Министерство образования и науки Российской Федерации
ФГБОУ ВО Ульяновский Государственный Университет

РЕФЕРАТ

к кандидатскому экзамену
по дисциплине «Философия технических наук.
История технических наук»

Направление подготовки 09.06.01 «Информатика и вычислительная техника»
направленность (профиль) подготовки 05.13.18 «Математическое моделирование, численные методы и комплексы программ (технические науки)»

Тема реферата: «История развития методов повышения производительности шифров»

Выполнил: Шлыков Д.И.
аспирант 1 года очной формы обучения кафедры «Телекоммуникационных технологий и сетей»
Подпись:_____

Научный руководитель:
Смагин А.А. зав. каф. «Телекоммуникационных технологий и сетей»
д.т.н., профессор
Подпись:_____

Проверил: Дубровский П.В.
к.т.н., доцент

Оценка: Хорошо
Дата: 7.03.19
Подпись:_____

Ульяновск 2019

Оглавление

Введение

С давних пор люди осознали, что информация имеет большую ценность и ее необходимо защищать от пристального внимания недругов и друзей. Тогда-то и возникла задача защиты этой информации от чрезмерно любопытных глаз. Древние пытались использовать для решения этой задачи самые разнообразные методы, и одним из них была тайнопись - умение составлять сообщения таким образом, чтобы его смысл был недоступен никому кроме посвященных в тайну. Есть свидетельства тому, что искусство тайнописи зародилось еще в доантичные времена и получило дальнейшее развитие в античные времена.

На протяжении всей своей многовековой истории, вплоть до совсем недавнего времени, это искусство служило немногим, в основном верхушке общества, не выходя за пределы резиденций глав государств, посольств и, конечно же, разведывательных миссий. И лишь несколько десятилетий назад все изменилось коренным образом - информация приобрела самостоятельную коммерческую ценность и стала широко распространенным, почти обычным товаром. Ее производят, хранят, транспортируют, продают и покупают, а значит - воруют и подделывают и, следовательно, ее необходимо защищать. Современное общество все в большей степени становится информационно обусловленным, успех любого вида деятельности все сильней зависит от обладания определенными сведениями и от отсутствия их у конкурентов. И чем сильней проявляется указанный эффект, тем больше потенциальные убытки от злоупотреблений в информационной сфере, и тем больше потребность в защите информации.

Среди всего спектра методов защиты данных от нежелательного доступа особое место занимают криптографические методы. В отличие от других методов, они опираются лишь на свойства самой информации и не используют свойства ее материальных носителей, особенности узлов ее обработки, передачи и хранения. Образно говоря, криптографические методы строят барьер между защищаемой информацией и реальным или потенциальным зло-

умышленником. Так уж исторически сложилось, что под криптографической защитой, в первую очередь, подразумевается шифрование данных. Раньше, когда эта операция выполнялось человеком вручную или с использованием различных приспособлений, при посольствах содержались многолюдные отделы шифровальщиков. Развитие криптографии сдерживалось проблемой реализации шифров, ведь придумать можно было все что угодно, но как это реализовать. Появление цифровых электронно-вычислительных машин, приведшее в конечном итоге к созданию мощной информационной индустрии, изменило все коренным образом и в этой сфере. С одной стороны, взломщики шифров получили в свои руки чрезвычайно мощное орудие, с другой стороны, барьер сложности реализации исчез и для создателей шифров открылись практически безграничные перспективы. Все это определило стремительный прогресс криптографии в последние десятилетия.

Параллельно с появлением и развитием криптографии росли и объемы обрабатываемой информации. С каждым годом ее становится все больше и больше в разы. Доля информации, которая должна быть надежно защищена, так же продолжает расти. Это, в свою очередь, предъявляет все более жесткие требования к пропускной способности систем обработки информации. Под обработкой может подразумеваться шифрование и расшифрование данных. Здесь могут быть применены ассиметричные шифры, симметричные блочные и поточные шифры. Они должны не только хорошо выполнять свою функцию, но и делать это достаточно быстро, не внося существенных задержек. Поэтому существует необходимость применения методов повышения производительности шифров.

История развития криптографии и методов шифрования

Наивная криптография

Для наивной криптографии (до начала XVI в.) характерно использование любых, обычно примитивных, способов запутывания противника относительно содержания передаваемых сообщений. На начальном этапе для защиты информации использовались методы кодирования и стеганографии, которые родственны, но не тождественны криптографии. Шифровальные системы сводились к использованию перестановки или замены букв на различные символы (другие буквы, знаки, рисунки, числа и т.п.). Одни и те же способы шифрования использовались повторно, ключи были короткими, использовались примитивные способы преобразования исходной информации в зашифрованное сообщение. Это позволяло, однажды установив алгоритм шифрования, быстро расшифровывать сообщения.

Уже в исторических документах древних цивилизаций — Индии, Египте, Китае, Месопотамии — имеются сведения о системах и способах составления шифрованного письма. Видимо, первые системы шифрования появились одновременно с письменностью в четвертом тысячелетии до нашей эры.

В древнеиндийских рукописях приводится более шестидесяти способов письма, среди которых есть и такие, которые можно рассматривать как криптографические. Имеется описание системы замены гласных букв согласными, и наоборот. Один из сохранившихся шифрованных текстов Месопотамии представляет собой табличку, написанную клинописью и содержащую рецепт изготовления глазури для гончарных изделий. В этом тексте использовались редко употребляемые значки, игнорировались некоторые буквы, употреблялись цифры вместо имен. В рукописях Древнего Египта шифровались религиозные тексты и медицинские рецепты. Шифрование использовалось в Библии. Некоторые фрагменты библейских текстов зашифрованы с помощью шифра, который назывался *атбаш*. Правило зашифрования состояло в замене $\boldsymbol{i}$-й буквы алфавита ($\boldsymbol{i} = 1, \boldsymbol{n}$) буквой с номером $\boldsymbol{n} - \boldsymbol{i} + \mathbf{1}$, где $\boldsymbol{n}$ — число букв алфавита. Происхождение слова *атбаш* объясняется принципом замены букв. Это слово составлено из букв Алеф, Тае, Бет и Шин, то есть первой и последней, второй и предпоследней букв древнесемитского алфавита.

Развитию криптографии способствовал переход от идеографичекого письма, основанного на использовании огромно числа иероглифов, к фонетическому письму. В древнем семитском алфавите во втором тысячелетии до

нашей эры было уже 30 знаков. Ими обозначались согласные звуки, а также некоторые гласные и слоги. Упрощение письма стимулировало развитие криптографии.

В Древней Греции криптография уже широко использовалась в разных областях деятельности, в особенности в государственной сфере. Плутарх сообщает, что жрецы, например, хранили в форме тайнописи свои прорицания. В Спарте в V — IV вв. до н. э. использовалось одно из первых шифровальных приспособлений — *Сцитала* (рис. 1).

Рисунок 1 — Сцитала

Это был жезл цилиндрической формы, на который наматывалась лента пергамента. Кроме жезла могли использоваться рукоятки мечей, кинжалов копий и т.д. Вдоль оси цилиндра на пергамент построчно записывался текст, предназначенный для передачи. После записи текста лента сматывалась с жезла и передавалась адресату, который имел точно такую же Сциталу. Ясно, что такой способ шифрования осуществлял перестановку букв сообщения. Ключом шифра служит диаметр Сциталы. Известен также и метод вскрытия такого шифра, приписываемый Аристотелю. Предлагалось заточить на конус длинный брус и, обернув вокруг него ленту, начать сдвигать ее по конусу от малого диаметра до самого большого. В том месте, где диаметр конуса совпадал с диаметром Сциталы, буквы текста сочетались в слоги и слова. После этого оставалось лишь изготовить цилиндр нужного диаметра [6].

Другим шифровальным приспособлением времен Спарты была *табличка Энея* (рис. 2). На небольшой табличке горизонтально располагался алфавит, а по ее боковым сторонам имелись выемки для наматывания нити. При зашифровании нить закреплялась у одной из сторон таблички и наматывалась на нее. На нити делались отметки (например, узелки) в местах, которые находились напротив букв данного текста. По алфавиту можно было двигаться лишь в одну сторону, то есть делать по одной отметке на каждом витке. После зашифрования нить сматывалась и передавалась адресату. Этот шифр представляет собой шифр замены букв открытого текста знаками, ко-

торые означали расстояния между отметками на нити. Ключом являлись геометрические размеры таблички и порядок расположения букв алфавита. Это был довольно надежный шифр: история не сохранила документов, подтверждающих сведения о методах его вскрытия.

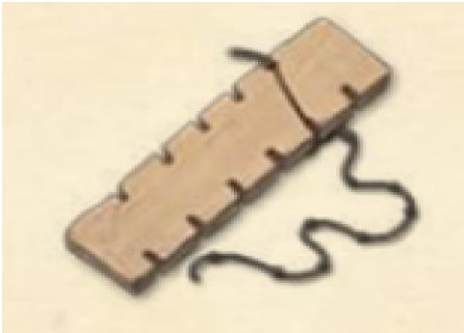


Рисунок 2 — Табличка Энея

Греческий писатель Полибий использовал систему сигнализации, которая была широко принята как метод шифрования. Он записывал буквы алфавита в квадратную таблицу и заменял их координатами: парами чисел (***i, j***), где ***i*** — номер строки, ***j*** — номер столбца. Применительно к латинскому алфавиту квадрат Полибия имеет следующий вид (рис. 3):

	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

Рисунок 3 — Квадрат Полибия

Пары (***i, j***) передавались с помощью факелов. Например, для передачи буквы О нужно было взять 3 факела в правую руку и 4 факела — в левую. Подобные шифровальные приспособления с небольшими изменениями просуществовали до эпохи военных походов Юлия Цезаря. Положение меняется в эпоху расцвета Рима, который первоначально представлял собой лишь небольшую гражданскую общину, со временем он разросся, подчинив себе сначала Италию, а затем и все Средиземноморье. Чтобы управлять наместниками в многочисленных провинциях, шифрованная связь для римских органов власти стала жизненно необходимой. Особую роль в сохранении тайны сыграл способ шифрования, предложенный Юлием Цезарем и изложенный им в «Записках о галльской войне» (I в. до н.э.). Вот что пишет о нем Гай Светоний: *«...существуют и его письма к Цицерону и письма к близким о до-*

машних делах: в них, если нужно было сообщить что-нибудь негласно, он пользовался тайнописью, то есть менял буквы так, чтобы из них не складывалось ни одного слова. Чтобы разобрать и прочитать их, нужно читать всякий раз четвертую букву вместо первой, например, О вместо А и так далее» [2]. Таким образом, Цезарь заменял буквы в соответствии с подстановкой, нижняя строка которой представляет собой алфавит открытого текста, сдвинутый циклически на три буквы влево.

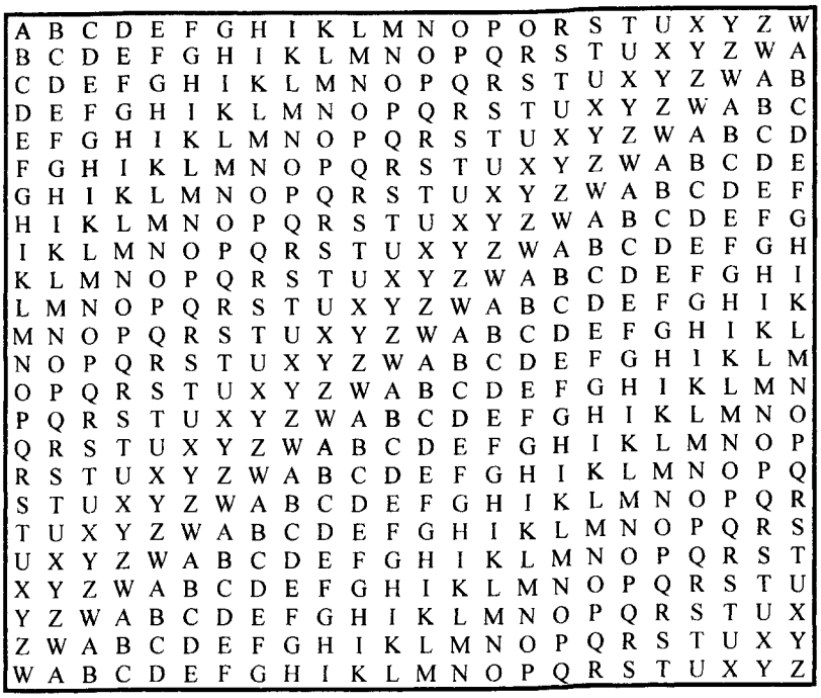
Со времен Цезаря до XV в. шифровальное дело претерпело много изменений, однако нам мало известно о методах и системах шифрования, применяемых в этот период времени. В мрачные годы средневековья практика шифрования сохранялась в строжайшей тайне. Так, в годы крестовых походов шифровальщики, служившие у Папы Римского, после года работы подлежали физическому уничтожению.

В эпоху Возрождения в итальянских городах-государствах параллельно с расцветом культуры и науки активно развивается криптография. Нередко ученые зашифровывали научные гипотезы, чтобы не прослыть еретиками и не подвергнуться преследованиям инквизиции.

Формальная криптография

Этап формальной криптографии (конец XV – начало XX вв.) связан с появлением формализованных и относительно стойких к ручному криптоанализу шифров. В европейских странах это произошло в эпоху Возрождения, когда развитие науки и торговли вызвало спрос на надежные способы защиты информации.

Богатым на новые идеи в криптографии оказался XVI в. Многоалфавитные шифры получили развитие в вышедшей в 1518 г. первой печатной книге по криптографии под названием «Полиграфия». Автором книги был один из самых знаменитых ученых того времени аббат Иоганнес Тритемий. В этой книге впервые в криптографии появляется квадратная таблица. Шифралфавиты записаны в строки таблицы один под другим, причем каждый из них сдвинут на одну позицию влево по сравнению с предыдущим (рис. 5).



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

Рисунок 5 — Таблица Тритемия

Тритемий предлагал использовать эту таблицу для многоалфавитного зашифрования самым простым из возможных способов: первая буква текста шифруется первым алфавитом, вторая буква — вторым и т. д. В этой таблице не было отдельного алфавита открытого текста, для этой цели служил алфавит первой строки. Таким образом, открытый текст, начинающийся со слов **HUNC CAVETO VITUM**, приобретал вид **HXPF GFBMCZ FUEIB.**

Преимущество этого метода шифрования по сравнению с методом Альберти состоит в том, что с каждой буквой задействуется новый алфавит. Альберти менял алфавиты лишь после трех или четырех слов. Поэтому его шифртекст состоял из отрезков, каждый из которых обладал закономерностями открытого текста, которые помогали вскрыть криптограмму. Побуквенное зашифрование не дает такого преимущества. *Шифр Тритемия* является также первым нетривиальным примером *периодического шифра*. Так называется многоалфавитный шифр, правило зашифрования которого состоит в использовании периодически повторяющейся последовательности простых замен.

В 1553 г. Джованни Баттиста Белазо предложил использовать для многоалфавитного шифра буквенный, легко запоминаемый ключ, который он назвал *паролем*. Паролем могло служить слово или фраза. Пароль периодически записывался над открытым текстом. Буква пароля, расположенная над буквой текста, указывала на алфавит таблицы, который использовался для зашифрования этой буквы. Например, это мог быть алфавит из таблицы Три-

темия, первой буквой которого являлась буква пароля. Однако Белазо, как и Тритемий, использовал в качестве шифралфавитов обычные алфавиты.

Воскресить смешанные алфавиты, которые применял Альберти, и объединить идеи Альберти с идеями Тритемия и Белазо в современную концепцию многоалфавитной замены выпало на долю итальянца Джованни де ла Порта. Ему было 28 лет, когда он в 1563 г. опубликовал книгу «О тайной переписке». По сути, эта книга являлась учебником по криптографии, содержащим криптографические познания того времени. Порта предложил использовать квадратную таблицу с периодически сдвигаемым смешанным алфавитом и паролем. Он советовал выбирать длинный ключ. Впервые им был предложен *шифр простой биграммной замены*, в котором пары букв представлялись одним специальным графическим символом. Они заполняли квадратную таблицу размером 20 x 20, строки и столбцы которой занумерованы буквами алфавита.

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

Например, биграмма **EA** заменялась символом Δ, биграмма **LF** — символом ■ и т. д. В своей книге Порта ввел многоалфавитный шифр, определяемый таблицей на рисунке 6:

A	a	b	c	d	e	f	g	h	i	k	l	m
B	n	o	p	q	r	s	t	u	x	y	z	w
C	a	b	c	d	e	f	g	h	i	k	l	m
D	o	p	q	r	s	t	u	x	y	z	w	n
E	a	b	c	d	e	f	g	h	i	k	l	m
F	p	q	r	s	t	u	x	y	z	w	n	o
G	a	b	c	d	e	f	g	h	i	k	l	m
H	q	r	s	t	u	x	y	z	w	n	o	p
I	a	b	c	d	e	f	g	h	i	k	l	m
K	r	s	t	u	x	y	z	w	n	o	p	q
L	a	b	c	d	e	f	g	h	i	k	l	m
M	s	t	u	x	y	z	w	n	o	p	q	r
N	a	b	c	d	e	f	g	h	i	k	l	m
O	t	u	x	y	z	w	n	o	p	q	r	s
P	a	b	c	d	e	f	g	h	i	k	l	m
Q	u	x	y	z	w	n	o	p	q	r	s	t
R	a	b	c	d	e	f	g	h	i	k	l	m
S	x	y	z	w	n	o	p	q	r	s	t	u
T	a	b	c	d	e	f	g	h	i	k	l	m
U	y	z	w	n	o	p	q	r	s	t	u	x
X	a	b	c	d	e	f	g	h	i	k	l	m
Y	z	w	n	o	p	q	r	s	t	u	x	y
Z	a	b	c	d	e	f	g	h	i	k	l	m
W	w	n	o	p	q	r	s	t	u	x	y	z

Рисунок 6 — Таблица Порта

Шифрование осуществляется при помощи лозунга, который пишется над открытым текстом. Буква лозунга определяет алфавит (заглавные буквы первого столбца), расположенная под ней буква открытого текста ищется в верхнем или нижнем полуалфавите и заменяется соответствующей ей буквой второго полуалфавита. Например, фраза, начинающаяся словами **HUNC CAVETO VIRUM**, будет зашифрована при помощи лозунга **DE LA PORTA** в **XFHP YTMOGA FQEAS**.

Еще одно важное усовершенствование многоалфавитных систем, состоящее в идее использования в качестве ключа текста самого сообщения или же шифрованного текста, принадлежит Джероламо Кардано и Блезу де Виженеру. Такой шифр был назван *самоключом*. В книге Виженера «Трактат о шифрах» самоключ представлен следующим образом. В простейшем случае за основу бралась таблица Тритемия с добавленными к ней в качестве первой строки и первого столбца алфавитами в их естественном порядке. Позже такая таблица стала называться *таблицей Виженера*. Подчеркнем, что в общем случае таблица Виженера состоит из циклически сдвигаемых алфавитов, причем первая строка может быть произвольным смешанным алфавитом (рис. 7).

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

Рисунок 7 — Таблица Виженера

Первая строка служит алфавитом открытого текста, а первый столбец — алфавитом ключа. Для зашифрования открытого сообщения ($T_0 = t_1t_2...$) Виженер предлагал в качестве *ключевой последовательности* (Г) использо-

вать само сообщение (T_0) с добавленной к нему в качестве первой буквы (t_0), известной отправителю и получателю (этим идея Виженера отличалась от идеи Кардано, у которого не было начальной буквы и система которого не обеспечивала однозначности расшифрования). Последовательности букв подписывались друг под другом:

$$\Gamma = t_0\ t_1\ t_2\ \ldots t_{i-1}\ \ldots$$
$$\underline{T_0 = t_1\ t_2\ t_3\ \ldots t_i\ \ldots}$$
$$T_ш = s_1\ s_2\ s_3\ \ldots s_i\ \ldots$$

При этом пара букв, стоящих друг под другом в Γ и T_0, указывала, соответственно, номера строк и столбцов таблицы, на пресечении которых находится знак s_i, шифрованного текста ($T_ш$). Например, фраза **HUNC CAVETO VIRUM**, использованная в предыдущих примерах, и начальная буква ***P*** дают

Шифртекст **YCHP ECUWZH IDAMG.**

Во втором варианте Виженер предлагал в качестве ключевой последовательности использовать шифрованный текст:

$$\Gamma = s_0\ s_1\ s_2\ \ldots s_{i-1}\ \ldots$$
$$\underline{T_0 = t_1\ t_2\ t_3\ \ldots t_i\ \ldots}$$
$$T_ш = s_1\ s_2\ s_3\ \ldots s_i\ \ldots$$

Самоключ Виженера был незаслуженно забыт на долгое время, а под шифром Виженера до сих пор понимают самый простой вариант с коротким ключевым словом и с таблицей, состоящей из обычных алфавитов.

Математическая криптография

Много новых идей в криптографии принес XIX в. Изобретение в середине XIX в. телеграфа и других технических видов связи дало новый толчок развитию криптографии. Информация передавалась в виде токовых и бестоковых посылок, то есть представлялась в двоичном виде. Поэтому возникла проблема «рационального» представления информации, которая решалась с помощью кодов. Коды позволяли передать длинное слово или целую фразу двумя-тремя знаками. Появилась потребность в высокоскоростных способах шифрования и в корректирующих кодах, необходимых в связи с неизбежными ошибками при передаче сообщений.

XX в. «прославился» двумя мировыми войнами. Эти войны оставили свой отпечаток на всех процессах, происходивших в человеческом обществе.

Они не могли не сказаться и на развитии криптографии. В период первой мировой войны в качестве полевых шифров широко использовались ручные шифры, в первую очередь шифры перестановки с различными усложнениями.

Первая мировая война явилась поворотным пунктом в истории криптографии: если до войны криптография представляла собой достаточно узкую область, то после войны она стала широким полем деятельности. Причина этого состояла в необычайном росте объема шифрпереписки и необходимости высокоскоростных способов шифрования, передаваемой информации по различным каналам связи. Криптоанализ стал важнейшим элементом разведки.

Прогресс этой области криптографии характеризовался и изменениями в самом криптоанализе. Эта наука переросла методы индивидуальной работы криптоаналитика над криптограммой. Системы секретной связи перестали быть настолько малочисленными и однородными, что один специалист мог овладеть всеми специализациями. Характер используемых шифров потребовал для их вскрытия скрупулезного анализа переписки, поиска ситуаций, благоприятствующих успешному криптоанализу, знания соответствующей обстановки.

Кроме того, криптоанализ обогатился большим опытом использования в годы войны ошибок неопытных или ленивых шифровальщиков. Еще Ф. Бэкон писал, что «в результате неловкости и неискусности тех рук, через которые проходят величайшие секреты, эти секреты во многих случаях оказывались обеспеченными слабейшими шифрами» [2]. Этот печальный опыт привел к необходимости введения строгой дисциплины среди шифровальщиков.

Несмотря на указанные последствия, первая мировая война не породила никаких новых научных идей в криптографии. Наоборот, полностью исчерпали свои возможности ручное шифрование, с одной стороны, и техническая сторона криптоанализа, состоявшая в подсчете частот встречаемости знаков, с другой.

Почти половина XX в. была связана с использованием колесных шифраторов. Различные их конструкции были запатентованы примерно в одно и то же время (в период 1917 — 1919 гг.) в разных странах: американцем Э. Х. Хеберном, голландцем Х. Ф. Кохом, немцем А. Шербиусом и шведом А. Г. Даммом.

Чертежи своей схемы на основе шифрующего диска Хеберн представил в 1917 г., и уже в следующем году был построен первый дисковый аппарат, получивший одобрение ВМС США. В 1921 г. Хеберн основал первую в США компанию по производству шифрмашин, которую через десять лет ждал бесславный конец, связанный с финансовыми трудностями.

Что представлял собой шифрующий диск? Корпус диска (имевшего размеры хоккейной шайбы) состоял из изоляционного материала, например твердой резины. По окружностям каждой из его сторон были вмонтированы на равном расстоянии друг от друга 26 электрических контактов (см. рис. 8). Каждый контакт был соединен внутри корпуса с некоторым контактом на другой стороне. Контакты на входной стороне представляли буквы открытого текста, контакты на выходной стороне — буквы шифртекста.

Диск устанавливался на оси между двумя неподвижными пластинами (розетками), каждая из которых также была изготовлена из изолятора и имела 26 контактов, соответствующих расположению контактов на диске. Контакты входной розетки соединялись с клавиатурой пишущей машинки, печатающей буквы открытого текста. Контакты выходной розетки соединялись с выходным устройством, указывающим буквы шифртекста, например, с помощью лампочек. При фиксированном угловом положении диска электрические цепи, соединяющие входные и выходные контакты, реализовывали одноалфавитную замену. При повороте же диска (на углы $\boldsymbol{2\pi k/26}$) схема реализовывала многоалфавитную замену (с 26 простыми заменами).

Рядом с одним диском можно было установить и другие диски. Тем самым схема токопрохождения удлинялась и число возможных простых замен, реализуемых многодисковой схемой, значительно возрастало. При движении $\boldsymbol{k}$ дисков по простейшей схеме одометра получался период, равный $26^{\boldsymbol{k}}$, который можно было сделать астрономическим числом.

Подобные шифрмашины обслуживали значительную часть линий связи высшего командования ВМС США, начиная с 20-х годов и, они значительно ускоряли процессы шифрования и расшифрования информации.

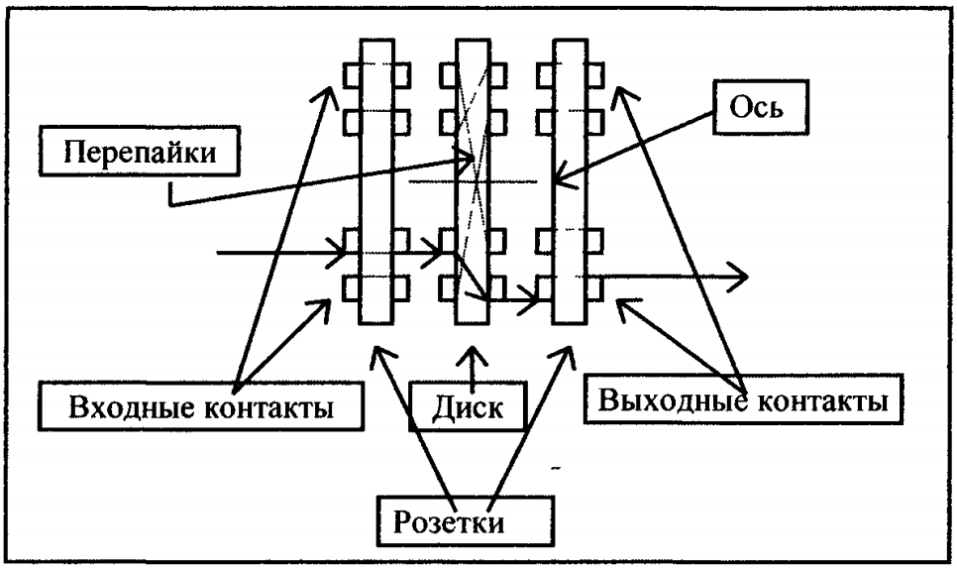


Рисунок 8 — Шифрующие диски

Х. Ф. Кох предлагал конструкцию шифрующего диска, в котором роль электричества выполняла пневматика. Речь идет о каналах, соединяющих входные и выходные контакты, по которым может проходить поток воздуха, водная или масляная струя и т. п. Любопытно, что подобные дисковые системы на основе пневматики были реально изготовлены и использовались на практике.

Принцип шифрующего диска использовали и шифрмашины, разработанные А. Шербиусом. Самой знаменитой из них была «*Энигма*» (рис. 9), которая в двух отношениях отличалась от других дисковых машин. Во-первых, после блока дисков была расположена неподвижная *обратимая розетка*, контакты которой были попарно соединены друг с другом. Импульс тока, приходивший на этот контакт, заворачивался и вновь проходил через блок дисков в противоположном направлении. Это давало двойное шифрование каждой буквы. Другая особенность «Энигмы» заключалась в неравномерном движении дисков, которое управлялось зубчатыми колесами.

Рисунок 9 — Энигма

С «Энигмой» теснейшим образом связан ход многих событий периода второй мировой войны. Дело в том, что она являлась источником ценнейших сведений для английских спецслужб, читавших переписку «Энигмы» (в рамках операции «Ультра»). Эта информация стоила так дорого, что У. Черчилль пожертвовал городом Ковентри, когда ему стал известен план германской бомбардировки этого английского города. С «Энигмой» связано также появление первой в истории вычислительной машины, сконструированной в 1942 г. для перебора ключевых элементов группой специалистов криптографов под руководством известного математика А. Тьюринга.

В дальнейшем ключевой вехой в развитии криптографии является фундаментальный труд Клода Шеннона «Теория связи в секретных системах» (англ. Communication Theory of Secrecy Systems) - секретный доклад, представленный автором в 1945 г., и опубликованный им в «Bell System Technical Journal» в 1949 г. В этой работе, по мнению многих современных криптографов, был впервые показан подход к криптографии в целом как к математической науке.

В 1960-х годах начали появляться различные блочные шифры, которые обладали большей криптостойкостью по сравнению с результатом работы роторных машин. Однако они предполагали обязательное использование цифровых электронных устройств – ручные или полумеханические способы шифрования уже не использовались.

В 1967 г. выходит книга Дэвида Кана «Взломщики кодов». Хотя книга не содержала сколько-нибудь новых открытий, она подробно описывала имеющиеся на тот момент результаты в области криптографии, большой исторический материал, включая успешные случаи использования криптоанализа, а также некоторые сведения, которые правительство США полагало всё ещё секретными. Но главное – книга имела заметный коммерческий успех и познакомила с криптографией десятки тысяч людей. С этого момента начали понемногу появляться работы и в открытой печати.

Примерно в это же время Хорст Фейстель переходит из Военно-воздушных сил США на работу в лабораторию корпорации IBM. Там он занимается разработкой новых методов в криптографии и разрабатывает ячейку Фейстеля, являющуюся основой многих современных шифров, в том числе шифра Lucifer, ставшего прообразом шифра DES – бывшего стандарта шифрования США, первого в мире открытого государственного стандарта на шифрование данных. На основе ячейки Фейстеля были созданы и другие блочные шифры, в том числе TEA (1994 г.), Twofish (1998 г.), IDEA (2000 г.), а также бывший (ГОСТ 28147-89) и действующий (ГОСТ 34.12-2015) российские стандарты шифрования.

В 1976 г. публикуется работа Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» (англ. «New Directions in Cryptography»). Данная работа открыла новую область в криптографии, теперь известную как криптография с открытым ключом. Также в работе содержалось описание алгоритма Диффи - Хеллмана - Меркла, позволявшего сторонам сгенерировать общий секретный ключ, используя открытый канал связи.

Хотя работа Диффи-Хеллмана создала большой теоретический задел для открытой криптографии, первой реальной криптосистемой с открытым ключом считают алгоритм RSA (названный по имени авторов - Рон Ривест (R. Rivest), Ади Шамир (A. Shamir) и Леонард Адлеман (L. Adleman)). Опубликованная в августе 1977 г., работа позволила сторонам обмениваться секретной информацией, не имея заранее выбранного секретного ключа. Стоит отметить, что и алгоритм Диффи - Хеллмана - Меркла, и RSA были впервые открыты в английских спецслужбах, но не были ни опубликованы, ни запатентованы из-за секретности.

В России для шифрования с открытым ключом стандарт отсутствует, однако для электронной цифровой подписи (органически связанной с шиф-

рованием с открытым ключом) принят ГОСТ Р 34.10-2001 (ГОСТ Р 34.10-2012), использующий криптографию на эллиптических кривых.

Создание асимметричных криптосистем подтолкнуло математиков и криптоаналитиков к изучению способов факторизации, дискретного логарифмирования, операций над эллиптическими кривыми в конечном поле и т.д.

Относительно новым методом является вероятностное шифрование. Вероятностное шифрование предложили Шафи Гольдвассер (Goldwasser) и Сильвио Микэли (Micali). Шифрование было названо «вероятностным» в связи с тем, что один и тот же исходный текст при шифровании с использованием одного и того же ключа может преобразовываться в совершенно различные шифротексты. При использовании криптосистем с открытым ключом существует возможность подбора открытого текста сопоставлением перехваченного шифротекста с результатом шифрования. Вероятностное шифрование позволяет на порядки увеличить сложность такого вида атаки.

Чарльз Беннет (Charles Bennet) и Жиль Брассард (Gilles Brassard), опираясь на работу Стивена Уиснера (Stephen Wiesner), разработали теорию квантовой криптографии, которая базируется скорее на квантовой физике, нежели на математике. Процесс отправки и приёма информации выполняется посредством объектов квантовой механики (например, при помощи электронов в электрическом токе, или фотонов в линиях волоконно-оптической связи). Основанная на принципах квантовой механики, эта система, в отличие от обычной криптографии, теоретически позволяет гарантированно защитить информацию от злоумышленника, даже если тот обладает самой современной технологией и неограниченными вычислительными мощностями. На данный момент, разрабатываются только прототипы квантовых криптосистем.

В то же время эффекты квантовой физики, возможно, смогут использоваться и для криптоанализа. Если будут построены квантовые компьютеры, то это поставит под вопрос существование современной криптографии.

Симметричные блочные шифры

Используемые в настоящее время системы шифрования делятся на два класса: блочные и поточные системы. Основной критерий такого разделения – мощность алфавита, над знаками которого производится операция шифро-

вания. Если открытый текст перед шифрованием разбивается на блоки, состоящие из нескольких знаков, то есть исходное сообщение обрабатывается блоками, то мы имеем дело с блочным шифром. Если каждый знак сообщения шифруется отдельно, то такой шифр — поточный.

Разделение шифров на поточные и блочные связано с алгоритмическими и техническими особенностями реализации шифрующих преобразований, использующими возможности существующей элементной базы (разрядность процессоров, быстродействие микросхем, объем памяти компьютера). При увеличении мощности алфавита необходимо исследовать, прежде всего, вопросы о выборе преобразований, реализуемых криптосхемой, и способе их практической реализации, влияющем на эффективность функционирования криптосхемы с точки зрения эксплуатационных характеристик.

Естественно, что точное значение мощности алфавита, начиная с которого шифр следует считать уже не поточным, а блочным, назвать нельзя. Более того, с развитием техники эта характеристика меняется в сторону увеличения. Например, в настоящее время используются 16- и 32-разрядные процессоры, а перспективная шифровальная техника проектируется уже на 64-разрядных процессорах. Поэтому при построении поточных шифров могут быть использованы алфавиты мощностью 2^{32} и 2^{64}.

Следует отметить, что переход от поточного к блочному шифрованию открывает дополнительные возможности для повышения стойкости криптографических алгоритмов. Все естественные языки обладают большой информационной избыточностью. Интегральной характеристикой избыточности служит энтропия текста. При шифровании текстов с малой энтропией имеется возможность применения методов, аналогичных методам вскрытия шифра простой замены. С увеличением мощности алфавита энтропия на один знак в «новом алфавите» также увеличивается. Таким образом, использование статистических закономерностей открытых сообщений при проведении криптографического анализа блочных шифров существенно затрудняется. Кроме того, анализ блочных шифров неразрывно связан с исследованием преобразований алфавитов большой мощности, а как правило, увеличение размеров задачи приводит к нелинейному росту трудоемкости ее решения, что также приводит к снижению эффективности известных методов криптографического анализа.

Оборотной стороной сложности анализа блочных криптосхем является трудность обоснования их криптографических качеств и получения доказуе-

мых оценок стойкости. Приходится разрабатывать методы анализа, учитывающие специфику схем блочного шифрования. К недостаткам блочных шифров следует отнести также сложность реализации преобразований алфавитов большой мощности. Однако этот недостаток удается преодолеть путем использования преобразований специального вида.

Ко всему прочему, симметричные блочные шифры являются самыми быстрыми и в плане обработки информации. Они проектируются и разрабатываются под целевые архитектуры систем, что дает много возможностей по их оптимизациям для улучшения производительности. Ниже буду рассмотрены некоторые из таких шифров.

Существующие симметричные блочные шифры

Алгоритм шифрования **Blowfish** основан в 1993 году Брюсом Шнаером. В общем случае алгоритм состоит из двух этапов — расширение ключа и шифрование/расшифрование исходных данных. Сложная схема выработки ключа сильно усложняет атаку на алгоритм, если пытаться взломать ее методом перебора, однако делает его непригодным для использования в системах, где ключ часто меняется, и на каждом ключе шифруется небольшие по объему данные. Алгоритм лучше всего подходит для систем, в которых на одном и том же ключе шифруются большие массивы данных.

Алгоритм шифрования **DES** основан в 1975 году фирмой IBM. С 1977 по 2001 г. Являлся Федеральным стандартом шифрования США. Симметричный алгоритм шифрования, в котором используется один ключ, как для получателя, так и для отправителя, то есть этот ключ используется как для расшифрования, так и для шифрования. DES имеет блоки по 64 бит и 16 цикловую структуру сети Фейстеля, для шифрования использует ключ с длиной 56 бит.

Алгоритм шифрования **Triple DES** (**3DES**) — симметричный блочный шифр, созданный в 1978 году на основе алгоритма DES, с целью устранения главного недостатка последнего — малой длины ключа (56 бит), который может быть взломан методом полного перебора. Скорость работы 3DES в 3 раза ниже, чем у DES, но криптостойкость намного выше. 3DES является простым способом устранения недостатков DES.

Алгоритм шифрования **AES** (**Rijndael**) разработан в 1997 году и на данный момент является Федеральным стандартом шифрования США. В ос-

нове этого алгоритма лежит симметричный блочный шифр, который работает с блоками данных длиной 128 бит и использует ключи длиной 128, 192 и 256 бит. Алгоритм может работать и с другими длинами блоков и ключей, но они в стандарт не вошли.

Алгоритм шифрования **ГОСТ 28147-89** («Магма») основан в 1989 году в СССР и в результате стал Федеральным стандартом шифрования Российской Федерации. В основе алгоритма лежит сеть Фейстеля. Использует 128 битный ключ шифрования и является надежным. Быстродействие достаточно низкое, но позволяет увеличить скорость работы за счет возможности изменения настроек со снижением криптостойкости.

Алгоритм шифрования из **ГОСТ Р 34.12-2015** («Кузнечик») – симметричный алгоритм блочного шифрования с размером блока 128 бит и длиной ключа 256 бит и использующий для генерации раундовых ключей сеть Фейстеля. Данный шифр утверждён (наряду с блочным шифром «Магма») в качестве стандарта в ГОСТ Р 34.12-2015 «Информационная технология. Криптографическая защита информации. Блочные шифры» приказом от 19 июня 2015 года № 749-ст.

Модели повышения производительности симметричных блочных шифров

К сожалению, единого подхода для увеличения производительности шифров нет. Это связано с тем, что для каждого шифра необходимо индивидуально рассматривать данный вопрос. Поскольку здесь рассматриваются симметричные блочные шифры, которые лучше всего поддаются оптимизациям, то далее речь пойдет именно о некоторых из них.

Модель для шифра AES

В статьях [12-16] рассматриваются три реализации для шифра AES. Первая реализация – **программная**. Такое название она получила, потому что все этапы шифрования полностью описаны на языке C++ и исполняются на CPU. Данная реализация пользуется математическими и программными оптимизациями.

Рассмотрим математическую сторону шифра AES. Каждый раунд состоит из трех различных обратимых преобразований, называемых слоями:

1) нелинейный слой реализован с помощь S-боксов, имеющих оптимальную нелинейность, и предотвращает возможность использования дифференциального, линейного и других современных методов криптоанализа;
2) линейный смешивающий слой гарантирует высокую степень взаимопроникновения символов блока для маскировки статистических связей;
3) слой сложения с ключом выполняет непосредственно шифрование.

Если записать все эти операции в виде матричных преобразований, то получится следующее:

$$\begin{bmatrix} Y_{0,j} \\ Y_{1,j} \\ Y_{2,j} \\ Y_{3,j} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} S[U_{0,j}] \\ S[U_{1,j-1}] \\ S[U_{2,j-2}] \\ S[U_{3,j-3}] \end{bmatrix} \oplus \begin{bmatrix} W_{i,0,j} \\ W_{i,1,j} \\ W_{i,2,j} \\ W_{i,3,j} \end{bmatrix},$$

где S – таблицы замены (S-боксы), U – исходное слово, W – раундовый ключ шифрования, Y – конечное зашифрованное слово [13]. Раскрывая матричное умножение, получаем:

$$Y_j = S[U_{0,j}] \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 3 \end{bmatrix} \oplus S[U_{1,j-1}] \cdot \begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \end{bmatrix} \oplus S[U_{2,j-2}] \cdot \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} \oplus S[U_{3,j-3}] \cdot \begin{bmatrix} 1 \\ 1 \\ 3 \\ 2 \end{bmatrix} \oplus W_{i,j}$$

Из предыдущего уравнения определим четыре таблицы, которые содержат в себе заранее просчитанные умножения числа на S-боксы:

$$T_0[b] = \begin{bmatrix} S[b] \cdot 2 \\ S[b] \\ S[b] \\ S[b] \cdot 3 \end{bmatrix}, \quad T_1[b] = \begin{bmatrix} S[b] \cdot 3 \\ S[b] \cdot 2 \\ S[b] \\ S[b] \end{bmatrix},$$

$$T_2[b] = \begin{bmatrix} S[b] \\ S[b] \cdot 3 \\ S[b] \cdot 2 \\ S[b] \end{bmatrix}, \quad T_3[b] = \begin{bmatrix} S[b] \\ S[b] \\ S[b] \cdot 3 \\ S[b] \cdot 2 \end{bmatrix},$$

где T – таблицы замены.

В результате мы получаем 2 таблицы S-боксов для шифрования и расшифрования, 4 таблиц T для шифрования и 4 таблицы T для расшифрования. Всего получается 10 таблиц, которые можно заранее просчитать и вшить в программный код в виде статических массивов для конкретной единицы

трансляции, в которой будут выполняться вычисления. В итоге все шифрование и расшифрование сводится к работе с побитовыми сдвигами для обращений к заготовленным массивам и совсем небольшим вычислениям.

По результатам измерений быстродействия программной реализации получился следующий график [13]:

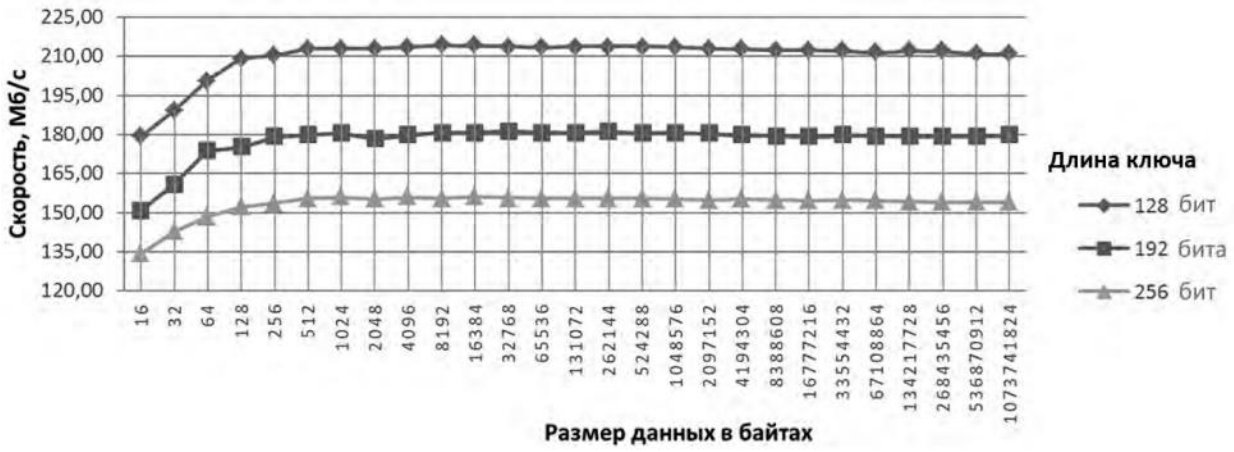


Рисунок 10 — Скоростные показатели шифрования данных размером от 16 байт до 1 Гб посредством программной реализации

Вторая реализация – **аппаратная**. Поскольку шифр AES является федеральным стандартом шифрования в США, то известные производители процессоров такие, как Intel и AMD внедрили в свои современные процессоры дополнительный набор инструкций AES-NI [1], которые позволяют очень значительно ускорить процесс шифрования и расшифрования. Данная реализация уже не использует заранее заготовленные таблицы для вычислений. Вместо этого используются специальные инструкции, которые посылают блок данных в процессор на обработку, где он выполняет один раунд шифрования или расшифрования у себя в регистрах, используя вшитые в него таблицы и операции преобразований. Данную процедуру можно повторять необходимое количество раз (в зависимости от длины ключа). Стоит заметить, что эти инструкции также позволяют выполнять операции расширения ключа, необходимые для шифрования и расшифрования, которые намного эффективнее реализаций, написанных на C++. То есть даже на этом этапе можно значительно ускорить процесс.

По результатам измерений быстродействия аппаратной реализации получился следующий график [13]:

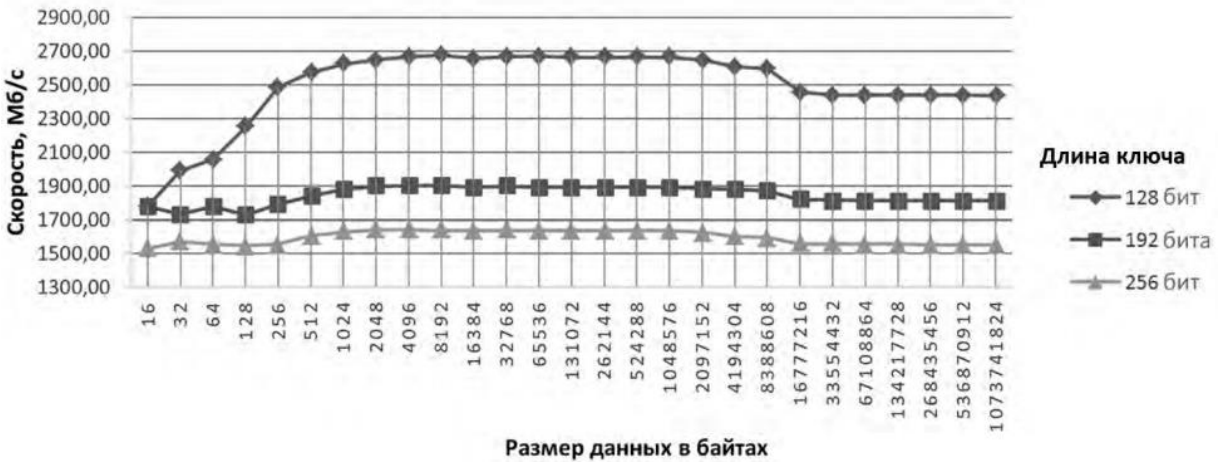


Рисунок 11 — Скоростные показатели шифрования данных объемом от 16 байт до 1 Гб посредством аппаратной реализации

По графикам видно, что аппаратная реализация достигает 2700 Мб/с, по сравнению с программной реализацией, где максимальная скорость приблизительно 210 Мб/с. Но это справедливо только для длины ключа в 128 бит, где требуется наименьшее количество раундов для шифрования или расшифрования, а следовательно, они будут самыми быстрыми.

Третья реализация – это **реализация с применением видеокарты**. Поскольку с помощью видеокарты удобно распараллеливать вычислительные процессы, то ее можно использовать для параллельного шифрования больших объемов данных, когда аппаратное шифрование недоступно. Для того, чтобы задействовать ресурсы GPU был использован язык OpenCL.

OpenCL (Open Computing Language) – открытый стандарт для универсального параллельного программирования различных типов процессоров. Стандарт предоставляет программистам переносимый и эффективный доступ ко всем возможностям гетерогенных вычислительных платформ. Для координации работы всех устройств гетерогенной системы всегда есть одно главное устройство, которое взаимодействует со всеми остальными посредством OpenCL API. Такое устройство называется «хост» и определяется вне OpenCL. Хост посылает пакеты с информацией и исполнительные команды на ускорители и получает готовые данные. Хост обрабатывает и выполняет программный код, представляющий собой тело программы, на центральном процессоре под управлением операционной системы, используя C++ или другой язык программирования. Ускорители выполняют OpenCL код, написанный на языке OpenCL C99. Существует определенный OpenCL компилятор для центрального процессора, для графического процессора и для специальных карт-ускорителей.

С использованием языка OpenCL были написаны алгоритмы шифрования и расшифрования данных, внутри которых используются эффективные табличные реализации (как и в программной реализации). Наивысшую эффективность данные алгоритмы показывают в режиме электронной кодовой книги, где каждый блок данных независимо от других шифруется и расшифровывается (рис. 12).

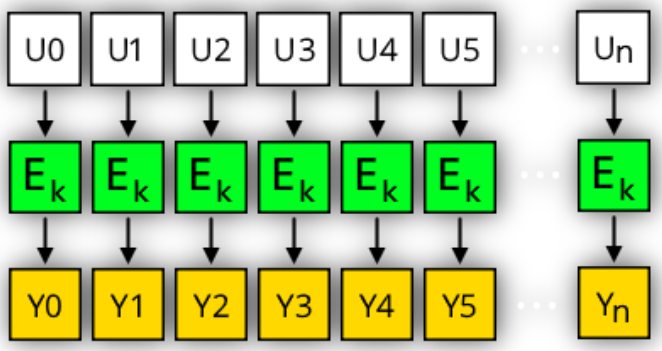


Рисунок 12 — Визуализация распараллеливания процесса шифрования: U — исходный блок, Y — отшифрованный блок, E_k — шифрование на ключе K

На CPU достичь высокой степени параллелизма достаточно сложно, из-за отсутствия большого числа ядер. На современных процессорах в среднем можно создать от 4-16 потоков, тогда как на видеокарте их число может достигать 1000 и больше.

По результатам измерений быстродействия для реализации на видеокарте получился следующий график [13]:

Рисунок 13 — Скоростные показатели шифрования данных объемом от 16 байт до 1 Гб с использованием видеокарты

Из графика на рисунке 13 видно, что скорости шифрования достигают 2000 Мб/с и более, что обгоняет программную реализацию. Но проявляется

другой эффект, связанный с размером данных. Поскольку необходимо передать данные с хоста на видеокарту, отшифровать их и потом обратно отослать на хост, происходит процесс взаимодействия оборудования на различных уровнях. Это накладывает свой отпечаток в виде «бугра» на графике. Здесь уже играют роли всякие низкоуровневые тонкости по типу пропускной способности шины, драйверов видеокарты, драйверов OpenCL, различных контроллеров между оперативной памятью, процессором и видеокартой. Из рисунка 13 видно, что наивысший пик производительности достигался при размере буфера данных в 128 мегабайт. То есть, используя данную реализацию для шифрования больших объемов данных с буфером данных в 128 Мб – можно получить достаточно хороший выигрыш в производительности по сравнению с другими реализациями.

Поскольку замеры производились на видеокарте от AMD, то она поддерживала только библиотеку OpenCL. Для видеокарт от Nvidia доступна другая библиотека – CUDA, которая полностью поддерживается данными видеокартами и разработана самой фирмой Nvidia для своих видеокарт. При помощи данной библиотеки также можно разработать алгоритмы шифрования и расшифрования с получением еще большей производительности, за счет совместимости и лучшей поддержки на аппаратном уровнем и уровне драйверов видеокарты.

Модель для шифра ГОСТ Р 34.12-2015 «Кузнечик»

Шифр принадлежит к классу LSX-шифров: его базовое преобразование (функция шифрования блока) представляется десятью циклами последов тельных преобразований L (линейное преобразование), S (подстановка) и X (смешивание с цикловым ключом) [3].

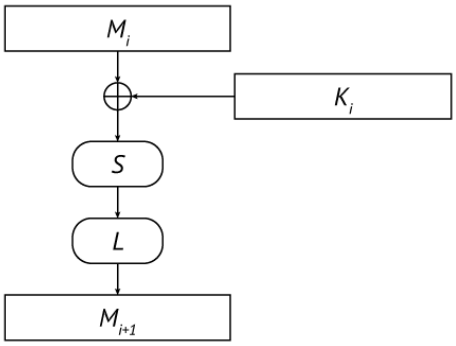


Рисунок 14 — Полный цикл базового преобразования

Команде исследователей из ОАО «ИнфоТеКС», а конкретно Михаилу Бородину и Андрею Рыбкину, удалось позаимствовать алгоритмическую оптимизацию умножения вектора на столбец у скоростных реализаций шифра AES (Rijndael), которая позволяет заменить классическую реализацию умножения за $O(n^2)$ умножений в поле на O(n) сложений по модулю два векторов длины O(n) с использованием предвычисленных таблиц, и о которой было доложено на конференции РусКрипто.

Вкратце, оптимизация заключается в следующем: допустим, в результате произведения некоторого вектора U

$$U = (u_0, u_1, \ldots, u_{n-1}) \in (\mathrm{GF}(256))^n$$

на матрицу A

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{pmatrix}_{n \times n} \in (GF(256))^{n \times n}$$

получился вектор V:

$$V = U \times A = (v_0, v_1, \ldots, v_{n-1}).$$

Традиционный способ вычисления компонент этого вектора заключается в последовательном скалярном умножении строки вектора U на столбцы матрицы A.

А можно пойти другим путем. Вместо того, чтобы вычислять каждую компоненту вектора-произведения как сумму n однобайтовых произведений, воспользуемся особенностью умножения вектора на матрицу и будем вычислять сразу все компоненты вектора как сумму n векторов:

$$V = \left(\bigoplus_j m_j a_{j,0}, \bigoplus_j m_j a_{j,1}, \ldots, \bigoplus_j m_j a_{j,n-1} \right) = \bigoplus_j m_j \left(a_{j,0}, a_{j,1}, \ldots, a_{j,n-1} \right).$$

Те же операции, но в другом порядке. Замечу, однако, что, во-первых, повысилась разрядность слагаемых с одного байта до n байт, и такие суммы можно (и нужно) вычислять в длинных регистрах, а, во-вторых, каждое слагаемое является покомпонентным произведением одного байта вектора на фиксированную строку матрицу. Например, можно заранее вычислить произведения вида:

$$m_i \times (a_{i,0}, a_{i,1}, \ldots, a_{i,n-1}) \quad \forall m_i \in \mathrm{GF}(256),$$

то есть умножить известную строку i матрицы A на все возможные значения байта i вектора U, и вместо умножения очередного байта на эту строку просто считывать произведение из таблицы. Тогда умножение вектора на матрицу сводится к считыванию n строк-произведений ***из заранее вычисленной таблицы*** и побитовое сложение этих строк для получения результирующего вектора V. Вот так, достаточно просто, можно сильно упростить умножение вектора на матрицу до O(n), если сложение векторов считать элементарными операциями.

В случае ГОСТ Р 34.12-2015 n = 16, так, вектора имеют длины в 16 байт, или 128 бит, и очень здорово помещаются в длинные XMM регистры, а для их сложения предусмотрены дополнительные процессорные инструкции, например, pxor.

Оптимизация с использованием инструкций процессора: для работы с блоками использовался набор инструкций SSE2, а именно, инструкции movdqu и movdqa для загрузки и выгрузки данных в регистры, инструкции pxor, pand, pandn для булевых операций, инструкции psrlw и psllw для побитовых сдвигов, pextrw для выгрузки байт регистра.

Кроме общеалгоритмических оптимизаций вроде описанных выше, для дальнейшего ускорения производительности нужно учитывать особенности ассемблера и особенности планировщика, который инструкции может поставить на параллельное исполнение на разных исполняющих устройствах.

Учет специфики микроархитектуры: большинство современных процессоров Intel и AMD имеют два, или более, исполняющих АЛУ, способных одновременно производить арифметические действия с различными регистрами, и планировщик, способный распределить блок выполняемых инструкций по различным АЛУ для параллельного выполнения с целью экономии процессорного времени.

Так, чередуя команды, использующие различные регистры, можно помочь процессору выполнять код параллельно. Склеенное LS-преобразование является двоичной суммой шестнадцати различных 128-битных чисел Xi, и, скорее всего, на современных процессорах может быть осуществлена в два параллельных потока (на одном ядре) с использованием двух аккумуляторов: левого и правого кэшей.

В результате описанных техник ускорений базового преобразования, позволяющих существенно повысить производительность шифра, были по-

лучены результаты для одного ядра Intel Core i7-2677M Sandy Bridge 1.80 GHz, где скорость достигла 120 МБ/с [3].

В статье [4] автор применят похожие методы для повышения производительности шифра ГОСТ 28147-89 в плане обработки данных на CPU. Он использует SSE-регистры и AVX-команды. В результате задействования различных регистров, правильной загрузки планировщика команд и использования гипертрейдинга получилось достичь скорости до 500 Мб/с.

В статье [7] автор проводит исследования оптимизации шифра ГОСТ Р 34.12–2015 «Магма» с применением технологий nVidia CUDA, OpenCL, OpenGL и обычная программа на CPU, где были получены следующие результаты:

Используемая технология	Средняя скорость обработки входящей информации, МБ/с
CPU	46.79
CUDA	365.35
OpenCL	235.13
OpenGL	176.18

По результатам измерений видно, что технология CUDA дает наивысшую производительность.

В статье [5] автор приводит пример повышения производительности ГОСТ Р 34.12-2015 «Кузнечик» на основе построения таблиц предвычислений, которые адаптированы под 64-х разрядные системы. В результате тестов на HP Pavilion dv6 Notebook PC Intel(R) Core(TM) i5 CPU M 480 2.67GHz, в системе Linux Mint 17.2 Rafaela, реализация шифрования по описанному принципу на языке Python показала скорость около 5,4 МБ/с, а на языке C – 54 МБ/с.

Иные методы повышения быстродействия

Приведенные выше примеры используют либо программные реализации, которые написаны на распространенных высокоуровневых языках программирования и исполняются на CPU или GPU, либо задействуют аппаратные ресурсы CPU, если таковые имеются, либо используют очень низкоуровневые языки ассемблера для эффективной работы со специализированными инструкциями процессора. Но можно пойти иначе. Существуют патенты на устройства, которые объединяют в себе сразу несколько алгоритмов

шифрования на выбор. Данные устройства позволяют эффективно выполнять шифрование и расшифрование данных. В некоторых из них делается упор на сокращение аппаратных затрат, за счет применения общего накопителя данных и общего накопителя ключа. Есть устройства, где реализованы различные режимы шифрования для симметричных блочных шифров, что является большим плюсом при шифровании больших файлов или потоков данных [8-11].

Заключение

Криптография и по сей день остается важнейшей наукой по защите информации. Она развивалась достаточно долго и играла важные роли в истории развития мира. Ее активно применяли во время мировых воин, где она внесла значительный вклад в победы и поражения.

С развитием этой науки развивались и шифры. Придумывались новые способы, новые методы, новая математика. В результате, на сегодняшний день мы имеем достаточно криптостойкие шифры, которые позволяют надежно шифровать данные. Помимо этого, на сегодняшний день есть множество решений проблем связанных с производительностью шифров, поскольку объем обрабатываемой информации постоянно растет и предъявляются все более жесткие требования к оборудованию и его пропускной способности. В данной области есть куда двигаться и развиваться, проводить исследования, экспериментировать с алгоритмами, новым оборудованием и функциональными возможностями, чтобы достичь наивысших результатов.

Литература

1. AES instruction set // Wikipedia URL: https://en.wikipedia.org/wiki/AES_instruction_set (дата обращения: 15.02.2019).
2. Kahn D. The codebreakers. The story of secret writing. — Macmillan, N.Y., 1967
3. ГОСТ Р 34.12 '15 на SSE2, или Не так уж и плох Кузнечик // Habr URL: https://habr.com/ru/post/312224/ (дата обращения: 03.03.2019).
4. Достигаем феноменальной скорости на примере шифрования ГОСТ 28147-89 // xakep.ru URL: https://xakep.ru/2013/10/19/shifrovanie-gost-28147-89/#toc01. (дата обращения: 15.02.2019).
5. Ищукова Е.А., Кошуцкий Р.А., Бабенко Л.К. Разработка и реализация высокоскоростного шифрования с использованием алгоритма Кузнечик // электронный научный журнал Курского государственного университета. - 2015. - №4 (08).
6. Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин, - 2 изд. - М.: Гелиос АРВ, 2002.
7. Павлов В.Э., Удальцов В.А. Оптимизация скорости работы блочных алгоритмов шифрования // Приоритетные направления развития образования и науки. - Чебоксары: Общество с ограниченной ответственностью "Центр научного сотрудничества "Интерактив плюс", 2017. - С. 76-80.
8. Патент РФ № 2006137366/08, 10.03.2005. Устройство с интегральной схемой // Патент России № 2412479. 2011. Бюл. № 5. / ЛИНДИНГЕР Андреас (DE), РОМБАХ Герхард (DE), ЛАНГЕ Роланд (DE), КИМЕС Йохен (DE), АСПЕРГЕР Карл (AT).
9. Патент РФ № 2012113897/08, 10.04.2012. Устройство шифрования данных по стандартам гост 28147-89 и AES // Патент России № 2494471. 2013. Бюл. № 27. / Архипкин В.Я., Иванов А.В., Ерохин В.В.
10. Патент РФ № 2015107429/08, 04.03.2015. Устройство шифрования данных (варианты), система на кристалле с его использованием (варианты) // Патент России № 2585988. 2016. Бюл. № 16. / Гнатюк В.Л., Осипенко П.Н., Красик К., Гурин К.Л., Хренов Г.Ю., Стариковский А.Ю., Витковский А.А., Лукьянов В.А., Шимко С.Н.

11. Патент РФ № 2017107217, 06.03.2017. Устройство шифрования данных по стандарту гост р 34.12-2015 и алгоритмам "Магма" и AES // Патент России № 2649429. 2018. Бюл. № 10.
12. Рацеев С.М., Корсунский А.С., Шлыков Д.И. О надежных и высокоскоростных симметричных шифрах // Технические науки – от теории к практике : сб. ст. по матер. LXIX междунар. науч.-практ. конф. – Новосибирск : АНС «СибАК», 2017. – № 4 (64). – С. 12–19.
13. Шлыков Д. И. О скоростной реализации шифра AES в библиотеке SDICRYPT // Автоматизация процессов управления. - 2018. - №53. - С. 34-40.
14. Шлыков Д.И. Высокоскоростная программная реализация блочного шифра AES. // Ученые записки Ульяновского государственного университета. Фундаментальные проблемы математики и механики. Ульяновск: УлГУ, 2016. С. 118-124. — ISSN 2313-2140.
15. Шлыков Д.И. Разработка высокоскоростных кроссплатформенных программных средств защиты информации на основе шифра AES // Тезисы докладов Межрегионального форума «Кибердружина». - Ульяновск: ФГБОУ ВПО «УлГПУ им. И.Н. Ульянова», 2015. С. 14.
16. Шлыков Д.И. Разработка высокоскоростных кроссплатформенных программных средств защиты информации на основе шифра AES. // ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ЗАЩИТА ПЕРСОНАЛЬНЫХ ДАННЫХ. ПРОБЛЕМЫ И ПУТИ ИХ РЕШЕНИЯ: Материалы VIII Всероссийской научно-практической конференции. Брянск: БГТУ, 2016. С. 177-180. — ISBN 978-5-89838-958-1.