Лабораторная работа № 0

(Предварительная. В которой нужно вспомнить, как пользоваться языками программирования)

В работах можно пользоваться любыми языками программирования, но скриптовые подойдут лучше. Также будет удобнее использовать системы UNIX из-за предоставляемых ими утилит.

Шестнадцатеричная система счисления (hex)

Шестнадцатеричная система счисления (шестнадцатеричные числа) — позиционная система счисления по целочисленному основанию 16.

Обычно в качестве шестнадцатеричных цифр используются десятичные цифры от 0 до 9 и латинские буквы от A до F для обозначения цифр от 10_{10} до 15_{10}, то есть (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

Перевод чисел из шестнадцатеричной системы в десятичную

Для перевода шестнадцатеричного числа в десятичное необходимо это число представить в виде суммы произведений степеней основания шестнадцатеричной системы счисления на соответствующие цифры в разрядах шестнадцатеричного числа.

Например, требуется перевести шестнадцатеричное число 5A3 в десятичное. В этом числе 3 цифры. В соответствии с вышеуказанным правилом представим его в виде суммы степеней с основанием 16:

$$5A3_{16} = 3 \cdot 16^0 + 10 \cdot 16^1 + 5 \cdot 16^2 =$$
$$= 3 \cdot 1 + 10 \cdot 16 + 5 \cdot 256 = 3 + 160 + 1280 = 1443_{10}$$

Кодировка Base64

Представление символов в вычислительных машинах

В вычислительных машинах символы не могут храниться иначе, как в виде последовательностей битов (как и числа). Для передачи символа и его корректного отображения ему должна соответствовать уникальная последовательность нулей и единиц. Для этого были разработаны таблицы кодировок.

Base64

Символ	Значение				Символ	Значение				Символ	Значение				Символ	Значение			
	10	2	8	16		10	2	8	16		10	2	8	16		10	2	8	16
A	0	000000	00	00	Q	16	010000	20	10	g	32	100000	40	20	w	48	110000	60	30
B	1	000001	01	01	R	17	010001	21	11	h	33	100001	41	21	x	49	110001	61	31
C	2	000010	02	02	S	18	010010	22	12	i	34	100010	42	22	y	50	110010	62	32
D	3	000011	03	03	T	19	010011	23	13	j	35	100011	43	23	z	51	110011	63	33
E	4	000100	04	04	U	20	010100	24	14	k	36	100100	44	24	0	52	110100	64	34
F	5	000101	05	05	V	21	010101	25	15	l	37	100101	45	25	1	53	110101	65	35
G	6	000110	06	06	W	22	010110	26	16	m	38	100110	46	26	2	54	110110	66	36

H	7	000111	07	07	X	23	010111	27	17	n	39	100111	47	27	3	55	110111	67	37
I	8	001000	10	08	Y	24	011000	30	18	o	40	101000	50	28	4	56	111000	70	38
J	9	001001	11	09	Z	25	011001	31	19	p	41	101001	51	29	5	57	111001	71	39
K	10	001010	12	0A	a	26	011010	32	1A	q	42	101010	52	2A	6	58	111010	72	3A
L	11	001011	13	0B	b	27	011011	33	1B	r	43	101011	53	2B	7	59	111011	73	3B
M	12	001100	14	0C	c	28	011100	34	1C	s	44	101100	54	2C	8	60	111100	74	3C
N	13	001101	15	0D	d	29	011101	35	1D	t	45	101101	55	2D	9	61	111101	75	3D
O	14	001110	16	0E	e	30	011110	36	1E	u	46	101110	56	2E	+	62	111110	76	3E
P	15	001111	17	0F	f	31	011111	37	1F	v	47	101111	57	2F	/	63	111111	77	3F

Операция XOR

Определение

Сложение по модулю 2 (логическое сложение, исключающее «ИЛИ», строгая дизъюнкция, XOR, поразрядное дополнение, побитовый комплемент) — булева функция, а также логическая и битовая операция. В случае 2 переменных результат выполнения операции является истинным тогда и только тогда, когда лишь один из аргументов является истинным. Для функции трёх и более переменных результат выполнения операции будет истинным только тогда, когда количество аргументов равных 1, составляющих текущий набор - нечетное. Такая операция естественным образом возникает в кольце вычетов по модулю 2, откуда и происходит название операции.

Таблица истинности

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Алгоритмы шифрования. AES

Определение

Симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES. Этот алгоритм хорошо проанализирован и сейчас широко используется, как это было с его предшественником DES. Национальный институт стандартов и технологий США (англ. National Institute of Standards and Technology, NIST) опубликовал спецификацию AES 26 ноября 2001 года после пятилетнего периода, в ходе которого были созданы и оценены 15 кандидатур. 26 мая 2002 года AES был объявлен стандартом шифрования. По состоянию на 2009 год AES является одним из самых распространённых алгоритмов симметричного шифрования. Поддержка AES (и только его) введена фирмой Intel в семейство процессоров x86 начиная с Intel Core i7-980X Extreme Edition, а затем на процессорах Sandy Bridge.

Алгоритм, режимы работы и дополнительная информация в Приложении 1

Задания лабораторной работы

В следующих заданиях постоянно будут использоваться строки в кодировании base64 или hex.

Задание 1. Напишите код, конвертирующий hex-представление массива байт в base64-представление, и код, проводящий обратное преобразование.
Строка:

```
faea8766efd8b295a633908a3c0828b22640e1e9122c3c9cfb7b59b7cf3c9d448bf04d72cde3aaa0
```

После перекодирования в base64 должна представлять собой:

```
+uqHZu/YspWmM5CKPAgosiZA4ekSLDyc+3tZt888nUSL8E1yzeOqoA==
```

Задание 2. Напишите функцию, которая принимает в качестве аргументов два буфера одинакового размера и возвращает их XOR. (Каждый из байтов нового буфера является результатом операции XOR соответствующих байтов буферов).
Строка:

```
8f29336f5e9af0919634f474d248addaf89f6e1f533752f52de2dae0ec3185f818c0892fdc873a69
```

после hex-декодирования и XOR со строкой:

```
bf7962a3c4e6313b134229e31c0219767ff59b88584a303010ab83650a3b1763e5b314c2f1e2f166
```

должна выдать строку:

```
305051cc9a7cc1aa8576dd97ce4ab4ac876af5970b7d62c53d495985e60a929bfd739ded2d65cb0f
```

Задание 3. XOR по одному символу.
Выберите строку в соответствии со своим порядковым номером в списке:

```
1. 2b4a0605040d4a1e03070f4a0b0d05464a03044a0b4a0d0b060b12134a0c0b18464a0c0b184a0b1d0b1344444444
2. 3904501903501150001502191f14501f1650131906191c500711025e50221512151c
3. 0506171513051e1f06055a560502041f1d1f1811561004191b5617561e1f12121318
4. 0e0d1f09404c040d1a094c1b03024c180409051e4c0a051e1f184c1a050f18031e15
5. 191f1911160b0c580c101d581d0e1114583f1914191b0c111b583d1508110a1d56
6. 213140f080146120e0346040712120a034a46340304030a4615160f0315460b070807010302
7. 021956050213171a560513150413025606 1a1718055602195602 1e1356331b061f04135105
8. 0d140c1115190c1d580f1d1908171654580c101d583c1d190c10582b0c190a54581916
9. 0e1d02001d0a0b4f1c1f0e0c0a4f1c1b0e1b0600014f18061b074f0a01001a0807
10. 130c14061143170c4307061017110c1a43020d43060d170a110643130f020d06174d
```

11. 2104030204141551130851051914513 41c011803145602510218 1f18020514035110161 41f05025d
12. 34160d0a070117174428010d0544160507011744 0c0b09014405060b051600440c0116
13. 1f180d1e1f04051c404c0f191f180308050d024c030a4c1804094c1f18030009024c1c000d 021f
14. 041811045013111e5003110615501815025000151f001c1550111e1450021503041f0215
15. 081c0b0b0a01034e1a014e1a060b4e090f020f1617404040

К строке была применена операция XOR по 1 символу (все операции только с латинским символами в кодировке ASCII). XOR по одному символу – это операция, в которой над каждым байтом массива производится операция XOR с одним постоянным значением. Например, если массив состоит из 3 элементов [a,b,c], то после преобразования по d будет [xor(a,d),xor(b,d),xor(c,d)]. Найдите ключ и расшифруйте сообщение.
Напишите код, делающий это за вас. Придумайте какой-нибудь способ оценки (алгоритм) вероятности того, что расшифрованный текст является текстом английского языка. (Частота встречаемости символов – хорошая оценка). Оцените каждый выход и выберите единственный с лучшей оценкой. Улучшайте свой алгоритм, пока не заработает.

В случае возникновения трудностей (лишь в этом случае) воспользуйтесь следующими подсказками. Задания в следующих лабораторных будут на порядок сложнее и без подсказок.

В начале записываем массив с частотами латинских символов в формате чисел с плавающей точкой. Значения можно взять в Приложении 2.
Далее перебираем всевозможные начальные строки (т.е. делаем XOR по одному символу для всех возможных паролей). Для каждой начальной строки считаем частоту встречаемости символов английского алфавита. Считаем, сколько раз встречается каждая буква в строке, потом делим на общее количество букв (не соответствует длине строки из-за наличия знаков препинания и т.д.). Считаем суммарную разницу между вероятностями (каждая из разностей берется по модулю). Это будет нашей оценкой. Теперь выбираем начальную строку с наименьшей оценкой. Эта строка с высокой долей вероятности будет правильной. Если p_i – табличное значений частот встречаемости буквы i, $p_i(a)$ – частота встречаемости символа i в расшифрованном тексте при ключе a, то необходимо найти байт a, при котором $\sum_i |p_i - p_i(a)|$ принимает минимальное значение.

Задание 4. Детектирование XOR по одному символу.

В архиве detectSingleCharacterXor 15 файлов. Возьмите файл с соответствующим вашему номером. В файле 60 строк, одна из них – результат шифрования XOR по одному символу. Определите ее и расшифруйте. Код из задания 3 должен вам в этом помочь.

Задание 5. XOR с повторяющимся ключом.

Эта операция проводит XOR каждого символа с соответствующим символом ключа. Реализуйте алгоритм. Строка (55 символов):

Never trouble about trouble until trouble troubles you!

С ключом (повторяющаяся строка «ICE», 55 символов):

ICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEICEI

должна превратиться в строку:

```
690d203f263769373726362725266528212a3c37653d312a3c21292c633027372c2563313b2c302b2f20693737263627252636693
```

Задание 6. Взлом XOR с повторяющимся ключом.

В файле *breakRepeatedKeyXor.txt* находится зашифрованная строка в base64. Попробуйте взломать её. Для этого:

1) Перебираем длину ключа **k** от 2 до 40;
2) Для каждой длины ключа создаем **k** строк. В строку **i** переносим все символы изначальной строки с номером n таким, что (**n mod k)=i**;
3) Расшифровываем каждую из полученных строк по алгоритму XOR по одному символу;
4) Соединяем расшифрованные строки;
5) Из 39 строк визуально выбираем нужный нам текст.

Задание 7. AES в режиме шифрования ECB.

Содержание файла *decryptAesEcb.txt* зашифровано AES-128-ECB и переведено в base64. Использован ключ:

```
YELLOW SUBMARINE
```

Расшифруйте.

Если выполняете задание на python, то используйте openssl. Воспользуйтесь любой удобной библиотекой, позволяющей работать с AES. Соответствующую документацию можно найти на сайте разработчика библиотеки.

Задание 8. Детектирование ECB.

В файле *detectEcb.txt* находятся строки в hex. Одна из них – результат шифрования в режиме ECB, размер блока – 16 байт. Определите эту строку.

Напоминаем, проблема с ECB в том, что он не использует состояния и вывод всегда детерминирован: одни и те же 16 байтов открытого текста при одном ключе всегда произведут те же 16 байтов зашифрованного текста.

Для улучшения знаний по теме предлагается прочитать статью в приложении 3.