степени  $n - 1$  имеет форму

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x^1 + a_0x^0$$

где  $x^i$  назван термином "i-тый элемент", а  $a_i$  называется коэффициентом i-того элемента. Хотя мы знаем полиномы в алгебре, но при представлении n-битовых слов полиномами необходимо следовать некоторым правилам:

а. степень  $x$  определяет позицию бита в n-битовых слов. Это означает, что крайний левый бит находится в нулевой позиции (связан с  $x^0$ ), самый правый бит находится в позиции  $n-1$  (связан с  $x^{n-1}$ );

b. коэффициенты сомножителей определяют значение битов. Каждый бит принимает только значение 0 или 1, поэтому наши полиномиальные коэффициенты могут иметь значение 0 или 1.

### Пример 2

Использование полиномов для предоставления слова из 8 бит (10011001) показано на [рис. 2](#).

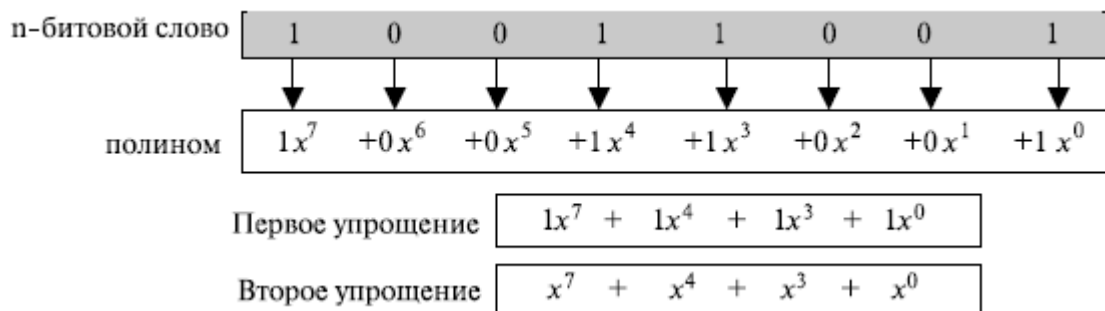


Рисунок 2 - Представление 8-ми битового слова полиномом

Заметим, что элемент полностью пропущен, если его коэффициент равен 0, и пропущен только коэффициент, если это 1. Также заметим, что элемент  $x^0$  равен 1.

### Пример 3

Чтобы найти слово на 8 битов, связанное с полиномом  $X^5 + X^2 + X$ , мы сначала восстановим пропущенные сомножители. Мы имеем  $n = 8$ , это означает полином степени 7. Расширенный полином имеет вид

$$0X^7 + 0X^6 + 1X^5 + 0X^4 + 1X^3 + 1X^2 + 1X^1 + 0X^0$$

Он связан со словом на 8 битов 00100110.

### Операции

Обратите внимание, что любая операция на полиномах фактически включает две операции: операции И коэффициентов двух полиномов. Другими словами, мы должны определить два поля: одно для коэффициентов и одно для полиномов. Коэффициенты равны 0 или 1; для этой цели мы можем использовать GF(2)-поле. Мы уже говорили о таком поле (см. пример 1). Для полиномов нам нужно поле GF( $2^n$ ), которое мы коротко обсудим ниже.

**Полиномы, представляющие n-битовые слова, используют два поля: GF(2) и GF(2<sup>n</sup>).**

Для генерирования элементов поля GF(2<sup>4</sup>) используйте полином  $f(x) = x^4 + x + 1$ .

**Решение**

Элементы 0,  $g^0$ ,  $g^1$ ,  $g^2$  и  $g^3$  могут быть сгенерированы достаточно просто, потому что в 4-битовом поле они представлены 0,  $x^0$ ,  $x^1$ ,  $x^2$  и  $x^3$  (не требуется деления на полином). Элементы от  $g^4$  до  $g^{14}$ , которые представляют  $g^4$  до  $g^{14}$  от  $x^4$  до  $x^{14}$ , нужно разделить на неприводимый полином. Для такого деления можно использовать полином  $f(g) = g^4 + g + 1 = 0$ . Применяв это отношение, мы имеем  $g^4 = -g - 1$ . поскольку сложение полей и вычитание полей — та же самая операция,  $g^4 = g + 1$ . Мы используем это отношение, чтобы найти значение всех элементов в виде 4-битовых слов:

$$0 = 0 = 0 = 0 \rightarrow 0 = (0000)$$

$$g^0 = g^1 = g^1 \rightarrow g^1 = (0010)$$

$$g^2 = g^2 = g^2 \rightarrow g^2 = (0100)$$

$$g^3 = g^3 = g^3 \rightarrow g^3 = (1000)$$

$$g^4 = g^4 = g + 1 \rightarrow g^4 = (0011)$$

$$g^5 = g(g + 1) = g^2 + g \rightarrow g^5 = (0110)$$

$$g^6 = g(g^2 + g) = g^3 + g^2 \rightarrow g^6 = (1100)$$

$$g^7 = g(g^3 + g) = g^3 + g + 1 \rightarrow g^7 = (1011)$$

$$g^8 = g(g^3 + g + 1) = g^2 + 1 \rightarrow g^8 = (0101)$$

$$g^9 = g(g^2 + 1) = g^3 + g \rightarrow g^9 = (1010)$$

$$g^{10} = g(g^3 + g) = g^2 + g + 1 \rightarrow g^{10} = (0111)$$

$$g^{11} = g(g^2 + g + 1) = g^3 + g^2 + g \rightarrow g^{11} = (1110)$$

$$g^{12} = g(g^3 + g^2 + g) = g^3 + g^2 + g + 1 \rightarrow g^{12} = (1111)$$

$$g^{13} = g(g^3 + g^2 + g + 1) = g^3 + g^2 + 1 \rightarrow g^{13} = (1101)$$

$$g^{14} = g(g^3 + g^2 + 1) = g^3 + 1 \rightarrow g^{14} = (1001)$$

Главная идея состоит в том, что вычисление элементов поля от  $g^4$  до  $g^{14}$  сводится к использованию соотношения  $g^4 = g + 1$  и предыдущих вычислений. Например,

$$g^{12} = g(g^{11}) = g(g^3 + g^2 + g) = g^4 + g^3 + g^2 = g^3 + g^2 + g + 1$$

После сокращения можно просто преобразовать степени в n-битовое слово. Скажем,  $g^3 + 1$  эквивалентно 1001, потому что присутствуют элементы со степенью 0 и 3. Заметим, что элементы с одинаковой степенью при таком процессе вычисления отменяют друг друга. Например,  $g^2 + g^2 = 0$ .

### Инверсии

Нахождение инверсий при использовании приведенного выше метода представления достаточно просто.

#### Аддитивные инверсии

Аддитивная инверсия каждого элемента — элемент непосредственно, потому что сложение и вычитание в этом поле — одна и та же операция,  $g^3 = g^3$ .

#### Мультипликативные инверсии

Найти мультипликативную инверсию каждого элемента также очень просто. Например, может найти мультипликативную инверсию элемента  $g^3$ , как показано ниже:

$$(g^3)^{-1} = g^{-3} = g^{12} = g^3 + g^2 + g + 1 \rightarrow (1111)$$

Заметим, что в этом случае степень рассчитывается по модулю  $2^n - 1$ ,  $2^4 - 1 = 15$ .

Поэтому  $-3 \bmod 15 = 12 \bmod 15$ .

Можно легко доказать, что  $g^3$  и  $g^{12}$  есть инверсные (обратные числа), потому что  $g^3 g^{12} = g^{15} = g^0 = 1$ .

#### Сложение и вычитание

Сложение и вычитание — это одинаковые операции. Промежуточные результаты могут быть упрощены, как проиллюстрировано в следующем примере.

### Пример 12

Этот пример показывает результаты операций сложения и вычитания:

а.

$$(g^3 + g^{12} + g^7) = g^3 + (g^3 + g^2 + g + 1) + (g^3 + g + 1) = g^3 + g^2 \rightarrow (1100)$$

б.  $g^3 - g^6 = g^3 + g^6 = g^3 + (g^3 + g^2) = g^2 \rightarrow (0100)$

Умножение и деление

Умножение есть сложение степени по модулю  $2^n - 1$ . Деление — это умножение, которое использует мультипликативную инверсию.

### Пример 13

Ниже показаны операции умножения и деления:

а.  $g^9 \times g^{11} = g^{20} = g^{20 \bmod 15} = g^5 = g^2 + g \rightarrow (0110)$

б.  $g^3 / g^8 = g^3 \times g^7 = g^{10} = g^2 + g + 1 \rightarrow (0111)$

Конечное поле  $GF(2^n)$  может использоваться для того, чтобы определить четыре операции — сложение, вычитание, умножение и деление  $n$ -битных слов. Только деление на нуль не определено. Каждое  $n$ -битовое слово может быть представлено как полином степени  $n - 1$  с коэффициентами в  $GF(2)$ , — это означает, что операции на  $n$ -битовых словах могут быть представлены как операции на этом полиноме. При умножении двух полиномов необходимо сделать эти операции операциями по модулю. Для этого мы должны определить неприводимый полином степени  $n$ . Чтобы найти мультипликативные инверсии к полиномам, может быть применен расширенный алгоритм Евклида.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие №15

### АЛГОРИТМ ШИФРОВАНИЯ БЕЗ ПЕРЕДАЧИ КЛЮЧА

#### Цель и содержание:

1. Углубить знания по основам построения алгебраических систем, используемых при шифровании.
2. Изучить правила получения минимальных многочленов поля  $GF(2^n)$

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретическое обоснование

Перед определением операций на полиномах мы должны поговорить о полиномах-модулях. Сложение двух полиномов никогда не создает полином, выходящий из множества. Однако умножение двух полиномов может создать полином со степенью большей, чем  $n - 1$ . Это означает, что мы должны делить результат на модуль и сохранять только остаток, как мы сделали в модульной арифметике. Для множеств полиномов в  $GF(2^n)$  группа полиномов степени  $n$  определена как модуль. Модуль в этом случае действует как полиномиальное простое число. Это означает, что никакие полиномы множества не могут делить этот полином. Простое полиномиальное число не может быть разложено в полиномы со степенью меньшей, чем  $n$ . Такие полиномы называются **неприводимые полиномы**. [Таблица 1](#) показывает примеры полиномов 1-5 степеней.

Для каждого значения степени часто есть более чем один неразлагаемый полином, — это означает, что когда мы определяем наш  $GF(2^n)$ , мы должны объявить, какой неприводимый полином мы используем как модуль.

Таблица 1. Список неприводимых полиномов

| Степень | Неприводимый полином |
|---------|----------------------|
| 1       | $(x+1)x$             |
| 2       | $(x^2+x+1)$          |

|   |                                                                                     |
|---|-------------------------------------------------------------------------------------|
| 3 | $(x^3+x^2+1)(x^3+x+1)$                                                              |
| 4 | $(x^4+x^3+x^2+x+1)(x^4+x^3+1)(x^4+x+1)$                                             |
| 5 | $(x^5+x^2+1)(x^5+x^3+x^2+x+1)(x^5+x^4+x^3+x+1)(x^5+x^4+x^3+x^2+1)(x^5+x^4+x^2+x+1)$ |

### Сложение

Теперь определим операцию сложения для полиномов с коэффициентом в GF(2). Операция сложения очень простая: мы складываем коэффициенты соответствующих элементов полинома в поле GF(2). Обратите внимание, что сложение двух полиномов степени  $n - 1$  всегда дает полином со степенью  $n - 1$  — это означает, что мы не должны использовать вычитание из модуля их результата.

### Пример 4

Произведем сложение  $(x^5 + x^2 + x) \oplus (x^3 + x^2 + 1)$  в GF(2<sup>8</sup>). Мы используем символ  $\oplus$  для обозначения полиномиального сложения. Ниже показана процедура

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0 \oplus$$

$$0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0$$

---


$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0 \rightarrow x^5 + x^3 + x + 1$$

В упрощенном полиноме (показан справа) сохранены элементы с коэффициентом 1 и удалены элементы с коэффициентом 0. Кроме того, удалены совпадающие элементы обоих полиномов, а несовпадающие сохраняются. Другими словами,  $x^5$ ,  $x^3$ , и  $x^1$  сохраняются, а  $x^2$ , который является совпадающим в этих двух полиномах, удален.

### Пример 5

Поскольку сложение в GF(2) означает операцию ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR), мы можем получить результат ИСКЛЮЧАЮЩЕГО ИЛИ для этих двух слов бит за битом. В предыдущем примере  $x^5 + x^2 + x$  есть 00100110, или полином, и  $x^3 + x^2 + 1$  есть 00001101. Результат — 00101011 или, в полиномиальном обозначении,  $x^5 + x^3 + x + 1$ .

**Аддитивный нейтральный элемент — тождество.** Аддитивный нейтральный элемент полинома — нулевой полином (полином со всеми коэффициентами, равными нулю), потому что, прибавляя этот полином к самому себе, в результате получаем нулевой полином.

**Аддитивная инверсия** полинома с коэффициентами в  $GF(2)$  — сам полином. Это означает, что операция вычитания та же самая, что и операция сложения.

Сложение и операции вычитания на полиномах — та же самая операция.

**Умножение**

Умножение в полиномах — сумма умножения каждого элемента одного полинома с каждым элементом второго полинома. Однако необходимо отметить три особенности.

Первая: умножение коэффициента проводится в поле  $GF(2)$ .

Вторая: умножение  $x^i$  на  $x^j$  дает результат  $x^{i+j}$ .

Третья: умножение может создать элементы со степенью большей, чем  $n-1$ , и это означает, что результат должен быть уменьшен с использованием полинома-модуля.

Сначала покажем, как умножить два полинома согласно вышеупомянутому определению. Позже будет дан более эффективный алгоритм, который может использоваться компьютерной программой.

### **Пример 6**

Найдите результат  $(x^5 + x^2 + x) \otimes (x^7 + x^4 + x^3 + x^2 + x)$  в  $GF(2^8)$  с неразлагаемым полиномом  $(x^8 + x^4 + x^3 + x + 1)$ . Обратите внимание, что для обозначения умножения двух полиномов используется символ  $\otimes$ .

**Решение**

Сначала умножаем эти два полинома так, как мы это делали в обычной алгебре. Обратите внимание, что в этом процессе пара элементов с равной степенью удаляется. Например, результат  $x^9 + x^9$  полностью удален, потому что он нулевой полином, по причине, которую мы обсуждали раньше при рассмотрении операции сложения.

$$P_1 \otimes P_2 = x^5(x^7 + x^4 + x^3 + x^2 + x) + x^2(x^7 + x^4 + x^3 + x^2 + x) + x(x^7 + x^4 + x^3 + x^2 + x)$$

$$P_1 \otimes P_2 = x^{12} + x^9 + x^8 + x^7 + x^6 + x^9 + x^6 + x^5 + x^4 + x^3 + x^8 + x^5 + x^4 + x^3 + x^2$$

$$P_1 \otimes P_2 = (x^{12} + x^7 + x^2) \bmod (x^8 + x^4 + x^3 + x + 1) = x^5 + x^3 + x^2 + x + 1$$

Чтобы найти конечный результат, разделим полином степени 12 на полином степени 8 (модуль) и сохраним только остаток. Процесс деления тот же самый, что и в обычной алгебре, но мы должны помнить, что здесь вычитание то же самое, что и сложение. [Рисунок 3](#) показывает процесс деления.

|                                  |                           |
|----------------------------------|---------------------------|
| $x^{12} + x^7 + x^2$             | $x^8 + x^4 + x^3 + x + 1$ |
| $x^{12} + x^8 + x^7 + x^5 + x^4$ | $x^4 + 1$                 |
| $x^8 + x^5 + x^4 + x^2$          |                           |
| $x^8 + x^4 + x^3 + x + 1$        |                           |
| $x^5 + x^3 + x^2 + x + 1$        | Остаток                   |

Рисунок 3 - Полиномиальное деление с коэффициентами в поле GF (2).

**Мультипликативное тождество** — всегда равно 1. Например, в GF(2<sup>8</sup>) мультипликативная инверсия — в побитном изображении 00000001.

**Мультипликативная инверсия.** Поиск мультипликативной инверсии требует привлечения расширенного алгоритма Евклида. Алгоритм Евклида должен быть применен к модулю и полиному, выполнение алгоритма является таким же, как и для целых чисел.

### Пример 7

В GF(2<sup>4</sup>) найдите инверсию  $(x^2 + 1) \bmod (x^4 + x + 1)$ .

Решение

Мы используем расширенный евклидов алгоритм, как это показано в [таблице 2](#):

Таблица 2. Алгоритм Евклида для упражнения 7

| q                 | r <sub>j</sub>        | r <sub>2</sub>      | r   | t <sub>j</sub>      | t <sub>2</sub>        | t                     |
|-------------------|-----------------------|---------------------|-----|---------------------|-----------------------|-----------------------|
| x <sup>2</sup> +1 | (x <sup>4</sup> +x+1) | (x <sup>2</sup> +1) | (x) | (0)                 | (1)                   | (x <sup>2</sup> +1)   |
| (x)               | (x <sup>2</sup> +1)   | (x)                 | (1) | (1)                 | (x <sup>2</sup> +1)   | (x <sup>3</sup> +x+1) |
| (x)               | (x)                   | (1)                 | (0) | (x <sup>2</sup> +1) | (x <sup>3</sup> +x+1) | (0)                   |

$$(1) \quad (0) \quad (x^3+x+1) \quad (0)$$

Это означает, что  $(x^2 + 1)^{-1} \bmod (x^4 + x + 1)$  есть  $(x^3 + x + 1)$ . Ответ может быть проверен просто: надо перемножить эти два полинома и найти остаток. В этом случае результат деления на модуль равен

$$[(x^2 + 1) \otimes (x^3 + x + 1)] \bmod (x^4 + x + 1) = 1$$

### Пример 8

В  $GF(2^8)$  найдите инверсию  $(x^5) \bmod (x^8 + x^4 + x^3 + x + 1)$ .

Решение

Будем использовать расширенный евклидов алгоритм, как это показано в

[Таблице 3:](#)

| q                 | r <sub>1</sub>                                        | r <sub>2</sub>                        | r                                                     | t <sub>1</sub>                                      | t <sub>2</sub>                                      | t                                                   |
|-------------------|-------------------------------------------------------|---------------------------------------|-------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|
| (x <sup>3</sup> ) | (x <sup>8</sup> +x <sup>4</sup> +x <sup>3</sup> +x+1) | (x <sup>5</sup> )                     | (x <sup>4</sup> +x <sup>3</sup> +x <sup>2</sup> +x+1) | (0)                                                 | (1)                                                 | (x <sup>3</sup> )                                   |
| (x+1)             | (x <sup>5</sup> )                                     | (x <sup>4</sup> +x <sup>3</sup> +x+1) | (x <sup>3</sup> +x <sup>2</sup> +1)                   | (1)                                                 | (x <sup>3</sup> )                                   | (x <sup>4</sup> +x <sup>3</sup> +1)                 |
| (x)               | (x <sup>4</sup> +x <sup>3</sup> +x+1)                 | (x <sup>3</sup> +x <sup>2</sup> +1)   | (1)                                                   | (x <sup>3</sup> )                                   | (x <sup>4</sup> +x <sup>3</sup> +1)                 | (x <sup>5</sup> +x <sup>4</sup> +x <sup>2</sup> +x) |
|                   | (x <sup>3</sup> +x <sup>2</sup> +1)                   | (1)                                   | (0)                                                   | (x <sup>4</sup> +x <sup>3</sup> +1)                 | (x <sup>5</sup> +x <sup>4</sup> +x <sup>2</sup> +x) | (0)                                                 |
|                   | (1)                                                   | (0)                                   |                                                       | (x <sup>5</sup> +x <sup>4</sup> +x <sup>2</sup> +x) | (0)                                                 |                                                     |

Это означает, что  $(x^5)^{-1} \bmod (x^8 + x^4 + x^3 + x + 1)$  есть  $(x^5 + x^4 + x^2 + x)$ .

Результат может быть легко проверен умножением этих двух полиномов и определением остатка деления по модулю.

$$[(x^5) \otimes (x^5+x^4+x^3+x)] \bmod (x^8+x^4+x^3+x+1) =$$

Умножение, использующее компьютер

Операция деления порождает проблему написания эффективной программы умножения двух полиномов. Лучший алгоритм для компьютерной реализации использует неоднократное умножение уменьшенного полинома на  $x$ . Например, вместо того чтобы находить результат  $(x^2 \otimes P_2)$ , программа находит результат  $(x \otimes (x \otimes P_2))$ . Преимущества этой стратегии будет обсуждаться далее, но сначала рассмотрим пример, чтобы проиллюстрировать алгоритм.

### Пример 9

Найдите результат умножения  $P_1 = (x^5 + x^2 + x)$  на  $P_2 = (x^7 + x^4 + x^3 + x^2 + x)$  в поле  $GF(2^8)$  с неприводимым полиномом  $(x^8 + x^4 + x^3 + x + 1)$ , используя алгоритм, изложенный выше.

Решение

Процесс показан в [таблице 4](#). Мы сначала находим промежуточный результат умножения  $x^0, x^1, x^2, x^3, x^4$  и  $x^5$ . Заметим, что необходимы только три составляющие произведения  $x^m \otimes P_2$  для  $m$  от 0 до 5, каждое вычисление зависит от предыдущего результата.

Таблица 4. Эффективный алгоритм умножения, использующий полиномы (пример 9)

| Степень                                                                        | Операция                        | Новый результат     | Вычитание |
|--------------------------------------------------------------------------------|---------------------------------|---------------------|-----------|
| $x^0 \otimes P_2$                                                              |                                 | $x^7+x^4+x^3+x^2+x$ | НЕТ       |
| $x^1 \otimes P_2$                                                              | $x \otimes (x^7+x^4+x^3+x^2+x)$ | $x^5+x^2+x+1$       | ДА        |
| $x^2 \otimes P_2$                                                              | $x \otimes (x^5+x^2+x+1)$       | $x^6+x^3+x^2+x$     | НЕТ       |
| $x^3 \otimes P_2$                                                              | $x \otimes (x^6+x^3+x^2+x)$     | $x^7+x^4+x^3+x^2$   | НЕТ       |
| $x^4 \otimes P_2$                                                              | $x \otimes (x^7+x^4+x^3+x^2)$   | $x^5+x+1$           | ДА        |
| $x^5 \otimes P_2$                                                              | $x \otimes (x^5+x+1)$           | $x^6+x^2+x$         | НЕТ       |
| $P_1 \times P_2 = (x^6+x^2+x)+(x^6+x^3+x^2+x)+(x^5+x^2+x+1) = x^5+x^3+x^2+x+1$ |                                 |                     |           |

Рассмотренный выше алгоритм имеет два преимущества. Первое — умножение полинома на  $x$  может быть выполнено простым сдвигом одного бита в  $n$ -битовом слове; операция может быть реализована на любом языке программирования. Второе — результат может быть использован, если максимальная степень полинома  $n-1$ . В этом случае сокращение может быть сделано просто с помощью применения операции ИСКЛЮЧАЮЩЕЕ ИЛИ с заданным модулем. В нашем примере самая высокая степень — только 8. Мы можем разработать простой алгоритм для нахождения промежуточных результатов.

1. Если старший разряд предыдущего результата равен 0, тогда надо сдвинуть предыдущий результат на один бит влево.

2. Если старший бит предыдущего результата равен 1: а. надо сдвинуть на один бит влево, и б. применить к нему операцию ИСКЛЮЧАЮЩЕЕ ИЛИ с модулем, исключив из этой операции старший разряд.

Повторим пример 9 для двоичной последовательности размером 8 бит. Пусть  $P_1 = 000100110$ ,  $P_2 = 10011110$ , модуль = 100011010 (девять битов). Обозначим операцию ИСКЛЮЧАЮЩЕЕ ИЛИ как  $\oplus$ . Пример приведен в [таблице 5](#).

Таблица 5. Эффективное умножение с применением n-битового слова

| Степень                                                                        | Операция сдвига влево | ИСКЛЮЧАЮЩЕЕ ИЛИ                              |
|--------------------------------------------------------------------------------|-----------------------|----------------------------------------------|
| $x^0 \otimes P_2$                                                              |                       | 10011110                                     |
| $x^1 \otimes P_2$                                                              | 00111100              | (00111100) $\oplus$ (00011010) =<br>00100111 |
| $x^2 \otimes P_2$                                                              | 01001110              | 01001110                                     |
| $x^3 \otimes P_2$                                                              | 10011100              | 10011100                                     |
| $x^4 \otimes P_2$                                                              | 00111000              | (00111000) $\oplus$ (00011010) =<br>00100011 |
| $x^5 \otimes P_2$                                                              | 01000110              | 01000110                                     |
| $P_1 \otimes P_2 = (000100110) \oplus (01001110) \oplus (01000110) = 00101111$ |                       |                                              |

В этом случае для умножения этих двух полиномов нам надо только пять операций левого сдвига и четыре ИСКЛЮЧАЮЩЕЕ ИЛИ. Вообще, для умножения двух полиномов степени  $n-1$  необходимо максимально  $(n-1)$  операций левого сдвига и  $2n$  операций ИСКЛЮЧАЮЩЕЕ ИЛИ.

**Умножение полиномов в  $GF(2^n)$  может быть выполнено с помощью операций левого сдвига и ИСКЛЮЧАЮЩЕЕ ИЛИ.**

**Пример 10**

Поле  $GF(2^3)$  состоит из 8 элементов. Покажем умножение и сложение таблиц для этого поля, используя неприводимый полином  $x^3+x^2+1$ . Мы будем оперировать с трехбитовым словом и полиномом. Заметим, что имеется два полинома третьей степени (см. [таблицу 1](#)). Другой полином  $(x^3 + x + 1)$  для умножения имеет таблицу, полностью отличающуюся от первой. [Таблица 6](#) показывает сложение. Затемненные клетки (нулевые) дают обратные пары для сложения.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие №16

### Алгоритм шифрования Эль-Гамала

#### Цель и содержание:

1. Углубить знания по основам построения алгебраических систем, используемых при шифровании.
2. Изучить правила выполнения прямого преобразования из позиционного кода в модулярный код

#### Формируемые компетенции: ОПК-6, ОПК-9.

#### Теоретическое обоснование

Известно, что модулярный код системы остаточных классов в качестве оснований использует взаимно простые числа  $p_i$ ,  $i = 1, 2, \dots, k$ . Произведение оснований кода СОК определяет рабочий диапазон

$$\mathfrak{D}_{\text{оак}} = \prod_{i=1}^k p_i . \quad (1)$$

Целое число  $A$  можно однозначно представить в виде наборов остатков, которые получены при делении исходного числа на основания СОК, если справедливо условие  $A < \mathfrak{D}_{\text{оак}}$ . В этом случае целое число  $A$ , в коде СОК можно записать в виде

$$A = (\alpha_1, \alpha_2, \dots, \alpha_k) , \quad (2)$$

где  $\alpha_i \equiv A \bmod p_i$ ;  $i = 1, 2, \dots, k$ .

Коды СОК позволяют операции над операндами  $A = (\alpha_1, \alpha_2, \dots, \alpha_k)$  и  $\hat{A} = (\beta_1, \beta_2, \dots, \beta_k)$  представить в виде совокупности соответствующих операций над остатками

$$A \circ \hat{A} = (|\alpha_1 \circ \beta_1|_{p_1}^+, |\alpha_2 \circ \beta_2|_{p_2}^+, \dots, |\alpha_k \circ \beta_k|_{p_k}^+) , \quad (3)$$

где  $\circ$  – операции сложения, вычитания и умножения.

Данные операции называются модульными. Используя эти операции можно повысить скорость вычислений за счет параллельной и независимой

обработки малоразрядных данных. Пусть заданы основания СОК  $p_1 = 2$ ,  $p_2 = 3$ ,  $p_3 = 5$ . Их произведение дает рабочий диапазон  $\mathfrak{D}_{\text{оää}} = \prod_{i=1}^k p_i = 2 \cdot 3 \cdot 5 = 30$ .

| Число | Остатки |       |       |
|-------|---------|-------|-------|
|       | mod 2   | mod 3 | mod 5 |
| 0     | 0       | 0     | 0     |
| 1     | 1       | 1     | 1     |
| 2     | 0       | 2     | 2     |
| 3     | 1       | 0     | 3     |
| 4     | 0       | 1     | 4     |
| 5     | 1       | 2     | 0     |
| 6     | 0       | 0     | 1     |
| 7     | 1       | 1     | 2     |
| ...   | ...     | ...   | ...   |
| 11    | 1       | 2     | 1     |
| ...   | ...     |       |       |
| 28    | 0       | 1     | 3     |
| 29    | 1       | 2     | 4     |

#### Сложение в СОК

|        |       |  |       |  |       |
|--------|-------|--|-------|--|-------|
|        | mod 2 |  | mod 3 |  | mod 5 |
| X = 7  | 1     |  | 1     |  | 2     |
| +      | +     |  | +     |  | +     |
| Y = 4  | 0     |  | 1     |  | 4     |
| C = 11 | 1     |  | 2     |  | 1     |

#### Вычитание в СОК

|       |       |  |       |  |       |
|-------|-------|--|-------|--|-------|
|       | mod 2 |  | mod 3 |  | mod 5 |
| X = 7 | 1     |  | 1     |  | 2     |
| -     | -     |  | -     |  | -     |
| Y = 4 | 0     |  | 1     |  | 4     |
| C = 3 | 1     |  | 0     |  | 3     |

#### Умножение в СОК

|       |       |  |       |  |       |
|-------|-------|--|-------|--|-------|
|       | mod 2 |  | mod 3 |  | mod 5 |
| X = 7 | 1     |  | 1     |  | 2     |
| *     | *     |  | *     |  | *     |

|          |   |  |   |  |   |
|----------|---|--|---|--|---|
| $Y = 4$  | 0 |  | 1 |  | 4 |
| $C = 28$ | 0 |  | 1 |  | 3 |

Для получения правильного результата должно выполняться условие невыхода за пределы рабочего диапазона  $P_{\text{раб}}$ .

Однако, в коде СОК существуют две обязательные немодульные операции:

- операция прямого преобразования из кода позиционной системы счисления (ПСС) в код СОК;
- операция обратного преобразования из кода СОК в код ПСС.

Рассмотрим выполнение операции прямого преобразования ПСС-СОК.

Очевидно, что самый простой способ получения остатка – операция деления. Однако, в кодах СОК данная операция не определена. Это связано со свойствами алгебраической системы – кольца.

Рассмотрим алгоритмы получения остатка без операции деления.

#### **Алгоритм понижения разрядности числа:**

1. Вычисляются остатки от весов разрядов числа по модулю  $p_i$ , т.е.

$$C_j \equiv 2^j \bmod p_i. \quad (4)$$

2. Вычисляется арифметическая сумма остатков согласно равенства

$$A_1 = \sum_{j=0}^L A_j C_j^{(1)} < A, \quad (5)$$

3. Если выполняется условие

$$A_1 < p_i, \quad (6)$$

то получен остаток где;  $\alpha_i = A_1 = A \bmod p_i$ . В противном случае – осуществляется переход к 2 этапу алгоритма, а вместо исходного числа  $A$  берется значение  $A_1$ .

4. Алгоритм повторяется до тех пор, пока не выполнится условие (6).

При вычислении остатка используется простой арифметический сумматор.

**Пример.** Пусть задано число  $A = 108$ . Необходимо найти остатки по для оснований  $p_1 = 3$ ,  $p_2 = 5$ . Получаем  $\alpha_1 = 108 \bmod 3 = 0$ .

Представим число  $A$  в двоичном коде. Получаем

$$108_{10} = 1101100_2 = 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 8 + 1 \cdot 4.$$

Вычислим остатки степеней двойки по модулю 5. Получаем

$$C_6 = 2^6 \bmod 3 = 1; C_5 = 2^5 \bmod 3 = 2; C_4 = 2^4 \bmod 3 = 1; C_3 = 2^3 \bmod 3 = 2;$$

$$C_2 = 2^2 \bmod 3 = 1; C_1 = 2^1 \bmod 3 = 2; C_0 = 2^0 \bmod 3 = 1.$$

Вычислим сумму  $A_1$  согласно (5)

$$A_1 = \tilde{N}_6 + \tilde{N}_5 + \tilde{N}_3 + \tilde{N}_2 = 1 + 2 + 2 + 1 = 6 < 108.$$

Так как  $A_1 > p_1 = 3$ , то повторяем алгоритм. Тогда сумма в двоичном коде

$$6_{10} = 110_2 = 1 \cdot 4 + 1 \cdot 2.$$

Воспользуемся выражением (5). Тогда

$$A_2 = 1 \cdot \tilde{N}_3 + 1 \cdot \tilde{N}_2 + 1 \cdot \tilde{N}_1 = 3 < 13.$$

Так как  $A_2 = 3$ , то остаток  $\alpha_1 = 108 \bmod 3 = 0$ .

Найдем остаток по модулю  $p_2 = 5$ . Получаем  $\alpha_1 = 108 \bmod 5 = 3$ .

Представим число  $A$  в двоичном коде. Получаем

$$108_{10} = 1101100_2 = 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 8 + 1 \cdot 4.$$

Вычислим остатки степеней двойки по модулю 5. Получаем

$$C_6 = 2^6 \bmod 5 = 4; C_5 = 2^5 \bmod 5 = 2; C_4 = 2^4 \bmod 5 = 1; C_3 = 2^3 \bmod 5 = 3;$$

$$C_2 = 2^2 \bmod 5 = 4; C_1 = 2^1 \bmod 5 = 2; C_0 = 2^0 \bmod 5 = 1.$$

Вычислим сумму  $A_1$  согласно (5)

$$A_1 = 1 \cdot \tilde{N}_6 + 1 \cdot \tilde{N}_5 + 1 \cdot \tilde{N}_3 + 1 \cdot \tilde{N}_2 = 13 < 108.$$

Так как  $A_1 > p_1 = 5$ , то повторяем алгоритм. Тогда сумма в двоичном коде

$$13_{10} = 1101_2 = 1 \cdot 8 + 1 \cdot 4 + 1.$$

Воспользуемся выражением (5). Тогда

$$A_2 = 1 \cdot \tilde{N}_3 + 1 \cdot \tilde{N}_2 + 1 \cdot \tilde{N}_1 = 8 < 13.$$

Так как  $A_2 > 5$ , то повторяем алгоритм. Тогда сумма в двоичном коде

$$8_{10} = 1000_2 = 1 \cdot 8.$$

Воспользуемся выражением (5). Тогда

$$A_3 = 1 \cdot \tilde{N}_3 = 3 < 5.$$

В результате получен остаток  $\alpha_1 = 108 \bmod 5 = 3$ .

2. Алгоритм непосредственного суммирования.

Преобразование исходного числа  $A$  в код СОК осуществляется с помощью набора констант, являющихся эквивалентами степеней оснований  $2^j$  и коэффициентов при соответствующих степенях оснований  $a^j$ , представленных в системе класса вычетов. В этом случае прямое преобразование из ПСС в код СОК осуществляется в соответствии с выражением

$$\alpha_i = \left| \sum_{j=0}^L (a_j \cdot 2^j) \bmod p_i \right|_{p_i}^+, \quad (7)$$

где  $i = 1, 2, 3, \dots, k$ .

Для повышения скорости выполнения преобразования ПСС-СОК надо двоичный код числа разбить на блоки, состоящие из  $B$  разрядов

$$\begin{aligned} |X|_{p_i}^+ &= \left( \left| 2^n a_n + \dots + 2^{n-\hat{A}} a_{n-\hat{A}} \right|_{p_i}^+ + \left| 2^{n-\hat{A}-1} a_{n-\hat{A}-1} + \dots + 2^{n-2\hat{A}} a_{n-2\hat{A}} \right|_{p_i}^+ + \right. \\ &\left. + \dots + \left| 2^{\hat{A}-1} a_{\hat{A}-1} + \dots + 1 \right|_{p_i}^+ \right) \bmod p_i = (B_L + B_{L-1} + \dots + B_1) \bmod p_i \end{aligned} \quad (8)$$

Затем определяются остатки по модулю каждого из блоков  $B_1, \dots, B_L$

$$\begin{aligned} |\hat{A}_1|_p^+ &= \left| 2^{B-1} a_{B-1} + \dots + 2^0 a_0 \right|_p^+ = \left| 2^0 \right|_p^+ (a_{\hat{A}-1}^1 + \dots + a_0^1) \Big|_p^+, \\ &\vdots \\ |\hat{A}_L|_p^+ &= \left| 2^n a_n + \dots + 2^{n-\hat{A}} a_{n-\hat{A}} \right|_p^+ = \left| 2^{n-\hat{A}} \right|_p^+ (a_{\hat{A}-1}^L + \dots + a_0^L) \Big|_p^+, \end{aligned} \quad (9)$$

где  $(a_{\hat{A}-1}^j + \dots + a_0^j)$  - двоичный код  $B_j$  блока.

Пусть необходимо вычислить остаток числа  $X = 32015$  по модулю  $p = 17$ . Представим в двоичном коде число  $X = 32015 = 0111 \ 1101 \ 0000 \ 1111_2$ . Воспользуемся последовательным алгоритмом (7) Тогда получаем

$$\begin{aligned} |32015|_{17}^+ &= |1111101000111|_{17}^+ = \left| 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^8 + 2^3 + 2^2 + 2^1 + 1 \right|_{17}^+ = \\ &= \left| 2^{14} \right|_{17}^+ + \left| 2^{13} \right|_{17}^+ + \left| 2^{12} \right|_{17}^+ + \left| 2^{11} \right|_{17}^+ + \left| 2^{10} \right|_{17}^+ + \left| 2^8 \right|_{17}^+ + \left| 2^3 \right|_{17}^+ + \left| 2^2 \right|_{17}^+ + \left| 2^1 \right|_{17}^+ + \left| 2^0 \right|_{17}^+ \Big|_{17}^+ = \\ &= |13 + 15 + 16 + 8 + 4 + 1 + 8 + 4 + 2 + 1|_{17}^+ 4. \end{aligned}$$

Увеличим размерность блока до  $B = 2$  разрядов. В этом случае число

$X = 32015_{10} = 01 \ 11 \ 11 \ 01 \ 00 \ 00 \ 11 \ 11_2$  Тогда

$$|32015|_{17}^+ = \left| 1 \cdot 2^{14} \right|_{17}^+ + \left| 3 \cdot 2^{12} \right|_{17}^+ + \left| 3 \cdot 2^{10} \right|_{17}^+ + \left| 1 \cdot 2^8 \right|_{17}^+ + \left| 3 \cdot 2^2 \right|_{17}^+ + \left| 3 \cdot 2^0 \right|_{17}^+ \Big|_{17}^+ =$$

$$= \left| 1 \cdot 13 \right|_{17}^+ + \left| 3 \cdot 16 \right|_{17}^+ + \left| 3 \cdot 4 \right|_{17}^+ + \left| 1 \cdot 1 \right|_{17}^+ + \left| 3 \cdot 4 \right|_{17}^+ + \left| 3 \cdot 1 \right|_{17}^+ \Big|_{17}^+ = 4.$$

Двоичный код числа  $X = 32015$  в и разобьем на блока по  $B = 4$  бит. Тогда получаем  $X = 32015 = 0111\ 1101\ 0000\ 1111$ . В результате получили 4 числа  $B_4 = 7, B_3 = 13, B_2 = 0, B_1 = 15$ . Тогда остаток по модулю 17 будет равен

$$|32015|_{17}^+ = \left| 7 \cdot 2^{12} \right|_{17}^+ + \left| 13 \cdot 2^8 \right|_{17}^+ + \left| 0 \cdot 2^4 \right|_{17}^+ + \left| 15 \cdot 2^0 \right|_{17}^+ \Big|_{17}^+ =$$

$$= \left| 7 \cdot 16 \right|_{17}^+ + \left| 13 \cdot 1 \right|_{17}^+ + \left| 0 \cdot 16 \right|_{17}^+ + \left| 15 \cdot 1 \right|_{17}^+ \Big|_{17}^+ = \left| 10 + 13 + 0 + 15 \right|_{17}^+ = 4.$$

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 17

### Построение электронной цифровой подписи RSA

#### Цель и содержание:

1. Углубить знания по основам построения алгебраических систем, используемых при шифровании.
2. Изучить правила выполнения обратного преобразования из модулярного кода в позиционный код

#### Формируемые компетенции: ОПК-6, ОПК-9.

#### Теоретический материал

Так как код СОК относится к непозиционным кодам, где число представляется в виде остатков, полученных при его делении на выбранные основания, то необходимо выполнять обратное преобразование из СОК в позиционный код.

Второй обязательной немодульной операцией является обратное преобразование из кода СОК в код ПСС (СОК-ПСС). В основу обратного преобразования СОК-ПССд положена китайская теорема об остатках

$$\dot{A} = \left| \alpha_1 B_1 + \alpha_2 B_2 + \dots + \alpha_k B_k \right|_{P_{\text{оаа}}}^+ = \sum_{i=1}^k \alpha_i B_i - r_A P_{\text{оаа}}, \quad (10)$$

где  $B_i$  – ортогональные базисы кода СОК;  $r_A$  – ранг числа  $A$ , показывающий количество превышений рабочего диапазона  $P_{\text{раб}}$ ;  $i = 1, 2, \dots, k$ .

При переводе из кода СОК в ПСС ортогональный базис используется в позиционном виде. Очевидно, что ортогональный базис можно представить в непозиционном виде как

$$\begin{aligned} B_1 &= (1, 0, \dots, 0); \\ B_2 &= (0, 1, \dots, 0); \\ &\vdots \\ B_k &= (0, 0, \dots, 1). \end{aligned} \quad (11)$$

Так как ортогональный базис  $B_i$ ,  $i = 1, 2, \dots, k$ , содержит нули, т.е.  $\alpha_u = 0$ , где  $u \neq i$ , кроме одной единицы  $\alpha_i = 1$ , то он должен делиться без остатка на другие основания, или на константу  $P_i = \frac{P_{\delta\alpha\alpha}}{p_i} = \prod_{\substack{u=1 \\ u \neq i}}^k p_u$ . Так как для полученного

значения может не выполняться условие

$$\delta_i = P_i \bmod p_i = 1, \quad (12)$$

то вводится вес ортогонального базиса  $m_i$ , такой, что справедливо

$$\delta_i m_i \equiv 1 \bmod p_i. \quad (13)$$

Тогда значение ортогонального базиса определяется

$$B_i = m_i D_i. \quad (14)$$

В результате ортогональный базис, удовлетворяющий условию

$$B_i \equiv 1 \bmod p_i. \quad (15)$$

Алгоритм вычисления ортогональных базисов:

1 этап. На данном этапе осуществляется вычисления значения

$$D_i = \frac{D_{\delta\alpha\alpha}}{p_i} = \prod_{\substack{u=1 \\ u \neq i}}^k p_u. \quad (16)$$

2 этап. Вычисление остатка

$$\delta_i = P_i \bmod p_i. \quad (17)$$

3 этап. Чтобы выполнялось условие (15) вычисляем значение веса ортогонального базиса, такое что

$$m_i \delta_i \equiv 1 \bmod p_i \quad (18)$$

4 этап. Вычисляется ортогонального базиса

$$B_i = m_i D_i^*. \quad (19)$$

Вычисленные ортогональные базисы должны удовлетворять условию

$$(\hat{A}_1 + \hat{A}_2 + \dots + B_k) \bmod P_{\delta\alpha\alpha} = 1. \quad (20)$$

**Пример.** Пусть задан код СОК с основаниями  $p_1 = 3$ ,  $p_2 = 5$ ,  $p_3 = 11$ . Тогда

рабочий диапазон равен  $D_{\delta\alpha\alpha} = \prod_{i=1}^3 p_i = 3 \cdot 5 \cdot 11 = 165$ . Найдем первый

ортогональный базис  $B_1$ .

1 Вычислим значение

$$D_1^* = \delta_2 \cdot \delta_3 = 5 \cdot 11 = 55.$$

2 этап. Вычислим остаток

$$\delta_1 = P_1 \bmod p_1 = 55 \bmod 3 = 1.$$

3 этап. Вычислим значение веса ортогонального базиса, такое что

$$m_1 \delta_1 \equiv 1 \bmod p_1.$$

$$m_1 \cdot 1 \equiv 1 \bmod p_1$$

Тогда имеем, что вес ортогонального базиса равен  $m_1 = 1$ .

4 этап. Значение ортогонального базиса равно

$$B_1 = m_1 D_1^* = 1 \cdot 55 = 55.$$

Вычислим значение второго ортогонального базиса  $B_2$ .

1 Вычислим значение

$$D_2 = \delta_1 \cdot \delta_3 = 3 \cdot 11 = 33.$$

2 этап. Вычислим остаток

$$\delta_2 = P_2 \bmod p_2 = 33 \bmod 5 = 3.$$

3 этап. Вычислим значение веса ортогонального базиса, такое что

$$m_2 \delta_2 \equiv 1 \bmod p_2.$$

$$m_2 \cdot 3 \equiv 1 \bmod 5$$

Методом полного перебора чисел от 1 до 4 найдем значение  $m_2$ .

Пусть  $m_2 = 1$ . Тогда  $m_2 \cdot 3 = 1 \cdot 3 \bmod 5 = 3$ . Значит  $m_2 = 1$  – не вес базиса.

Пусть  $m_2 = 2$ . Тогда  $m_2 \cdot 3 = 2 \cdot 3 \bmod 5 = 1$ . Значит  $m_2 = 2$  – вес базиса.

4 этап. Значение ортогонального базиса равно

$$B_2 = m_2 D_2 = 2 \cdot 33 = 66.$$

Вычислим значение третьего ортогонального базиса  $B_3$ .

1 Вычислим значение

$$D_3 = \delta_1 \cdot \delta_2 = 3 \cdot 5 = 15.$$

2 этап. Вычислим остаток

$$\delta_3 = P_3 \bmod p_3 = 15 \bmod 11 = 4.$$

3 этап. Вычислим значение веса ортогонального базиса, такое что

$$m_3 \delta_3 \equiv 1 \pmod{p_3}.$$

$$m_3 \cdot 4 \equiv 1 \pmod{11}$$

Методом полного перебора чисел от 1 до 10 найдем значение  $m_3$ .

Пусть  $m_3 = 1$ . Тогда  $m_3 \cdot 4 = 1 \cdot 4 \pmod{11} = 4$ . Значит  $m_3 = 1$  – не вес базиса.

Пусть  $m_3 = 2$ . Тогда  $m_3 \cdot 4 = 2 \cdot 4 \pmod{11} = 8$ . Значит  $m_3 = 2$  – не вес базиса.

Пусть  $m_3 = 3$ . Тогда  $m_3 \cdot 4 = 3 \cdot 4 \pmod{11} = 1$ . Значит  $m_3 = 3$  – вес базиса.

4 этап. Значение ортогонального базиса равно

$$B_3 = m_3 D_3 = 3 \cdot 15 = 45.$$

Выполним проверку полученных ортогональных базисов

$$(\hat{A}_1 + \hat{A}_2 + B_3) \pmod{P_{\text{оаа}}} = (55 + 66 + 45) \pmod{165} = 1.$$

Ортогональные базисы вычислены правильно.

Осуществим перевод числа  $A = (2, 3, 9)$  в позиционный код.

Тогда на основе китайской теоремы об остатках получаем

$$\hat{A} = |\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3 B_3|_{P_{\text{оаа}}}^+ = |2 \cdot 55 + 3 \cdot 66 + 9 \cdot 45|_{165}^+ = |713|_{165}^+ = 53.$$

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Практическое занятие № 18**

### **Построение электронной цифровой подписи Эль-Гамала**

#### **Цель и содержание:**

1. Углубить знания по основам построения схем разделения секрета.
2. Изучить схему разделения секрета на основе китайской теоремы об остатках

#### **Формируемые компетенции: ОПК-6, ОПК-9.**

#### **Теоретический материал**

Для того, чтобы составить представление о задачах, для решения которых потребовалось создавать схемы разделения секрета, приведем следующий исторический пример:

В книге «Gent und seine Schoenheiten» (Thill-Verlag, Bruessel, 1990) описывается следующий исторический пример. В IX-XIV вв. в бельгийском городе Генте была построена ратушная башня. В «секрете», то есть самом надежном помещении, хранились уставы и привилегии, имевшие важное значение. Помещение имело две двери, каждая с тремя замками. Ключи от этих замков находились во владении различных цехов. Документы хранились в шкафу, запиравшемся на три замка. Один ключ от шкафа хранился у фогта, а два других – у главного шеффена.

Таким образом, получить доступ к документам могли только совместно собравшиеся представители трех цехов, фогт и шеффен.

Этот исторический пример приведен для того, чтобы показать, что протоколы разделения секрета были реализованы задолго до появления криптографических протоколов как научного направления.

Протоколы разделения секрета применяются для распределенного хранения информации. Чаще всего такой информацией оказываются секретные ключи или пароли какого-либо абонента.

Например, главный бухгалтер предприятия держит секретную рабочую информацию зашифрованной, а ключ хранит в надежном месте, известном только ему. Но что случится, если главный бухгалтер уволится, попадет под машину, или тайник с ключом сгорит?

Тогда ключ будет утерян, а дешифрование в современных криптосистемах может занять миллионы лет! То есть информация тоже окажется утерянной.

Может быть, выдать по копии ключа заместителю главного бухгалтера и директору предприятия?

Но если заместитель захочет занять место главного бухгалтера, воспользуется своей копией ключа и подменит важную информацию? Или продаст ее конкурентам? Опасно давать свой секретный ключ другим людям.

Можно предложить следующий выход: разделить ключ, состоящий, скажем, из 64 бит, на четыре части по 16 бит и выдать одну часть генеральному директору, другую – его заместителю, третью – заместителю главного бухгалтера, а четвертую – супругу (супруге) главного бухгалтера.

Но что, если заместители договорятся сместить своих начальников и воспользуются своими частями ключа? Тогда для того, чтобы восстановить ключ, злоумышленникам потребуется подобрать всего лишь 32 бита, что потребует всего  $2^{32} \approx 4,3 \cdot 10^9$  итераций вместо  $2^{64} \approx 18,5 \cdot 10^{18}$  при подборе 64 битов. Злоумышленники смогут восстановить ключ за вполне обозримое время. Разумным будет разделить этот 64-битовый ключ  $k$  так, чтобы каждому досталось по 16 бит. Как это сделать? Генеральному директору, и заместителям можно выдать по случайной 16-битовой строке  $s_1, s_2, s_3$  соответственно, а супругу(е) главного бухгалтера – строку

$$s_4 = k - s_1 - s_2 - s_3 \bmod 2^{16}.$$

Тогда каждый из них будет обладать случайной строкой бит, по которой ключ  $k$  можно восстановить только перебором  $2^{16}$  – битового числа. Даже соединив три любых части, нельзя будет получить никакой информации о ключе, и нельзя будет уменьшить количество перебираемых битов. Но, соединив все четыре части,

$$s_1 + s_2 + s_3 + s_4 \bmod 2^{64} = k,$$

можно будет однозначно восстановить  $k$ .

Мы только что описали простейшую схему разделения секрета с одной разрешенной группой участников, состоящей из 4-х абонентов.

### Схема на основе Китайской теоремы об остатках

Сформулируем теорему, на основе которой можно построить пороговую схему разделения секрета.

#### Китайская теорема об остатках (I век до н.э. Сунь-Цзе)

Пусть  $m_1, \dots, m_k$  – попарно простые числа  $\Rightarrow$  система сравнений вида

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ x \equiv b_2 \pmod{m_2} \\ \dots\dots\dots \\ x \equiv b_k \pmod{m_k} \end{cases}$$

, где  $m_1, m_2, \dots, m_k$  – попарно простые числа, имеет единственное решение

$$x_0 \equiv \sum_{i=1}^k b_i M_i M'_i \pmod{M}, \text{ где } M = \prod_{i=1}^k m_i, M_i = \frac{M}{m_i}, M'_i = M_i^{-1} \pmod{m_i}.$$

□

Схема разделения секрета на основе китайской теоремы об остатках выглядит следующим образом:

Пусть  $N$  – общий секрет.

#### Фаза разделения секрета:

Берем  $p_1, p_2, \dots, p_n$  – различные простые числа.

Часть секрета, выдаваемая  $i$ -му участнику схемы, есть число  $x_i$ , вычисляемое как  $x_i \equiv N \pmod{p_i}$ .

Заметим, что числа  $p_1, p_2, \dots, p_n$  должны быть такими, чтобы произведение любых  $k$  из них было больше, чем  $N$ . А это достигается, когда для всех  $i$  выполняется  $p_i > \sqrt[k]{N}$ .

Для того, чтобы  $k-1$  участников не смогли восстановить секрет без  $k$ -го участника, необходимо, чтобы  $p_i \ll \sqrt[k-1]{N}$ .

Итак, относительно чисел  $p_1, p_2, \dots, p_n$  должны выполняться условия:

$$\sqrt[k]{N} < p_i < \sqrt[k-1]{N} \quad i \in \{1, \dots, n\}.$$

### Фаза восстановления секрета:

Собравшись вместе,  $k$  участников составляют и решают систему сравнений относительно неизвестного  $N$ :

$$\begin{cases} N \equiv x_1 \pmod{p_1} \\ N \equiv x_2 \pmod{p_2} \\ \dots \\ N \equiv x_k \pmod{p_k} \end{cases}.$$

Решив систему, участники получают общий секрет  $N$ .

### Пример.

Построим (3,3) – пороговую схему на основе Китайской теоремы об остатках для деления секрета  $N=549$ .

Определим область, из которой будем выбирать простые числа.

$\sqrt[3]{N} \approx 8,2$ ,  $\sqrt[2]{N} \approx 23,4$ . Итак,  $8,2 < p_i < 23,4$ .

Выберем  $p_1=11$ ,  $p_2=13$ ,  $p_3=17$ .

Фаза раздачи секрета:

$$x_1 = N \bmod p_1 = 549 \bmod 11 = 10;$$

$$x_2 = N \bmod p_2 = 549 \bmod 13 = 3;$$

$$x_3 = N \bmod p_3 = 549 \bmod 17 = 5.$$

Доля  $P_1$ :  $(x_1=10, p_1=11)$ ;

Доля  $P_2$ :  $(x_2=3, p_2=13)$ ;

Доля  $P_3$ :  $(x_3=5, p_3=17)$ .

Фаза восстановления секрета:

Участники  $P_1$ ,  $P_2$  и  $P_3$ , составляют систему сравнений

$$\begin{cases} N \equiv 10 \pmod{11} \\ N \equiv 3 \pmod{13} \\ N \equiv 5 \pmod{17} \end{cases}$$

и решают ее по Китайской теореме об остатках:

$$N=10\cdot 221\cdot 1+3\cdot 187\cdot 8+5\cdot 143\cdot 5 \bmod 11\cdot 13\cdot 17 = 2210+4488+3575 \bmod 2431 = 2210+2057+1144 \bmod 2431 = 549.$$

Действительно, секрет был восстановлен тремя участниками.

Теперь посмотрим, что случится, если секрет захотят восстановить только два участника – например,  $P_1$  и  $P_2$ . Эти участники составят с помощью своих долей секрета систему:

$$\begin{cases} N \equiv 10 \pmod{11} \\ N \equiv 3 \pmod{13} \end{cases}$$

и получают ее решение:

$$N = 10\cdot 13\cdot 6+3\cdot 11\cdot 6 \bmod 11\cdot 13 = 780+198 \bmod 143 = 978 \bmod 143 = 120.$$

Как видим, получилось совсем другое значение, а не секрет. То есть неразрешенная группа из двух человек не сумела восстановить секрет, пользуясь своими долями.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 19

### Ключевой обмен на основе алгоритма Диффи-Хеллмана

#### Цель и содержание:

1. Углубить знания по основам построения алгебраических систем, используемых при шифровании.
2. Изучить правила получения минимальных многочленов поля  $GF(2^n)$

#### Формируемые компетенции: ОПК-6, ОПК-9.

#### Теоретическое обоснование

Ранее мы рассмотрели схемы разделения секрета как протокол, в котором участники доверяют раздающему на фазе разделения секрета, и доверяют друг другу на фазе восстановления секрета. Рассмотрим теперь случай, когда некоторые участники протокола могут оказаться противниками. Задачей протокола разделения секрета в таком случае является защита честных участников от мошенничества противников.

Конечно, если противником является раздающий, он может просто саботировать выполнение протокола, и невозможно предложить криптографическую защиту от такого рода действий. Но саботаж протокола мгновенно выявляется и может быть пресечен другими методами. Честные участники в таком случае доверят разделение секретов другим, честным, дилерам.

Для защиты же от более хитрых действий противника возникла идея **проверяемого разделения секрета**.

Идея такова: на этапе разделения секрета дилер публикует секрет в зашифрованном виде так, чтобы, пользуясь этой информацией, никто не мог восстановить секрет, но чтобы каждый абонент, используя свою долю секрета, мог проверить, с того ли секрета была получена эта доля.

Кроме дилера, злоумышленниками могут являться некоторые из абонентов. Нечестные участники могут попытаться помешать восстановлению секрета, посылая не свои доли секрета, а какие-то другие значения. От протокола

требуется, чтобы все честные участники, если их не менее  $t$ , всегда правильно восстанавливали значение секрета  $m$ .

Для построения схемы разделения секрета на основе китайской теоремы об остатках с обнаружением сфальсифицированной доли необходимо использовать избыточные модулярные коды.

Характерной чертой модулярных кодов является отсутствие обмена промежуточными результатами между вычислительными каналами, реализующих вычисление по модулям кода СОК. Это отмеченное выше свойство модулярных кодов можно использовать для повышения отказоустойчивости вычислительных устройств, функционирующих в МК. Очевидно, что ошибка, которая возникает по модулю  $p_i$ ,  $i = 1, 2, \dots, k$ , не сможет перемещаться от одного вычислительного тракта к другому, так как между ними отсутствует связь. При этом ошибку можно обнаружить и исправить во время выполнения обратного преобразования из модулярного кода в позиционную систему счисления.

Для разработки кодов способных обнаруживать и и корректировать ошибки необходимо увеличивать избыточность такого кода. Пусть задан безизбыточный МК, который содержит  $k$  взаимно простых оснований  $p_1, p_2, \dots, p_k$ . Для данного набора оснований рабочий диапазон будет равен

$$\mathbb{D}_{\text{о\ddot{a}a}} = p_1 \cdot p_2 \cdot \dots \cdot p_k = \prod_{i=1}^k \delta_i. \quad (1)$$

Для однозначного представления двоичного кода числа  $A$  в модулярном коде необходимо, что выполнялось условие

$$\dot{A} = (\alpha_1, \alpha_2, \dots, \alpha_k) < \mathbb{D}_{\text{о\ddot{a}a}} = \prod_{i=1}^k \delta_i. \quad (2)$$

Значит каждому набору остатков  $(\alpha_1, \alpha_2, \dots, \alpha_k)$  в соответствие ставится только одно число  $A$ , которое принадлежит рабочему диапазону. Воспользуемся данным свойством для построения корректирующих модулярных кодов.

Введем в модулярный код минимальную избыточность. Для этого возьмем дополнительное  $k+1$  –е основание, которое удовлетворяет условиям

$$\text{НОД}(p_i, p_{k+1}) = 1, \quad (3)$$

$$p_1 < p_2 < \dots < p_k < p_{k+1}, \quad (4)$$

где  $i = 1, 2, \dots, k$ .

В результате введения одного избыточного основания произойдет расширение диапазона модулярного кода. Тогда произведение модулей избыточного кода даст полный диапазон, который равен

$$D_{\text{нб}} = p_1 \cdot p_2 \cdot \dots \cdot p_{k+1} = \prod_{i=1}^k \delta_i \cdot \delta_{k+1} = D_{\text{ддд}} \delta_{k+1}. \quad (5)$$

Таким образом, диапазон модулярного кода увеличился в  $p_{k+1}$  раз по сравнению с рабочим диапазоном. В результате этого общее количество возможных кодовых комбинаций такого избыточного кода  $N_o = P_{\text{нб}}$ .

Разрешенные комбинации в избыточном коде СОК должны удовлетворять условию

$$\dot{A} = (\alpha_1, \alpha_2, \dots, \alpha_{k+1}) < P_{\text{ддд}}. \quad (6)$$

Рассмотрим однократную ошибку в коде СОК, под которой понимается искажение одного остатка в кодовой комбинации. Пусть ошибка произошла по первому основанию. Ее глубина может быть  $\Delta\alpha_1 = \{1, 2, \dots, \delta_1 - 1\}$ . Тогда ошибочный остаток будет определяться

$$\alpha_1^* = (\alpha_1 + \Delta\alpha_1) \bmod p_1. \quad (7)$$

Тогда ошибочная комбинация будет иметь вид

$$\dot{A}^* = (\alpha_1^*, \alpha_2, \dots, \alpha_{k+1}) = ((\alpha_1 + \Delta\alpha_1) \bmod p_1, \alpha_2, \dots, \alpha_{k+1}). \quad (8)$$

Воспользуемся китайской теоремой об остатках для выполнения обратного преобразования из кода СОК в позиционный код. Тогда

$$\begin{aligned} \dot{A}^* &= (\alpha_1^* B_1 + \alpha_2 B_2 + \dots + \alpha_{k+1} B_{k+1}) \bmod P_{\text{нб}} = \\ &= ((\alpha_1 B_1 + \Delta\alpha_1 B_1 + \alpha_2 B_2 + \dots + \alpha_{k+1} B_{k+1}) \bmod P_{\text{нб}}) = (A + \Delta\alpha_1 B_1) \bmod P_{\text{нб}} \end{aligned} \quad (9)$$

Анализ выражения (9) показывает, что при возникновении ошибки кодовая комбинация выходит за пределы рабочего диапазона, то есть

$$\dot{A}^* = (\alpha_1^*, \alpha_2, \dots, \alpha_{k+1}) = (A + \Delta\alpha_1 B_1) \bmod P_{\text{нб}} > D_{\text{ддд}}. \quad (10)$$

Именно на этом свойстве кодов СОК строятся алгоритмы поиска и коррекции ошибок в МК.

**Пример.**

Пусть заданы основания информационные основания кода СОК  $p_1 = 2$ ,  $p_2 = 3$ ,  $p_3 = 5$ . Тогда рабочий диапазон будет равен  $P_{\text{раб}} = 2 \cdot 3 \cdot 5 = 30$ . Тогда число разрешенных комбинаций равно 30 – от нулевой до 29-й. Введем контрольное основание  $p_4 = 11$ . Это приведет к увеличению диапазона до полного  $P_{\text{полн}} = 2 \cdot 3 \cdot 5 \cdot 11 = 30 \cdot 11 = 330$ . Разрешенные комбинации удовлетворяют условию

$$\dot{A} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) < P_{\text{оаа}} = 30.$$

Возьмем в качестве разрешенной комбинации  $A_{29} = (1, 2, 4, 7) = 29$ . Выполним обратное преобразование из СОК в ПСС. Для этого вычислим значения ортогональных базисов.

Для вычисления ортогонального базиса  $B_1$ :

1 этап. Вычислим  $\dot{D}_1 = \dot{D}_{\text{оаа}} / \delta_1 = 330 / 2 = 165$ .

2 этап. Определим остаток по модулю  $p_1 = 2$ . Получаем

$$\delta_1 = \dot{D}_1 \bmod p_1 = 165 \bmod 2 = 1.$$

3 этап. Значение веса  $B_1$  из условия  $\delta_1 m_1 \equiv 1 \bmod p_1$ , то имеем.  $m_1 = 1$ .

4 этап. Вычислим ортогональный базис  $B_1 = m_1 \dot{D}_1 = 165$ .

Для вычисления второго ортогонального базиса  $B_2$ :

1 этап. Определим константу  $\dot{D}_2 = \dot{D}_{\text{псб}} / \delta_2 = 110$ .

2 этап. Вычислим остаток по модулю  $p_2 = 3$ . Получаем

$$\delta_2 = \dot{D}_2 \bmod p_2 = 110 \bmod 3 = 2.$$

3 этап. Определим значение веса  $B_2$  из условия  $\delta_2 m_2 \equiv 1 \bmod p_2$ . Получаем вес базиса равен  $m_2 = 2$ .

4 этап. Вычислим ортогональный базис  $B_2 = m_2 \dot{D}_2 = 220$ .

Для вычисления третьего ортогонального базиса кода СОК:

1 этап. Определим значение константы  $\dot{D}_3 = \dot{D}_{\text{псб}} / \delta_3 = 66$ .

2 этап. Вычислим значение остатка по модулю  $p_3 = 5$ . Получаем

$$\delta_3 = \dot{D}_3 \bmod p_3 = 66 \bmod 5 = 1.$$

3 этап. Определим значение веса  $B_3$  из условия  $\delta_3 m_3 \equiv 1 \bmod p_3$ . Получаем, что вес ортогонального базиса равен  $m_3 = 1$ .

4 этап. Вычислим ортогональный базис  $B_3 = m_3 \dot{D}_3 = 66$ .

Для вычисления четвертого ортогонального базиса кода СОК:

1 этап. Определим значение константы  $\mathbb{D}_4 = \mathbb{D}_{\text{нѳі}} / \delta_4 = 30$ .

2 этап. Вычислим значение остатка по модулю  $p_4 = 11$ . Получаем

$$\delta_4 = \mathbb{D}_4 \bmod p_4 = 30 \bmod 11 = 8.$$

3 этап. Определим значение веса  $B_4$  из условия  $\delta_4 m_4 \equiv 1 \bmod p_4$ . Получаем, что вес ортогонального базиса равен  $m_4 = 7$ .

4 этап. Вычислим ортогональный базис  $B_4 = m_4 \mathbb{D}_4 = 210$ .

Проверим полученные значения ортогональных базисов

$$(B_1 + B_2 + B_3 + B_4) \bmod P_{\text{полн}} = (165 + 220 + 66 + 210) \bmod 330 = 1$$

Переведем  $A_{29} = (1, 2, 4, 7)$  в позиционный код

$$\begin{aligned} \dot{A} &= (\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3 B_3 + \alpha_4 B_4) \bmod P_{\text{нѳі}} = \\ &= (1 \cdot 165 + 2 \cdot 220 + 4 \cdot 66 + 7 \cdot 210) \bmod 330 = (2339) \bmod 330 = 29 \end{aligned}$$

Пусть искажениям подвергается третье основание. Тогда глубина ошибки может быть  $\Delta\alpha_3 = \{1, 2, 3, 4\}$ . Запишем ошибочные комбинации и переведем их значения в позиционный код.

Пусть глубина ошибки равна  $\Delta\alpha_3 = 1$ . Тогда ошибочный остаток

$$\alpha_3^{\Delta=1} = (\alpha_3 + \Delta\alpha_3) \bmod p_3 = (4 + 1) \bmod 5 = 0.$$

Тогда ошибочная комбинация будет иметь вид

$$\dot{A}_3^{\Delta=1} = (1, 2, 0, 7).$$

Переведем ее в позиционный код. Получаем

$$\begin{aligned} \dot{A}_3^{\Delta=1} &= (\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3^{\Delta=1} B_3 + \alpha_4 B_4) \bmod P_{\text{нѳі}} = \\ &= (1 \cdot 165 + 2 \cdot 220 + 0 \cdot 66 + 7 \cdot 210) \bmod 330 = (2075) \bmod 330 = 95 > 30. \end{aligned}$$

Пусть глубина ошибки равна  $\Delta\alpha_3 = 2$ . Тогда ошибочный остаток

$$\alpha_3^{\Delta=2} = (\alpha_3 + \Delta\alpha_3) \bmod p_3 = (4 + 2) \bmod 5 = 1.$$

Тогда ошибочная комбинация будет иметь вид

$$\dot{A}_3^{\Delta=2} = (1, 2, 1, 7).$$

Переведем ее в позиционный код. Получаем

$$\begin{aligned} \dot{A}_3^{\Delta=2} &= (\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3^{\Delta=2} B_3 + \alpha_4 B_4) \bmod P_{\text{нѳі}} = \\ &= (1 \cdot 165 + 2 \cdot 220 + 1 \cdot 66 + 7 \cdot 210) \bmod 330 = (2141) \bmod 330 = 161 > 30. \end{aligned}$$

Пусть глубина ошибки равна  $\Delta\alpha_3 = 3$ . Тогда ошибочный остаток

$$\alpha_3^{\Delta=3} = (\alpha_3 + \Delta\alpha_3) \bmod p_3 = (4 + 3) \bmod 5 = 2.$$

Тогда ошибочная комбинация будет иметь вид

.

Переведем ее в позиционный код. Получаем

$$\begin{aligned} \dot{A}_3^{\Delta=3} &= (\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3^{\Delta=3} B_3 + \alpha_4 B_4) \bmod P_{\text{ннн}} = \\ &= (1 \cdot 165 + 2 \cdot 220 + 2 \cdot 66 + 7 \cdot 210) \bmod 330 = (2207) \bmod 330 = 227. \end{aligned}$$

Ошибочная комбинация  $\dot{A}_3^{\Delta=3} = (1, 2, 2, 7) = 227 > 30$ .

Пусть глубина ошибки равна  $\Delta\alpha_3 = 4$ . Тогда ошибочный остаток

$$\alpha_3^{\Delta=4} = (\alpha_3 + \Delta\alpha_3) \bmod p_3 = (4 + 4) \bmod 5 = 3.$$

Тогда ошибочная комбинация будет иметь вид

$$\dot{A}_3^{\Delta=4} = (1, 2, 3, 7).$$

Переведем ее в позиционный код. Получаем

$$\begin{aligned} \dot{A}_3^{\Delta=4} &= (\alpha_1 B_1 + \alpha_2 B_2 + \alpha_3^{\Delta=4} B_3 + \alpha_4 B_4) \bmod P_{\text{ннн}} = \\ &= (1 \cdot 165 + 2 \cdot 220 + 3 \cdot 66 + 7 \cdot 210) \bmod 330 = (2273) \bmod 330 = 293 > 30. \end{aligned}$$

Получаем, что все ошибочные комбинации вышли за пределы рабочего диапазона.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 20

### Модифицированный метод ключевого обмена на основе алгоритма Диффи-Хеллмана

#### Цель и содержание:

1. Углубить знания по основам построения электронных цифровых подписей.
2. Изучить правила получения электронной цифровой подписи на основе алгоритма RSA

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретический материал

В качестве электронной цифровой подписи рассмотрим асимметричный алгоритм шифрования с открытым ключом алгоритм RSA. Алгоритм асимметричного шифрования RSA является одним из первых полноценных алгоритмов с открытым ключом. Он был разработан Рональдом Риверсом, Адрианом Шамиром и Леонардо Адлеманом в 1976 году.

Рассмотрим алгоритм построения электронной цифровой подписи на основе RSA. Он состоит в следующем.

1. Пусть абоненты  $A$  и  $B$  решили наладить между собой секретную переписку с использованием электронной цифровой подписи. Тогда каждый из них, независимо от другого, выбирает 2 больших простых числа, находит их произведение, функцию Эйлера  $\varphi(P)$  от произведения, этих чисел, а затем выбирает случайное число согласно алгоритма:

$$A \Rightarrow p_1, p_2; P_1 = p_1 \cdot p_2; \Rightarrow \varphi(P_1) = (p_1 - 1)(p_2 - 1);$$

$$\hat{A} \Rightarrow 0 < a < \varphi(P_1), (a, \varphi(P_1)) = 1;$$

$$B \Rightarrow q_1, q_2; Q_1 = q_1 \cdot q_2; \Rightarrow \varphi(Q_1) = (q_1 - 1)(q_2 - 1);$$

$$\hat{A} \Rightarrow 0 < b < \varphi(Q_1), (b, \varphi(Q_1)) = 1.$$

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная: 1-5*

*Дополнительная: 1-6*

## Практическое занятие №21

### СХЕМА РАЗДЕЛЕНИЯ СЕКРЕТА ШАМИРА

#### Цель и содержание:

1. Углубить знания, по основам использования алгебраических структур для шифрования с двумя секретными ключами.
2. Исследовать основные характеристики алгоритма шифрования.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Затем открытые ключи абонента А:  $a$  и  $P_1$  находятся в открытом доступе.

Открытые ключи абонента В:  $b$  и  $Q_1$  печатаются в телефонной книге.

Каждый из абонентов независимо от другого выбирает свой секретный ключ согласно условия

$$A : \alpha \Rightarrow a \cdot \alpha \equiv 1 \pmod{\phi(P_1)}; \quad (1)$$

$$B : \beta \Rightarrow b \cdot \beta \equiv 1 \pmod{\phi(Q_1)}. \quad (2)$$

Пусть абонент А решает подписать сообщение  $m$ , передаваемое абоненту В и пусть  $0 < m < P_1$ , иначе текст делят на блоки.

1. Абонент А подписывает сообщение  $m$  своим секретным ключом

$$m_1 \equiv m^a \pmod{P_1}. \quad (3)$$

Подписанное сообщение передается абоненту В

2. Абонент В, используя открытый ключ абонента А проверяет правильность подписи

$$m \equiv m_1^a \pmod{P_1}. \quad (4)$$

Пусть абонент В решает подписать сообщение  $n$  и переслать его абоненту А и пусть  $0 < n < Q_1$ , иначе текст делят на блоки.

1. Абонент В подписывает сообщение  $n$  своим секретным ключом и находит:

$$n_1 \equiv n^b \pmod{Q_1}. \quad (5)$$

Подписанное сообщение пересылают абоненту А.

2. Абонент А проверяет правильность подписи и расшифровывает сообщение открытым ключом абонента В:

$$n \equiv n_1^b \pmod{Q_1}$$

На этом алгоритм заканчивается.

Студены выполняют индивидуальные задания согласно таблицы 1.

Таблица 1 – Задание для изучения электронной цифровой подписи RSA

| Вариант | Первое число | Второе число | Сообщение $m$ |
|---------|--------------|--------------|---------------|
| 1       | 23           | 7            | 17            |
| 2       | 29           | 13           | 18            |
| 3       | 31           | 11           | 19            |
| 4       | 37           | 17           | 20            |
| 5       | 41           | 19           | 21            |
| 6       | 43           | 5            | 22            |
| 7       | 47           | 7            | 23            |
| 8       | 53           | 11           | 24            |
| 9       | 23           | 13           | 16            |
| 10      | 29           | 17           | 15            |
| 11      | 31           | 23           | 14            |
| 12      | 37           | 29           | 13            |
| 13      | 41           | 11           | 28            |
| 14      | 43           | 13           | 29            |
| 15      | 47           | 17           | 30            |
| 16      | 53           | 5            | 31            |

Студенты самостоятельно выбирают значение ключей для обеих сторон (А и В) и исследуют процесс создания и проверки ЭЦП сообщения.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6



## Практическое занятие № 22

### Протокол аутентификации Шнорра

#### Цель и содержание:

1. Углубить знания по основам построения схем разделения секрета.
2. Изучить правила получения образов ключей и их восстановления в схеме

Блекли

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретический материал

### 4.2. Пороговые схемы разделения секрета

**$(n, k)$  – пороговой схемой разделения секрета** ( $k \leq n$ ) называется такая схема, в которой секрет разделяется между  $n$  участниками, причем разрешенной группой является любая группа, насчитывающая не менее  $k$  абонентов.

#### 4.2.1. Схема Блекли.

Как известно, система  $k$  линейно независимых сравнений от  $k$  неизвестных по простому модулю имеет ровно одно решение.

На этом основана пороговая схема Блекли, созданная в 1979 году. Секрет  $m$  разделяется между  $n$  абонентами так, что любая группа, состоящая не менее, чем из  $k$  абонентов, является разрешенной.

Параметрами схемы являются:

$p$  – большое простое число.  $p$  должно быть больше любого секрета, который предполагается разделять в этой схеме ( $p > m$ ). Тогда  $m \in \mathbb{Z}_p$ .

$n$  – число долей секрета.

$k$  – минимальный размер разрешенной группы.

#### Подготовительная фаза:

Случайным образом раздающий выбирает числа  $x_2^*, x_3^*, \dots, x_k^* \in \mathbb{Z}_p$ . Тем самым получена секретная точка  $Q = (m, x_2^*, x_3^*, \dots, x_k^*)$ .

### Фаза раздачи секрета:

Для  $i$ -го ( $i=1, \dots, n$ ) участника раздающий выбирает случайные равномерно распределенные на  $Z_p$  коэффициенты  $a_{1i}, a_{2i}, a_{3i}, \dots, a_{ki}$  и вычисляет  $b_i = a_{1i}m + a_{2i}x_2^* + \dots + a_{ki}x_k^* \pmod p$ .

После чего раздающий  $D$  посылает абоненту  $P_i$  сравнение

$$a_{1i}x_1 + a_{2i}x_2 + \dots + a_{ki}x_k \equiv b_i \pmod p$$

(а, точнее, его коэффициенты) с неизвестными  $x_1, x_2, \dots, x_k$ .

При этом раздающий должен следить за тем, чтобы любые  $k$  сравнений были линейно независимы.

### Фаза восстановления секрета:

Собравшись вместе,  $k$  абонентов могут составить из своих сравнений систему

$$a_{11}x_1 + a_{21}x_2 + \dots + a_{k1}x_k \equiv b_1 \pmod p$$

$$a_{12}x_1 + a_{22}x_2 + \dots + a_{k2}x_k \equiv b_2 \pmod p$$

.....

$$a_{1k}x_1 + a_{2k}x_2 + \dots + a_{kk}x_k \equiv b_k \pmod p$$

решив которую они отыщут точку  $Q$ , первая координата которой как раз и будет секретом  $m$ .

### Замечание 1.

При достаточно большом  $p$  при условии, что коэффициенты сравнений выбираются случайным образом, проверкой линейной независимости сравнений можно пренебречь, так как вероятность нарушения этого условия крайне низка.

### Замечание 2.

Если свои доли секретов объединят  $t < k$  абонентов (то есть неразрешенная группа), то система сравнений, составленная ими, даст в качестве решения не точку, а множество точек, лежащих на гиперплоскости порядка  $t$  в  $k$ -мерном пространстве, и определить наверняка, какая из них содержит секрет,

невозможно. В этом случае секретом может оказаться любое значение из  $Z_p$  с равной вероятностью.

### Пример.

Построим (3,3) – пороговую схему Блекли. В качестве секрета возьмем  $m=5$ , в качестве модуля  $p=11$ . В качестве секретной точки выберем  $Q=(m=5, x_2^*=2, x_3^*=1)$ .

Фаза раздачи:

Вычислим доли секрета для каждого абонента:

Для  $P_1$ : Выберем  $a_{11}=1, a_{21}=1, a_{31}=1$ . Тогда  $b_1=1 \cdot 5 + 1 \cdot 2 + 1 \cdot 1 \bmod 11 = 8$ .

Для  $P_2$ : Выберем  $a_{12}=3, a_{22}=2, a_{32}=7$ . Тогда  $b_2=3 \cdot 5 + 2 \cdot 2 + 7 \cdot 1 \bmod 11 = 4$ .

Для  $P_3$ : Выберем  $a_{13}=8, a_{23}=1, a_{33}=10$ . Тогда  $b_3=8 \cdot 5 + 1 \cdot 2 + 10 \cdot 1 \bmod 11 = 8$ .

Заметим, что после фазы раздачи у каждого из участников оказалось по 4 коэффициента линейного сравнения.

Фаза восстановления:

Участники  $P_1, P_2, P_3$  составляют систему линейных сравнений. У каждого из них есть по одному сравнению. Получившаяся система состоит из трех сравнений и имеет три неизвестных. Если матрица системы невырожденная, то система имеет единственное решение.

$$\begin{cases} x_1 + x_2 + x_3 \equiv 8 \pmod{11} \\ 3x_1 + 2x_2 + 7x_3 \equiv 4 \pmod{11} \\ 8x_1 + x_2 + 10x_3 \equiv 8 \pmod{11} \end{cases}$$

Представим эту систему в виде расширенной матрицы и решим методом Гаусса, учитывая, что операции производятся в  $Z_{11}$ .

$$\left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 3 & 2 & 7 & 4 \\ 8 & 1 & 10 & 8 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 0 & -1 & 4 & -20 \\ 0 & -7 & 2 & -56 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 0 & 10 & 4 & 2 \\ 0 & 4 & 2 & 10 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 0 & 1 & 7 & 9 \\ 0 & 4 & 2 & 10 \end{array} \right)_{11} \Rightarrow$$

$$\left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 0 & 1 & 7 & 9 \\ 0 & 0 & 7 & 7 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & 0 & 7 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \end{array} \right)_{11}.$$

Итак, решив систему сравнений, участники нашли единственную точку  $x_1=5$ ,  $x_2=2$ ,  $x_3=1$ , которая совпадает с точкой Q, и восстановили секрет  $m=5$ .

Посмотрим, что случится, если свои секреты объединят не три, а два участника. Удастся ли им узнать значение  $m$ , или хотя бы очертить круг возможных его значений?

Пусть секрет пытаются восстановить участники  $P_1$  и  $P_2$ . Тогда расширенная матрица системы, составленной ими, будет

$$\left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 3 & 2 & 7 & 4 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 0 & -1 & 4 & -20 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 1 & 1 & 8 \\ 0 & 1 & 7 & 9 \end{array} \right)_{11} \Rightarrow \left( \begin{array}{ccc|c} 1 & 0 & 1 & 10 \\ 0 & 1 & 7 & 9 \end{array} \right)_{11}$$

Таким образом, решение, полученное этими участниками, будет

$$m = x_1 = 10 - x_3 \pmod{11}$$

$$x_2 = 9 - 7x_3 \pmod{11}$$

Но тогда  $m$  может принимать любое значение от 0 до 10, то есть участники  $P_1$  и  $P_2$ , собравшись вместе, не увеличили своих знаний о секрете.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Практическое занятие №23.**

### **Протокол аутентификации для систем электронных платежей.**

#### **Цель и содержание:**

1. Углубить знания по основам построения скрытых каналов, используемых при шифровании.
2. Изучить правила организации скрытого канала на основе схемы Густава-Симмонсона

**Формируемые компетенции:** ОПК-6, ОПК-9.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Практическое занятие №24.**

### **Отечественная криптографическая хеш-функция.**

#### **Цель и содержание:**

1. Углубить знания по основам построения протоколов аутентификации.
2. Изучить правила построения протокола аутентификации Фейге-Фиат-Шамира

#### **Формируемые компетенции: ОПК-6, ОПК-9.**

#### **Теоретический материал**

Повысить имитостойкость системы защиты данных от НСД можно за счет построения системы на основе протокола с нулевым разглашением, которые используются в протоколах аутентификации. Особое место среди таких протоколов занимают протоколы аутентификации типа «запрос-ответ» в котором одна сторона (претендент) доказывает свою идентичность другой стороне (проверяющему) посредством демонстрации знания определенного секрета. Претендент должен ответить на запрос, меняющийся во времени, причем ответ должен зависеть и от значения его секрета и от запроса.

В работе [1] приведены наиболее известные вычислительно-сложные задачи теории чисел, на которых базируется протокол защиты данных от НСД. Предложенный алгоритм аутентификации, используемый в разрабатываемой системе, основан на доказательстве с нулевым разглашением и относится к задаче распознавания Диффи-Хеллмана (DDHP). Стойкость задач Диффи-Хеллмана базируется на сложности вычисления дискретного логарифма. В настоящее время решить данную задачу дискретного логарифмирования за минимальное время можно только с помощью субэкспоненциального алгоритма, что значительно больше, чем время перебора возможных коммутаций, выполняемых шифровальной колодкой, приведенной в [1].

Рассмотрим реализацию протокола аутентификации Fiat-Shamir, алгоритм которого приведен в [6,7], с использованием параллельных кодов СОК. В данном протоколе есть предварительный этап.

1. Претендент использует в качестве модулей кода СОК большие простые числа  $m_1$  и  $m_2$ . В этом случае рабочий диапазон кода  $M_2$  определяется выражением (1). Данный диапазон  $M_2$  используется в качестве открытого ключа.

2. Претендент определяет свой секретный ключ – число  $S$ , которое меньше диапазона  $M_2$  и взаимно простое с ним. Секретный ключ представляется в коде СОК  $S = (S_1, S_2)$ .

3. Претендент находит квадратичный вычет  $G$  по модулю  $M_2$ . Данное число представляется в коде СОК  $G = (G_1, G_2)$ . При этом должны быть справедливыми условия

$$G_i = S_i^2 \bmod m_i. \quad (1)$$

$$G_i G_i^{-1} \equiv 1 \bmod m_i. \quad (2)$$

Рассмотрим второй этап протокола – аутентификацию претендента.

1. Претендент выбирает случайное число  $H$ , а затем преобразует его в код СОК  $H = (H_1, H_2)$ . Используя данные числа выполняется

$$Q_i \equiv H_i^2 \bmod m_i. \quad (3)$$

Полученный в СОК результат  $Q = (Q_1, Q_2)$  передается запросчику.

2. Запросчик, после получения кодограммы СОК  $Q = (Q_1, Q_2)$ , выбирает случайное число  $L \in \{0, 1\}$ , которое передается претенденту.

3. Претендент, получив вопрос  $L \in \{0, 1\}$ , определяет «ответ» в СОК

$$V_i = H_i S_i^L \bmod m_i. \quad (4)$$

Ответ в виде кода поступает запросчику.

4. Получив ответ  $V = (V_1, V_2)$ , запросчик приступает к его проверке. При этом используются два выражения

$$B_i = V_i^2 \bmod m_i. \quad (5)$$

$$B_i = (V_i^2 G_i) \bmod m_i. \quad (6)$$

Выражение (5) используется, если был задан вопрос  $L = 0$ , а выражение (6) – если был задан вопрос  $L = 1$ . Претенденту будет присвоен статус «свой» только при выполнении сравнения

$$B_i \equiv Q_i \bmod m_i. \quad (7)$$

В качестве недостатка протокола аутентификации Fiat-Shamir можно выделить значительные временные затраты на опознавание претендента. Это связано с тем, что для обеспечения требуемой имитостойкости протокола необходимо выполнить от 30 до 40 раундов. Поэтому данный протокол нецелесообразно применять в системах аутентификации космического аппарата. Рассмотрим протоколы, которые позволяют определить статус спутника за один раунд.

Рассмотрим пример реализации данного протокола опознавания. Пусть выбраны два простых числа  $Q = 5$ ,  $G = 7$ . Тогда их произведение будет равно  $M = 35$ .

Определим квадратичные вычеты по модулю  $M = 35$ . Квадратным вычетом является число, удовлетворяющее следующему выражению

$$S^2 \equiv H \bmod M \quad (A.1)$$

где  $1 \leq S \leq n$ .

В ходе проведенных исследований были найдены следующие значения квадратичных вычетов

$$S = \{1, 4, 6, 9, 11, 14, 15, 16, 21, 25, 29, 30, 34\}.$$

Если значение числа  $H = 16$ . Тогда обратное мультипликативное значение числа равно его  $H^{-1} = 11$ , так как

$$(16 \cdot 11) \bmod 35 = 176 \bmod 35 = 1$$

Вычислим значение секретного ключа, используя выражение (2). Тогда получаем  $S = 9$ , так как  $9^2 \bmod 35 = 81 \bmod 35 = 11$ .

Таким образом, получили открытые ключи  $(M, H) = (35, 11)$ .

Рассмотрим алгоритм опознавания для данного протокола. Данный порядок операций, выполненный один раз называется аккредитацией.

1. Пусть претендент Р выбирает случайное числа К из условия  $K \in \{1, 2, \dots, M-1\}$ . Пусть  $K=8$ . Затем претендент вычисляет значение используя (4). Тогда имеем

$$E = K^2 \bmod M = 8^2 \bmod 35 = 29$$

Данное значение  $E=29$  передается проверяющему В как свидетельство.

2. Проверяющий абонент V выбирает случайное число  $B = 0$ . Данное значение передается претенденту Р.

3. Претендент Р производит вычисление, используя равенство (5). Так как значение проверочного бита  $B = 0$ , то проверяющему абоненту V будет передано число  $Y = K = 8$ .

4. Проверяющий V, получив ответ, производит проверку согласно условия (2.6). Так как значение проверочного бита  $B = 0$ , то абонент V вычисляет

$$L = Y^2 \bmod M = 8^2 \bmod 35 = 29$$

Так как полученное значение делает истинным выражение (2.8), т.е.  $L = E = 29 \bmod 35$ , то проверяющий абонент V делает вывод - претендент Р имеет статус «свой».

Рассмотрим ситуацию, когда проверяющий V пересылает претенденту Р значение проверочного бита  $B = 1$ . Тогда получаем.

3. Претендент Р производит вычисление, используя равенство (2.5). Так как значение  $B = 1$ , то проверяющему абоненту V будет передано число

$$\hat{I} = (KS) \bmod M = (8 \cdot 9) \bmod 35 = 2$$

4. Проверяющий V, получив ответ, производит проверку согласно условия (2.7). Так как значение проверочного бита  $B = 1$ , то абоненту V необходимо вычислить

$$L = (Y^2 H) \bmod M = (2^2 \cdot 16) \bmod 35 = 29$$

Так как полученное значение делает истинным выражение (2.8), т.е.  $L = E = 29 \bmod 35$ , то проверяющий абонент V делает вывод - претендент Р имеет статус «свой».

Студенты получают задание для индивидуального исследования протокола аутентификации Фиат-Шамира.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 25

### Скрытый канал Эль-Гамалья

#### Цель содержание:

1. Углубить знания по основам построения протоколов аутентификации.
2. Изучить правила построения протокола аутентификации Guillou-Quisquater

#### Формируемые компетенции: ОПК-6, ОПК-9.

#### Теоретический материал

Среди протоколов аутентификации особое место занимает протокол аутентификации Guillou-Quisquater. Данный протокол базируется на доказательстве с нулевым разглашением. Проведем его модификацию с использованием параллельного кода системы остаточных классов.

Перед началом функционирования каждый из претендентов получает личный идентификатор  $I_p$ . Можно положить, что значение данного идентификатора будет определяться номером космического аппарата. Кроме того в состав  $I_p$  можно включить и другие сведения, например, тип космического аппарата, дата начала работы на орбите. На предварительном этапе претендент реализует процедуру вычисления открытого и секретного ключей.

1. В качестве оснований кода СОК претенденту необходимо выбрать большие простых числа  $m_1$  и  $m_2$ . Их произведение дает диапазон кода СОК  $\dot{I}_k = m_1 m_2$ .

2. Претендент определяет значение хеш-функции  $F$ , где в качестве аргумента используется идентификатор  $I_p$

$$H = F(I_p). \quad (1)$$

Вычисленное значение представляется в СОК  $\dot{I} = (\dot{I}_1, \dot{I}_2)$ .

3. Претендент вычисляет значение секретного ключа  $L = (L_1, L_2)$ , представленного в СОК. Для этого используется сравнение

$$H_i L_i^Y \equiv 1 \bmod m_i . \quad (2)$$

Наряду с секретным ключом в протоколе используется открытый ключ  $(m_1 \parallel m_2 \parallel Y_1 \parallel Y_2 \parallel H_1 \parallel H_2)$ .

Процедура аутентификации претендента состоит из этапов.

1. Претендент осуществляет выбор случайного числа  $P$ , где  $D < I_k - 1$ . Это число представляется в коде СОК  $P = (P_1, P_2)$ . Затем претендент производит вычисление

$$K_i = P_i^Y \bmod m_i . \quad (3)$$

Вычисленное значение параметра  $K = (K_1 \parallel K_2)$  пересылается проверяющему.

2. Проверяющий после получения  $K = (K_1 \parallel K_2)$  производит выбор случайного числа  $B$ , где  $A \leq Y - 1$ . Данное число выступает в качестве запроса. Оно передается претенденту.

3. Претендент после получения запроса  $B$  приступает к вычислению ответа

$$D_i = P_i L_i^B \bmod m_i . \quad (4)$$

Ответы, представленные в коде СОК  $(D_1 \parallel D_2)$ , передаются проверяющему.

4. Проверяющий после получения ответов  $(D_1 \parallel D_2)$  выполняет проверку претендента. Для этого вычисляется

$$K_i^* = D_i^Y H_i^B \bmod m_i . \quad (5)$$

Если выполняется равенство  $K_i^* = K_i$ , то претенденту присваивается статус «свой». В противном случае – у претендента получается статус «чужой».

Характерной чертой рассмотренного протокола аутентификации Guillou-Quisquater являются меньшие временные затраты по сравнению с протоколом Fiat-Shamir. Следовательно, данный протокол обладает большим потенциалом для реализации в системах опознавания «свой-чужой» для низкоорбитальных систем спутниковой связи.

Рассмотрим выполнение разработанного протокола аутентификации Guillou-Quisquater, реализованного в параллельном коде СОК.

1. В качестве оснований кода СОК претендент выбрал простые числа  $m_1 = 7, m_2 = 13$ . Их произведение дает диапазон  $I_k = 91$ .

2. Пусть хеш-функция, идентификатора претендента  $I_p$  равна  $H = 5$ . Вычисленное значение представляем в СОК  $\dot{I} = (5, 5)$ .

3. Претендент вычисляет значение секретного ключа, представленного в СОК. Для этого используется сравнение (2).

$$\begin{aligned} 5 \cdot L_1^{Y_1} &\equiv 1 \pmod{m_1} \\ 5 \cdot L_2^{Y_2} &\equiv 1 \pmod{m_2} \end{aligned}$$

Получили, что  $(L_1^{Y_1}, L_2^{Y_2}) = (3, 8)$ . Получаем, что секретный ключ равен  $L = (L_1, L_2) = (3, 5)$ . При этом показатели степени  $Y = (Y_1, Y_2) = (0, 7)$ .

Тогда открытый ключ  $(m_1 \parallel m_2 \parallel Y_1 \parallel Y_2 \parallel H_1 \parallel H_2) = (7 \parallel 13 \parallel 0 \parallel 7 \parallel 5 \parallel 5)$ .

Процедура аутентификации претендента.

1. Претендент выбирает случайного числа  $P = 8$ , и представляет в коде СОК  $P = (1, 8)$ . Затем претендент производит вычисление согласно (3)

$$\begin{aligned} K_1 &= P_1^{Y_1} \pmod{m_1} = 1^0 \pmod{7} = 1 \\ K_2 &= P_2^{Y_2} \pmod{m_2} = 8^7 \pmod{13} = 5 \end{aligned}$$

Это значение  $K = (K_1 \parallel K_2) = (1 \parallel 5)$  пересылается проверяющему.

2. Проверяющий производит выбор случайного числа  $B = 5 = (5 \parallel 5)$ , которое передается претенденту.

3. Претендент, получив  $B = 5 = (5 \parallel 5)$ , вычисляет ответ согласно (4)

$$\begin{aligned} D_1 &= P_1 L_1^{B_1} \pmod{m_1} = (1 \cdot 3^5) \pmod{7} = 5, \\ D_2 &= P_2 L_2^{B_2} \pmod{m_2} = (8 \cdot 5^5) \pmod{13} = 1. \end{aligned}$$

Ответы  $(D_1 \parallel D_2) = (5 \parallel 1)$  передаются проверяющему.

4. Проверяющий после получения ответов выполняет проверку претендента согласно(5)

$$K_1^* = D_1^{Y_1} H_1^{B_1} \bmod m_1 = (5^0 \cdot 5^5) \bmod 7 = 1,$$

$$K_2^* = D_2^{Y_2} H_2^{B_2} \bmod m_2 = (1^7 \cdot 5^1) \bmod 13 = 5.$$

Так как выполнилось равенство  $K_i^* = K_i = (1 \parallel 5)$ , то претенденту присваивается статус «свой».

Студенты получают задание для индивидуального исследования протокола аутентификации Guillou-Quisquater

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Практическое занятие № 22

### ПРОТОКОЛ АУТЕНТИФИКАЦИИ ШНОРРА

#### Цель содержание:

1. Углубить знания по основам построения протоколов аутентификации.
2. Изучить правила построения протокола аутентификации Фейге-Фиат-Шамира

**Формируемые компетенции:** ОПК-6, ОПК-9.

#### Теоретический материал

Устранить многораундовый характер протокола аутентификации Фиат-Шамира может протокол аутентификации Шнорра,. Рассмотрим реализацию данного протокола в модулярном коде. Для выполнения данного протокола претендент Р должен обладать определенной идентификационной информацией IP. В состав этих данных может входить тип КА, его личный идентификатор, срок выхода на орбиту, и.т.д.

На этапе получения открытого и секретного выполняется алгоритм, который состоит из следующих шагов.

Выбираем в качестве оснований простые числа  $m_1, m_2, \dots, m_k$ , для которых ищем простое число  $q_i$ , - делитель  $m_i$ . Затем определяется число  $g_i$ , которое удовлетворяет условию

$$g_i^{q_i} \equiv 1 \pmod{m_i}. \quad (1)$$

Секретным ключом выбирают число  $X=(x_1, x_2, \dots, x_k)$ , которое удовлетворяет условию

$$X < Q = \prod_{i=1}^k q_i. \quad (2)$$

Открытым ключом является число  $Y = (y_1, y_2, \dots, y_k)$ , где справедливо

$$y_i = g_i^{-x_i} \pmod{m_i}. \quad (3)$$

Протокол аутентификации выполняется следующим образом.

1. Претендент А выбирает случайное число  $S = (s_1, s_2, \dots, s_k)$ , которое удовлетворяет условию  $S < Q$ . Затем осуществляется вычисление числа

$R = (r_1, r_2, \dots, r_k)$ , согласно

$$r_i = g_i^{s_i} \bmod m_i. \quad (4)$$

Вычисленное значение передается проверяющему абоненту В.

2. Абонент В выбирает случайное число  $E = (e_1, e_2, \dots, e_k) \in \{1, 2, \dots, 2^t - 1\}$ , где  $t$  - некоторый параметр. Данное число посылается абоненту А.

3. Абонент А, получив число  $E$ , вычисляет число  $D = (d_1, d_2, \dots, d_k)$ , согласно равенства

$$d_i = (s_i + e_i x_i) \bmod q_i. \quad (5)$$

Вычисленное значение  $D = (d_1, d_2, \dots, d_k)$ , посылается абоненту В.

4. Абонент В, получив ответ  $D = (d_1, d_2, \dots, d_k)$ , проверяет правильность ответа

$$W_i = g_i^{d_i} y_i^{e_i} \bmod m_i. \quad (6)$$

Если вычисленное значение  $W = (w_1, w_2, \dots, w_k)$ , совпадает с числом  $R = (r_1, r_2, \dots, r_k)$ , то претендент А – «свой». В противном случае - абонент А является «чужим».

Рассмотрим выполнение протокола аутентификации Шнорра в модулярном коде. Выбираем основания модулярного кода МК

$m_1 = 11, m_2 = 23, m_3 = 29$ .

Рабочий диапазон системы равен  $P_{\text{раб}} = 7337$ .

Определим делители оснований МК  $q_1 = 5, q_2 = 11, m_3 = 7$ , которые имеют  $Q = 385$ .

Воспользуемся условием (5) и выберем  $g_1 = 3, g_2 = 2, g_3 = 7$ .

Пусть секретный ключ абонента А равен  $X = (3, 5, 5)$ . Тогда первая часть открытого ключа, представленного в модулярном коде, будет равна

$$y_1 = g_1^{-x_1} \bmod m_1 = 3^{-3} \bmod 11 = 3^7 \bmod 11 = 9;$$

$$y_2 = g_2^{-x_2} \bmod m_2 = 2^{-5} \bmod 23 = 2^6 \bmod 23 = 18;$$

$$y_3 = g_3^{-x_3} \bmod m_3 = 7^{-5} \bmod 29 = 2^{23} \bmod 29 = 20.$$

В этом случае открытый ключ имеет вид  $(y_i, p_i, g_i) = ((9, 18, 20)(11, 23, 29)(3, 2, 7))$ .

1. Претендент А выбирает числа  $S = (2, 7, 3)$  и вычисляет число  $R = (r_1, r_2, r_3)$ , согласно

$$r_1 = g_1^{s_1} \bmod m_1 = 3^2 \bmod 11 = 9;$$

$$r_2 = g_2^{s_2} \bmod m_2 = 2^7 \bmod 23 = 13;$$

$$r_3 = g_3^{s_3} \bmod m_3 = 7^3 \bmod 29 = 24.$$

Вычисленное значение  $R = (9, 13, 24)$  передается запросчику В.

2. Запросчик В выбирает число  $E = (4, 8, 4)$  которое передается претенденту А.

3. Претендент А вычисляет ответ на поставленный вопрос е, используя равенство (9)

$$d_1 = (s_1 + e_1 x_1) \bmod q_1 = (2 + 4 \cdot 3) \bmod 5 = 4;$$

$$d_2 = (s_2 + e_2 x_2) \bmod q_2 = (7 + 8 \cdot 5) \bmod 11 = 3;$$

$$d_3 = (s_3 + e_3 x_3) \bmod q_3 = (3 + 4 \cdot 5) \bmod 7 = 2.$$

Вычисленное значение  $D = (4, 3, 2)$ , посылается абоненту В.

4. Абонент В проверяет правильность ответа согласно (10)

$$w_1 = g_1^{s_1} y_1^{e_1} \bmod m_1 = (3^4 \cdot 9^4) \bmod 11 = 3^2 \bmod 11 = 9.$$

$$w_2 = g_2^{s_2} y_2^{e_2} \bmod m_2 = (2^3 \cdot 18^8) \bmod 23 = 13.$$

$$w_3 = g_3^{s_3} y_3^{e_3} \bmod m_3 = (7^2 \cdot 20^4) \bmod 29 = 24.$$

Так как справедливо, что  $W = (9, 13, 24) = R$ , то статус претендента А - «свой».

Студенты получают задание для индивидуального исследования протокола аутентификации Шнорра

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная: 1-5*

*Дополнительная: 1-6*

## Практическое занятие № 23

# ПРОТОКОЛ АУТЕНТИФИКАЦИИ с НУЛЕВЫМ РАЗГЛАШЕНИЕМ ЗНАНИЙ, РЕАЛИЗОВАННЫЙ В МОДУЛЯРНОМ КОДЕ

### Цель содержание:

1. Углубить знания по основам построения протоколов аутентификации.
2. Изучить правила построения протокола аутентификации с нулевым разглашением, реализованным в модулярном коде

### Формируемые компетенции: ОПК-6, ОПК-9.

### Теоретический материал

Анализ модифицированных протоколов Shnor и Guillou-Quisquater показал, что этап аутентификации включает в себя четыре шага, необходимые для определения статуса претендента. Сократить число таких шагов позволяет протокол аутентификации, рассмотренный на лабораторной работе. В данном протоколе процедура опознавания сокращена до трех шагов – вопрос проверяющего, вычисление ответа претендентом и проверка правильности ответа. Однако данный протокол для выполнения аутентификации использует большое простое число, по модулю которого и выполняются все вычисления. Произведем разработку данного протокола в коде СОК.

Предварительный этап протокола.

1. Претендент имеет секретный ключ  $W$ , начальный параметр  $C$  для вычисления  $n$ -го сеансового ключа  $C(n)$ , где  $n = 1, 2, \dots$ , число  $B$ , с помощью которого итерационно вычисляют аргумент  $B(n)$ , позволяющий определить факт двойного использования сеансового ключа. Данные параметры представляются в коде СОК  $(W_1(n), W_2(n), \dots, W_k(n)), (C_1(n), C_2(n), \dots, C_k(n)), (\hat{A}_1(n), \hat{A}_2(n), \dots, \hat{A}_k(n))$ . При этом основания кода СОК выбираются такими, чтобы они имели одинаковый порождающий элемент  $\dot{a}_1 = \dot{a}_2 = \dots = \dot{a}_k$ .

2. Претендент перед началом  $n$ -го сеанса обмена данными вычисляет исходный статус КА

$$\begin{aligned} U_1(n) &= (a_1^{W_1(n)} a_1^{C_1(n)} a_1^{B_1(n)}) \bmod m_1, \\ &\vdots \\ U_k(n) &= (a_k^{W_k(n)} a_k^{C_k(n)} a_k^{B_k(n)}) \bmod m_k. \end{aligned} \quad (1)$$

3. Претендент выбирает случайные числа  $\Delta W(n), \Delta C(n), \Delta \hat{A}(n)$ , где  $1 < \Delta W(n), \Delta C(n), \Delta \hat{A}(n) < \prod_{i=1}^k \varphi(m_i) - 1$ . Представляет их в параллельном коде СОК, а затем изменяет параметры  $W, C(n), \hat{A}(n)$  с помощью выражения

$$\begin{aligned} W_i^*(n) &= (W_i + \Delta W_i(n)) \bmod \varphi(m_i), \\ C_i^*(n) &= (C_i(n) + \Delta C_i(n)) \bmod \varphi(m_i), \\ B_i^*(n) &= (B_i(n) + \Delta B_i(n)) \bmod \varphi(m_i). \end{aligned} \quad (2)$$

4. Претендент вычисляет измененный статус космического аппарата

$$\begin{aligned} U_1^*(n) &= (a_1^{W_1^*(n)} a_1^{C_1^*(n)} a_1^{B_1^*(n)}) \bmod m_1, \\ &\vdots \\ U_k^*(n) &= (a_k^{W_k^*(n)} a_k^{C_k^*(n)} a_k^{B_k^*(n)}) \bmod m_k. \end{aligned} \quad (3)$$

Этап аутентификации претендента

1. Проверяющая сторона выбирает «вопрос» в виде случайного числа, удовлетворяющего условию  $\hat{I}(n) < \prod_{i=1}^k \varphi(m_i) - 1$ , и представляет его в коде СОК

$\hat{I} = (\hat{I}_1(n), \hat{I}_2(n), \dots, \hat{I}_k(n))$ . Данный «вопрос» передается претенденту.

2. Претендент, получив вопрос, осуществляет вычисление ответов

$$\begin{cases} Q_1^1(n) = (W_1^*(n) - H_1 W_1(n)) \bmod \varphi(m_1), \\ \vdots \\ Q_k^1(n) = (W_k^*(n) - H_k W_k(n)) \bmod \varphi(m_k). \end{cases} \quad (4)$$

$$\begin{cases} Q_1^2(n) = (C_1^*(n) - H_1 C_1(n)) \bmod \varphi(m_1), \\ \vdots \\ Q_k^2(n) = (C_k^*(n) - H_k C_k(n)) \bmod \varphi(m_k). \end{cases} \quad (5)$$

$$\begin{cases} Q_1^3(n) = (B_1^*(n) - H_1 B_1(n)) \bmod \varphi(m_1), \\ \vdots \\ Q_k^3(n) = (B_k^*(n) - H_k B_k(n)) \bmod \varphi(m_k). \end{cases} \quad (6)$$

Претендент пересылает проверяющему представленные в коде СОК  $U(n), U^*(n), Q^1(n), Q^2(n), Q^3(n)$ .

3. Проверяющий осуществляет проверку ответов

$$\begin{aligned} R_1(n) &= (U_1^{H_1(n)} a_1^{Q_1^1(n)} a_1^{Q_1^2(n)} a_1^{Q_1^3(n)}) \bmod m_1, \\ &\vdots \\ R_k(n) &= (U_k^{H_k(n)} a_k^{Q_k^1(n)} a_k^{Q_k^2(n)} a_k^{Q_k^3(n)}) \bmod m_k. \end{aligned} \quad (7)$$

Если после проверки получается

$$(R_1(n), R_2(n), \dots, R_k(n)) = (U_1^*(n), U_2^*(n), \dots, U_k^*(n)), \quad (8)$$

то проверяющий присваивает претенденту статус «свой».

Проведем сравнительный анализ рассмотренных протоколов аутентификации, реализованных в коде СОК.

Рассмотрим выполнение разработанного протокола аутентификации в модулярном коде. Предварительный этап протокола.

1. Претендент выбирает основания СОК  $m_1 = 19, m_2 = 29, m_3 = 39$ , так как они имеют порождающий элемент  $\dot{a}_1 = \dot{a}_2 = \dot{a}_3 = 2$ . Учитывая диапазон кода

$$M_3 = \prod_{i=1}^3 m_i = 21489, \text{ выбираем секретный ключ } W = 10102 = (13, 10, 1),$$

сеансовый ключ  $C(1) = 365 = (2, 17, 32)$  и число  $B(1) = 452 = (15, 17, 8)$ .

2. Претендент перед началом первого сеанса обмена данными вычисляет исходный статус КА согласно (1)

$$\begin{aligned} U_1(1) &= (a_1^{W_1(n)} a_1^{C_1(n)} a_1^{B_1(n)}) \bmod m_1 = |2^{13} \cdot 2^2 \cdot 2^{15}|_{19}^+ = |2^{12}|_{19}^+ = 11, \\ U_2(1) &= (a_2^{W_2(n)} a_2^{C_2(n)} a_2^{B_2(n)}) \bmod m_2 = |2^{10} \cdot 2^{17} \cdot 2^{17}|_{29}^+ = |2^{16}|_{29}^+ = 25, \\ U_3(1) &= (a_3^{W_3(n)} a_3^{C_3(n)} a_3^{B_3(n)}) \bmod m_3 = |2^1 \cdot 2^{32} \cdot 2^8|_{37}^+ = |2^5|_{37}^+ = 32. \end{aligned}$$

3. Претендент выбирает случайные числа  $\Delta W(n) = 221 = (5, 25, 5)$ ,  $\Delta \tilde{N}(n) = 101 = (11, 17, 29)$ ,  $\Delta \hat{A}(n) = 58 = (4, 2, 22)$ . Изменяем параметры  $W, C(n), \hat{A}(n)$  согласно (2)

$$\begin{aligned} W^*(n) &= (|13 + 5|_{18}^+, |10 + 25|_{28}^+, |1 + 5|_{36}^+) = (0, 7, 6), \\ C^*(n) &= (|4 + 11|_{18}^+, |17 + 17|_{28}^+, |32 + 29|_{36}^+) = (15, 6, 25), \\ B^*(n) &= (|15 + 4|_{18}^+, |17 + 2|_{28}^+, |8 + 22|_{36}^+) = (1, 19, 30). \end{aligned}$$

4. Претендент вычисляет измененный статус КА согласно (3)

$$\begin{aligned}
U_1^*(1) &= (a_1^{W_1^*(n)} a_1^{C_1^*(n)} a_1^{B_1^*(n)}) \bmod m_1 = |2^0 \cdot 2^{15} \cdot 2^1|_{19}^+ = |2^{16}|_{19}^+ = 5, \\
U_2^*(1) &= (a_2^{W_2^*(n)} a_2^{C_2^*(n)} a_2^{B_2^*(n)}) \bmod m_2 = |2^7 \cdot 2^6 \cdot 2^{19}|_{29}^+ = |2^4|_{29}^+ = 16, \\
U_3^*(1) &= (a_3^{W_3^*(n)} a_3^{C_3^*(n)} a_3^{B_3^*(n)}) \bmod m_3 = |2^6 \cdot 2^{25} \cdot 2^{30}|_{37}^+ = |2^{25}|_{37}^+ = 20.
\end{aligned}$$

Этап аутентификации претендента.

1. Проверяющий выбирает «вопрос»  $\dot{I} = 367 = (6, 19, 34)$ , который передает претенденту.

2. Претендент, получив вопрос, вычисляет ответы согласно (4)-(6)

$$\begin{cases}
Q_1^1(1) = |W_1^*(1) - H_1 W_1(1)|_{\varphi(m_1)}^+ = |0 - 6 \cdot 13|_{18}^+ = 12, \\
Q_2^1(1) = |W_2^*(1) - H_2 W_2(1)|_{\varphi(m_2)}^+ = |7 - 19 \cdot 10|_{28}^+ = 13, \\
Q_3^1(1) = |W_3^*(1) - H_3 W_3(1)|_{\varphi(m_3)}^+ = |6 - 34 \cdot 1|_{36}^+ = 8.
\end{cases}$$

$$\begin{cases}
Q_1^2(1) = |\tilde{N}_1^*(1) - \dot{I}_1 \tilde{N}_1(1)|_{\varphi(m_1)}^+ = |15 - 6 \cdot 4|_{18}^+ = 9, \\
Q_2^2(1) = |\tilde{N}_2^*(1) - H_2 \tilde{N}_2(1)|_{\varphi(m_2)}^+ = |6 - 19 \cdot 17|_{28}^+ = 19, \\
Q_3^1(1) = |\tilde{N}_3^*(1) - H_3 \tilde{N}_3(1)|_{\varphi(m_3)}^+ = |25 - 34 \cdot 32|_{36}^+ = 17.
\end{cases}$$

$$\begin{cases}
Q_1^3(1) = |\hat{A}_1^*(1) - H_1 \hat{A}_1(1)|_{\varphi(m_1)}^+ = |1 - 6 \cdot 15|_{18}^+ = 1, \\
Q_2^3(1) = |\hat{A}_2^*(1) - H_2 \hat{A}_2(1)|_{\varphi(m_2)}^+ = |19 - 19 \cdot 17|_{28}^+ = 4, \\
Q_3^3(1) = |\hat{A}_3^*(1) - H_3 \hat{A}_3(1)|_{\varphi(m_3)}^+ = |30 - 34 \cdot 8|_{36}^+ = 10.
\end{cases}$$

Претендент пересылает проверяющему представленные в коде СОК  $U(1), U^*(1), Q^1(1), Q^2(1), Q^3(1)$ .

3. Проверяющий осуществляет проверку ответов согласно (7)

$$\begin{aligned}
R_1(1) &= |U_1^{H_1(n)} a_1^{Q_1^1(n)} a_1^{Q_1^2(n)} a_1^{Q_1^3(n)}|_{m_1}^+ = |11^6 \cdot 2^{12} \cdot 2^9 \cdot 2^1|_{19}^+ = 5, \\
R_2(1) &= |U_2^{H_2(n)} a_2^{Q_2^1(n)} a_2^{Q_2^2(n)} a_2^{Q_2^3(n)}|_{m_2}^+ = |25^{19} \cdot 2^{12} \cdot 2^{19} \cdot 2^4|_{29}^+ = 16, \\
R_3(1) &= |U_3^{H_3(n)} a_3^{Q_3^1(n)} a_3^{Q_3^2(n)} a_3^{Q_3^3(n)}|_{m_3}^+ = |32^{34} \cdot 2^8 \cdot 2^{17} \cdot 2^{10}|_{37}^+ = 20.
\end{aligned}$$

Так как в результате проверки получено равенство

$$(R_1(1), R_2(1), R_3(1)) = (U_1^*(1), U_2^*(1), U_3^*(1)) = (5, 16, 20),$$

то проверяющий присваивает претенденту статус «свой».

Студенты получают задание для индивидуального исследования протокола аутентификации с нулевым разглашением знаний, реализованный в модулярном коде

**Список литературы, рекомендуемый к использованию по данной теме.**

**1. Основная литература:**

- Рябко Б.Я. Фионов А.Н. Криптографические методы защиты информации. – М.: "Горячая линия-Телеком", 2012. - 229 с.
- Музыкантский А.И., Фурин В.В. Лекции по криптографии. – М.: МЦНМО, 2011. - 68 с.
- Глухов М. М. Круглов И. А., Пичкур А. Б., Черемушкин А. В. Введение в теоретико-числовые методы криптографии. – СПб.: "Лань", 2011. - 400 стр.
- Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.: ДиК, 2008. – 448 с.
- Основы криптографии: Учебное пособие/Под ред. Алферова П.П. – М.: Гелиос, 2008. – 480 с.

**2. Дополнительная литература:**

- Торстейнсон П., Ганеш Г.А. Криптография и безопасность в технологии .NET. – М.: "Бином. Лаборатория знаний", 2013. – 480 с.
- Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.: "ДМК Пресс", 2008. – 448 с.
- Алферов А.П., Зубов А.Ю. и др. Основы криптографии: Учебное пособие, 2-е – М.: Гелиос АРВ, 2006 – 480 с.
- Баричев С.Г. Основы современной криптографии. Учебный курс. – М.: Телеком, 2007. – 129 с.
- Молдовян А.А. и др. Криптография. – СПб.: Лань, 2007. – 224 с.
- Введение в криптографию/Под ред В.В.Ященко. – М.: МЦНМО, 2006. – 288 с.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

## **Методические указания**

по выполнению лабораторных работ  
по дисциплине «Методы и средства криптографической защиты информации»  
для студентов направления подготовки  
10.03.01 Информационная безопасность  
Направленность (профиль) «Организация и технологии защиты информации»  
(по отрасли и с сфере профессиональной деятельности)

## Содержание

### 5 семестр

Лабораторная работа 1. Модели шифров

Лабораторная работа 2. Классификация методов криптографической защиты информации

Лабораторная работа 3. Алгоритм шифрования таблицей Виженера

Лабораторная работа 4. Шифры перестановки их свойства

Лабораторная работа 5. Расшифрование на основе генераторов много тактовых кодовых фильтров

+4 ЛР (Их нет в МУ)

Лабораторная работа 6. Шифрование с помощью самосинхронизирующихся поточных шифров

Лабораторная работа 7. Расшифрование с помощью самосинхронизирующихся поточных шифров

Лабораторная работа 8. Основы построения поточного шифра S-A5

Лабораторная работа 9. Расшифрование с помощью поточного шифра S-A5

### 6 семестр

Лабораторная работа 10. Арифметика целых чисел

Лабораторная работа 11. Модульная арифметика

Лабораторная работа 12. Алгебраические структуры

Лабораторная работа 13. Расширение полей Галуа  $GF(2^m)$

Лабораторная работа 14. Математические модели расширенных полей Галуа

### Больше работ нет

Лабораторная работа 15. Алгоритм шифрования без передачи ключа

Лабораторная работа 16. Алгоритм шифрования Эль-Гамала

Лабораторная работа 17. Построение электронной цифровой подписи RSA

Лабораторная работа 18. Построение электронной цифровой подписи  
Эль-Гамала

+ 7 ЛР(Их нет в МУ)

Лабораторная работа 19. Ключевой обмен на основе алгоритма Диффи-Хеллмана

Лабораторная работа 20. Модифицированный метод ключевого обмена на основе алгоритма Диффи-Хеллмана

Лабораторная работа 21. Схема разделения секрета Шамира

Лабораторная работа 22. Протокол аутентификации Шнорра

Лабораторная работа 23. Протокол аутентификации для систем электронных платежей

Лабораторная работа 24. Отечественная криптографическая хеш-функция

Лабораторная работа 25. Скрытый канал Эль-Гамала

Указания по технике безопасности .....

Список рекомендуемой литературы .....

## Лабораторная работа №1

### МОДЕЛИ ШИФРОВ

#### Цель и содержание:

1. Углубить знания, по основам алгоритма блочного шифрования .
2. Исследовать вопросы получения ключей для алгоритма шифрования.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Упрощенный DES по свойствам и структуре он подобен DES, но имеет гораздо меньше параметров. Данный алгоритм был разработан профессором Эдвардом Шейфером (Edward Schaefer) из Университета Санта-Клара.

В алгоритме S-DES используется 10-битовый ключ, который находится у отправителя и у получателя сообщения. Из этого ключа на определенных этапах шифрования и дешифрования генерируется два 8-битовых ключа. На рисунке 1 показана схема процедуры создания подключей. Сначала выполняется перестановка битов ключа следующим образом. Если 10-битовый ключ представить в виде  $k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9 k_{10}$ , то перестановку P10 можно задать формулой:

$$P10 (k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9 k_{10}) = (k_3 k_5 k_2 k_7 k_4 k_{10} k_1 k_9 k_8 k_6)$$

Можно также представить перестановку P10 в табличной форме:

| P 10 |   |   |   |   |    |   |   |   |   |
|------|---|---|---|---|----|---|---|---|---|
| 3    | 5 | 2 | 7 | 4 | 10 | 1 | 9 | 8 | 6 |

Эту таблицу следует читать слева направо. Каждый ее элемент идентифицирует позицию бита исходных данных в генерируемой выходной последовательности. Иными словами, первым битом в выходной последовательности будет третий бит исходной последовательности, вторым - пятый и т.д. Например, в соответствии с данной таблицей ключ  $(1010000010)$  будет преобразован к виду  $(1000001100)$ . После этого отдельно для первых пяти битов и отдельно для вторых выполняется циклический сдвиг влево (LS-1), который еще называют вращением. В нашем случае в результате будет получена последовательность  $(00001 11000)$ .

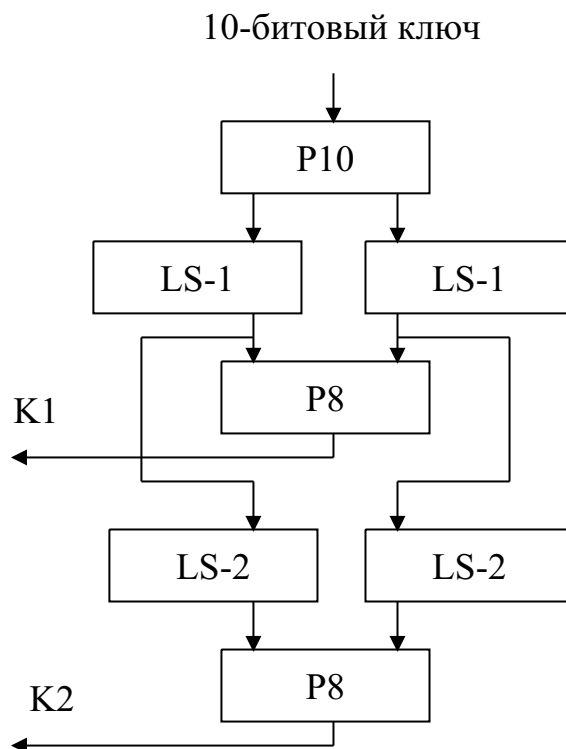


Рисунок 1 – Вычисление ключей S-DES

Затем применяется перестановка P8, в результате которой из 10-битового ключа сначала выбираются, а затем переставляются 8 битов по правилу:

| P8 |   |   |   |   |   |    |   |
|----|---|---|---|---|---|----|---|
| 6  | 3 | 7 | 4 | 8 | 5 | 10 | 9 |

В результате этой операции получается первый подключ (K1). В нашем примере он будет иметь вид *(10100100)*.

Теперь нужно вернуться к двум 5-битовым строкам, полученным в результате применения функций LS-1, и выполнить с каждой из этих строк циклический сдвиг влево на две позиции (LS-2). В нашем конкретном случае значение *(00001 11000)* будет преобразовано к виду *(00100 00011)*. Наконец, применив к полученной в результате последовательности перестановку P8, получим подключ (K2). Для нашего примера результатом будет *(01000011)*.

### Методика и порядок выполнения работы

3. Изучить теоретический материал работы.
4. Провести исследование ключевой системы алгоритма S-DES.

Каждый студент получает индивидуальное задание по выработке ключа для алгоритма упрощенного S-DES. Задание представлено в таблице 1.

Таблица 1 – Задание на разработку ключей для S-DES

| № п/п | Исходные данные | № п/п | Исходные данные |
|-------|-----------------|-------|-----------------|
| 1.    | 1100110101      | 15.   | 1111110101      |
| 2.    | 1010100101      | 16.   | 1100100101      |
| 3.    | 0001010101      | 17.   | 0111010101      |
| 4.    | 1010111101      | 18.   | 1110111101      |
| 5.    | 1011101011      | 19.   | 1001101011      |
| 6.    | 1100111101      | 20.   | 1100100001      |
| 7.    | 0011010101      | 21.   | 0100010101      |
| 8.    | 1100001111      | 22.   | 1101101111      |
| 9.    | 0100110101      | 23.   | 0111110101      |
| 10.   | 1011010101      | 24.   | 1011100101      |
| 11.   | 0100001100      | 25.   | 1111101100      |
| 12.   | 0011001010      | 26.   | 1111001010      |
| 13.   | 1110101000      | 27.   | 0100101001      |
| 14.   | 1100011111      | 28.   | 0001100110      |

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования систем по своему варианту и ответы на контрольные вопросы.

При защите студенты предоставляют программу с помощью которой провели исследования ключевой системы данного алгоритма шифрования.

### **Вопросы для защиты работы**

1. Поясните особенности шифрования с использованием S-DES. Какие виды шифров вы можете отнести к этой группе.
2. Какие виды ключей используются в S-DES.
3. Основные этапы получения ключей в алгоритме S-DES

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Лабораторная работа 2**

# **КЛАССИФИКАЦИЯ МЕТОДОВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ**

### **Цель и содержание:**

1. Углубить знания, по основам алгоритма блочного шифрования .
2. Исследовать вопросы шифрования данных .

### **Формируемые компетенции ОПК-6, ОПК-9**

### **Теоретическое обоснование**

На рисунке 1 показана общая структура упрощенного алгоритма DES, который мы в дальнейшем будем для краткости называть S-DES. Этот алгоритм получает на входе 8-битовый блок открытого текста (например, 10111101) и 10-битовый ключ, а на выходе генерируется 8-битовый блок закрытого текста.

Функция  $f_k$ , использует в качестве исходных данных не только шифруемый текст, но и 8-битовый ключ. Алгоритм можно построить так, чтобы он работал с 16-битовым ключом, состоящим из двух восьмибитовых подключей, применяемых по отдельности каждый для своего вызова криптографической функции  $f_k$ .

Можно использовать и 8-битовый ключ, для чего просто следует ввести его дважды

Наконец, можно прибегнуть к комбинированному решению, когда требуется 10-битовый ключ, из которого генерируются два 8-битовых, как показано на рисунке 1. В этом случае ключ сначала преобразуется путем перестановки (P10). После этого применяется операция сдвига, а полученные в ее результате данные поступают на вход перестановки (P8), которая генерирует первый 8-битовый ключ (K1). Те же полученные в результате операции сдвига данные поступают на вход другой операции сдвига и другой функции перестановки (P8), в результате чего генерируется второй подключ (K2).

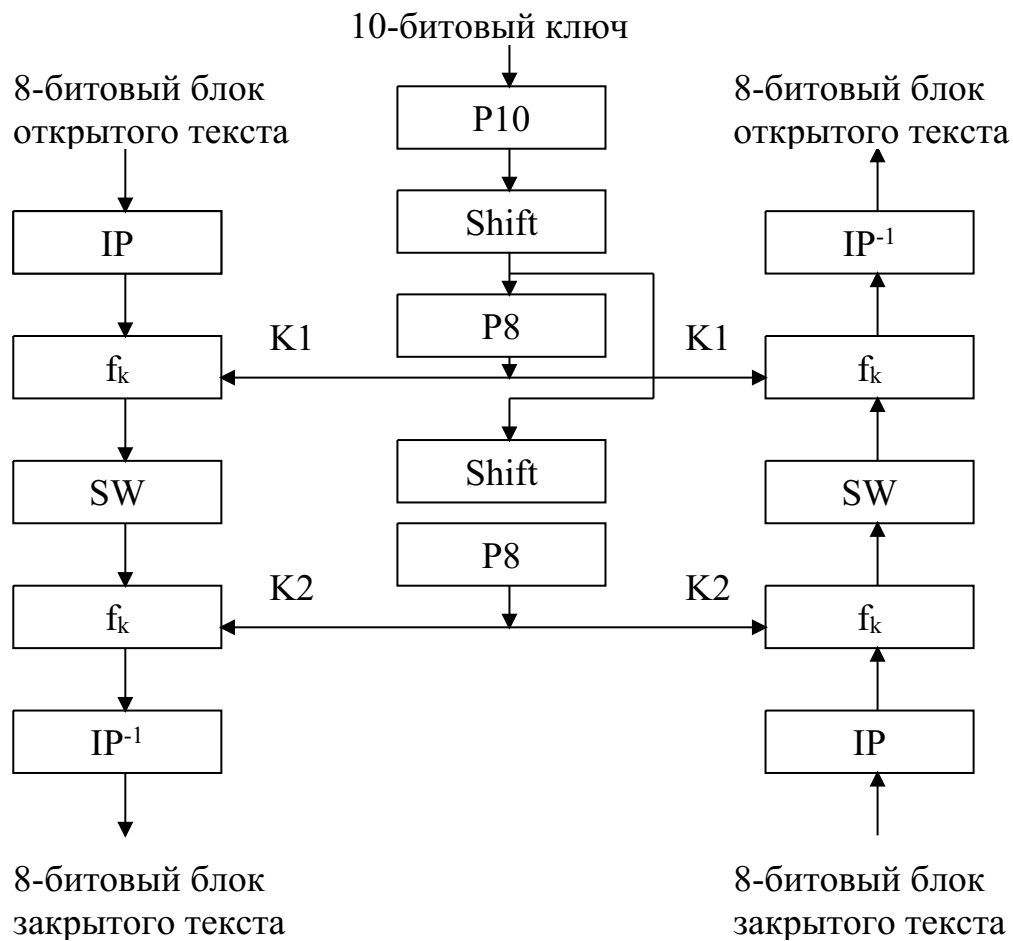


Рисунок 1 – Схема упрощенного алгоритма DES

На рисунке 2 представлена более подробная схема алгоритма шифрования S-DES. Как уже упоминалось, процесс шифрования представляет собой последовательное выполнение пяти операций, которые мы рассмотрим здесь каждую в отдельности

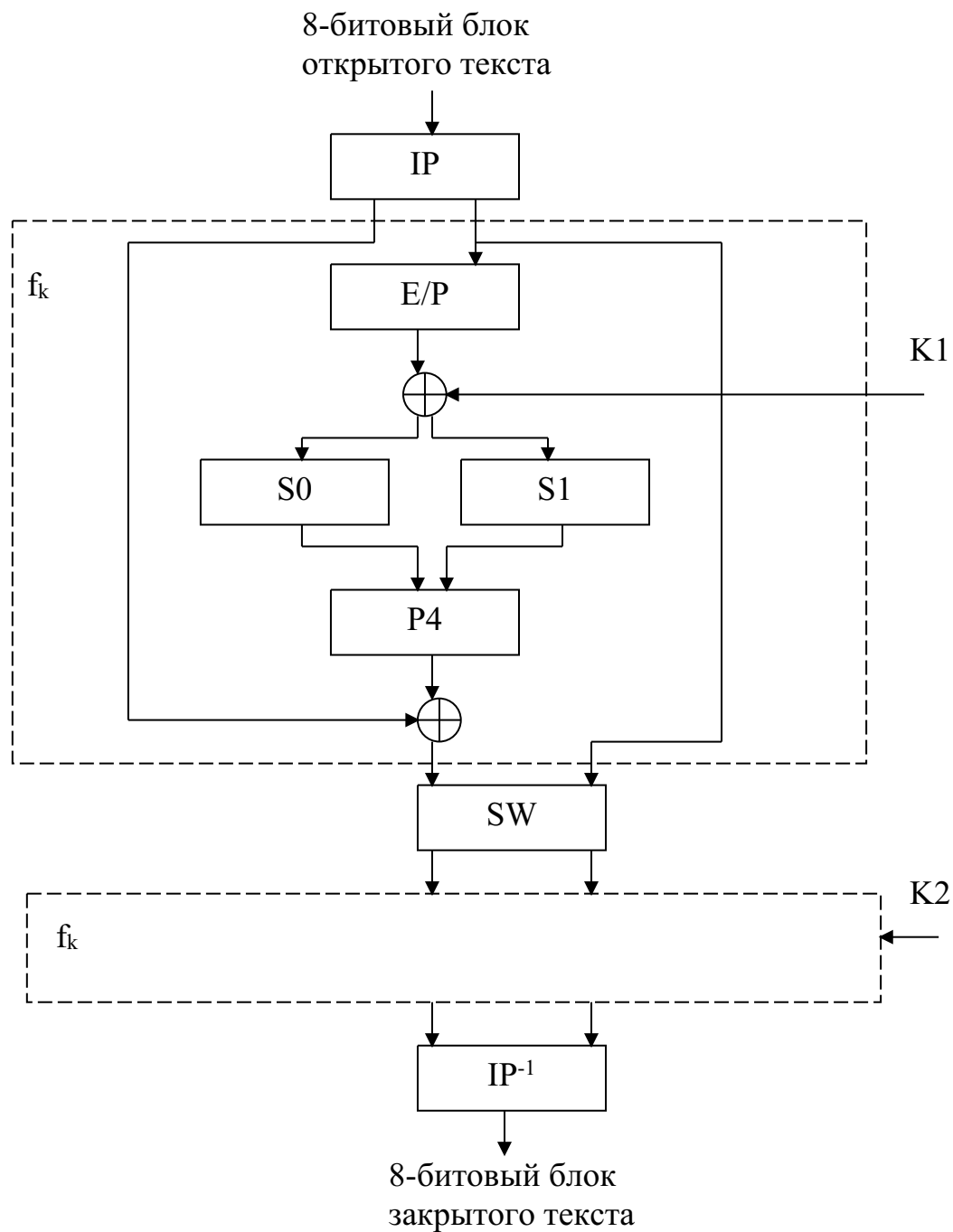


Рисунок 2 – Подробная схема шифрования S-DES

Данный алгоритм можно представить в виде композиции функций:

$$IP^{-1} \circ f_{k2} \circ SW \circ f_{k1} \circ IP,$$

или, иначе:

$$\text{шифрованный текст} = IP^{-1}(f_{k2}(SW(f_{k1}(IP(\text{открытый текст}))))))$$

где

$$K_1 = P8(\text{сдвиг}(P10(\text{ключ}))),$$

$$K_2 = P8(\text{сдвиг}(\text{сдвиг}(P10(\text{ключ}))))).$$

Процесс дешифрования, также представленный на рисунке 1, по сути, является процессом, обратным процессу шифрования:

$$\text{открытый текст} = IP^{-1}(f_{k1}(SW(f_{k2}(IP(\text{шифрованный текст}))))))$$

### Начальная и завершающая перестановки

На вход алгоритма поступает 8-битовый блок открытого текста, к которому применяется начальная перестановка, заданная функцией IP.

| IP |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 2  | 6 | 3 | 1 | 4 | 8 | 5 | 7 |

Все 8 битов открытого текста сохраняют свои значения, но меняется порядок их следования. На завершающей стадии алгоритма выполняется обратная перестановка.

| IP <sup>-1</sup> |   |   |   |   |   |   |   |
|------------------|---|---|---|---|---|---|---|
| 4                | 1 | 3 | 5 | 7 | 2 | 8 | 6 |

Как легко убедиться с помощью простой проверки, вторая из приведенных выше перестановок действительно является обратной по отношению к первой, т.е.  $IP^{-1}(IP(X)) = X$ .

### Функция $f_k$

Самым сложным компонентом S-DES является функция  $f_k$ , представляющая собой комбинацию перестановки и подстановки. Пусть  $L$  и  $R$  означают соответственно первые 4 бита и последние 4 бита 8-битовой последовательности, подаваемой на вход 4, и пусть  $F$  – некоторое отображение пространства 4-битовых строк в себя, не обязательно являющееся взаимно однозначным. Тогда

$$f_k(L, R) = (L + F(E, SK), R),$$

где  $SK$  обозначает подключ, а  $+$  – операцию XOR (побитовое исключаящее ИЛИ). Например, если в результате применения функции IP (рисунок 3) получено значение  $(1011\ 1101)$  и  $F(1101, SK) = (1110)$  для некоторого ключа  $SK$ , то  $f_k(10111101) = (01011101)$ , так как  $(1011) + (1110) = (0101)$ .

Теперь опишем отображение  $F$ . На входе этого отображения имеем 4-битовое значение  $(n_1 \ n_2 \ n_3 \ n_4)$ . Первой операцией является операция расширения/перестановки.

| $E/P$ |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|
| 4     | 1 | 2 | 3 | 2 | 3 | 4 | 1 |

Для дальнейшего рассмотрения удобнее представить результат в следующей форме:

$$\begin{array}{cc} n_4 & n_1 \\ n_2 & n_3 \end{array} \quad \begin{array}{cc} n_2 & n_3 \\ n_4 & n_1 \end{array}$$

К этому значению с помощью операции XOR добавляется 8-битовый подключ  $K_1 = (k_{11}, k_{12} \ k_{13} \ k_{14} \ k_{15} \ k_{16} \ k_{17} \ k_{18})$ :

$$\begin{array}{cc} n_4 + k_{11} & n_1 + k_{12} \\ n_2 + k_{15} & n_3 + k_{16} \end{array} \quad \begin{array}{cc} n_2 + k_{13} & n_3 + k_{14} \\ n_4 + k_{17} & n_1 + k_{18} \end{array}$$

Переименуем полученные в результате 8 битов, как показано ниже:

$$\begin{array}{cc} P_{00} & P_{01} \\ P_{10} & P_{11} \end{array} \quad \begin{array}{cc} P_{02} & P_{03} \\ P_{12} & P_{13} \end{array}$$

Первые четыре бита (первая строка приведенной выше матрицы) поступают на вход модуля  $S0$ , на выходе которого получается 2-битовая последовательность, а оставшиеся четыре бита (вторая строка матрицы) – на вход модуля  $S1$ , на выходе которого получается другая 2-битовая последовательность. Эти S-модули (матрицы кодирования) работают следующим образом. Первый и четвертый биты входной последовательности рассматриваются как 2-битовые числа, определяющие строку, а второй и третий – как числа, определяющие столбец  $S$ -матрицы. Модули  $S0$  и  $S1$  можно определить следующим образом:

$$S0 = \begin{bmatrix} 0 & 3 & 2 \\ 2 & 1 & 0 \\ 2 & 1 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

$$S1 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

Элементы, находящиеся на пересечении соответствующей строки и столбца, задают двух битовые выходные значения. Например, если  $(P_{0,0} P_{0,3}) = (00)$  и  $(p_{0,1} p_{02}) = (10)$ , то выходные 2 бита задаются значением, которое находится на пересечении строки 0 и столбца 2 матрицы  $S0$  (оно равно 3 или  $(11)$  в двоичном представлении). Точно так же  $(P_{1,0} P_{1,3})$  и  $(p_{1,1} p_{12})$  служат для определения строки и столбца матрицы  $S1$ , на пересечении которых стоит значение, задающее вторые 2 бита.

Теперь 4 бита, полученные на выходе модулей  $S0$  и  $S1$ , преобразуются с помощью перестановки следующим образом:

| P4 |   |   |   |
|----|---|---|---|
| 2  | 4 | 3 | 1 |

Результат применения перестановки P4 и является результатом функции F.

### Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Провести исследование процесса шифрования .

Каждый студент получает индивидуальное задание по осуществлению процесса шифрования. Открытый текст приведен в таблице 1. В качестве ключа для алгоритма S-DES использовать результаты предыдущего занятия.

Таблица 1 – Задание на разработку ключей для S-DES

| № п/п | Исходные данные | № п/п | Исходные данные |
|-------|-----------------|-------|-----------------|
| 1.    | 1111110101      | 15.   | 1100110101      |
| 2.    | 1100100101      | 16.   | 1010100101      |
| 3.    | 0111010101      | 17.   | 0001010101      |
| 4.    | 1110111101      | 18.   | 1010111101      |
| 5.    | 1001101011      | 19.   | 1011101011      |
| 6.    | 1100100001      | 20.   | 1100111101      |
| 7.    | 0100010101      | 21.   | 0011010101      |
| 8.    | 1101101111      | 22.   | 1100001111      |
| 9.    | 0111110101      | 23.   | 0100110101      |
| 10.   | 1011100101      | 24.   | 1011010101      |
| 11.   | 1111101100      | 25.   | 0100001100      |
| 12.   | 1111001010      | 26.   | 0011001010      |
| 13.   | 0100101001      | 27.   | 1110101000      |

| № п/п | Исходные данные | № п/п | Исходные данные |
|-------|-----------------|-------|-----------------|
| 14.   | 0001100110      | 28.   | 1100011111      |

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования систем по своему варианту и ответы на контрольные вопросы.

### **Вопросы для защиты работы**

1. Поясните особенности процесса шифрования в S-DES.
2. Сколько раундов используется в S-DES.
3. Основные операции шифрования с помощью алгоритма S-DES.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Лабораторная работа № 3

### Алгоритм шифрования таблицей Виженера

#### Цель и содержание:

1. Углубить знания, по основам построения алгоритма шифрования .
2. Исследовать вопросы расшифрования принятого сообщения.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Алгоритм дешифрования S-DES в качестве исходных данных использует 8-битовый блок шифрованного текста и тот же 10-битовый ключ, который применялся для шифрования, а в результате работы алгоритм дешифрования должен генерировать 8-битовый блок открытого текста.

Алгоритм дешифрования включает последовательное выполнение пяти операций; начальной перестановки  $IP$ , сложной функции, являющейся композицией операций перестановки и подстановки и зависящей от полученного ключа, перестановки  $SW$ , при которой две половинки последовательности данных просто меняются местами, еще раз функции  $f$  и, наконец, перестановки, обратной начальной ( $IP^{-1}$ ). Подробная схема дешифрования с использованием S-DES приведена на рисунке 1.

На вход алгоритма поступает 8-битовый блок закрытого текста, к которому применяется начальная перестановка, заданная функцией  $IP$ .

| $IP$ |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|
| 2    | 6 | 3 | 1 | 4 | 8 | 5 | 7 |

Все 8 битов закрытого текста сохраняют свои значения, но меняется порядок их следования. На завершающей стадии алгоритма выполняется обратная перестановка.

| $IP^{-1}$ |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|
| 4         | 1 | 3 | 5 | 7 | 2 | 8 | 6 |

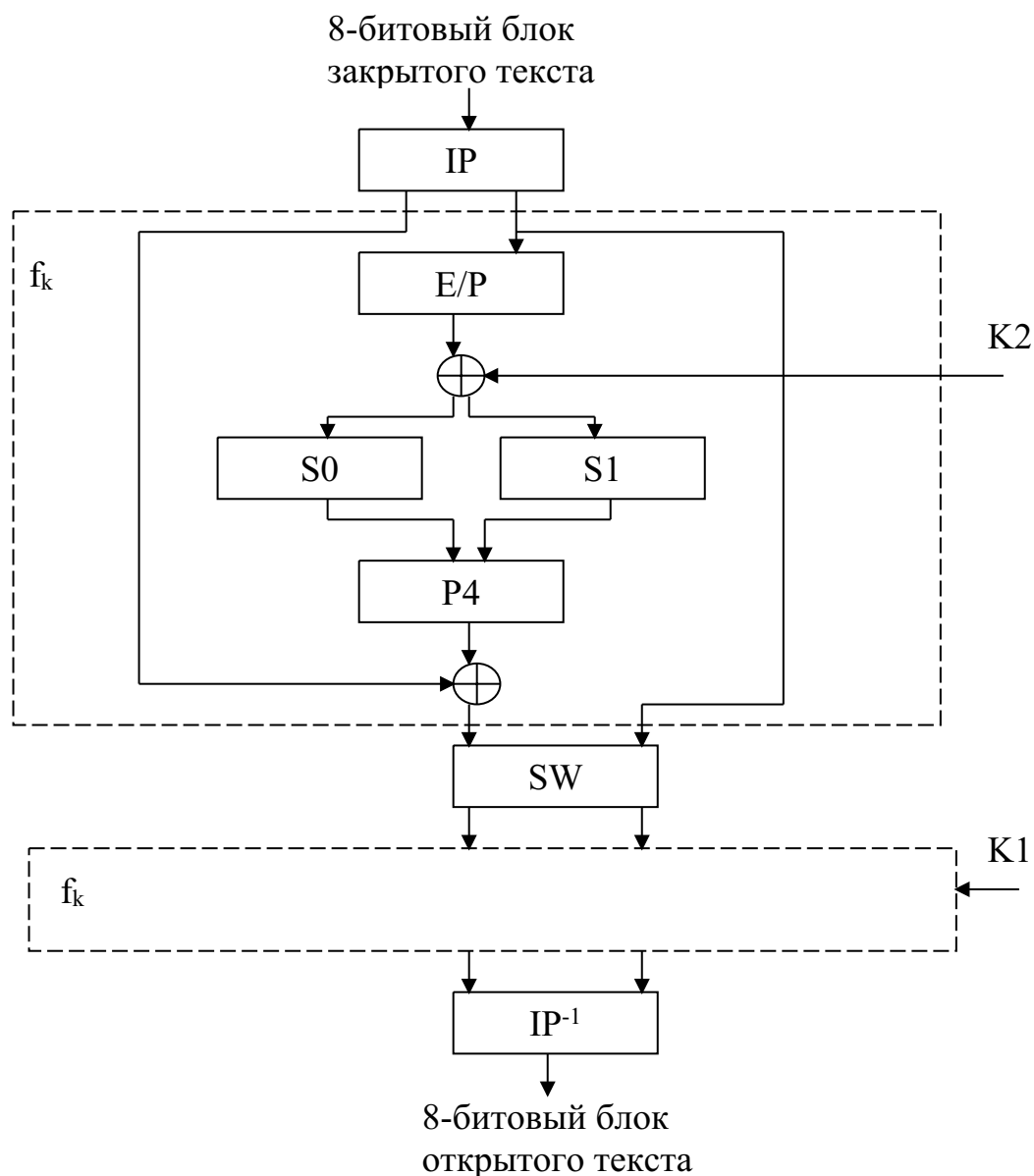


Рисунок 1 – Подробная схема расшифрования S-DES

Как легко убедиться с помощью простой проверки, вторая из приведенных выше перестановок действительно является обратной по отношению к первой, т.е.  $IP^{-1}(IP(X)) = X$ .

### Функция $f_k$

Самым сложным компонентом S-DES является функция  $f_k$ , представляющая собой комбинацию перестановки и подстановки. Пусть  $L$  и  $R$  означают соответственно первые 4 бита и последние 4 бита 8-битовой последовательности, подаваемой на вход 4, и пусть  $F$  – некоторое отображение

пространства 4-битовых строк в себя, не обязательно являющееся взаимно однозначным. Тогда

$$f_k(L, B) = (L + F(E, SK), R),$$

где  $SK$  обозначает подключ, а  $+$  – операцию XOR (побитовое исключающее ИЛИ). Например, если в результате применения функции  $IP$  (рисунок 3) получено значение  $(1011\ 1101)$  и  $F(1101, SK) = (1110)$  для некоторого ключа  $SK$ , то  $f_k(10111101) = (01011101)$ , так как  $(1011) + (1110) = (0101)$ .

Теперь опишем отображение  $F$ . На входе этого отображения имеем 4-битовое значение  $(n_1\ n_2\ n_3\ n_4)$ . Первой операцией является операция расширения/перестановки.

| $E/P$ |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|
| 4     | 1 | 2 | 3 | 2 | 3 | 4 | 1 |

Для дальнейшего рассмотрения удобнее представить результат в следующей форме:

$$\begin{array}{cc} n_4 & n_1 \\ n_2 & n_3 \end{array} \quad \begin{array}{cc} n_2 & n_3 \\ n_4 & n_1 \end{array}$$

К этому значению с помощью операции XOR добавляется 8-битовый подключ  $K_1 = (k_{11},\ k_{12}\ k_{13}\ k_{14}\ k_{15}\ k_{16}\ k_{17}\ k_{18})$ :

$$\begin{array}{cc} n_4 + k_{11} & n_1 + k_{12} \\ n_2 + k_{15} & n_3 + k_{16} \end{array} \quad \begin{array}{cc} n_2 + k_{13} & n_3 + k_{14} \\ n_4 + k_{17} & n_1 + k_{18} \end{array}$$

Переименуем полученные в результате 8 битов, как показано ниже:

$$\begin{array}{cc} P_{00} & P_{01} \\ P_{10} & P_{11} \end{array} \quad \begin{array}{cc} P_{02} & P_{03} \\ P_{12} & P_{13} \end{array}$$

Первые четыре бита (первая строка приведенной выше матрицы) поступают на вход модуля  $S0$ , на выходе которого получается 2-битовая последовательность, а оставшиеся четыре бита (вторая строка матрицы) – на вход модуля  $S1$ , на выходе которого получается другая 2-битовая последовательность. Эти S-модули (матрицы кодирования) работают

следующим образом. Первый и четвертый биты входной последовательности рассматриваются как 2-битовые числа, определяющие строку, а второй и третий – как числа, определяющие столбец  $S$ -матрицы. Модули  $S0$  и  $S1$  можно определить следующим образом:

$$S0 = \begin{bmatrix} 0 & 3 & 2 \\ 2 & 1 & 0 \\ 2 & 1 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

$$S1 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

Элементы, находящиеся на пересечении соответствующей строки и столбца, задают двух битовые выходные значения. Например, если  $(P_{0,0} P_{0,3}) = (00)$  и  $(p_{0,1} p_{0,2}) = (10)$ , то выходные 2 бита задаются значением, которое находится на пересечении строки 0 и столбца 2 матрицы  $S0$  (оно равно 3 или  $(11)$  в двоичном представлении). Точно так же  $(P_{1,0} P_{1,3})$  и  $(p_{1,1} p_{1,2})$  служат для определения строки и столбца матрицы  $S1$ , на пересечении которых стоит значение, задающее вторые 2 бита.

Теперь 4 бита, полученные на выходе модулей  $S0$  и  $S1$ , преобразуются с помощью перестановки следующим образом:

| P4 |   |   |   |
|----|---|---|---|
| 2  | 4 | 3 | 1 |

Результат применения перестановки P4 и является результатом сложной функции F.

### Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Провести исследование процесса расшифрования.

Каждый студент получает индивидуальное задание по осуществлению дешифрования. В качестве зашифрованного текста, а так же в качестве ключа для упрощенного S-DES использовать результаты предыдущих занятий

### Содержание отчета и его форма

Отчет по лабораторной работе, оформленный письменно в рабочей

тетради, должен содержать процесс исследования системы по своему варианту и ответы на контрольные вопросы.

**Вопросы для защиты работы**

1. Поясните особенности процесса дешифрования в S-DES.
2. Сколько раундов в S-DES для осуществления дешифрования.
3. Проведите сравнение алгоритма DES и S-DES по этапу расширения передаваемого сообщения.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Лабораторная работа № 4

### ШИФРЫ ПЕРЕСТАНОВКИ ИХ СВОЙСТВА

#### Цель и содержание:

1. Углубить знания, по основам поточного шифрования.
2. Исследовать вопросы получения синхронного ПСП.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Шифр Вернама можно считать исторически первым поточным шифром. Так как поточные шифры, в отличие от блочных, осуществляют поэлементное шифрование потока данных без задержки в криптосистеме, их важнейшим достоинством является высокая скорость преобразования, соизмеримая со скоростью поступления входной информации. Таким образом, обеспечивается шифрование практически в реальном масштабе времени вне зависимости от объема и разрядности потока преобразуемых данных.

Простейшие устройства синхронного и самосинхронизирующегося шифрования с использованием ГПК, реализованного на основе  $N$ -разрядного регистра сдвига с линейной обратной связью - *LFSR* (Linear Feedback Shift Register), называются *скремблерами*, а сам процесс преобразования – скремблированием.

В синхронных поточных шифрах гамма формируется независимо от входной последовательности, каждый элемент (бит, символ, байт и т. п.) которой таким образом шифруется независимо от других элементов. В синхронных поточных шифрах отсутствует эффект размножения ошибок, т. е. число искаженных элементов в расшифрованной последовательности равно числу искаженных элементов зашифрованной последовательности, пришедшей из канала связи.

Вставка или выпадение элемента зашифрованной последовательности недопустимы, так как из-за нарушения синхронизации это приведет к

неправильному расшифрованию всех последующих элементов. Синхронное поточное шифрование с использованием LFSR показано на рисунке 1.

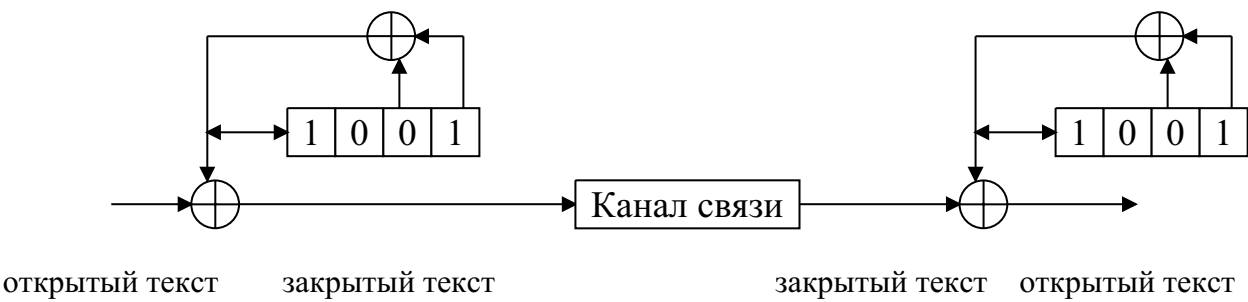


Рисунок 1 – Синхронное поточное шифрование

В таблице 1 показан пример поточного шифрования и расшифрования двоичной последовательности *11100101010110* с использованием гаммы формируемой 4-разрядным *LFSR* при начальном состоянии **1001**.

Зашифрованная последовательность имеет вид *01001010010010*.

При отсутствии искажений в канале связи после расшифрования с использованием той же гаммы получается исходная последовательность

Таблица 1 – Пример поточного шифрования и расшифрования двоичной последовательности, когда отсутствуют ошибки в принятой комбинации

| Процедура шифрования последовательности |          |               |   |   |   | Процедура расшифрования, когда отсутствуют ошибки в принятой комбинации |          |               |   |   |   |
|-----------------------------------------|----------|---------------|---|---|---|-------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                      |          |               |   |   |   | Приемная сторона                                                        |          |               |   |   |   |
| <i>c</i>                                | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                | <i>p</i> | Генератор ПСП |   |   |   |
| 0                                       | 1        | 1             | 1 | 0 | 0 | 0                                                                       | 1        | 1             | 1 | 0 | 0 |
| 1                                       | 1        | 0             | 1 | 1 | 0 | 1                                                                       | 1        | 0             | 1 | 1 | 0 |
| 0                                       | 1        | 1             | 0 | 1 | 1 | 0                                                                       | 1        | 1             | 0 | 1 | 1 |
| 0                                       | 0        | 0             | 1 | 0 | 1 | 0                                                                       | 0        | 0             | 1 | 0 | 1 |
| 1                                       | 0        | 1             | 0 | 1 | 0 | 1                                                                       | 0        | 1             | 0 | 1 | 0 |
| 0                                       | 1        | 1             | 1 | 0 | 1 | 0                                                                       | 1        | 1             | 1 | 0 | 1 |

| Процедура поточного шифрования двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда отсутствуют ошибки в принятой комбинации |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|-----------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                  |          |               |   |   |   |
| $c$                                                        | $p$      | Генератор ПСП |   |   |   | $c$                                                                               | $p$      | Генератор ПСП |   |   |   |
| <b>1</b>                                                   | <b>0</b> | 1             | 1 | 1 | 0 | <b>1</b>                                                                          | <b>0</b> | 1             | 1 | 1 | 0 |
| <b>0</b>                                                   | <b>1</b> | 1             | 1 | 1 | 1 | <b>0</b>                                                                          | <b>1</b> | 1             | 1 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 1 | 1 | 1 | <b>0</b>                                                                          | <b>0</b> | 0             | 1 | 1 | 1 |
| <b>1</b>                                                   | <b>1</b> | 0             | 0 | 1 | 1 | <b>1</b>                                                                          | <b>1</b> | 0             | 0 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 0 | 1 | <b>0</b>                                                                          | <b>0</b> | 0             | 0 | 0 | 1 |
| <b>0</b>                                                   | <b>1</b> | 1             | 0 | 0 | 0 | <b>0</b>                                                                          | <b>1</b> | 1             | 0 | 0 | 0 |
| <b>1</b>                                                   | <b>1</b> | 0             | 1 | 0 | 0 | <b>1</b>                                                                          | <b>1</b> | 0             | 1 | 0 | 0 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 1 | 0 | <b>0</b>                                                                          | <b>0</b> | 0             | 0 | 1 | 0 |

В таблице 2 рассмотрена ситуация, когда при передаче зашифрованной последовательности был потерян четвертый бит, и вместо правильной последовательности к получателю пришла последовательность **0101010010010**.

Таблица 2 – Пример поточного шифрования и расшифрования двоичной последовательности, когда при передаче был потерян четвертый бит

| Процедура поточного шифрования двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда при передаче был потерян четвертый бит |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|---------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                |          |               |   |   |   |
| $c$                                                        | $p$      | Генератор ПСП |   |   |   | $c$                                                                             | $p$      | Генератор ПСП |   |   |   |
| <b>0</b>                                                   | <b>1</b> | 1             | 1 | 0 | 0 | <b>0</b>                                                                        | <b>1</b> | 1             | 1 | 0 | 0 |
| <b>1</b>                                                   | <b>1</b> | 0             | 1 | 1 | 0 | <b>1</b>                                                                        | <b>1</b> | 0             | 1 | 1 | 0 |
| <b>0</b>                                                   | <b>1</b> | 1             | 0 | 1 | 1 | <b>0</b>                                                                        | <b>1</b> | 1             | 0 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 1 | 0 | 1 | <b>1</b>                                                                        | <b>1</b> | 0             | 1 | 0 | 1 |
| <b>1</b>                                                   | <b>0</b> | 1             | 0 | 1 | 0 | <b>0</b>                                                                        | <b>1</b> | 1             | 0 | 1 | 0 |
| <b>0</b>                                                   | <b>1</b> | 1             | 1 | 0 | 1 | <b>1</b>                                                                        | <b>0</b> | 1             | 1 | 0 | 1 |
| <b>1</b>                                                   | <b>0</b> | 1             | 1 | 1 | 0 | <b>0</b>                                                                        | <b>1</b> | 1             | 1 | 1 | 0 |
| <b>0</b>                                                   | <b>1</b> | 1             | 1 | 1 | 1 | <b>0</b>                                                                        | <b>1</b> | 1             | 1 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 1 | 1 | 1 | <b>1</b>                                                                        | <b>1</b> | 0             | 1 | 1 | 1 |
| <b>1</b>                                                   | <b>1</b> | 0             | 0 | 1 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 0 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 0 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 0 | 0 | 1 |
| <b>0</b>                                                   | <b>1</b> | 1             | 0 | 0 | 0 | <b>1</b>                                                                        | <b>0</b> | 1             | 0 | 0 | 0 |
| <b>1</b>                                                   | <b>1</b> | 0             | 1 | 0 | 0 | <b>0</b>                                                                        | <b>0</b> | 0             | 1 | 0 | 0 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 1 | 0 |                                                                                 |          |               |   |   |   |

Видно, что после расшифрования всех битов, следующих после выпавшего, происходят искажения информации. В результате вместо битовой строки **0101010110** будет получена строка **1101110000**.

В таблице 3 рассмотрена ситуация, когда при передаче зашифрованной последовательности произошло искажение пятого (1 - 0) и восьмого (0 - 1) битов и вместо правильной последовательности к получателю пришла последовательность **01000011010010**. Видно, что после расшифрования вместо правильной строки будет получена строка **11101100010110** с искаженным пятым (0—1) и восьмым (1—0) битами.

Таблица 3 – Пример поточного шифрования и расшифрования двоичной последовательности, когда при передаче произошло искажение битов

| Процедура поточного шифрования двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда при передаче произошло искажение битов |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|---------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                        | <i>p</i> | Генератор ПСП |   |   |   |
| <b>0</b>                                                   | <b>1</b> | 1             | 1 | 0 | 0 | <b>0</b>                                                                        | <b>1</b> | 1             | 1 | 0 | 0 |
| <b>1</b>                                                   | <b>1</b> | 0             | 1 | 1 | 0 | <b>1</b>                                                                        | <b>1</b> | 0             | 1 | 1 | 0 |
| <b>0</b>                                                   | <b>1</b> | 1             | 0 | 1 | 1 | <b>0</b>                                                                        | <b>1</b> | 1             | 0 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 1 | 0 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 1 | 0 | 1 |
| <b>0</b>                                                   | <b>0</b> | 1             | 0 | 1 | 0 | <b>0</b>                                                                        | <b>1</b> | 1             | 0 | 1 | 0 |
| <b>0</b>                                                   | <b>1</b> | 1             | 1 | 0 | 1 | <b>0</b>                                                                        | <b>1</b> | 1             | 1 | 0 | 1 |
| <b>1</b>                                                   | <b>0</b> | 1             | 1 | 1 | 0 | <b>1</b>                                                                        | <b>0</b> | 1             | 1 | 1 | 0 |
| <b>1</b>                                                   | <b>1</b> | 1             | 1 | 1 | 1 | <b>1</b>                                                                        | <b>0</b> | 1             | 1 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 1 | 1 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 1 | 1 | 1 |
| <b>1</b>                                                   | <b>1</b> | 0             | 0 | 1 | 1 | <b>1</b>                                                                        | <b>1</b> | 0             | 0 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 0 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 0 | 0 | 1 |
| <b>0</b>                                                   | <b>1</b> | 1             | 0 | 0 | 0 | <b>0</b>                                                                        | <b>1</b> | 1             | 0 | 0 | 0 |
| <b>1</b>                                                   | <b>1</b> | 0             | 1 | 0 | 0 | <b>1</b>                                                                        | <b>1</b> | 0             | 1 | 0 | 0 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 1 | 0 | <b>0</b>                                                                        | <b>0</b> | 0             | 0 | 1 | 0 |

### Методика и порядок выполнения работы

5. Изучить теоретический материал работы.
6. Провести исследование системы поточного шифрования.

Используя данные представленные в таблице 7 произвести разработку генератора ПСП и осуществить процедуры зашифрования и расшифрования.

Таблица 7 – Исходные данные

| M | A0 | A1 | A2 | A3 | A4 | A5 | A6 |
|---|----|----|----|----|----|----|----|
| 4 | 1  | 0  | 0  | 1  | 1  |    |    |
| 4 | 1  | 1  | 0  | 0  | 1  |    |    |
| 5 | 1  | 0  | 0  | 1  | 0  | 1  |    |
| 5 | 1  | 0  | 1  | 0  | 0  | 1  |    |
| 5 | 1  | 1  | 1  | 1  | 0  | 1  |    |
| 5 | 1  | 1  | 0  | 1  | 1  | 1  |    |
| 5 | 1  | 1  | 1  | 0  | 1  | 1  |    |
| 6 | 1  | 0  | 0  | 0  | 0  | 1  | 1  |

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования системы по своему варианту и ответы на контрольные вопросы.

### **Вопросы для защиты работы**

1. Определение синхронного поточного шифрования.
2. Свойства синхронного поточного шифрования.
3. Основные способы построения М-последовательностей.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Лабораторная работа №5

### РАСШИФРОВАНИЕ НА ОСНОВЕ ГЕНЕРАТОРОВ МНОГО ТАКТОВЫХ КОДОВЫХ ФИЛЬТРОВ

#### Цель и содержание:

1. Углубить знания, по основам поточного шифрования.
2. Исследовать вопросы получения самосинхронизирующейся ПСП.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

В самосинхронизирующихся поточных шифрах элементы входной последовательности зашифровываются с учетом  $N$  предшествующих элементов (рисунок 1), которые принимают участие в формировании ключевой последовательности. В самосинхронизирующихся шифрах имеет место эффект размножения ошибок, в то же время в отличие от синхронных, восстановление синхронизации происходит автоматически через  $N$  элементов зашифрованной последовательности.

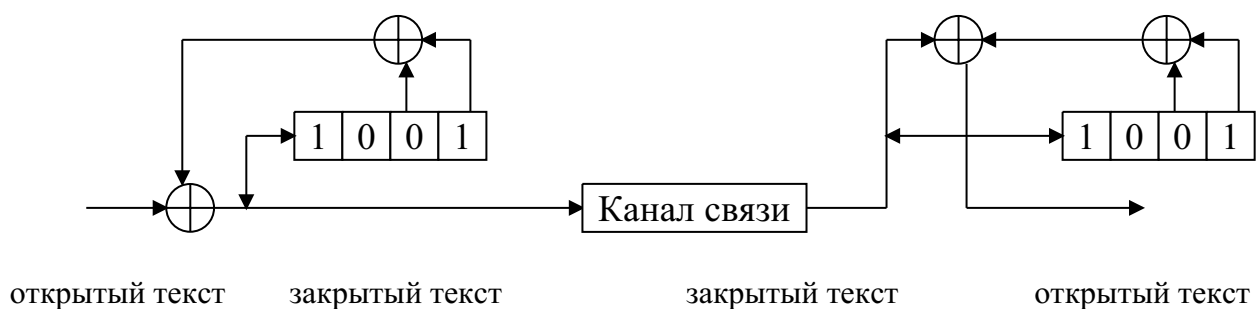


Рисунок 1 – Схема устройства зашифрования с ПСП

Таблица 1 – Пример поточного шифрования и расшифрования двоичной последовательности, когда отсутствуют ошибки в принятой комбинации

|                      |                    |  |                          |                 |
|----------------------|--------------------|--|--------------------------|-----------------|
| Процедура шифрования | поточного двоичной |  | Процедура расшифрования, | поточного когда |
|----------------------|--------------------|--|--------------------------|-----------------|

| последовательности |     |               |   |   |   | отсутствуют ошибки в принятой комбинации |     |               |   |   |   |
|--------------------|-----|---------------|---|---|---|------------------------------------------|-----|---------------|---|---|---|
| Передающая сторона |     |               |   |   |   | Приемная сторона                         |     |               |   |   |   |
| $c$                | $p$ | Генератор ПСП |   |   |   | $c$                                      | $p$ | Генератор ПСП |   |   |   |
| 0                  | 1   | 0             | 1 | 0 | 0 | 0                                        | 1   | 0             | 1 | 0 | 0 |
| 1                  | 1   | 1             | 0 | 1 | 0 | 1                                        | 1   | 1             | 0 | 1 | 0 |
| 0                  | 1   | 0             | 1 | 0 | 1 | 0                                        | 1   | 0             | 1 | 0 | 1 |
| 1                  | 0   | 1             | 0 | 1 | 0 | 1                                        | 0   | 1             | 0 | 1 | 0 |
| 1                  | 0   | 1             | 1 | 0 | 1 | 1                                        | 0   | 1             | 1 | 0 | 1 |
| 0                  | 1   | 0             | 1 | 1 | 0 | 0                                        | 1   | 0             | 1 | 1 | 0 |
| 0                  | 1   | 0             | 0 | 1 | 1 | 0                                        | 1   | 0             | 0 | 1 | 1 |
| 1                  | 1   | 1             | 0 | 0 | 1 | 1                                        | 1   | 1             | 0 | 0 | 1 |
| 1                  | 0   | 1             | 1 | 0 | 0 | 1                                        | 0   | 1             | 1 | 0 | 0 |
| 1                  | 1   | 1             | 1 | 1 | 0 | 1                                        | 1   | 1             | 1 | 1 | 0 |
| 1                  | 0   | 1             | 1 | 1 | 1 | 1                                        | 0   | 1             | 1 | 1 | 1 |
| 1                  | 1   | 1             | 1 | 1 | 1 | 1                                        | 1   | 1             | 1 | 1 | 1 |
| 0                  | 0   | 0             | 1 | 1 | 1 | 0                                        | 0   | 0             | 1 | 1 | 1 |
| 0                  | 0   | 0             | 0 | 1 | 1 | 0                                        | 0   | 0             | 0 | 1 | 1 |

В таблице 1 показан пример шифрования и расшифрования двоичной последовательности **11100111010100** с использованием 4-разрядного *LFSR* при начальном состоянии, равном 1001. Зашифрованная последовательность имеет вид **01011001111100**. При отсутствии искажений в канале после расшифрования получается исходная последовательность. В таблице 2 рассмотрена ситуация, когда при передаче зашифрованной последовательности был потерян третий, равный нулю бит и вместо правильной последовательности к получателю пришла последовательность **01111001111100**.

Видно, что после расшифрования может произойти искажение не более 4 бит (в общем случае не более  $N$ ), следующих после выпавшего символа. В рассмотренном примере вместо 4-битовой строки 0011 будет получена строка 0010. Все остальные биты будут приняты без искажений.

Таблица 2 – Пример поточного шифрования и расшифрования двоичной последовательности, когда при передаче был потерян третий бит

| Процедура поточного шифрования двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда при передаче был потерян третий бит |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                             |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                     | <i>p</i> | Генератор ПСП |   |   |   |
| 0                                                          | 1        | 0             | 1 | 0 | 0 | 0                                                                            | 1        | 0             | 1 | 0 | 0 |
| 1                                                          | 1        | 1             | 0 | 1 | 0 | 1                                                                            | 1        | 1             | 0 | 1 | 0 |
| 0                                                          | 1        | 0             | 1 | 0 | 1 | 1                                                                            | 0        | 0             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 0 | 1 | 0 | 1                                                                            | 0        | 1             | 0 | 1 | 0 |
| 1                                                          | 0        | 1             | 1 | 0 | 1 | 0                                                                            | 1        | 1             | 1 | 0 | 1 |
| 0                                                          | 1        | 0             | 1 | 1 | 0 | 0                                                                            | 0        | 0             | 1 | 1 | 0 |
| 0                                                          | 1        | 0             | 0 | 1 | 1 | 1                                                                            | 1        | 0             | 0 | 1 | 1 |
| 1                                                          | 1        | 1             | 0 | 0 | 1 | 1                                                                            | 0        | 1             | 0 | 0 | 1 |
| 1                                                          | 0        | 1             | 1 | 0 | 0 | 1                                                                            | 1        | 1             | 1 | 0 | 0 |
| 1                                                          | 1        | 1             | 1 | 1 | 0 | 1                                                                            | 0        | 1             | 1 | 1 | 0 |
| 1                                                          | 0        | 1             | 1 | 1 | 1 | 1                                                                            | 1        | 1             | 1 | 1 | 1 |
| 1                                                          | 1        | 1             | 1 | 1 | 1 | 0                                                                            | 0        | 1             | 1 | 1 | 1 |
| 0                                                          | 0        | 0             | 1 | 1 | 1 | 0                                                                            | 0        | 0             | 1 | 1 | 1 |
| 0                                                          | 0        | 0             | 0 | 1 | 1 |                                                                              |          |               |   |   |   |

В таблице 3 рассмотрена ситуация, когда при передаче зашифрованной последовательности произошло искажение первого (0 - 1) бита и вместо правильной последовательности пришла последовательность 11011001111100. Видно, что после расшифрования помимо неправильно принятого бита, могут исказиться еще не более 4 последующих. В примере будет неправильно принят первый бит и вместо правильной 4-битовой строки 1100 будет получено - 1111.

Таблица 3 – Пример поточного шифрования и расшифрования, двоичной последовательности, когда при передаче произошло искажение битов

| Процедура поточного шифрования двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда при передаче произошло искажение битов |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|---------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                        | <i>p</i> | Генератор ПСП |   |   |   |
| 1                                                          | 1        | 0             | 1 | 0 | 0 | 1                                                                               | 0        | 0             | 1 | 0 | 0 |
| 1                                                          | 1        | 1             | 0 | 1 | 0 | 1                                                                               | 1        | 1             | 0 | 1 | 0 |
| 0                                                          | 1        | 0             | 1 | 0 | 1 | 0                                                                               | 1        | 0             | 1 | 0 | 1 |
| 1                                                          | 0        | 1             | 0 | 1 | 0 | 1                                                                               | 1        | 1             | 0 | 1 | 0 |
| 1                                                          | 0        | 1             | 1 | 0 | 1 | 1                                                                               | 1        | 1             | 1 | 0 | 1 |
| 0                                                          | 1        | 0             | 1 | 1 | 0 | 0                                                                               | 1        | 0             | 1 | 1 | 0 |
| 0                                                          | 1        | 0             | 0 | 1 | 1 | 0                                                                               | 1        | 0             | 0 | 1 | 1 |
| 1                                                          | 1        | 1             | 0 | 0 | 1 | 1                                                                               | 1        | 1             | 0 | 0 | 1 |

| Процедура шифрования поточного двоичной последовательности |          |               |   |   |   | Процедура поточного расшифрования, когда при передаче произошло искажение битов |          |               |   |   |   |
|------------------------------------------------------------|----------|---------------|---|---|---|---------------------------------------------------------------------------------|----------|---------------|---|---|---|
| Передающая сторона                                         |          |               |   |   |   | Приемная сторона                                                                |          |               |   |   |   |
| <i>c</i>                                                   | <i>p</i> | Генератор ПСП |   |   |   | <i>c</i>                                                                        | <i>p</i> | Генератор ПСП |   |   |   |
| <b>1</b>                                                   | <b>0</b> | 1             | 1 | 0 | 0 | <b>1</b>                                                                        | <b>0</b> | 1             | 1 | 0 | 0 |
| <b>1</b>                                                   | <b>1</b> | 1             | 1 | 1 | 0 | <b>1</b>                                                                        | <b>1</b> | 1             | 1 | 1 | 0 |
| <b>1</b>                                                   | <b>0</b> | 1             | 1 | 1 | 1 | <b>1</b>                                                                        | <b>0</b> | 1             | 1 | 1 | 1 |
| <b>1</b>                                                   | <b>1</b> | 1             | 1 | 1 | 1 | <b>1</b>                                                                        | <b>1</b> | 1             | 1 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 1 | 1 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 1 | 1 | 1 |
| <b>0</b>                                                   | <b>0</b> | 0             | 0 | 1 | 1 | <b>0</b>                                                                        | <b>0</b> | 0             | 0 | 1 | 1 |

### Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Провести исследование системы с самосинхронизирующейся ПСП.

Используя данные представленные в таблице 7 произвести разработку генератора ПСП и осуществить процедуры зашифрования и расшифрования (исходное заполнение генератора произвольное).

Таблица 7 – Исходные данные для шифрования и расшифрования сообщений в самосинхронизирующихся системах

| М | A0 | A1 | A2 | A3 | A4 | A5 | A6 |
|---|----|----|----|----|----|----|----|
| 4 | 1  | 0  | 0  | 1  | 1  |    |    |
| 4 | 1  | 1  | 0  | 0  | 1  |    |    |
| 5 | 1  | 0  | 0  | 1  | 0  | 1  |    |
| 5 | 1  | 0  | 1  | 0  | 0  | 1  |    |
| 5 | 1  | 1  | 1  | 1  | 0  | 1  |    |
| 5 | 1  | 1  | 0  | 1  | 1  | 1  |    |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 5 | 1 | 1 | 1 | 0 | 1 | 1 |   |
| 6 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования систем по своему варианту и ответы на контрольные вопросы.

### **Вопросы для защиты работы**

1. Определение самосинхронизирующихся поточных шифров.
2. Свойства самосинхронизирующихся поточных шифров.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## 6 семестр

### Лабораторная работа № 6

#### Шифрование с помощью самосинхронизирующихся поточных шифров.

##### Цель и содержание:

1. Углубить знания, по основам использования алгебраических структур для шифрования с двумя секретными ключами.
2. Исследовать основные характеристики алгоритма шифрования.

##### Формируемые компетенции ОПК-6, ОПК-9

##### Теоретическое обоснование

##### Вопрос 1. Элементы расширений полей Галуа

Переход от двоичной записи к полиномиальной форме

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | = |

$$1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x + 1 =$$
$$= x^7 + x^4 + x^3 + 1$$

Получили  $10011001 = x^7 + x^4 + x^3 + 1$ .

Такие многочлены используются в теории кодирования. Поэтому необходимо напомнить о правилах сложения и умножения их коэффициентов, принадлежащих полю  $GF(2^n)$ :

##### Операции над многочленами в поле $GF(2)$ .

##### Сложение

Сложение двух полиномов  $a(x)$  и  $c(x)$  в поле  $GF(2)$  дает третий полином  $u(x)$ , у которого коэффициенты при соответствующих степенях «х» суммируются по модулю два.

$$x^4 + x^3 + x^2 + 1$$

$$x^7 + x^5 + x^3 + x + 1$$

---

**Вычитание – суммирование по модулю два.**

### **Умножение**

Умножение элементов поля  $GF(2^n)$  выполняется по правилу умножения представляющих их многочленов с последующим суммированием по модулю два соответствующих коэффициентов промежуточных результатов

$$\begin{array}{r} x^7 + x^5 + x^3 + x + 1 \\ x^3 + x^2 + 1 \\ \hline x^{10} + x^8 + x^6 + x^4 + x^3 \\ x^9 + x^7 + x^5 + x^3 + x^2 \\ x^7 + x^5 + x^3 + x + 1 \\ \hline \end{array}$$

### **Деление**

$$\begin{array}{r|l} x^7 + x^5 + x^3 + x + 1 & x^3 + x^2 + 1 \\ x^7 + x^6 + x^4 & x^4 + x^3 + x + 1 \\ \hline x^6 + x^5 + x^4 + x^3 + x + 1 & \\ x^6 + x^5 + x^3 & \\ \hline x^4 + x + 1 & \\ x^4 + x^3 + x & \\ \hline x^3 + 1 & \\ x^3 + x^2 + 1 & \\ \hline x^2 & \end{array}$$

В полях Галуа  $GF(2^n)$  часто выполняются операции умножения по модулю порождающего полинома  $P(x)$ , степень которого равна  $n$

$$\deg P(x) = n$$

Пусть  $p(x) = x^4 + x + 1$ . Найдем произведение двух полиномов по модулю  $p(x)$ .

$$\begin{array}{r} x^3 + x + 1 \\ x^3 + 1 \\ \hline x^6 + x^4 + x^3 \\ x^3 + x + 1 \\ \hline x^6 + x^4 + x + 1 \end{array}$$

Полученное произведение разделим на  $p(x) = x^4 + x + 1$  до получения остатка  $r(x)$  (полином должен иметь степень меньше  $n$ , т.е.  $\deg r(x) < n$ )

$$\begin{array}{r|l} x^6 + x^4 + x + 1 & x^4 + x + 1 \\ x^6 + x^3 + x^2 & \hline \hline x^4 + x^3 + x^2 + x + 1 & \\ x^4 + x + 1 & \\ \hline x^3 + x^2 & \end{array}$$

Тогда имеем

$$(x^3 + x + 1)(x^3 + 1) \bmod (x^4 + x + 1) = x^3 + x^2$$

Довольно часто при построении помехоустойчивых кодов используются многочлены над полем  $GF(2)$ , коэффициенты которых складывались и умножались по правилам сложения и умножения элементов  $GF(2)$ . В то же время все приведенные выше операции справедливы для многочленов над полем  $GF(2^n)$ , которое является расширением поля  $GF(2)$  (и имеет характеристику 2).

Поле  $GF(2^n)$  является расширением поля  $GF(2)$  и содержит  $2^n$  элементов.  $2^n - 1$  - ненулевые элементы, которые образуют циклическую мультипликативную группу.

Чтобы получить элементы расширенного поля  $GF(2^n)$  используется порождающий полином степени  $n$ . Это неприводимые полиномы.

$$GF(2^3) \quad P(x)=x^3+x+1; \quad GF(2^4) \quad P(x)=x^4+x+1; \quad GF(2^5) \quad P(x)=x^5+x^2+1.$$

Проверим, является ли полином  $P(x)=x^3+x+1$  неприводимым в поле  $GF(2)$ , в котором заданы два элемента  $\{0, 1\}$ . Подставим их в полином.

$$P(x=0) = 0^3 + 0 + 1 = 1 \quad 0 - \text{не корень } p(x).$$

$$P(x=1) = 1^3 + 1 + 1 = 1 \quad 1 - \text{не корень } p(x).$$

Элемент  $\beta$  называется **примитивным (порождающим) элементом** ненулевой мультипликативной группы расширенного поля  $GF(2^n)$ . Для этого необходимо возводить в степень элемент  $\beta$  по модулю порождающего полинома  $p(x)$ .

Рассмотрим пример на поле  $GF(7)$ , которое содержит ненулевые элементы  $\{1, 2, 3, 4, 5, 6\}$ . В данной группе есть, как минимум, один порождающий элемент.

Пусть  $\beta = 2$ .

$$\beta^0 = 2^0 \bmod 7 = 1$$

$$\beta^1 = 2^1 \bmod 7 = 2.$$

$$\beta^2 = 2^2 \bmod 7 = 4$$

$$\beta^3 = 2^3 \bmod 7 = 8 \bmod 7 = 1.$$

Пусть  $\beta = 3$ .

$$\beta^0 = 3^0 \bmod 7 = 1$$

$$\beta^1 = 3^1 \bmod 7 = 3.$$

$$\beta^2 = 3^2 \bmod 7 = 9 \bmod 7 = 2$$

$$\beta^3 = 3^3 \bmod 7 = (3 \cdot 3^2) \bmod 7 = (3 \cdot 2) \bmod 7 = 6.$$

$$\beta^4 = 3^4 \bmod 7 = (3 \cdot 3^3) \bmod 7 = (3 \cdot 6) \bmod 7 = 18 \bmod 7 = 4.$$

$$\beta^5 = 3^5 \bmod 7 = (3 \cdot 3^4) \bmod 7 = (3 \cdot 4) \bmod 7 = 12 \bmod 7 = 5.$$

$$\beta^6 = 3^6 \bmod 7 = (3 \cdot 3^5) \bmod 7 = (3 \cdot 5) \bmod 7 = 15 \bmod 7 = 1 = \beta^0.$$

Каждый элемент поля  $GF(2^n)$  можно представить в виде многочлена  $GF(2)$ , степень которого меньше  $n$ .

Рассмотрим поле  $GF(2^3)$ , определяемое многочленом  $P(x) = x^3 + x + 1$ . Вычислим  $2^3 - 1 = 7$  ненулевых элементов этого поля. В качестве порождающего элемента выбираем  $\beta = x$ . Чтобы получить все остальные элементы будем порождающий элемент возводить в степень по модулю многочлена  $P(x) = x^3 + x + 1$ .

$$\beta^0 \bmod P(x) = \tilde{\alpha}^0 = 001$$

$$\beta^1 \bmod P(x) = \tilde{\alpha}^1 = \tilde{\alpha} = 010$$

$$\beta^2 \bmod P(x) = \tilde{\alpha}^2 \bmod x^3 + x + 1 = \tilde{\alpha}^2 = 100$$

$$\beta^3 \bmod P(x) = \tilde{\alpha}^3 \bmod x^3 + x + 1 = \tilde{\alpha} + 1 = 011$$

Проверим методом деления

Таким образом, последовательно вычисляя  $\beta^2 = \beta * \beta$ ;  $\beta^3 = \beta * \beta * \beta$  и так далее, как произведения по модулю многочлена  $P(x)$ , определяем все элементы поля. Ясно, что поле  $GF(2^n)$  полностью определяется многочленом  $P(x)$ .

Поскольку  $\beta$  - корень  $P(x)$  можно записать  $\beta^3 + \beta + 1 = 0$ ;  $\beta^3 = \beta + 1$ . Тогда восемь элементов  $GF(2^3)$  можно записать в виде

$$\beta^0 = 1 = 001; \beta^1 = x = 010; \beta^2 = x^2 = 100;$$

$$\beta^3 = x + 1 = 011; \beta^4 = x^2 + x = 110; \beta^5 = x^2 + x + 1 = 111;$$

$$\beta^6 = x^2 + 1 = 101.$$

Вычислим значения

минимальных полиномов для данного поля Галуа. Воспользуемся основными свойствами расширенных полей Галуа. Между многочленами и элементами конечных полей имеется дополнительная, еще не рассмотренная ранее связь. Эта связь позволяет глубже изучить свойства многочленов, порождающих помехоустойчивые коды. Элементы конечного поля играют для многочленов ту же роль, которую комплексные числа играют для многочленов над вещественным полем. Аналогично тому, как каждый многочлен с вещественными коэффициентами можно разложить на множители, допустив введение комплексных чисел, каждый многочлен с коэффициентами из конечного поля можно разложить на множители с помощью элементов из некоторого расширения этого поля. Связь между сомножителями и корнями

многочленов с коэффициентами из конечного поля полностью аналогична связи между корнями и сомножителями многочленов с вещественными коэффициентами. Например, заданный многочлен может быть представлен в виде

$$m(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1} = (x - \beta^1)(x - \beta^2)\dots(x - \beta^j), \quad (1)$$

где  $\beta^j$  - корни многочлена  $a(x)$ .

Многочлены и элементы поля  $GF(2^n)$  обладают рядом свойств, которые оказываются полезными при описании кодов и при построении декодеров. Перечислим эти свойства без доказательства. Каждое из свойств проиллюстрируем примерами, показывающими, как их использовать.

**Свойство 1.** Если  $f(x)$  - произвольный многочлен, коэффициенты которого лежат в  $GF(n)$ , то

$$f(x^n) = f^n(x)$$

**Свойство 2.** Сомножители  $x^m - 1$ .

Порядком  $m$  элемента  $\beta$  конечного поля называется наименьшее значение  $m$ , для которого  $\beta^m = 1$ . По определению  $\beta$  является корнем многочлена  $x^m - 1$ . Если  $\beta$  является также корнем некоторого неприводимого многочлена  $f(x)$ , то  $f(x)$  должен быть делителем  $x^m - 1$ .

**Свойство 3.** Поле  $GF(2^n)$  и корни  $x^{2^n-1} - 1$ .

Корни многочлена  $x^{2^n-1} - 1$  совпадают с ненулевыми элементами  $GF(2^n)$ . Так, многочлен  $f(x) = x^7 - 1$  раскладывается на множители следующим образом:

$$x^7 - 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$$

Очевидно, что корнями  $x^3 + x + 1$  являются  $\beta$ ,  $\beta^2$  и  $\beta^4$ ;

корнями  $x^3 + x^2 + 1$  -  $\beta^3$ ,  $\beta^5$  и  $\beta^6$ ;

корнем  $x + 1$  является 1.

**Свойство 4.**

Минимальным многочленом над полем  $GF(2^n)$  для элемента  $b$  поля  $GF(2^m)$  называется **нормированный многочлен**  $m(x)$  наименьшей степени с коэффициентами из  $GF(2^n)$ , не приводимый над этим полем, такой, что  $m(b) = 0$ .

Одно из свойств многочлена  $m(x)$  над  $GF(2^n)$  состоит в том, что если  $\beta$  является его корнем, то элементы  $\beta^{2^n}$  тоже являются его корнями, то есть корнями являются  $\beta^2, \beta^4, \beta^8, \dots$ . Так как степень многочлена  $m(x)$  конечна, то конечно и число его корней. Если  $\deg m(x) = d$ , то можно показать, что  $\beta, \beta^2, \dots, \beta^{2^d-1}$  - это все корни  $m(x)$  (после этого элементы начинают повторяться). Опираясь на сформулированное свойство корней минимального многочлена, можно эффективно находить минимальный многочлен элемента в конечном поле.

### Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Провести исследование свойств расширения поля Галуа  $GF(2^5)$ .

Используя порождающий полином, представленный в таблице 1 произвести вычисление всех ненулевых элементов поля Галуа  $GF(2^5)$ . На основании полученных элементов поля определить минимальные многочлены.

Таблица 1 – Порождающие полиномы

| Варианты          | A0 | A1 | A2 | A3 | A4 | A5 |
|-------------------|----|----|----|----|----|----|
| 1, 7, 13, 19, 25  | 1  | 0  | 0  | 1  | 0  | 1  |
| 2, 8, 14, 20, 26  | 1  | 0  | 1  | 0  | 0  | 1  |
| 3, 9, 15, 21, 27  | 1  | 1  | 1  | 1  | 0  | 1  |
| 4, 10, 16, 22, 28 | 1  | 1  | 0  | 1  | 1  | 1  |
| 5, 11, 17, 23, 29 | 1  | 1  | 1  | 0  | 1  | 1  |
| 6, 12, 18, 24, 30 | 1  | 0  | 0  | 0  | 0  | 1  |

### Содержание отчета и его форма

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования алгебраической системы по своему варианту и ответы на контрольные вопросы.

**Вопросы для защиты работы**

1. Определение конечного поля
2. Какие операции выполняются в конечном поле
3. Неприводимый полином и его свойства.
4. Минимальный многочлен и его свойства.
5. Корни минимального многочлена.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Лабораторная работа № 7

# РАСШИФРОВАНИЕ С ПОМОЩЬЮ САМОСИНХРОНИЗИРУЮЩИХСЯ ПОТОЧНЫХ ШИФРОВ

### Цель и содержание:

1. Углубить знания, по основам использования алгебраических структур для шифрования с двумя секретными ключами.
2. Исследовать основные характеристики алгоритма шифрования.

### Формируемые компетенции ОПК-6, ОПК-9

### Теоретическое обоснование

В основу данного алгоритма защиты информации от НСД получены положения теории чисел.

Положительные целые числа могут быть разделены на три группы: число 1, простые числа и составные объекты, как это показано на рис. 1. Положительное целое число — простое число тогда и только тогда, когда оно точно делимо без остатка на два целых числа — на 1 и на само себя. Составной объект — положительное целое число больше с чем двумя делителями.

Простое число делимо без остатка только на себя и 1.

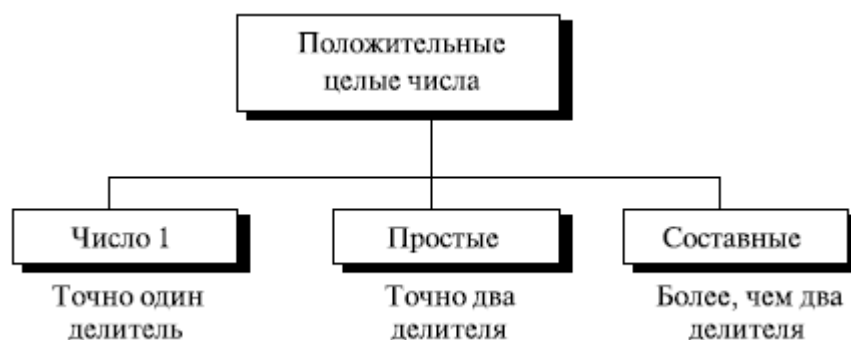


Рисунок 1 - Три группы положительных целых чисел

### Взаимно простые числа

Два положительных целых числа  $a$  и  $b$  являются взаимно простыми (coprime), если  $\text{НОД}(a, b) = 1$ , потому что число 1 является взаимно простым с

любым целым числом. Если  $p$  — простое число, тогда все числа от 1 до  $p-1$  являются взаимно простыми к  $p$ . В лекции 1 обсуждали множество  $Z_n^*$ , чьи элементы — все числа, взаимно простые с  $n$ . Множество  $Z^*$  является тем же самым, за исключением того, что модуль  $(p)$  — простое число.

### Количество простых чисел

После того как понятие простых чисел было определено, естественно возникает вопрос: число простых чисел конечно или бесконечно? Возьмем число  $n$ . Сколько есть простых чисел меньших, чем это число, или равных  $n$ ?

### Число простых чисел

Число простых чисел бесконечно. Приведем нестрогое доказательство: предположим, что множество простых чисел конечно (ограничено), и пусть  $p$  — наибольшее простое число. Перемножим все простые числа, входящие в это множество, и получим результат  $P = 2 \times 3 \times \dots \times p$ . Целое число  $(P + 1)$  не может иметь простого делителя  $q \leq p$  ( $p$  — наибольшее простое число). Тогда этот делитель должен быть одним из множителей, входящих в  $P$ . Это значит, что  $q$  делит  $P$ . Если  $q$  также делит  $(P + 1)$ , то  $q$  делит  $(P + 1) - P = 1$ . Единственное число, которое делит 1, — это сама 1, которая не является простым числом. Поэтому  $q$  должно быть большим, чем  $p$ , и ряд простых чисел не исчерпывается принятым конечным множеством.

Число простых чисел бесконечно.

Чтобы рассмотреть вторую возможность, введем функцию  $\pi(n)$ , которая определяет число простых чисел, меньших или равных  $n$ . Ниже показаны значения этой функции для различного  $\pi(n)$ .

$$\pi(1) = 0 \quad \pi(2) = 1 \quad \pi(3) = 2 \quad \pi(10) = 4 \quad \pi(20) = 8 \quad \pi(50) = 15 \quad \pi(100) = 25$$

Но если  $n$  является очень большим, как мы можем вычислить  $\pi(n)$ ? Для ответа мы можем использовать только приближение, которое показано ниже:

$$\left[ \frac{n}{\ln n} \right] < \pi(n) < \left[ \frac{n}{\ln n - 1.08366} \right]$$

Гаусс обнаружил верхний предел; Лагранж обнаружил нижний предел.

Пример 4

Найдите количество простых чисел, меньших, чем 1 000 000.

Решение

Приближение дает диапазон от 72 383 до 78 543. Фактическое число простых чисел — 78 498.

Проверка на простое число

Пример Действительно ли 97 — простое число?

Решение. Наибольшее ближайшее целое число —  $\sqrt{n} = 9$ . Простые числа меньше чем 9 — 2, 3, 5 и 7. Проверим, делимо ли без остатка 97 любым из этих номеров. Ответ: не делимо, так что 97 — простое число.

Пример 6 Действительно ли 301 — простое число?

Решение Наибольшее ближайшее целое число  $\sqrt{301} = 17$ . Мы должны проверить 2, 3, 5, 7, 11, 13 и 17. Числа 2, 3 и 5 не делят 301, но 7 — делит. Поэтому 301 — не простое число.

### Решето Эратосфена

Греческий математик Эратосфен изобрел метод, как найти все простые числа, меньшие, чем  $n$ .

Метод назван решетом Эратосфена. Предположим, что мы хотим найти все числа, меньшие, чем 100. Мы записываем все числа между 2 и 100. Поскольку  $\sqrt{100} = 10$ , мы должны видеть, делим ли без остатка любой номер меньше чем 100 на числа 2, 3, 5 и 7. Таблица 1 показывает результат.

Таблица 1. Решето Эратосфена

|    | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  |
|----|----|----|----|----|----|----|----|----|-----|
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20  |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30  |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40  |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50  |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60  |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70  |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80  |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90  |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |

Процесс состоит в следующем:

1. Вычеркнуть все числа, делимые без остатка на 2 (кроме самого 2).
2. Вычеркнуть все числа, делимые без остатка на 3 (кроме самого 3).
3. Вычеркнуть все числа, делимые без остатка на 5 (кроме самого 5).
4. Вычеркнуть все числа, делимые без остатка на 7 (кроме самого 7).
5. Оставшиеся числа – простые.

Phi-функция Эйлера

Phi-функция Эйлера,  $\varphi(n)$ , которую иногда называют тотиентой Эйлера, играет очень важную роль в криптографии. Функция  $\varphi(n)$  находит из ряда чисел  $0, 1, \dots, n-1$  числа, взаимно простые с  $n$ . Можно вспомнить из [лекции 2](#), что множество  $Z_n^*$  — числа, которые не больше чем  $n$  и взаимно простые с  $n$ . Функция  $\varphi(n)$  вычисляет число элементов этого множества. Ниже показано, как найти это значение:

1.  $\varphi(1) = 0$ .
2.  $\varphi(p) = p - 1$ , если  $p$  — простое число.
3.  $\varphi(m \times n) = \varphi(m) \times \varphi(n)$ , если  $m$  и  $n$  — взаимно простые.
4.  $\varphi(p^e) = p^e - p^{e-1}$ , если  $p$  — простое.

Мы можем объединить эти четыре правила, предназначенные для нахождения  $\varphi(n)$ .

$$\varphi(n) = (p_1^{e_1} - p_1^{e_1-1}) \times (p_2^{e_2} - p_2^{e_2-1}) \times \dots \times (p_k^{e_k} - p_k^{e_k-1})$$

Очень важно заметить, что значение  $\varphi(n)$  для больших чисел может быть найдено, если может быть найдено число  $n$  и если  $n$  может быть представлено в виде разложения простых чисел. Другими словами, трудность нахождения  $\varphi(n)$  зависит от трудности нахождения разложения  $n$ . Это рассматривается в следующем разделе. Трудность нахождения  $\varphi(n)$  зависит от трудности нахождения разложения  $n$ .

Пример 7 Какое значение имеет  $\varphi(13)$ ?

Решение Поскольку 13 — простое число,  $\varphi(13) = (13 - 1) = 12$ .

Пример 8 Какое значение имеет  $\varphi(10)$ ?

Решение Мы можем использовать третье правило:

$\varphi(10) = \varphi(2) \times \varphi(5) = 1 \times 4 = 4$ , поскольку 2 и 5 — простые числа.

Пример 9 Какое значение имеет  $\varphi(240)$ ?

Решение Мы можем записать  $240 = 2^4 \times 3^1 \times 5^1$ .

Тогда  $\varphi(240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 64$

Пример 10 Можно ли утверждать, что  $\varphi(49) = \varphi(7) \times \varphi(7) = 6 \times 6 = 36$ ?

Решение Нет.  $\varphi(49) = 7^2 - 7^1 = 42$

Пример 11 Какие числа являются элементами в  $Z_{14}^*$ ?

Решение  $\varphi(14) = \varphi(7) \times \varphi(2) = 6 \times 1 = 6$ . Элементы – это 1, 3, 5, 9, 11 и 13.

Интересный факт: если  $n > 2$ , значение  $\varphi(n)$  — четное.

Рассмотренные выше теоретические положения позволят понять алгоритм работы криптографической системы защиты информации с двумя секретными ключами.

Рассмотрим данный алгоритм. Пусть стороны  $A$  и  $B$  решили организовать секретную передачу информации между собой. Для этого выбирается большое число  $p$ , такое, что  $p-1$  хорошо разлагается на простые множители.

Далее абоненты независимо друг от друга осуществляют выбор ключей  $a$  и  $b$ , которые являются первыми секретными, согласно следующих правил:

$$1 < a < p - 2; 1 < b < p - 2.$$

Затем стороны производят вычисления вторых ключей, которые также являются секретными, согласно условия:

$$\begin{aligned} A : a \rightarrow \alpha \quad \alpha \cdot a &\equiv 1 \bmod \varphi(p) \\ B : b \rightarrow \beta \quad \beta \cdot b &\equiv 1 \bmod \varphi(p) \end{aligned}$$

где  $\varphi(p)$  – функция Эйлера.

1. Пусть абонент  $A$  решает передать сообщение  $n$ . Тогда сторона  $A$  зашифровывает это сообщение своим первым закрытым ключом:

$$n_1 = n^a \bmod p$$

Полученное сообщение отправляется на сторону  $B$ .

2. Тогда сторона  $B$  зашифровывает принятое сообщение своим первым закрытым ключом:

$$n_2 = n_1^b \bmod p$$

Полученное сообщение отправляется на сторону  $B$ .

3. Тогда сторона  $A$  зашифровывает принятое сообщение своим вторым закрытым ключом:

$$n_3 = n_2^a \bmod p$$

Полученное сообщение отправляется на сторону  $B$ .

4. Тогда сторона  $B$  зашифровывает принятое сообщение своим первым закрытым ключом:

$$n = n_3^b \bmod p$$

Полученное сообщение отправлялось на сторону  $B$  от абонента  $A$ .

Основным недостатком рассмотренной системы криптозащиты является неоднократная передача информации от одного пользователя к другому, что в конечном итоге приводит к низкой скорости передачи информации. Кроме того, довольно сложно подобрать число  $p$ , чтобы обеспечить необходимую надёжность защиты.

### **Аппаратура и материалы**

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.

2. Операционная система Windows 2000\XP.

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.

2. Провести исследование системы шифрования без передачи ключей.

Передаваемое сообщение  $m$  и кодирующее число  $p$  представлены в таблице 1.

Таблица 1 – Задание для исследования системы шифрования

| Вариант | Кодирующее число $p$ | Сообщение $m$ |
|---------|----------------------|---------------|
| 1       | 23                   | 17            |
| 2       | 29                   | 18            |
| 3       | 31                   | 19            |
| 4       | 37                   | 20            |

| Вариант | Кодирующее число $p$ | Сообщение $m$ |
|---------|----------------------|---------------|
| 5       | 41                   | 21            |
| 6       | 43                   | 22            |
| 7       | 47                   | 23            |
| 8       | 53                   | 24            |
| 9       | 23                   | 16            |
| 10      | 29                   | 15            |
| 11      | 31                   | 14            |
| 12      | 37                   | 13            |
| 13      | 41                   | 28            |
| 14      | 43                   | 29            |
| 15      | 47                   | 30            |
| 16      | 53                   | 31            |

Студенты самостоятельно выбирают значение ключей для обеих сторон (А и В) и исследуют процесс зашифрования и расшифрования сообщения.

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать исследование процесса зашифрования и расшифрования сообщения без передачи ключей по своему варианту и ответы на вопросы.

### **Вопросы для защиты работы**

1. Дайте определение и классификацию целых чисел.
2. Поясните алгоритм определения простоты числа.
3. Дайте определение функции Эйлера.
4. Вычислите функцию Эйлера для конкретного примера.
5. Принципы построения данного алгоритма шифрования с двумя секретными ключами.
6. Основные характеристики алгоритма шифрования.
7. Достоинства и недостатки шифрования с двумя секретными ключами.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6



## Лабораторная работа № 8

### ОСНОВЫ ПОСТРОЕНИЯ ПОТОЧНОГО ШИФРА S-A5

#### Цель и содержание:

1. Углубить знания, по основам шифрования с использованием теории чисел.
2. Исследовать основные характеристики алгоритма шифрования с открытым ключом.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Асимметрично-ключевая криптография широко использует простые числа. Тема простых чисел — большая часть любой книги по теории чисел.

В основу асимметричных криптографических алгоритмов положены теоремы Ферма и Эйлера.

#### Малая теорема Ферма

Малая теорема Ферма играет очень важную роль в теории чисел и криптографии. Ниже мы приводим две версии теоремы.

#### Первая версия

Первая версия говорит, что если  $p$  — простое число и  $a$  — целое число, такое, что  $p$  не является делителем  $a$ , тогда  $a^{p-1} \equiv 1 \pmod{p}$ .

#### Вторая версия

Вторая версия вводит ограничивающее условие на  $a$ . Она утверждает, что если  $p$  — простое число и  $a$  — целое число, то  $a^p \equiv a \pmod{p}$ .

#### Приложения

Хотя мы будем рассматривать приложения этой теоремы позже в этой лекции, теорема очень полезна для того, чтобы решить некоторые проблемы.

Возведение в степень. Малая теорема Ферма иногда полезна для того, чтобы быстро найти решение при возведении в степень. Следующие примеры показывают это.

Пример 1. Найдите результат  $6^{10} \bmod 11$ .

Решение Мы имеем  $6^{10} \bmod 11 \equiv 1$ . Это первая версия малой теоремы Ферма, где  $p = 11$ .

Пример 2. Найдите результат  $3^{12} \bmod 11$ .

Решение Здесь степень (12) и модуль (11) не соответствуют условиям теоремы Ферма. Но, применяя преобразования, мы можем привести решение к использованию малой теоремы Ферма.

$$3^{12} \bmod 11 \equiv (3^{11} \times 3) \bmod 11 \equiv (3^{11} \bmod 11)(3 \bmod 11) \equiv (3 \times 3) \bmod 11 = 9$$

### Мультипликативные инверсии.

Очень интересное приложение теорема Ферма находит для некоторых мультипликативных инверсий, если модуль — простое число. Если  $p$  — простое число и  $a$  — целое число, такое, что  $p$  не является его делителем, тогда  $a^{-1} \bmod p = a^{p-2} \bmod p$ . Это может быть легко доказано, если мы умножим обе стороны равенства на  $a$  и используем первую версию малой теоремы Ферма:

$$a \times a^{-1} \bmod p \equiv a \times a^{p-2} \bmod p \equiv a^{p-1} \bmod p \equiv 1 \bmod p$$

Это приложение позволяет не использовать расширенный алгоритм Евклида для нахождения мультипликативных инверсий.

Пример 3. Инверсии по модулю простого числа могут быть найдены без использования расширенного Евклидова алгоритма:

a.  $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$

b.  $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$

c.  $6^{-1} \bmod 101 = 6^{101-2} \bmod 101 = 6^{99} \bmod 101 = 32 \bmod 101$

d.  $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

### Теорема Эйлера

Теорему Эйлера можно представить как обобщения малой теоремы Ферма. Модуль в теореме Ферма — простое число, модуль в теореме Эйлера — целое число. Мы вводим две версии этой теоремы.

## Первая версия

Первая версия теоремы Эйлера подобна первой версии малой теоремы Ферма. Если  $a$  и  $n$  – взаимно простые, то  $a^{\varphi(n)} \equiv 1 \pmod n$ .

## Вторая версия

Вторая версия теоремы Эйлера подобна второй версии малой теоремы Ферма; она устраняет условие, что  $n$  должно быть взаимно простым с  $a$ . Если  $n = p \times q, a < n$ , а  $k$  — целое число, то  $a^{k \times \varphi(n)+1} \equiv a \pmod n$ .

Приведем нестрогое доказательство второй версии, основанной на первой версии. Поскольку  $a < n$ , то возможны три случая:

1. Если  $a$  не кратно ни числу  $p$ , ни числу  $q$ , то  $a$  и  $n$  – взаимно простые.

$$a^{k \times \varphi(n)+1} \pmod n = (a^{\varphi(n)})^k \times a \pmod n = (1)^k \times a \pmod n = a \pmod n$$

2. Если  $a$  — кратное числу  $p$ ,  $a = (i \times p)$ , но не кратно числу  $q$ .

$$a^{\varphi(n)} \pmod q = (a^{\varphi(q)} \pmod q)^{\varphi(p)} \pmod q = 1 \rightarrow a^{\varphi(n)} \pmod q = 1$$

$$a^{k \times \varphi(n)} \pmod q = (a^{\varphi(n)} \pmod q)^k \pmod q = 1 \rightarrow a^{k \times \varphi(n)} \pmod q = 1$$

$$a^{k \times \varphi(n)} \pmod q = 1 \rightarrow a^{k \times \varphi(n)} = 1 + j \times q \text{ (интерпретация сравнения)}$$

$$a^{k \times \varphi(n)+1} = a \times (1 + j \times q) = a + j \times q \times a = a + (i \times j) \times q \times p = a + (i \times j) \times n$$

$$a^{k \times \varphi(n)+1} = a + (i \times j) \times n \rightarrow a^{k \times \varphi(n)+1} \equiv a \pmod n \text{ (отношение конгруэнтности)}$$

3. Если  $a$  кратно  $q$  ( $a = i \times q$ ), но не кратно  $p$ , доказательство второго случая то же самое, но  $p$  и  $q$  меняются местами.

Вторая версия теоремы Эйлера используется в криптографической системе RSA.

В качестве асимметричного алгоритма защиты данных с открытым ключом рассмотрим алгоритм RSA. Алгоритм асимметричного шифрования RSA является одним из первых полноценных алгоритмов с открытым ключом. Он был разработан Рональдом Риверсом, Ади Шамиром и Леонардо Адлеманом в 1976 году.

Суть данного алгоритма шифрования состоит в следующем. Пусть абоненты  $A$  и  $B$  решили наладить между собой секретную переписку с открытым ключом. Тогда каждый из них, независимо от другого, выбирает 2 больших простых числа, находит их произведение, функцию Эйлера  $\varphi(P)$  от

произведения, этих чисел, а затем выбирает случайное число согласно алгоритма:

$$A:\Rightarrow p_1, p_2; P_1 = p_1 \cdot p_2; \Rightarrow \varphi(P_1) = (p_1 - 1)(p_2 - 1); \Rightarrow 0 < a < \varphi(P_1), (a, \varphi(P_1)) = 1;$$

$$B:\Rightarrow q_1, q_2; Q_1 = q_1 \cdot q_2; \Rightarrow \varphi(Q_1) = (q_1 - 1)(q_2 - 1); \Rightarrow 0 < b < \varphi(Q_1), (b, \varphi(Q_1)) = 1.$$

Затем открытые ключи  $a$  и  $P_1$ , а так же  $b$  и  $Q_1$  печатаются в телефонной книге.

Каждый из абонентов независимо от другого выбирает свой секретный ключ согласно условия

$$A:\alpha \Rightarrow a \cdot \alpha \equiv 1 \pmod{\varphi(P_1)};$$

$$B:\beta \Rightarrow b \cdot \beta \equiv 1 \pmod{\varphi(Q_1)}.$$

Пусть абонент  $A$  решает послать сообщение  $m$  абоненту  $B$  и пусть  $0 < m < P_1$ , иначе текст делят на блоки.

3. Абонент  $A$  шифрует сообщение  $m$  открытым ключом абонента  $B$ , который есть телефонной книге, и находит:

$$m_i \equiv m^b \pmod{Q_1}$$

4. Абонент  $B$  расшифровывает сообщение своим секретным ключом  $\beta$  :

$$m \equiv m_i^\beta \pmod{Q_1}$$

Пусть абонент  $B$  решает послать сообщение  $n$  абоненту  $A$  и пусть  $0 < n < Q_1$ , иначе текст делят на блоки.

3. Абонент  $B$  шифрует сообщение  $n$  открытым ключом абонента  $A$ , который есть телефонной книге, и находит:

$$n_i \equiv n^a \pmod{P_1}$$

4. Абонент  $A$  расшифровывает сообщение секретным ключом  $\alpha$  :

$$n \equiv n_i^\alpha \pmod{P_1}$$

### **Аппаратура и материалы**

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.

2. Операционная система Windows 2000\XP.

### **Методика и порядок выполнения работы**

5. Изучить теоретический материал работы.

## 6. Провести исследование системы RSA.

Передаваемое сообщение  $m$  и кодирующее число  $p$  представлены в таблице 1

Таблица 1 – Задание для исследования системы RSA

| Вариант | Кодирующее<br>число | Кодирующее<br>число | Сообщение $m$ |
|---------|---------------------|---------------------|---------------|
| 1       | 23                  | 7                   | 17            |
| 2       | 29                  | 13                  | 18            |
| 3       | 31                  | 11                  | 19            |
| 4       | 37                  | 17                  | 20            |
| 5       | 41                  | 19                  | 21            |
| 6       | 43                  | 5                   | 22            |
| 7       | 47                  | 7                   | 23            |
| 8       | 53                  | 11                  | 24            |
| 9       | 23                  | 13                  | 16            |
| 10      | 29                  | 17                  | 15            |
| 11      | 31                  | 23                  | 14            |
| 12      | 37                  | 29                  | 13            |
| 13      | 41                  | 11                  | 28            |
| 14      | 43                  | 13                  | 29            |
| 15      | 47                  | 17                  | 30            |
| 16      | 53                  | 5                   | 31            |

Студенты самостоятельно выбирают значение ключей для обеих сторон ( $A$  и  $B$ ) и исследуют процесс зашифрования и расшифрования сообщения.

### Содержание отчета и его форма

Отчет по лабораторной работе должен содержать процесс исследования системы RSA по своему варианту и ответы на вопросы.

### Вопросы для защиты работы

1. Поясните первую версию малой теоремы Ферма
2. Поясните вторую версию малой теоремы Ферма.
3. Покажите на примерах использование теоремы Ферма.
4. Поясните первую версию малой теоремы Эйлера
5. Поясните вторую версию малой теоремы Эйлера.
6. Покажите на примерах использование теоремы Эйлера
7. Основные принципы построения алгоритма RSA.

8. Основные характеристики алгоритма RSA.
9. По заданным числам определить открытый и секретный ключи для симметричного алгоритма защиты данных.
10. Достоинства и недостатки симметричного и RSA шифрования.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## Лабораторная работа № 9

### РАСШИФРОВАНИЕ С ПОМОЩЬЮ ПОТОЧНОГО ШИФРА S-A5

#### Цель и содержание:

1. Углубить знания, по основам асимметричного шифрования.
2. Исследовать основы алгоритма шифрования Эль-Гамала.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Криптосистема Эль-Гамала была предложена в 1985 году. Она используется для получения, как для цифровой подписи, так и для шифрования. Криптостойкость асимметричного алгоритма шифрования определяется трудоемкостью вычисления дискретного логарифма в конечном поле Галуа. Для генерации пары ключей выбирается простое число  $p$  и два случайных числа  $g$  и  $x$ , причем  $g < p$ ,  $x < p$ . Затем вычисляется значение:

$$y = g^x \bmod p$$

Открытым ключом являются:

1.  $p$  - простое число (может быть общим для группы абонентов).
2.  $g < p$  (может быть общим для группы абонентов).
3.  $y = g^x \bmod p$ .

Секретным ключом является значение:  $x < p$ .

Процесс шифрования осуществляется следующим образом. Для шифрования сообщения  $M$  выбирается случайное число  $k$ , такое, что

$$\text{НОД}(k, (p-1)) = 1.$$

Затем вычисляется первая часть шифрования:

$$a = g^k \bmod p,$$

и вторая часть шифрования:

$$b = y^k \cdot M \bmod p$$

Пара  $(a, b)$  называется шифротекстом. Следует отметить, шифротекст имеет длину в два раза больше длины исходного текста. Полученная пара передается на противоположную сторону.

Процесс расшифрования осуществляется следующим образом:

$$M = \frac{b}{a^x} \bmod p$$

Рассмотрим процесс зашифрования текста с использованием ассиметричного алгоритма Эль-Гамала. Пусть в качестве ключей выбрали  $p=11$  и  $g=2$ , ( $2 < 11$ ). Определяем секретный ключ  $x=8$ , ( $8 < 11$ ). Тогда вычислением значения,

$$y = 2^8 \bmod 11 = 3$$

В справочнике печатается: *пользователь А*  $y=2, p=11, y=3$ .

Пусть необходимо передать тест  $M=5$ .

Выбираем число  $k=9$ , так как  $\text{НОД}(9, 10)=1$ ,  $k(p-1)=11-1=10$ .

Определяем значение первой части шифра:

$$a = g^k \bmod 11 = 2^9 \bmod 11 = 6$$

Определяем вторую часть шифра:

$$b = y^k M \bmod p = 3^9 \cdot 5 \bmod 11 = 27 \bmod 11 = 5$$

Зашифрованное сообщение в виде пары  $(6, 5)$  передается на другую сторону к пользователю *В*. Затем проводится процесс дешифрования. Тогда:

$$M = \frac{9}{6^8} \bmod 11 = \frac{1}{9} \bmod 11 = 5$$

### **Аппаратура и материалы**

3. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.

4. Операционная система Windows 2000\XP.

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.
2. Провести исследование криптосистемы Эль-Гамала.

Передаваемое сообщение  $m$  открытые ключи  $p$ ,  $g$ , и секретный ключ представлены в таблице 1.

Таблица 1 – Задание для исследования криптосистемы Эль-Гамала

| Вариант | Открытый ключ |     | Закрытый ключ,<br>$x$ | Сообщение $m$ |
|---------|---------------|-----|-----------------------|---------------|
|         | $p$           | $g$ |                       |               |
| 1       | 23            | 10  | 11                    | 17            |
| 2       | 29            | 11  | 12                    | 23            |
| 3       | 23            | 12  | 13                    | 19            |
| 4       | 31            | 13  | 14                    | 20            |
| 5       | 17            | 14  | 15                    | 15            |
| 6       | 19            | 15  | 16                    | 18            |
| 7       | 37            | 7   | 10                    | 30            |
| 8       | 41            | 6   | 11                    | 24            |
| 9       | 23            | 12  | 13                    | 21            |
| 10      | 29            | 14  | 15                    | 23            |
| 11      | 23            | 15  | 14                    | 17            |
| 12      | 31            | 10  | 10                    | 14            |
| 13      | 17            | 12  | 11                    | 13            |
| 14      | 19            | 17  | 12                    | 15            |
| 15      | 37            | 18  | 13                    | 21            |
| 16      | 41            | 9   | 20                    | 11            |

Студенты самостоятельно выбирают значение  $k$  согласно условию и исследуют процедуру зашифрования и расшифрования согласно варианту.

### Содержание отчета и его форма

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования криптосистемы Эль-Гамала по своему варианту и ответы на контрольные вопросы.

### Вопросы для защиты работы

1. Основные принципы построения алгоритма Эль-Гамала.
2. Основные характеристики алгоритма Эль-Гамала.
3. Достоинства и недостатки алгоритма шифрования Эль-Гамала.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6



## Лабораторная работа № 10

### АРИФМЕТИКА ЦЕЛЫХ ЧИСЕЛ

#### Цель и содержание:

1. Углубить знания, по основам использования цифровой подписи.
2. Исследовать основные характеристики алгоритма построения электронной подписи Эль-Гамала.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Одним из основных методов обеспечения безопасности информационных систем является цифровые шифрования. В современных шифровочных устройствах имеют дело с большим объёмом информации, записанной в  $q$ -ной системе счисления ( $q = 2^4 - 10^4$ ). В процессе переработки данной информации осуществляются различные арифметические операции или преобразования. При этом возникает задача выбора эффективного метода выполнения этих действий или преобразований.

Криптосистема «открытый ключ» неудобна в том смысле, что получатель сообщения не знает, кто является отправителем сообщения. Этого недостатка лишена система «электронная подпись».

Для получения цифровой подписи выбирается простое число  $p$  и два случайных числа  $g$  и  $x$ , причем  $g < p$ ,  $x < p$ . Затем вычисляется значение

$$y = g^x \bmod p$$

Открытым ключом являются:

1.  $p$  - простое число (может быть общим для группы абонентов).
2.  $g < p$  (может быть общим для группы абонентов).
3.  $y = g^x \bmod p$ .

Секретным ключом является значение:  $x < p$ .

Чтобы подписать сообщение  $M$ , сначала выбирается число  $k$ , взаимно-простое с  $p-1$ . Затем вычисляется:

$$a = g^k \bmod p$$

Затем находится  $b$  из условия

$$M = (xa + kb) \bmod (p - 1)$$

Подписью является пара чисел  $a$  и  $b$ . Случайное значение  $k$  хранится в секрете. Для проверки подписи надо убедиться, что

$$y^a a^b \bmod p = g^M \bmod p$$

Каждая подпись требует нового значения  $k$ , следовательно,  $k$  выбирается случайным образом.

Рассмотрим процесс построения электронной подписи Эль-Гамала.

Пусть  $p = 11$  и случайно выбираем  $g = 2$ . Пусть секретный ключ  $x = 8$ , тогда

$$y = g^x \bmod p$$

$$y = 2^8 \bmod 11 = 3$$

Открытым ключом являются  $y = 3$ ,  $g = 2$  и  $p = 11$ . Необходимо подписать  $M = 5$ . Выбираем случайное число  $k = 9$ , оно взаимно простое с  $p-1$ , которое равно 10. Далее вычисляем

$$a = g^k \bmod 11 = 2^9 \bmod 11 = 6$$

Затем с помощью расширенного алгоритма Евклида найдем  $b$  из уравнения

$$M = (ax + kb) \bmod p - 1$$

$$5 = (6 \cdot 8 + 9b) \bmod 10$$

$$8 + 9 \cdot 1 \bmod 10 = 7$$

$$8 + 9 \cdot 2 \bmod 10 = 6$$

$$8 + 9 \cdot 3 \bmod 10 = 5$$

следовательно,  $b = 3$ .

Таким образом, подпись представляет собой пару чисел  $a = 6$  и  $b = 3$ .

Проверяем правильность подписи согласно

$$y^a a^b \bmod p = g^M \bmod p$$

(пересылаем  $M$  и совместно с ним  $a$ ,  $b$ , следовательно, на приемной стороне при истинном значении последнего выражения принимается решение о достоверности переданного).

**Аппаратура и материалы**

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.

2. Операционная система Windows 2000\XP.

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.

2. Провести исследование системы цифровой подписи.

Передаваемое сообщение  $m$  открытые ключи  $p$ ,  $g$ , и секретный ключ представлены в таблице 1.

Таблица 1 – Задание для исследования системы

| Вариант | Открытый ключ |     | Закрытый ключ,<br>$x$ | Сообщение $m$ |
|---------|---------------|-----|-----------------------|---------------|
|         | $p$           | $g$ |                       |               |
| 1       | 23            | 12  | 13                    | 21            |
| 2       | 29            | 14  | 15                    | 23            |
| 3       | 23            | 15  | 14                    | 17            |
| 4       | 31            | 10  | 10                    | 14            |
| 5       | 17            | 12  | 11                    | 13            |
| 6       | 19            | 17  | 12                    | 15            |
| 7       | 37            | 18  | 13                    | 21            |
| 8       | 41            | 9   | 20                    | 11            |
| 9       | 23            | 10  | 11                    | 17            |
| 10      | 29            | 11  | 12                    | 23            |
| 11      | 23            | 12  | 13                    | 19            |
| 12      | 31            | 13  | 14                    | 20            |
| 13      | 17            | 14  | 15                    | 15            |
| 14      | 19            | 15  | 16                    | 18            |
| 15      | 37            | 7   | 10                    | 30            |
| 16      | 41            | 6   | 11                    | 24            |

Студенты самостоятельно выбирают значение  $k$  согласно условию и исследуют процедуру получения цифровой подписи согласно варианту.

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследование системы цифровой подписи на основе алгоритма Эль-Гамала по своему варианту и ответы на вопросы.

### **Вопросы для защиты работы**

1. Основные принципы построения цифровой подписи Эль-Гамала.

2. Основные характеристики алгоритма Эль-Гамала, используемого для получения цифровой подписи.

3. Сравнительная характеристика алгоритмов получения цифровой подписи RSA и Эль-Гамала.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Лабораторная работа №11**

### **МОДУЛЬНАЯ АРИФМЕТИКА**

#### **Цель и содержание:**

1. Углубить знания, по основам применения теории чисел для разработки протоколов аутентификации.
2. Исследовать основы протокола аутентификации с использованием доказательства с нулевым разглашением.

#### **Формируемые компетенции ОПК-6, ОПК-9**

#### **Теоретическое обоснование**

Повысить имитостойкость системы защиты данных от НСД можно за счет построения системы на основе протокола с нулевым разглашением, которые используются в протоколах аутентификации. Особое место среди таких протоколов занимают протоколы аутентификации типа «запрос-ответ» в котором одна сторона (претендент) доказывает свою идентичность другой стороне (проверяющему) посредством демонстрации знания определенного секрета. Претендент должен ответить на запрос, меняющийся во времени, причем ответ должен зависеть и от значения его секрета и от запроса.

В работе [1] приведены наиболее известные вычислительно-сложные задачи теории чисел, на которых базируется протокол защиты данных от НСД. Предложенный алгоритм аутентификации, используемый в разрабатываемой системе, основан на доказательстве с нулевым разглашением и относится к задаче распознавания Диффи-Хеллмана (DDHP). Стойкость задач Диффи-Хеллмана базируется на сложности вычисления дискретного логарифма. В настоящее время решить данную задачу дискретного логарифмирования за минимальное время можно только с помощью субэкспоненциального алгоритма, что значительно больше, чем время перебора возможных коммутаций, выполняемых шифровальной колодкой, приведенной в [1].

**Рассмотрим алгоритм протокола** аутентификации с использованием доказательства с нулевым разглашением. Такие протоколы, нашли широкое применение в системах электронных платежей, которые работают с электронной наличностью. Примеры реализаций таких протоколов приведены [1-2]. Использование разработанных протоколов с нулевым разглашением позволяет обеспечить решение такой проблемы, так как данные протоколы обладают хорошей криптографической стойкостью.

Для физической реализации данных протоколов необходимо выполнение алгоритма [14,15].

**Первый этап.** В память вычислительного устройства ответчика вводятся числа  $U, S, T$ . В этом случае число  $U$  выступает в качестве долгосрочного секретного ключа.

Числа  $S$  и  $T$  являются базовой основой, для вычисления сеансовых ключей  $S(i)$  и  $T(i)$ .

Для определения сеансовых ключей  $S(i)$  и  $T(i)$  предлагается использовать псевдослучайную функцию (ПСФ). Данная ПСФ, принимающая на входную последовательность  $(x_1, \dots, x_n)$  и ключ  $(g, s_1, \dots, s_n)$ , обеспечивает выполнение

$$F((s_1, \dots, s_n), (g, x_1, \dots, x_n)) = g^{\left( \frac{1}{\prod_{i=1}^n (s_i + x_i)} \right)}, \quad (1)$$

где  $g$  – первообразный элемент мультипликативной группы.

При этом, имитостойкость данной ПСФ основывается на доказательстве о сложности решения  $\lambda$ -DDH проблемы. Тогда имеем

$$S(i) = g^{\frac{1}{S+i+1}} \bmod q, \quad (2)$$

$$T(i) = g^{\frac{1}{T+i+1}} \bmod q, \quad (3)$$

где  $q$  – мультипликативная группа;  $g$  – первообразный элемент этой группы;  $i$  – номер проводимого сеанса.

**Второй этап.** Используя полученные данные  $U$ ,  $S(i)$  и  $T(i)$ , вычислительное устройство пользователя вычисляет истинный статус объекта согласно

$$\tilde{N}(i) = g^U g^{S(i)} g^{T(i)} \bmod q. \quad (4)$$

Значение истинного статуса записывается в блок памяти претендента (пользователя).

**Третий этап.** Затем в вычислительном устройстве, используя полученные данные  $U$ ,  $S(i)$  и  $T(i)$ , производится их зашумление

$$U^*(i) = U + \Delta U \bmod q, \quad (5)$$

$$S^*(i) = S(i) + \Delta S \bmod q, \quad (6)$$

$$T^*(i) = T(i) + \Delta T \bmod q, \quad (7)$$

где  $\Delta U, \Delta S, \Delta T$  - величины зашумления  $U, S(i)$  и  $T(i)$  соответственно.

После этого ответчик вычисляет зашумленный статус

$$\tilde{N}^*(i) = g^{U^*} g^{S^*(i)} g^{T^*(i)} \bmod q. \quad (8)$$

Значение зашумленного статуса записывается в блок памяти претендента (пользователя).

**Четвертый этап.** При появлении запроса от объекта проверяющий генерирует «запросное число»  $d$  и пересылает его претенденту.

**Пятый этап.** Получив «запросное число»  $d$  претендент (пользователь) производит вычисление ответов

$$r_1 = U^* - dU \bmod \phi(q), \quad (9)$$

$$r_2 = S(i)^* - dS(i) \bmod \phi(q), \quad (10)$$

$$r_3 = T(i)^* - dT(i) \bmod \phi(q), \quad (11)$$

где  $\phi(q)$  - функция Эйлера числа  $q$ .

**Шестой этап.** Окончив выполнение вычислений, претендент (пользователь) передает проверяющему сигнал, который содержит:

- вычисленный истинный статус  $C(i)$ ;
- вычисленный зашумленный статус  $C^*(i)$ ;
- ответы на вопросы  $r_1, r_2, r_3$ .

**Седьмой этап.** Проверяющий, получив этот сигнал, производит вычисление результата

$$Y(i) = \tilde{N}^d(i) g^{r_1} g^{r_2} g^{r_3} \bmod q. \quad (12)$$

Если вычисленное значение  $Y(i)$  совпадет со значением зашумленного статуса претендента (пользователя), т.е.  $Y(i) = \tilde{N}^*(i)$ , то принимается решение, что статус претендента (пользователя) «свой». Между абонентским терминалом, производится обмен данными.

Если вычисленное значение  $Y(i)$  не совпадет со значением зашумленного статуса претендента (пользователя), т.е.  $Y(i) \neq \tilde{N}^*(i)$ , то принимается решение, что статус претендента (пользователя) «чужой» и обмен информацией не производится.

Рассмотрим пример реализации протокола. Пусть в качестве числа  $q=11$ . Для данной мультипликативной группы выбираем  $g=2$ . В качестве элементов секрета выбираем  $U=3$ ,  $S=4$ ,  $T=6$ . Тогда согласно (4) получаем, что значение истинного статуса

$$C = q^3 \cdot q^4 \cdot q^6 \bmod q = 2^{13} \bmod 11 = 2^3 \bmod 11 = 8$$

Пусть осуществляется «зашумление значений  $U$ ,  $S$ ,  $T$ ». для этого снимаются значения  $\Delta U=2$ ,  $\Delta S=5$ ,  $\Delta T=4$ . Тогда

$$U^* = U + \Delta U \bmod q = 3 + 2 \bmod 11 = 5$$

$$S^* = S + \Delta S \bmod q = 4 + 5 \bmod 11 = 9$$

$$T^* = T + \Delta T \bmod q = 6 + 4 \bmod 11 = 10$$

Полученные зашумленные  $U^*$ ,  $S^*$  и  $T^*$  подаются на входы первого вычислительного устройства 9, которое реализует (8). Тогда

$$C^* = g^{U^*} \cdot g^{S^*} \cdot g^{T^*} \bmod q = 2^5 \cdot 2^9 \cdot 2^{10} \bmod 11 = 2^{24} \bmod 11 = 16 \bmod 11 = 5$$

Данное значение  $C^*$  через второй блок коммутации 10, а затем записывается во второй регистр 17.

Пусть проверяющий передает вопрос  $d=6$ .

Проведем определение «ответа» на поставленные вопрос  $d=6$

$$r_1 = U^* - dU \bmod \phi(q) = 5 - 6 \cdot 3 \bmod 10 = -3 \bmod 10 = 7$$

$$r_2 = S^* - dS \bmod \phi(q) = 9 - 6 \cdot 4 \bmod 10 = -5 \bmod 10 = 5$$

$$r_3 = T^* - dT \bmod \phi(q) = 10 - 6 \cdot 6 \bmod 10 = -6 \bmod 10 = 4$$

Затем вычисленные значения  $(C, C^*, r_1, r_2, r_3)$  поступают на передатчик ответчика и передаются проверяющему.

Проверяющий реализует выражение (12).

$$Y = C^d \cdot g^{r_1} \cdot g^{r_2} \cdot g^{r_3} \mod q = 8^6 \cdot 2^7 \cdot 2^5 \cdot 2^4 \mod 11 = \\ = 2^{18} \cdot 2^{16} \mod 11 = 2^{34} \mod 11 = 2^4 \mod 11 = 5$$

Так как значение  $C^*$  совпало со значением  $Y$ , то выход системы опознавания поступает сигнал «1», что соответствует статусу «свой».

### Аппаратура и материалы

5. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 7.

6. Операционная система Windows 7.

### Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Провести исследование протокола аутентификации с использованием доказательства с нулевым разглашением.

Исходные данные для исследований представлены в таблице 1.

Таблица 1 – Задание для исследования протокола аутентификации

| Вариант | Число $p$ | Секретный ключ, $U$ | Значение параметра $S$ | Значение параметра $T$ |
|---------|-----------|---------------------|------------------------|------------------------|
| 1       | 23        | 10                  | 11                     | 17                     |
| 2       | 29        | 11                  | 12                     | 23                     |
| 3       | 23        | 12                  | 13                     | 19                     |
| 4       | 31        | 13                  | 14                     | 20                     |
| 5       | 17        | 14                  | 15                     | 15                     |
| 6       | 19        | 15                  | 16                     | 18                     |
| 7       | 37        | 7                   | 10                     | 30                     |
| 8       | 41        | 6                   | 11                     | 24                     |
| 9       | 23        | 12                  | 13                     | 21                     |
| 10      | 29        | 14                  | 15                     | 23                     |
| 11      | 23        | 15                  | 14                     | 17                     |
| 12      | 31        | 10                  | 10                     | 14                     |
| 13      | 17        | 12                  | 11                     | 13                     |
| 14      | 19        | 17                  | 12                     | 15                     |
| 15      | 37        | 18                  | 13                     | 21                     |
| 16      | 41        | 9                   | 20                     | 11                     |

Студенты самостоятельно выбирают значение  $k$  согласно условию и исследуют процедуру проверки статуса претендента (пользователя) с использованием протокола аутентификации согласно варианту.

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно, должен содержать процесс исследования протокола аутентификации с использованием доказательства с нулевым разглашением по своему варианту и ответы на контрольные вопросы.

### **Вопросы для защиты работы**

1. Основные принципы построения протокола аутентификации с использованием доказательства с нулевым разглашением
2. Поясните алгоритм получения истинного статуса претендента (пользователя)
3. Поясните алгоритм получения зашумленного статуса претендента (пользователя)
4. Приведите пример вычисления сеансовых ключей протокола аутентификации с использованием доказательства с нулевым разглашением.
5. Основные характеристики протокола аутентификации с использованием доказательства с нулевым разглашением.
6. Достоинства и недостатки данного протокола.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

## **Лабораторная работа №12**

### **АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ**

#### **Цель и содержание:**

1. Углубить знания, полученные на лекциях, по основам ключевого экспоненциального ключевого обмена.
2. Исследовать основные характеристики метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана.

#### **Формируемые компетенции ОПК-6, ОПК-9**

#### **Теоретическое обоснование**

Различают следующие типы протоколов распределения ключей:

- протоколы передачи (уже сгенерированных) ключей;
- протоколы (совместной) выработки общего ключа (открытое распределение ключей);
- схемы предварительного распределения ключей.

Различают также протоколы распределения ключей между отдельными участниками и между группами участников информационного взаимодействия.

Метод экспоненциального ключевого обмена Диффи-Хелмана основан на открытом ключе. Криптостойкость определяется трудностью вычисления дискретного логарифма.

#### **Описание алгоритма Диффи-Хелмана**

Выбираются:

- простое число  $q$ ;
- число  $\alpha$ , являющееся первообразным конем  $q$ .

Сторона  $A$  выбирает случайное число  $X_A < q$  и вычисляет значение:

$$Y_A = \alpha^{X_A} \bmod q$$

Пользователь  $B$  случайно выбирает  $X_B < q$  и вычисляет:

$$Y_B = \alpha^{X_B} \bmod q$$

Стороны обмениваются полученными значениями  $Y_A$  и  $Y_B$ , сохраняя в тайне значения  $X_A$  и  $X_B$ .

Тогда сторона  $A$  вычисляет ключ по формуле:

$$K_A = (Y_B)^{X_A} \bmod q$$

При этом сторона  $B$  определяет ключ согласно:

$$K_B = (Y_A)^{X_B} \bmod q = (\alpha^{X_A})^{X_B} \bmod q$$

Следует отметить, что данный алгоритм можно использовать при односторонней генерации ключей

### **Односторонняя генерация ключа.**

1. Сторона  $A$  выбирает целое число  $X_A$  и генерирует

$$k = \alpha^{X_A} \bmod q$$

2. Сторона  $B$  выбирает случайное целое  $X_B$  и передаёт на сторону  $A$  значение

$$Y_B = \alpha^{X_B} \bmod q$$

3. Сторона  $A$  посылает стороне  $B$  значение

$$Y_A = (Y_B)^{X_A} \bmod q = \alpha^{X_B \cdot X_A} \bmod q$$

4. Сторона  $B$  вычисляет сначала значение

$$Z = X_B^{-1} \bmod q$$

А затем осуществляет вычисление значения ключа

$$k^* = (Y_A)^Z \bmod q = (\alpha^{X_B \cdot X_A})^Z \bmod q = \left( \alpha^{X_B \cdot X_A \cdot \frac{1}{X_B}} \right) \bmod q = \alpha^{X_A} \bmod q$$

Имеем ключ  $k = k^*$ .

Описанный протокол ключевого обмена можно расширить и для многих участников.

**Первый этап.** Сторона  $A$  выбирает значение  $X_A$  вычисляет и посылает на сторону  $B$  вычисленное значение:

$$Y_A = \alpha^{X_A} \bmod q$$

При этом сторона  $A$  получила  $Y_C$  от стороны  $C$ .

Сторона  $B$  выбирает значение  $X_B$  и вычисляет:

$$Y_B = \alpha^{X_B} \bmod q$$

Полученное значение посылается на сторону С, получив при этом  $Y_A$ .

Сторона С выбирает значение  $X_C$  и вычисляет:

$$Y_C = \alpha^{X_C} \bmod q$$

Полученное значение было послано на сторону А. При этом сторона С получила значение  $Y_B$ .

**Второй этап.** Станция А пересылает станции В значения

$$Z_A = (Y_C)^{X_A} \bmod q = (\alpha^{X_C \cdot X_A}) \bmod q$$

Станция В пересылает станции С значения

$$Z_B = (Y_A)^{X_B} \bmod q = (\alpha^{X_A \cdot X_B}) \bmod q$$

Станция С пересылает на станцию А значения

$$Z_C = (Y_B)^{X_C} \bmod q = (\alpha^{X_B \cdot X_C}) \bmod q$$

**Третий этап.** Станция А вычисляет ключ

$$K_A = (Z_C)^{X_A} \bmod q = (\alpha^{X_B \cdot X_C}) \bmod q = \alpha^{X_A X_B X_C} \bmod q$$

Станция В вычисляет ключ

$$K_B = (Z_A)^{X_B} \bmod q = (\alpha^{X_A \cdot X_C})^{X_B} \bmod q$$

Станция С вычисляет ключ

$$K_C = (Z_B)^{X_C} \bmod q = (\alpha^{X_A \cdot X_B})^{X_C} \bmod q$$

## Аппаратура и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.

2. Операционная система Windows 2000\XP.

## Методика и порядок выполнения работы

1. Изучить теоретический материал работы.

2. Провести исследование метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана для двух сторон.

Кодирующее число  $g$  числа  $X_A$  и  $X_B$  представлены в таблице 1.

Таблица 1 – Задание для исследования метода экспоненциального

ключевого обмена на основе алгоритма Диффи-Хелмана для двух сторон.

| Вариант | Кодирующее число $q$ | Число $X_A$ | Число $X_B$ |
|---------|----------------------|-------------|-------------|
| 1       | 23                   | 7           | 17          |
| 2       | 29                   | 13          | 18          |
| 3       | 31                   | 11          | 19          |
| 4       | 37                   | 17          | 20          |
| 5       | 41                   | 19          | 21          |
| 6       | 43                   | 5           | 22          |
| 7       | 47                   | 7           | 23          |
| 8       | 53                   | 11          | 24          |
| 9       | 23                   | 13          | 16          |
| 10      | 29                   | 17          | 15          |
| 11      | 31                   | 23          | 14          |
| 12      | 37                   | 29          | 13          |
| 13      | 41                   | 11          | 28          |
| 14      | 43                   | 13          | 29          |
| 15      | 47                   | 17          | 30          |
| 16      | 53                   | 5           | 31          |

Студенты самостоятельно выбирают значение первообразный порождающий элемент и исследуют процесс двухстороннего обмена.

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана для двух сторон по своему варианту и ответы на контрольные вопросы.

### **Вопросы для защиты работы**

11. Основные принципы построения экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана.

12. Основные характеристики экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана.

13. Односторонняя генерация ключей с использованием алгоритма Диффи-Хелмана.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная: 1-5*

*Дополнительная: 1-6*

## Лабораторная работа № 13

### РАСШИРЕНИЕ ПОЛЕЙ ГАЛУА GF(2M)

#### Цель и содержание:

1. Углубить знания, полученные на лекциях, по основам ключевого экспоненциального ключевого обмена.
2. Исследовать основные характеристики модифицированного метода экспоненциального ключевого обмена Диффи-Хелмана.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Известно, что метод ключевого обмена Диффи-Хелмана имеет недостаток. Он подвержен атаке противник посередине. Устранить данный недостаток позволяет протокол МТИ (MTI) – (авторы Мацумото, Такашима, Иман) предназначен для реализации метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана и предлагает вместо цифровой подписи использовать модифицированную процедуру выработки общего ключа.

Пусть пользователи имеют секретные ключи, которые выбираются из условия:

$$A \rightarrow a, 1 < a < p-2$$

$$B \rightarrow b, 1 < b < p-2$$

Затем обе стороны публикуют открытые ключи:

$$A \rightarrow Z_A, \quad Z_A = \alpha^a \bmod p$$

$$\hat{A} \rightarrow Z_B, \quad Z_B = \alpha^b \bmod p$$

Для выработки секретного ключа  $K$  обе стороны генерируют случайные числа вида:

$$1 < X_A < p-2$$

$$1 < X_B < p-2$$

Затем вычисляются соответствующие значения, которыми обе стороны обмениваются:

$$Y_A = \alpha^{X_A} \bmod p$$

$$Y_B = \alpha^{X_B} \bmod p$$

После этого стороны вычисляют значения ключа согласно выражений:

$$B: K = (Y_A)^{X_B} \cdot Z_A^b \bmod p = (\alpha^{X_A \cdot X_B}) \cdot \alpha^{ab} \bmod p = \alpha^{X_A \cdot X_B + ab} \bmod p$$

$$A: K = (Y_B)^{X_A} \cdot Z_B^a \bmod p = (\alpha^{X_A \cdot X_B}) \cdot \alpha^{ab} \bmod p = \alpha^{X_A \cdot X_B + ab} \bmod p$$

Рассмотрим процесс исследования метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана, предлагающего вместо цифровой подписи использовать модифицированную процедуру выработки общего ключа МТИ.

1. Выбираем в качестве значений  $p=7$ ,  $\alpha=3$ .
2. Выбираем секретные ключи  $A: a=2$ ;  $B: b=6$ .
3. Вычисляем открытые ключи

$$A \rightarrow Z_A = 3^2 \bmod 7 = 2;$$

$$B \rightarrow Z_B = 3^6 \bmod 7 = 1.$$

Полученные открытые ключи печатаются.

4. Выбираем случайные числа для шифрования и осуществляем процесс вычисления ключей, которыми будут обмениваться две стороны.

$$A \rightarrow X_A = 5; \quad Y_A = \alpha^5 \bmod p = 3^5 \bmod 7 = 5;$$

$$B \rightarrow X_B = 4; \quad Y_B = \alpha^4 \bmod p = 3^4 \bmod 7 = 4.$$

5. Осуществляем вычисление ключа для шифрования

$$A \rightarrow K = (Y_B)^{X_A} Z_B^a \bmod 7 = 4^5 \cdot 1^2 \bmod 7 = 2;$$

$$B \rightarrow K = (Y_A)^{X_B} Z_A^b \bmod 7 = 5^4 \cdot 2^6 \bmod 7 = 2.$$

Полученные значения ключей на стороне  $A$  и  $B$  совпадают

### **Аппаратура и материалы**

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.
2. Операционная система Windows 2000\XP.

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.

2. Провести исследование метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана для двух сторон.

Кодирующее число  $g$  числа  $X_A$  и  $X_B$  представлены в таблице 1.

Таблица 1 – Задание для исследования метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана для двух сторон.

| Вариант | Кодирующее число $g$ | Число $X_A$ | Число $X_B$ |
|---------|----------------------|-------------|-------------|
| 1       | 23                   | 7           | 17          |
| 2       | 29                   | 13          | 18          |
| 3       | 31                   | 11          | 19          |
| 4       | 37                   | 17          | 20          |
| 5       | 41                   | 19          | 21          |
| 6       | 43                   | 5           | 22          |
| 7       | 47                   | 7           | 23          |
| 8       | 53                   | 11          | 24          |
| 9       | 23                   | 13          | 16          |
| 10      | 29                   | 17          | 15          |
| 11      | 31                   | 23          | 14          |
| 12      | 37                   | 29          | 13          |
| 13      | 41                   | 11          | 28          |
| 14      | 43                   | 13          | 29          |
| 15      | 47                   | 17          | 30          |
| 16      | 53                   | 5           | 31          |

Студенты самостоятельно выбирают значение первообразный порождающий элемент и исследуют процесс двухстороннего обмена.

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования метода экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана для двух сторон по своему варианту и ответы на контрольные вопросы.

### **Вопросы для защиты работы**

1. Основные принципы построения экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана.

2. Основные характеристики экспоненциального ключевого обмена на основе алгоритма Диффи-Хелмана.

3. Односторонняя генерация ключей с использованием алгоритма Диффи-

Хелмана.

**Список литературы, рекомендуемый к использованию по данной теме.**

*Основная: 1-5*

*Дополнительная: 1-6*

## Лабораторная работа №14

### МАТЕМАТИЧЕСКИЕ МОДЕЛИ РАСШИРЕННЫХ ПОЛЕЙ ГАЛУА

#### Цель и содержание:

1. Углубить знания, полученные на лекциях, по основам вычисления секретного ключа.
2. Исследовать основные принципы получения распределенного секретного ключа на основе схемы Шамира.

#### Формируемые компетенции ОПК-6, ОПК-9

#### Теоретическое обоснование

Идея состоит в разделении секретного ключа на компоненты с последующим их распределением среди легальных пользователей. Восстановление ключа возможно только в том случае, если коалиция легальных пользователей будет содержать определенное число участников, при этом по одной похищенной части нельзя восстановить весь ключ.

Схема Шамира использует полиномиальное уравнение в ключевом поле. Выбирается большое простое число  $p$ , которое больше числа долей и больше самого большого секрета. Чтобы сделать секрет общим, генерируется полином степени  $m-1$ . Чтобы сделать  $(3,n)$  пороговую схему для восстановления сообщения  $M$  требуется три доли, при этом генерируется квадратичный полином

$$(ax^2 + bx + M) \bmod p$$

Коэффициенты  $a$  и  $b$  выбираются случайным образом и хранятся в тайне,  $M$ -сообщение. Простое число  $p$  - открыто публикуется. Доли получаются с помощью вычисления многочлена в  $n$  различных точках  $k_i = F(x_i)$

Первой долей, может быть значение многочлена при  $x=1$ , второй долей – значение многочлена при  $x=2$ , и.т.д.

Поскольку в квадратичных многочленах имеется три неизвестных коэффициента  $a$ ,  $b$ ,  $M$ , для создания трех уравнений используют любые три доли.

Рассмотрим исследование процесса вычисления секретного ключа на основе схемы Шамира. Определить секрет  $M$ . Пусть необходимо создать схему

(3,5)-пороговую схему в которой три человека из пяти могут восстановить секрет – значение  $M$ . Для этого получим квадратное уравнение с числами  $a=7$  и  $b=8$ . Положим, что  $M=11$ . В качестве  $p$  выбираем число 13. Тогда имеем:

$$F(x) = (7x^2 + 8x + 11) \bmod 13$$

Пятью долями являются:

$$k_1 = F(1) = 7 + 8 + 11 \equiv 0 \bmod 13$$

$$k_2 = F(2) = 28 + 16 + 11 \equiv 3 \bmod 13$$

$$k_3 = F(3) \equiv 7 \bmod 13$$

$$k_4 = F(4) \equiv 12 \bmod 13$$

$$k_5 = F(5) \equiv 5 \bmod 13$$

Чтобы восстановить секрет  $M$  воспользуемся тремя долями  $k_2, k_3, k_5$ .

$$(a \cdot 2^2 + b \cdot 2 + M) \equiv 3 \bmod 13$$

$$(a \cdot 3^2 + b \cdot 3 + M) \equiv 7 \bmod 13$$

$$(a \cdot 5^2 + b \cdot 5 + M) \equiv 5 \bmod 13$$

Для определения секрета, а он представляет собой точку пересечения с осью  $Y$  воспользуемся интерполяционными полиномами Лагранжа.

Коэффициенты интерполяционных полиномов Лагранжа применяется для интерполяции функции  $y(x)$ , заданной рядом ординатой  $y_0 - y_n$  при абсциссах  $x_0 - x_n$ . В качестве нулевого узла можно использовать любой узел, например, при 3 узлах и нулевом центральном узле значения функций обозначаются  $y_{-1} = y(x_{-1})$ ,  $y_0 = y(x_0)$ ,  $y_1 = y(x_1)$ . Полином Лагранжа может быть записан в виде:

$$y(x) = A_0(x)y_0 + A_1(x)y_1 + \dots + A_n(x)y_n = \sum_{m=0}^n A_m(x)y_m$$

где коэффициенты определяются из формулы:

$$A_m(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{m-1})(x - x_{m+1}) \dots (x - x_n)}{(x_m - x_0)(x_m - x_1) \dots (x_m - x_{m-1})(x_m - x_{m+1}) \dots (x_m - x_n)}$$

Ввиду сложности последней формулы коэффициенты сложно вычислить при  $n+1$  ординатах. При нормировке

$$x = x_0 + ph$$

где  $h$  - разность абсцисс соседних узлов

$$p = (x - x_0) / h$$

и будет вычисляться значение нормированных коэффициентов Лагранжа:

|         |          |
|---------|----------|
| $X_0=1$ | $Y_0=0$  |
| $X_1=2$ | $Y_1=3$  |
| $X_2=3$ | $Y_2=7$  |
| $X_3=4$ | $Y_3=12$ |
| $X_4=5$ | $Y_4=5$  |

Следовательно, имеем:

|                         |
|-------------------------|
| $X_1=2$ , тогда $A_0=3$ |
| $X_2=3$ , тогда $A_1=7$ |
| $X_4=5$ , тогда $A_2=5$ |

$$y(x) = A_0(x)y_0 + A_1(x)y_1 + A_2(x)y_2$$

$$y(x) = A_0(x) \cdot 3 + A_1(x) \cdot 7 + A_2(x) \cdot 5$$

$$A_0(x) = \frac{(x - x_2) \cdot (x - x_4)}{(x_1 - x_2) \cdot (x_1 - x_4)} = \frac{x^2 - x \cdot (x_2 + x_4) + x_2 \cdot x_4}{(2 - 3) \cdot (2 - 5)} =$$

$$= \frac{x^2 - x \cdot (3 + 5) + 15}{(-1) \cdot (-3)} = \frac{x^2 - 8x + 15}{3}$$

$$A_1(x) = \frac{(x - x_1) \cdot (x - x_4)}{(x_2 - x_1) \cdot (x_3 - x_1)} = \frac{x^2 - x(x_1 + x_4) + x_1 \cdot x_4}{(3 - 2) \cdot (3 - 5)} = -\frac{x^2 - 7x + 10}{2}$$

$$A_2(x) = \frac{(x - x_1) \cdot (x - x_2)}{(x_4 - x_1) \cdot (x_4 - x_2)} = \frac{x^2 - x(x_1 + x_2) + x_1 \cdot x_2}{(5 - 2) \cdot (5 - 3)} = \frac{x^2 - 5x + 6}{6}$$

$$y(x) = \frac{x^2 - 8x + 15}{3} \cdot y_0 - \frac{x^2 - 7x + 10}{2} \cdot y_1 + \frac{x^2 - 5x + 6}{6} \cdot y_2$$

Найдём значение в точке  $x = 0$ . Это и есть секрет значения функции в нулевой точке  $F(0)=11$ . Тогда

$$y(0) = \frac{15 \cdot 3}{3} - \frac{10 \cdot 7}{2} + \frac{6}{6} \cdot 5 \bmod 13 = 15 - 35 + 5 \bmod 13 = 11 \bmod 13$$

Исходный полином

$$F(x) = 7x^2 + 8x + 11 \bmod 13$$

Полученное значение секрета  $M=11$ .

### Аппаратура и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 2000\XP.

2. Операционная система Windows 2000\XP.

### **Методика и порядок выполнения работы**

1. Изучить теоретический материал работы.

2. Провести исследование вычисления секретного ключа на основе использования схемы Шамира, используя (3,5)-пороговую схему в которой три человека из пяти могут восстановить секрет – значение  $M$  системы RSA.

Кодирующее число  $p$  числа  $a$ ,  $b$  и секрет  $M$  представлены в таблице 1.

Таблица 1 – Задание для исследования.

| Вариант | Кодирующее<br>число $p$ | Число $a$ | Число $b$ | Секрет $M$ |
|---------|-------------------------|-----------|-----------|------------|
| 1       | 23                      | 7         | 12        | 17         |
| 2       | 29                      | 13        | 14        | 18         |
| 3       | 31                      | 11        | 21        | 19         |
| 4       | 37                      | 17        | 23        | 20         |
| 5       | 41                      | 19        | 11        | 21         |
| 6       | 43                      | 5         | 12        | 22         |
| 7       | 47                      | 7         | 13        | 23         |
| 8       | 29                      | 11        | 14        | 24         |
| 9       | 23                      | 13        | 15        | 16         |
| 10      | 29                      | 17        | 16        | 15         |
| 11      | 31                      | 23        | 17        | 14         |
| 12      | 37                      | 29        | 18        | 13         |
| 13      | 41                      | 11        | 17        | 28         |
| 14      | 43                      | 13        | 16        | 29         |
| 15      | 29                      | 17        | 15        | 14         |
| 16      | 23                      | 5         | 14        | 21         |

Студенты самостоятельно выбирают значение 3 пользователей для определения секретного ключа на основе интерполяционных полиномов Лагранжа.

### **Содержание отчета и его форма**

Отчет по лабораторной работе, оформленный письменно в рабочей тетради, должен содержать процесс исследования вычисления секретного ключа на основе использования схемы Шамира по своему варианту и ответы на

контрольные вопросы.

### **Вопросы для защиты работы**

1. Назначение и особенности методов вычисления секретного ключа.
2. Основные характеристики метода Шамира.
3. Применение полиномов Лагранжа при вычислении секретного ключа.

### **Список литературы, рекомендуемый к использованию по данной теме.**

*Основная:* 1-5

*Дополнительная:* 1-6

### **Указания по технике безопасности**

1. Перед началом работы пользователь ПК обязан проверить, чтобы все кабели питания находились как можно дальше в компактном положении с тыльной стороны рабочего места.
2. Компьютер рекомендуется подключать к отдельной розетке. Розетка, используемая для подключения компьютер должна быть трехполюсной.
3. Запрещается приступать к работе при: а) обнаружении неисправности оборудования; б) отсутствии защитного заземления устройств.
4. Пользователю ПК во время работы запрещается:
  - а) касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала);
  - б) прикасаться к задней панели системного блока;
  - в) производить переключения интерфейсных кабелей периферийных устройств при включенном питании;
  - г) производить отключение питания во время выполнения задачи.

5. Категорически запрещается работать с ПК при снятом корпусе; оставлять включенный ПК без присмотра; самостоятельно вскрывать корпус ПК.

## ЛИТЕРАТУРА

### 1. Основная литература:

- Рябко Б.Я. Фионов А.Н. Криптографические методы защиты информации. – М.: "Горячая линия-Телеком", 2012. - 229 с.
- Музыкантский А.И., Фурин В.В. Лекции по криптографии. – М.: МЦНМО, 2011. - 68 с.
- Глухов М. М. Круглов И. А., Пичкур А. Б., Черемушкин А. В. Введение в теоретико-числовые методы криптографии. – СПб.: "Лань", 2011. - 400 стр.
- Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.: ДиК, 2008. – 448 с.
- Основы криптографии: Учебное пособие/Под ред. Алферова П.П. – М.: Гелиос, 2008. – 480 с.

### 2. Дополнительная литература:

- Торстейнсон П., Ганеш Г.А. Криптография и безопасность в технологии .NET. – М.: "Бином. Лаборатория знаний", 2013. – 480 с.
- Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.: "ДМК Пресс", 2008. – 448 с.
- Алферов А.П., Зубов А.Ю. и др. Основы криптографии: Учебное пособие, 2-е – М.: Гелиос АРВ, 2006 – 480 с.
- Баричев С.Г. Основы современной криптографии. Учебный курс. – М.: Телеком, 2007. – 129 с.
- Молдовян А.А. и др. Криптография. – СПб.: Лань, 2007. – 224 с.
- Введение в криптографию/Под ред В.В.Ященко. – М.: МЦНМО, 2006. – 288 с.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для обучающихся по организации и проведению самостоятельной работы  
по дисциплине «Методы и средства криптографической защиты информации»

для студентов направления подготовки  
10.03.01 «Информационная безопасность»

Направленность (профиль):

«Организация и технологии защиты информации (по отрасли или в сфере профессиональной  
деятельности)»



## **СОДЕРЖАНИЕ**

### **Введение**

1. Общая характеристика самостоятельной работы студента
2. Технологическая карта самостоятельной работы студента
3. Самостоятельное изучение тем лекций
4. Паспорт фонда оценочных средств для проверки самостоятельной работы
5. Список рекомендуемой литературы

## *Введение*

Целью дисциплины «Методы и средства криптографической защиты информации» является рассмотрение теоретических основ и закономерностей построения и функционирования систем, в том числе и организационных, методологических принципов их анализа и синтеза, применение изученных закономерностей для формирования набора общекультурных и профессиональных компетенций будущего бакалавра по направлению подготовки 10.03.01 «Информационная безопасность».

Основными задачами изучения дисциплины являются:

1. Изучение основных естественнонаучных законов для адекватного применения математического аппарата и выявления сущности проблем, возникающих в сфере защиты информации и информационной безопасности.
2. Формирование способностей применять методы теории систем и системного анализа при проведении предварительного технико-экономического анализа и обоснования проектных решений по обеспечению информационной безопасности.
3. Формирование способностей самостоятельно проводить эксперименты по заданной методике, обрабатывать результаты, оценивать погрешность и достоверность результатов.

### **1. Общая характеристика самостоятельной работы студента**

Основными видами учебной работы по достижению результатов освоения дисциплины являются лекции, лабораторные работы и самостоятельная работа студентов (СРС).

На лекциях раскрываются основные положения и понятия курса, формируются знания в области математических основ теории систем. На лабораторных работах формируются умения и навыки, необходимые для решения задач профессиональной деятельности.

Приступая к изучению учебной дисциплины, необходимо ознакомиться с учебной программой, получить в библиотеке рекомендованные учебные пособия, а также получить у ведущего преподавателя в электронном виде конспекты лекций, методические рекомендации к лабораторным работам и тестовые, завести новую тетрадь для конспектирования лекций и выполнения лабораторных работ.

Для изучения дисциплины предлагается список основной и дополнительной литературы. Основная литература предназначена для обязательного изучения, дополнительная – поможет более глубоко освоить отдельные вопросы, подготовить исследовательские задания и выполнить задания для самостоятельной работы.

В ходе лекционных занятий студент обязан осуществлять конспектирование учебного материала, особое внимание, обращая на категории, формулировки, раскрывающие содержание тех или иных понятий, физических явлений, процессов, эффектов. В рабочих конспектах желательно оставлять поля, на которых следует делать пометки из рекомендованной литературы, дополнять материал прослушанной лекции, формулировать научные выводы и практические рекомендации. Студент имеет право задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Самостоятельная работа студентов над материалом учебной дисциплины является неотъемлемой частью учебного процесса и должна предполагать углубление знания учебного материала, излагаемого на аудиторных занятиях, и приобретение дополнительных знаний по отдельным вопросам самостоятельно.

Основными видами самостоятельной работы студентов по учебной дисциплине являются:

- самостоятельное изучение учебного материала по заданным темам;
- подготовка к лабораторным работам, оформление отчета по выполненному лабораторному занятию и подготовка к собеседованию по результатам работы.

Основными методами самостоятельной работы студентов являются:

- 1) для овладения знаниями:

- чтение текста (конспекта лекций, учебника, дополнительной литературы) и конспектирование текста;
- работа с нормативными документами;
- поиск информации в Интернет;
- 2) для закрепления и систематизации знаний:
  - работа с конспектом лекции (обработка текста);
  - повторная работа над учебным материалом (учебника, дополнительной литературы, слайдами);
  - составление плана и тезисов ответа или графическое изображение структуры ответа;
  - составление таблиц для систематизации учебного материала;
  - изучение нормативных документов;
  - ответы на контрольные вопросы;
- 3) для формирования умений:
  - подготовка к лабораторным занятиям;
  - решение задач и лабораторных заданий;
  - подготовка отчетов по лабораторным занятиям;
  - рефлексивный анализ профессиональных умений.

В случае пропуска учебного занятия студент может воспользоваться содержанием различных блоков учебно-методического комплекса (лекции, лабораторные занятия, контрольные вопросы и тесты) для самоподготовки и освоения темы. Для самоконтроля необходимо использовать вопросы и задания, предлагаемые к лабораторным занятиям, а также варианты тестовых заданий.

## 2. Технологическая карта самостоятельной работы студента

| Коды реализуемых компетенций, индикатора(ов)         | Вид деятельности студентов          | Средства и технологии оценки | Объем часов, в том числе |                                    |       |
|------------------------------------------------------|-------------------------------------|------------------------------|--------------------------|------------------------------------|-------|
|                                                      |                                     |                              | СРС                      | Контактная работа с преподавателем | Всего |
| 5 семестр                                            |                                     |                              |                          |                                    |       |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к лабораторной работе    | Собеседование                | 14.00                    | 2.00                               | 16.00 |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к лекции                 | Конспект                     | 12.00                    | 2.00                               | 14.00 |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка доклада                  | Доклад                       | 12.00                    | 2.00                               | 14.00 |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Самостоятельное изучение литературы | Конспект                     | 12.00                    | 2.00                               | 14.00 |

|                                                      |                                           |                                  |       |       |       |
|------------------------------------------------------|-------------------------------------------|----------------------------------|-------|-------|-------|
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к<br>практическому<br>занятию  | Конспект.<br>Письменный<br>отчет | 12.00 | 2.00  | 14.00 |
| <b>Итого за 5 семестр</b>                            |                                           |                                  | 62.00 | 10.00 | 72.00 |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к<br>практическому<br>занятию  | Конспект.<br>Письменный<br>отчет | 1.00  | 1.00  | 2.00  |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к<br>лабораторной<br>работе    | Собеседование                    | 1.00  | 1.00  | 2.00  |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к<br>лекции                    | Конспект                         | 1.00  | 1.00  | 2.00  |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка к<br>экзамену                  | Вопросы к<br>экзамену            | 1.00  | 1.00  | 2.00  |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Самостоятельное<br>изучение<br>литературы | Конспект                         | 1.00  | 1.00  | 2.00  |
| ОПК-6 ИД-2<br>ОПК-9 ИД-1<br>ОПК-9 ИД-2<br>ОПК-9 ИД-3 | Подготовка<br>доклада                     | Доклад                           | 1.00  | 1.00  | 2.00  |
| <b>Итого за 6 семестр</b>                            |                                           |                                  | 6.00  | 6.00  | 12.00 |
| <b>Итого</b>                                         |                                           |                                  | 68.00 | 16.00 | 84.00 |

### 3. Порядок выполнения самостоятельной работы

#### 3.1 Методические рекомендации по изучению теоретического материала

Комплексное изучение предлагаемой студентам учебной дисциплины «Методы и средства криптографической защиты информации» предполагает овладение материалами лекций, учебников, программы, творческую работу студентов в ходе проведения лабораторных работ, а также систематическое выполнение заданий для самостоятельной работы.

В ходе лекций раскрываются основные вопросы в рамках рассматриваемых тем, делаются акценты на наиболее сложные и интересные положения изучаемого материала, которые должны быть приняты студентами во внимание. Материалы лекций являются основой для подготовки студентов к лабораторным занятиям.

#### ***Работа с конспектом лекций***

Просмотрите конспект сразу после занятий. Отметьте материал конспекта лекций, который вызывает затруднения для понимания. Попытайтесь найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю.

Каждую неделю отводите время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам и тестам.

#### ***Выполнение лабораторных работ по дисциплине «Методы и средства криптографической защиты информации»***

На первом занятии получите у преподавателя задания по курсу, планы подготовки к лабораторным работам. Обзаведитесь всем необходимым методическим обеспечением.

Перед лабораторными работами изучите теорию вопроса, предполагаемого к исследованию, ознакомьтесь с опытом других исследователей в этом направлении.

Целью лабораторных работ является:

- закрепление методов приложения теории к решению практических задач анализа и синтеза психологического знания;
- проверка уровня понимания студентами вопросов, рассмотренных на лекциях и по учебной литературе, степени и качества усвоения материала;
- обучение навыкам освоения методики эксперимента и работы с нормативно-справочной литературой;
- восполнение пробелов в пройденной теоретической части курса и оказание помощи в его усвоении.

### **3.2 Методические рекомендации по написанию и оформлению доклада**

#### **1. Общие положения**

1.1. Доклад, как вид самостоятельной работы в учебном процессе, способствует формированию навыков исследовательской работы, расширяет познавательные интересы, учит критически мыслить.

1.2. При написании доклада по заданной теме студент составляет план, подбирает основные источники.

1.3. В процессе работы с источниками систематизирует полученные сведения, делает выводы и обобщения.

1.4. К докладу по крупной теме могут привлекать несколько студентов, между которыми распределяются вопросы выступления.

#### **2. Выбор темы доклада**

2.1. Тематика доклада обычно определяется преподавателем, но в определении темы инициативу может проявить и студент.

2.2. Прежде чем выбрать тему доклада, автору необходимо выявить свой интерес, определить, над какой проблемой он хотел бы поработать, более глубоко ее изучить.

### 3. Этапы работы над докладом

3.1. Формулирование темы, причем она должна быть не только актуальной по своему значению, но и оригинальной, интересной по содержанию.

3.2. Подбор и изучение основных источников по теме (как правильно, при разработке доклада используется не менее 8-10 различных источников).

3.3. Составление списка использованных источников.

3.4. Обработка и систематизация информации

3.5. Разработка плана доклада.

3.6. Написание доклада.

3.7. Публичное выступление с результатами исследования.

### 4. Структура доклада:

- титульный лист

- оглавление (в нем последовательно излагаются названия пунктов доклада, указываются страницы, с которых начинается каждый пункт);

- введение (формулирует суть исследуемой проблемы, обосновывается выбор темы, определяются ее значимость и актуальность, указываются цель и задачи доклада, дается характеристика используемой литературы);

- основная часть (каждый раздел ее, доказательно раскрывая отдельную проблему или одну из ее сторон, логически является продолжением предыдущего; в основной части могут быть представлены таблицы, графики, схемы);

- заключение (подводятся итоги или дается обобщенный вывод по теме доклада, предлагаются рекомендации);

- список использованных источников

### 5. Структура и содержание доклада

5.1. Введение - это вступительная часть научно-исследовательской работы. Автор должен приложить все усилия, чтобы в этом небольшом по объему разделе показать актуальность темы, раскрыть практическую значимость ее, определить цели и задачи эксперимента или его фрагмента.

5.2. Основная часть. В ней раскрывается содержание доклада.

Как правило, основная часть состоит из теоретического и практического разделов. В теоретическом разделе раскрываются история и теория исследуемой проблемы, дается критический анализ литературы и показываются позиции автора. В практическом разделе

излагаются методы, ход, и результаты самостоятельно проведенного эксперимента или фрагмента.

В основной части могут быть также представлены схемы, диаграммы, таблицы, рисунки и т.д.

5.3. В заключении содержатся итоги работы, выводы, к которым пришел автор, и рекомендации. Заключение должно быть кратким, обязательным и соответствовать поставленным задачам.

5.4. Список использованных источников представляет собой перечень использованных книг, статей, фамилии авторов приводятся в алфавитном порядке, при этом все источники даются под общей нумерацией литературы. В исходных данных источника указываются фамилия и инициалы автора, название работы, место и год издания.

5.5. Приложение к докладу оформляются на отдельных листах, причем каждое должно иметь свой тематический заголовок и номер, который пишется в правом верхнем углу, например: «Приложение 1»

## 6. Требования к оформлению доклада

6.1. Объем доклада может колебаться в пределах 5-15 печатных страниц; все приложения к работе не входят в ее объем.

6.2. Доклад должен быть выполнен грамотно, с соблюдением культуры изложения.

6.3. Обязательно должны иметься ссылки на используемую литературу.

6.4. Должна быть соблюдена последовательность написания библиографического аппарата.

## 7. Критерии оценки доклада

- актуальность темы исследования;
- соответствие содержания теме;
- глубина проработки материала; правильность и полнота использования источников;
- соответствие оформления доклада стандартам.

По усмотрению преподавателя доклады могут быть представлены на семинарах, научно-практических конференциях, а также использоваться как зачетные работы по пройденным темам.

## **3.3 Методические рекомендации по конспектированию первоисточников**

Изучение и анализ литературных источников является обязательным видом самостоятельной работы студентов. Изучение литературы по избранной теме имеет своей задачей проследить характер постановки и решения определенной проблемы различными

авторами, аргументацию их выводов и обобщений, провести анализ и систематизировать полученный материал на основе собственного осмысления с целью выяснения современного состояния вопроса.

Проработка отобранного материала обязательно должна идти с одновременным ведением записей прочитанного и своих замечаний. Запись может иметь как форму конспекта, так и выписок, а также картотеку положений, тезисов, идей, методик, что в дальнейшем облегчит классификацию и систематизацию полученного материала. Такого рода записи являются лучшим способом накопления и первичной обработки материала, одной из обязательных форм организации умственного труда.

**Планом** удобно пользоваться при подготовке к устному выступлению по выбранной теме. Каждый пункт плана должен раскрывать одну из сторон избранной темы, а весь план должен охватывать ее целиком.

**Тезисы** предполагают сжатое изложение основных положений текста в форме утверждения или отрицания. Они являются более совершенной формой записей и представляют основу для дискуссии. К тому же их легко запомнить.

**Конспектирование** – процесс мысленной переработки и письменной фиксации информации, в виде краткого изложения основного содержания, смысла какого-либо текста.

Результат конспектирования – запись, позволяющая конспектирующему немедленно или через некоторый срок с нужной полнотой восстановить полученную информацию. Конспект в переводе с латыни означает «обзор». По существу его и составлять надо как обзор, содержащий основные мысли текста без подробностей и второстепенных деталей. Конспект носит индивидуализированный характер: он рассчитан на самого автора и поэтому может оказаться малопонятным для других.

Для того чтобы осуществлять этот вид работы, в каждом конкретном случае необходимо грамотно решить следующие задачи:

1. Сориентироваться в общей композиции текста (уметь определить вступление, основную часть, заключение).
2. Увидеть логико-смысловую канву сообщения, понять систему изложения автором информации в целом, а также ход развития каждой отдельной мысли.
3. Выявить «ключевые» мысли, т. е. основные смысловые вехи, на которые «нанизано» все содержание текста.
4. Определить детализирующую информацию.
5. Лаконично сформулировать основную информацию, не перенося на письмо все целиком и дословно.

Во всяком научном тексте содержится информация 2-х видов: основная и вспомогательная. Основной является информация, имеющая наиболее существенное значение для раскрытия содержания темы или вопроса. К ней относятся: определения научных понятий, формулировки законов, теоретических принципов и т. д. Назначение вспомогательной информации – помочь читателю лучше усвоить предлагаемый материал. К этому типу информации относятся разного рода комментарии. Основную – записываем как можно полнее, вспомогательную, как правило, опускаем. Содержание конспектирования составляет переработка основной информации в целях ее обобщения и сокращения. Обобщить – значит представить ее в более общей, схематической форме, в виде тезисов, выводов, отдельных заголовков, изложения основных результатов и т. п. Читая, мы интуитивно используем некоторые слова и фразы в качестве опорных. Такие опорные слова и фразы называются ключевыми. Ключевые слова и фразы несут основную смысловую и эмоциональную нагрузку содержания текста.

Выбор ключевых слов – это первый этап смыслового свертывания, смыслового сжатия материала.

Важными требованиями к конспекту являются наглядность и обозримость записей и такое их расположение, которое давало бы возможность уяснить логические связи и иерархию понятий.

**По форме конспекты подразделяются на формализованные и графические.**

**1. Формализованные** (все записи вносятся в заранее подготовленные таблицы).

Это удобно, во-первых, при конспектировании материалов, когда перечень характеристик описываемых предметов или явлений более или менее постоянен, во-вторых, при подготовке единого конспекта по нескольким источникам. Особенно если есть необходимость сравнения отдельных данных. Разновидностью формализованного конспекта является запись, составленная в форме ответов на заранее подготовленные вопросы, обеспечивающие исчерпывающие характеристики однотипных предметов или явлений.

**2. Графические** (элементы конспектируемой работы располагаются в таком виде, при котором видна иерархия понятий и взаимосвязь между ними).

По каждой работе может быть не один, а несколько графических конспектов, отображающих книгу в целом и отдельные ее части. Ведение графического конспекта – наиболее совершенный способ изображения внутренней структуры книги, а сам этот процесс помогает усвоению ее содержания.

Можно выделить следующие основные **типы конспектов: плановый, текстуальный, сводный, тематический.**

**Плановый** – легко получить с помощью предварительно сделанного плана произведения, каждому вопросу плана отвечает определенная часть конспекта:

а) вопросно-ответный (на пункты плана, выраженные в вопросительной форме, конспект дает точные ответы);

б) схематичный плановый конспект (отражает логическую структуру и взаимосвязь отдельных положений).

**Текстуальный** – это конспект, созданный в основном из цитат.

**Сводный конспект** – сочетает выписки, цитаты, иногда тезисы; часть его текста может быть снабжена планом.

**Тематический** – дает более или менее исчерпывающий ответ (в зависимости из числа привлеченных источников и другого материала, например, своих же записей) на поставленный вопрос – тему: обзорный; хронологический.

Роль конспекта – чисто учебная: он помогает зафиксировать основные понятия и положения первичного текста и в нужный момент их воспроизвести, например, при написании реферата или подготовке к экзамену.

**Способы конспектирования.**

- **Тезисы** – это кратко сформулированные основные мысли, положения изучаемого материала. Тезисы лаконично выражают суть читаемого, дают возможность раскрыть содержание. Приступая к освоению записи в виде тезисов, полезно в самом тексте отмечать места, наиболее четко формулирующие основную мысль, которую автор доказывает (если, конечно, это не библиотечная книга). Часто такой отбор облегчается шрифтовым выделением, сделанным в самом тексте.

- **Линейно-последовательная запись текста.** При конспектировании линейно – последовательным способом целесообразно использование плакатно-оформительских средств, которые включают в себя следующие:

- сдвиг текста конспекта по горизонтали, по вертикали;
- выделение жирным (или другим) шрифтом особо значимых слов;
- использование различных цветов;
- подчеркивание;
- заключение в рамку главной информации.

- **Способ «вопросов – ответов».** Он заключается в том, что, поделив страницу тетради пополам вертикальной чертой, конспектирующий в левой части страницы самостоятельно формулирует вопросы или проблемы, затронутые в данном тексте, а в правой части дает ответы на них.

Одна из модификаций способа «вопросов – ответов» - таблица, где место вопроса занимает формулировка проблемы, поднятой автором (лектором), а место ответа – решение данной проблемы. Иногда в таблице могут появиться и дополнительные графы: например, «мое мнение» и т. п.

- **Схема с фрагментами** – способ конспектирования, позволяющий ярче выявить структуру текста, - при этом фрагменты текста (опорные слова, словосочетания, пояснения всякого рода) в сочетании с графикой помогают созданию рационально - лаконичного конспекта.

- **Простая схема** – способ конспектирования, близкий к схеме с фрагментами, объяснений к которой конспектирующий не пишет, но должен уметь давать их устно. Этот способ требует высокой квалификации конспектирующего. В противном случае такой конспект нельзя будет использовать.

- **Параллельный способ** конспектирования. Конспект оформляется на двух листах параллельно или один лист делится вертикальной чертой пополам и записи делаются в правой и в левой части листа.

**Однако** лучше использовать разные способы конспектирования для записи одного и того же материала.

**Комбинированный конспект** – вершина овладения рациональным конспектированием. При этом умело используются все перечисленные способы, сочетая их в одном конспекте (один из видов конспекта свободно перетекает в другой в зависимости от конспектируемого текста, от желания и умения конспектирующего). Именно при комбинированном конспекте более всего проявляется уровень подготовки и индивидуальность студента.

### **Принципы составления конспекта прочитанного.**

1. Записать все выходные данные источника: автор, название, год и место издания. Если текст взят из периодического издания (газеты или журнала), то записать его название, год, месяц, номер, число, место издания.

2. Выделить поля слева или справа, можно с обеих сторон. Слева на полях отмечаются страницы оригинала, структурные разделы статьи или книги (названия параграфов, подзаголовки и т. п.), формулируются основные проблемы. Справа – способы фиксации прочитанной информации.

Один из видов чтения – углубленное – предполагает глубокое усвоение прочитанного и часто сохранение информации в целях последующего обращения к ней. Эффективность такого чтения повышается, если прочитанное зафиксировано не только в памяти, но и на бумаге. Психологи утверждают, что записанное лучше и полнее усваивается, прочнее откладывается в памяти. Установлено, что если прочитать 1000 слов и затем записать 50, подытоживающих прочитанное, то коэффициент усвоения будет выше, чем, если прочитать 10000 слов, не записав ни одного. Кроме того, при записи прочитанного формируется навык свертывания информации. И, наконец, чередование чтения и записывания уменьшает усталость, повышает работоспособность и производительность умственного труда.

#### *1. Резюмирование.*

Резюме – краткий итог прочитанного, содержащий его оценку. Резюме характеризует основные выводы книги, главные итоги.

Выбор языковых средств для построения резюме-выводов подчинен основной задаче свертывания информации: минимум языковых средств – максимум информации. Это обычно одно – три четких, кратких, выразительных предложения, раскрывающих, по мнению автора, самую суть описываемого объекта.

## *2. Аннотация.*

Аннотация – краткая обобщенная характеристика печатной работы (книги, статьи), включающая иногда и его оценку. Это наикратчайшее изложение содержания первичного документа, дающее общее представление о теме.

Основное ее назначение – дать некоторое представление о книге (статье, научной работе) с тем, чтобы рекомендовать ее определенному кругу читателей или воспользоваться своими записями при выполнении работы исследовательского, реферативного характера. Поэтому аннотации не требуются изложения содержания произведения, в ней лишь перечисляются вопросы, которые освещены в первоисточнике (содержание этих вопросов не раскрывается). Аннотация отвечает на вопрос: «О чем говорится в первичном тексте?», дает представление только о главной теме и перечне вопросов, затрагиваемых в тексте первоисточника.

Текст аннотации не стандартизирован. В научной литературе можно встретить различные требования к составлению аннотаций. Например, текст справочной аннотации может включать следующие сведения:

- тип и название аннотируемого документа (монография, диссертация, сборник, статья и т. п.)
- задачи, поставленные автором аннотируемого документа
- метод, которым пользовался автор (эксперимент, сравнительный анализ, компиляция других источников)
- принадлежность автора к определенной научной школе или направлению
- структуру аннотируемого документа
- предмет и тему произведения, основные положения и выводы автора
- характеристику вспомогательных иллюстративных материалов, дополнений, приложений, справочного аппарата, включая указатели и библиографию.

## *3.4 Методические указания по подготовке к экзамену*

Контроль и оценка знаний студентов является неотъемлемой составной частью учебно-воспитательного процесса в вузе. Экзаменационная сессия – самая ответственная пора в студенческой жизни: она является итогом большой работы студентов в семестре. Экзамен – это метод проверки знаний студентов по части или полному курсу учебной дисциплины, произведенный путем постановки устных или письменных вопросов. Он дает объективную официально фиксируемую оценку успехов студентов за определенный отрезок времени.

Экзамен преследует многогранную цель: во-первых, это – проверка знаний студента, во-вторых, они сами по себе являются важным звеном в овладении наукой, в-третьих, это продолжение учебного процесса; наконец, он имеет большое значение как фактор стимулирования глубокого изучения предмета.

Подготовка к экзамену состоит из двух взаимосвязанных этапов. Первый – систематический труд на протяжении семестра, учебного года, охватывающий все формы

учебного процесса: лекции, изучение и конспектирование рекомендованной литературы, активное участие в семинарских (практических) и лабораторных занятиях.

Второй – подготовка непосредственно перед экзаменом. Она позволяет студентам за сравнительно короткий отрезок времени охватить всю перспективу изученного и лучше понять основные закономерности и явления.

Ограниченность времени (3-4 дня) для непосредственной подготовки к экзамену по предмету требует от студентов спокойно, без нервной суеты и спешки еще раз внимательно продумать изученный в течение семестра материал, тщательно отработать вопросы, недостаточно изученные или плохо понятые, с тем, чтобы по возможности устранить все пробелы в своих знаниях.

Готовиться надо по строго продуманному графику, последовательно переходя от темы к теме, не пропуская ни одну из них.

Сложные вопросы, недостаточно уясненные в процессе подготовки к экзамену, необходимо записать и получить на них разъяснения у преподавателей во время предэкзаменационных консультаций.

Следует заметить, что студенты, которые не ходят на консультации из-за отсутствия, по их мнению, сложных вопросов, поступают опрометчиво. На консультациях очень часто лектор не только отвечает на заданные вопросы, но и по собственной инициативе разъясняет наиболее трудные разделы курса.

Итак, основной задачей подготовки студентов к экзамену является систематизация знаний учебного материала, его творческое осмысливание. Важнейшим учебным пособием на этом этапе работы студента является собственный конспект прослушанных лекций и самостоятельно проработанных тем курса.

В соответствии с положением об экзаменах их, как правило, принимают преподаватели, читающие курс лекций.

Получив билет, студент должен хорошо продумать содержание поставленных вопросов. Значительное число неудачных ответов объясняется неясным пониманием поставленной проблемы. Правильное понимание вопроса обеспечит успех при ответе на него. При подготовке к ответу на билет нужно составить развернутый план по каждому вопросу

Отвечая на вопросы экзаменационного билета, не следует спешить. Надо излагать материал спокойно, не торопясь, владеть собой, следить за построением фраз. Следует избегать подходов издали, общих рассуждений. Рекомендуются строить ответы четко, последовательно, конкретно, по возможности исчерпывающе. Вместе с тем весьма желательно быстро и правильно иллюстрировать свой ответ примерами.

От экзаменуемого требуется: определение понятий, обоснование выдвинутых положений, свободное оперирование фактическим материалом, дать альтернативные подходы по отдельным проблемам. Логичность, стройность, литературная грамотность изложения являются неотъемлемыми чертами полноценного ответа. Нельзя при ответе допускать ни излишней краткости, переходящей в схематизм, ни многословия. И то, и другое не оправдано. Краткость не дает преподавателю возможности понять, владеет ли студент учебным материалом, а многословие может показать, что студент не умеет акцентировать внимание на главном и говорит слишком расплывчато.

## **Вопросы к экзамену**

### **Вопросы для проверки уровня обученности**

1. Основные понятия теории криптографической защиты информации
2. История криптографии
3. Классификация методов шифрования
4. Шифры подстановки. Криптоанализ шифров подстановки
5. Шифры перестановки. Криптоанализ шифров перестановки
6. Шифр Виженера.

7. Шифр Кардано.
8. Принципы построения блочных шифров
9. Петля Фейстеля
10. Американский стандарт шифрования DES. Основные показатели
11. Алгоритм шифрования DES
12. Ключевая система алгоритма DES
13. Практическое применение американского стандарта шифрования DES. Двойной DES
14. Тройной DES
15. История конкурса на новый стандарт шифрования AES
16. Математические основы построения алгоритма AES
17. Основные элементы реализации шифра AES
18. Преобразования в алгоритме AES
19. Основные параметры отечественного стандарта шифрования Магма
20. Алгоритм шифрования Магма
21. Отечественный стандарт отечественного стандарта шифрования Кузнечик
22. Основные параметры алгоритма Кузнечик
23. Алгоритм шифрования Кузнечик
24. Принципы построения синхронных поточных систем шифрования
25. Схемные реализации синхронных поточных шифраторов. Генератор Фибоначчи.
26. Схемные реализации синхронных поточных шифраторов. Генератор на основе многотактовых кодовых фильтров
27. Принципы построения самосинхронизирующихся поточных систем шифрования
28. Схемные реализации самосинхронизирующихся поточных шифраторов
29. Алгоритм поточного шифрования RC4
30. Алгоритм поточного шифрования A5
31. Режимы использования блочных шифров и их виды.
32. Режим электронной кодовой книги
33. Режим сцепления блоков шифрованного текста
34. Режим шифрования обратной связи (CFB)
35. Режим внешней обратной связи (OFB)
36. Режим счетчика
37. Надежность шифров
38. Энтропия и избыточность языка
39. Расстояние единственности
40. Теоретическая стойкость шифров
41. Практическая стойкость шифров
42. Основные принципы построения асимметричных систем шифрования
43. Алгоритм шифрования без передачи ключа
44. Алгоритм шифрования RSA
45. Алгоритм шифрования Эль-Гамала
46. Электронная цифровая подпись. Назначение. Функции ЭЦП.
47. Основные принципы построения электронной цифровой подписи
48. Цифровые подписи на основе шифрсистем с открытыми ключами
49. Электронная цифровая подпись на основе RSA
50. Электронная цифровая подпись на основе алгоритма Эль-Гамала
51. Управление ключами. Двухсторонние протоколы
52. Управление ключами. Трехсторонние протоколы
53. Метод экспоненциального ключевого обмена на основе алгоритма Диффи-Хеллмана.
54. Алгоритм односторонней генерации ключа Диффи-Хеллмана
55. Модификация алгоритма Диффи-Хеллмана
56. Криптографические схемы разделения секрета. Виды схем разделения секрета.
57. Основные принципы построения схем разделения секрета
58. Схема разделения секрета Шамира
59. Протоколы аутентификации. Назначение. Классификация протоколов аутентификации
60. Парольная аутентификация
61. Аутентификация методом "запрос-ответ"

62. Принципы построения протоколов аутентификация, построенных на доказательстве с нулевым разглашением знаний
63. Протоколы аутентификация, основанные на доказательствах с нулевым разглашением
64. Скрытый канал. Назначение. Решаемые задачи.
65. Построение скрытого канала на основе схемы Онга-Шнорра-Шамирах
66. Построение скрытого канала на основе схемы Эль-Гамала
67. Криптографические хеш-функции. Назначение хеш-функции. Свойства хеш-функции
68. Функции хеширования и целостность данных
69. Бесключевые функции хеширования

#### **Задания для проверки уровня обученности.**

1. Зашифровать открытый текст с помощью таблицы Виженера.
  2. Расшифровать закрытый текст с помощью таблицы Виженера.
  3. Вычислить сеансовые ключи для алгоритма шифрования S-DES.
  4. Зашифровать открытый текст с помощью алгоритма S-DES.
  5. Расшифровать закрытый текст с помощью алгоритма S-DES.
  6. Построить шифратор синхронного поточного шифра на основе генератора Фибоначчи.
- Зашифровать 11 бит открытого текста
7. Построить дешифратор синхронного поточного шифра на основе генератора Фибоначчи. Расшифровать 11 бит открытого текста
  8. Построить шифратор синхронного поточного шифра на основе генератора многотактовых кодовых фильтров. Зашифровать 11 бит открытого текста
  9. Построить дешифратор синхронного поточного шифра на основе генератора многотактовых кодовых фильтров. Расшифровать 11 бит открытого текста
  10. Построить шифратор самосинхронизирующегося поточного шифра на основе генератора Фибоначчи. Зашифровать 11 бит открытого текста
  11. Построить дешифратор самосинхронизирующегося поточного шифра на основе генератора Фибоначчи. Расшифровать 11 бит открытого текста
  12. Зашифровать 7 бит открытого текста с помощью алгоритма S-A5.
  13. Зашифровать сообщение с помощью алгоритма без передачи ключа
  14. Зашифровать сообщение с помощью алгоритма RSA.
  15. Зашифровать сообщение с помощью алгоритма Эль-Гамала.
  16. Вычислить секретный ключ с помощью алгоритма Диффи-Хеллмана.
  17. Провести аутентификацию претендента с использованием протокола, построенного на доказательстве с нулевым разглашением знаний.

### **Список литературы, рекомендуемый к использованию по данной теме:**

#### **Основная литература:**

- Басалова, Г. В. Основы криптографии : учебное пособие / Басалова Г. В. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 282 с. - Книга находится в базовой версии ЭБС IPRbooks
- Кукина, Е. Г. Введение в криптографию : Сборник задач и упражнений / Кукина Е. Г. - Омск : Омский государственный университет им. Ф.М. Достоевского, 2013. - 91 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-7779-1588-7

#### **Дополнительная литература:**

- Применение искусственных нейронных сетей и системы остаточных классов в криптографии / [авт. кол.: Червяков Н.И., Евдокимов А.А., и др.]. - Москва : ФИЗМАТЛИТ, 2012. - 280 с. : ил. - Прил.: с. 269-279. - Библиогр.: с. 269-279 (191 назв.) и в конце гл. - ISBN 978-5-9221-1386-1.

#### **Интернет-ресурсы:**

[http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=5170](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=5170) Малюк А.А., Теория защиты информации, издательство "Горячая линия-Телеком", 2012"  
<http://siblec.ru/index.php?dn=html&way=> - банк лекций  
[http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=4427](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=4427) Панин.В.В., Основы теории информации, издательство "Бином. Лаборатория знаний", 2012

**Список литературы, рекомендуемый к использованию в самостоятельной работе студентов:**

**Основная литература:**

Басалова, Г. В. Основы криптографии : учебное пособие / Басалова Г. В. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 282 с. - Книга находится в базовой версии ЭБС IPRbooks  
Кукина, Е. Г. Введение в криптографию : Сборник задач и упражнений / Кукина Е. Г. - Омск : Омский государственный университет им. Ф.М. Достоевского, 2013. - 91 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-7779-1588-7

**Дополнительная литература:**

Применение искусственных нейронных сетей и системы остаточных классов в криптографии / [авт. кол.: Червяков Н.И., Евдокимов А.А., и др.]. - Москва : ФИЗМАТЛИТ, 2012. - 280 с. : ил. - Прил.: с. 269-279. - Библиогр.: с. 269-279 (191 назв.) и в конце гл. - ISBN 978-5-9221-1386-1.

**Интернет-ресурсы:**

[http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=5170](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=5170) Малюк А.А., Теория защиты информации, издательство "Горячая линия-Телеком", 2012"  
<http://siblec.ru/index.php?dn=html&way=> - банк лекций  
[http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=4427](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=4427) Панин.В.В., Основы теории информации, издательство "Бином. Лаборатория знаний", 2012