МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский
государственный университет аэрокосмического приборостроения

И. Л. Ерош

ДИСКРЕТНАЯ МАТЕМАТИКА.

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ

Учебное пособие

Санкт-Петербург
2001

УДК 512.54
Е78
ББК 22.1

Ерош И. Л.

Е78 Дискретная математика. Математические вопросы криптографии: Учеб. пособие/СПбГУАП. СПб., 2001. 56 с.

В учебном пособии кратко изложены основные положения криптографии, которая по используемому математическому аппарату может рассматриваться как раздел дискретной математики. Первый раздел пособия практически не требует специальной математической подготовки и доступен школьникам старших классов. Второй и последующий разделы используют некоторые понятия и теоремы теории чисел. Для знакомства с ними полезным может оказаться учебное пособие "Дискретная математика. Теория чисел" того же автора или любая литература по теории чисел.

Пособие ориентировано на студентов технических университетов, аспирантов и преподавателей дисциплины "Дискретная математика" технических вузов.

Рецензенты:
кафедра радиосистем Санкт-Петербургского
электротехнического университета;
канд. техн. наук доцент *В. Н. Сасковец*

Утверждено
редакционно-издательским советом университета
в качестве учебного пособия

ВВЕДЕНИЕ

Криптография – наука о тайнописи существует уже не одну тысячу лет. Однако до последнего времени она обслуживала правителей, военных, дипломатов. И только с середины 70-х годов начала оформляться в строгую математическую теорию (в связи с разработкой принципов открытого распределения ключей). Область применения современных криптографических систем не ограничивается как ранее межправительственными переговорами, а служит для защиты информационных потоков в банковских сетях, для организации выборов с использованием компьютерных сетей и других случаях.

Часто под термином *криптография* понимают действия легальных отправителей и получателей сообщений. Под термином *криптоанализ* понимают действия врага (незаконного перехватчика сообщений). Общая схема криптосистемы выглядит следующим образом (рис. 1).

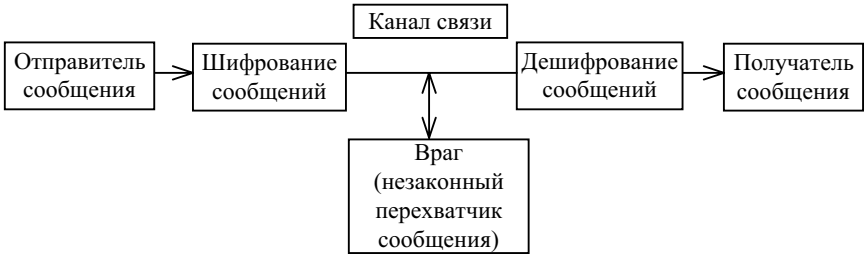


Рис. 1. Общая схема криптосистемы

Введем некоторые символьные определения: pt – исходный текст; $E(pt) = ct$ – зашифрованный текст (криптотекст); $D(ct) = pt$ – расшифрованный текст.

Зашифрование и расшифрование текстов производятся в рамках *криптосистемы*.

Криптосистема состоит из следующих компонентов:

1. Пространство исходных сообщений PT, которое содержит всевозможные исходные тексты pt.

2. Ключевое пространство K. Каждому ключу k в K соответствует алгоритм зашифрования E_k и расшифрования D_k. Если к сообщению pt применить E_k, а к результату D_k, то снова получим исходный текст pt, т. е. $D_k(E_k(pt)) = pt$.

3. Пространство криптотекстов CT, т. е. набор всевозможных криптотекстов ct.

Элементами CT являются результаты применения к элементам PT методов шифрования E_k, где k пробегает все пространство K.

При передаче текстовых сообщений по различным каналам связи каждая буква предварительно кодируется комбинацией из двоичных символов {0, 1}, а затем уже все сообщение подвергается зашифрованию.

Мощность пространства ключей не должна быть очень маленькой, так как перехватчик не должен иметь возможности проверить все ключи. Часто пространство ключей K бесконечно. Как определить, является ли данная криптосистема хорошей? Сэр Френсис Бекон сформулировал требования к криптосистемам:

1. По заданным E_k и исходному сообщению pt легко вычислить ct. По заданным D_k и ct легко вычислить исходное сообщение pt.

2. Не зная D_k, , нельзя вычислить pt из криптотекста ct.

3. Криптотекст не должен вызывать подозрений, т. е. должен выглядеть естественно.

Нам представляется третье требование не очень важным, в первом требовании подразумевается, что для легальных пользователей криптосистема не должна быть очень сложной, а во втором – “невозможность” следует заменить на трудновычислимость.

1. ЭЛЕМЕНТЫ КЛАССИЧЕСКОЙ КРИПТОГРАФИИ

1.1. Древние криптографические системы

Познакомимся с некоторыми старыми криптосистемами (даются в современной интерпретации) [2].

Система Цезаря

Пусть требуется передать текстовое сообщение, написанное на английском языке (содержит 26 букв, пробелы игнорируются): WE GO TO CITY. Все буквы от A до Z нумеруются цифрами от 0 до 25 и в сообщении производится замена, например со сдвигом на три:

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

В этом случае исходный текст будет зашифрован как ZHJRWRFLW*B*. Ключевое пространство системы Цезаря состоит из 26 чисел от 0 до 25 и определяет сдвиг всех букв алфавита при заменах. E_k и D_k легко вычисляются одно из другого, так как $D_k = E_{26-k}$. Множество замен $\{E_k\}$образует коммутативную группу сдвигов, причем $E_k^{-1} = D_k$

Обобщением шифра Цезаря может служить шифр перестановок. Поскольку число перестановок 26 букв равно 26!, то пространство ключей такого шифра равно 26!, т. е. достаточно велико. Однако полное множество перестановок n элементов образует группу, т. е. по E_k просто и однозначно находится D_k

Криптосистема Хилла

Криптосистема основана на свойствах линейной алгебры. Все буквы английского алфавита кодируются цифрами, как и в предыдущем случае: A–0, B–1, ..., Z –26. Матрица для шифрования M имеет размер $d \times d$. Все операции выполняются по модулю 26. Все сообщение разбивается на блоки длины d. Для работы криптосистемы необходимо, чтобы матрица M имела бы обратную.

Например, $M = \begin{pmatrix} 3...3 \\ 2...5 \end{pmatrix}$ $M^{-1} = \begin{pmatrix} 15...17 \\ 20.....9 \end{pmatrix}$.

Исходное сообщение HEL*P* определяется двумя векторами

$$P_1 = \begin{pmatrix} H \\ E \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix}; \quad P_2 = \begin{pmatrix} L \\ P \end{pmatrix} = \begin{pmatrix} 11 \\ 15 \end{pmatrix}.$$

Из уравнений $MP_1 = C_1 \begin{pmatrix} 7 \\ 8 \end{pmatrix}; \quad MP_2 = C_2 \begin{pmatrix} 0 \\ 19 \end{pmatrix}$, что соответствует шифрованному сообщению HIAT.

Расшифровать сообщение можно умножением обратной матрицы M^{-1} на C_1 и C_2.

Криптосистема Ришелье

Эта система относится к группе методов называемых “информация среди мусора”. Для шифрования сообщений используется лист картона с прорезями, например:

X	*X*	*X*	*X*	*X*	*X*	*X*		*X*	*X*
X	*X*	*X*	*X*	*X*	*X*	*X*	*X*		*X*
X	*X*	*X*	*X*	*X*		*X*	*X*	*X*	*X*
X	*X*	*X*	*X*			*X*	*X*	*X*	*X*
	X	*X*	*X*	*X*			*X*		*X*
X		*X*	*X*	*X*	*X*	*X*	*X*	*X*	
X	*X*	*X*	*X*	*X*	*X*	*X*	*X*		

Картон с прорезями накладывается на лист бумаги и в прорези вписывается сообщение. Затем картон снимается и в остальные клеточки вписывается “мусор”, который полностью затеняет ранее написанный текст. Так для примера можно написать сообщение:

I		L	O	V	E		Y	O	U
I		H	A	V	E		Y	O	U
D	E	E	P		U	N	D	E	R
M	Y		S	K	I	N		M	Y
L	O	V	E		L	A	S	T	S
F	O	R	E	V	E	R		I	N
H	Y	P	E	R	S	P	A	C	E

Оно выглядит совершенно невинно. Но если наложить картон с прорезями, получим зловещее сообщение YOU KILL AT ONCE.

Известны различные разновидности криптосистемы Ришелье. В частности, картон с прорезями можно взять в виде квадратной матрицы, а прорези в картоне можно сделать так, чтобы при последовательных поворотах на 90, 180 и 270 градусов буквами заполнялись все клеточки.

Одноалфавитными системами называются системы подстановок, в которых буквы заменяются одинаково на протяжении всего текста. Существуют многоалфавитные подстановки. Простейшим примером многоалфавитной подстановки может служить следующая. Исходное сообщение поделено на блоки по три буквы в каждом. При шифровании первая буква каждого блока становится третьей, а вторая и третья становятся соответственно первой и второй. Так, сообщение LETUSGOTOFRANCE станет ETLSGUTOORAFCEN. Система легко обобщается. Пусть длина блока равна d, тогда число вариантов блоковой замены равно $d!$

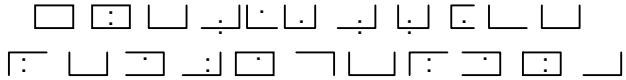
В некоторых случаях в одноалфавитных системах криптотекст использует алфавит, отличный от алфавита исходного текста. Например, шифр замены букв может иметь такой вид:

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	

Тогда исходное сообщение WE TALK ABOUT IT MANY TIMES будет выглядеть так:



Если исходное сообщение достаточно длинно и написано на любом естественном языке, то для его расшифровки можно использовать статистические данные о вероятности появления букв в произвольных сообщениях. Если же алфавиты исходного сообщения и криптотекста совпадают и нет никакой дополнительной информации, то расшифровка существенно усложняется (26! вариантов перебора подстановок).

Считается, что одной из самых древних криптосистем является *система Полибия.* Рассмотрим квадрат замен (доска Полибия):

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

Каждая буква α представляется в криптотексте парой букв, соответствующих ее координатам. Так, например, сообщение HALLOW будет представлено как BCAACACACDEB. В нашей классификации система Полибия есть одноалфавитная система замен. Буква *J* исключена, для того чтобы получить квадрат. Алфавит шифротекста уже алфавита исходного текста.

Аффинная криптосистема

Эта система определяется двумя целыми числами a и b, причем $0 \le a, b \le 25$, $(a, 26) = 1$. Буква α заменяется на $\alpha a + b \bmod 26$. Для $a = 3$ и $b = 5$ получаем следующие замены букв:

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

Условие взаимной простоты a и 26 обеспечивает взаимную однозначность отображения.

Подсчитаем, сколько разных ключей имеется в приведенной аффинной системе. Число a может принимать следующие значения: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25. Независимо от значения a число b принимает 26 значений (от 0 до 25). Случай $a = 1$; $b = 0$ следует исключить. Тогда общее число ключей равно 311. В таком простейшем случае криптоаналитику не потребуется много времени для подбора ключа.

Система Цезаря с ключевым словом

Она является одноалфавитной системой замены и определяется некоторым числом a и ключевым словом (фразой) с возможным повторением букв. Пусть система задана числом $a = 8$ и фразой: HOW MANY ELKS. Как и ранее выписываются подряд все буквы латинского алфавита и под буквой, стоящей на 8 месте, без пробелов пишется ключевая фраза. Оставшиеся буквы первой строки в алфавитном порядке дописываются далее с циклическим переносом:

0 8 25
A B C D E F G H I J K L M N O P Q R S T U V WX Y Z
P Q R T U V X Z **H 0 W M A N Y E L K S** B C D F G I J

Простейшая защита против атак криптоаналитика, основанных на подсчете частот появления различных букв в текстах, используется в криптосистеме *омофонов* (HOMOPHONES), которая также является одноалфавитной. Буквы исходного сообщения имеют несколько замен, причем число замен каждой буквы пропорционально вероятности ее появления в тексте сообщения.

1.2. Многоалфавитные системы

Система Плайфейра (*PLAYFAIR*)

Система названа в честь барона Плайфейра. Буквы английского алфавита, среди которых отсутствует *J*, упорядочиваются в виде квадрата 5×5, например

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

Квадрат используется для зашифрования и расшифрования по следующим правилам.

1. Исходный текст делится на блоки по 2 буквы в каждом. Текст имеет четную длину и в нем не должно быть блоков, содержащих 2 одинаковые буквы. Если эти требования не выполнены, текст модифицируется (даже в ущерб правилам грамматики). Например, AL LM EN; KI **SS** ME; WH ER EA RE YO **U.** Первый текст допустим, второй содержит в блоке 2 одинаковые буквы, третий имеет нечетную длину.

2. Если пара букв блока не попадает в одну строку или столбец, то эта пара шифруется парой букв прямоугольника. Если же пара букв попадает в одну строку или столбец, то она шифруется буквами со сдвигом вправо (в строке) или вниз (в столбце). Так, AL шифруется FP; LM–PV; EN–TO.

Зашифруем криптотекст CR YP TO EN IG MA–HI DI NG TO UN DO. EN IG MA – шифровальная машина, на которой была основана криптосистема, используемая немцами во время ВОВ.

При циклическом смещении строк и столбцов квадрата Плайфейра получаем эквивалентный квадрат. Известны модификации квад-

рата Плайфейра: прямоугольники размером 4×4; 3×9 и с иным способом замены.

Система Плайфейра с ключевым словом

Она строится следующим образом: выбирается ключевое слово без повтора букв и записывается в первых строках квадрата, а затем выписываются оставшиеся буквы в алфавитном порядке. Так, для ключевой фразы: HOW MANY ELKS получаем квадрат для шифрования:

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

Многоалфавитная система Виженера (1523–1596).

Квадрат Виженера выглядит следующим образом:

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

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

Система Виженера подобна системе Цезаря, в которой ключ меняется от шага к шагу. Квадрат Виженера используется как для зашифрования, так и для расшифрования. Для зашифрования читаем исходное сообщение из строк и ключей системы Цезаря из столбцов. Так, для исходного сообщения PURPLE и ключевого слова CRYPTO находим пересечение *P*-строки и *C*-столбца, получаем *R*. Криптотекст имеет вид: RLPEES. Для расшифрования находим, в какой строке в *C*- столбце лежит *R*? Получаем *P*.

Существует много других подобных квадратов; одним из наиболее известных является квадрат Бьюфорта, строками которого являются строки квадрата Виженера, записанные в обратном порядке.

Если сообщение длиннее ключевого слова, то последнее периодически повторяется. Если известен период повторения, то криптоанализ сводится к криптоанализу одноалфавитных систем. В 1860 г. немецким криптоаналитиком Ф.У. Казизки был изобретен метод для вскрытия периодических криптосистем с неизвестным периодом. Метод Казизки выявляет период с помощью обнаружения одинаковых слов в криптотексте.

Криптосистема AUTOKLAVE (приписываемая математику 16 века Дж. Кардано, известному своими формулами для решения уравнений 3 и 4-й степени)

Система является дальнейшей модификацией системы Виженера. Исходное сообщение с некоторым сдвигом само является и ключом шифровки. В следующем сообщении ключ равен 6:

Сообщение: A I D S I S T R A N S M I T T E D T H R O U G H
Ключ: A I D S I S TRA N SMI T T E D T

Ключ используется, как в системе Виженера для подстановки Цезаря для каждой буквы. Пустые символы в начале ключа могут заполняться циклическим концом сообщения или ключевым словом. Так для ключевого слова IMMUNE получаем следующее начало для криптотекста:

Сообщение: A I D S I S T R A N SM I T T E D T H R O U G H
Ключ: I M M U N E A I D S I S T R A N SM I T T E D T
Криптотекст: I U P M V W T Z D F A E B K T R V F P K H Y J A

В другом варианте модификации системы *AUTOKLAVE* в качестве ключа используется криптотекст, записанный после ключевого слова.

В данном случае предыдущий пример будет зашифрован следующим образом:

Сообщение: A I D S I S T R A N S M I T T E D T H R O U G H
Ключ: I M M U N E I U P M V W B L P Z N I J E I D O B
Криптотекст: I U P M V W B L P Z N I J E I D Q B Q V W X W I

Для криптоанализа последней версии криптоаналитику достаточно только найти длину ключа.

Для повышения секретности зашифрования можно исходное сообщение предварительно перевести на некоторый малораспространенный язык, а затем уже использовать одну из криптосистем.

Мы классифицировали криптосистемы как одноалфавитные и многоалфавитные. Другая классификация делит криптосистемы на контекстно–свободные и контекстно–зависимые. В первых шифруются индивидуальные буквы, во вторых – группы букв. Примеры криптосистем различных типов даны в следующей таблице:

Системы	Контекстно–свободные	Контекстно–зависимые
Одноалфавитные	Цезаря	Плайфейра
Многоалфавитные	Виженера	Плайфейра с периодом

Здесь система Плайфейра с периодом означает систему Плайфейра, где вместо одного используется несколько квадратов, например, три. Первая пара букв сообщения шифруется первым квадратом, вторая – вторым, третья – третьим, затем четвертая – опять первым и т. д.

Система "кодовая книга" (*CODE BOOK*).

Зашифрование осуществляется в соответствии с кодовой книгой, например,

Оригинал	Перевод
ATTACK	FISHING
.....	.....
IN	BETWEEN
.....	.....
MORNING	WORK HOUR
.....	.....
THE	THE

В этом случае сообщение ATTACK IN THE MORNING (атака утром) будет зашифрована как FISHING BETWEEN THE WORK HOURSE (рыбалка в рабочем перерыве). Чтобы сделать криптотекст синтаксически правильным, в него добавляются разумные концовки.

Одноразовый блокнот (*ONE – TIME PAD*)

*С*читается криптосистемой с идеальной секретностью [10]. Сообщением является последовательность битов ограниченной длины. В виде ключа выступает также двоичная последовательность. Ключ для шифрования и расшифрования общий. Например:

Сообщение *S*:	1	1	0	1	0	0	0	1	0	1	1	1	0	0	1	0	0	1
Ключ: *K*:	1	0	0	0	1	0	1	1	0	1	1	1	1	0	0	1	0	0
Шифр: $S \oplus K$:	0	1	0	1	1	0	1	0	0	0	0	0	1	0	1	1	0	1

Для расшифровки сообщения достаточно к $S \oplus K$ прибавить по модулю 2 еще раз K. После приема сообщения ключ уничтожается.

Очевидным недостатком такой криптосистемы является то, что ключ должен быть достаточно большим и отдельно передаваться получателю сообщения.

1.3. Роторные криптографические машины

Рассмотренные криптосистемы могут быть сделаны более быстродействующими и секретными при использовании специализированных машин. История создания таких машин насчитывает уже несколько сотен лет. Основная идея использовалась уже в старейшей машине Томаса Джеферсона – колесе Джеферсона.

Колесо Джеферсона состоит из одинаковых дисков, надетых на одну ось и имеющих возможность свободно вращаться друг относительно друга. По образующей диска написаны на равном расстоянии буквы английского алфавита, причем порядок букв на каждом диске различен. Если используется 10 дисков, то исходное сообщение разбивается на блоки длиной 10, которые отдельно кодируются с определенным сдвигом. Вращением дисков на одну линию устанавливается шифруемый блок и со сдвигом, одинаковым для всех дисков, снимается зашифрованный блок. Колесо Джеферсона реализует многоалфавитную подстановку.

До начала 50-х годов в армии США использовалась машина С-36 известного разработчика криптографических машин Бориса Хэйглина.

Некоторые известные криптографические машины, такие как немецкая ENIGMA, американская SIGABA, японские RED и PURPLE вре-

мен второй мировой войны, являются электромеханическими. Основным блоком в них является диск в виде кодового колеса с проволочными перемычками внутри, называемый ротором.

1.4. Криптографический стандарт DES

В 1977 г. был опубликован стандарт шифрования данных (DES–Data Encryption Standard) Национального бюро стандартов. DES–алгоритм был разработан специально для электронных устройств для зашифрования и расшифрования данных. До опубликования DES не было открытых публикаций, содержащих полный алгоритм для практического криптографического применения. Хотя в криптографии предполагается, что криптоаналитик знает используемую криптосистему, однако, большинство разработчиков криптосистем стараются скрыть детали их алгоритмов. DES является исключением. Существует два противоположных мнения о целесообразности опубликования общей системы шифрования данных. С одной стороны, этот шаг рассматривается как вызов всем тем, кто пытается вскрывать системы. С другой стороны, разработчики лучше всех знают слабости данной системы, что может позволять государственным организациям контролировать переговоры между банками, предприятиями и частными лицами.

Рассмотрим работу алгоритмов криптосистемы DES.

Пользователи выбирают ключ, содержащий 56 битов. Один и тот же ключ используется при зашифровании и расшифровании сообщений, поэтому храниться и передаваться он должен секретной почтой. В позиции 8, 16, 24, ..., 64 ключа добавляются двоичные символы так, чтобы сумма единиц в байтах была нечетной. Это позволяет проводить проверку ключа при передаче и хранении. 56 битов ключа, находящиеся на позициях 1, 2, 3, ..., 7, 9, 19, 11, ...17, 19, 20, 21, ...63, подвергаются следующей перестановке:

57	49	41	33	25	17	9	
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	Блок C_0
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	Блок D_0
21	13	5	28	20	12	4	

Перестановка определяется двумя блоками C_0 и D_0 по 28 бит в каждом. Далее используются итеративные процедуры преобразования. Получив некоторые блоки C_{n-1}, D_{n-1}, строим блоки C_n, D_n для $n = 1, 2, 3, 4, ...,$ 16 одним или двумя левыми сдвигами из блоков C_{n-1}, D_{n-1} в соответствии со следующей таблицей сдвигов:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Число левых сдвигов	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

При левых сдвигах все элементы блока смещаются влево на одну или две позиции циклически в пределах данного блока. Из блоков C_nD_n строятся перестановки K_n, состоящие из 48 бит (биты 9, 18, 22, 25, 35, 38, 43, 54 в перестановки не входят). Остальные биты переставляются следующим образом:

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

Все приведенные вычисления являются предварительными. Из исходного ключа вычислено 16 последовательностей K_n по 48 бит в каждой.

Для шифрования сообщений они представляются двоичными блоками w, содержащими по 64 бита. Сначала блок подвергается начальной перестановке:

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	18	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

Полученный блок w' условно разбивается на два блока $w' = L_0R_0$, где каждый содержит по 32 бита. Далее используется итерационная процедура. Построив блоки L_{n-1} и R_{n-1} , $1 \le n \le 16$, определим L_n и R_n следующим образом:

$$L_n = R_{n-1},$$
$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n),$$

где $\oplus$ – сложение по модулю 2, а функция f будет определена далее.

Расшифрование сообщения производится достаточно просто: приведенные уравнения могут быть представлены в виде:

$$R_{n-1} = L_n.,$$
$$L_{n-1} = R_n \oplus f(L_n, K_n).$$

Можно вычислять L_n и R_n, спускаясь до L_0, R_0, после чего расшифрование становится очевидным.

Функция f строит из 32-битовых блоков R_{n-1} или L_n и 48-битового блока K_n 32-битовый блок следующим образом. Блок из 32 бит расширяется до 48 бит в соответствии с таблицей:

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

Таким образом, некоторые биты из 32-битового блока повторяются в 48-битовом блоке. После такого расширения два 48-битовых блока побитно складываются по модулю 2. Результирующий 48-битовый блок делится на 8 блоков по 6 бит каждый $B = B_1B_2B_3...B_8$. Затем каждый из этих восьми блоков B_i трансформируется в 4-битовый блок B_i' с помощью соответствующей таблицы (для каждого блока B_i – своя таблица) и ряда правил.

	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
S_1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

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

Преобразование осуществляется следующим образом. Пусть блок B_3 равен 111011. Первый и шестой разряд представляют число x, $0 \le x \le 3$, а разряды со второго по пятый – число y, $0 \le y \le 15$. Для рассматривае-

мого блока $x = 3$, а $y = 11$. В табл. S_3 на пересечении 3-й строки и 11-го столбца (отсчет начинается с нулевых значений) находим число 3. Его двоичное представление в 4 байтах имеет вид 0011. Значение f получается применением перестановки:

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

к полученному 32-битовому блоку $B_1'B_2'B_3'...B_8'$.

DES-алгоритмы работают очень быстро на специализированном оборудовании. С другой стороны, криптоанализ приводит к многочисленным системам нелинейных уравнений. Число всевозможных ключей, которые необходимо перебрать криптоаналитику очень велико (2^{56}). Кроме того, система DES обладает очень полезной, с точки зрения секретности, особенностью: незначительные изменения исходного сообщения или ключа приводят к значительным искажениям криптотекста.

Российский аналог криптосистемы DES (ГОСТ 28147–89) использует ключ в 256 бит.

2. КРИПТОСИСТЕМЫ С ОТКРЫТЫМ РАСПРЕДЕЛЕНИЕМ КЛЮЧЕЙ

2.1. Общие соображения о системе с открытым распределением ключей

Криптосистемы, описанные ранее, имеют очень серьезный недостаток. По алгоритму зашифрования можно найти алгоритм расшифрования. Поэтому ключи должны храниться в строжайшем секрете или передаваться от отправителя к получателю по сверхсекретной почте. Даже в очень сложной системе DES ключи зашифрования и расшифрования совпадают. Часто ключ по длине равен сообщению, поэтому возникает вопрос, а не проще ли вместо передачи ключей передавать по сверхсекретной почте само сообщение? Конечно, это целесообразно далеко не во всех случаях. Например, ключ можно передать заранее, когда сообщение еще не сформировано или воспользоваться одним и тем же ключом несколько раз.

Однако существуют системы, в которых можно раскрыть принцип шифрования без существенного уменьшения секретности передачи в целом. Такие системы называют системами с открытым распределением ключей. Впервые идея таких систем была представлена Диффи и Хелманом. Идея необычайно проста по существу, но произвела революцию в криптографии.

Идея основана на использовании так называемых односторонних функций. Пусть по аргументу x легко вычисляется функция $f(x)$. А по значениям $f(x)$ аргумент x вычисляется очень трудно. Тогда перехватив сообщение $f(x)$, недоброжелатель не сможет легко найти x – истинное передаваемое сообщение. Однако эти трудности, казалось бы, будут существовать и для легального получателя сообщения? Но для него предлагается оставить “лазейку” в виде некоторых знаний, которые помогут ему быстро расшифровать полученное сообщение.

Одним из хороших примеров, иллюстрирующих эту идею, является следующий [2].

Для ловли рыбы в северных странах используются так называемые “морды” – плетеные корзины с конусообразным углублением, с одной стороны, и маленьким отверстием в центре этого углубления. Рыба легко заходит в эту корзину, а обратный выход из нее найти рыбе трудно. В то же время рыбак открывает заднюю крышку и легко выбирает рыбу.

Приведем еще один пример, наглядно иллюстрирующий принципы работы системы с открытым распределением ключей.

Пусть шифрование сообщений выполняется с использованием телефонного справочника. Для большого города такой справочник может содержать несколько томов, в которых фамилии каждого абонента сопоставлен телефонный номер. Зашифруем сообщение: СОВЕРШЕННО СЕКРЕТНО. В таблице показано, как это можно сделать.

Шифруемая буква сообщения	Фамилия абонента	Телефон абонента
С	Соловьев	5830247
О	Омельченко	1422576
В	Владимиров	3523397
Е	Евдокимов	2355495
Р	Родионов	1152843
Ш	Шапиро	4326567
Е	Емельянов	3768789
Н	Новиков	1754328
Н	Николаев	5377891
О	Осипов	2844390
С	Сасковец	3155639
Е	Есиков	4166720
К	Коновалов	1944267
Р	Ресин	2255129
Е	Елкин	4933221
Т	Тимофеев	2288546
Н	Никитин	2190032
О	Окунев	3811020

Вместо буквы исходного сообщения передается номер телефона абонента, фамилия которого начинается с этой буквы. Поскольку фамилия по начальной букве выбирается случайным образом, то одинаковым буквам могут соответствовать различные номера телефонов. Однако расшифрование при этом производится однозначно. Криптоаналитику, пытающемуся вскрыть шифр, придется решать обратную задачу: по номеру телефона искать фамилию абонента, имеющего этот телефон. Эта задача при наличии только алфавитного телефонного справочника может оказаться чрезвычайно громоздкой. Однако легальный получатель сообщения имеет "лазейку" в виде телефонного справочника, составленного в соответствии с возрастающими номерами телефонов. В этом случае найти по телефону фамилию абонента труда не представляет. Конечно, этот пример приводится не для практического использования, так как всем понятно, что можно составить обратный справочник. Однако он иллюстрирует идею открытого распределения ключей и принцип "лазейки".

В настоящее время разработаны различные криптосистемы с открытым распределением ключей. Все они основаны на односторонних функциях, когда задача шифрования выполняется легко, а обратная задача – очень сложно. Большая часть этих идей использует математический аппарат теории чисел. Приведем перечень некоторых фундаментальных задач теории чисел, для которых оценка сложности еще не определена, но решение которых представляется достаточно трудоемким.

FACTOR (n). Найти разложение большого числа n на множители.

PRIMALITY (n). Решить вопрос о том, является ли n простым числом?

FIND-PRIME ($>n$). Найти простое число, большее n.

SQUAREFREENESS (n). Решить, делит или нет квадрат простого числа число n?

QUAD-RESIDUE (a, n). Решить, выполняется или нет сравнение $x^2 \equiv a \bmod n$ для некоторого x?

SQUAREROOT (a, n). Найти, если возможно, такое число x, что $x^2 \equiv a \bmod n$.

DISCRETE-LOG (a, b, n). Найти, если возможно, такое x, что $a^x \equiv b \bmod n$.

Общий прием, используемый при построении криптосистем с открытым ключом, состоит в следующем:

1. Выбирается односторонняя функция, такая, что по x легко находится $f(x)$, но по $f(x)$ значение x находится очень трудно. Функция $f(x)$ объявляется открыто.

2. Находится легкая подзадача определения по $f(x)$ значения x.

3. Значения передаваемой функции $f(x)$ перемешиваются ("взбиваются"), так, чтобы для криптоаналитика она выглядела как труднорешаемая, а для легального получателя открывается способ сведения передаваемых значений функции к легкой подзадаче, что является "лазейкой" для него.

Очень показательными в этом случае являются так называемые "рюкзачные" криптосистемы, к рассмотрению которых мы и переходим.

2.2. Рюкзачные системы

Рассмотрим вектор A = (a_1 a_2 a_3 ...a_i ... a_n), где a_i – различные целые положительные числа. Пусть, кроме того, задано некоторое положительное число k. Поставим задачу: выбрать числа a_i среди элементов вектора A, такие, чтобы их сумма в точности равнялась k. В этом случае можно говорить, что k определяет размер рюкзака, а выбранные a_i – размеры предметов, которые помещаются в рюкзак.

Пусть вектор A = (43 129 215 473 903 302 561 1165 697 1523), а k = 3231. С помощью перебора можно найти: 3231 = 129 + 473 + 903 + 561 + 1165.

С другой стороны, если взять вектор-столбец вида (0 1 0 1 1 0 1 1 0 0) и умножить вектор-строку A на этот вектор-столбец:

$$(43\,129\,215\,473\,903\,302\,561\,1165\,697\,1523) * \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 3231,$$

то получим значение k.

Если каждую букву английского алфавита закодировать двоичным пятиразрядным кодом и разбить передаваемое сообщение на пары букв, то каждая пара будет кодироваться 10 двоичными символами. Будем умножать вектор-строку A на векторы-столбцы кодов пар букв, при этом будем получать шифрованное сообщение $k_1\ k_2\ k_3$ ... Шифрование осуществляется очень просто, а вот для расшифровки полученного сообщения потребуется решать задачу о рюкзаке: по значению k находить двоичный 10-разрядный вектор-столбец, при умножении на который вектора-строки A будет получено данное k. Эта задача явно односторонняя: легкое шифрование и очень трудное расшифрование. В принципе, можно ее решать с помощью полного перебора (2^{10} вариантов). Однако трудности расшифрования будут одинаковыми как для криптоаналитика, так и для легального получателя, что, безусловно, нежелательно. Для того чтобы облегчить легальному получателю расшифровку сообщения, нужно предложить ему "лазейку". В данном случае "лазейка" может быть устроена следующим образом. Рассмотрим сверхрастущие векторы $A = (a_1\ a_2\ a_3\ ...a_i\ ...\ a_n)$, в которых $a_i > \sum_{l=1}^{i-1} a_l$, т. е. каждый элемент вектора A больше суммы всех предыдущих элементов. Например, A = (25 27 56 112 231 452 916 1803). Нахождение элементов вектора A, сумма которых дает заданное число k, элементарно (если такое решение имеется). Действительно, пусть k = 1449. Поскольку 1449<1803, то последний элемент вектора не входит в решение. Далее, так как 1449>916, то 916 обязательно входит в решение, так как сумма всех предыдущих элементов меньше 916. Рассуждая аналогично, получаем код позиций выбираемых элементов: (1 0 1 0 0 1 1 0). Однако если опубликовать этот сверхрастущий вектор, то и для криптоаналитика задача расшифрования сообщений станет элементарной. Тогда мы преобразуем сверхрастущий вектор A в вектор B по следующему правилу: выберем некоторый модуль m, больший суммы всех элементов A, возьмем некоторое t, такое, что $(t, m) = 1$ и каждый элемент вектора B будем вычислять по правилу: $b_i \equiv a_i \cdot t \bmod m$. Вектор B уже не будет казаться сверхрастущим, и он может быть опубликован в качестве открытого ключа. А в качестве "лазейки" для легального пользователя сообщим ему значения t и m.

2.3. Пример использования рюкзачных систем для криптографии с открытым распределением ключей

Пусть A – сверхрастущая последовательность чисел: (1 2 4 8 16).

Легко видеть, что $a_i > \sum_{l=1}^{i-1} a_l$, т. е. суммы всех предыдущих значений.

Передаваемые сообщения пусть представляют собой следующие пятиразрядные двоичные коды:

$w_1 = (1\ 0\ 1\ 1\ 0)$; $w_2 = (\ 0\ 1\ 1\ 0\ 1)$; $w_3 = (\ 1\ 0\ 0\ 0\ 1)$.

Выполним сильное модульное преобразование вектора A. Для этого выберем значение модуля m так, чтобы он был больше суммы всех a_i. Одновременно нужно выбрать t так, чтобы $(t, m) = 1$.

При этом желательно, чтобы даже первые значения $a_i t$ были бы больше модуля m (чтобы и по ним нельзя было бы сразу определить t). Выбираем $t = 40$ и $m = 37$.

Элементы вектора B получаются следующим образом: $b_i \equiv a_i t \bmod m$. Сам вектор $B = (3\ 6\ 12\ 24\ 11)$. Он уже не является сверхрастущим вектором, поэтому криптоанализ будет затруднен. Умножая вектор B на матрицу, составленную из двоичных векторов w_i , получаем зашифрованный текст:

$$B \cdot W = (3\ 6\ 12\ 24\ 11) \cdot \begin{pmatrix} 1\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 1\ 0 \\ 1\ 0\ 0 \\ 0\ 1\ 1 \end{pmatrix} = (39\ 29\ 14) = X,$$

где X – вектор шифрованного сообщения.

Именно этот шифрованный текст передается по каналу связи и может быть перехвачен криптоаналитиком. Кроме того, считается, что ключ (вектор B) также открыто сообщается всем (как получателю сообщения, так и может быть перехвачен любым пользователем сети). Значения t и m являются “лазейкой” для легального получателя сообщений. Он находит значение u такое, что $t \cdot u \equiv 1 \bmod m$. В нашем случае легко находится $u = 25$, так как $40 \cdot 25 \equiv 1 \bmod 37$.

Затем легальный получатель сообщения также легко находит преобразование вектора X:

$39 \cdot 25 \equiv 13 \bmod 37$; $29 \cdot 25 \equiv 22 \bmod 37$; $14 \cdot 25 \equiv 17 \bmod 37$. Теперь получатель имеет вектор X'= (13 22 17) и ему остается вычислить исходный вектор A , зная вектор B и u:

$3 \cdot 25 \equiv 1 \bmod 37$; $6 \cdot 25 \equiv 2 \bmod 37$; $12 \cdot 25 \equiv 4 \bmod 37$; $24 \cdot 25 \equiv 8 \bmod 37$; $11 \cdot 25 \equiv 16 \bmod 37$. Исходный вектор A = (1 2 4 8 16) является сверхрастущим, поэтому по вектору A и X' легко дешифруется переданное сообщение:

$$\begin{pmatrix} 1\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 1\ 0 \\ 1\ 0\ 0 \\ 0\ 1\ 1 \end{pmatrix}$$

2.4. Плотные рюкзаки

Задача о рюкзаке, лежащая в основе базисного варианта систем с открытым распределением ключей, имеет низкую плотность, в том смысле, что компоненты рюкзачного вектора располагаются в отрезке от 1 до n очень редко. Плотность рюкзака определяется следующим образом:

$$d(\mathrm{A}) = \frac{n}{\log_2 \max \mathrm{A}}.$$

Так, для сверхрастущего вектора A = (1 2 4 8 16 32 64) d(A) = 7/6.

Увеличить плотность рюкзачного вектора можно с использованием полей Галуа GF(p^h), где p – простое; h – целое.

Конечное поле F(p^h) содержит p^h элементов. Основное поле F(p) , которое является подполем поля F(p^h), содержит p элемента (0, 1, 2, 3, ..., p–1) и 2 операции: $\oplus \bmod p$ и $\oplus \bmod p$.

Элемент α называется алгебраическим степени h над полем F(p), если и только если α удовлетворяет в F(p) уравнению: $P(x) = 0$, где $P(x)$ – многочлен степени h, но не удовлетворяет никакому уравнению с многочленом меньшей степени. Это влечет неприводимость многочле-

на $P(x)$. Все p^h элементов поля $F(p^h)$ могут быть представлены в виде

$$\sum c_j \alpha^i ,$$

где $0 \le c_j \le p-1$; $0 \le i \le h-1$.

При вычислениях степень α^s, где $s \ge h$ заменяется на меньшую в соответствии с уравнением $P(\alpha) = 0$.

Пусть, например, $p = 3$, $h = 2$ и α удовлетворяет уравнению $x^2 - x - 1 = 0$. Элементы поля $F(3^2)$ можно выразить как:

$$0, 1, 2, \alpha, \alpha + 1, \alpha + 2, 2\alpha, 2\alpha + 1, 2\alpha + 2.$$

В вычислениях понижение степеней производится с использованием равенства $\alpha^2 = \alpha + 1$. Например, $(2\alpha + 1)(\alpha + 2) = 2\alpha^2 + \alpha + 4\alpha + 2 = 2(\alpha + 1) + 5\alpha + 2 = 7\alpha + 4 = \alpha + 1$.

Элемент $\beta \ne 0$ поля $F(p^h)$ называется образующей $F\cdot(p^h)$ мультипликативной группы ненулевых элементов поля $F(p^h)$, если степени β^i , $i = 1, 2, 3, ..., p^h - 1$ пробегают все ненулевые элементы поля $F(p^h)$. Образующая может рассматриваться как основание log. Такие логарифмы называются дискретными логарифмами. Рассмотрим, например, все 8 степеней корня α в приведенном примере и запишем результат в виде таблицы:

i	1	2	3	4	5	6	7	8
α^i	α	$\alpha+1$	$2\alpha+1$	2	2α	$2\alpha+2$	$\alpha+2$	1

Из таблицы видно, что α является образующей. Эта таблица может быть представлена как таблица дискретных логарифмов. Для этого в верхней строке запишем упорядоченные элементы поля, а в нижней – значения степеней образующего элемента, при которых получаем данный элемент поля:

y	1	2	α	$\alpha+1$	$\alpha+2$	2α	$2\alpha+1$	$2\alpha+2$
$\log_\alpha y$	8	4	1	2	7	5	3	6

Считается, что вычисление дискретных логарифмов является трудной задачей, как и задача факторизации (разложения на множители). Таблица логарифмов может использоваться для выполнения умножения и деления элементов поля. Заметим, что операции выполняются по модулю $p^h - 1$, в данном примере по модулю $3^2 - 1 = 8$. Для примера:

$\log((\alpha + 2)(2\alpha + 1)) = \log(\alpha + 2) + \log(2\alpha + 1) = 7 + 3 = 10 \equiv 2 \bmod 8$. Что соответствует элементу $\alpha + 1$.

$\log((\alpha + 1)/(2\alpha + 2)) = 2 - 6 = -4 \equiv 4 \bmod 8$, что соответствует элементу 2.

Можно проверить, что кроме элемента α образующими также являются элементы $2\alpha + 1$, $\alpha + 2$ и 2α. Если $s = p^h - 1$ есть наименьшая положительная степень, удовлетворяющая уравнению $\beta^s = 1$, то β является образующей. Поэтому число образующих элементов поля равно $\varphi(p^h - 1)$, где φ – функция Эйлера. Для нашего примера $\varphi(8) = 4$.

Рассмотрим вспомогательную задачу. Пусть имеется рюкзачный вектор $A = (a_1\ a_2\ a_3\ ...a_i\ ...\ a_n)$. Рассмотрим различные суммы элементов, состоящие из h компонент. Задача состоит в том, чтобы для заданных n и h построить вектор A такой, чтобы все суммы из h элементов были бы попарно различны. Для рюкзаков низкой плотности это сделать легко: $a^i = h^{i-1}$, $1 \le i \le n$. Например,

$$A = (1\ 2\ 4\ 8\ 16\ 32\ 64),\ h = 2,\ n = 7;$$

$$A = (1\ 3\ 9\ 27\ 81\ 243\ 729\ 2187\ 6561),\ h = 3,\ n = 9.$$

Легко проверяется, что любые пары в первом векторе различны и любые тройки во втором также различны.

Такое построение векторов A соответствует рюкзакам низкой плотности, так как они являются сверхрастущими.

Пусть $n = p$ – простое число. Показано, что для простого p и целого $h \ge 2$ всегда можно построить рюкзачный вектор $A = (a_1\ a_2\ a_3\ ...a_i\ ...\ a_p)$, удовлетворяющий следующим условиям:

а) $1 \le a_i \le p^h - 1$, для $1 \le i \le p$.

б) Если x_i и y_i – неотрицательные целые числа, такие, что $(x_1, x_2, ..., x_p) \ne (y_1, y_2, ..., y_p)$, но $\sum x_i = \sum y_i = h$, тогда

$$\sum x_i a_i \ne \sum y_i a_i \quad (*)$$

При построении вектора A можно взять $a_i = \log_g(\alpha + i - 1)$, $1 \le i \le p$, где α– корень неприводимого многочлена; g – образующая группы $F \cdot (p^h)$.

Аналогичное построение может быть выполнено для $n = p^s$, где p – простое; s – целое. Кроме того, неравенство (*) может быть заменено на более сильное: $\sum x_i a_i \not\equiv \sum y_i a_i \bmod (p^h - 1)$.

При построении криптосистемы исходный текст должен состоять из p-разрядных блоков, в каждом из которых сумма элементов равняется h. Для того чтобы обеспечить это условие, необходимо после кодирова-

ния букв сообщений произвести перекодирование равновесными кодами длины p веса h (следует отметить, что значения элементов равны 0, 1, 2, ..., $p - 1$)

Пример 1.

Пусть имеем поле $F(3^2)$ и α корень уравнения $x^2 = x + 1$.Выберем образующую $g = 2\alpha + 1$. Поскольку логарифмы элементов α, $\alpha + 1$, $\alpha + 2$ есть соответственно 3, 6 и 5, то вектор A = (3 6 5). Исходные тексты есть векторы размерности 3 с суммой компонент, равной 2. Возьмем такие векторы исходного текста: (2 0 0), (0 1 1), (0 2 0), (1 0 1). Произведем шифрование исходных текстов:

$$(3\ 6\ 5)\cdot\begin{pmatrix}2 & 0 & 0 & 1\\ 0 & 1 & 2 & 0\\ 0 & 1 & 0 & 1\end{pmatrix} = (6\ 3\ 4\ 8).$$

Это и будет криптотекст.

Напомним, что мы производим вычисления по модулю $p^h - 1$. В данном случае по модулю 8. Легальный получатель вычисляет:

$$(2\alpha + 1)^6 = \alpha + 1,$$
$$(2\alpha + 1)^3 = \alpha,$$
$$(2\alpha + 1)^4 = 2,$$
$$(2\alpha + 1)^8 = 1.$$

Если к полученным степеням прибавлять многочлен $\alpha^2 - \alpha - 1$, получим соответственно:

α^2 , $\alpha^2 - 1 = (\alpha + 1)(\alpha + 2)$, $2 = (\alpha^2 - \alpha + 1) = (\alpha + 1)(\alpha + 1)$; $1 = \alpha^2 - -\alpha = \alpha(\alpha + 2)$. Откуда получаем исходные коды:

$$(2\ 0\ 0),\ (0\ 1\ 1),\ (0\ 2\ 0),\ (1\ 0\ 1).$$

Открытым ключом являются A, p, h. Секретной лазейкой являются α и g.

В некоторых применениях такой криптосистемы дополнительно после зашифрования производится перемешивание π и сдвиг (шум) d. Эти значения являются дополнительной лазейкой для легального получателя сообщений.

Пример 2.

Возьмем конечное поле $F(64) = F(2^6)$. Здесь $p = 2$, $h = 6$. Многочлен $x^6 - x - 1$ неприводим над полем F(2), так как ни 0, ни 1 не обращают его в 0. Кстати, при $p = 2$ операции + и – равнозначны. Следовательно, все

элементы поля $F(2^6)$ могут быть представлены через корень α этого многочлена. Более конкретно, все 64 элемента поля $F(2^6)$ могут быть представлены в виде

$$\sum_{i=1}^{6} x_i \alpha^{6-i}, \text{ где } x_i \in \{0, 1\}.$$

Элементы поля при $p = 2$ и $h = 6$ представляются как двоичные наборы длины 6. Так, $\alpha^4 + \alpha^2 + \alpha + 1$ представляется набором: 0 1 0 1 1 1. Возьмем образующую поля $g = \alpha$ – корню неприводимого многочлена. Тогда таблица логарифмов может быть представлена следующим образом:

Элемент	Логарифм	Элемент	Логарифм	Элемент	Логарифм	Элемент	Логарифм
000001	63	010001	24	100001	62	110001	61
000010	1	010010	33	100010	25	110010	46
000011	6	010011	16	100011	11	110011	30
000100	2	010100	14	100100	34	110100	50
000101	12	010101	52	100101	31	110101	22
000110	7	010110	36	100110	17	110110	39
000111	26	010111	54	100111	47	110111	43
001000	3	011000	9	101000	15	111000	29
001001	32	011001	45	101001	23	111001	60
001010	13	011010	49	101010	53	111010	42
001011	35	011011	38	101011	51	111011	21
001100	8	011100	28	101100	37	111100	20
001101	48	011101	41	101101	44	111101	59
001110	27	011110	19	101110	55	111110	57
001111	18	011111	56	101111	40	111111	58
010000	4	100000	5	110000	10		

Поскольку $\log_\alpha \alpha = 1$, а $\log_\alpha(\alpha + 1) = 6$, то вектор A = (1 6). Введем еще шум в виде дополнительного сдвига на $d = 60$. В результате получим открытый вектор B = (61 3), так как элементы b_i вычисляются следующим образом:

$$b_i \equiv a_i \cdot d \bmod (2^6 - 1).$$

Исходными векторами являются векторы (x, y), в которых $x + y = 6$. Используя открытый ключ зашифрования B и $p = 2$, $h = 6$, зашифруем векторы исходного текста:

$$(\,61\;3) \cdot \begin{pmatrix} 6 & 1 & 2 & 3 & 4 & 5 & 0 \\ 0 & 5 & 4 & 3 & 2 & 1 & 6 \end{pmatrix} = (51\;13\;8\;3\;61\;56\;18)$$

Легальный получатель вычитает из каждого шифрованного сообщения *hd* по модулю 63 и получает следующий набор чисел:

$$\begin{gathered} 51 - 6\cdot 60 \equiv 6 \bmod 63, \\ 13 - 360 \equiv 31 \bmod 63, \\ 8 - 360 \equiv 26 \bmod 63, \\ 3 - 360 \equiv 21 \bmod 63, \\ 61 - 360 \equiv 16 \bmod 63, \\ 56 - 360 \equiv 11 \bmod 63, \\ 18 - 360 \equiv 36 \bmod 63. \end{gathered}$$

Из таблицы логарифмов получаем:

$$\begin{aligned} &\alpha^6 = \alpha + 1 = \alpha^6, \\ &\alpha^{31} = \alpha^5 + \alpha + 1 = \alpha\,(\alpha + 1)^5, \\ &\alpha^{26} = \alpha^2 + \alpha + 1 = \alpha^2(\alpha + 1)^4, \\ &\alpha^{21} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1 = \alpha^3(\alpha + 1)^3, \qquad (\cdot) \\ &\alpha^{16} = \alpha^4 + \alpha + 1 = \alpha^4(\alpha + 1)^2, \\ &\alpha^{11} = \alpha^5 + \alpha + 1 = \alpha^5(\alpha + 1), \\ &\alpha^{36} = \alpha^4 + \alpha^2 + \alpha = (\alpha + 1)^6. \end{aligned}$$

Для вычисления значений $(*)$ заметим, что правая часть должна быть представлена в виде: $\alpha^k(\alpha + 1)^{6-k}$, где $k = 1, 2, 3, 4, 5, 6$. В левой части имеются значения α^x, где $x = 6, 31, 26, 21, 16, 11, 36$. Таким образом, при преобразовании $(\cdot)$ решаются уравнения: $\alpha^x = \alpha^k(\alpha + 1)^{6-k}$. Если прологарифмировать его по основанию α, то получим: $x = k + (6-k)\log_\alpha(\alpha + 1)$. Из таблицы дискретных логарифмов находим: $\log_\alpha(\alpha + 1) = 6$, следовательно, $x = 36 - 5k$, откуда получаем $k = (36-x)/5$. Вычисляем:

$$\begin{aligned} &\alpha^6\ (x = 6) = \alpha^6, \\ &\alpha^{31}(x = 31) = \alpha(\alpha + 1)^5\,, \\ &\alpha^{26}(\,x = 26) = \alpha^2(\alpha + 1)^4 \text{ и т.д.} \end{aligned}$$

После этого сразу же получаем исходный кодовый набор:

$$\begin{pmatrix} 6 & 1 & 2 & 3 & 4 & 5 & 0 \\ 0 & 5 & 4 & 3 & 2 & 1 & 6 \end{pmatrix}$$

Пример 3.

Рассмотрим самый общий случай шифрования с использованием конечных полей Галуа. Пусть $p = 5$, $h = 2$. Поле $F(5^2)$ содержит 25 элементов. Легко проверяется, что неприводимым полиномом в подполе $F(5)$ основного поля является многочлен $X^2 + 2$, так как никаким из 5 элементов (0, 1, 2, 3, 4) он не обращается в 0. Пусть α – корень полинома. При вычислениях будем производить замену $\alpha^2 = 3$ (поскольку $-2 \equiv 3 \bmod 5$). Порождающим элементом является $g = \alpha + 1$, что легко проверяется. Составим таблицу логарифмов для элементов поля $F\cdot(5^2)$, где исключен только элемент, равный 0. Так как все элементы поля будут представляться в виде $a_1\alpha + a_2$, где a_1 и $a_2 \in \{0, 1, 2, 3, 4\}$, то таблица логарифмов будет выглядеть следующим образом:

Элемент	Логарифм	Элемент	Логарифм	Элемент	Логарифм
00		20	21	40	15
01	24	21	22	41	5
02	18	22	19	42	16
03	6	23	11	43	20
04	12	24	2	44	13
10	3	30	9		
11	1	31	14		
12	8	32	23		
13	4	33	7		
14	17	34	10		

Составим вектор A по правилу: элемент $a_i = \log_g(\alpha + i - 1)$, где $i = 1, 2, 3, 4, 5$. Тогда получим

$$a_1 = \log_g(\alpha + 0) = 3,$$
$$a_2 = \log_g(\alpha + 1) = 1,$$
$$a_3 = \log_g(\alpha + 2) = 8,$$
$$a_4 = \log_g(\alpha + 3) = 4,$$
$$a_5 = \log_g(\alpha + 4) = 17.$$

Вектор A = (3 1 8 4 17). Сначала введем преобразование π элементов вектора, например, по такому правилу подстановок:

$\pi = \begin{pmatrix} a\ b\ c\ d\ e \\ b\ e\ d\ c\ a \end{pmatrix}$, тогда получим вектор A' = πA = (1 17 4 8 3).

Введем шумовую составляющую в виде сдвига $d = 20$ (по модулю $5^2 - 1 = 24$) всех элементов вектора A', в результате получим открытый вектор B:

$$B = (21\ 13\ 0\ 4\ 23).$$

Шифруемые блоки исходного сообщения должны иметь элементы, сумма которых в каждом столбце равна 2 (в общем случае h). Возьмем несколько произвольных элементов, удовлетворяющих этому условию, и произведем зашифрование:

$$(21\ 13\ 0\ 4\ 23)\cdot\begin{pmatrix}1&0&0&0&0\\0&1&2&0&0\\1&0&0&0&2\\0&0&0&2&0\\0&1&0&0&0\end{pmatrix} = (21\ 12\ 2\ 8\ 0)$$

По каналу связи передается зашифрованное сообщение: 21 12 2 8 0. Кроме того, открытым ключом также являются $p = 5$ и $h = 2$. Легальный получатель, кроме того, знает d, g и π. Поэтому он сразу приступает к расшифровыванию сообщения. Он вычитает из полученных чисел шум hd по модулю 24:

$$21 - 2\cdot 20 \equiv \mathbf{5} \bmod 24.$$
$$12 - 2\cdot 20 \equiv \mathbf{20} \bmod 24.$$
$$2 - 2\cdot 20 \equiv \mathbf{10} \bmod 24.$$
$$8 - 2\cdot 20 \equiv \mathbf{16} \bmod 24.$$
$$0 - 2\cdot 20 \equiv \mathbf{8} \bmod 24.$$

Затем легальный получатель по таблице логарифмов находит значения $gx = (\alpha + 1)^x$, где x – степени, выделенные жирным шрифтом в последних вычислениях. Добавляет к полученному выражению $\alpha^2 + 2$ и раскладывает результат на два множителя:

$$(\alpha + 1)^5 = 4\alpha + 1 + \alpha^2 + 2 = \alpha^2 + 4\alpha + 3 = (\alpha + 1)(\alpha + 3),$$
$$(\alpha + 1)^{20} = 4\alpha + 3 + \alpha^2 + 2 = \alpha^2 + 4\alpha = \alpha(\alpha + 4),$$
$$(\alpha + 1)^{10} = 3\alpha + 4 + \alpha^2 + 2 = \alpha^2 + 3\alpha + 1 = (\alpha + 4)^2,$$
$$(\alpha + 1)^{16} = 4\alpha + 2 + \alpha^2 + 2 = \alpha^2 + 4\alpha + 4 = (\alpha + 2)^2,$$
$$(\alpha + 1)^8 = \alpha + 2 + \alpha^2 + 2 = \alpha^2 + \alpha + 4 = (\alpha + 3)^2.$$

В результате получаем расшифрованные сообщения, которые отличаются от передаваемых только перестановкой π:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 2 & 0 & 0 \end{pmatrix}.$$

После перестановки получаем исходное сообщение:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Плотность построенного рюкзачного вектора равна

$$\mathrm{d}(\mathrm{A}) = \frac{p}{\log_2 \max \mathrm{A}} = \frac{5}{2} = 2{,}5.$$

3. КРИПТОСИСТЕМА RSA

3.1. Идея криптосистемы

Наиболее широко распространенной системой с открытым ключом является криптосистема *RS*A (*Rivest*, *Shamir*, A*dleman*). Идея системы состоит в том, что очень сложно разложить произведение двух простых чисел на сомножители, т. е. найти эти сомножители. Сама же идея системы *RS*A исключительно проста.

Пусть p и q – два случайно выбранных простых числа (каждое примерно по 100 десятичных разрядов). Обозначим:

$$n = pq \text{ и } \varphi(n) = (p-1)(q-1),$$

где $\varphi(n)$ – функция Эйлера от n.

Случайно выбирается большое число $d >> 1$, такое, что $(d, \varphi(n)) = 1$, и вычисляется e, $1 < e < \varphi(n)$, удовлетворяющее сравнению:

$$ed \equiv 1 \bmod \varphi(n).$$

Числа n, e и d называются соответственно модулем, экспонентой зашифрования и экспонентой расшифрования соответственно.

Числа n и e образуют открытый ключ, а p, q, $\varphi(n)$ и d секретную лазейку. При этом секретная лазейка включает в себя взаимозависимые величины. Так, если известно p (и, конечно, n и e), то остальные числа лазейки вычисляются просто:

$$q = n/p;\ \varphi(n) = (p-1)(q-1);$$

d находится из условия: $ed \equiv 1 \bmod \varphi(n)$.

Зашифрование обеспечивается возведением числового фрагмента текста S в степень e по модулю n.

Расшифрование достигается возведением результата предыдущего шага в степень d.

При зашифровании получаем $S^e \equiv C \bmod n$. Здесь C – зашифрованный фрагмент текста.

При расшифровании $C^d = S^{ed} = S^{1+\varphi(n)k} = S^{\varphi(n)k}\, S \equiv S \bmod n$. (*)

Справедливость (*) легко видна, так как из сравнения $ed \equiv 1 \bmod \varphi(n)$ следует, что $ed = 1 + \varphi(n)k$, где k – некоторое целое.

Пример.

Пусть $p = 11$, $q = 13$. Тогда $n = 143$, $\varphi(n) = 120$.

Выберем d из условия: $(d, \varphi(n)) = 1$, например, $d = 37$, тогда из сравнения: $ed \equiv 1 \bmod \varphi(n)$ находим $e = 13$. Действительно,

$$13 \cdot 37 = 481 \equiv 1 \bmod 120.$$

Для зашифрования возьмем фрагмент текста, который закодирован, например, числом $S = 42$.

$$42^{13} \equiv 3 \bmod 143, \text{ т. е. шифр фрагмента } C = 3.$$

Для расшифрования возведем число 3 в степень 37:

$$3^{37} \equiv 42 \bmod 143.$$

Таким образом, легальный получатель вычисляет значение исходного кода фрагмента.

3.2. Использование систем с открытым распределением ключей для абонентских сетей

Рассмотрим несколько близких задач, в которых абоненты обмениваются секретной информацией по открытому каналу.

1. Пусть несколько абонентов A, B, C, договорились об обмене информацией. Они могут выбрать некоторое общее простое число p, такое, что $p - 1$ раскладывается на простые сомножители в первой степени. Число вида $N = p_1 p_2 p_3 \ldots p_k$ называется эвклидовым числом. Каждый из участников выбирает два числа меньших и взаимно простых с $p - 1$ так, чтобы:

$$a_1 a_2 \equiv b_1 b_2 \equiv c_1 c_2 \equiv 1 \bmod (p-1).$$

Пусть абонент А хочет передать сообщение S абоненту B. Он кодирует свое сообщение возведением в степень a_1: $S^{a_1} \equiv S_1 \bmod p$ и передает его B. Тот, в свою очередь, кодирует полученное сообщение возведением в степень b_1 : $S_1^{b_1} \equiv S_2 \bmod p$ и возвращает его А. А возводит его в степень a_2 и передает его B: $S_2^{a_2} \equiv S_3 \bmod p$. B возводит его в степень b_2 и читает сообщение. Справедливость результата следует из сравнения: $a_1 b_1 a_2 b_2 \equiv 1 \bmod (p - 1)$.

Пример. Пусть абоненты A, B и C выбрали число $p = 103$. Это число простое, причем $103 - 1 = 102$ – эвклидово число, так как представляется в виде произведения простых чисел в первых степенях: $102 = 2 \cdot 3 \cdot 17$. Каждый из участников выбирает пару секретных ключей, например:

$$A: a_1 = 25,\ a_2 = 49,$$
$$B: b_1 = 19,\ b_2 = 43,$$
$$C: c_1 = 35,\ c_2 = 35.$$

Пусть теперь A посылает к B сообщение $S = 67$. Он возводит его в степень 25 и находит остаток по модулю 103: $67^{25} \equiv 86 \bmod 103$.

B возводит его в степень $b_2 = 19$ и отправляет результат к А:

$$86^{19} \equiv 96 \bmod 103.$$

A возводит полученное сообщение в степень 49 и передает его B:

$$96^{49} \equiv 21 \bmod 103.$$

B, получив сообщение, возводит его в степень 43 и читает исходный текст: $21^{43} \equiv 67 \bmod 103$. Таким образом, $S = 67$.

Открытым ключом в этой системе является модуль p. Недостатком такой системы является большое число передач от одного абонента к другому.

2. Пусть имеется абонентская сеть и требуется обеспечить связь между любой парой пользователей. Если из одного центра заранее передать открытые ключи g и p каждому пользователю, то они могут выработать общий ключ следующим образом. Пусть абонент А сам придумал ключ k_1, а абонент B ключ k_2 (это индивидуальные секретные ключи абонентов). А посылает к B сообщение $g^{k_1} \bmod p$, а B посылает к А сообщение $g^{k_2} \bmod p$. B возводит полученное сообщение в степень k_2, а А в степень k_1. В результате они выработают одинаковый общий ключ: $g^{k_1 k_2} = g^{k_2 k_1} \equiv K \bmod p$, после чего возможен обмен информацией по открытому каналу с использованием любой классической (симметричной) криптосистемы.

4. КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

В традиционных (классических) криптографических системах предполагалось, что два лица, которые обмениваются секретной информацией, полностью доверяют друг другу и пытаются защитить свои сообщения от третьих лиц (перехватчиков, врагов, криптоаналитиков).

Криптография с открытым ключом значительно расширила класс задач, решаемых с помощью криптографических методов. В результате появилась необходимость в интерактивных , многоразовых двусторонних обменах сообщениями между участниками, которые не всегда доверяют друг другу, в передаче информации между несколькими участниками. Последовательность действий участников обмена информацией, использующих криптографические приемы для решения нетрадиционных задач, называют криптографическими протоколами.

4.1. Банки и вкладчики

Рассмотрим задачу, в которой вкладчики банка v_1, v_2, $\ldots v_k$ передают шифрованное распоряжение ответственному работнику банка B (банкиру). При этом кроме конфиденциальности должна обеспечиваться узнаваемость вкладчика, чтобы по полученному сообщению банкир B сумел идентифицировать автора сообщения и выполнить именно его распоряжение.

Банкир B выбирает некоторое число $N = PQ$, где P и Q – большие простые числа. Каждый из вкладчиков v_i ($i = 1, 2, 3, \ldots, k$) также выбирает свои значения $n_i = p_i q_i$, причем желательно, чтобы $N > n_i$. Затем как банкир, так и вкладчики находят значения $\varphi(N)$ – банкир и $\varphi(n_i)$ – все вкладчики. После чего каждый выбирает свой открытый ключ: S, а банкир и s_i – вкладчики из условий:

$$0 < S < \varphi(N),\ (S, \varphi(N)) = 1 \text{ – банкир и}$$
$$0 < s_i < \varphi(n_i),\ (si, \varphi(n_i)) = 1 \text{ – вкладчики.}$$

Затем банкир и вкладчики находят свои секретные ключи T и t_i из сравнений:

$$ST \equiv 1 \bmod \varphi(N),\ 0 < T < \varphi(N) \text{ – банкир,}$$

$st \equiv 1 \bmod \varphi(n_i)$, $0 < t < \varphi(n_i)$ – вкладчики.

После этих операций открыто публикуется телефонная книга с открытыми ключами:

$$B\colon N, S$$

$$v_i\colon n_i, s_i.$$

Пусть теперь некоторый вкладчик v_i хочет передать распоряжение m банкиру B. Он шифрует его сначала своим секретным ключом (возводя m в степень t_i по mod n_i, а затем открытым ключом банкира: $m_1 \equiv m^{ti} \bmod n_i$, $m_2 \equiv m_1^S \bmod N$. Сообщение m_2 передается по открытому каналу связи. Банкир, получив сообщение m_2, сначала расшифровывает его своим секретным ключом T, а затем открытым ключом вкладчика vi. В результате получает:

$$m_3 \equiv m_{2T} \bmod N,\ m_4 \equiv m_3^{ti} \bmod n_i.$$

При этом $m_4 = m$, т. е. банкир B расшифровывает переданное ему распоряжение, при этом заодно и идентифицирует (узнает) вкладчика. Это похоже на проверку подписи вкладчика и иногда называется “электронная подпись”. Если вкладчик из открытой телефонной книги узнает, что банкир выбрал число $N < n_i$, то изменив порядок шифровки, получит тот же результат, если банкир также изменит порядок расшифровки.

Пример.

Пусть банкир выбрал простые числа $P = 23$, $Q = 11$; вкладчик v: $p = 13$, $q = 7$. После чего и банкир, и вкладчик вычисляют сначала функции Эйлера: $\varphi(23 \cdot 11) = 220$; $\varphi(13 \cdot 7) = 72$, затем выбирают открытые и вычисляют секретные ключи, например: $S = 71$, $T = 31$; $s = 29$, $t = 5$. Открыто публикуются числа: $P \cdot Q = 253$, $p \cdot q = 91$, $S = 71$, $s = 29$. Секретным ключом банкира является число $T = 31$, а секретным ключом вкладчика $t = 5$.

Пусть вкладчик решил дать секретное поручение банкиру в виде числа $m = 41$. Он шифрует его своим секретным ключом t, а затем открытым ключом банкира S:

$$41^5 \equiv 6 \bmod 91,\ 6^{71} \equiv 94 \bmod 253.$$

Это сообщение (число 94) по открытому каналу передается банкиру. Банкир расшифровывает сообщение сначала своим секретным ключом T, а затем открытым ключом вкладчика s:

$$94^{31} \equiv 6 \bmod 253,\ 6^{29} \equiv 41 \bmod 77.$$

Банкир принимает указание вкладчика в виде числа 41.

4.2. Электронные платежи

Рассмотрим протокол электронных платежей, при которых обеспечивается требование неотслеживаемости покупателя (пояснение термина приведено ниже). Пусть в электронных платежах участвуют три стороны: банк, обеспечивающий снятие со счета денег, зачисления на счет и подпись электронной банкноты, покупатель, снимающий со счета в банке некоторую сумму и передающий ее продавцу, и сам продавец (товаров или услуг), получающий электронную банкноту и передающий ее в банк для проверки и зачисления на свой счет.

Будем использовать три транзакции: снятие со счета, платеж и депозит.

При расчетах электронными деньгами необходимо обеспечить:

– безопасность банка, в виде невозможности подделать подпись банка или по набору подлинных банкнот, подписанных банком, создать новую фальшивую банкноту с банковской подписью,

– неотслеживаемость покупателя в виде невозможности сопоставления выданных банкнот, которые возвращаются в банк, и покупателя, получившего эти банкноты.

Эти, на первый взгляд, парадоксальные требования обеспечиваются с помощью так называемой "затемненной" подписи в соответствии со схемой RSA.

Банк выбирает два больших простых числа p и q, открытый ключ e и вычисляет секретный ключ d из условия: $e^d \equiv 1 \bmod \varphi\,(N = pq)$, затем открыто публикует N, e и некоторую одностороннюю функцию $f: Z_N \to Z_N$.

Генерация подписи банка состоит в применении к сообщению m функции дешифрования: $s \equiv m^d \bmod N$. В простейшем варианте электронной подписи ключом d соответствует банкнота достоинством в 1 денежную единицу (рубль, сто рублей, тысяча и т.п.). Для получения нескольких денежных единиц в данном протоколе необходимо каждый раз обращаться в банк. Более сложный протокол обеспечивает получение электронной банкноты произвольного достоинства.

При снятии со счета покупатель выбирает некоторое случайное число $n \in Z_N$ и вычисляет $f(n)$.Покупатель хочет получить от банка подпись на банкноте, т. е. $f(n)^d$. Однако если он просто пошлет в банк значение $f(n)$, то нарушится условие неотслеживаемости, так как банк, снимая со счета покупателя сумму, сопоставит покупателю выданную (подписанную банком) банкноту. Для устранения этого покупатель выбирает некоторое случайное число $r \in Z_N$., $r \neq 0$, вычисляет $f(n)r^e \bmod$

N, и посылает результат в банк. Банк вычисляет значение $f(n)^d r \bmod N$ и возвращает его покупателю. Покупатель "снимает" затемняющий множитель r и получает подписанную банкноту в виде $(n, f(n)^d \bmod N)$. Этой банкнотой он будет рассчитываться с продавцом.

В транзакции платежа покупатель передает продавцу электронную банкноту $(n, f(n)^d \bmod N)$, которую тот может проверить самостоятельно, вычислив по n значение $f(n)$, а затем проверив условие $f(n) \equiv \equiv (f(n)^d)^e \bmod N$.

Однако для того, чтобы исключить оплату одной и той же банкнотой нескольких покупок, продавец отправляет электронную банкноту в банк для проверки и зачисления на свой счет. Банк по реестру проверяет, не была ли ранее потрачена эта банкнота и зачисляет денежную квоту на счет продавца.

При снятии со счета у банка остается некоторое значение $f(n)^d r \bmod N$, которое из-за затемняющего множителя r представляет собой просто случайное число.

Пример.

Пусть банком выбраны простые числа $p = 17$ и $q = 19$. $N = 323$, $\varphi(323) = 288$. Открытый ключ $e = 11$, секретный ключ $d = 131$ и односторонняя функция $f(x) = x^2 \bmod 323$.

Открыто публикуются $N = 323$, $e = 11$ и вид функции f.

При снятии со счета покупатель выбирает случайное число $n = 25$, вычисляет $f(25) = 25^2 \equiv 302 \bmod 323$, выбирает затемняющий множитель $r = 20$, вычисляет $f(n)r^e \bmod N$, которое в данном примере будет равно: $302 \cdot 20^{11} \bmod 323 \equiv 74 \bmod 323$ и посылает в банк число 74. Банк возводит полученное число в степень $d = 131$ и получает: $74^{131} \equiv 63 \bmod 323$. Покупатель снимает затемняющий множитель $r = 20$, решая сравнение: $f(25)^{131} \equiv 310 \bmod 323$. Таким образом, подписанная банком электронная банкнота имеет вид: (25, 310). Эта банкнота отправляется продавцу. Если продавец хочет самостоятельно проверить подлинность банкноты, то он может вычислить $F(25) \equiv 302 \bmod 323$ и затем $310^{11} \equiv 302 \bmod 323$. Совпадение полученных значений свидетельствует о подлинности банкноты. Однако для гарантии того, что эта банкнота не использовалась в других платежах, продавец отправляет ее в банк, который по реестру проверяет, что она впервые используется в платеже и зачисляет одну денежную единицу на счет продавца.

4.3. Проверка подлинности авторства передаваемых документов

В некоторых случаях, сделка, совершенная по сети, должна быть юридически признаваемой, чтобы ни один из партнеров не мог отказаться от авторства переданного сообщения. Для этого существует специальный прием, называемый “цифровой подписью”. В принципе, протокол, рассмотренный в предыдущем разделе, также позволяет банкиру проверить авторство присланного сообщения и может рассматриваться как процедура “цифровой подписи”. Известно несколько разновидностей цифровых подписей. Рассмотрим протокол, предложенный Шнором и называемый протоколом аутентификации (доказательство того, что автор сообщения владеет секретным ключом).

Пусть p и q – простые числа, причем q делит $p - 1$. Например, $p = 23$, $q = 11$, или $p = 59$, $q = 29$. Пусть выбрано g такое, что $g^q \equiv 1 \bmod p$. Для первого примера g = 2, так как $2^{11} \equiv 1 \bmod 23$, для второго примера g = 4, так как $4^{29} \equiv 1 \bmod 59$.

Обычно протоколы описываются, как обмен информацией между двумя абонентами, например, Алисой и Бобом. Алиса выбирает некоторое случайное число x из диапазона чисел: $\{0, 1, 2, \ldots q-1\}$ и вычисляет ключ $y = g^{-x} \bmod p$, который открыто публикует.

Далее Алиса выбирает случайное число k из множества $\{0, 1, 2, \ldots q-1\}$, вычисляет $r = g^k \bmod p$ и r отправляет к Бобу.

Боб выбирает некоторый случайный запрос e из множества

$$\{0, 1, 2, \ldots 2^t - 1\},$$

где t – некоторое целое и посылает e к Алисе.

Алиса вычисляет $s = k + xe \bmod q$ и посылает s к Бобу на проверку.

Боб проверяет соотношение: $r = g^s y^e \bmod p$ и, если оно выполняется, принимает доказательство Алисы, что она владеет секретным ключом, в противном случае доказательство отвергается.

Пример.

Пусть $p = 59$, $q = 29$, тогда можно взять g = 4. Случайное число x, выбираемое Алисой, $x = 9$, тогда $y = 4^{-9} \equiv 17 \bmod 59$. Число 17 является открытым ключом Алисы.

Далее Алиса выбирает k например, $k = 12$, вычисляет $r = 4^{12} \equiv 35 \bmod 59$ и число $r = 35$ посылается к Бобу. Пусть случайный запрос Боба $e = 10$, который он посылает к Алисе. Алиса вычисляет: $s = 12 + 9 \cdot 10 \bmod 29 \equiv 15 \bmod 29$. Число $s = 15$ посылается к Бобу.

Боб проверяет соотношение:

$$4^{15}17^{10} \equiv 57 \cdot 12 \equiv 35 \bmod 59.$$

Поскольку $r = 35$, то доказательство принимается.

4.4. Разделение секрета

Рассмотрим случай, когда руководитель банка или какой-либо другой организации не полностью доверяет своим сотрудникам и хочет подстраховаться при использовании секретного ключа. Он может разделить весь секретный ключ (двоичная или десятичная последовательность символов) на отдельные фрагменты и эти фрагменты раздать нескольким сотрудникам так, чтобы при общем числе сотрудников n полный ключ мог быть ими составлен, если соберутся вместе не менее h сотрудников.

Наиболее просто поставленная задача решается при $h = n$, т. е. когда ключ раздается n сотрудником и требуется наличие всех n фрагментов ключа, чтобы собрать полностью секретный ключ S. Выберем некоторое простое число p и пусть секретный ключ представляется в виде набора $(s_1, s_2, s_3, \ldots, s_k)$, где все s_i являются элементами поля $GF(p)$. Разделим секретный ключ на n фрагментов следующим образом. Будем генерировать произвольные случайные числа: $(a_{11}, a_{12}, a_{13}, \ldots, a_{1k})$ – фрагмент секретного ключа 1-го сотрудника,
$(a_{21}, a_{22}, a_{23}, \ldots, a_{2k})$ – фрагмент секретного ключа 2-го сотрудника,
……
$(a_{n-11}, a_{n-12}, a_{n-13}, \ldots, a_{n-1k})$ – фрагмент секретного ключа n–1-го сотрудника.

А последнему n-му сотруднику вычислим элементы его фрагмента секретного ключа по следующему правилу:

$$a_{n1} = s_1 - a_{11} - a_{21} - a_{31} - \ldots - a_{n1} \bmod p.$$
$$a_{n2} = s_2 - a_{12} - a_{22} - a_{32} - \ldots - a_{n2} \bmod p.$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_{nk} = s_k - a_{1k} - a_{2k} - a_{3k} - \ldots - a_{nk} \bmod p.$$

В этом случае только при сложении всех фрагментов ключа по модулю p получим полный секретный ключ.

Пример.

Пусть $p = 29$ и секретный ключ имеет вид: (26, 13, 21, 8, 0, 18). Требуется разделить его на 5 фрагментов для раздачи 5 сотрудникам. Для первых четырех из них генератор случайных чисел по модулю 29 пусть выработал фрагменты:

(26, 0, 13, 11, 23, 25)
(2, 7, 15, 12, 27, 6)
(1, 3 , 24, 6, 0, 16)
(12, 2, 7, 0, 7, 0)

Для последнего, пятого сотрудника вычисленный фрагмент имеет вид:

(14, 1, 20, 8, 1, 0).

Легко проверяется, что сложение всех фрагментов (каждый элемент складывается по модулю 29) дает полный секретный ключ (26, 13, 21, 8, 0, 18)

Следует заметить, что сложность подбора секретного ключа в рассмотренном случае зависит только от значения модуля p и числа элементов k и практически не зависит от количества сотрудников n.

Рассмотрим случай, когда $h < n$. Имеется несколько вариантов решения такой пороговой задачи.Приведем алгоритм, описанный в [2]. Он основан на модульной арифметике и китайской теореме об остатках.

Имеется n участников $A_1, A_2, A_3, \ldots, A_n$. Пусть m_i, $i = 1, 2, \ldots n$, целые числа, большие 1, такие, что $(m_i, m_j) = 1$ при $i \neq j$. Обозначим M – произведение всех чисел m_i, т. е. $M = m_1 m_2 m_3 \ldots, m_i, \ldots m_n$.

Обозначим также M_i – произведение всех m_j ($j = 1, 2, \ldots, i–1, i+1, \ldots, n$), кроме m_i, т. е. $M_i = M / m_i$. Вычислим значения N_i из условия: $M_i N_i \equiv \equiv 1 \bmod m_i$. Поскольку $(M_i, m_i) = 1$, то решение всегда существует и все N_i будут найдены.

Если имеется n сравнений вида: $x \equiv a_i \bmod m_i$, $i = 1, 2, \ldots, n$, a_i – целые, то общее решение этих сравнений имеет вид:

$$x = \sum_{i=1}^{n} a_i M_i N_i.$$

Кроме того, это решение единственное, т. е. любое другое решение y удовлетворяет сравнению: $y \equiv x \bmod M$.

Пусть теперь h фиксированный порог, $1 < h \leq n$. Обозначим через $\min(h)$ – наименьшее из h произведений m_i, а $\max(h–1)$ – наибольшее из $h–1$ произведений m_i. Если выполнены условия:

$$\min(h) – \max(h–1) \geq 3 \max(h–1) \quad (*)$$
$$\text{и } \max(h–1) < c < \min(h) \quad (**)$$

то множество $\{a_1, a_2, \ldots, a_t\}$, где $a_i \equiv c \bmod m_i$, образует (h, n) пороговую схему для c [2]. Это означает, что если c – некоторый секретный ключ,

а a_i – фрагменты ключа, розданные n участникам, то любые h из участников смогут восстановить значение c по его фрагментам, а любые h–1 участников сделать это не смогут (без перебора вариантов). При этом чем больше в (*) разность, тем труднее h–1 участникам подобрать секретный ключ по своим фрагментам.

Пример.

Пусть $n = 5$ и $m_1 = 97$; $m_2 = 98$, $m_3 = 99$, $m_4 = 101$, $m_5 = 103$.

Возьмем $h = 3$ и вычислим $\min(3) = (97 \cdot 98 \cdot 99) = 941094$; $\max(2) = (101 \cdot 103) = 10403$. Неравенство (*) примет вид:

$$941094 - 10403 = 930691 > 3 \cdot 10403 = 31209.$$

Секретное число с должно лежать в пределах (**). Пусть оно известно некоторому сотруднику, разделяющему секрет, который вычислил значения a_i ($i = 1, 2, \ldots, 5$) из условий $a_i \equiv c \bmod m_i$ и раздал фрагменты секрета пяти участникам: $a_1 = 62$, $a_2 = 4$, $a_3 = 50$, $a_4 = 50$, $a_5 = 38$.

Пусть теперь трое из пяти участников, например, A_2, A_3 и A_4 пытаются восстановить секретный ключ c по своим фрагментам. Поскольку каждый участник знает только свое значение m_i, то они вычисляют:

$$M_2' = m_3 \cdot m_4 = 9999;\ M_3' = m_2 \cdot m_4 = 9898;\ M_4' = m_2 \cdot m_3 = 9702,$$

а затем соответствующие значения N_i: $N_2 = 33$, $N_3 = 49$, $N_4 = 17$. После чего находят значение $y = 4 \cdot 9999 \cdot 33 + 50 \cdot 9898 \cdot 49 + 50 \cdot 9702 \cdot 17 = 33816668$. Секретный ключ вычисляется из сравнения: $c \equiv y \bmod (m_2 \cdot m_3 \cdot m_4)$. Таким образом,

$$c \equiv 33816668 \bmod (98 \cdot 99 \cdot 101) \equiv 500000 \bmod 979902.$$

Если взять любую другую тройку клиентов, например, A_1, A_4, A_5, то они вычислят тот же секретный ключ $c = 500000$.

Пусть теперь двое клиентов, например, A_2 и A_5 пытаются найти секретный ключ c. Они вычисляют значения $y = 4 \cdot 103 \cdot 59 + 38 \cdot 98 \cdot 41 = 176992 \equiv 5394 \equiv c \bmod 10094$. Они понимают, что истинный секретный ключ находится из условий: $5394 + i \cdot 10094$, но значения i не знают. Количество значений i определяется как целая часть дроби:

$$\frac{\min(h) - \max(h-1) - 1}{\max(h-1)}.$$

Для рассматриваемого примера целая часть дроби равна 89, т. е. двум участникам потребуется перебрать 89 вариантов ключа. В реальных условиях количество вариантов может быть сделано существенно большим.

Рассмотрим еще один пример разделения секретного ключа, описанный в [9].

Пусть к приему сообщения допущено n сотрудников, из которых не все могут оказаться на месте во время приема. Фрагменты ключа распределяются между сотрудниками по определенному правилу, причем так, что ни один сотрудник не имеет полного набора фрагментов ключа. Сообщение может быть расшифровано, если соберутся h или более сотрудников (т. е. h сотрудников должны иметь полный набор фрагментов), при этом $1 \le h \le n$. В дальнейшем слова "фрагмент ключа" будем заменять на слово "ключ", имея в виду, что этот ключ является частью общего секретного ключа.

Требуется по заданным параметрам n и h определить число ключей h и дать правило распределения этих ключей между сотрудниками.

Сначала рассмотрим случай, когда n – нечетное число и $h = (n+1)/2$.

1. Мажоритарный принцип

Замечание 1. Известно, что n-разрядными равновесными кодами веса q называют двоичные n-разрядные комбинации, содержащие ровно q единиц и $n - q$ нулей. Полным равновесным кодом длины n веса q будем называть набор всех кодов, отвечающих данным условиям, и обозначать $R(n, q)$.

Замечание 2. $P(a, b)$ – обозначают количество перестановок из a объектов первого вида и b объектов второго, это число:

$$P(a,b) = \frac{(a+b)!}{a!b!} = C^{a}_{(a+b)} = C^{b}_{(a+b)}. \qquad (1)$$

Для $R(n, q)$ число комбинаций равно $P(n–q, q)$

Рассмотрим поставленную задачу в случае, когда n – нечетное число, а порог $h = (n+1)/2$.

Теорема 1. Если n – нечетное число и порог $h = (n+1)/2$, тогда количество фрагментов ключа k равно числу n разрядных двоичных кодов веса h, а именно

$$k = P\left(\frac{n+1}{2}, \frac{n-1}{2}\right) = \frac{n!}{\left((n+1)/2\right)!\left((n-1)/2\right)!}, \qquad (2)$$

а правило распределения ключей между сотрудниками соответствует столбцам полного равновесного кода $R(n, q)$.

Доказательство необходимости и достаточности приведено ниже.

Пример 1.

Пусть $n = 5$ и $h = 3$. Построим таблицу равновесных 5-разрядных кодов веса 3 , т. е. R (5, 3).

Таблица 1

	1	2	3	4	5
1)	1	1	1	0	0
2)	1	1	0	1	0
3)	1	1	0	0	1
4)	1	0	1	1	0
5)	1	0	1	0	1
6)	1	0	0	1	1
7)	0	1	1	1	0
8)	0	1	1	0	1
9)	0	1	0	1	1
10)	0	0	1	1	1

В таблице цифрами 1, 2, 3, 4, 5 обозначены члены приемной команды, 1), 2), 3), ...обозначены номера ключей. Возьмем для примера 3-го члена команды. Он имеет ключи с номерами 1), 4), 5), 7), 8), 10). Любая тройка членов приемной команды имеет полный набор ключей и может составить полный секретный ключ S. При этом никакие пары членов приемной команды не имеют полного набора. Количество ключей в данном примере равно 10. Единицы в вертикальном коде соответствуют номерам ключей, которые имеет данный член команды.

2. Принцип с произвольным порогом

В некоторых случаях может оказаться более удобным принцип, основанный на произвольном пороге h. Для этого сформулируем и докажем следующую теорему.

Теорема 2.

Если максимальное число людей, имеющих ключи, равно n (здесь уже n – любое целое положительное число) и требуется обеспечить решение при пороге h, то количество фрагментов равно числу кодовых комбинаций в $R(n, n–h + 1)$, т. е.

$$k = P\left(h-1, n-h+1\right) = \frac{n!}{\left(n-1\right)!\left(n-h+1\right)!}. \qquad (3)$$

Пример 2.

Пусть $n = 6$, $h = 2$. Построим равновесный двоичный код $R(6, 5)$

Таблица 2

	1	2	3	4	5	6
1)	1	1	1	1	1	0
2)	1	1	1	1	0	1
3)	1	1	1	0	1	1
4)	1	1	0	1	1	1
5)	1	0	1	1	1	1
6)	0	1	1	1	1	1

Дизъюнкция любых двух столбцов дает код, содержащий все единицы. Доказательство необходимости и достаточности приведено далее.

Теорема 2 является обобщением теоремы 1. Действительно, если положить в теореме 2 значение $h = (n+1)/2$, то получим

$$R(n, (n+1)/2).$$

Обе теоремы являются конструктивными, так как дают правило распределения ключей между членами принимающей команды.

Доказательство.

Возьмем i-й столбец из $R(n, q)$ и обозначим его $r(i)$. Его вес будем определять как число единиц в коде и обозначать $|r(i)|$ Определим операцию дизъюнкции столбцов $r(i)$ и $r(j)$ в виде результирующего вектора той же размерности, все компоненты которого получены путем дизъюнкции соответствующих компонентов векторов $r(i)$ и $r(j)$.Аналогично введем операцию дизъюнкции «v» над s векторами: $r(i_1)$ v $r(i_2)$ v $r(i_3)$ v...v $r(i_s)$.

Возьмем полную таблицу кодов длины n веса q, которую мы обозначали $R(n, q)$. Из этой таблицы выберем произвольный столбец с номером i, т. е. $r(i)$. Определим его вес. Легко видеть, что при любом i

$$\left|r(i)\right| = P(n-q, q-1) = C_{n-q}^{q-1}. \qquad (4)$$

Вес кода дизъюнкции любых 2 столбцов с номерами i и j равен

$$\left|r(i) \vee r(j)\right| = C_{n-1}^{q-1} + C_{n-2}^{q-1}.$$

Вес кода дизъюнкции любых s столбцов с номерами $i_1, i_2, i_3, \ldots i_s$ при $1 < s < n-q+1$ равен

$$\left|r\left(i_1\right)\mathrm{v}r\left(i_2\right)\mathrm{v}r\left(i_3\right)\mathrm{v}...\mathrm{v}r\left(i_s\right)\right| = C_{n-1}^{q-1} + C_{n-2}^{q-1} + C_{n-3}^{q-1} + ... + C_{n-1}^{q-1}. \qquad (5)$$

Если $s = n–q+1$, то сумма (5) точно равна количеству кодов в $R(n, q)$, т. е.

$$C_n^q = \frac{n!}{q!(n-q)}.$$

Действительно, последний член в (5) равен 1.

Если число членов s ряда (5) меньше $n–q+1$, то сумма (5) меньше значения (1), что доказывает теорему, так как никакие члены команды, если их меньше, чем $n–q+1$ не имеют полного набора ключей. В частном случае, если $q = (n+1)/2$, имеем, что при $s = (n+1)/2$ в совокупности имеется полный набор ключей для расшифровки сообщения.

Приведенный алгоритм распределения секретных ключей (фрагментов общего секретного ключа) достаточно прост и позволяет найти распределение секрета при любых значениях n и любых h ($1 \le h \le n$). Это является достоинством приведенного алгоритма. Этот алгоритм имеет существенный недостаток, если общий ключ просто разделять на отдельные участки: чем больше соберется сотрудников (хотя их может быть и меньше h), тем легче им будет подобрать значения недостающих фрагментов. Например, если общий ключ представлял собой 20-разрядное десятичное число и его разделить на фрагменты по 2 разряда, то когда соберутся любые 2 сотрудника, им достаточно будет подобрать значения двух недостающих разрядов, т. е. проверить всего 102 вариантов.

Секретность приведенного алгоритма можно существенно усилить, если формировать фрагменты так же, как формировались фрагменты в первом примере (при $h = n$).

Пример 3.

Пусть секретный ключ S ($S1$, $S2$, $S3$) имеет вид S = (23, 8, 11), т. е. представляет собой совокупность трех элементов, где каждый элемент – произвольное положительное целое меньшее некоторого простого числа p.

Девять из десяти фрагментов получим с помощью генератора случайных чисел, при этом совершенно необязательно, чтобы элементы находились в пределах $0 – p – 1$. Например,

1) (5, 32, 18)

2) (0, 19, 3)
3) (36, 7, 16)
4) (9, 11, 35)
5) (16, 1, 28)
6) (25, 39, 46)
7) (3, 0, 21)
8) (15, 14, 2)
9) (35, 20, 20)

А десятый фрагмент (a_{10}, b_{10}, c_{10}) получим по следующему правилу:

$a_{10} = S_1 - a_1 - a_2 - \ldots - a_9 \bmod p$

$b_{10} = S_2 - b_1 - b_2 - \ldots - b_9 \bmod p$

$c_{10} = S_3 - c_1 - c_2 - \ldots - c_9 \bmod p$

Для рассматриваемого примера получаем:

10) (24, 10, 25)

Если соберутся вместе любые три или больше сотрудников, то ключ *S* легко восстанавливается сложением соответствующих элементов всех десяти фрагментов по модулю 29. Если соберется вместе менее трех сотрудников, то на подбор ключа *S* им потребуется столько же усилий, сколько требуется любому одному сотруднику.

В общем случае при заданных n и *h* число фрагментов ключей равно

$P(h - 1, n - h + 1) = n!/((h - 1)!(n - h + 1)!)$

Криптостойкость ключа не зависит от числа сотрудников, если их меньше *h,* и ключ легко собирается, если число сотрудников равно или больше *h*.

Приведенный метод позволяет производить разделение секрета при различных степенях доверия к сотрудникам. Так, если разделяющий секрет доверяет какому-то лицу больше, чем остальным, он может выделить ему две или даже три "квоты" ключа. Порог, как и ранее можно выбирать любой (но не менее, чем число "квот", выданных наиболее доверенному сотруднику). Естественно, число сотрудников в такой схеме соответственно уменьшается.

5. НЕКОТОРЫЕ ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ, ИСПОЛЬЗУЮЩИЕ ИДЕИ С ОТКРЫТЫМ РАСПРЕДЕЛЕНИЕМ КЛЮЧЕЙ

5.1. Бросание жребия по открытому каналу

Предположим, что Алиса и Боб недавно развелись и хотят с помощью жребия решить, кому достанется, например, их общая машина.

Вопрос о принадлежности машины они хотят решить по телефону путем бросания жребия, при этом естественно стремление обоих к тому, чтобы игра была честной.

Протокол обмена сообщениями при этом может выглядеть следующим образом:

Все целые числа делятся на два класса: четные и нечетные. Пусть нечетным числам соответствует решка монеты, а четным – орел. Так, числам 1, 3, 5, 7, 9,…соответствует решка, а числам 2, 4, 6, 8, 10,… – орел. Алиса и Боб договариваются об использовании некоторой односторонней функции, например, модульном возведении в некоторую степень: $x^k \equiv a \bmod p$. Фактически они должны согласовать значения k и p. Затем Алиса выбирает некоторое целое число x , возводит его в степень k и полученное а передает Бобу. Боб, получив а, не знает, соответствует оно четному числу или нет и наугад принимает решение, например, считает, что Алиса выбрала число y, которое сообщает Алисе. Тогда Алиса передает Бобу число x, которое она использовала для вычисления a. Боб сравнивает x и y по mod 2, если $x \in y \bmod 2$, то Боб угадал и машина достается ему, если же $x \neq y \bmod 2$, то машина достается Алисе.

5.2. Определение “лишних” абонентов

Пусть секрет разделен между всеми участниками совещания по открытому каналу так, как это было сделано в примере 3 разд. 4.4. "Разделение секрета".

Руководитель опрашивает всех участников и каждый присылает свой набор фрагментов секрета. Руководитель отбирает только отличающиеся фрагменты и если их число равно R, то складывает их по модулю p, получая общий секретный ключ S, совпадающий с секретным ключом, известным ему. Если число разных фрагментов меньше R, а число участников равно или больше h, то имеются "лишние". Если число различных фрагментов больше R, то также имеются "лишние абоненты", случайно или намеренно "затесавшиеся" и желающие участвовать в обмене информацией. При наличии своих, которых меньше h, секрет S не сможет быть собран.

5.3. Идентификация пользователей по слепкам паролей

Рассмотрим ситуацию, когда требуется обратиться в банк и предъявить свой пароль. Если пароли хранятся в какой-то области памяти, то это не безопасно. Лучше всего, если они будут храниться в виде некоторых слепков, например результатов применения однонаправленных функций. Тогда даже системный администратор не сможет передать их противнику. В то же время, обращающийся клиент посылает свой истинный пароль, а компьютер вычисляет его слепок. Одной из реализаций этой схемы может быть следующая: клиент сообщает свой пароль в виде некоторого числа, например, x. ЭВМ вычисляет значение $x^n \equiv \equiv s \bmod p$ и в таблице слепков паролей находит, что приславший сообщение действительно является клиентом (или не является им). Для того, чтобы исключить угадывание паролей типа "герой", "сокол", "ястреб", можно делать к паролю случайную добавку.

5.4. Обеспечение дополнительной секретности

Недостатком описанного выше алгоритма является то, что при наличии подслушивающего противника паролем можно воспользоваться только один раз. А далее противник может замаскироваться под "своего". Целесообразно сделать так, чтобы пароль никогда не передавался по открытому каналу и даже на центральном пункте о нем не было бы ничего известно. Такой алгоритм может быть реализован следующим образом. Каждый пользователь выбирает свою схему шифрования и расшифрования (например, модульное возведение в степень), вычисляет два ключа t и d, открытый ключ t сообщает на центральный пункт, а секретный d оставляет у себя. При обращении клиента на центральный

пункт , ему посылается случайный запрос x, зашифрованный открытым ключом t пользователя. Пользователь, получив этот запрос, расшифровывает его своим секретным ключом d и посылает расшифрованный запрос на центральный пункт, по которому там могут судить, что обратился “свой”.

ЗАКЛЮЧЕНИЕ

Прошло более 50 лет со времени выхода в свет работ великого ученого современности Клода Шеннона [10], посвященных теории связи и криптографии. В этих работах дан анализ математических структур различных криптосистем и хотя во времена К.Шеннона не известны были криптосистемы с открытым ключом, его работы не потеряли актуальности в наше время. Им доказан ряд теорем, обеспечивающих построение систем с совершенной и близкой к совершенной секретностью, в которых число различных ключей приближается к числу передаваемых сообщений, а размерности сообщений и ключей имеют один порядок. Однако наиболее интересным представляется положение К. Шеннона, что секретность системы должна соответствовать задачам защиты информации. Так, секретность системы в соответствии с теоремой К. Шеннона должна быть такова, чтобы на вскрытие шифра потребовалось бы время при использовании самых современных вычислительных мощностей большее, чем время, за которое информация устареет. Таким образом, еще 50 лет назад К. Шеннон предвидел, что вопросы защиты информации в конечном счете будут интересовать очень широкий круг людей и перед ними встанет выбор: какую из криптосистем использовать в зависимости от важности передаваемой информации, времени ее жизни и цены на создание и эксплуатацию системы.

Библиографический список

1. *Виноградов И. М.* Основы теории чисел. М.: Наука, 1965. 172 с.

2. *Arto Salomaα.*Public-Key Criptography:. Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, Barselona Springer-Verlag. 1990.(Пер. с англ. *Арто Саломаа*. Криптография с открытым ключом. М.: Мир, 1995. 364 с.

3. *Ерош И.Л.* Основы теории конечных групп. Учеб. пособие / СПбГУАП, 1998. 38 с.

4. 1.*Cram*er *R*. *at a*ll. A Secure and Optimally Efficient Multi-Authority Election Scheme. Proceedinds of EUROCRYPT'97. Vol.8, No. 5. September-October 1997.

5. *Нечаев В. И.* Элементы криптографии. Основы теории защиты информации. М.: Высшая школа, 1999. 110 с.

6. *Молдовян А. А., Молдовян Н. А., Советов Б. Я.* Криптография. Санкт-Петербург: Лань. 2000. 218 с.

7. *Жельников В.*Криптография от папирусов до компьютеров. М.: ABF., 1997. 334 с.

8. *Ященко В. В.* и др. Введение в криптографию. М.: МЦНМО – ЧеРо, 1999. 272 с.

9. *Ерош И. Л.* Система передачи данных с закрытыми ключами // Информационные системы в экономике и промышленности / Под ред. И. Л Ероша. СПбГУАП. СПб., 1999. С. 114–117.

10. *Шеннон К*. Работы по теории информации и кибернетике. М.: ИЛ, 1963.

Содержание

Учебное издание

Ерош Игорь Львович

ДИСКРЕТНАЯ МАТЕМАТИКА. Математические вопросы криптографии

Учебное пособие

Редактор *В. П. Зуева*
Компьютерная верстка *Колешко А. Н., Бардуковой Ю. С.*

Лицензия ЛР №020341 от 07.05.97. Сдано в набор 15.01.01 Подписано к печати 28.04.01 Формат 60×84 1/16. Бумага тип. №3. Печать офсетная. Усл. печ. л. 3,25. Усл. кр.-отт. 3,37. Уч. -изд. л. 3,5. Тираж 100 экз. Заказ № 172

Редакционно-издательский отдел
Сектор компьютерно-издательских технологий
Отдел оперативной полиграфии
СПбГУАП
190000, Санкт-Петербург, ул. Б. Морская, 67