Модуль 3
Защита информации с использованием шифровальных (криптографических) средств

АКАДЕМИЯ АЙТИ

www.academy.it.ru

Лекция 3.1.2
Теоретические основы криптографии

Место криптографии ….

Криптология (1935) - наука, занимающаяся исследованиями криптографических преобразований.

Криптология состоит из двух частей - ***криптография*** и ***криптоанализ***.

Криптография занимается разработкой методов криптографических преобразований информации.

Криптоанализ (1920) - занимается оценкой сильных и слабых сторон методов шифрования, а также разработкой методов, позволяющих взламывать криптосистемы.

Основные понятия и определения

Криптогра́фия (от др.-греч. *κρυπτός* - скрытый и *γράφω* - пишу, *«тайнопись»*) – наука о преобразовании информации с целью ее защиты при хранении, обработке и передаче по каналам связи, находящимся под контролем противника, а также о методах преодоления соответствующих защитных мер противника.

Криптогра́фия - наука о методах обеспечения ***конфиденциальности*** (невозможности прочтения информации посторонним), ***целостности*** данных (невозможности незаметного изменения информации), ***аутентификации*** (проверки подлинности авторства или иных свойств объекта), а также ***невозможности отказа от авторства***.

https://ru.wikipedia.org/wiki/Криптография

Основные понятия и определения

В качестве информации, подлежащей шифрованию и расшифрованию, а также электронной подписи будут рассматриваться ***тексты (сообщения)***, построенные на некотором ***алфавите***. Под этими терминами понимается следующее:

Алфавит - конечное множество используемых для кодирования информации знаков.

Текст (сообщение) - упорядоченный набор из элементов алфавита.

Основные понятия и определения

Зашифрование **-** (encryption): Обратимое преобразование данных с помощью шифра, которое формирует шифртекст из открытого текста. [ИСО/МЭК 18033–1, статья 2.18]

> Вместо термина *открытый текст* (plaintext) часто употребляются термины *«открытые данные»* или *исходный текст*, а вместо *шифртекст (шифрованный текст)* (ciphertext) *- «зашифрованные данные».*

Расшифрование **-** (decryption): Операция, обратная к зашифрованию. [ИСО/МЭК 18033-1, статья 2.13].

Под ***шифрованием*** понимается процесс ***зашифрования*** или ***расшифрования***. Также термин ***шифрование*** (в узком смысле) используется как синоним ***зашифрования***.

ГОСТ 28147-89, ГОСТ Р 34.12 - 2015

В некоторых источниках отдельно выделяют термин ***дешифрование***, подразумевая под этим восстановление исходного текста на основе шифрованного без знания ключа, то есть ***методами криптоанализа***.

Основные понятия и определения

Шифр (криптографическая система) представляет собой совокупность (семейство ***T****) обратимых преобразований открытых данных на множество всевозможных зашифрованных данных, осуществляемых по определенным правилам с применением ключей.

*Членам этого семейства можно взаимно однозначно сопоставить число ***K***, называемое ***ключом***. Преобразование T_K определяется соответствующим алгоритмом и значением ключа ***K***.

ГОСТ 28147-89

Шифр (cipher): Криптографический метод, используемый для обеспечения конфиденциальности данных, включающий алгоритм зашифрования и алгоритм расшифрования. [ИСО/МЭК 18033–1, статья 2.20]

ГОСТ Р 34.12 - 2015

Основные понятия и определения

Ключ **-** конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования, обеспечивающее выбор одного преобразования из совокупности всевозможных для данного алгоритма преобразования.

ГОСТ 28147-89

Ключ (key): Изменяемый параметр в виде последовательности символов, определяющий криптографическое преобразование. [ИСО/МЭК 18033–1, статья 2.21]

Итерационный ключ (round key): Последовательность символов, вычисляемая в процессе развертывания ключа шифра, и определяющая преобразование на одной итерации блочного шифра.

Развертывание ключа (key schedule): Вычисление итерационных ключей из ключа шифра.

ГОСТ Р 34.12 - 2015

Пространство ключей **-** набор всех возможных значений ключа.

Следует отличать понятия ***Ключ*** и ***Пароль***.
Пароль также является секретной последовательностью знаков алфавита, однако используется не для шифрования, а для аутентификации субъектов.

Основные понятия и определения

Открытый (публичный) и ***закрытый (личный) ключ*** – пара ключей, выбираемых таким образом, чтобы тогда, когда один из них (первый) применяется для зашифрования, второй можно было бы использовать для расшифрования.

Ключ подписи (signature key): Элемент секретных данных, специфичный для субъекта и используемый только данным субъектом в процессе формирования цифровой подписи. [ИСО/МЭК 14888-1:2008]

Ключ проверки подписи (verification key): Элемент данных, математически связанный с ключом подписи и используемый проверяющей стороной в процессе проверки цифровой подписи. [ИСО/МЭК 14888-1:2008]

ГОСТ Р 34.10 - 2012

Основные понятия и определения

Электронная (цифровая) подпись - присоединяемый к тексту результат его криптографического преобразования, которое позволяет при получении текста другим пользователем проверить *авторство* и *целостность* (*подлинность*) сообщения.

Электронная цифровая подпись (signature); ЭЦП: Строка бит, полученная в результате процесса формирования подписи. [ИСО/МЭК 14888-1, статья 3.12]

ГОСТ Р 34.10 - 2012

В ***ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012*** в целях сохранения терминологической преемственности по отношению к действующим отечественным нормативным документам и опубликованным научно-техническим изданиям установлено, что термины ***«электронная подпись», «цифровая подпись»*** **и** ***«электронная цифровая подпись»*** **являются синонимами**

Основные понятия и определения

Шифратор – аппаратное, программно-аппаратное или программное средство, реализующее шифр.

Противник – субъект (или физическое лицо), не знающий ключа или открытого текста и стремящийся получить его.

Подлинность – принадлежность сообщения конкретному автору и неизменность содержания сообщения.

Имитозащита - защита системы шифрованной связи от навязывания ложных данных.

Имитовставка - отрезок информации фиксированной длины, полученный по определенному правилу из открытых данных и ключа и добавленный к зашифрованным данным для обеспечения имитозащиты.

ГОСТ 28147-89

Классификация ..... методов преобразования информации

Шифр Бэкона

1	a	AAAAA
2	e	AABAA
3	i	ABAAA
4	n	ABBAA
5	r	BAAAA
6	w	BABAA
7	b	AAAAB
8	f	AABAB
9	k	ABAAB
10	o	ABBAB
11	s	BAAAB
12	x	BABAB

13	c	AAABA
14	g	AABBA
15	l	ABABA
16	p	ABBBA
17	t	BAABA
18	y	BABBA
19	d	AAABB
20	h	AABBB
21	m	ABABB
22	q	ABBBB
23	v	BAABB
24	z	BABBB

Фрэнсис Бэкон,
22.01.1561 – 09.04.1626

Телеграфный трехрегистровый код МТК-2

1	a	AAAAA
2	e	AABAA
3	i	ABAAA
4	n	ABBAA
5	r	BAAAA
6	w	BABAA
7	b	AAAAB
8	f	AABAB
9	k	ABAAB
10	o	ABBAB
11	s	BAAAB
12	x	BABAB

13	c	AAABA
14	g	AABBA
15	l	ABABA
16	p	ABBBA
17	t	BAABA
18	y	BABBA
19	d	AAABB
20	h	AABBB
21	m	ABABB
22	q	ABBBB
23	v	BAABB
24	z	BABBB

Код	Лат.	Рус.	Циф.	Код	Лат.	Рус.	Циф.
11000	A	А	-	11101	Q	Я	1
10011	B	Б	?	01010	R	Р	4
01110	C	Ц	:	10100	S	С	'
10010	D	Д		00001	T	Т	5
10000	E	Е	3	11100	U	У	7
10110	F	Ф	Э	01111	V	Ж	=
01011	G	Г	Ш	11001	W	В	2
00101	H	Х	Щ	10111	X	Ь	1
01100	I	И	8	10101	Y	Ы	6
11010	J	Й	Ю	10001	Z	З	+
11110	K	К	(	00010		CR	
01001	L	Л	)	01000		LF	
00111	M	М	.	11111		ЛАТ	
00110	N	Н	,	11011		ЦИФ	
00011	O	О	9	00100		SP	
01101	P	П	0	00000		РУС	

Был принят в СССР **в 1963 году**. Код 5-битовый (всего 32 разных комбинации), поэтому используются 3 разных регистра (русский, латинский, цифры), переключаемые управляющими символами РУС, ЛАТ, ЦИФ. Букв Ъ и Ё нет; вместо буквы Ч использовали цифру 4.

Основан на международном телеграфном коде №2 (ITA2), рекомендованном Международным консультативным комитетом по телефонии и телеграфии **в 1932 году** (в международном коде 00000 не используется).

Прим. С 1995 года этот комитет официально называется ITU-T — (англ. International Telecommunication Union - Telecommunication sector) сектор стандартизации электросвязи Международного союза электросвязи.

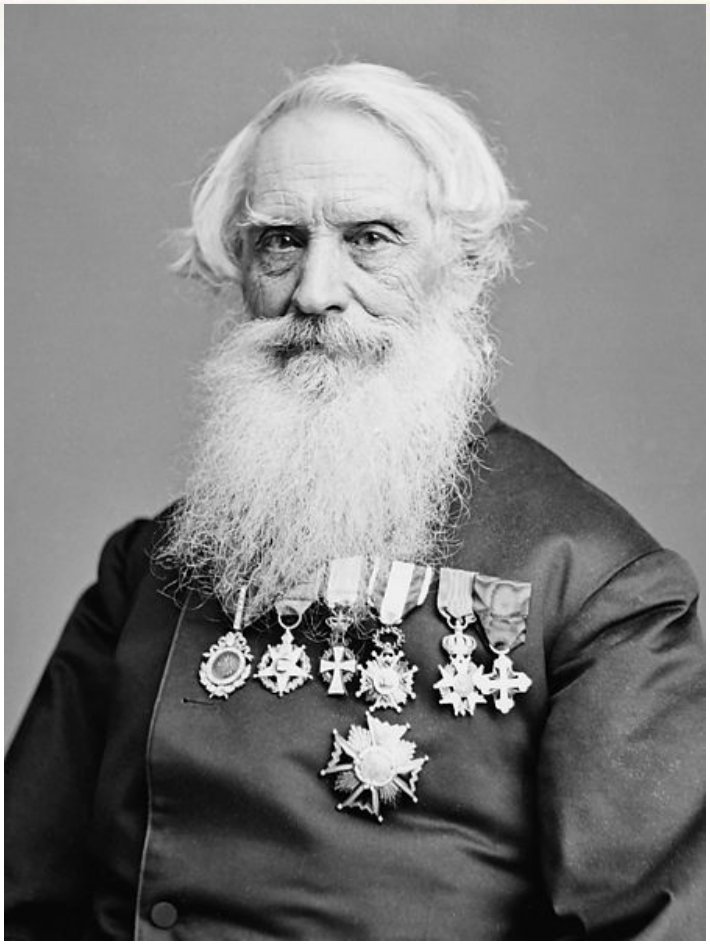
Соответствие между английским и русским регистрами, принятое в МТК-2, было использовано при создании компьютерных кодировок КОИ-7 и КОИ-8.

Азбука Морзе

INTERNATIONAL MORSE CODE

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to five dots.

A • —		U • • —	
B — • • •		V • • • —	
C — • — •		W • — —	
D — • •		X — • • —	
E •		Y — • — —	
F • • — •		Z — — • •	
G — — •			
H • • • •			
I • •			
J • — — —			
K — • —		1 • — — — —	
L • — • •		2 • • — — —	
M — —		3 • • • — —	
N — •		4 • • • • —	
O — — —		5 • • • • •	
P • — — •		6 — • • • •	
Q — — • —		7 — — • • •	
R • — •		8 — — — • •	
S • • •		9 — — — — •	
T —		0 — — — — —	



Сэмюэл Морзе,
27.04.1791 – 02.04.1872

Стеганография - наука о скрытой передаче информации путём сохранения в тайне самого факта передачи. В отличие от криптографии, которая скрывает содержимое сообщения, стеганография скрывает само его существование.

Примеры:

В V веке до н.э. тиран Гистией, находясь под надзором царя Дария в Сузах, захотел послать секретное сообщение своему зятю Аристагору в анатолийский город Милет. Он побрил наголо своего раба и вытатуировал послание на его голове. Когда волосы снова отросли, раб отправился в путь.

Геродот. История. Книга V. Терпсихора. 35.

Немного истории…

История криптографии - ровесница истории человеческого языка. Более того, первоначально письменность сама по себе была своеобразной криптографической системой, так как в древних обществах ею владели только избранные.

Священные книги древнего Египта, древней Индии тому примеры.

«История – не учительница, а назидательница, наставница жизни; она ничему не учит, а только наказывает за незнание уроков»

(В.О. Ключевский)

Немного истории…

АКАДЕМИЯ АЙТИ

История криптографии - ровесница истории человеческого языка. Более того, первоначально письменность сама по себе была своеобразной криптографической системой, так как в древних обществах ею владели только избранные.

Священные книги древнего Египта, древней Индии тому примеры.

The Jayamaṅgalā of Yaśodhara (possibly, fl. 13th century)

Немного истории…

Историю криптографии условно можно разделить на 4 этапа:

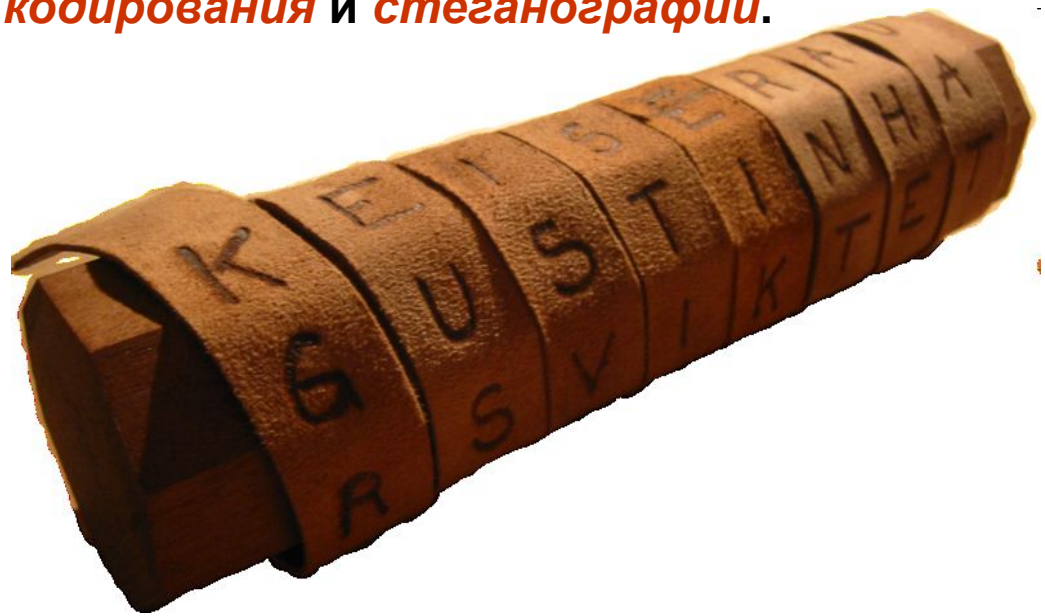
1. **Наивная криптография.** (до начала XVI века)
2. **Формальная криптография.** (конец XV века - начало XX века)
3. **Научная криптография.** (30-е - 60-е годы XX века)
4. **Компьютерная криптография.** (с 70-х годов XX века)

НАИВНАЯ КРИПТОГРАФИЯ

(до начала XVI века)

Для *наивной криптографии* характерно использование любых (обычно примитивных) способов запутывания противника относительно содержания шифруемых текстов.

На начальном этапе для защиты информации использовались методы *кодирования* и *стеганографии*.



Использовавшийся в Древней Греции шифр «скитала», известный так же как «шифр Древней Спарты» (реконструкция показана на фото), вероятно был *первым устройством для шифрования.*

Линейка Энея

Линейка Энея реализует ***шифр замены***.

Вместо диска использовались линейка с отверстиями по числу букв алфавита и прорезью и катушка с нитью.

Для ***шифрования*** нить протягивалась через прорезь и отверстие, после чего на нити завязывался очередной узел.

Для ***расшифрования*** необходимо было иметь саму ***нить*** и ***линейку с аналогичным расположением отверстий***.

Таким образом, даже зная алгоритм шифрования, но не имея ключа (линейки), прочитать сообщение было невозможно.

Аналог: *«Узелковое письмо» («кипу») получило распространение у индейцев Центральной Америки. Свои сообщения они также передавали в виде нитки, на которой завязывались разноцветные узелки, определявшие содержание сообщения.*

Система шифрования Цезаря

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

Одноалфавитная подстановка (*K = 3, m = 26*)

При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путем смещения по алфавиту от исходной буквы на ***K*** букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении ***K = 3***.

Такой шифр замены можно задать ***таблицей подстановок***, содержащей соответствующие пары букв открытого текста и шифртекста.

Например, послание Цезаря

«Пришел, Увидел, Победил»

VENI VIDI VICI

выглядело бы в зашифрованном виде

YHQL YLGL YLFL

Полибианский квадрат

При шифровании в ***полибианском квадрате*** находили очередную букву открытого текста и записывали в шифртекст букву, расположенную ниже ее в том же столбце. Если буква текста оказывалась в нижней строке таблицы, то для шифртекста брали самую верхнюю букву из того же столбца.

λ	ε	υ	ω	γ
ρ	ξ	δ	σ	ο
μ	η	β	ζ	τ
ψ	π	0	α	χ
%	ν		φ	ι

Полибианский квадрат, заполненный случайным образом 24 буквами греческого алфавита и пробелом

Полибианский квадрат

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

ДОМ

↓

15 33 31

этап ФОРМАЛЬНОЙ КРИПТОГРАФИИ

(конец XV века - начало XX века)

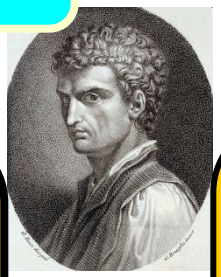
Связан с появлением формализованных и относительно стойких к ***РУЧНОМУ*** криптоанализу шифров.

В европейских странах это произошло в эпоху Возрождения, когда развитие науки и торговли вызвало спрос на надежные способы защиты информации.

Леон Батисте Альберти , итальянский архитектор

одним из первых предложил

Многоалфавитную подстановку



Работа (1466)
«Трактат о шифре»
считается первой научной работой по криптологии

Данный шифр, получивший имя дипломата XVI века *Блеза де Вижинера*, состоял в последовательном «сложении» букв исходного текста с ключом (процедуру можно облегчить с помощью специальной таблицы или диска).

Шифр Виженера

Берется небольшое целое число ***m*** и алфавит после каждой символьной подстановки сдвигается на ***m*** символов. Например, для ***m=4***. Пусть ключом будет слово ***МЫШЬ***, тогда получим:

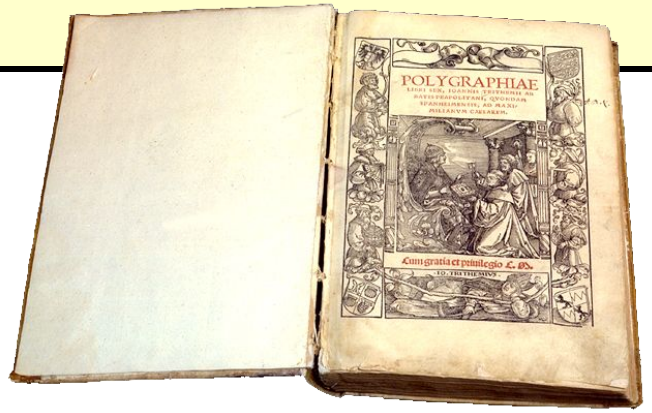
а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я	
м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л	1
ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ	2
ш щ ъ ы ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч	3
ь э ю я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы	4

Исходный текст разбивается на группы по m символов (в рассмотренном случае по 4). Для каждой группы первый символ заменяется соответствующей буквой первого алфавита, вторая – из второго и т.д. Например, фраза «***от улыбки каждый день светлей***» будет преобразована следующим образом:

- отул ыбки кажд ыйде ньсв етле й
- ынлз зьге чыяа зеьб ъчйю сндб ц

Иоганн Тритемий, немецкий аббат

«Полиграфия» (1518 г.) - это одна из первых печатных работ, в которой обобщены и сформулированы известные на тот момент алгоритмы шифрования



этап ФОРМАЛЬНОЙ КРИПТОГРАФИИ

(конец XV века - начало XX века)

АКАДЕМИЯ АЙТИ

Иоганн Тритемий, немецкий аббат

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

Ему принадлежат два небольших, но важных открытия:

1 **СПОСОБ ЗАПОЛНЕНИЯ ПОЛИБИАНСКОГО КВАДРАТА**
(первые позиции заполняются с помощью легко запоминаемого ключевого слова, остальные - оставшимися буквами алфавита)

2 **ШИФРОВАНИЕ ПАР БУКВ (биграмм)**

этап ФОРМАЛЬНОЙ КРИПТОГРАФИИ
(конец XV века - начало XX века)

В начале XIX века *сэр Чарльз Уитстон* открыл простой, но стойкий *способ многоалфавитной замены* (подстановки биграмм) - *шифр Плейфера*.
(сам шифр назван в честь Лиона Плэйфера, который сделал его применение обязательным в министерстве иностранных дел Великобритании).

Сделал важное усовершенствование - *шифрование «двойным квадратом».*

Сэр Чарльз Уитстон

Лорд Лайон Плейфер

Шифры *Плейфера* и *Уитстона* использовались вплоть до первой мировой войны, так как с трудом поддавались ручному криптоанализу.

этап ФОРМАЛЬНОЙ КРИПТОГРАФИИ

(конец XV века - начало XX века)

Голландский лингвист Огюст Керкгоффс в 1883 г **сформулировал главное требование к криптографическим системам, которое остается актуальным и поныне:**

1. Система должна быть физически, если не математически, невскрываемой
2. **Нужно, чтобы не требовалось сохранение системы в тайне; попадание системы в руки врага не должно причинять неудобств**;
3. Хранение и передача ключа должны быть осуществимы без помощи бумажных записей; корреспонденты должны располагать возможностью менять ключ по своему усмотрению
4. Система должна быть пригодной для сообщения через телеграф
5. Система должна быть легко переносимой, работа с ней не должна требовать участия нескольких лиц одновременно
6. Наконец, от системы требуется, учитывая возможные обстоятельства её применения, чтобы она была проста в использовании, не требовала значительного умственного напряжения или соблюдения большого количества правил

Секретность шифров должна быть основана на секретности ключа, но не алгоритма.

Роторные криптосистемы

Последним словом на этапе формальной криптографии, которое обеспечило еще более высокую криптостойкость, а также позволило автоматизировать (механизировать) процесс шифрования стали ***РОТОРНЫЕ криптосистемы***

Одной из первых подобных систем стала изобретенная в 1790 году ***Томасом Джефферсоном*** механическая машина. Многоалфавитная подстановка с помощью роторной машины реализуется вариацией взаимного положения вращающихся роторов, каждый из которых осуществляет «прошитую» в нем подстановку.

Практическое распространение роторные машины получили только в начале XX века. Одной из первых практически используемых машин, стала немецкая ***ENIGMA***, разработанная в 1917 году ***Эдвардом Хеберном*** и усовершенствованная ***Артуром Кирхом***.

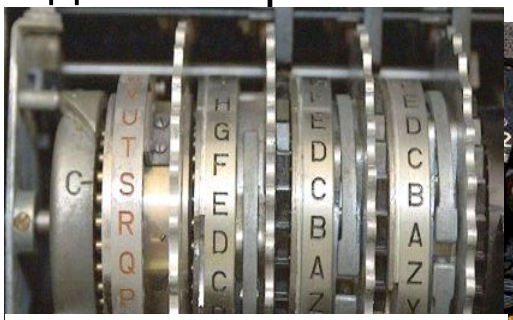
Помимо немецкой машины ***Enigma*** использовались также устройства ***Sigaba*** (США), ***Typex*** (Великобритания). ***Red***, ***Orange*** и ***Purple2*** (Япония).

Роторные системы - вершина формальной криптографии, так как относительно просто реализовывали очень стойкие шифры.

Успешные криптоатаки на роторные системы стали возможны только с ***появлением ЭВМ***.

Роторные криптосистемы

Главной деталью **роторной машины** является ротор с проволочными перемычками внутри. На каждой стороне ротора расположены равномерно по окружности ***m*** электрических контактов, где ***m*** - число знаков алфавита (в случае латинского алфавита ***m = 26***). Каждый контакт на передней стороне ротора соединен с одним из контактов на задней стороне.



Kriegsmarine M4
или Abwehr variant

В результате электрический сигнал, представляющий знак, будет переставлен в соответствии с тем, как он проходит через ротор от передней стороны к задней. В процессе работы роторы вращались.

Wehrmacht Enigma

Появление криптосистем со строгим математическим обоснованием криптостойкости.

К началу 30-х годов окончательно сформировались разделы математики, являющиеся научной основой **криптологии**: ***теория вероятностей и математическая статистика, общая алгебра, теория чисел***, начали активно развиваться ***теория алгоритмов, теория информации, кибернетика***.

Своеобразным водоразделом стала работа ***Клода Шеннона*** «***Теория связи в секретных системах***» (1949), где сформулированы теоретические принципы криптографической защиты информации.

Шеннон обосновал возможность создания сколь угодно ***стойких криптосистем.***

В 60-х годах ведущие криптографические школы подошли к созданию ***блочных шифров***, еще более стойких по сравнению с роторными криптосистемами, однако допускающие практическую реализацию ***только в виде цифровых электронных устройств.***

КОМПЬЮТЕРНАЯ КРИПТОГРАФИЯ

(с 70-х годов XX века)

Обязана своим ***появлением вычислительных средств*** с производительностью, достаточной для реализации криптосистем, обеспечивающих при большой скорости шифрования на несколько порядков более высокую криптостойкость, чем «ручные» и «механические» шифры.

Первым классом криптосистем, практическое применение которых стало возможно с появлением мощных и компактных вычислительных средств, стали ***блочные шифры***.

В 70-е годы был разработан ***американский стандарт шифрования DES*** (принят в 1978 году). Один из его авторов ***Хорст Фейстель*** (сотрудник *IBM*), описал модель блочных шифров, на основе которой были построены другие, более стойкие симметричные криптосистемы.

С появлением ***DES*** обогатился и криптоанализ, для атак на алгоритм было создано несколько ***новых видов криптоанализа*** (***линейный, дифференциальный и т.д.***), практическая реализация которых ***возможна только с применением мощных вычислительных систем.***

КОМПЬЮТЕРНАЯ КРИПТОГРАФИЯ

(с 70-х годов XX века)

В середине 70-х годов произошел настоящий прорыв в современной криптографии - ***появление асимметричных криптосистем***, которые не требовали передачи секретного ключа между сторонами.

Здесь отправной точкой принято считать работу «***Новые направления в современной криптографии***», опубликованную ***Уитфилдом Диффи*** и ***Мартином Хеллманом*** в 1976 году. В ней впервые сформулированы ***принципы обмена шифрованной информацией без обмена секретным ключом***.

Независимо к идее асимметричных криптосистем подошел ***Ральф Меркли***. Несколькими годами позже ***Рон Ривест***, ***Ади Шамир*** и ***Леонард Адлеман*** разработали систему ***RSA***, первую практическую асимметричную крипто-систему, стойкость которой была основана на проблеме факторизации больших простых чисел.

Асимметричная криптография открыла сразу несколько новых прикладных направлений, в частности: системы ***электронной цифровой подписи*** и ***электронных денег***.

Актуальной остается и задача совершенствования ***симметричных криптосистем***.

В 80-90-х годах были разработаны ***нефейстеловские шифры (SAFER, RC6*** *и др.)*.

В 2000 году после открытого международного конкурса был принят ***новый национальный стандарт шифрования*** США - ***AES***

КОМПЬЮТЕРНАЯ КРИПТОГРАФИЯ

(с 70-х годов XX века)

Современная криптография включает в себя четыре раздела:

- Симметричные криптосистемы.
- Криптосистемы с открытым ключом.
- Системы электронной подписи.
- Управление ключами.

Основные направления использования криптографических методов - **передача конфиденциальной информации по каналам связи** (например, электронная почта), **установление подлинности** передаваемых сообщений, **хранение информации** (документов, баз данных и т.п.) **на носителях** в зашифрованном виде.

Требования к криптосистемам

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно.

Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д.

Программная реализация более практична, допускает известную гибкость в использовании.

Требования к криптосистемам

Независимо от способа реализации для современных криптографических систем защиты информации сформулированы следующие *общепринятые требования:*

1. ***Знание нарушителем алгоритма шифрования не должно снижать криптостойкости шифра.***

Это фундаментальное требование было сформулировано Керкхоффом и разделяет криптосистемы общего использования (алгоритм доступен потенциальному нарушителю) и ограниченного использования (алгоритм держится в секрете). Взлом шифров в системе сотовой связи ***GSM***, стандарта ***DECT*** или в ***защите дисков DVD*** от незаконного копирования/воспроизведения - примеры последствий, к которым может привести несоблюдение этого требования.

Требования к криптосистемам

2. Зашифрованное сообщение должно поддаваться чтению только при наличии ключа.

Используемое в программе MS Word 6.0/95 «шифрование» документа на самом деле только запрещало его открытие в данной программе. Сам же текст не шифровался и был доступен для чтения в любом текстовом редакторе.

3. Шифр должен быть стойким даже в случае если нарушителю известно достаточно большое количество исходных данных и соответствующих им зашифрованных данных.

4. Число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку и должно либо выходить за пределы возможностей современных вычислительных средств *(с учетом возможности организации сетевых вычислений),* ***либо требовать создания (применения) дорогостоящих вычислительных систем.***

Требования к криптосистемам

5. Незначительное изменение ключа или исходного текста должно приводить к существенному изменению вида зашифрованного текста.

Этому требованию не соответствуют практически все шифры донаучной криптографии.

6. Структурные элементы алгоритма шифрования должны быть неизменными.

7. Длина шифрованного текста должна быть равной длине исходного текста.

8. Дополнительные биты, вводимые в сообщение в процессе шифрования, должны быть полностью и надежно скрыты в шифрованном тексте.

Требования к криптосистемам

9. Не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемыми в процессе шифрования.

10. Любой ключ из множества возможных должен обеспечивать равную криптостойкость.

В этом случае принято говорить о линейном (однородном) пространстве ключей.

Основы криптоанализа

Главным действующим лицом ***в криптоанализе*** выступает ***нарушитель*** (или ***криптоаналитик***). Под ним понимают лицо (группу лиц), целью которых является прочтение или подделка защищенных криптографическими методами сообщений.

В отношении нарушителя принимается ряд допущений, которые как правило кладутся в основу математических или иных моделей:

1. Нарушитель ***знает алгоритм шифрования*** (или выработки ЭП) и особенности его реализации в конкретном случае, но не знает секретного ключа.

2. Нарушителю ***доступны все зашифрованные тексты.*** Нарушитель ***может иметь доступ к*** некоторым ***исходным текстам***, для которых известны соответствующие им зашифрованные тексты.

3. Нарушитель ***имеет в своем распоряжении*** вычислительные, людские, временные и иные ***ресурсы***, объем которых оправдан потенциальной ценностью информации, которая будет добыта в результате криптоанализа.

Основы криптоанализа

Криптоатакой или ***атакой на шифр*** называют попытку прочтения или подделки зашифрованного сообщения, вычисления ключа методами криптоанализа.

Удачную криптоатаку называют ***взломом*** или ***вскрытием***.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к расшифрованию без знания ключа (т.е. криптоатаке).

Показатель криптостойкости - главный параметр любой криптосистемы. В качестве показателя криптостойкости можно выбрать:

- количество всех возможных ключей или вероятность подбора ключа за заданное время с заданными ресурсами;
- количество операций или время (с заданными ресурсами), необходимое для взлома шифра с заданной вероятностью;
- стоимость вычисления ключевой информации или исходного текста.

Все эти показатели должны учитывать также уровень возможной криптоатаки.

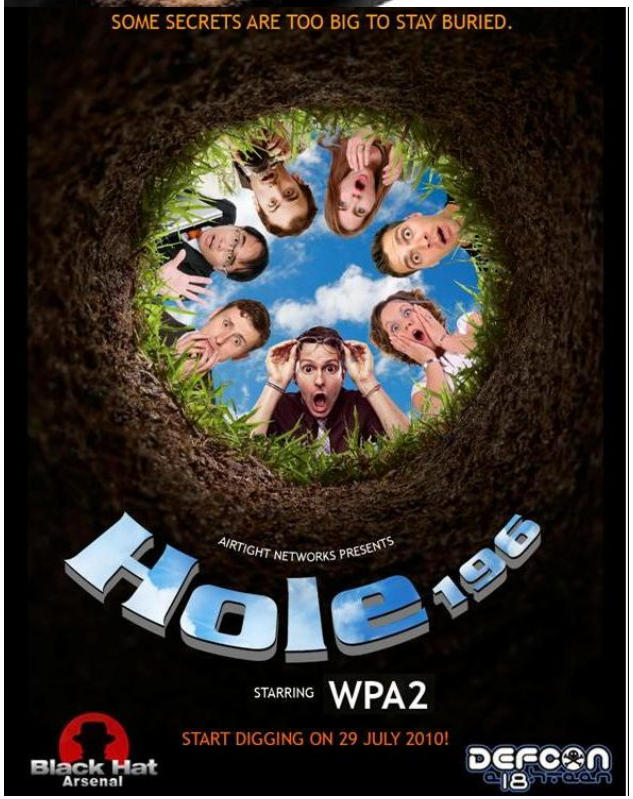
Эффективность защиты информации криптографическими методами ***зависит*** не только от криптостойкости шифра, но и ***от множества*** других ***факторов***, включая ***вопросы реализации*** криптосистем в виде устройств или программ.

При анализе криптостойкости шифра необходимо учитывать и ***человеческий фактор («атака на человека»).***

Современный криптоанализ опирается на такие математические науки как:

- ***теория вероятностей***
- ***математическая статистика***
- ***алгебра***
- ***теория чисел***
- ***теория алгоритмов и д.***

... вопросы реализации....



AirTight NETWORKS

23 июля 2010 года ***AirTight Networks*** опубликовала информацию об уязвимости ***Hole196*** в протоколе ***WPA2***.

(196-я страница IEEE 802.11 Standard (revision, 2007))

Используя эту уязвимость, авторизовавшийся в сети злоумышленник может расшифровывать данные других пользователей, используя свой закрытый ключ.

Никакого взлома ключей не требуется.

27 апреля 2011 опубликован эксплойт!!

... вопросы реализации....

RSA Security заявила о наличии АНБ-бэкдора в своих продуктах

Александр Панасенко пт, 20/09/2013 - 14:38

Компания RSA Security, один из крупнейших поставщиков средств коммерческого шифрования данных, сегодня порекомендовала клиентам не использовать функции шифрования в программном обеспечении RSA Data Protection и RSA Bsafe, так как в их компонентах содержатся бэкдоры, созданные в Агентстве национальной безопасности США.В бюллетене по безопасности RSA, разосланном сегодня компанией, говорится, что в крипто-механизме обоих продуктов содержится генератор ключей Dual EC_DRBG. Этот генератор использует механизм, ранее утвержденный институтом NIST (National Institute of Standards and Technology). На этой неделе стало известно, что NIST сертифицировал этот механизм со встроенным бэкдором АНБ. Механизм работает с генератором случайных чисел, который в реальности генерирует не случайные числа, а позволяет обладателю данных о бэкдоре предугадывать генерации.

"Чтобы гаранитровать высокий уровень безопасности наших приложений, RSA настоятельно рекомендует клиентам более не применять Dual EC_DRBG и использовать иную систему генерации случайных чисел", - говорят в компании. "Технический гид по тому, как переключиться на другой механизм, доступен в разделе документация для клиентов".

Методы криптоанализа

Все методы криптоанализа в целом укладываются в *четыре* направления:

1. **Статистический криптоанализ.** Исследует возможности взлома криптосистем на основе изучения статистических закономерностей исходных и зашифрованных сообщений. Его применение осложнено тем, что в реальных криптосистемах информация перед шифрованием подвергается сжатию (превращая исходный текст в случайную последовательность символов), или в случае гаммирования используются псевдослучайные последовательности большой длины.

2. **Алгебраический криптоанализ.** Он занимается поиском математически слабых звеньев криптоалгоритмов. Например, в 1997 году в эллиптических системах был выявлен класс ключей, которые существенно упрощали криптоанализ.

Абу Юсуф Якуб ибн Исхак ибн Саббах аль-Кинди (араб. أبو يوسف يعقوب إبن إسحاق الكندي, умер. ок. 873) - знаменитый арабский философ и учёный. В Западной Европе был известен под именем *Alkindus.*

Аль-Кинди является автором около 290 книг по метафизике, логике, этике, математике, астрономии, медицине, метеорологии, оптике, лингвистике, музыке.

Его самый знаменитый трактат - ***«Рукопись по дешифрованию криптографических сообщений»***, был обнаружен заново лишь в 1987 году в оттоманском архиве Сулайманийа в Стамбуле.

В нем содержится подробный анализ статистики, фонетики и синтаксиса арабского языка, революционная система криптоанализа аль-Кинди умещается в два коротких абзаца:

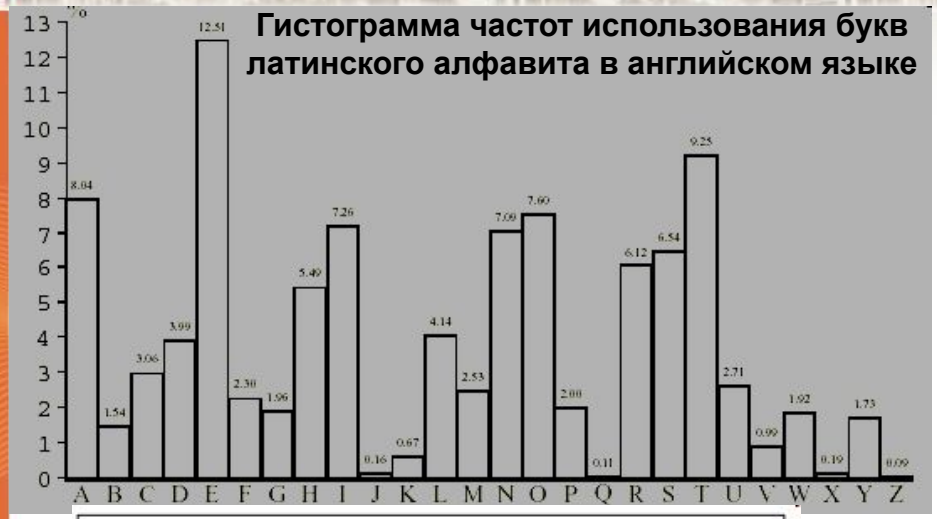
История криптоанализа
Аль-Кинди

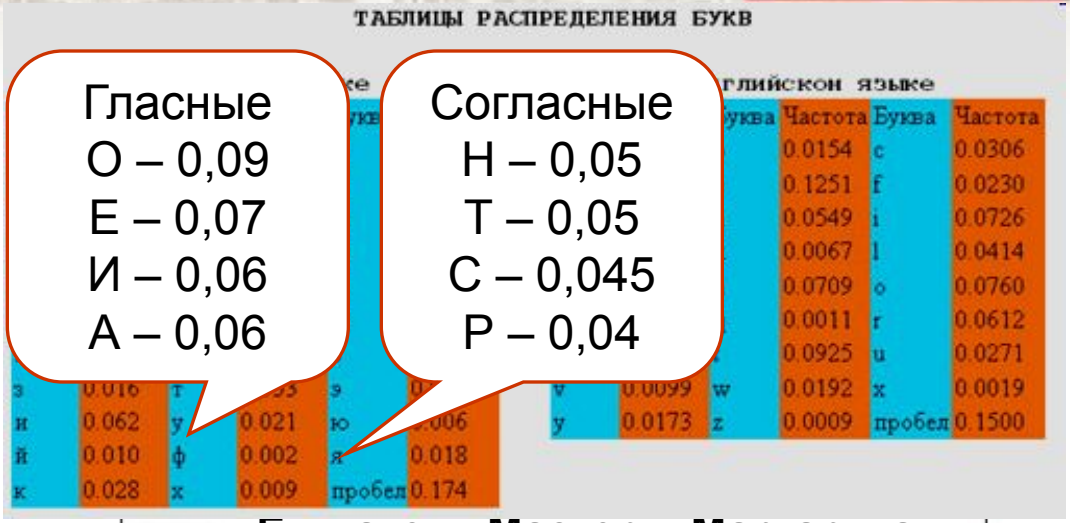
«Один из способов прочесть зашифрованное сообщение, если мы знаем язык, на котором оно написано, - это взять другой незашифрованный текст на том же языке, размером в страницу или около того, и затем подсчитать появление в нем каждой из букв. Назовем *наиболее часто встречающуюся букву* «первой», букву, которая по частоте появления стоит на втором месте, назовем «вторая», букву, которая по частоте появления стоит на третьем месте, назовем «третья» и так далее, пока не будут сочтены все различные буквы в незашифрованном тексте.

Затем посмотрим на зашифрованный текст, который мы хотим прочитать, и таким же способом проведем сортировку его символов.

Найдем *наиболее часто встречающийся символ* и заменим его «первой» буквой незашифрованного текста, второй по частоте появления символ заменим «второй» буквой, третий по частоте появления символ заменим «третьей» буквой и так далее, пока не будут заменены все символы зашифрованного сообщения, которое мы хотим дешифровать».

Статистический криптоанализ





1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

3. Дифференциальный (или разностный) криптоанализ (1990 г.). Основан на анализе зависимости изменения шифрованного текста от изменения исходного текста. Впервые использован ***Мерфи***, улучшен ***Бихэмом*** и ***Шамиром*** для атаки на ***DES***.

4. Линейный криптоанализ (1993 г.). Метод, основанный на поиске линейной аппроксимации между исходным и шифрованным текстом. Предложенный ***Мацуи***, также впервые был применен при взломе ***DES***. Как и дифференциальный анализ в реальных криптосистемах может быть применен только для анализа отдельных блоков криптопреобразований.

Основы криптоанализа

Методы криптоанализа

Классический криптоанализ:

- Частотный анализ
- Метод Касиски
- Метод индексных совпадений
- Метод взаимных индексных совпадений

Симметричные алгоритмы:

- Дифференциальный криптоанализ
- Линейный криптоанализ
- Интегральный криптоанализ
- Статистический криптоанализ
- Вычетный криптоанализ
- XSL атака
- Атака со скольжением

Асимметричные алгоритмы (алгоритм RSA):

- Решение задачи разложения числа на множители
- Решение задачи дискретного логарифма
- Метод бесключевого чтения RSA

Другие методы:

- Атака «дней рождения»
- Атака «человек посередине»
- Атака «грубой силой»
- Метод «бумеранга» (1999 г.)
- Сдвиговая атака (1999 г.)
- «Встреча посередине»
- Метод интерполяции (1997 г.)

Пошутим?..

В древнее время – методы ***точечно-почечного*** и ***точечно-печеночного*** криптоанализа,

Никто не может точно датировать, когда впервые были применены методы ***терморектального криптоанализа.***

Известно, что в 13 веке инквизиторы использовали ***похожие методики*** для определения демонов. В то время были распространены методики ***свинцового терморектального*** и ***термоорального анализа***.

Основной метод - вливание расплавленного свинца в полость.

К сожалению, как показала практика, использование термоорального криптоанализа не дает новых сведений, так как клиент после «исследования» не многое может сказать…

А вот терморектальный анализ нашел последователей и со временем получил развитие….

В Китае, например, использовали подожженную ветку бамбука…

В последствии, с появлением электричества, «исследователи» перешли на высокотехнологичные инструменты. Так и появился современный (классический) ***метод терморектального анализа*** с использованием паяльника.

В каждой шутке есть только доля шутки…

«*Терморектальный криптоанализ*» — вариант криптоанализа, предусматривающий применение методов соответствующего воздействия не к самой криптографической системе, а к человеку или группе людей, обладающих информацией или способами её получения.

http://termorect.narod.ru/archiv.html

Терморектальный криптоанализ™

современные технологии на службе вашего бизнеса

Терморектальный криптоанализ™ — молодая, бурно развивающаяся компания на рынке услуг бизнес-консалтинга и информационного менеджмента.

скрывает, криптоанализ позволяет это узнать. Традиционные методы криптоанализа малоэффективны, они требуют больших результата. Вернее, результат может быть достигнут, но через несколько тысяч лет. Современный бизнес, в котором все нужно сделать «еще вчера» не будет столько ждать. Именно поэтому был разработан перспективный метод терморектального криптоанализа, успешно применяемый на практике нашими специалистами. Он не требует сверхсложного оборудования, но удивительно эффективен в применении. Кроме того, в отличие от традиционных методов и продвинутых, но все же не гарантирующих результата методов точечно-почечного и точечно-печеночного криптоанализа, терморектальный криптоанализ имеет веское преимущество, выраженное научным языком.

На главную
Истории успеха
Технические детали
Литература
Теория
Практика
Записки специалистов
С чего все начиналось
Контакты

Основная теорема терморектального криптоанализа — время, необхо… шифрования сообщения не зависит от алгоритма шифрования и длины ключа.

Что это значит? Представьте себе пароль, состоящий из латинских символов. Достаточно до… одну(!) букву, и его вскрытие станет сложнее в 26 раз! Если ранее традиционные методы, скажем, позволял… утки, то теперь Вам потребуется почти месяц. За месяц может произойти все что угодно — дефолт, дев… ия, оранжевая революция. Разумеется, для бизнеса, который хочет быть стабильным, это неприемлемо. …рморектальный криптоанализ настолько востребован, редко когда для получения пароля любой длины требуется больше 5 минут.

Некоторые недальновидные бизнесмены считают, что терморектальный криптоанализ опасен и может иметь проблемы с законом. Однако это не так. Наша компания — честный игрок рынка и предоставляет только проверенные и надежные услуги

Готово

«Намного проще находить дефекты в людях, чем в криптосистемах»

Криптоаналитики нередко вскрывают стойкие криптосистемы используя просчеты в распределении ключей.

Зачем Нарушителю заниматься вскрытием криптографического алгоритма целиком, если он может восстановить ключ из-за неаккуратного хранения ключа?

Зачем тратить 10 000 000 долларов на создание супермашины для взлома, если достаточно подкупить клерка за 1000 долларов?

1 000 000 долларов связисту, сидящему на теплом месте в дипломатическом посольстве - превосходная сделка. ***Это намного дешевле***, чем строить огромные машины для взлома и нанимать блестящих криптоаналитиков.

Джон Энтони Уокер (John Anthony Walker, — старший уоррент-офицер 3 класса, дежурный офицер связи штаба командующего подводным флотом США в атлантическом регионе, арестован в 1985 г.) с 1968 г. передавал Советской разведке ключи шифрования ВМС США.

Начальник советского отдела управления внешней контрразведки ЦРУ ***Олдрич Хейзен Эймс*** (Aldrich Hazen Ames) стоил меньше 2 миллионов долларов, включая жену. Арестован в 1994 г.

Нарушитель может выкрасть ключи.

Можно арестовать или похитить кого-то, кто знает ключи.

Нарушитель (***и не только «она»***) может совращать кого-то и получать ключи таким образом

Морские пехотинцы, охранявшие посольство США в Москве не устояли перед подобной атакой. Они устраивали «экскурсии» в шифровальный отдел для своих русских подруг

Б. Шнайер. Прикладная криптография, 2002

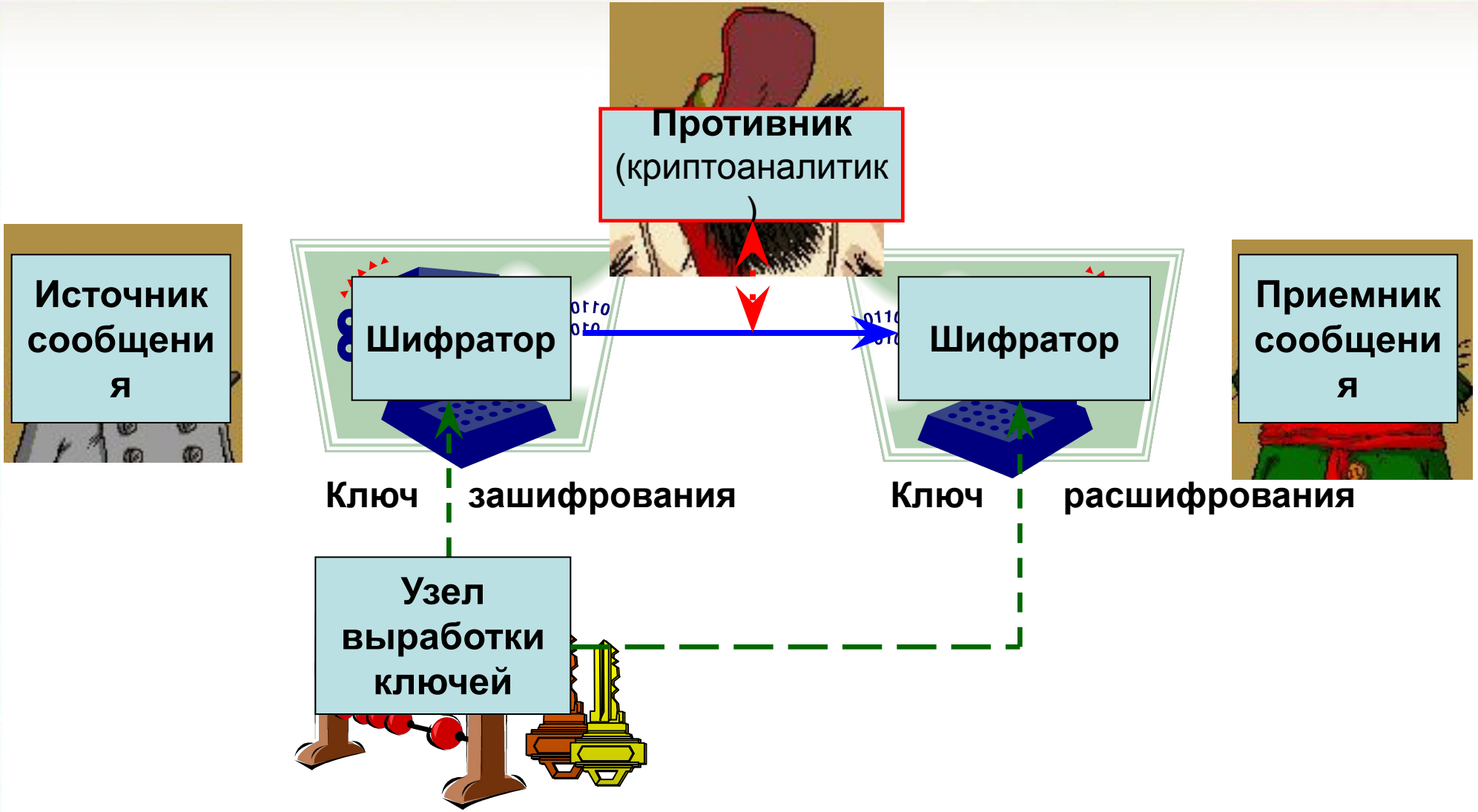
Основные задачи, решаемые в современных ИТКС с использованием СКЗИ:

- **Обеспечение конфиденциальности** - **защита содержимого** информации от лиц, не имеющих к ней доступа.
- **Обеспечение целостности** - гарантирование невозможности **несанкционированного изменения** информации.
- **Обеспечение аутентификации** – разработка и внедрение методов **подтверждения подлинности** сторон и самой информации.
- **Обеспечение невозможности отказа от авторства** - **предотвращение возможности отказа** субъектов от некоторых совершенных ими действий.

Основные трудности обеспечения информационной безопасности с помощью СКЗИ:

- средство реализации криптографического алгоритма в компьютерной системе представляет собой **равноправный** с прочими **ресурс**
- ключевая информация СКЗИ является данными компьютерной системы
- функционирование СКЗИ происходит не автономно, а выполняется **под управлением операционной системы** и различных программ-посредников
- программная среда, в которой работает СКЗИ, **устроена иерархично**
- работа СКЗИ сопряжена с возникновением **ошибочных ситуаций** в аппаратной и программной среде КС

Общая схема использования СКЗИ



Надежность традиционного (симметричного) шифрования

Для надежности традиционного шифрования необходимо выполнение следующих требований:

- **Алгоритм шифрования должен быть достаточно стойким.** Как минимум, он не должен дать возможность расшифровать весь текст или вычислить ключ, даже если у противника в распоряжении окажется несколько фрагментов шифрованного текста вместе с соответствующими им фрагментами открытого текста.

- **Отправитель и получатель должны некоторым образом получить копии секретного ключа и сохранять его в тайне.** Необходимо отметить, что надежность традиционного шифрования зависит от секретности ключа, а не от секретности алгоритма.

Надежность шифра

Стойкость шифрующего преобразования - это ***трудоемкость*** задачи нахождения параметра преобразования (ключа), либо открытого текста при тех или иных условиях.

Т.о. - ***надежность*** или ***стойкость*** шифра определяется объемом работы криптоаналитика, необходимой для его взлома.

Правило Кирхгоффа: противнику известно ***всё*** (все параметры шифра), ***кроме ключа***, использованного для шифрования данного текста.

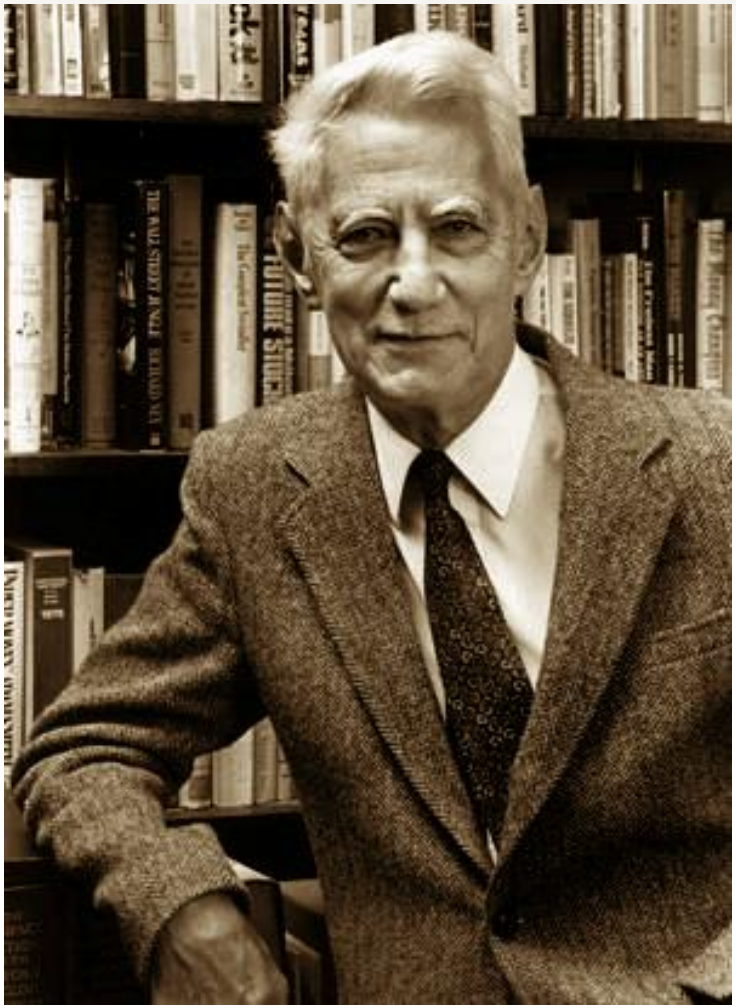
Вывод: шифр называется защищенным по вычислениям, если он удовлетворяет хотя бы одному из критериев:

- стоимость работы по дешифрованию зашифрованной информации превышает ее стоимость
- время, требуемое для дешифрования зашифрованной информации, превышает время, в течение которого эта информация является актуальной

Надежность шифра

Идеальный случай - шифр является **абсолютно стойким**, т.е. открытый текст **X** невозможно найти **никогда**.

Условия для такого шифра сформулировал ***Клод Шеннон*** - ***... чтобы противник не получил никакой информации о ключе, либо об открытом тексте необходимо и достаточно, чтобы суммарная вероятность всех ключей, переводящих открытый текст в зашифрованный текст должна быть равна вероятности получения зашифрованного текста и не должна зависеть от открытого текста...***



«Теория связи в секретных системах», 1945 г.

Надежность шифра

Следствия :

- число всевозможных ключей не должно быть меньше числа сообщений
- для каждого зашифрованного текста должен найтись ключ, который переводит любой открытый текст в данный зашифрованный текст (условие транзитивности).

Сформулированные условия являются **необходимыми**, чтобы шифр был **абсолютно стойким** ***(по Шеннону)***.

Т.е, строгое выполнение условий Шеннона возможно только ***для шифров с длиной ключа, равной длине передаваемого текста.***

Классификация криптографических алгоритмов

Подстановки (замены)

- **Моноалфавитные** (шифр простой замены). Заключаются в замене символов исходного сообщения на другие (того же алфавита) по более или менее сложному правилу. ***Пример – шифр Цезаря.***
- **Многоалфавитные**. В отличие от моноалфавитных закон изменения символов отличается от символа к символу ***Пример – шифр Виженера.***
- Вместо замены символа может происходить замена группы символов.
  Пример – шифр Плейфера

Перестановки

Заключаются в перестановке местами символов исходного текста по некоторому правилу:

Пример 1 – переписать символы исходного сообщения сзади на перёд полностью:

ПРИВЕТ МЕДВЕД → ДЕВДЕМ ТЕВИРП

Перестановки

Заключаются в перестановке местами символов исходного текста по некоторому правилу:

Пример 1 – переписать символы исходного сообщения сзади на перёд полностью:

ПРИВЕТ МЕДВЕД → ДЕВДЕМ ТЕВИРП

Пример 2 – переписать символы исходного сообщения сзади на перёд в группах по три знака, сохраняя общую последовательность:

ПРИВЕТ МЕДВЕД → ИРПТЕВ ДЕМДЕВ

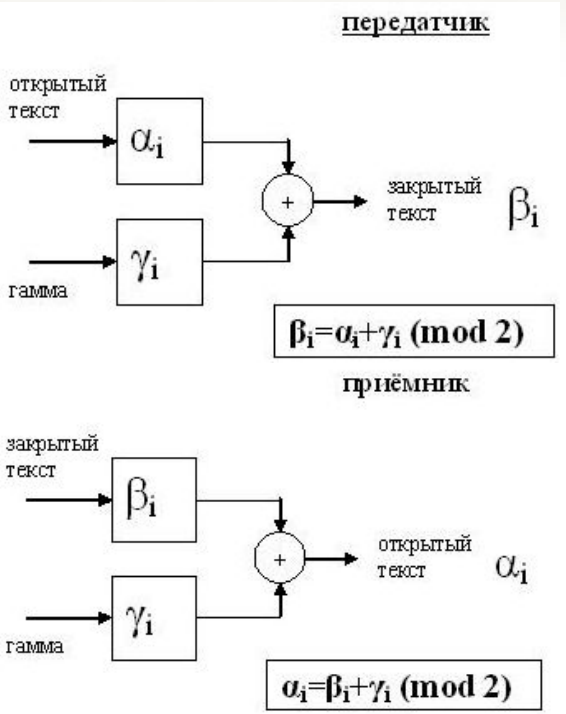
Гаммирование….

Преобразование символов исходного сообщения, при котором символы исходного сообщения складываются по модулю равному мощности алфавита с символами ключа.

Примечание:

Ввиду того что преобразования происходит в ЭВМ, то в качестве алфавита выступает {0,1} и соответственно сложение происходит по модулю 2 (исключающее или – XOR)

Шеннон доказал что если длина гаммы (ключа) равна длине сообщения, то такой шифр взломать нельзя....



Пример:

Алфавит Z_2 – {0,1}

Сложение по модулю 2:

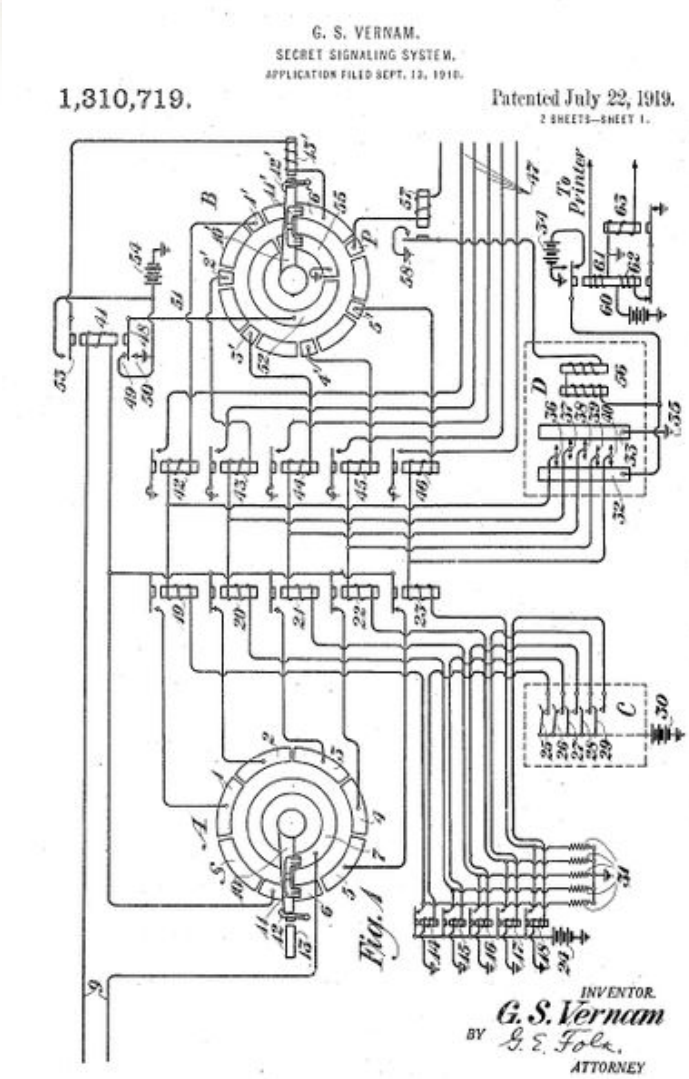
$0 \oplus 1 = 1$

$1 \oplus 1 = 0$ } XOR

$0 \oplus 0 = 0$

исходный текст :	**1001010001111010001101 01....**
гамма:	**1001000101000110101010 10....**
шифртекст :	**0000010100111100100111 11....**

Шифр Вернама



Гилберт Стандфорд Вернам (03.04.1890 – 07.02.1960), инженер по телекоммуникациям, сотрудник *Bell Laboratories*, *AT&T*

В 1917 году изобрёл ***поточный шифр*** и ввёл в обращение успешную инновацию - ***шифроблокнот*** с одноразовыми ключами, так называемый ***«шифр Вернама»***, для которого доказана ***«абсолютная криптографическая стойкость»***.

*Предложил готовить перфоленту со случайными знаками (так называемую «**гамму**») заранее и затем электромеханически складывать ее импульсы с импульсами знаков открытого текста. Полученная сумма представляла собой шифртекст, предназначенный для передачи по линии связи. Вернам установил следующее правило суммирования: если сразу оба импульса являются «плюсами» или «минусами», то итоговый импульс будет «минусом», а если эти импульсы различны, то в результате получится «плюс».*

Шифр Вернама

Вернам сумел слить воедино два процесса – **шифрование** (как ***зашифрование***, так и ***расшифрование***) и **передачу** сообщения.

Он создал то, что впоследствии назвали ***«линейным шифрованием»***, чтобы отличать его от ставшего традиционным ***предварительного шифрования***.

«…. Вернам освободил процесс шифрования от «оков времени и ошибок», исключив из этого процесса человека - выдающийся вклад, внесенный в практику шифрования»…

Шифр Вернама

Проблемы, возникающие при изготовлении, рассылке и уничтожении «гаммы», могут показаться пустячными, однако в условиях быстроменяющейся обстановки объемы переписки зачастую очень велики и удивляют даже самых бывалых связистов.

В течение суток может понадобиться зашифровать сотни тысяч слов, а для этого требуется изготовить миллионы знаков «гаммы». И поскольку «гамма» для каждого сообщения должна быть единственной и неповторимой, то ее ***общий объем будет эквивалентен объему всей переписки за это время!!!***

Майор ***Джозеф Освальд Моборн***, устроив первое испытание шифрсистемы Вернама, установил его машины сразу в трех городах. Даже при небольшом объеме переписки (до 135 коротких сообщений в день) оказалось невозможным изготовить ***достаточное количество качественной «гаммы».***

Не найдя другого выхода из затруднительного положения, Моборн стал комбинировать две относительно короткие «гаммы», чтобы получать из них более длинную «гамму», как это первоначально предлагал делать сам Вернам.

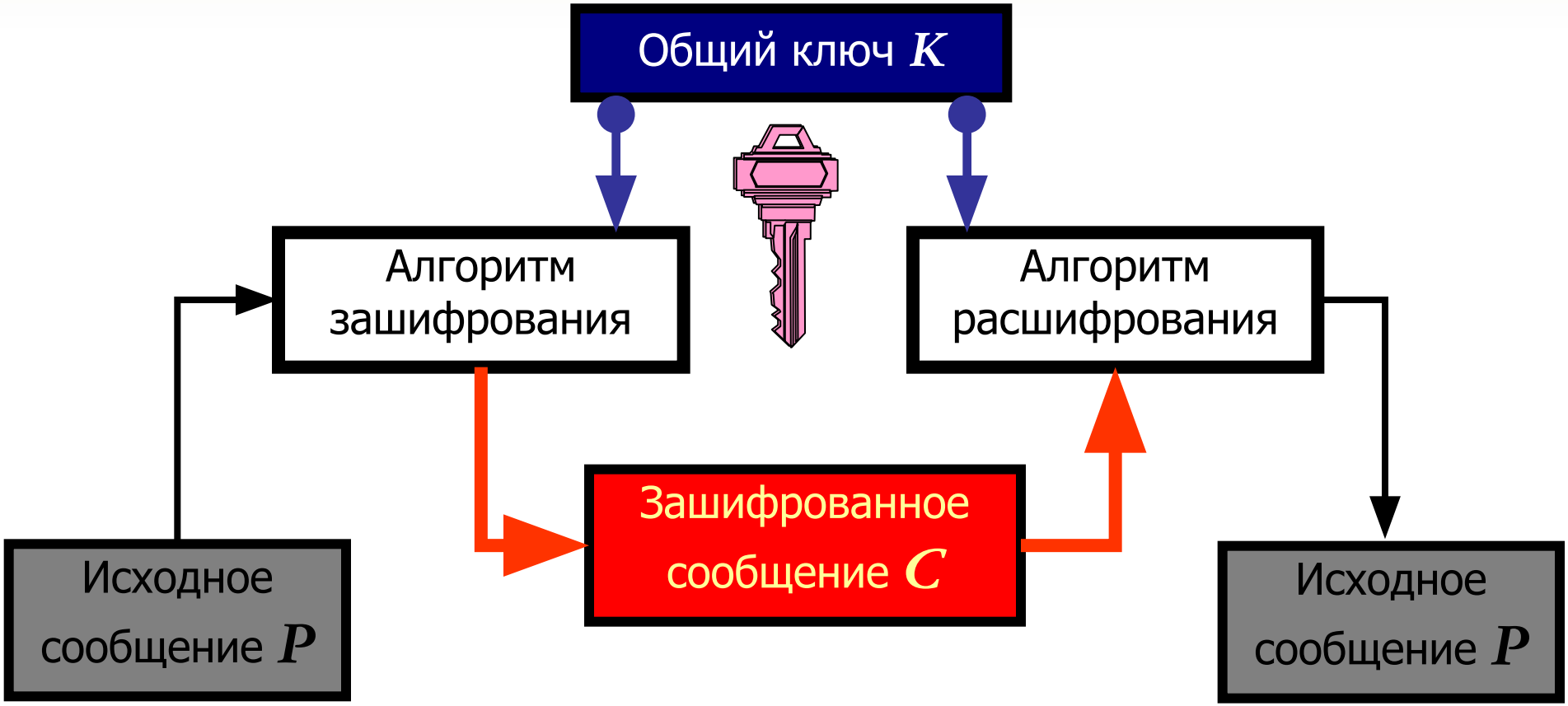
Шифр Вернама

Шифроблокнот Вернама - Моборна


```
ZDXWWW EJKAWO FECIFE WSNZIP PXPKIY URMZHI JZTLBC YLGDYJ
HTSVTV RRYYEG EXNCGA GGQVRF FHZCIB EWLGGR BZXQDQ DGGIAK
YHJYEQ TDLCQT HZBSIZ IRZDYS RBYJFZ AIRCWI UCVXTW YKPQMK
CKHVEX VXYVCS WOGAAZ OUVVON GCNEVR LMBLYB SBDCDC PCGVJX
QXAUIP PXZQIJ JIUWYH COVWMJ UZOJHL DWHPER UBSRUJ HGAAPR
CRWVHI FRNTQW AJVWRT ACAKRD OZKIIB VIQGBK IJCWHF GTTSSE
EXFIPJ KICASQ IOUQTP ZSGXGH YTYCTI BAZSTN JKMFXI RERYWE
```

Одноразовый шифрблокнот содержит случайную «гамму», которая используется ***один, и только один раз***. При этом для каждого знака открытого текста, принадлежащего всей совокупности сообщений, которые уже были посланы данной группой корреспондентов или еще только будут посланы ею в обозримом будущем, предусматривается использование абсолютно ***нового и не поддающегося предсказанию знака «гаммы».***

Общая схема симметричного (традиционного) шифрования



Симметричные шифры

Постулатом для симметричных криптосистем является ***секретность ключа***.

Симметричные криптосхемы в настоящее время принято подразделять на **блочные** и **поточные**.

Блочные криптосистемы разбивают текст сообщения (файла, документа и т.д.) на отдельные блоки и затем осуществляют преобразование этих блоков с использованием ключа.

Поточные криптосистемы работают несколько иначе. На основе ключа системы вырабатывается некая последовательность - так называемая ***выходная гамма***, которая затем накладывается на текст сообщения. Таким образом, преобразование текста осуществляется потоком по мере выработки гаммы.

Блочные шифры

Представляют семейство обратимых преобразований блоков (частей фиксированной длины) исходного текста.

Фактически блочный шифр – система подстановки на алфавите блоков

N-разрядным *блоком* называют последовательность из нулей и единиц длины ***N***

X – *можно рассматривать как вектор или как число*

$$X = (x_0, x_1, x_2, \ldots x_{N-1})$$

Зашифрование – замена исходного блока ***X*** на блок ***Y*** в соответствие с заданным алгоритмом и ключом ***K***

Расшифрование – замена блока ***Y*** на блок ***X*** в соответствие с заданным алгоритмом и ключом ***K***

Сеть Фейстеля

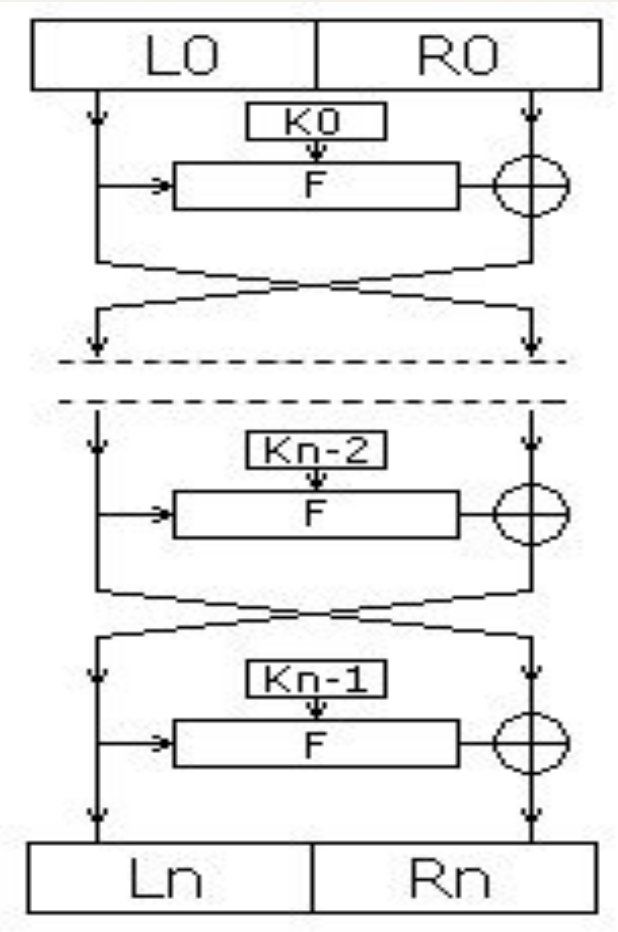
В 1971 году **Хорст Фейстель** (*Horst Feistel*) запатентовал два устройства, реализовавшие различные алгоритмы шифрования, названные затем общим названием «Люцифер» (*Lucifer*). Одно из устройств использовало конструкцию, впоследствии названную «***сетью Фейстеля***» («*Feistel cipher*», «*Feistel network*»).

Сеть Фейстеля имеет следующую структуру.

- Входной блок делится на несколько равной длины подблоков, называемых ветвями.
- В случае, если блок имеет длину 64 бита, используются две ветви по 32 бита каждая.
- Каждая ветвь обрабатывается независимо от другой, после чего осуществляется циклический сдвиг всех ветвей влево.
- Такое преобразование выполняется несколько циклов или раундов.

Зашифровывание

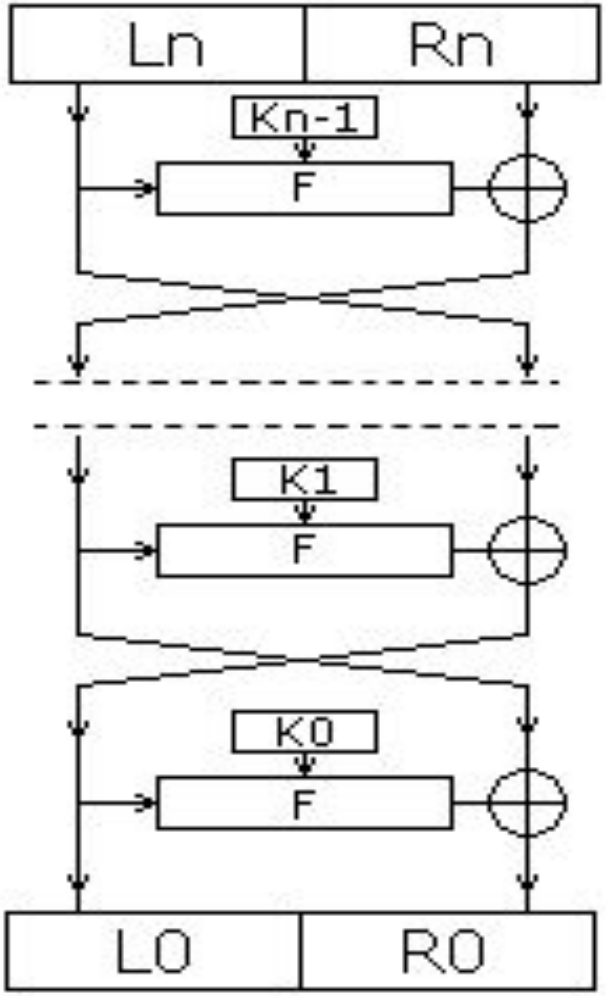
- Выбранный блок делится на два равных подблока — «левый» (***L0***) и «правый» (***R0***).
- «Левый подблок» ***L0*** видоизменяется функцией ***f(L0,K0)*** в зависимости от раундового ключа ***K0***, после чего он складывается по модулю 2 с «правым подблоком» ***R0***.
- Результат сложения присваивается новому левому подблоку ***L1***, который будет половиной входных данных для следующего раунда, а «левый подблок» ***L0*** присваивается без изменений новому правому подблоку ***R1***, который будет другой половиной.
- После чего операция повторяется ***N-1*** раз, при этом при переходе от одного этапа к другому меняются раундовые ключи (***K0*** на ***K1*** и т. д.) по какому-либо математическому правилу, где ***N*** — количество раундов в заданном алгоритме



Генерация раундовых ключей происходит на базе ключа шифрования и зависит от алгоритма шифрования

Расшифровывание

Происходит так же, как и зашифровывание, с тем лишь исключением, что ключи идут в обратном порядке, то есть не от первого к N-ному, а от N-го к первому.



Особенности Сети Фейстеля

Увеличение количества раундов значительно увеличивает криптостойкость алгоритма.

Возможно, эта особенность и повлияла на столь активное распространение сети Фейстеля, так как для большей криптостойкости достаточно просто увеличить количество раундов, не изменяя сам алгоритм. В последнее время количество раундов не фиксируется, а лишь указываются допустимые пределы.

Сеть Фейстеля является обратимой даже в том случае, если функция F не является таковой, так как для дешифрования не требуется вычислять F^{-1}.

Для дешифрования используется тот же алгоритм, но на вход подается зашифрованный текст, и ключи используются в обратном порядке.

Особенности Сети Фейстеля

В настоящее время все чаще используются различные разновидности сети Фейстеля для 128-битного блока с четырьмя ветвями.

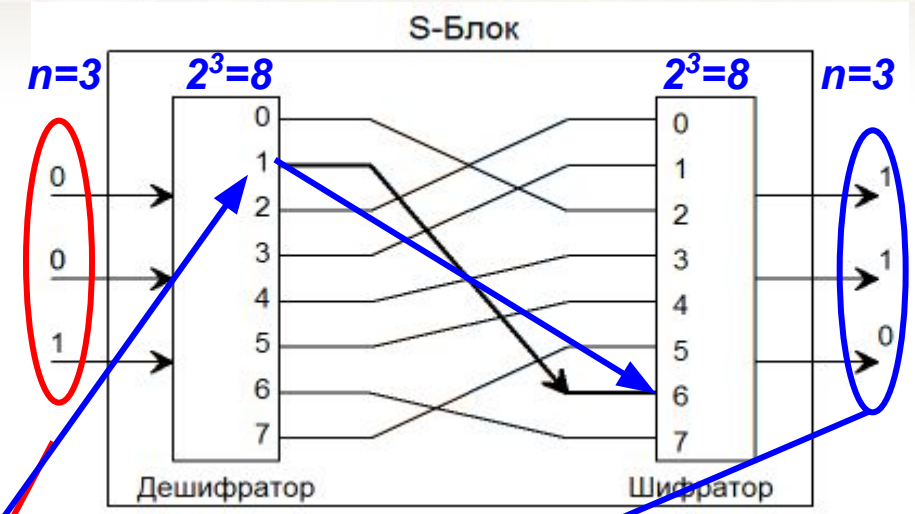
Увеличение количества ветвей, а не размерности каждой ветви связано с тем, что наиболее популярными до сих пор остаются процессоры с 32-разрядными словами, следовательно, оперировать 32-разрядными словами эффективнее, чем с 64-разрядными.

Основной характеристикой алгоритма, построенного на основе сети Фейстеля, является функция *F*. Различные варианты касаются также начального и конечного преобразований. Подобные преобразования, называемые забеливанием (*whitening*), осуществляются для того, чтобы выполнить начальную рандомизацию входного текста.

Функция *F* = *S-box* (подстановка, замена)

Блок подстановок (***S-блок***) **состоит из:**

- **Дешифратора (демультиплексора)**, преобразующего ***n***-разрядный двоичный сигнал в одноразрядный сигнал по основанию 2^n
- **Системы коммутаторов** внутренних соединений (всего соединений $2^n!$)
- **Шифратора (мультиплексора)**, переводящего сигнал из одноразрядного 2^n-ричного в ***n***-разрядный двоичный.



Эквивалентная таблица замен для рассматриваемого трех разрядного S-Box

№ комбинации	0	1	2	3	4	5	6	7
Вход	000	001	010	011	100	101	110	111
Выход	010	110	000	001	011	100	111	101

В электронике можно непосредственно применять приведённую схему, в программировании же генерируют ***таблицы замены***. Оба этих подхода являются эквивалентными, то есть файл, зашифрованный на компьютере, можно расшифровать на электронном устройстве и наоборот

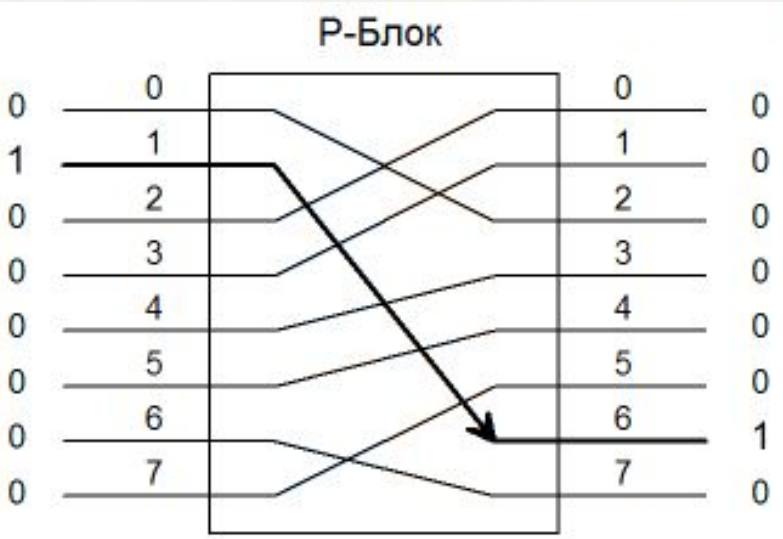
Алгоритмы блочного шифрования

Substitution box, пример (AES)

S-box

```
Sbox = array(
        0x63,0x7c,0x77,0x7b,0xf2,0x6b,0x6f,0xc5,0x30,0x01,0x67,0x2b,0xfe,0xd7,0xab,0x76,
        0xca,0x82,0xc9,0x7d,0xfa,0x59,0x47,0xf0,0xad,0xd4,0xa2,0xaf,0x9c,0xa4,0x72,0xc0,
        0xb7,0xfd,0x93,0x26,0x36,0x3f,0xf7,0xcc,0x34,0xa5,0xe5,0xf1,0x71,0xd8,0x31,0x15,
        0x04,0xc7,0x23,0xc3,0x18,0x96,0x05,0x9a,0x07,0x12,0x80,0xe2,0xeb,0x27,0xb2,0x75,
        0x09,0x83,0x2c,0x1a,0x1b,0x6e,0x5a,0xa0,0x52,0x3b,0xd6,0xb3,0x29,0xe3,0x2f,0x84,
        0x53,0xd1,0x00,0xed,0x20,0xfc,0xb1,0x5b,0x6a,0xcb,0xbe,0x39,0x4a,0x4c,0x58,0xcf,
        0xd0,0xef,0xaa,0xfb,0x43,0x4d,0x33,0x85,0x45,0xf9,0x02,0x7f,0x50,0x3c,0x9f,0xa8,
        0x51,0xa3,0x40,0x8f,0x92,0x9d,0x38,0xf5,0xbc,0xb6,0xda,0x21,0x10,0xff,0xf3,0xd2,
        0xcd,0x0c,0x13,0xec,0x5f,0x97,0x44,0x17,0xc4,0xa7,0x7e,0x3d,0x64,0x5d,0x19,0x73,
        0x60,0x81,0x4f,0xdc,0x22,0x2a,0x90,0x88,0x46,0xee,0xb8,0x14,0xde,0x5e,0x0b,0xdb,
        0xe0,0x32,0x3a,0x0a,0x49,0x06,0x24,0x5c,0xc2,0xd3,0xac,0x62,0x91,0x95,0xe4,0x79,
        0xe7,0xc8,0x37,0x6d,0x8d,0xd5,0x4e,0xa9,0x6c,0x56,0xf4,0xea,0x65,0x7a,0xae,0x08,
        0xba,0x78,0x25,0x2e,0x1c,0xa6,0xb4,0xc6,0xe8,0xdd,0x74,0x1f,0x4b,0xbd,0x8b,0x8a,
        0x70,0x3e,0xb5,0x66,0x48,0x03,0xf6,0x0e,0x61,0x35,0x57,0xb9,0x86,0xc1,0x1d,0x9e,
        0xe1,0xf8,0x98,0x11,0x69,0xd9,0x8e,0x94,0x9b,0x1e,0x87,0xe9,0xce,0x55,0x28,0xdf,
        0x8c,0xa1,0x89,0x0d,0xbf,0xe6,0x42,0x68,0x41,0x99,0x2d,0x0f,0xb0,0x54,0xbb,0x16,
    );
```

Функция *F* = *P-box* (перестановка)



Блок перестановок (***P-блок***) всего лишь изменяет положение цифр и является ***линейным устройством.***

Этот блок может иметь очень большое количество входов-выходов, однако в силу линейности систему нельзя считать криптоустойчивой.

Вход	**01000000**	**10000000**	**....**	**11000000**
Выход	**00000010**	**00100000**	**....**	**00100010**

Криптоанализ ключа для n-разрядного P-блока проводится путём подачи на вход n-1 различных сообщений, каждое из которых состоит из n-1 нуля («0») и 1 единицы («1»).

Основные режимы работы блочных шифров

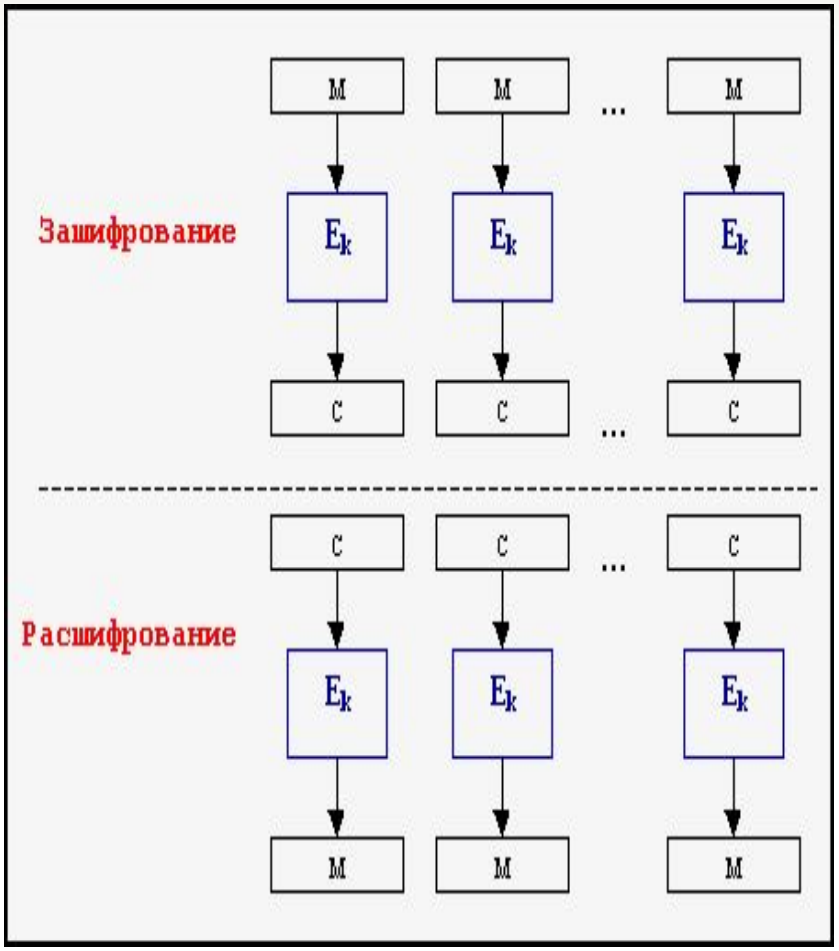
- электронная кодовая книга - ***ECB (Electronic Code Book)***
- сцепление блоков шифртекста - ***CBC (Cipher Block Chaining)***
- обратная связь по шифртектсту - ***CFB (Cipher Feed Back)***
- обратная связь по выходу - ***OFB (Output Feed Back)***

Электронная кодовая книга (ECB)

Режим простой замены

Исходный текст разбивается на блоки (***M***), равные размеру блока шифра (***E_K***).

Затем с каждый блок шифруют независимо от других с использованием одного ключа шифрования.



Непосредственно этот режим применяется для шифрования небольших объемов информации, размером не более одного блока или для шифрования ключей. Это связано с тем, что одинаковые блоки открытого текста преобразуются в одинаковые блоки шифротекста, что может дать взломщику (криптоаналитику) определенную информацию о содержании сообщения. К тому же, если он предполагает наличие определенных слов в сообщении (например, слово «Здравствуйте» в начале сообщения или «До свидания» в конце), то получается, что он обладает как фрагментом открытого текста, так и соответствующего шифротекста, что может сильно облегчить задачу нахождения ключа.

Основным достоинством этого режима является простота реализации

Алгоритмы блочного шифрования

Сцепление блоков шифртекста (CBC)

Режим простой замены с зацеплением

Наиболее часто применимый режим шифрования для обработки больших количеств информации.

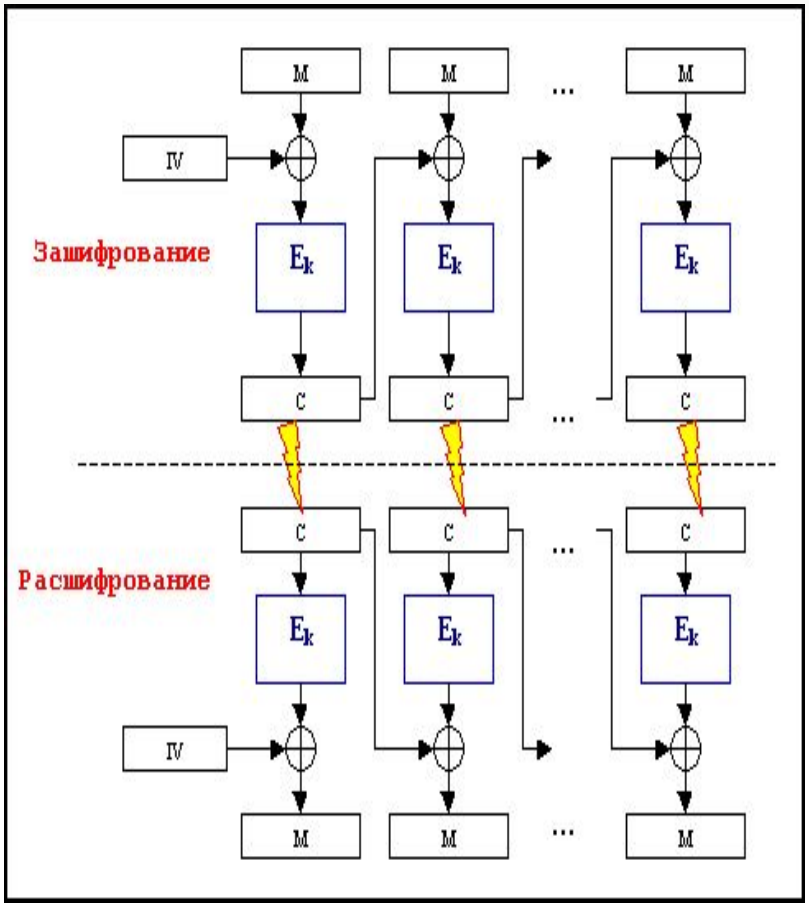
Первый блок складывается побитно по модулю **2** (***XOR***) с неким значением ***IV*** - начальным вектором (***Initialization Vector***), который выбирается независимо перед началом шифрования.

Полученное значение зашифровывается.

Полученный в результате блок шифртекста отправляется получателю и одновременно служит начальным вектором ***IV*** для следующего блока открытого текста.

Расшифрование осуществляется в обратном порядке.

В виде формулы, преобразование в режиме ***CBC*** можно представить как $C_i=E_k(M_i \oplus C_{i-1})$, где i - номер соответствующего блока. Из-за использования такого сцепления блоков шифртекста с открытым текстом пропадают недостатки режима ***ECB***, поскольку каждый последующий блок зависит от всех предыдущих.



Алгоритмы блочного шифрования

Обратная связь по шифртексту (CFB)

Режим гаммирования с обратной связью

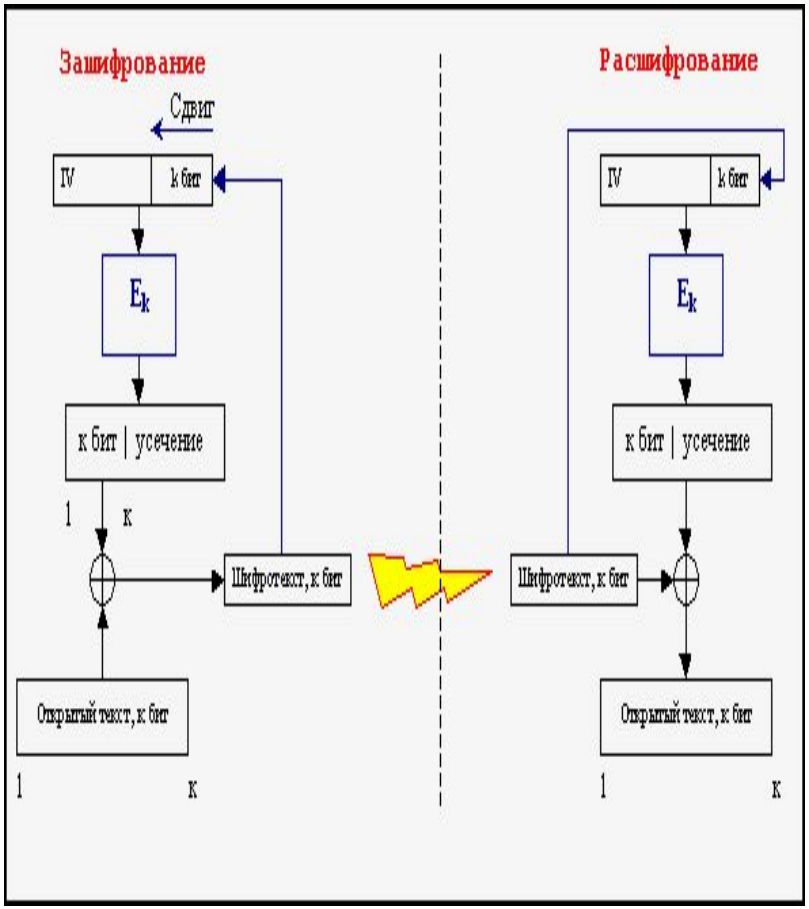
Режим может использоваться для получения ***поточного*** шифра из ***блочного***. Размер блока в данном режиме *меньше, либо равен размеру блока шифра*.

1. ***IV*** **представляет собой сдвиговый регистр. Вначале** ***IV*** **заполняется неким значением, которое называется** ***синхропосылкой*****, не является секретным и передается перед сеансом связи получателю.**
2. **Значение** ***IV*** **шифруется.**
3. **Берутся первые** ***k*** **бит зашифрованного значения** ***IV*** **и складываются** ***(XOR)*** **с** ***k*** **битами открытого текста - получается блок шифртекста из** ***k*** **бит.**
4. **Значение** ***IV*** **сдвигается на** ***k*** **битов влево, а вместо него становится значение шифрованного текста.**
5. **Затем опять 2 пункт и т.д до конца.**

Расшифрование осуществляется аналогично.

Особенностью данного режима является распространение ошибки на весь последующий текст. Рекомендованные значения ***k: 1 =< k =< 8***.

Применяется, как правило, для шифрования потоков информации типа оцифрованной речи,



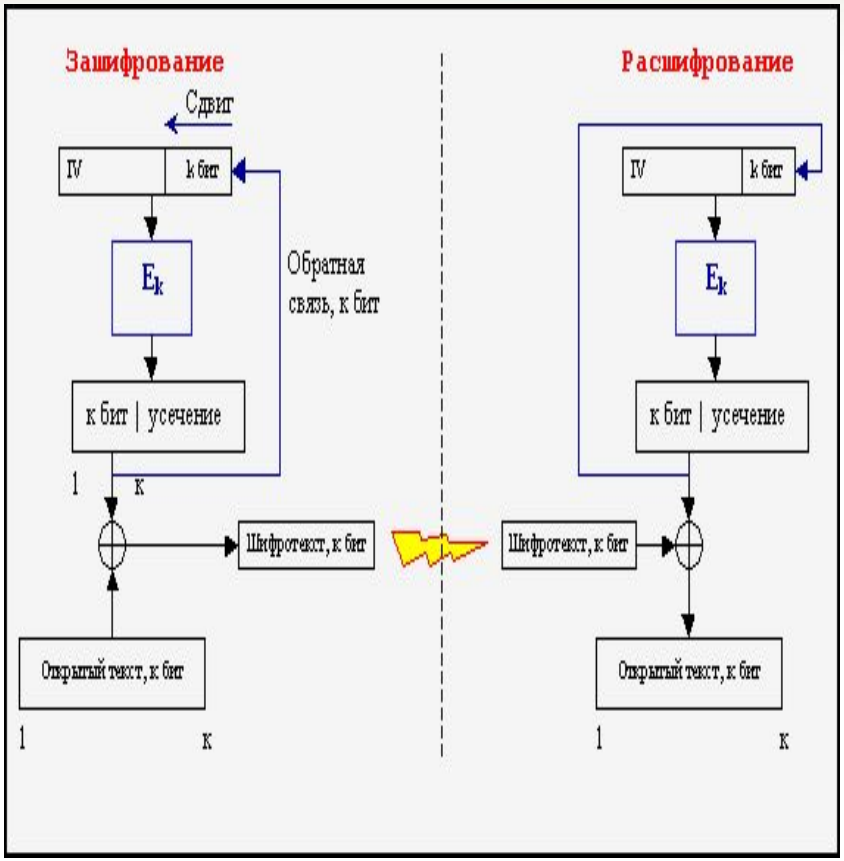
Обратная связь по выходу (OFB)

Режим гаммирования

Данный режим примечателен тем, что позволяет получать поточный шифр в его классическом виде, в отличии от режима ***CFB***, в котором присутствует связь с шифртекстом. Принцип работы схож с принципом работы режима ***CFB***, но сдвиговый регистр ***IV*** заполняется не битами шифртекста, а битами, выходящими из под усечения.

Для любого блока длины ***k*** операция зашифрования выглядит следующим образом: $C_i=M_i \oplus G_i$, где G_i - результат зашифрования некоторого вектора, являющегося заполнением сдвигового регистра. Расшифрование осуществляется аналогично. **Главное свойство шифра - *единичные ошибки не распространяются*,** т.к заполнение сдвигового регистра осуществляется не зависимо от шифротекста.

Область применения: потоки видео, аудио или данных, для которых необходимо обеспечить оперативную доставку. Широко используется в военной сфере наряду с поточными шифрами.



Симметричные блочные алгоритмы

Название	Длина ключа, бит	Размер блока, бит	Количество раундов
Rijndael	128, 192 или 256	128, 160,192, 228 или 256	10, 12, 14
AES	128, 192 или 256	128	10, 12, 14
Blowfish	32-448	64	16
CAST-128	128	64	16
DES	56	64	16
IDEA	128	64	8
RC2	до 1024	64	16
RC5	до 2048	32, 64 или 128	0...255
ГОСТ 28147-89	256	64	16 или 32
Магма	256	64	32
Кузнечик	256	128	10

Алгоритмы блочного шифрования

DES - Data Encryption Standard

Является самым распространенным и наиболее известным алгоритмом симметричного шифрования. Алгоритм был разработан в 1977 году, а в 1980 году был принят *NIST (National Institute of Standards and Technolody)* США в качестве ***стандарта*** (*FIPS PUB 46*).

Является *классической сетью Фейстеля* с двумя ветвями. Алгоритм преобразует за несколько раундов 64-битный входной блок в 64-битный выходной блок. Длина ключа равна 56 битам.

Процесс шифрования состоит из четырех этапов.

На **первом** из них выполняется начальная перестановка (***IP***) 64-битного исходного текста (забеливание), во время которой биты переупорядочиваются в соответствии со ***стандартной таблицей IP***.

Второй этап состоит из 16 раундов одной и той же функции, которая использует операции сдвига и подстановки.

На **третьем** этапе левая и правая половины выхода последней (16-й) итерации меняются местами.

На **четвертом** этапе выполняется перестановка IP^{-1} результата, полученного на третьем этапе. Перестановка IP^{-1} инверсная начальной перестановке.

Проблемы *DES*.

Так как длина ключа равна ***56*** битам, существует всего $2^{56} = 7{,}2*10^{16}$ возможных ключей.

На сегодня такая длина ключа недостаточна, поскольку допускает успешное применение лобовых атак.

Также без ответа пока остается вопрос, возможен ли криптоанализ с использованием существующих характеристик алгоритма ***DES***. Основой алгоритма являются ***восемь таблиц подстановки***, (***S-boxes***), которые применяются в каждой итерации.

Существует опасность, что эти ***S-boxes*** конструировались таким образом, что криптоанализ возможен со стороны взломщика, который знает слабые места этих ***S-boxes***.

В течение многих лет обсуждалось как стандартное, так и неожиданное поведение ***S-boxes***, но все-таки никому не удалось обнаружить их фатально слабые места.

Криптоанализ

Основные результаты усилий по взлому DES:

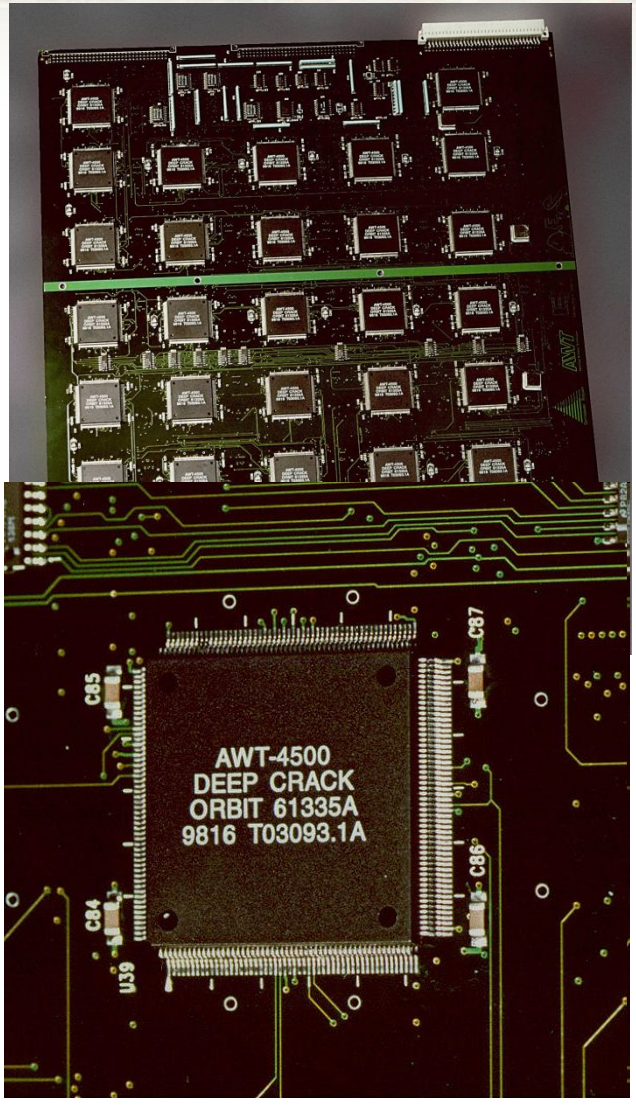
- 1987 г. - Дэвис - метод вычисления ключа DES, основанный на специфических свойствах таблиц замен DES
- 1991 г. - Эли Бихам и Эди Шамир - атака, в которой ключ шифрования вычисляется методом дифференциального криптоанализа при наличии у атакующего возможности генерации 2^{47} специально выбранных пар «открытый текст – зашифрованный текст»
- 1993 г.- Мицуру Мацуи доказал возможность вычисления ключа шифрования методом линейного криптоанализа при наличии у атакующего 2^{47} пар «открытый текст – зашифрованный текст». В дальнейшем эта атака была усилена = 2^{43} пар вместо 2^{47}.
- 1994 г. - Эли Бихам и Алекс Бирюков усилили метод Дэвиса. Их метод позволяет вычислить 6 бит ключа, остальные 50 бит – полным перебором возможных вариантов при наличии 2^{50} пар известных открытых текстов и шифртекстов или вычислить 24 бита ключа при наличии 2^{52} пар.

Алгоритмы блочного шифрования
DES - Data Encryption Standard

Криптоанализ

- 1998 г. используя суперкомпьютер стоимостью 250 тыс. долл., сотрудники RSA Laboratory «взломали» DES менее чем за три дня. Эксперимент проходил в рамках исследования DES Challenge II, проводимого RSA Laboratory под руководством общественной организации Electronic Frontier Foundation (EFF). Суперкомпьютер, построенный в RSA Laboratory для расшифровки данных, закодированных методом DES по 56-разрядному ключу, получил название ***EFF DES Cracker.***

  Проблема усугубляется тем, что стоимость постройки подобного суперкомпьютера неуклонно снижается. ***«Через четыре-пять лет такие компьютеры будут стоять в любой школе»,*** — считает Джон Гилмор, руководитель проекта DES Challenge и один из основателей EFF.



Алгоритмы блочного шифрования

Тройной DES (Triple DES)

Т.к. основным недостатком ***DES*** считается малая длина ключа, давно начали разрабатываться различные альтернативы этому алгоритму шифрования. Один из подходов состоит в том, чтобы разработать новый алгоритм, и успешный тому пример - ***IDEA***. Другой подход предполагает повторное применение алгоритма ***DES*** с различными ключами.

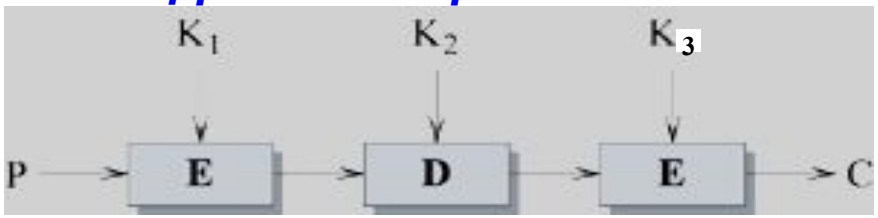
В этом случае выполняется последовательность зашифрование – расшифрование - зашифрование ***(E ⇒ D ⇒ E):*** $C = EK_1 [DK_2 [EK_3 [P]]]$

Тройной DES является достаточно популярной альтернативой DES и используется при управлении ключами в стандартах ***ANSI X9.17, ISO 8732*** и ***PEM (Privacy Enhanced Mail)***.

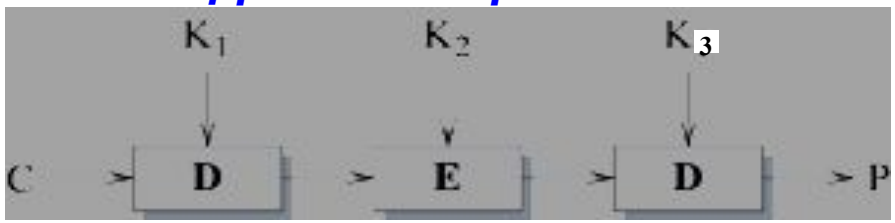
Известных криптографических атак на тройной *DES* не существует.

Цена подбора ключа в ***тройном DES*** равна 2^{112}.

Шифрование тройным DES



Расшифрование тройным DES



Алгоритмы блочного шифрования

IDEA *(International Data Encryption Algorithm)*

Использует 128-битный ключ. Открытый текст разбивается на блоки по 64 бит.

Модификация сети Фейстеля.

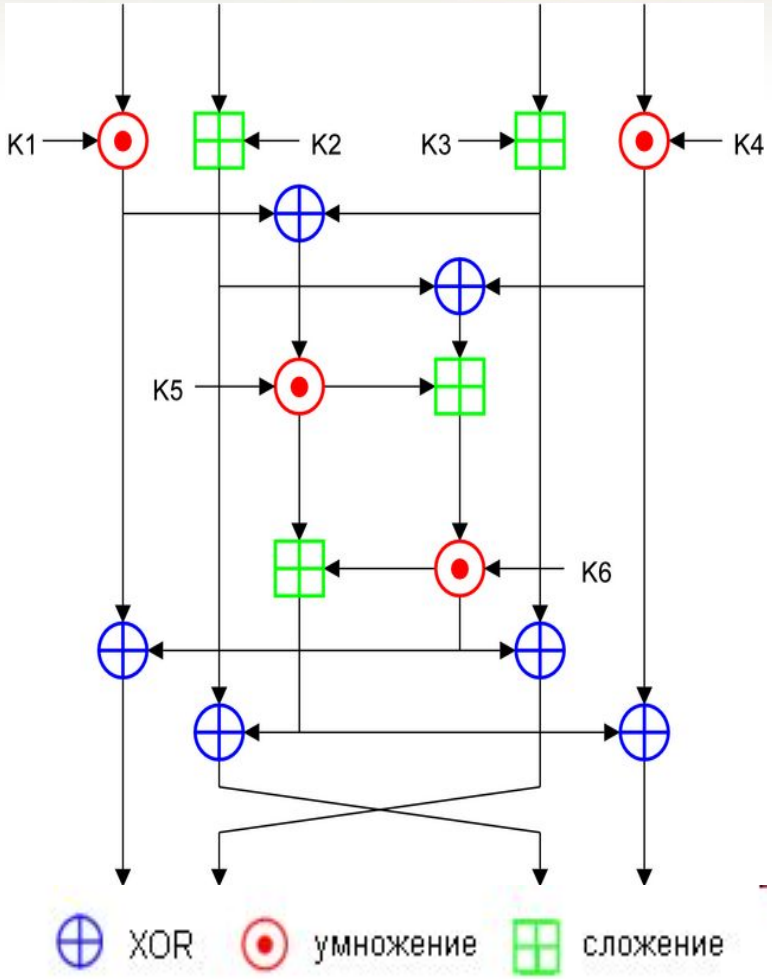
Если такое разбиение не возможно, используются различные режимы шифрования.

Каждый исходный незашифрованный 64-битный блок делится на четыре подблока по 16 бит каждый, так как все алгебраические операции, использующиеся в процессе шифрования, совершаются над 16-битными числами.

Новым в алгоритме является использование операций из разных алгебраических групп:

- сложение по модулю 2^{16}
- умножение по модулю $2^{16} + 1$
- побитовое исключающее ИЛИ (XOR).

Применение этих операций затрудняет криптоанализ IDEA по сравнению с DES, который основан исключительно на операции XOR, а также позволяет отказаться от использования S-блоков и таблиц замены.



K1-K6 – 16-битные подключи

AES

2 января 1997 года *NIST* объявил о начале разработки ***AES (Advanced Encryption Standard)***.

12 сентября 1997 года были представлены официальные требования к нему. В них указывалось, что целью является разработка неклассифицированного, хорошо проанализированного алгоритма шифрования, доступного для широкого применения.

«Как минимум алгоритм должен был быть симметричным и блочным и поддерживать длину блока 128 бит и длину ключа 128, 192 и 256 бит».

Алгоритмы блочного шифрования

AES

20 августа 1998 года *NIST* анонсировал ***пятнадцать*** кандидатов ***CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, Twofish***, и было предложено прокомментировать их характеристики.

Данные алгоритмы были разработаны специалистами ***двенадцати стран***.

Вторая конференция была проведена в марте 1999 года. Результат голосования: ***Rijndael***: 86 за, 10 против...

В августе 1999 года были представлены ***пять финалистов: MARS, RC6™, Rijndael, Serpent*** и ***Twofish***.

Третья конференция: 13-14.04 2000 г.

> *«За три года проведения конкурса на звание нового стандарта шифрования каждый из пяти финалистов подвергся столь же тщательному исследованию государственного и гражданского криптологического сообщества всего мира, сколь и DES за 20 лет своей жизни».*

Алгоритмы блочного шифрования

AES

Джоан Даймен (1965)
и
Винсент Риджмен (1970)

Rijndael, разработка криптографов из Бельгии, ***Винсента Риджмена и Джоана Даймена,*** был выбран благодаря простому дизайну, облегчающему его анализ, малому размеру исполняемого кода и требованиям к памяти, высокой скорости и другим параметрам, отличающим его от конкурентов, даже несмотря на незначительно меньший «запас прочности» (т.е. способность противостоять атакам в своих ослабленных вариантах с меньшим числом раундов), тем не менее, не несущий практической угрозы безопасности алгоритма.

Алгоритмы блочного шифрования

AES

Алгоритм ***Rijndael*** представляет блок данных в виде двухмерного байтового массива размером 4X4, 4X5, 4X6, 4X7 или 4X8, *(т.е. допускается использование нескольких фиксированных размеров шифруемого блока информации: 128, 160, 192, 224, 256)*, поэтому схема этого алгоритма стала называться ***«квадрат (square)»***. Все операции выполняются с отдельными байтами массива, а также с независимыми столбцами и строками.

Алгоритм ***Rijndael*** выполняет ***четыре*** преобразования:

SB *(SubBytes)* - табличная замена каждого байта массива;

SR *(ShiftRow)* - сдвиг строк массива. При этой операции первая строка остается без изменений, а остальные циклически побайтно сдвигаются влево на фиксированное число байт, зависящее от размера массива. Например, для массива размером 4X4 строки 2, 3 и 4 сдвигаются соответственно на 1, 2 и 3 байта;

MC *(MixColumn)* - операция над независимыми столбцами массива, когда каждый столбец по определенному правилу умножается на фиксированную матрицу ***c(x)***;

AK *(AddRoundKey)* - добавление ключа раунда; каждый бит массива складывается по модулю 2 с соответствующим битом ***ключа раунда, который,*** в свою очередь, определенным образом ***вычисляется из ключа шифрования***.

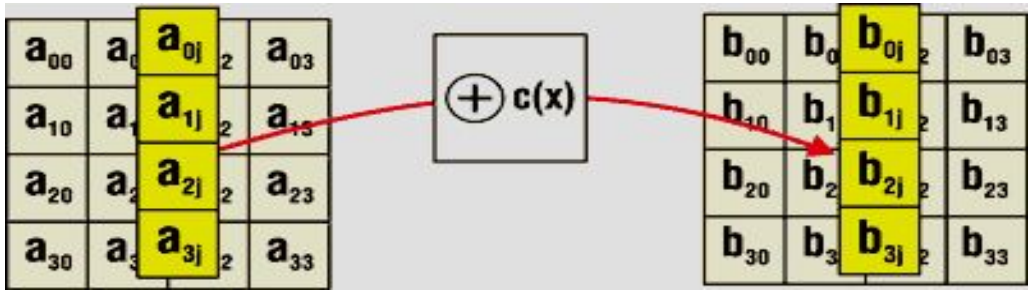
Алгоритмы блочного шифрования

AES. Схемы операций алгоритма *Rijndael*

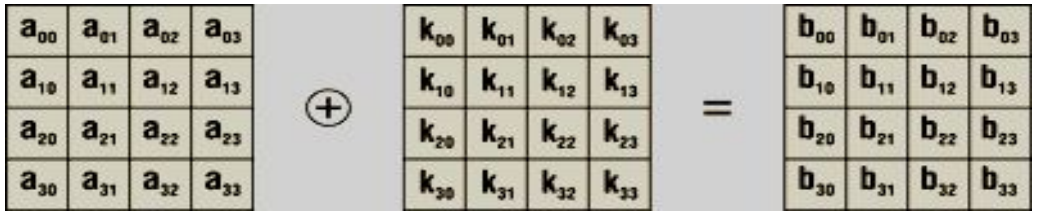
Операция *SB*

Операция *SR*

Операция *MC*



Операция *AK*



AES. Резюме

Rijndael стал ***стандартом шифрования AES*** благодаря ряду преимуществ перед другими алгоритмами-конкурсантами:

- обеспечивает высокую скорость шифрования на всех платформах: как при программной, так и при аппаратной реализации.
- лучшие возможности распараллеливания вычислений по сравнению с другими алгоритмами, представленными на конкурс.
- требования к ресурсам для его работы минимальны, что важно при его использовании в устройствах, обладающих ограниченными вычислительными возможностями.

Недостатком алгоритма считается его нетрадиционная схема

Аппаратная поддержка ***AES*** ***(и только его)*** введена фирмой ***Intel*** в семейство процессоров ***x86*** начиная с ***Intel Core i7-980X Extreme Edition***, далее на процессорах с ядром ***Sandy Bridge***.

ГОСТ 28147-89

«Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования»

Алгоритм опубликован в 1989 году.

Представляет собой классическую сеть Фейстеля.

Алгоритм имеет длину ключа шифрования ***256*** бит.

Шифрует информацию блоками по 64 бита, которые затем разбиваются на два субблока по 32 бита (***N_1*** и ***N_2***).

Субблок ***N_1*** обрабатывается определенным образом, после чего его значение складывается со значением субблока ***N_2*** (сложение выполняется по модулю 2, т. е. применяется логическая операция ***XOR***, а затем субблоки меняются местами.

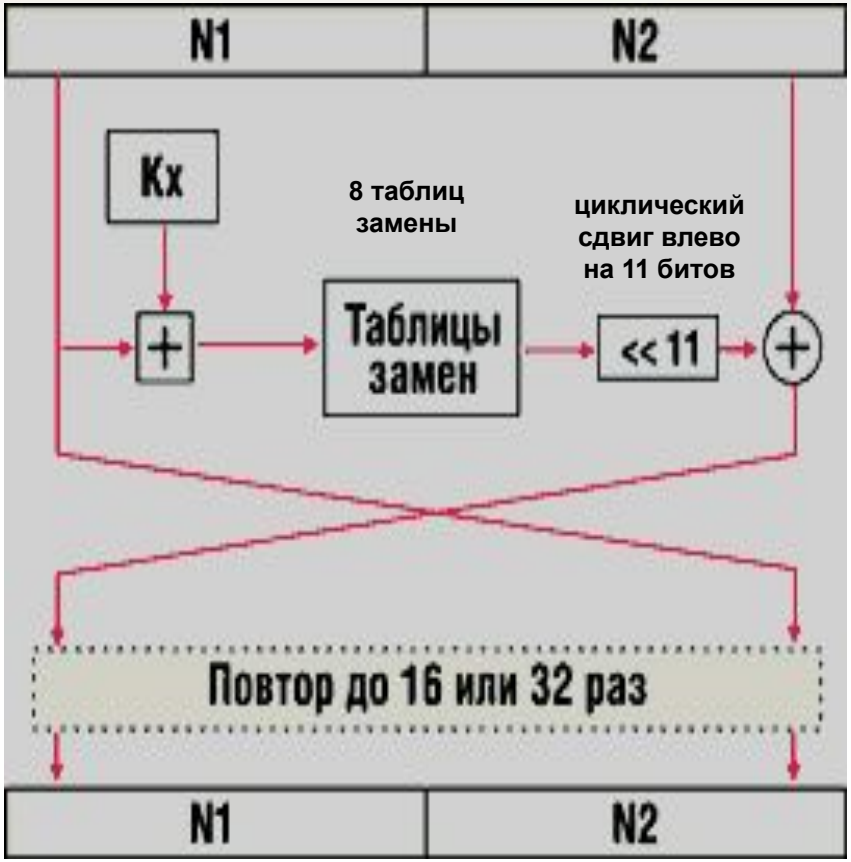
Данное преобразование выполняется определенное число раундов - 16 или 32 в зависимости от режима работы алгоритма.

Алгоритмы блочного шифрования

ГОСТ 28147-89

В каждом раунде выполняются две операции:

- ***наложение ключа*** - содержимое субблока N_1 складывается по модулю 2 с 32-битовой частью ключа K_X. Полный ключ шифрования представляется в виде конкатенации 32-бит подключей: K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7. В процессе шифрования используется один из этих подключей - в зависимости от номера раунда и режима работы алгоритма;
- ***табличная замена N_1*** - после наложения ключа субблок разбивается на 8 частей по 4 бита, значение каждой из которых заменяется в соответствии с таблицей замены для данной части субблока, затем выполняется побитовый циклический сдвиг субблока влево на 11 бит.



Алгоритмы блочного шифрования

ГОСТ 28147-89

Считается очень сильным алгоритмом - в настоящее время для его взлома не предложено никаких более эффективных методов, чем метод прямого перебора.

Высокая стойкость достигается за счет большой длины ключа - ***256*** бит.

При использовании секретной синхропосылки эффективная длина ключа увеличивается до ***320*** бит, а засекречивание таблиц замены увеличивает до ***610*** бит.

Кроме того, криптостойкость зависит от количества раундов преобразований, которых по ***ГОСТ 28147-89*** должно быть ***32*** (*полный эффект рассеивания входных данных достигается уже после **8** раундов*).

ГОСТ 28147-89. Режимы шифрования

Алгоритм, определяемый ***ГОСТ 28147-89***, предусматривает четыре режима работы:

- ***простой замены***
- ***гаммирования***
- ***гаммирования с обратной связью***
- ***генерации имитовставок***

В них используется одно и то же шифрующее преобразование, но поскольку назначение режимов различно, осуществляется это преобразование в каждом из них по-разному.

ГОСТ 28147-89. Генерация ключей

256-битный ключ разбивается на восемь 32-битных подключей. Алгоритм имеет 32 раунда, поэтому каждый подключ используется в четырех раундах по следующей схеме:

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

Критика ГОСТ 28147-89

Основная критика ГОСТа связана с неполнотой стандарта в части генерации ключей и таблиц замен.

- Достаточно просто доказывается, что у ГОСТа существуют «слабые» ключи и таблицы замен, но в стандарте не описываются критерии выбора и отсева «слабых».
- Стандарт не специфицирует алгоритм генерации таблиц замен (***S-блоков***). С одной стороны, это может являться дополнительной секретной информацией (помимо ключа), а с другой, поднимает ряд проблем:
  - a) **нельзя определить криптостойкость алгоритма, не зная заранее таблицы замен**
  - b) **реализации алгоритма от различных производителей могут использовать разные таблицы замен и могут быть несовместимы между собой**
  - c) **потенциальная возможность (отсутствие запрета в стандарте) использования таблиц замены, в которых узлы не являются перестановками, что может привести к чрезвычайному снижению стойкости шифра**

Совместимость реализаций

- ФГУП НТЦ «Атлас»
- ООО «КРИПТО-ПРО»
- ООО «Фактор-ТС»
- ЗАО «МО ПНИЭИ»
- ОАО «Инфотекс»
- ЗАО Санкт-Петербургский Региональный Центр Защиты Информации
- ООО «Криптоком»
- ООО «Р-Альфа»

С участием: Демос, Мобильные ТелеСистемы, Волгоградский Государственный Технический Университет, Центральный Телеграф и др.

RFC 4490 [CPCMS]
RFC 4491 [CPPK]
RFC 4357 [CPALGS]

приняты национальными органами по стандартизации Армении, Белоруссии, Казахстана, Кыргызии, Молдовы, Украины, Таджикистана, Узбекистана и России.
ГОСТ 34.311-95, ГОСТ 34.310-95, ГОСТ 34.310-2004, ГОСТ 28147-89

ГОСТ 28147-89

http://www.cryptopro.ru/blog/2013/08/27/gost-28147-89-ne-speshi-ego-khoronit-chast-1-stoikost-algoritma

Сухой остаток: какова стойкость на практике?

В заключение приведем таблицу, содержащую данные обо всех известных международному криптографическому сообществу результатах строго описанных и обоснованных атак на ГОСТ 28147-89. Отметим, что сложность приводится в операциях зашифрования алгоритма ГОСТ 28147-89, а память и материал указаны в блоках алгоритма (64 бита = 8 байт).

Атака	Трудоемкость	Память	Требуемый материал
Исобе	2^{224}	2^{64}	2^{32}
Динур-Данкельман-Шамир, FP, 2DMitM	2^{192} 6,27…e+57	2^{64} 512 Gb	2^{64} 134217728 Tb
Динур-Данкельман-Шамир, FP, low-memory	2^{204}	2^{19}	2^{64}
Динур-Данкельман-Шамир, Reflection, 2DMitM	2^{224}	2^{36}	2^{32}
Динур-Данкельман-Шамир, Reflection, 2DMitM	2^{236}	2^{19}	2^{32}
Полный перебор 1,1…e+77	2^{256}	1	4
Количество наносекунд с возникновения Вселенной	2^{89}		

Cipher security against publicly known feasible attacks

Cipher			Protocol version				
Type	Algorithm	Strength (bits)	SSL 2.0	SSL 3.0 [n 1][n 2][n 3][n 4]	TLS 1.0 [n 1][n 3]	TLS 1.1 [n 1]	TLS 1.2 [n 1]
Block cipher with mode of operation	AES GCM[24][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure
	AES CCM[25][n 5]		N/A	N/A	N/A	N/A	Secure
	AES CBC[n 6]		N/A	N/A	Depends on mitigations	Secure	Secure
	Camellia GCM[26][n 5]	256, 128	N/A	N/A	N/A		
	Camellia CBC[27][n 6]		N/A	N/A	Depends on mitigations		
	ARIA GCM[28][n 5]	256, 128	N/A	N/A	N/A		
	ARIA CBC[28][n 6]		N/A	N/A	Depends on mitigations		Secure
	SEED CBC[29][n 6]	128	N/A	N/A	Depends on mitigations	Secure	Secure
	3DES EDE CBC[n 6]	112[n 7]	Insecure	Insecure	Low strength, Depends on mitigations	Low strength	Low strength
	GOST 28147-89 CNT[23]	256	N/A	N/A	Secure	Secure	Secure
	IDEA CBC[n 6][n 8]	128	Insecure	Insecure	Depends on mitigations	Secure	N/A
	DES CBC[n 6][n 8]	56	Insecure	Insecure	Insecure	Insecure	N/A
		40[n 9]	Insecure	Insecure	Insecure	N/A	N/A
	RC2 CBC[n 6]	40[n 9]	Insecure	Insecure	Insecure	N/A	N/A
Stream cipher	ChaCha20-Poly1305[33][n 5]	256	N/A	N/A	N/A	N/A	Secure
	RC4[n 10]	128	Insecure	Insecure	Insecure	Insecure	Insecure
		40[n 9]	Insecure	Insecure	Insecure	N/A	N/A
None	Null[n 11]	-	N/A	Insecure	Insecure	Insecure	Insecure

ГОСТ 28147-89

http://www.cryptopro.ru/blog/2013/08/27/gost-28147-89-ne-speshi-ego-khoronit-chast-1-stoikost-algoritma

Несмотря на достаточно масштабный цикл исследований в области стойкости алгоритма ГОСТ 28147-89, на данный момент не известно ни одной атаки, условия для осуществления которой являлись бы достижимыми при сопутствующих длине блока в 64 бита эксплуатационных требованиях. Вытекающие из параметров шифра (битовая длина ключа, битовая длина блока) ограничения на объем материала, который может быть обработан на одном ключе, существенно строже минимального объема, который необходим для осуществления любой из известных на данный момент атак. Следовательно, при выполнении существующих эксплуатационных требований ни один из предложенных к настоящему моменту методов криптоанализа ГОСТ 28147-89 не позволяет определять ключ с трудоемкостью меньшей полного перебора.

Sweet32: Birthday attacks on 64-bit block ciphers in TLS and OpenVPN

CVE-2016-2183, CVE-2016-6329

Cryptographic protocols like TLS, SSH, IPsec, and OpenVPN commonly use block cipher algorithms, such as AES, Triple-DES, and Blowfish, to encrypt data between clients and servers. To use such algorithms, the data is broken into fixed-length chunks, called blocks, and each block is encrypted separately according to a mode of operation. Older block ciphers, such as Triple-DES and Blowfish use a block size of 64 bits, whereas AES uses a block size of 128 bits.

It is well-known in the cryptographic community that a short block size makes a block cipher vulnerable to birthday attacks, even if there are no cryptographic attacks against the block cipher itself. We observe that such attacks have now become practical for the common usage of 64-bit block ciphers in popular protocols like TLS and OpenVPN. Still, such ciphers are widely enabled on the Internet. Blowfish is currently the default cipher in OpenVPN, and Triple-DES is supported by nearly all HTTPS web servers, and currently used for roughly 1-2% of HTTPS connections between mainstream browsers and web servers.

We show that a network attacker who can monitor a long-lived Triple-DES HTTPS connection between a web browser and a website can recover secure HTTP cookies by capturing around 785 GB of traffic. In our proof-of-concept demo, this attack currently takes less than two days, using malicious Javascript to generate traffic. Keeping a web connection alive for two days may not seem very practical, but it worked easily in the lab. In terms of computational complexity, this attack is comparable to the recent attacks on RC4. We also demonstrate a similar attack on VPNs that use 64-bit ciphers, such as OpenVPN, where long-lived Blowfish connections are the norm.

Countermeasures are currently being implemented by browser vendors, OpenSSL, and the OpenVPN team, and we advise users to update to the latest available versions.

ГОСТ Р 34.12-2015

ТК 26 ГОСТР

ТЕХНИЧЕСКИЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ «КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ» (ТК 26)

АКАДЕМИЯ АЙТИ

Национальные стандарты Российской Федерации

• ГОСТ Р 34.12–2015 "Информационная технология. Криптографическая защита информации. Блочные шифры"
(Утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 19 июня 2015г. № 749-ст)

• ГОСТ Р 34.13–2015 "Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров"
(Утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 19 июня 2015г. № 750-ст)

3 Общие положения

В настоящем стандарте приведено описание двух базовых блочных шифров с длинами блоков n = 128 бит и n = 64 бит и длинами ключей k = 256 бит.

Примечания

1 На описанный в настоящем стандарте шифр с длиной блока n = 128 бит можно ссылаться как на блочный шифр «Кузнечик» («Kuznyechik»).

2 На описанный в настоящем стандарте шифр с длиной блока n = 64 бит можно ссылаться как на блочный шифр «Магма» («Magma»).

ГОСТ Р 34.12-2015

ТК 26 ГОСТР

ТЕХНИЧЕСКИЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ «КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ» (ТК 26)

АКАДЕМИЯ АЙТИ

24.06.2015

Новые национальные криптографические стандарты

Блочный шифр является важным криптографическим механизмом, который может использоваться как самостоятельный криптографический алгоритм, так и входить в состав других криптографических алгоритмов и протоколов для защиты данных, передаваемых по сетям засекреченной связи или в информационно-телекоммуникационной сети «Интернет».

Учитывая, что алгоритм криптографического преобразования ГОСТ 28147-89 хорошо зарекомендовал себя в программной и аппаратной реализации и по своим свойствам он не накладывает ограничения на степень конфиденциальности защищаемой информации, признано целесообразным включить в новый криптографический стандарт ГОСТ Р 34.12-2015 «Информационная технология. Криптографическая защита информации. Блочные шифры» описание этого шифра с размером блока 64 бит с зафиксированными блоками нелинейной подстановки (шифр «Магма»). Фиксация блоков нелинейной подстановки сделает алгоритм, описанный в стандарте, более унифицированным и поможет исключить использование «слабых» блоков нелинейной подстановки.

В стандарт также включен новый блочный шифр (шифр «Кузнечик») типа «подстановочно-перестановочная сеть» с размером блока 128 бит. Данный тип шифров является хорошо изученным и относительно простым с точки зрения криптографического анализа и обоснования требуемых свойств. Ожидается, что блочный шифр с длиной блока 128 бит будет устойчив ко всем известным на сегодняшний день атакам на блочные шифры.

Режимы работы n-битного блочного шифра (режим простой замены, режим гаммирования, режим гаммирования с обратной связью по выходу, режим гаммирования с обратной связью по шифртексту, режим простой замены с зацеплением и режим выработки иммитовставки) выведены в отдельный национальный стандарт ГОСТ Р 34.13-2015 «Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров», что соответствует принятой международной практике.

ГОСТ Р 34.12-2015
«Кузнечик»

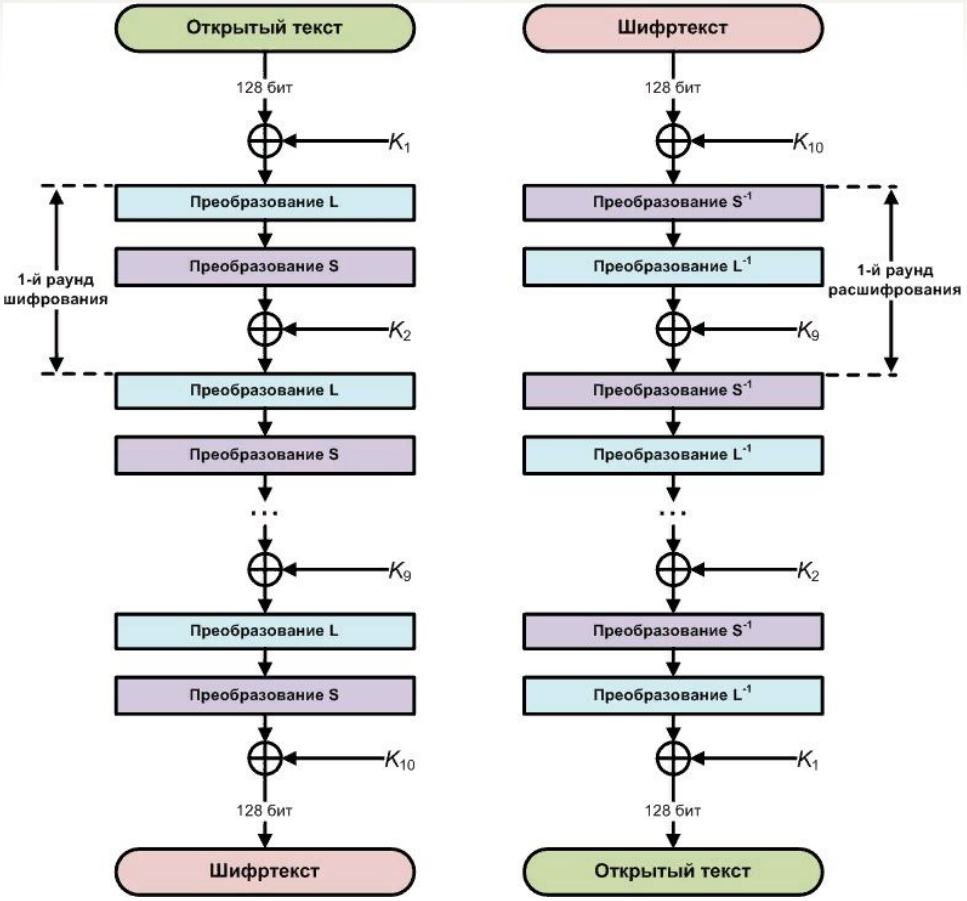
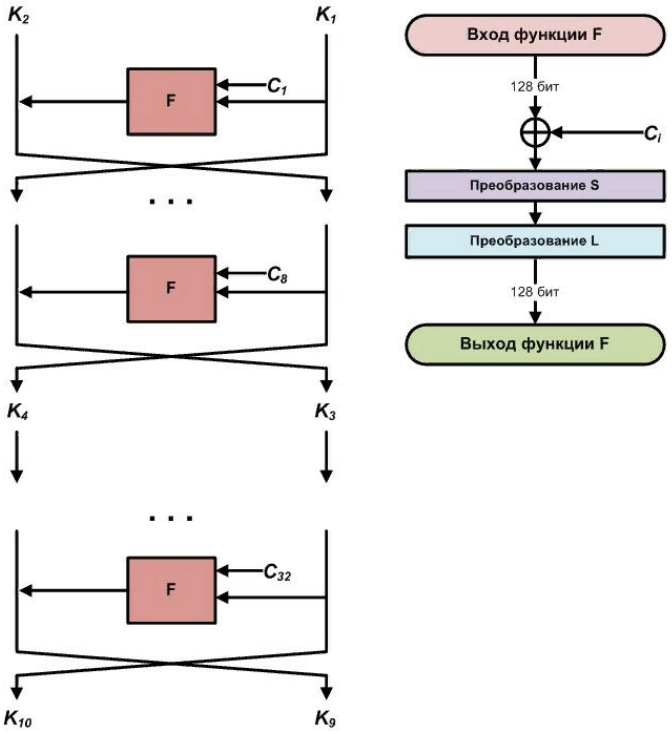


Схема получения итерационных (раундовых) ключей



производительность шифра; в качестве примера и только !!! - на одном ядре Intel Core i7-2677M Sandy Bridge, 1.80 ГГц в версии с SSE инструкциями **= 120 МБ/с**

ГОСТ Р 34.13-2015

ТК 26 ГОСТР

ТЕХНИЧЕСКИЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ
«КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ» (ТК 26)

АКАДЕМИЯ АЙТИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 34.13– 2015

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Режимы работы блочных шифров

ГОСТ Р 34.13-2015

Стандарт определяет следующие режимы работы блочных шифров:

- ***простой замены***
- ***простой замены с зацеплением***
- ***гаммирования***
- ***гаммирования с обратной связью по выходу***
- ***гаммирования с обратной связью по шифртексту***
- ***выработки имитовставки***

Поточные шифры

Шифрование в поточных шифрах осуществляется на основе сложения некоторой ключевой последовательности (***гаммы***) с открытым текстом сообщения. Сложение осуществляется познаково посредством XOR. Уравнение зашифрования выглядит следующим образом:

Поточные шифры

Зашифрование осуществляется наложением потокового ключа - гаммы (***зашифрование гаммированием***).

Иметь ключ, равный по размеру шифруемым данным представляется проблематичным.

Поэтому поточные шифры вырабатывают выходную гамму на основе некоторого секретного ключа небольшого размера, а значит ***основной задачей поточных шифров*** является ***выработка*** некоторой ***последовательности (выходной гаммы)*** для шифрования сообщения.

Основная проблема симметричного шифрования

Основная проблема при применении симметричных криптосистем для связи заключается в **сложности передачи обеим сторонам секретного ключа**.

Распределение ключей между обменивающимися информацией сторонами, использующими алгоритмы симметричного шифрования, можно организовать несколькими способами:

- ключ может быть выбран одной стороной и физически доставлен другой стороне
- ключ может выбрать третья сторона физически доставить его обеим обменивающимися информацией сторонам
- если обменивающиеся информацией стороны уже используют некоторый общий ключ, одна из сторон может передать новый ключ другой стороне в шифрованном виде, используя старый ключ
- если обе обменивающиеся информацией стороны имеют защищенные каналы связи с третьей стороной, то последняя может доставить ключ обменивающимися информацией сторонам по этим защищенным каналам

Спасибо за внимание!

Контакты:
academy@it.ru
www.academy.it.ru