Задания по практикуму на ЭВМ № 1. 319 группа. 2017 год.

Содержание

Список изменений

2017-09-15

- Аффинный шифр:
  - Добавлены комментарии по поводу обратимости матрицы $(X|1)$, исправлен пример взлома.
- Квадрат Полибия:
  - Исправлен пример шифрования, в качестве кодирования по таблице выбран порядок "номер строки"номер столбца".
- Шифр Плейфера:
  - Поддержка русского языка теперь необязательна.
- Добавлены комментарии по поводу поддержки русского языка.
- Добавлены сроки сдачи заданий.

Структура документа

Документ состоит из четырех частей. Первая часть посвящена описанию задания. Во второй части находится список литературы. Третья часть представляет собой описание требований к программной реализации. В заключительной части освещены некоторые организационные вопросы.

Описание задания

Задание состоит из пяти частей и представляет собой реализацию самых простых и известных методов шифрования. Целью данного задания является частичное ознакомления с историей криптографии и формирование понимания, почему данные методы шифрования не могут использоваться на практике. Необходимо реализовать следующие методы шифрования.

Аффинный шифр

Шифрование/Расшифрование

На вход подаются параметры шифра a и b, и строка открытого текста составленная из символов алфавита A: $|A| = m$. Символ открытого текста нумеруются числами $x_i \in [0, ..., m-1]$ согласно порядку в алфавите. Тогда каждый символ открытого текста x и заменяется символом y, полученным линейным преобразованием из оригинального: $y = ax+b \mod m$[1]. Например, зашифруем слово «DEDUCTION» используя параметры $a = 3, b = 1$. Получаем: KNKJHGZRO.

Вскрытие*

Процесс шифрования можно представить следующим образом

$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_k \end{pmatrix} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \\ x_k & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

$$Y = \begin{pmatrix} X \mid 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} X \mid 1 \end{pmatrix}^{-1} Y$$

Таким образом, зная результат шифрования всего двух символов, можно восстановить параметры оригинального шифра. Однако матрица $(X|1)$ должна быть обратима в кольце Z_m, так что символы открытого текста не могут быть любыми.

На вход подаётся два символа открытого текста и шифротекста, а также строка шифротекста. Программа должна найти параметры оригинального шифра, используя которые расшифровать остальной шифротекст.

Входные параметры

На вход программа принимает имя файла со следующим содержанием. В первой строке указывается действие, которое необходимо сделать: encrypt/decrypt/break. На следующей строке указывается ключ в виде двух чисел - множитель и сдвиг (для режима break подаётся пара открытый текст и шифротекст через пробел). На третьей строке указывается открытый текст или шифротекст.

Вывод работы программы

Результатом работы программы должен являться полученный шифртекст или открытый текст. Ошибки необходимо выводить в stderr.

[1] Каким ограничениям должна удовлетворять a для возможности однозначного расшифрования?

Примеры

Пример ввода	Пример вывода
encrypt 3 1 DEDUCTION	KNKJHGZRO
decrypt 5 2 PUT OU OWMJWT	NOT SO SECRET
decrypt 2 5 BDFHJL	Error: 2 is no coprime with 26
break EH NY HY ZHY JATKQP TV CAPJNU Q_Q	AH MAH KRIPTO IS BROKEN T_T

Шифр маршрутной перестановки

Шифрование/Расшифрование

Шифры маршрутных перестановок используют некоторую геометрическую фигуру (плоскую или объемную). Преобразования состоят в том, что отрезок открытого текста записывается в такую фигуру по некоторой траектории, а выписывается по другой траектории.

В данном задании роль геометрической фигуры будет играть таблица. Пусть число k определяет последовательность $1, 2, \ldots, k-1, k$ чисел от 1 до k. Зададим произвольную перестановку этих чисел $a_1, a_2, \ldots, a_k$. Число k и зафиксированная перестановка $a_1, a_2, \ldots, a_k$ являются секретным ключом системы. Для шифрования составляется таблица с k столбцами. В эту таблицу последовательно записывается открытый текст с учётом пробелов. Согласно зафиксированной перестановке считывается текст по столбцам в указанном порядке. В случае если количество символов открытого текста не кратно числу k, сообщение следует дополнить пробелами.

Например, зашифруем сообщение «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ». Фиксируем число $k = 6$ и перестановку $(2, 6, 3, 4, 5, 1)$. Получаем следующую таблицу:

2	6	3	4	5	1
А	Б	Р	А	М	О
В	_	И	Л	Ь	Я
_	С	Е	Р	Г	Е
Е	В	И	Ч	_	_

При шифровании сообщения «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» результат будет «ОЯЕ_АВ_ЕРИЕИАЛРЧМ
(Символом «_» обозначен пробел).

Входные параметры

На вход программа принимает имя файла со следующим содержанием. В первой строке указывается действие, которое необходимо сделать: encrypt/decrypt. На следующей строке указывается ключ в виде заданной перестановки. Элементы перестановки отделены запятой. На третьей строке указывается открытый текст или шифротекст.

Вывод работы программы

Результатом работы программы должен являться полученный шифртекст или открытый текст.

Примеры

Пример ввода	Пример вывода
encrypt 2,6,3,4,5,1 АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ	ОЯЕ АВ ЕРИЕИАЛРЧМЬГ Б СВ
decrypt 3,2,5,1,4 нкосечстпно е аьрел еенои	очень секретное послание

Квадрат Полибия

Шифрование/Расшифрование

Для шифрования с помощью квадрата Полибия необходимо составить таблицу шифрования. В зависимости от количества букв в исходном алфавите (в рамках задания рассматриваются русский и английский языки) квадрат может быть размером 5×5 или 6×6. В полученную таблицу вписываются все буквы алфавита по порядку. При нехватке клеток можно вписать в одну клетку две буквы. Возможные таблицы:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

	1	2	3	4	5	6
1	А	Б	В	Г	Д	Е
2	Ё	Ж	З	И	Й	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ъ	Ы	Ь
6	Э	Ю	Я	_	.	,

Существует несколько методов шифрования с помощью квадрата Полибия. В данном задании предлагается реализовать следующий вариант. Сообщение преобразуется в координаты по квадрату Полибия. Например, шифруем слово «SOMETEXT»:
43 34 32 15 44 15 53 44
Затем координаты циклически сдвигаются влево на нечётное число символов. Для сдвига в один символ имеем:
33 43 21 54 41 55 34 44
Что соответствует шифротексту «NSFYQZOT».

Входные параметры

На вход программа принимает имя файла со следующим содержанием. В первой строке указывается действие, которое необходимо сделать: encrypt/decrypt. На следующей строке указывается язык english/russian и,через пробел, нечётное число. На третьей строке указывается открытый текст или шифротекст.

Вывод работы программы

Результатом работы программы должен являться полученный шифртекст или открытый текст. В английском варианте шифрования пробелы и знаки препинания нужно опустить. В обоих вариантах при наличии недопустимых символов ожидается сообщение об ошибке в stderr.

Примеры

Пример ввода	Пример вывода

`encrypt`
`english 1`
`SOMETEXT`
	`NSFYQZOT`

`decrypt`
`russian 1`
`TMPDUCYS`
	`Error: полученный текст содержит недопустимые символы.`

Шифр Виженера

Шифрование/Расшифрование

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая таблица Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
C	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
D	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
E	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
F	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
G	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
H	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
I	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
J	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
K	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
L	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
M	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M	N
N	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K	L
P	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J	K
Q	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I	J
R	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H	I
S	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G	H
T	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F	G
U	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E	F
V	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D	E
W	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C	D
X	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B	C
Y	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	B
Z	z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

Например, зашифруем следующее сообщение: ATTACKATDAWN. В качестве ключа выберем слово LEMON. Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: LEMONLEMONLE. Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа, то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст:	`ATTACKATDAWN`
Ключ:	`LEMONLEMONLE`
Зашифрованный текст:	`LXFOPVEFRNHR`

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным обра-

зом. Программа должна уметь шифровать и расшифровывать получаемые сообщения с заданным ключом.

Входные параметры

На вход программа принимает имя файла со следующим содержанием. В первой строке указывается действие, которое необходимо сделать: encrypt/decrypt. На следующей строке через пробел указывается длина ключевого слова и само ключевое слово. На третьей строке указывается длина текста и сам текст.

Вывод работы программы

Результатом работы программы должен являться полученный шифртекст или открытый текст. Пробелы и знаки препинания должны попадать в вывод как есть. Ошибки должны выводится в stderr.

Примеры

Пример ввода	Пример вывода
encrypt 5 Lemon 14 Attack at dawn	Lxfopv ef rnhr
decrypt 8 TOPSECRET 11 Zixfic kiqh	Guinea text

Шифр Плейфера

Шифрование/Расшифрование

Шифр Плейфера использует матрицу 5x5 (в данном задании шифр применяет для английского алфавита), содержащую ключевое слово или фразу. Для создания матрицы и использования шифра достаточно запомнить ключевое слово и четыре простых правила. Чтобы составить ключевую матрицу, в первую очередь нужно заполнить пустые ячейки матрицы буквами ключевого слова (не записывая повторяющиеся символы), потом заполнить оставшиеся ячейки матрицы символами алфавита, не встречающимися в ключевом слове, по порядку (символы «I» и «J» объединяются в одну ячейку). Ключевое слово должно быть записано в верхней строке матрицы слева направо. Ключевое слово, дополненное алфавитом, составляет матрицу 5x5 и является ключом шифра.

Для того чтобы зашифровать сообщение, необходимо разбить его на биграммы (группы из двух символов), например «Hello World» становится «HE LL OW OR LD», и отыскать эти биграммы в таблице. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Определяем положения углов этого прямоугольника относительно друг друга. Затем, руководствуясь следующими 4 правилами, зашифровываем пары символов исходного текста:

- Если два символа биграммы совпадают (или если остался один символ), добавляем после первого символа «X», зашифровываем новую пару символов и продолжаем.
- Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.
- Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящиеся непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.
- Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Например, выберем ключевое слово «DEVELOPER», тогда получаем матрицу:

D	E	V	L	O
P	R	A	B	C
F	G	H	I/J	K
M	N	Q	S	T
U	W	X	Y	Z

Зашифруем слово «PROPERTY». Делим на биграммы: PR OP ER TY. P и R в одной строке, потому вместо каждой из этих букв берём по сути правого соседа (правило 2). Получаем RA. C OP так не пройдёт - тут и строки, и столбцы разные. Берём правило 4. Получаем DC. E и R в одном столбце. Берём соседей снизу (правило 3). Получаем RG. TY заменяется на SZ - правило 4. Итог - RADCRGSZ.

Входные параметры

На вход программа принимает имя файла со следующим содержимым:

- В первой строке указывается действие, которое необходимо сделать: encrypt/decrypt.
- На второй строке через пробел указывается ключевое слово.

- На третьей строке указывается текст или шифртекст (набор символов английского языка).
- Поддержка русского языка в данном задании необязательна, но приветствуется.

Вывод работы программы

Результатом работы программы должен являться полученный шифртекст или открытый текст. Пробелы и знаки препинания должны попадать в вывод как есть. Ошибки должны выводится в stderr. *Примечание:* Если поданна строка нечётной длины, то необходимо дополнить её символом X.

Примеры

Пример ввода	Пример вывода
encrypt DEVELOPER PROPERTY	RADCRGSZ
decrypt QWERTY Nbtr fx la mdwwrmtc	Here is my password

Список литературы

1. Аграновский А.В., Хади Р.А. «Практическая криптография»
2. Яковлев А.В., Безбогов А. А., Родин В.В., Шамкин В.Н. «Криптографическая защита информации»
3. Новиков Е.А., Шитов Ю.А. «Криптографические методы защиты информации»
4. Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черёмушкин А. В. «Основы криптографии»
5. Бабаш А. В., Шанкин Г. П. «Криптография»
6. Дориченко С.А., Ященко В.В. «25 этюдов о шифрах: Популярно о современной криптографии»
7. Реализация некоторых описанных шифров для проверки: http://rumkin.com/tools/cipher/

Общие требования к реализации и сдачи программ

Требования к функциональности

1. Корректная обработка всех видов входных данных и возможных ошибок.
2. Поддержка двух алфавитов (английский и русский), если не оговорено обратное. В ином случае программа должна быть легко модифицируемой для использования русского алфавита. По умолчанию, русский текст предполагается подавать в Юникоде.
3. Режим русского языка должен подключаться параметром командной строки –lang=russian, если иное не оговорено в условии задачи.
4. Программа должна принимать на вход параметр - путь к входному файлу.
5. Вывод программы необходимо проводить в stdout.
6. Вывод ошибок – в stderr.

Требования к коду

1. Программа должна быть написана на языке C/C++.
2. Код не должен быть скопирован у другого студента.
3. Код должен быть написан по одному из стилей. Главное требование: читаемость.
4. Каждая функция в коде не должна быть длиннее 25 строк (и только в исключительных случаях больше).
5. Функции и переменные должны иметь осмысленные имена (length, array и т.д.).
6. Реализация не должна быть платформозависимой. Предусмотреть возможность сборки на UNIX-подобных системах компилятором gcc версии 4.9.2.
7. Допускается использование возможностей из последних стандартов c++ в мере поддерживаемой компилятором gcc версии 4.9.2.
8. По возможности рекомендуется добиться отсутствия предупреждений со стороны компилятора с опцией компиляции -Wall (/Wall для компилятора Visual Studio).

Формат приёма заданий

1. Студенты присылают письмо на почту cmcmsu.aspa2016.ib@gmail.com следующего формата:
   (a) Тема письма в формате «Задание_[номер_часть]_Ф_И_О».
   (b) В теле письма находится архив с проектом, без лишних сборочных файлов, примеры командной строки, на которой проверялась работоспособность программы, а также Makefile для сборки программы.
   (c) В Makefile обязательно использование опции -Wall для включения предупреждений компилятора и рекомендуется использование 2 уровня оптимизаций компилятора (флаг -O2).
   (d) При наличии поддержки русского алфавита, следует указать кодировку принимаемых файлов[2]. По умолчанию, кодировка считается utf-8.
2. В случае нахождения ошибки аспирант отправляет Вам комментарий и свои аргументы командной строки. Процесс повторяется до тех пор, пока не будут выполнены все требования или не закончится срок приема задания.
3. После успешного прохождения первого этапа назначается встреча для обсуждения деталей реализации задания.
4. Задание можно присылать по частям.

Сроки приёма заданий

1. Реализацию каждого из выданных заданий рекомендуется выслать на почту не позднее 21:00 2 октября, и необходимо выслать не позднее 21:00 9 октября.
2. Этап исправления ошибок и обсуждения деталей реализации должен закончиться не позднее 21:00 16 октября.

Организационные вопросы

В случае возникновения вопросов, которые требуют личной встречи для разъяснений и которые возникли у подавляющего большинства, просьба заранее написать не позднее 22:00 8 сентября 2016 года.
Ответ на небольшие вопросы можно получить в Telegram-канале:
https://t.me/joinchat/AAAAAEOJIk8ITkd1yYWw7A
или по почте: cmcmsu.aspa2016.ib@gmail.com

[2] и иные полумёртвые стандарты не приветствуются