Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«Академия управления городской средой, градостроительства и печати»

УТВЕРЖДАЮ
Заместитель директора
по учебно-производственной работе
_____ О.В. Фомичева
«__» _____ 2025 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по выполнению практических работ
по МДК.03.05 Основы криптографической защиты данных
ПМ.03 ЭКСПЛУАТАЦИЯ ОБЪЕКТОВ СЕТЕВОЙ ИНФРАСТРУКТУРЫ

для специальности

09.02.06 Сетевое и системное администрирование

Санкт-Петербург
2025 г.

Методические рекомендации рассмотрены на заседании методического совета СПб ГБПОУ «АУГСГиП»
Протокол № 2 от «22» 11 202 3 г.

Методические рекомендации одобрены на заседании цикловой комиссии информационных технологий
Протокол № 4 от «21» 11 202 3 г.

Председатель цикловой комиссии: Караченцева М.С. _____

Разработчики: преподаватели СПб ГБПОУ «АУГСГиП»

С О Д Е Р Ж А Н И Е

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая тетрадь по выполнению практических работ предназначены для организации работы на практических занятиях по МДК.03.05 «Основы криптографической защиты данных», которая является важной составной частью в системе подготовки специалистов среднего профессионального образования по специальности 09.02.06 «Сетевое и системное администрирование».

Практические занятия являются неотъемлемым этапом изучения учебной дисциплины и проводятся с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными рабочей программой учебной дисциплины;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- готовности использовать теоретические знания на практике.

Практические занятия по МДК.03.05 «Основы криптографической защиты данных» способствуют формированию в дальнейшем при изучении профессиональных модулей, следующих общих и профессиональных компетенций:

ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

3.7 Применять криптографические аппаратные средства защиты информации на защищаемых объектах

В рабочей тетради предлагаются к выполнению практические работы, предусмотренные учебной рабочей программой МДК.03.05 «Основы криптографической защиты данных».

При разработке содержания практических работ учитывался уровень сложности освоения студентами соответствующей темы, общих и профессиональных компетенций, на формирование которых направлена дисциплина.

Выполнение практических работ в рамках МДК.03.05 «Основы криптографической защиты данных» позволяет освоить комплекс работ по выполнению практических заданий по всем темам МДК.03.05 «Основы криптографической защиты данных».

Рабочая тетрадь по МДК.03.05 «Основы криптографической защиты данных» имеют практическую направленность и значимость. Формируемые в процессе практических занятий умения могут быть использованы студентами в будущей профессиональной деятельности.

Рабочая тетрадь предназначена для студентов колледжа, изучающих МДК.03.05 «Основы криптографической защиты данных».

Оценки за выполнение практических работ выставляются по пятибалльной системе. Оценки за практические работы являются обязательными текущими оценками и выставляются в журнале теоретического обучения.

1. Перечень практических работ по МДК.03.04 «Основы криптографической защиты данных»

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
Тема 5.2. Классификация шифров	создавать программы, реализующие алгоритмы и протоколы защищенной передачи данных	ОК 1-ОК 9 ПК 3.7	Практическая работа № 1 Применение шифров перестановки	2
			Практическая работа № 2 Алгоритмизация шифра Цезаря	2
			Практическая работа № 3 Декодирование моноалфавитного подстановочного шифра частотным методом	2
			Практическая работа № 4 Применение основ модулярной арифметики, проверка простоты и факторизация чисел.	2
	конструировать криптостойкие алгоритмы и протоколы		Практическая работа № 5 Применение шифров гаммирования	2
			Практическая работа № 6 Применение комбинированных шифров	2
Тема 5.3. Криптографические протоколы	применять на практике алгоритмы шифрования секретным ключом	ОК 1-ОК 9 ОК 1-ОК 9 ОК 1-ОК 9	Практическая работа № 7 Метод шифрования с открытым ключом RSA	2
			Практическая работа № 8 Разработка хэш-функции	2
			Практическая работа № 9 Использование шифросистемы Эль-Гамаля	2
			Практическая работа № 10 Применение бесключевого протокола Шамира	2
			Практическая работа № 11 Применение электронной подписи (ГОСТы 34.10-94 и 34.10-2001)	2
			Практическая работа № 12 Настройка ПО для работы с электронной подписью	2
Тема 5.4. Основы криптоанализа	проводить анализ криптостойкости алгоритмов и протоколов	ОК 1-ОК 9 ОК 1-	Практическая работа № 13 Изучение частотного метода криптоанализа симметричных криптосистем	2

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
		ОК 9 ОК 1-ОК 9	Практическая работа № 14 Изучение методов криптоанализа криптосистем гаммирования с периодической гаммой	2
			Практическая работа № 15 Изучение метода линейного криптоанализа блочных симметричных криптосистем	2
			Практическая работа № 16 Изучение метода дифференциального (разностного) криптоанализа блочных симметричных криптосистем	2
			Практическая работа № 17 Методы оценки качества криптографических генераторов	2
Тема 5.5. Стеганография	проводить анализ данных на наличие скрытой информации	ПК 3.7	Практическая работа № 18 Применение текстовой криптографии	2
			Практическая работа № 19 Исследование методов цифровой стеганографии для защиты информации	2
			Практическая работа № 20 Решение ситуационных задач	2
			Практическая работа № 21 Применение LSB-стеганографии	2
			Практическая работа № 22 Применение метода замены цифровой палитры	2
			Практическая работа № 23 Анализ графических изображений на наличие скрытой информации.	2
			Практическая работа № 24 Применение ОС Kali Linux в стеганографии	2
			Практическая работа № 25 Решение ситуационных задач	2

2. Описание порядка выполнения практических работ

2.1. Практическая работа №1 «Применение шифров перестановки»

Задание:

Шифр **Цезаря** — один из древнейших шифров. При шифровании каждый символ заменяется другим, отстоящим от него в алфавите на фиксированное число позиций. Шифр Цезаря можно классифицировать как шифр подстановки, при более узкой классификации — шифр простой замены.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки. Естественным развитием шифра Цезаря стал шифр Виженера. С точки зрения современного криптоанализа, шифр Цезаря не имеет приемлемой стойкости.

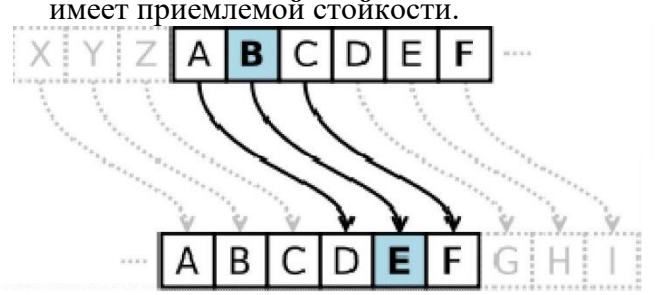


Рисунок 1 Шифр Цезаря

Математическая модель

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами:

$$y = x + k \ (mod\ n)$$
$$x = y - k \ (mod\ n),$$

Где x		— символ открытого текста
y		— символ шифрованного текста
n	— мощность алфавита (кол-во символов)	
k		— ключ

Можно заметить, что суперпозиция двух шифрований на ключах k1 и k2 — есть просто шифрование на ключе k1+k2. Более общее, множество шифрующих преобразований шифра Цезаря образует группу Z. Алфавит:

Буква	**А**	**Б**	**В**	**Г**	**Д**	**Е**	**Ё**	**Ж**	**З**	**И**	**Й**
Номер	1	2	3	4	5	6	7	8	9	10	11
Буква	**К**	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**
Номер	12	13	14	15	16	17	18	19	20	21	22
Буква	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Ы**	**Ъ**	**Э**	**Ю**	**Я**
Номер	23	24	25	26	27	28	29	30	31	32	33

Пример:

Сообщение	К	Р	И	П	Т	О	Г	Р	А	Ф	И	Я
Номер 1	12	18	10	17	20	16	4	18	1	22	10	33
Номер 1 +5	17	23	15	22	25	21	9	23	6	27	15	5
Шифр	П	Х	Н	Ф	Ч	У	З	Х	Е	Щ	Н	Д

Ответ: «Пхнфчузхещнд»

Задание для работы

Зашифровать шифром Цезаря предложения (КЛЮЧ 3)

байты сохраняются в виде файлов

ФИО

2.2. Практическая работа № 2 «Алгоритмизация шифра Цезаря»

Задание:

1. Ознакомьтесь с теоретической частью практической работы.
2. Загрузите программу Microsoft Excel.
3. На первом листе электронной книги запишите в столбец А буквы русского алфавита. В столбце В – номер букв, в столбце С – опять буквы (такая запись будет необходима для использования функции ВПР).

	A	B	C	D
1	а	0	а	
2	б	1	б	
3	в	2	в	
4	г	3	г	
5	д	4	д	
6	е	5	е	
7	ж	6	ж	
8	з	7	з	
9	и	8	и	
10	й	9	й	
11	к	10	к	
12	л	11	л	
13	м	12	м	
14	н	13	н	
15	о	14	о	
16	п	15	п	
17	р	16	р	
18	с	17	с	
19	т	18	т	
20	у	19	у	
21	ф	20	ф	
22	х	21	х	
23	ц	22	ц	
24	ч	23	ч	
25	ш	24	ш	
26	щ	25	щ	
27	ъ	26	ъ	
28	ы	27	ы	
29	ь	28	ь	
30	э	29	э	
31	ю	30	ю	
32	я	31	я	

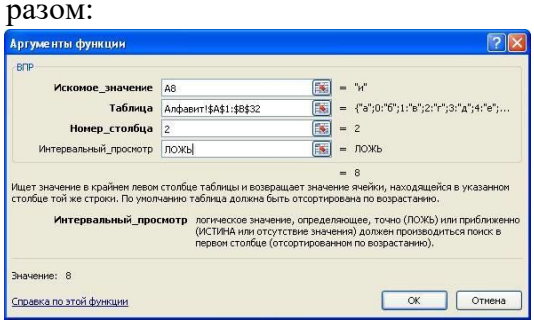
4. Переименуйте лист1 в Алфавит.

5. На втором листе электронной книги запишите название работы, ключ и название столбцов таблицы (S – исходные символы, X – числа исходных символов, Y – пересчитанные по формуле значения, S1 – символы закрытого текста). Значение ключа можно взять любым и обязательно его значение записать в отдельную ячейку (B5). В столбец S, начиная с 8 строки, впишите фамилию и имя, каждую букву в отдельной ячейке.

	A	B	C	D	E	F	G
1					Шифр Цезаря		
2							
3	1. Зашифровывание						
4							
5	k=	5					
6							
7	S	X	Y	S1			
8	и						
9	в						
10	а						
11	н						
12	о						
13	в						
14	а						
15	н						
16	д						
17	р						
18	е						
19	й						
20							

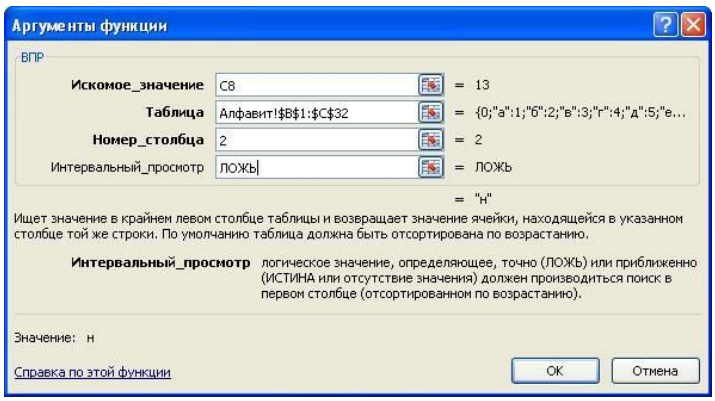
6. В столбце X должны быть числовые значения символов из столбца S. Эти значения хранятся на листе Алфавит. Чтобы получить их, можно воспользоваться функцией **ВПР** (категория – ссылки и массивы).

Встаем в ячейку B8 и вызываем функцию ВПР. Заполняем ее окно следующим образом:

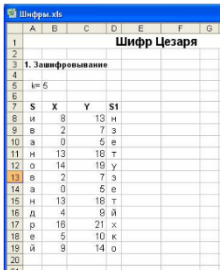


7. Растянуть формулу вниз до конца таблицы.
8. В ячейку C8 (столбец Y) записывается формула для шифрования. Исходная формула метода Цезаря имеет вид: $y_i = (x_i + k) \bmod n$. Операции mod в Excel соответствует функция **ОСТАТ(число; делитель)**. В нашем случае **число** – это (x_i +k), а **делитель** – 32.

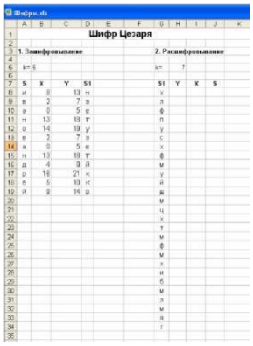
   Т.е. функция **ОСТАТ** будет иметь вид **=ОСТАТ((B8+$B$5);32)**.
9. Эту формулу необходимо растянуть вниз до конца таблицы.
10. В ячейку D8 (столбец S1) опять записываем функцию **ВПР**, которая по числу Y найдет букву. Эта функция будет выглядеть следующим образом:



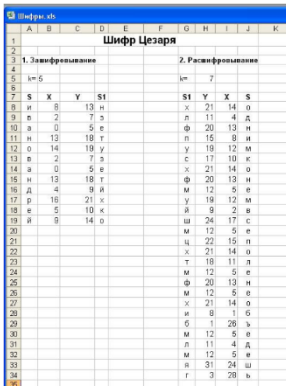
11. Окончательно таблица должна выглядеть следующим образом:



12. Рядом приготовьте место для дешифрования информации. Получите у преподавателя карточку с закрытым текстом и впишите его в столбец S1 новой таблицы.



13. Проведите дешифрования текста по аналогии с зашифровыванием. Для расшифровывания (столбца X) используйте формулу $x_i = (y_i + (32-k)) \bmod 32$.



14. Запишите полученную фразу.

2.3. Практическая работа № 3 «Декодирование моноалфавитного подстановочного шифра частотным методом»

Задание:

Исходные данные:

Зашифрованный текст, перечень наиболее часто встречающихся букв в тексте, перечень наиболее часто используемых в русском языке букв.

Выходные данные: Расшифрованный текст.

Порядок выполнения:

1. Запустить на выполнение файл labw01.exe

На экране появится окно выполнения лабораторной работы (рис. 1):

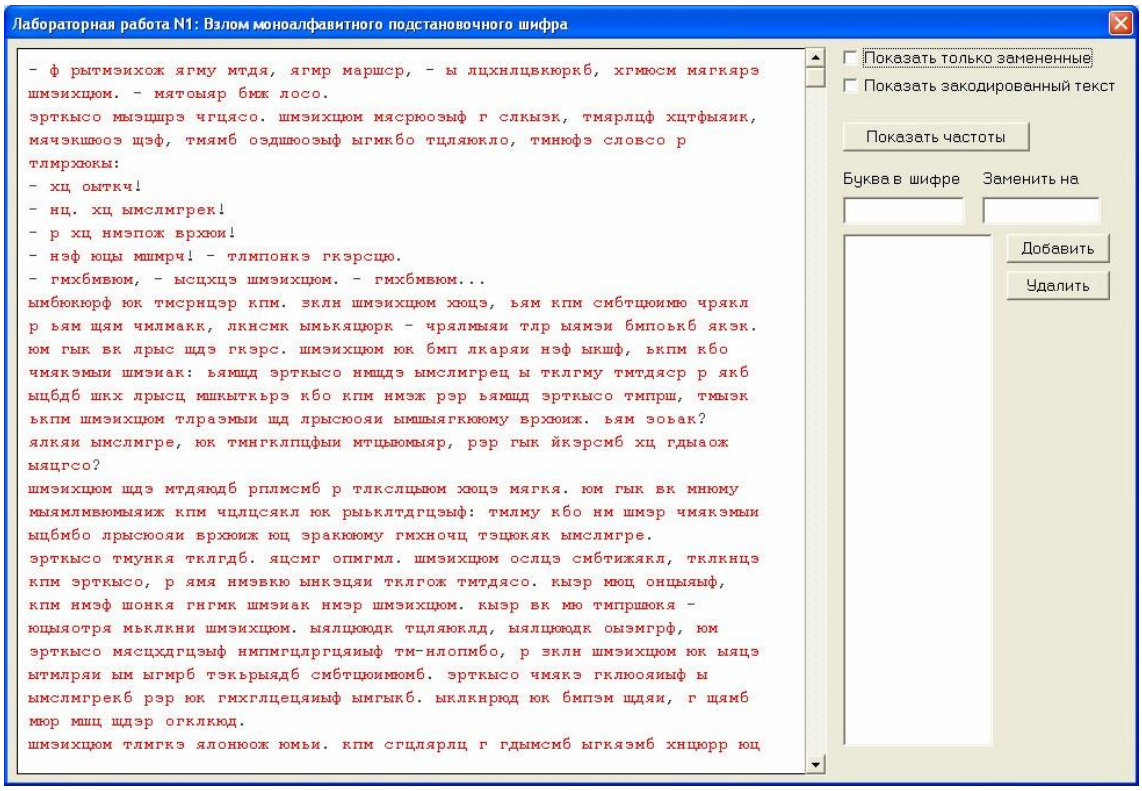


Рисунок 1. Окно выполнения лабораторной работы

В левой части окна находится зашифрованный текст (буквы, выделенные красным цветом). В процессе расшифровки расшифрованные (правильно или неправильно) буквы текста меняют цвет с красного на черный.

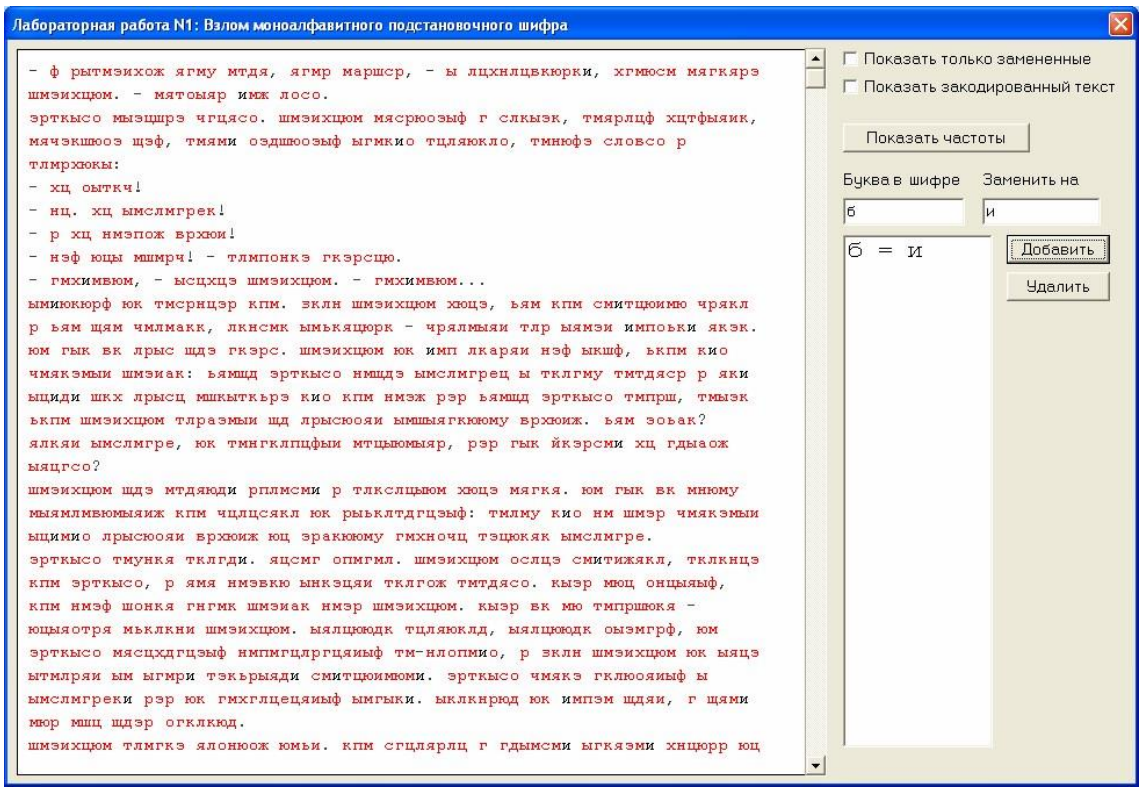


Рисунок 2. Изменения окна лабораторной работы после расшифровки одной буквы

Чтобы указать для какой-либо буквы шифра ее истинное (расшифрованное) значение, нужно в поле «Буква в шифре» указать значение буквы, например, “б”, а в поле «Заменить на» - ее истинное значение, например, “и”, а затем нажать кнопку “Добавить“. Результат такого действия приведен на рис. 2.

На рис. 3. Приведено окно выполнения лабораторной работы после добавления расшифровок нескольких букв.

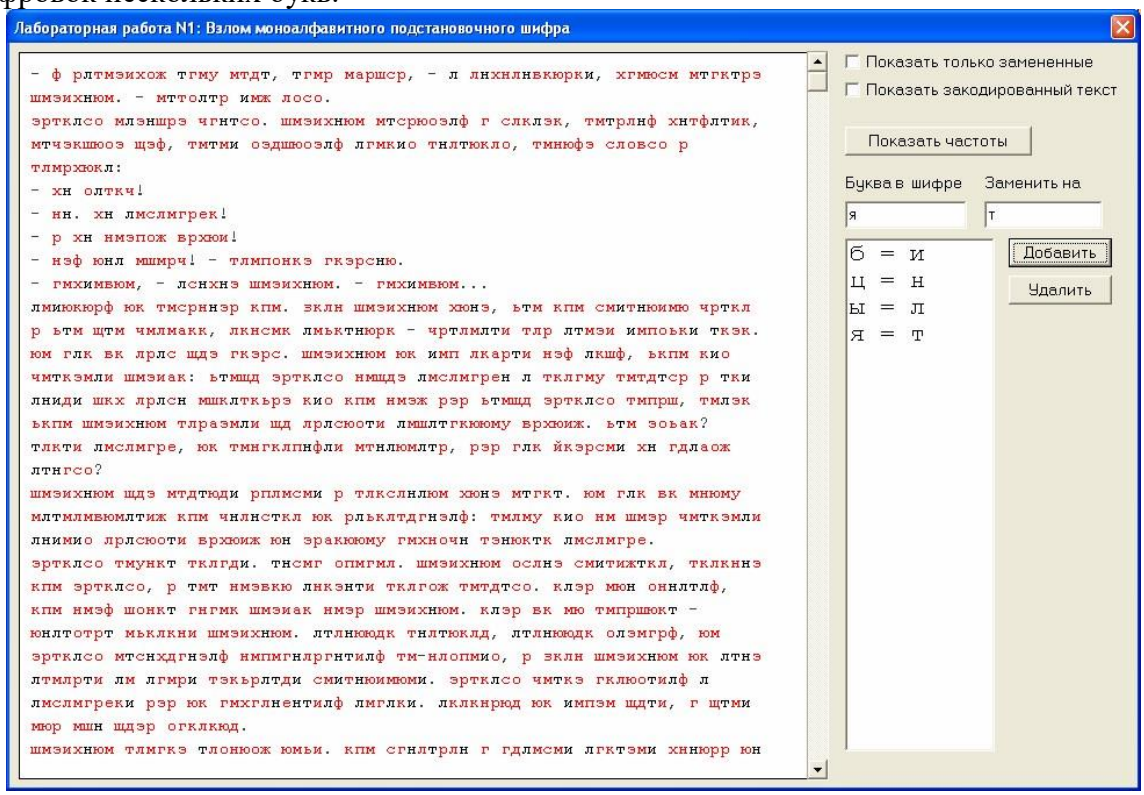


Рисунок 3. Окно лабораторной работы после расшифровки нескольких букв

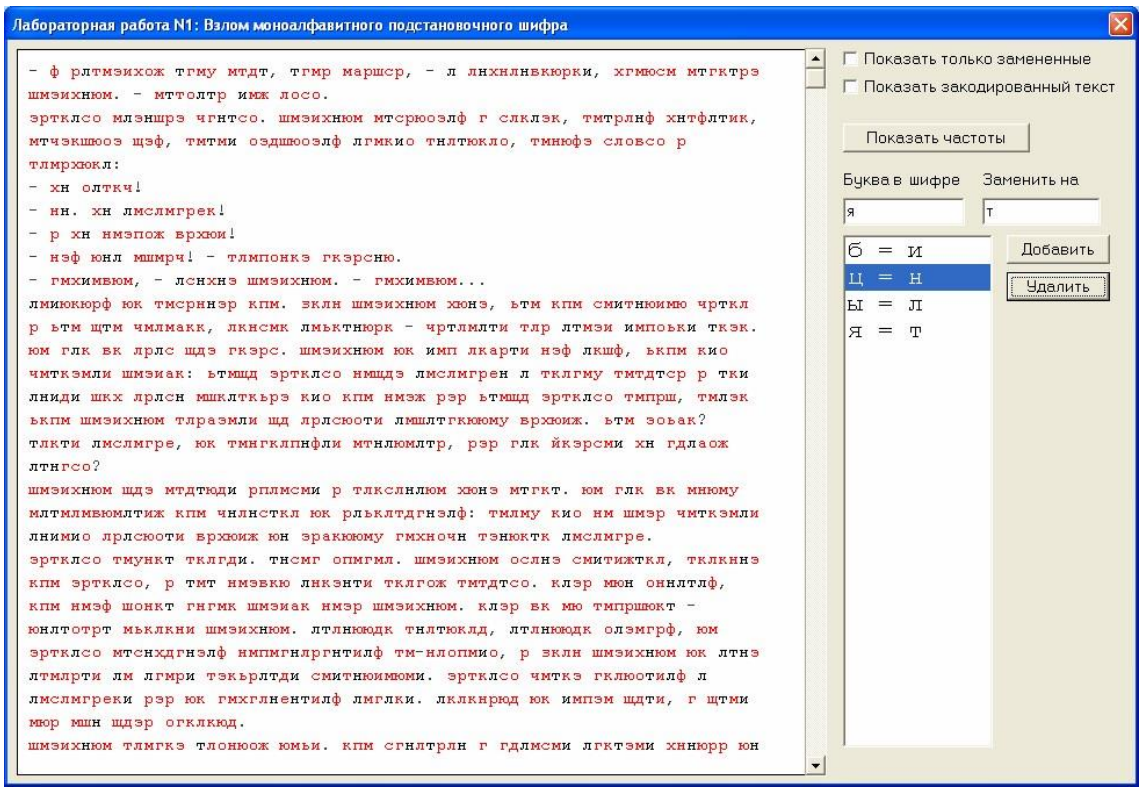


Рисунок 4. Процедура удаления ошибочно указанных расшифровок

Чтобы отменить указанную расшифровку буквы, нужно в списке расшифровок мышкой указать соответствующую пару букв и нажать кнопку «Удалить» (рис. 4).

Полоса вертикального скроллинга служит для навигации по расшифровываемому тексту.

2. Начинается частотная атака с анализа частот встречаемости букв в шифровке. Для этих целей в окне выполнения лабораторной работы предусмотрена кнопка «Показать частоты». При ее нажатии на экран выводится перечень десяти наиболее часто встречаемых букв в шифре, а также перечень букв, наиболее часто встречаемых в русском языке (рис. 5).

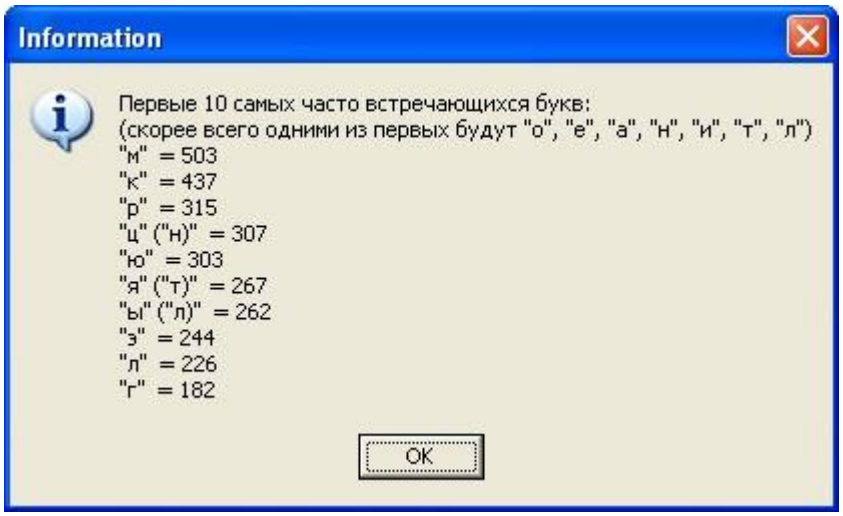


Рисунок 5. Информация о частотах встречаемости букв в шифре

Первым шагом в расшифровке текста может быть указание расшифровки для самой часто встречаемой буквы - буквы «о». Для случая, приведенного на рис. 5, указывается «о» как расшифровка буквы «м» шифра (см. рис. 6).

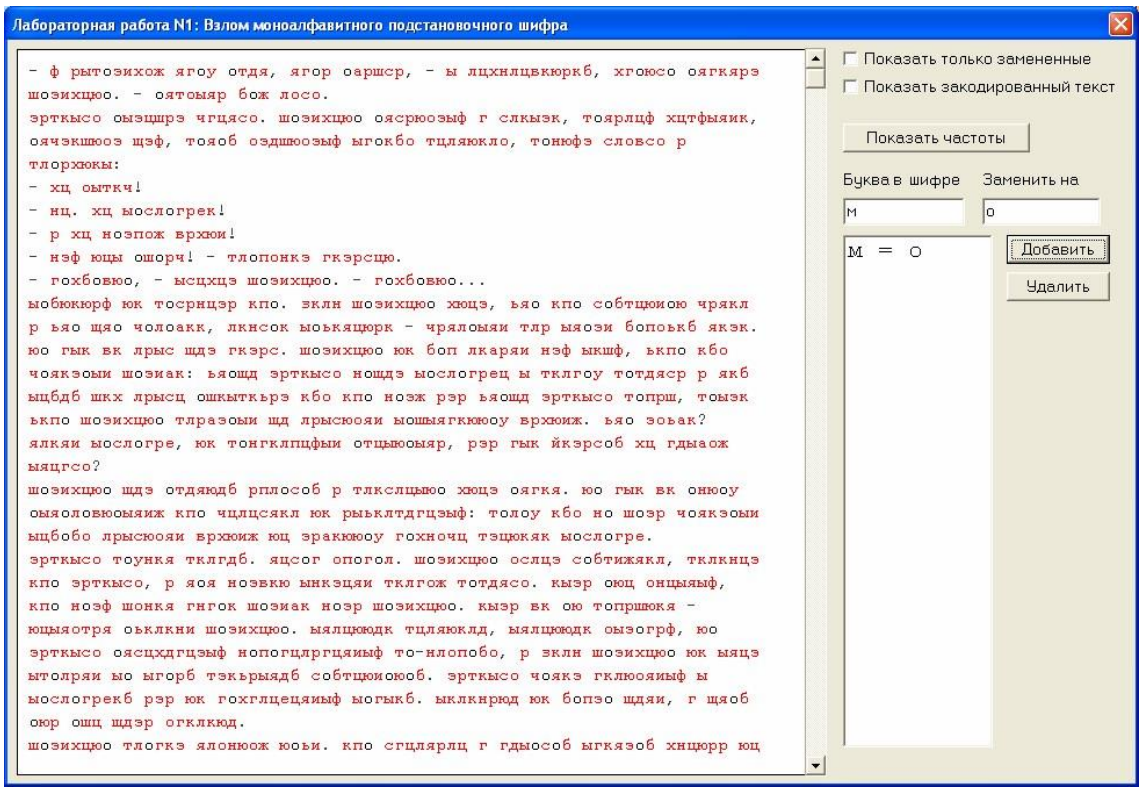


Рисунок 6. Первый шаг расшифровки - указание расшифровки буквы «о»

Следует помнить, что для конкретного текста частота встречаемости букв может быть несколько иной, чем в среднем для русского языка. Если в русском языке, например, буква «т» встречается чаще, чем буква «л», то в каком-то конкретном тексте буква «л» вполне может встречаться чаще буквы «т». Поэтому слепо опираться на данные частотного анализа не следует.

3. В зашифрованном тексте осуществляется поиск коротких слов, зашифрованные буквы которых можно предсказать по уже расшифрованным буквам и частотной информации из рис. 5. На рис. 7. в верхней строчке есть фрагмент текста « ою », где «о» уже известно

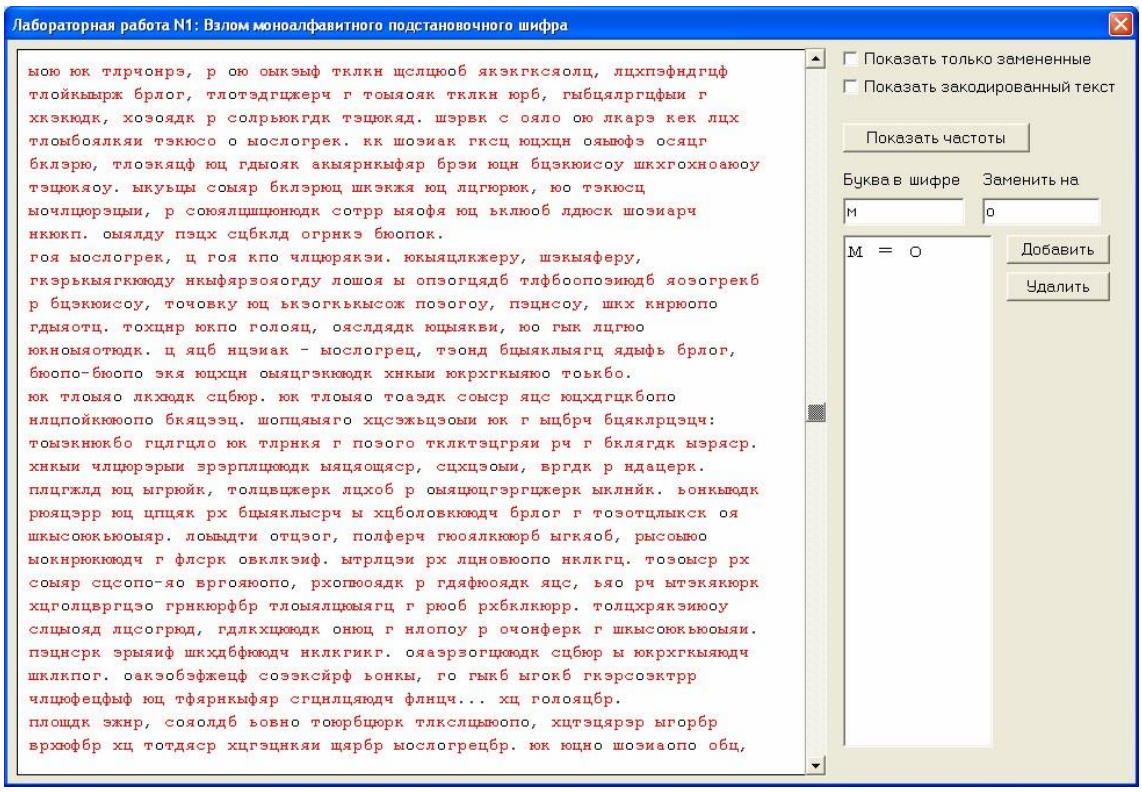


Рисунок 7. Поиск коротких слов

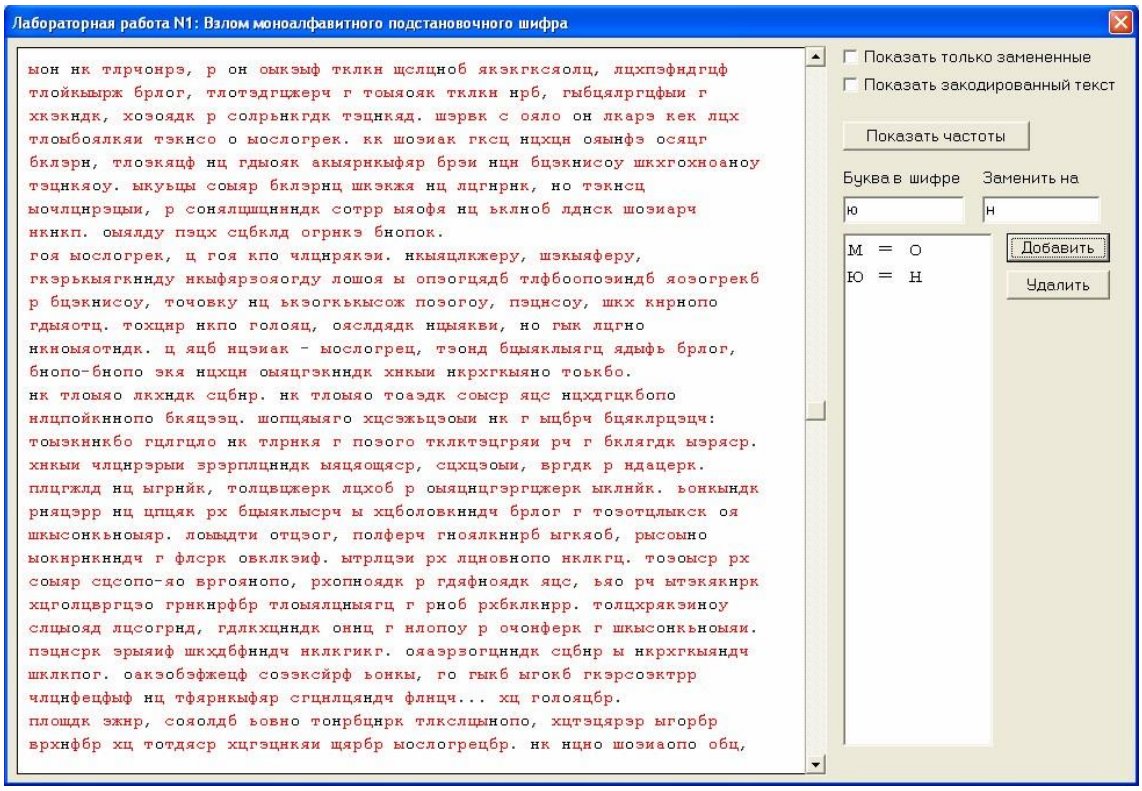


Рисунок 8. Результат расшифровки букв «о» и «н»

Этот фрагмент может быть скорее всего словом « он » В таблице частот (рис. 5) буква «ю» шифра стоит на 5-м месте, что примерно соответствует позиции буквы «н» русского языка (4-е место). Значит разумно попробовать поменять «ю» на «н». Результат приведен на рис. 8.

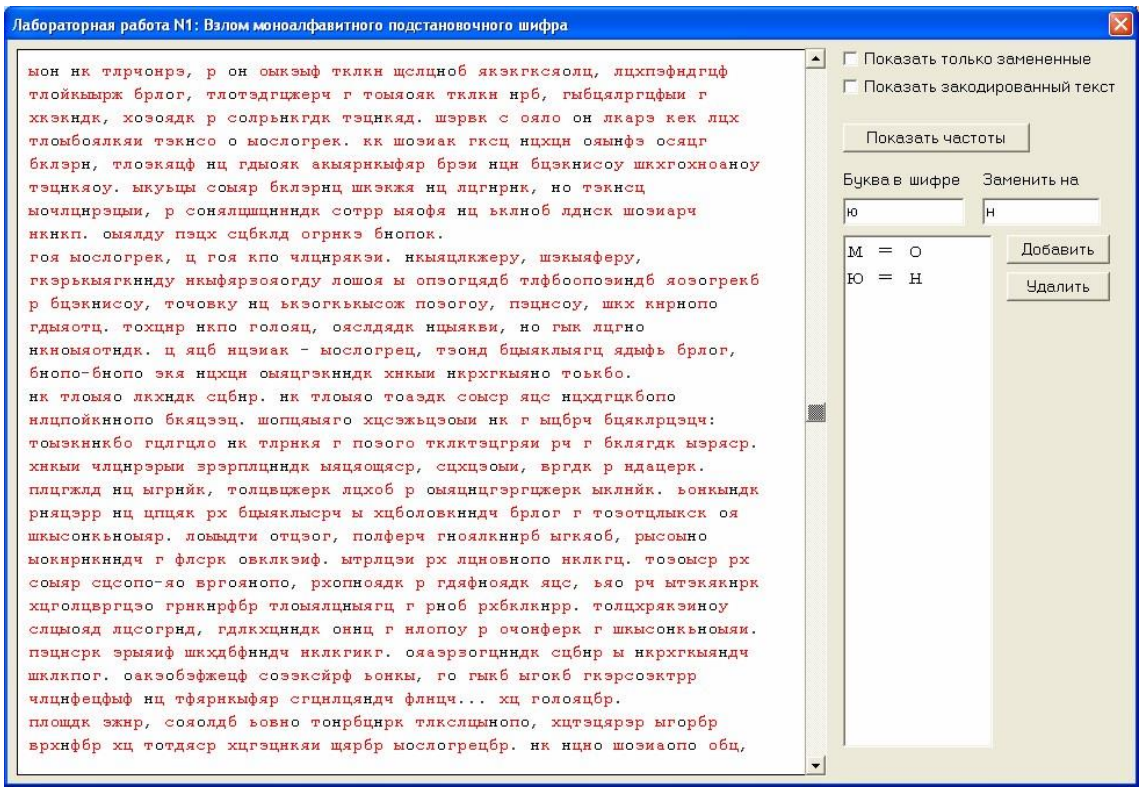


Рисунок 9. Продолжение поиска коротких понятных слов

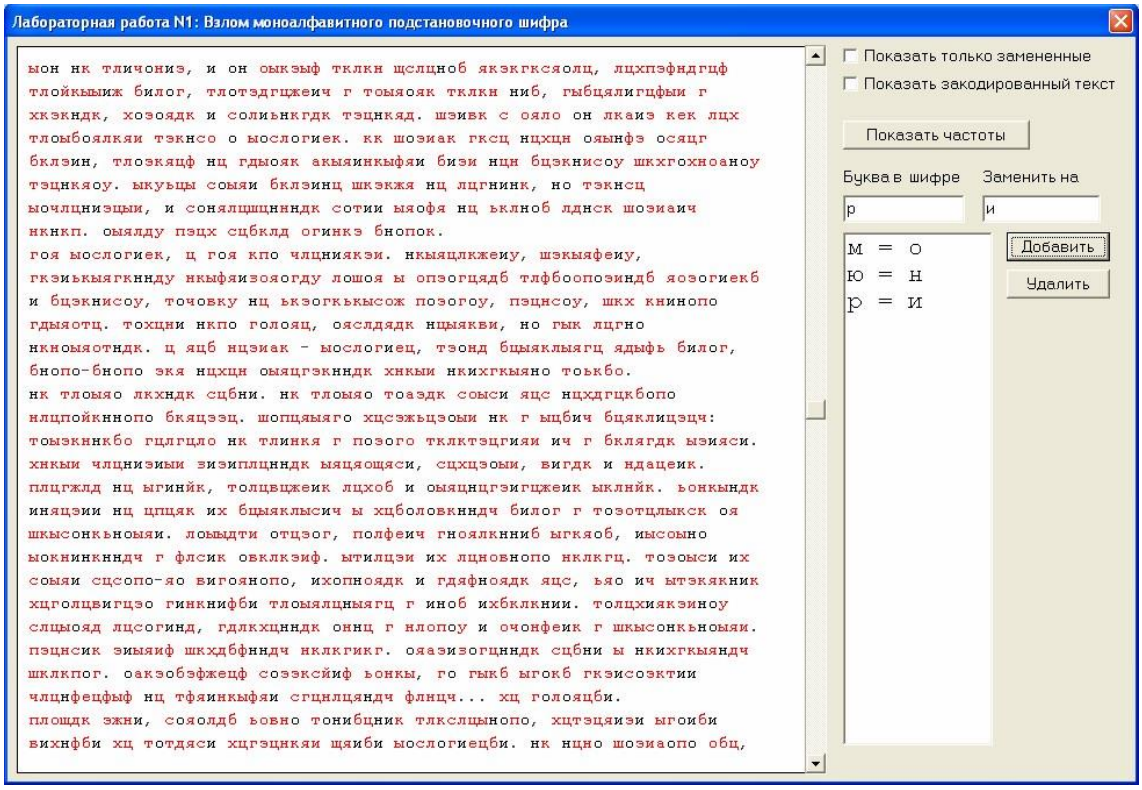


Рис. 10. Результат расшифровки букв «о», «н» и «и»

Далее повторяется поиск коротких слов, в которых можно догадаться о значении зашифрованных букв. На рис. 9 в первой и третей строках есть отдельно стоящее «р». Скорее всего это предлог «и», что согласуется и с информацией на рис. 5. Результат замены приведен на рис. 10.

На рис. 11 в первой строке обнаруживается слово из двух известных «и» и шифрованной буквы «э» между ними. Скорее всего это буква «л», образующая слово «или» (рис. 12).

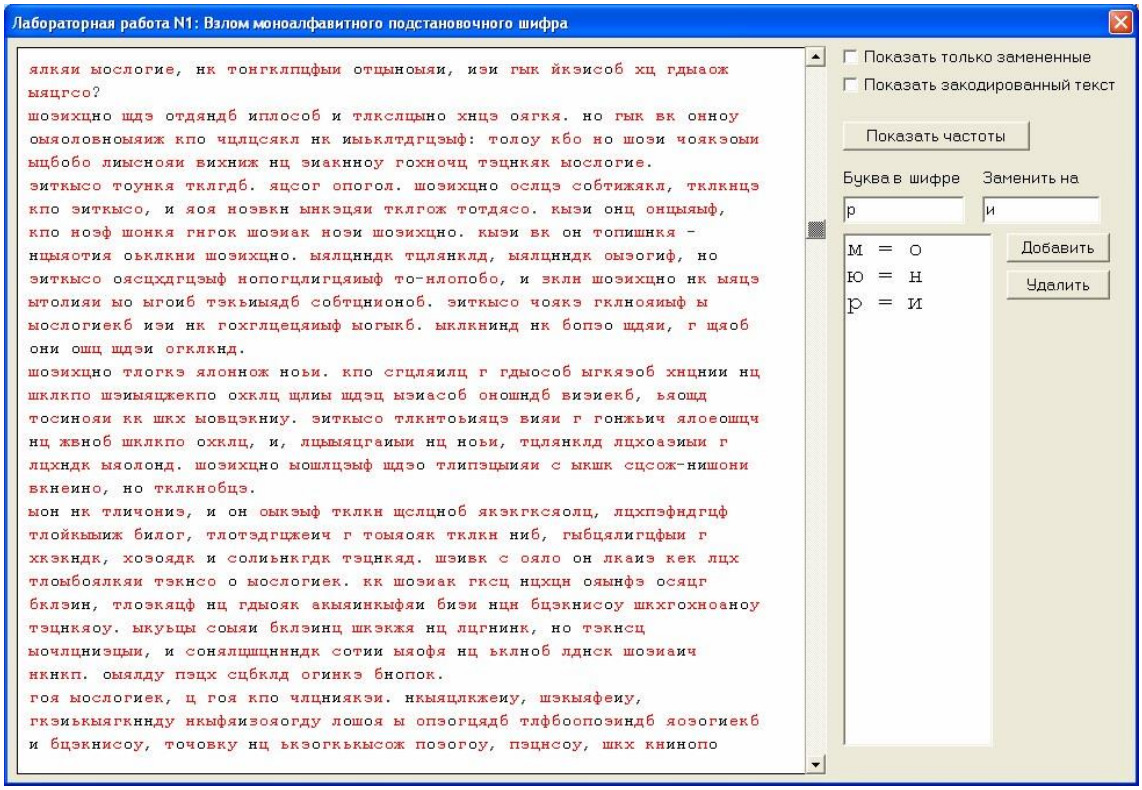


Рисунок 11. Продолжение поиска коротких понятных слов

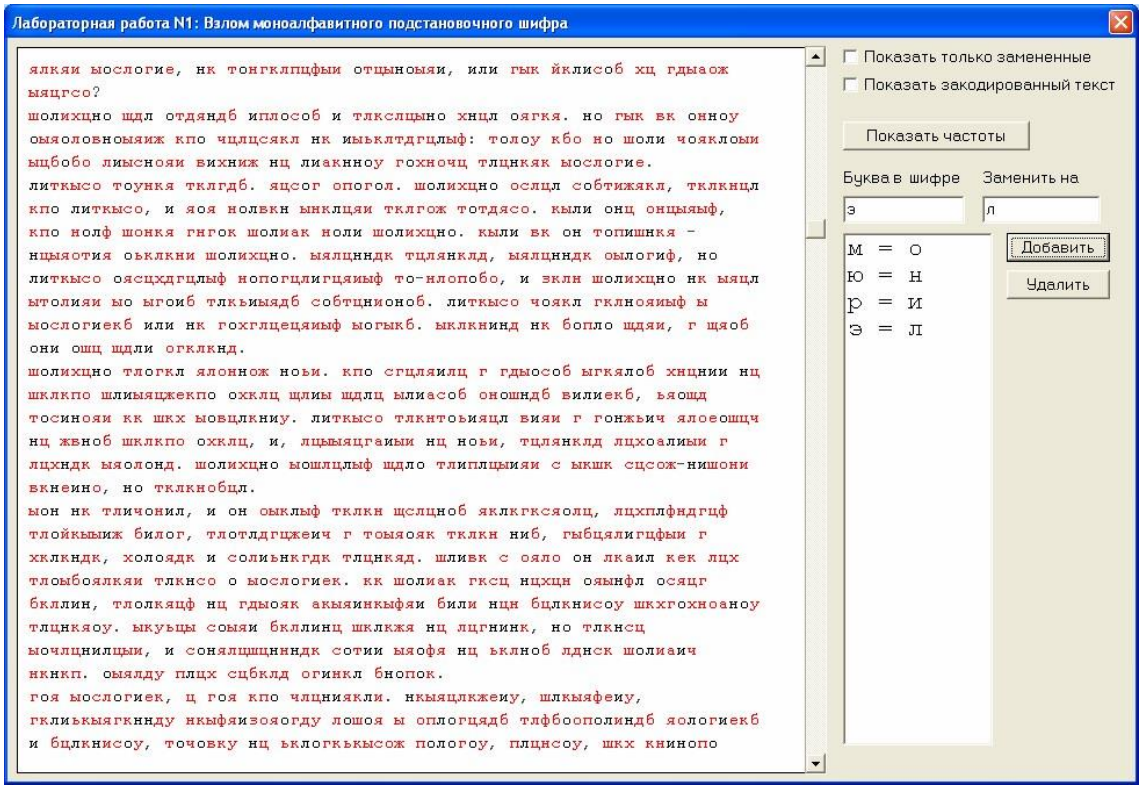


Рисунок 12. Результат расшифровки букв «о», «н», «и» и «л»

После расшифровки аналогичным образом букв «к» на «е», «ц» на «а» и «я» на «т» окно выполнения лабораторной работы приобретает следующий вид (рис. 13):

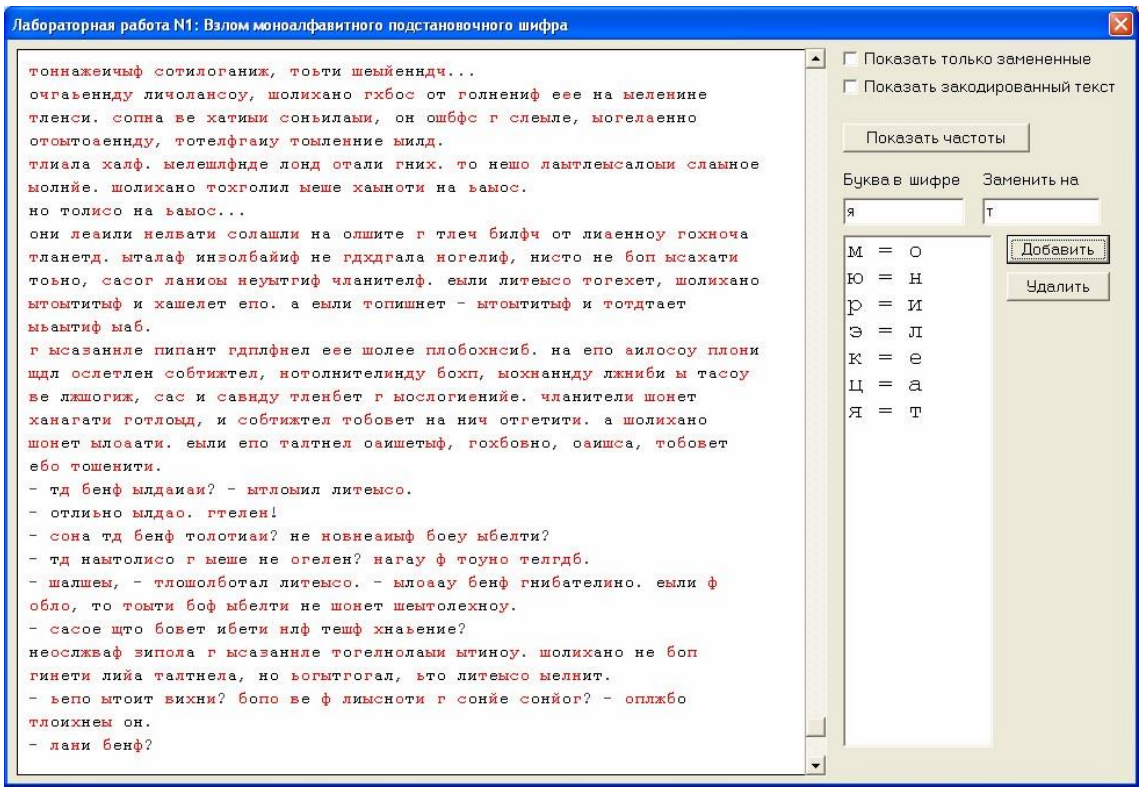


Рисунок 13. Окно выполнения лабораторной работы после расшифровки семи букв

Когда так много букв уже известно, зашифрованные буквы могут мешать для понимания слов. Для облегчения дальнейшего анализа в программе предусмотрена возможность выставления флага «Показать только замененные», при выставлении которого все зашифрованные буквы выводятся на экран в виде символов решетки (рис. 14).

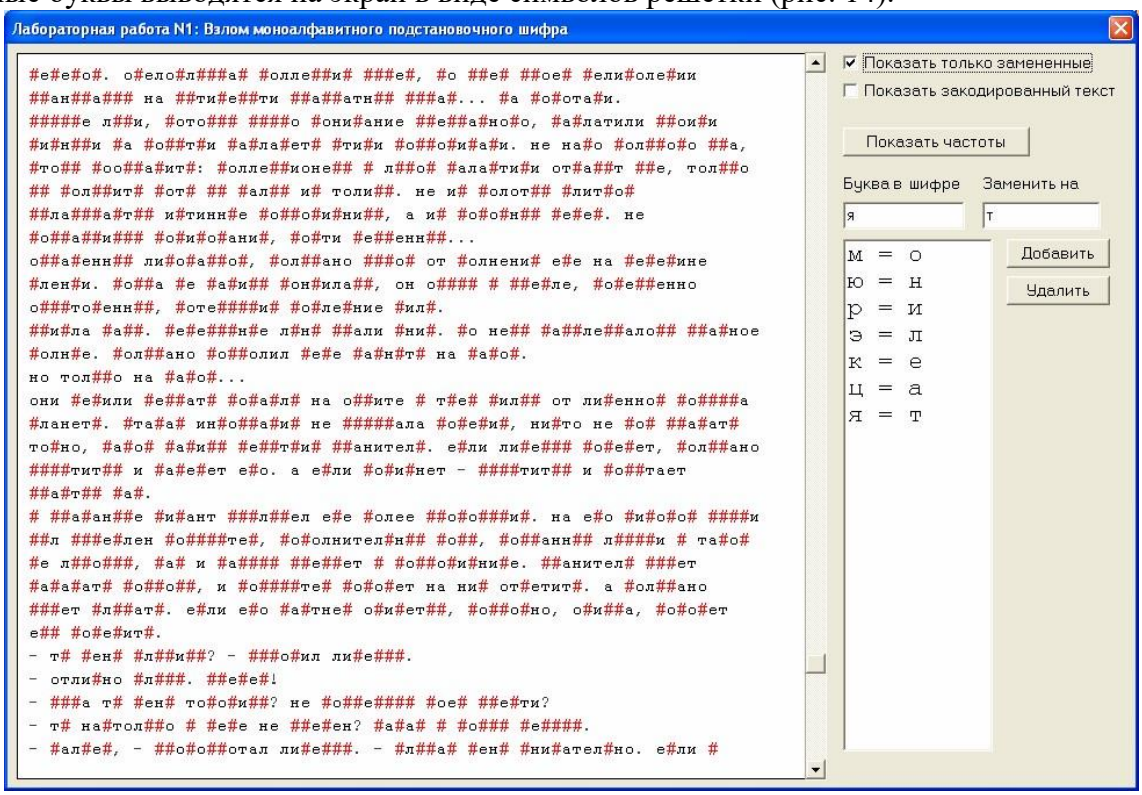


Рис. 14. Использование флага «Показать только замененные»

Теперь видно, что слово «##о#о##отал» в нижней строке вполне может быть словом «пробормотал». Если теперь выключить флаг, то можно получить косвенное подтвержде-

ние этого - на позициях двух букв «р» в этом слове в шифре также находится одинаковая буква «л» (рис. 15).

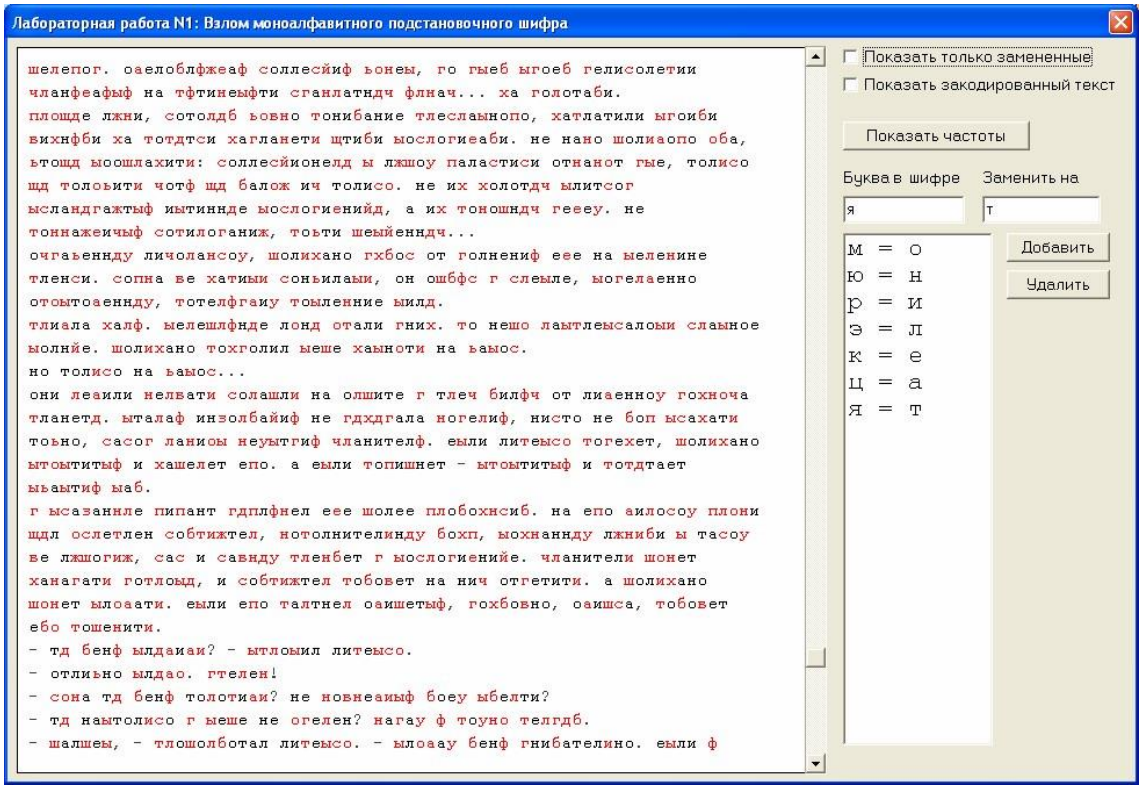


Рисунок 15. Проверка гипотезы отключением флага

Если заменить теперь букву «т» на «п», «л» на «р», «ш» на «б» и «б» на «м», то окно выполнения лабораторной работы станет выглядеть так(рис. 16):

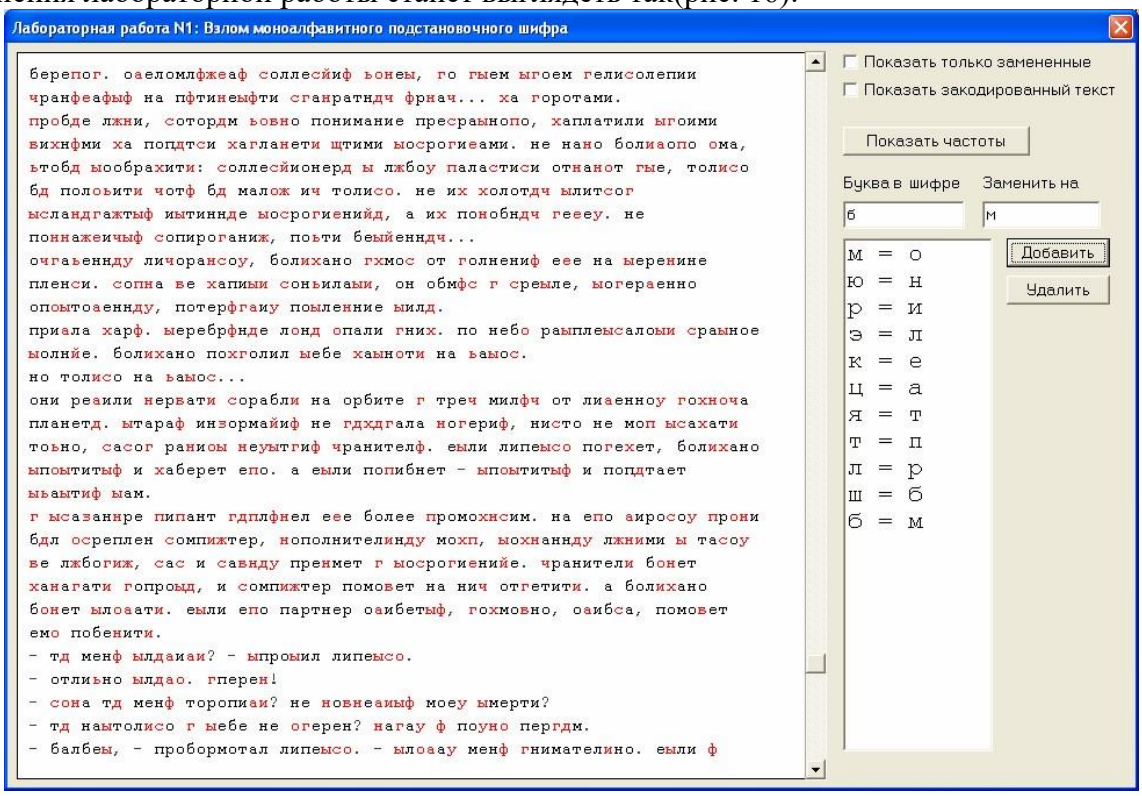


Рисунок 16. Окно лабораторной работы после расшифровки букв «п», «р», «б» и «м».

Хорошо видно, что дальнейший анализ значительно упрощается. Например, очевидно по слову «хаплатили», что буква «х» шифра соответствует букве «з» исходного текста. На рис. 17 приведено окно программы, когда анализ уже близок к завершению (осталось совсем немного нерасшифрованных букв).

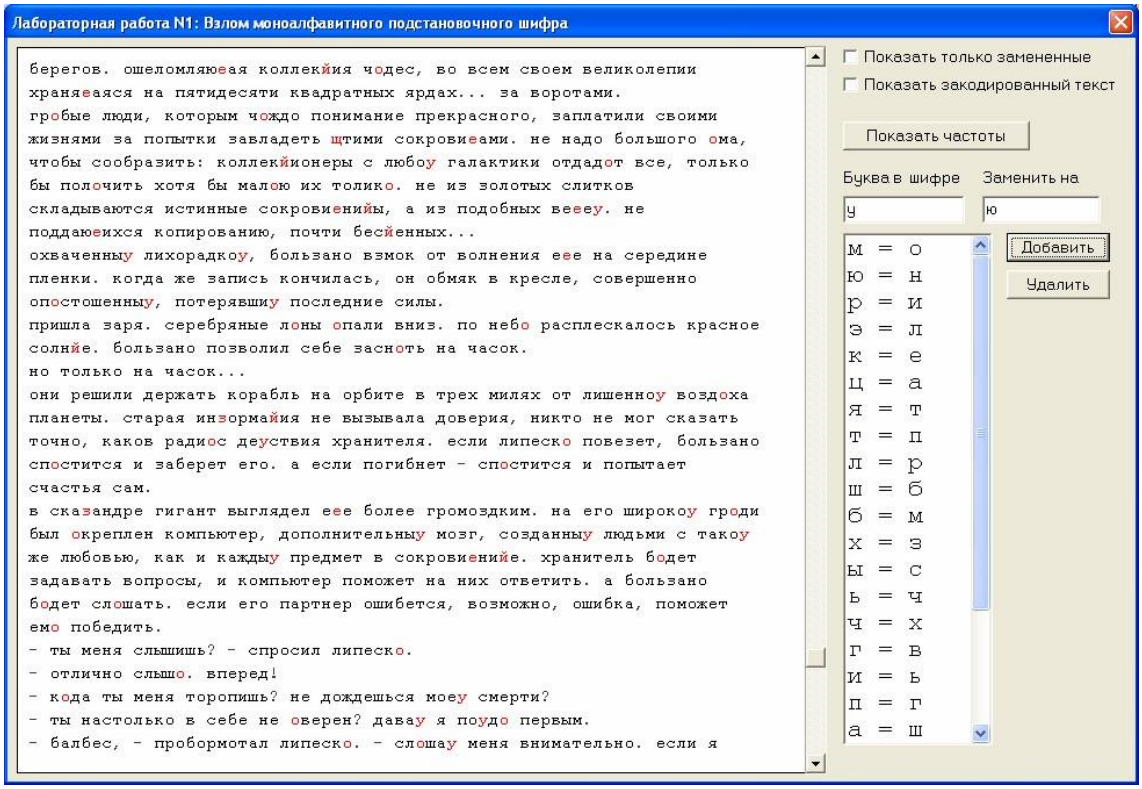


Рисунок 17. Расшифрованы почти все буквы текста

Когда же все буквы текста расшифрованы, на экран выводится информационное окно (рис. 18):

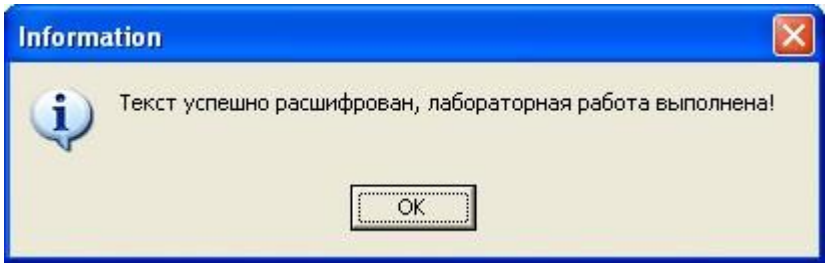


Рисунок 18. Информационное окно, свидетельствующее о успешной расшифровке текста

Появление этого окна на экране свидетельствует об успешном выполнении практической работы.

2.4. Практическая работа № 4 «Применение основ модулярной арифметики, проверка простоты и факторизация чисел.»

Задание:

Задание 1.1. Найти наибольший общий делитель НОД(a,b) двух чисел $a \in Z$, $b \in N$ методом Евклида.

Алгоритм и расчетные формулы метода Евклида:

$$r_{j-2} = q_j r_{j-1} + r_j,\ q_j = \left[\frac{r_{j-2}}{r_{j-1}}\right],\ j = 0,1,2\ldots, \text{ где } r_{-2} = a, r_{-1} = b.$$

Решением задачи является последний ненулевой остаток $r_j > 0$.

Пример. Найти НОД$(1547,560)$. Решение приведено в табл. 1.3.

Таблица 1.3 – Выполнение алгоритма Евклида для примера

шаг	кратность	разложение	остаток
$j=1$	$q_1 = \left[\frac{a}{b}\right] = \left[\frac{1547}{560}\right] = 2$	$a = q_1 b + r_0$: $1547 = 2 \cdot 560 + 427$	$r_0 = 427$
$j=2$	$q_2 = \left[\frac{b}{r_0}\right] = \left[\frac{560}{427}\right] = 1$	$b = q_2 r_0 + r_1$: $560 = 1 \cdot 427 + 133$	$r_1 = 133$
$j=3$	$q_3 = \left[\frac{r_0}{r_1}\right] = \left[\frac{427}{133}\right] = 3$	$r_0 = q_3 r_1 + r_2$: $427 = 3 \cdot 133 + 28$	$r_2 = 28$
$j=4$	$q_4 = \left[\frac{r_1}{r_2}\right] = \left[\frac{133}{28}\right] = 4$	$r_1 = q_4 r_2 + r_3$: $133 = 4 \cdot 28 + 21$	$r_3 = 21$
$j=5$	$q_5 = \left[\frac{r_2}{r_3}\right] = \left[\frac{28}{21}\right] = 1$	$r_2 = q_5 r_3 + r_4$: $28 = 1 \cdot 21 + 7$	$r_4 = 7$
$j=6$	$q_6 = \left[\frac{r_3}{r_4}\right] = \left[\frac{21}{7}\right] = 3$	$r_3 = q_6 r_4 + r_5$: $21 = 3 \cdot 7 + 0$	$r_5 = 0$

Получено: $r_5 = 0$, следовательно, выполнение алгоритма окончено. Решением является предшествующий (ненулевой) остаток $r_4 = 7$: НОД$(1547,560) = 7$.

Задание 1.2. Найти результат преобразования методом, основанным на теореме Эйлера.

Функция Эйлера: $\varphi(n)=\left|\left\{0\le b<n \middle| НОД(b,n)=1\right\}\right|$ обладает следующими ключевыми свойствами:

1) $\varphi(1)=1$;

2) для любого простого p: $\varphi(p)=p-1$;

3) для любого простого p в степени α: $\varphi\left(p^{\alpha}\right)=p^{\alpha}-p^{\alpha-1}$;

4) $\forall m,n\in N \mid НОД(m,n)=1: \varphi(m\cdot n)=\varphi(m)\cdot\varphi(n)$.

Используется теорема Эйлера:

$$\forall a,m\in N \mid НОД(a,m)=1: a^{\varphi(m)}\equiv 1(\mathrm{mod}\, m),$$

и следствие из нее: если НОД$(a,m)=1$ и n' – наименьший неотрицательный вычет n по модулю $\varphi(m)$, то $a^{n}\equiv a^{n'}(\mathrm{mod}\, m)$.

Пример. Провести преобразование: $5^{17}\bmod 7$

Решение. $a=5; n=17; m=7$. При этом 7 – простое число.

Находим $\varphi(m)=\varphi(7)=7-1=6$;

$n'=17\bmod(6)=(2\cdot 6+5)\bmod(6)=5$;

$5^{17}\bmod 7=5^{5}\bmod 7=3125\bmod 7=(446\cdot 7+3)\bmod 7=3$.

Задание 1.3. Найти обратное значение числа по модулю:

– при помощи определения кратности k («по определению»);

– с использованием функции Эйлера.

Сравнить полученные результаты и оценить трудоемкость.

По определению обратные числа по модулю означают, что $a\cdot b\equiv 1(\mathrm{mod}\, m)$. Обратное число $b\equiv a^{-1}(\mathrm{mod}\, m)$ гарантированно

существует, если НОД$(a,m)=1$ (a и m – взаимно простые). При этом выполнено диофантово уравнение: $a\cdot b-k\cdot m=1$, где k – кратность m в числе $a\cdot b$. Внимание: число k должно быть получено *целым* числом.

Пример: найти $x=9^{-1}(\mathrm{mod}\,5)$.

Решение «по определению». Получаем уравнение $9\cdot x-k\cdot 5=1$;

$$x=(1+k\cdot 5):9.$$

Производим последовательный перебор возможных кратностей k как чисел последовательности натуральных чисел $\{1,2,3,4,....\}$ до тех пор, пока не получим x как *целое* число. В данном примере при $k=7$ получено целое число $x=4$.

Итак, получено $9^{-1}(\mathrm{mod}\,5)=4$.

Проверка: $9\cdot 4-7\cdot 5=36-35=1$.

Решение по теореме Эйлера: $a^{\varphi(m)}\equiv 1(\mathrm{mod}\,m)$. Домножим левую и правую части этого уравнения на $a^{-1}(\mathrm{mod}\,m)$. Получаем правило нахождения числа, обратного по модулю (в кольце), *с использованием функции Эйлера:*

$$a^{\varphi(m)-1}\equiv a^{-1}(\mathrm{mod}\,m).$$

Пример тот же: $x=9^{-1}(\mathrm{mod}\,5)$.

Решение: $a=9, m=5$. НОД$(9,5)=1$, $m=5$ – простое число.

Находим $\varphi(m)=\varphi(5)=5-1=4$;

$x=9^{4-1}(\mathrm{mod}\,5)=729(\mathrm{mod}\,5)=(145\cdot 5+4)(\mathrm{mod}\,5)=4$, т.е. $b=x=4$.

Получен тот же результат без перебора неподходящих вариантов k. Если в дальнейшем потребуется k как коэффициент диофантова уравнения, его можно будет определить из уравнения $a\cdot b-k\cdot m=1$.

Задание 1.4. Найти результат преобразования методом цепочек.

При больших значениях чисел $a^{n'}$ (задание 1.2) или $a^{\varphi(m)-1}$ (задание 1.3) определение вычетов по $(\mathrm{mod}\, m)$ может быть проведено с использованием метода цепочек модулярной арифметики:

$$(a \pm b) \bmod m = \big((a \bmod m) \pm (b \bmod m)\big) \bmod m;$$

$$(a * b) \bmod m = \big((a \bmod m) * (b \bmod m)\big) \bmod m;$$

$$\big(a * (b \pm c)\big) \bmod m = \big(((a * b) \bmod m) \pm ((a * c) \bmod m)\big) \bmod m.$$

Пример 1. Пусть требуется представить в форме цепочки $a^8 (\mathrm{mod}\, m)$. Степень четная, более того $8 = 2^3$.

Цепочка преобразований

$$(((a^2 (\mathrm{mod}\, m)^2 \bmod m)^2 \bmod m.$$

При использовании сложных чисел в выражении $a^n (\mathrm{mod}\, m)$ как в качестве a, так и в качестве степени n (n' или $(\varphi(m)-1)$) может быть использована факторизация чисел (разложение их на простые сомножители).

Пример 2. Найти методом цепочек $a^n (\mathrm{mod}\, m) = 15^{63} (\mathrm{mod}\, 7)$.

Решение. $a = 15 = 3 \cdot 5;\ n = 63 = 3^2 \cdot 7.$

$$15^{63} (\mathrm{mod}\, 7) = ((3^{63} \bmod 7) \cdot (5^{63} \bmod 7)) \bmod 7 = (A \cdot B) \bmod 7;$$

$$A = 3^{63} \bmod 7 = 3^{3 \cdot 3 \cdot 7} \bmod 7 = ((3^3 \bmod 7)^3 \bmod 7)^7 \bmod 7;$$

$$3^3 \bmod 7 = 27 \bmod 7 = (3 \cdot 7 + 6) \bmod 7 = 6;$$

$$6^3 \bmod 7 = 216 \bmod 7 = (30 \cdot 7 + 6) \bmod 7 = 6;$$

$$6^7 \bmod 7 = 6^{2 \cdot 2 + 3} \bmod 7 = ((6^{2 \cdot 2} \bmod 7) \cdot (6^3 \bmod 7)) \bmod 7;$$

$$6^{2 \cdot 2} \bmod 7 = (6^2 \bmod 7)^2 \bmod 7;\ 6^2 \bmod 7 = 36 \bmod 7 = (5 \cdot 7 + 1) \bmod 7 = 1;$$

$$1^2 \bmod 7 = 1;\ 6^7 \bmod 7 = (1 \cdot 6) \bmod 7 = 6;\ \ A = 6.$$

Аналогичными действиями для $5^{63} \bmod 7$ получаем $B = 6$.

$$15^{63} (\mathrm{mod}\, 7) = A \cdot B \bmod 7 = (6 \cdot 6) \bmod 7 = 36 \bmod 7 = (5 \cdot 7 + 1) \bmod 7 = 1.$$

В прикладной криптографии (при использовании двоичного кода представления целых чисел) особое значение имеет возможность построения на основе метода цепочек алгоритма повторного возведения в квадрат, который реализуется при выполнении заданий лабораторного практикума.

Обозначим промежуточный результат вычисления a. В конце работы алгоритма a примет значение наименьшего неотрицательного вычета $b^n (\mathrm{mod}\, m)$. Пусть $n = n_0 \cdot 2^0 + n_1 \cdot 2^1 + n_2 \cdot 2^2 + \ldots + n_{k-1} \cdot 2^{k-1}$, где n_j, $j = 0, 1, 2, \ldots, k-1$ – цифры двоичной записи числа n. Каждое n_j равно либо 1, либо 0. Принимаем начальное значение $a = 1$. Первые шаги алгоритма метода повторного возведения в квадрат представлены в табл. 1.6.

Таблица 1.6 – Первые шаги алгоритма метода

$j = 0$	$b_0 = b (\mathrm{mod}\, m)$	при $n_0 = 1 \Rightarrow a = b_0$
$j = 1$	$b_1 = b^2 (\mathrm{mod}\, m)$	при $n_1 = 1 \Rightarrow a = (a \cdot b_1) \bmod m$
$j = 2$	$b_2 = b_1^2 (\mathrm{mod}\, m)$	при $n_2 = 1 \Rightarrow a = (a \cdot b_2) \bmod m$
$j = 3$	$b_3 = b_2^2 (\mathrm{mod}\, m)$	при $n_3 = 1 \Rightarrow a = (a \cdot b_3) \bmod m$

Алгоритм продолжается для всех $j = 0, 1, 2, \ldots, k-1$. При $n_j = 0$ достигнутое значение a не меняется. На j-том шаге, получим $b_j = b^{2^j} (\mathrm{mod}\, m)$. Если $n_j = 1$, то есть – когда 2^j входит в двоичное представление числа n, поэтому используем b_j как множитель для вычисления нового значения a и не делаем этого при $n_j = 0$. После выполнения шагов по всем j, получим искомое $a = b^n (\mathrm{mod}\, m)$.

Пример 3. Найти $5^{17} \bmod 7$ методом повторного возведения в квадрат.

Получение бинарного («двоичного») представления числа 17 показано в табл. 1.7. Строки таблицы заполняются справа–налево.

Таблица 1.7 – Построение двоичного представления числа 17

Четное число без остатка					16
Делим на 2	1	2	4	8	17
Остаток (бинарное представление числа)	1	0	0	0	1
Позиция (разряд)	4	3	2	1	0

Проверка: $n = 1 \cdot 2^0 + 1 \cdot 2^4 = 1 + 16 = 17$.

Процедура «повторного возведения в квадрат» для определения $5^{17} \bmod 7$ представлена в табл. 1.8.

Таблица 1.8 – «Повторное возведение в квадрат» для $5^{17} \bmod 7$

j	n_j	b_j	a_j
0	1	$b_0 = 5(\bmod 7) = 5$	$a = b_0 = 5$
1	0	$b_1 = 5^2(\bmod 7) = 25 \bmod 7 = (3 \cdot 7 + 4) \bmod 7 = 4$	$a = a = 5$
2	0	$b_2 = 4^2(\bmod 7) = 16 \bmod 7 = (2 \cdot 7 + 2) 16 \bmod 7 = 2$	$a = a = 5$
3	0	$b_3 = 2^2(\bmod 7) = 4 \bmod 7 = 4$	$a = a = 5$
4	1	$b_4 = 4^2(\bmod 7) = 16 \bmod 7 = (2 \cdot 7 + 2) \bmod 7 = 2$	$a = (a \cdot b_4) \bmod m = (5 \cdot 2) \bmod 7 = 10 \bmod 7 = (7 + 3) \bmod 7 = 3$

Ответ: $5^{17} \bmod 7 = 3$. Результат совпадает с ответом примера, представленного в задании 1.2.

2.5. Практическая работа № 5 «Применение шифров гаммирования»

Задание:

- зашифровать свою фамилию с помощью шифров гаммирования по модулю N и модулю 2. При оформлении отчета необходимо привести исходное сообщение (фамилию), гамму и таблицы зашифрования/дешифрования;

- сгенерировать гамму с помощью регистра сдвига с линейной обратной связью (принять полином $x^8 + x^4 + x^3 + x^2 + 1$) и алгоритма Блюм - Блюм - Шуба. При оформлении отчета необходимо привести расчет исходного значения и таблицы генерации гамм для 10 итераций. Исходное значение определить сложением по модулю 2 всех букв фамилии в соответствии с кодировкой Windows 1251. Например, для фамилии "АБРАМОВ" расчет исходного значения для генераторов гамм будет выглядеть следующим образом.

1100 0000 А
1100 0001 Б
1101 0000 Р
1100 0000 А
1100 1100 М
1100 1110 О
1100 0010 В

11
01 0001 209_{10}

2.6. Практическая работа № 6 «Применение комбинированных шифров»

Задание:

Среди комбинированных методов шифрования наиболее распространенными являются методы блочного шифрования. **Блочное шифрование** предполагает разбиение исходного

открытого текста на равные блоки, к которым применяется однотипная процедура шифрования. В настоящее время блочные шифры широко используются на практике. Российский и бывший американский стандарты шифрования относятся именно к этому классу шифров.

DES (Data Encryption Standard, стандарт шифрования данных) - федеральный стандарт шифрования США в 1977-2001 годах **для использования во всех несекретных правительственных каналах связи** (FIPS PUB 46 «Data Encryption Standard»). Несмотря на то, что в настоящий момент федеральным стандартом шифрования США является Rijndael (AES - Advanced Encryption Standard, расширенный стандарт шифрования; тип – подстановочно-перестановочная сеть), рассмотрение DES позволяет понять основные принципы блочного шифрования.

В алгоритме, лежащем в основе DES, используются методы замены, перестановки и гаммирования (сложение по модулю 2).

Открытое сообщение разбивается на блоки длиной 64 бита. Если длина сообщения не кратна 64, оно дополняется справа недостающим количеством битов.

Данные шифруются ключом длиной 56 бит. На самом деле ключ имеет размер 64 бита, однако реально для выработки ключевых элементов используются только 56 из них. Самые младшие биты каждого байта ключа (8-ой, 16-ый, …, 64-ый) не попадают в ключевые элементы и служат исключительно для контроля четности. Требуется, чтобы сумма битов каждого байта ключа, включая контрольный, была четной.

Для решения разнообразных криптографических задач, разработаны **четыре рабочих режима**, реализующих DES:

- электронная кодовая книга ECB (Electronic Code Book);
- сцепление блоков шифра CBC (Cipher Block Chaining);
- обратная связь по шифртексту CPB (Cipher Feed Back);
- обратная связь по выходу OFB (Output Feed Back).

Режим ECB (электронная кодовая книга - Electronic Code Book).

Открытое сообщение разбивают на 64-битовые блоки. Каждый из них шифруют независимо с использованием одного и того же ключа шифрования.

Общая схема шифрования блока изображена на рис.22.

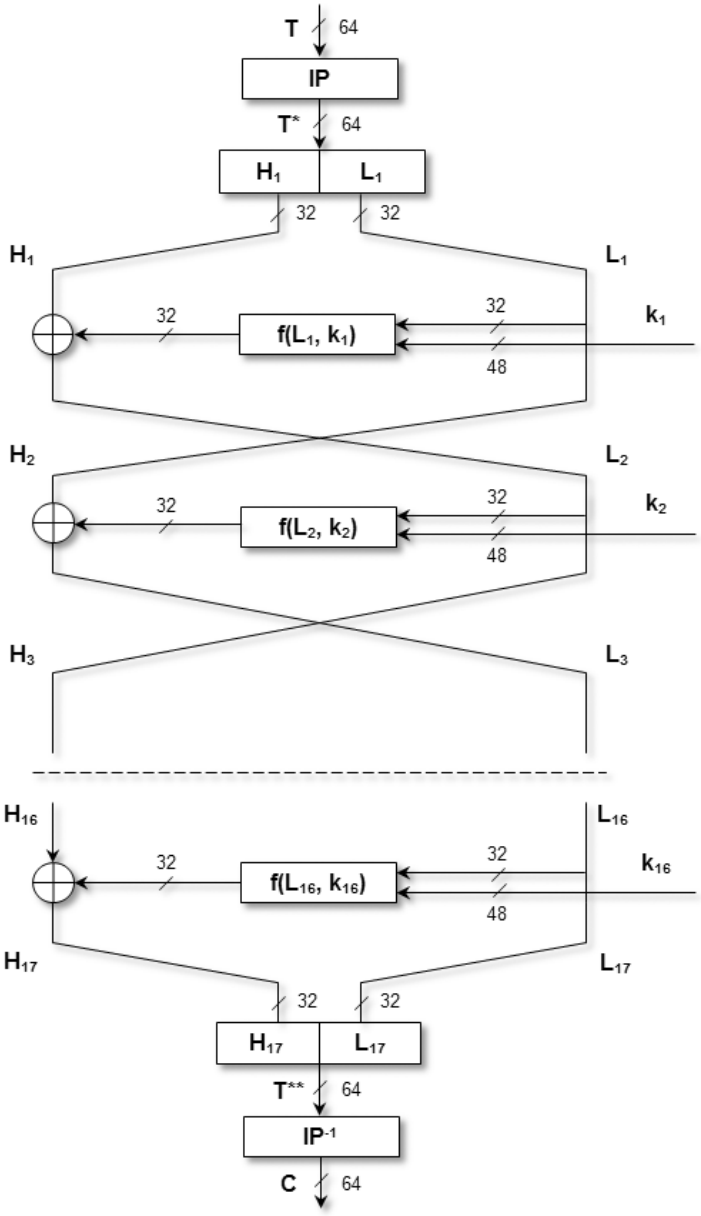


Рис.22. Схема шифрования блока

1. Шифрование 64-битового блока данных **T** начинается с начальной перестановки битов **IP**.

Таблица 5. Начальная перестановка IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

В таблице указывается новое положение соответствующего бита. Таким образом, при выполнении начальной перестановки 58-ый бит станет 1-ым, 50-ый – 2-ым, 42-ой – 3-им и т.д.

2. Результат перестановки **T*** разделяется на две 32-битовые части $\mathbf{H_1}$ и $\mathbf{L_1}$, с которыми выполняются 16 раундов преобразования.

3. В каждом раунде **i** старшая половина $\mathbf{H_i}$ блока модифицируется путем побитового прибавления к ней по модулю 2 ($\oplus$) результата вычисления функции шифрования **f**, зависящей от младшей половины блока $\mathbf{L_i}$ и 48-битового ключевого элемента $\mathbf{k_i}$. Ключевой элемент $\mathbf{k_i}$ вырабатывается из ключа шифрования. Между раундами старшая и младшая половины блока меняются местами. В последнем раунде происходит то же самое, за исключением обмена значениями половинок блока.

4. Полублоки $\mathbf{H_{17}}$ и $\mathbf{L_{17}}$ объединяются в полный блок **T****, в котором выполняется конечная битовая перестановка $\mathbf{IP^{-1}}$, обратная начальной.

Таблица 6. Конечная перестановка IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Результат последней операции и является выходным значением цикла шифрования – зашифрованным блоком **C**.

Все перестановки в таблицах **IP** и $\mathbf{IP^{-1}}$ подобраны разработчиками таким образом, чтобы максимально затруднить процесс расшифровки путём подбора ключа.

Схема функции шифрования **f** приведена на рис.23.

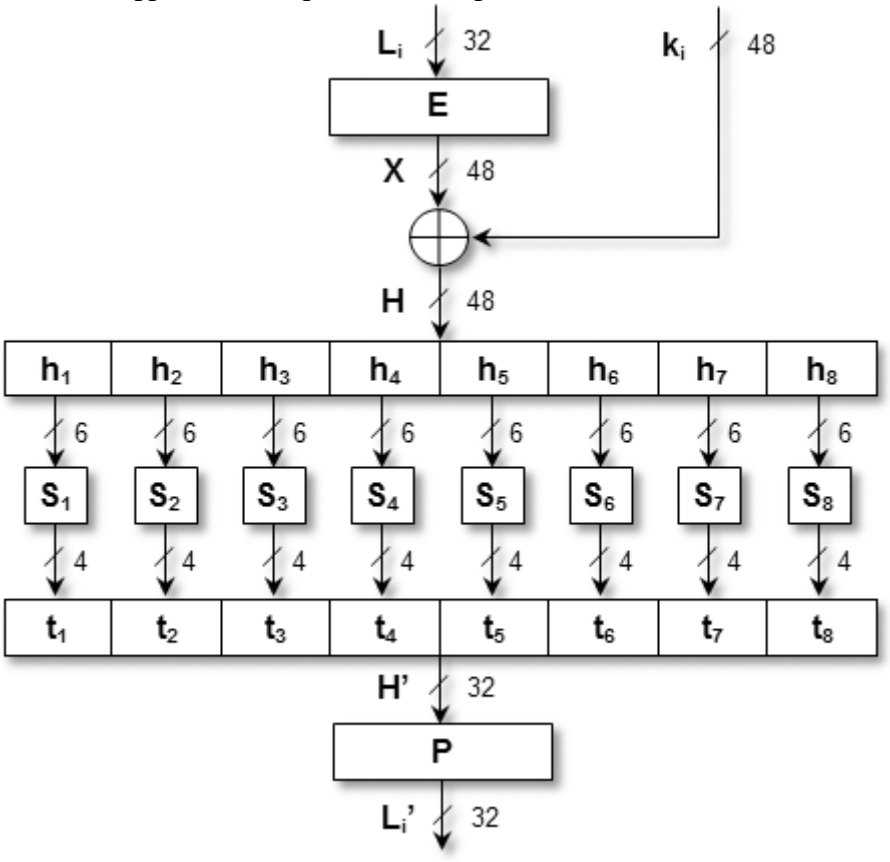


Рис.23. Схема функции шифрования

1. На вход поступает 32-битовая половина шифруемого блока $\mathbf{L_i}$ и 48-битовый ключевой элемент $\mathbf{k_i}$.

2. $\mathbf{L_i}$ разбивается на 8 тетрад по 4 бита. Каждая тетрада по циклическому закону дополняется крайними битами из соседних тетрад до 6-битного слова (функция расширения **E**). Цикличность означает, что первый бит $\mathbf{L_i}$ добавляется последним в последнее слово, а последний бит $\mathbf{L_i}$ добавляется первым в первое слово. Далее выполняется объединение тетрад в 48-битный блок **X**. Например, $\mathbf{L_i}$ = 0111 0110 1… …0 1101, тогда **X** = 101110 101101 … 011010.

3. **X** побитово суммируется по модулю 2 ($\oplus$) с ключевым элементом $\mathbf{k_i}$.

4. 48-битовый блок данных **H** разделяется на восемь 6-битовых элементов, обозначенных $\mathbf{h_1, h_2, \ldots, h_8}$.

5. Каждое из значений $\mathbf{h_j}$ преобразуется в новое 4-битовое значение $\mathbf{t_j}$ с помощью соответствующего узла замены $\mathbf{S_j}$.

Таблица 7. Узлы замен

	Номер столбца																
Номер строки	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S_2
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S_3
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S_4
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S_5
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S_6
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S_7
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S_8
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Если на вход $\mathbf{S_j}$ поступает блок $\mathbf{h_j=b_1b_2b_3b_4b_5b_6}$, то двухбитовое число $\mathbf{b_1b_6}$ указывает номер строки матрицы, а четырёхбитовое число $\mathbf{b_2b_3b_4b_5}$ - номер столбца в таблице узлов замен. В результате применения узла замены $\mathbf{S_j}$ к блоку $\mathbf{h_j}$ получается число (от 0 до 15), которое преобразуется в $\mathbf{t_j}$. Например, в узел замены S_3 поступает h_3=101011. Тогда, номер строки равен 3 (b_1b_6=11), номер столбца – 5 ($b_2b_3b_4b_5$=0101), t_3=1001 (9).

6. Полученные восемь элементов $\mathbf{t_j}$ вновь объединяются в 32-битовый блок **H'**.

7. В **H'** выполняется перестановка битов **P**.

Таблица 8. Перестановка P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Результат последней операции и является выходным значением функции шифрования $\mathbf{L_i'}$.

Ключевые элементы вырабатываются из ключа с использованием сдвигов и битовых выборок-перестановок. Таким образом, ключевые элементы состоят исключительно из битов исходного ключа, «перетасованных» в различном порядке. Схема выработки ключевых элементов показана на следующем рисунке.

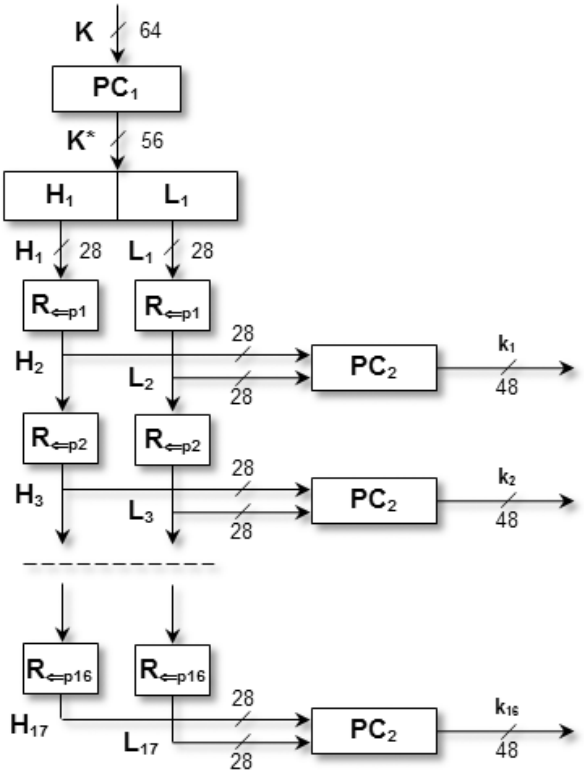


Рис.24. Схема выработки ключевых элементов

1. Выработка ключевых элементов из ключа **K** начинается со входной выборки-перестановки битов $\mathbf{PC_1}$, которая отбирает 56 из 64 битов ключа и располагает их в другом порядке.

Таблица 9. Перестановка PC_1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

2. Результат выборки-перестановки **K*** разделяется на две 28-битовые части: старшую $\mathbf{C_1}$ и младшую $\mathbf{D_1}$.

3. 16 раз выполняется процедура.

3а. В зависимости от номера итерации обе части циклически сдвигаются на 1 или 2 бита влево.

Таблица 10. Циклический сдвиг

Номер итерации	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг (бит)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3б. Из полученных блоков с помощью выходной битовой выборки-перестановки **PC_2** отбираются первые 48 битов, которые и формируют очередной ключевой элемент.

Таблица 11. Перестановка PC_2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Алгоритмы шифрования и расшифрования DES-ECB в общем виде выражаются следующими схемами

$$C = DES(T) = IP(T) \rightarrow H_1 \oplus f(L_1, k_1),\ L_1 \rightarrow \ldots \rightarrow H_{16} \oplus f(L_{16}, k_{16}),\ L_{16} \rightarrow IP^{-1}(2^{32}*L_{17}+H_{17}), \quad (8)$$

$$T = DES^{-1}(C) = IP(C) \rightarrow H_1 \oplus f(L_1, k_{16}),\ L_1 \rightarrow \ldots \rightarrow H_{16} \oplus f(L_{16}, k_1),\ L_{16} \rightarrow IP^{-1}(2^{32}*L_{17}+H_{17}). \quad (9)$$

Таким образом, для расшифрования необходимо «прогнать» DES с тем же ключом в обратном направлении.

Использование различных методов шифрования:

- замена - функция расширения **E**, узлы замены **S**;
- перестановка – перестановки **IP**, **IP^{-1}**, **P**, **PC_1**, **PC_2**, чередование **L_i** и **H_i**, циклический сдвиг;
- гаммирование – $\oplus$.

Из-за небольшого числа возможных ключей (всего 256), появляется возможность их полного перебора на быстродействующей вычислительной технике за реальное время. В 1998 году Electronic Frontier Foundation используя специальный компьютер DES-Cracker, удалось взломать DES за 3 дня. По неподтвержденным данным, Агентство национальной безопасности США уже в 1996 г. могло вскрывать ключ DES за 3-15 мин. с помощью устройства стоимостью 50000 долларов.

Задание на лабораторную работу.

В лабораторной работе необходимо зашифровать по алгоритму DES-ECB сообщение, состоящее из первых восьми букв своей фамилии. Если количество букв в фамилии меньше 8 букв, то необходимо добавить недостающее количество букв из имени. В качестве ключа выбрать первые 7 букв шифруемого сообщения.

При оформлении отчета необходимо привести:

- шифруемое сообщение (8 букв фамилии) в символьном и битовом представлении в соответствии с кодировкой Windows 1251 (табл.3);
- ключ (7 букв фамилии) в символьном и битовом представлении в соответствии с кодировкой Windows 1251 (табл.3);
- ключ в битовом представлении с учетом битов контроля четности;
- ключевые элементы k_i;
- результат начальной перестановки IP;
- полублоки H_i и L_i, $f(k_i, L_i)$, $H_i \oplus f(k_i, L_i)$;

- результат конечной перестановки IP^{-1}.

2.7. Практическая работа № 7 Метод шифрования с открытым ключом RSA

Задание:

Задание 1. Известны значения модуля шифрования N, открытого ключа e и открытого текста. Закодировать символы сообщения с помощью табл. 1 (буквы «е» и «ё» не различаются), а затем зашифровать сообщение по алгоритму RSA с помощью открытого ключа (N, e).

Таблица 1

Таблица кодирования символов открытого текста

Символ	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
Код	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Символ	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
Код	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42

1. Выбрать параметры шифра и открытый текст из табл. 2 в соответствии с номером варианта (от 1 до 5). Выполнить кодирование, разбиение на блоки и шифрование блоков текста аналогично рассмотренному ниже примеру.

Таблица 2

Варианты задания

Номер варианта	N	e	Открытый текст	Криптограмма У
1	2279	281	сон	18221993
2	2773	113	лес	13081874
3	1643	127	вид	381131
4	1517	193	сто	367712
5	1711	235	гол	18384

Пример 1

N= 1739, e = 653, требуется зашифровать по алгоритму RSA текст «май».

2. Подготовить открытый текст к шифрованию, закодировав его с помощью табл. 3.7: м — 23, а — 11, й - 20.

Получили открытое сообщение X = 231120.

3. Разбить открытый текст X на блоки $x_к$, такие, что $x_к<N$. В рассматриваемом примере N= 1739, поэтому сообщение X можно разбить на два блока — x_1= 231, x_2 = 120.

4. Теперь можно зашифровать блоки x_1 используя формулу $y_к = x_к \bmod N$. Для вычислений можно воспользоваться табличным процессором MSExcel. Подготовить реализацию алгоритма для быстрого вычисления степени по модулю последовательным возведением в квадрат с хранением промежуточных результатов:

- перевести значение степени e в двоичное представление. В среде MS Excel для этих целей можно воспользоваться функцией ДЕС.В.ДВ группы Инженерные.

Данная функция осуществляет перевод значений только в диапазоне от 512 до 511.
Если число e выходит за рамки указанного диапазона, следует воспользоваться стандартным приложением MS Windows Калькулятор, режим (вид) Программист. В этом случае следует установить переключатель системы счисления в позицию Dec (десятичная), ввести число e, а затем установить переключатель в позицию Bin (двоичная). Число будет переведено в двоичную систему счисления.
В примере e = 653 > 511, поэтому перевод в двоичную систему счисления осуществлен с помощью приложения Калькулятор: $e = 1010001101_2$.
Занести значение e в десятеричной и двоичной системах счисления на лист MS Excel;

- определить p — число разрядов двоичного представления числа e. В среде MS Excel для этих целей можно воспользоваться функцией ДЛСТР группы Текстовые. Пусть значение e в двоичной системе счисления занесено в ячейку A3. Тогда в ячейку A4 следует занести формулу =ДЛСТР(A3);
- теперь следует сформировать таблицу для вычисления степени e по модулю N. В ячейки столбца C занести значения от 0 до p - 1 (в примере - от 0 до 9), задав заголовок столбца — i;
- в соответствующие ячейки столбца D занести значения двоичных разрядов b_i (начиная с младшего разряда), для чего воспользоваться функцией ПСТР группы Текстовые. Если двоичное значение e находится

  в ячейке A3, число разрядов h занесено в ячейку A4, а значения i содержатся в ячейках C2:C11, формула в ячейке D2 примет вид: =ПСТР($A$3;$A$4-C2;1). Ссылки на значения e и г должны быть абсолютными (преобразовать ссылку щелкнув на ней мышью, а затем нажав кнопку F4). Скопировать сформированную формулу в диапазон ячеек столбца D (D3:D11 в примере) рис. 1;

D2 =ПСТР($A$3;$A$4-C2;1)

	A	B	C	D	E	F
1	*e*		*i*	b_i		
2	653		0	1		
3	1010001101		1	0		
4	10		2	1		
5			3	1		
6	*N*		4	0		
7	1739		5	0		
8	*x*		6	0		
9			7	1		
10			8	0		
11			9	1		
12						

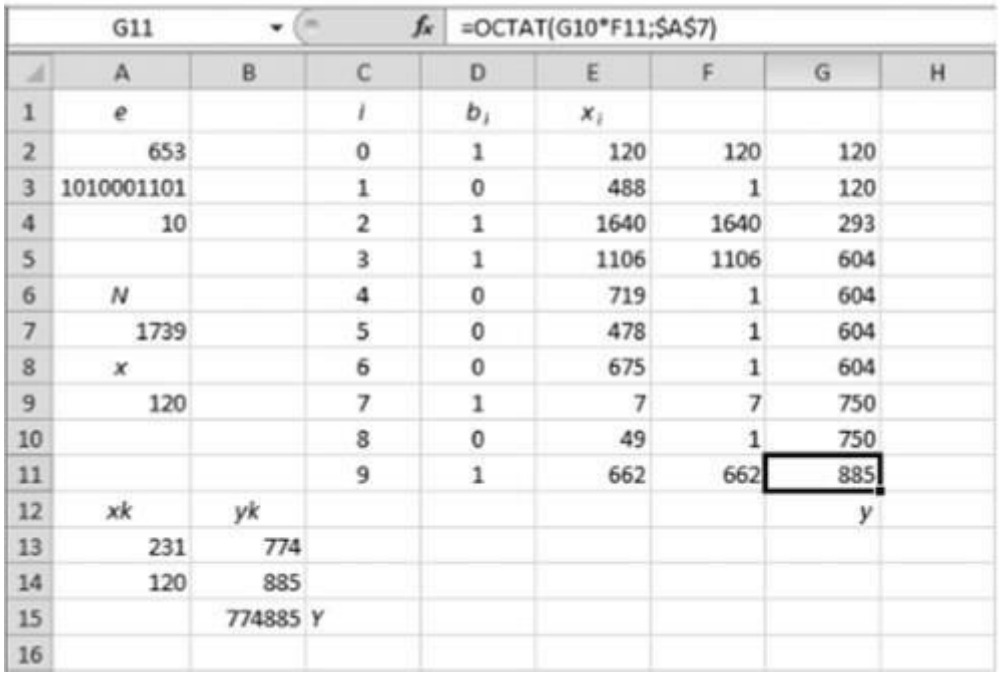
Рис. 1. Занесение на лист MS Excel разрядов числа e

- занести в первый столбец значение N (в примере — 1739). Пусть значение 1739 занесено в ячейку A7, ячейка A6 содержит соответствующую подпись. Тогда в ячейку A8 занести подпись x, значения блоков для шифрования будут заноситься в дальнейшем в ячейку A9;

- в ячейках столбца E вычислить значения ряда x_2 задав заголовок столбца X_j — в ячейку E2 занести формулу =A9, в ячейку E3 — формулу =ОСТАТ(E2^2;$A$7), ссылка на значение N должна быть абсолютной. Скопировать формулу на оставшийся диапазон ячеек столбца E (E4:E11 в примере);
- в ячейки столбца F занести значение «1», если соответствующее значение бита = 0 (находится в столбце D), или значением E2 (из столбца E), если $b_i = 1$. Для этих целей следует воспользоваться функцией ЕСЛИ группы Логические. Формула в ячейке F2 имеет вид: =ЕСЛИ(D2=“0”;1;E2). Значение бита является текстовым, поэтому заключается в двойные кавычки. Скопировать формулу на диапазон ячеек столбца F (F3:F11 в примере);
- в столбце G подсчитать произведение значений из столбца F по модулю. Для этого в ячейку G2 ввести формулу =F2, в ячейку G3 — формулу =ОСТАТ(G2*F3;$A$7). Ссылка на значение N должна быть абсолютной.

  Скопировать формулу на оставшийся диапазон ячеек столбца G (G4:G11 в примере);
- последняя заполненная ячейка столбца G (G11 в примере) содержит результат вычисления степени по модулю. Подписать эту ячейку как y.

5. Получить значения блоков шифротекста y_k, последовательно занося значения блоков $x_к$ в подготовленную для этого ячейку A9.



	A	B	C	D	E	F	G	H
1	*e*		*i*	b_i	x_i			
2	653		0	1	120	120	120	
3	1010001101		1	0	488	1	120	
4	10		2	1	1640	1640	293	
5			3	1	1106	1106	604	
6	*N*		4	0	719	1	604	
7	1739		5	0	478	1	604	
8	*x*		6	0	675	1	604	
9	120		7	1	7	7	750	
10			8	0	49	1	750	
11			9	1	662	662	885	
12	*xk*	*yk*					*y*	
13	231	774						
14	120	885						
15		774885	Y					
16								

Рис. 2. Вычисление блоков шифротекста

Значения блоков $x_к$ и полученные $y_к$ с подписями занести на лист (например, в диапазон ячеек A12:B14) - рис. 2.

Значения блоков шифротекста: $y_1 = 774$, $y_2 = 885$.

Ниже сформированных блоков шифротекста получить полное значение У, использовав операцию конкатенации &. В примере в ячейку B15 следует занести формулу =B13&B14 и подписать эту ячейку как У. Получена криптограмма У = 774885.

Задание 2. Криптограмма У получена RSA шифрованием на известном открытом ключе (N, e). Определить секретный ключ d и получить открытый текст, если кодирование символов сообщения осуществлялось с помощью табл. 1.
Выбрать значения открытого ключа (N, e) и криптограммы У из табл. 2 в соответствии с номером варианта (от 1 до 5). Выполнить дешифрование криптограммы по аналогии с рассмотренным ниже примером.

Пример 2

N = 1739, e = 653, требуется дешифровать RSA криптограмму
У = 12231108.

Рис. 3. Включение надстройки «Поиск решения»

В окне Надстройки установить флажок рядом с пунктом Поиск решения и нажать ОК (рис. 3);

- выбрать ячейку B2 (в которой подсчитано произведение двух множителей) и вызвать инструмент Поиск решения на вкладке Данные;
- в окне Поиск решения установить целевую ячейку $B$2 равной значению N (в примере — 1739), в поле Изменяя ячейки переменных выделить диапазон ячеек $A$1:$B$1, в группе В соответствии с ограничениями нажать кнопку Добавить, в окне Добавление ограничения в поле Ссылка на ячейку выделить диапазон ячеек $A$1:$B$1, в следующем поле выбрать значение цел и нажать ОК (рис. 4). Будет установлено ограничение $A$1:$B$1 = целое. Результирующий вид окна настроек инструмента Поиск решения показан на рис. 5;

Рис. 4. Задание ограничений на изменяемые ячейки

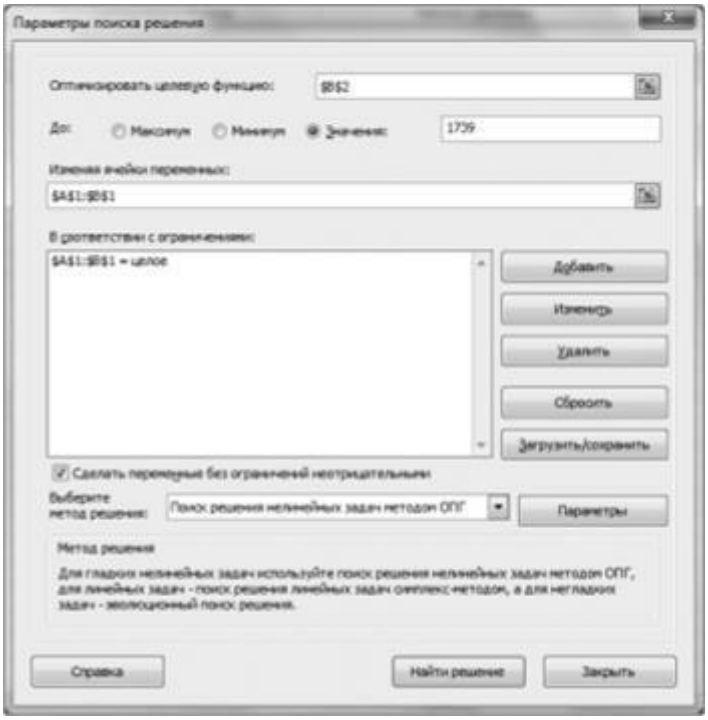


Рис. 5. Настройка инструмента Поиск решения

- в окне Поиск решения выбрать метод решения «Поиск решения нелинейных задач методом ОПГ», затем нажать кнопку «Параметры» и на вкладке «Все методы» установить Максимальное время — 1000 и Предельное число итераций - 10 000. Нажать ОК;
- после того как инструмент Поиск решения полностью настроен, в окне Поиск решения нажать кнопку Выполнить. Будет выдано окно Результаты поиска решения с сообщением о том, что решение найдено — установить переключатель в позицию «Сохранить найденное решение» и нажать ОК. В ячейках A1 и B1 будут получены значения простых множителей числа N.

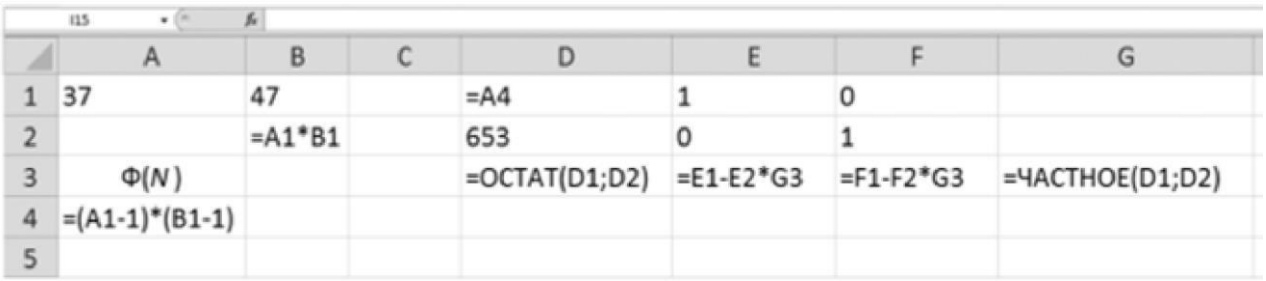
В рассматриваемом примере после выполнения поиска решения в ячейке A1 будет установлено значение 37, а в ячейке B1 — 47. Это и есть множители числа N = 1739. Примечание: если один из множителей получен равным единице, то следует изменить начальные значения в ячейках A1 и B1, а затем повторно выполнить поиск решения.

9. Получили: p = 37, q = 47. Поскольку оба числа простые, легко вычислить значение Φ(N) по формуле: Φ(N) = ϕ(p • q) = (p - 1) • (q - 1). Для вычисления значения Φ(N) занести в ячейку A4 формулу =(A1-1)(B1-1), в ячейку A3 занести подпись к значению. Получено Φ(N) = 1656.

10. Зная значение Φ(N) и e, можно вычислить секретный ключ d. Для вычисления d следует воспользоваться расширенным алгоритмом Евклида:

- сформировать первую строку (U) расширенного алгоритма Евклида: в ячейку D1 занести значение Φ(N) (1656 в рассматриваемом примере), в ячейку E1 —1, в ячейку F1 — 0;
- сформировать вторую строку (V) расширенного алгоритма Евклида: в ячейку D2 занести значение e (в примере — 653), в ячейку E2 — 0, в ячейку F2- 1;

- сформировать строку расширенного алгоритма Евклида: в ячейку G3 занести формулу =ЧАСТНОЕ(D1;D2), в ячейку D3 — формулу =ОСТАТ(D1;D2), в ячейку E3 — формулу =E1-E2*G3, в ячейку F3 — формулу: =F1-F2*G3 (рис. 6);



	A	B	C	D	E	F	G
1	37	47		=A4	1	0	
2		=A1*B1		653	0	1	
3	Φ(*N*)			=ОСТАТ(D1;D2)	=E1-E2*G3	=F1-F2*G3	=ЧАСТНОЕ(D1;D2)
4	=(A1-1)*(B1-1)						
5							

Рис. 6. Формулы для расчета по алгоритму Евклида

Результаты реализации расширенного алгоритма Евклида для рассматриваемого примера показаны на рис. 7. Получено значение d = 317.

	A	B	C	D	E	F	G	H
1	37	47		1656	1	0		
2		1739		653	0	1		
3	Φ(*N*)			350	1	-2	2	
4	1656			303	-1	3	1	
5				47	2	-5	1	
6	*d*			21	-13	33	6	
7	317			5	28	-71	2	
8				1	-125	317	4	
9				0	653	-1656	5	
10				#ДЕЛ/0!	#ДЕЛ/0!	#ДЕЛ/0!	#ДЕЛ/0!	
11				#ДЕЛ/0!	#ДЕЛ/0!	#ДЕЛ/0!	#ДЕЛ/0!	
12				#ДЕЛ/0!	#ДЕЛ/0!	#ДЕЛ/0!	#ДЕЛ/0!	
13								

Рис. 7. Пример реализации расширенного алгоритма Евклида

11. Подготовить последовательность Y к расшифрованию, разбив ее на части $y_к$ таким образом, что $y_к<N$, $y_к$ не содержит ведущих нулей. В рассматриваемом примере N = 1739, Y = 12231108, Y может быть разбито на два блока — $y_1 = 1223$, $y_2 = 1108$.

12. Аналогично п. 4 задания 1 подготовить реализацию алгоритма быстрого вычисления степени d по модулю N для дальнейшего определения блоков открытого текста по формуле $x_k = y_к \bmod N$:

- перевести значение степени d в двоичное представление, занести его в ячейку A8. В рассматриваемом примере d = 317 < 511, поэтому можно воспользоваться функцией MS Excel ДЕС.В.ДВ группы Инженерные, тогда в ячейку A8 можно занести формулу =ДЕС.В.ДВ(A7). Получено значение 100111101;

- определить p — число разрядов двоичного представления числа d с помощью функции ДЛСТР, поместить результат в ячейку А9. Получено p = 9;
- занести в ячейку А10 подпись У, в ячейку А2 — подпись N;
- в столбцах I — M сформировать таблицу для вычисления степени d по модулю N: задать заголовки столбцов I, J и К (i, b_i, y_i); в ячейки столбца I занести значения от 0 до 8; в ячейку J2 занести формулу
  =ПСТР($A$8;$A$9-I2;1), в ячейку К2 — =А11, в ячейку КЗ — =ОСТАТ(К2^2;$B$2), в ячейку L2 — =ЕСЛИ(J2="0";1;K2), в ячейку М2- =L2, в ячейку М3— формулу =ОСТАТ(М2*L3;$B$2). Скопировать последние формулы каждого столбца на оставшийся диапазон ячеек столбца;
- последняя заполненная ячейка столбца М (М10 в примере) содержит результат вычисления степени по модулю. Подписать эту ячейку как x.

13. Получить значения блоков открытого текста $x_к$, последовательно занося значения блоков $y_к$ в подготовленную для этого ячейку А11. Значения блоков $y_к$ и полученные $x_к$ с подписями занести на лист (например, в диапазон ячеек А13:В15) (рис. 8).

Значения блоков открытого текста: x, = 283, x2 = 827.

14. Ниже сформированных блоков открытого текста получить полное значение X, использовав операцию конкатенации &. В примере в ячейку В16 следует занести формулу =В14&В15 и подписать эту ячейку как X. Получено числовое представление открытого текста X = 283827.

M10 | =ОСТАТ(M9*L10;$B$2)

	A	B	C	H	I	J	K	L	M	N
1	37	47			*i*	b_i	y_i			
2	*N*	1739			0	1	1108	1108	1108	
3	*Φ(N)*				1	0	1669	1	1108	
4	1656				2	1	1422	1422	42	
5					3	1	1366	1366	1724	
6	*d*				4	1	9	9	1604	
7	317				5	1	81	81	1238	
8	100111101				6	0	1344	1	1238	
9	9				7	0	1254	1	1238	
10	*y*				8	1	460	460	827	
11	1108								*x*	
12										
13	*yk*	*xk*								
14	1223	283								
15	1108	827								
16		283827	*X*							
17										

Рис. 8. Вычисление блоков открытого текста

15. Разбить X на двузначные числа и провести обратное преобразование чисел в символы языка по табл. 1.

28 - с, 38 - ы, 27 - р.
Получен открытый текст «сыр».

16. Полученный файл MS Excel показать преподавателю.

2.8. Практическая работа № 8 Расчет основных параметров локальной сети

Задание:

В практической работе необходимо по алгоритму MD5 получить хеш-образ сообщения, состоящего из первых трех букв своей фамилии.

При оформлении отчета необходимо привести:

- теоретическую часть, включающую "Шаг основного цикла вычисления хеша", "Шаг цикла раунда" и таблицу "Раундовые функции RF";
- исходное сообщение (3 буквы фамилии) в символьном и десятичном представлениях в соответствии с кодировкой Windows 1251;
- прообраз сообщения в шестнадцатеричном представлении (512 бит), включая вспомогательные единичный и нулевые биты, а также биты, определяющие длину сообщения;
- исходные значения переменных A, B, C и D в шестнадцатеричном представлении (по 32 бита);
- для 1-го, 2-го и 16-го 32-битового блока прообраза - результаты вычислений переменных A, B, C и D в шестнадцатеричном представлении для всех раундов;

Раунд	Итерация	Значения переменных							
		в начале итерации				в конце итерации			
		A	B	C	D	A	B	C	D
1	1								
	2								
	...								
	16								
2	1								
	2								
	...								
	16								
3	1								
	2								
	...								
	16								
4	1								
	2								
	...								
	16								

- результат итогового сложения по модулю 2^{32} исходных значений переменных A, B, C и D со значениями этих переменных, полученных после 4-го раунда в шестнадцатеричном представлении (128 бит) до и после перестановки байт.

2.9 Практическая работа № 9 Использование шифросистемы Эль-Гамаля

Задание:

Генерация ключей

1. Генерируется случайное простое число p.
2. Выбирается целое число g — первообразный корень p.
3. Выбирается случайное целое число x такое, что 1 x p .
4. Вычисляется y g x mod p .
5. Открытым ключом является тройка (*p, g, y*), закрытым ключом — число x.

Шифрование

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

Выбирается сессионный ключ — случайное целое число k такое, что 1 k p 1

Вычисляются числа a g k mod p и b y M k mod p.

Пара чисел (*a, b*) является шифротекстом.

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

Расшифровывание

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

M b a x 1 mod p

При этом нетрудно проверить, что

a x 1 g kx (mod p) и поэтому b a x 1 (y

M k)g kx g M g xk xk M mod p .

Для практических вычислений больше подходит следующая формула:

M b a x 1mod p ba p 1 x mod p

Пример:

Шифрование

Допустим, что нужно зашифровать сообщение*M* 5.

Произведем генерацию ключей:

Пусть p 11, g 2. Выберем x 8 - случайное целое число x такое, что 1 x p.

Вычислим y g x mod p 28 mod11 3 .

Итак, открытым ключом является тройка p,g, y 11,2,3 , а закрытым ключом - число x 8.

Выбираем случайное целое число k такое, что $1 < k < p-1$. Пусть $k = 9$.

Вычисляем число a $a = g^k \bmod p = 29 \bmod 11 = 512 \bmod 11 = 6$.

Вычисляем число b y $b = M^k \bmod p = 35^9 \bmod 11 = 19683 \bmod 11 = 9$.

Полученная пара $(a, b) = (6,9)$ является шифротекстом.

Расшифрование

Необходимо получить сообщение $M = 5$ по известному шифротексту $(a, b) = (6,9)$ и закрытому ключу $x = 8$.

$$M = b \cdot a^{p-1-x} \bmod p = 9 \cdot 6^{11-8} \bmod 11 = 5$$

Вычисляем M по формуле:

Получили исходное сообщение $M = 5$.

Так как в схему Эль-Гамаля вводится случайная величина k ,то шифр Эль-Гамаля можно назвать шифром многозначной замены. Из-за случайности выбора числа k такую схему еще называют схемой вероятностного шифрования. Вероятностный характер шифрования является преимуществом для схемы ЭльГамаля, так как у схем вероятностного шифрования наблюдается большая стойкость по сравнению со схемами с определенным процессом шифрования. Недостатком схемы шифрования Эль-Гамаля является удвоение длины зашифрованного текста по сравнению с начальным текстом. Для схемы вероятностного шифрования само сообщение M и ключ не определяют шифротекст однозначно. В схеме Эль-Гамаля необходимо использовать различные значения случайной величины k для шифровки различных сообщений

M и M’. Если использовать одинаковые k, то для соответствующих шифротекстов (ab) и $(a' b')$ выполняется соотношение $b b' \equiv M M' \bmod 11$. Из этого выражения можно легко вычислить M', если известно M.

4. ВАРИАНТЫ ЗАДАНИЙ

№	Исходный текст
1	Шумит дубравушка к непогодушке
2	Утром вороны каркают к дождю
3	Сорока на хвосте принесла
4	Снег холодный, а от мороза укрывает
5	Сирень или берèза, а всè дерево
6	Сегодня не тает, а завтра кто знает
7	Розы без шипов не бывает
8	Не высок лесок, а от ветра защищает
9	На всех и солнышко не светит
10	Красна ягодка, да на вкус горька
11	В осеннее ненастье семь погод на дворе
12	Ветром ветра не смеряешь
13	Пропущенный час годом не нагонишь

14	Счастливые часов не наблюдают
15	Друг неиспытанный, как орех не расколотый
16	Дружи с теми, кто лучше тебя самого
17	Крепкую дружбу и топором не разрубишь
18	Кто друг прямой, тот брат родной
19	лучше выслушать упрèки друга, чем потерять его
20	Одна пчела много мèду не принесèт
21	С тем не ужиться, кто любит браниться
22	Старый друг лучше новых двух
23	На чужой сторонушке рад родной воробушке
24	Народы нашей страны дружбой сильны
25	Поднявший меч от меча и погибнет
26	При солнце тепло, при Родине добро
27	Старая слава новую любит
28	Любишь кататься - люби и саночки возить
29	Кто пахать не ленится, у того хлеб родится
30	На печи не храбрись, а в поле не трусь

2.10 Практическая работа № 10 Применение бесключевого протокола Шамира

Задание:

Обмен информацией с использованием протокола Шамира

Краткое изложение основ и правил применения протокола Шамира с привязкой к практическим аспектам рассматриваются при проведении практического занятия на конкретном примере. Алгоритм и расчетные формулы подробно рассматриваются в разделе теоретического обучения (либо на лекционных занятиях, либо при выполнении СРС).

***Пример**. Дано*: $p_A = 38177$, $p_B = 52631$,$M =$"*XYZ*".

Этап 1. Генерация ключей

Абонент *A*: выберем значение $e_A = 51407$ по *НОД*(e p_A, −1) =1. Находим условию значение d_A:

$d_A = e_A^{-1} (\text{mod}\ p_A - 1) = 51407^{-1} (\text{mod}(51407 - 1)) = 24335.$

Абонент *B*: выберем значение $e_B = 42239$ по *НОД*(e p_B, −1) =1. Находим условию значение d_B:

$d_B = e_B^{-1} (\text{mod}\ p_B - 1) = 42239^{-1} (\text{mod}(52631\ 1)) - = 32229.$ *Закрытый ключ абонента A*: $e_A = 51407$.

Закрытый ключ абонента B: $e_B = 42239$.

Этап 2. Преобразование сообщения в числовой эквивалент

Преобразуем передаваемую триграмму “*XYZ*” в числовой эквивалент для последующей обработки $M =$"*XYZ*"$= 23\ 26\cdot\ ^2 + 24\ 26\cdot\ ^1 + 25\ 26\cdot\ ^0 = 16197$.

Этап 3. Трехпроходный алгоритм Шамира

Используя обозначения, используемые в протоколе Шамира, введем значения α и β: $\alpha = e_A = 51407$; $\beta = e_B = 42239$. Выполним 1-й *шаг* алгоритма Шамира:

$C_1 = E_\alpha(M) = M^{eA} (\text{mod}\ p_A) = 16197^{51407} (\text{mod}38177) = 30944.$ Выполним 2-й *шаг* алгоритма Шамира:

$C_2 = E\ C_\beta(\ _1) = C_1^{eB} (\mathrm{mod}\ p_B) = 30944^{42239} (\mathrm{mod} 52631) = 34848$. Выполним 3-й *шаг* алгоритма Шамира:

$$C_3 = D_\alpha(C_2) = C_2^{dA} (\mathrm{mod}\ p_A) = 34848^{24335} (\mathrm{mod} 28177) = 14648.$$

Вычислим передаваемое абоненту *B* значение из выражения:

$$M = D_\beta(C_3) = D_\beta(E_\beta(^M)):$$

$$M = D\ C_\beta(\ _3) = C_3^{dB} (\mathrm{mod}\ p_B) = 14648^{34229} (\mathrm{mod} 52631) = 16197.$$

Этап 4. Преобразование полученного значения к формату текста

$$M = 216197 = 3\ 26\cdot\ ^2 + 24\ 26\cdot\ ^1 + 25\ 26\cdot\ ^0 = "XYZ".$$

Полученное значение: *“XYZ*”.

2.11 Практическая работа № 11 Применение электронной подписи (ГОСТы 34.10-94 и 34.10-2001)

Задание:

В лабораторной работе необходимо привести последовательность выполнения процедур генерации и проверки ЭЦП с использованием следующих способов:

- на базе алгоритма RSA;
- по ГОСТ 34.10-94;
- по ГОСТ 34.10-2001.

При оформлении отчета необходимо привести таблицы генерации ключей, отправки сообщения с ЭЦП и получения сообщения с ЭЦП. В качестве хеш-образа исходного сообщения h(T) принять коды, соответственно, 1-ой, 2-ой и 3-ей буквы своей фамилии согласно их положению в алфавите.

2.12 Практическая работа № 12 Настройка ПО для работы с электронной подписью

Задание:

Практическая работа выполняется учащимися в паре для полноценного обмена ключами, зашифрованными и подписанными сообщениями. Каждый из учащихся в паре работает на компьютере с установленной системой GnuPG для Windows. Компьютеры должны быть объединены в сеть для оперативного обмена файлами.

Порядок работы каждого из учащихся в паре.

- Создайте пару ключей в менеджере ключей Cleopatra.
- Экспортируйте сертификат открытого ключа из своей пары ключей в файл и передайте его своему напарнику.
- Получив файл с экспортированным ключом от напарника, импортируйте его в менеджер ключей. Установите для импортированного ключа полное доверие.
- Зашифруйте с использованием импортированного ключа напарника произвольный текст на диске. Передайте зашифрованный текст напарнику.
- Получив зашифрованный файл от напарника, дешифруйте его своим закрытым ключом. Убедитесь, что файл был успешно дешифрован.
- Используя свой закрытый ключ, подпишите произвольный файл на диске электронной подписью. Передайте подписанный документ напарнику.
- Получив от напарника документ с подписью, убедитесь, что подпись верна.

Отчет по практической работе должен содержать следующие сведения:

- ✓ Название и цель работы;
- ✓ Экранные формы основных этапов работы с системой и описание к ним;

Ответьте на контрольные вопросы

2.13 Практическая работа № 13 Изучение частотного метода криптоанализа симметричных криптосистем

Задание:

Криптосистема Цезаря определяется выражением:

$$y_i = (x_i + k) \bmod m,\ i = \overline{1,n},$$

где y_i - буква криптограммы, x_i - буква открытого сообщения, k - ключ шифра, n - длина криптограммы (открытого текста), m - мощность алфавита. Выражения для расшифрования имеет вид:

$$x_i = (y_i - k) \bmod m.$$

Метод частотного криптоанализа базируется на реализации методов теории статистических решений, а именно, на методе максимального правдоподобия [4]. В соответствии с этим методом оценкой ключа шифра k^* является такое его значение, которое доставляет максимальное значение логарифму функции правдоподобия $l(k)$. Для криптосистемы Цезаря оценка формируется в соответствии с выражением:

$$k^* = \arg\max_k l(k),\ l(k) = \sum_{j=0}^{m-1} \nu_{(j+k) \bmod m} \log p_1(j), \quad (1)$$

где $p_1(j)$ - оценка вероятности встречаемости j -й буквы алфавита мощности m в открытых текстах, ν_j - частость встречаемости j -й буквы в криптограмме.

Выражение (1) справедливо, если источник открытых сообщений представляет собой стационарный источник дискретных сообщений без памяти. В случае, когда источник открытых сообщений представляет собой однородную цепь Маркова, оценка ключа будет определяться в соответствии с выражением:

$$k^* = \arg\max_k \left\{ \sum_{j=0}^{m-1} \delta_{y_1,(j+k) \bmod m} + \sum_{j,s=0}^{m-1} \nu_{} + \max \log p_1(j) \right\}$$

2 Порядок выполнения работы

2.1 При подготовке к лабораторной работе

На этапе подготовки к лабораторной работе студенты должны, используя литературу [1,2,3,4] и материалы лекций углубить свои знания по криптосистеме Цезаря и частотному методу криптоанализа простейших шифров.

Студенты на предстоящее лабораторное занятие готовят русский и английский алфавиты со значениями вероятностей встречаемости букв.

2.2 Во время проведения занятия.

Преподаватель перед проведением занятия проводит контрольный опрос студентов и определяет степень их готовности к лабораторной работе. Затем преподаватель разбивает группу студентов на несколько подгрупп по два студента в каждой.

Каждая подгруппа получает от преподавателя индивидуальный вариант задания на лабораторную работу, который представляет собой криптограмму, зашифрованную с помощью криптосистемы Цезаря.

Студенты должны:

1. Определить частотные характеристики криптограммы, для чего рассчитать значение частоты встречаемости символов $j \in A_m$ в криптограмме.

2. Определить вероятностные характеристики алфавита, для чего вычислить значение логарифма вероятности встречаемости символа $\log p_1(j)$ для заданного алфавита.

3. Полученные значения свести в таблицу 1.

Таблица 1.

Буква	А	Б	…	Ю	Я
$j \in A_m$					
$\log p_1(j)$					
$\nu_j(Y)$					

4. В соответствии с выражением (1) определить значение логарифма функции правдоподобия $l(K)$ и построить соответствующую графическую зависимость.

5. Определить в соответствии с выражением (1) оценку ключа k^*. 6. Дешифровать заданную криптограмму, используя оценку ключа k^*. При получении осмысленного текста подготовить отчет и представить его преподавателю.

4 Содержание отчета

Отчет должен включать в себя следующие пункты:

1. Задание на выполнение лабораторной работы (исходную криптограмму).
2. Основные расчетные соотношения.
3. Результаты расчетов, сведенные в табл. 1.
4. Графическую зависимость $l(k)$ и значение оценки ключа k^*.
5. Полученный дешифрованием открытый текст.

2.14 Практическая работа № 14 Изучение методов криптоанализа криптосистем гаммирования с периодической гаммой

Задание:

Криптосистема Виженера представляет собой шифр гаммирования с использованием периодической гаммы малого периода. В криптосистеме Виженера ключ $k^d \in$

набором из d символов. Такие наборы подписываются под открытым текстом $x_1, x_2,..., x_n$, $x_i \in A_m$, до получения

периодической ключевой последовательности $\tilde{k} = k_1, k_2,..., k_n$, $n = sd + r$, где s - число полных периодов $\bar{k}^d$, $r = n \bmod d$.

Уравнение шифрования для криптосистемы Виженера:

$$y_i = (x_i + k_i) \bmod m.$$

При дешифровании криптосистемы Виженера решаются две взаимосвязанные задачи:

- задача определение периода d ключевой последовательности $\tilde{k}$;
- задача дешифрования криптограммы Y при известном периоде d длине ключевой последовательности $\tilde{k}$.

Основным инструментом решения задачи определения периода ключевой последовательности криптосистемы Виженера являются методы Фридмана, основанные на понятии индекса совпадения. Индексом совпадения последовательности $X = x_1, x_2,..., x_n$ называется величина

$$I(X) = \sum_{i=1}^{m} \frac{F_i(F_i - 1)}{n(n-1)}, \quad (1)$$

где $X \in A_m$ - некоторая последовательность; F_i - частота встречаемости (число мест в тексте) i - буквы в последовательности X. Для криптосистемы Виженера, получаемого шифрованием открытого текста $X = a_1, a_2,..., a_n$ с помощью равновероятного выбора ключа k из множества всех локально-периодических последовательностей K_n^d периода d и $n = sd + r$ справедливо

$$M(I) = \frac{(s-1)r}{n(n-1)} + \frac{s(s-1)(d-r)}{n(n-1)} + \sum_i p_i^2 \left(\frac{(s-1)r}{n(n-1)} + \frac{s(s-1)(d-r)}{n(n-1)}\right). \quad (2)$$

Первый метод Фридмана состоит в том, что вычисляется индекс совпадения $I(Y)$ для имеющейся криптограммы в соответствии с выражением (1) и затем его значение сравнивается с (2) при $d = 1,2,3,...$. При достаточной близости индекса совпадения к одному из значений (2), при некотором d, предполагают, что период равен этому значению d. Первый метод Фридмана эффективен для $d \le 5$, т.к. значение $M(I)$ для фиксированного периода d совпадает со значениями целого ряда различных периодов ключевой последовательности.

Суть *второго метода Фридмана* состоит в опробовании возможных периодов d по следующей схеме. Из исходной криптограммы $Y = y_1, y_2, ..., y_n$ для предполагаемого периода d ключевой последовательности выписывается d подпоследовательностей:

1) $y_1, y_{1+d}, y_{1+2d}, ...$

2) $y_2, y_{2+d}, y_{2+2d}, ...$

.....

d) $y_d, y_{d+d}, y_{d+2d}, ...$

Для каждой подпоследовательности подсчитывается ее индекс совпадения (Y_d). Если все индексы совпадения в среднем близки к значению

$\frac{1}{N}\sum_{i=1}^{N} p_i^2$ (среднее значение индекса совпадения случайных криптограмм, d_i полученных с помощью гамм периода 1), то принимают величину d за истинный период, в противном случае опробуется следующая величина периода. Второй метод Фридмана позволяет эффективно определять периоды $d \le 30$.

Метод «протяжки» вероятного слова. При известном периоде ключевой последовательности d выписываются две подпоследовательности исходной криптограммы:

$y_1, y_2, ..., y_{(k-1)d+r}; \quad y_i, ... y$

$y_{1+d}, y_{2+d}, ..., y_{i+d}, ..., y_{kd+r}, \quad n = kd + r.$ y Формируется называемое «множество вероятных слов», которые, по мнению криптоаналитика, могут быть началом искомого открытого текста.

Для слова $a_1^*, a_2^*, ..., a_r^*$ из этого множества, находятся первые символы ключевой последовательности $k_1^*, k_2^*, ..., k_r^*$. Правильность угадывания вероятного слова, а, следовательно, и первых символов ключевой последовательности, проверяется на «читаемость» следующего фрагмента расшифрованного текста

$f^{-1}(y_{1+d}), f^{-1}(y_{2+d}), ..., f^{-1}(y_{r+d}).$ $k_1^* k_2^* ... k_r^*$

Если полученная последовательность символов «читаемая», то полагают, что $k_1, k_2, ..., k_r = k_1^*, k_2^*, ..., k_r^*$, т.е. найдены первые r символов ключевой последовательности. Далее анализируя фрагменты расшифрованной криптограммы, ищут возможные продолжения открытого текста, таким образом, чтобы получить недостающую часть ключевой последовательности $k_{r+1}, k_{r+2}, ..., k_d$ $k_{r+1}, k_{r+2}, ..., k_d$.

После того, как определен весь ключ, оставшаяся часть криптограммы расшифровывается на найденном ключе. Если же последовательность символов принимается как случайная (т.е. «нечитаемая»), то опробуется следующее вероятное слово. Вероятные слова могут выбираться также из предположения, что они является окончанием открытого текста или, в общем случае, располагаются на других позициях открытого текста.

Метод чтения в колонках. Рассмотрим два случая:

- априорные вероятности символов ключевой последовательности не известны;
- априорные вероятности символов ключевой последовательности известны.

Априорные вероятности символов ключевой последовательности не известны.

Пусть дана криптограмма $Y = y_1, y_2, ..., y_n$. Известен период ключевой последовательности d. Сформируем две подпоследовательности исходной криптограммы

$$y_1, y_2, ..., y_{(k-1)d+r}; \qquad y_i, ... y_{kd+r}, \quad n = kd + r.$$

$$y_{1+d}, y_{2+d}, ..., y_{i+d}, ..., y_{kd+r},$$

Будем полагать, что открытыми текстами, подлежащими шифрованию, являются содержательные тексты с известными вероятностями букв алфавита $P(a_j) = p_j, j = \overline{1,m}$, где j - порядкоый номер буквы алфавита. Также будем считать, что на множестве K задано равномерное распределение, т.е. ключом является реализация выборки объемом d из равномерного распределения K. Тогда вероятность того, что i -я и $(i+d)$-я буквы открытого текста были равны соответственно, s и l, при условии, что i -я и $(i+d)$-я буквы криптограммы равны соответственно, y_i и y_{i+d} определяется выражением

$$P(x_i = s, x_{i+d} = l \mid y_i, y_{i+d}) = \frac{P(x_i = s, x_{i+d} = l; y_i, y_{i+d})}{P(y_i, y_{i+d})}.$$

Если числитель не равен нулю, то справедливо равенство

$$P(x_i = s, x_{i+d} = l \mid y_i, y_{i+d}) = \frac{p_s p_l}{\sum_{f \in K} p_{f^{-1}(y_i)} p_{f^{-1}(y_{i+d})}} = \tilde{p}_{sl}.$$

Для каждой пары y_i и y_{i+d} букв исходной криптограммы упорядочим в соответствии с убыванием полученных значения условных вероятностей (5) пары букв открытого текста s и l. Построив такие колонки для каждого i, в результате получаем таблицу (табл. 1), в которой верхние пары имеют большую условную вероятность, чем нижние.

Таблица 1.

$i=1$	…	$i=j$	…	$i=n$
$\frac{y_1}{y_{1+d}}$	…	$\frac{y_j}{y_{j+d}}$	…	$\frac{y_{(k-1)d+r}}{y_{kd+r}}$
….	…	$\left(\frac{x_j = s}{x_{j+d} = l}, \tilde{p}_{sl}\right)$	…	…

Буквы искомого содержательного текста будут находится вероятнее всего в первых строках и задача сводится к подбору таких пар букв, чтобы в результате получался осмысленный текст.

Априорные вероятности символов ключевой последовательности известны. Если априорно известны вероятности символов ключевой последовательности, то задача дешифрования криптограммы аналогична задаче восстановления текста, зашифрованного неравновероятной гаммой. Пусть p_i , $i\ 1,m$, есть вероятность использования символа в ключевой последовательности. Рассмотрим простую задачу, когда некоторые символы вовсе не встречаются в ключевой последовательности. Положим, что

$$p_1 \geq p_2 \geq \ldots \geq p_l, p_{l+1} = p_{l+2} = \ldots = p_m = 0, l < m.$$

Составим таблицу (см. таблицу 2), в которой по строкам расположены символы, полученные путем расшифровки криптограммы одним символом ключевой последовательности k_i , $i\ 1,l$. Задача заключается в подборе по столбцам символов таким образом, чтобы в результате получился осмысленный текст. Если нельзя исключить использования ни одного символа в ключевой последовательности, то тогда поступают следующим образом. Для составления таблицы исключают из рассмотрения ml наименее вероятных букв алфавита, дальнейшие действия аналогичны рассмотренным выше. Надежность такого метода меньше, так как не исключена возможность частичной, а может и полной, потери истинного открытого текста.

Таблица 2.

ki / yi	y_1	….	….	y_n
k_1	$y\hat{}_1(k_1)$	….	…..	$y\hat{}_n(k_1)$
k_2	$y\hat{}_1(k_2)$	….	….	$y\hat{}_n(k_2)$
….	….	….	….	….
k_l	$y\hat{}_1(k_l)$	….	….	$y\hat{}_n(k_l)$

2 Порядок выполнения работы

2.1. При подготовке к лабораторной работе

На этапе подготовки к лабораторной работе студенты должны, используя литературу [1,2,3,4], материалы лекций углубить свои знания по следующим вопросам: криптосистема Виженера; методы определения периода ключевой последовательности; методы бесключевого чтения (метод «протяжки» вероятного слова, метод чтения в колонках).

Студенты на предстоящее лабораторное занятие готовят алфавиты русского и английского языка со значениями вероятностей встречаемости символов.

2.2. Во время проведения занятия

Преподаватель перед проведением занятия проводит контрольный опрос студентов и определяет степень их готовности к лабораторной работе. Затем преподаватель разбивает группу студентов на подгруппы по два человека. Каждая подгруппа получает от преподавателя индивидуальный вариант задания на лабораторную работу, который представляет собой криптограммы, зашифрованные с помощью криптосистемы Виженера.

Лабораторная работа состоит из двух частей. В первой части работы студенты дешифрируют первую заданную криптограмму с использованием первого метода Фридмана и метода чтения в колонках. При этом студенты должны: определить период ключевой последовательности с помощью первого метода Фридмана; используя метод чтения в колонках для заданного случая дешифрировать первую криптограмму. Вторая часть работы заключается в дешифровании второй криптограммы с применением второго метода Фридмана и метода «протяжки» вероятного слова. На этом этапе студенты должны: используя вто-

рой метод Фридмана определить период ключевой последовательности; на основании вычисленного значения периода ключевой последовательности используя метод «протяжки» вероятного слова дешифрировать вторую криптограмму.

Если в результате дешифрирования заданных криптограмм получены осмысленные тексты, студенты оформляют отчет и представляют его преподавателю.

3 Содержание отчета

Отчет должен включать в себя следующие пункты:

1. Задание на выполнение лабораторной работы (исходные криптограммы).
2. Основные расчетные соотношения.
3. Результаты расчетов, представленные в виде табл. 1 и 2. 4. Результаты анализа, т.е. дешифрованные криптограммы.

2.15 Практическая работа № 15 Изучение метода линейного криптоанализа блочных симметричных криптосистем

Задание:

Блочные симметричные криптосистемы (БСК) представляют собой семейство обратимых криптографических преобразований блоков (частей фиксированной дины) исходного текста.

В настоящее время разработано большое количество БСК, многие из которых являются национальными стандартами. Наибольшую известность приобрели системы DES, IDEA, AES (Rijndael), ГОСТ 28147-89. Эти системы находятся под пристальным вниманием криптоаналитиков, основной задачей которых является поиск «слабых мест» в этих системах.

В настоящей работе метод линейного криптоанализа БСК рассматривается применительно к криптосистеме S-DES, являющейся упрощенной версией криптосистемы DES.

1. Алгоритм шифрования (расшифрования) криптосистемы S-DES. На рис. 1 иллюстрируется алгоритм шифрования (расшифрования).

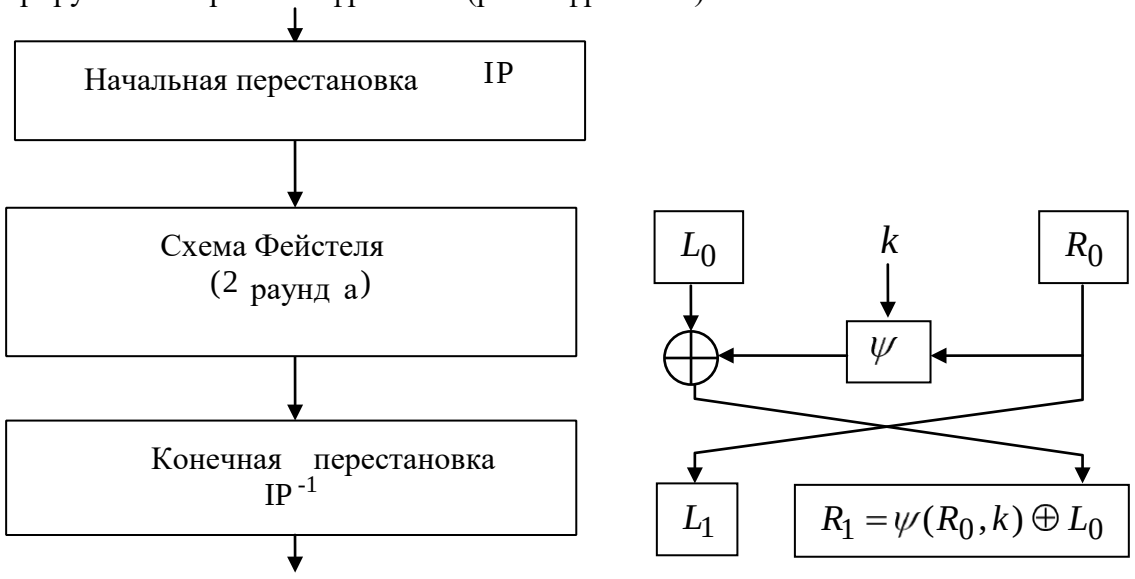


Рисунок 1 – Схема алгоритма шифрования S-DES с сетью Фейстеля

Входной 8-битовый блок вначале подвергается начальной перестановке

(IP), в соответствии с табл.1. Биты подблока пронумерованы от 0 до 7, причем бит с наибольшим порядковым номером 7 является младшим битом, и наоборот.

Таблица 1 – Начальная перестановка IP

7	6	4	0	2	5	1	3

Таблица разделена на две части, верхняя часть определяет подблок левых бит L_0, а нижняя часть определяет подблок правых бит R_0. Таким образом, после начальной перестановки IP, подблоки L_0 и R_0 подвергаются первому раунду шифрования. На выходе первого раунда получаются два выходных подблока L_1 и R_1, полученные в соответствии с выражением:

$$\overline{L1R0;R1} = L0 \oplus \psi\ (R0,k(8)_1).$$

Функция ψ, называемая функцией усложнения и аналогичная функции усложнения алгоритма DES, зависит от ключа, а ее вид будет описан ниже.

Подблоки L_1 и R_1 являются входными для второго раунда шифрования, на выходе которого получаются подблоки L_2 и R_2. Далее производиться объединение подблоков $L_2 \| R_2$ в блок, который подвергается перестановке, являющейся инверсией начальной перестановки. В результате получаем выходной блок криптограммы.

2. Алгоритм формирования раундовых ключей. Основной 10-битный ключ шифра $k_{(10)}$ используется для генерирования двух раундовых 8-битных ключей $k_{(8)1}$ и $k_{(8)2}$. Основной ключ шифра $k_{(10)}$, биты которого пронумерованы от 0 до 9, подвергается перестановке PC-1, определяемой табл. 2.

Таблица 2 – Перестановка PC-1

9	7	3	8	0
2	6	5	1	4

Верхняя строка таблицы определяют биты (9,7,3,8,0) подблока C_0, а нижняя - биты (2,6,5,1,4) подблока D_0. Подблоки C_0 и D_0 подвергаются единичному сдвигу влево, результатом которого является подблоки C_1 и D_1. Результат объединения подблоков $C_1 \| D_1$ подвергается перестановке, в соответствии с табл. 3.

Таблица 3 – Перестановка PC-2

5	3	9	7	2	8	6	4

Результатом перестановки PC-2 является первый раундовый ключ $k_{(8)1}$. Процедура формирования второго раундового ключа $k_{(8)2}$ аналогична, отличие заключается в том, что подблоки C_1 и D_1 подвергаются двум сдвигам влево.

3. Функция усложнения. На рис. 2 представлена схема функции усложнения ψ.

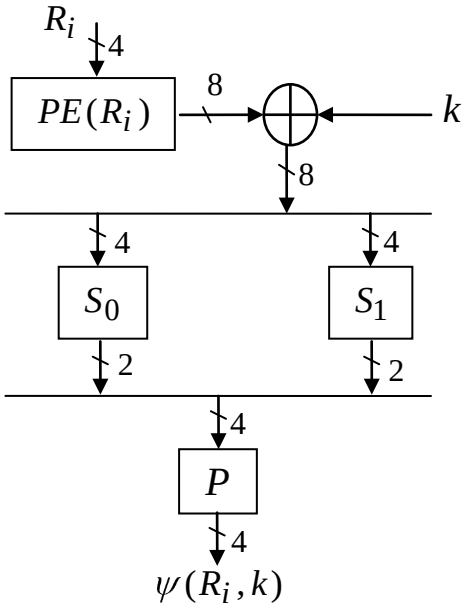


Рисунок 2 – Схема функции усложнения

Вначале 4-х битный подблок подвергается перестановке с расширением (РЕ), в соответствии с табл. 4, на выходе которой получается 8-ми битный блок. Полученный результат складывается по mod2 с битами 8-ми битного раундового ключа $k_{(8)i}$, i 1,2 и подвергается перестановке в блоках замены S_0 и S_1 (см. табл. 5 и табл. 6).

Таблица 4 – Перестановка с расширением РЕ

3	0	1	2	1	2	3	0

Таблица 5 – Блок замены S_0

S_0	№ столбца			
№ строки	**0**	**1**	**2**	**3**
0	1	0	2	3
1	3	1	0	2
2	2	0	3	1
3	1	3	2	0

Таблица 6 – Блок замены S_1

S_1	№ столбца			
№ строки	**0**	**1**	**2**	**3**
0	0	3	1	2
1	3	2	0	1
2	1	0	3	2
3	2	1	3	0

Причем, результат операции сложения по mod2 затем разбивается на два подблока, первые четыре бита (0,1,2,3) образуют подблок B_0, оставшиеся биты (4,5,6,7) образуют подблок B_1. Подблоки B_0 и B_1 подвергаются преобразованию в блоках замены S_0 и S_1, соответственно. Крайние биты входного 4-битного подблока определяют строку таблицы, а средние биты – столбец. После преобразования в блоках замены выходные 2-битные подблоки объединяются $S_0(B_0) \parallel S_1(B_1)$. Полученный 4-битный подблок подвергается перестановке (Р), в соответствии с табл. 7.

Таблица 7 – Перестановка (P)

1	0	3	2

Результатом перестановки будет выходное значение функции усложнения $R_{i,k}(8)i$, i ψ 1,2.

Метод линейного криптоанализа. Метод линейного криптоанализа разработан в 1993 году японским криптологом Митсуру Матсуи. В первоначальном виде этот метод сформулирован применительно к криптосистеме DES, в настоящее время создаются новые модификации этого метода [4].

Идея метода линейного криптоанализа основана на том, что существует возможность заменить нелинейную функцию криптографического преобразования ее линейным аналогом. Линейный криптоанализ базируется на знании криптоаналитиком пар «открытый текст-криптограмма», а также алгоритма шифрования.

Будем считать, что при генерации исходного текста X случайные биты независимы и равновероятны $P(x_i 1)\ p,\ P(x_i 0)\ 1\ p$, $p\ 0{,}5$. *Линейным статистически аналогом* (или приближенным линейным аналогом) называется выражение:

$$\lambda(X,Y) = \sum_{i=1}^{n} a_i x_i \oplus \sum_{i=1}^{n} b_i y_j = \sum_{k=1}^{L} c_k k_k , \qquad (1)$$

если вероятность

$$P\left\{\lambda(X, f(X,K)) = \sum_{k=1}^{L} c_k k_k\right\} = 0{,}5 + \Delta$$

Величина $\Delta = |1-2p|$ называется *эффективностью линейного аналога*, а коэффициенты a_i 1 , b_i 1 , c_k 1 - параметрами линейного аналога. По существу Δ характеризует степень линейности функции .

криптографического преобразования и имеет максимальное значение $\max\ 0{,}5$. При применении метода линейного криптоанализа решаются две взаимосвязанные задачи:

1) нахождение эффективного линейного статистического аналога и вычисление его вероятности;

2) определение ключа (или нескольких бит ключа) с использованием эффективного линейного статистического аналога.

Практическая реализация метода линейного криптоанализа связана с реализацией следующих последовательных шагов.

1. Тщательно анализируется криптографическая функция и определяется множество линейных статистических аналогов. На этом шаге в первую очередь анализируются S-блоки функции усложнения ψ. Для этого формируются таблицы значений $Q_t(i, j)$, где: $t 0{,}1$ - номер S-блока, $i\ 1{,}4$, $j\ 1{,}2$. Значение $Q_t(i, j)$ представляет собой количество совпадений суммы по mod2 некоторых битов входных данных с суммой по mod2 некоторых битов выходных данных. В ходе анализа прослеживаются все возможные комбинации двоичных векторов i, j . Каждая пара векторов используется в качестве маски, которая накладывается на возможные пары «вход-выход» S-блока. Эти маски указывают на биты входа и выхода, которые необходимо сложить по mod2, а затем сравнить полученные результаты.

* *

Далее проводится анализ полученных таблиц $Q_t(i, j)$ и отыскиваются такие значения i , j , для которых выполняется условие:

$$Qt(i^*, j^*): \max Qt(i, j) \quad -8 \cdot \qquad (2)$$

В соответствии с полученной парой i , j , и учитывая в схеме алгоритма шифрования перестановки и сложение по mod2, формируется эффективный линейный статистический аналог:

$$\lambda^*(X,Y) = \sum_{i=1}^{n} a_i^* x_i \oplus \sum_{i=1}^{n} b_i^* y_j = \sum_{k=1}^{L} c_k^* k_k \qquad , P_{эа} = \frac{Q(i^*, j^*)}{16}.$$

2. Генерируется множество независимых исходных текстов $X(1), X(2), \ldots, X(M)$ и регистрируются соответствующие им криптограммы $Y(1), Y(2), \ldots, Y(M)$.

3. Для каждой пары $X(m), Y(m)$, m $1, M$ вычисляется значение левой части эффективного линейного статистического аналога:

$$\lambda^*(X(m), Y(M)) = \sum_{i=1}^{n} a_i^* x_{im} \oplus \sum_{i=1}^{n} b_i^* y_{im} . \qquad (4)$$

4. Определяется частота получения «1» при вычислении M значений (4):

$$\nu = \frac{1}{M} \sum_{m=1}^{M} \lambda^* (X(m), Y(m)), \qquad (5)$$

и строится оценка максимального правдоподобия в соответствии с правилом:

$$d = \begin{cases} 1, & \nu \geq 0{,}5, \\ 0, & \nu < 0{,}5. \end{cases} \qquad (6)$$

5. Строится система линейных уравнений, причем каждое уравнение системы представляет собой равенство правой части (4) и соответствующего значения (6)

$$\sum_{k=1}^{L} c_k^* k_k = d . \qquad (7)$$

Единственное решение полученной системы (7) используется в качестве оценки ключа $k^* = k_1^*, k_2^*, \ldots, k_L^*$.

2 Порядок выполнения работы

Лабораторная работа выполняется с использование программы «Cryptoanaliz».

2.1. При подготовке к лабораторной работе

На этапе подготовки к лабораторной работе студенты должны, используя литературу [1-4], материалы лекций углубить свои знания по следующим вопросам: блочные симметричные криптосистемы (определение, основные характеристики, достоинства и недостатки), блочная криптосистема S-DES, метод линейного криптоанализа блочных криптосистем, а также изучить инструкцию по использованию программы «Cryptoanaliz».

2.2. Во время проведения занятия

Преподаватель перед проведением занятия проводит контрольный опрос студентов и определяет степень их готовности к лабораторной работе. Преподаватель разбивает группу студентов на 5 подгрупп, для каждой подгруппы определяется номер индивидуального задания на предстоящую лабораторную работу. Варианты индивидуальных заданий заложены в программе «Cryptoanaliz».

В процессе выполнения работы студенты должны:

1. Запустить на исполнение программу «Cryptoanaliz» и пройти предлагаемый контрольный тест.

2. В соответствии с заданием определенным преподавателем студенты выбирают номер варианта, количество известных текстов и осуществляют зашифрование случайным образом сгенерированных открытых текстов.

3. Используя таблицы Q_0 и Q_1, и учитывая таблицы перестановки и сложение по mod2, студенты определяют эффективные линейные аналоги и вычисляют их вероятности. Полученный результат студенты заносят в табл 8.

Таблица 8 – Эффективные линейные статистические аналоги

№ блока	Эффективный линейный аналог	p	$\Delta = \lvert 1 - 2p \rvert$
S_0			
S_1			

4. Для каждого из полученных линейных аналогов студенты определяют в соответствии с выражениями (5), (6) значение правой части уравнений используя модуль «Анализ».

5. Используя полученные результаты, студенты формируют систему уравнений (7). Решение системы уравнений позволяет определить все или часть битов 8-битных раундовых ключей. Используя алгоритм формирования раундовых ключей криптосистемы S-DES, студенты определяют основный 10битный ключ шифра. Возможные варианты 10-битного ключа шифра и соответствующие ему 8-битные раундовые ключи студенты заносят в отчет по лабораторной работе.

6. Используя модуль «Проверка» студенты проверяют правильность каждого из полученных вариантов ключей шифра.

7. При совпадении результатов анализа с истинным ключом шифра студенты оформляют, в соответствии с требованиями настоящего пособия отчет и представляют его преподавателю для защиты.

3 Содержание отчета

Отчет должен включать в себя следующие пункты:

1. Схему блочной криптосистемы S-DES и исходные данные индивидуального задания.

2. Таблицы статистического анализа Q_0 и Q_1, и таблицу с эффективными линейными статистическими аналогами (табл. 8).

3. Систему линейных уравнений для определения битов ключа.

4. Варианты полученных ключей.

5. Результат проверки подтверждающий правильность определенного в работе ключа.

2.16 Практическая работа № 16 Изучение метода дифференциального (разностного) криптоанализа блочных симметричных криптосистем

Задание:

Криптосистема S-DES подробно рассмотрена в лабораторной работе №3.

Метод дифференциального (разностного) криптоанализа предложен Э. Байхэмом и А. Шамиром, и, по мнению ряда специалистов компании IBM, является общим методом криптоанализа блочно-итерационных криптосистем. Идея заключается в анализе процесса изменения несходства для пары открытых текстов $X\ X\ X$, имеющих определенные исходные различия, в процессе прохождения через циклы шифрования с одним и тем же ключом.

Пусть задана пара входов X и X', с несходством $X\ X$. Известны перестановка IP и перестановка с расширением E, а следовательно, известны и несходства А на входе блоков замены S_0 и S_1. Выходы Y и Y' известны, следовательно, известно и несходство $Y\ Y\ Y$, а значит, при известных перестановках IP^{-1} и P известны несходства ΔС на выходе блоков замены S_0 и S_1.

Доказано, что для любого заданного ΔА не все значения ΔС равновероятны. Комбинация ΔА и ΔС позволяет предположить значения битов для $E(X)$ k_i и $E(X')$ k_i . То, что $E(X)$ и $E(X')$ известны, дает информацию о k_i .

Несходство различных пар открытых текстов приводит к несходству получаемых шифр-текстов с определенной вероятностью. Эти вероятности можно определить, построив таблицы для каждого из блоков замены. Таблицы сроятся по следующему принципу: по вертикали располагаются все возможные комбинации А, по горизонтали – все возможные комбинации ΔС, а на пересечении – число соответствий данного ΔС данному А.

Число наибольших совпадений указывает нам пару ΔА и ΔС, с помощью которой можно определить секретный ключ. Пара открытых текстов, соответствующих данным ΔА и ΔС называется *правильной* парой, а пара открытых текстов, не соответствующих данным ΔА и ΔС – *неправильной* парой. Правильная пара подскажет правильный ключ цикла, а неправильная пара – случайный. Чтобы найти правильный ключ, необходимо просто собрать достаточное число предположений. Один из подключей будет встречаться чаще, чем все остальные. Фактически правильный подключ появляется из всех возможных случайных подключей.

2 Порядок выполнения работы

Лабораторная работа выполняется с использование программы «Cryptoanaliz».

2.1. При подготовке к лабораторной работе

На этапе подготовки к лабораторной работе студенты должны, используя литературу [1-4], материалы лекций углубить свои знания по следующим вопросам: блочные сим-

метричные криптосистемы (определение, основные характеристики, достоинства и недостатки), блочная криптосистема S-DES, метод дифференциального (разностного) криптоанализа блочных криптосистем, а также изучить инструкцию по использованию программы «Cryptoanaliz».

2.2. Во время проведения занятия

Преподаватель перед проведением занятия проводит контрольный опрос студентов и определяет степень их готовности к лабораторной работе.

Преподаватель разбивает группу студентов на 5 подгрупп, для каждой подгруппы определяется номер индивидуального задания на предстоящую лабораторную работу. Варианты индивидуальных заданий заложены в программе «Cryptoanaliz».

В процессе выполнения работы студенты должны:

1. Запустить на исполнение программу «Cryptoanaliz» и пройти предлагаемый контрольный тест.

2. В соответствии с заданием определенным преподавателем студенты выбирают номер варианта и количество известных текстов.

3. Используя таблицы анализа несходств ($\overset{\Delta}{A}, \overset{\Delta}{C}$) для блоков замены S_0 и S_1 студенты определяют оптимальный дифференциал ($\overset{*}{\Delta} A$, $\overset{*}{\Delta} C$) и осуществляют зашифрование случайным образом сгенерированных открытых текстов. Программа выбирает из множества пар текстов пары удовлетворяющие оптимальному дифференциалу ($\overset{*}{\Delta} A$, $\overset{*}{\Delta} C$) и представляет их в виде в табл. 1.

Таблица 1 – Пары текстов удовлетворяющие оптимальному дифференциалу

№	X	$E(X)$	$S(E(X))$	Y
1				
…				
№	X'	$E(X)'$	$S(E(X))'$	Y'
1				
…				

4. Студенты анализируют пары открытых текстов и определяют множество раундовых ключей шифра и, соответственно, множество основных ключей шифра. Ключ, получаемый чаще остальных и будет наиболее вероятным ключом шифра.

5. Используя модуль «Проверка» студенты проверяют правильность определенного анализом ключей шифра.

6. При совпадении результатов анализа с истинным ключом шифра студенты оформляют, в соответствии с требованиями настоящего пособия отчет и представляют его преподавателю для защиты.

4 Содержание отчета

Отчет должен включать в себя следующие пункты:

1. Структуру алгоритма S-DES и таблицы перестановок и замен, соответствующие заданному варианту.

2. Результаты анализа таблиц замен S_0 и S_1.

3. Результаты анализа пар открытых текстов.
4. Множество возможных раундовых ключей.
5. Результаты проверки, подтверждающие правильность определенного в работе ключа.

2.17 Практическая работа № 17 Методы оценки качества криптографических генераторов

Задание:

Криптографический генератор – это аппаратно или программно реализованный имитатор источника равномерно распределенной случайной последовательности (РРСП) чисел, которая вычисляется по известному детерминированному рекуррентному соотношению. Основное требование к выходной последовательности x криптографического генератора, $i = \overline{0,\infty}$, состоит в минимальных отличиях по статистическим характеристикам последовательности x от РРСП.

Тесты, применяемые для оценки качества КГ, делятся на графические и оценочные [3].

К ***графическим тестам*** относится: гистограмма распределения элементов последовательности; распределение на плоскости; проверка серий; автокорреляционная функция; графический спектральный тест; проверка на монотонность; профиль линейной сложности.

Гистограмма распределения элементов – данный метод позволяет оценить равномерность распределения символов, а также определить частоту появления каждого символа. Для исследуемой последовательности $_i$,i 1,n подсчитывается сколько раз встречается каждый элемент, после чего строиться график зависимости числа появления элементов от их численного представления.

Распределение на плоскости – метод предназначен для определения зависимостей между элементами последовательностей. Построение распределение на плоскости осуществляется по правилу $_i$’ i 1 ,i 1,(n 1). Если между элементами последовательности связь отсутствует, то точки на плоскости расположены хаотично, в случае когда на плоскости наблюдаются «узоры» - между элементами последовательности существует взаимосвязь, т.е. последовательность не является случайной.

Проверка серий – данный метод позволяет оценить равномерность распределения символов в исследуемой последовательности на основе анализа частоты появления нулей и единиц и серий состоящих из k бит. Построение осуществляется следующим образом: подсчитывается сколько раз проявляются нули, единицы, серии-двойки (00, 01, 10, 11), серии-тройки (000, 001, 010, 011, 100, 111) и т.д. в битовом представлении исследуемой последовательности $_i$,i 1,n. Результат представляется в графическом виде. У последовательности, чьи свойства близки к свойствам РРСП разбросы между числом появлений нулей и единиц, между числом появлений серий каждого вида должны стремиться к нулю.

Проверка на монотонность – данный метод позволяет оценить равномерность распределения символов. В исследуемой последовательности на основе анализа длин участков невозрастания и неубывания элементов последовательности. Исследуемая последовательность $_i$,i 1,n представляется в виде непересекающихся участков невозрастания и неубывания следующих друг за другом. У последовательности, чьи свойства близки к свойствам РРСП вероятность появления участка невозрастания (неубывания) обратно пропорциональна его длине.

Автокорреляционная функция (АКФ) – данный метод предназначен для оценки корреляции между сдвинутыми копиями исследуемой последовательности $_i$,i 1,n. Метод позволяет обнаруживать зависимость между подпоследовательностями исследуемой последовательности. *Битовая* АКФ. Исследуемая последовательность $_i$,i 1,n представляется в битовом виде и затем нормируется по правилу ~$_i$ (1)1 ,i 1,n. После этого вычисляются всплески корреляции:

$$r_j = \frac{\sum_{i=1}^{n} \tilde{\gamma}_i \tilde{\gamma}_{(i+j) \bmod n}}{\sum_{i=1}^{n} \tilde{\gamma}_i}, \quad j = \overline{1,n}.$$

Символьная АКФ. Исследуемая последовательность нормируется по правилу $\tilde{\gamma}_i = \sum_{j=0}^{R-1} (-1)^{a_i} 2^j, \overline{i =}$ 1,n, R - разрядность числа, a_i - двоичная запись i – го элемента исследуемой последовательности. Далее вычисляются всплески корреляции.

Для последовательности, чьи свойства близки к свойствам РРСП, значения всплесков корреляции должно стремиться к нулю, во всех точках, кроме тех, чье значение кратно длине последовательности в символах (в битах) для символьной (битовой) АКФ.

Профиль линейной сложности – данный метод позволяет исследовать последовательность на случайность анализируя зависимость линейной сложности последовательности от ее длины. Для исследуемой двоичной последовательности $_i$,i 1,n рассматриваются подпоследовательности $\gamma^{(}$ содержащие первые k элементов и строится графическая зависимость линейной сложности L от длины подпоследовательности k . У последовательности, чьи свойства близки к свойствам РРСП линия графика k должна стремиться к линии L. $= -$ 2

Графический спектральный метод – данный метод позволяет определить равномерность распределения 0 и 1 в исследуемой последовательности на основе анализа высоты выбросов преобразования Фурье. Исследуемая двоичная последовательность $_i$,i 1,n преобразуется по правилу $\sim_i 2\ _i 1$ в последовательность ~$_i$,i 1,n к которой применяется дискретное преобразование Фурье. У последовательности, чьи свойства близки к свойствам РРСП число гармоник, чьи длительности значительно превышают среднюю длину гармоники должно стремиться к нулю.

Оценочные тесты, в отличии от графических тестов, позволяют по результатам тестирования сделать практически однозначный вывод о возможности использования криптографического генератора. Существует множество различных оценочных тестов, сгруппированных в наборы: тесты Д.Кнута, тест DieHard, тест NIST и др. [3]. В настоящей лабораторной работе рассматриваются некоторые тесты, входящие в набор NIST.

$_n = n_1 -$

Вычисляется статистика $s = \frac{|S_n|}{\sqrt{n}}$

Частотный тест (frequency test) служит для проверки равно-

где $erfc(x) = 1 - erf(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-\tau^2} \tau$

$(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\tau^2} d\tau$ - функция ошибок.

мерности появления 0 и 1 в исследуемой последовательности. В исследуемой последовательности длинной n подсчитывается количество нулей n_0 и единиц n_1, а затем вычисляется их разница S_{n_0}.

$$= erfc\left(\frac{s}{\sqrt{2}}\right),$$

и определяется значение P

d - дополнительный интеграл вероятностей,

$erf\, x$

Связь стандартного нормального распределения x и функции ошибок $\Phi(x)$ имеет вид:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{\tau^2}{2}} d\tau = 0{,}5\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right).$$

Значение P должно быть больше 0,01.

Частотный тест в последовательностях (frequency test within a block) необходим для проверки равномерности появления 0 и 1 в подпоследовательности. Исследуемая двоичная последовательность разбивается на $N = \left[\frac{n}{M}\right]$ M-битных последовательностей. Лишние биты отбрасываются. Определяется доля 1 в каждой подпоследовательности $\pi_i = \frac{\sum_{j=1}^{M} \varepsilon_{(i-1)M+j}}{M}$. Вычисляется статистика $s = 4M \sum_{i=1}^{M} (\pi_i - 0{,}5)^2$ и значение $P = igamc\left(\frac{N}{2}, \frac{s}{2}\right)$, где: $igamc(a, x) = 1 - p(a, x)$, $p(a, x) = \frac{\Gamma_x(a)}{\Gamma(a)}$, $\Gamma(a) = \int_0^{\infty} t^{a-1} e^{-t} dt$ - гамма-функция, $\Gamma_x(a) = \int_0^{x} t^{a-1} e^{-t} dt$ - неполная гамма-функция.

Значение P должно быть больше 0,01.

Тест «дырок» (runs test) служит для проверки равномерности распределения 0 и 1 в исследуемой последовательности на основе анализа количества появлений «блоков» - подпоследовательностей, состоящих из одних 1 или одних 0 («дырок»). Определяется предтестовая статистика π – доля 1 в исследуемой последовательности $\pi = \frac{\sum_{i=1}^{n} \varepsilon_i}{n}$. Если $|\pi - 0{,}5| \geq \tau = \frac{2}{\sqrt{n}}$ - тест (количество блоков и «дырок») $\nu_n = \sum_{k=1}^{n-1} r(k) +$ считается $= \begin{cases} 0, & \varepsilon_i = \varepsilon_{i+1} \\ 1, & \varepsilon_i \neq \varepsilon_{i+1} \end{cases}$

$$= erfc\left(\frac{|\nu_n - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right)$$

пройденным, в противном случае вычисляем статистику

,

1, где: $r(k)$. Затем

вычисляется P . Значение P должно быть больше 0,01.

Проверка рангов матриц (binary matrix rank test) служит для проверки равномерности распределения 0 и 1 в исследуемой последовательности на основе анализа количества появлений матриц различных рангов. Исследуемая

$$= \left\lfloor \frac{n}{MQ} \right\rfloor$$

двоичная последовательность длины n разбивается на N непересекающихся подпоследовательностей. Лишние биты отбрасываются.
Каждая подпоследовательность представляется как бинарная матрица размером $M\ Q$.

Определяется ранг каждой матрицы. Пусть F_M - число матриц ранга M , $F_{M\,1}$ - число матриц ранга $M\ 1$, $N\ F_M\ F_{M\,1}$ - число оставшихся матриц.
Вычисляется статистика

$$s = \frac{FM - 0{,}2888N}{0{,}2888N} + \frac{FM1 - 0{,}5776N}{0{,}5776N} + \frac{N - FM - FM1 - 0{,}1336N}{0{,}1336N},$$

а затем значение $P\ igamc\ 1,$. Значение P должно быть больше 0,01. 2 *Спектральный тест* (spectral test) служит для проверки равномерности 0 и 1 в исследуемой последовательности на основе анализа высоты выбросов преобразования Фурье.

Исследуемая двоичная последовательность i , $i\ 1,n$ преобразовывается в последовательность x_i по правилу $x_i\ 2\ _i\ 1$. К полученной последовательности применяется дискретное преобразование
Фурье S , из которого формируется подпоследовательность S , n содержащая первые членов S . Члены последовательности S являются 2

комплексными числами. Определяются модули $m_i\ S_i$ каждого из элементов последовательности S . Последовательность m_i дает последовательность выбросов высот преобразования Фурье.

$$= \sqrt{\quad} \quad 0{,}95n$$

Вычисляются T $3n,\ N_0$ и определяется статистика

$$= \frac{}{2}$$

$s = \frac{}{\sqrt{\quad}}$, где N_1 - число элементов m_i , меньших чем T .

$$\frac{N\ 1\ N_0}{0{,}05n}$$

$$0{,}95$$

$$2$$

Вычисляется значение $P = erfc\left(\frac{|s|}{\sqrt{2}}\right)$. Значение P должно быть больше 0,01.

2 Порядок выполнения работы

2.1. При подготовке к лабораторной работе

На этапе подготовки к лабораторной работе студенты должны, используя литературу [1,2,3], материалы лекций углубить свои знания по следующим вопросам: классификация криптографических генераторов, методы усложнения алгоритмов генерации псевдослучайных последовательностей, графические и оценочные тесты качества криптографических генераторов.

2.2. Во время проведения занятия

Преподаватель перед проведением занятия проводит контрольный опрос студентов и определяет степень их готовности к лабораторной работе. Преподаватель разбивает группу студентов на подгруппы, для каждой подгруппы выдается индивидуальное задание на предстоящую лабораторную работу. Индивидуальное задание представляет собой тип криптографического генератора с исходными данными, который требуется исследовать.

В процессе выполнения работы студенты должны:

1. Сформировать, для заданного преподавателем типа криптографического генератора, псевдослучайную последовательность.
2. Оценить качество криптографического генератора с помощью рассмотренных в учебных материалах тестов и построить соответствующие графические зависимости.
3. Сформулировать вывод о возможности использования криптографического генератора в алгоритмах шифрования.

3 Содержание отчета

Отчет по лабораторной работе должен включать в себя следующие пункты:

1. Задание на лабораторную работу.
2. Графические зависимости, иллюстрирующие результаты применения графических тестов и таблицу с результатами оценочного тестирования.

Выводы о возможности использования заданного криптографического генератора в алгоритмах шифрования.

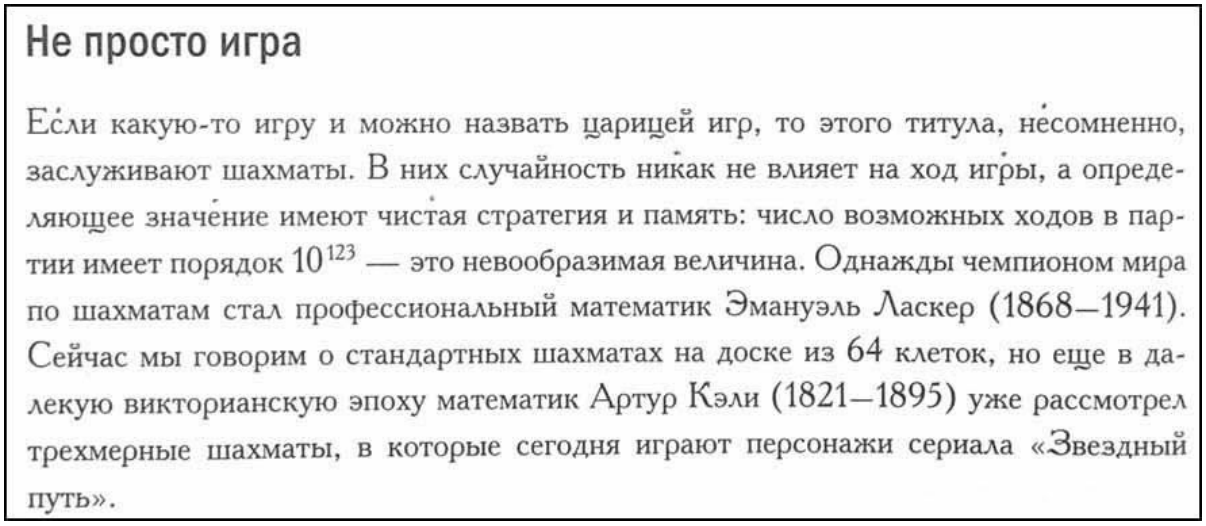
2.18 Практическая работа № 18 Применение текстовой криптографии

Задание:

Компьютерные стеганографические методы как самостоятельно, так и совместно с криптографией, получили широкое распространение в целях защиты конфиденциальной информации. В лабораторной работе рассматривается стеганографическое сокрытие секретных сообщений в текстовых документах редактора Microsoft Word за счет специфического форматирования символов текста. Принципы сокрытия базируются на других известных стеганографических методах.

1. Микроточки. Использование микроточек для передачи секретных сообщений описал греческий ученый Эней Тактик в сочинении «Об обороне укреплённых мест». Суть предложенного им так называемого «книжного шифра» заключалась в прокалывании малозаметные дырок в книге или в другом документе над буквами секретного сообщения. Во

время Первой мировой войны германские шпионы использовали аналогичный шифр, заменив дырки на точки, наносимые симпатическими чернилами на буквы газетного текста.



Не просто игра

Е́сли какую-то игру и можно назвать царицей игр, то этого титула, не́сомненно, заслуживают шахматы. В них случайность ника́к не влияет на ход игры́, а определяющее значе́ние имеют чиста́я стратегия и память: число возможных ходов в партии имеет порядок 10^{123} — это невообразимая величина. Однажды чемпионом мира по шахматам стал профессиональный математик Эмануэль Ласкер (1868–1941). Сейчас мы говорим о стандартных шахматах на доске из 64 клеток, но еще в далекую викторианскую эпоху математик Артур Кэли (1821–1895) уже рассмотрел трехмерные шахматы, в которые сегодня играют персонажи сериала «Звездный путь».

Рис.1. Сокрытие сообщения «секрет» в тексте за счет малозаметных точек (Хоакин Наварро. Тайная жизнь чисел. Мир математики – том 31)

По аналогии с микроточками скрываемая в тексте секретная информация специальным образом помечается (форматируется).

2. Использование особенностей человеческого зрения. Подобные методы широко используются для сокрытия информации в мультимедийных файлах (в частности, метод LSB, Least Significant Bit - наименьший значащий бит) за счет их избыточности. По аналогии с ними, в обычном тексте символы, составляющие секретное сообщение, могут форматируются так, что это будет незаметно для глаза неискушенного читателя текста. В частности, символы секретного сообщения могут выделяться другим цветом, незначительно отличающегося от цвета остальных символов.

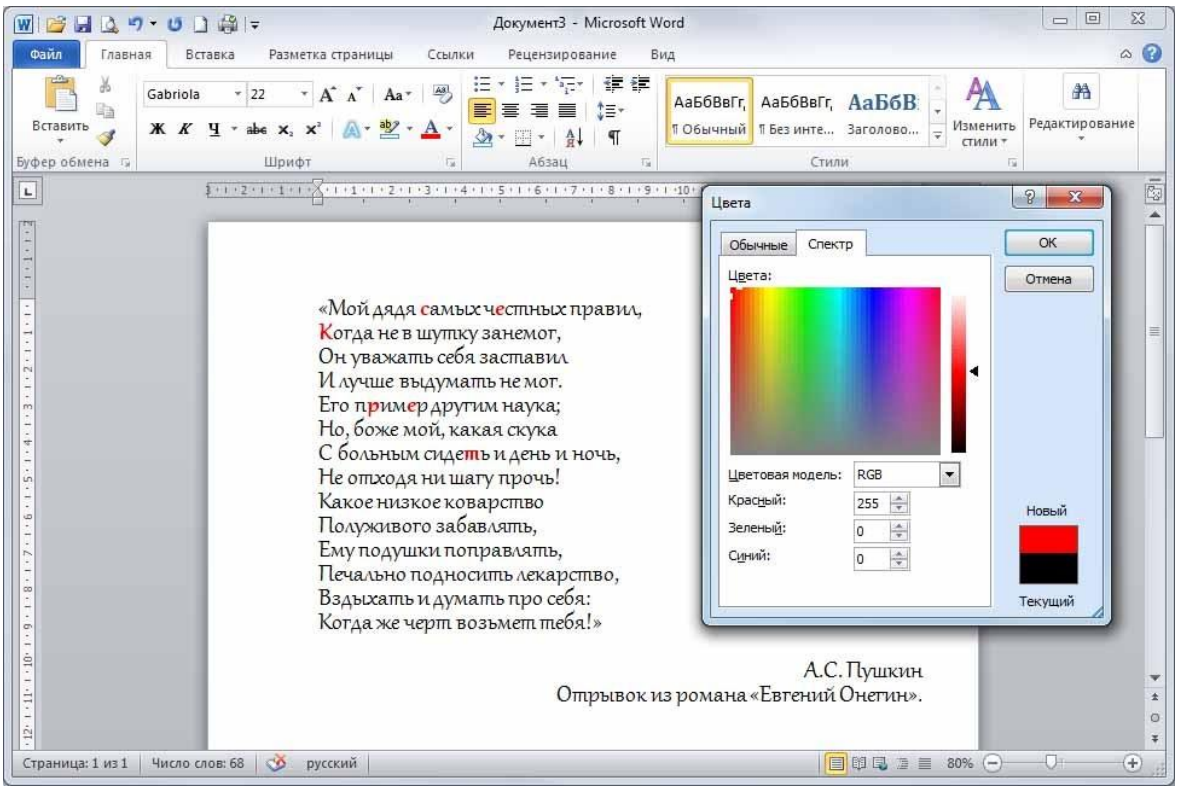


Рис.2. Принцип форматирования символов секретного сообщения «секрет» (цвет символов красный – RGB(255, 0, 0))

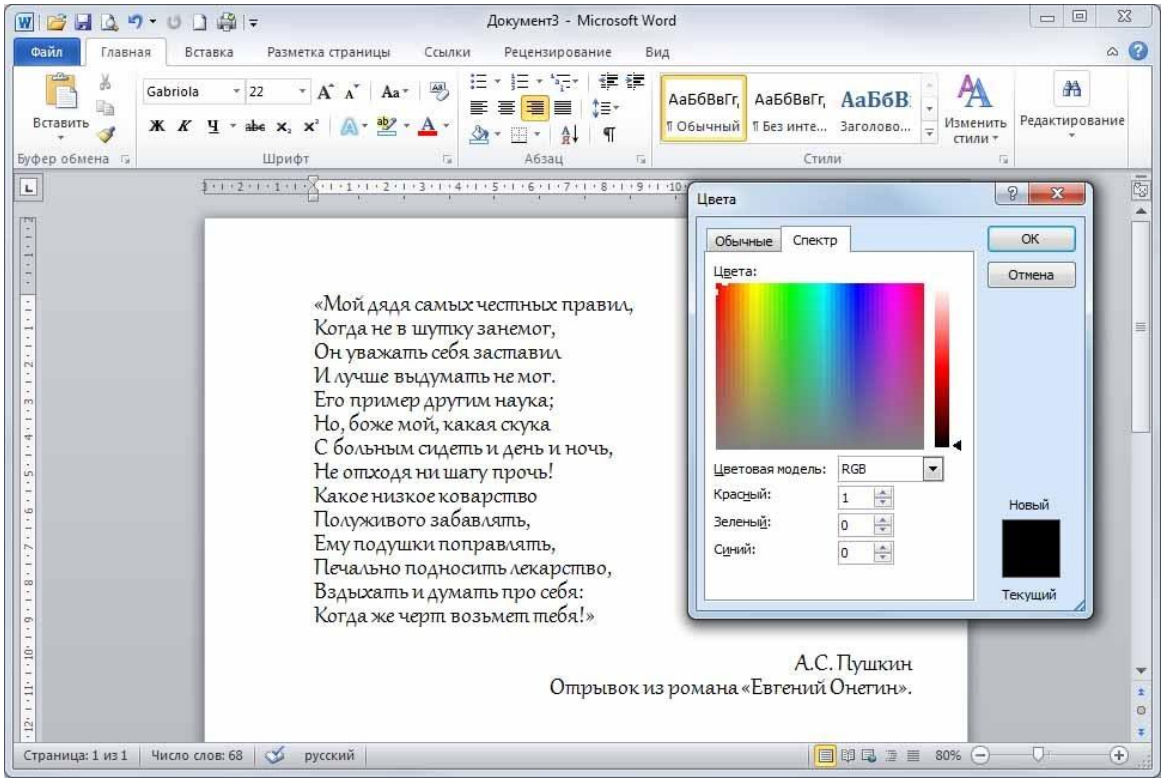


Рис.3. Стеганографическое сокрытие символов секретного сообщения «секрет» (цвет символов «почти черный» – RGB(1, 0, 0))

На рис.3 цвет символов секретного сообщения RGB(1, 0, 0) практически не отличается от цвета символов остального текста RGB(0, 0, 0).

3. Семаграммы и кодирование. Предыдущий метод можно усилить за счет использования предварительного кодирования символов секретного сообщения (например, азбукой Морзе или Windows 1251). Перед форматированием символы секретного сообщения вначале кодируются битовыми строками длиной n согласно принятой кодировке. В исходном тексте выбираются n первых символов, которые будут соответствовать битовому представлению первого символа секретного сообщения. Для нулей битовой строки оставляют исходное форматирование, для единиц – незначительно меняют (см. рис. 3). Процедуру последовательно повторяют для оставшихся символов секретного сообщения. Например, слово «секрет» согласно кодировке Windows 1251 в битовом представлении будет выглядеть 11110001 11100101 11101010 11110000 11100101 11110010_2.

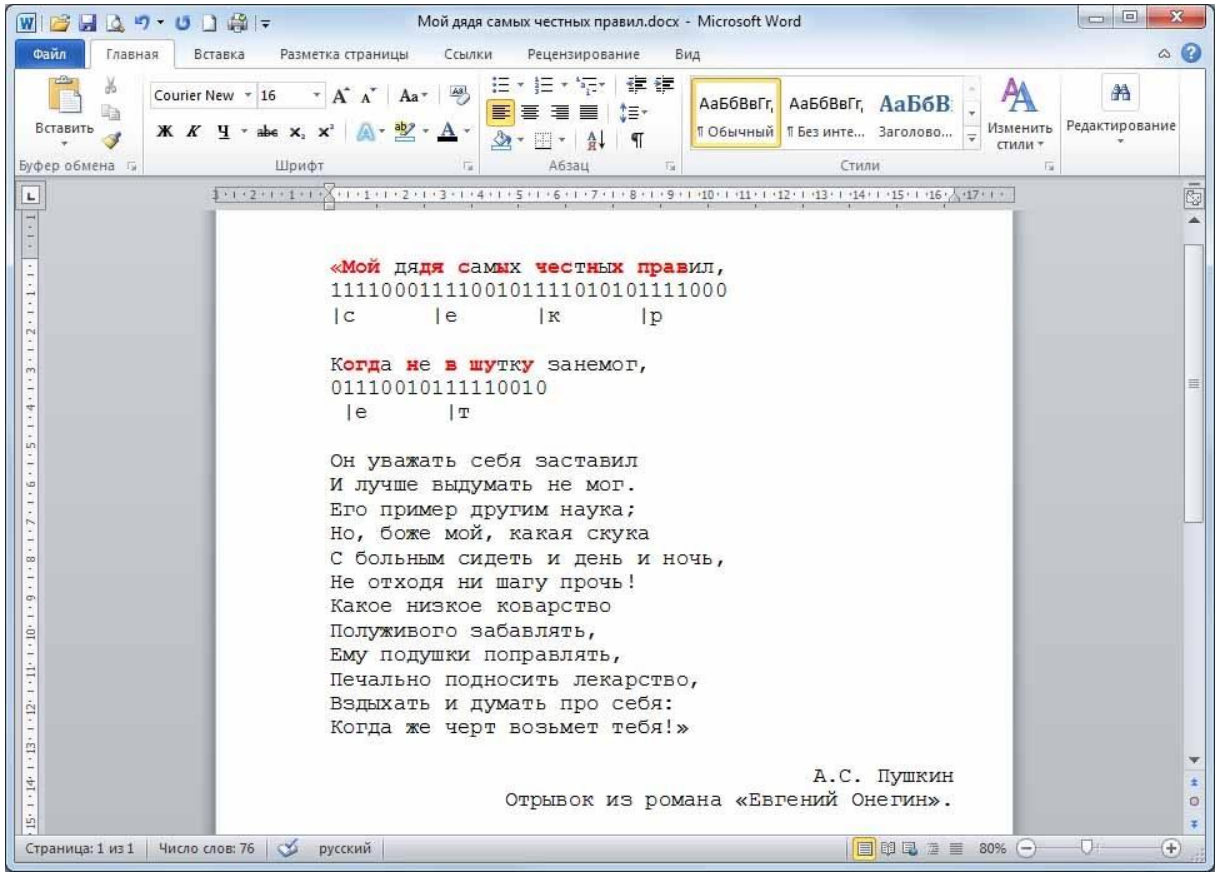


Рис.4. Принцип кодирования и форматирования символов секретного сообщения «секрет» (цвет нулей черный – RGB(0, 0, 0); цвет единиц красный – RGB(255, 0, 0))

2. Задание на выполнение лабораторной работы

1) Для заданного файла необходимо определить скрытое сообщение и использованный метод его стеганографического сокрытия.

2) Способы форматирования символов, применяемые для секретных сообщений (символов целиком, нулей или единиц):

- цвет символов;
- цвет фона;
- размер шрифта;
- масштаб шрифта;
- межсимвольный интервал.

3) Применяемые двоичные кодировки символов:

- без кодировки;
- код Бодо (МТК-2);

- КОИ-8R;
- ср866;
- Windows 1251.

4) Варианты индивидуальных заданий (выбираются согласно номеру в журнале):

В качестве текстов использованы стихи Агнии Барто, секретных сообщений – японские пословицы и поговорки. Файлы с заданиями сформированы с помощью программы, разработанной Максимом Вячеславовичем Орловым (ДВГУПС, студент 240 гр., 2015 г.).

5) Отчет по лабораторной работе должен содержать:

- фрагмент стиха, содержащий секретное сообщение (см. рис.4):
    - с подчеркиванием символов, соответствующих единицам (вместо выделения красным цветом);
    - с битовыми строками;
    - с символами секретного сообщения;
- вывод (например, «В файле «variant01.docx», скрыта фраза «Один бог забыл - другой поможет.» посредством использования кодировки ср866 и размера символов: для нулей – 14пт, для единиц – 14.5пт»).

2.19 Практическая работа № 19 Исследование методов цифровой стеганографии для защиты информации

Задание:

Стеганография – наука, которая изучает способы и методы скрытия конфиденциальной информации.

Задача стеганографии: скрытие самого факта существования секретных данных при их передаче, хранении или обработке.

В отличие от криптографии, где противник точно может определить, является ли передаваемое сообщение зашифрованным текстом, методы стеганографии позволяют встраивать секретные сообщения в открытые послания таким образом, чтобы было невозможным заподозрить существование самого встроенного послания.

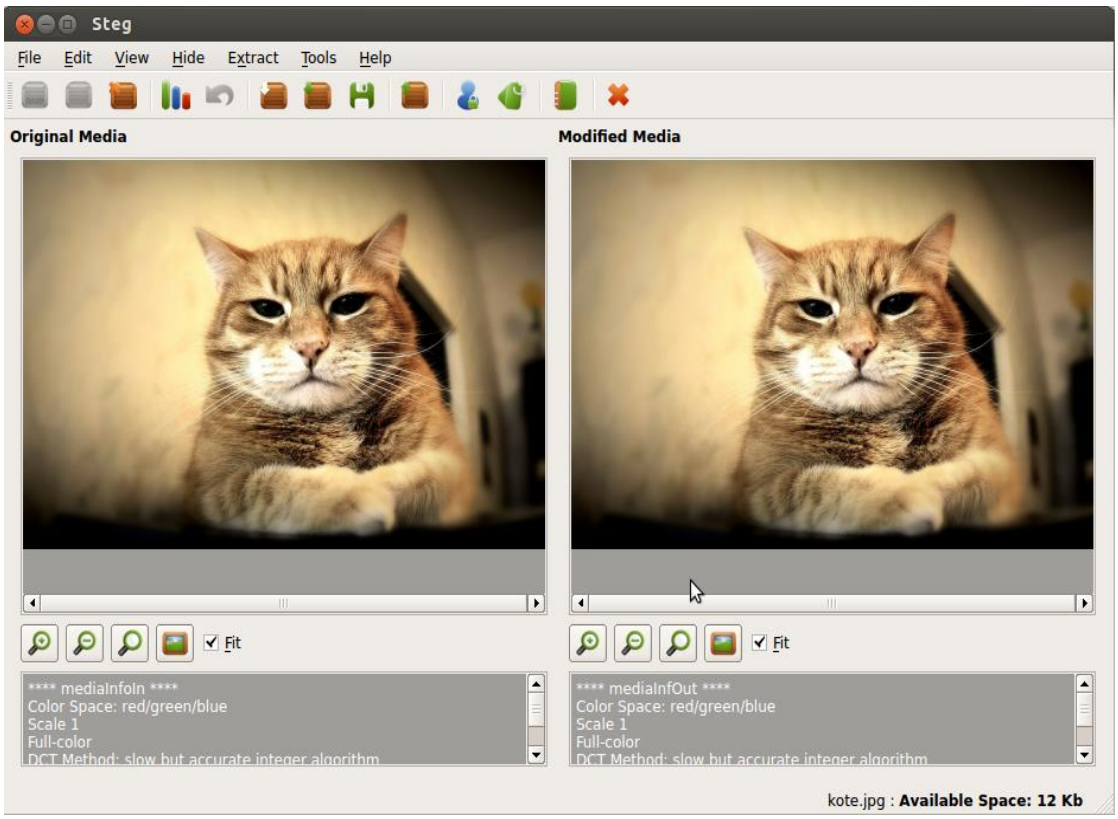
Таким образом, если цель криптографии состоит в блокировании несанкционированного доступа к информации путём шифрования содержания секретных сообщений, то цель стеганографии – в скрытии самого факта существования секретного сообщения.

При необходимости оба способа могут быть объединены и использованы для повышения эффективности защиты информации.

Компьютерная стеганография базируется на двух основных принципах.

Первый принцип заключается в том, что файлы, содержащие оцифрованное изображение или звук, могут быть до некоторой степени видоизменены без потери их функциональности в отличие от других типов данных, требующих абсолютной точности.

Второй принцип заключается в неспособности органов чувств человека различать незначительные изменения в цвете изображения или качестве звука. Этот принцип особенно легко применять к изображению или звуку, несущему избыточную информацию.



Структурная схема стегосистемы представлена на рис.1.

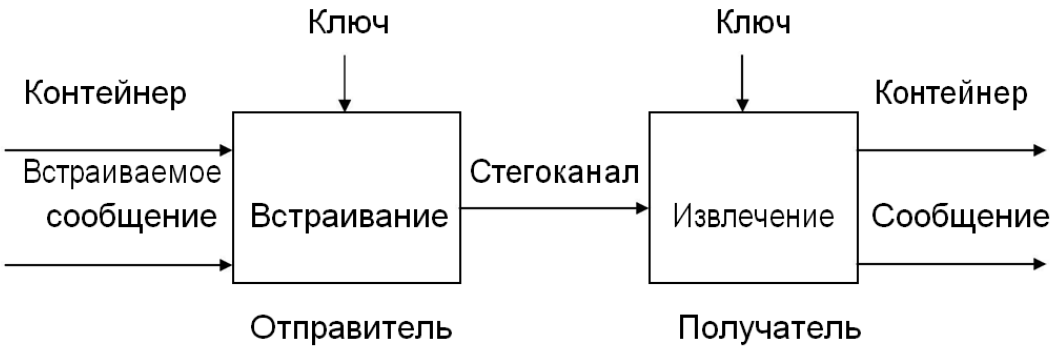


Рис.1. Структурная схема стегосистемы

Задание.

1. Исследовать основные понятия и возможности компьютерной и цифровой стеганографии.

Основные понятия: контейнер, виды контейнеров, встроенное сообщение, стеганографический канал (стегоканал), ключ.

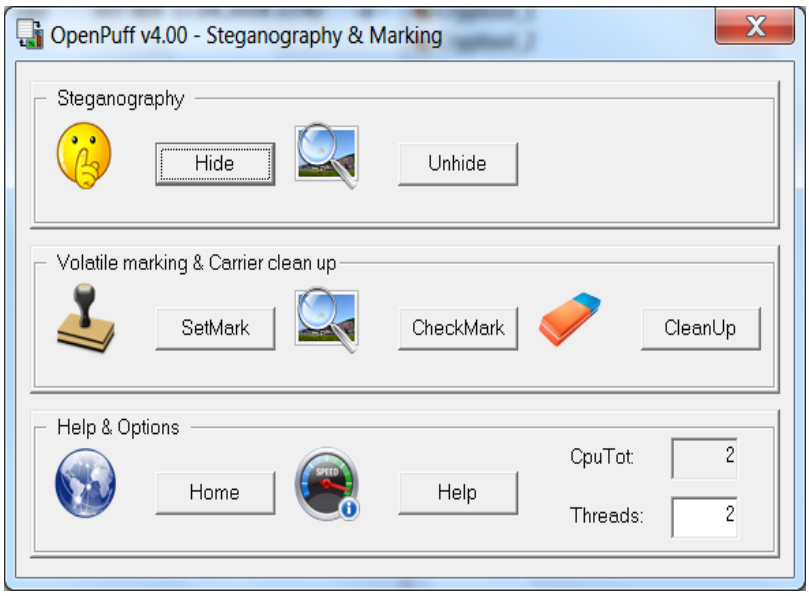
Методы стеганографии: метод замены младших бит (Least Significant Bit) и другие.

Воспользуйтесь ссылками:

- Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. - К.: "МК-Пресс", 2006. - 288 с. - http://web.archive.org/web/20140221205846/http://er.nau.edu.ua/bitstream/NAU/8049/1/CompSteganoRU.pdf
- Стеганография в XXI веке. Цели. Практическое применение. Актуальность - https://habrahabr.ru/post/253045/

2. Сделать обзор программ компьютерной (цифровой) стеганографии (таблица в отчете).

3. С помощью программы OpenPuff скрыть данные Вашего файла (.*txt, *.doc).



OpenPuff - профессиональный инструмент для стеганографии. Поддерживает два основных режима: скрытие данных в файле и подписывание файла (Цифровой водяной знак).

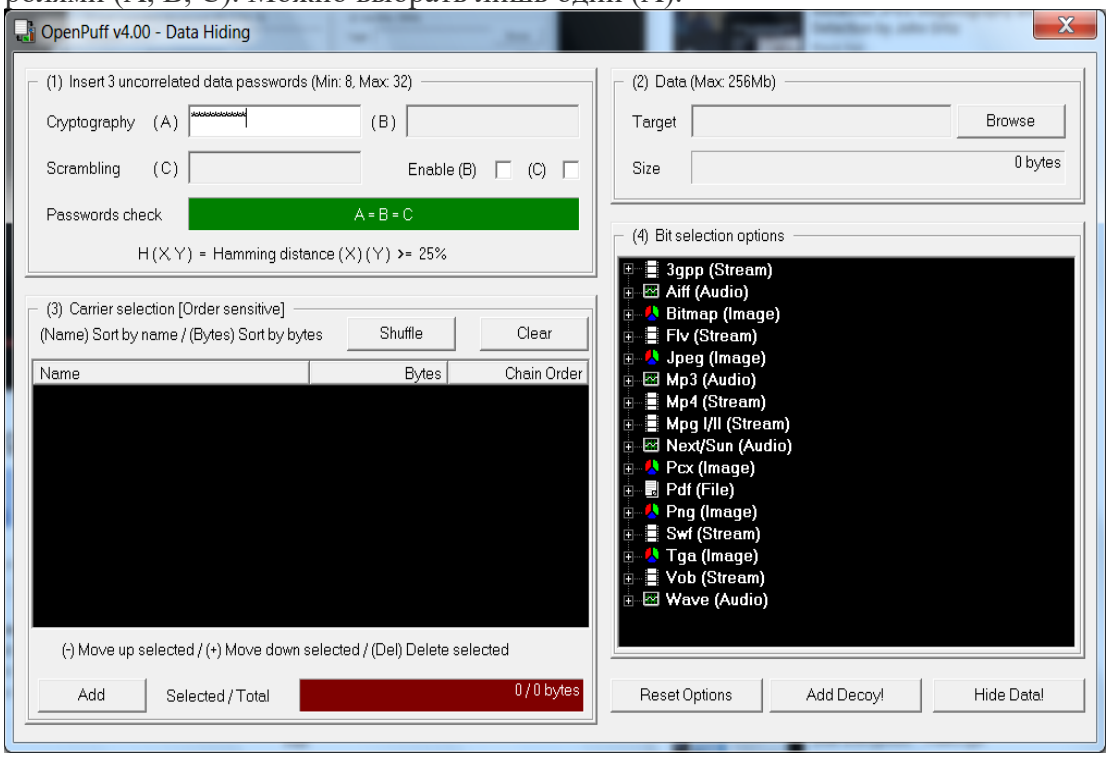
OpenPuff работает с такими форматами, как:

- изображения (BMP, JPG, PCX, PNG, TGA);
- аудио (AIFF, MP3, NEXT/SUN, WAV);
- видео (3GP, MP4, MPG, VOB);
- Flash (FLV, SWF, PDF).

3.1 Запустите файл OpenPuff.exe.

В открывшемся окне выберите Hide (функция для скрытия).

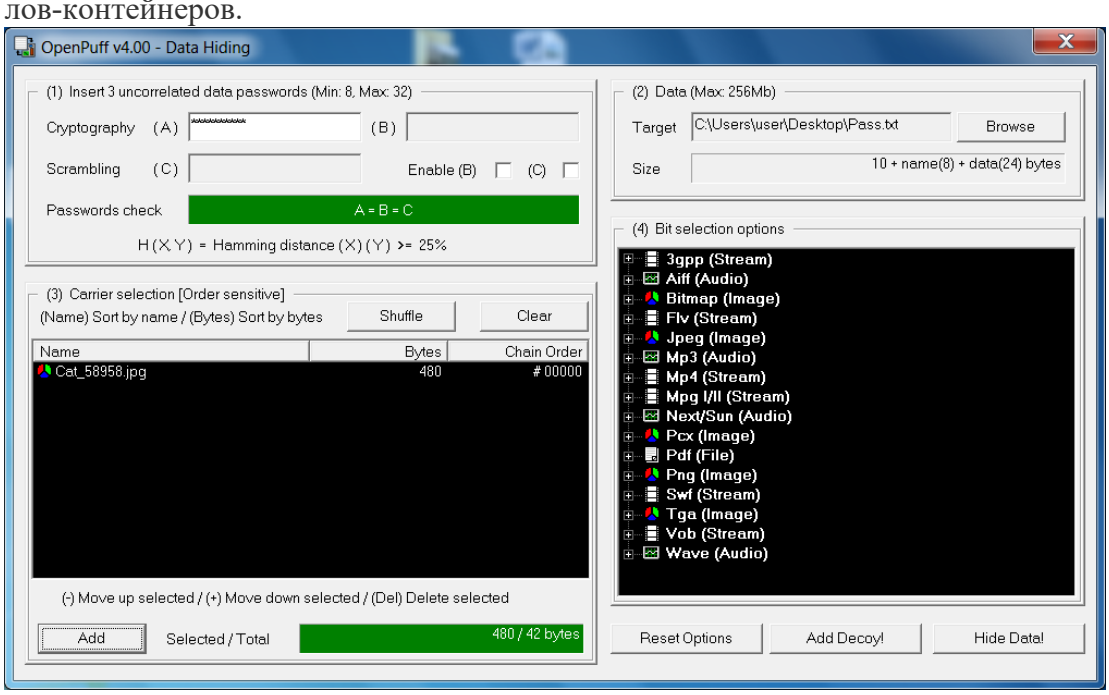
По умолчанию OpenPuff предложит Вам защитить информацию тремя различными паролями (A, B, C). Можно выбрать лишь один (A).



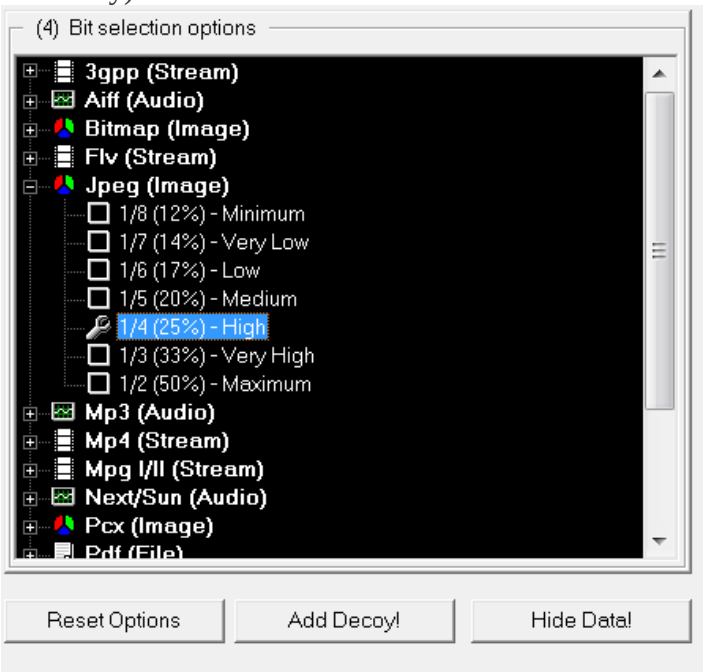
3.2 Выберите файл (doc), который будете скрывать: (2) Data / Browse.

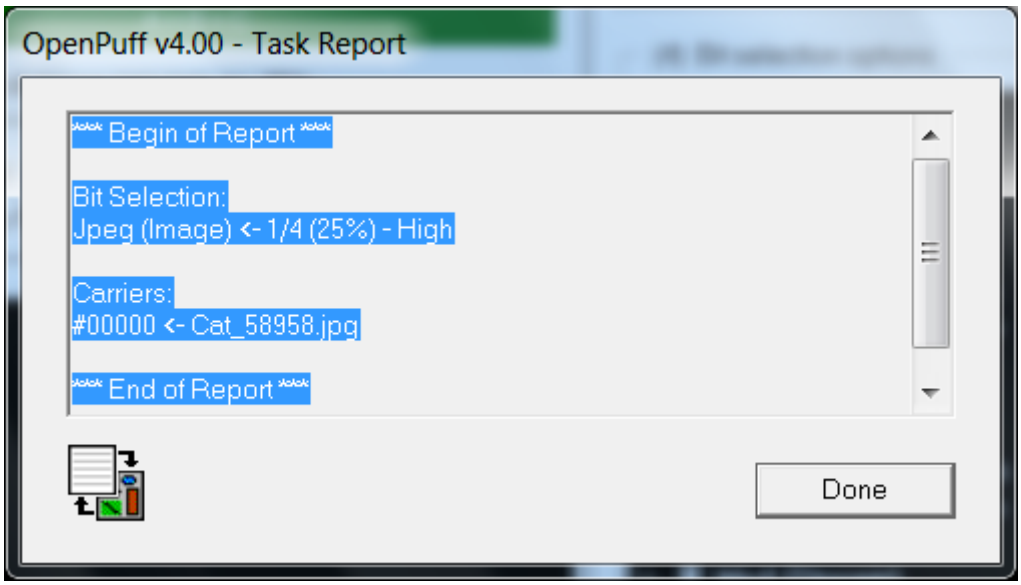
3.3 Выберите файл-контейнер (изображение *.jpeg, *.png, *.bmp): (3) Carrier selection / Add.

Если Вам нужно скрыть большой объем информации, то нужно выбрать несколько файлов-контейнеров.



3.4 В "Bit selection options" выберите соответствующий формат файла-контейнера. В открывшемся списке установите опцию " 1/4 (25%) - High", нажмите "Hide Data" и укажите на диске папку, в которой будет сохранен файл со скрытыми данными (Select Output Directory) .





Скрытие данных завершено при появлении отчета.

3.5 Откройте исходный файл-контейнер и файл со встроенными данными. Сравните их.

4. Извлеките информацию из файла-контейнера.

4.1 Запустите файл OpenPuff.exe.

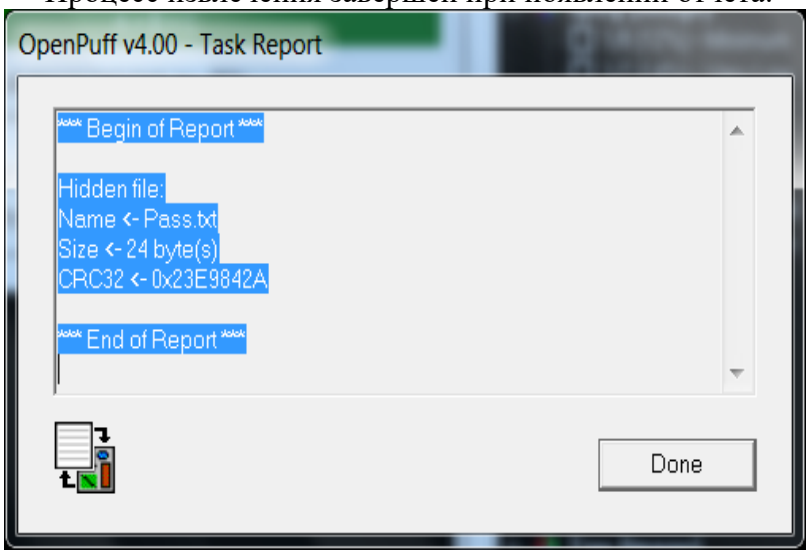
4.2 В открывшемся окне выберите UnHide. Введите пароль (A).

4.3 Выберите файл-контейнер со скрытой информацией: Add Carriers.

4.4 В "Bit selection options" выберите соответствующий формат файла-контейнера, установите опцию " 1/4 (25%) - High" и нажмите "UnHide".

4.4 Укажите на диске папку, в которой будет сохранен исходный файл (Select Output Directory).

Процесс извлечения завершен при появлении отчета.



4.5 Проверьте содержание исходного файла после извлечения.

2.20 Практическая работа № 20 Решение ситуационных задач

Задание:

BMP (англ. **Bitmap** Picture) - это стандартный, не сжатый битовый графический формат, используемый в Windows.

JPG (англ. **Joint Photographic Experts Group**, по названию организации-разработчика) - это растровый формат записи и хранения графических изображений, высокая степень сжатия в котором достигается за счет потери качества изображения. Основан на кодировании плавных цветовых переходов и позволяет в разы уменьшить объем данных при записи изображения в файл.

Определите требуемый объем видеопамяти (в Кб) для различных графических режимов экрана монитора с заданным разрешением экрана. Заполните таблицу в тетради.

Рисунок	**Разрешение экрана**	**Глубина цвета (бит на точку)**	**Объем видеопамяти (в Кб)**		**Размер файла *.jpg**
			свойства файла	**по формуле**	
Черно-белый *Домик.bmp*	640*480	24			
Цветной *Домик2.bmp*	640*480	24			
Цветной *Домик3.bmp*	640*480	4			-
Цветной *Домик4.bmp*					-

1. Откройте графический редактор Paint.
2. Задайте размер рисунка 640*480 точек.
3. Нарисуйте черно-белый домик и сохраните его (*Сохранить как*) в своей папке с именем *Домик.bmp*, указав тип файла *24-разрядный рисунок (*.bmp;*dib)*. Впишите в таблицу размер получившегося файла.

4. Разукрасьте домик, дорисуйте произвольные объекты (солнышко, забор и т.п.). Сохраните его (*Сохранить как*) в своей папке с именем *Домик2.bmp*, указав тип файла *24-разрядный рисунок (*.bmp;*dib)*. Впишите в таблицу размер получившегося файла.
5. Рассчитайте объем видеопамяти **I** (в Кб) для данного рисунка по формуле **I = K * i**, где **K** – разрешение экрана, **i** – глубина цвета. Впишите в таблицу получившееся значение (расчеты записать в тетрадь).
6. Сравните размеры ч/б и цветного рисунков; размер файла по формуле и через свойства. Запишите вывод о сравнении объемов (*Почему?*).
7. Откройте и сохраните в формате *jpg* файлы *Домик.bmp*, *Домик2.bmp*. Впишите в таблицу размеры получившихся файлов. Сравните с форматом *bmp.*
8. Откройте файл *Домик2.bmp* и сохраните его в своей папке с именем *Домик3.bmp*, указав тип файла *16-цветный рисунок (*.bmp;*dib)*. Впишите в таблицу размер получившегося файла.
9. Рассчитайте по формуле объем видеопамяти (в Кб) для рисунка *Домик3.bmp*. Впишите в таблицу получившееся значение (расчеты записать в тетрадь).
10. Сравните размеры файлов *Домик2.bmp* и *Домик3.bmp.* Объясните различие объемов файлов.

Как изменится размер файла *Домик2.bmp* (во сколько раз), если размер рисунка уменьшить до 320*240 точек? Ответ подтвердить с помощью формулы (и) или преобразованием рисунка.

2.21 Практическая работа № 21 Применение LSB-стеганографии

Задание:

1.Произвести вложение и извлечение информации при различных скоростях вложения.

2. Оценить эффективность обнаружения факта вложения при использовании различных атак (визуальное обнаружение без преобразований, визуальное обнаружение после преобразования к двоичному изображению, атака по критерию c^2, атака 2-го порядка с учетом корреляции яркостей пикселей).

Порядок выполнения

1.Для начала выполнения работы перейти в каталог, содержащий рабочие программы **ЛабСтег/LSB(1)**. Запустить программу **test.exe**. Рекомендуется скопировать папку ЛабСтег на рабочий стол учебного компьютера.

2.Нажать кнопку «Открыть файл» и выбрать один из предложенных файлов, содержащих изображения, с именем NN.bmp. Нажать кнопку «Вложить сообщение». В появившемся диалоговом окне выбрать вид вкладываемой информации – текст. Ввести сообщения, которое будет вложено. Выбрать вероятность вложения 100%.

3.Нажать кнопки «Визуальная атака». Сравнить покрывающее сообщение и стеганограмму, а так же визуальные атаки на эти изображения. Нажать кнопку «Статистические атаки», в появившемся диалоговом окне выбрать обе атаки. Нажать кнопку «Таблица» и записать результаты статистических атак.

4.Повторить пункты 1 и 2 для вероятностей вложения 50%, 10%, 5%, 2%. Перед каждым новым вложением нажимать кнопку «Очистить» и открывать файл NN.bmp заново. Сделать выводы об эффективности визуальной атаки при различных вероятностях вложе-

ния. По завершению записи всех данных статистических атак нажать кнопку «Очистить таблицу».

5.Открыть файл с вложением NN100%.bmp и нажать кнопку «Извлечь сообщение». Ничего не менять в появившемся диалоговом окне. Проверить, что выдается то сообщение, которое было вложено.

6.Нажать кнопку «Очистить».

7.По аналогии с пунктами 1 и 2 провести статистические атаки на стеганограммы с вероятностью вложения 1%, 0,1%, 0,05%, 0,01% и 1 двоичный символ для трех различных покрывающих сообщений. Проанализировать эффективность статистических атак.

8.Проведите процедуру удаления сообщения, вложенного в стеганограмму про 100% скорости вложения. Убедитесь, что при извлечении информации получается случайный набор символов.

Описание выполнения лабораторной работы

Программа работает с файлами формата BMP оттенки серого (8 разрядов) с параметрами изображения 300x200. Общее количество пикселей в таких изображениях равно 60000. Как уже говорилось выше, формат BMP представляет собой точечный рисунок. В данном формате один пиксель описывается одним байтом. Максимальное количество секретной информации, которое можно вложить в файл – 60000 бит, при этом каждый пиксель изображения будет содержать информацию. Программа позволяет вложить информацию в каждый пиксель с вероятность вложения 100% или с меньшей вероятностью, в случайные пиксели изображения.

Главное окно программы показано на рисунке 1.

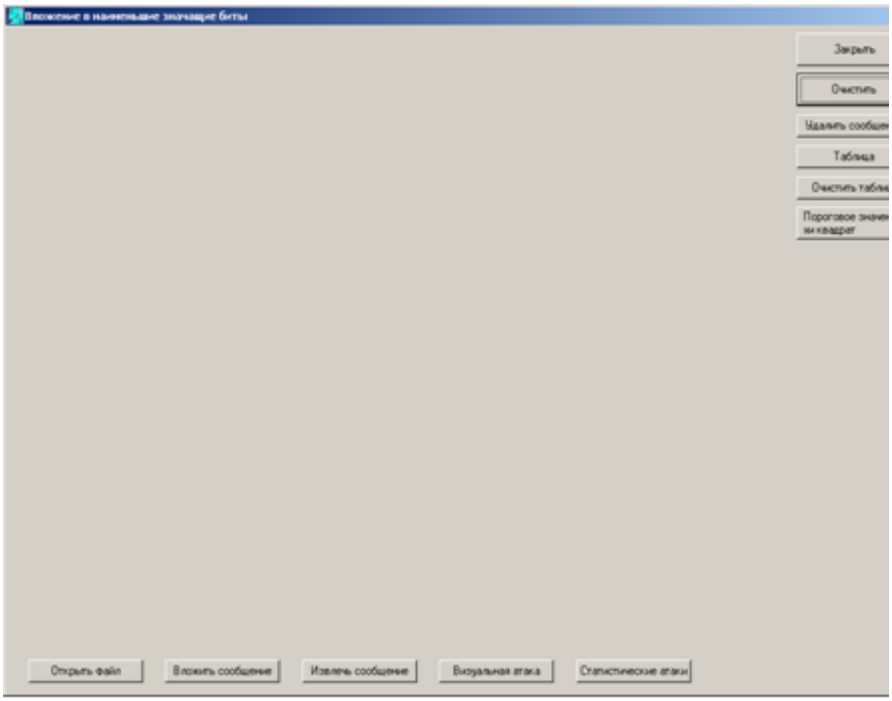


Рис.1. Главное окно лабораторной работы.

Рассмотрим более подробно кнопки главного окна.

Кнопка «Открыть файл» позволяет открыть любой файл формата BMP оттенки серого (8 разрядов) размером 300x200 пикселей. Открываемый файл может быть покрывающим сообщением, в которое нужно вложить информацию, возможной стеганограммой, которую нужно проверить на наличие вложения, или стеганограммой, полученной легальным пользователем, из которой надо извлечь секретное сообщение. Изображение открытого файла появляется в левом верхнем углу (рисунок4).

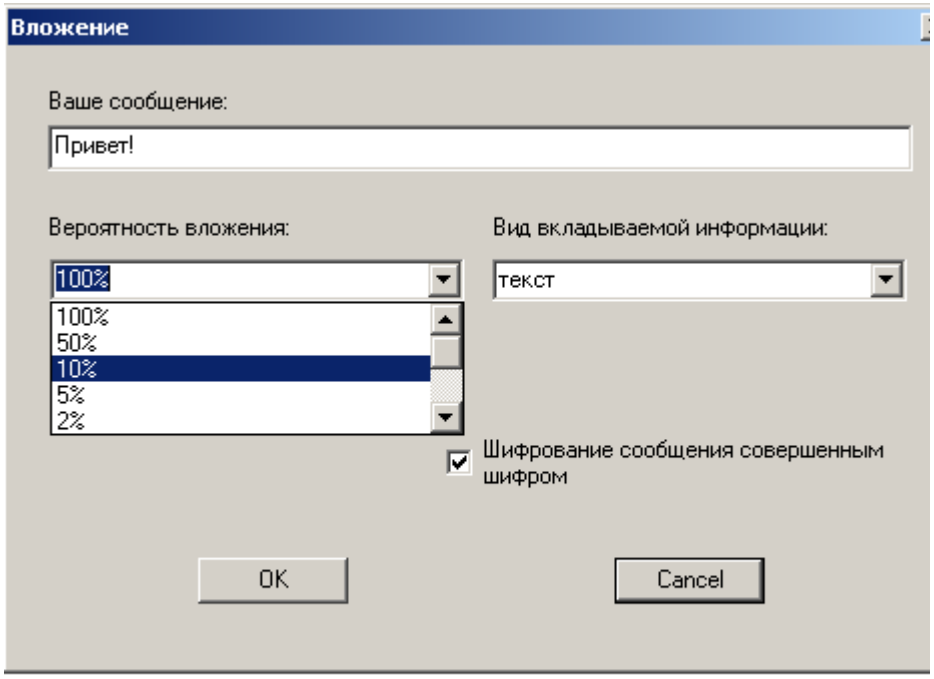


Рис. 2. Окно вложения сообщения.

Кнопка «Вложить сообщение» позволяет вложить информацию в покрывающее сообщение. В качестве вкладываемого сообщения можно использовать как текст, так и двоичную последовательность, состоящую 0 и 1. Вероятность вложения можно выбрать из набора: 100%; 50%; 10%; 5%; 2%; 1%; 0,1%; 0,05%; 0,01%. Введенное сообщение повторяется, чтобы получилась выбранная вероятность вложения. Если при выбранной вероятности вложения количество бит, которое можно вложит, меньше бит сообщения, то последние биты сообщения не будут вложены. При желании можно зашифровать сообщение совершенным шифром. Диалоговое окно, появляющееся при нажатии кнопки «Вложить сообщение», показанно на рисунке 2. Полученная в результате вложения стеганограмма появится справа от покрывающего сообщения (рисунке 5).

Кнопка «Извлечь сообщение» позволяет легальному пользователю извлечь сообщение. До извлечения сообщения надо открыть файл со стеганограммой. Легальному получателю стеганограммы должно быть известно, было ли зашифровано сообщение совершенным шифром или нет. Так же получатель должен знать, в каком виде вкладывалось сообщение – в виде текста или двоичной последовательности. Выбор данных параметров производится с помощью диалогового окна, показанного на рисунке 3.

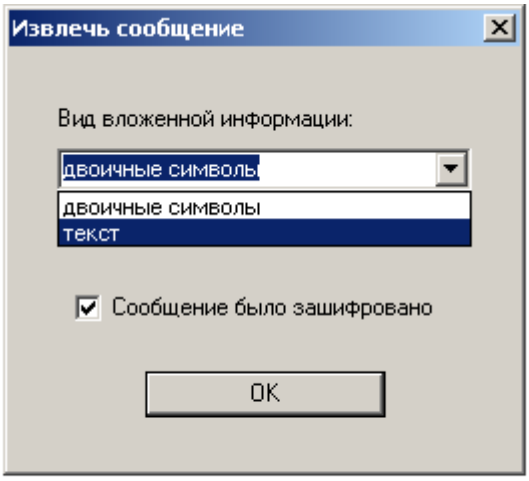


Рис. 3. Окно извлечения сообщения

Сообщение, извлеченное из стеганограммы, появится справа от изображения, как показано на рисунке 4.

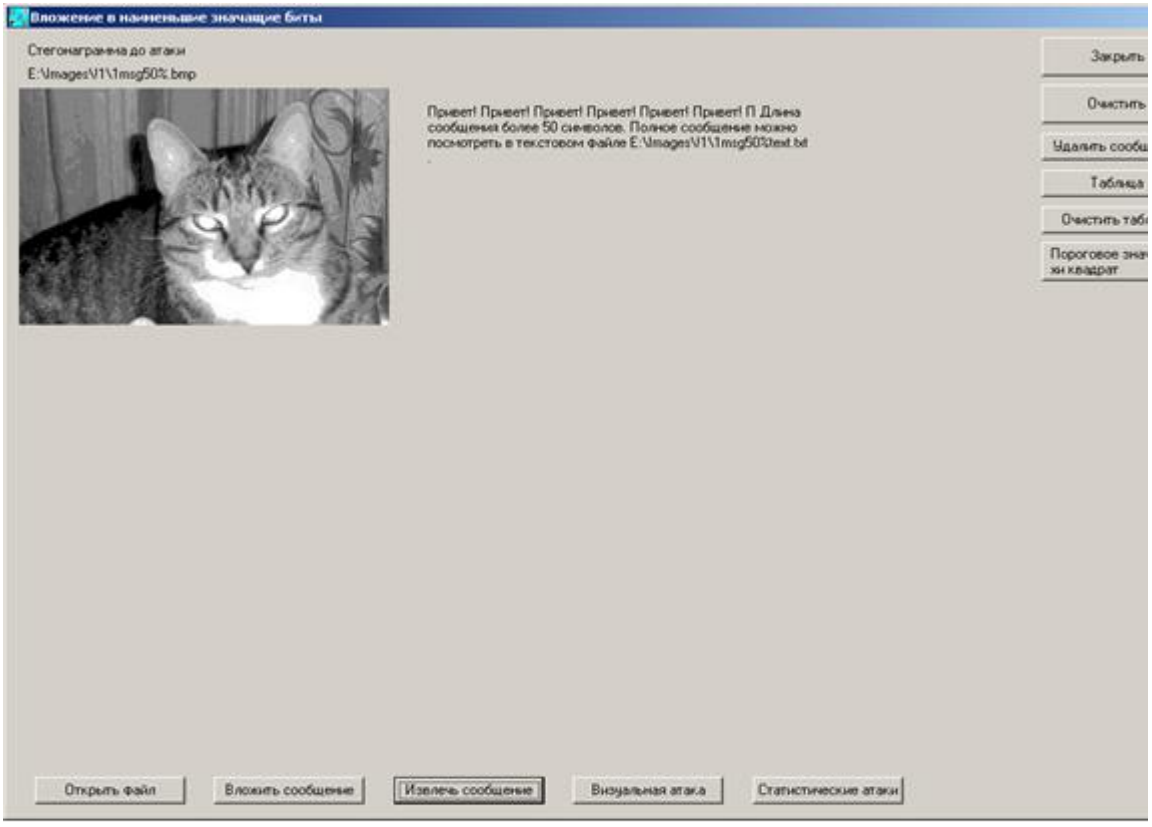


Рис. 4. Извлечение сообщения.

Если вложенное сообщение повторялось несколько раз, при извлечение сообщения получится периодически повторяющийся текст или периодически повторяющаяся двоичная последовательность. Если количество символов в извлеченном тексте или двоичной последовательности более пятидесяти, то появится только пятьдесят первых символов (чтобы не загружать экран), и будет указан путь к файлу, в котором можно прочитать полное извлеченное сообщение

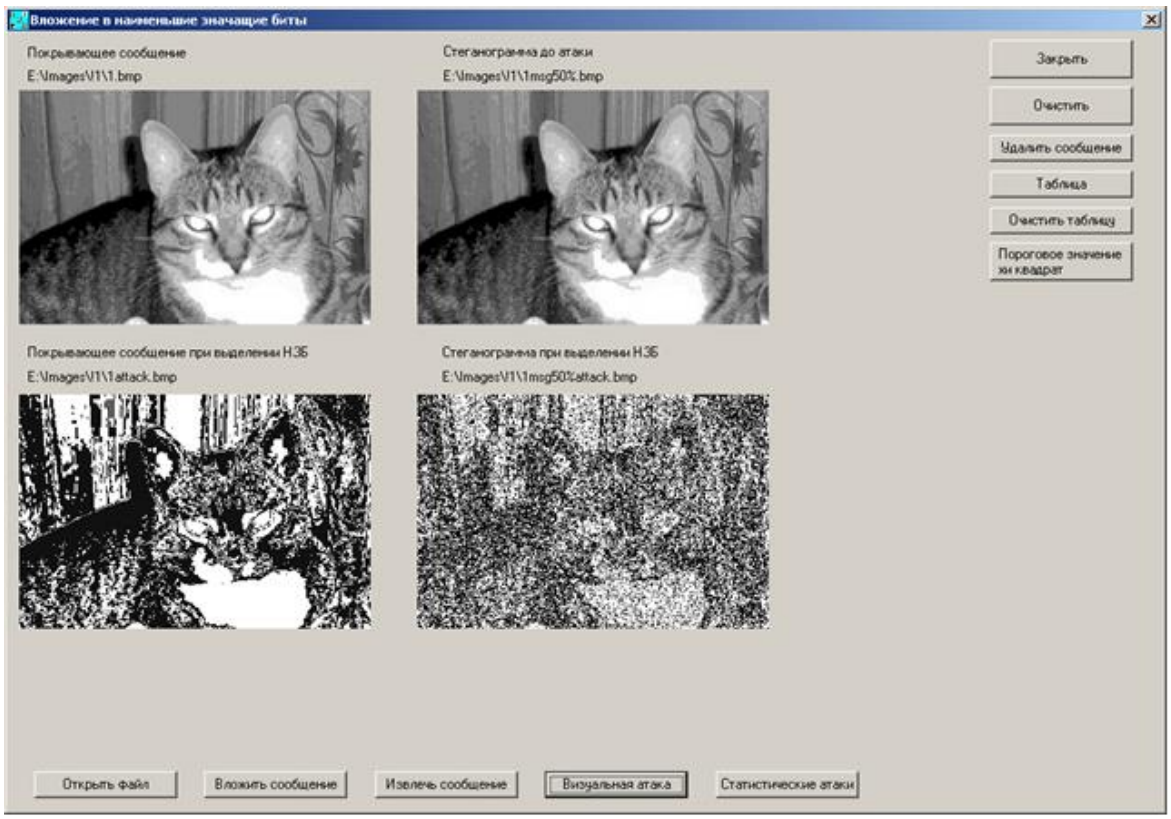


Рис. 5. Главное окно с результатами визуальной атаки.

Кнопка «Визуальная атака» позволяет провести визуальную атаку. Изображения, полученное в результате атаки появятся под атакованным изображением. Главное окно с результатами проведения визуальной атаки показано на рисунке 5.

Атаку можно произвести на покрывающее сообщение или стеганограмму, в зависимости от того, какой файл был открыт. Если вначале открыть покрывающее сообщение, а потов вложить сообщение, то в главном окне будет отображаться изображения покрывающего сообщения (слева) и стеганограммы (справа). Тогда визуальную атаку можно провести на оба изображения сразу, что позволяет сравнит результаты атаки для покрывающего сообщения и стеганограммы.

Конечно, у атакующего нет покрывающего сообщения, а значит нет возможности сравнения результаты атак. Но для студентов сравнение дает возможность понять, на какие особенности изображения предполагаемой стеганограммы после визуальной атаки стоит обратить особое внимание. Так же сравнение помогает выявить насыщенность возможного шума при различных вероятностях вложения, понять, при какой вероятности после визуальной атаки можно уверенно сказать, что в изображении есть вложение, а при какой стоит провести другие, более эффективные атаки, например одну из статистических атак или обе сразу.

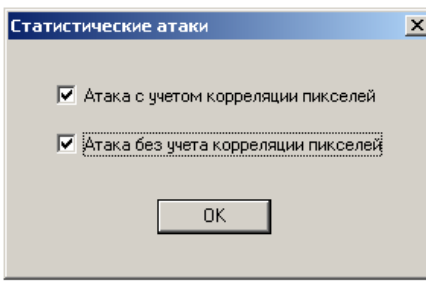


Рис. 6 Статистические атаки.

Кнопка «Статистические атаки» позволяет провести статистическую атаку без учета корреляции пикселей, основанную на гистограммах изображений, и статистическую атаку с учетом корреляции пикселей. Можно провести одну из атак, поставив галочку напротив выбранной атаки, а можно обе атаки одновременно, поставив две галочки напротив двух атак сразу.

Диалоговое окно для выбора атаки показано на рисунке 6.

Атаковать можно одно изображение (покрывающее сообщение) или стеганограмму. Атака на покрывающее сообщение позволяет набрать статистику для выбора порогового значение c_{a^2}. Можно провести атаку на оба изображения сразу и сравнить полученные результаты.

Результаты статистических атак заносятся в таблицу. Таблица вызывается нажатием кнопки «Таблица». В таблице 10 строк. Если две атаки проводились одновременно, то в строке будут заполнены оба столбца, если проводилась только одна атака, то результат атаки появится в столбце, соответствующем этой атаке. Слева, напротив результатов атак, появится полное имя атакуемого файла.

Таблица появляется в отдельном окне, поверх главного окна. Внешний вид таблицы с результатами двух атак на покрывающее изображение и стеганограмму показан на рисунке 7.

С помощью таблицы можно сравнить эффективность статистической атаки, основанной на гистограммах изображений, и статистической атаки с учетом корреляции пикселей. В таблице можно накопить данные, например, по значению c^2 покрывающих сообщений, для выбора порогового значения c_{a^2}.

Так же можно сравнивать различные результаты атак на различные предполагаемые стеганограммы, или сравнить результаты атак на покрывающее сообщение и стеганограмму.

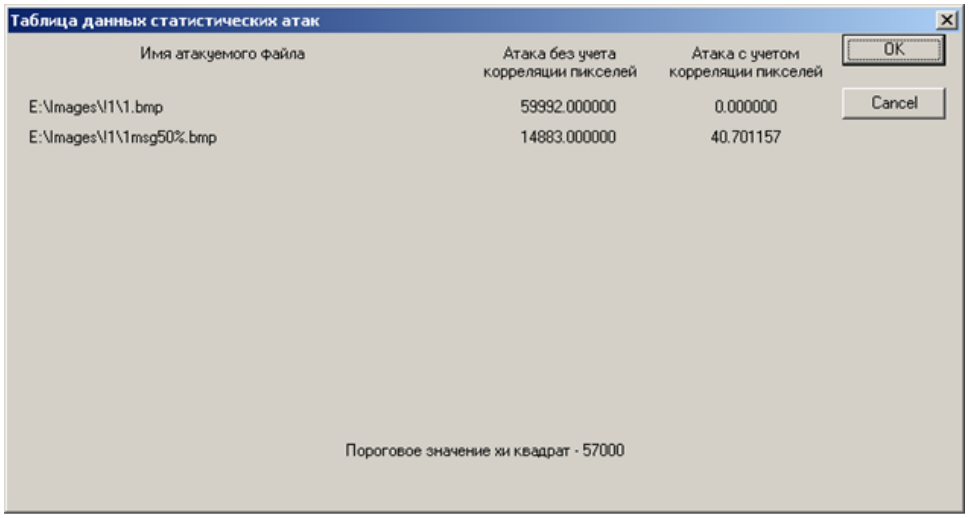


Рис. 7. Таблица данных статистических атак.

В нижней строчке таблицы выведено пороговое значение для атаки, основанной на гистограммах изображений, – c_{a^2}. Его можно выбрать, проанализировав c^2 различных покрывающих сообщений.

Пороговое значение вводится с помощью кнопки «Пороговое значение хи квадрат», находящейся в главном окне программы. При нажатии кнопки появляется диалоговое окно, показанное на рисунке 8.

До введения порогового значения нижняя строчка в таблице будет пустая. Введенное значение c_{a^2} будет отражаться в нижней строчке таблицы. Результаты, полученные после статистической атаки, основанной на гистограммах изображений, можно сравнить с пороговым значением и сделать вывод о наличие или отсутствии вложения.

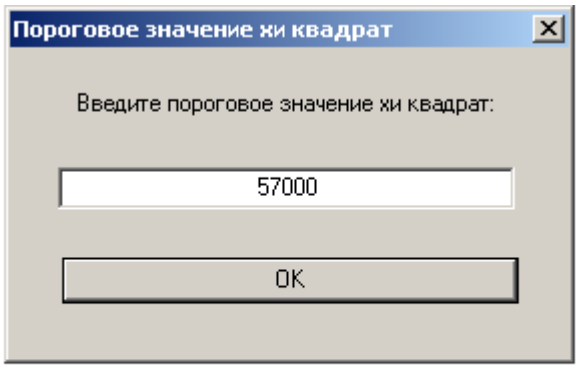


Рис. 8. Пороговое значение хи квадрат

В любой момент пороговое значение можно изменить, для этого надо вновь нажать кнопку «Пороговое значение хи квадрат» и в появившемся диалоговом окне ввести новое значение c_{a^2}. Потом в нижней строчке таблицы будет отображаться новое пороговое значение.

С помощью кнопки «Очистить таблицу» можно удалить из таблицы все ранее занесенные в нее данные, что позволяет начать накапливать статистические данные заново. При этом нижняя строчка, содержащая пороговое значение c_{a^2}, не исчезнет, что позволяет проверить много изображений на наличие вложения с одним и тем же пороговым значение c_{a^2}.

Кнопка «Удалить сообщение» позволяет удалить сообщение из стеганограммы. Атаку по удалению сообщения можно применить для любого файла с изображением. При этом не обязательно быть уверенным, что это стеганограмма.

При атаке по удалению сообщения все наименьшие значащие биты атакованного файла заполняются случайным образом 0 и 1, при этом не важно, в каких именно битах было вложено сообщение, поскольку изменяются все без исключения пиксели. Из стеганограммы после проведения атаки по удалению сообщения извлечь первоначальный текст или двоичную последовательность уже нельзя.

Результат проведения атаки по удалению показан на рисунке 9.

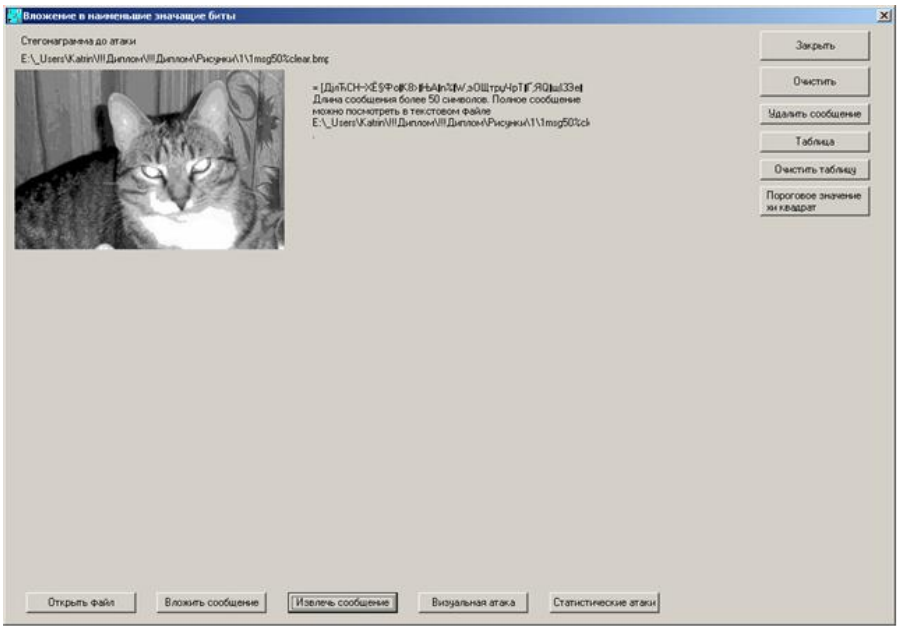


Рис. 9. Результат атаки по удалению сообщения.

На рисунке 9 изображено главное окно с стеганограммой, после удаления из нее сообщения. Изначально в стеганограмме было вложено сообщение «Привет! » с вероятностью вложения 50% и зашифровано совершенным шифром.

Если бы стеганограмма не подверглась атаке по удалению сообщения, то при извлечении легальный пользователь получил бы периодически повторяющийся текст «Привет! Привет! Привет!…». Но так как сообщение было удалено, вместо осмысленного текста легальный пользователь получает набор случайных символов «= [ДjлЋСН— ХЁ§ФоК8>ЊА…».

Данная атака очень удобна, если основной задачей атакующего является не допустить передачу вложенного сообщения. Ее можно применять ко всем изображениям, и быть уверенным, что даже если какой-то файл и содержал секретное сообщения, после атаки легальный пользователь все равно не сможет его причитать.

При нажатии кнопки «Очистить» очищается главный экран программы, при этом таблица не меняется, и данные из нее не удаляются.

При нажатии кнопки «Закрыть» закрывается главное окно, программа прекращает работу. Все статистические данные, находящиеся в таблице и пороговое значение c_{a^2} после завершения работы программы удаляются, их уже не восстановить. Но все ключи (стегоключ и, если он был, ключ для совершенного шифра) и созданные изображения, такие как стеганограммы с вложенной информацией, изображения после визуальных атак и так далее, сохраняются на диске. При желании их можно посмотреть любыми программными средствами, которые соответствуют их форматам.

Отчет

1.Вкладываемый текст и текст, полученным при извлечении.

2.Данные по всем проведенным статистическим атакам.

3.Выводы об эффективности различных атак.

2.22 Практическая работа № 22 Применение метода замены цифровой палитры

Задание:

1.Ознакомиться со словарем синонимов русского языка.

2.Наблюдать изменение коротких (специально подобранных) фраз, в зависимости от изменения короткой двоичной цепочки вкладываемой в них секретной информации.

3.Произвести вложение заранее выбранной 10-битовой последовательности в один из текстов, используя специальную программу, оперирующую с лингвистической базой данных (словарем синонимов).

4.Произвести извлечение 10-битовой последовательности из полученной в п.3 стеганограммы.

5.Оценить скрытность секретной информации и скорость ее вложения.

Порядок выполнения

Для начала работы перейти в каталог, содержащий рабочие программы: **ЛабСтег/LingvLab(3)**. Для работы используются программы: *Information Processor, Information Retriever*.

1.Для знакомства со словарем синонимов перейти в подкаталог «TestTexts» и открыть файл «_SynonymDictionary» для чтения.

2.Для демонстрации метода вложения короткой цепочки бит в четыре, заранее подобранные фразы:

- Запустить программу «*Information Processor*». (Для работы программы необходимо после запуска загрузить Текстовый файл – контейнер и словарь синонимов).

- Произвести вложение различных битовых последовательностей в текстовый файл «TestSentence».

- Наблюдать изменения слов во фразах при сохранении основного содержания последних.

Сделать пометки, если, на ваш взгляд, содержание фраз хотя бы незначительно изменится или произойдут нарушения грамматики языка.

3.Для демонстрации автоматического вложения скрытной информации в смысловые тексты значительного объема на основе использования словаря синонимов:

- Запустить программу «*Information Processor*».

- Создать произвольную двоичную цепочку длиной 10 бит для погружения в выбранный смысловой текст.

- Произвести вложение битовой последовательности в текстовый файл «Text_n_Author» (где n – номер бригады).

4.Произвести сравнение текста, полученного в ходе вложения, с оригиналом (Compare).

Сделать выводы о сохранении (или нет) основного содержания и грамматики текста. Рассчитать скорость вложения секретной информации в битах на байт текста.

5.Сохранить полученную стеганограмму в некотором файле.

6.Извлечь стеганограмму из файла и произвести декодирование скрытой в ней информации, запустив программу *Information Retriever.*

(Для работы программы необходимо после запуска загрузить Текстовый файл – стеганограмму и словарь синонимов).

Сравнить выделенную информацию с той, которая была вложена соседней бригадой.

7.Сделать выводы о незаметности (или заметности) вложения секретной информации.

Отчет

1.Титульный лист.

2.Исходные короткие фразы, вложенная в них секретная информация и соответствующие ей стеганограммы.

Выводы по секретности вложения и сохранения основного содержания и грамматики коротких фраз.

3.10-битовая цепочка Выводы по секретности вложения и сохранения основного содержания и грамматики текстов.

4.Расчет скорости вложения в бит/байт покрывающего текста.

5.10-битовая цепочка, выделенная из стеганограммы. Её совпадение (или нет) с вложенной цепочкой.

2.23 Практическая работа № 23 Анализ графических изображений на наличие скрытой информации.

Задание:

Приступая к работе скачайте и распакуйте архив PR5.zip, программа и тестовое изображение в нем. В ходе выполнения работы можно указывать программе, в какие компоненты JPEG изображения внедрить больше информации, а в какие меньше. Суммарное количество информации при этом остается прежним и меняется только ее распределение между компонентами изображения.

После любых изменений в настройках нажимайте на надпись: «Изображение с внедренным сообщением» и тогда на экран будет выведено это изображение, а справа от него будет выводиться количественная оценка его качества в дБ (PSNR - пиковое соотношение сигнал/шум). Для выполнения лабораторной работы необходимо, чтобы качество было больше 43 дБ.

С помощью ползунка можно выбирать баланс распределения информации между цветовой и яркостной компонентой. Определите, в какой из компонент искажения заметнее?

С помощью ползунка «Маскирование в текстурных участках» можно перераспределять информацию между однородными (небо) участками изображения и текстурными (листья, рябь на воде).

В восьми строках ввода можно для каждого коэффициента ДКП (при JPEG сжатии выполняется дискретное косинусное преобразование в блоках 8x8 пикселей и для каждого блока осуществляется квантование) отдельно задать долю внедрения туда информации. Эти коэффициенты должны быть положительными числами, большими, чем ноль. 0-й коэффициент соответствует самой низкой частоте, а 9-й - самой высокой. Чем больший коэффициент будет задан, тем больше информации будет внедрено в соответствующие частоты изображения. Определите, в каких частотах человеческий глаз лучше замечает искажения?

2.24 Практическая работа № 24 Применение ОС Kali Linux в стеганографии

Задание:

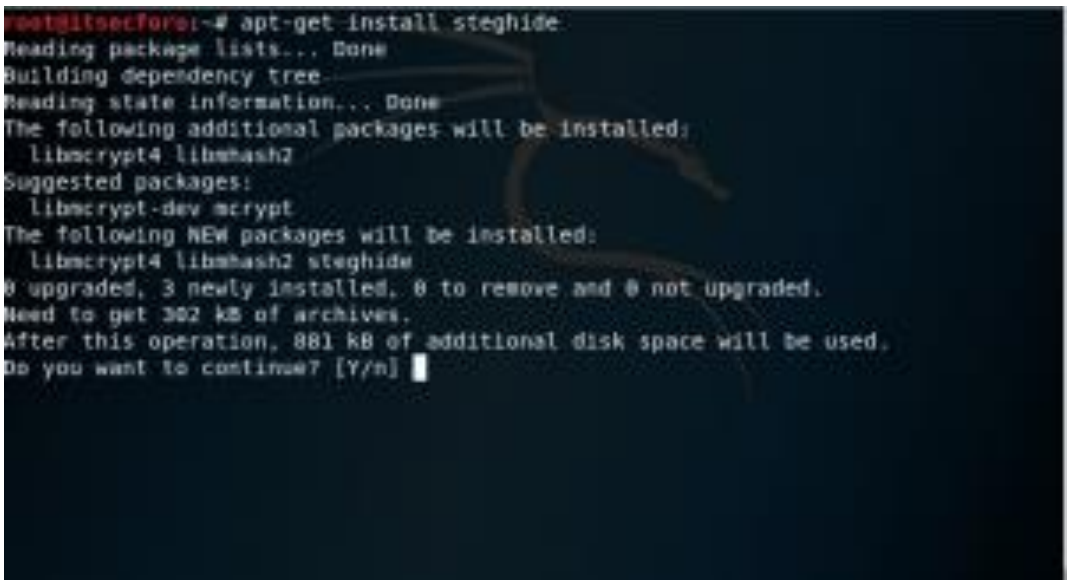
а. Steghide
В Kali Linux есть два основных инструментария для стеганографического использования.
Steghide – это программа стеганографии, которая позволяет скрыть данные в различных типах изображений и аудиофайлах.
Цветно-чувствительные частоты выборок не изменяются, что делает встраивание устойчивым к статистическим испытаниям первого порядка.
Особенности:

- сжатие встроенных данных
- шифрование встроенных данных
- встраивание контрольной суммы для проверки целостности экстрадированных данных
- поддержка файлов JPEG, BMP, WAV и AU

б. StegoSuite
Stegosuite – бесплатный инструмент стеганографии, написанный на Java. С помощью Stegosuite вы можете скрыть информацию в файлах изображений.
Особенности:

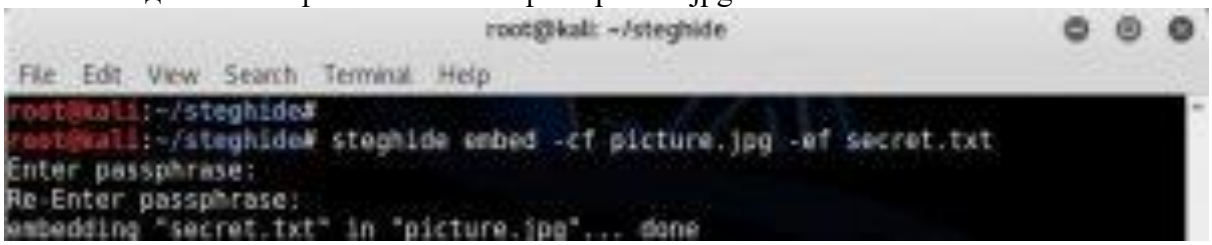
- Поддержка BMP, GIF и JPG
- Шифрование AES встроенных данны
- Автоматическое предотвращение однородных областей (только встраивание данных в шумные области)
- Встраивание текстовых сообщений и нескольких файлов любого типа
- Легко использовать

Скрытие данных в изображении с помощью steghide
Установка Steghide
Установка в Kali Linux проста, так как steghide уже доступен в репозитории Kali Linux.
Выполните следующую команду, и все готово:



Скрыть текстовый файл в изображении

Я создал папку steguide в корневой домашней папке и разместил файл picture.jpg и secret.txt там же.
picture.jpg – это файл, в котором я собираюсь скрыть файл secret.txt.
Я собираюсь показать здесь команды по всему этому безобразию.
Чтобы скрыть текстовый файл в картинке в Kali Linux с помощью steghide, используйте следующую команду:
steghide embed -cf picture.jpg -ef secret.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret.txt" in "picture.jpg"... done
Эта команда вставит файл secret.txt в файл picture.jpg.



Теперь вы можете отправлять, делиться или делать что-либо с этим новым файлом picture.jpg без необходимости беспокоиться о том, что кто-то сможет разоблачить ваши данные.

Извлечение текстового файла из изображения

После того, как вы внесете свои секретные данные, как показано выше, вы можете отправить файл picture.jpg человеку, который должен получить секретное сообщение.
Принимающий должен использовать steghide следующим образом:
steghide extract -sf picture.jpg
Enter passphrase:
the file "secret.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "secret.txt".
Если поставленная кодовая фраза верна, содержимое исходного файла secret.txt будет извлечено из файла picture.jpg и сохранено в текущем каталоге.
Чтобы быть в безопасности, я проверяю содержимое секретного файла, который я извлек.
После проверки убеждаемся,что все в порядке.
head -3 secret.txt

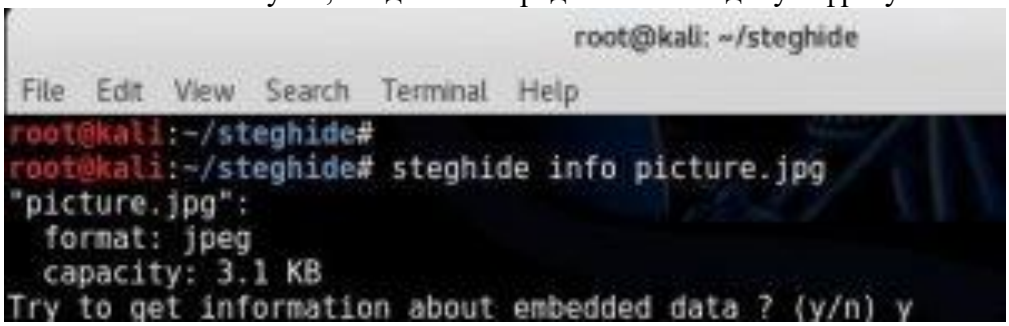
Просмотр информации о встроенных данных

Если вы получили файл, содержащий встроенные данные, и хотите получить информацию об этом до его извлечения, используйте команду info:

```
# steghide info picture.jpg
"picture.jpg":
format: jpeg
capacity: 3.1 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
embedded file "secret.txt":
size: 6.5 KB
encrypted: rijndael-128, cbc
compressed: yes
```

После вывода некоторых общих сведений о файле (формат, емкость) вам будет задан вопрос, следует ли steghide попытаться получить информацию о встроенных данных.

Если вы ответите «yes», вы должны предоставить кодовую фразу.



Затем Steghide попытается извлечь внедренные данные с помощью этой фразы и – если это удастся – выведет некоторую информацию об этом.

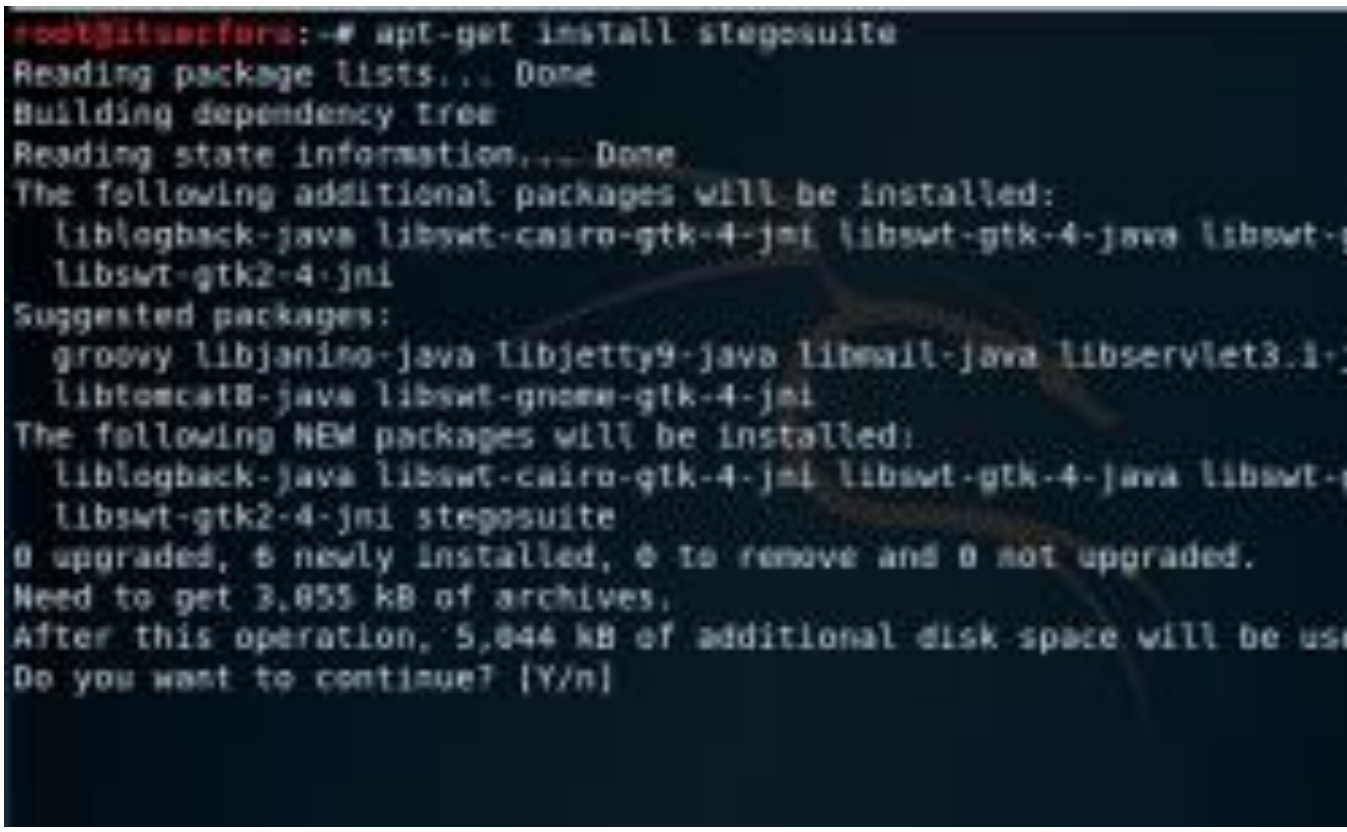
Скрытие данных на изображении с помощью Stegosuite

Установка stegosuite

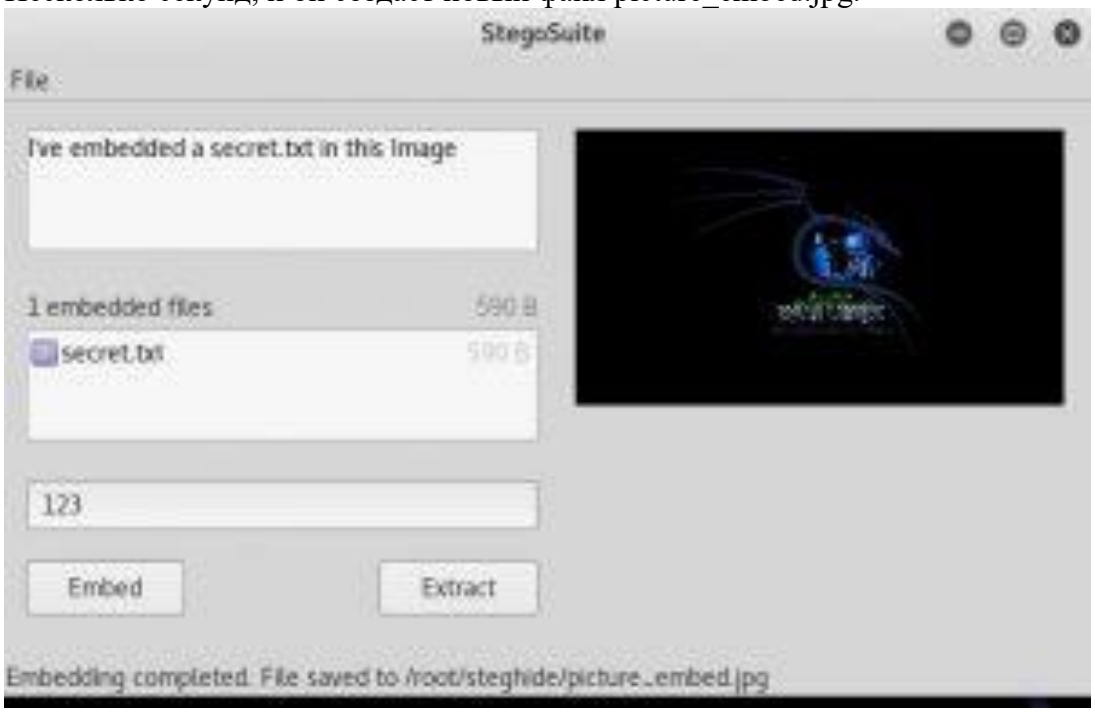
Установка в Kali Linux очень проста, так как stegosuite уже доступен в репозитории Kali Linux.

Выполните следующую команду, и все будет установлено:

```
#  apt-get install stegosuite
```



Вставить текстовый файл в изображение с помощью Stegosuite
Вам нужно запустить его из меню «Application» (или просто выполнить поиск). Откройте File > Open и откройте изображение, которое хотите использовать.
Щелкните правой кнопкой мыши раздел файла и выберите файл и выберите файл secret.txt.
Введите ключевую фразу и нажмите «embed».
Несколько секунд, и он создаст новый файл picture_embed.jpg.



Извлечение текстового файла из изображения с использованием Stegosuite
Если вы хотите извлечь текстовый файл или данные из изображения, просто откройте изображение, введите парольную фразу и нажмите «extract».

2.25 Практическая работа № 25 Решение ситуационных задач

Задание:

Для работы используется программа **LingvSteg.exe**в каталоге **ЛабСтег/ДемоЛСГ(4)**и дополнительное матобеспечение NetFramework 4.0.
1.Выбрать текстовый документ для вложения.
2.Установить основные параметры, используемые при вложении.
3.Задать короткое сообщение и произвести его вложение в текстовый документ.
4.Произвести извлечение информации из стеганограммы.
5.Оценить секретность СГ и скорость вложения информации оператором.

Панель установки параметров

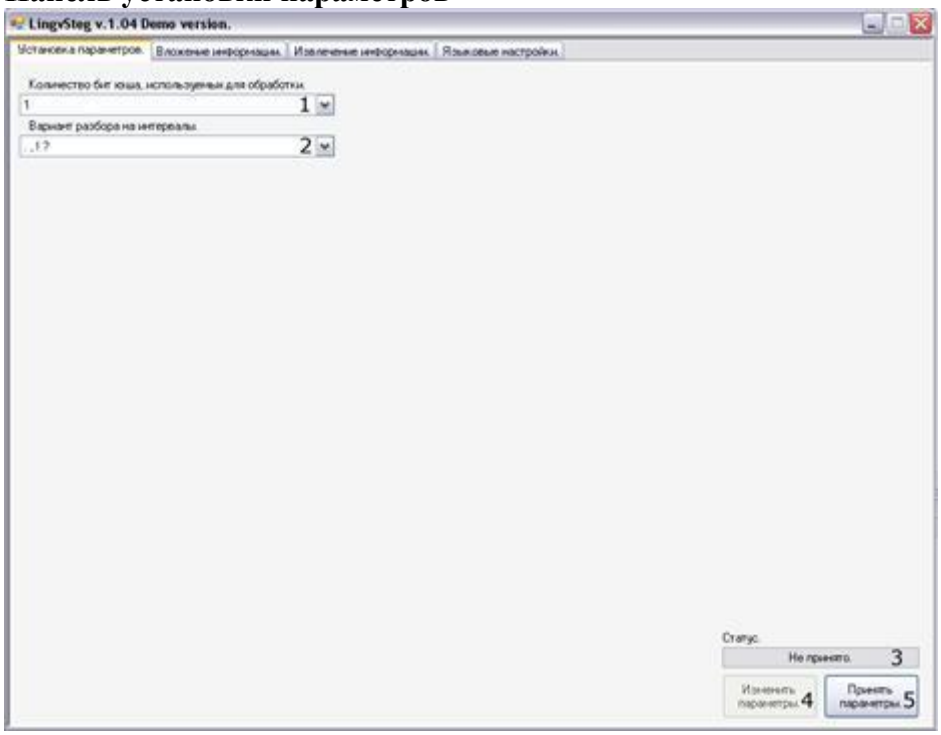


Рис. 10. Панель установки параметров.

1 - поле для выбора количества бит хэша, используемых для вложения. Используются первые биты хэша в указанном количестве.
2 - поле для выбора варианта разбора покрывающего сообщения на интервалы для хеширования.
3 - поле, показывающее успешно ли параметры были применены.
4 - кнопка для изменения уже принятых параметров.
5 – кнопка для принятия введенных параметров.

Вложение информации

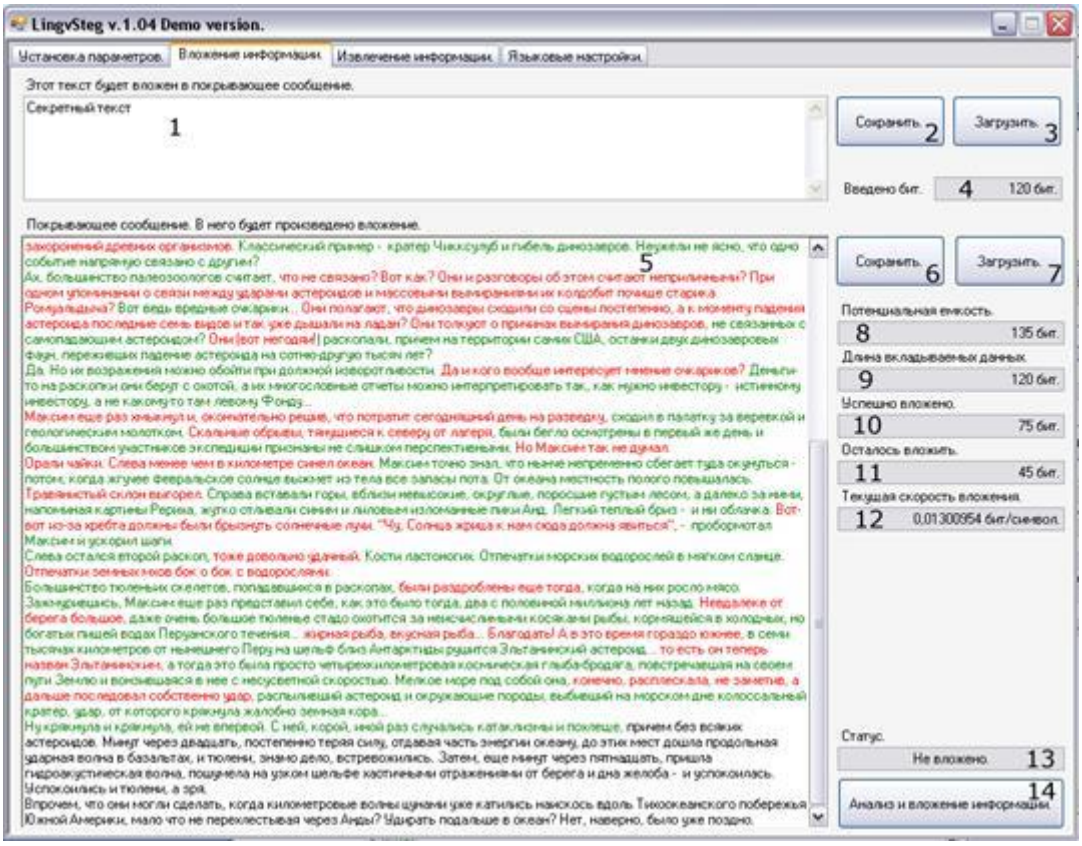


Рис. 11. Панель вложения информации.

1 - поле для ввода секретного сообщения, оно будет вложено в покрывающее сообщение.
2 - кнопка, позволяющая сохранить секретное сообщение в файл.
3 - кнопка, позволяющая загрузить секретное сообщение из файла.
4 - поле, показывающее сколько бит данных введено в поле для ввода секретного сообщения.
5 - поле для ввода и редактирования покрывающего сообщения. Цвета указывают на совпадение или несовпадение хэшей интервалов с подблоками секретного сообщения. Красный текст - текст, нуждающийся в дальнейшем редактировании, зеленый текст- текст, который уже успешно отредактирован. Черный текст - текст, в который вложение не производится, потому, что все данные уже могут быть вложены в предшествующий ему текст.
6 - кнопка, позволяющая сохранить покрывающее сообщение в файл.
7 - кнопка, позволяющая загрузить покрывающее сообщение из файла.
8 - поле, показывающее сколько максимально можно вложить бит в текст введенного покрывающего сообщения.
9 - поле, показывающее длину секретного сообщения в битах.
10 - поле, показывающее сколько бит уже успешно вложено в покрывающее сообщение.
11 - поле, показывающее сколько бит еще не совпадает с битами интервалов.
12 - поле, показывающее текущую скорость вложения информации.
13 - поле, указывающее, успешно ли осуществлено вложение секретного сообщения в покрывающее.
14 - кнопка, по нажатию которой проверяется успешность вложения секретного сообщения в покрывающее и происходит обновление цветовой индикации в поле редактирования покрывающего сообщения.

Извлечение информации

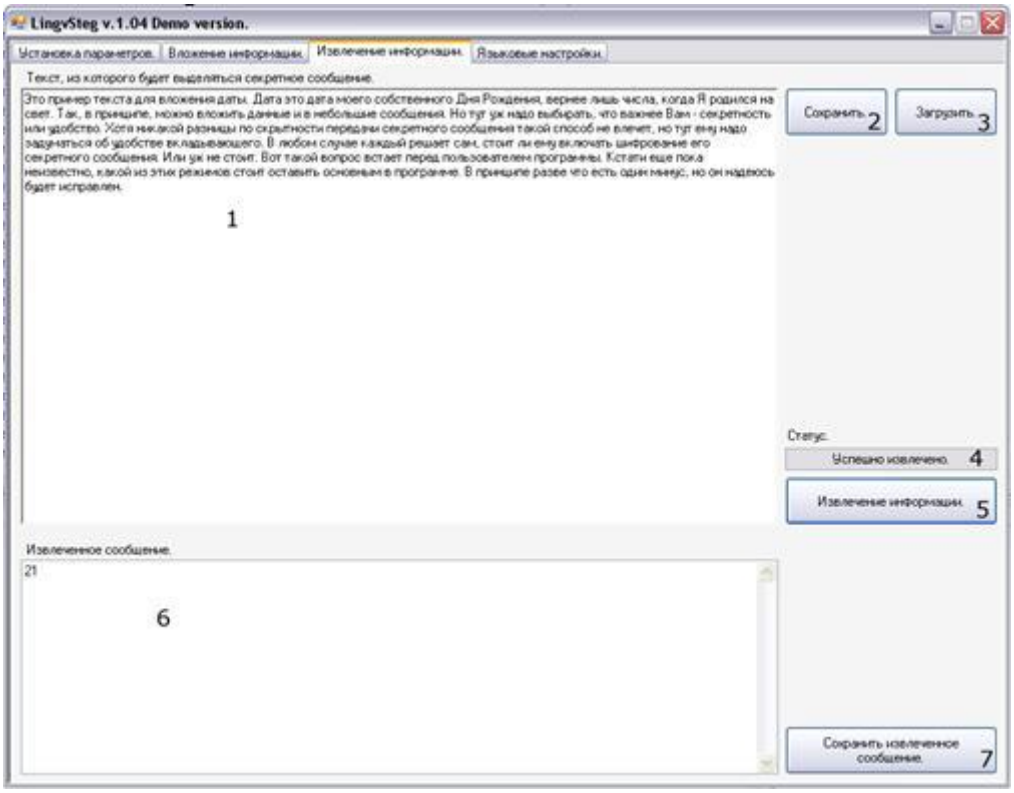


Рис. 12. Панель извлечения информации.

1 - поле для ввода сообщения, из которого будет проходить извлечение информации.
2 - кнопка, позволяющая сохранить стеготекст в файл.
3 - кнопка, позволяющая загрузить стеготекст из файла.
4 - поле, указывающее успешно ли осуществлена попытка извлечения информации из введенного сообщения.
5 - кнопка, по нажатию которой будет произведена попытка извлечь информацию из введенного сообщения.
6 - поле, в котором будет отображено секретное сообщение в случае успешного извлечения.
7 - кнопка, позволяющая сохранить секретное сообщение в файл.

Языковые настройки

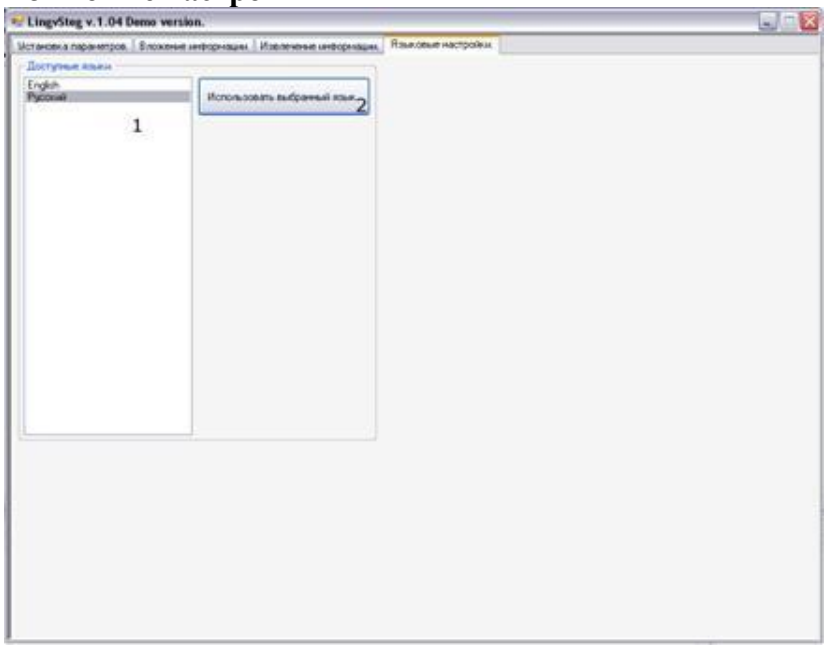


Рис. 13. Панель языковых настроек.

1 - список, в котором отображены те языки, на которых может быть представлен интерфейс программы.
2 - кнопка, по нажатию которой будет установлен выбранный язык интерфейса.

Порядок выполнения работы

Вложение информации

1. Задайте требуемые параметры во вкладке «Установка параметров».
2. Нажмите кнопку «Принять параметры» (5). После этого в случае корректно указанных параметров ввод в поля будет заблокирован, и указанные параметры будут применяться для дальнейшей работы. Поле «Статус» (3) отобразит информацию о том, что параметры успешно приняты. В случае же некорректно указанных параметров будет выведено сообщение том, какой из параметров нуждается в корректировке, а поле «Статус» (3) отобразит информацию о том, что параметры не приняты.
3. После того, как параметры будут успешно заданы, перейдите на вкладку «Вложение информации».
4. Введите в поле для ввода секретного сообщения (1) сообщение, которое хотите вложить, или загрузите его из файла, используя кнопку «Загрузить»(3). Поле «Введено бит.»(4) будет отображать его длину в битах
5. Начинайте вводить в поле для покрывающего сообщения (5) текст в который будет производиться вложение, или загрузите этот текст из файла, используя кнопку «Загрузить»(7).
6. Нажмите кнопку «Анализ и вложение информации» (14). Программа произведет попытку вложения в текст покрывающего сообщения. Поля справа (8-12) отобразят различную информацию о ходе вложения. Текст покрывающего сообщения станет окрашен в несколько цветов. Цвета указывают на совпадение или несовпадение битов хэшей интервалов с битами секретного сообщения. Красный текст - текст, нуждающийся в дальнейшем редактировании, зеленый текст - текст, который уже успешно отредактирован. Черный текст - текст, в который вложение не производится, потому, что все данные уже могут быть вложены в предшествующий ему текст. Длина черного текста ограничена, не стоит вводить текста более чем в такой, в который потенциально можно вложить 8 бит, иначе из этого участка тоже будут извлекаться данные.
7. Редактирование покрывающего сообщения и последующее нажатие кнопки «Анализ и вложение информации» (14) необходимо производить до тех пор, пока поле «Статус» (13) не отобразит информацию о том, что секретное сообщение было успешно вложено.
8. Готовое покрывающее сообщение можно или скопировать прямо из поля редактирования (5), или сохранить в файл, используя кнопку «Сохранить» (6).

Извлечение информации

9. Задайте требуемые параметры во вкладке «Установка параметров».
10. Нажмите кнопку «Принять параметры» (5). После этого в случае корректно указанных параметров ввод в поля будет заблокирован, и указанные параметры будут применяться для дальнейшей работы. Поле «Статус» (3) отобразит информацию о том, что параметры успешно приняты. В случае же некорректно указанных параметров будет выведено сообщение том, какой из параметров нуждается в корректировке, а поле «Статус» (3) отобразит информацию о том, что параметры не приняты.
11. После того, как параметры будут успешно заданы, перейдите на вкладку «Извлечение информации».
12. В поле для ввода стеготекста (1) введите текст сообщения, из которого планируется извлечение информации, или загрузите текст из файла, используя кнопку «Загрузить» (3).
13. Нажать кнопку «Извлечение информации» (5). Результат попытки извлечения будет указан в поле «Статус» (4). В случае успешного извлечения извлеченное сообщение появится в поле для извлеченного сообщения (6).
14. Извлеченное сообщение можно или скопировать прямо из этого поля извлеченного сообщения (6), или сохранить в файл, используя кнопку «Сохранить извлеченное сообщение» (7).

Примечание

В данной программе вкладываемое сообщение предварительно не шифруется (для упрощения) и поэтому обнаружение данной лингвистической СГ оказывается тривиальным при известном алгоритме вложения и извлечения. Однако, при использовании стойкого шифрования такая СГ будет необнаруживаемой.

Отчет

1.Титульный лист.
2.Текст вкладываемого сообщения.
3.Текст извлеченного сообщения.
4.Объём покрывающего сообщения.
5.Время вложения информации (затраченное оператором) и скорость вложения по отношению к объёму покрывающего сообщения.