Занимательная информатика

Криптография

Манга

マンガでわかる

暗号

三谷 政昭・佐藤 伸一／共著
ひのき いでろう／作画
ウェルテ／制作

Ohmsha

ОБРАЗОВАТЕЛЬНАЯ МАНГА

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

КРИПТОГРАФИЯ

Митани Масааки, Сато Синъити
Хиноки Идэро

Перевод
А. Б. Клионского,
Научный редактор
Д. М. Белявский

Москва
ДМК Пресс, 2019

УДК 003.26
ББК 32.81
М66

Митани Масааки, Сато Синъити
М66 Занимательная информатика. Криптография. Манга / Митани Масааки, Сато Синъити (авторы), Хиноки Идэро (худож.); пер. с яп. Клионского А. Б., научн. ред. Д. М. Белявский. – М.: ДМК Пресс, 2019. – 238 с.: ил. – (Серия «Образовательная манга»). – Доп. тит. л. яп.

ISBN 978-5-97060-603-2

Из музея исскуств один за другим дерзко крадут ценные произведения, а преступник каждый раз оставляет зашифрованные сообщения. Проницательный инспектор Мэгуро, его сестра – математик Рика, – и эрудированная журналистка Ёнэда Рио бросают вызов дерзкому похитителю, но для этого им требуется разгадать загадку шифра. Книга познакомит читателя с общими понятиями криптологии и лежащими в её основе интересными математическими закономерностями, а также с тем, как криптография используется в нашей повседневной жизни.

УДК 003.26
ББК 32.81

Original Japanese edition
Manga de wakaru Ango (The Manga Guide to Cryptology)
By Mitani Masaaki and Sato Shinichi (Authors), Hinoki Idero (Illustrator) and Verte Corp. (Producer)

ISBN 978-4-274-06674-0 (яп.)
ISBN 978-5-97060-603-2 (рус.)

ПРЕДИСЛОВИЕ

Развитие информационного общества, ядром которого является сеть интернет, сделало нашу жизнь очень удобной, позволив свободно получать информацию, публикуемую на веб-сайтах, общаться по электронной почте, пользоваться услугами интернет-магазинов и интернет-банкинга.

Но, наслаждаясь этими удобствами, нам всем почему-то часто приходится слышать вызывающие некоторое беспокойство слова: «информационная безопасность», «защита личной информации» и, наконец, «шифрование». В чём же заключается проблема?

Дело в том, что пользоваться сетью – значит обмениваться по ней разнообразной информацией, в том числе и конфиденциальной, то есть такой, которую требуется держать в секрете. К ней относятся, например, номера кредитной карты и банковского счета, история болезни и кредитная история, адрес электронной почты и т. п. Попав в руки злоумышленников, такие сведения могут быть использованы для совершения различных преступлений, поэтому защита информации, несомненно, является главной задачей в области сетевых технологий. Основой для построения безопасных систем, предоставляющих разнообразные сетевые услуги с надёжной аутентификацией (установлением подлинности) данных, защитой от спуфинга (злонамеренных действий под видом законных пользователей), перехвата информации и фальсификации данных является шифрование.

За последние годы в развитии криптографии※ произошёл огромный скачок: она перестала быть уделом только специалистов по информационной безопасности и прочно вошла в жизнь обычных людей, пользующихся услугами информационных сетей.

Каким же образом шифрование обеспечивает информационную безопасность и защиту личной информации?

В этой книге на основе манги описываются механизмы шифрования и его роль в нашей жизни. Объяснения сложных математических понятий, без которых понимание криптологии невозможно, даются в легком для понимания виде, поэтому вы сможете освоить их без особого напряжения, просто следя за развитием сюжета. В самом повествовании, конечно же, тоже заложен шифр, разгадав который, читатель получит дополнительное удовольствие. Надеюсь, что эта книга поможет вам овладеть базовыми знаниями в области криптологии※ и информационной безопасности.

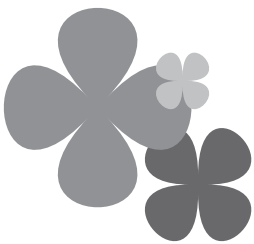
В завершение хотим поблагодарить коллектив Отдела разработок издательства Ohmsha и художника Хиноки Идэро, рисовавшего мангу.

Апрель 2007 *Авторы*

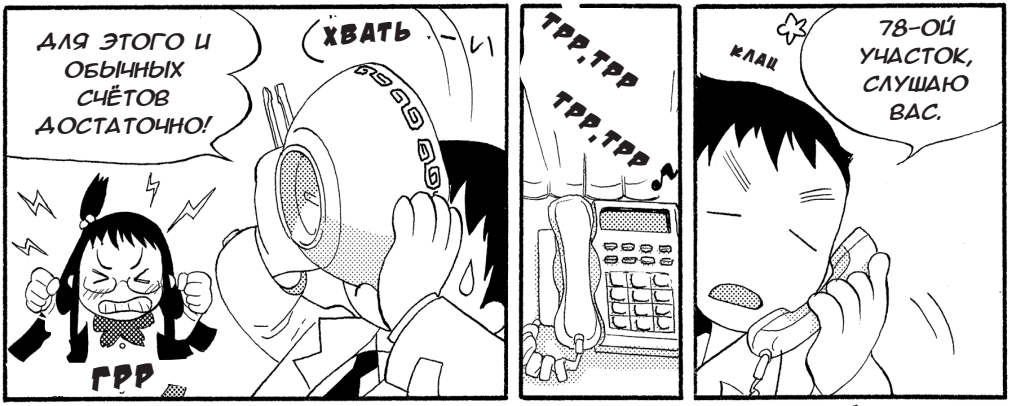
※ Криптография – раздел криптологии, в котором изучают собственно методы шифрования. В другом разделе криптологии – криптоанализе, – занимаются поиском уязвимости шифров.

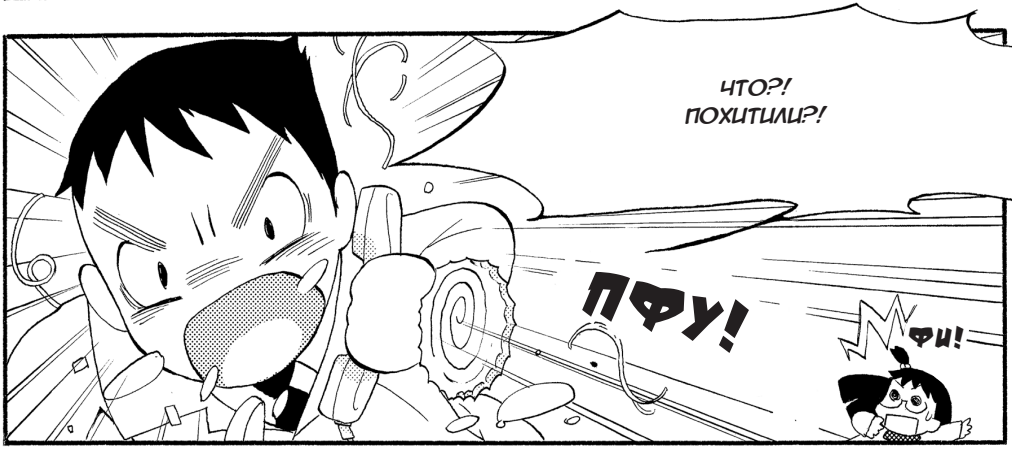
СОДЕРЖАНИЕ

ПРОЛОГ

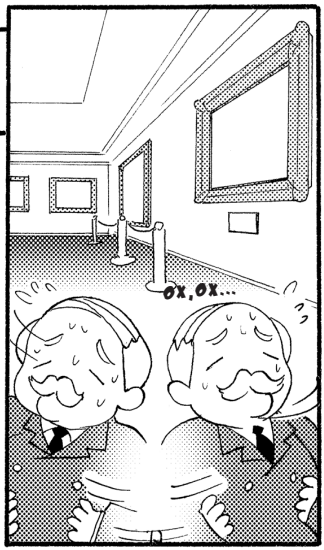


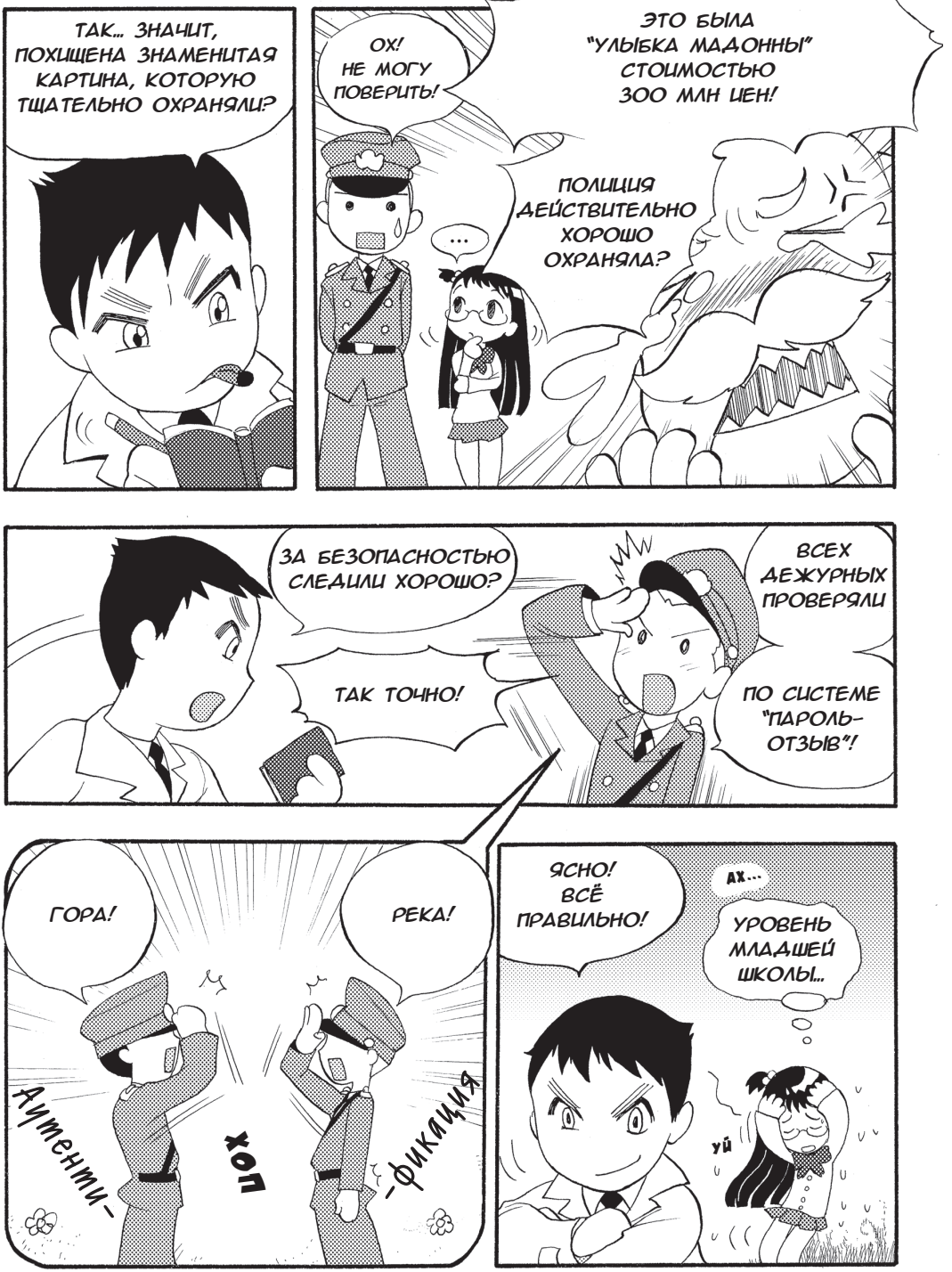

Полицейский участок № 78
в каком-то городе
Безопасность дорожного движения
отдел
.....
Приёмная
Мегуро Рика
БРАТЕЦ, НУ КУПИ!
Мегуро
помощник инспектора
НЕТ! КОМПЬЮТЕР ДЛЯ ШКОЛЬНИЦЫ – СЛИШКОМ БОЛЬШАЯ РОСКОШЬ!
ФШШ
ОН НУЖЕН МНЕ ДЛЯ ИЗУЧЕНИЯ МАТЕМАТИКИ!
$ax + by = gcd(a,b)$
$a^{p-1} \equiv 1 \pmod{p}$
$\varphi(p) = p - 1$
x
y^2
.....
ХОЧУ!


ДЛЯ ЭТОГО И ОБЫЧНЫХ СЧЁТОВ ДОСТАТОЧНО!
ХВАТЬ
ГРР
ТРР,ТРР
ТРР,ТРР
КЛАЦ
78-ОЙ УЧАСТОК, СЛУШАЮ ВАС.

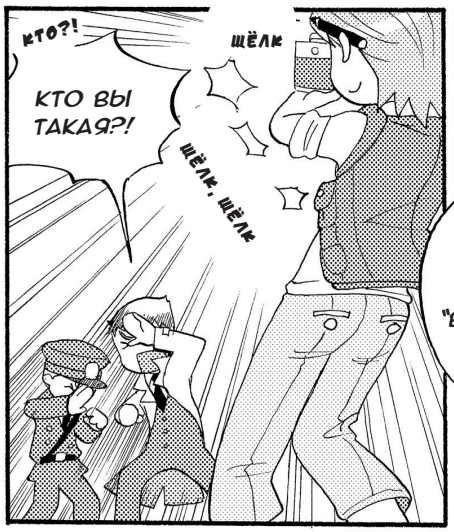

ЧТО?! ПОХИТИЛИ?!
ПФУ!
ФИ!


Музей исскуств «Мраморный»
ВИУ
ВИУ
ПРОВЕРЬТЕ ЗАПАДНОЕ КРЫЛО!
ПЕРЕКРОЙТЕ ВОН ТОТ ВЫХОД!
ВОСТОЧНОЕ КРЫЛО!

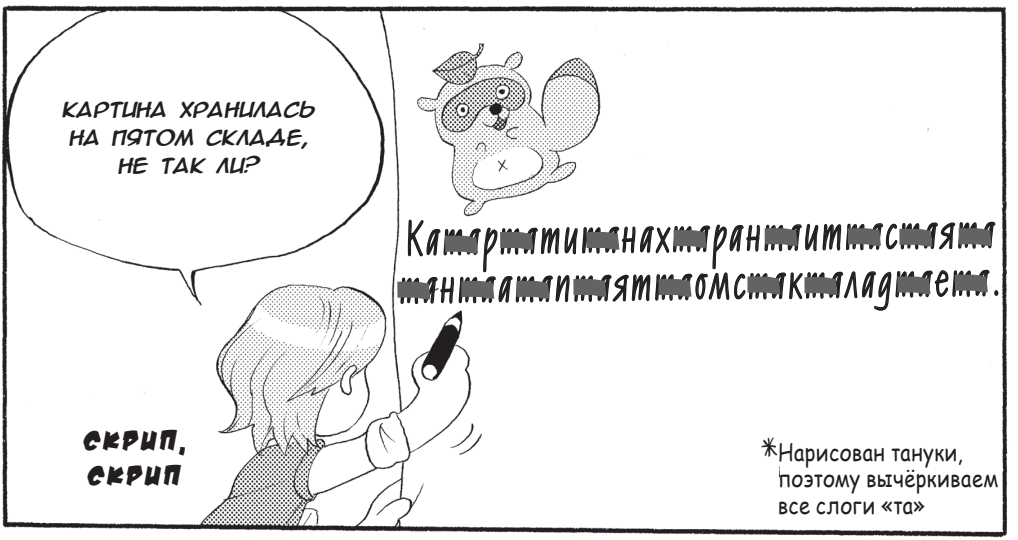

ОХ,ОХ...

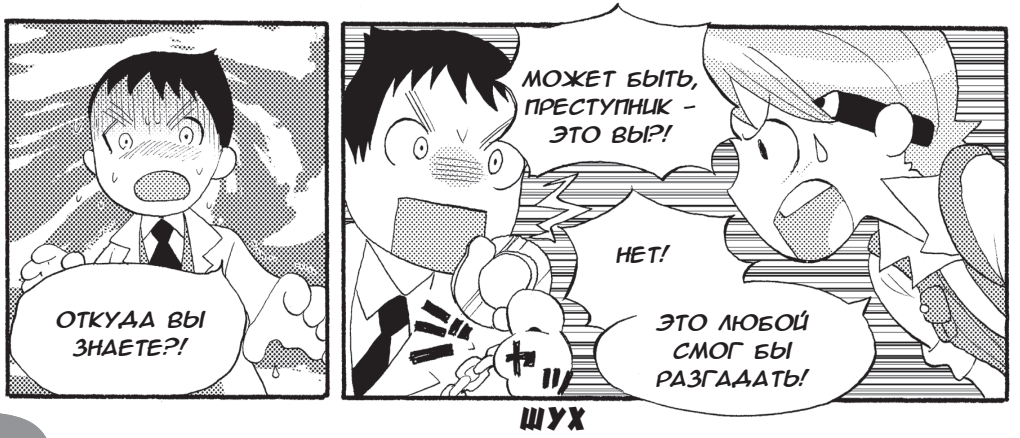

ТАК... ЗНАЧИТ, ПОХИЩЕНА ЗНАМЕНИТАЯ КАРТИНА, КОТОРУЮ ТЩАТЕЛЬНО ОХРАНЯЛИ?
ОХ! НЕ МОГУ ПОВЕРИТЬ!
ЭТО БЫЛА "УЛЫБКА МАДОННЫ" СТОИМОСТЬЮ 300 МЛН ЕН!
ПОЛИЦИЯ ДЕЙСТВИТЕЛЬНО ХОРОШО ОХРАНЯЛА?
...
ЗА БЕЗОПАСНОСТЬЮ СЛЕДИЛИ ХОРОШО?
ТАК ТОЧНО!
ВСЕХ ДЕЖУРНЫХ ПРОВЕРЯЛИ
ПО СИСТЕМЕ "ПАРОЛЬ-ОТЗЫВ"!
ГОРА!
РЕКА!
Аутентификация
ЯСНО! ВСЁ ПРАВИЛЬНО!
АХ...
УРОВЕНЬ МЛАДШЕЙ ШКОЛЫ...
УЙ

※ Изображена енотовидная собака тануки.

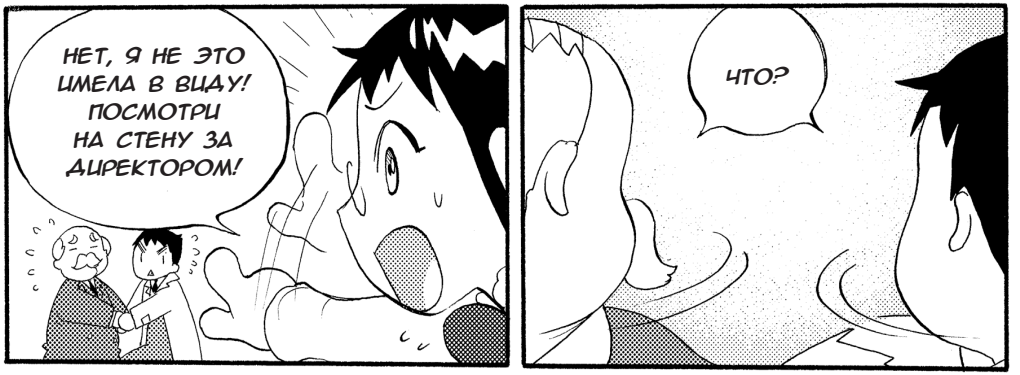


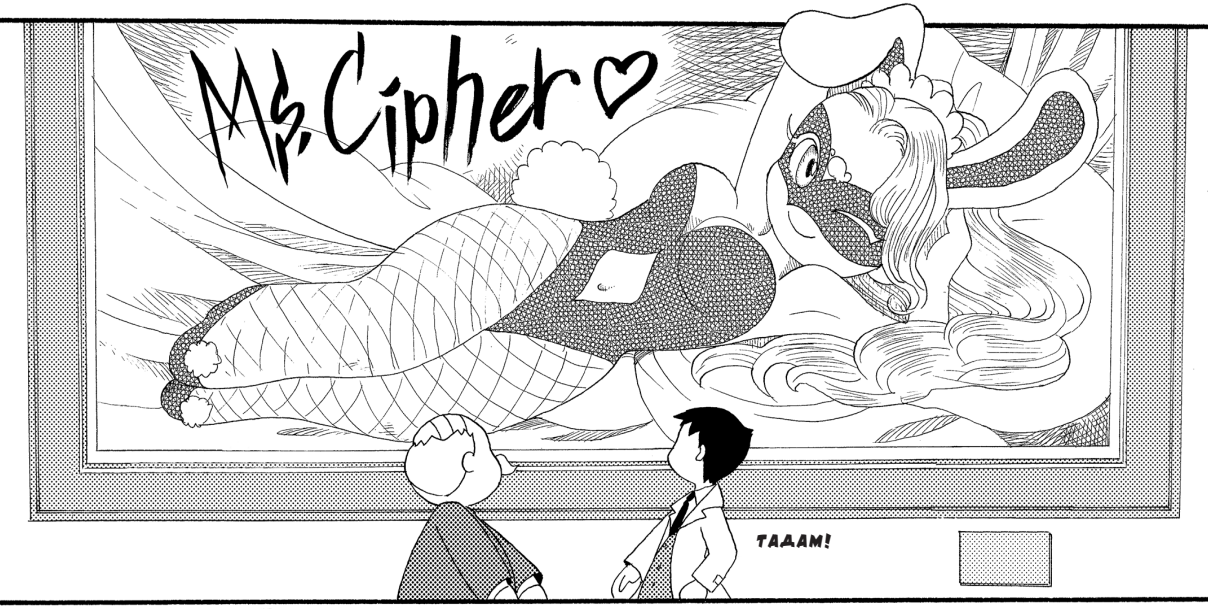

КАК?! ГАЗЕТЧИКИ УЖЕ ВСЁ ПРОНЮХАЛИ?
ЧТО БЫЛО НЕ ТАК С БЕЗОПАСНОСТЬЮ?!
ПАРОЛЬ И ШИФР НИКУДА НЕ ГОДЯТСЯ!
ЧТО?! ПОЧЕМУ?!

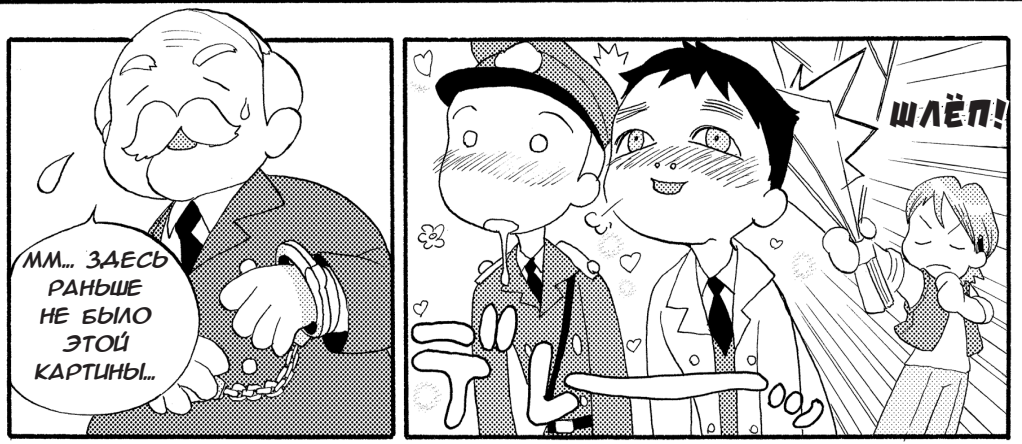

КАРТИНА ХРАНИЛАСЬ НА ПЯТОМ СКЛАДЕ, НЕ ТАК ЛИ?
Катартинанахранитастаятанатапятомскладтета.
СКРИП, СКРИП
*Нарисован тануки, поэтому вычёркиваем все слоги «та»


ОТКУДА ВЫ ЗНАЕТЕ?!
МОЖЕТ БЫТЬ, ПРЕСТУПНИК – ЭТО ВЫ?!
НЕТ!
ЭТО ЛЮБОЙ СМОГ БЫ РАЗГАДАТЬ!
ШУХ


КАК? НЕУЖЕЛИ?
ИНСПЕКТОР МЭГУРО, НЕ ТРАТЬТЕ ВРЕМЯ! ПОСКОРЕЕ НАЙДИТЕ ПРЕСТУПНИКА!
ГРР...
ДА... ПРЕСТУПНИКА..
ВОТ ОН, ПРЕСТУПНИК!
ЧТО?!
ДИРЕКТОР, ВЫ АРЕСТОВАНЫ!
ЩЁЛК!
ОЙ!

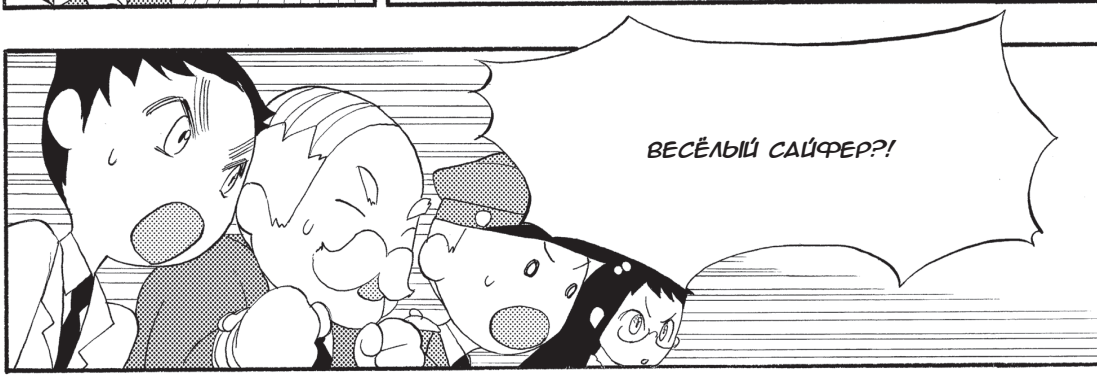

НЕТ, Я НЕ ЭТО ИМЕЛА В ВИДУ! ПОСМОТРИ НА СТЕНУ ЗА ДИРЕКТОРОМ!
ЧТО?


Ms. Cipher♡
ТАДАМ!


ММ... ЗДЕСЬ РАНЬШЕ НЕ БЫЛО ЭТОЙ КАРТИНЫ...
ШЛЁП!
ГЛЫГ

Я – Весёлый сайфер.
Это я украла картину.
В следующий раз
украду VDVIRCU.

Спокойной ночи ♥




Это я украла картину. В следующий раз украду VDVIRCU.
ЭТО ЖЕ ШИФР! УКАЗАНА ВЕЩЬ, КОТОРАЯ БУДЕТ УКРАДЕНА СЛЕДУЮЩЕЙ.
ЧТО ОЗНАЧАЕТ ЭТО VDVIRCU?
У МЕНЯ С АНГЛИЙСКИМ НЕ ОЧЕНЬ...
ЭХ...
НО ЭТО ЯВНО НЕ ШИФР "ТАНУКИ": ВЫЧЁРКИВАНИЕ БУКВ НЕ ДАЁТ НИЧЕГО ОСМЫСЛЕННОГО.
ДАВАЙТЕ ТОГДА ИЗУЧИМ КРИПТОЛОГИЮ И ПОКАЖЕМ ЭТОМУ ВЕСЁЛОМУ САЙФЕРУ!
(ГОРЯЧО)
ЭТО ЖЕ НЕ ШПИОНСКИЙ РОМАН.... КАКОЙ НАМ ПРОК ОТ ЭТИХ ШИФРОВ?

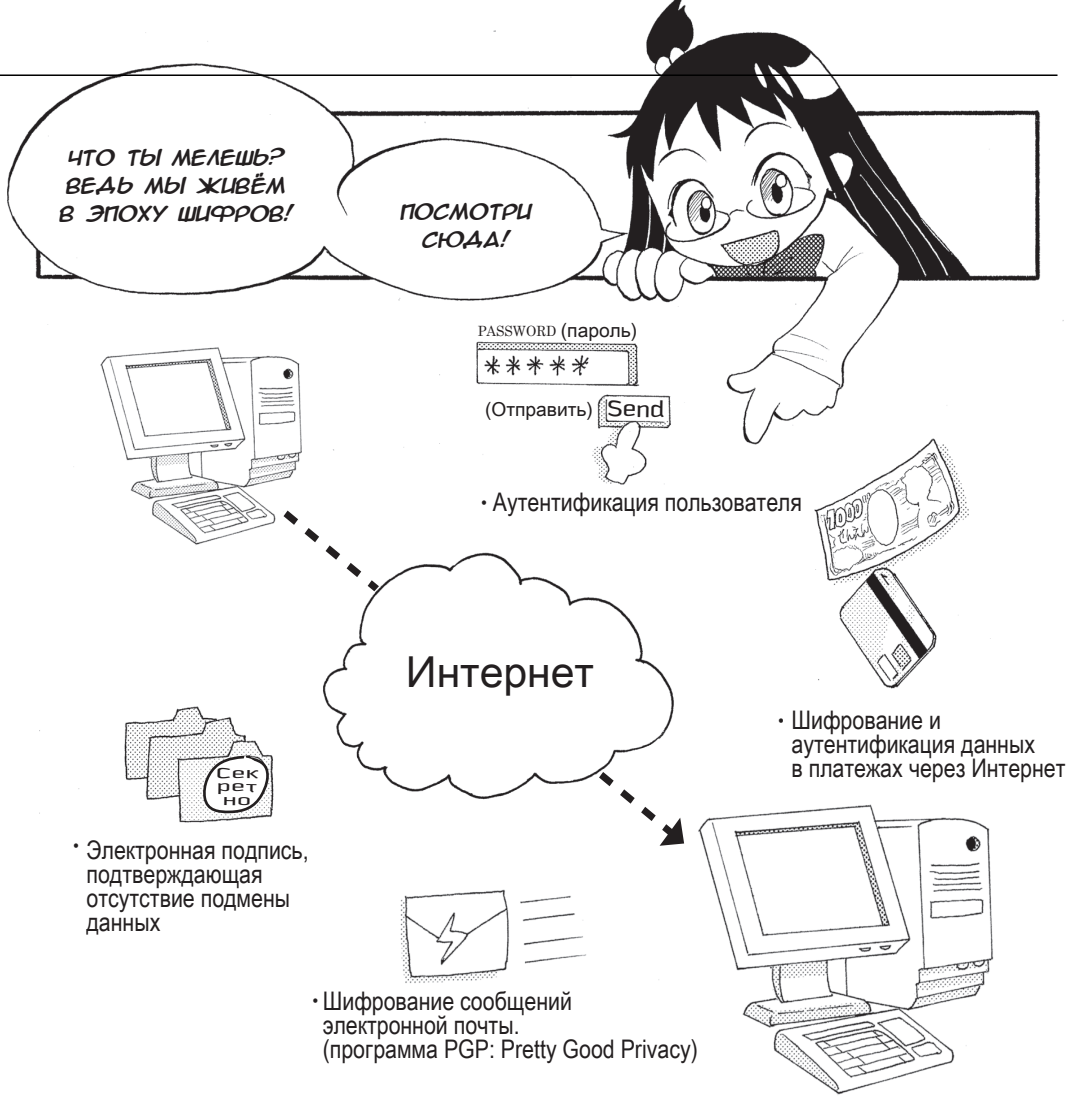
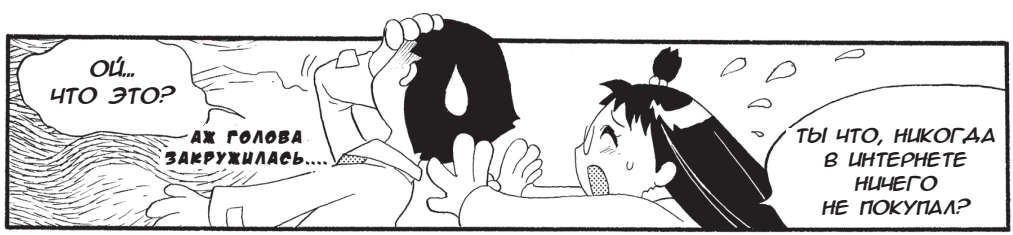


Рис. 0.1. Роль криптографии в современном обществе

Как показано на рис. 0.1, в нашу эпоху компьютеров и связи шифрование незаменимо для борьбы с подменой данных, перехвата информации и т. п.



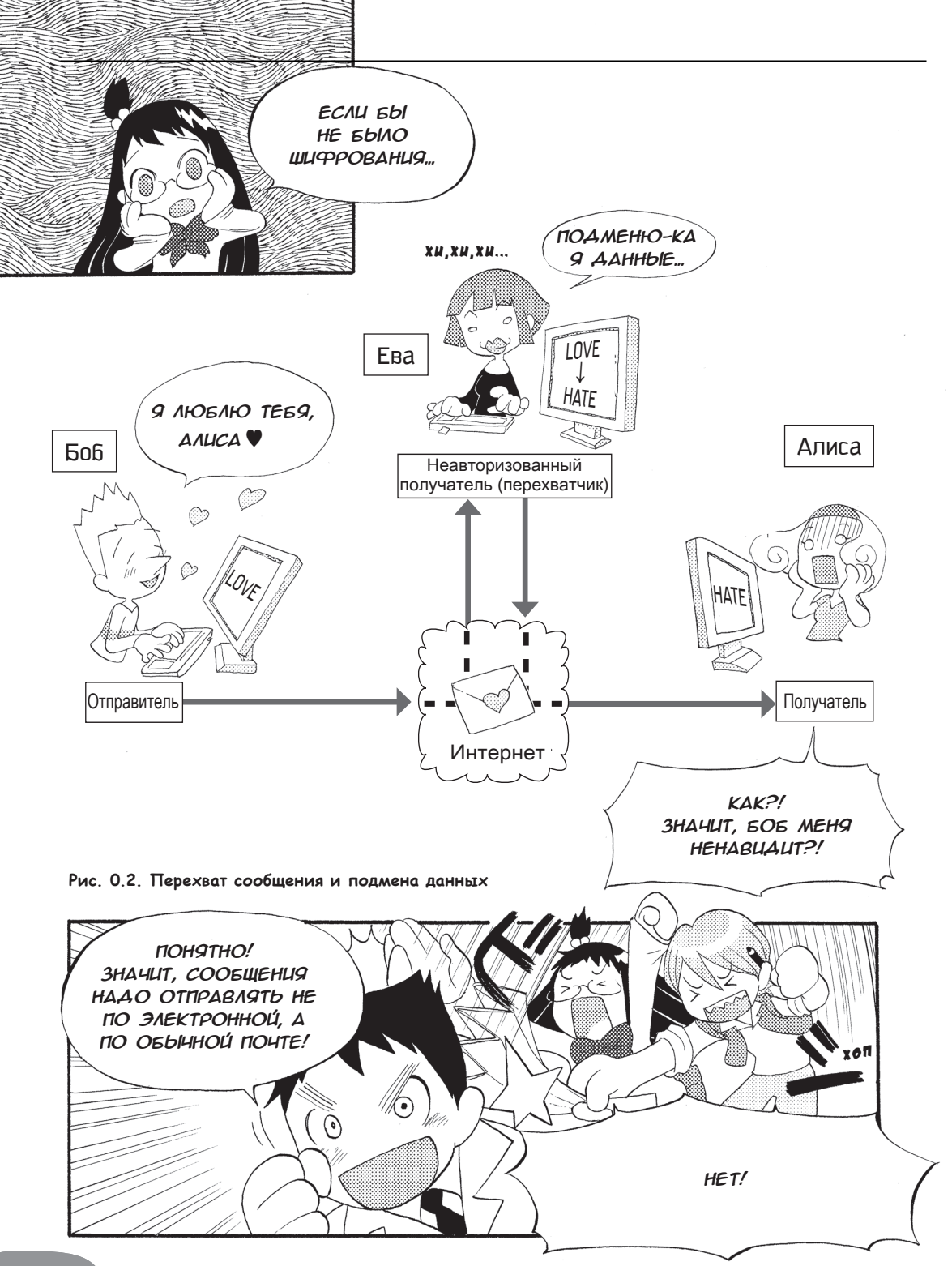


Рис. 0.2. Перехват сообщения и подмена данных

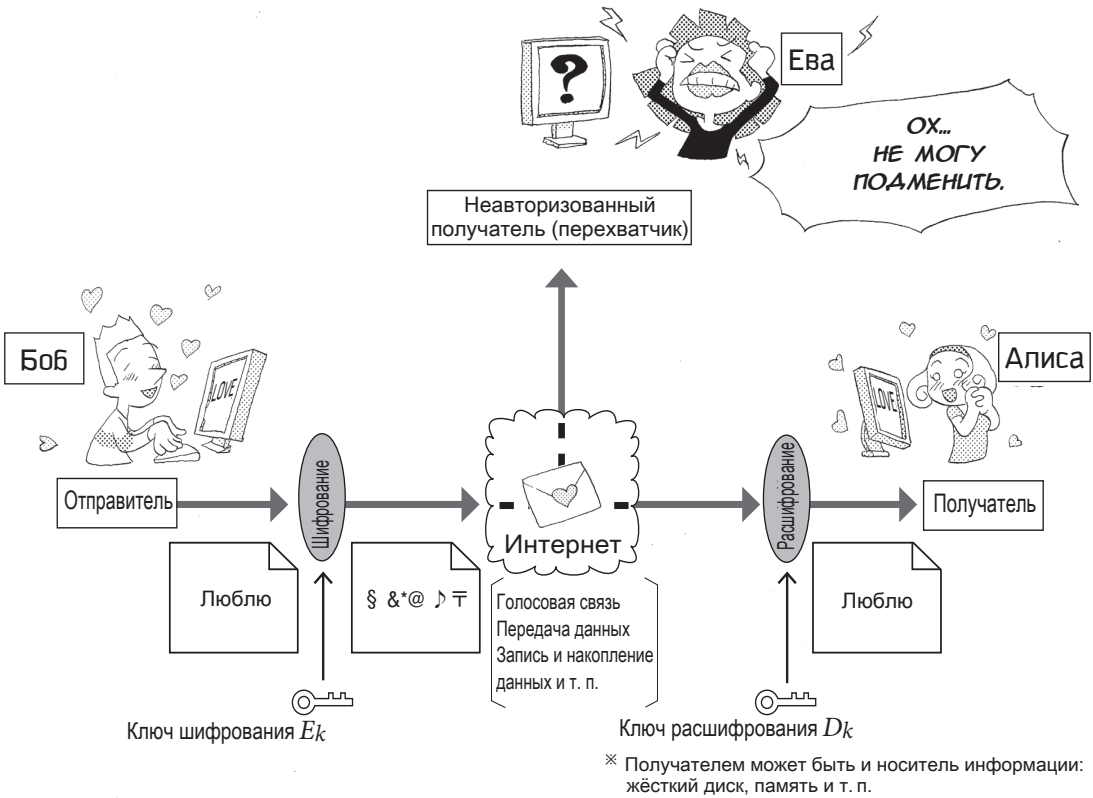
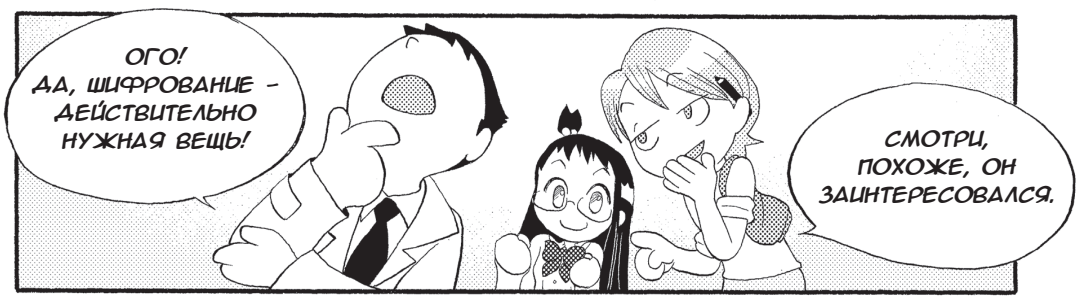


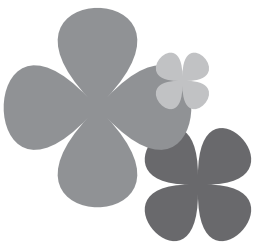
Рис. 0.3. Модель шифрования (криптосистема)



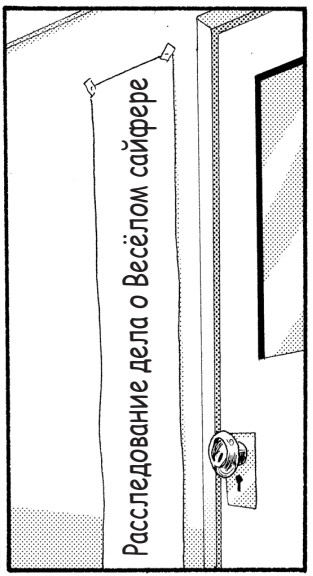

ЕСЛИ ХОЧЕШЬ,
Я МОГУ ПРЕПОДАВАТЬ
ТЕБЕ ШИФРОВАНИЕ!
ДА, ХОЧУ!
Я ИЗУЧУ ШИФРОВАНИЕ
И ПОЙМАЮ САЙФЕРА!
ИНСПЕКТОР...
УВАЖАЕМЫЙ ДИРЕКТОР,
НЕ БЕСПОКОЙТЕСЬ!
ПРЕСТУПЛЕНИЕ СКОРО
БУДЕТ РАСКРЫТО!
ОХ...
СНИМИТЕ
С МЕНЯ
НАРУЧНИКИ!

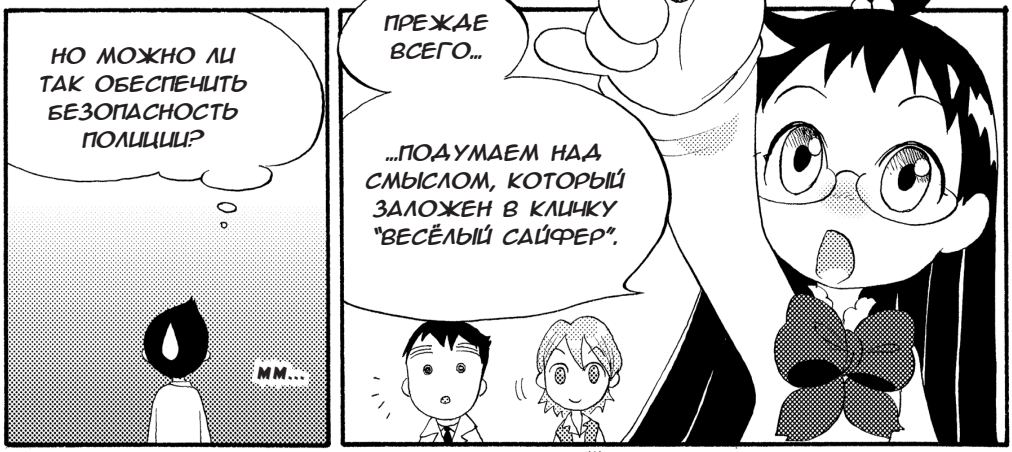
ГЛАВА 1

ОСНОВЫ КРИПТОГРАФИИ

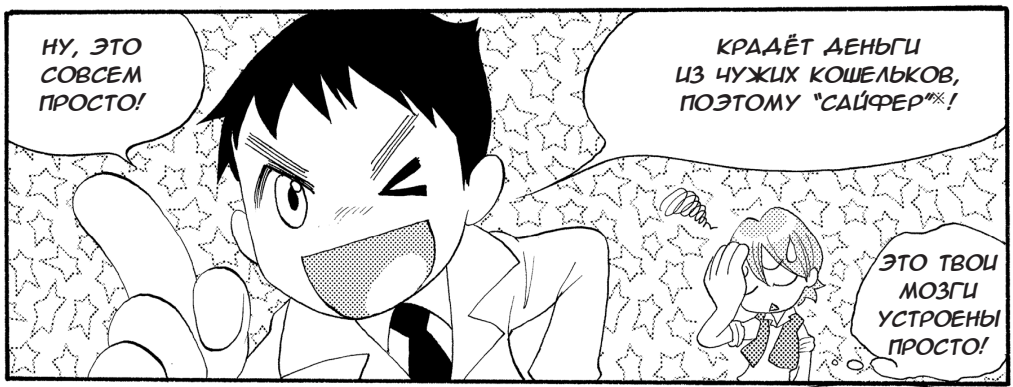


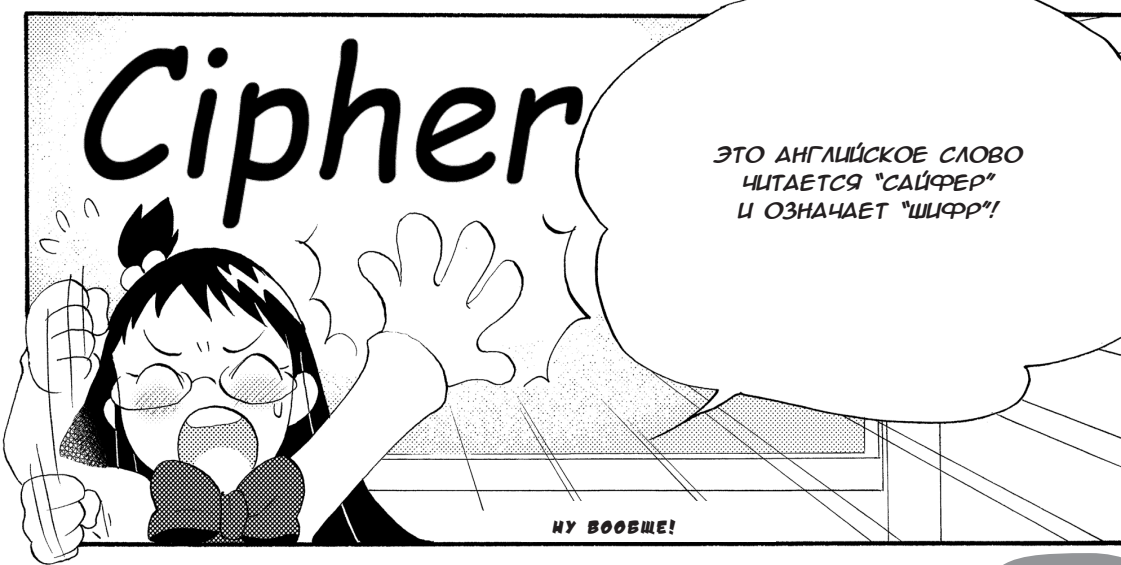
1-1 Основные понятия криптографии





※ По-японски «сайфу» означает «кошелёк».





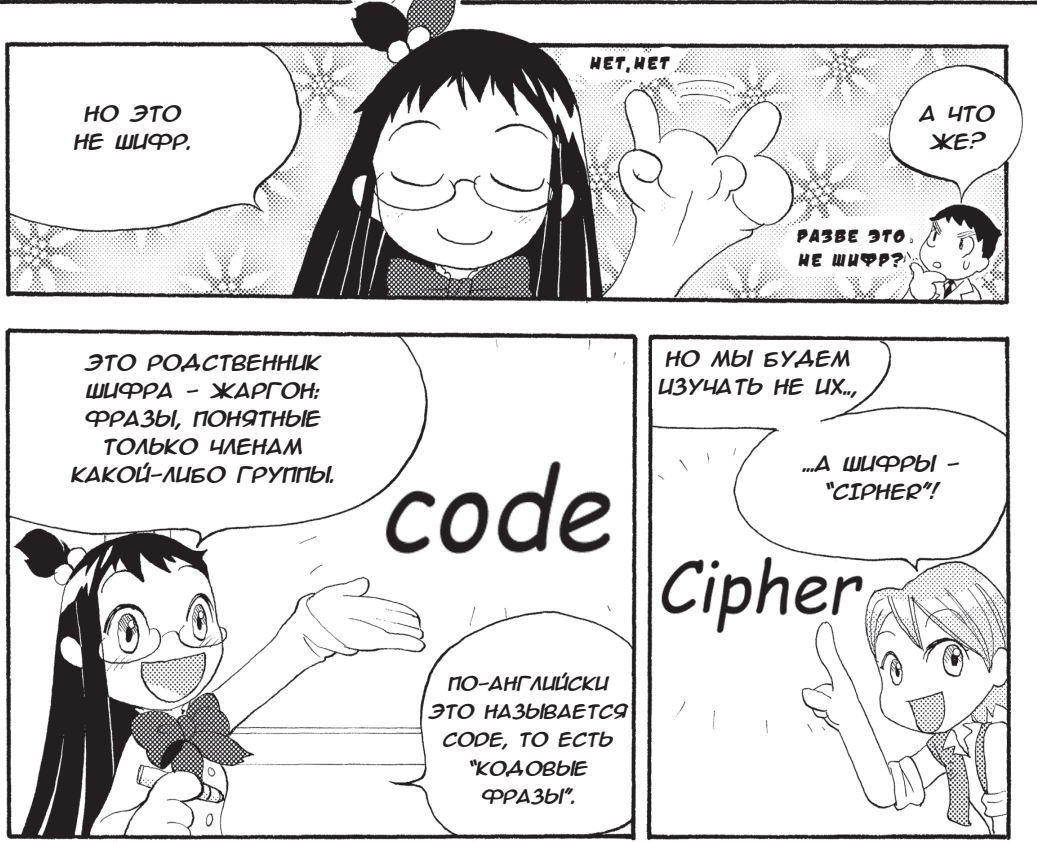
«Ниитака-яма ноборэ» ※1 = «Начать атаку»

«Тора тора тора» ※2 = «Атака была успешной»

※1 Дословно: «Поднимайтесь на гору Ниитака».

※2 Дословно: «Тигр, тигр, тигр».

※ Обе кодовые фразы использовались Императорским флотом Японии при нападении на Пёрл-Харбор во время Второй мировой войны.



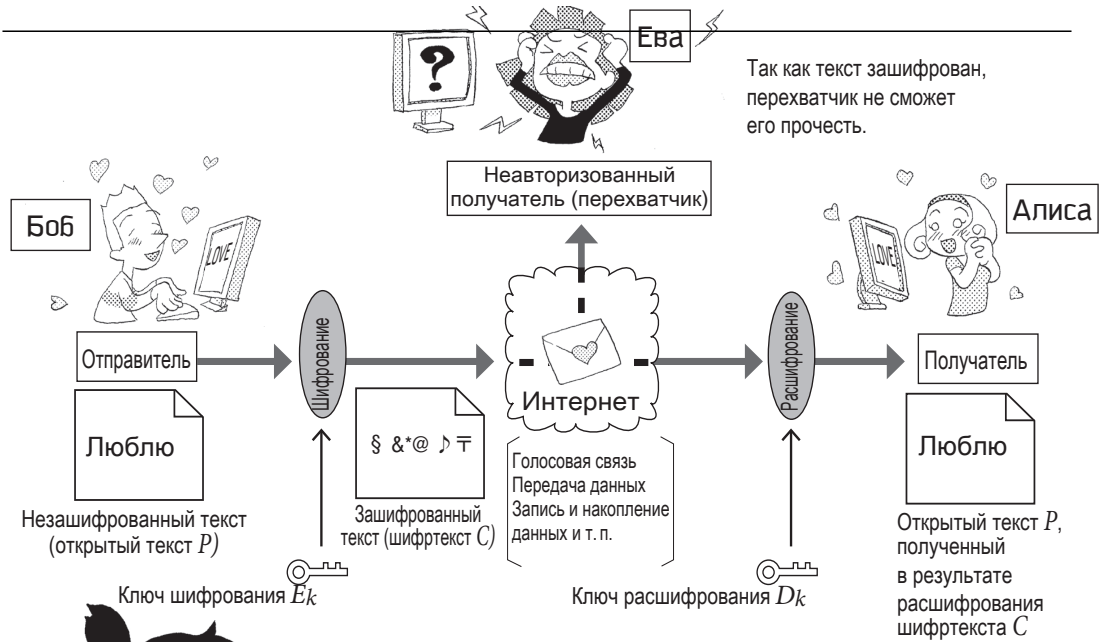
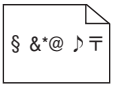


Рис. 1.1. Модель шифрования (криптосистема, шифр)

✤ Термины криптографии

Открытый текст *P* (Plain text): обычный незашифрованный текст.

Шифртекст *C* (Cipher text): зашифрованный текст. Другое название – криптограмма.



Шифрование (Encryption/Encipherment): преобразование открытого текста в шифртекст.

Расшифрование (Decryption/Decipherment): преобразование шифртекста в открытый текст.

Ключ шифрования E_k (Encryption Key): ключ, используемый для шифрования.

Ключ расшифрования D_k (Decryption Key): ключ, используемый для расшифрования.

✽ Связь между ключами E_k и D_k

Отправитель зашифровывает открытый текст: используя открытый текст P и ключ шифрования E_k (функцию шифрования), он генерирует шифртекст C.

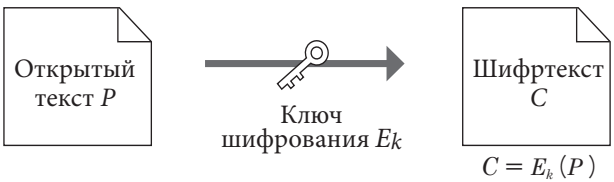


Рис. 1.2. Шифрование с использованием ключа E_k

Получатель расшифровывает шифртекст: используя шифртекст C и ключ расшифрования D_k (функцию расшифрования), он генерирует открытый текст P.

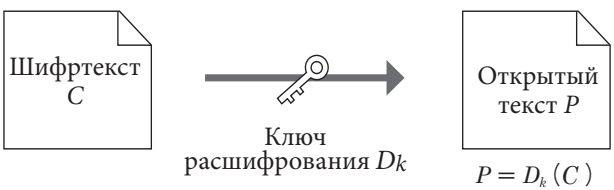
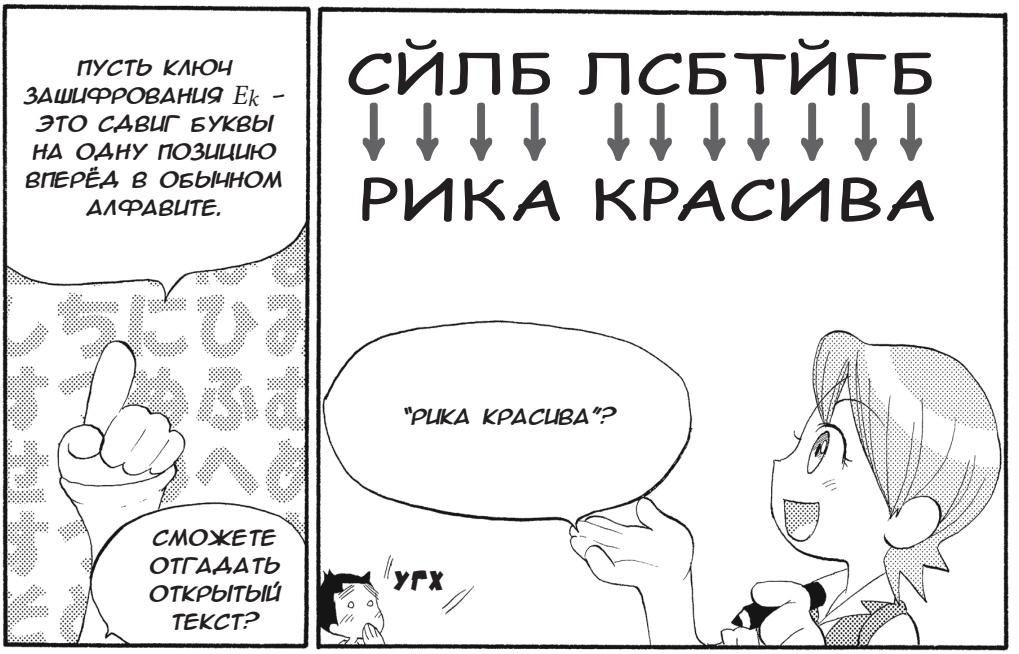
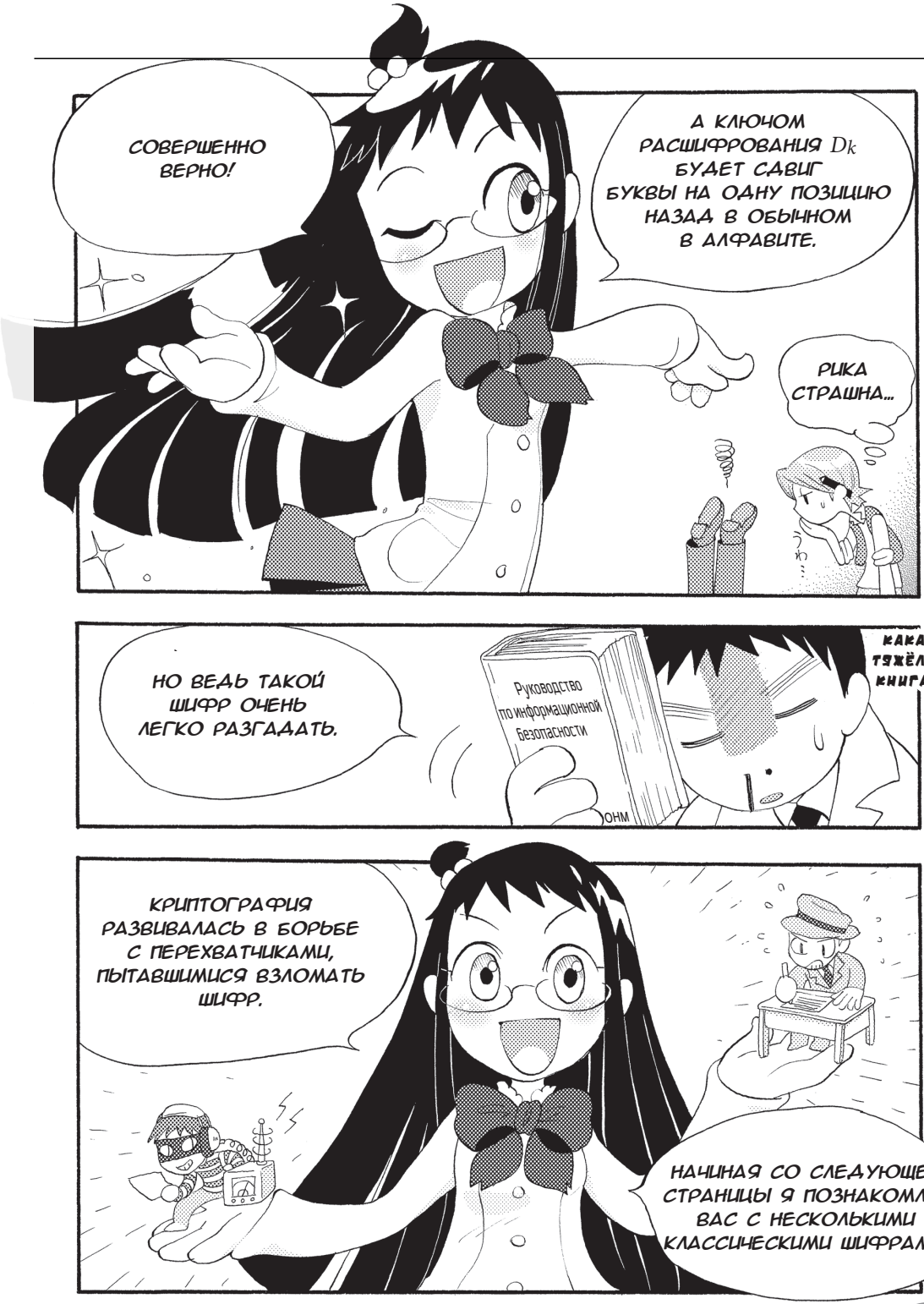


Рис. 1.3. Расшифрование с использованием ключа D_k


А ТЕПЕРЬ – ЗАДАЧКА!
СЙЛБ ЛСБТЙГБ
СКРИП, СКРИП


ПУСТЬ КЛЮЧ ЗАШИФРОВАНИЯ Ek – ЭТО СДВИГ БУКВЫ НА ОДНУ ПОЗИЦИЮ ВПЕРЁД В ОБЫЧНОМ АЛФАВИТЕ.
СМОЖЕТЕ ОТГАДАТЬ ОТКРЫТЫЙ ТЕКСТ?
СЙЛБ ЛСБТЙГБ
РИКА КРАСИВА
"РИКА КРАСИВА"?
УГХ


ТУДУХ
ХА,ХА,ХА! "РИКА КРАСИВА" – ЭТО ОЧЕВИДНАЯ ВСЕМ ОШИБ...
БАХ
РУ!

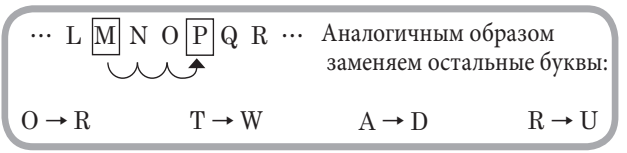

СОВЕРШЕННО ВЕРНО!
А КЛЮЧОМ РАСШИФРОВАНИЯ D_k БУДЕТ СДВИГ БУКВЫ НА ОДНУ ПОЗИЦИЮ НАЗАД В ОБЫЧНОМ В АЛФАВИТЕ.
РИКА СТРАШНА...
НО ВЕДЬ ТАКОЙ ШИФР ОЧЕНЬ ЛЕГКО РАЗГАДАТЬ.
Руководство по информационной безопасности
ОНМ
КАКАЯ ТЯЖЁЛАЯ КНИГА...
КРИПТОГРАФИЯ РАЗВИВАЛАСЬ В БОРЬБЕ С ПЕРЕХВАТЧИКАМИ, ПЫТАВШИМИСЯ ВЗЛОМАТЬ ШИФР.
НАЧИНАЯ СО СЛЕДУЮЩЕЙ СТРАНИЦЫ Я ПОЗНАКОМЛЮ ВАС С НЕСКОЛЬКИМИ КЛАССИЧЕСКИМИ ШИФРАМИ.

1-2 Классические шифры

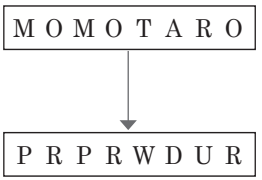
❀ Шифр Цезаря

Каждая буква открытого текста заменяется на букву, сдвинутую относительно неё на *n* позиций в алфавите. В качестве примера попробуем зашифровать слово MOMOTARO.

Примем $n = 3$.

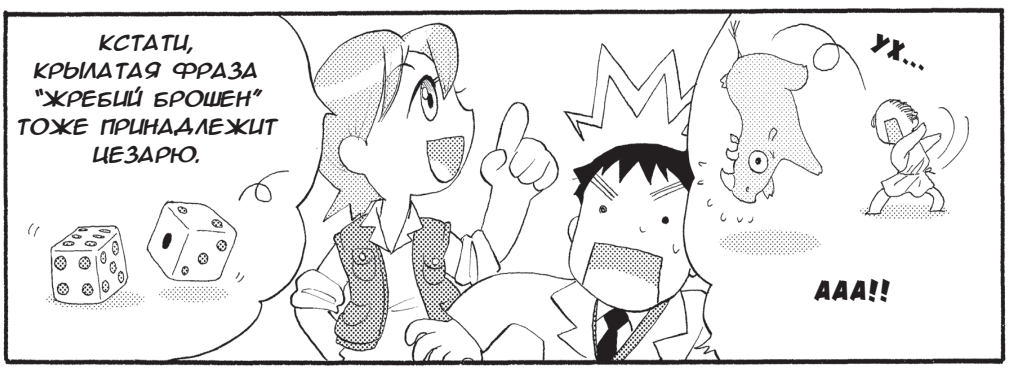


Таким образом, у нас получится шифртекст.

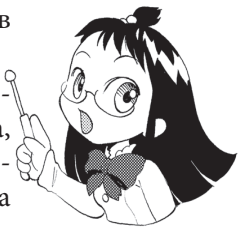


Последним трём буквам алфавита соответствуют первые.

Цезарь – это древнеримский полководец и политик Гай Юлий Цезарь (100 г. до н. э. – 44 г. до н. э.), придумавший этот шифр во время Галльской войны для обмена с союзниками сообщениями, которые не мог прочесть неприятель.



❀ Шифр одноалфавитной замены



Если немного усложнить шифр Цезаря, изменяя сдвиг в зависимости от буквы, то мы получим шифр замены.

Шифр замены, в котором есть взаимно-однозначное соответствие между буквами открытого текста и шифртекста, называется шифром одноалфавитной (или простой) замены. Шифр Цезаря тоже является разновидностью шифра одноалфавитной замены.

Положим, что 26 букв английского алфавита заменяются так, как показано ниже.

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

Правило замены σ («сигма»)

Тогда шифрование будет осуществляться следующим образом.

M	O	M	O	T	A	R	O

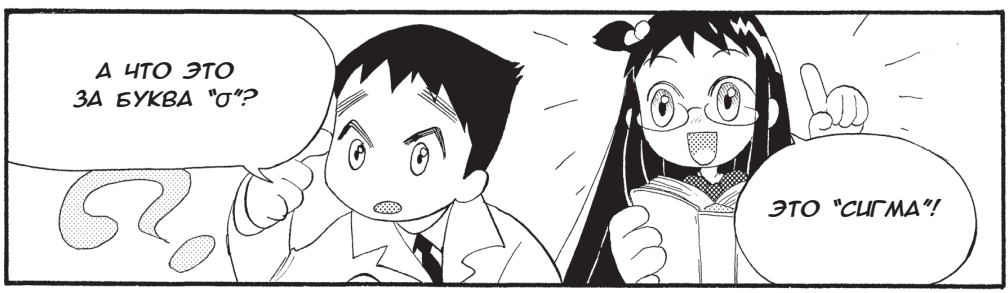
(Открытый текст *P)*

↓ Заменяем буквы по правилу замены σ

D	G	D	G	Z	Q	K	G

(Шифртекст *C)*

В этом шифре алгоритмом является замена букв, а ключом шифрования E_k – правило замены σ.



❖ Шифр многоалфавитной замены (шифр Виженера)

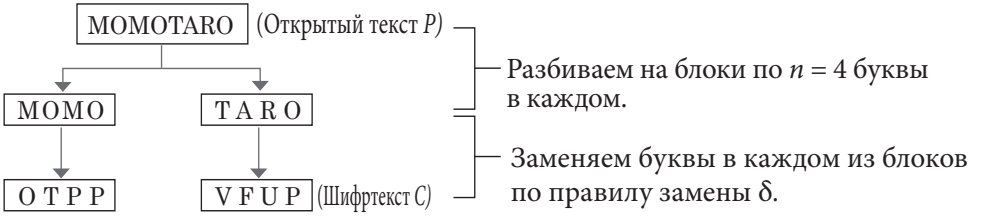
Разбив открытый текст на блоки по n букв в каждом, изменяют величину сдвига в зависимости от позиции каждой буквы внутри блока. Можно сказать, что шифр Виженера является расширением шифра Цезаря.

Положим $n = 4$ и определим правило замены δ следующим образом.

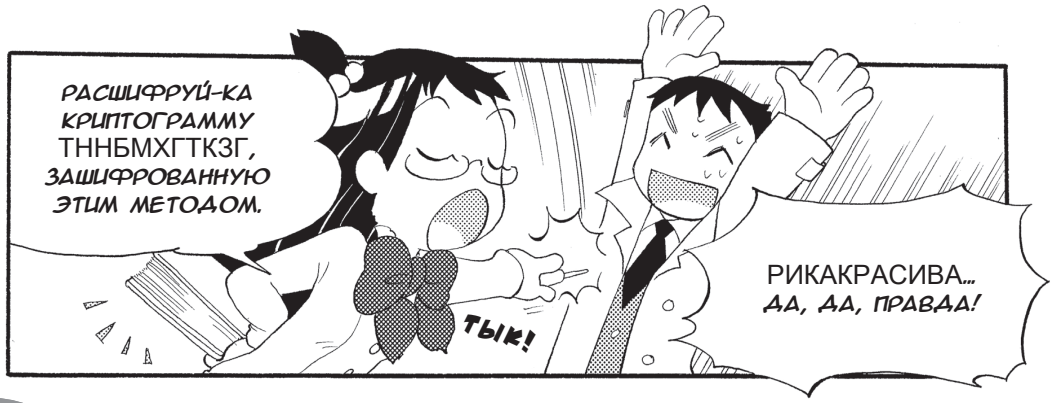
1-я буква → сдвиг = 2
2-я буква → сдвиг = 5
3-я буква → сдвиг = 3
4-я буква → сдвиг = 1

Правило замены δ

В этом случае мы получим следующий шифртекст.



В данном шифре ключ шифрования – это длина блока и правило замены, то есть последовательность величин сдвига.



✤ Шифр перестановки

Разбив открытый текст на блоки по n букв в каждом, меняют местами буквы в каждом из блоков.

Положим $n = 4$ и определим правило перестановки τ следующим образом.

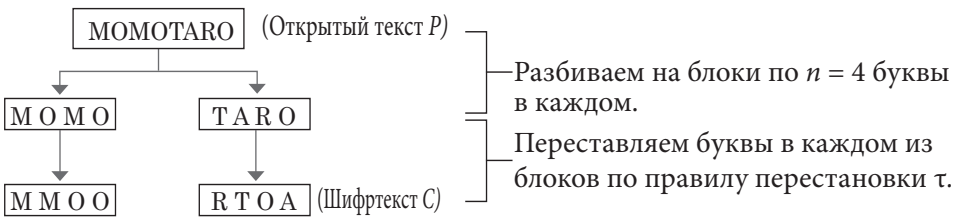
$$\tau = \begin{pmatrix} 1234 \\ 2413 \end{pmatrix}$$

Верхняя формула означает, что перестановка осуществляется следующим образом.

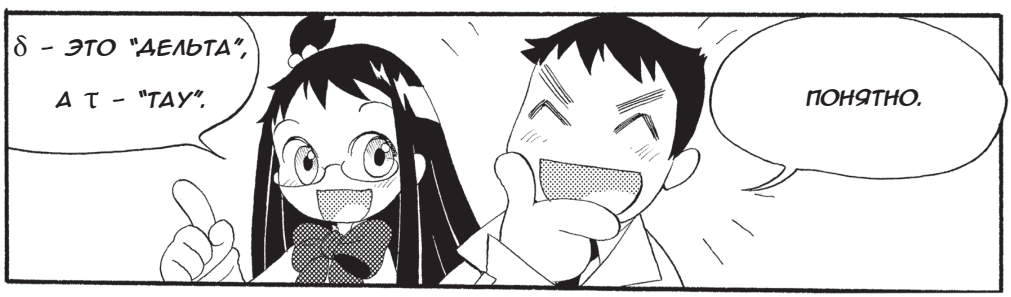
1-я буква → 2-я буква
2-я буква → 4-я буква
3-я буква → 1-я буква
4-я буква → 3-я буква

Правило перестановки τ

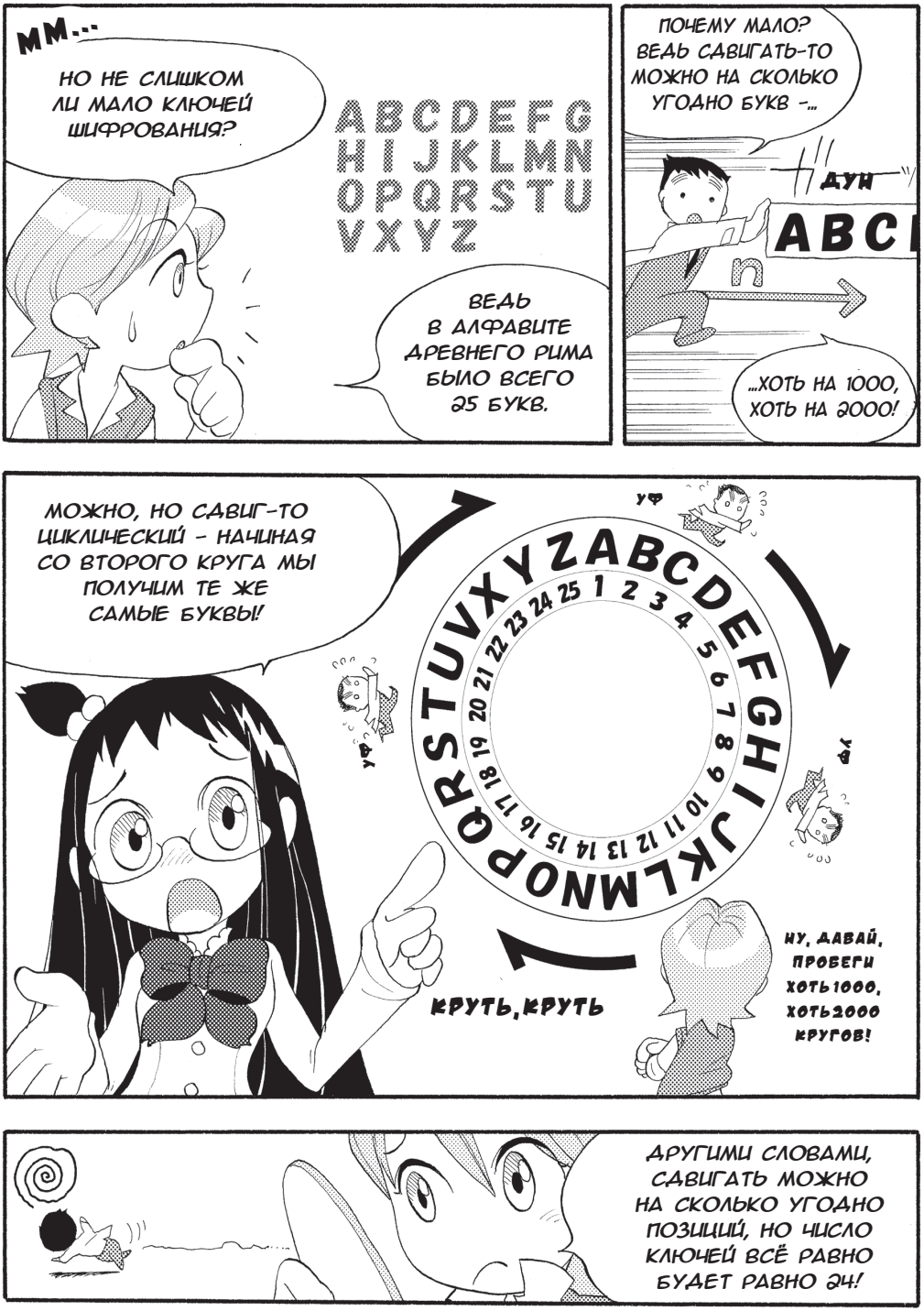
В этом случае мы получим следующий шифртекст.

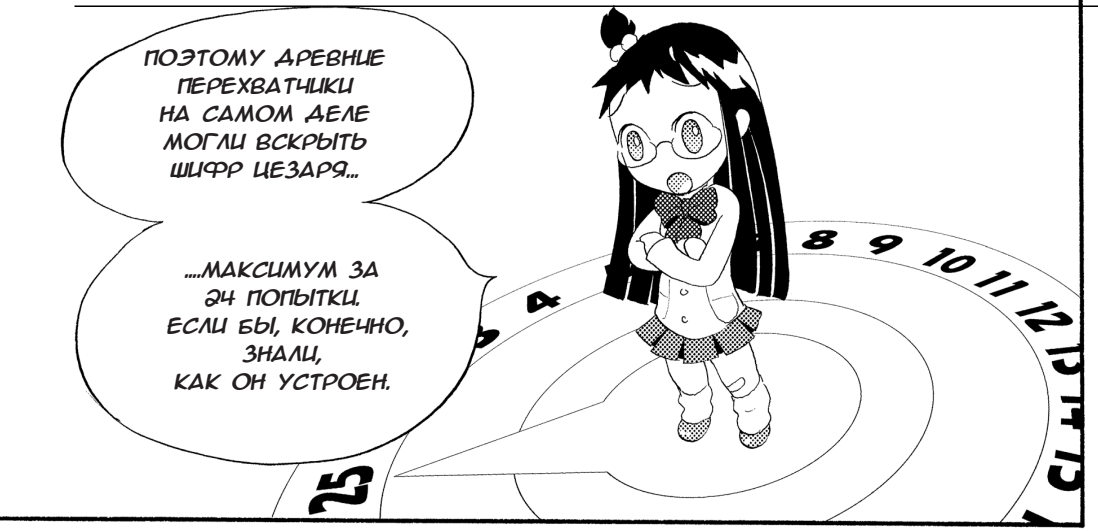


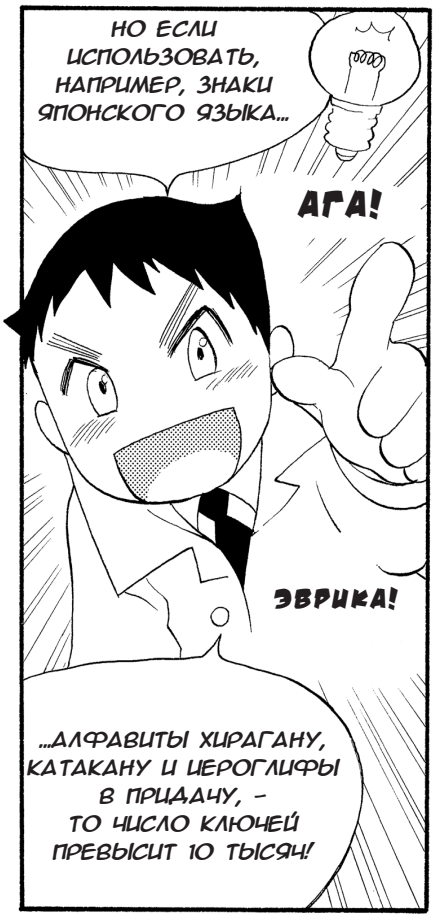
В данном шифре алгоритм шифрования – изменение порядка следования букв, а ключ шифрования – длина блока и правило перестановки.

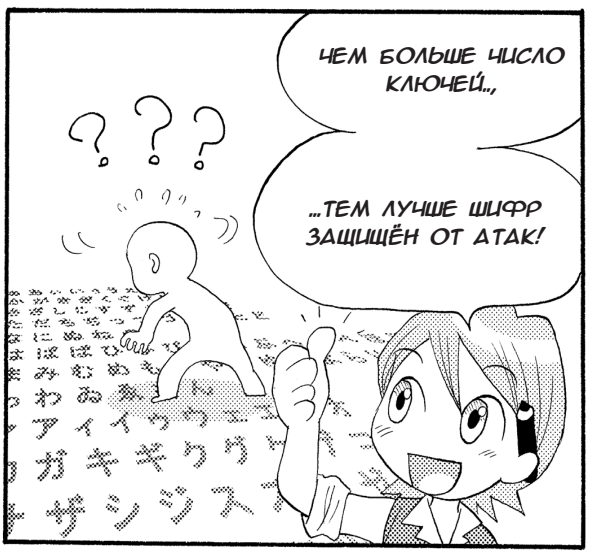


1-3 Стойкость шифра


ММ...
НО НЕ СЛИШКОМ ЛИ МАЛО КЛЮЧЕЙ ШИФРОВАНИЯ?
A B C D E F G
H I J K L M N
O P Q R S T U
V X Y Z
ВЕДЬ В АЛФАВИТЕ ДРЕВНЕГО РИМА БЫЛО ВСЕГО 25 БУКВ.
ПОЧЕМУ МАЛО? ВЕДЬ СДВИГАТЬ-ТО МОЖНО НА СКОЛЬКО УГОДНО БУКВ -...
ДУН
A B C
...ХОТЬ НА 1000, ХОТЬ НА 2000!
МОЖНО, НО СДВИГ-ТО ЦИКЛИЧЕСКИЙ - НАЧИНАЯ СО ВТОРОГО КРУГА МЫ ПОЛУЧИМ ТЕ ЖЕ САМЫЕ БУКВЫ!
УФ
A B C D E F G H I J K L M N O P Q R S T U V X Y Z
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
КРУТЬ, КРУТЬ
НУ, ДАВАЙ, ПРОБЕГИ ХОТЬ 1000, ХОТЬ 2000 КРУГОВ!
ДРУГИМИ СЛОВАМИ, СДВИГАТЬ МОЖНО НА СКОЛЬКО УГОДНО ПОЗИЦИЙ, НО ЧИСЛО КЛЮЧЕЙ ВСЁ РАВНО БУДЕТ РАВНО 24!


ПОЭТОМУ ДРЕВНИЕ ПЕРЕХВАТЧИКИ НА САМОМ ДЕЛЕ МОГЛИ ВСКРЫТЬ ШИФР ЦЕЗАРЯ...
....МАКСИМУМ ЗА 24 ПОПЫТКИ. ЕСЛИ БЫ, КОНЕЧНО, ЗНАЛИ, КАК ОН УСТРОЕН.
25
8 9 10 11 12


НО ЕСЛИ ИСПОЛЬЗОВАТЬ, НАПРИМЕР, ЗНАКИ ЯПОНСКОГО ЯЗЫКА...
АГА!
ЭВРИКА!
...АЛФАВИТЫ ХИРАГАНУ, КАТАКАНУ И ИЕРОГЛИФЫ В ПРИДАЧУ, – ТО ЧИСЛО КЛЮЧЕЙ ПРЕВЫСИТ 10 ТЫСЯЧ!


ЧЕМ БОЛЬШЕ ЧИСЛО КЛЮЧЕЙ...,
...ТЕМ ЛУЧШЕ ШИФР ЗАЩИЩЁН ОТ АТАК!


ИТАК, ТЕПЕРЬ МЫ ПОСМОТРИМ ЧИСЛО КЛЮЧЕЙ В ДРУГИХ ШИФРАХ.

✤ Число ключей шифра одноалфавитной замены

Будем считать, что в этом и в последующих шифрах используется английский алфавит, состоящий из 26 букв. Общее число ключей шифра замены будет равно числу перестановок множества из 26 букв, рассчитываемому по следующей формуле:

$$P_{26} = 26! = 26 \times 25 \times 24 \times \cdots \times 3 \times 2 \times 1 \approx 4.03291461 \times 10^{26}.$$

Перестановкой (Permutation) *n* элементов называется любой упорядоченный набор этих элементов. Полученное выше значение является довольно большим: время поиска ключа (временная сложность криптоанализа) на компьютере, перебирающем 100 млн перестановок в секунду, может составить до 128 млрд лет.

Таким образом, хотя теоретически ключ найти возможно, практически шифр одноалфавитной замены считается вычислительно стойким.

Однако известно, что этот шифр уязвим для частотного криптоанализа, использующего такую его особенность, как совпадение частот появления букв в открытом тексте и шифртексте.

В связи с этим с практической точки зрения вычислительно криптостойким считается шифр одноалфавитной замены с одноразовым ключом (one time pad).

Теперь поговорим о разделе математики под названием комбинаторика. Кроме вышеописанной перестановки, существуют также понятия размещения и сочетания. Размещение из *n* по *r* – это упорядоченный набор *r* элементов, выбранных из множества *n* различных элементов. Таким образом, перестановка тоже является частным случаем размещения. Число размещений вычисляется по нижеприведённой формуле.

$${}_nA_r = n \times (n-1) \times (n-2) \times \cdots\cdots \times (n-r+1) = \frac{n!}{(n-r)\ !}.$$

Сочетание из *n* по *r* – это набор *r* элементов, выбранных из множества *n* различных элементов. Оно обозначается заглавной буквой *C*, от слова Combination.

$${}_nC_r = \frac{{}_nP_r}{r!} = \frac{n!}{(n-r)\ !\ r!}.$$

Различие между размещением и сочетанием заключается в том, что в размещении важен порядок следования элементов, поэтому AB и BA будут разными размещениями, а сочетание – это способ выбора элементов из множества, а порядок их следования здесь не важен, поэтому AB и BA будут одинаковыми сочетаниями. Кстати, восклицательный знак (!) – это факториал. Знак факториала после *n* означает произведение всех чисел от 1 до *n* включительно.

$$n! = n \times (n-1)\ \cdots\cdots \times 3 \times 2 \times 1.$$

✤ Число ключей шифра многоалфавитной замены

Примем длину одного блока равной n буквам. Так как сдвиги букв в блоке нам неизвестны, мы должны будем перебрать 26 значений сдвига 1-й позиции, для каждого из значений сдвига 1-й позиции – по 26 значений сдвига 2-й позиции, для каждого из значений сдвига 2-й позиции – по 26 значений сдвига 3-й позиции и так далее до n-й позиции в блоке. Общее число ключей будет следующим.

$$\underbrace{26 \times 26 \times \cdots \times 26 \times 26}_{n \text{ множителей}} = 26^n$$

Для $n = 4$ получим следующее.

$$\underbrace{26 \times 26 \times 26 \times 26}_{4 \text{ множителя}} = 26^4$$

$$26^4 = 456\,976$$

При увеличении n количество ключей резко увеличивается. Так, для $n = 10$ оно превысит 140 трлн.

✤ Число ключей шифра перестановки

Приняв длину одного блока равной n буквам, получим следующее.

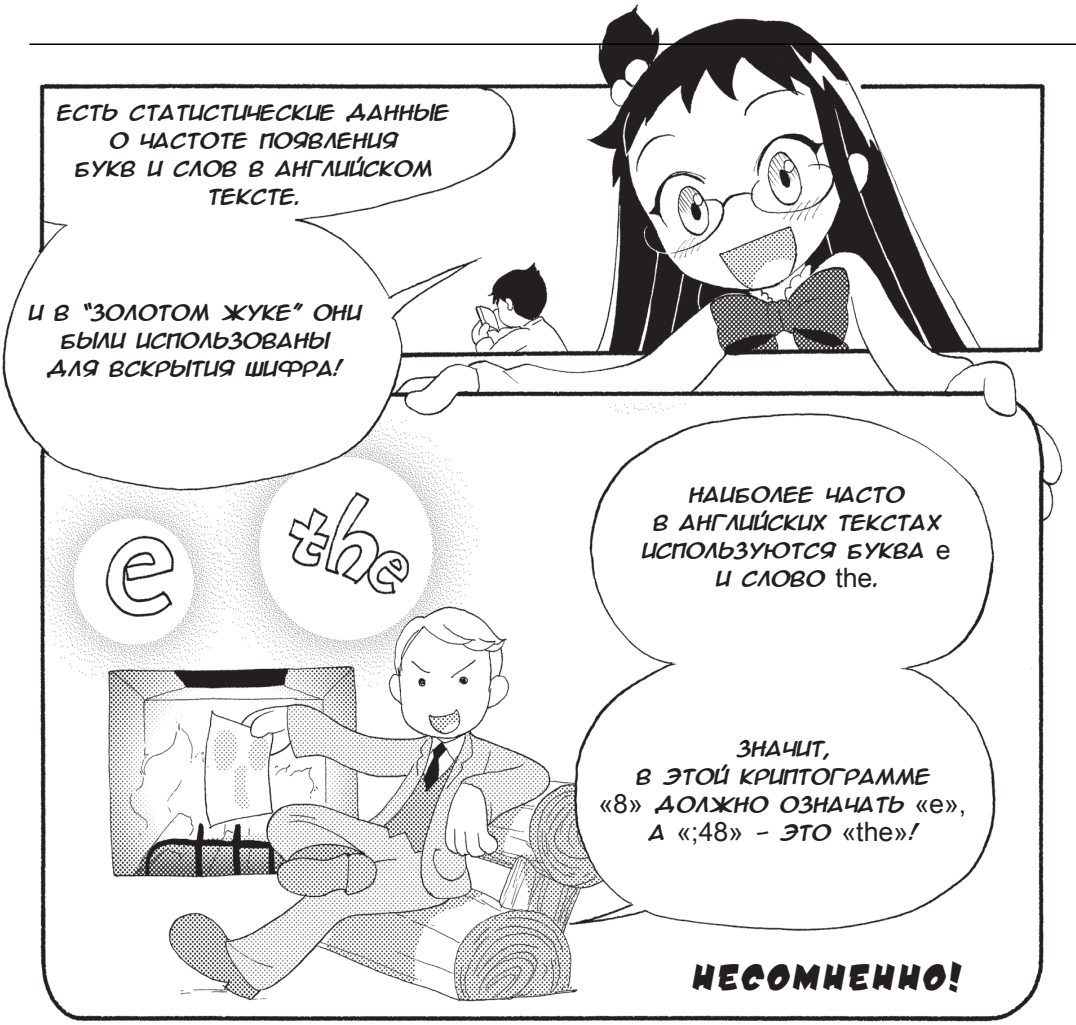
$$P_n = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1 = n!$$

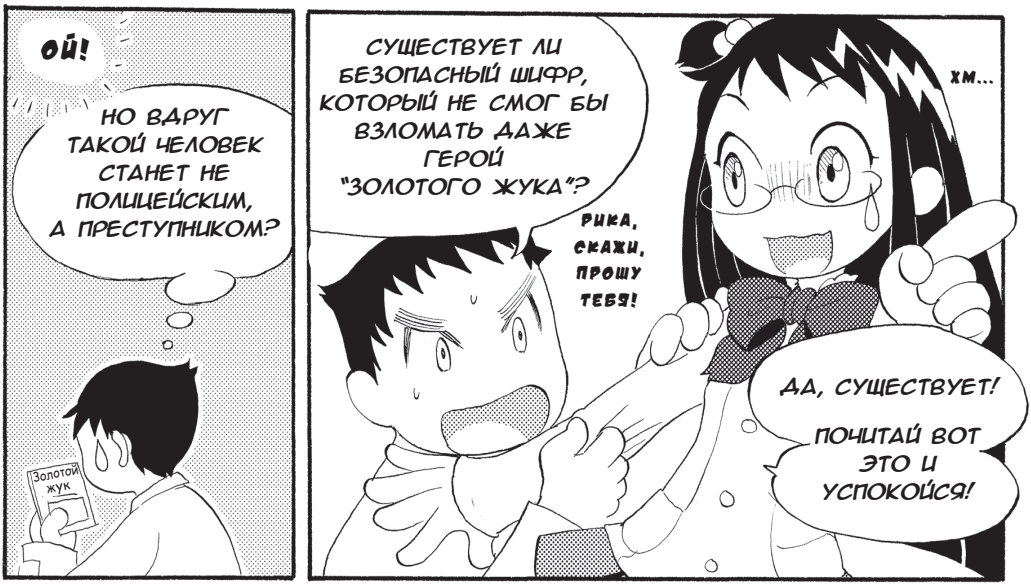
Для блока, состоящего из 4 букв ($n = 4$), общее число ключей Ek будет следующим:

$$4! = 4 \times 3 \times 2 \times 1 = 24.$$

При увеличении n число ключей будет возрастать, повышая тем самым стойкость шифра. Так, для $n = 26$ число ключей будет таким же, как для шифра одноалфавитной замены.


ЕСЛИ СЧИТАТЬ, ЧТО ЧЕМ БОЛЬШЕ ЧИСЛО КЛЮЧЕЙ, ТЕМ БЕЗОПАСНЕЕ ШИФР, ТО САМЫМ БЕЗОПАСНЫМ БУДЕТ ШИФР ОДНОАЛФАВИТНОЙ ЗАМЕНЫ.
ММ...
Шифр одноалфавитной замены
Шифр многоалфавитной замены
ОДНАКО ШИФР МНОГОАЛФАВИТНОЙ ЗАМЕНЫ ВЫГЛЯДИТ ПОСЛОЖНЕЕ.
ПРИ ИСПОЛЬЗОВАНИИ ДЛИННЫХ ШИФРТЕКСТОВ ШИФР ОДНОАЛФАВИТНОЙ ЗАМЕНЫ СТАНОВИТСЯ УЯЗВИМЫМ.
Золотой жук
Эдгар Аллан По
ЗОЛОТОЙ ЖУК? ЭТО АТЛАС НАСЕКОМЫХ?
БОГАЧ, НАВЕРНОЕ...
НЕТ, ЭТО ДЕТЕКТИВ ПРО ВСКРЫТИЕ ШИФРА.
ОГО, ЭТО НОВЕЛЛА!
Часть криптограммы из «Золотого жука»
53 ‡ ‡ † 305)) 6 * ; 4826)
4 ‡ .) 4 ‡) ; 806 * ; 48 †
8 ¶ 60)) 85 ; 1 ‡ (; : ‡ * 8


ЕСТЬ СТАТИСТИЧЕСКИЕ ДАННЫЕ О ЧАСТОТЕ ПОЯВЛЕНИЯ БУКВ И СЛОВ В АНГЛИЙСКОМ ТЕКСТЕ.
И В "ЗОЛОТОМ ЖУКЕ" ОНИ БЫЛИ ИСПОЛЬЗОВАНЫ ДЛЯ ВСКРЫТИЯ ШИФРА!
e
the
НАИБОЛЕЕ ЧАСТО В АНГЛИЙСКИХ ТЕКСТАХ ИСПОЛЬЗУЮТСЯ БУКВА e И СЛОВО the.
ЗНАЧИТ, В ЭТОЙ КРИПТОГРАММЕ «8» ДОЛЖНО ОЗНАЧАТЬ «e», А «;48» – ЭТО «the»!
НЕСОМНЕННО!
ХОРОШО СООБРАЖАЕТ!
ВОТ БЫ ПРИГЛАСИТЬ ЕГО НА РАБОТУ В ПОЛИЦИЮ!
ЧЕМ ДЛИНЕЕ ШИФРТЕКСТ, ТЕМ БОЛЬШЕ В НЁМ ЗАЦЕПОК ДЛЯ ВЗЛОМА.
КОРОТКИЕ ШИФРТЕКСТЫ ВЗЛАМЫВАТЬ ТРУДНО.



✤ Возможность криптоанализа

В общем случае существуют следующие условия, делающие возможным вскрытие шифра.

① Известен алгоритм шифрования.

② Известны статистические свойства открытого текста: повторяемость букв и т. п.

③ Имеется большое количество образцов шифртекста.

✤ Совершенно стойкий шифр

Используя одноразовые ключи, основанные на случайных числах, можно создать шифр, теоретически не поддающийся вскрытию.

Практически шифртекст *C* в подобном шифре генерируется путём применения ряда случайных чисел к открытому тексту *P*, причём длина этого ряда равна длине открытого текста. Эта криптосистема, изобретённая в 1917 году телеграфистом AT&T Гильбертом Вернамом, называется шифром Вернама. Этот шифр, в котором используются одноразовые ключи из шифровального блокнота (one-time pad), обладает абсолютной криптостойкостью, то есть принципиально не поддаётся вскрытию, как было доказано в 1949 году Шенноном (см. стр. 19).

Вот простой пример шифра Вернама.
Сначала мы ставим в соответствие коды символов (числа) буквам алфавита.

Таблица 1.1. Коды символов алфавита

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Коды шифртекста мы будем получать, находя остаток от деления результата сложения чисел на 26.

① Заменяем буквы алфавита на коды символов.

Открытый текст	M	O	M	O	T	A	R	O
	↓	↓	↓	↓	↓	↓	↓	↓
	12	14	12	14	19	0	17	14

② Складываем коды символов с однократно используемым рядом случайных чисел.

	12	14	12	14	19	0	17	14
Ряд случайных чисел (ключ шифрования)	+	+	+	+	+	+	+	+
	9	20	15	23	27	2	15	8
	=	=	=	=	=	=	=	=
	21	34	27	37	46	2	32	22

③ Вычисляем остаток от деления на 26.

	21	34	27	37	46	2	32	22
	↓	↓	↓	↓	↓	↓	↓	↓
	21	8	1	11	20	2	6	22

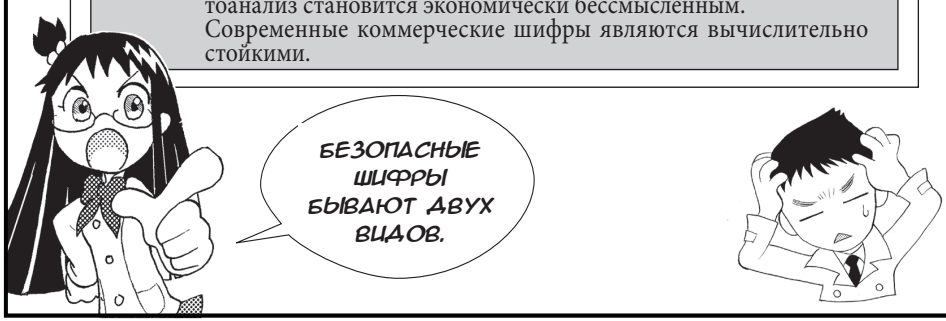
④ Используя коды символов, заменяем числа на буквы алфавита.

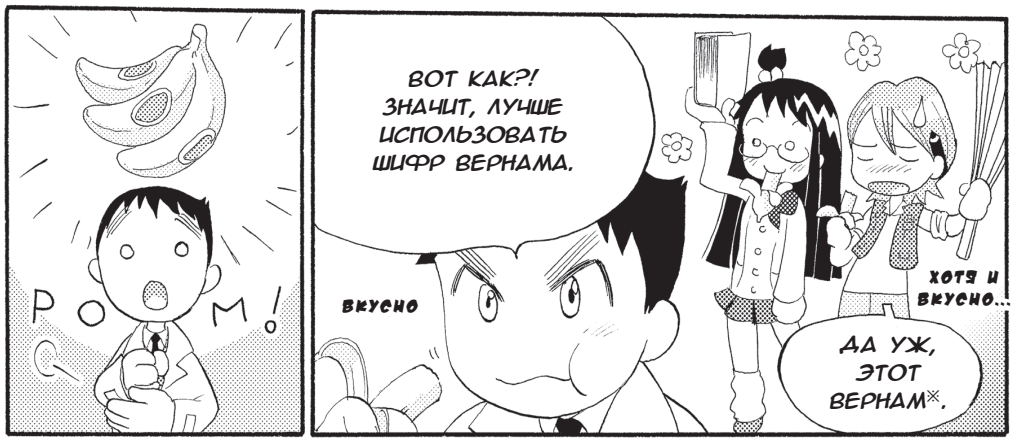
	21	8	1	11	20	2	6	22
Шифртекст	↓	↓	↓	↓	↓	↓	↓	↓
	V	I	B	L	U	C	G	W

✤ Типы криптостойкости

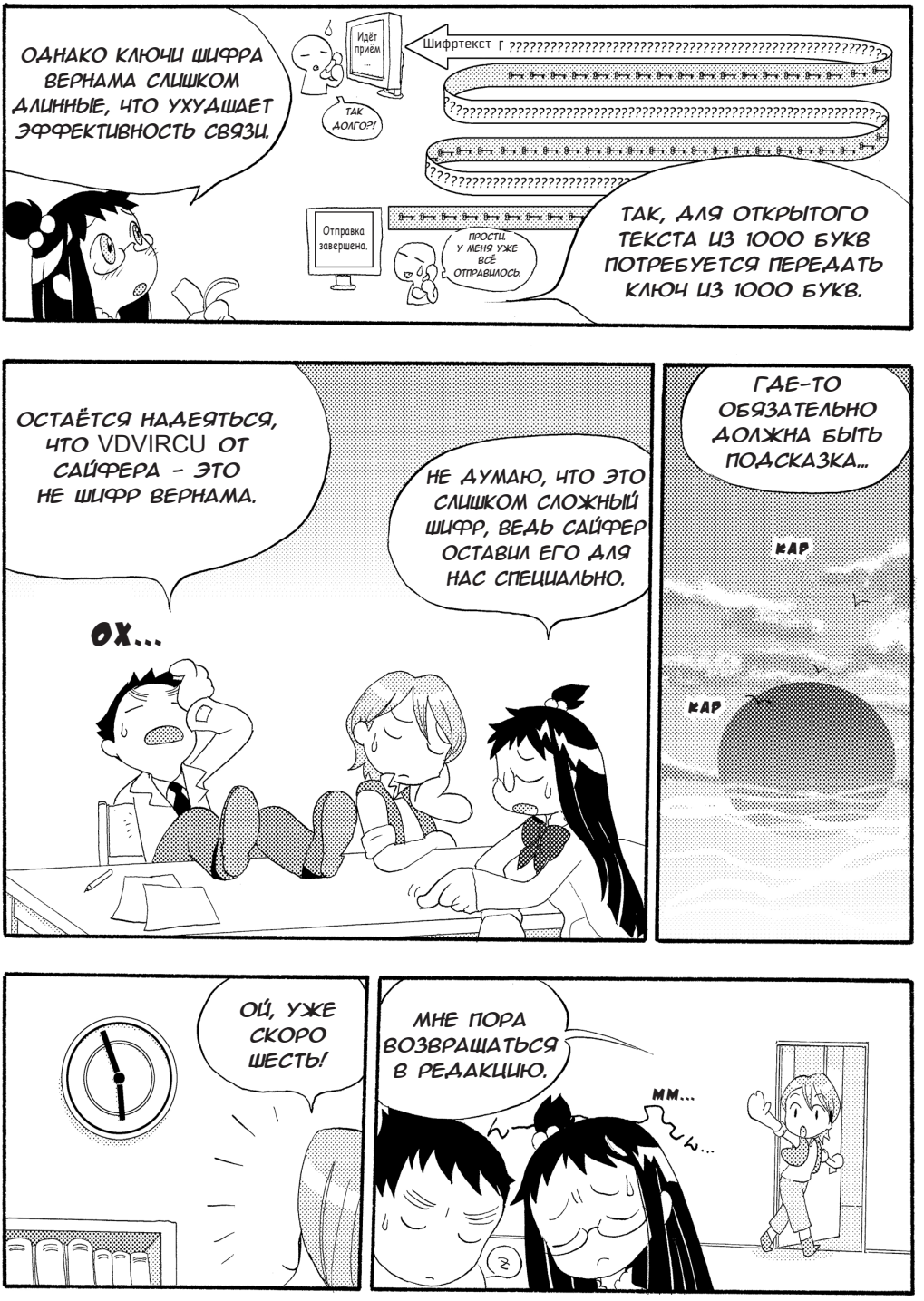
① **Совершенно стойкий шифр:**
Шифр, который теоретически невозможно взломать, как шифр Вернама.

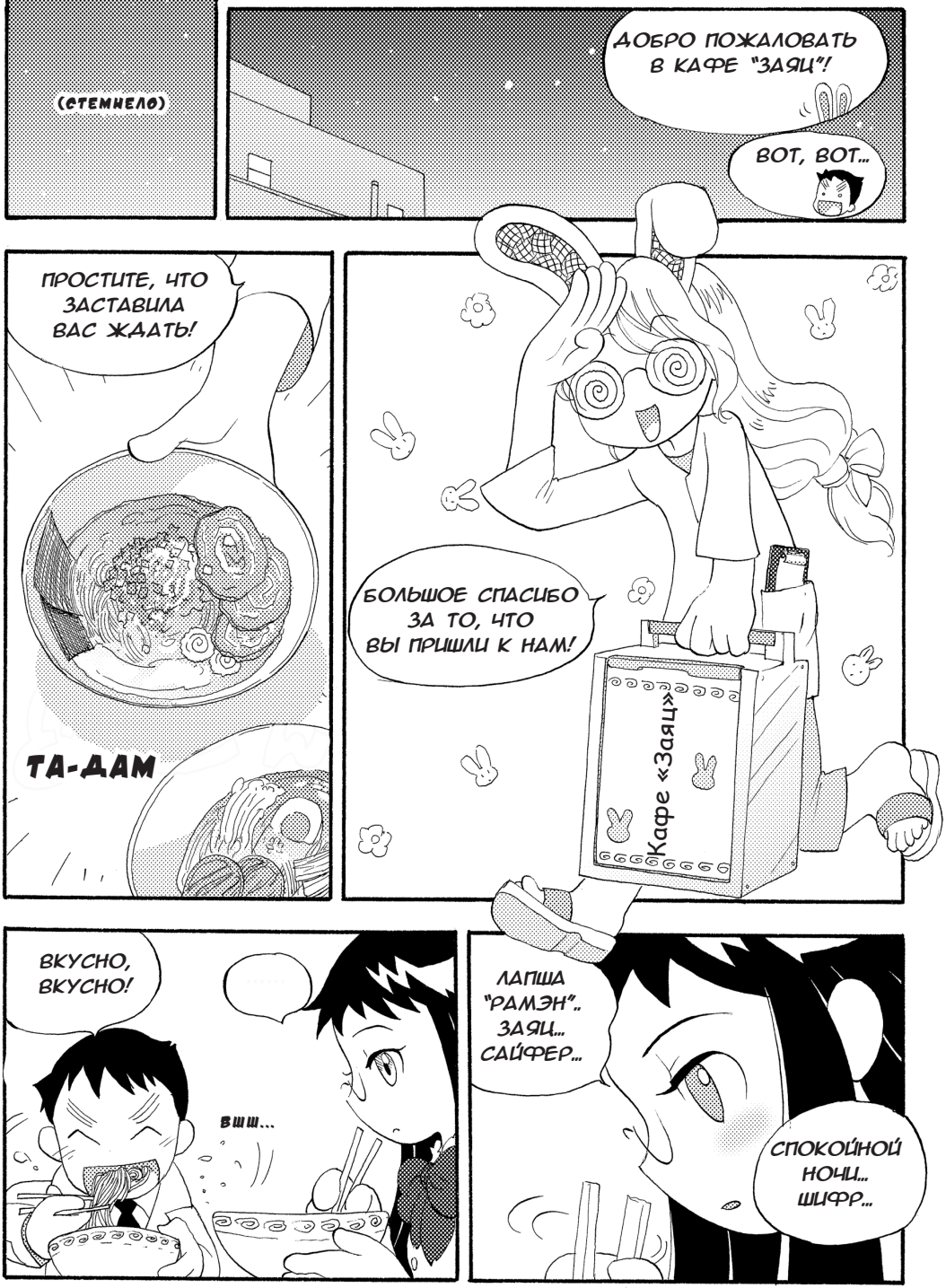
② **Вычислительно стойкий шифр:**
Для взлома требуется столько времени и трудозатрат, что криптоанализ становится экономически бессмысленным. Современные коммерческие шифры являются вычислительно стойкими.

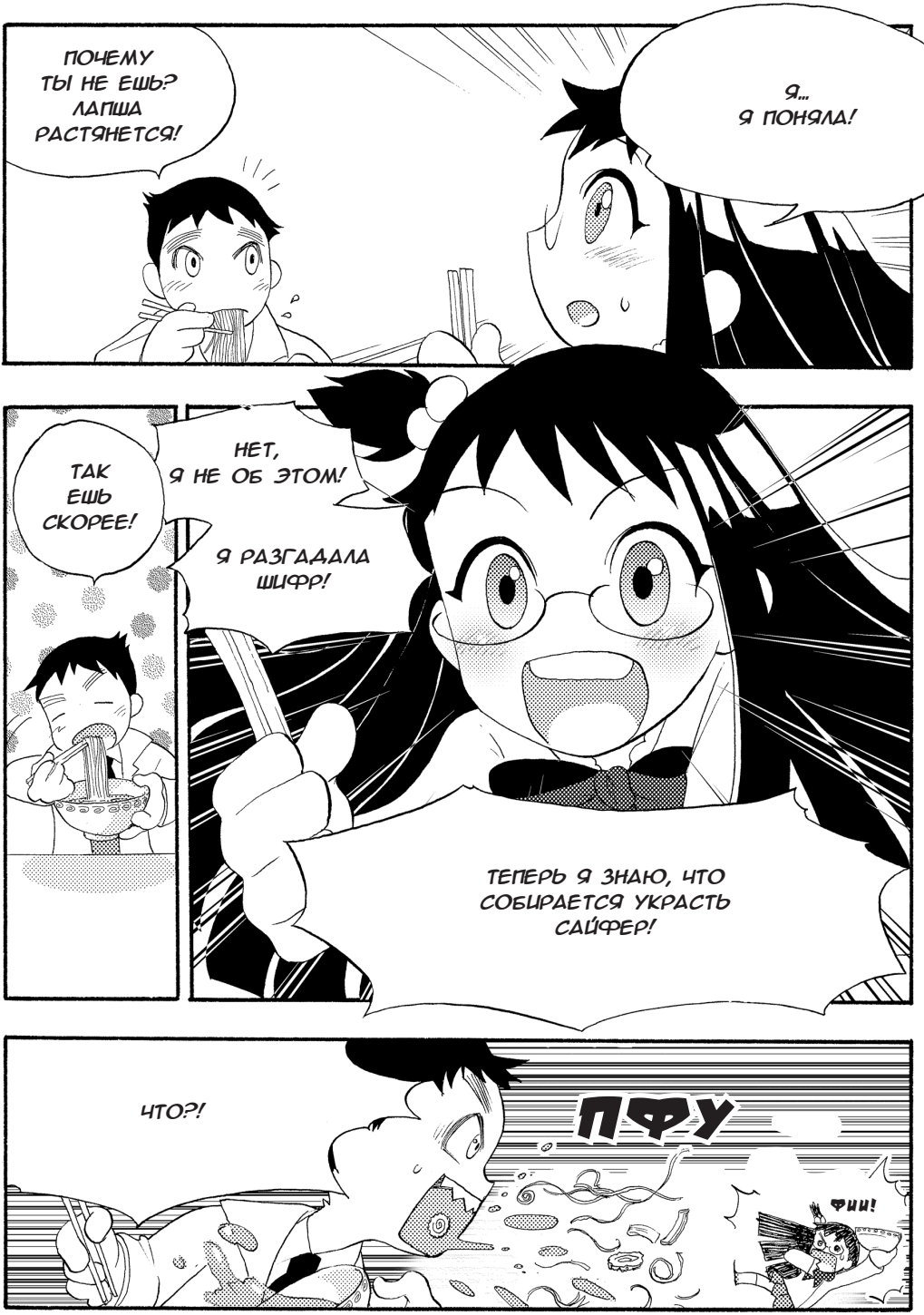




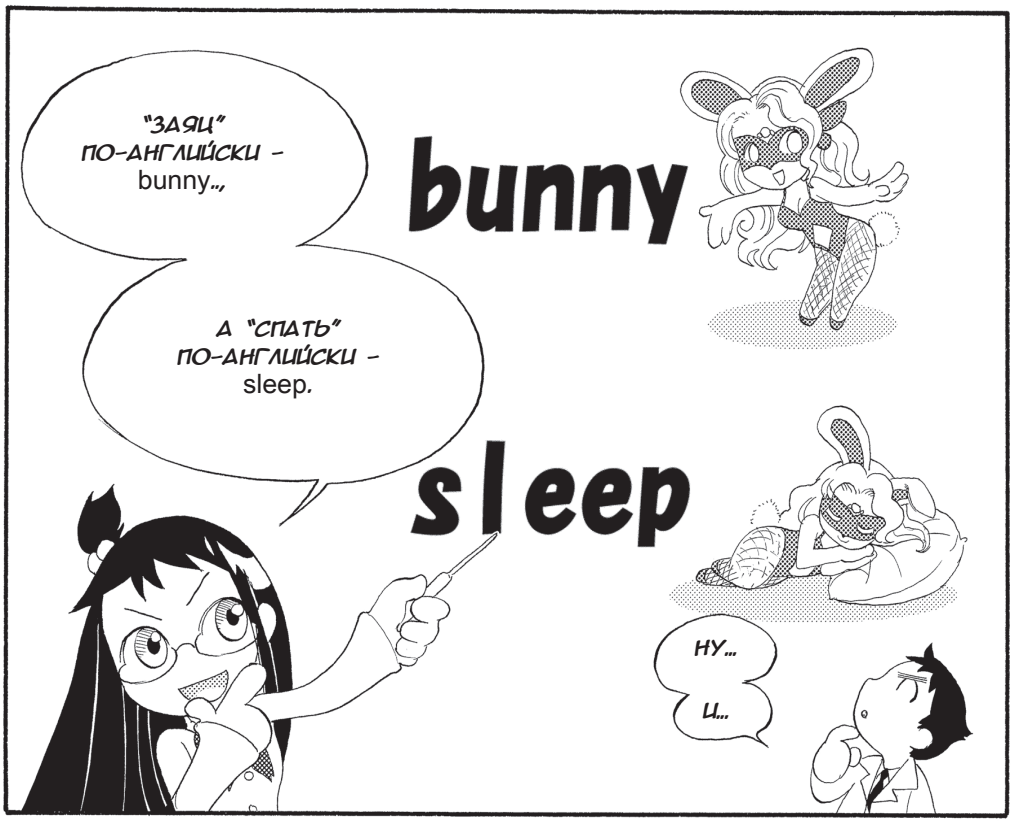
※ Бананы появляются в манге потому, что по-японски имя Вернам звучит как Банам.


Однако ключи шифра Вернама слишком длинные, что ухудшает эффективность связи.
Идёт приём...
Так долго?!
Шифртекст Г ??
Отправка завершена.
Прости, у меня уже всё отправилось.
Так, для открытого текста из 1000 букв потребуется передать ключ из 1000 букв.
Остаётся надеяться, что VDVIRCU от Сайфера – это не шифр Вернама.
Ох...
Не думаю, что это слишком сложный шифр, ведь Сайфер оставил его для нас специально.
Где-то обязательно должна быть подсказка...
КАР
КАР
Ой, уже скоро шесть!
Мне пора возвращаться в редакцию.
ММ...

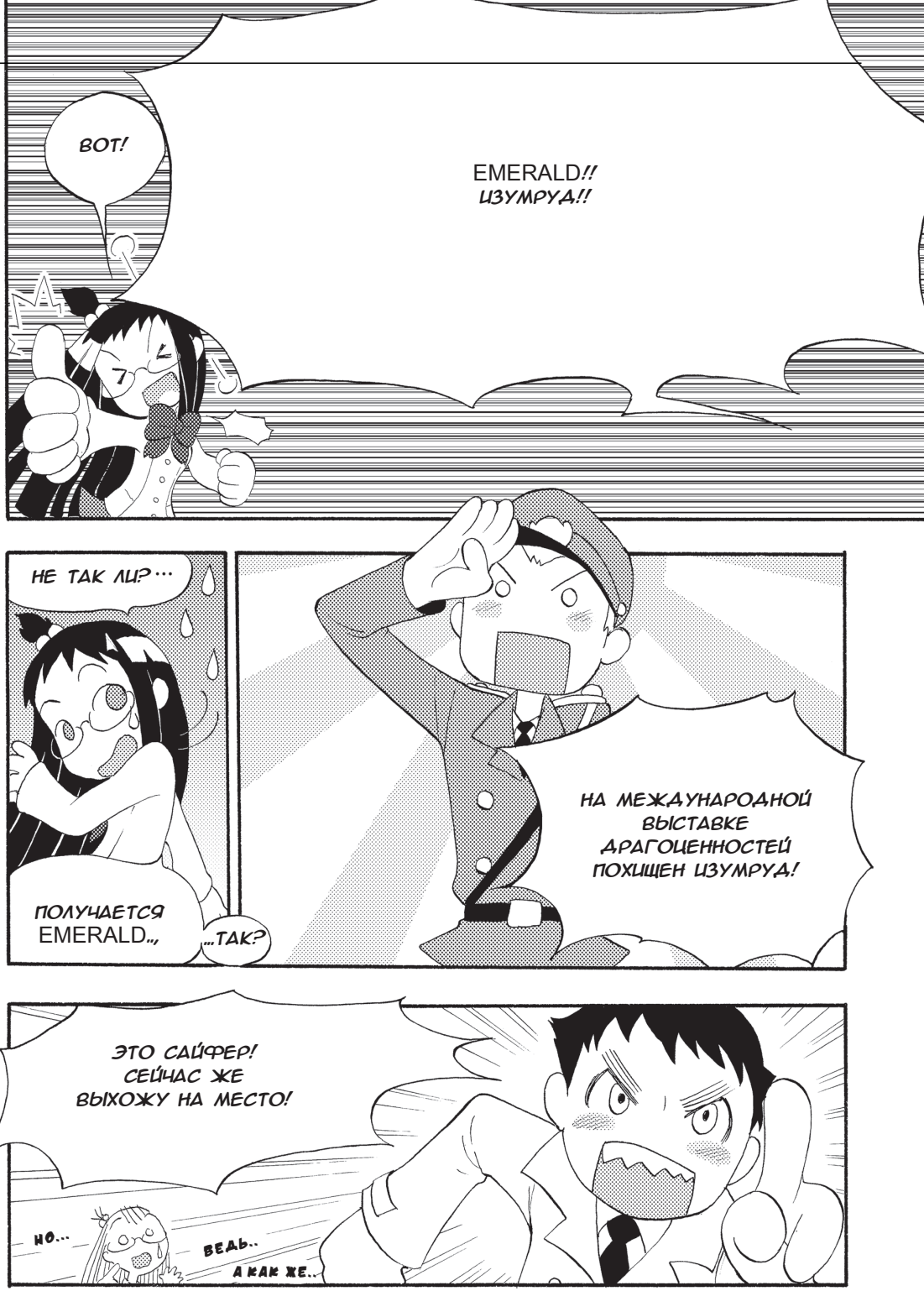

(стемнело)
Добро пожаловать в кафе "Заяц"!
Вот, вот...
Простите, что заставила вас ждать!
Та-дам
Большое спасибо за то, что вы пришли к нам!
Кафе «Заяц»
Вкусно, вкусно!
Вшш...
Лапша "рамэн".. Заяц... Сайфер...
Спокойной ночи... Шифр...

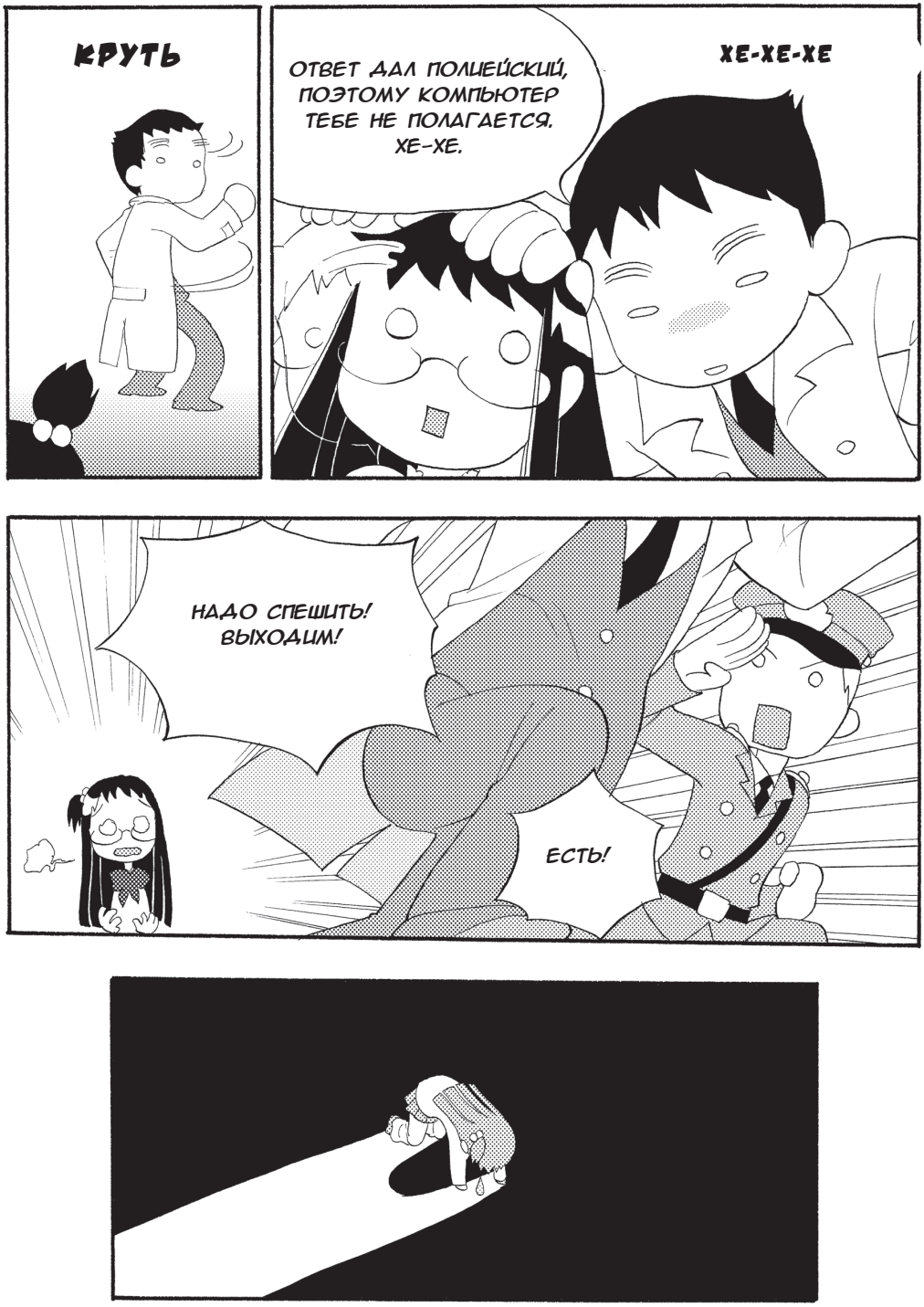

ПОЧЕМУ ТЫ НЕ ЕШЬ? ЛАПША РАСТЯНЕТСЯ!
Я... Я ПОНЯЛА!
ТАК ЕШЬ СКОРЕЕ!
НЕТ, Я НЕ ОБ ЭТОМ!
Я РАЗГАДАЛА ШИФР!
ТЕПЕРЬ Я ЗНАЮ, ЧТО СОБИРАЕТСЯ УКРАСТЬ САЙФЕР!
ЧТО?!
ПФУ
ФИИ!


ЕСЛИ Я СКАЖУ ТЕБЕ, ЧТО УКРАДЁТ САЙФЕР, КОМПЬЮТЕР МНЕ КУПИШЬ?
ЛАДНО... ДЕВАТЬСЯ НЕКУДА.
ПОДСКАЗКОЙ БЫЛИ "ЗАЯЦ" И "СПОКОЙНОЙ НОЧИ".
Я – Весёлый сайфер. Это я украла картину. В следующий раз украду VDVIRCU.
Спокойной ночи
ЧТО?!


"ЗАЯЦ" ПО-АНГЛИЙСКИ – bunny.,,
bunny
А "СПАТЬ" ПО-АНГЛИЙСКИ – sleep.
sleep
НУ...
И...

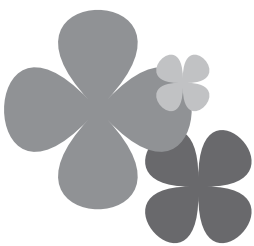

ЕСЛИ КАЖДУЮ БУКВУ СЛОВА bunny СДВИНУТЬ В АЛФАВИТЕ НА 17 ПОЗИЦИЙ ВПЕРЁД...
bunny
На 17 позиций вперёд
sleep
...ТО МЫ ПОЛУЧИМ СЛОВО sleep!
ВОТ КАК! ЗНАЧИТ, КЛЮЧОМ БЫЛ СДВИГ НА 17 ПОЗИЦИЙ?!
ДА!
ХА
ХА
А ЕСЛИ ТЕПЕРЬ СДВИНУТЬ ВСЕ БУКВЫ VDVIRCU НА 17 ПОЗИЦИЙ НАЗАД В АЛФАВИТЕ...
ХА
ХА
ХА
...ТО У НАС ПОЛУЧИТСЯ...
ГЛЫГ


ВОТ!
EMERALD!!
ИЗУМРУД!!
НЕ ТАК ЛИ?…
ПОЛУЧАЕТСЯ EMERALD…
…ТАК?
НА МЕЖДУНАРОДНОЙ ВЫСТАВКЕ ДРАГОЦЕННОСТЕЙ ПОХИЩЕН ИЗУМРУД!
ЭТО САЙФЕР! СЕЙЧАС ЖЕ ВЫХОЖУ НА МЕСТО!
НО…
ВЕДЬ..
А КАК ЖЕ..

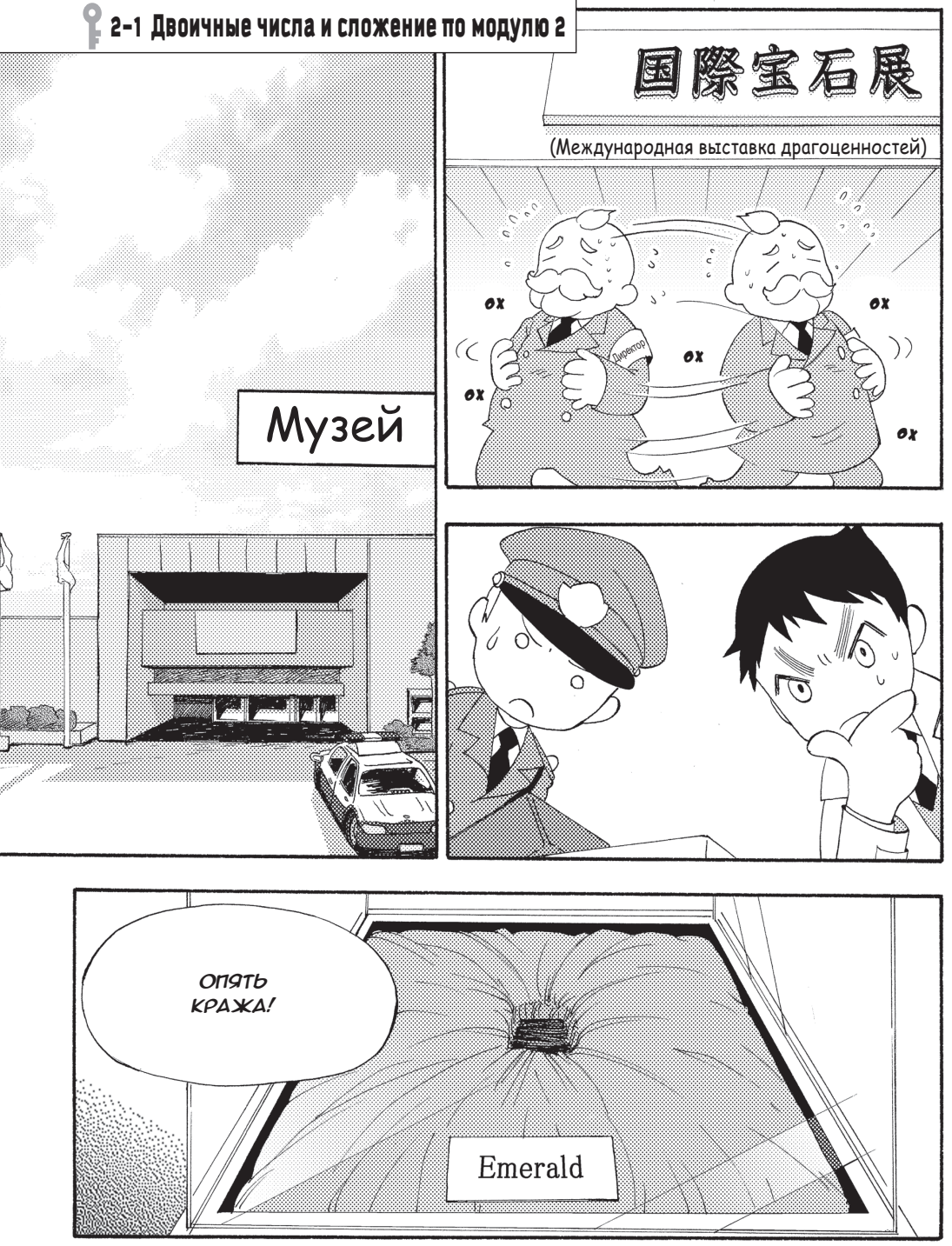

КРУТЬ
ОТВЕТ ДАЛ ПОЛИЕЙСКИЙ, ПОЭТОМУ КОМПЬЮТЕР ТЕБЕ НЕ ПОЛАГАЕТСЯ. ХЕ-ХЕ.
ХЕ-ХЕ-ХЕ
НАДО СПЕШИТЬ! ВЫХОДИМ!
ЕСТЬ!

ГЛАВА 2

ОДНОКЛЮЧЕВОЙ ШИФР



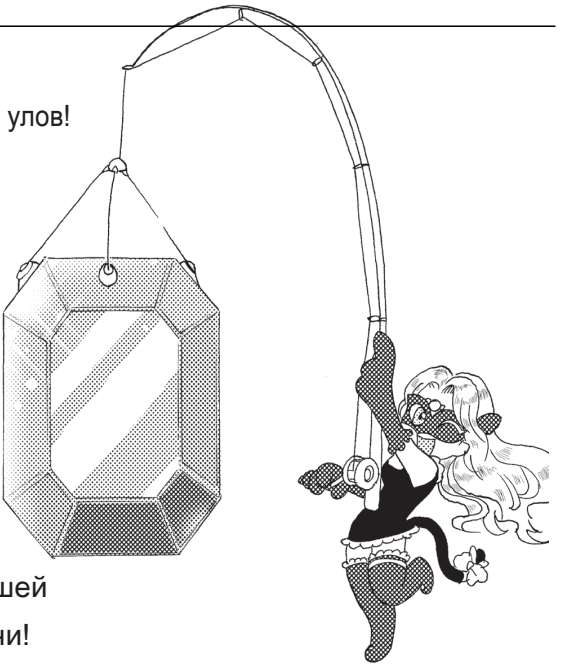
2-1 Двоичные числа и сложение по модулю 2




ЭТО БЫЛ ИЗУМРУД СТОИМОСТЬЮ В 300 МЛН ЕН!
ААА!
БЫЛ ЛИ ЗАКРЫТ НА КЛЮЧ ВЫСТАВОЧНЫЙ ШКАФ?
ВООБЩЕ-ТО, ДА, НО ПОТОМ ЕГО КТО-ТО ОТКРЫЛ НЕЗАМЕТНО ДЛЯ НАС.
ЗНАЧИТ, ДРАГОЦЕННЫЙ КАМЕНЬ КТО-ТО ВЫУДИЛ С ЧЕРДАКА.
А ВОТ И Я!
ОТКУДА ТЕБЕ ИЗВЕСТНО О СПОСОБЕ КРАЖИ?
ВСЁ-ТАКИ ЭТО ТВОИХ РУК ДЕЛО...
ФИТЬ
НЕТ, ПРОСТО К НАМ В РЕДАКЦИЮ ПРИШЛО ВОТ ЭТО ПИСЬМО!

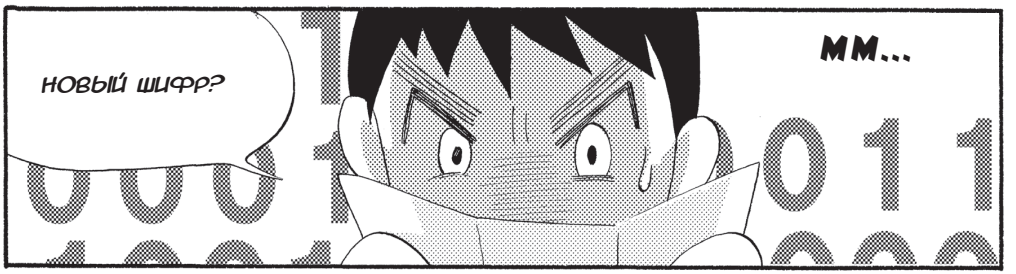
Сегодня опять хороший улов!
Я выудила большой
драгоценный камень.♥

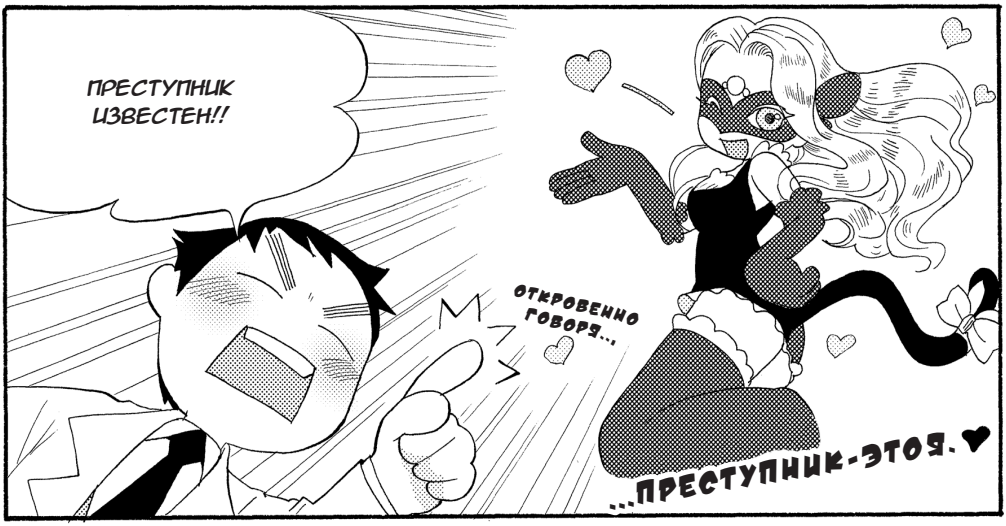
Весёлый сайфер

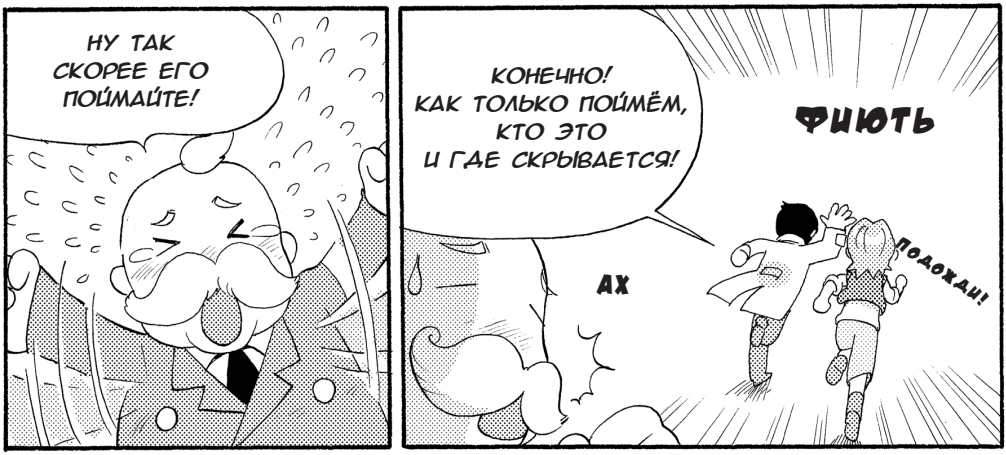


P. S.:
Изучите это до нашей
следующей встречи!

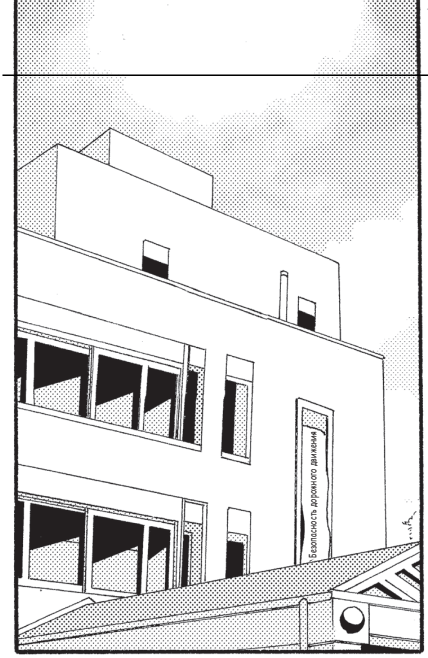
00110001 00101011 00110001 00111101 00110000

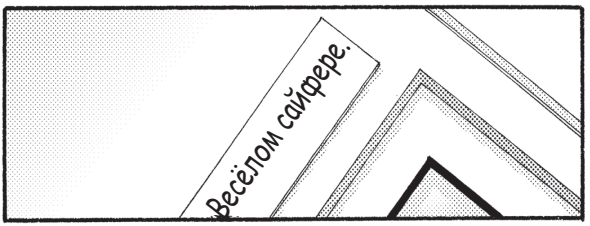


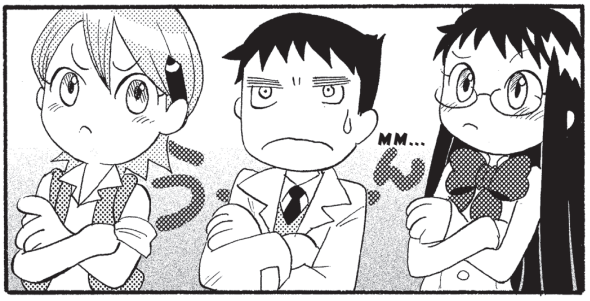

ПРЕСТУПНИК ИЗВЕСТЕН!!
ОТКРОВЕННО ГОВОРЯ...
...ПРЕСТУПНИК-ЭТО Я.♥


НУ ТАК СКОРЕЕ ЕГО ПОЙМАЙТЕ!
КОНЕЧНО! КАК ТОЛЬКО ПОЙМЁМ, КТО ЭТО И ГДЕ СКРЫВАЕТСЯ!
ФИЮТЬ
ПОДОЖДИ!
АХ

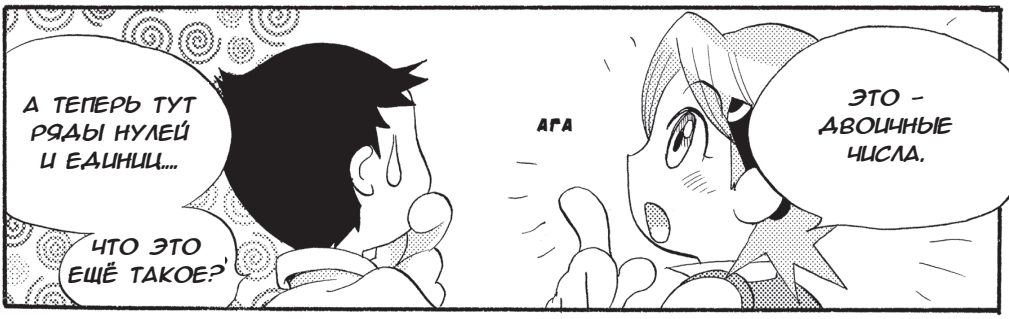

ПРОСТО ВОПИЮЩАЯ БЕЗОТВЕТСТВЕННОСТЬ!

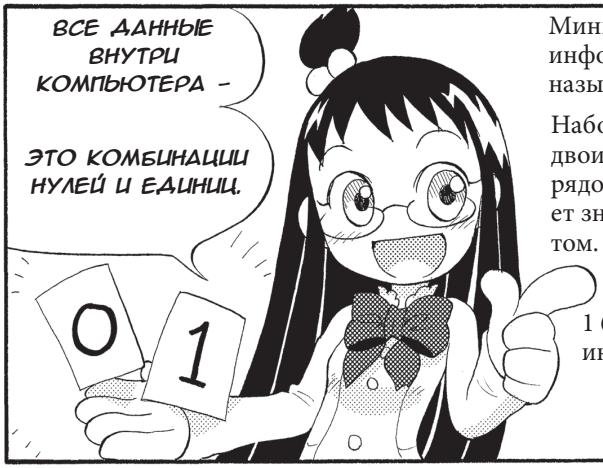






00110001 00101011
00110001 00111101
00110000





Минимальная единица информации, равная 0 или 1, называется битом.

Набор из 8 битов, другими словами, двоичное число, состоящее из 8 разрядов, каждый из которых принимает значения 0 или 1), называется байтом.

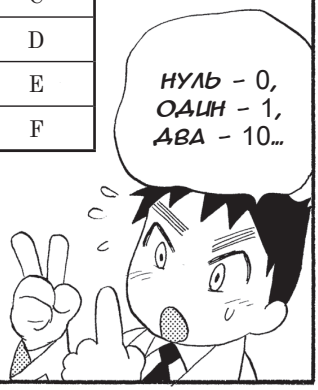
1 байтом можно выразить $2^8 = 256$ информационных сообщений.

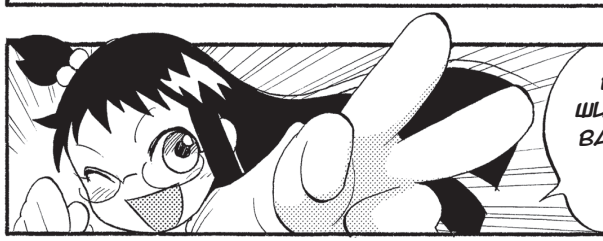
Таблица 2.1. *Соответствие двоичных, десятичных и шестнадцатеричных чисел*

Двоичные числа	Десятичные числа	Шестнадцатеричные числа
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

Двоичные числа	Десятичные числа	Шестнадцатеричные числа
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Так как при увеличении двоичных чисел резко увеличивается их разрядность, их часто выражают в шестнадцатеричном виде.
Для того чтобы показать, что число является шестнадцатеричным, как правило, перед ним ставят знак «0x»: шестнадцатеричное число 0xA выражает десятичное число 10.





※Кодировка JIS X 0201 расширяет международную стандартную кодировку ASCII (7 бит) до 1 байта (8 бит), давая возможность выражать в дополнение к цифрам и английским буквам знаки азбуки «катакана» половинной ширины и др.

※Заголовки столбцов таблицы соответствуют младшим 4 битам, а заголовки строк – старшим 4 битам шестнадцатеричного числа.

Таблица 2.2. Кодировка JIS X 0201

4 младших бита \ 4 старших бита	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00		DE		0	@	P		p				ー	タ	ミ		
01	SH	D1	!	1	A	Q	a	q			。	ア	チ	ム		
02	SX	D2	"	2	B	R	b	r			「	イ	ツ	メ		
03	EX	D3	#	3	C	S	c	s			」	ウ	テ	モ		
04	ET	D4	$	4	D	T	d	t			、	エ	ト	ヤ		
05	EQ	NK	%	5	E	U	e	u			・	オ	ナ	ユ		
06	AK	SN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
07	BL	EB	'	7	G	W	g	w			ァ	キ	ヌ	ラ		
08	BS	CN	(	8	H	X	h	x			ィ	ク	ネ	リ		
09	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
0A	LF	SB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
0B	HM	EC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
0C	CL	→	,	<	L	¥	l	|			ャ	シ	フ	ワ		
0D	CR	←	-	=	M	]	m	}			ュ	ス	ヘ	ン		
0E	SO	↑	.	>	N	^	n	‾			ョ	セ	ホ	゛		
0F	SI	↓	/	?	O	_	o				ッ	ソ	マ	゜		

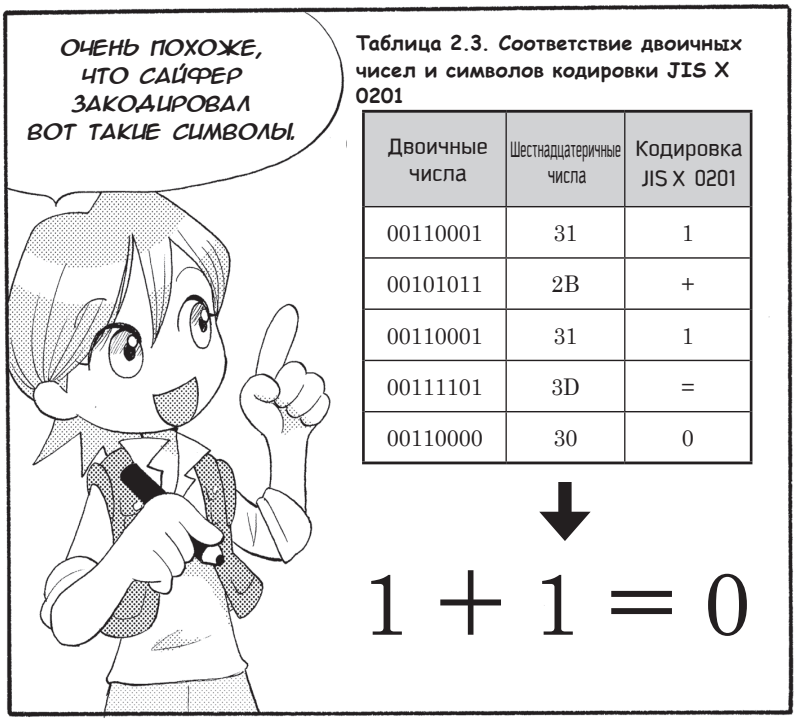
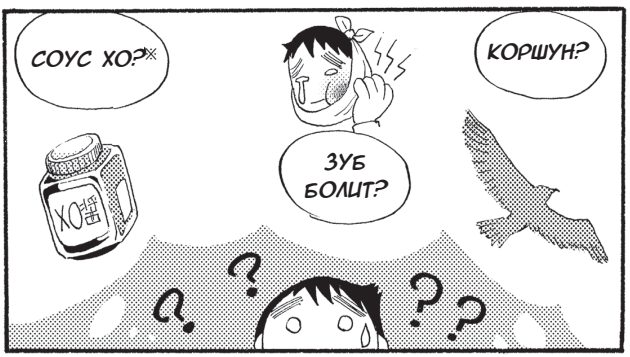
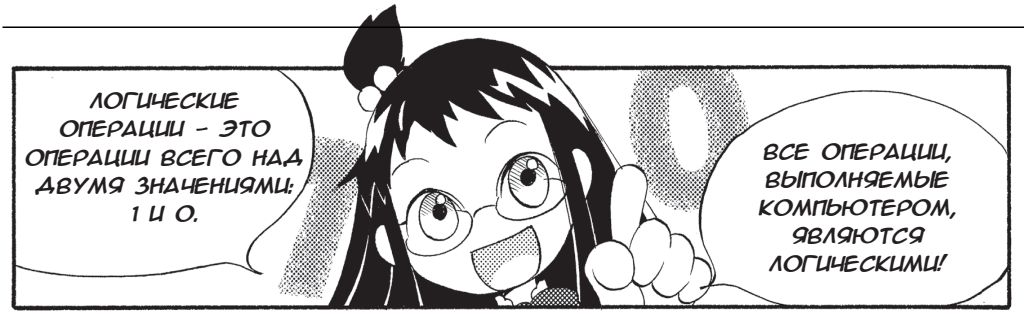


Таблица 2.3. Соответствие двоичных чисел и символов кодировки JIS X 0201

Двоичные числа	Шестнадцатеричные числа	Кодировка JIS X 0201
00110001	31	1
00101011	2B	+
00110001	31	1
00111101	3D	=
00110000	30	0

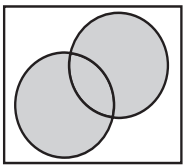


※ Непереводимая игра слов с термином «исключающее ИЛИ».



OR (логическое сложение) $A + B$

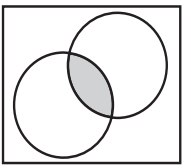
A	B	A+B
0	0	0
1	0	1
0	1	1
1	1	1



Если A или B равно 1, то результат равен 1.

AND (логическое умножение) $A \cdot B$

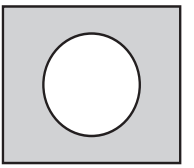
A	B	A·B
0	0	0
1	0	0
0	1	0
1	1	1



Если A и B равны 1, то результат равен 1.

NOT (инверсия) $\overline{A}$

A	$\overline{A}$
1	0
0	1

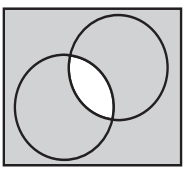


Если A равно 0, то результат равен 1.

NAND (штрих Шеффера)

$\overline{A \cdot B}$

A	B	AB
0	0	1
1	0	1
0	1	1
1	1	0

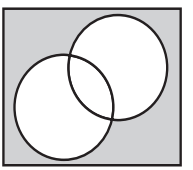


Если A или B равно 0, то результат равен 1.

NOR (стрелка Пирса)

$\overline{A+B}$

A	B	$\overline{A+B}$
0	0	1
1	0	0
0	1	0
1	1	0

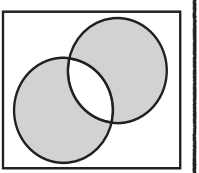


Если A и B равно 0, то результат равен 1.

XOR (сложение по модулю 2)

$\overline{A} \cdot B + A \cdot \overline{B} = (A \oplus B)$

A	B	A ⊕ B
0	0	0
1	0	1
0	1	1
1	1	0



Если A и B отличаются, то результат равен 1. (В связи с этим называется также «логической неравнозначностью»)

Рис. 2.1. Логические операции

Пусть (1101) – это открытый текст, а (1001) – ключ шифрования. Тогда операция XOR даст нам следующий результат:

$$\underset{\text{Открытый текст}}{(1101)} \oplus \underset{\text{Ключ шифрования}}{(1001)} = \underset{\text{Шифртекст}}{(0100)}$$

Результат операции – (0100), – будем считать шифртекстом. Теперь выполним операцию XOR над шифртекстом и ключом расшифрования (1001).

$$\underset{\text{Шифртекст}}{(0100)} \oplus \underset{\text{Ключ расшифрования}}{(1001)} = \underset{\text{Открытый текст}}{(1101)}$$

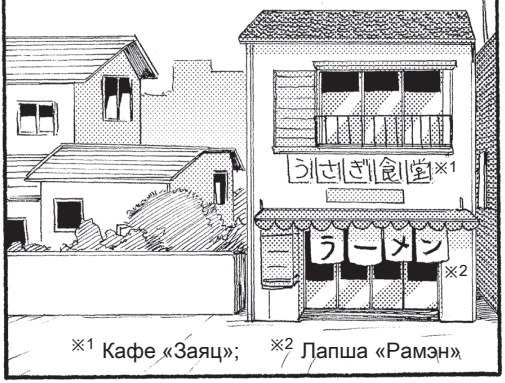
При этом произойдёт расшифрование – мы получим открытый текст. Теперь мы выполним операцию XOR над шифртекстом и открытым текстом. Как видите, у нас получился ключ.

$$\underset{\text{Шифртекст}}{(0100)} \oplus \underset{\text{Открытый текст}}{(1101)} = \underset{\text{Ключ шифрования (равен ключу расшифрования)}}{(1001)}$$

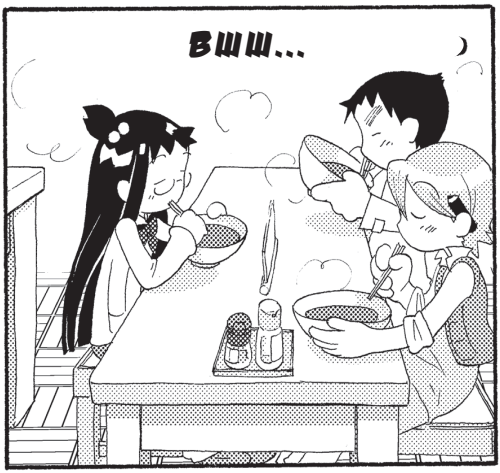
Таким образом, если из трёх наборов данных: открытый текст, ключ шифрования (ключ расшифрования), шифртекст; – у нас имеются два, мы сможем найти недостающий набор данных.

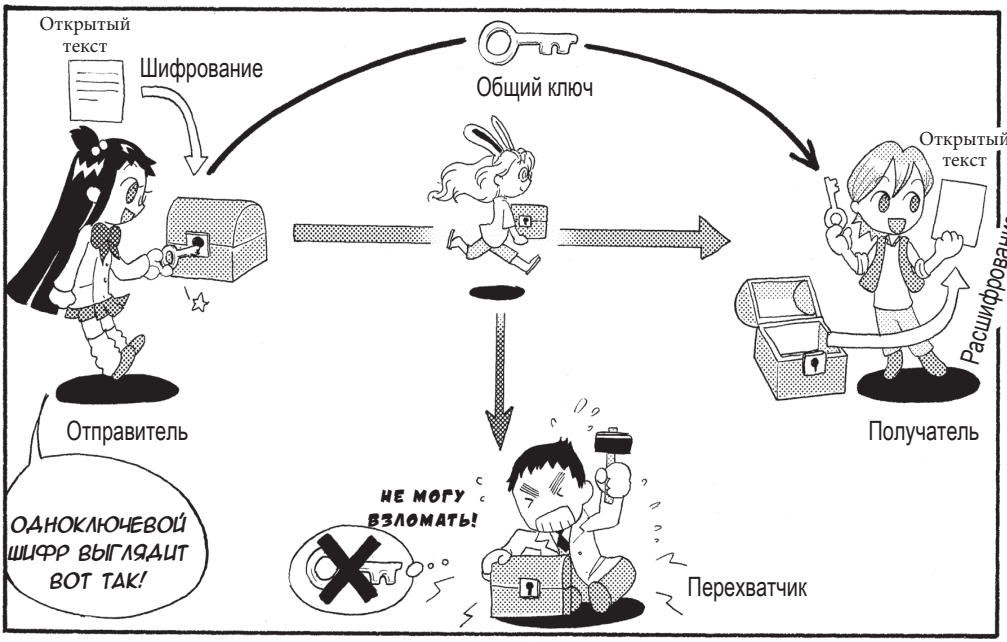
※ Обычно этот шифр называют «симметричным шифром». – *Прим. ред.*

2-2 Что такое одноключевой шифр?

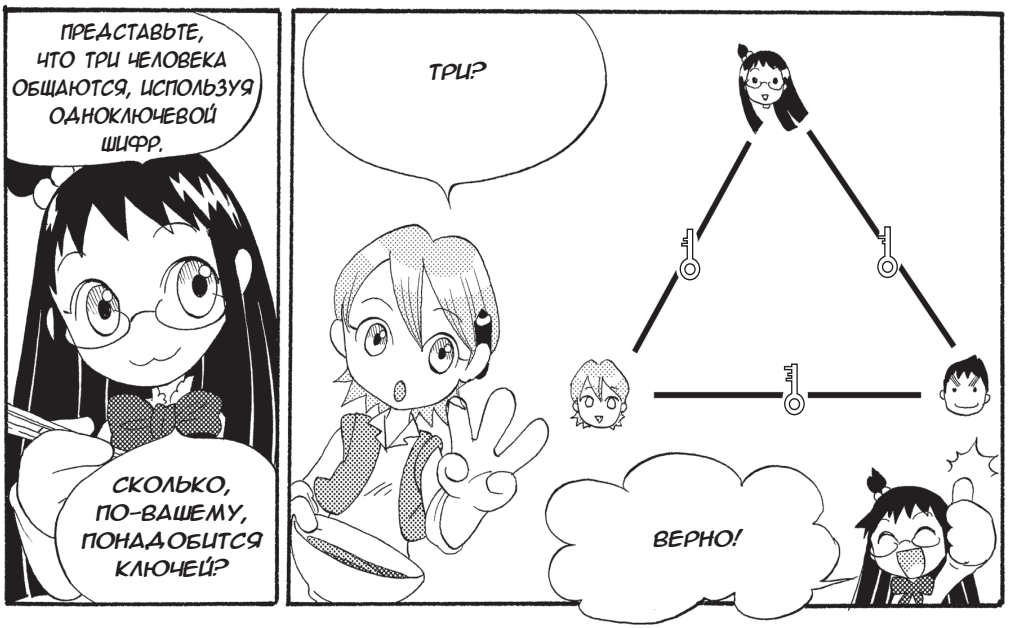


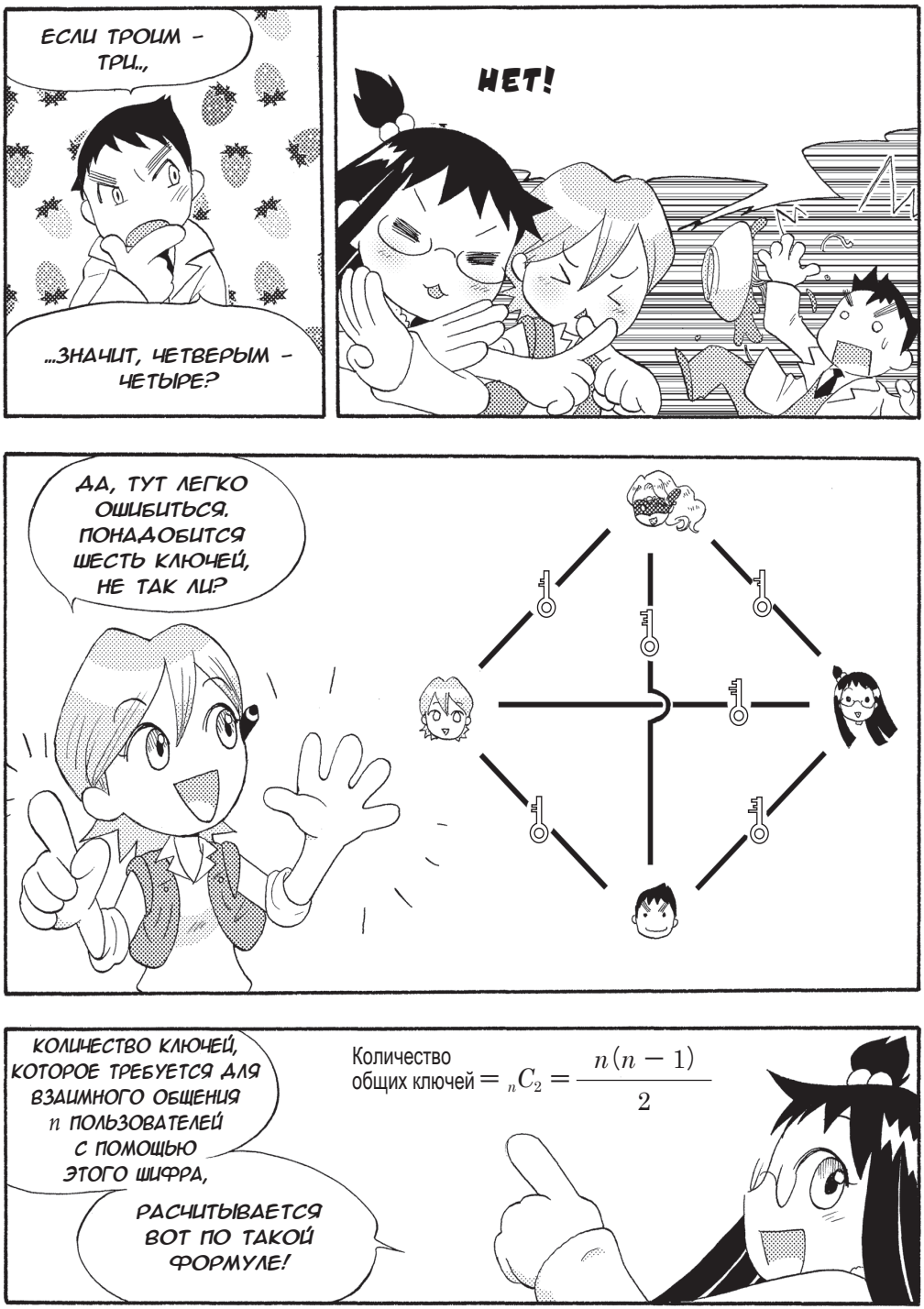
※1 Кафе «Заяц»; ※2 Лапша «Рамэн»

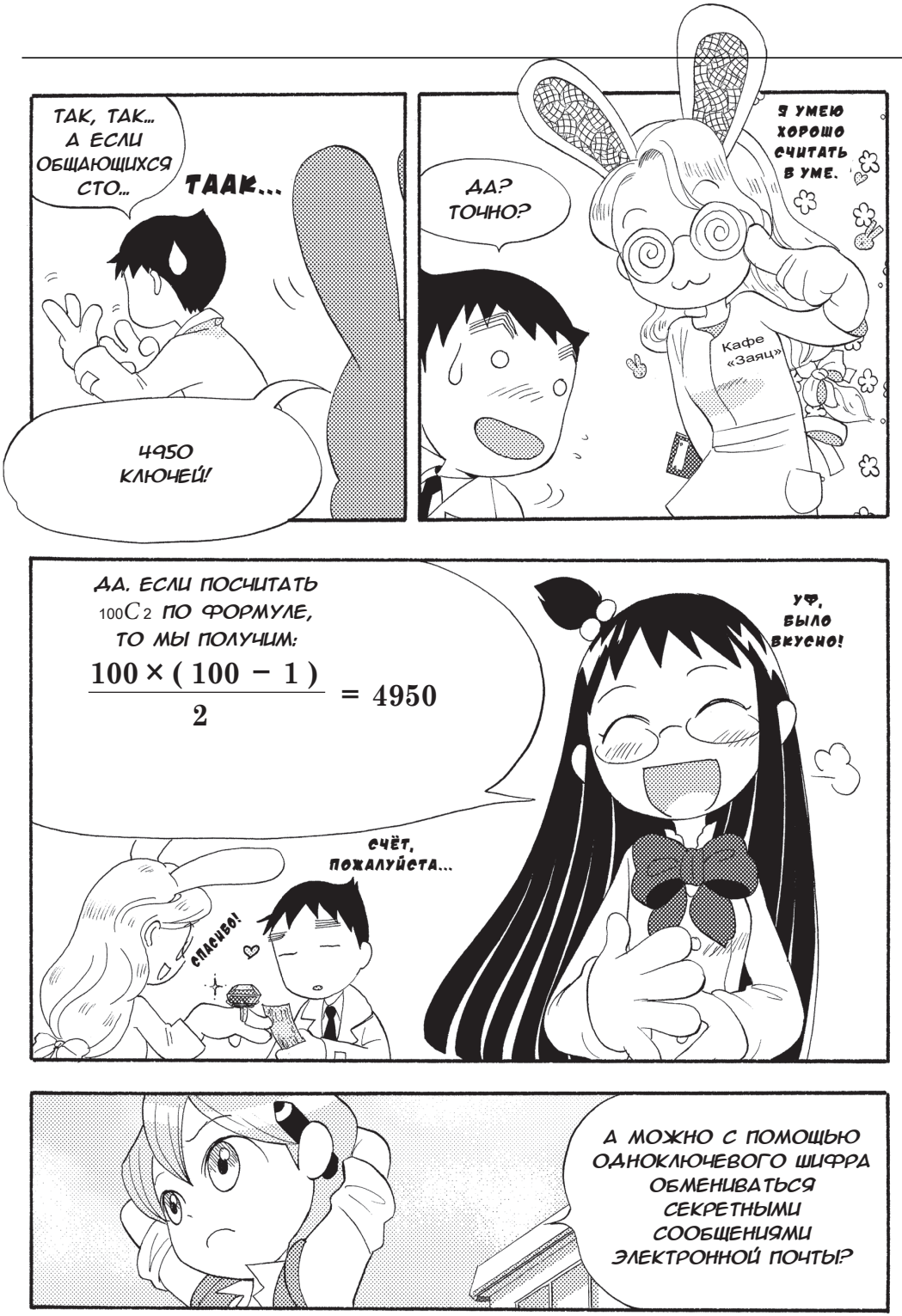


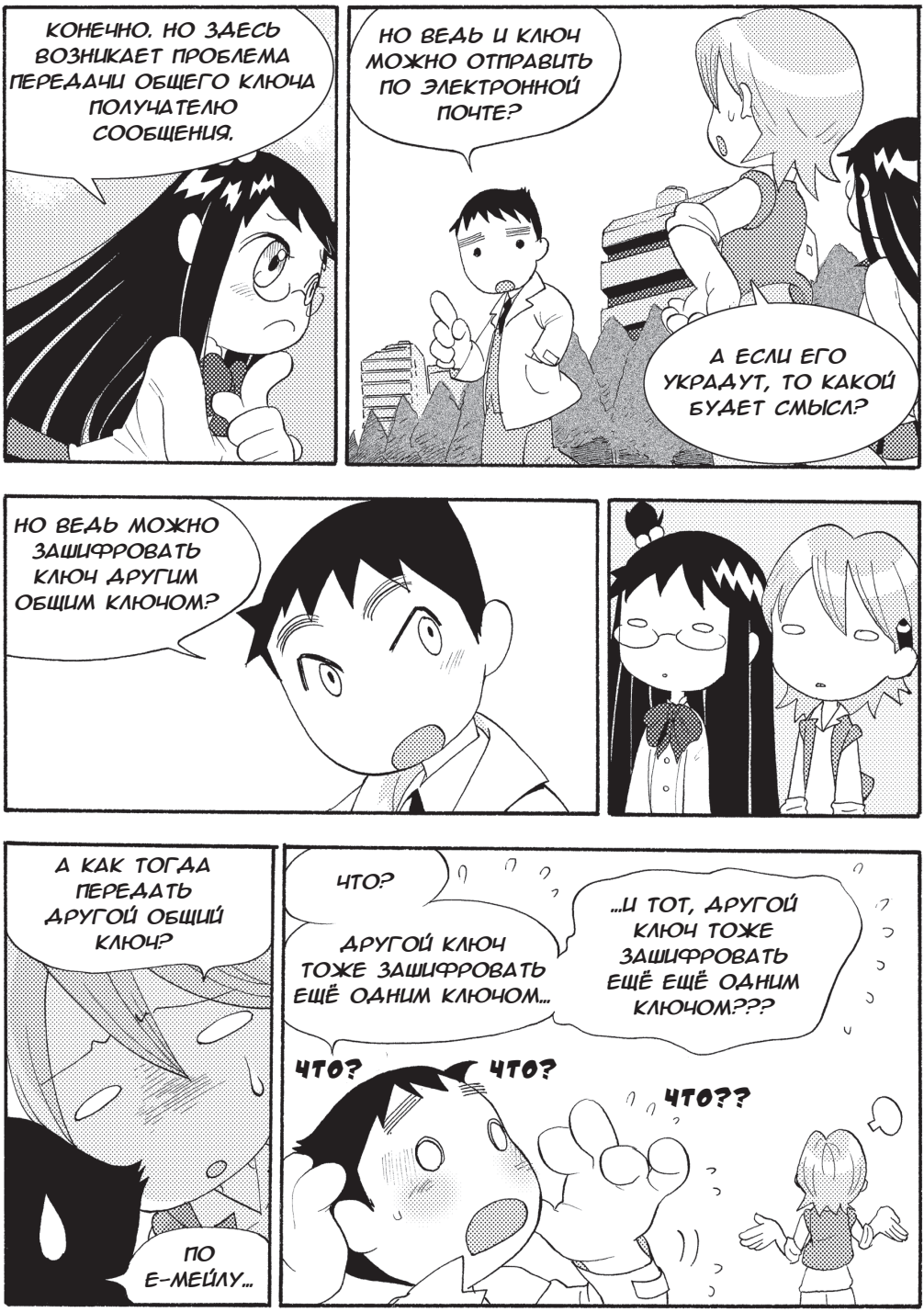


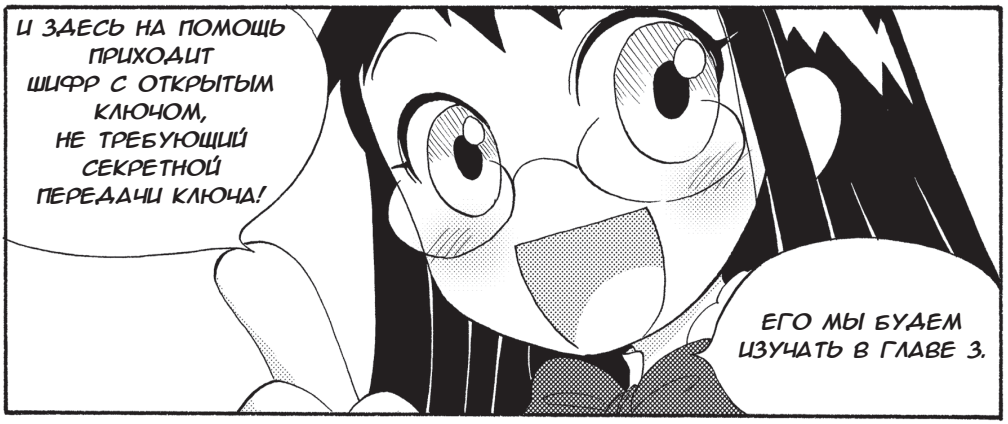
Одноключевой шифр (Common Key Cryptography) за его особенности называют также симметричным шифром (Symmetric Key Cryptography), или шифром с секретным ключом (Secret Key Cryptorgathy).
Все исторические шифры тоже были одноключевыми.




ЕСЛИ ТРОИМ – ТРИ...,
...ЗНАЧИТ, ЧЕТВЕРЫМ – ЧЕТЫРЕ?
НЕТ!
ДА, ТУТ ЛЕГКО ОШИБИТЬСЯ. ПОНАДОБИТСЯ ШЕСТЬ КЛЮЧЕЙ, НЕ ТАК ЛИ?
КОЛИЧЕСТВО КЛЮЧЕЙ, КОТОРОЕ ТРЕБУЕТСЯ ДЛЯ ВЗАИМНОГО ОБЩЕНИЯ n ПОЛЬЗОВАТЕЛЕЙ С ПОМОЩЬЮ ЭТОГО ШИФРА,
РАСЧИТЫВАЕТСЯ ВОТ ПО ТАКОЙ ФОРМУЛЕ!
Количество общих ключей $= {}_nC_2 = \frac{n(n-1)}{2}$

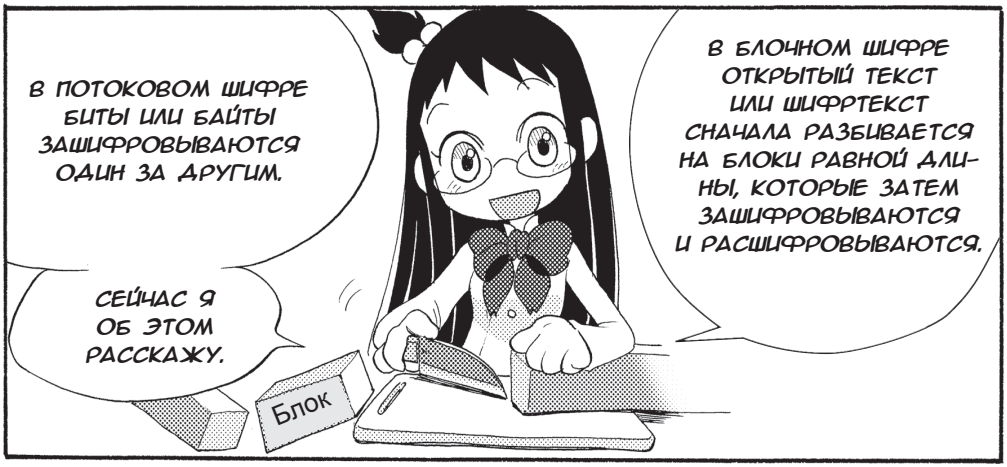

ТАК, ТАК... А ЕСЛИ ОБЩАЮЩИХСЯ СТО...
ТААК...
4950 КЛЮЧЕЙ!
ДА? ТОЧНО?
Я УМЕЮ ХОРОШО СЧИТАТЬ В УМЕ.
Кафе «Заяц»
ДА. ЕСЛИ ПОСЧИТАТЬ ${}_{100}C_2$ ПО ФОРМУЛЕ, ТО МЫ ПОЛУЧИМ:
$\frac{100 \times (100 - 1)}{2} = 4950$
УФ, БЫЛО ВКУСНО!
СЧЁТ, ПОЖАЛУЙСТА...
СПАСИБО!
А МОЖНО С ПОМОЩЬЮ ОДНОКЛЮЧЕВОГО ШИФРА ОБМЕНИВАТЬСЯ СЕКРЕТНЫМИ СООБЩЕНИЯМИ ЭЛЕКТРОННОЙ ПОЧТЫ?


КОНЕЧНО. НО ЗДЕСЬ ВОЗНИКАЕТ ПРОБЛЕМА ПЕРЕДАЧИ ОБЩЕГО КЛЮЧА ПОЛУЧАТЕЛЮ СООБЩЕНИЯ.
НО ВЕДЬ И КЛЮЧ МОЖНО ОТПРАВИТЬ ПО ЭЛЕКТРОННОЙ ПОЧТЕ?
А ЕСЛИ ЕГО УКРАДУТ, ТО КАКОЙ БУДЕТ СМЫСЛ?
НО ВЕДЬ МОЖНО ЗАШИФРОВАТЬ КЛЮЧ ДРУГИМ ОБЩИМ КЛЮЧОМ?
А КАК ТОГДА ПЕРЕДАТЬ ДРУГОЙ ОБЩИЙ КЛЮЧ?
ПО Е-МЕЙЛУ...
ЧТО?
ДРУГОЙ КЛЮЧ ТОЖЕ ЗАШИФРОВАТЬ ЕЩЁ ОДНИМ КЛЮЧОМ...
...И ТОТ, ДРУГОЙ КЛЮЧ ТОЖЕ ЗАШИФРОВАТЬ ЕЩЁ ЕЩЁ ОДНИМ КЛЮЧОМ???
ЧТО?
ЧТО?
ЧТО??

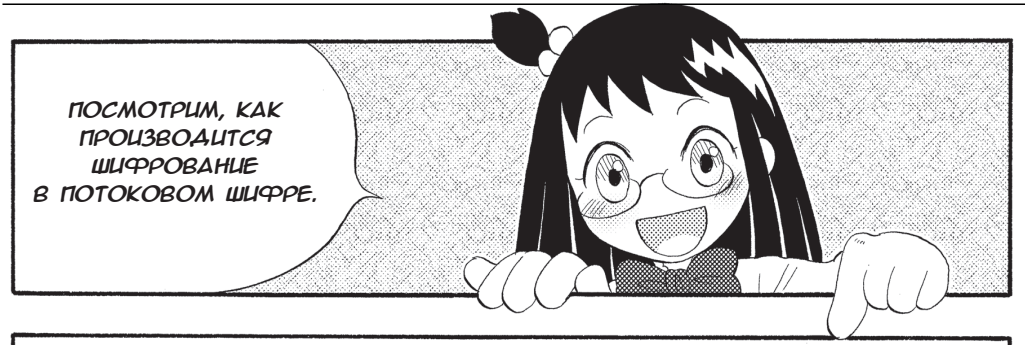


✤ Особенности одноключевого шифра

- Требуется осторожность при передаче и хранении ключа, так как его нужно держать в секрете.
- Подходит для передачи больших объёмов данных, так как низкая сложность вычислений позволяет проводить шифрование и расшифрование с высокой скоростью.
- Необходимо хранить большое количество ключей, что делает шифр не подходящим для коммуникации с неограниченно большим количеством партнёров.



2-3 Устройство потокового шифра



В потоковом шифре шифрование и расшифрование осуществляется последовательно, поэтому он применяется в связи, например в мобильных телефонах.

В качестве ключа используется длинная последовательность псевдослучайных чисел (чисел, как бы взятых «с потолка»), сгенерированная на компьютере. Для шифрования достаточно последовательно выполнять операцию XOR над данными открытого текста и ключа.

Позволяет обрабатывать данные быстро благодаря простоте алгоритма, по сравнению с блочным шифром.

Типичные алгоритмы потокового шифра: Salsa20, SEAL и др.

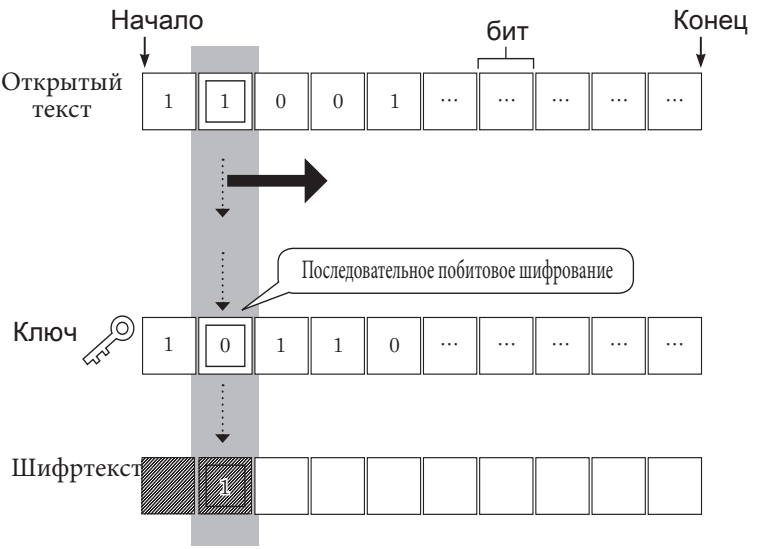
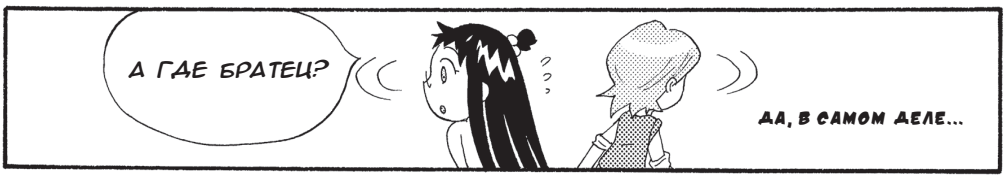


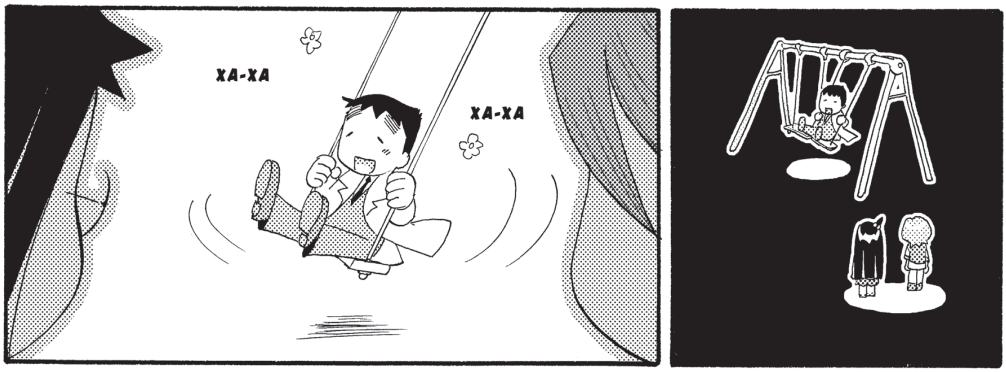
Рис. 2.2. Устройство потокового шифра


А ЧТО ТАКОЕ ПОСЛЕДОВАТЕЛЬНОСТЬ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ?
ЭТО ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ, ПОХОЖИХ НА СЛУЧАЙНЫЕ ЧИСЛА!
(См. стр. 225)


КСТАТИ, И ПОТОКОВЫЙ, И БЛОЧНОЙ ШИФРЫ МОЖНО КОГДА-НИБУДЬ ВЗЛОМАТЬ, ПЕРЕБИРАЯ КЛЮЧИ ОДИН ЗА ДРУГИМ.
(См. стр. 79)
ДРУГИМИ СЛОВАМИ, ОНИ ОБЛАДАЮТ ТОЛЬКО ВЫЧИСЛИТЕЛЬНОЙ СТОЙКОСТЬЮ.
ДА


А ПОСЛЕДОВАТЕЛЬНОСТЬ НАСТОЯЩИХ СЛУЧАЙНЫХ ЧИСЕЛ СГЕНЕРИРОВАТЬ НЕЛЬЗЯ?
ВЕДЬ, СГЕНЕРИРОВАВ ПОСЛЕДОВАТЕЛЬНОСТЬ ИСТИННО СЛУЧАЙНЫХ ЧИСЕЛ ДЛИНОЙ НЕ МЕНЕЕ ОТКРЫТОГО ТЕКСТА, МЫ ПОЛУЧИМ СОВЕРШЕННО СТОЙКИЙ ШИФР ВЕРНАМА!
НА КОМПЬЮТЕРЕ ЭТО СДЕЛАТЬ СЛОЖНО.


А ГДЕ БРАТЕЦ?
ДА, В САМОМ ДЕЛЕ...


ХА-ХА
ХА-ХА

2-4 Устройство блочного шифра

В отличие от потокового шифра, в котором используется побитовое шифрование, в блочном шифре информация зашифровывается блоками определённой длины (рис. 2.3). Длина одного блока зависит от разновидности шифра, имеются 64-битные, 128-битные блочные шифры. Кстати, так как однобайтовый символ кодируется восемью битами, 64-битный блок содержит 8 символов.

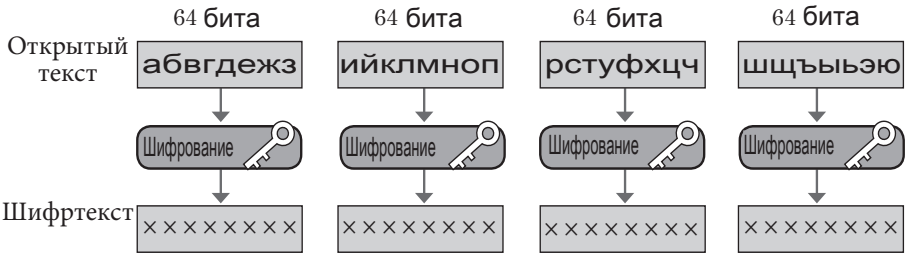
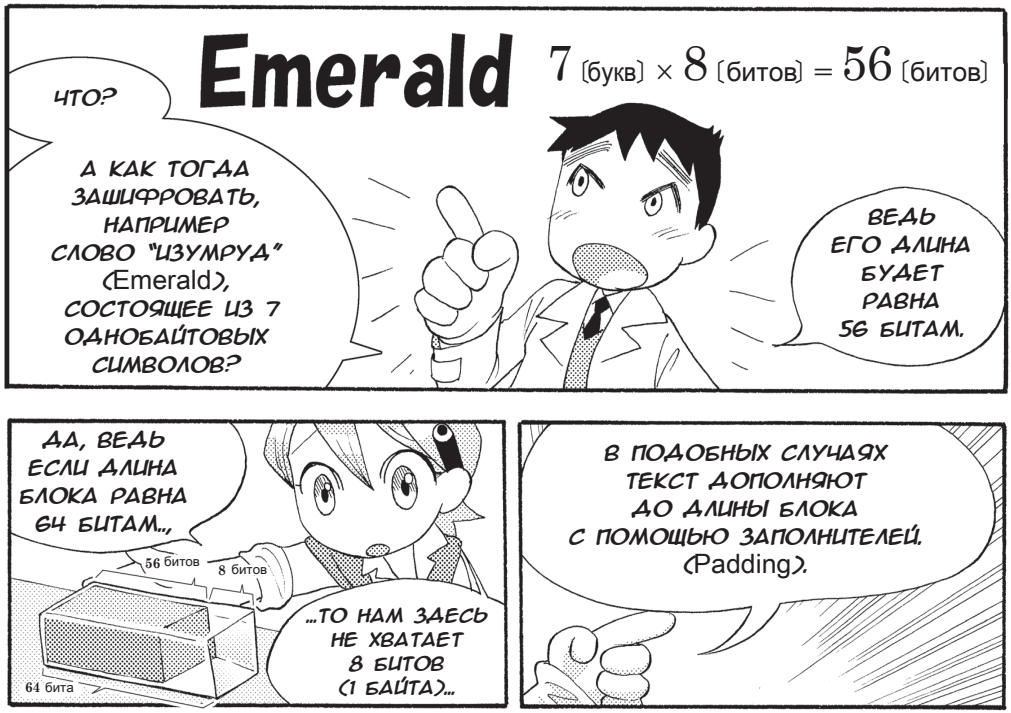


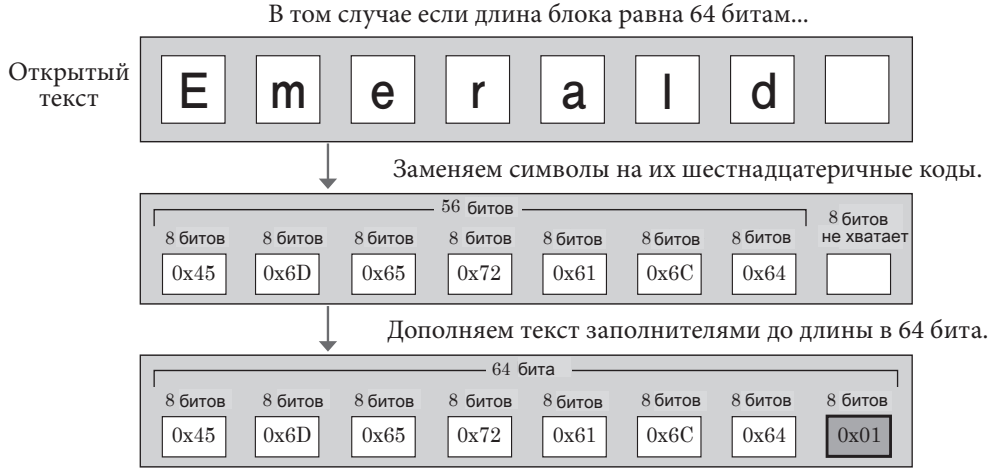
Рис. 2.3. Устройство блочного шифра

Длину блока сделали равной 64 битам потому, что высокопроизводительные компьютеры могут обрабатывать блоки такой длины с меньшим количеством вычислений, а также потому, что при малом числе битов снижается безопасность блочного шифра.

Таблица 2.4. Разновидности блочных шифров

Название шифра	Длина блока в битах	Длина ключа в битах
DES	64	64
AES	128	128 192 256





※ Коды однобайтовых символов выражены в виде двухзначных шестнадцатеричных чисел (на это указывает префикс 0x перед числами).

Рис. 2.4. Пример дополнения битами-заполнителями

В вышеприведённом примере текст дополняется до длины блока в 64 бита (8 байтов) с помощью байта-заполнителя. Этот байт, равный 0x01, то есть 1, показывает число байтов-заполнителей, которое будет удалено при расшифровании. Существуют и другие методы дополнения.

Пусть в открытом тексте два раза встречается последовательность символов «**my love,**».

…my love,……my love ,

↓ Шифрование

…/"!#$&%*……/"!#$&%*

Тогда в шифртексте будет два раза повторяться последовательность символов «/»**!$&%***», что может стать зацепкой для раскрытия шифра.

❀ Режим сцепления блоков шифртекста (CBC)

Режим CBC – это метод, позволяющий получать на выходе криптосистемы разный шифртекст при одинаковом открытом тексте на её входе.

После разбиения открытого текста блоки входных или выходных данных используются повторно (это называется «обратной связью»), что позволяет рассеивать непрерывные данные открытого текста между блоками, повышая тем самым стойкость шифра※.

Рис. 2.5. Устройство режима CBC

В случае, показанном на рис. 2.5, предыдущий блок шифртекста C_{i-1} складывается по модулю 2 со следующим блоком открытого текста P_i :

$$X_i = C_{i-1} \oplus P_i \text{ , где } i = 1, 2, 3...,$$

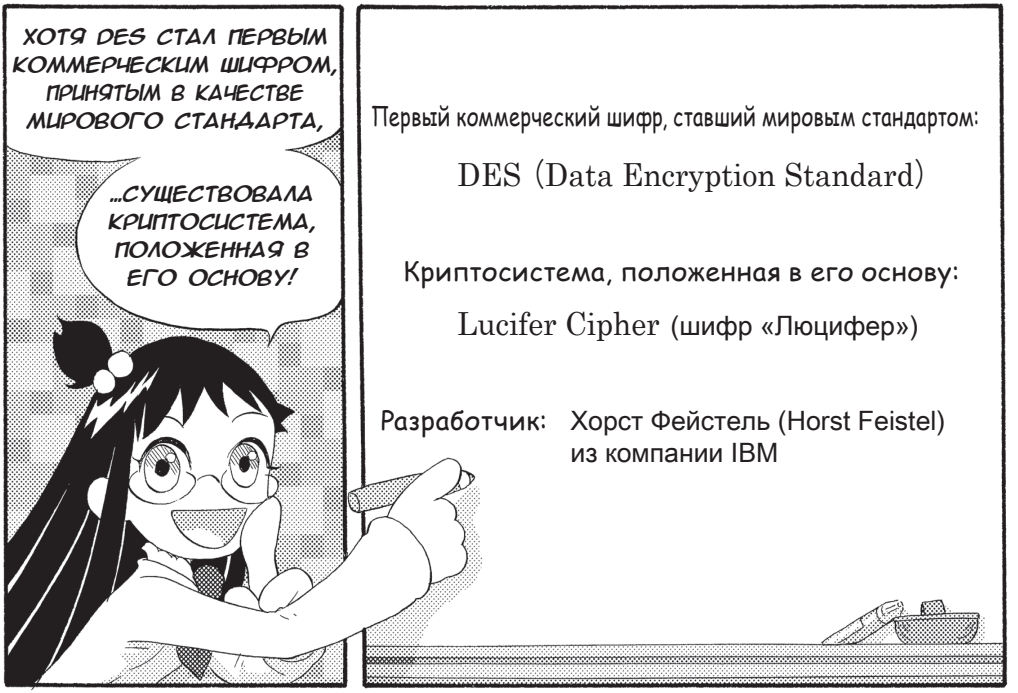
и результат этой операции X_i используется в качестве входных данных шифрования. Таким образом, даже при наличии одинаковых блоков открытого текста одинаковые блоки шифртекста сгенерированы не будут. Правда, для самого первого блока X_1 входных данных приходится в качестве предыдущего блока выходных данных использовать заранее сформированный вектор инициализации C_0.

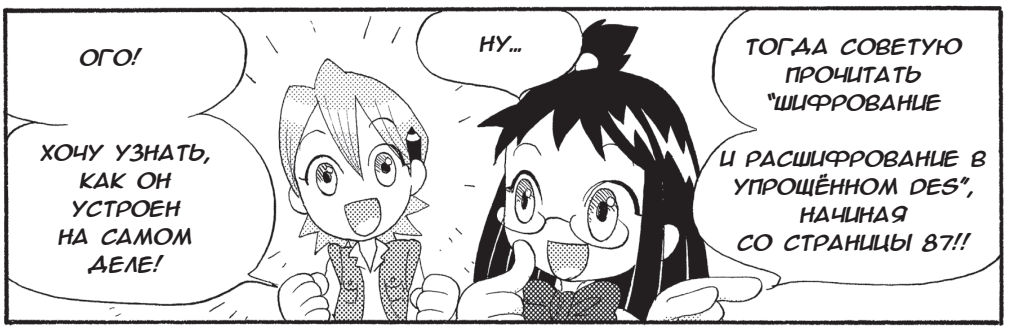
Правило использования нескольких блоков называется «режимом шифрования». Кроме вышеописанного режима CBC и описанного на предыдущей страницы режима ECB, существуют также такие режимы, как OFB, CFB, и другие.

※ Однако на больших объёмах данных у режима CBC есть уязвимости – *Прим. ред.*

2-5 Устройство шифра DES

※ Для более глубокого понимания устройства шифра DES рекомендуем прочитать раздел «Шифрование и расшифрование в упрощённом DES», начинающийся со стр. 87.

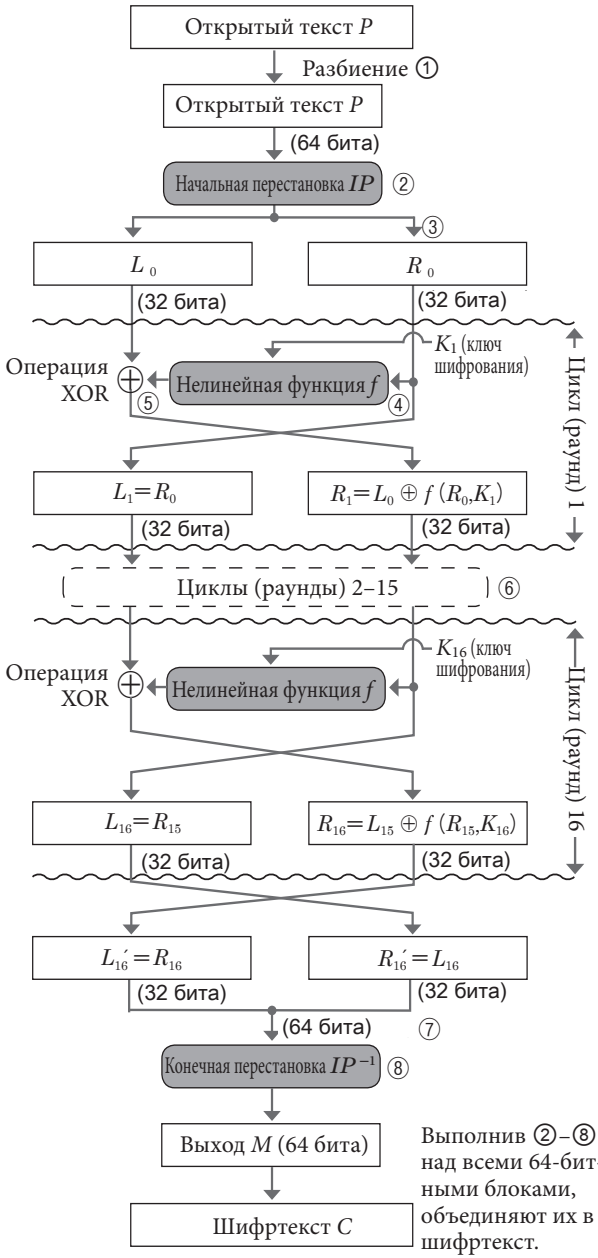




Основы строения сети Фейстеля

Фейстель предложил следующий метод шифрования информации.

※ *IP* : Initial Permutation



① Разбиваем открытый текст на 64-битные блоки.

② Выполняем начальную перестановку *IP* битов 64-битного блока.

③ Делим 64-битный блок на два 32-битных блока: левый L_0 и правый R_0.

④ Выполняем замену и перестановку битов блока R_0 с помощью нелинейной функции *f*, использующей ключ шифрования K_1.

⑤ Сложив по модулю 2 блок L_0 и значение функции $f(R_0, K_1)$, получаем правый 32-битный блок R_1. Левый 32-битый блок L_1 принимаем равным R_0.

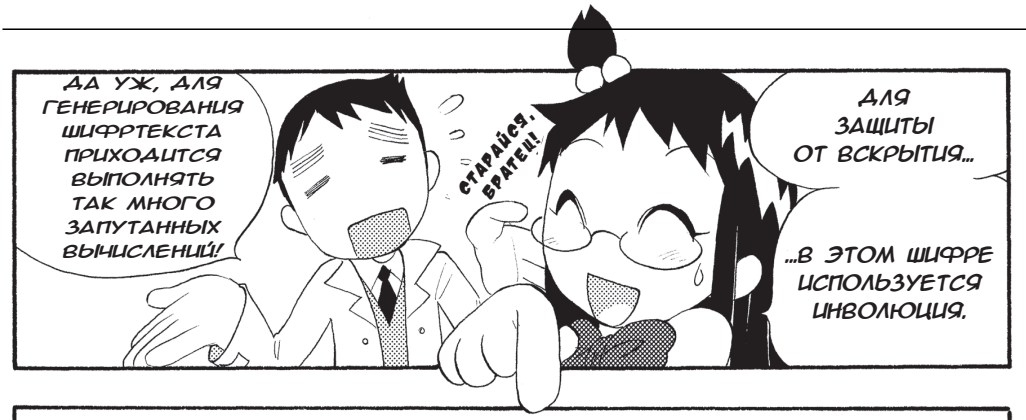
⑥ Выполняем ④–⑤ в цикле ещё 15 раз (раунды 2–16).

⑦ Объединяем левый и правый 32-битные блоки L_{16} и R_{16} в один 64-блок.

⑧ Выполняем конечную перестановку IP^{-1} битов 64-битного блока, обратную начальной перестановке *IP*.

Выполнив ②–⑧ над всеми 64-битными блоками, объединяют их в шифртекст.

Рис. 2.6. Порядок шифрования DES (генерирование шифртекста)



Инволюция

Инволюция – это такое преобразование, при двухкратном применении которого происходит возврат к первоначальному виду.

Рассмотрим, например, такое преобразование, при котором 1 превращается в 4, 2 – в 3, 3 – в 2, 4 – в 1. В этом преобразовании каждому входному значению соответствует одно выходное значение. Попробуем выполнить его два раза.

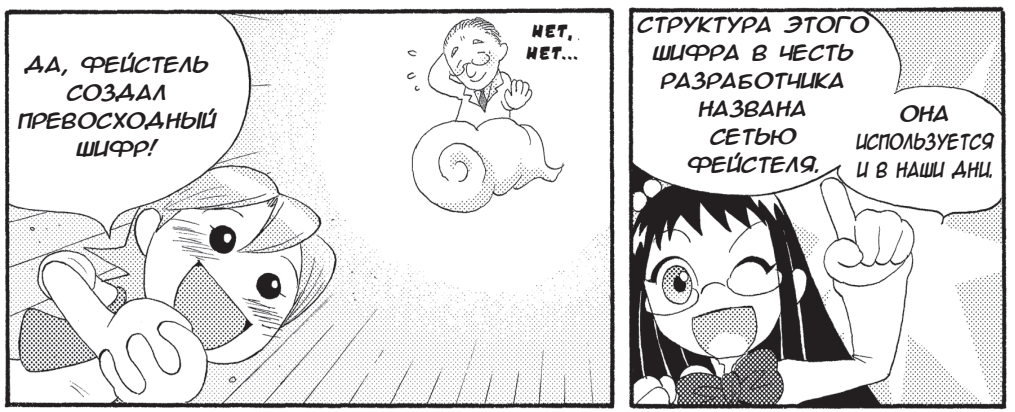
$1 \rightarrow 4 \rightarrow 1$

$2 \rightarrow 3 \rightarrow 2$

$3 \rightarrow 2 \rightarrow 3$

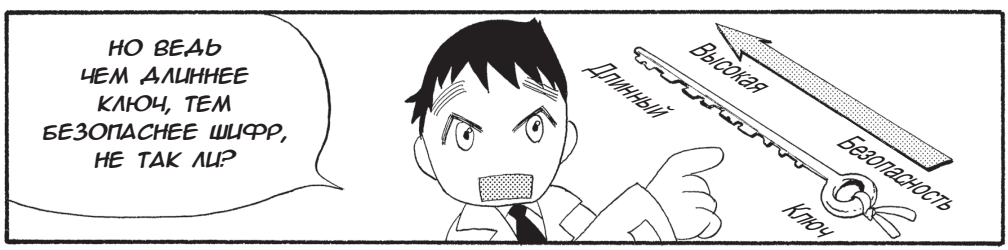
$4 \rightarrow 1 \rightarrow 4$

Как видите, мы получили первоначальные значения. Такое преобразование называется инволюцией, которая тесно связана с механизмом шифрования и расшифрования DES.

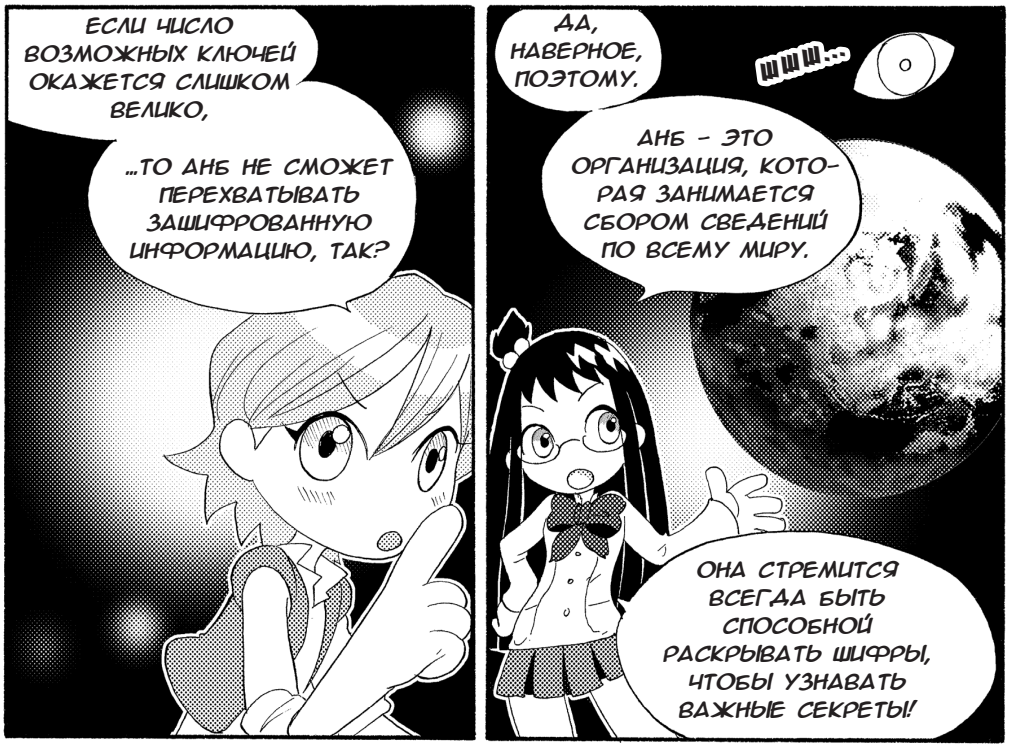


Длина ключа DES по спецификации равна 64 битам, но в действительности в качестве ключа используется 56 бит, а остальные 8 бит нужны для проверки чётности.

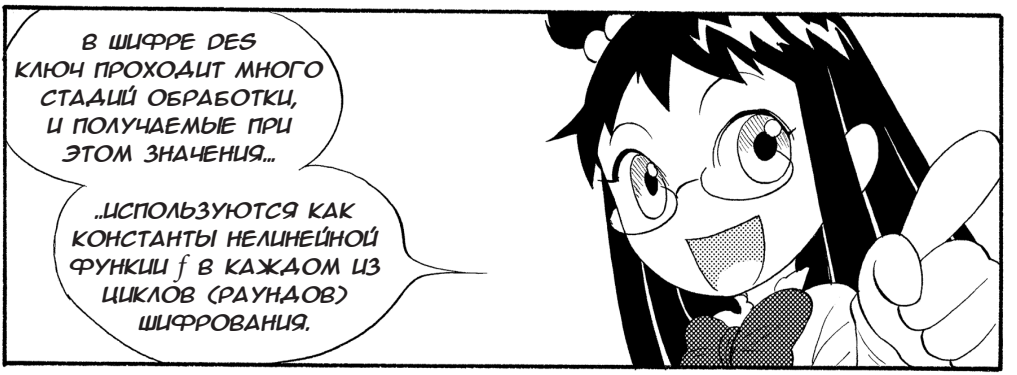
Проверка чётности проводится для обнаружения ошибок, которые могут возникнуть в данных под действием шумов, в результате ошибки чтения и т. п.



※Миллион– 10^6, миллиард – 10^9, триллион – 10^{12}, квадриллион – 10^{15}, квинтиллион – 10^{18}.

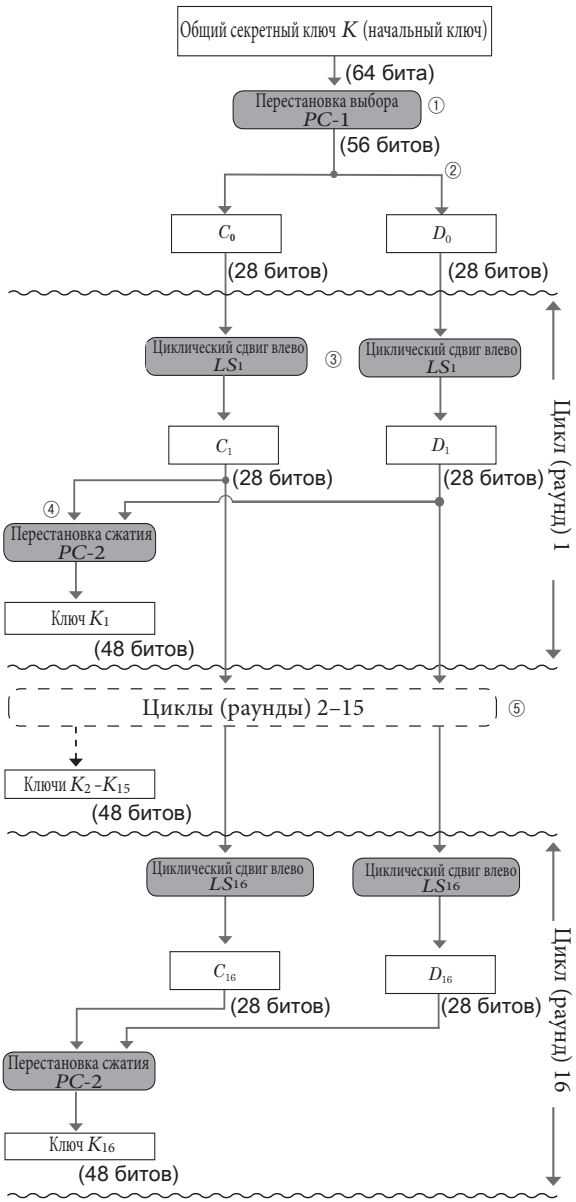

ЕСЛИ ЧИСЛО ВОЗМОЖНЫХ КЛЮЧЕЙ ОКАЖЕТСЯ СЛИШКОМ ВЕЛИКО,
...ТО АНБ НЕ СМОЖЕТ ПЕРЕХВАТЫВАТЬ ЗАШИФРОВАННУЮ ИНФОРМАЦИЮ, ТАК?
ДА, НАВЕРНОЕ, ПОЭТОМУ.
ШШШ...
АНБ – ЭТО ОРГАНИЗАЦИЯ, КОТОРАЯ ЗАНИМАЕТСЯ СБОРОМ СВЕДЕНИЙ ПО ВСЕМУ МИРУ.
ОНА СТРЕМИТСЯ ВСЕГДА БЫТЬ СПОСОБНОЙ РАСКРЫВАТЬ ШИФРЫ, ЧТОБЫ УЗНАВАТЬ ВАЖНЫЕ СЕКРЕТЫ!


БРР...
ОЙ, КАК СТРАШНО...


В ШИФРЕ DES КЛЮЧ ПРОХОДИТ МНОГО СТАДИЙ ОБРАБОТКИ, И ПОЛУЧАЕМЫЕ ПРИ ЭТОМ ЗНАЧЕНИЯ...
...ИСПОЛЬЗУЮТСЯ КАК КОНСТАНТЫ НЕЛИНЕЙНОЙ ФУНКЦИИ f В КАЖДОМ ИЗ ЦИКЛОВ (РАУНДОВ) ШИФРОВАНИЯ.

✤ Генерирование ключей шифрования DES

Ключи K_1, K_2, K_3, .., K_{16}, используемые для шифрования DES, генерируются так, чтобы ключи, используемые в каждом из раундов, отличались друг от друга.



※ *PC:* Permuted Choice

① Удаляем 8 битов проверки чётности из 64-битного начального ключа и переставляем остальные биты.

② Делим 56 битов на два 28-битных блока: левый C_0 и правый D_0.

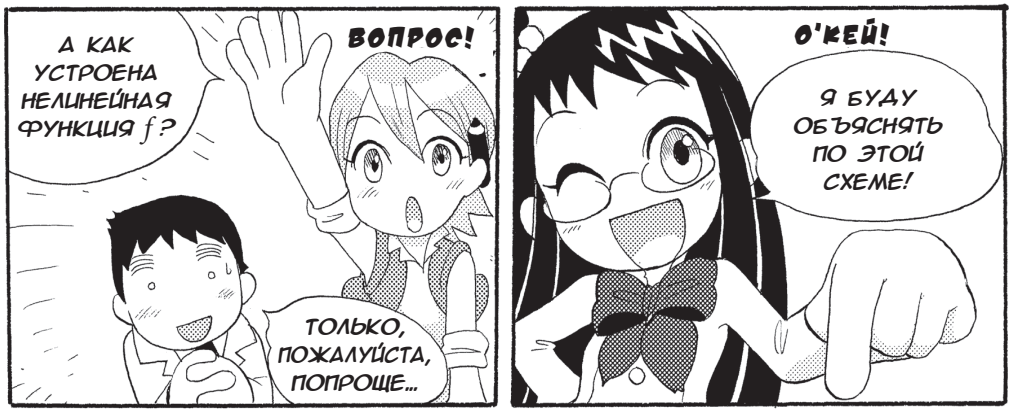
③ Циклически сдвинув влево на определённое число битов (LS_1) блоки C_0 и D_0, получаем C_1 и D_1.

④ Объединив C_1 и D_1, переставляем биты со сжатием (PC-2) и получаем 48-битный ключ K_1.

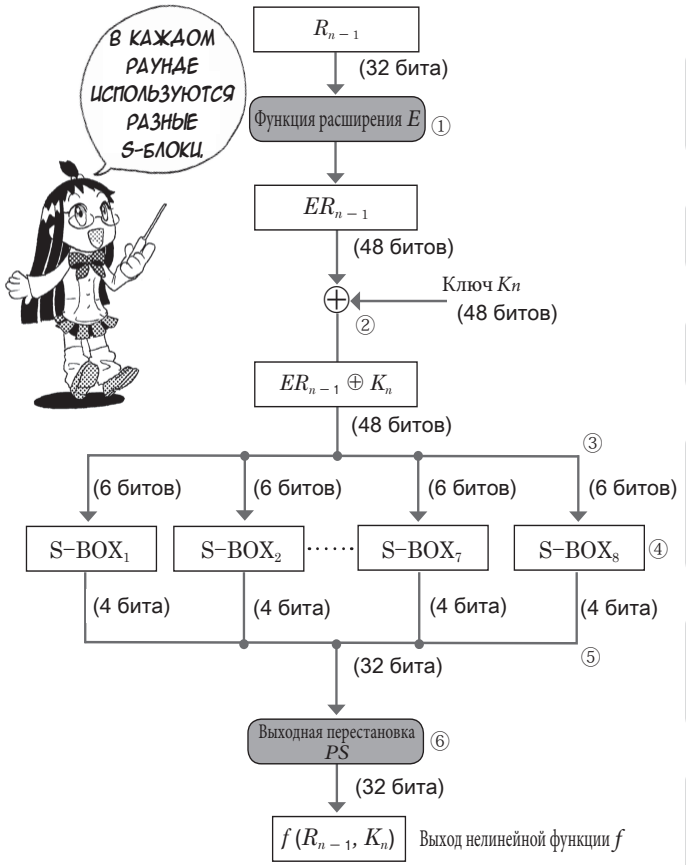
⑤ Выполняя вычисления ③–④ в цикле, получаем ключи K_n (n = 2, 3,.., 16) для каждого из раундов шифрования.

※ При генерировании ключей расшифрования вместо циклических сдвигов влево выполняем циклические сдвиги вправо. Кроме того, ключи расшифрования генерируются в обратном порядке: K_{16}, K_{15},.., K_1.

Рис. 2.7. Порядок генерирования ключей шифрования и расшифрования DES



✤Устройство нелинейной функции *f* шифра DES



① Выполняем расширение введённого 32-битного блока до 48 битов (длина ключа).

② Полученный 48-битный блок складываем по модулю 2 с ключом.

③ Разбиваем результат на восемь 6-битных блоков.

④ Преобразуем все 6-битные блоки в 4-битные с помощью восьми S-блоков (таблиц замены).

⑤ Объединяем 4-битные блоки в 32-битный блок в порядке их следования.

⑥ Выполнив выходную перестановку *PS*, получаем выходное значение функции *f*.

Рис. 2.8. Устройство нелинейной функции *f* шифра DES

✤ Обобщённая модель шифрования и расшифрования DES

Шифрование и расшифрование DES проводится следующим образом. Шифрование открытого текста и расшифрование шифртекста представляют собой взаимно обратные процессы.

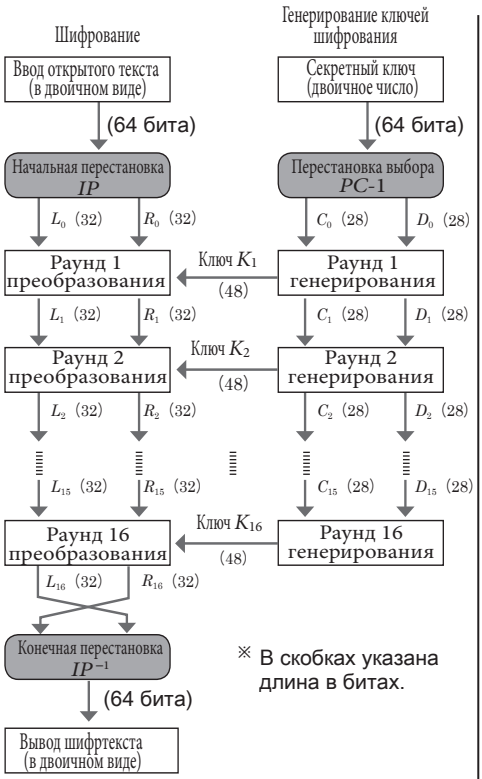


Рис. 2.9. Модель шифрования DES

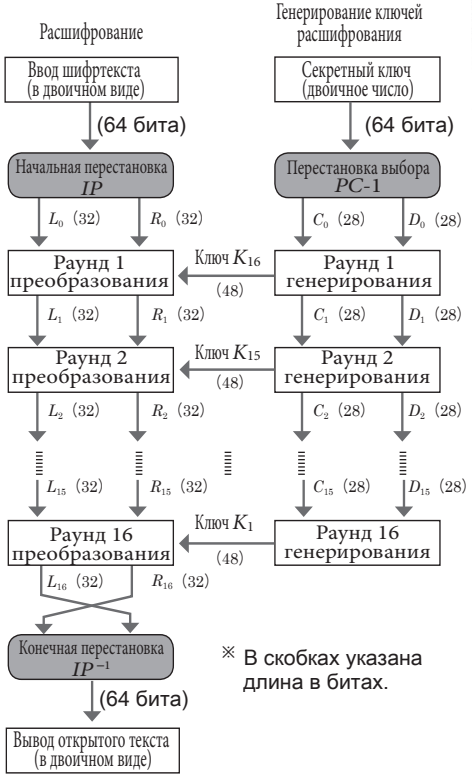
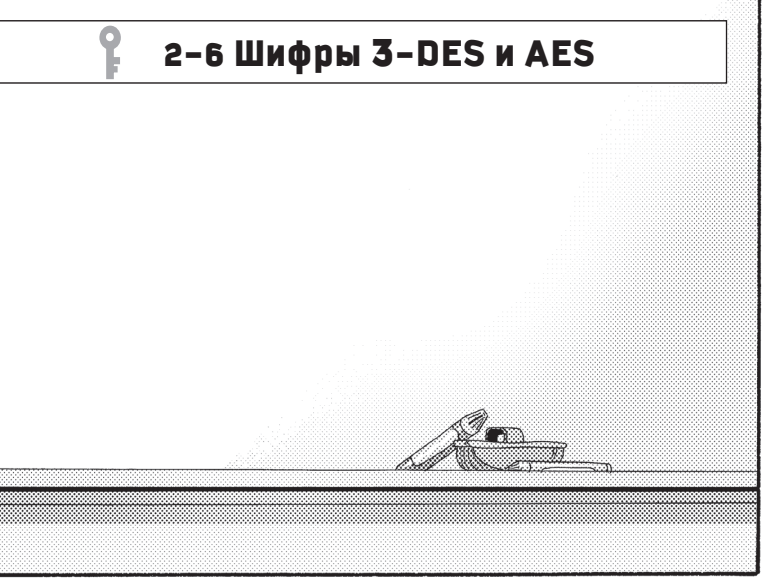
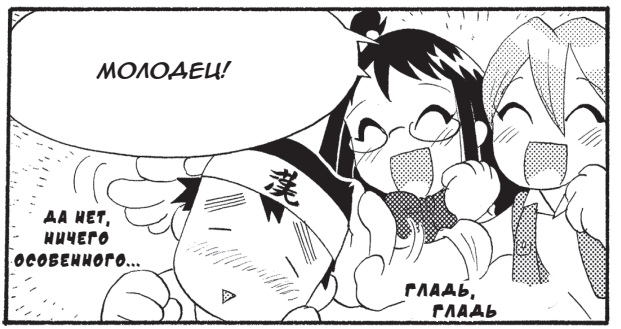


Рис. 2.10. Модель расшифрования DES

2-6 Шифры 3-DES и AES





※ Надпись на повязке: «Храбость».

Недостатки шифра DES

- Малая длина ключа: замедление скорости обработки и снижение стойкости шифра.
- Отсутствие стандарта на S-блоки: возможно появление слабых реализаций.

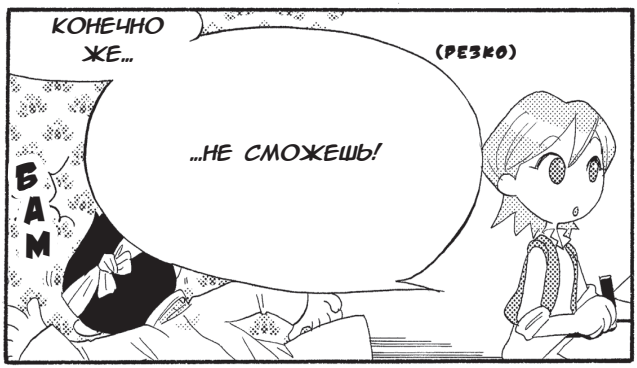
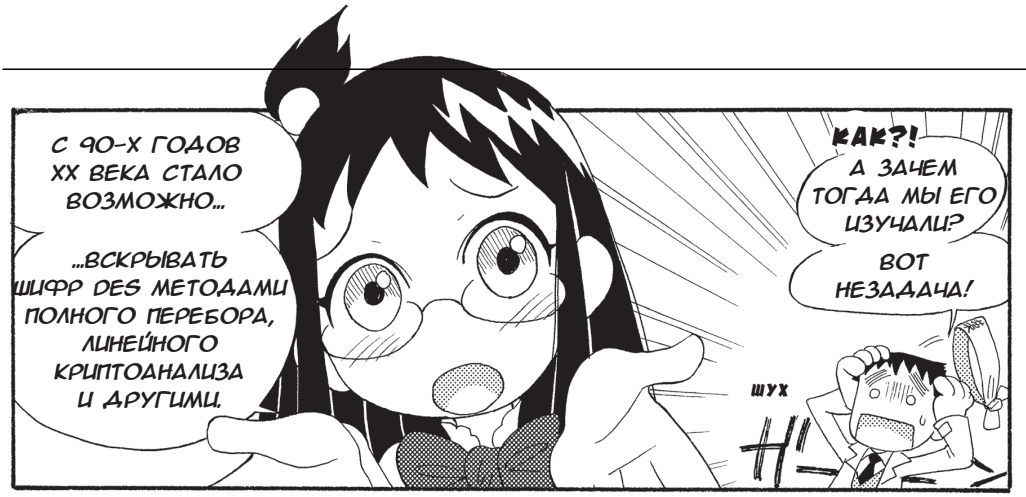


Таблица 2.3. Методы криптоанализа блочных шифров

Полный перебор (метод «грубой силы»)	Проверяют один за другим все возможные ключи
Дифференциальный криптоанализ	Метод поиска ключа, основанный на свойстве операции XOR: разность (результат сложения по модулю 2) входных данных передаётся на выход без изменений
Линейный криптоанализ	Метод вероятностной оценки выходных данных, основанный на линейной аппроксимации S-блоков



※ 3-DES уже считается небезопасным – *Прим. ред.*

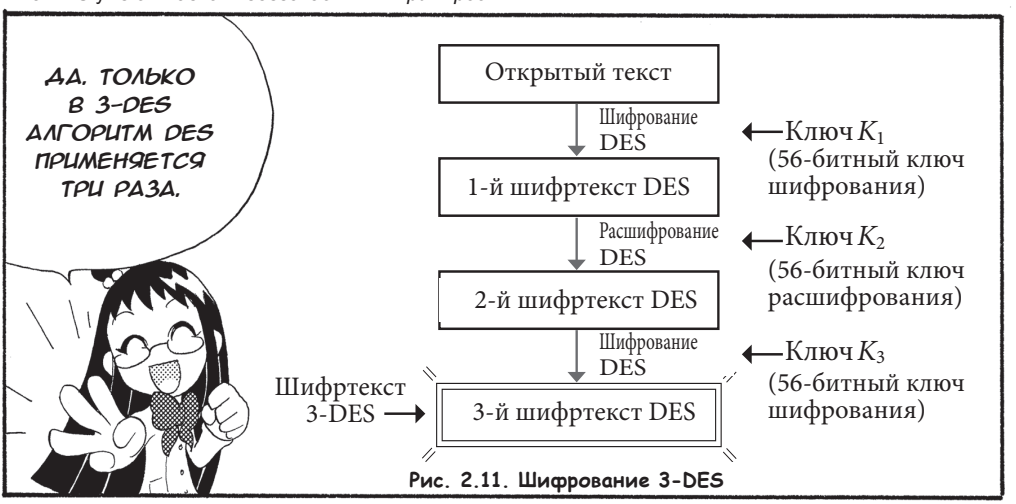
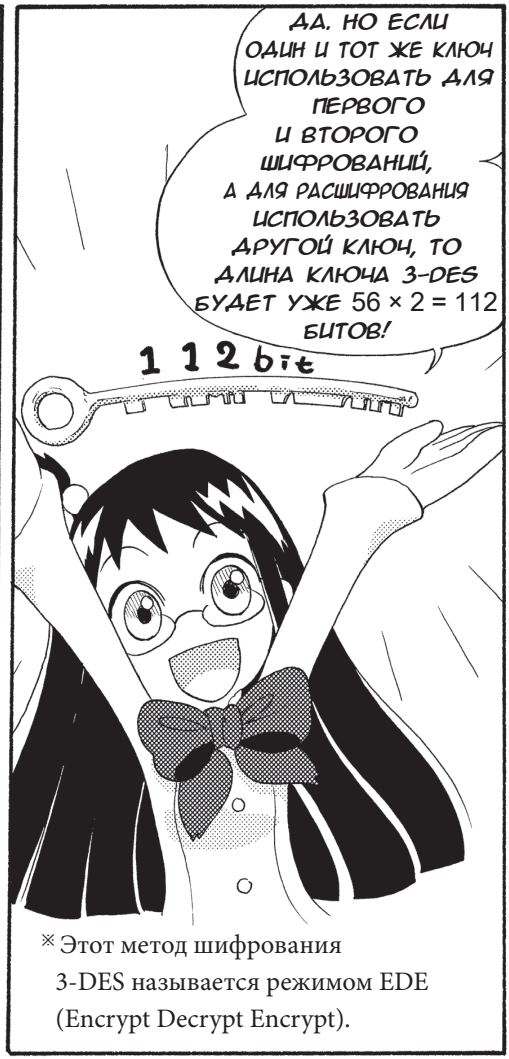
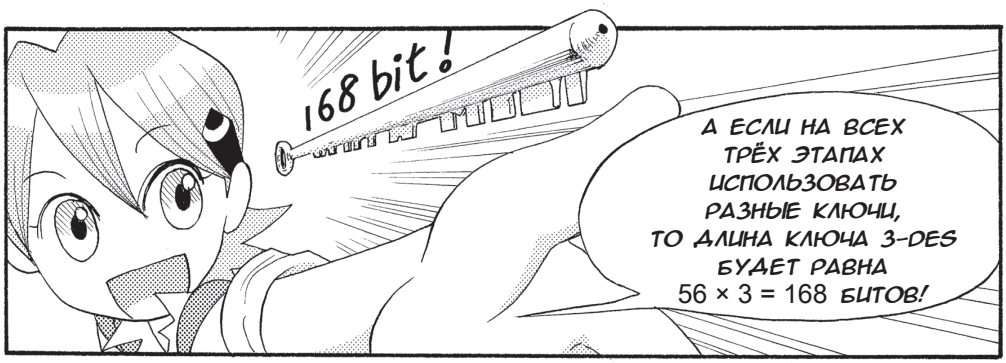


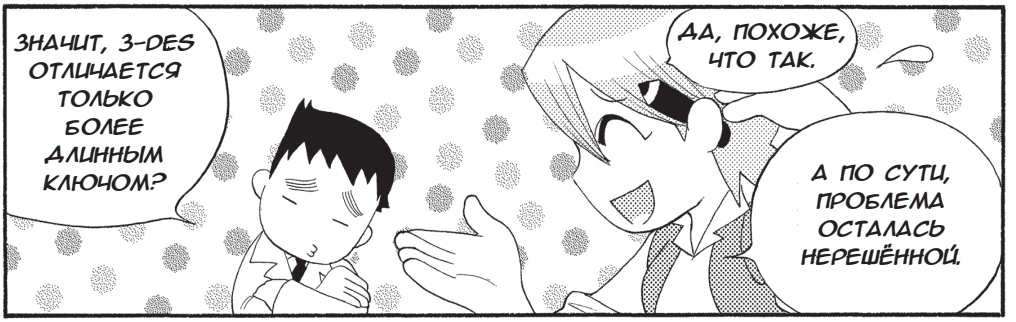
Рис. 2.11. Шифрование 3-DES

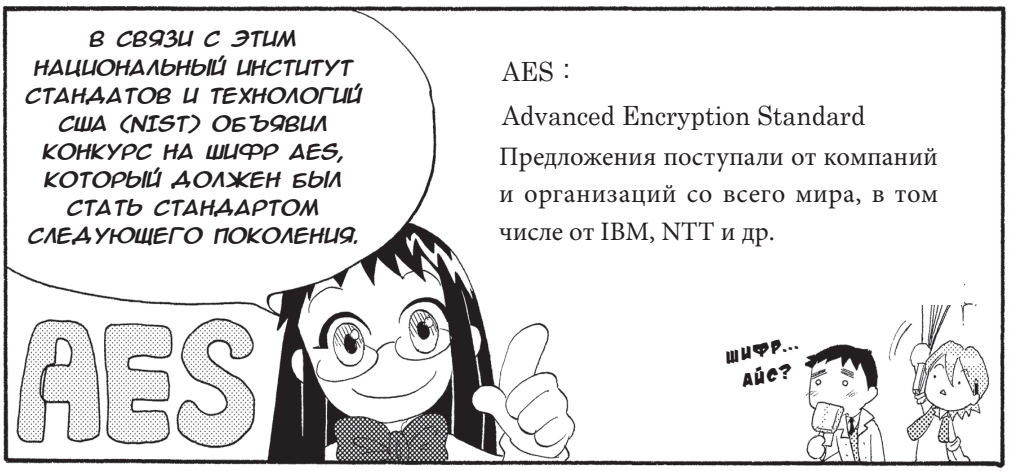
※ В этом случае эффективная длина ключа так и останется равна 56 битам.



※ Этот метод шифрования 3-DES называется режимом EDE (Encrypt Decrypt Encrypt).







AES :

Advanced Encryption Standard

Предложения поступали от компаний и организаций со всего мира, в том числе от IBM, NTT и др.

✤ Общие сведения о шифре AES

В 2000 году алгоритм «Рэндел» был выбран в качестве стандарта FIPS (Federal Information Processing Standarts: федеральные стандарты обработки информации США). Название Rijndael (по-голландски произносится «Рэндел») составлено из имён разработчиков: Йоана Даймена (Joan Daemen) и Висцента Рэймена (Viscent Rijmen) из Лёвенского католического университета Бельгии.

В зависимости от длины ключа существует три типа AES, показанных в табл. 2.4.

Таблица 2.4. Типы шифра AES

Тип	Длина ключа в битах	Длина блока в битах	Число раундов
AES–128	128	128	10
AES–192	192	128	12
AES–256	256	128	14

Стойкость шифра тем выше, чем длиннее ключ и чем больше число раундов.

В основе данного шифра лежит не сеть Фейстеля, а подстановочно-перестановочная сеть (Substitution Permutation Network: SPN, SP-сеть), состоящая из множества раундов, в которых над результатом сложения по модулю 2 входного блока и ключа соответствующего раунда одновременно выполняются замены и перестановки.

Можно предположить, что шифр AES, обладающий высокой криптостойкостью, в будущем заменит шифр DES※.

※ AES уже по факту заменил DES – *Прим. ред.*.


ОКОЛО ОДНОГО ГОДА?
А ВОТ И НЕТ! 149 ТРИЛЛИОНОВ ЛЕТ, КАК СЧИТАЕТСЯ!
АА!!
НЕСМОТРЯ НА РАСКРЫТИЕ УСТРОЙСТВА S-БЛОКОВ И РАБОТУ МНОГИХ ИССЛЕДОВАТЕЛЕЙ,
...НИКАКИХ ПРОБЛЕМ НА ДАННЫЙ МОМЕНТ ОБНАРУЖИТЬ НЕ УДАЛОСЬ.
ЗНАЧИТ, AES ОСТАНЕТСЯ БЕЗОПАСНЫМ И В БУДУЩЕМ?!
НО КРИПТОАНАЛИЗ ТОЖЕ РАЗВИВАЕТСЯ!
НЕКОТОРЫЕ СЧИТАЮТ, ЧТО AES БУДЕТ ОСТАВАТЬСЯ БЕЗОПАСНЫМ ЕЩЁ ЛЕТ ДЕСЯТЬ.


ЭХ...
КСТАТИ, БРАТЕЦ,
КОГДА ЖЕ ТЫ, НАКОНЕЦ, КУПИШЬ МНЕ КОМПЬЮТЕР?! Я ВЕДЬ УЖЕ ТАК МНОГО РАССКАЗАЛА ПРО ШИФРОВАНИЕ!
А...
Я ЗНАЛ, ЧТО ТЫ ЭТО СКАЖЕШЬ, И КОЕ-ЧТО ПРИГОТОВИЛ!
ДА?! ПРАВДА?!
АХ!
АХ!
АХ!
ВОТ!
on
off
MR
M-
M+
AC
C


ЭТА ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА НАМНОГО УДОБНЕЕ, ЧЕМ СЧЁТЫ!
РАДУЙСЯ!
ДРР
ДРР
ДРР
Я ЖЕ ПРОСИЛА НЕ КАЛЬКУЛЯТОР...
...А КОМПЬЮТЕР!!
КАКОЙ ЖЕ ТЫ, БРАТЕЦ, ДУРАК...
ТЕПЕРЬ ПОПРАКТИКУЙТЕСЬ В ШИФРОВАНИИ И РАСШИФРОВАНИИ С ПОМОЩЬЮ ОПИСАНИЯ УПРОЩЁННОГО DES, НАЧИНАЮЩЕГОСЯ НА СЛЕДУЮЩЕЙ СТРАНИЦЕ.
ОЙ...
А В ГЛАВЕ 3 ВАС ЖДЁТ КОЕ-ЧТО ИНТЕРЕСНОЕ: Я ОТКРОЮ КЛЮЧ, КОТОРЫЙ ДО ЭТОГО БЫЛ СЕКРЕТНЫМ!

Пример использования упрощённого DES

Как же осуществляется шифрование и расшифрование DES? Попробуем разобраться в этом с помощью облегчённой версии DES.

Преобразование в двоичные данные

Данные, используемые во всех современных шифрах, включая DES, вообще говоря, являются двоичными, поэтому открытый текст, состоящий из букв и цифр, необходимо преобразовать в двоичные числа. Здесь мы будем использовать только 16 символов (один из которых является ничего не значащим «исключённым символом»). Преобразовав эти символы в соответствующие им 4-битные двоичные коды, мы выразим данные в виде ряда, состоящего из нулей и единиц.

Таблица 2.7. Символы и их двоичные коды

Символы	Двоичные коды
A	0000
B	0001
C	0010
D	0011
E	0100
F	0101
G	0110
H	0111

Символы	Двоичные коды
I	1000
J	1001
K	1010
L	1011
M	1100
N	1101
O	1110
(Исключённый символ)	1111

Генерирование шифртекста DES

Длина блока в реальном шифре DES равна 64 битам, но здесь мы будем использовать упрощённый шифр DES с 8-битным блоком и двумя раундами шифрования, что позволит более наглядно описать общие закономерности. Генерирование шифртекста DES основано на двух процессах: шифровании и генерировании ключей (рис. 2.12).

Сначала с помощью табл. 2.7 представим открытый текст, который мы хотим зашифровать, в виде последовательности нулей и единиц, как показано на рис. 2.12, а затем перемешаем 8-битные двоичные данные с помощью начальной перестановки *IP* в соответствии с табл. 2.8.

Смысл табл. 2.8 заключается в том, что, например, 1-й слева бит 8-битного блока введённого открытого текста становится 5-м слева битом на выходе начальной перестановки, 2-й слева бит –1-м слева битом и т. д. (рис. 2.13).

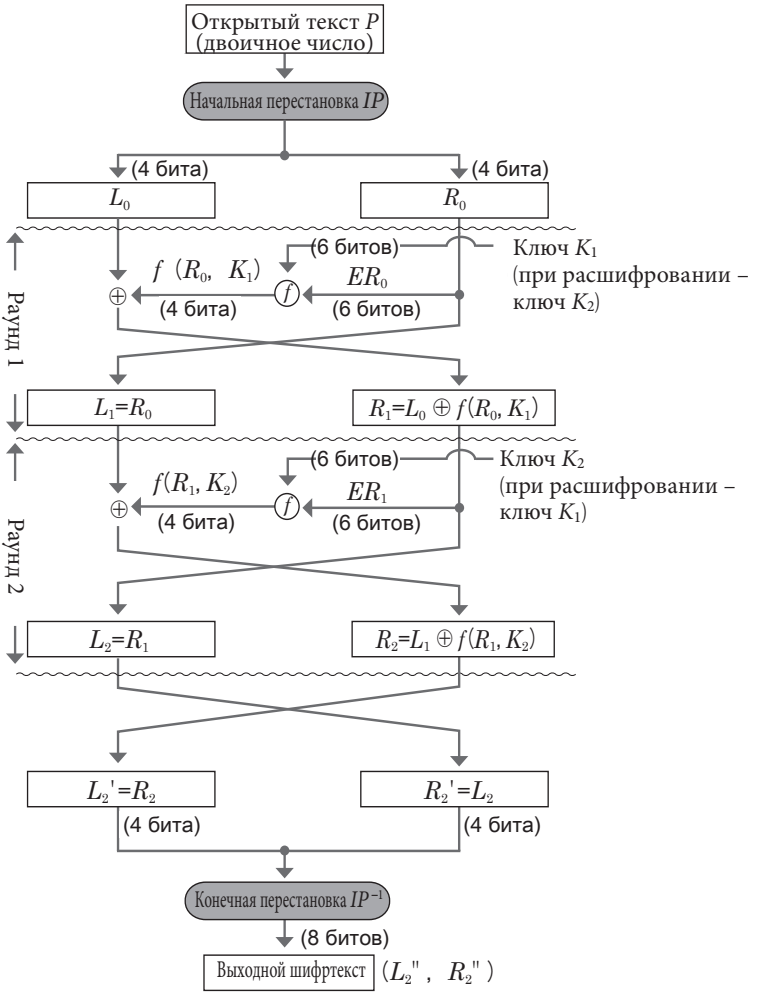


Рис. 2.12. Порядок генерирования шифртекста упрощённого DES

Таблица 2.8. Начальная перестановка *IP*

Позиции входных битов, j	1	2	3	4	5	6	7	8
Позиции выходных битов, k	5	1	6	2	7	3	8	4

Таблица 2.9. Начальная перестановка (другая форма записи табл. 2.8)

Позиции выходных битов, k	1	2	3	4	5	6	7	8
Позиции входных битов, j	2	4	6	8	1	3	5	7

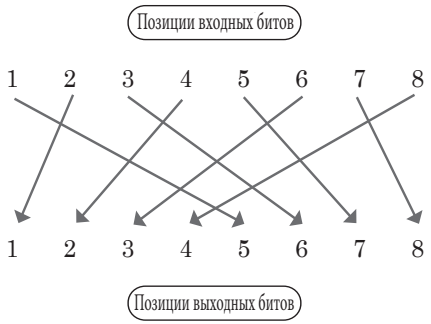


Рис. 2.13. Начальная перестановка *IP*

Кроме того, табл. 2.8 можно выразить в виде табл. 2.9, переставив колонки в порядке следования выходных битов. В табл. 2.9 1-му выходному биту соответствует 2-й входной бит, 2-му выходному биту – 4-й входной бит и т.д. (рис. 2.14).

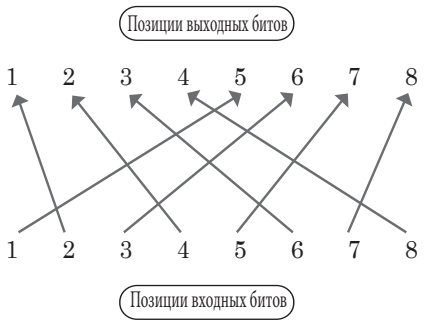


Рис. 2.14. Другая форма записи начальной перестановки *IP*

Полученный после начальной перестановки ряд битов (двоичных данных) после двух раундов генерирования шифртекста подвергается конечной перестановке IP^{-1} (табл. 2.10), возвращающей биты в те же самые позиции, которые они имели до начальной перестановки.

Таблица 2.10. Конечная перестановка IP^{-1}

Позиции входных битов, k	1	2	3	4	5	6	7	8
Позиции выходных битов, j	2	4	6	8	1	3	5	7

Другими словами, совместив табл. 2.8 и 2.10, можно заметить, что, например, 5-й входной бит в табл. 2.8 оказывается на выходе 7-м битом, а затем, в табл. 2.10, 7-й бит опять возвращается в первоначальную позицию 5-го бита (рис. 2.15).

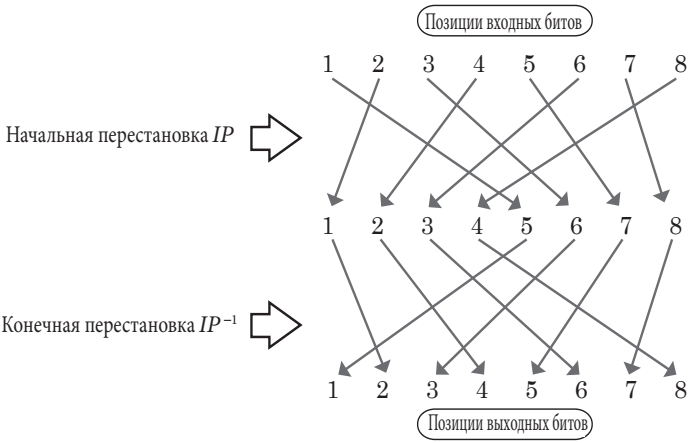


Рис. 2.15. Совместное действие перестановок: начальной *IP* и конечной IP^{-1}

В примере, описывающем процесс шифрования упрощённого DES, мы будем использовать следующие два ключа:

$$K_1 = (110001), K_2 = (111000) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

(о генерировании ключей будет рассказано позже). Далее мы попробуем превратить строку символов MC в строку шифртекста упрощённого DES, считая, что символы выражаются 4-битными двоичными числами. Согдасно табл. 2.7, строка символов MC будет соответствовать двоичному числу 11000010.

Ниже будет описан конкретный упрощённый пример процесса генерирования шифртекста DES. В целях углубления понимания рекомендуем читателям тщательно проверять каждую операцию.

Шаг 1

Используя табл. 2.8, выполняем начальную перестановку *IP* открытого текста 11000010 = «MC» (рис. 2.16).

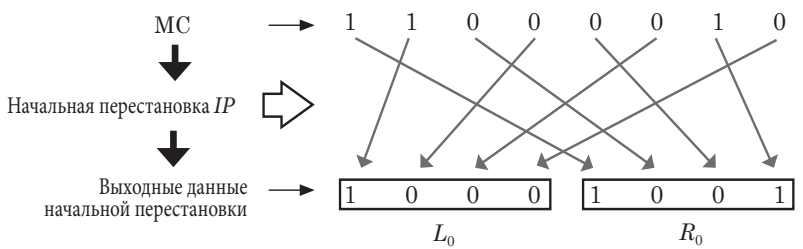


Рис. 2.16. Начальная перестановка (*IP*) открытого текста

Шаг 2

Делим выходные данные начальной перестановки, полученные на шаге 1, на старший (левый) 4-битный блок L_0 и правый (младший) 4-битный блок R_0, как показано на рис. 2.16.

$$L_0 = (1000) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

$$R_0 = (10\underline{01}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Шаг 3

Используя табл. 2.11, проводим перестановку с расширением E (Expansion Permutation) блока R_0, продублировав 3-й и 4-й биты, подчёркнутые в выражении (3) (в результате этой операции битность числа увеличится с 4 до 6, кроме того, изменятся номера позиций битов).

$$ER_0 = (\underline{01}1\underline{00}\underline{1}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

Таблица 2.11. Перестановка с расширением *E*

Позиции выходных битов, k	1	2	3	4	5	6
Позиции входных битов, j	3	4	1	2	3	4

Шаг 4

Складываем по модулю 2 результат перестановки с расширением ER_0 (4) и ключ $K_1 = 110001$ (рис. 2.17).

$$ER_0\,(K_1) = ER_0 \oplus K_1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5)$$

$$= (011001) \oplus (110001)$$

$$= (101000) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

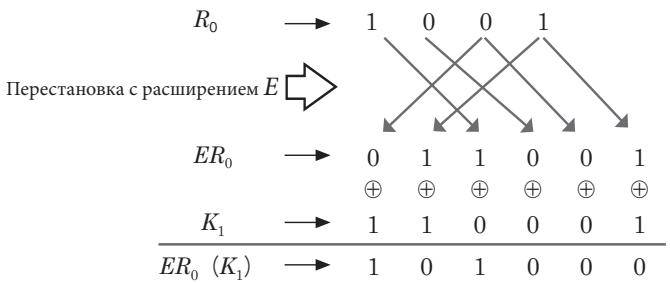


Рис. 2.17. Порядок вычислений шагов 3 и 4

Шаг 5

Используя табл. 2.12, выполняем над выражением (6) замену со сжатием *S* (Substitution) (в результате этой операции битность уменьшится с 6 до 4).

Таблица 2.12. Замена со сжатием *S*

Номера столбцов \ Номера строк	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	④	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Таблица 2.12 состоит из четырёх строк с номерами 0, 1, 2, 3, каждую из которых можно рассматривать как выходной алфавит шифра моноалфавитной замены. Выбираем строку по значению 2-битного двоичного числа (диапазон принимаемых значений $(0)_{10}$–$(3)_{10)}$, составленного из 1-го (крайнего слева) и 6-го (крайнего справа) битов выражения (6): $(1\cancel{0}\cancel{1}\cancel{0}0)_2 = (10)_2 = (2)_{10}$. Выбираем столбец по значению 4-битного двоичного числа (диапазон принимаемых значений: $(0)_{10}$–$(15)_{10)}$, составленного из остальных битов выражения (6): $(\cancel{1}0100\cancel{0})_2 = (0100)_2 = (4)_{10}$.

Выбираем в табл. 2.12 ячейку на пересечении строки № 2 и столбца № 4 и преобразуем содержащееся в ней десятичное число к двоичному виду: $(13)_{10} = (1101)_2$. Полученное двоичное число подвергаем выходной перестановке *PS* по табл. 2.13: $(1101)_2 \rightarrow (0111)_2$ (рис. 2.18). Использованные здесь обозначения $(\ )_2$ и $(\ )_{10}$ указывают, является ли число в скобках двоичным или десятичным соответственно.

Таблица 2.13. Выходная перестановка *PS* после замены со сжатием

Позиции входных битов, *j*	1	2	3	4
Позиции выходных битов, *k*	3	4	1	2

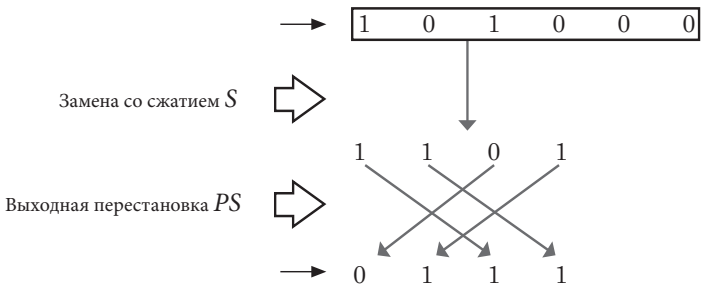


Рис. 2.5. Порядок вычислений шага 5

Рассматривая преобразования шага 4: сложение по модулю 2 с ключом K_1, замену со сжатием S и выходную перестановку PS как единую нелинейную функцию f, мы можем записать:

$$f(R_0, K_1) = (0111) \quad \cdots\cdots \quad (7)$$

Функция f нелинейна, так как она не удовлетворяет условию $f(ax + by) = af(x) + bf(y)$. Например, проходящая через начало координат функция 1-го порядка $f(x) = 2x$ удовлетворяет этому условию и поэтому является линейной, а квадратичная функция $f(x) = x^2$ этому условию не удовлетворяет и поэтому является нелинейной.

Шаг 6

В соответствии с рис. 2.12, используя выражения (2), (3) и (7), находим выходное значение 1-го раунда, состоящее из двух 4-битных блоков: старшего (левого) L_1 и младшего (правого) R_1.

$$L_1 = R_0 = (1001) \quad \cdots\cdots \quad (8)$$

$$R_1 = L_0 \oplus f(R_0, K_1) \quad \cdots\cdots \quad (9)$$

$$= (1000) \oplus (0111) = (1111) \quad \cdots\cdots \quad (10)$$

1-й раунд генерирования шифртекста DES (шаги 3–6) закончен, и мы переходим ко 2-му раунду (шаги 7–10), в котором будут выполняться те же самые вычисления.

Шаг 7

Используя табл. 2.11, проводим перестановку с расширением E (Expansion Permutation) блока R_1 (10):

$$ER_1 = (111111) \quad \cdots\cdots \quad (11)$$

Шаг 8

Складываем по модулю 2 результат перестановки с расширением ER_1 (11) и ключ $K_2 = 111000$:

$$ER_1\ (K_2) = ER_1 \oplus K_2 \quad \cdots\cdots \quad (12)$$

$$= (111111) \oplus (111000)$$

$$= (000111) \quad \cdots\cdots \quad (13)$$

Шаг 9

Чтобы выполнить над выражением (13) перестановку со сжатием S, выбираем в табл. 2.12 ячейку на пересечении строки с номером (0~~0011~~1)$_2$ = (01)$_2$ = (1)$_{10}$ и столбца с номером (~~0~~0011~~1~~)$_2$ = (0011)$_2$ = (3)$_{10}$.

Преобразовав десятичное значение этой ячейки в двоичное число: $(4)_{10} = (0100)_2$, подвергаем его выходной перестановке *PS* по табл. 2.13:

$$(0100) \rightarrow (0001) \quad \cdots\cdots (14)$$

Все преобразования шага 9 можно записать в виде:

$$f(R_1, K_2) = (0001) \quad \cdots\cdots (15)$$

Шаг 10

В соответствии с рис. 2.12, используя выражения (8), (10) и (15), находим выходное значение 2-го раунда, состоящее из двух 4-битных блоков: старшего (левого) L_2 и младшего (правого) R_2.

$$L_2 = R_1 = (1111) \quad \cdots\cdots (16)$$

$$R_2 = L_1 \oplus f(R_1, K_2) \quad \cdots\cdots (17)$$

$$= (1001) \oplus (0001) = (1000) \quad \cdots\cdots (18)$$

Шаг 11

В соответствии с рис. 2.12 меняем местами старший (L_2) и младший (R_2) 4-битные блоки (рис. 2.19).

$$L_2' = R_2 = (1000) \cdots\cdots (19)$$

$$R_2' = L_2 = (1111) \cdots\cdots (20)$$

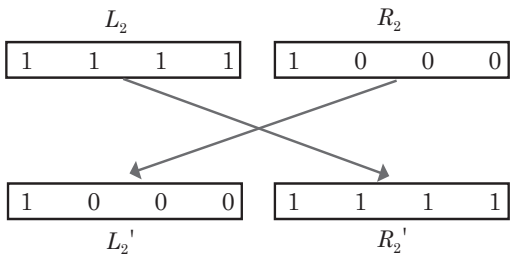


Рис. 2.19. Вычисления шага 11

Шаг 12

В соответствии с табл. 2.10 выполняем конечную перестановку IP^{-1} двоичных данных рис. 2.19. 8-битное число, составленное из старшего (L_2'') и младшего (R_2'') 4-битных блоков, является искомым шифртекстом DES (рис. 2.20).

$$L_2'' = (1110) \quad \ldots\ldots (21)$$

$$R_2'' = (1010) \quad \ldots\ldots (22)$$

Шифртекст DES (11101010) $\ldots\ldots$ (23)

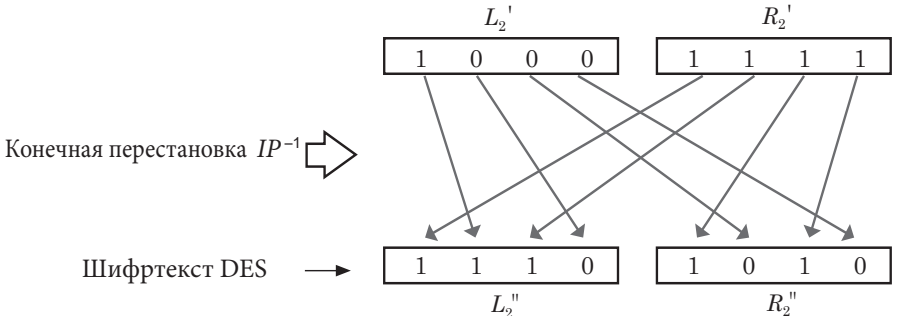


Рис. 2.20. Шифртекст DES, полученный после конечной перестановки IP^{-1}

✤ Расшифрование шифртекста DES

Теперь попробуем вернуться от шифртекста DES (рис. 2.20) к открытому тексту.

Порядок расшифрования будет совершенно таким же, как порядок генерирования шифртекста (рис. 2.12), однако ключи будут использоваться в порядке, обратном порядку использования ключей при шифровании. Другими словами, мы будем использовать ключ K_2 в 1-м раунде и ключ K_1 во 2-м раунде расшифрования.

Шаг 1

Используя табл. 2.8, выполняем начальную перестановку *IP* шифртекста 11101010 (рис. 2.21).

Шаг 2

Делим выходные данные начальной перестановки, полученные на шаге 1, на старший (левый) 4-битный блок L_0 и правый (младший) 4-битный блок R_0. Сравнивая рис. 2.21 с выражениями (19) и (20), мы можем записать:

$$L_0 = (1000) \ (= L_2') \quad \ldots\ldots (24)$$

$$R_0 = (1111) \ (= R_2') \quad \ldots\ldots (25)$$

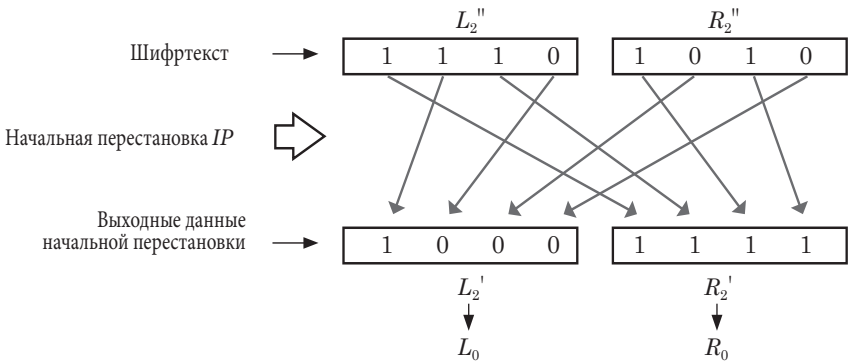


Рис. 2.21. Начальная перестановка (*IP*) шифртекста

Шаг 3

Используя табл. 2.11, проводим перестановку с расширением E блока R_0, продублировав 3-й и 4-й биты, подчёркнутые в выражении (25).

$$ER_0 = (\underline{1}11\underline{1}1\underline{1}) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (26)$$

Шаг 4

Складываем по модулю 2 результат перестановки с расширением ER_0 (26) и ключ $K_2 = 111000$.

$$ER_0\,(K_2) = ER_0 \oplus K_2 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (27)$$

$$= (111111) \oplus (111000)$$

$$= (000111) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (28)$$

Шаг 5

Чтобы выполнить над выражением (28) замену со сжатием S, выбираем в табл. 2.12 ячейку на пересечении строки с номером $(0\cancel{0}\cancel{0}\cancel{1}\cancel{1}1)_2 = (01)_2 = (1)_{10}$ и столбца с номером $(\cancel{0}0011\cancel{1})_2 = (0011)_2 = (3)_{10}$.

Преобразовав десятичное значение этой ячейки в двоичное число: $(4)_{10} = (0100)_2$, подвергаем его выходной перестановке PS по табл. 2.13: $(0100)_2 \rightarrow (0001)_2$.

Все преобразования шага 5 можно записать в виде:

$$f\,(R_0, K_2) = (0001) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (29)$$

Шаг 6

В соответствии с рис. 2.12, используя выражения (24), (25) и (29), находим выходное значение 1-го раунда, состоящее из двух 4-битных блоков: старшего (левого) L_1 и младшего (правого) R_1.

$$L_1 = R_0 = (1111) \quad \cdots\cdots (30)$$

$$R_1 = L_0 \oplus f(R_0, K_2) \quad \cdots\cdots (31)$$

$$= (1000) \oplus (0001) = (10\underline{01}) \quad \cdots\cdots (32)$$

1-й раунд расшифрования DES (шаги 3–6) закончен, и мы переходим ко 2-му раунду расшифрования(шаги 7–10), в котором будут выполняться те же самые вычисления.

Шаг 7

Используя табл. 2.11, проводим перестановку с расширением E (Expansion Permutation) блока R_1 (32):

$$ER_1 = (\underline{01}10\underline{01}) \quad \cdots\cdots (33)$$

Шаг 8

Складываем по модулю 2 результат перестановки с расширением ER_1 (33) и ключ $K_1 = 110001$:

$$ER_1(K_1) = ER_1 \oplus K_1 \quad \cdots\cdots (34)$$

$$= (011001) \oplus (110001)$$

$$= (101000) \quad \cdots\cdots (35)$$

Шаг 9

Чтобы выполнить над выражением (35) замену со сжатием S, выбираем в табл. 2.12 ячейку на пересечении строки с номером $(1\cancel{0}10\cancel{00})_2 = (10)_2 = (2)_{10}$ и столбца с номером $(\cancel{1}0100\cancel{0})_2 = (0100)_2 = (4)_{10}$.

Преобразовав десятичное значение этой ячейки в двоичное число: $(13)_{10} = (1101)_2$, подвергаем его выходной перестановке PS по табл. 2.13:

$$(1101) \rightarrow (0111) \quad \cdots\cdots (36)$$

Все преобразования шага 9 можно записать в виде:

$$f(R_1, K_1) = (0111) \quad \cdots\cdots (37)$$

Шаг 10

В соответствии с рис. 2.12, используя выражения (30), (31) и (37), находим выходное значение 2-го раунда, состоящее из двух 4-битных блоков: старшего (левого) L_2 и младшего (правого) R_2.

$$L_2 = R_1 = (1001) \quad \cdots\cdots (38)$$

$$R_2 = L_1 \oplus f(R_1,\ K_1) \quad \cdots\cdots (39)$$

$$= (1111) \oplus (0111) = (1000) \quad \cdots\cdots (40)$$

Шаг 11

В соответствии с рис 2.12 меняем местами старший (L_2) и младший (R_2) 4-битные блоки (рис. 2.19).

$$L_2' = R_2 = (1000) \cdots\cdots (41)$$

$$R_2' = L_2 = (1001) \cdots\cdots (42)$$

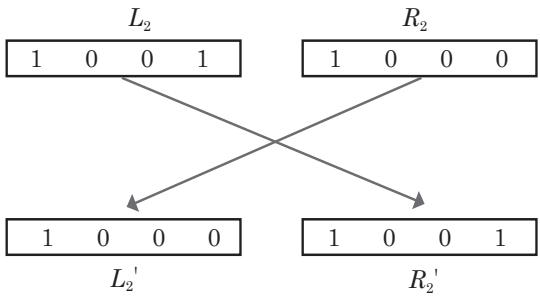


Рис. 2.22. Вычисления шага 11

Шаг 12

В соответствии с табл. 2.10, выполняем конечную перестановку IP^{-1} двоичных данных рис. 2.22 (рис. 2.23).

$$L_2'' = (1100) \cdots\cdots (43)$$

$$R_2'' = (0010) \cdots\cdots (44)$$

Открытый текст 1100 0010

「M」 「C」

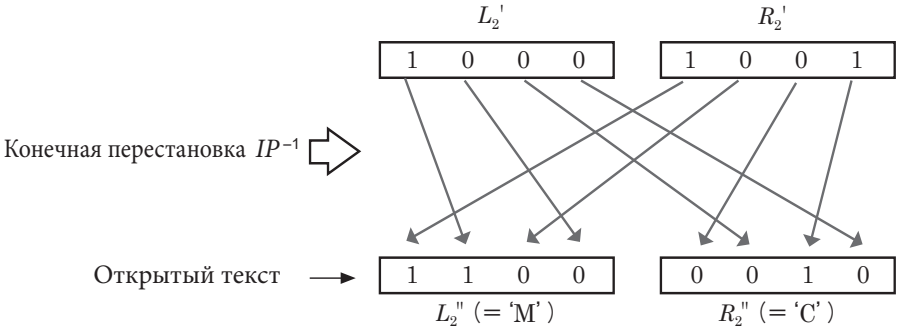


Рис. 2.23. Вычисления шага 12

Полученное нами 8-битное двоичное число является открытым текстом, а двоичные коды выражений (43) и (44) соответствуют символам «M», «C» по табл. 2.7. Таким образом, расшифрование шифртекста DES было проведено правильно.

На основании вышеизложенного, сопоставив этапы шифрования и расшифрования DES, мы можем убедиться, что расшифрование происходит благодаря выполнению в обратном порядке того же самого процесса, который использовался для шифрования (рис. 2.24).

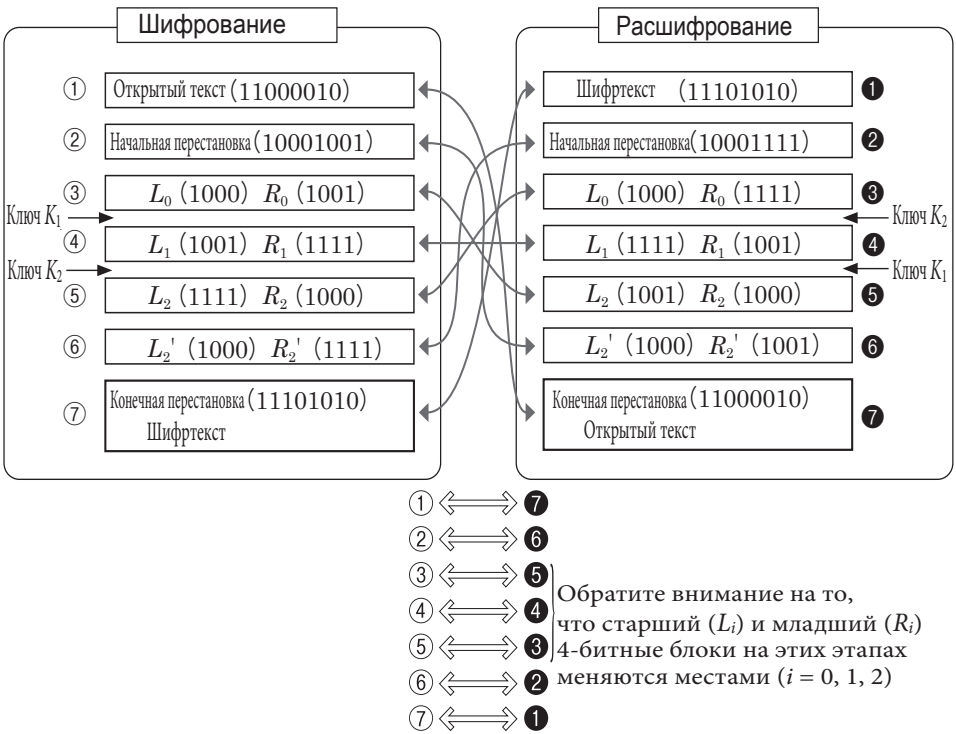


Рис. 2.24. Взаимосвязь шифрования и расшифрования

❀ Генерирование ключей шифрования DES

Теперь опишем порядок генерирования ключей шифрования и расшифрования из общего ключа DES. Сначала, положив 8-битный общий ключ (начальный ключ) K_0 равным, например, следующему значению:

$$K_0 = (10011001) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (45)$$

проследим порядок генерирования ключей шифрования для 1-го раунда (K_1) и для 2-го раунда (K_2) (рис. 2.25).

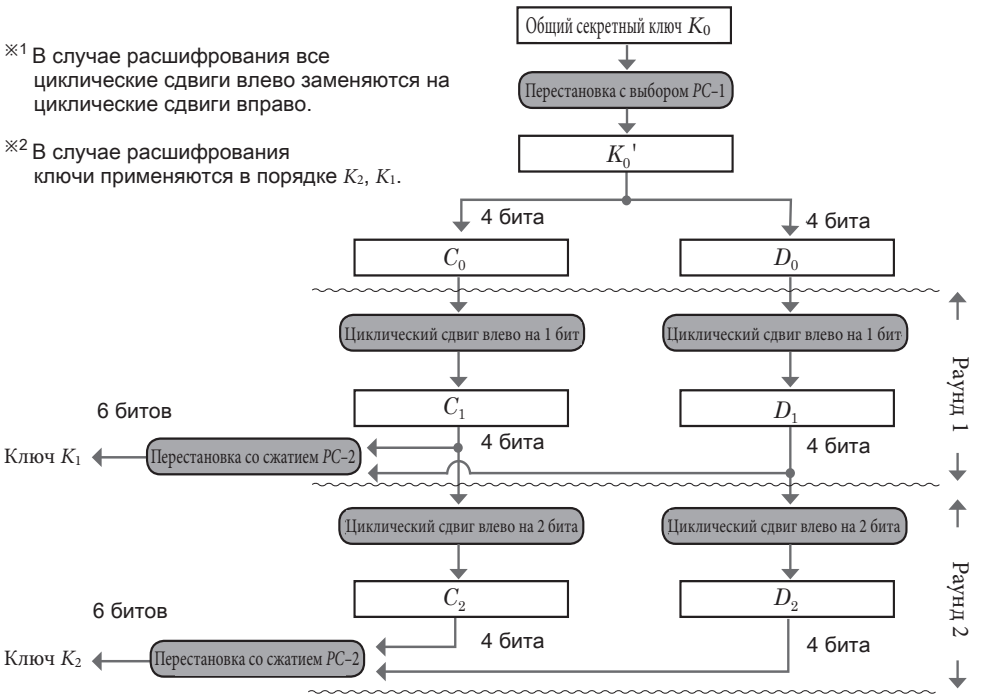


Рис. 2.25. Порядок генерирования ключей шифрования и расшифрования

Шаг 1

В соответствии с табл. 2.14 перемешиваем биты общего (секретного) ключа K_0 (45) с помощью операции перестановки с выбором *PC*-1 (рис. 2.26).

$$K_0' = (00110101) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (46)$$

Разделяем ключ K_0' (46) на старший (C_0) и младший (D_0) 4-битные блоки.

$$C_0 = (0011) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (47)$$

$$D_0 = (0101) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (48)$$

Табл. 2.14. Перестановка с выбором *PC*-1

	Старший 4-битный блок C_i				Младший 4-битный блок D_i			
Позиции входных битов, j	1	2	3	4	5	6	7	8
Позиции выходных битов, k	8	7	1	3	6	2	5	4

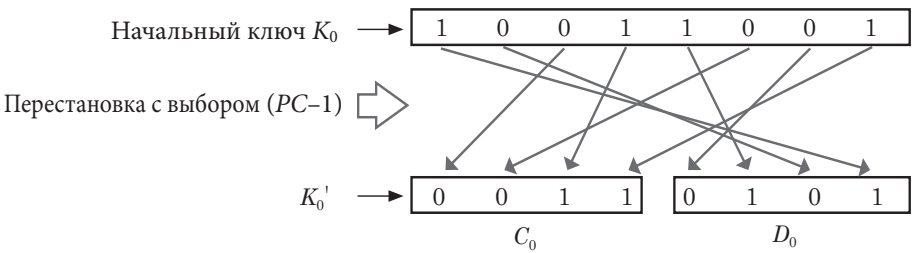


Рис. 2.26. Перастановка с выбором *PC*-1

Шаг 2

В соответствии с табл. 2.15 величина циклического сдвига в 1-м раунде равна 1 биту, поэтому мы циклически сдвигаем все биты каждого из блоков C_0 и D_0 по отдельности на одну позицию влево и обозначаем полученные результаты C_1 и D_1 (рис. 2.27).

$$C_1 = (0110) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (49)$$

$$D_1 = (1010) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (50)$$

Таблица 2.15. Величины циклического сдвига влево

Номера раундов	1	2
Величины циклического сдвига	1	2

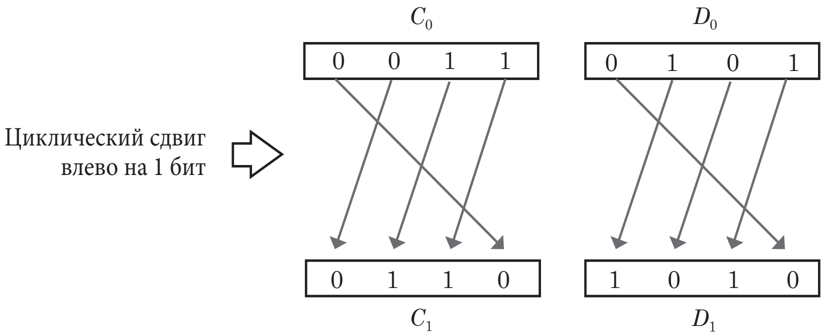


Рис. 2.27. Операции циклического сдвига влево (шаг 2)

Шаг 3

В соответствии с табл. 2.16 объединяем блоки C_1 (49) и D_1 (50) с помощью перестановки со сжатием *PC*-2 (битность уменьшается с 8 до 6) и получаем ключ K_1, который будет использоваться в первом раунде шифрования (рис. 2.28).

$$K_1 = (110001) \quad \cdots\cdots \quad (51)$$

Таблица 2.16. Перестановка со сжатием *PC*-2

Позиции выходных битов, k	1	2	3	4	5	6
Позиции входных битов, j	7	5	1	8	6	2

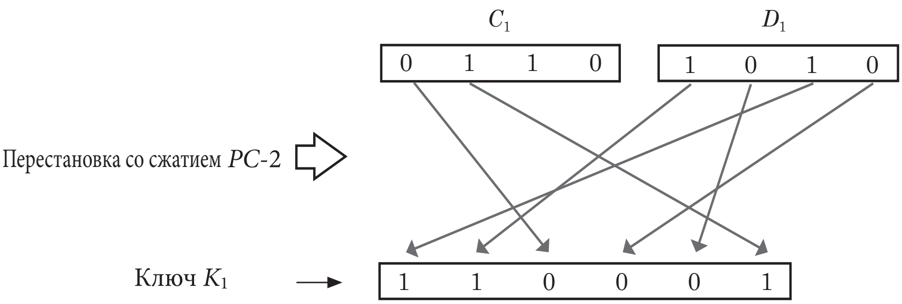


Рис. 2.28. Перестановка со сжатием *PC*-2 (шаг 3)

Раз за разом повторяя вычисления шагов 2–3, мы сможем один за другим получать ключи, которые будут использоваться для шифрования.

Шаг 4

В соответствии с табл. 2.15 величина циклического сдвига влево во 2-м раунде равна 2 битам, поэтому мы циклически сдвигаем все биты каждого из блоков C_1 и D_1 по отдельности влево на 2 бита и получаем C_2 и D_2 (рис. 2.29).

$$C_2 = (1001) \quad (52)$$

$$D_2 = (1010) \quad (53)$$

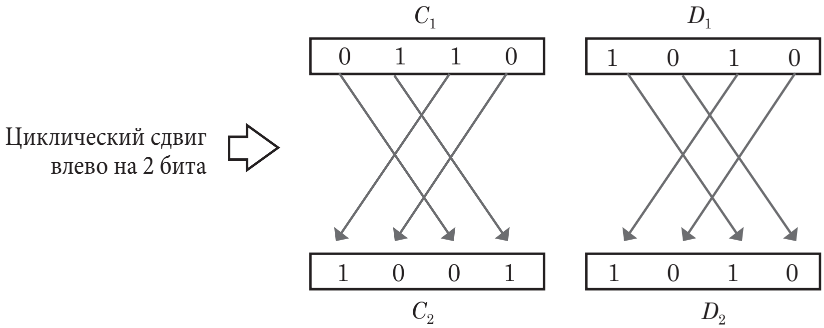


Рис. 2.29. Операции циклического сдвига влево (шаг 4)

Шаг 5

В соответствии с табл. 2.16 объединяем блоки C_2 (52) и D_2 (53) с помощью перестановки со сжатием *PC*-2 (битность уменьшается с 8 до 6) и получаем ключ K_2, который будет использоваться во втором раунде шифрования (рис. 2.30).

$$K_2 = (111000) \quad (54)$$

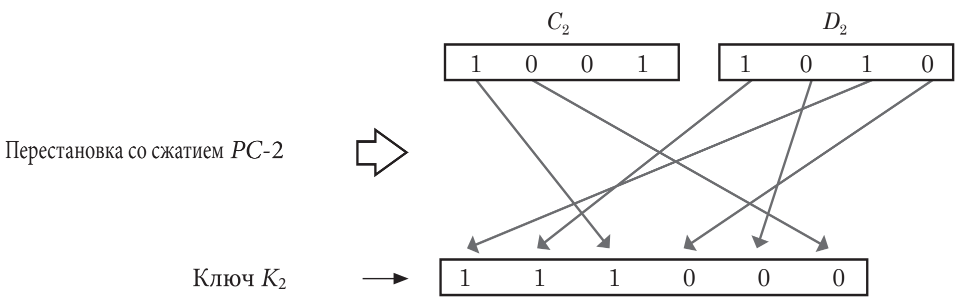


Рис. 2.30. Перестановка со сжатием *PC*-2 (шаг 5)

❖ Генерирование ключей расшифрования DES

Ключи шифрования мы генерировали в порядке K_1, K_2 из общего ключа (начального ключа) K_0 = 10011001 (45). Ключи расшифрования необходимо генерировать из общего ключа K_0 в обратном порядке – K_2, K_1. При использовании алгоритма, показанного на рис. 2.25, это может быть достигнуто благодаря замене операций циклического сдвига влево, выполнявшихся в процессе генерирования ключей шифрования, на операции циклического сдвига вправо. Ниже мы проследим порядок генерирования ключей расшифрования на основе рис. 2.25.

Шаг 1

В соответствии с табл. 2.14 перемешиваем биты общего (секретного) ключа K_0 (45) с помощью операции перестановки с выбором *PC*-1.

$$K_0' = (00110101) \quad \cdots\cdots (55)$$

$$C_0 = (0011) \quad \cdots\cdots (56)$$

$$D_0 = (0101) \quad \cdots\cdots (57)$$

Шаг 2

В соответствии с табл. 2.17 величина циклического сдвига вправо в 1-м раунде равна 1 биту, поэтому мы циклически сдвигаем все биты каждого из блоков C_0 и D_0 по отдельности на одну позицию вправо и обозначаем полученные результаты C_1 и D_1 (рис. 2.31).

$$C_1 = (1001) \quad \cdots\cdots (58)$$

$$D_1 = (1010) \quad \cdots\cdots (59)$$

Таблица 2.17. Величины циклического сдвига вправо

Номера раундов	1	2
Величины циклического сдвига	1	2

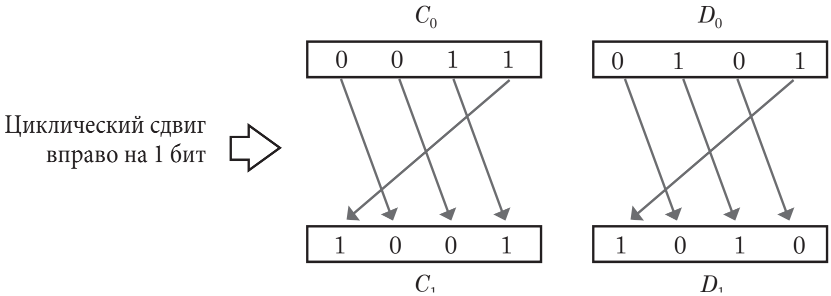


Рис. 2.31. Операции циклического сдвига вправо (шаг 2)

Шаг 3

В соответствии с табл. 2.16 объединяем блоки C_1 (58) и D_1 (59) с помощью перестановки со сжатием *PC*-2 (битность уменьшается с 8 до 6) и получаем ключ K_2, который будет использоваться в первом раунде расшифрования (рис. 2.32).

$$K_2 = (111000) \quad \cdots\cdots \quad (60)$$

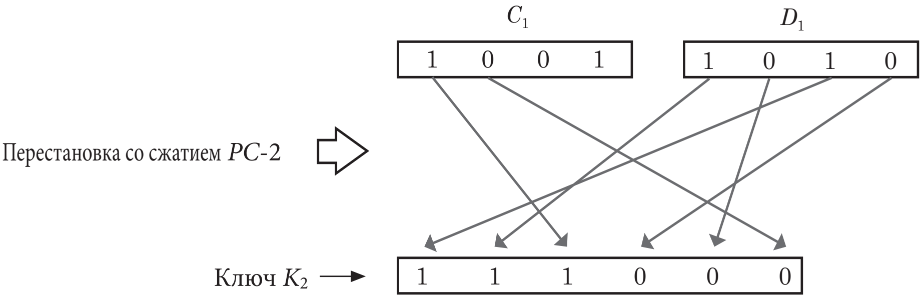


Рис. 2.32. Перестановка со сжатием *PC*-2 (шаг 3)

Шаг 4

В соответствии с табл. 2.17 величина циклического сдвига вправо во 2-м раунде равна 2 битам, поэтому мы циклически сдвигаем все биты каждого из блоков C_1 и D_1 по отдельности на две позиции вправо на 2 бита и получаем C_2 и D_2 (рис.2.33).

$$C_2 = (0110) \quad \cdots\cdots \quad (61)$$

$$D_2 = (1010) \quad \cdots\cdots \quad (62)$$

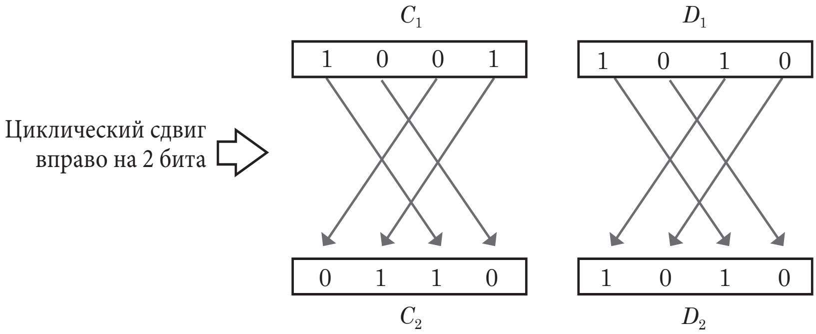


Рис. 2.33. Операции циклического сдвига вправо (шаг 4)

Шаг 5

В соответствии с табл. 2.16 объединяем блоки C_2 (61) и D_2 (62) с помощью перестановки со сжатием *PC*-2 (битность уменьшается с 8 до 6) и получаем ключ K_1, который будет использоваться во втором раунде расшифрования (рис. 2.34).

$$K_1 = (110001) \quad\quad (63)$$

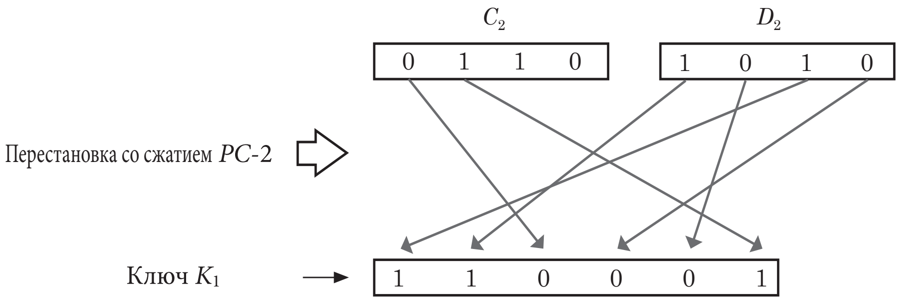
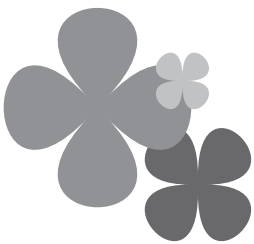


Рис. 2.34. Перестановка со сжатием *PC*-2 (шаг 5)

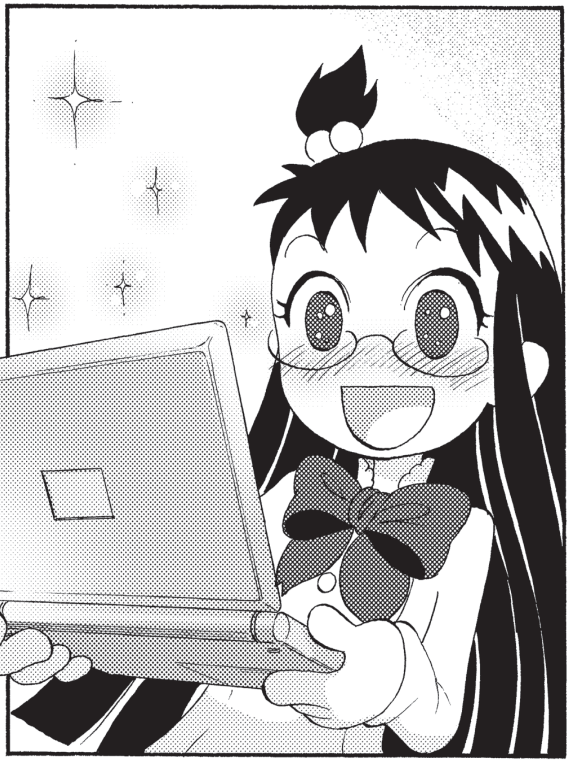
На основании вышеприведённых результатов, сравнивая ключи расшифрования (60), (63) с ключами шифрования (51), (54), можно сделать вывод, что вышеописанным способом можно получать ключи расшифрования в порядке K_2, K_1, обратном порядку получения ключей шифрования (K_1, K_2).

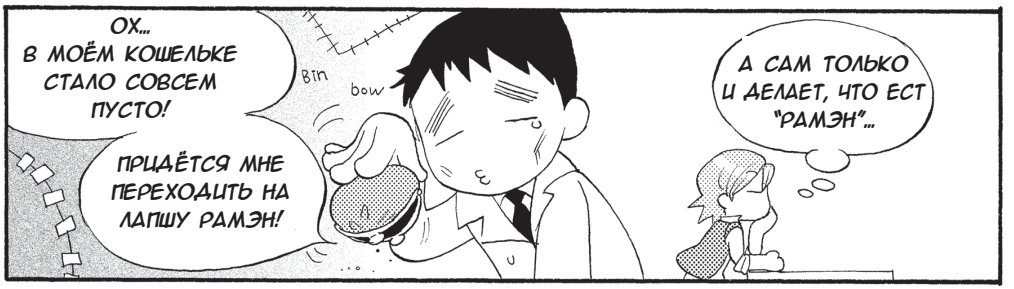
ГЛАВА 3

ШИФР С ОТКРЫТЫМ КЛЮЧОМ



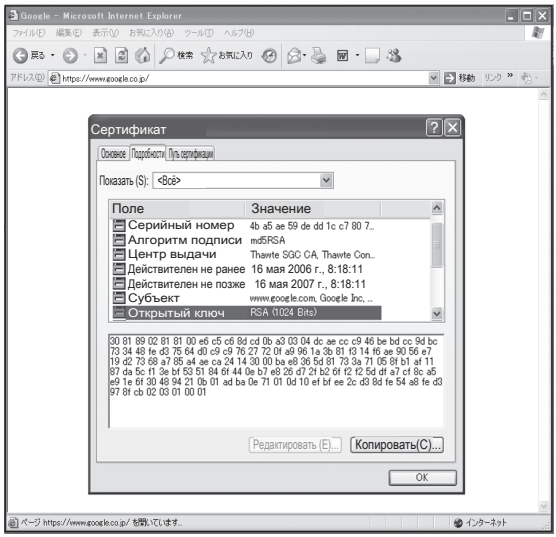
3-1 Основы шифра с открытым ключом






С ПОМОЩЬЮ КОМПЬЮТЕРА МОЖНО МНОГОЕ...
ГЛАДЬ
ГЛАДЬ
ОБЩЕНИЕ ПО Е-МЕЙЛ, ИНТЕРНЕТ-ШОППИНГ, ОНЛАЙН-ИГРЫ...
ЧТО?! ТЫ ЖЕ ГОВОРИЛА, ЧТО ДЛЯ УЧЁБЫ?
ХЕ-ХЕ
Я ТОЖЕ ЧАСТО ДЕЛАЮ ПОКУПКИ В ИНТЕРНЕТЕ!
И ОПЛАЧИВАЮ ИХ БАНКОВСКОЙ КАРТОЙ. ЭТО ТАК УДОБНО!
BOOK
CD
SHOES
НО НЕ ОПАСНО ЛИ ЭТО? ВЕДЬ ИНФОРМАЦИЮ КРЕДИТНОЙ КАРТЫ МОГУТ УКРАСТЬ...
ЧТОБЫ НЕ УКРАЛИ, ИСПОЛЬЗУЮТ ШИФРОВАНИЕ С ОТКРЫТЫМ КЛЮЧОМ!

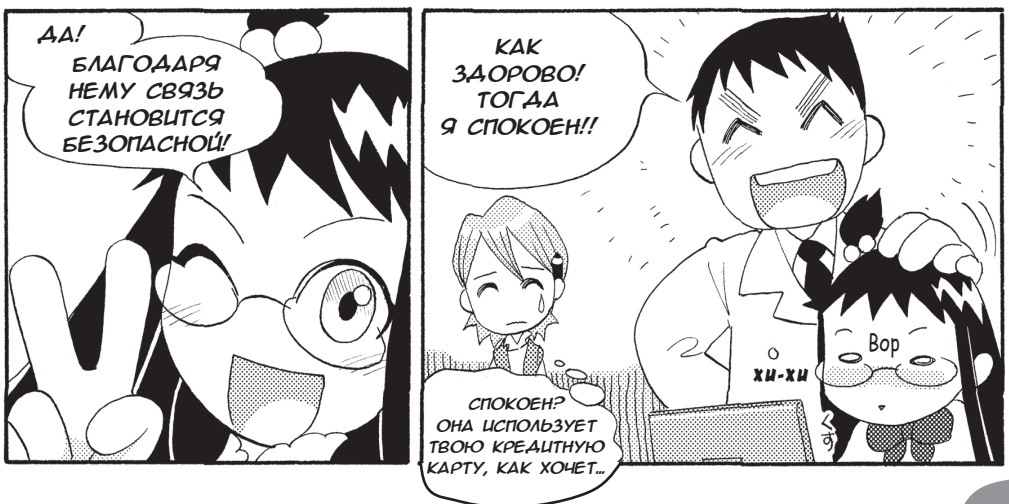

Интернет-магазин
Рай покупок
Сумка класса «люкс»
¥ 300,000
Купить
Описание
СМОТРИ, КАКАЯ КРАСИВАЯ СУМКА!
ДА, НАВЕРНОЕ...
ХОТЯ Я НЕ ОЧЕНЬ ХОРОШО РАЗБИРАЮСЬ...
ДАВАЙ ИСПЫТАЕМ ТВОЮ КРЕДИТНУЮ КАРТУ.
ЭТО ЕЩЁ ЗАЧЕМ?
ТЕПЕРЬ ПОПРОБУЙ ВВЕСТИ СЮДА ИНФОРМАЦИЮ.
КЛАЦ
КЛАЦ
CARD
МОЛОДЕЦ! А ТЕПЕРЬ МЫ КЛИКНЕМ ПО ЭТОЙ КНОПОЧКЕ!
¥ 300,0
Купить
Описание
КЛАЦ
ЧТО ТЫ НАДЕЛАЛА?!
ПОСМОТРИ ВОТ НА ЭТО!

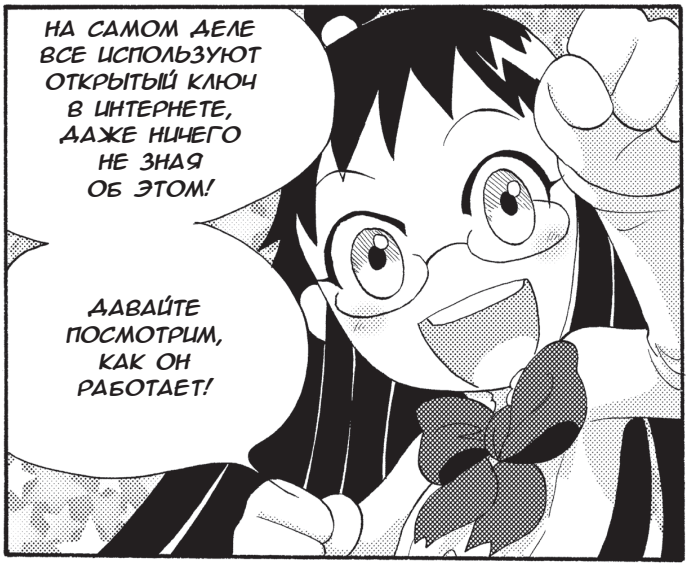


Открытый ключ отображается только в состоянии шифрованного соединения. Если в качестве браузера используется Internet Explorer, то для отображения открытого ключа необходимо пройти по пунктам меню: **Файл** → **Свойства** → **Сертификат** и открыть вкладку **Подробности**.

При использовании таких браузеров, как Safari или Firefox, необходимо кликнуть по изображению замка перед адресной строкой и просмотреть подробную информацию.

※ В современных браузерах всё не так – *Прим. ред.*.

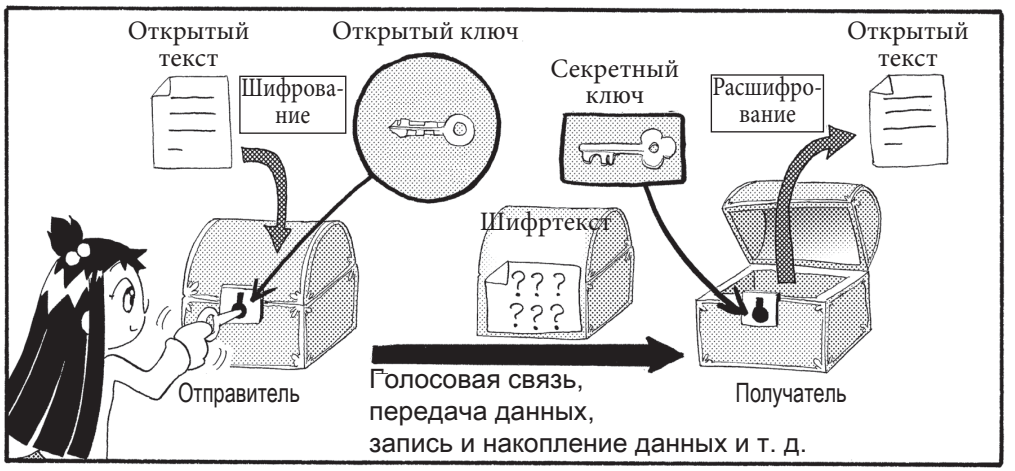


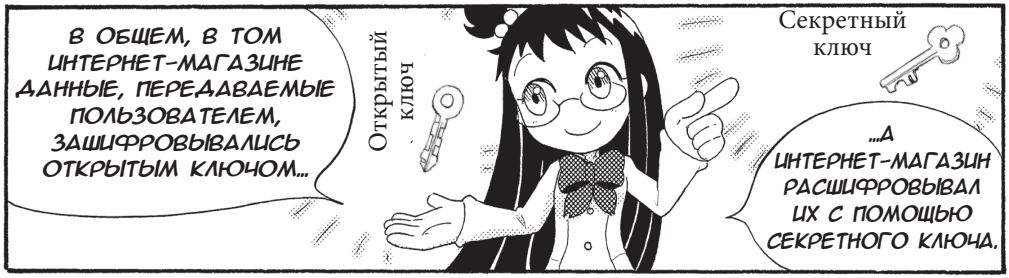

НА САМОМ ДЕЛЕ ВСЕ ИСПОЛЬЗУЮТ ОТКРЫТЫЙ КЛЮЧ В ИНТЕРНЕТЕ, ДАЖЕ НИЧЕГО НЕ ЗНАЯ ОБ ЭТОМ!
ДАВАЙТЕ ПОСМОТРИМ, КАК ОН РАБОТАЕТ!

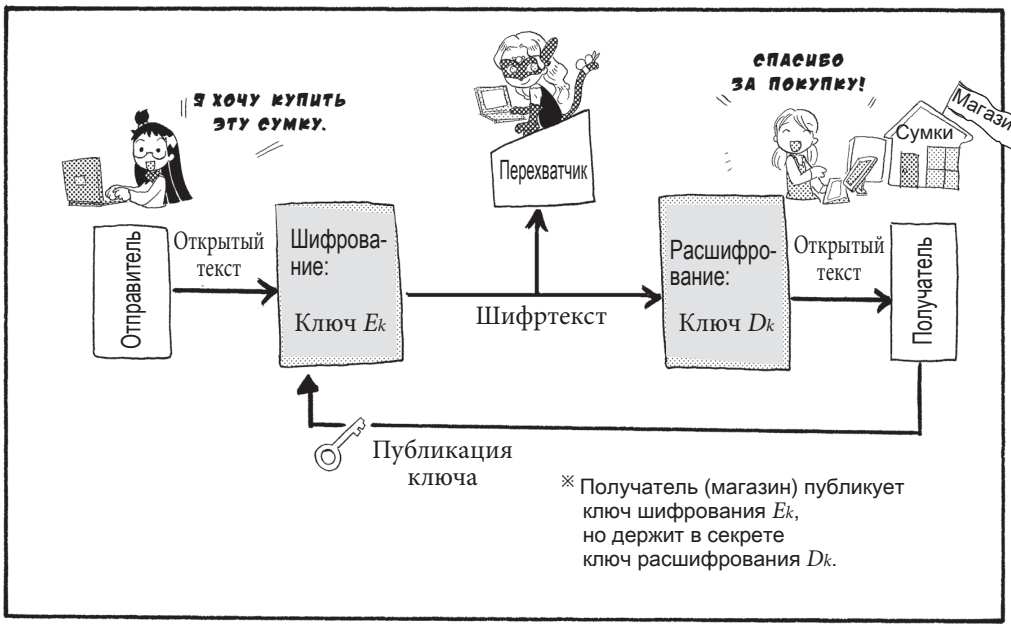

НО ВЕДЬ В ШИФРАХ, КОТОРЫЕ МЫ ИЗУЧАЛИ ДО ЭТОГО, КЛЮЧ НУЖНО БЫЛО ДЕРЖАТЬ В СЕКРЕТЕ...
ДА, ДА!!


ТОГДА ПОЧЕМУ ЗДЕСЬ ЕГО МОЖНО СПОКОЙНО РАСКРЫТЬ?

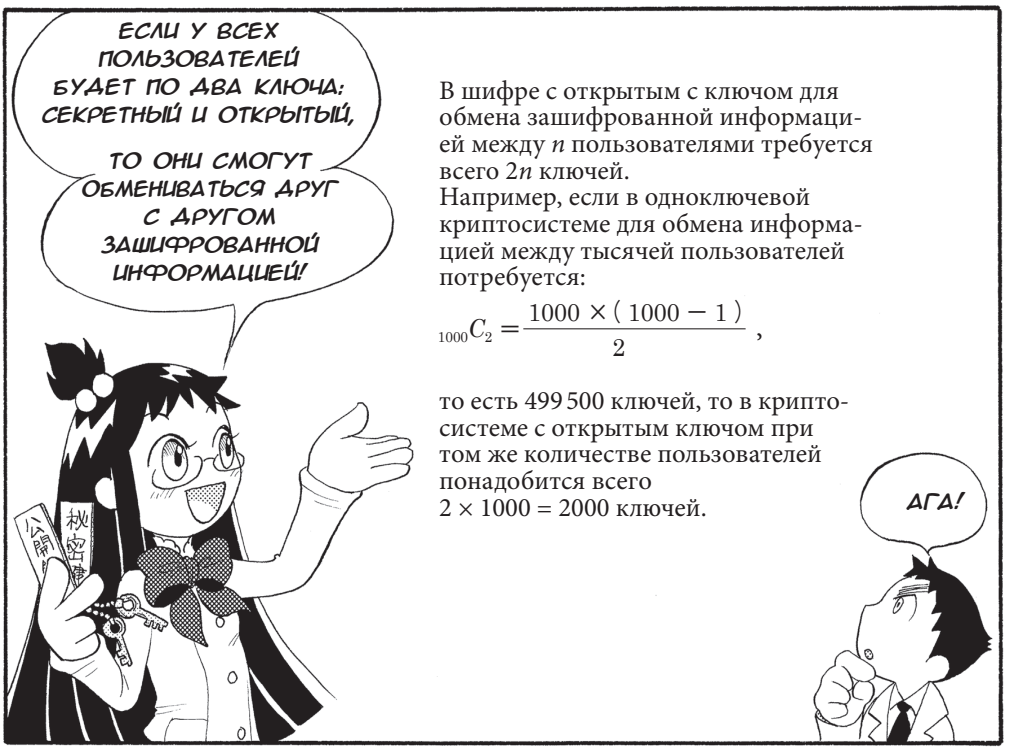

НА САМОМ ДЕЛЕ, КРОМЕ ОТКРЫТОГО КЛЮЧА,
ЗДЕСЬ ИСПОЛЬЗУЕТСЯ И СЕКРЕТНЫЙ КЛЮЧ ТОЖЕ!


Открытый текст
Шифрование
Открытый ключ
Секретный ключ
Расшифрование
Открытый текст
Шифртекст
???
???
Отправитель
Голосовая связь,
передача данных,
запись и накопление данных и т. д.
Получатель


В ОБЩЕМ, В ТОМ ИНТЕРНЕТ-МАГАЗИНЕ ДАННЫЕ, ПЕРЕДАВАЕМЫЕ ПОЛЬЗОВАТЕЛЕМ, ЗАШИФРОВЫВАЛИСЬ ОТКРЫТЫМ КЛЮЧОМ...
Открытый ключ
Секретный ключ
...А ИНТЕРНЕТ-МАГАЗИН РАСШИФРОВЫВАЛ ИХ С ПОМОЩЬЮ СЕКРЕТНОГО КЛЮЧА.


Я ХОЧУ КУПИТЬ ЭТУ СУМКУ.
Перехватчик
СПАСИБО ЗА ПОКУПКУ!
Магазин
Сумки
Отправитель
Открытый текст
Шифрование: Ключ Ek
Шифртекст
Расшифрование: Ключ Dk
Открытый текст
Получатель
Публикация ключа
※ Получатель (магазин) публикует ключ шифрования Ek, но держит в секрете ключ расшифрования Dk.

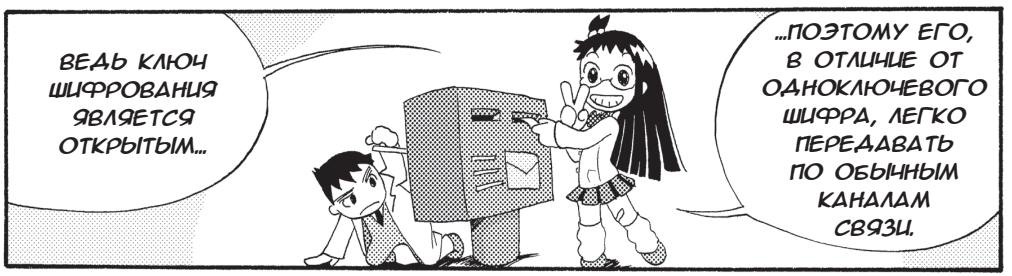

ТО ЕСТЬ ДЛЯ ШИФРОВАНИЯ И РАСШИФРОВАНИЯ ИСПОЛЬЗУЮТСЯ РАЗНЫЕ КЛЮЧИ?
ДА. ПОЭТОМУ ТАКОЙ ШИФР НАЗЫВАЮТ ЕЩЁ АСИММЕТРИЧНЫМ.
Криптосистема с открытым ключом (Public Key Cryptosystem)
||
Асимметричная криптосистема (Asymmetric Key Cryptosystem)

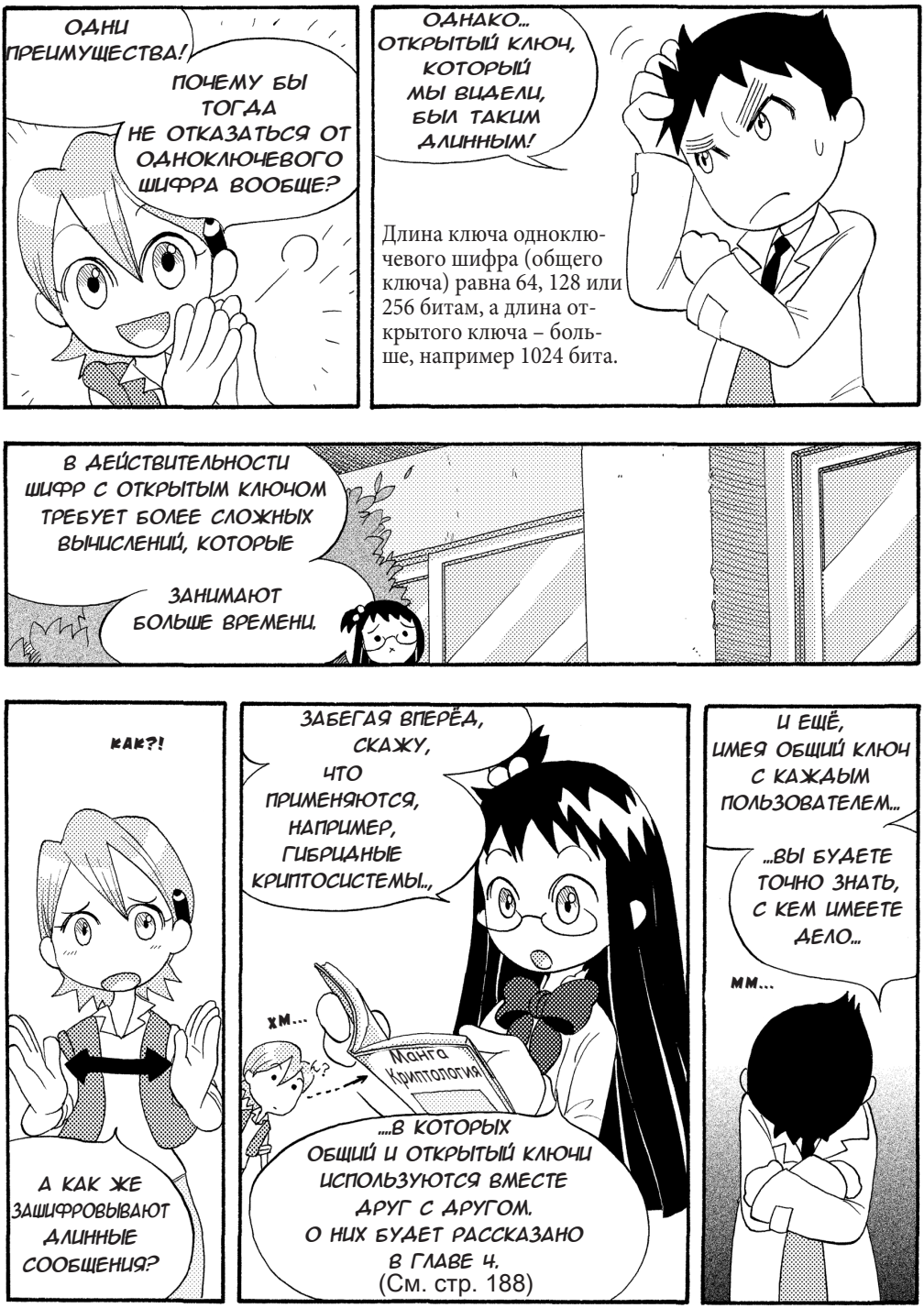


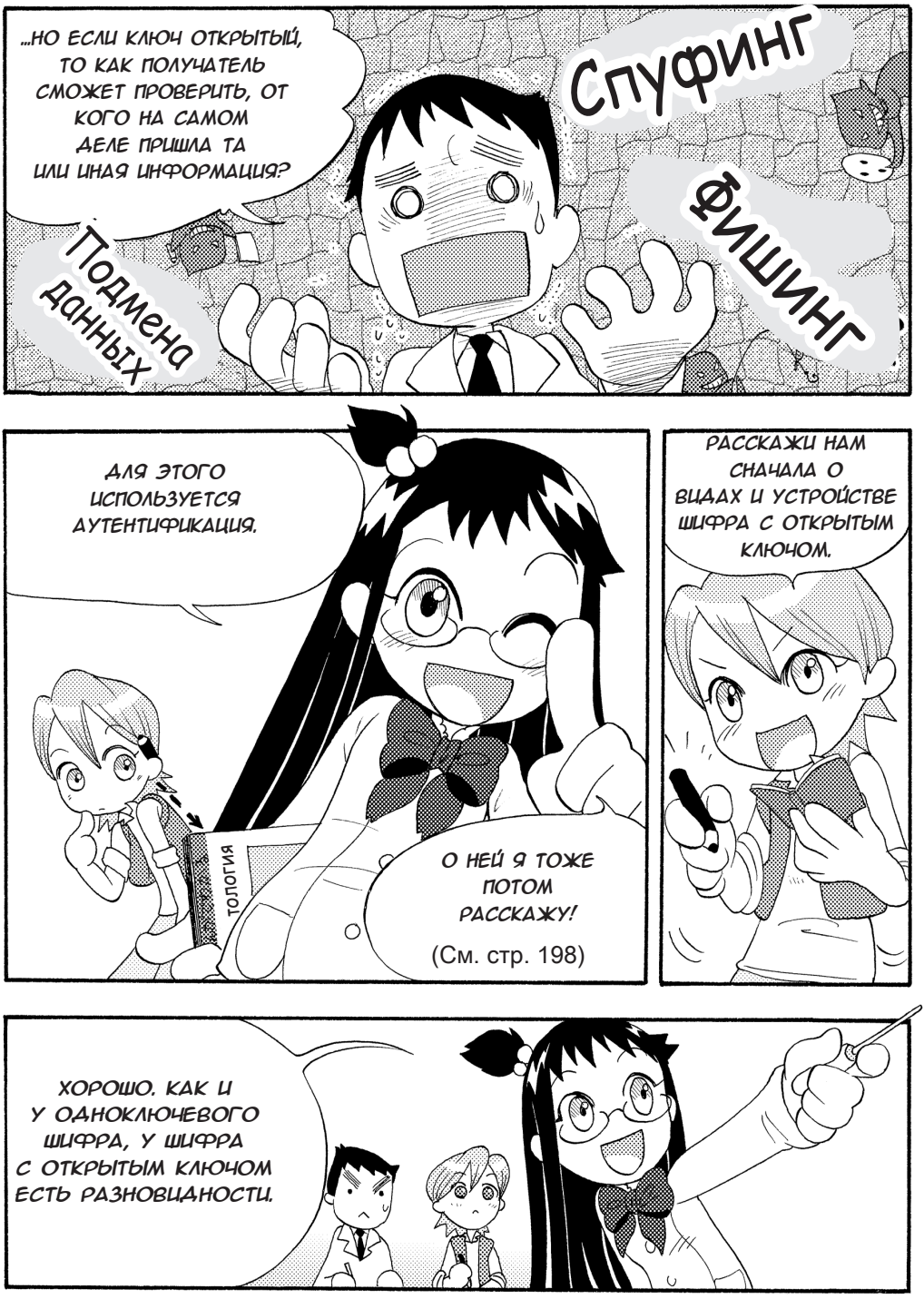
В шифре с открытым с ключом для обмена зашифрованной информацией между n пользователями требуется всего $2n$ ключей.
Например, если в одноключевой криптосистеме для обмена информацией между тысячей пользователей потребуется:

$$_{1000}C_2 = \frac{1000 \times (1000 - 1)}{2},$$

то есть 499 500 ключей, то в криптосистеме с открытым ключом при том же количестве пользователей понадобится всего $2 \times 1000 = 2000$ ключей.




ОДНИ ПРЕИМУЩЕСТВА!
ПОЧЕМУ БЫ ТОГДА НЕ ОТКАЗАТЬСЯ ОТ ОДНОКЛЮЧЕВОГО ШИФРА ВООБЩЕ?
ОДНАКО... ОТКРЫТЫЙ КЛЮЧ, КОТОРЫЙ МЫ ВИДЕЛИ, БЫЛ ТАКИМ ДЛИННЫМ!
Длина ключа одноключевого шифра (общего ключа) равна 64, 128 или 256 битам, а длина открытого ключа – больше, например 1024 бита.
В ДЕЙСТВИТЕЛЬНОСТИ ШИФР С ОТКРЫТЫМ КЛЮЧОМ ТРЕБУЕТ БОЛЕЕ СЛОЖНЫХ ВЫЧИСЛЕНИЙ, КОТОРЫЕ ЗАНИМАЮТ БОЛЬШЕ ВРЕМЕНИ.
КАК?!
А КАК ЖЕ ЗАШИФРОВЫВАЮТ ДЛИННЫЕ СООБЩЕНИЯ?
ЗАБЕГАЯ ВПЕРЁД, СКАЖУ, ЧТО ПРИМЕНЯЮТСЯ, НАПРИМЕР, ГИБРИДНЫЕ КРИПТОСИСТЕМЫ...
ХМ...
Манга Криптология
...В КОТОРЫХ ОБЩИЙ И ОТКРЫТЫЙ КЛЮЧИ ИСПОЛЬЗУЮТСЯ ВМЕСТЕ ДРУГ С ДРУГОМ. О НИХ БУДЕТ РАССКАЗАНО В ГЛАВЕ 4.
(См. стр. 188)
И ЕЩЁ, ИМЕЯ ОБЩИЙ КЛЮЧ С КАЖДЫМ ПОЛЬЗОВАТЕЛЕМ...
...ВЫ БУДЕТЕ ТОЧНО ЗНАТЬ, С КЕМ ИМЕЕТЕ ДЕЛО...
ММ...


...НО ЕСЛИ КЛЮЧ ОТКРЫТЫЙ, ТО КАК ПОЛУЧАТЕЛЬ СМОЖЕТ ПРОВЕРИТЬ, ОТ КОГО НА САМОМ ДЕЛЕ ПРИШЛА ТА ИЛИ ИНАЯ ИНФОРМАЦИЯ?
Спуфинг
Фишинг
Подмена данных
ДЛЯ ЭТОГО ИСПОЛЬЗУЕТСЯ АУТЕНТИФИКАЦИЯ.
О НЕЙ Я ТОЖЕ ПОТОМ РАССКАЖУ!
(См. стр. 198)
РАССКАЖИ НАМ СНАЧАЛА О ВИДАХ И УСТРОЙСТВЕ ШИФРА С ОТКРЫТЫМ КЛЮЧОМ.
ХОРОШО. КАК И У ОДНОКЛЮЧЕВОГО ШИФРА, У ШИФРА С ОТКРЫТЫМ КЛЮЧОМ ЕСТЬ РАЗНОВИДНОСТИ.

✤ Основные разновидности шифра с открытым ключом

В зависимости от используемого фокуса криптографической магии шифры с открытым ключом делятся на две большие группы.

Используемый фокус: задача факторизации целых чисел

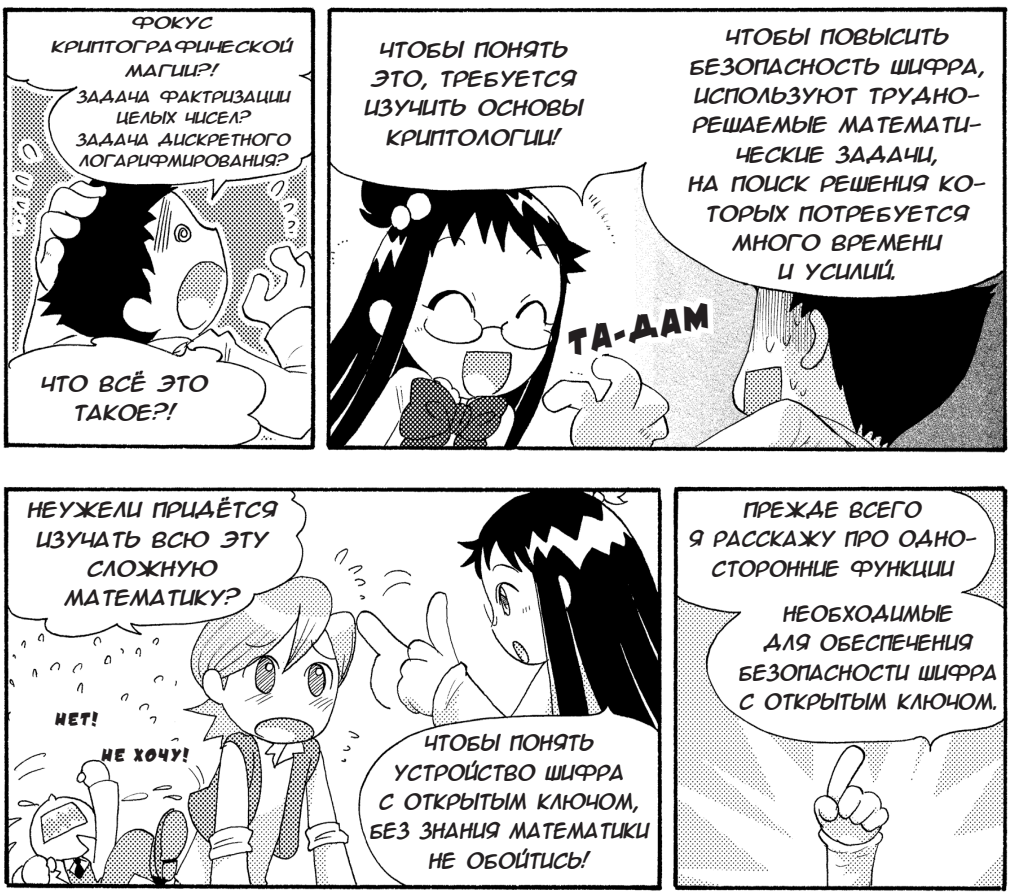
- Шифр RSA
- Шифр Рабина

и другие

Используемый фокус: задача дискретного логарифмирования

- Шифр Эль-Гамаля
- Криптосистемы на эллиптических кривых
- Алгоритм DSA

и другие



✤ Односторонние функции

Функцию, которую легко вычислить для любого входного значения, но вычислить входное значение по заданному значению функции, наоборот, чрезвычайно сложно, называют «односторонней».

Приведём здесь примеры односторонних функций.

(1) Задача факторизации целых чисел

Перемножить два больших простых числа достаточно легко, но чрезвычайно трудно, зная только результат этого произведения (составное число), определить, какие простые числа были перемножены (то есть разложить составное число на простые множители).

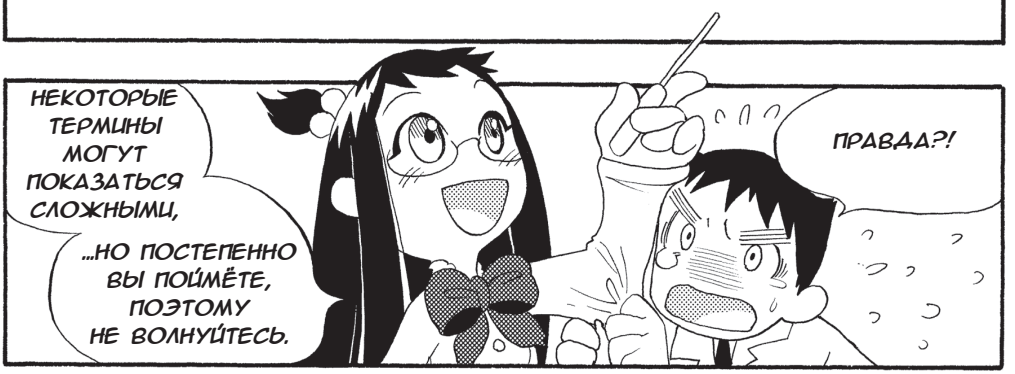
Эту задачу называют задачей факторизации целых чисел (см. стр. 122).

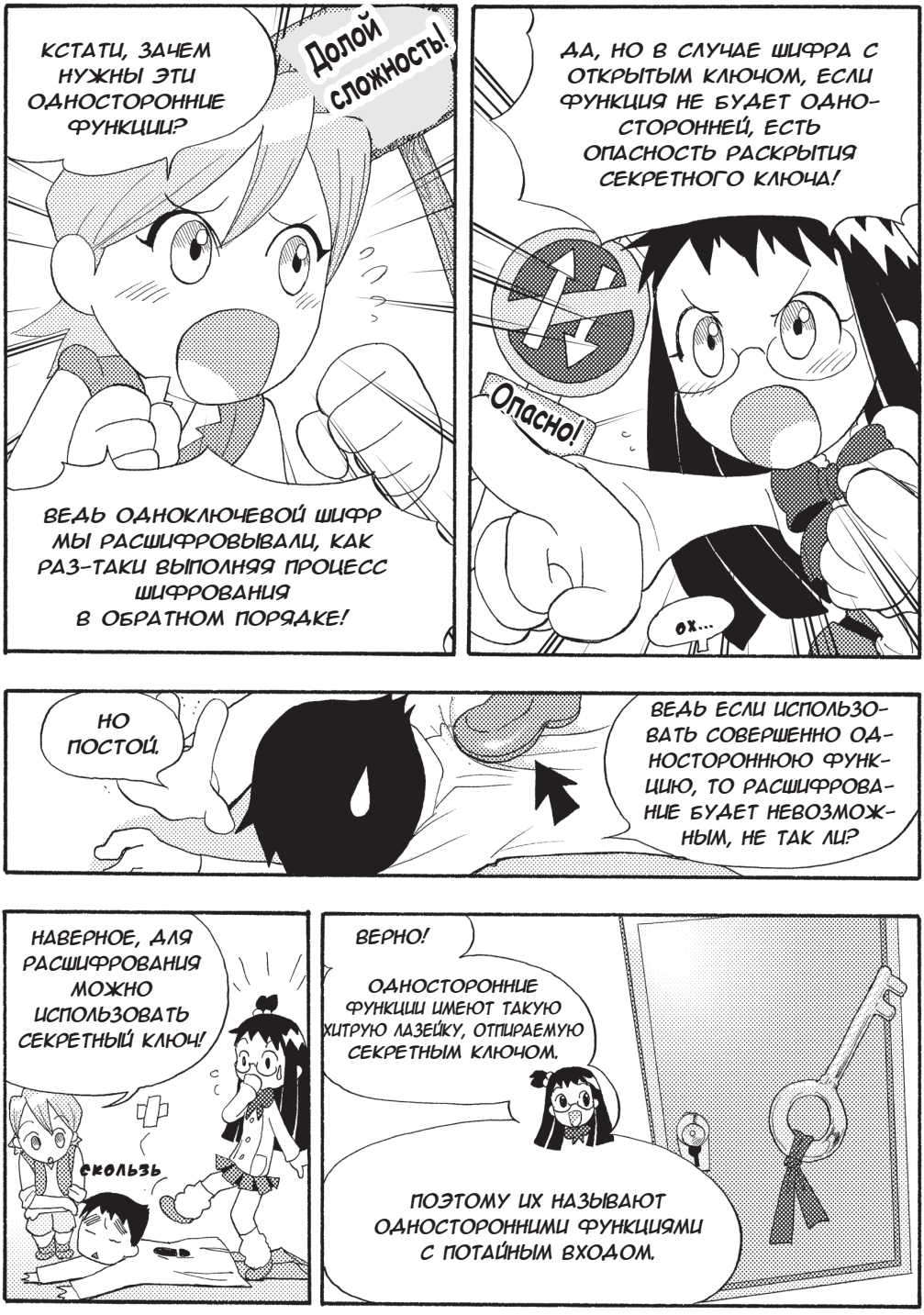
(2) Задача дискретного логарифмирования

Рассмотрим следующую формулу сравнения по модулю p.

$$a^x \equiv y \pmod{p}$$

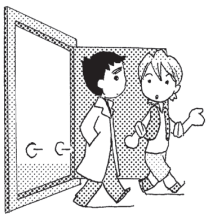
По известным значениям a и x достаточно легко найти значение y, однако по известным значениям a и y чрезвычайно трудно найти значение x, которое является логарифмом y. Это называется задачей дискретного логарифмирования. Слово «дискретный» означет величину, значение которой не может изменяться непрерывно, другими словами, она может принимать только отстоящие друг от друга значения (см. стр. 175).



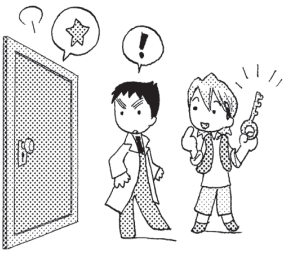

КСТАТИ, ЗАЧЕМ НУЖНЫ ЭТИ ОДНОСТОРОННИЕ ФУНКЦИИ?
Долой сложность!
ВЕДЬ ОДНОКЛЮЧЕВОЙ ШИФР МЫ РАСШИФРОВЫВАЛИ, КАК РАЗ-ТАКИ ВЫПОЛНЯЯ ПРОЦЕСС ШИФРОВАНИЯ В ОБРАТНОМ ПОРЯДКЕ!
ДА, НО В СЛУЧАЕ ШИФРА С ОТКРЫТЫМ КЛЮЧОМ, ЕСЛИ ФУНКЦИЯ НЕ БУДЕТ ОДНОСТОРОННЕЙ, ЕСТЬ ОПАСНОСТЬ РАСКРЫТИЯ СЕКРЕТНОГО КЛЮЧА!
Опасно!
ох...
НО ПОСТОЙ.
ВЕДЬ ЕСЛИ ИСПОЛЬЗОВАТЬ СОВЕРШЕННО ОДНОСТОРОННЮЮ ФУНКЦИЮ, ТО РАСШИФРОВАНИЕ БУДЕТ НЕВОЗМОЖНЫМ, НЕ ТАК ЛИ?
НАВЕРНОЕ, ДЛЯ РАСШИФРОВАНИЯ МОЖНО ИСПОЛЬЗОВАТЬ СЕКРЕТНЫЙ КЛЮЧ!
скользь
ВЕРНО!
ОДНОСТОРОННИЕ ФУНКЦИИ ИМЕЮТ ТАКУЮ ХИТРУЮ ЛАЗЕЙКУ, ОТПИРАЕМУЮ СЕКРЕТНЫМ КЛЮЧОМ.
ПОЭТОМУ ИХ НАЗЫВАЮТ ОДНОСТОРОННИМИ ФУНКЦИЯМИ С ПОТАЙНЫМ ВХОДОМ.

Если выйти наружу из комнаты с автоматически запирающейся дверью, не имея при себе ключа, то вы не сможете вернуться внутрь. Функции, имеющие такое устройство, называют односторонними функциями с потайным входом.

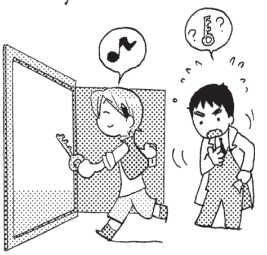
Из комнаты наружу можно выйти, даже не имея ключа.

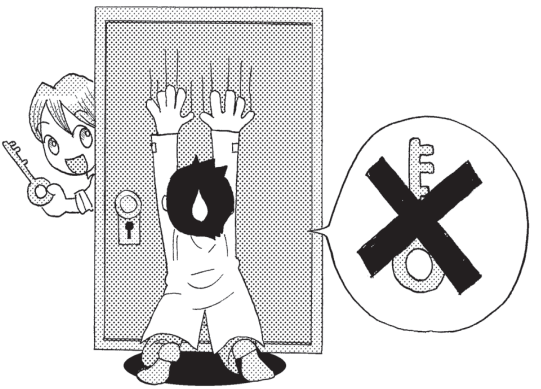


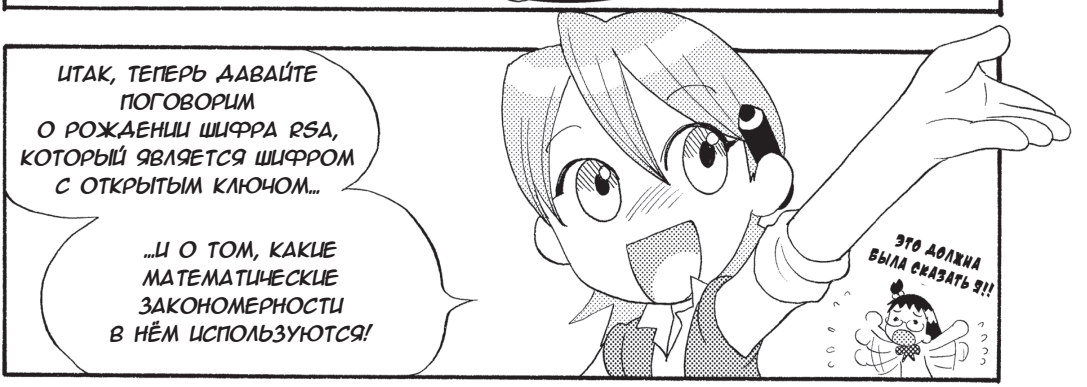
Войти в комнату без ключа не получится.



Если ключ есть, то можно войти в комнату.







✤ Рождение шифра RSA

Шифр RSA, опубликованный в 1977 году, является первым в мире шифром с открытым ключом.

Название RSA образовано из первых букв фамилий разработчиков – американских учёных Рональда Ривеста (Rivest), Ади Шамира (Shamir) и Леонарда Адлемана (Adleman).

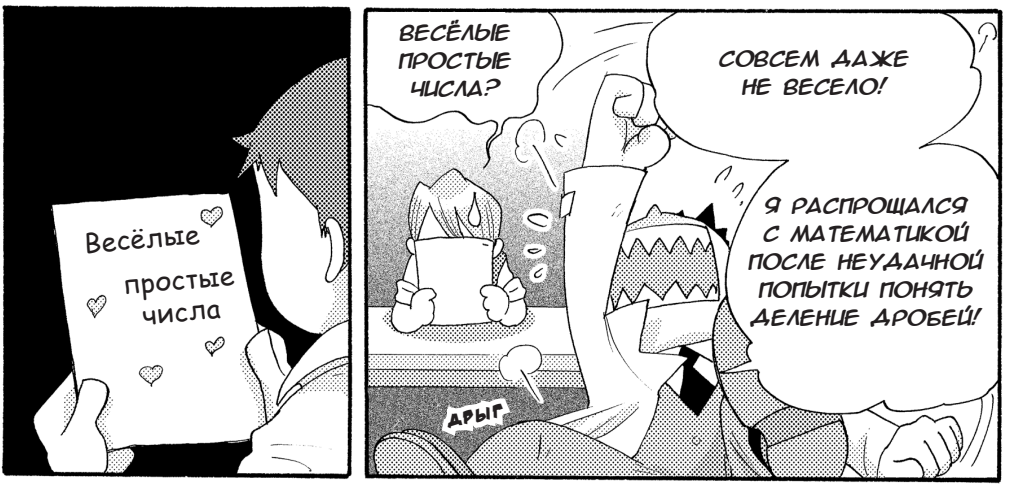
Стойкость шифра обеспечивает задача факторизации целых чисел. В 1977 году журнал Scientific American опубликовал задачу, предложенную этими тремя учёными, в которой предлагалось факторизовать (разложить на простые множители) нижеприведённое 129-значное натуральное число и прочитать зашифрованное сообщение.

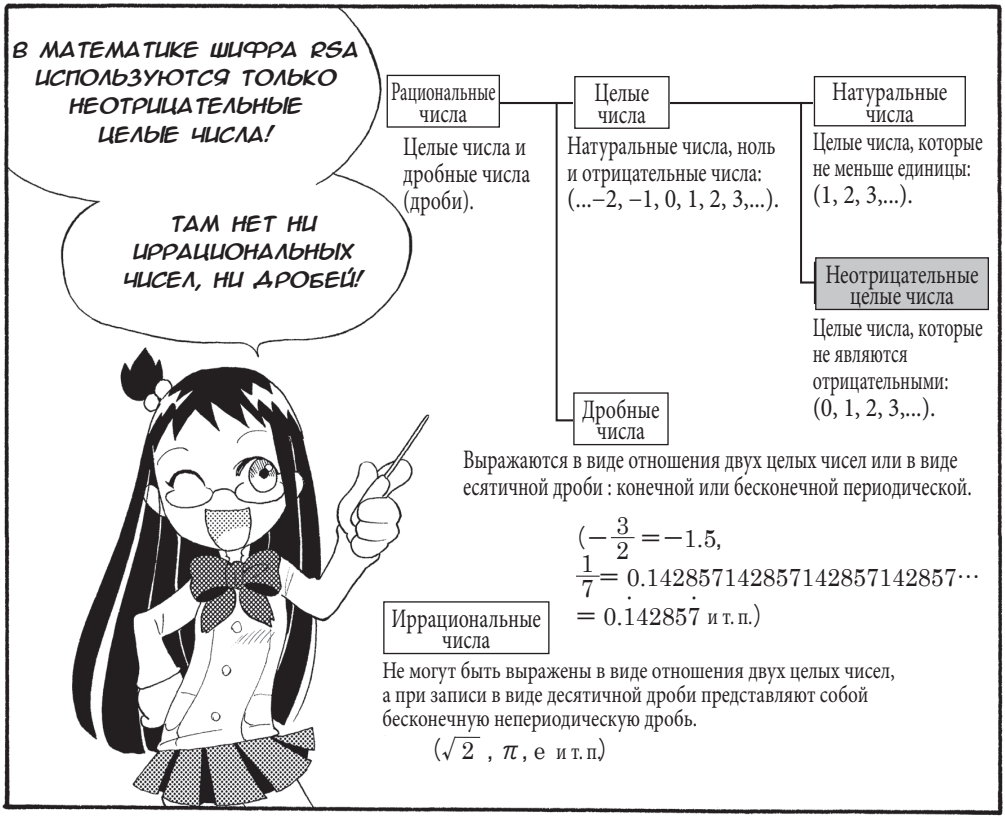
```
114381625757888867669235779976146612010218296721242
362562561842935706935245733897830597123563958705058
989075147599290026879543541
```

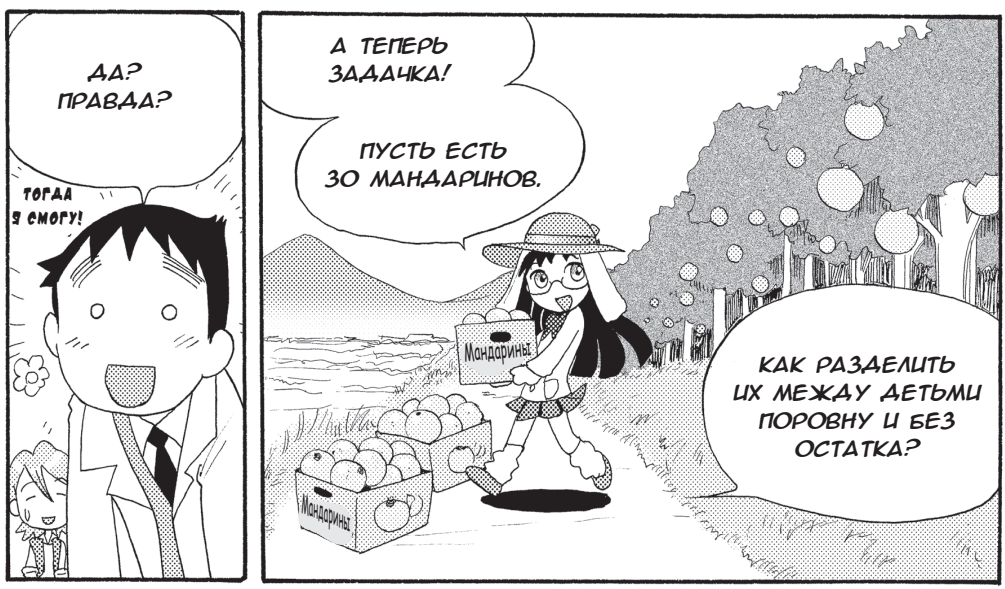
Эта задача на факторизацию была решена спустя 17 лет, в 1994 году, путём распределённых вычислений на 1600 компьютерах, и сообщение было расшифровано. Может показаться, что 17 лет – это очень большой срок, но один из разработчиков шифра RSA – Рональд Ривест, прогнозировал, что 1000 лет на расшифровку точно потребуется. Получается, что, с точки зрения разработчиков, шифр был вскрыт очень быстро. Кстати, зашифрованное сообщение было следующим:
THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE.

В настоящее время в шифре RSA используются десятичные числа, имеющие не менее 300 разрядов, для факторизации которых потребовались бы астрономические сроки.

3-2 Простые числа и факторизация




В МАТЕМАТИКЕ ШИФРА RSA ИСПОЛЬЗУЮТСЯ ТОЛЬКО НЕОТРИЦАТЕЛЬНЫЕ ЦЕЛЫЕ ЧИСЛА!
ТАМ НЕТ НИ ИРРАЦИОНАЛЬНЫХ ЧИСЕЛ, НИ ДРОБЕЙ!
Рациональные числа
Целые числа и дробные числа (дроби).
Целые числа
Натуральные числа, ноль и отрицательные числа: (...−2, −1, 0, 1, 2, 3,...).
Натуральные числа
Целые числа, которые не меньше единицы: (1, 2, 3,...).
Неотрицательные целые числа
Целые числа, которые не являются отрицательными: (0, 1, 2, 3,...).
Дробные числа
Выражаются в виде отношения двух целых чисел или в виде есятичной дроби : конечной или бесконечной периодической.
$(-\frac{3}{2} = -1.5,$
$\frac{1}{7} = 0.\dot{1}4285\dot{7}1428\dot{5}7142\dot{8}57142857\cdots$
$= 0.\dot{1}4285\dot{7}$ и т. п.)
Иррациональные числа
Не могут быть выражены в виде отношения двух целых чисел, а при записи в виде десятичной дроби представляют собой бесконечную непериодическую дробь.
$(\sqrt{2}\ ,\ \pi\ ,\ e$ и т. п.)


ДА? ПРАВДА?
ТОГДА Я СМОГУ!
А ТЕПЕРЬ ЗАДАЧКА!
ПУСТЬ ЕСТЬ 30 МАНДАРИНОВ.
Мандарины
Мандарины
КАК РАЗДЕЛИТЬ ИХ МЕЖДУ ДЕТЬМИ ПОРОВНУ И БЕЗ ОСТАТКА?

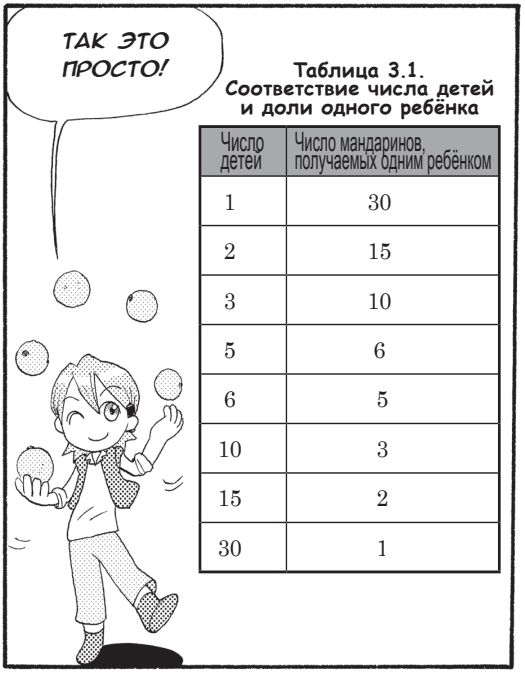
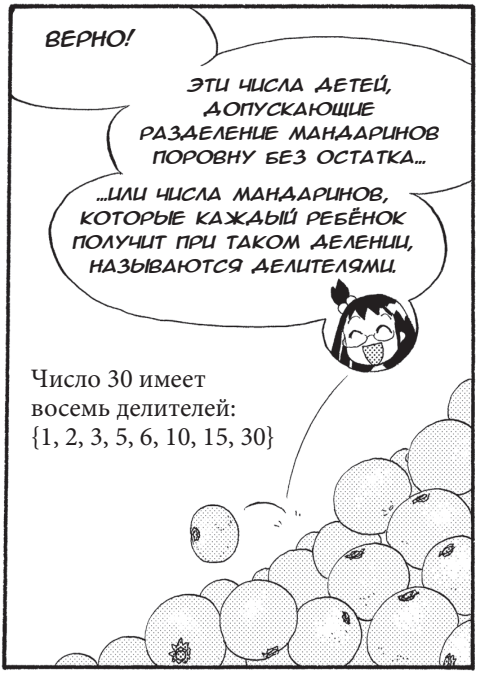


Таблица 3.1. Соответствие числа детей и доли одного ребёнка

Число детей	Число мандаринов, получаемых одним ребёнком
1	30
2	15
3	10
5	6
6	5
10	3
15	2
30	1



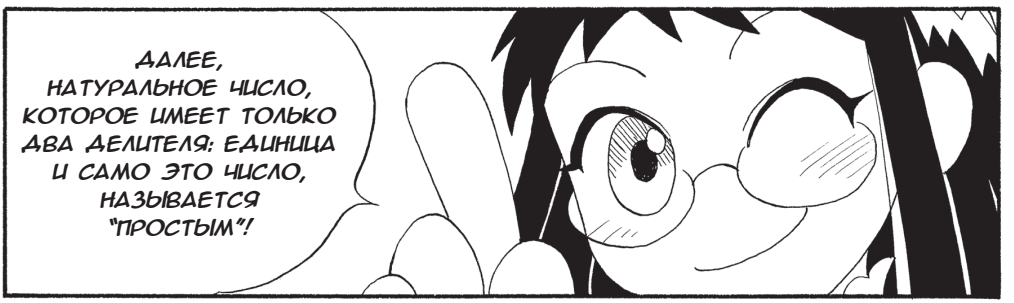
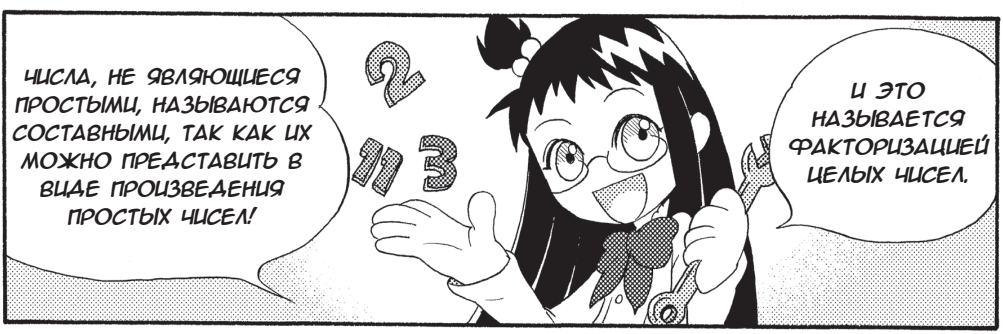


Таблица 3.2. Тест на простоту натуральных чисел от 2 до 20

2	Делится без остатка только на себя (2) и на 1	Простое число
3	Делится без остатка только на себя (3) и на 1	Простое число
4	Делится без остатка на 2	Непростое число
5	Делится без остатка только на себя (5) и на 1	Простое число
6	Делится без остатка на 2 и на 3	Непростое число
7	Делится без остатка только на себя (7) и на 1	Простое число
8	Делится без остатка на 2 и на 4	Непростое число
9	Делится без остатка на 3	Непростое число
10	Делится без остатка на 2 и на 5	Непростое число
11	Делится без остатка только на себя (11) и на 1	Простое число
12	Делится без остатка на 2, на 3, на 4 и на 6	Непростое число
13	Делится без остатка только на себя (13) и на 1	Простое число
14	Делится без остатка на 2 и на 7	Непростое число
15	Делится без остатка на 3 и на 5	Непростое число
16	Делится без остатка на 2, на 4 и на 8	Непростое число
17	Делится без остатка только на себя (17) и на 1	Простое число
18	Делится без остатка на 2, на 3, на 6 и на 9	Непростое число
19	Делится без остатка только на себя (19) и на 1	Простое число
20	Делится без остатка на 2, на 4, на 5 и на 10	Непростое число



$4 = 2^2 = 2 \times 2$

$6 = 2 \times 3$

$8 = 2^3 = 2 \times 2 \times 2$

$9 = 3^2 = 3 \times 3$

$10 = 2 \times 5$

$12 = 2^2 \times 3 = 2 \times 2 \times 3$

$14 = 2 \times 7$

$15 = 3 \times 5$

$16 = 2^4 = 2 \times 2 \times 2 \times 2$

$18 = 2 \times 3^2 = 2 \times 3 \times 3$

$20 = 2^2 \times 5 = 2 \times 2 \times 5$

ЛЮБОЕ СОСТАВНОЕ ЧИСЛО МОЖНО РАЗЛОЖИТЬ НА ПРОСТЫЕ МНОЖИТЕЛИ ТОЛЬКО ОДНИМ СПОСОБОМ, И ЭТО СВОЙСТВО НАЗЫВАЕТСЯ "ОДНОЗНАЧНОСТЬЮ ФАКТОРИЗАЦИИ ЦЕЛЫХ ЧИСЕЛ".

ЕДИНИЦУ НЕ ОТНОСЯТ К ПРОСТЫМ ЧИСЛАМ ИМЕННО С ЦЕЛЬЮ СОХРАНЕНИЯ ЭТОГО СВОЙСТВА ОДНОЗНАЧНОСТИ.

ВОТ КАК?!

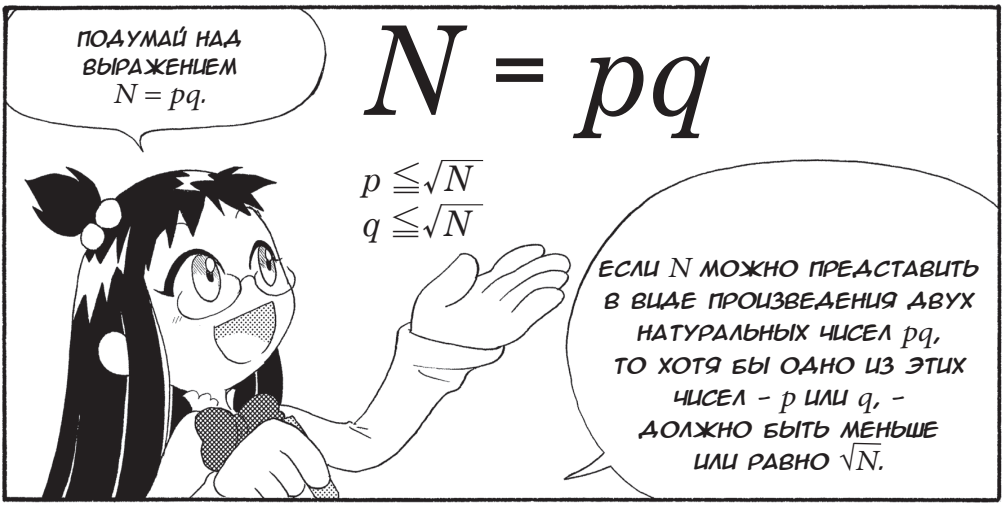
ЗНАЧИТ, ЧТОБЫ НАЙТИ ПРОСТЫЕ ЧИСЛА, НУЖНО ПРОВЕРЯТЬ НА ПРОСТОТУ ВСЕ ЧИСЛА ОДНО ЗА ДРУГИМ?

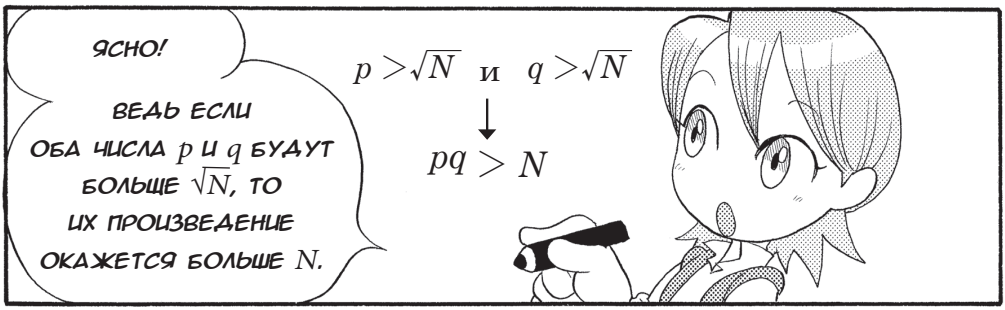
СУЩЕСТВУЕТ МЕТОД ПОД НАЗВАНИЕМ "РЕШЕТО ЭРАТОСФЕНА".

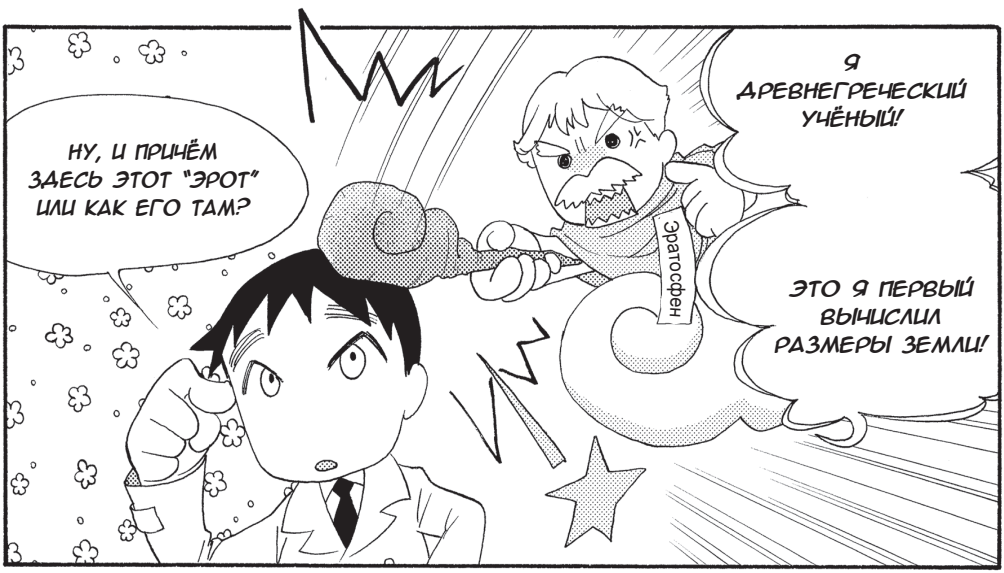
В НЁМ ИСПОЛЬЗУЕТСЯ ВОТ ТАКОЕ СВОЙСТВО!

Если натуральное число N не делится без остатка ни на одно из простых чисел, которые меньше или равны $\sqrt{N}$, то это натуральное число N является простым.

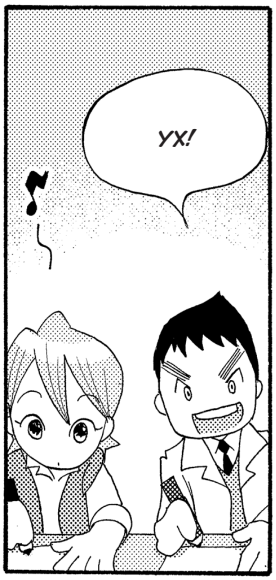
НО ПОЧЕМУ ЭТО ТАК?


ПОДУМАЙ НАД ВЫРАЖЕНИЕМ $N = pq$.
$N = pq$
$p \leqq \sqrt{N}$
$q \leqq \sqrt{N}$
ЕСЛИ N МОЖНО ПРЕДСТАВИТЬ В ВИДЕ ПРОИЗВЕДЕНИЯ ДВУХ НАТУРАЛЬНЫХ ЧИСЕЛ pq, ТО ХОТЯ БЫ ОДНО ИЗ ЭТИХ ЧИСЕЛ – p ИЛИ q, – ДОЛЖНО БЫТЬ МЕНЬШЕ ИЛИ РАВНО $\sqrt{N}$.

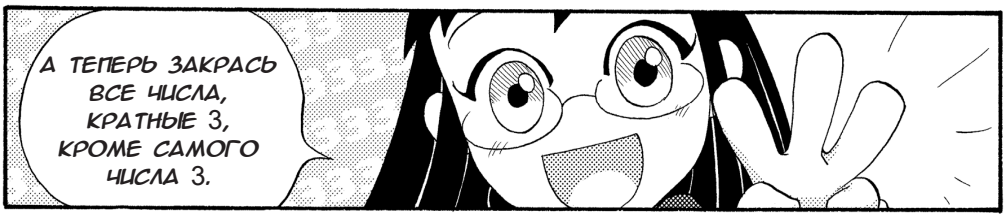

ЯСНО!
ВЕДЬ ЕСЛИ ОБА ЧИСЛА p И q БУДУТ БОЛЬШЕ $\sqrt{N}$, ТО ИХ ПРОИЗВЕДЕНИЕ ОКАЖЕТСЯ БОЛЬШЕ N.
$p > \sqrt{N}$ И $q > \sqrt{N}$
$pq > N$


НУ, И ПРИЧЁМ ЗДЕСЬ ЭТОТ "ЭРОТ" ИЛИ КАК ЕГО ТАМ?
Я ДРЕВНЕГРЕЧЕСКИЙ УЧЁНЫЙ!
Эратосфен
ЭТО Я ПЕРВЫЙ ВЫЧИСЛИЛ РАЗМЕРЫ ЗЕМЛИ!

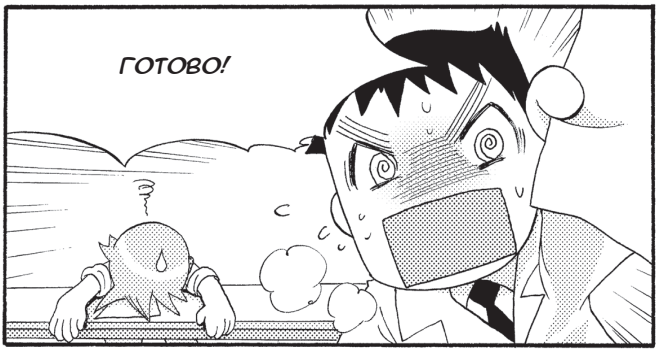
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400



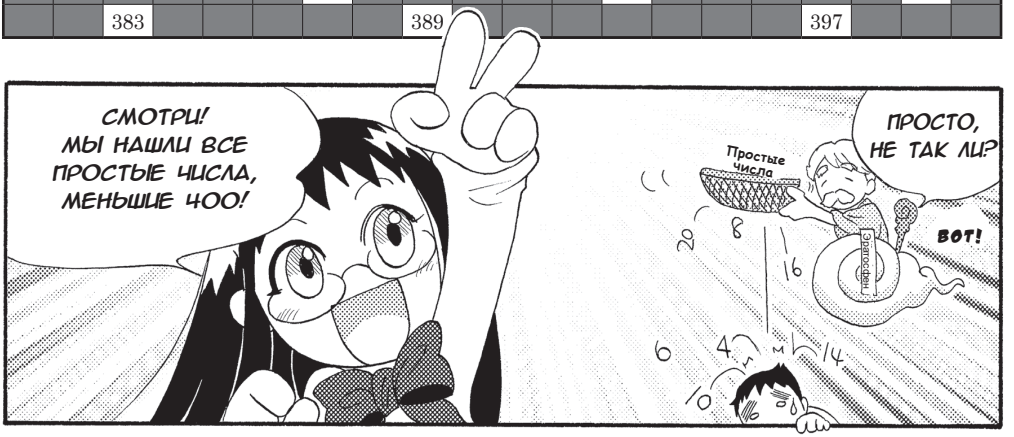
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280
281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380
381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

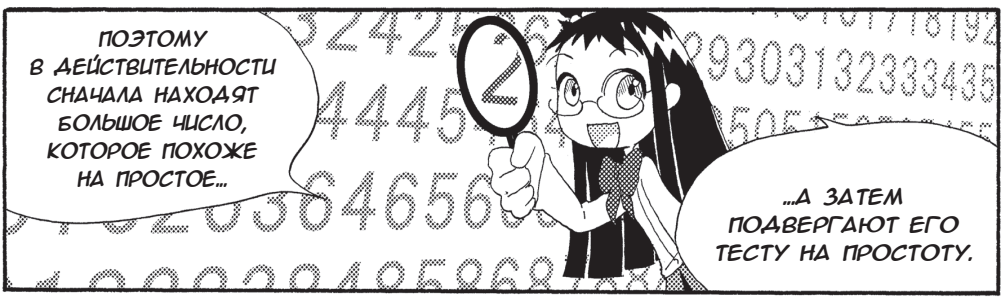


	2	3		5		7		9		11		13		15		17		19	
21		23		25		27		29		31		33		35		37		39	
41		43		45		47		49		51		53		55		57		59	
61		63		65		67		69		71		73		75		77		79	
81		83		85		87		89		91		93		95		97		99	
101		103		105		107		109		111		113		115		117		119	
121		123		125		127		129		131		133		135		137		139	
141		143		145		147		149		151		153		155		157		159	
161		163		165		167		169		171		173		175		177		179	
181		183		185		187		189		191		193		195		197		199	
201		203		205		207		209		211		213		215		217		219	
221		223		225		227		229		231		233		235		237		239	
241		243		245		247		249		251		253		255		257		259	
261		263		265		267		269		271		273		275		277		279	
281		283		285		287		289		291		293		295		297		299	
301		303		305		307		309		311		313		315		317		319	
321		323		325		327		329		331		333		335		337		339	
341		343		345		347		349		351		353		355		357		359	
361		363		365		367		369		371		373		375		377		379	
381		383		385		387		389		391		393		395		397		399	



	2	3		5		7				11		13				17		19	
		23						29		31						37			
41		43				47						53						59	
61						67				71		73						79	
		83						89								97			
101		103				107		109				113							
						127				131						137		139	
								149		151						157			
		163				167						173						179	
181										191		193				197		199	
										211									
		223				227		229				233						239	
241										251						257			
		263						269		271						277			
281		283										293							
						307				311		313				317			
										331						337			
						347		349				353						359	
						367						373						379	
		383						389								397			





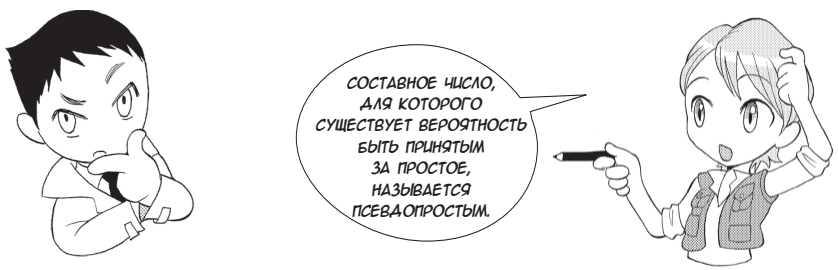
✤ Тест на простоту

Хотя решето Эратосфена является методом, позволяющим надёжно обнаруживать простые числа, на определение с его помощью простоты очень больших чисел требуется слишком много времени.

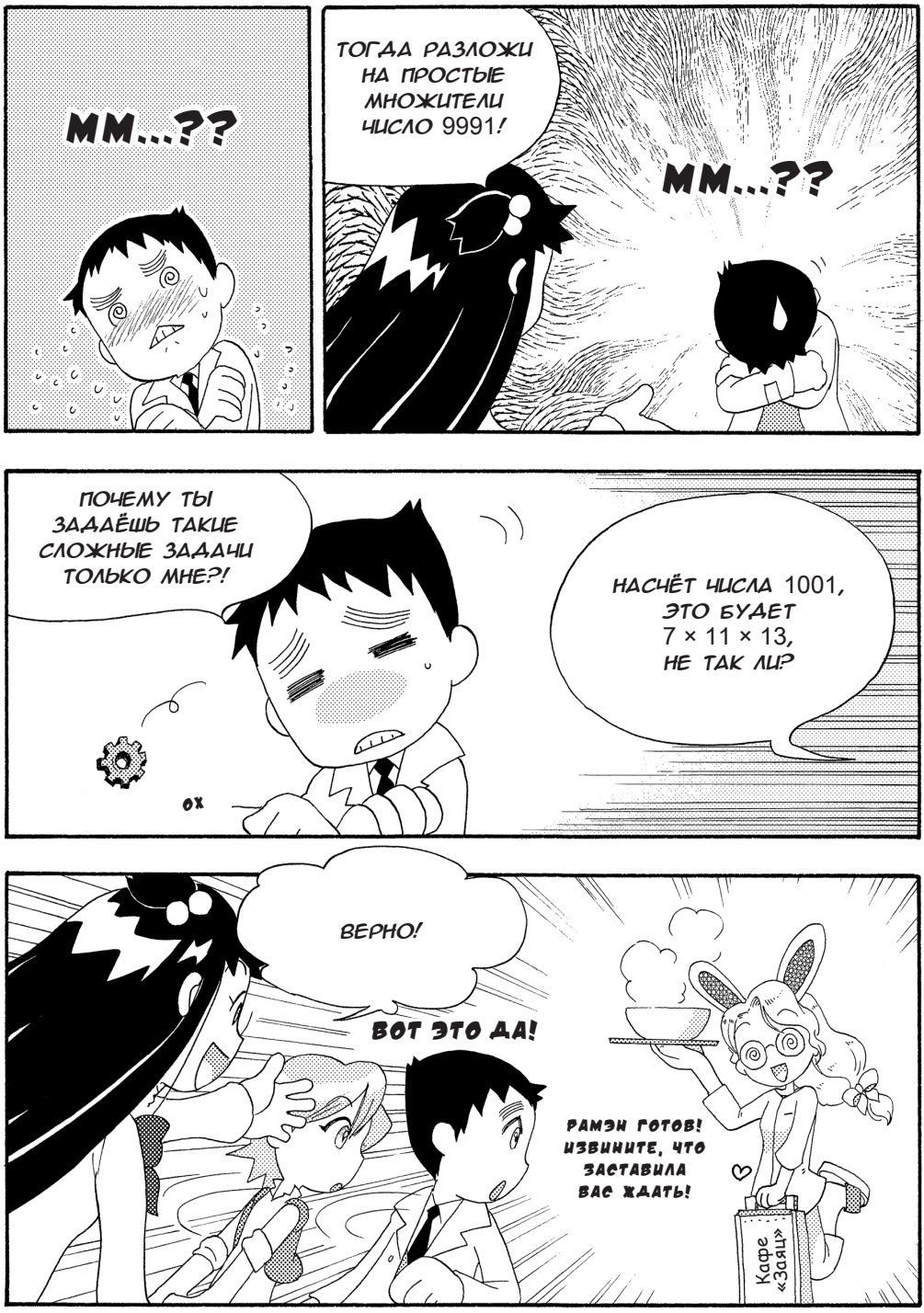
В связи с этим в качестве теста на простоту используются методы, не являющиеся достоверными на все 100 %, однако позволяющие с большой долей вероятности полагать, что исследуемое число является простым.

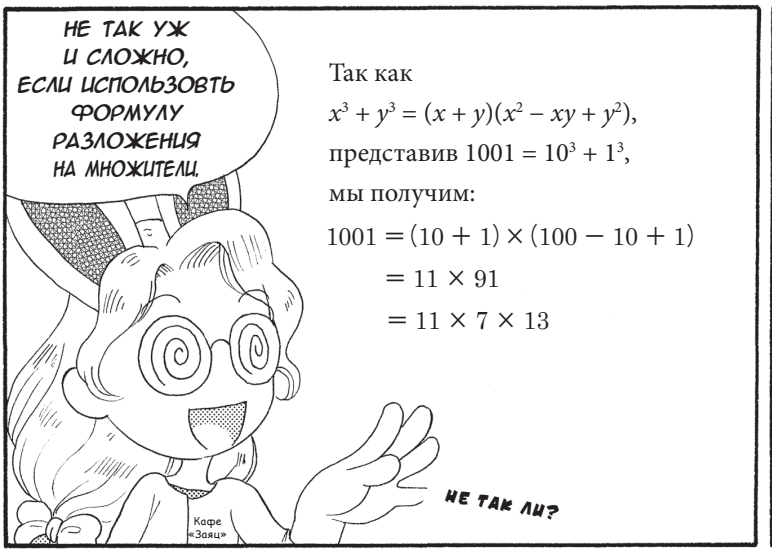
Тест Ферма позволяет с некоторой долей вероятности полагать, что число n является простым, если для любого целого числа a, которое не делится на n, выполняется сравнение $a^{n-1} = 1 \pmod{n}$ (см. стр. 156), но при его использовании существует опасность того, что не простое (то есть составное) число будет ошибочно принято за простое, хотя вероятность этого не так высока.

Этот недостаток теста Ферма был исправлен в тесте Миллера-Рабина, позволяющем снизить вероятность ошибочной оценки не менее чем в 4 раза, по сравнению с тестом Ферма, и таким образом почти надёжно обнаруживать простые числа.

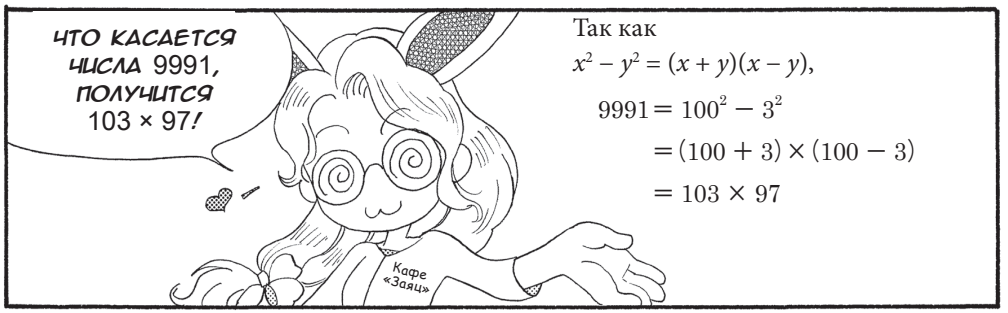


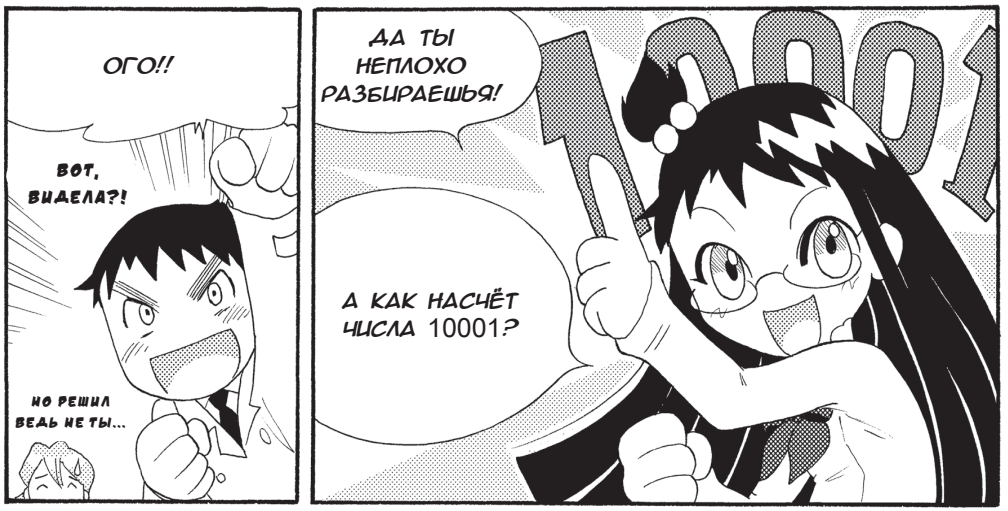

ТЕПЕРЬ ДАВАЙТЕ УБЕДИМСЯ В ТОМ, ЧТО ФАКТОРИЗАЦИЯ ЦЕЛЫХ ЧИСЕЛ – ЭТО СЛОЖНАЯ ЗАДАЧА!
РИО, РАЗЛОЖИ НА ПРОСТЫЕ МНОЖИТЕЛИ ЧИСЛО 35!
НУ, ЭТО ПРОСТО!
5 × 7
ВЕРНО!
А ТЕПЕРЬ – ЧИСЛО 77.
7 × 11
ПРАВИЛЬНО!
ТЕПЕРЬ ТВОЯ ОЧЕРЕДЬ, БРАТЕЦ!
РАЗЛОЖИ-КА НА ПРОСТЫЕ МНОЖИТЕЛИ ЧИСЛО 1001!
ОЙ!


ММ...??
ТОГДА РАЗЛОЖИ НА ПРОСТЫЕ МНОЖИТЕЛИ ЧИСЛО 9991!
ММ...??
ПОЧЕМУ ТЫ ЗАДАЁШЬ ТАКИЕ СЛОЖНЫЕ ЗАДАЧИ ТОЛЬКО МНЕ?!
ОХ
НАСЧЁТ ЧИСЛА 1001, ЭТО БУДЕТ 7 × 11 × 13, НЕ ТАК ЛИ?
ВЕРНО!
ВОТ ЭТО ДА!
РАМЭН ГОТОВ! ИЗВИНИТЕ, ЧТО ЗАСТАВИЛА ВАС ЖДАТЬ!
Кафе «Заяц»

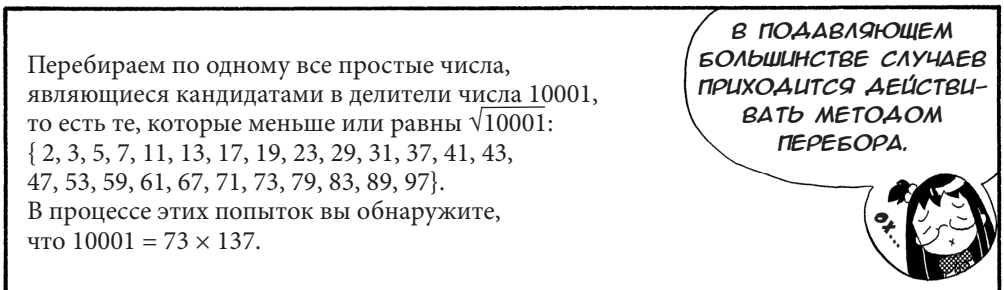

НЕ ТАК УЖ И СЛОЖНО, ЕСЛИ ИСПОЛЬЗОВТЬ ФОРМУЛУ РАЗЛОЖЕНИЯ НА МНОЖИТЕЛИ.
Так как
$x^3 + y^3 = (x + y)(x^2 - xy + y^2)$,
представив $1001 = 10^3 + 1^3$,
мы получим:
$1001 = (10 + 1) \times (100 - 10 + 1)$
$= 11 \times 91$
$= 11 \times 7 \times 13$
НЕ ТАК ЛИ?
Кафе «Заяц»

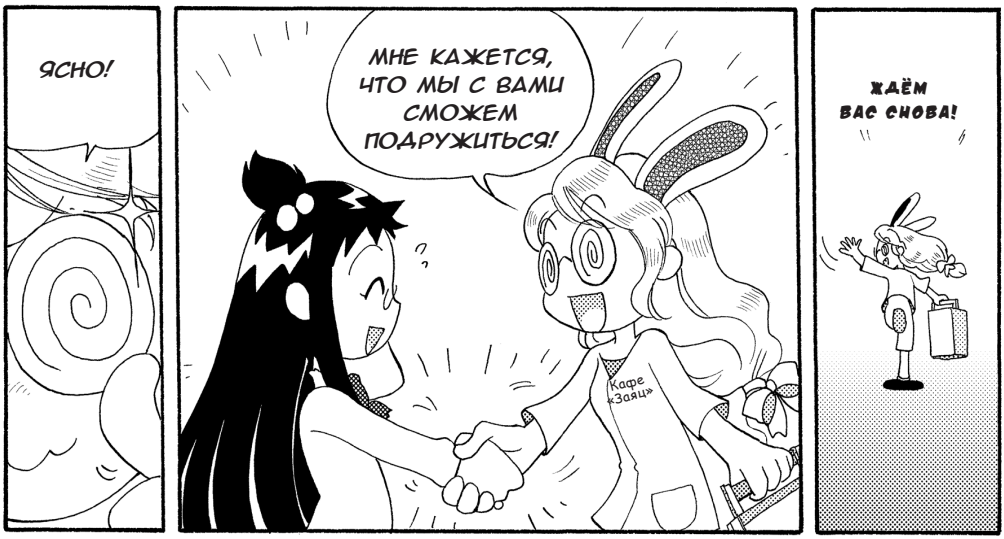

А, ВОТ ОНО ЧТО!
ОН И ВПРАВДУ ВСЁ ПОНЯЛ?


ЧТО КАСАЕТСЯ ЧИСЛА 9991, ПОЛУЧИТСЯ 103 × 97!
Так как
$x^2 - y^2 = (x + y)(x - y)$,
$9991 = 100^2 - 3^2$
$= (100 + 3) \times (100 - 3)$
$= 103 \times 97$
Кафе «Заяц»

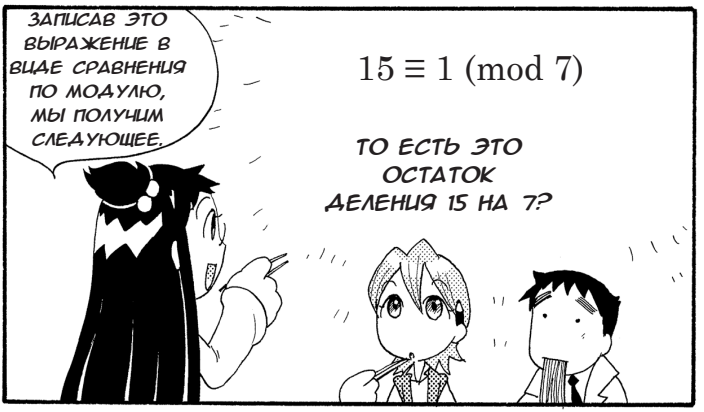

ОГО!!
ВОТ, ВИДЕЛА?!
НО РЕШИЛ ВЕДЬ НЕ ТЫ...
ДА ТЫ НЕПЛОХО РАЗБИРАЕШЬЯ!
А КАК НАСЧЁТ ЧИСЛА 10001?
10001

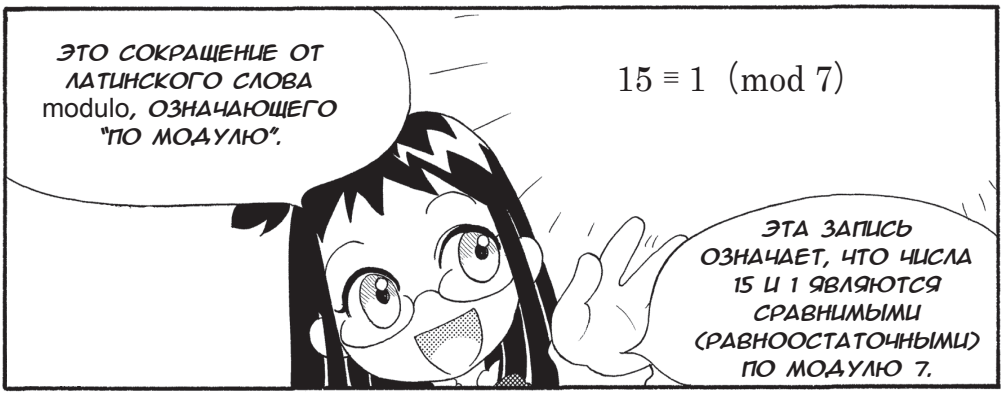
Перебираем по одному все простые числа, являющиеся кандидатами в делители числа 10001, то есть те, которые меньше или равны $\sqrt{10001}$:
{ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}.
В процессе этих попыток вы обнаружите, что $10001 = 73 \times 137$.





3-3 Модульная арифметика





$$a \equiv b \pmod N$$

Это общая форма записи сравнения по модулю.
Число N называется модулем сравнения.
Числа a и b в этой формуле называются сравнимыми (равноостаточными) числами по модулю N.
За рубежом вместо знака «≡» иногда используют знак «=».

mod

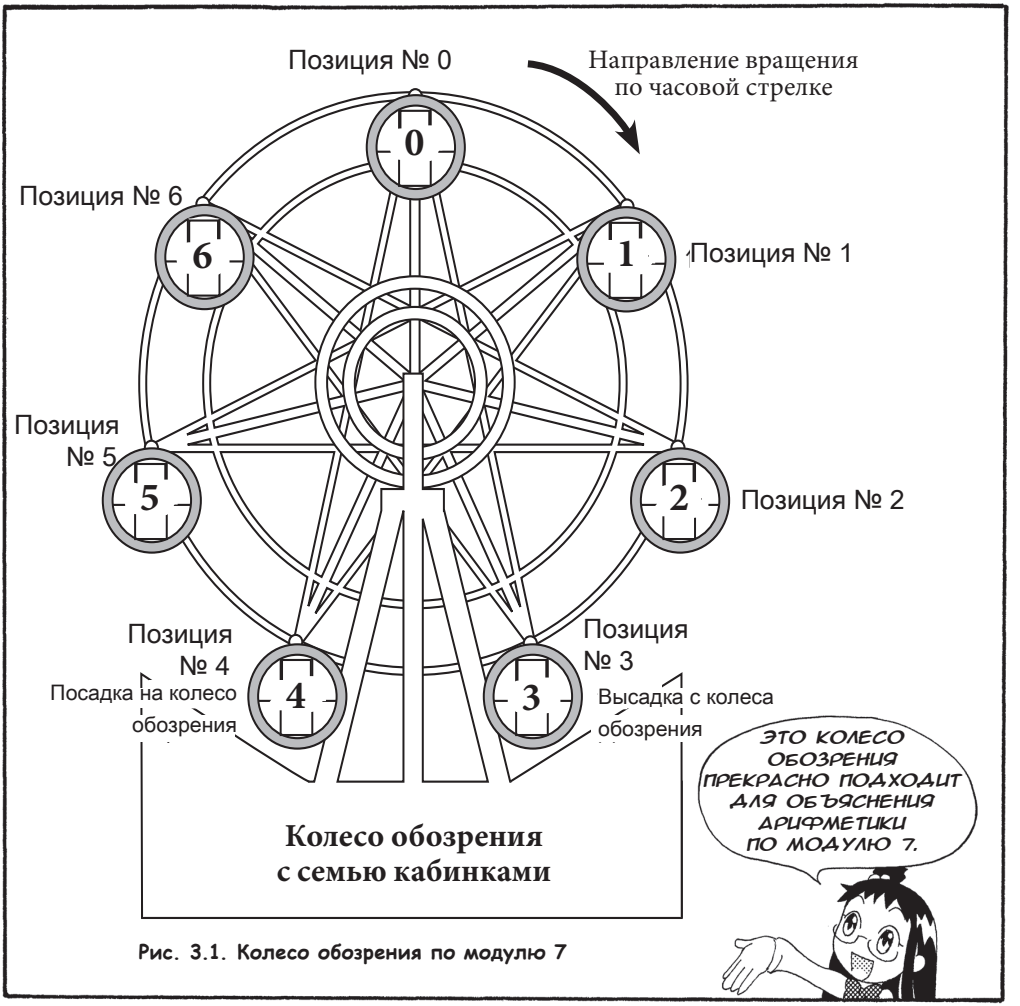


Рис. 3.1. Колесо обозрения по модулю 7

✤ Сложение по модулю и вычитание по модулю

Для объяснения модульной арифметики мы воспользуемся колесом обозрения, показанным на рис. 3.1. Это колесо имеет семь кабинок – от нулевой до шестой, кроме того, позиции, в которых эти кабинки могут находиться, тоже пронумерованы: от позиции № 0 в самой верхней точке колеса до позиции № 6. Площадка для высадки с колеса обозрения находится в позиции № 3, а площадка для посадки на колесо – в позиции № 4.

В начальном состоянии нулевая кабинка находится в позиции № 0, первая кабинка – в позиции № 1 и так далее, то есть номера всех кабинок совпадают с номерами позиций, в которых эти кабинки находятся. Далее, колесо обозрения спроектировано так, чтобы вращаться по часовой стрелке при операции сложения.

Рассмотрим сначала нулевую кабинку. Если колесо повернётся на 1/7 оборота по часовой стрелке, то нулевая кабинка переместится из позиции № 0 в позицию № 1. Определим это как операцию прибавления единицы (+1).

При повороте на 2/7 оборота нулевая кабинка переместится из позиции № 0 в позицию № 2. Это соответствует операции +2.

При повороте на 7/7 оборота, другими словами, когда колесо совершит один полный оборот, нулевая кабинка, начав движение с позиции № 0, вновь вернётся в позицию № 0. Это соответствует операции +7, которая эквивалентна прибавлению нуля, то есть случаю, когда кабинка не двигалась вообще.

Используя эту модель колеса обозрения, можно прекрасно объяснить таблицу сложения по модулю 7.

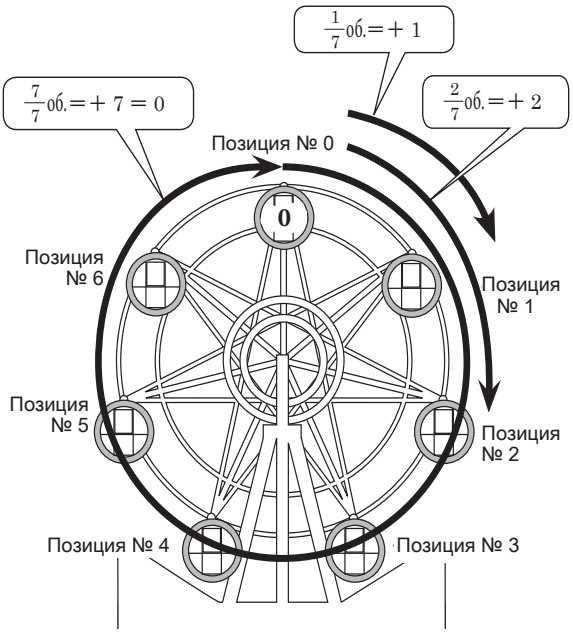


Таблица 3.2. Модель (1) сложения по модулю 7

Теперь рассмотрим, например, операцию сложения 5 + 6.

Представьте что первое слагаемое, то есть 5, – это пятая кабинка в начальном положении (в позиции № 5). В какую позицию она придёт, если колесо сделает 6/7 оборота по часовой стрелке? Как видно по рис. 3.3, при перемещении на 6 позиций по часовой стрелке пятая кабинка переместится из позиции № 5 в позицию № 4, что соответствует нижеприведённой формуле сравнения по модулю.

$$5 + 6 \equiv 4 \pmod 7.$$

Приняв номер кабинки, равный её номеру позиции в начальном состоянии, за слагаемое *a*, легко определить на рисунке её позицию после поворота колеса на *b*/7 оборота. Таким образом, можно убедиться в том, что для любых слагаемых *a* и *b* результат операции сложения по модулю 7 будет соответствовать приведённому в табл. 3.3.

Теперь объясним с помощью этого колеса обозрения операцию вычитания по модулю.

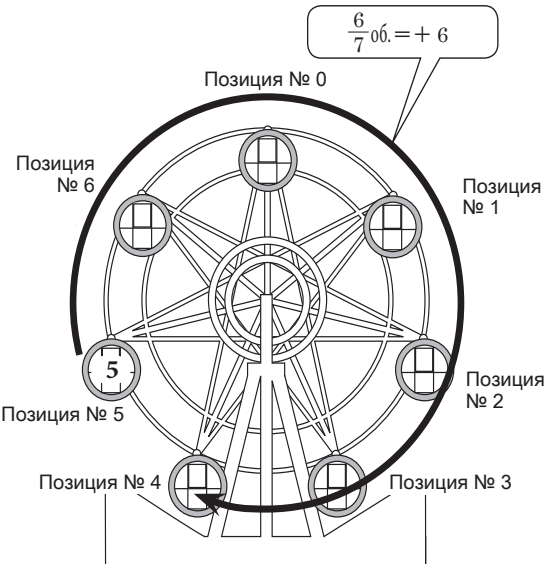


Рис. 3.3. Модель (2) сложения по модулю 7

Таблица 3.3. *Сложение a + b по модулю 7*

a \ *b*	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

Рассмотрим сначала нулевую кабинку. Если колесо повернётся на 1/7 оборота против часовой стрелки, то нулевая кабинка переместится из позиции № 0 в позицию № 6. Определим это как операцию вычитания единицы (–1).

При повороте на 2/7 оборота против часовой стрелки нулевая кабинка переместится из позиции № 0 в позицию № 5. Это соответствует операции вычитания двух (–2).

При повороте на один полный оборот против часовой стрелки нулевая кабинка, начав движение с позиции № 0, вновь вернётся в позицию № 0. Это соответствует операции вычитания семи (–7), которая эквивалентна вычитанию нуля, то есть случаю, когда кабинка не двигалась вообще.

Эта модель колеса обозрения позволяет прекрасно понять, почему в ячейках таблицы вычитания по модулю 7 содержатся те или иные значения.

Представьте что в операции вычитания 3–4 число 3 – это третья кабинка в начальном положении (в позиции № 3). Посмотрим по рис. 3.4, в какую позицию она придёт, если колесо сделает 4/7 оборота против часовой стрелки. Как ясно из рисунка, при перемещении на 4 позиции против часовой стрелки третья кабинка переместится из позиции № 3 в позицию № 6, что соответствует нижеприведённой формуле сравнения по модулю.

$$3 - 4 \equiv 6 \pmod 7.$$

Приняв номер кабинки, равный её номеру позиции в начальном состоянии, за уменьшаемое a, легко определить на рисунке её позицию после поворота колеса на $b/7$ оборота против часовой стрелки. Таким образом, можно убедиться в том, что для любых чисел a и b результат операции вычитания $a - b$ по модулю 7 будет соответствовать приведённому в табл. 3.4.

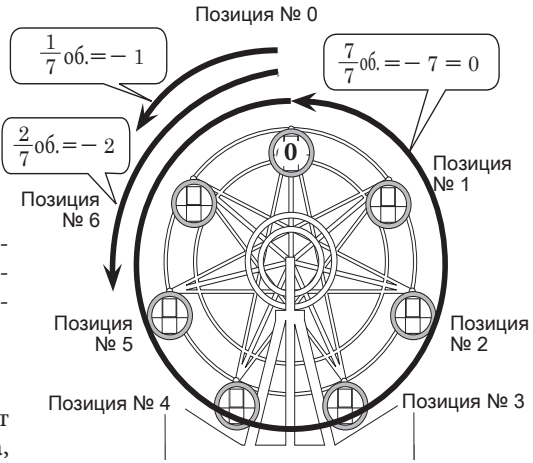


Рис. 3.4. Модель (1) вычитания по модулю 7

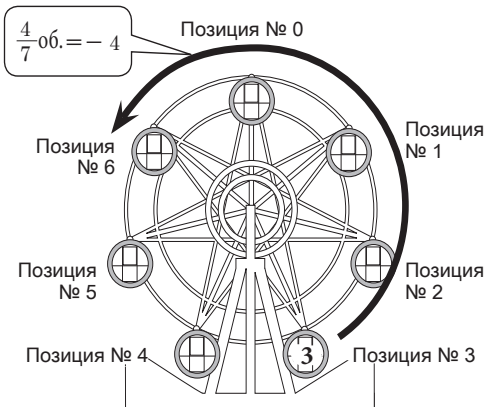
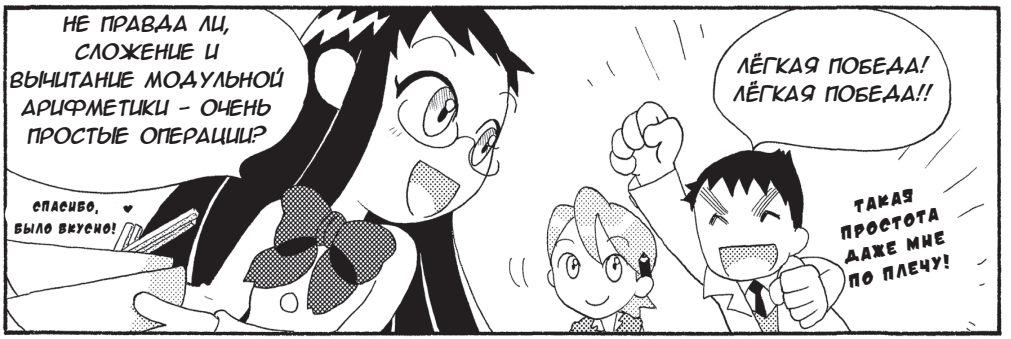
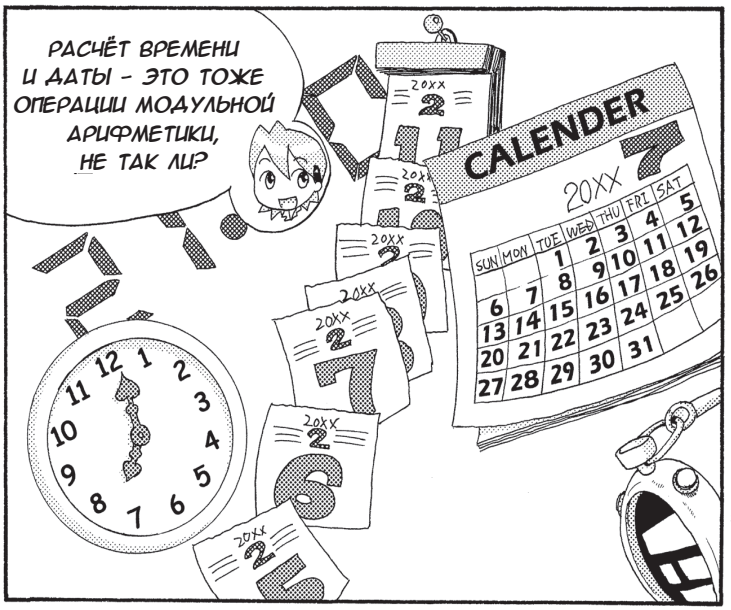


Рис.3.4. Модель (2) вычитания по модулю 7

Таблица 3.4. Вычитание $a - b$ по модулю 7

a \ b	0	1	2	3	4	5	6
0	0	6	5	4	3	2	1
1	1	0	6	5	4	3	2
2	2	1	0	6	5	4	3
3	3	2	1	0	6	5	4
4	4	3	2	1	0	6	5
5	5	4	3	2	1	0	6
6	6	5	4	3	2	1	0


НЕ ПРАВДА ЛИ, СЛОЖЕНИЕ И ВЫЧИТАНИЕ МОДУЛЬНОЙ АРИФМЕТИКИ – ОЧЕНЬ ПРОСТЫЕ ОПЕРАЦИИ?
СПАСИБО, БЫЛО ВКУСНО!
ЛЁГКАЯ ПОБЕДА! ЛЁГКАЯ ПОБЕДА!!
ТАКАЯ ПРОСТОТА ДАЖЕ МНЕ ПО ПЛЕЧУ!


РАСЧЁТ ВРЕМЕНИ И ДАТЫ – ЭТО ТОЖЕ ОПЕРАЦИИ МОДУЛЬНОЙ АРИФМЕТИКИ, НЕ ТАК ЛИ?
CALENDER
20XX 7
SUN MON TUE WED THU FRI SAT
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


А ТЕПЕРЬ ДАВАЙ ИЗУЧИМ УМНОЖЕНИЕ И ДЕЛЕНИЕ ПО МОДУЛЮ!
Я ГОТОВ!


НАПРИМЕР, УМНОЖЕНИЕ 5 × 4 ПО МОДУЛЮ 7 БУДЕТ ВОТ ТАКИМ?
$5 \times 4 = 7 \times 2 + 6.$
Следовательно,
$5 \times 4 \equiv 6 \pmod 7.$


СОВЕРШЕННО ВЕРНО!
ТАК ДЕРЖАТЬ!
ВОТ ТАБЛИЦА УМНОЖЕНИЯ ПО МОДУЛЮ 7!

Таблица 3.5. Умножение $a \times b$ по модулю 7

a \ b	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

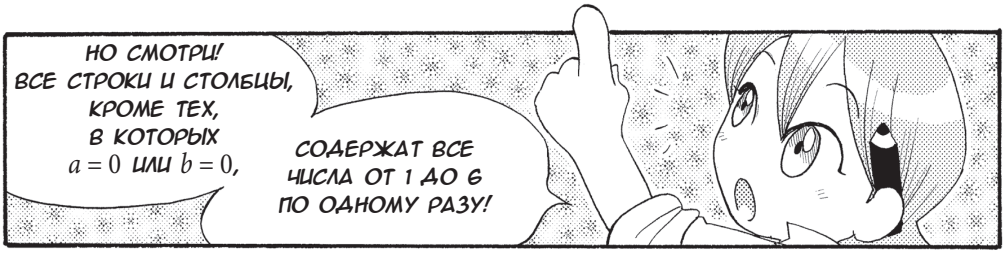
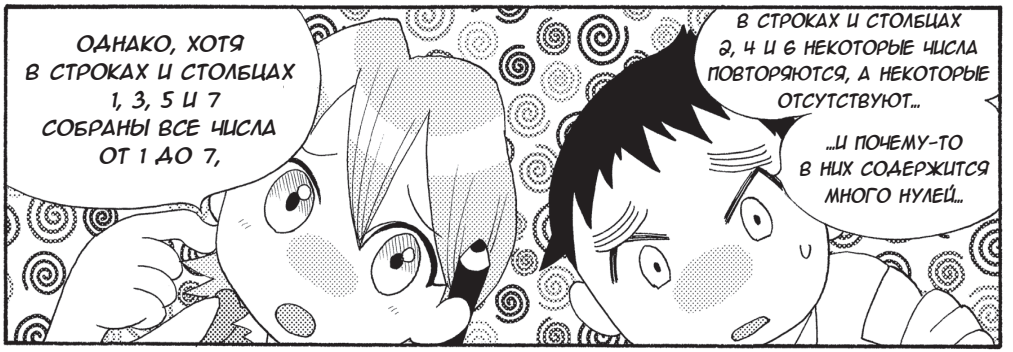


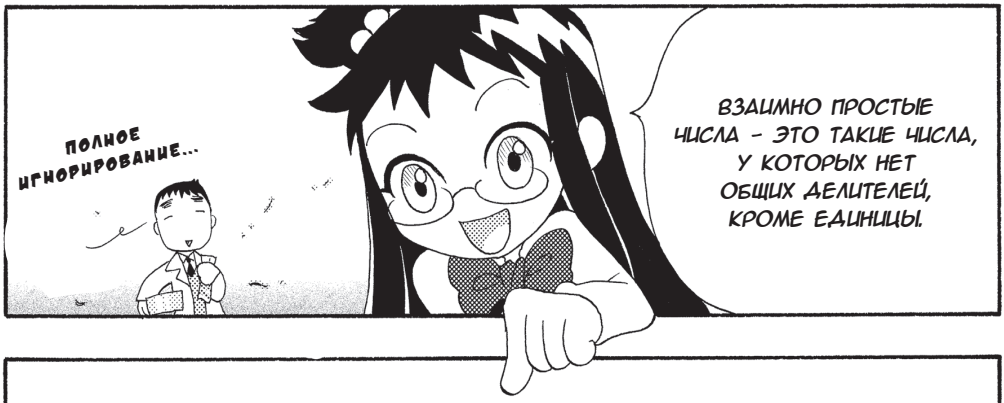
Таблица 3.6. Умножение $a \times b$ по модулю 8

a \ b	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1



Ведь перестаёт выполняться правило умножения:
если $a \times b = 0$, то хотя бы один из множителей, a или b, обязательно должен быть равен 0.

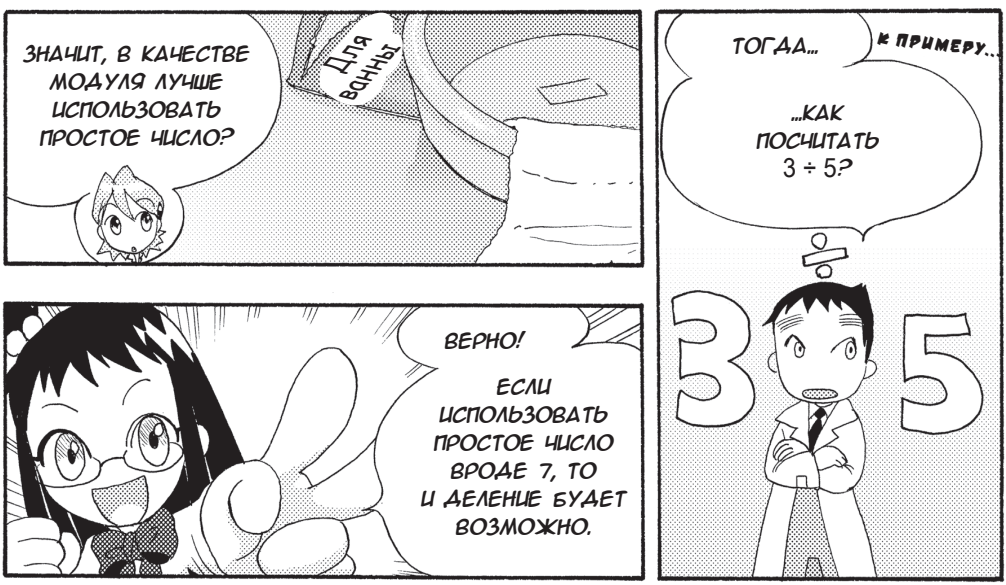
※ Игра слов с названием добавки для ванны «Простая добавка для горячих источников».



Например, числа 8 и 2 имеют общий делитель, отличный от единицы, – это число 2, поэтому они не являются взаимно простыми. Кроме того, числа 4 и 6, содержащиеся в таблице умножения по модулю 8, тоже имеют с числом 8 общий делитель – число 2, поэтому они не являются взаимно простыми по отношению к числу 8.

Напротив, числа 1, 3, 5 и 7 являются взаимно простыми по отношению к числу 8, так как их наибольший общий с числом 8 делитель равен единице.

Таким образом, все простые числа являются взаимно простыми с любыми числами, которые им не кратны. Именно это свойство позволило нам отыскивать простые числа с помощью решета Эратосфена.



$$a \div b = a \times \frac{1}{b}$$

Поделить a на b – это то же самое, что умножить a на число, обратное b. Обратное число называют также обратным элементом.

$$3 \times \frac{1}{3} = 1$$

Таблица 3.7. Умножение $a \times b$ по модулю 7

a \ b	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Согласно табл. 3.7, 1 обратно 1, 2 обратно 4, 3 обратно 5, 4 обратно 2, 5 обратно 3, 6 обратно 6.

Поделить 3 на 5 – значит умножить 3 на число, обратное 5. Следовательно,

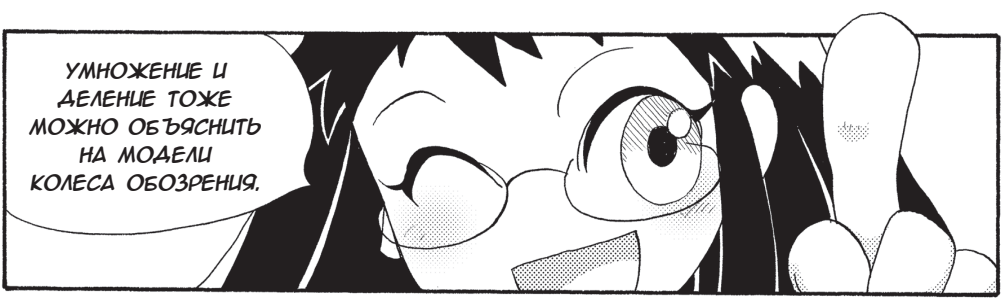
$$3 \div 5 \equiv 3 \times 3 = 9$$
$$9 = 7 + 2 \equiv 2 \pmod 7$$

Таким образом,

$$3 \div 5 \equiv 2 \pmod 7$$

Таблица 3.7. Деление $a \div b$ по модулю 7

a \ b	0	1	2	3	4	5	6
0	—	0	0	0	0	0	0
1	—	1	4	5	2	3	6
2	—	2	1	3	4	6	5
3	—	3	5	1	6	2	4
4	—	4	2	6	1	5	3
5	—	5	6	4	3	1	2
6	—	6	3	2	5	4	1



❀ Умножение по модулю и деление по модулю

Для объяснения умножения по модулю мы опять воспользуемся моделью колеса обозрения, имеющего семь кабинок.

В начальном состоянии нулевая кабинка находится в позиции № 0, первая кабинка – в позиции № 1 и так далее, то есть номера всех кабинок совпадают с номерами позиций, в которых эти кабинки находятся.

Для понимания умножения полезно поразмышлять над таким параметром этой модели, как скорость вращения колеса.

Если колесо вращается со скоростью 1/7 оборота в минуту (другими словами, делает один полный оборот за 7 минут), то номер позиции нулевой кабинки через 3 минуты после начала вращения рассчитывается следующим образом:

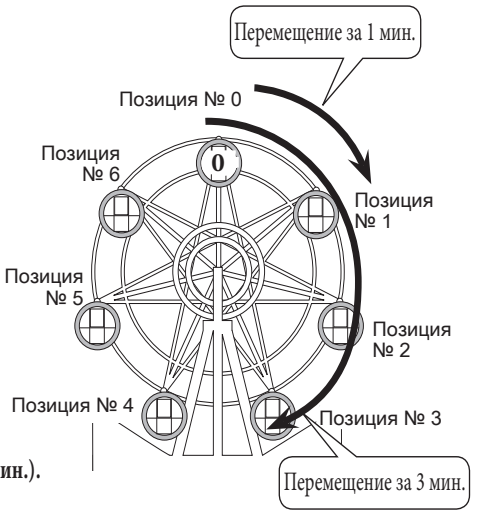


Рис. 3.6. Вращение со скоростью $\frac{1}{7}$ об/мин. (1)

1 (скорость) × 3 (время) = 3 (номер позиции через 3 мин.).

Если же колесо вращается со скоростью 5/7 оборота в минуту, то номер позиции нулевой кабинки через 6 минут после начала вращения вычисляется путём умножения по модулю 7.

$5 \times 6 = 30,$

$30 = 7 \times 4 + 2$ (30 ÷ 7 = 4, остаток 2).

Следовательно,

$5 \times 6 \equiv 2 \pmod 7$.

Другими словами, поделив 30 на 7, мы получим 4 полных оборота и 2 в остатке. Сделанные кабинкой 4 полных оборота в мире модульной арифметики эквивалентны 0 оборотов, поэтому нас интересует только этот остаток 2.

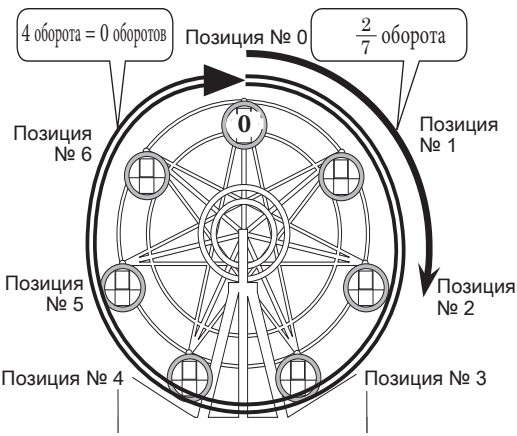


Рис. 3.7. Позиция через 6 мин. вращения со скоростью $\frac{5}{7}$ об/мин.

Для понимания деления по модулю мы применим подход, обратный умножению: будем находить время вращения, используя данные о конечной позиции нулевой кабинки и скорости вращения колеса, которые нам известны.

Пусть в результате вращения колеса со скоростью 1/7 оборота в минуту нулевая кабинка переместилась из начальной позиции № 0 в конечную позицию № 5. Попробуем найти время её вращения.

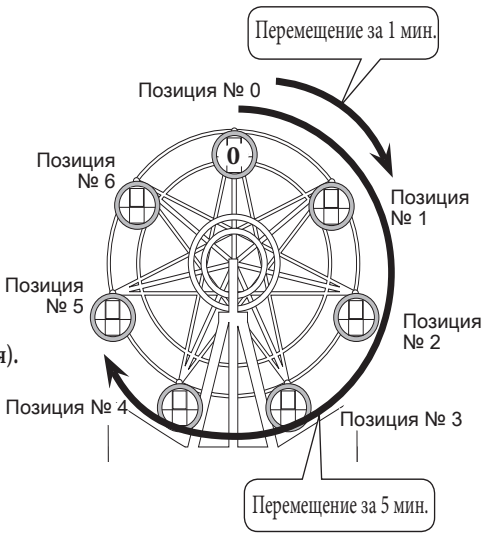


Рис. 3.8. Вращение со скоростью $\frac{1}{7}$ об/мин. (2)

5 (конечная позиция) ÷ 1 (скорость) = 5 (время вращения).

Таким образом, мы выяснили, что колесо вращалось 5 минут. На самом деле, нулевая кабинка будет в той же самой позиции № 5 и в случае, если колесо вращалось, например, 12 или 19 минут, то есть, в общем случае, $(5 + 7n)$ минут, но в мире арифметики по модулю 7 даже для измерения промежутков времени имеется всего 7 значений от 0 до 6 минут, а длительности, подобные 12 или 19 минутам, будут эквивалентны 5 минутам.

Теперь положим, что в результате вращения колеса со скоростью 2/7 оборота в минуту нулевая кабинка переместилась из начальной позиции № 0 в конечную позицию № 5. В этом случае время вращения колеса можно найти в таблице деления по модулю 7.

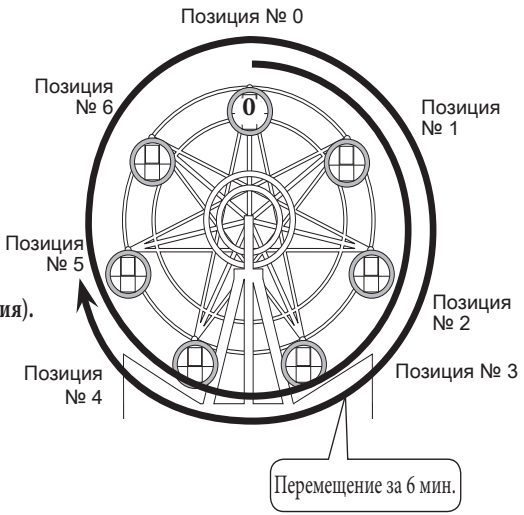
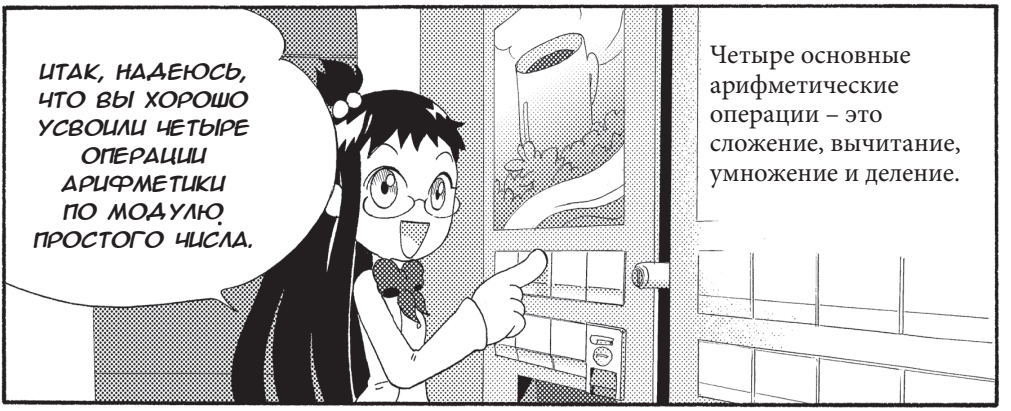


Рис. 3.9. Вращение со скоростью $\frac{2}{7}$ об/мин.

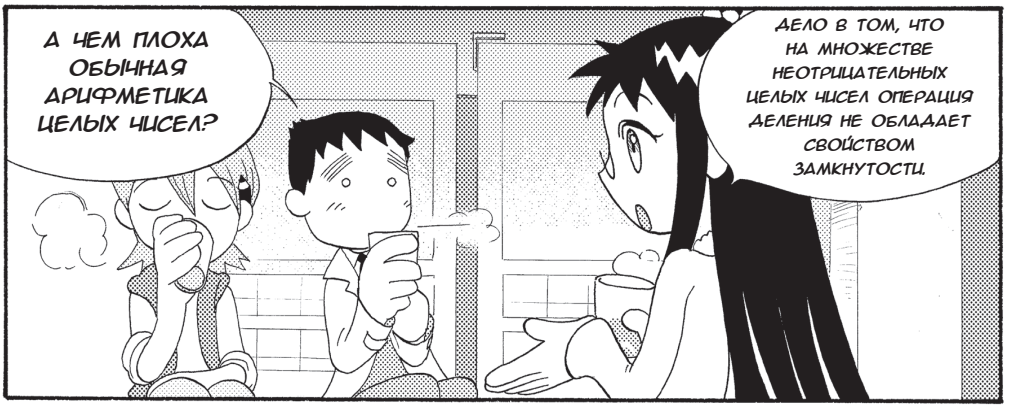
5 (конечная позиция) ÷ 2 (скорость) = 6 (время вращения).

У нас получился результат 6 минут, но как нам его интерпретировать? Рассуждать надо следующим образом: за 6 минут вращения нулевая кабинка пришла в конечную позицию № 5, но для этого ей пришлось сделать один лишний оборот. Другими словами, хотя конечной была на самом деле позиция № 12 ($6 \times 2 = 12$), по причине (mod 7) она была выражена как позиция № 5.

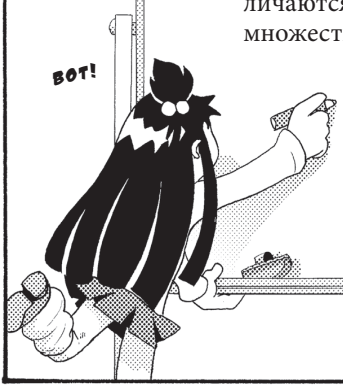
Следовательно, так как $12 \div 2 = 6$, найденный в таблице деления по модулю 7 ответ – «колесо вращалась 6 минут», – оказался абсолютно верным.


ИТАК, НАДЕЮСЬ, ЧТО ВЫ ХОРОШО УСВОИЛИ ЧЕТЫРЕ ОПЕРАЦИИ АРИФМЕТИКИ ПО МОДУЛЮ ПРОСТОГО ЧИСЛА.
Четыре основные арифметические операции – это сложение, вычитание, умножение и деление.


А ЧТО В ЭТОМ ОСОБЕННОГО?
ТАКОЙ БОЛЬШОЙ ВЫБОР ОПЕРАЦИЙ ОЧЕНЬ ПОДХОДИТ ДЛЯ МЕТОДОВ ШИФРОВАНИЯ И РАСШИФРОВАНИЯ.
НЕПОБЕДИМОСТЬ!
КРАСОТА!

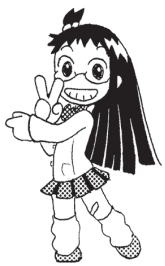

А ЧЕМ ПЛОХА ОБЫЧНАЯ АРИФМЕТИКА ЦЕЛЫХ ЧИСЕЛ?
ДЕЛО В ТОМ, ЧТО НА МНОЖЕСТВЕ НЕОТРИЦАТЕЛЬНЫХ ЦЕЛЫХ ЧИСЕЛ ОПЕРАЦИЯ ДЕЛЕНИЯ НЕ ОБЛАДАЕТ СВОЙСТВОМ ЗАМКНУТОСТИ.

Например, результат операции 3 ÷ 8 не принадлежит множеству натуральных чисел, так как является дробным числом.
С другой стороны, операции арифметики по модулю простого числа p, обладая такими же свойствами коммутативности, ассоциативности, дистрибутивности, отличаются тем, что их результат всегда принадлежит множеству чисел $\{0, 1,... p - 1\}$.



КОММУТАТИВНОСТЬ - ЭТО
$a + b = b + a$ ИЛИ $ab = ba$,
АССОЦИАТИВНОСТЬ - ЭТО
$(a + b) + c = a + (b + c)$ ИЛИ $(ab)c = a(bc)$,
А ДИСТРИБУТИВНОСТЬ - ЭТО
$a(b + c) = ab + bc$!

Типичным примером является так называемое»конечное поле».
Множество рациональных чисел содержит бесконечно большое количество элементов.
С другой стороны, множество, на котором выполнимы операции арифметики по модулю простого числа p, содержит только p элементов: 0, 1,... $p - 1$.
Таким образом, оно имеет конечный размер и поэтому называется «конечным полем».



※ Этот иероглиф 体 читается «тай» и в общей алгебре означает «поле».
Морской карась нарисован потому, что по-японски он тоже называется «тай».

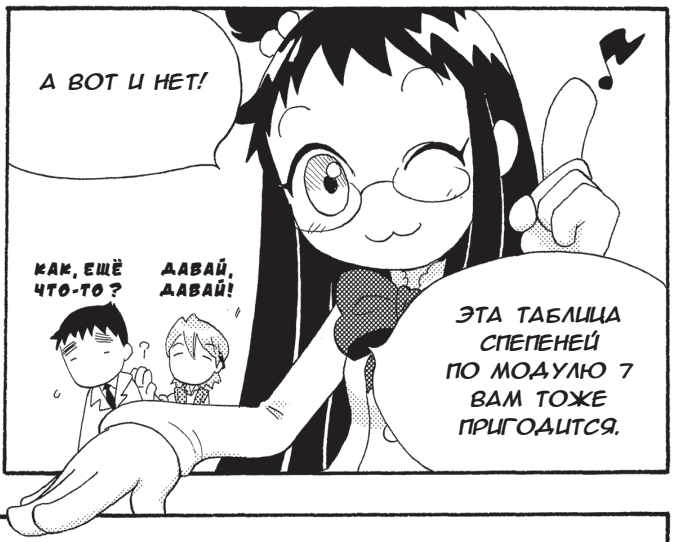
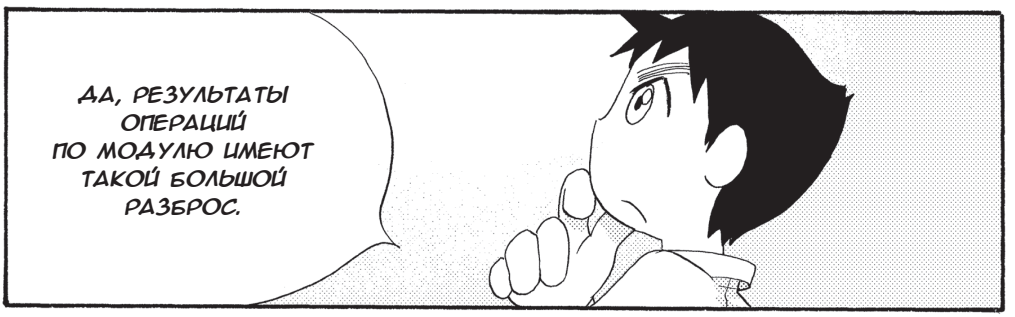


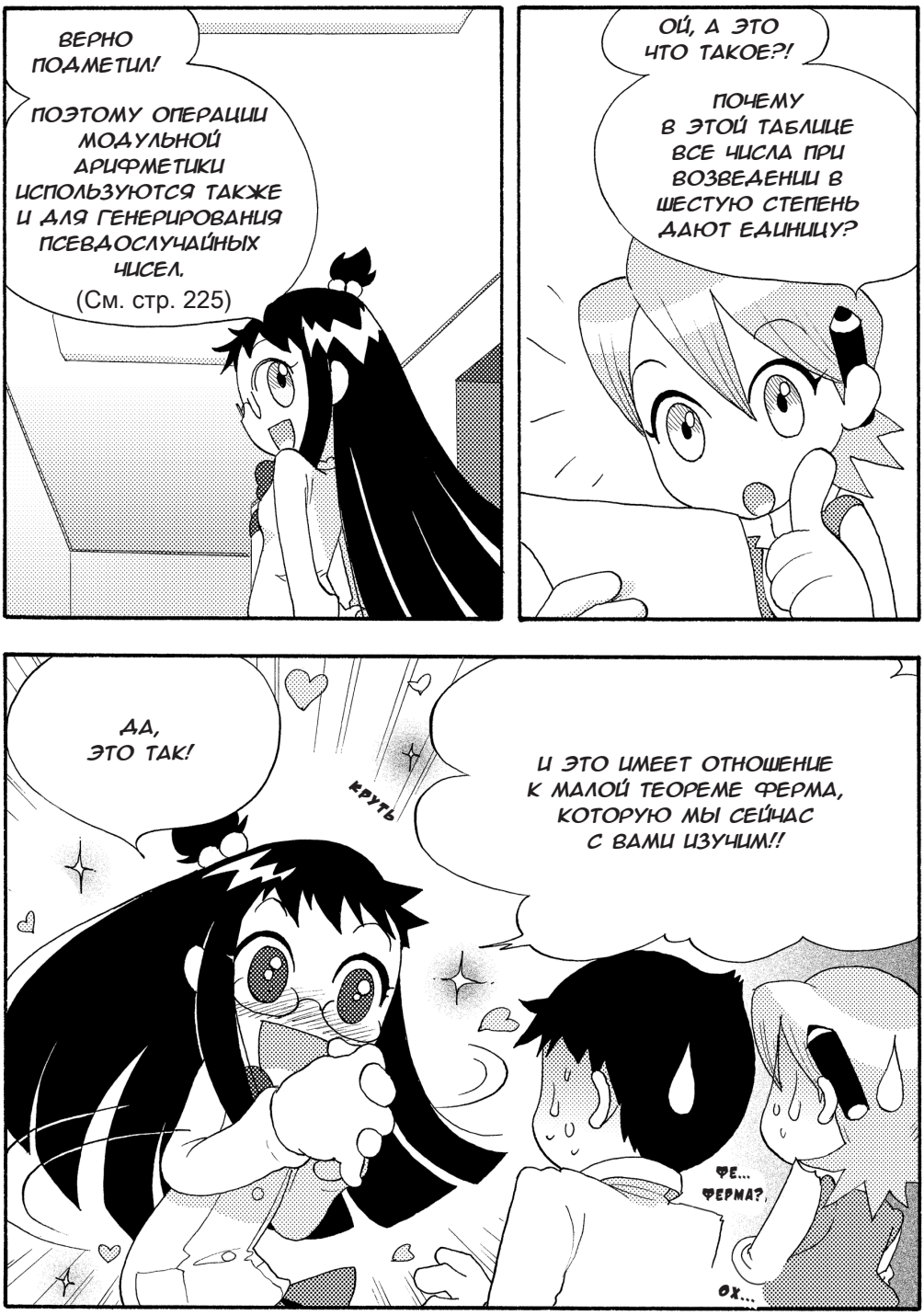
Таблица 3.9. Возведение в степень (a^b) по модулю 7

a \ b	1	2	3	4	5	6
1	1	1	1	1	1	1
2	2	4	1	2	4	1
3	3	2	6	4	5	1
4	4	2	1	4	2	1
5	5	4	6	2	3	1
6	6	1	6	1	6	1

Остатки от деления на 7 шестой степени любого из чисел в таблице равны 1.

$$
\begin{aligned}
1^6 &= 1 = 0 \times 7 + 1 \\
2^6 &= 64 = 9 \times 7 + 1 \\
3^6 &= 729 = 104 \times 7 + 1 \\
4^6 &= 4096 = 585 \times 7 + 1 \\
5^6 &= 15625 = 2232 \times 7 + 1 \\
6^6 &= 46656 = 6665 \times 7 + 1
\end{aligned}
$$




ВЕРНО ПОДМЕТИЛ!
ПОЭТОМУ ОПЕРАЦИИ МОДУЛЬНОЙ АРИФМЕТИКИ ИСПОЛЬЗУЮТСЯ ТАКЖЕ И ДЛЯ ГЕНЕРИРОВАНИЯ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ.
(См. стр. 225)
ОЙ, А ЭТО ЧТО ТАКОЕ?!
ПОЧЕМУ В ЭТОЙ ТАБЛИЦЕ ВСЕ ЧИСЛА ПРИ ВОЗВЕДЕНИИ В ШЕСТУЮ СТЕПЕНЬ ДАЮТ ЕДИНИЦУ?
ДА, ЭТО ТАК!
КРУТЬ
И ЭТО ИМЕЕТ ОТНОШЕНИЕ К МАЛОЙ ТЕОРЕМЕ ФЕРМА, КОТОРУЮ МЫ СЕЙЧАС С ВАМИ ИЗУЧИМ!!
ФЕ... ФЕРМА?
ОХ...

3-4 Малая теорема Ферма и теорема Эйлера

Малая теорема Ферма

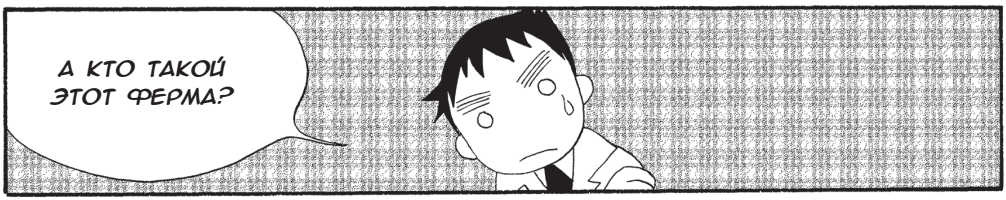
Если n – простое число, то для любого целого числа a, которое является взаимно простым с n (другими словами, не является кратным числу n), верно следующее сравнение:

$$a^{n-1} \equiv 1 \pmod{n}$$

Это означает, что остаток от деления числа a в степени $n - 1$ на число n будет равен 1.

Таблица 3.10. Возведение в степень (a^b) по модулю 7

a \ b	1	2	3	4	5	6
1	1	1	1	1	1	1
2	2	4	1	2	4	1
3	3	2	6	4	5	1
4	4	2	1	4	2	1
5	5	4	6	2	3	1
6	6	1	6	1	6	1



Ферма – отец теории чисел

Пьер де Ферма (1601–1665 гг.) – математик и юрист XVII века, внесший огромный вклад в теорию сравнений и теорию чисел.

Кроме малой теоремы Ферма, существует и великая теорема Ферма.

Она заключается в том, что для любого натурального $n > 2$ уравнение $x^n + y^n = z^n$ не имеет решений в виде натуральных чисел (x, y, z), однако сам Ферма не оставил доказательств этой теоремы.

Сама теорема очень лаконична и выглядит настолько простой, что кажется, что её мог бы доказать даже ученик средней школы. Как известно, теорема Пифагора гласит, что длины a, b и c трёх сторон прямоугольного треугольника (которые могут быть равны, например, 3, 4 и 5 метрам соответственно) удовлетворяют соотношению $a^2 + b^2 = c^2$. Великая теорема Ферма представляет собой уравнение, в котором равные двум показатели степеней уравнения $a^2 + b^2 = c^2$ заменены на $n > 2$.

Великая теорема Ферма была доказана в 1995 году, спустя 330 лет после смерти Ферма, английским математиком Эндрю Уайлсом (1953 г. р.).

Считается, что Ферма написал на полях такое примечание к своей великой теореме.

Выразим контрапозицию к малой теореме Ферма: если существует число a, взаимно простое по отношению к числу n, такое, что

$$a^{n-1} \not\equiv 1 \pmod{n},$$

то число n не является простым.

Если верно высказывание: «*любая манга увлекательна*»,

то можно утверждать, что и его контрапозиция:

«*если книга скучна, то это не манга*», –

тоже является верной.

✤ Тест Ферма и псевдопростые числа

При оценке простоты чисел с помощью теста Ферма нужно помнить о том, что верность сравнения

$$a^{n-1} \equiv 1 \pmod{n}$$

является необходимым, но не достаточным условием простоты числа n.

По этой причине возможны случаи, в которых число, которое не является простым, вероятностно оценивается как простое.

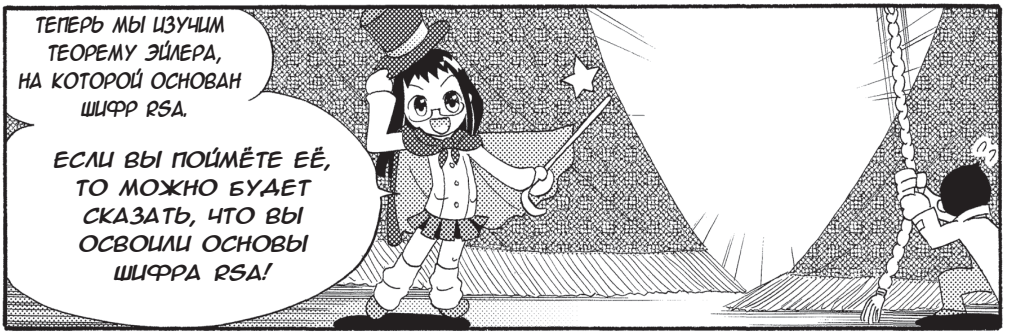
Например, хотя число $n = 3215031751$ и взаимно простые к нему числа 2, 3, 5 и 7 удовлетворяют сравнениям

$$2^{3215031750} \equiv 1 \pmod{3215031751},$$
$$3^{3215031750} \equiv 1 \pmod{3215031751},$$
$$5^{3215031750} \equiv 1 \pmod{3215031751},$$
$$7^{3215031750} \equiv 1 \pmod{3215031751},$$

простым числом оно не является, так как его можно разложить на простые множители следующим образом:

$$3215031751 = 151 \times 751 \times 28351.$$

Правда, в диапазоне натуральных чисел, не превышающих 25 миллиардов, $n = 3215031751$ является единственным составным числом, для которого остатки от деления четырёх простых чисел 2, 3, 5, 7 в степени $n - 1$ на n равны 1. На основе теста Ферма был создан также тест Миллера-Рабина, упоминавшийся на стр. 131, который позволяет повысить точность оценки.



✤ Теорема Эйлера

Для любого натурального числа n и взаимно простого с ним целого числа a верно следующее сравнение:

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Присутствующая в данном сравнении функция $\varphi(n)$ называется функцией Эйлера, значение которой равно количеству натуральных чисел в диапазоне от 1 до n, которые являются взаимно простыми по отношению к числу n.

Кроме того, так как $a \equiv a \pmod{n}$, перемножив левые и правые части:
$a^{\varphi(n)} \times a \equiv 1 \times a \pmod{n}$, получим, что
$a^{\varphi(n)+1} \equiv a \pmod{n}$.

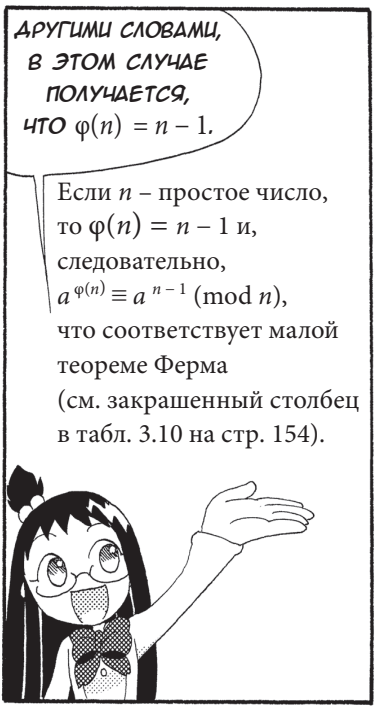
При дальнейшем повышении степени:
$a^{\varphi(n)} \times a^{\varphi(n)} \equiv 1 \times 1 \pmod{n}$, то есть
$a^{2\varphi(n)} \equiv 1 \pmod{n}$,
$a^{2\varphi(n)+1} \equiv a \pmod{n}$ и так далее.

Обобщая вышеизложенное, можно записать, что для любого натурального числа n и взаимно простого с ним целого числа a верны следующие два сравнения:

$a^{k\varphi(n)} \equiv 1 \pmod{n}$,
$a^{k\varphi(n)+1} \equiv a \pmod{n}$, где k – неотрицательное целое число.

Кроме того, в том случае, если натуральное число n можно представить в виде произведения отличных друг от друга простых чисел, то для всех целых чисел a от 1 до $n - 1$ верно следующее сравнение.

$$a^{k\varphi(n)+1} \equiv a \pmod{n} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$



✤ Математик Эйлер

Леонард Эйлер (1707–1783 гг.) – один из самых выдающихся математиков XVIII века, родившийся в Швейцарии. Он известен не только огромными достижениями в разнообразных областях математической науки, но также и активной работой в области физики, астрономии.

Среди его широко известных математических открытий есть так называемая формула Эйлера (формула Эйлера для комплексных чисел):

$$e^{i\theta} = \cos\theta + i\sin\theta$$

Она связывает комплексную экспоненту $e^{i\theta}$ с тригонометрическими функциями $\cos\theta$ и $\sin\theta$ посредством мнимой единицы $i = \sqrt{-1}$.

✤ Функция Эйлера от произведения двух простых чисел

Пусть число N можно разложить на два простых множителя: p и q. Здесь для вывода функции Эйлера мы будем подсчитывать количество целых чисел от 1 до N, которые не являются взаимно простыми с N. Очевидно, что таковыми являются только числа, кратные p, и числа, кратные q.

(1) Числа от 1 до pq, кратные p: p, $2p$, $3p$,..., qp – всего q штук.
(2) Числа от 1 до pq, кратные q: q, $2q$, $3q$,..., pq – всего p штук.
(3) Оба последних числа рядов (1) и (2), qp и pq, совпадают с N.

Чтобы найти $\varphi(N)$, нужно сначала из общего количества чисел от 1 до N, равного N ($= pq$) штук, вычесть количества чисел в рядах (1) и (2), а затем прибавить единицу к результату, для того чтобы учесть пару совпадающих чисел в рядах (1) и (2).

$$\varphi(N) = pq - p - q + 1 = (p - 1)(q - 1).$$

Таким образом, функцию Эйлера в нашем случае можно выразить как $(p - 1)(q - 1)$.

Кроме того, если p и q – простые числа, то $\varphi(p) = p - 1$ и $\varphi(p) = q - 1$, следовательно, $\varphi(pq) = \varphi(p)\,\varphi(q)$.

Кроме того, с учётом того, что $a^{p-1} \equiv 1 \pmod{p}$ и $a^{q-1} \equiv 1 \pmod{q}$, обозначив L наименьшее общее кратное чисел $(p - 1)$ и $(q - 1)$, мы можем записать следующее сравнение:

$$a^L \equiv 1 \pmod{p, q}.$$

Следовательно, для числа a, взаимно простого с N, верно следующее сравнение:

$$a^L \equiv 1 \pmod{N}.$$

Другими словами, в том случае, если N можно разложить на простые множители p и q, наименьшее общее кратное L чисел $(p - 1)$ и $(q - 1)$ может выполнять ту же самую роль, что и функция Эйлера $\varphi(N)$.

Далее, так как произведение двух целых чисел можно представить в виде произведения их наименьшего общего кратного (НОК) и наибольшего общего делителя (НОД):

$$(p-1)(q-1) = LG,$$

$$L = \frac{(p-1)(q-1)}{G}$$, где L – наименьшее общее кратное,

G – наибольший общий делитель чисел $(p-1)$ и $(q-1)$.

Перейдём к рассмотрению конкретного примера. Положим, что $p = 3$ и $q = 5$, тогда $N = 15$; $(p-1) = 2$; $(q-1) = 4$; $\varphi(N) = (p-1)(q-1) = 8$;
L = НОК(2, 4) = 4;   G = НОД(2, 4) = 2.

Аналогично тому, как было показано на стр. 158 для функции Эйлера, из сравнения $a^L \equiv 1 \pmod N$ следует обобщённая формула:

$a^{kL} \equiv 1 \pmod N$, где a – взамно простое с N число; $k = 0, 1, 2,...$

Возвращаясь к рассматриваемому случаю, для любого натурального числа a, взаимно простого с 15, будет верно сравнение:

$$a^{4k} \equiv 1 \pmod{15}.$$

Кроме того, заменив функцию $\varphi(N)$ в показателе степени сравнения (1) (см. стр. 158) на L, получим, что если $N = pq$, где p и q – простые числа, то для любого целого числа a от 1 до $(N-1)$ верно следующее сравнение:

$$a^{kL+1} \equiv a \pmod N \qquad (2)$$

В рассматриваемом случае

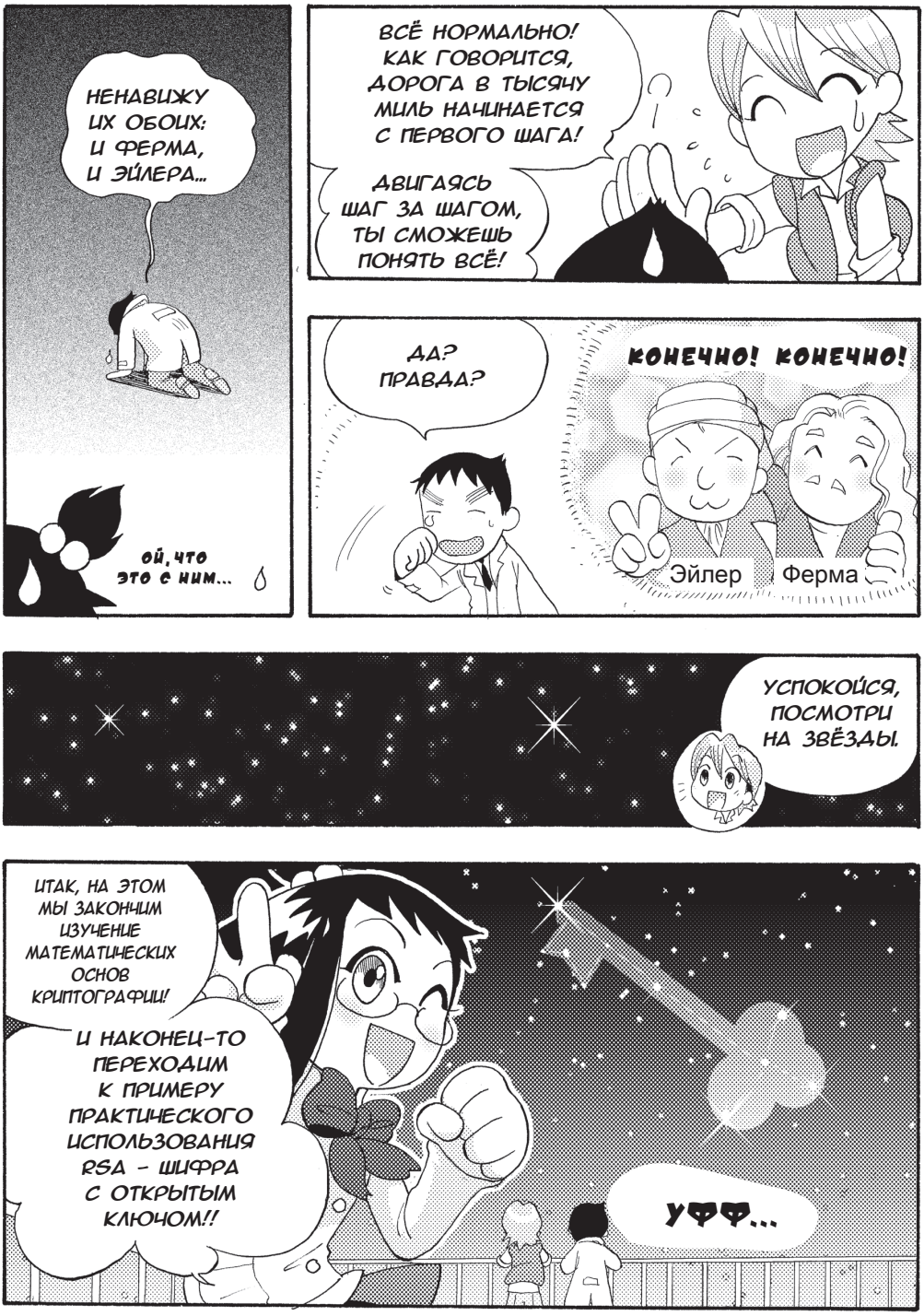
$$a^{4k+1} \equiv a \pmod{15}.$$

В верности данного сравнения можно убедиться с помощью табл. 3.11.

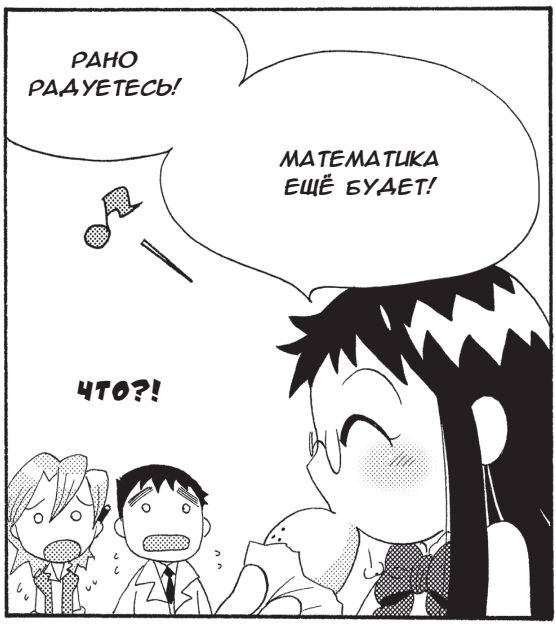
Таблица 3.11. Возведение в степень (a^b) по модулю 15
(для иллюстрации случая $N = 3 \times 5$, $\varphi(N) = 8$, $L = 4$, $G = 2$)

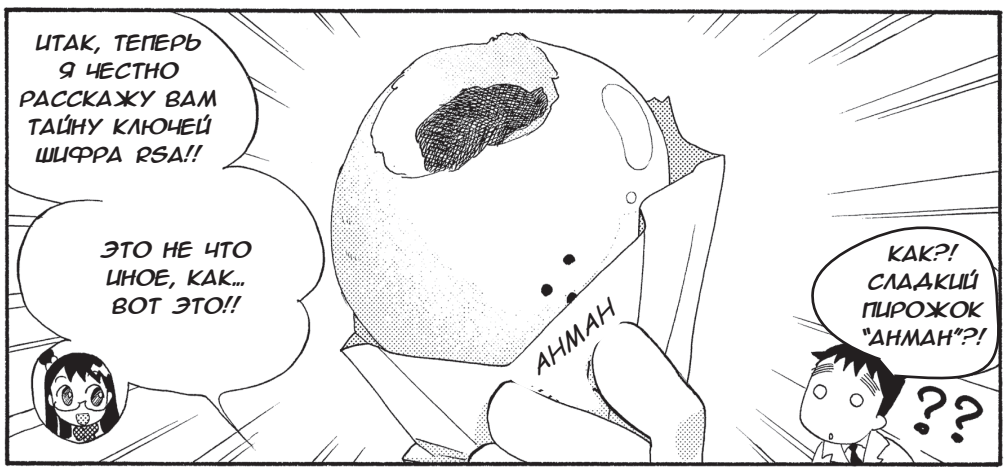
a \ b	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	2	4	8	1	2	4	8	1
3	3	9	12	6	3	9	12	6
4	4	1	4	1	4	1	4	1
5	5	10	5	10	5	10	5	10
6	6	6	6	6	6	6	6	6
7	7	4	13	1	7	4	13	1
8	8	4	2	1	8	4	2	1
9	9	6	9	6	9	6	9	6
10	10	10	10	10	10	10	10	10
11	11	1	11	1	11	1	11	1
12	12	9	3	6	12	9	3	6
13	13	4	7	1	13	4	7	1
14	14	1	14	1	14	1	14	1
	Первый период				Второй период			

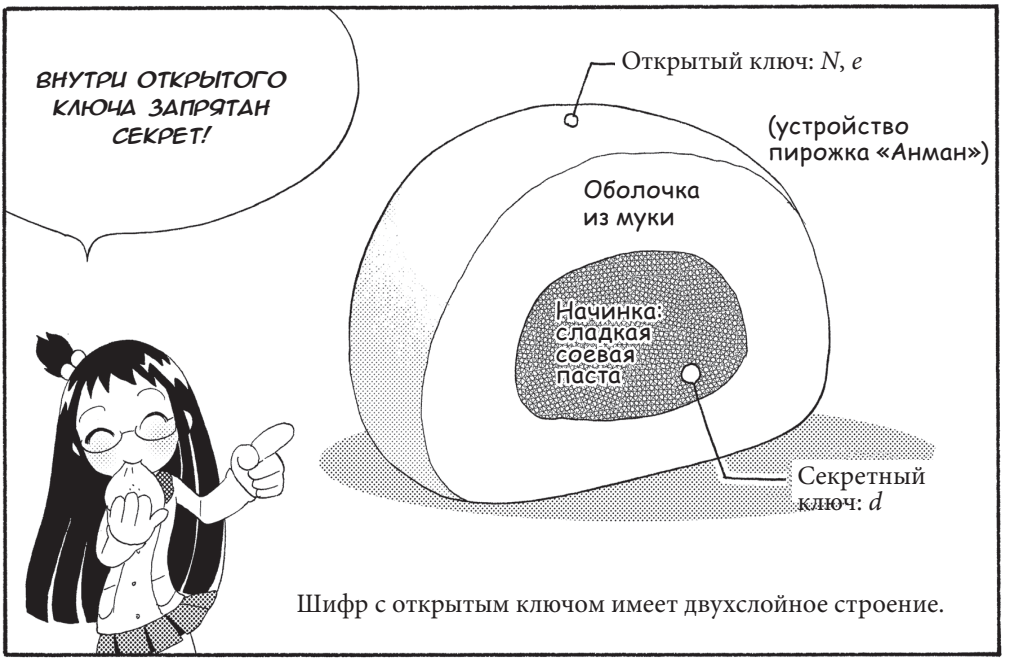
Закрашенные серым столбцы, соответствующие таким показателям степени (b), при которых степень сравнима по модулю с самим числом a, повторяются с перидом $L = 4$, что находится в соответствии со сравнением (2). В диапазоне значений показателей степени от 1 до $\varphi(N) = 8$ эти столбцы повторяются $G = 2$ раза.


ВСЁ НОРМАЛЬНО! КАК ГОВОРИТСЯ, ДОРОГА В ТЫСЯЧУ МИЛЬ НАЧИНАЕТСЯ С ПЕРВОГО ШАГА!
ДВИГАЯСЬ ШАГ ЗА ШАГОМ, ТЫ СМОЖЕШЬ ПОНЯТЬ ВСЁ!
НЕНАВИЖУ ИХ ОБОИХ: И ФЕРМА, И ЭЙЛЕРА...
ОЙ, ЧТО ЭТО С НИМ...
ДА? ПРАВДА?
КОНЕЧНО! КОНЕЧНО!
Эйлер
Ферма
УСПОКОЙСЯ, ПОСМОТРИ НА ЗВЁЗДЫ.
ИТАК, НА ЭТОМ МЫ ЗАКОНЧИМ ИЗУЧЕНИЕ МАТЕМАТИЧЕСКИХ ОСНОВ КРИПТОГРАФИИ!
И НАКОНЕЦ-ТО ПЕРЕХОДИМ К ПРИМЕРУ ПРАКТИЧЕСКОГО ИСПОЛЬЗОВАНИЯ RSA – ШИФРА С ОТКРЫТЫМ КЛЮЧОМ!!
УФФ...

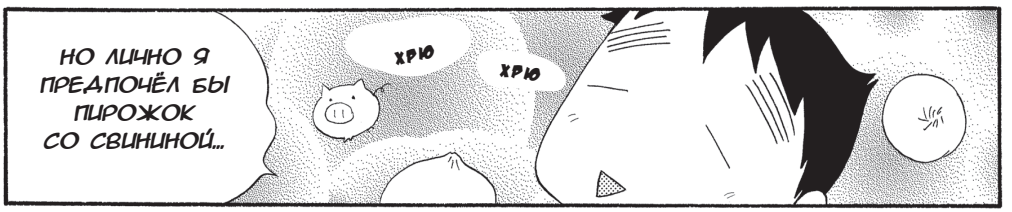
3-5 Устройство шифра RSA







Шифр с открытым ключом имеет двухслойное строение.



✤ Шифрование и расшифрование RSA

Обозначив P открытый текст и C – шифртекст, можно выразить процесс шифрования в виде следующей формулы:

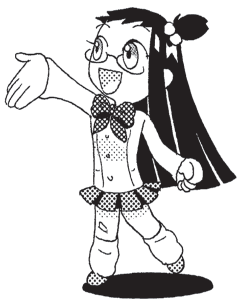
$$C \equiv P^e \pmod{N}.$$

Таким образом, результат возведения P (открытого текста) в степень e (составляющая открытого ключа) делится на N (другая составляющяя открытого ключа), и остатком от этого деления будет C (шифртекст).

Далее, процесс расшифрования можно выразить следующей формулой:

$$P \equiv C^d \pmod{N}.$$

Таким образом, результат возведения C (шифртекста) в степень d (секретный ключ) делится на N (открытый ключ), и остатком от этого деления будет P (открытый текст). Здесь N – это число, полученное перемножением двух отличных друг от друга больших простых чисел.




ВЕДЬ, ЗНАЯ e, C И N, МОЖНО БЫЛО БЫ РЕШИТЬ СРАВНЕНИЕ $x^e \equiv C \pmod N$ И ПРОЧИТАТЬ ШИФРТЕКСТ, НЕ ТАК ЛИ?
ОДНАКО ЕСЛИ ИСКАТЬ x МЕТОДОМ ПЕРЕБОРА, ПОДСТАВЛЯЯ ЗНАЧЕНИЯ ПО ОДНОМУ, ТО НА ЭТО ПОТРЕБУЕТСЯ ТАК МНОГО ВРЕМЕНИ, ЧТО СДЕЛАТЬ ЭТО БУДЕТ ПРАКТИЧЕСКИ НЕВОЗМОЖНО!
ТАКОЕ СВОЙСТВО НАЗЫВАЕТСЯ "ВЫЧИСЛИТЕЛЬНОЙ КРИПТОСТОЙКОСТЬЮ".


КСТАТИ, ЕСЛИ БЫ БЫЛО ИЗВЕСТНО ЗНАЧЕНИЕ ФУНКЦИИ ЭЙЛЕРА $\varphi(N)$, ТО ПРОЧИТАТЬ ШИФРТЕКСТ МОЖНО БЫЛО БЫ, ИСПОЛЬЗУЯ ТЕОРЕМУ ЭЙЛЕРА!
НО ДЛЯ ТОГО, ЧТОБЫ ВЫЧИСЛИТЬ ФУНКЦИЮ $\varphi(N)$, НУЖНО РАЗЛОЖИТЬ ЧИСЛО N НА ПРОСТЫЕ МНОЖИТЕЛИ, НЕ ТАК ЛИ?
ТИСК


ЕСЛИ N – ОЧЕНЬ БОЛЬШОЕ ЧИСЛО, ТО ДЛЯ РАЗЛОЖЕНИЯ ЕГО НА ПРОСТЫЕ МНОЖИТЕЛИ ПОНАДОБИТСЯ ОЧЕНЬ МНОГО ВРЕМЕНИ. ДРУГИМИ СЛОВАМИ, БЕЗОПАСНОСТЬ ЭТОГО ШИФРА ЗАВИСИТ ОТ СЛОЖНОСТИ ЗАДАЧИ ФАКТОРИЗАЦИИ ЦЕЛЫХ ЧИСЕЛ.
СЛУШАТЬ ВНИМАТЕЛЬНО!


ДРУГИМИ СЛОВАМИ, МЫ ПРИХОДИМ К МАТЕМАТИЧЕСКИ СЛОЖНОЙ ЗАДАЧЕ НА ФАКТОРИЗАЦИЮ ЦЕЛЫХ ЧИСЕЛ. ИМЕННО ОНА ДЕЛАЕТ ШИФР RSA СТОЙКИМ!!


В ШИФРЕ С ОТКРЫТЫМ КЛЮЧОМ БОЛЬШОЕ ВНИМАНИЕ УДЕЛЯЕТСЯ КЛЮЧАМ ШИФРОВАНИЯ И РАСШИФРОВАНИЯ, ПОЭТОМУ СЕЙЧАС МЫ ПО ПОРЯДКУ ИЗУЧИМ ПРОЦЕСС ГЕНЕРИРОВАНИЯ КЛЮЧЕЙ RSA!
БЕДОЛАГА...

✾ Метод генерирования ключей RSA

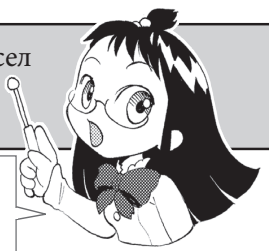
① Выбираем произвольно два достаточно больших простых числа p и q.

Произведение $p \times q$, другими словами, N, будет одной из составляющих открытого ключа.

② Вычисляем значение функции Эйлера $\varphi(pq) = (p - 1)(q - 1)$.

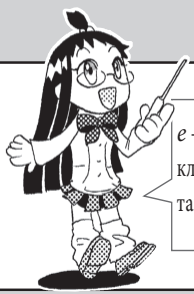
③ – это ещё одна подготовительная операция для генерирования открытого ключа.

③ Находим число L – наименьшее общее кратное чисел $(p - 1)$ и $(q - 1)$.



После вычисления функции Эйлера числа p и q нам уже не понадобятся! Их лучше уничтожить, чтобы кто-нибудь их случайно не узнал.

④ Выбираем произвольно положительное целое число e, так чтобы оно было немного меньше L и являлось взаимно простым с L.



e – это один из открытых ключей! Выберите его так, чтобы $P^e > N$!!

В случае если $P^e \leq N$, открытый текст преобразуется в шифртекст, минуя вычисления по модулю:
$P^e = C$, что сделает невозможным шифрование путём вычислений.

⑤ Выбираем произвольно положительное целое число d так, чтобы оно удовлетворяло следующему сравнению.
$ed \equiv 1 \pmod{L}$.
Это число d, мы выбираем его так, чтобы оно было меньше $\varphi(N)$, но больше чисел p и q.

Теперь давайте найдём ключ расшифрования d, образующий пару с ключом шифрования e. Из $ed \equiv 1 \pmod{L}$ следует, что $ed - 1 \equiv 0 \pmod{L}$, другими словами, что число $(ed - 1)$ кратно числу L.

$ed - 1 = kL$, где k – неотрицательное целое число.

Следовательно,

$ed = kL + 1$, где k – неотрицательное целое число.

Следовательно, согласно сравнению (2), приведённому в рассказе о функции Эйлера (см. стр. 161), для любого натурального числа P от 1 до $(N - 1)$ верно следующее сравнение.

$P^{ed} = P^{kL+1} \equiv P \pmod{N}$.

Это сравнение позволяет нам понять, что при возведении шифртекста C $(= P^e)$ в степень d (P^{ed}) произойдёт восстановление открытого текста из шифртекста (то есть расшифрование).

❀ Генерирование открытого и секретного ключей

Здесь мы попробуем найти открытый ключ N и секретный ключ d для двух простых чисел $p = 5$ и $q = 11$.

Шаг 1

Число N равно произведению чисел p и q.

$$N = pq = 5 \times 11 = 55$$

Шаг 2

Вычисляем значение функции Эйлера $\varphi(N)$ для найденного N.

$$\varphi(55) = (5 - 1) \times (11 - 1) = 4 \times 10 = 40$$

Шаг 3

Находим наименьшее общее кратное L чисел $(p - 1)$ и $(q - 1)$.
Наименьшее общее кратное чисел 4 и 10: $L = 20$.

Шаг 4

Выбираем натуральное число e от 1 до $(L - 1)$ так, чтобы оно было взаимно простым с наименьшим общим кратным L. Для $L = 20$ это может быть одно из следующих восьми чисел: {1, 3, 7, 9, 11, 13, 17, 19}.

Шаг 5

Находим обратный элемент d к ключу шифрования e, другими словами, ключ расшифрования d. Подумаем над тем, каким будет обратный элемент к $e = 17$ относительно операции умножения по модулю 20.

Необходимым и достаточным условием верности сравнения $ed = 1 \pmod{20}$ является выполнение равенства $ed = 20k + 1$, где k – неотрицательное целое число.
Решая уравнение относительно d, получим:

$$d = \frac{(20k + 1)}{17}$$

Так как правая часть уравнения является целым числом, нужно подобрать такое k, чтобы число $(20k + 1)$ было кратно 17. Последовательно перебирая k,

обнаруживаем, что при $k = 11$ числитель дроби $(20k + 1)$ равен числу 221, которое делится на 17 без остатка. Другими словами,

$$221 = 20 \times 11 + 1 = 17 \times 13$$
$$17 \times 13 \equiv 1 \pmod{20}$$

Таким образом, мы получили $d = 13$. Итак, в результате вышеприведённых вычислений мы получили следующие ключи.

Открытый ключ ($N = 55$, $e = 17$) ← Ключ шифрования
Закрытый ключ ($d = 13$) ← Ключ расшифрования

Подобрав числа d для всех возможных чисел e, полученных на шаге 4, можно получить следующие пары чисел, удовлетворяющих уравнению $ed = 20k + 1$, где k – неотрицательное целое число:

$$(e = 1,\ d = 1)、(e = 3,\ d = 7)、(e = 7,\ d = 3)、(e = 9,\ d = 9)、$$
$$(e = 11,\ d = 11)、(e = 17,\ d = 13)、(e = 19,\ d = 19)$$

Учитывая то, что в качестве ключей расшифрования и шифрования обычно выбирают отличные друг от друга целые числа ($e \neq d$), а также то, что в качестве ключа шифрования e желательно выбирать как можно большее целое число, здесь мы будем использовать пару $e = 17$, $d = 13$.

❀ Генерирование шифртекста RSA

Прежде всего я объясню порядок генерирования шифртекста с помощью открытого ключа RSA.

В этом примере мы будем зашифровывать открытый текст, состоящий из 4 букв английского алфавита (GOLF), используя ключ шифрования $e = 17$, сгенерированный в предущем примере.

Таблица 3.12. Таблица кодов символов

Буквы	Коды	Буквы	Коды	Буквы	Коды
a	0	s	18	K	36
b	1	t	19	L	37
c	2	u	20	M	38
d	3	v	21	N	39
e	4	w	22	O	40
f	5	x	23	P	41
g	6	y	24	Q	42
h	7	z	25	R	43
i	8	A	26	S	44
j	9	B	27	T	45
k	10	C	28	U	46
l	11	D	29	V	47
m	12	E	30	W	48
n	13	F	31	X	49
o	14	G	32	Y	50
p	15	H	33	Z	51
q	16	I	34	⋮	⋮
r	17	J	35	Пробел	63

Шаг 1

Прежде всего назначим буквам открытого текста целые числа, используя таблицу кодов символов.

G	O	L	F
↓	↓	↓	↓
32	40	37	31

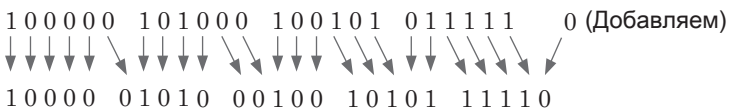
Шаг 2

Преобразуем целые числа в 6-битные двоичные строки.

32	40	37	31
↓	↓	↓	↓
100000	101000	100101	011111

Шаг 3

Представим двоичные данные в виде неотрицательных целых чисел, не превышающих число $(N - 1)$. Так как в нашем примере $N = 55$, $(N - 1) = 54$, мы перегруппируем данные в 5-битные строки: это позволит представить данные в виде целых чисел от 0 до 31, что удовлетворяет вышеуказанному условию. Разумеется, ничто не запрещает разделить данные и на 4-битные, и на 3-битные строки, но чем будет больше длина двоичной строки, тем более эффективен будет процесс шифрования.



(Последний ноль является битом-заполнителем последней 5-битной строки, в которой после перегруппировки не хватало одного бита.)

Шаг 4

Преобразуем двоичные данные в десятичные числа.

10000	01010	00100	10101	11110
↓	↓	↓	↓	↓
16	10	4	21	30

Шаг 5

Зашифровываем десятичные данные, используя ключ шифрования ($N = 55$, $e = 17$): находим остатки от деления на 55 результатов возведения десятичных данных в степень 17, другими словами, мы должны вычислить следующие степени по модулю 55.

$$16^{17} \pmod{55},\quad 10^{17} \pmod{55},\quad 4^{17} \pmod{55},\quad 21^{17} \pmod{55},\quad 30^{17} \pmod{55}$$

Покажем здесь подробно процесс возведения десятичного числа 16 в степень 17 по модулю 55. Так как верны следующие сравнения:

$$16^2 = 256 \equiv 36 \pmod{55} \qquad 36^2 = 1296 \equiv 31 \pmod{55}$$
$$31^2 = 961 \equiv 26 \pmod{55} \qquad 26^2 = 676 \equiv 16 \pmod{55}$$

последовательно применяя их, мы получим следующее.

$$\begin{aligned}
16^{17} &= 16^2 \times 16^2 \times 16^2 \times 16^2 \times 16^2 \times 16^2 \times 16^2 \times 16^2 \times 16 \\
&\equiv 36 \times 36 \times 36 \times 36 \times 36 \times 36 \times 36 \times 36 \times 16 \pmod{55} \\
&= 36^2 \times 36^2 \times 36^2 \times 36^2 \times 16 \\
&\equiv 31 \times 31 \times 31 \times 31 \times 16 \pmod{55} \\
&= 31^2 \times 31^2 \times 16 \\
&\equiv 26 \times 26 \times 16 \pmod{55} \\
&= 26^2 \times 16 \\
&\equiv 16 \times 16 \pmod{55} \\
&\equiv 36 \pmod{55}.
\end{aligned}$$

Шифрование остальных десятичных данных производится аналогичным образом, поэтому здесь мы приведём только результаты вычислений.

$$10^{17} \pmod{55} \equiv 10 \qquad 4^{17} \pmod{55} \equiv 49$$
$$21^{17} \pmod{55} \equiv 21 \qquad 30^{17} \pmod{55} \equiv 35$$

Итак, мы сгенерировали шифртекст, который в виде десятчных чисел выражается следующим образом.

$$36 \quad 10 \quad 49 \quad 21 \quad 35 \quad \cdots\cdots\cdots \quad (3)$$

Рассматривая (3) как последовательность кодов символов, назначим десятичным числам буквы по табл. 3.12.

36	10	49	21	35
↓	↓	↓	↓	↓
K	k	X	v	J

Теперь шифртекст RSA готов.

❀ Расшифрование RSA

Теперь я объясню порядок преобразования шифртекста в открытый текст с помощью секретного ключа RSA.

В этом примере будет продемонстрирован процесс расшифрования десятичных зашифрованных данных (3) до преобразования в открытый текст, состоящий из букв алфавита, с помощью секретного ключа расшифрования ($d = 13$).

Шаг 1

Используя ключ расшифрования ($d = 13$), вычисляем $C^d \pmod N$: находим остатки от деления на 55 результатов возведения десятичных данных (3) в степень 13, другими словами, мы должны вычислить следующие степени по модулю 55:

$$36^{13} \pmod{55},\quad 10^{13} \pmod{55},\quad 49^{13} \pmod{55},\quad 21^{13} \pmod{55},\quad 35^{13} \pmod{55}$$

Вычисления проводятся по тому же алгоритму, как на шаге 5 генерирования шифртекста. Например, число 36 возводится в 13-ю степень по модулю 55 следующим образом:

$$\begin{aligned}
36^{13} &= 36^2 \times 36^2 \times 36^2 \times 36^2 \times 36^2 \times 36^2 \times 36 \\
&\equiv 31 \times 31 \times 31 \times 31 \times 31 \times 31 \times 36 \pmod{55} \\
&\equiv 26 \times 26 \times 26 \times 36 \pmod{55} \\
&= 26^2 \times 26 \times 36 \\
&\equiv 16 \times 26 \times 36 \pmod{55} \\
&= 14976 \\
&\equiv 16 \pmod{55}.
\end{aligned}$$

Расшифрование остальных зашифрованных данных {10, 49, 21, 35} производится аналогичным образом, поэтому здесь мы приведём только результаты вычислений.

$$10^{13} \pmod{55} \equiv 10 \qquad 49^{13} \pmod{55} \equiv 4$$
$$21^{13} \pmod{55} \equiv 21 \qquad 35^{13} \pmod{55} \equiv 30$$

Следовательно, данные открытого текста в виде десятичных чисел записываются следующим образом:

16 10 4 21 30

Шаг 2

Преобразуем полученные десятичные данные открытого текста в 5-битные двоичные числа.

16	10	4	21	30
↓	↓	↓	↓	↓
10000	01010	00100	10101	11110

Шаг 3

Для того чтобы воспользоваться таблицей кодов символов, перегруппируем двоичные данные в 6-битные строки.

1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0

0 (Отбрасываем)

1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 1

(Последний 0 является битом-заполнителем, добавленным во время перегруппировки в 5-битные строки, поэтому отбрасываем его.)

Шаг 4

Преобразуем 6-битные двоичные строки в целые числа.

100000	101000	100101	011111
↓	↓	↓	↓
32	40	37	31

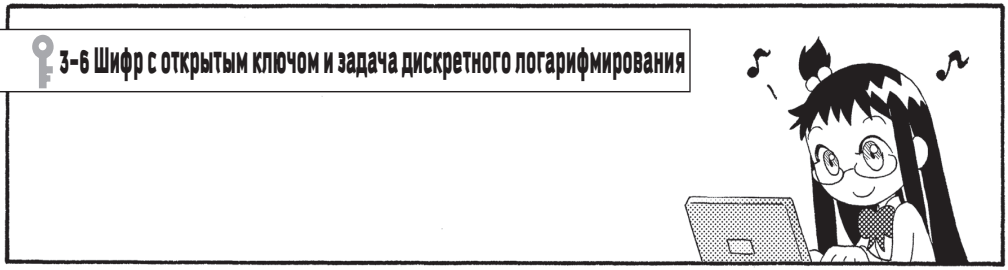
Шаг 5

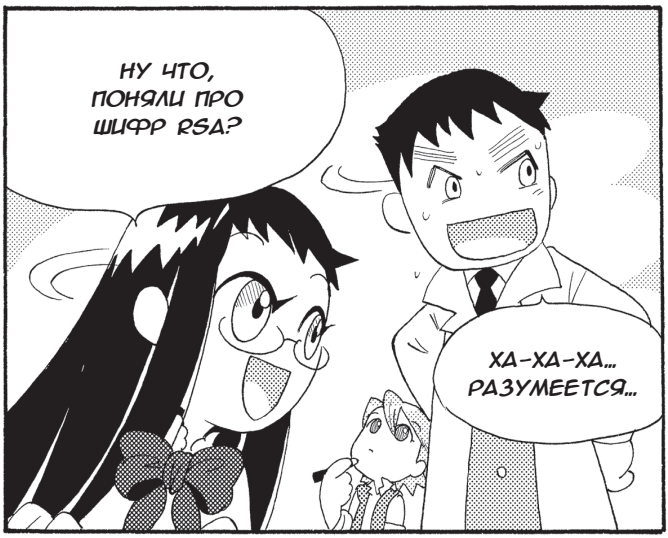
Используя таблицу кодов символов, заменяем целочисленные данные буквами.

32	40	37	31
↓	↓	↓	↓
G	O	L	F

Итак, расшифрование завершено.

3-6 Шифр с открытым ключом и задача дискретного логарифмирования





✤ Задача дискретного логарифмирования

Взгляните ещё раз на таблицу степеней по модулю 7.

В строке, соответствующей основанию 3, содержатся все значения от 1 до 6 без повторений.

Множество результатов арифметических операций по модулю 7 представляет собой конечное поле, состоящее из следующих элементов:

$$\{ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6 \}$$

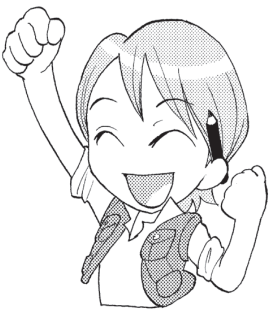
а с помощью степеней числа 3 можно выразить все эти элементы, за исключением 0. Число 3, выражающее все числа в табл. 3.13 от 1 до 6 по одному разу, называется первообразным корнем по модулю 7.

Таблица 3.13. Степени (a^b) по модулю 7

a \ b	1	2	3	4	5	6
1	1	1	1	1	1	1
2	2	4	1	2	4	1
3	3	2	6	4	5	1
4	4	2	1	4	2	1
5	5	4	6	2	3	1
6	6	1	6	1	6	1

Для любого модуля p, являющегося простым числом, обязательно существуют $\varphi(p-1)$ первообразных корней. Например, для простого модуля 7 существуют два первообразных корня.

$$\varphi(7-1) = \varphi(6) = \varphi(2\times 3) = (2-1)\times(3-1) = 2$$



Таким образом, кроме числа 3, для модуля 7 должен существовать ещё один первообразный корень. Изучив табл. 3.13, можно увидеть, что первообразным корнем является также число 5. Обозначив p – простой модуль, α – первообразный корень по модулю p, произвольный элемент Z_i конечного поля можно выразить следующим сравнением:

$\alpha^k \equiv Z_i \pmod p$, где $k \leq p-1$ – неотрицательное целое число.

Кроме того, показатель k степени первообразного корня α можно выразить следующим сравнением:

$k \equiv \log_\alpha Z_i \pmod p$.

В этом случае число k называют дискретным логарифмом по основанию α.

Пусть вас не смущает такой математический термин, как логарифм, так как здесь всё очень просто. Например, такое выражение, как 2^3, имеет точно такой же смысл, как все нижеприведённые выражения.

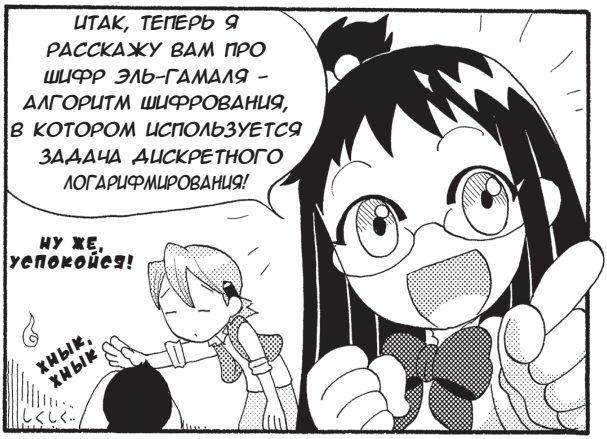
$$3 = 3\log_2 2 = \log_2 2^3 = \log_2 8$$

Ведь, например, фразу «2 в степени 3 равно 8» можно выразить и так: «Чтобы получить 8, надо 3 раза умножить 2 само на себя».

Как упоминалось на стр. 118, используя следующее сравнение

$$\alpha^k \equiv Z_i \pmod p,$$

легко найти Z_i по известным α, k и p, однако найти дискретный логарифм k по известным α, Z_i и p – чрезвычайно сложная задача. Это и называется задачей дискретного логарифмирования.



❀ Шифрование и расшифрование Эль-Гамаля

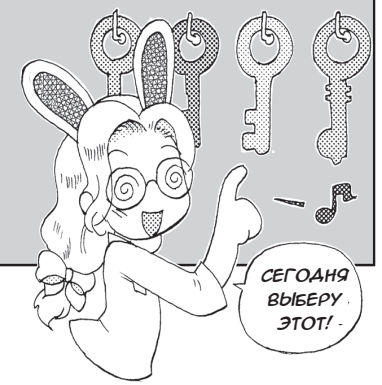
Пусть отправителем шифртекста будет Рика, а получателем – Лана.

① Получатель Лана выбирает большое простое число q и первообразный корень α.

② Получатель Лана произвольно выбирает секретный ключ d и публикует три числа: g, α и q, являющиеся решением сравнения

$$g \equiv \alpha^d \pmod{q},$$

в качестве открытого ключа.



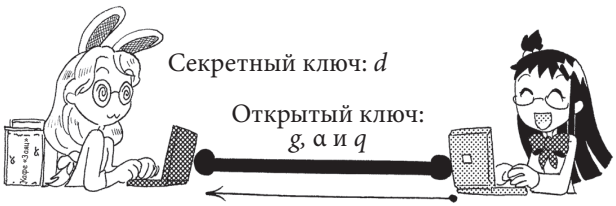
③ Отправитель Рика выбирает случайное число r и вычисляет

$$C_1 \equiv \alpha^r \pmod{q}.$$

Кроме того, используя открытый текст P, она вычисляет также и

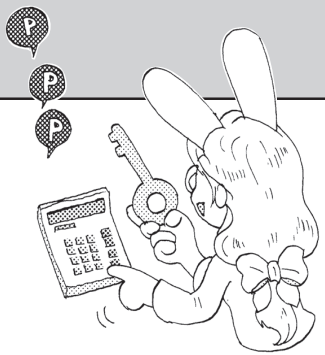
$$C_2 \equiv P \times g^r \pmod{q}.$$

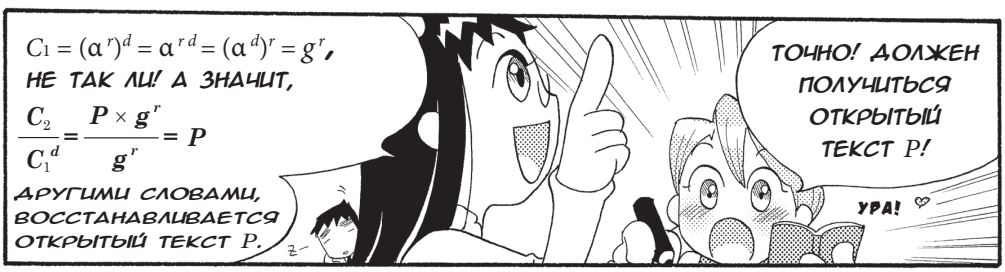
④ Отправитель Рика отправляет C_1 и C_2 получателю Лане.

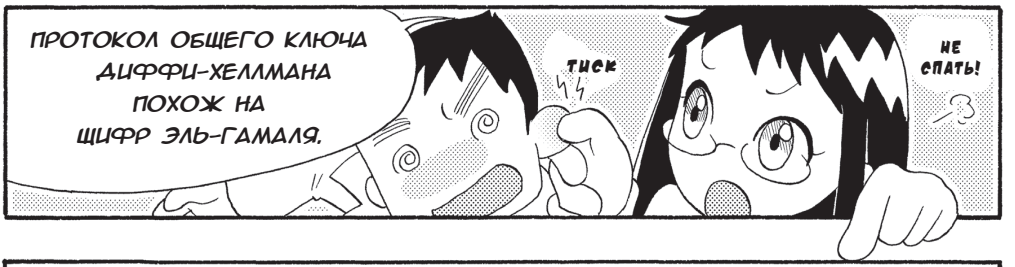


⑤ Используя секретный ключ d, получатель Лана расшифровывает шифртекст путём вычисления следующего сравнения:

$$P \equiv \frac{C_2}{C_1{}^d} \pmod{q}$$



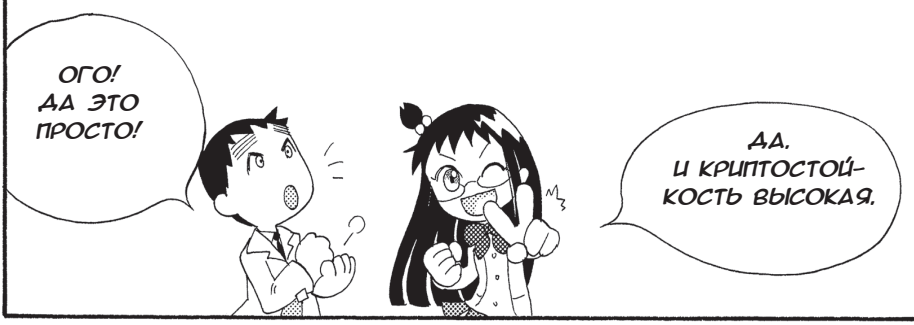


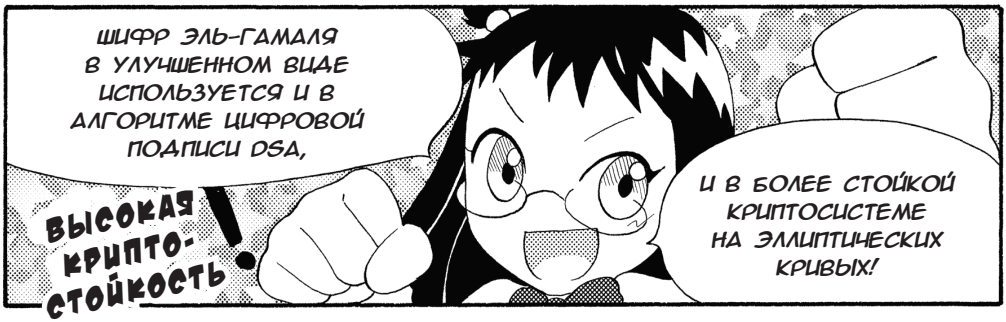


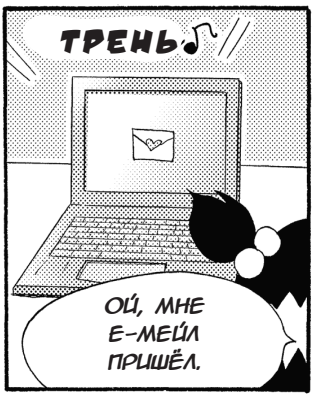
① Рика и Лана совместно обладают большим простым числом p и первообразным корнем α, которые не являются секретными.

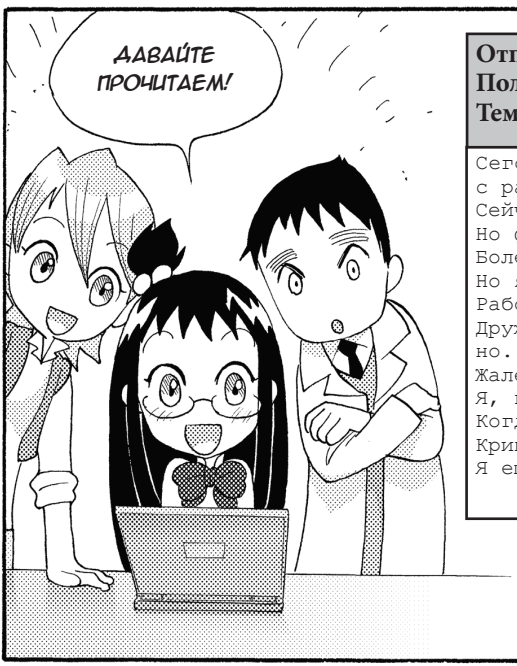
② Рика выбирает случайное число c и, сохраняя его в секрете, отправляет Лане результат вычисления $\alpha^c \pmod{p}$.
Лана выбирает случайное число d и, сохраняя его в секрете, отправляет Рике результат вычисления $\alpha^d \pmod{p}$.

③ Рика, используя свой секретный ключ c, получает ключ $(\alpha^d)^c \equiv \alpha^{cd} \pmod{p}$.
Лана, используя свой секретный ключ d, получает ключ $(\alpha^c)^d \equiv \alpha^{cd} \pmod{p}$.
Теперь у обеих девушек есть общий ключ.









Отправитель: Удзуки Лана
Получатель: Рика
Тема: До свиданья!

Сегодня ты опять преподавала? Я ушла с работы и нахожусь за границей!
Сейчас я скучаю по тебе.
Но фантастически ты знаешь шифры!
Более того, ты любишь математику!
Но я рассказывать хорошо не умею.
Работа - это для меня самое главное.
Дружба - это для меня тоже очень важно.
Жалею, что мы не поговорили по душам.
Я, конечно, когда-нибудь приеду опять.
Когда тебе будет скучно, напиши!
Криптография - очень полезная наука!
Я ещё напишу!


.....
ТВОЯ ПОДРУГА... КАК ГРУСТНО...
И МНЕ ТОЖЕ ГРУСТНО! НО ВАЖНЕЙ ТО, ЧТО Я ПРОГОЛОДАЛСЯ!
ПОЙДЁМ ЕСТЬ ЛАПШУ РАМЭН!
ПОЙДЁМ!
ПРИХОДИ, МЫ ТЕБЕ МЕСТО ЗАЙМЁМ!
СТУК
ДОРОГИЕ ЧИТАТЕЛИ, ВЫ ТОЖЕ, НАВЕРНОЕ, УЖЕ ДОГАДАЛИСЬ?!

Дополнительная информация

Расширенный алгоритм Евклида

Алгоритм Евклида – это метод нахождения наибольшего общего делителя (НОД) двух натуральных чисел, являющийся более эффективным, чем алгоритмы факторизации целых чисел. Алгоритм Евклида прост и надёжен: чтобы найти НОД двух натуральных чисел a и b ($a > b$), достаточно выполнять вычисления в следующем порядке.

① Поделив a на b, находим остаток r.

② Если $r = 0$, то НОД = b (поиск завершён).

③ Если $r \neq 0$, то возвращается к шагу ①, приняв $a = b$ и $b = r$.

Таким образом, наибольшим общим делителем будет делитель без остатка, полученный при выполнении вышеуказанного алгоритма, или, другими словами, последний ненулевой остаток.

Используя вышеописанный алгоритм Евклида, попробуем найти НОД, например, чисел 1365 и 77.

(a)	(b)	(r)		(a)	(b)	(r)
1365 ÷	77 =	17 (ост. 56)	→	1365 =	17 × 77 +	56
77 ÷	56 =	1 (ост. 21)	→	77 =	1 × 56 +	21
56 ÷	21 =	2 (ост. 14)	→	56 =	2 × 21 +	14
21 ÷	14 =	1 (ост. 7)	→	21 =	1 × 14 +	7
14 ÷	7 =	2 (ост. 0)	→	14 =	2 × 7 +	0

Таким образом, наибольший общий делитель чисел 1365 и 77 равен 7.

Нахождение решения неопределённого уравнения

Теперь попробуем использовать алгоритм Евклида для поиска наибольшего общего делителя двух взаимно простых чисел, например 20 и 17.

$$20 = 1 \times 17 + 3 \quad \cdots\cdots \quad (1)$$

$$17 = 5 \times 3 + 2 \quad \cdots\cdots \quad (2)$$

$$3 = 1 \times 2 + 1 \quad \cdots\cdots \quad (3)$$

$$2 = 2 \times 1 + 0$$

НОД оказался равным 1, о чём мы знали и так, поэтому может показаться, что в использовании алгоритма Евклида здесь нет никакой необходимости, однако в действительности из формул промежуточных вычислений по этому алгоритму можно извечь большую практическую пользу.

Перенося члены тождеств (1), (2) и (3), преобразуем их к следующему виду.

$$20 - 1 \times 17 = 3 \quad \cdots\cdots \quad (4)$$

$$17 - 5 \times 3 = ② \quad \cdots\cdots \quad (5)$$

$$3 - 1 \times ② = 1 \quad \cdots\cdots \quad (6)$$

Теперь подставляем левую часть тождества (5) вместо ② тождества (6) и группируем члены по множителям 3 и 17.

$$3 - 1 \times ② = 3 - 1 \times (17 - 5 \times 3) = 6 \times \boxed{3} - 1 \times 17 = 1 \quad \cdots\cdots \quad (7)$$

Теперь подставляем левую часть тождества (4) вместо $\boxed{3}$ тождества (7) и группируем члены по множителям 17 и 20.

$$6 \times \boxed{3} - 1 \times 17 = 6 \times (20 - 1 \times 17) - 1 \times 17 = 6 \times 20 - 7 \times 17 = 1$$

Перепишем результат этих преобразований следующим образом:

$$20 \times 6 + 17 \times (-7) = 1$$

Вышеприведённое тождество по форме соответствует уравнению $ax + by = c$, в котором a, b, c, x, y – целые числа. Уравнения подобного вида с неизвестными x и y называются неопределёнными уравнениями первого порядка.

Другими словами, используя формулы промежуточных вычислений по алгоритму Евлкида, мы нашли целочисленное решение неопределённого уравнения первого порядка для $a = 20$ и $b = 17$ – пару чисел $(x, y) = (6, -7)$. Этот метод, называемый расширенным алгоритом Евклида, имеет очень большую практическую ценность.

Для общего случая, когда a и b – не равные нулю целые числа, c – их наибольший общий делитель, уравнение

$$ax + by = c$$

имеет целочисленное решение (x_1, y_1), которое может быть найдено с использованием расширенного алгоритма Евклида. Правда, неопределённые уравнения первого порядка имеют множество решений, которые могут быть представлены в следующем виде (здесь k – произвольное целое число):

$$(x,\ y) = \left(x_1 + k \cdot \frac{b}{c},\ y_1 - k \cdot \frac{a}{c}\right) \quad \cdots\cdots \quad (8)$$

● Вычисление обратного элемента по модулю

Используя решение в общем виде (8), можно выразить множество решений неопределённого уравнения первого порядка $20x + 17y = 1$ в следующем виде:

$$(6 + 17k,\ -7 - 20k) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (9).$$

При $k = -1$ мы имеем решение $(x, y) = (-11, 13)$, подставив которое в неопределённое уравнение первого порядка $20x + 17y = 1$, получим следующее тождество:

$$20 \times (-11) + 17 \times 13 = 1.$$

Преобразуем выражение путём переноса члена в правую часть.

$$17 \times 13 = 1 + 11 \times 20 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (10).$$

Если внимательно посмотреть на тождество (10), то можно заметить, что оно означает верность следующего сравнения:

$$17 \times 13 \equiv 1 \pmod{20} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (11).$$

На стр. 168 говорилось, что если $ed \equiv 1 \pmod{L}$, то ключ расшифрования d – обратное число (обратный элемент) к ключу шифрования e относительно умножения по модулю L. Другими словами, сравнение (11) означает, что число 13 является обратным элементом к числу 17 относительно умножения по модулю 20.

Это означает, что, используя расширенный алгоритм Евклида, можно легко находить обратные элементы, а так как в шифре с открытым ключом это используется для генерации секретного ключа (ключа расшифрования), этот алгоритм широко используется и в криптографии.

Итак, выше мы смогли найти 17^{-1} – обратный элемент к 17 по модулю 20, но можно ли найти, например, 16^{-1} – обратный элемент к 16 (mod 20)? Так как НОД чисел 16 и 20 равен 4, мы можем найти решение уравнения $20x + 16y = 4$, как было показано выше. Однако неопределённое уравнение первого порядка $20x + 16y = 1$, решить которое нужно для нахождения обратного элемента, не имеет решений в области целых чисел, так как его левая часть всегда будет кратна 4. Другими словами, если два числа не являются взаимно простыми, то нельзя найти обратный элемент. Таким образом, получение обратного элемента возможно только в том случае, если два числа являются взаимно простыми.

В заключение попробуем на практике найти 73^{-1} в качестве обратного элемента к 73 по модулю 1001, используя расширенный алгоритм Евклида. Сначала с помощью алгоритма Евклида мы находим НОД чисел 73 и 1001.

$$1001 = 13 \times 73 + 52$$
$$73 = 1 \times 52 + 21$$
$$52 = 2 \times 21 + 10$$
$$21 = 2 \times 10 + 1$$
$$10 = 10 \times 1 + 0$$

Следовательно, наибольший общий делитель чисел 73 и 1001 равен 1, другими словами, 73 и 1001 являются взаимно простыми числами.

Теперь перенесём члены в этих тождествах так, чтобы в правых частях были остатки.

$$1001 - 13 \times 73 = 52 \quad \cdots\cdots (12)$$
$$73 - 1 \times 52 = 21 \quad \cdots\cdots (13)$$
$$52 - 2 \times 21 = 10 \quad \cdots\cdots (14)$$
$$21 - 2 \times 10 = 1 \quad \cdots\cdots (15)$$

Подставим выражение (14) вместо множителя 10 в выражение (15).

$$21 - 2 \times (52 - 2 \times 21) = 1$$
$$21 - 2 \times 52 + 4 \times 21 = 1 \quad \cdots\cdots (16)$$

В выражении (16) сгруппируем члены по множителям 21 и 52.

$$5 \times 21 - 2 \times 52 = 1 \quad \cdots\cdots (17)$$

Подставим выражение (13) вместо множителя 21 в выражение (17).

$$5 \times (73 - 1 \times 52) - 2 \times 52 = 1$$
$$5 \times 73 - 5 \times 52 - 2 \times 52 = 1 \quad \cdots\cdots (18)$$

В выражении (18) сгруппируем члены по множителям 52 и 73.

$$5 \times 73 - 7 \times 52 = 1 \quad \cdots\cdots (19)$$

Подставим выражение (12) вместо множителя 52 в выражение (19).

$$5 \times 73 - 7 \times (1001 - 13 \times 73) = 1$$
$$5 \times 73 - 7 \times 1001 + 91 \times 73 = 1 \quad \cdots\cdots (20)$$

В выражении (20) сгруппируем члены по множителям 73 и 1001.

$$96 \times 73 - 7 \times 1001 = 1 \quad \cdots\cdots (21)$$

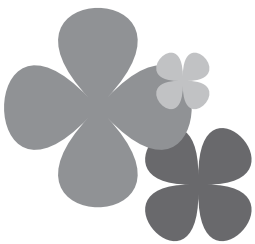
Перенесём член в выражении (21) так, как показано ниже.

$$96 \times 73 = 1 + 7 \times 1001$$

Так как это тождество означает верность сравнения $96 \times 73 \equiv 1 \pmod{1001}$, обратным элементом 73^{-1} к числу 73 по модулю 1001 является число 96.

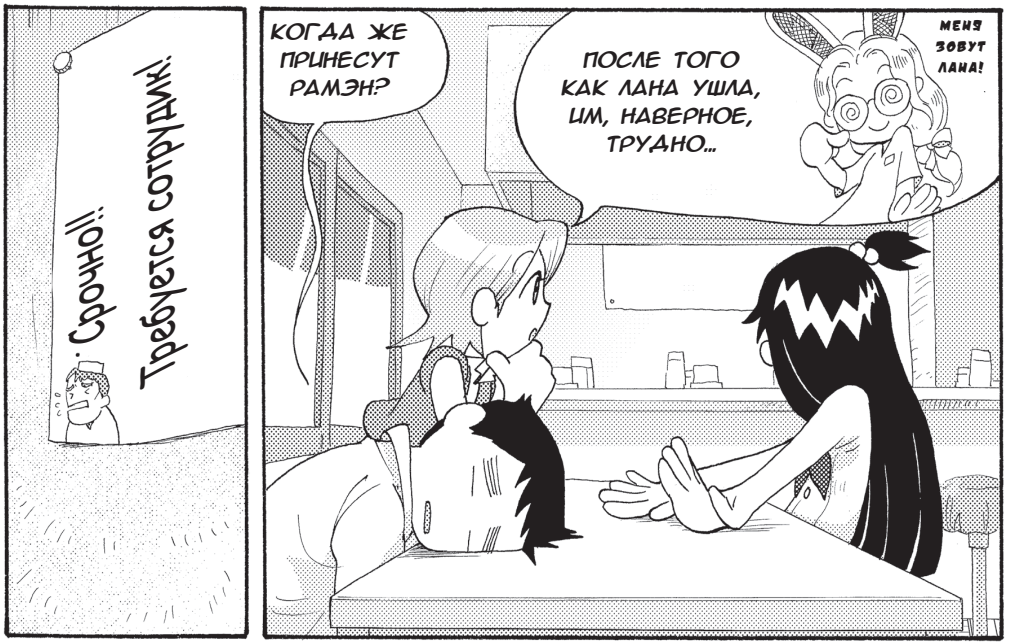
ГЛАВА 4

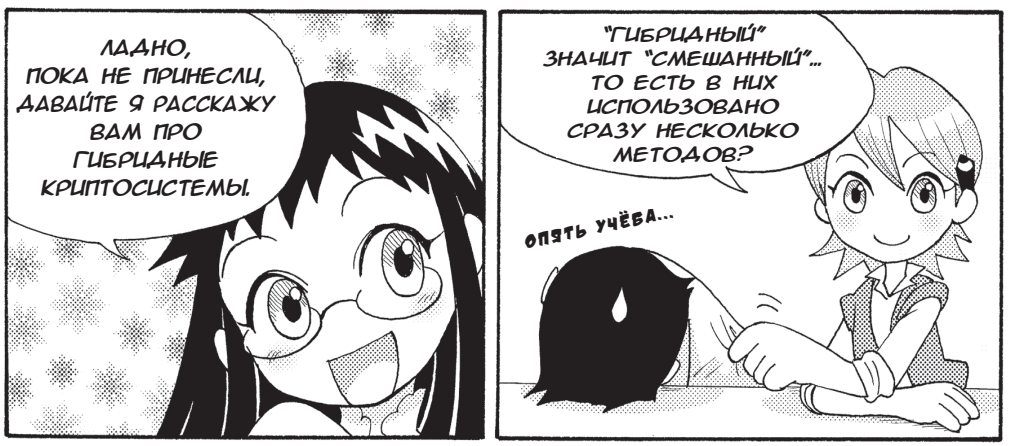
КАК ИСПОЛЬЗУЮТ ШИФР НА ПРАКТИКЕ?



4-1 Гибридные криптосистемы

Вывески: ※1 – кафе «Заяц», ※2 – лапша рамэн.





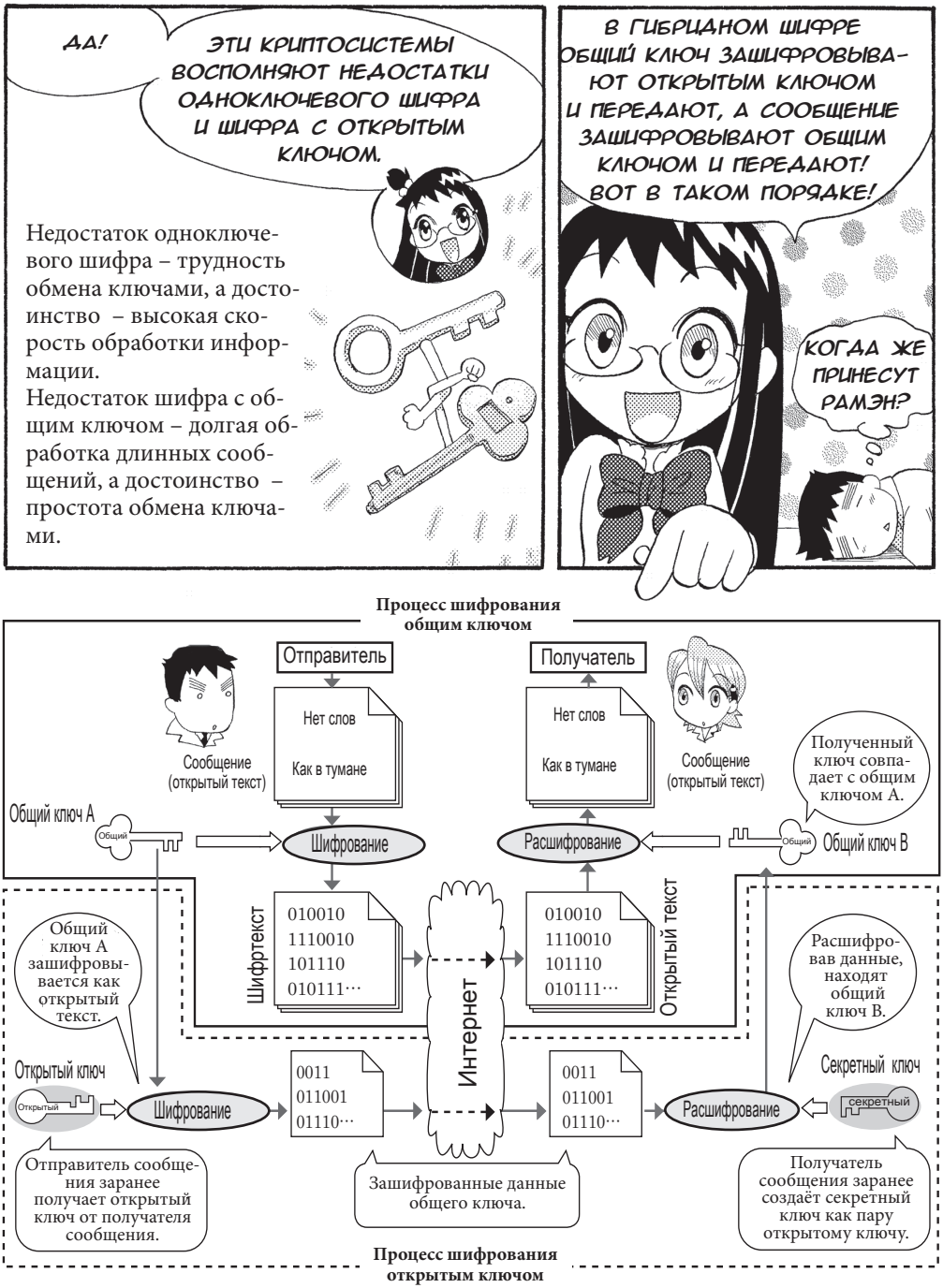
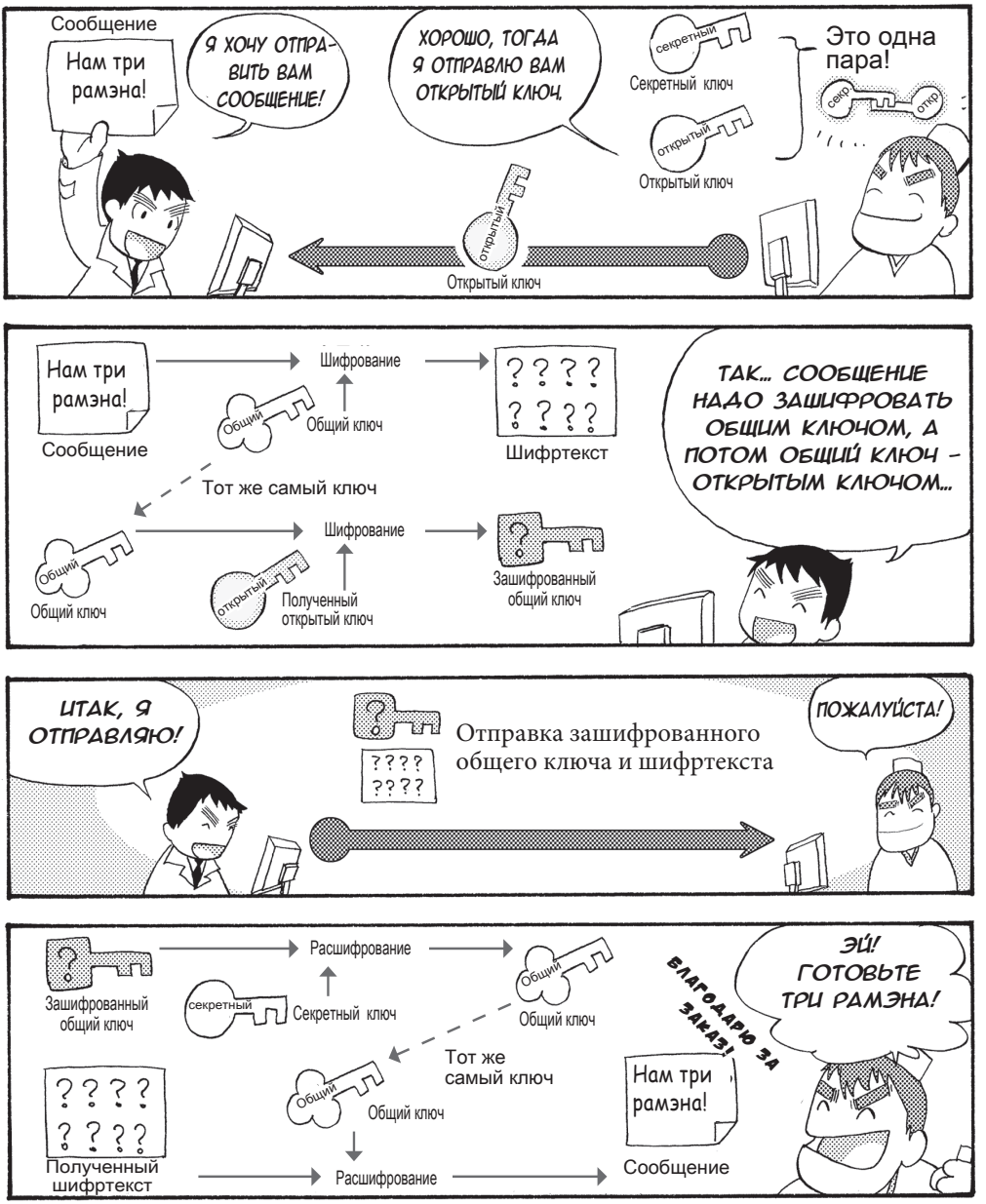
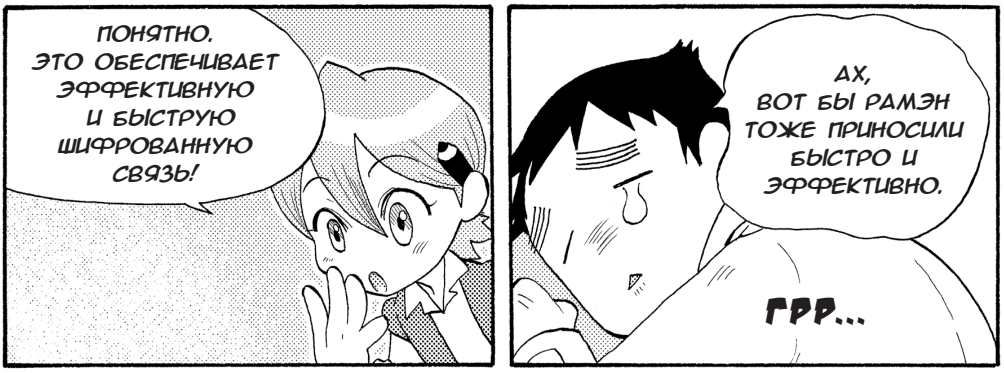


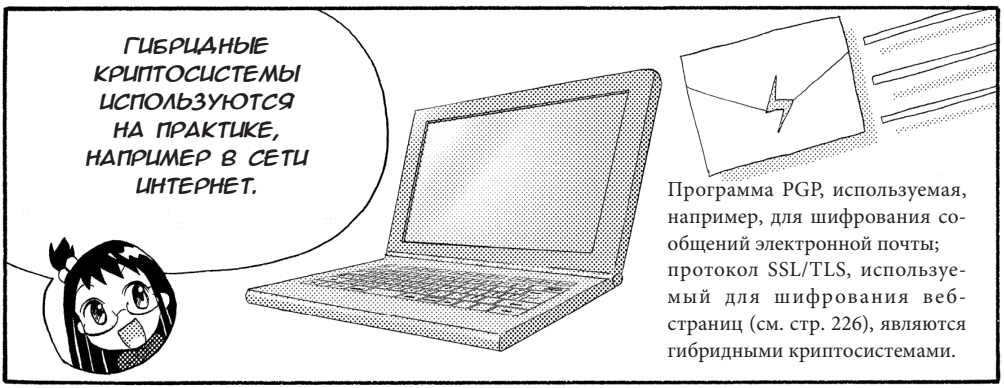
Рис. 4.1. *Общая картина шифрования и расшифрования в гибридной криптосистеме*

Как показано на рис. 4.1, шифр с открытым ключом используется только для шифрования и расшифрования общего ключа, а одноключевой шифр – только для шифрования и расшифрования сообщения. Таким образом, длинные сообщения быстро зашифровываются и расшифровываются с помощью общего ключа, а благодаря тому что общий ключ зашифровывается открытым ключом и передаётся по каналу связи, не возникает проблемы обмена ключами – самой большой уязвимости одноключевого шифра.

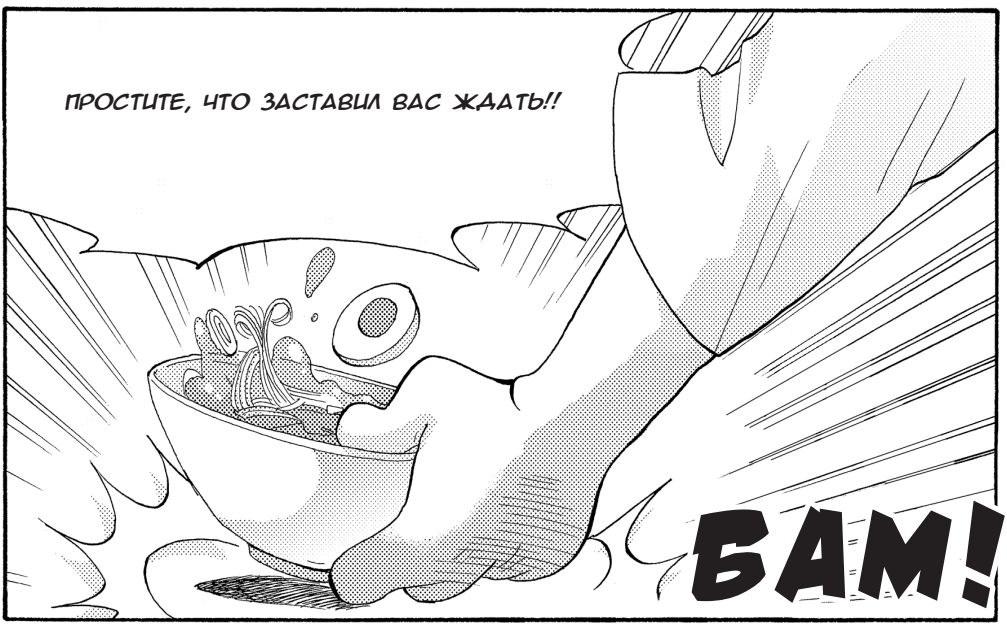
Итак, теперь давайте изучим практическое использование гибридной криптосистемы на примере заказа лапши рамэн.





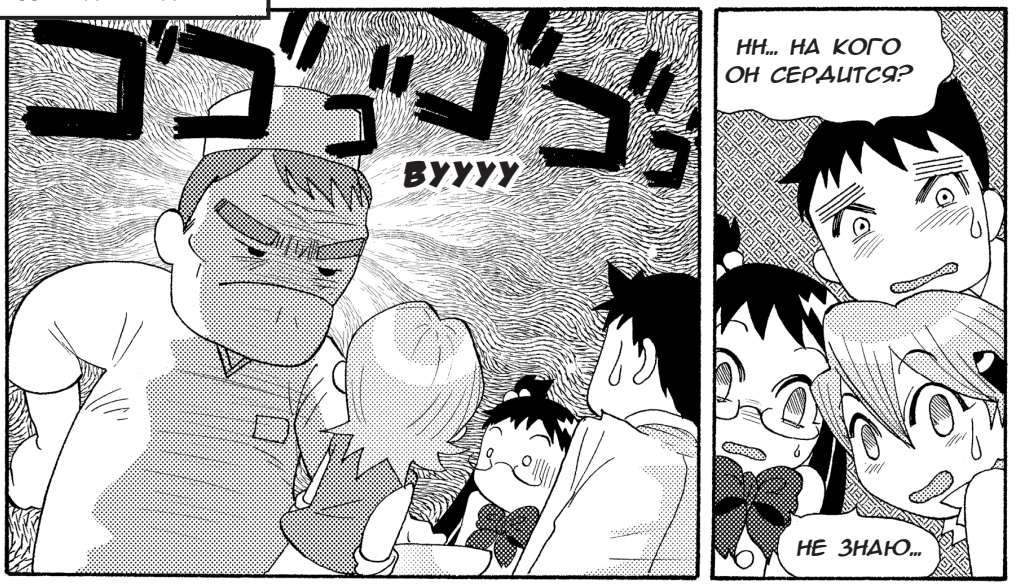


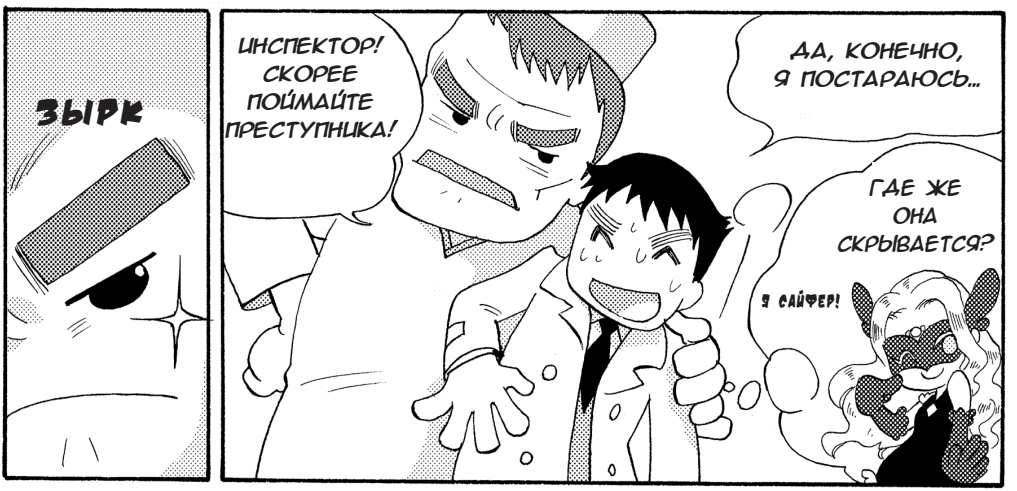
Программа PGP, используемая, например, для шифрования сообщений электронной почты; протокол SSL/TLS, используемый для шифрования вебстраниц (см. стр. 226), являются гибридными криптосистемами.

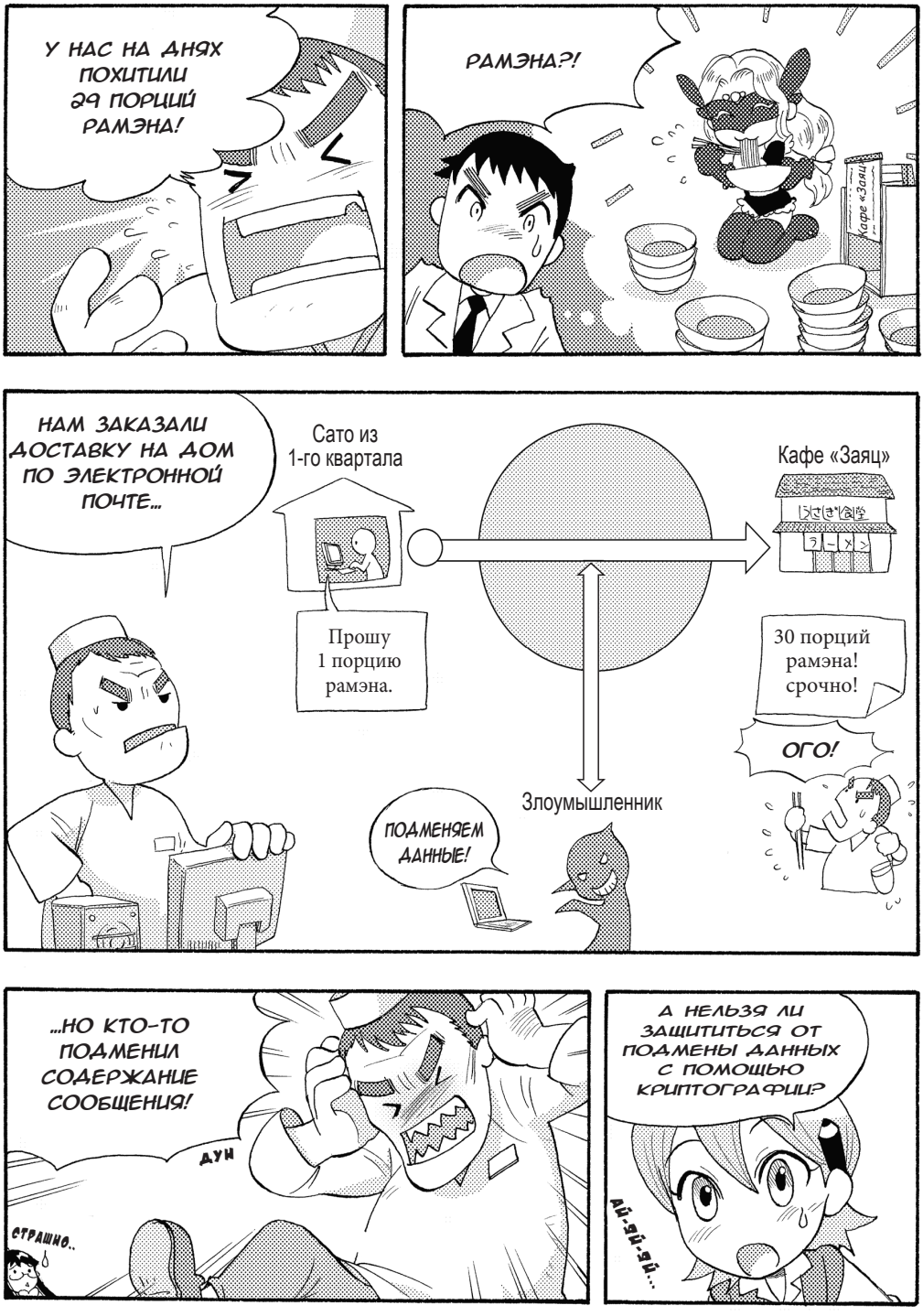


4-2 Хеш-функция и код аутентификации сообщения

Подмена данных






У НАС НА ДНЯХ ПОХИТИЛИ 29 ПОРЦИЙ РАМЭНА!
РАМЭНА?!
Кафе «Заяц»
НАМ ЗАКАЗАЛИ ДОСТАВКУ НА ДОМ ПО ЭЛЕКТРОННОЙ ПОЧТЕ...
Сато из 1-го квартала
Прошу 1 порцию рамэна.
Кафе «Заяц»
30 порций рамэна! срочно!
ОГО!
Злоумышленник
ПОДМЕНЯЕМ ДАННЫЕ!
...НО КТО-ТО ПОДМЕНИЛ СОДЕРЖАНИЕ СООБЩЕНИЯ!
ДУН
СТРАШНО..
А НЕЛЬЗЯ ЛИ ЗАЩИТИТЬСЯ ОТ ПОДМЕНЫ ДАННЫХ С ПОМОЩЬЮ КРИПТОГРАФИИ?
АЙ-ЯЙ-ЯЙ...


ДЛЯ ЗАЩИТЫ ОТ ПОДМЕНЫ ДАННЫХ МОЖНО ИСПОЛЬЗОВАТЬ ХЕШ-ФУНКЦИЮ!!
hash


ХЕШ? ЭТО КАК-ТО СВЯЗАНО С hashed beef?
НАСМЕШИЛ! ЭТО ЖЕ БЛЮДО ИЗ МЕЛКО НАРЕЗАННОЙ ВАРЁНОЙ ГОВЯДИНЫ!
ХА-ХА


СЛОВО Hash ПО-АНГЛИЙСКИ И ОЗНАЧАЕТ "МЕЛКО НАРЕЗАТЬ".
НО ЗДЕСЬ МЫ ВМЕСТО ГОВЯДИНЫ МЕЛКО НАРЕЗАЕМ СООБЩЕНИЕ И ПОЛУЧАЕМ ЕГО ХЕШ. ЭТО НАЗЫВАЕТСЯ ХЕШ-ФУНКЦИЕЙ.
ДА?
ОЧЕНЬ ВКУСНО, НАВЕРНОЕ...


А ЧТО ТАКОЕ ХЕШ?
ОНО ВООБЩЕ СЪЕДОБНОЕ?


ЭТО ЗНАЧЕНИЕ, ВЫЧИСЛЯЕМОЕ НА ОСНОВЕ СООБЩЕНИЯ. ОНО ПОДОБНО "ОТПЕЧАТКАМ ПАЛЬЦЕВ" ДЛЯ СЛЕДОВАТЕЛЯ
ЕГО ИСПОЛЬЗУЮТ ДЛЯ ТОГО, ЧТОБЫ ПРОВЕРИТЬ, НЕТ ЛИ ПОДМЕНЫ ДАННЫХ В СООБЩЕНИИ.
НЕТ, НЕСЪЕДОБНОЕ!

❀ Хеш-функция

Хеш-функция вычисляет хеш для собщения. Хеш можно уподобить отпечаткам пальцев, помогающим идентифицировать личность. Он представляет собой данные фиксированной длины, полученные путём свёртки данных исходного сообщения, и содержит меньше информации, чем исходное сообщение.

Отсутствие подмены данных в исходном сообщении называется целостностью данных, и её проверка возможна благодаря тому, что отправитель прилагает к исходному сообщению его хеш. Другими словами, с помощью этих «отпечатков пальцев» проверяют наличие или отсутствие в сообщении злонамеренных изменений. Получатель сообщения, используя ту же самую хеш-функцию, что и отправитель, вычисляет хеш полученного сообщения и сравнивает его с хешем, которое приложил к исходному сообщению отправитель. Если эти два хеша совпадают, то получатель убеждается в то, что данные в сообщении не были подменены.

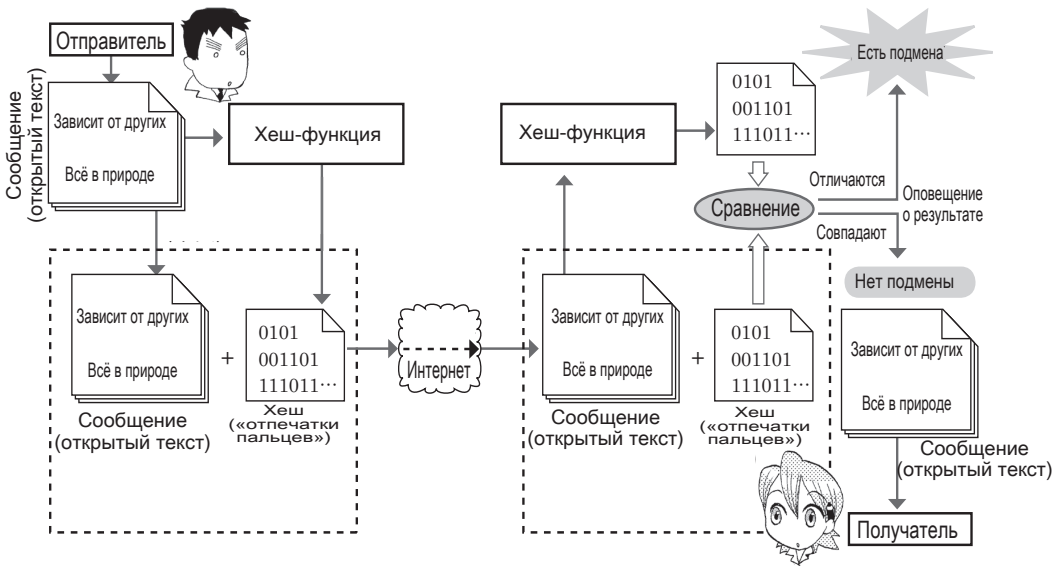


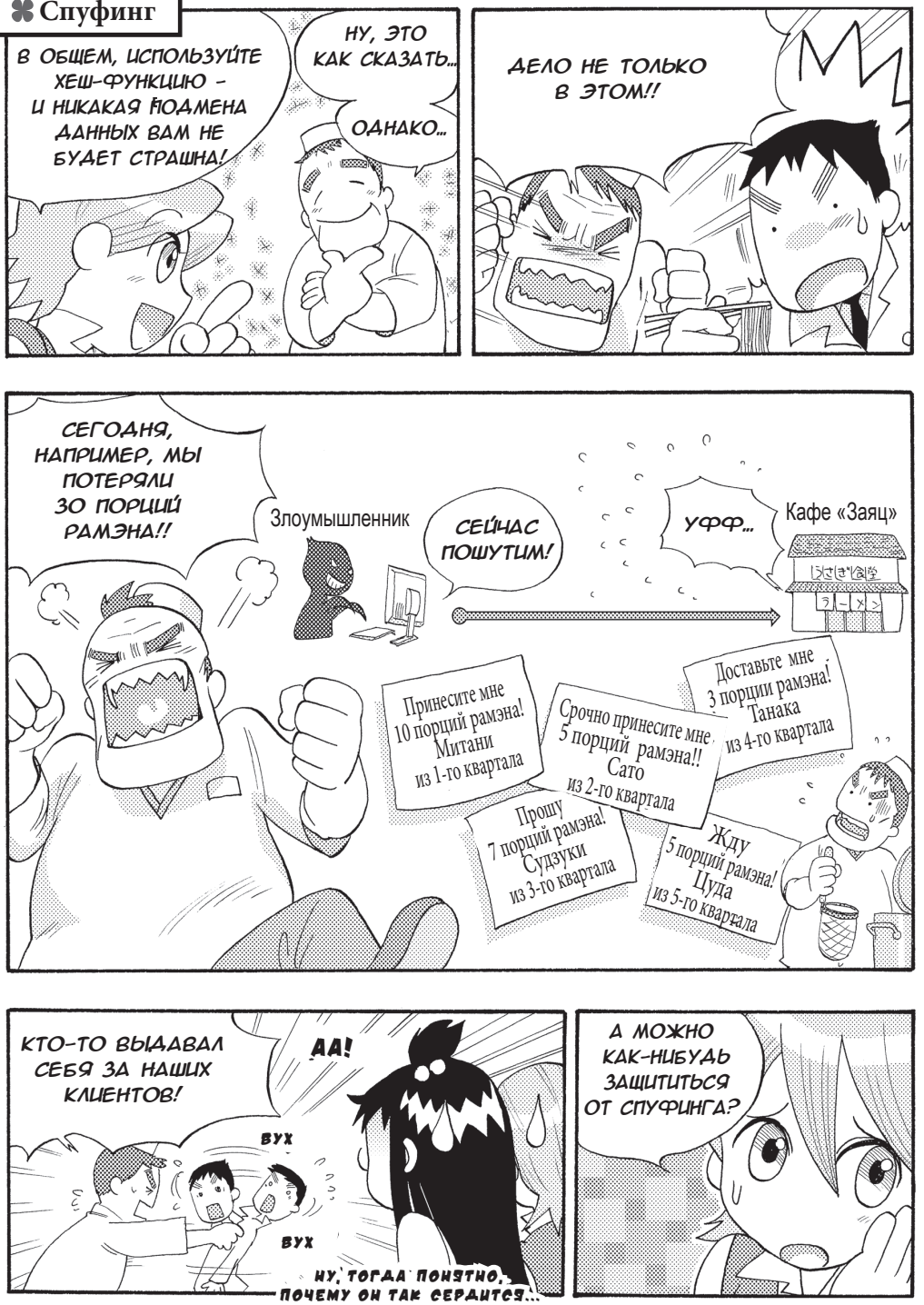
Рис. 4.2. Хеш-функция

Хеш-функция должна быть односторонней, чтобы по собщению можно было вычислить его хеш, но по хешу невозможно было восстановить исходное сообщение. Преобразование данных, выполняемое односторонней хеш-функцией, называется необратимым.

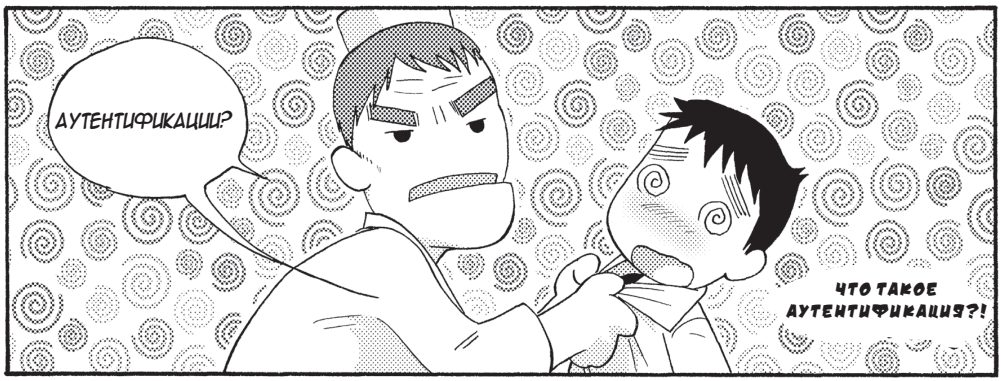
Кроме того, хеш-функция должна затруднять нахождение двух отличающихся друг от друга сообщений, имеющих одинаковый хеш. Это условие называется «сильной устойчивостью к коллизиям». Кроме того, при наличии одного сообщения с известным хешем хеш-функция должна затруднять нахождение другого сообщения, имеющего такой же хеш. Это условие называется «слабой устойчивостью к коллизиям». Для этих целей были специально разработаны такие хеш-функции, как MD5, SHA-1, SHA-512, RIPEMD-160 и др.※

※ MD5, SH-1, а также RIPEMD уже не считаются стойкими – *Прим. ред.*

✽ Спуфинг


В ОБЩЕМ, ИСПОЛЬЗУЙТЕ ХЕШ-ФУНКЦИЮ – И НИКАКАЯ ПОДМЕНА ДАННЫХ ВАМ НЕ БУДЕТ СТРАШНА!
НУ, ЭТО КАК СКАЗАТЬ...
ОДНАКО...
ДЕЛО НЕ ТОЛЬКО В ЭТОМ!!
СЕГОДНЯ, НАПРИМЕР, МЫ ПОТЕРЯЛИ 30 ПОРЦИЙ РАМЭНА!!
Злоумышленник
СЕЙЧАС ПОШУТИМ!
УФФ...
Кафе «Заяц»
Принесите мне 10 порций рамэна! Митани из 1-го квартала
Срочно принесите мне 5 порций рамэна!! Сато из 2-го квартала
Доставьте мне 3 порции рамэна! Танака из 4-го квартала
Прошу 7 порций рамэна! Судзуки из 3-го квартала
Жду 5 порций рамэна! Цуда из 5-го квартала
КТО-ТО ВЫДАВАЛ СЕБЯ ЗА НАШИХ КЛИЕНТОВ!
АА!
ВУХ
ВУХ
НУ, ТОГДА ПОНЯТНО, ПОЧЕМУ ОН ТАК СЕРДИТСЯ...
А МОЖНО КАК-НИБУДЬ ЗАЩИТИТЬСЯ ОТ СПУФИНГА?


МОЖНО, ЕСЛИ ИСПОЛЬЗОВАТЬ ИМИТОВСТАВКУ, ТО ЕСТЬ КОД АУТЕНТИФИКАЦИИ СООБЩЕНИЯ!
Имитовставка (код аутентификации сообщения)
MAC : Message Authentication Code


АУТЕНТИФИКАЦИИ?
ЧТО ТАКОЕ АУТЕНТИФИКАЦИЯ?!


ЭТО ПРОВЕРКА ПОДЛИННОСТИ СООБЩЕНИЯ, ТО ЕСТЬ ТОГО,
Рамэн, две порции!
Принесите мне один «Рамэн»!
Сато из 1-го квартала
Аутентично
рамэн, 3 порции! Накаи из 3-го квартала
Аутентично
порции»!
Аутентично
Пришлите мне рамэн и гёдза!
Аида из 6-го квартала
Аутентично
Вышлите «Рамэн», 2 порции!
Жду «Рам
Аутентично
...ТО СОБЩЕНИЕ ДЕЙСТВИТЕЛЬНО ОТПРАВИЛ НАСТОЯЩИЙ ОТПРАВИТЕЛЬ,
...А НЕ ЗЛОУМЫШЛЕННИК, КОТОРЫЙ ИМ ПРИТВОРЯЕТСЯ.


ОГО!
А КАК ЭТО МОЖНО ПРОВЕРИТЬ?

❀ Устройство имитовставки

Код аутентификации сообщения (MAC), называемый также имитовставкой, позволяет проверить целостность данных и аутентичность сообщения. Опишем здесь устройство имитовставки по рис. 4.3.

Отправитель прилагает к отправляемому сообщению MAC-код, который используется для проверки сообщения, как и уже изученное нами значение хеш-функции.

Получатель сравнивает MAC-код, который он вычислил на основе полученного сообщения, с MAC-кодом, который был приложен к этому сообщению отправителем. Это позволяет получателю убедиться в целостности и аутентичности сообщения. При этом и отправитель, и получатель используют для генерирования MAC-кода один и тот же общий ключ.

Если эти два MAC-кода совпадают, то это означает, во-первых, отсутствие подмены данных в полученном сообщении (целостность), и, во-вторых, аутентичность отправителя, обладающего тем же самым общим ключом, что и получатель.

Если же эти два MAC-кода отличаются друг от друга, то это может означать либо подмену данных сообщения в процессе передачи, либо отсутствие у отправителя сообщения общего ключа, которым должен обладать аутентичный отправитель.

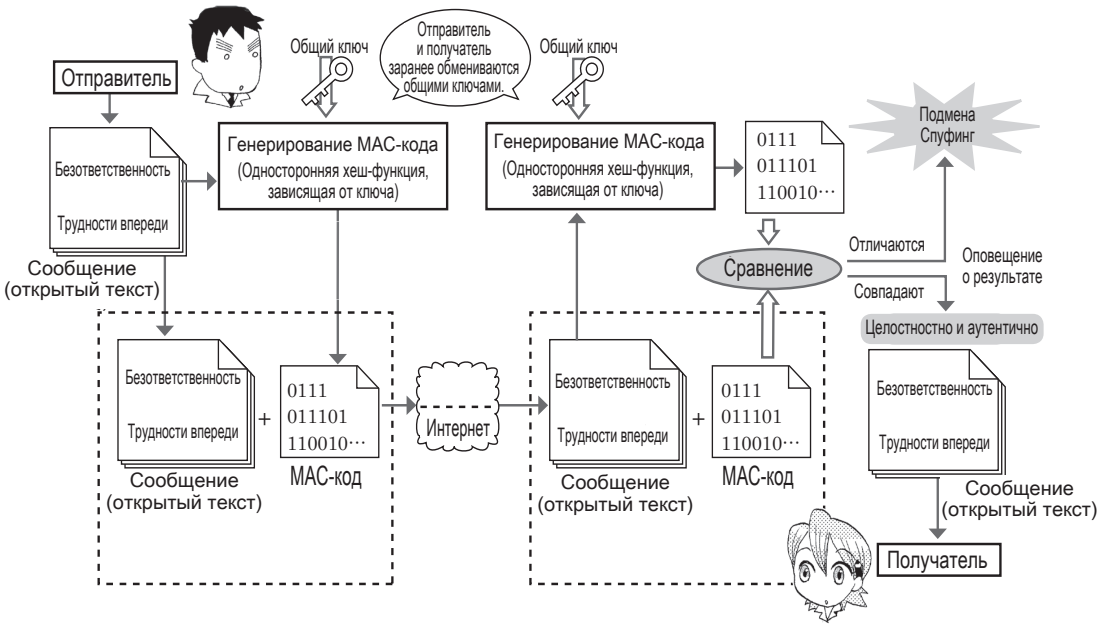


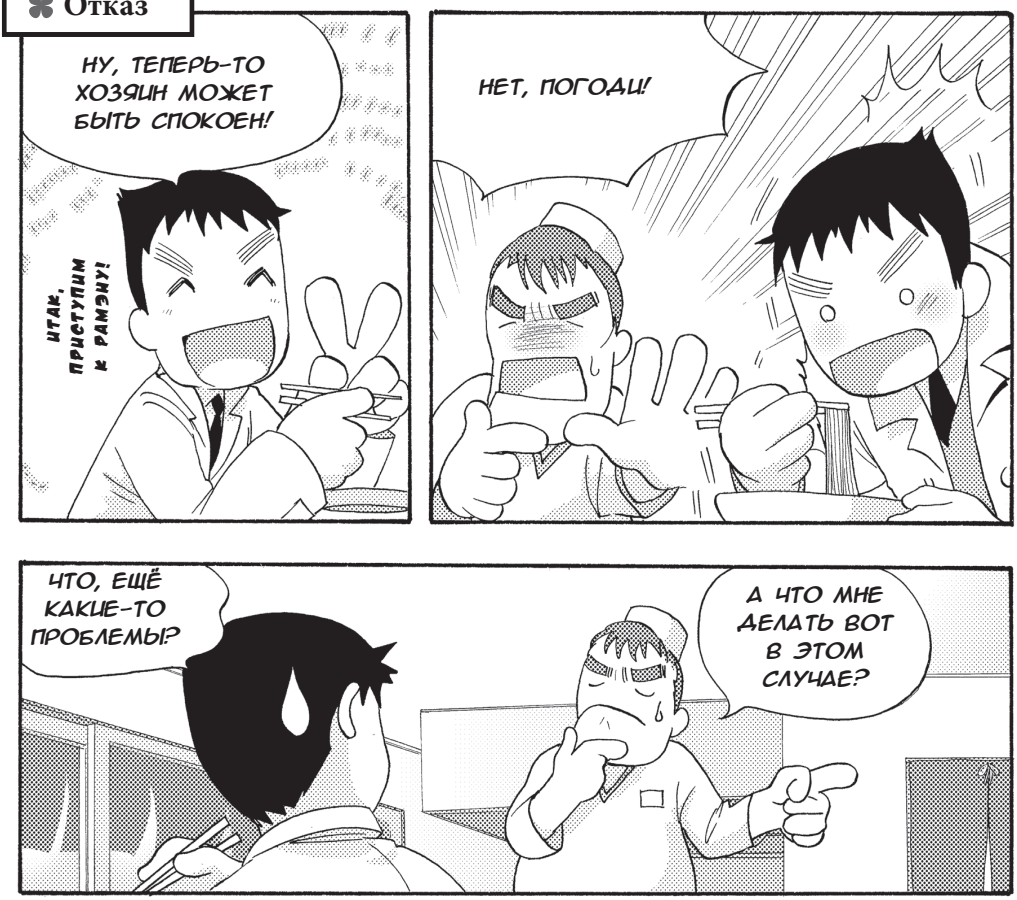
Рис. 4.3. Устройство имитовставки

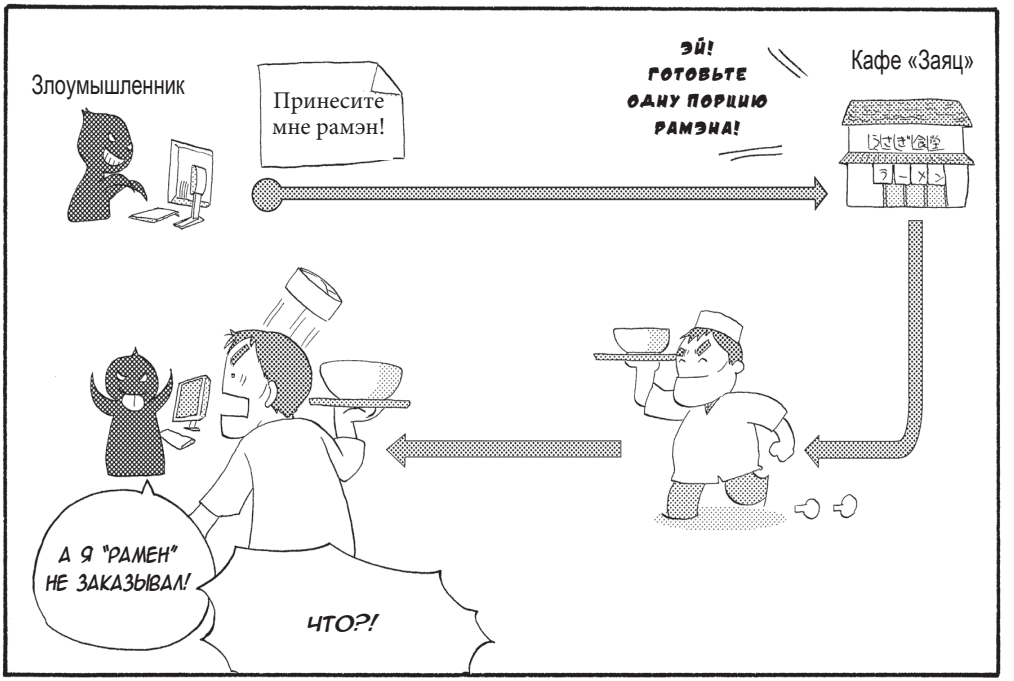
Имитовставку можно рассматривать как разновидность односторонней хеш-функции, снабжённую ключом. Она используется так же, как и хеш: отправитель и получатель независимо друг от друга вычисляют MAC-код, а целостность и аутентичность проверяются путём сравнения вычисленного и полученного MAC-кодов.

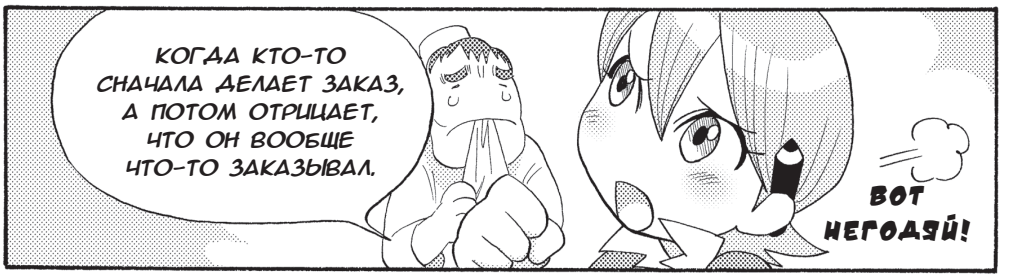
Для генерирования MAC-кода и отправитель, и получатель используют один и тот же общий ключ, которым должны обладать только они. Это позволяет убедиться в том, что MAC-код, приложенный к полученному сообщению, был сгенерирован отправителем, у которого есть тот же самый общий ключ, но порождает такой же недостаток, какой имеется у одноключевого шифра: проблему безопасности совместного обладания общим ключом.

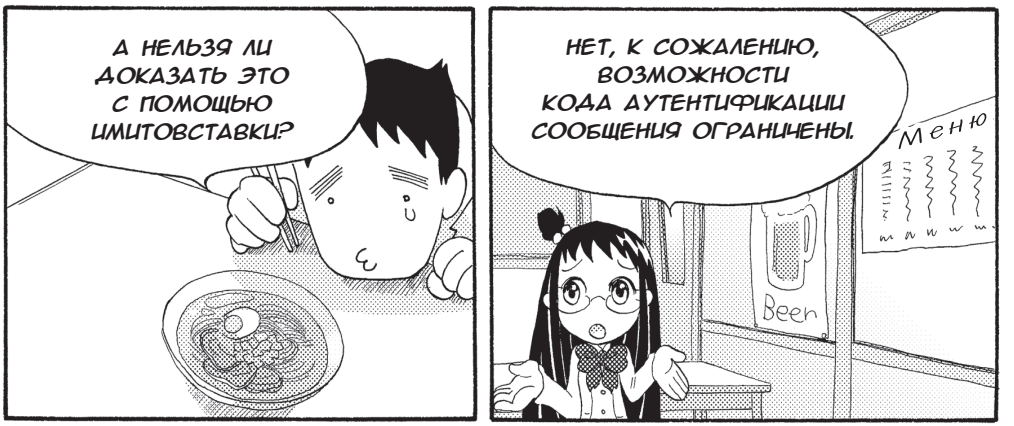
Имитоставка используется в международных денежных переводах между банками, в протоколе SSL/TLS, который используется, например, для интернет-шоппинга, и т. д.

✤ Отказ




Злоумышленник
Принесите мне рамэн!
ЭЙ! ГОТОВЬТЕ ОДНУ ПОРЦИЮ РАМЭНА!
Кафе «Заяц»
А Я "РАМЕН" НЕ ЗАКАЗЫВАЛ!
ЧТО?!

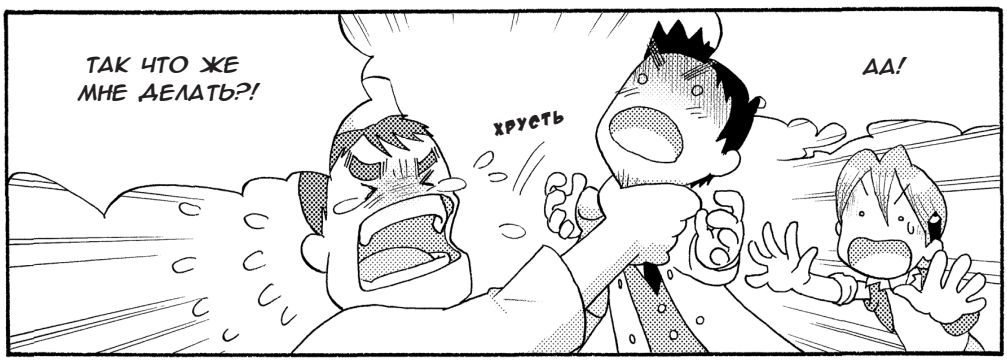

КОГДА КТО-ТО СНАЧАЛА ДЕЛАЕТ ЗАКАЗ, А ПОТОМ ОТРИЦАЕТ, ЧТО ОН ВООБЩЕ ЧТО-ТО ЗАКАЗЫВАЛ.
ВОТ НЕГОДЯЙ!


А НЕЛЬЗЯ ЛИ ДОКАЗАТЬ ЭТО С ПОМОЩЬЮ ИМИТОВСТАВКИ?
НЕТ, К СОЖАЛЕНИЮ, ВОЗМОЖНОСТИ КОДА АУТЕНТИФИКАЦИИ СООБЩЕНИЯ ОГРАНИЧЕНЫ.
Меню
Been

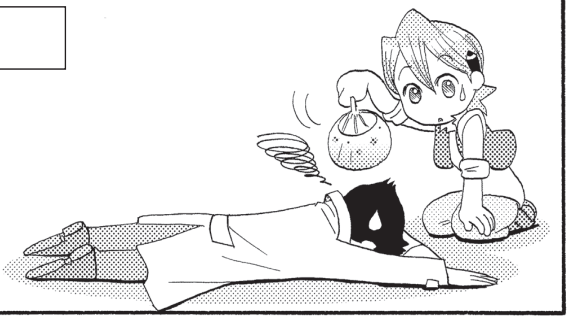
✾ Два недостатка имитовставки

(1) **Проблема невозможности предотвращения отказа** (non-repudiation)
Представим, что пользователь **A** сначала отправил пользователю **B** сообщение и MAC-код, а потом стал утверждать, что он ничего не отправлял: якобы сообщение и MAC-код пользователь **B** создал сам. В подобном случае **B** не сможет опровергнуть это утверждение **A**, ведь даже если попросить третью сторону разрешить спор, она не сможет этого сделать, так как не имеет никаких средств, позволяющих определить, кто на самом деле создал сообщение и MAC-код – **A** или **B**.

(2) **Проблема невозможности доказательства третьей стороне**
Представим, что пользователь **A** отправил пользователю **B** сообщение и MAC-код. Однако пользователь **B** не сможет доказать третьей стороне **C**, что это сообщение действительно отправил ему **A**, ведь и сообщение, и его MAC-код могли быть созданы как пользователем **A**, так и пользователем **B**. Другими словами, третья сторона не сможет вынести обоснованного заключения о том, кто создал сообщение и MAC-код – **A** или **B**, так как оба они обладают одним и тем же общим ключом.



4-3 Цифровая подпись



Защита от отказа

Таблица 4.1. Шифр с открытым ключом и цифровая подпись

Шифр с открытым ключом	Шифрование открытым ключом получателя	→	Шифр-текст	→	Расшифрование секретным ключом получателя
Цифровая подпись	Расшифрование открытым ключом отправителя	←	Цифровая подпись	←	Шифрование секретным ключом отправителя

❀ Устройство цифровой подписи

Отправитель, зашифровывая отправляемое сообщение своим секретным ключом, создаёт из него цифровую подпись и прилагает её к отправляемому сообщению.

Получатель, расшифровывая подпись с помощью открытого ключа отправителя, восстанавливает из неё сообщение, а затем сравнивает его с сообщением, к которому эта цифровая подпись была приложена.

Идентичность этих двух сообщений подтверждает как целостность данных сообщения, так и аутентичность отправителя.

Кроме того, благодаря использованию открытого ключа отправителя для расшифрования цифровой подписи и третья сторона, и сам получатель имеют возможность удостовериться в том, что сообщение и его цифровая подпись были созданы настоящим отправителем, а это решает проблему доказательства третьей стороне и проблему отказа отправителя.

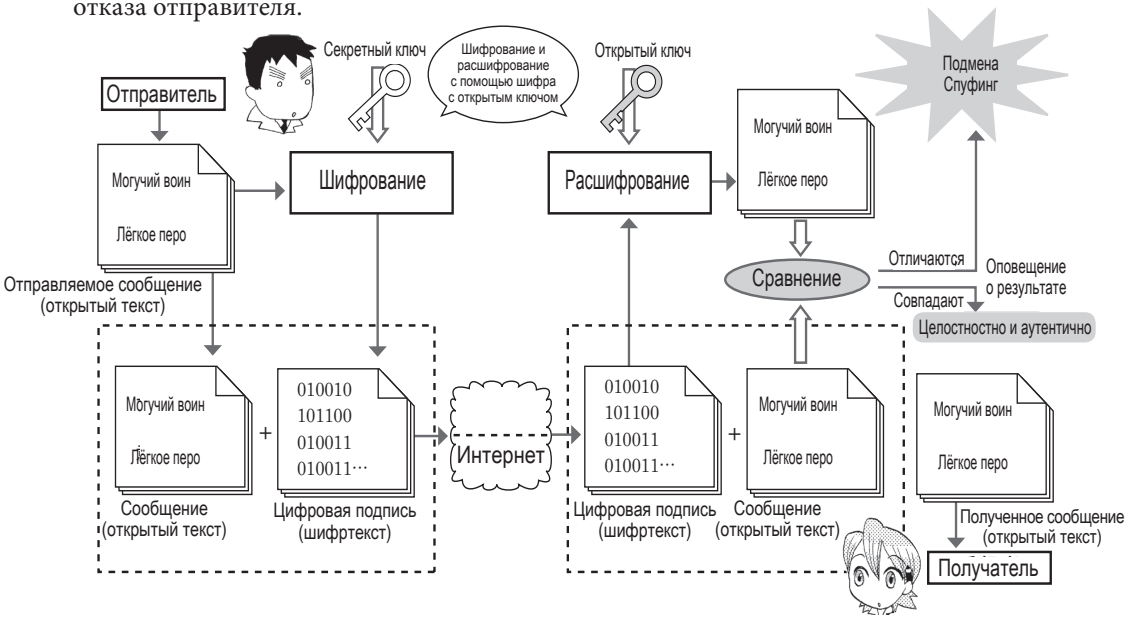


Рис. 4.4. Устройство (1) цифровой подписи
(в случае использования зашифрованного сообщения в качестве цифровой подписи)

В модели, показанной на рис 4.4, с целью упрощения и для лучшего понимания основной идеи цифровая подпись создаётся непосредственным шифрованием сообщения※, но шифрование и расшифрование полного сообщения с помощью шифра с открытым ключом заняло бы слишком много времени, поэтому на практике для создания цифровой подписи зашифровывается не всё сообщение, а его хеш, полученный с помощью односторонней хеш-функции.

※ Описание подписи как результат шифрования сообщения секретным ключом, вообще говоря, неверное. Это верно только для RSA – *Прим. ред.*

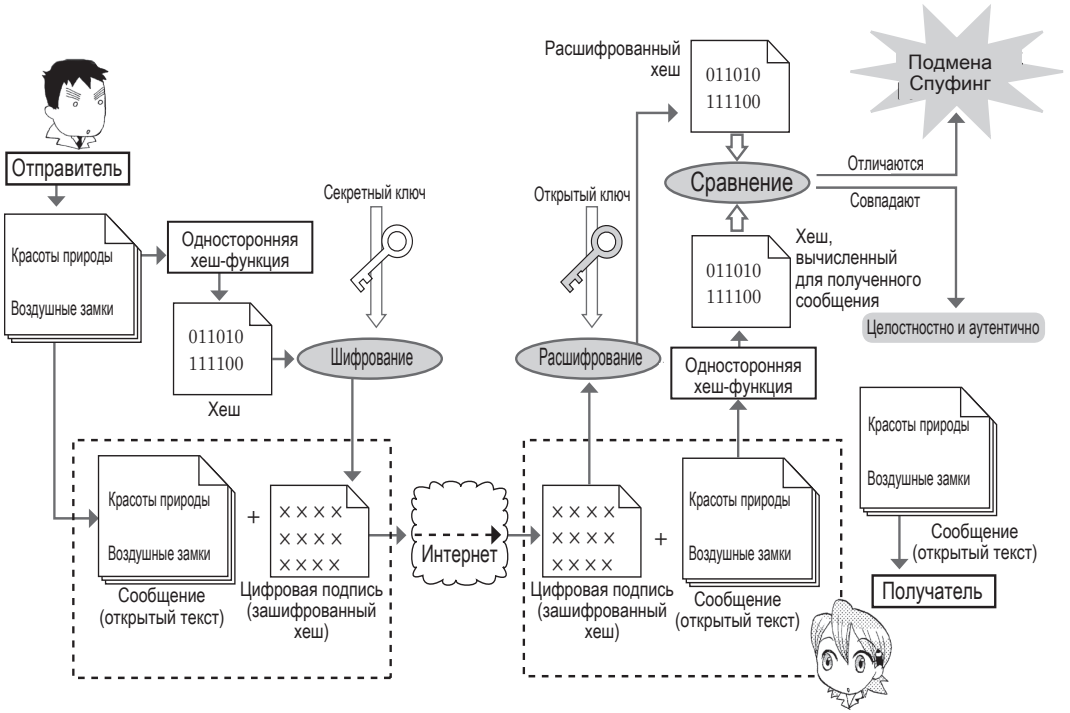
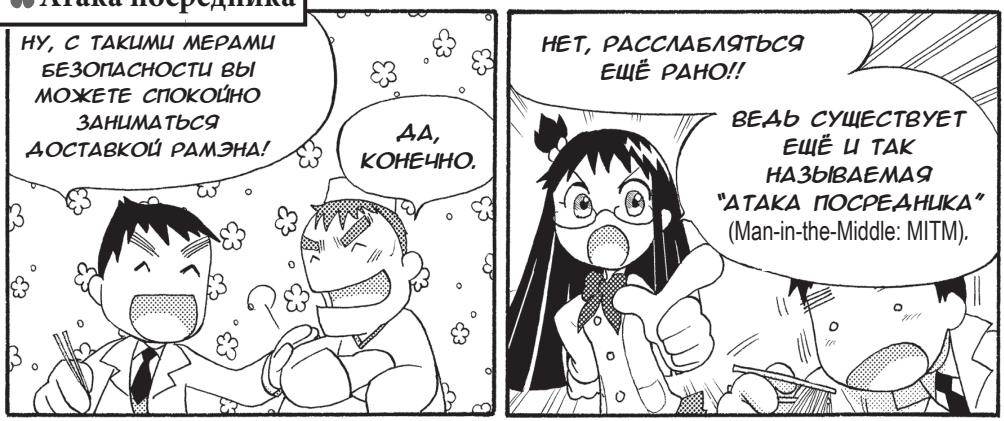


Рис. 4.5. Устройство (2) цифровой подписи
(в случае использования зашифрованного хеша сообщения в качестве цифровой подписи)

Цифровая подпись используется также и для создания сертификатов для проверки аутентичности серверов SSL/TLS. Сертификат – это открытый ключ (в данном случае открытый ключ сервера) с приложенной к нему цифровой подписью. Кроме того, цифровую подпись прилагают к приложениям, скачиваемым по сети, что позволяет предотвратить их изменение.

✤ Атака посредника



Пусть пользователь **A** – это Сато из 1-квартала, который хочет отправить сообщение, а пользователь **B** – это кафе «Заяц», которое должно получить это сообщение. Для осуществления этого пользователь **B** сначала должен отправить пользователю **A** свой открытый ключ. Однако в процессе передачи ключа злоумышленник может его перехватить и отправить пользователю **A** свой открытый ключ вместо открытого ключа пользователя **B**.

В подобном случае злоумышленник с помощью своего секретного ключа сможет расшифровать сообщение, отправленное пользователем **A** пользователю **B**, и, кроме того, изменить его содержание, зашифровать открытым ключом пользователя **B** и отправить пользователю **B** так, как будто бы это сообщение от пользователя **A**, ведь пользователь **B** не располагает никакими средствами, которые позволили бы ему обнаружить эту атаку.

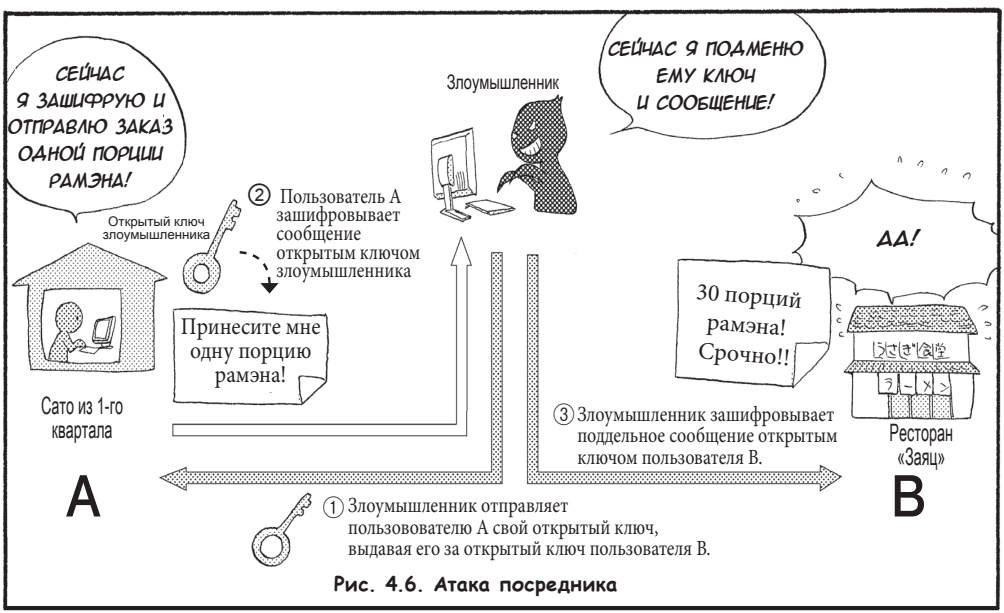


Рис. 4.6. Атака посредника

Сертификат и удостоверяющий центр

Сертификат, состоящий из открытого ключа и приложенной к нему цифровой подписи этого открытого ключа, выпускается удостоверяющими центрами (CA: Certification Authority). Пользователь, желающий опубликовать открытый ключ, регистрирует его в удостоверяющем центре, поручая ему выпуск сертификата.

В ответ на это поручение удостоверяющий центр проверяет подлинность пользователя и, если он отвечает критериям центра, создаёт на основе открытого ключа пользователя цифровую подпись, которая в паре с открытым ключом образует сертификат. Пара открытый ключ/секретный ключ в одних случаях может быть сгенерирована самим пользователем, в других – её генерирует удостоверяющий центр во время регистрации пользователя.

Механизм заверения открытого ключа с помощью сертификата является гарантией того, что этот открытый ключ действительно принадлежит пользователю **A**. Для этого пользователь **A** поручает заслуживающей доверия третьей стороне – удостоверяющему центру, – подтвердить подлинность открытого ключа, принадлежащего пользователю **A**. Давайте теперь изучим порядок процедуры сертификации, состоящей из шести этапов, по рис. 4.7.

① Пользователь **A** поручает удостоверяющему центру выпустить сертификат на свой открытый ключ.

② Удостоверяющий центр проверяет подлинность пользователя **A**, после чего выпускает сертификат, который состоит из открытого ключа пользователя **A** и цифровой подписи этого открытого ключа, созданной удостоверяющим центром.

③ Удостоверяющий центр сохраняет сертификат пользователя **A** в репозитории (хранилище данных).

④ Пользователь **B** скачивает сертификат пользователя **A** из репозитория.

⑤ Пользователь **B** с помощью открытого ключа удостоверяющего центра восстанавливает (расшифровывает) открытый ключ пользователя **A** из цифровой подписи, содержащийся в сертификате пользователя **A**.

⑥ Пользователь сравнивает открытый ключ пользователя **A**, расшифрованный из цифровой подписи, с открытым ключом пользователя **A**, содержащимся в сертификате в виде открытого текста. Совпадение этих двух ключей является гарантией того, что содержащийся в сертификате открытый ключ действительно принадлежит пользователю **A**.

Благодаря вышеуказанной процедуре пользователь **B** получает открытый ключ, гарантированно принадлежащий пользователю **A**, с помощью которого может проверить подлинность сообщения с электронной подписью, зашифрованной секретным ключом пользователя **A**. Подлинным может считаться сообщение, которое одновременно удовлетворяет трём нижеуказанным условиям.

① Содержащиеся в сообщении данные не были подменены.
② Сообщение не отправлено злоумышленником, выдающим себя за пользователя **A**.
③ Пользователь **A** не имеет возможности отказаться от того факта, что именно он отправил это сообщение пользователю **B**.

Подтверждение подлинности открытого ключа является гарантией того, что сообщение, снабжённое электронной подписью, соответствует условию подлинности сообщения. Вышеописанный механизм лежит в основе инфраструктуры открытых ключей (ИОК, англ. PKI: Public Key Infrastructure), о которой будет рассказано далее.

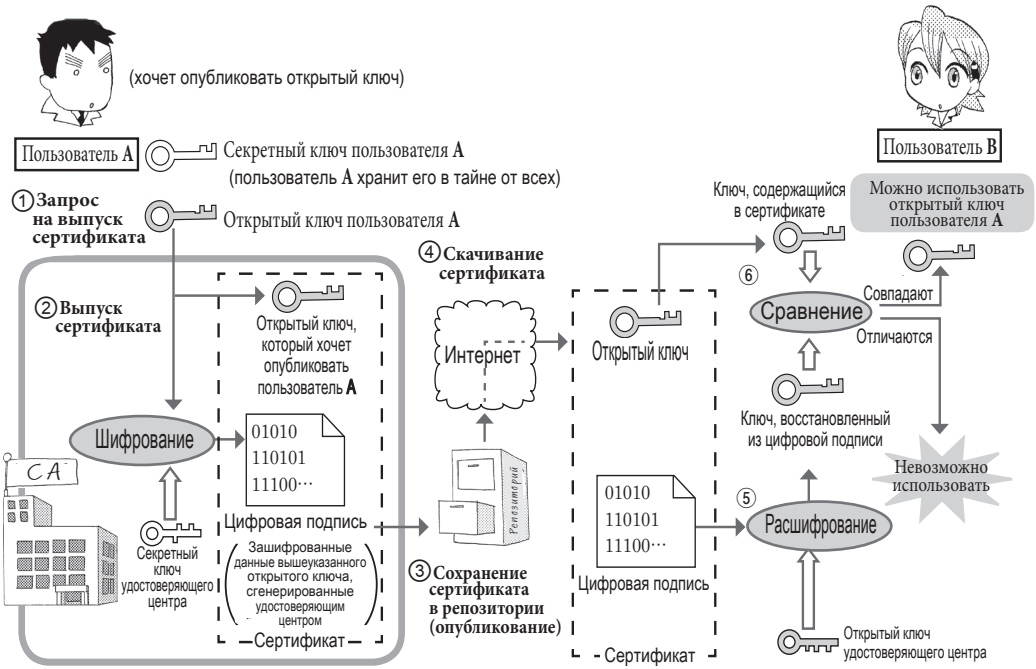
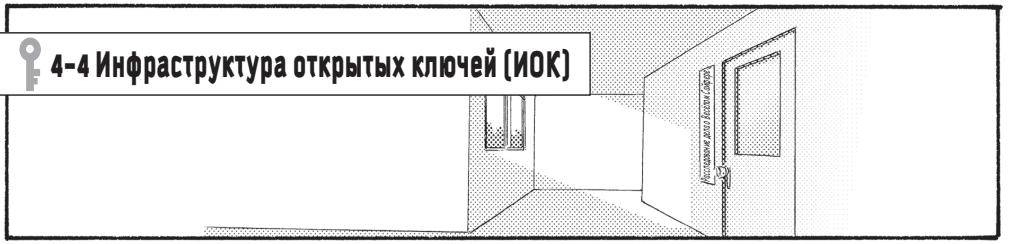
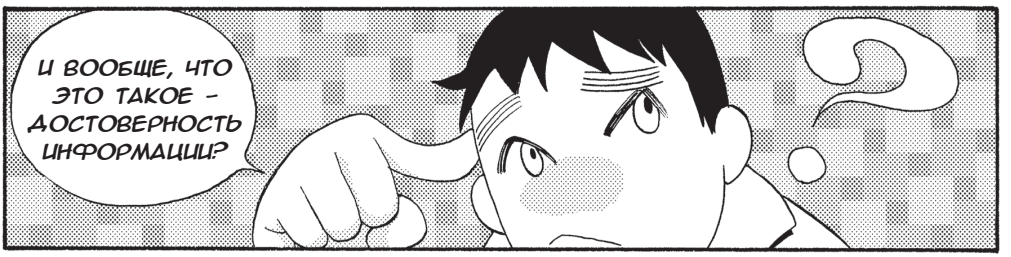
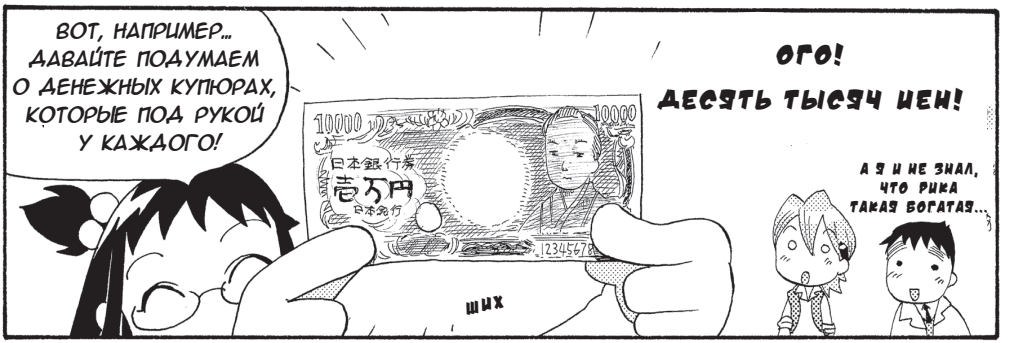


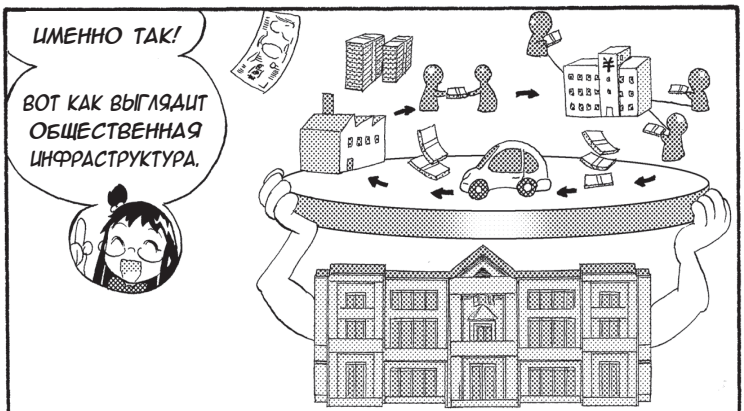
Рис. 4.7. Порядок выпуска сертификата

4-4 Инфраструктура открытых ключей (ИОК)

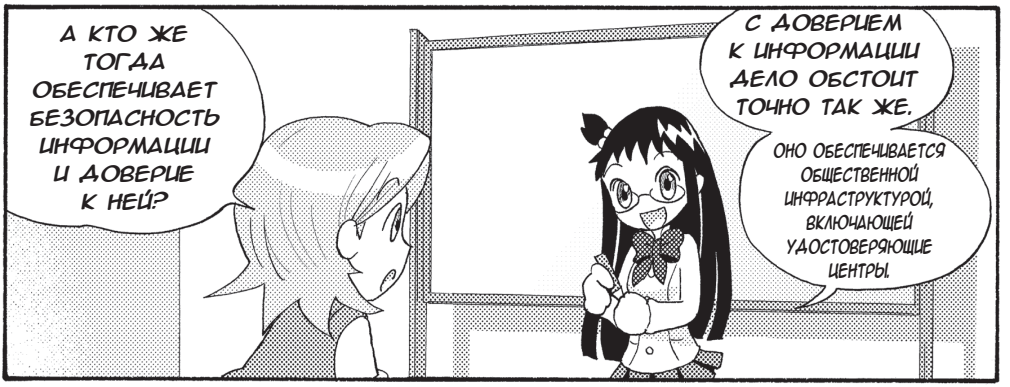






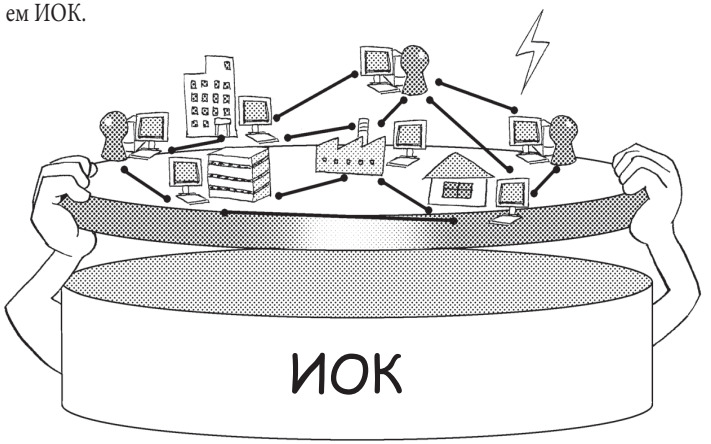


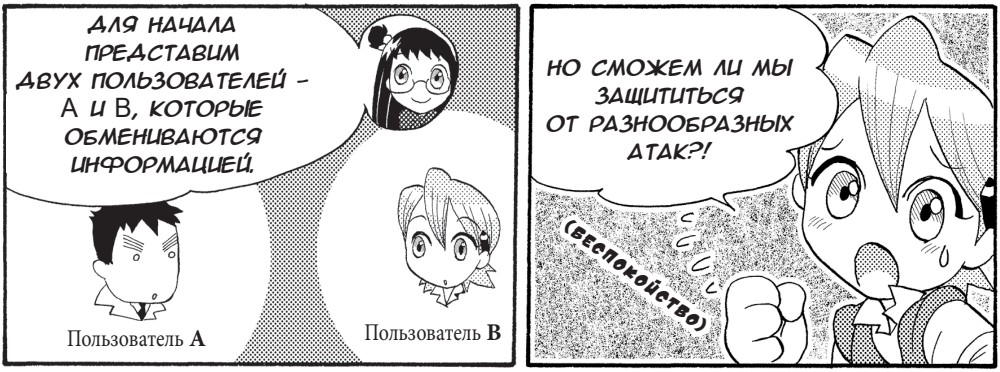
- Купюры выпускает только банк Японии (они не должны печататься кем попало).
- Банк Японии принимает различные меры для того, чтобы ценность денег (бумажных купюр) не снижалась.
- Государство предотвращает подделку купюр и охраняет доверие к ним с помощью законов.



Подобно тому, как безопасность денег и доверие к ним гарантируются разнообразными элементами общественной инфраструктуры, безопасность систем, в которых используется шифр с открытым ключом, и доверие к ним тоже обеспечиваются общественной инфраструктурой под названием ИОК.

Другими словами, мы можем безопасно обмениваться сообщениями электронной почты, совершать торговые сделки через интернет только благодаря существованию общественной инфраструктуры под названием ИОК.





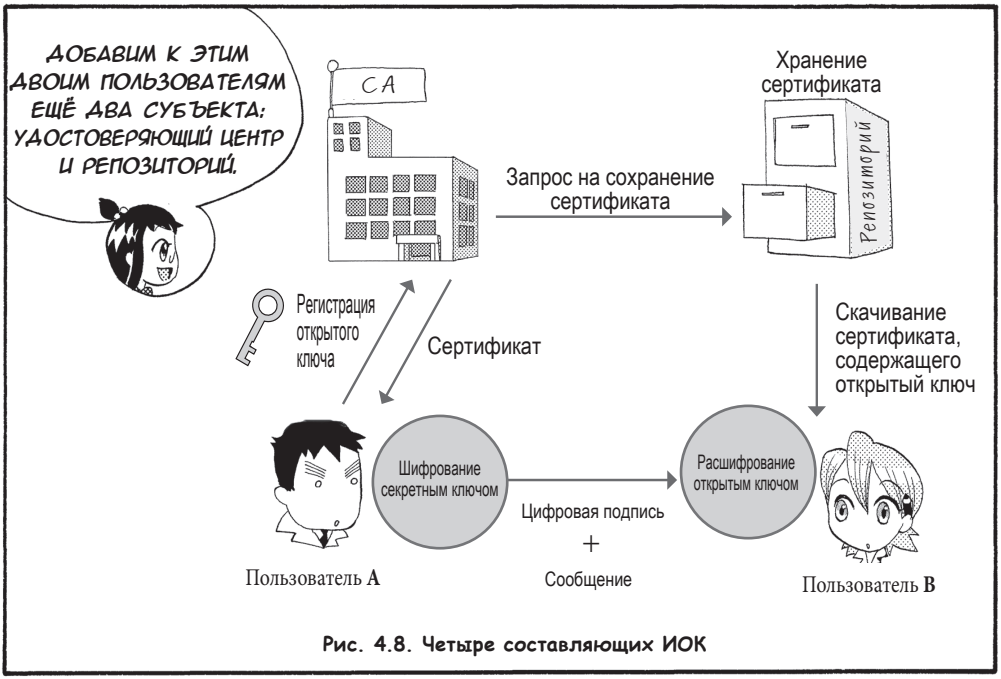
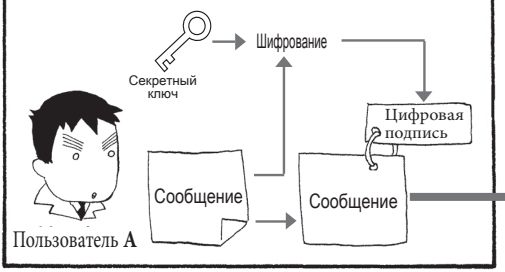


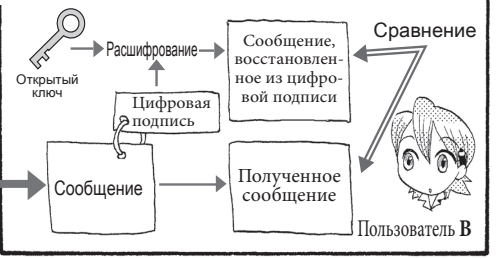
Рис. 4.8. Четыре составляющих ИОК

① Пользователь **B** хочет получить от пользователя **A** сообщение, избежав подмены данных, спуфинга и отказа!

② Пользователь **A** с помощью своего секретного ключа создаёт цифровую подпись и прилагает её к сообщению.



③ Пользователь **B** с помощью открытого ключа пользователя **A** проверяет полученную цифровую подпись, и если она в порядке, то считает сообщение подлинным.

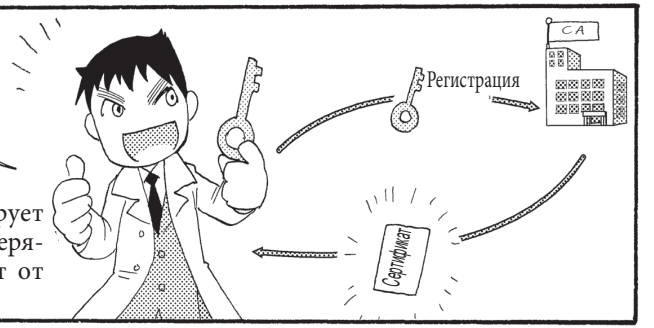


④ ОДНАКО ДЕЙСТВИТЕЛЬНО ЛИ ЭТОТ ОТКРЫТЫЙ КЛЮЧ ПРИНАДЛЕЖИТ ПОЛЬЗОВАТЕЛЮ A?

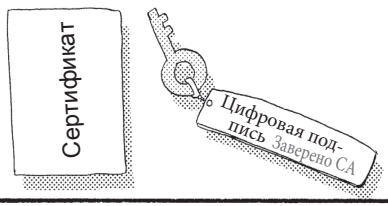
ПОДОЗРИТЕЛЬНО!

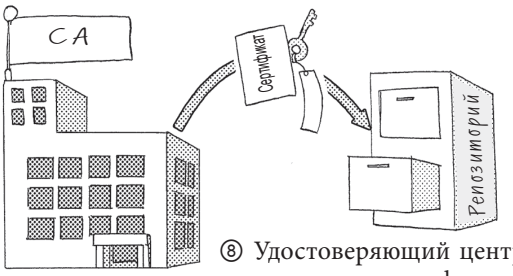
⑤ ПРИДУМАЛ! Я ПОПРОШУ НАДЁЖНЫЙ УДОСТОВЕРЯЮЩИЙ ЦЕНТР, ЧТОБЫ ОН ЗАВЕРИЛ МОЙ ОТКРЫТЫЙ КЛЮЧ!

⑥ Пользователь **A** регистрирует открытый ключ в удостоверяющем центре и получает от него сертификат.



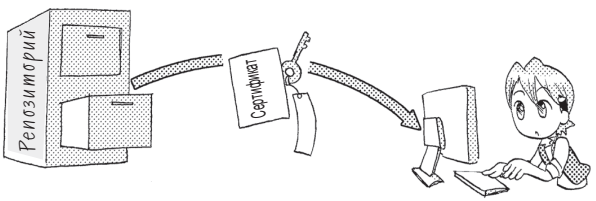
⑦ Сертификат состоит из открытого ключа пользователя **A** и цифровой подписи этого ключа, созданной удостоверяющим центром.



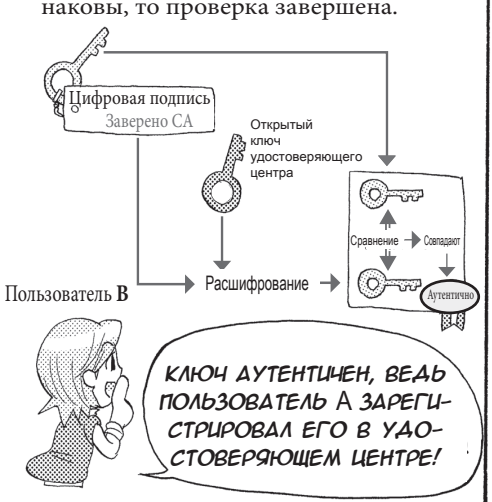


⑧ Удостоверяющий центр сохраняет сертификат в репозитории (хранилище данных).

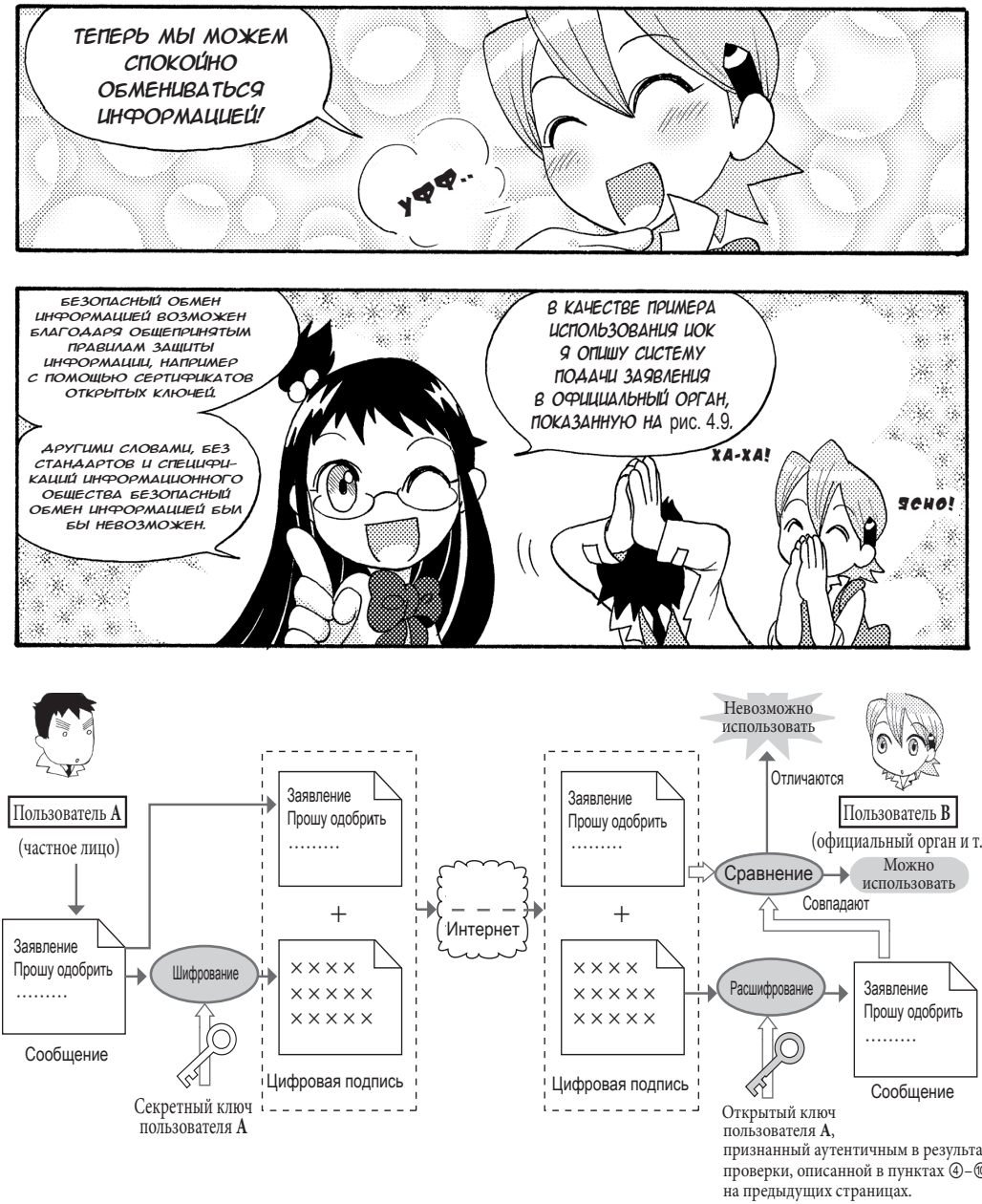
⑨ Пользователь **B** скачивает сертификат пользователя **A** из репозитория.



⑩ Пользователь **B** сравнивает открытый ключ, содержащийся в сертификате пользователя **A**, с открытым ключом, восстановленным путём расшифрования цифровой подписи. Если они одинаковы, то проверка завершена.

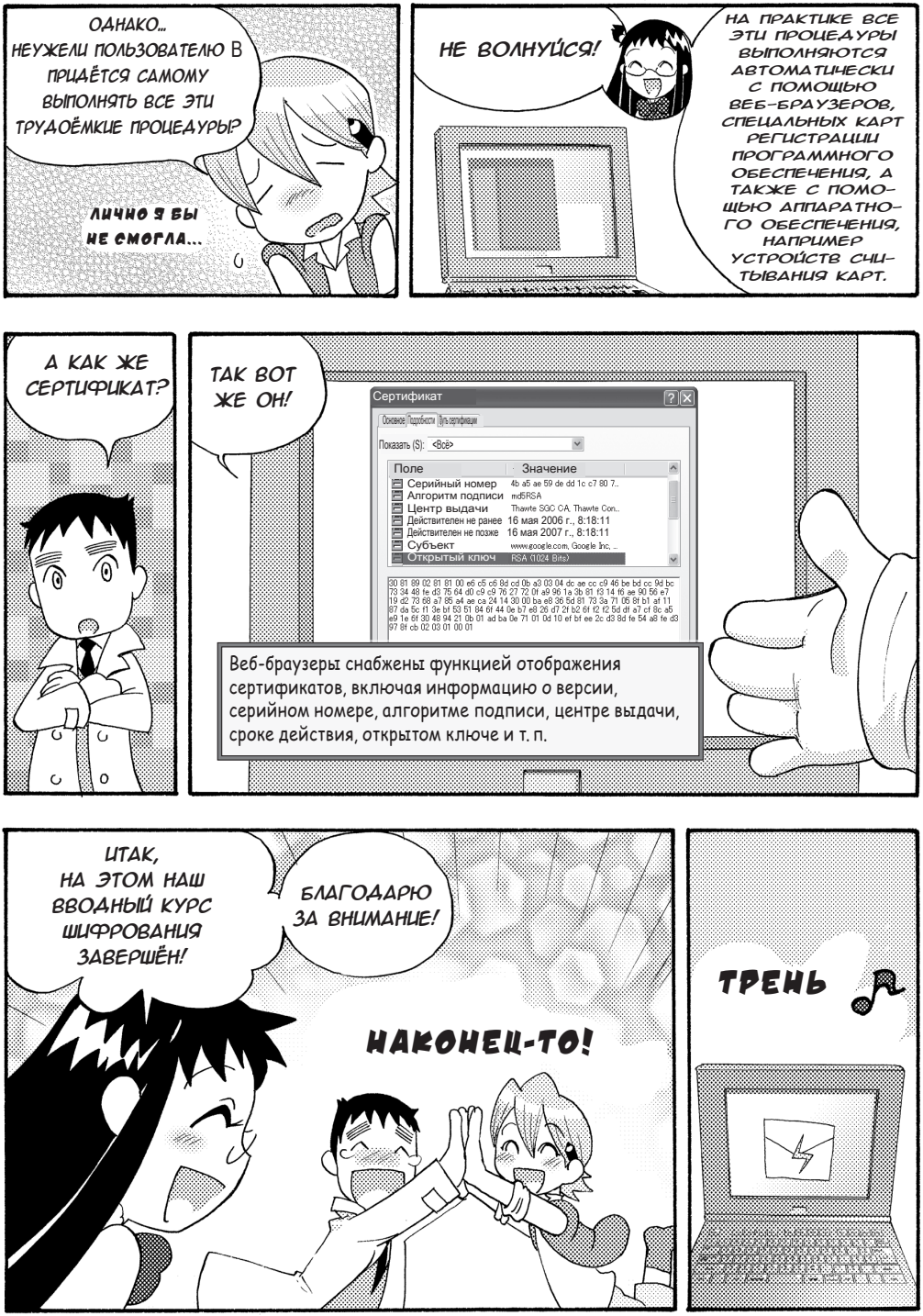


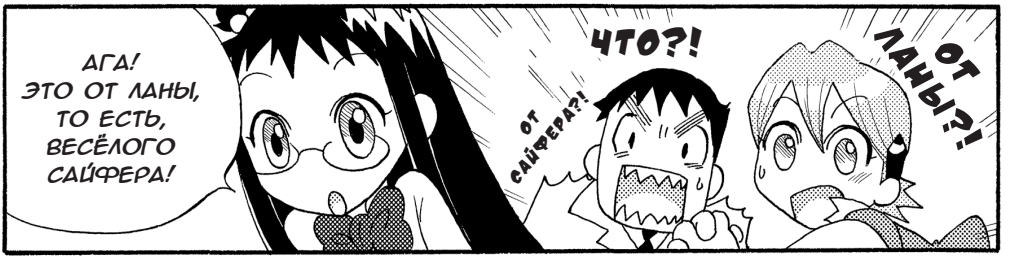
⑪ По завершении проверки пользователь **B** убеждается в аутентичности открытого ключа пользователя **A**, содержащегося в сертификате пользователя **A**, и, следовательно, в целостности и аутентичности сообщения, полученного на шаге ③ (оно не содержит подделок, отправлено аутентичным пользователем без возможности отказа).



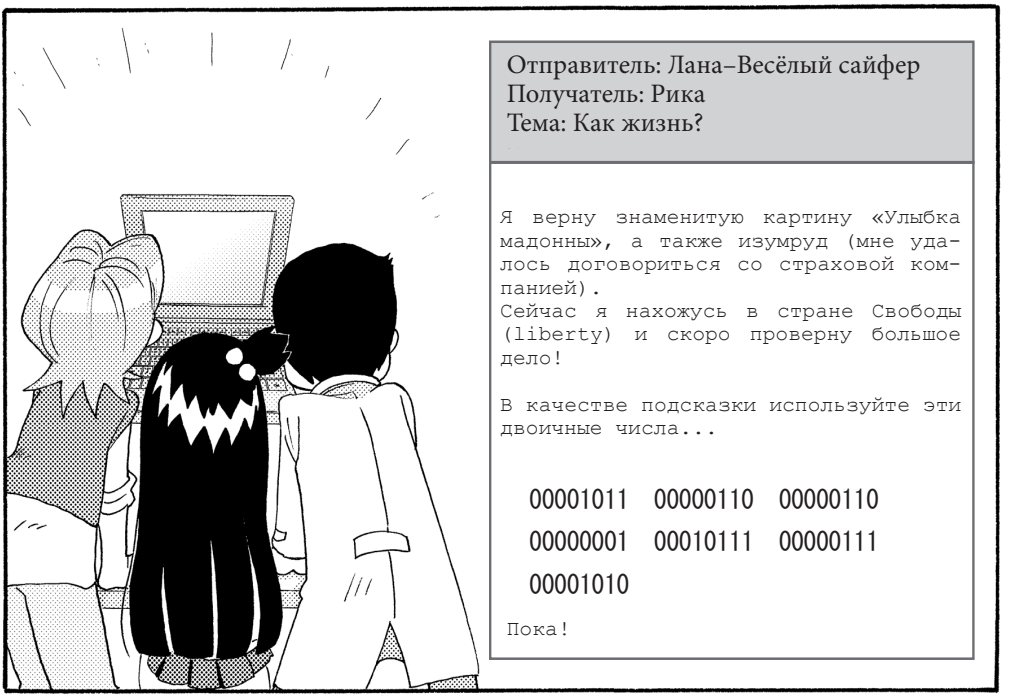
Открытый ключ пользователя **A** сертифицирован, а значит, совпадение двух заявлений: полученного в качестве сообщения и восстановленного путём расшифрования цифровой подписи, – исключает возможность того, что заявление подверглось подмене данных, было отправлено злоумышленником, выдающим себя за пользователя **A**, а также возможность отказа пользователя **A** от факта подачи им этого заявления.

Рис. 4.9. Пример процедуры подачи заявления с помощью ИОК

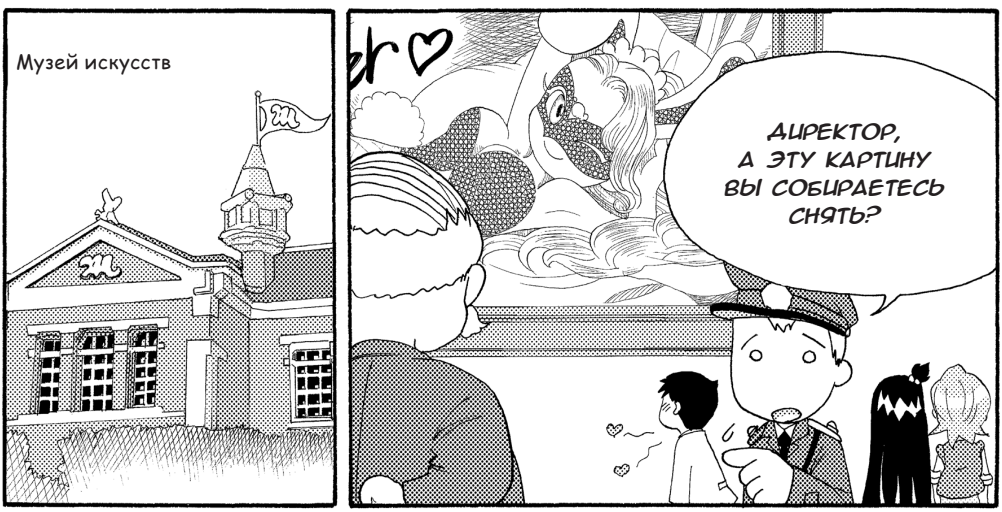

ОДНАКО… НЕУЖЕЛИ ПОЛЬЗОВАТЕЛЮ В ПРИДЁТСЯ САМОМУ ВЫПОЛНЯТЬ ВСЕ ЭТИ ТРУДОЁМКИЕ ПРОЦЕДУРЫ?
ЛИЧНО Я БЫ НЕ СМОГЛА…
НЕ ВОЛНУЙСЯ!
НА ПРАКТИКЕ ВСЕ ЭТИ ПРОЦЕДУРЫ ВЫПОЛНЯЮТСЯ АВТОМАТИЧЕСКИ С ПОМОЩЬЮ ВЕБ-БРАУЗЕРОВ, СПЕЦИАЛЬНЫХ КАРТ РЕГИСТРАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, А ТАКЖЕ С ПОМОЩЬЮ АППАРАТНОГО ОБЕСПЕЧЕНИЯ, НАПРИМЕР УСТРОЙСТВ СЧИТЫВАНИЯ КАРТ.
А КАК ЖЕ СЕРТИФИКАТ?
ТАК ВОТ ЖЕ ОН!
Сертификат
Основное Подробности Путь сертификации
Показать (S): <Все>
Поле
Значение
Серийный номер
Алгоритм подписи
Центр выдачи
Действителен не ранее
Действителен не позже
Субъект
Открытый ключ
Веб-браузеры снабжены функцией отображения сертификатов, включая информацию о версии, серийном номере, алгоритме подписи, центре выдачи, сроке действия, открытом ключе и т. п.
ИТАК, НА ЭТОМ НАШ ВВОДНЫЙ КУРС ШИФРОВАНИЯ ЗАВЕРШЁН!
БЛАГОДАРЮ ЗА ВНИМАНИЕ!
НАКОНЕЦ-ТО!
ТРЕНЬ

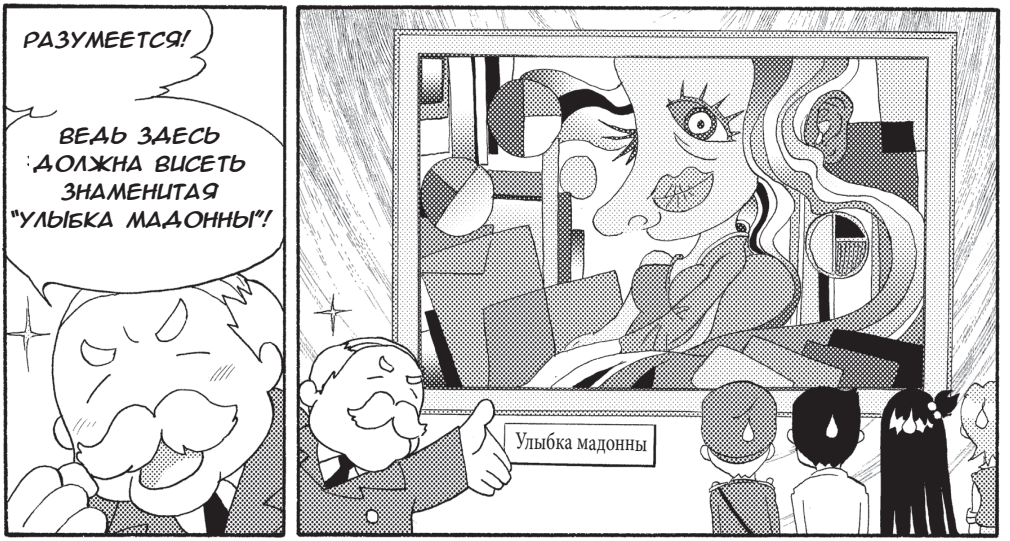


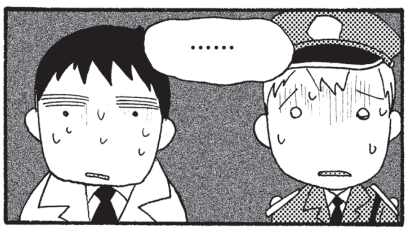
※ Прочитав текст е-мейла на стр. 181 по диагонали из левого верхнего угла, вы узнаете, кто такой Сайфер.



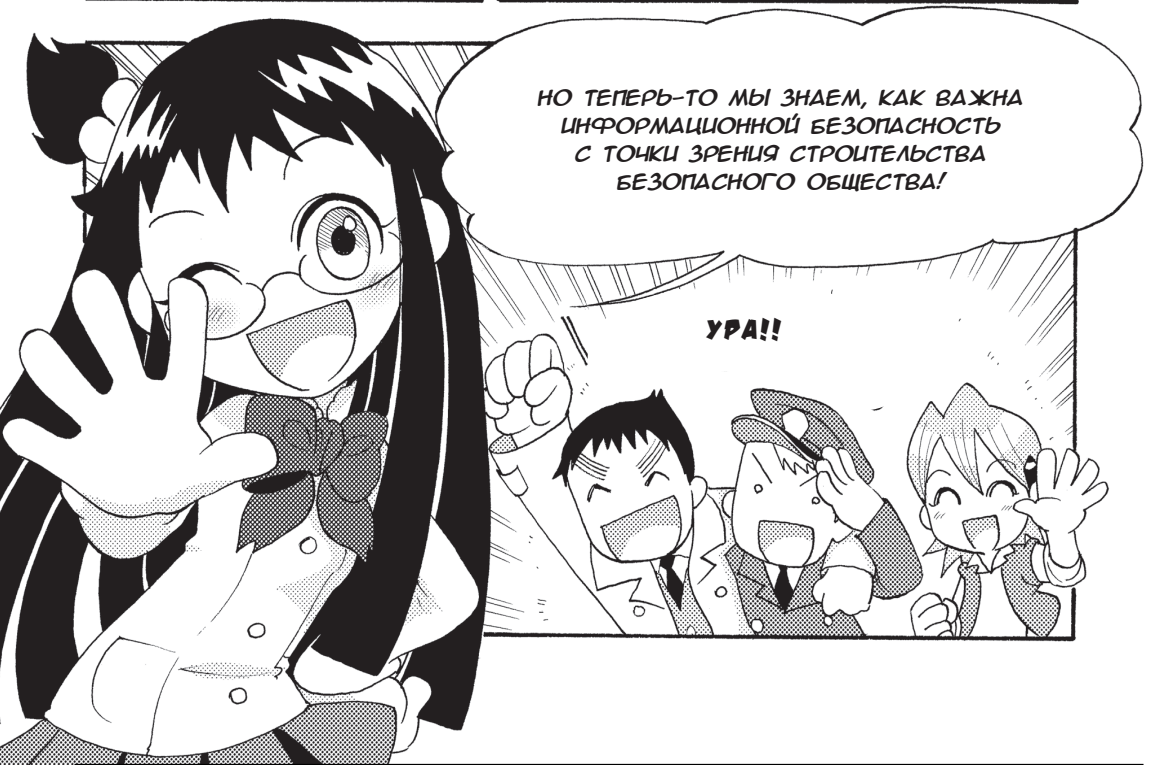
Спустя несколько дней




РАЗУМЕЕТСЯ!
ВЕДЬ ЗДЕСЬ ДОЛЖНА ВИСЕТЬ ЗНАМЕНИТАЯ "УЛЫБКА МАДОННЫ"!
Улыбка мадонны


.....


ИЗУМРУД ТОЖЕ ВЕРНУЛИ, ТАК ЧТО НА ЭТОТ РАЗ ВСЁ ОБОШЛОСЬ!
ЗДОРОВО!
КАК ХОРОШО!
ПОЗДРАВЛЯЮ!
БРАТ!


НО ТЕПЕРЬ-ТО МЫ ЗНАЕМ, КАК ВАЖНА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬ С ТОЧКИ ЗРЕНИЯ СТРОИТЕЛЬСТВА БЕЗОПАСНОГО ОБЩЕСТВА!
УРА!!

Дополнительная информация

Доказательство с нулевым разглашением

В последние годы время от времени происходят случаи, когда при оплате кредитной картой злоумышленники крадут информацию, после чего обладатель карты получает счета на оплату товаров или услуг, которых он никогда не приобретал. Таким образом, при разглашении личной информации с целью идентификации личности всегда существует опасность её кражи. В связи с этим появилась необходимость в методе идентификации личности (аутентификации карты) без разглашения личной информации.

В ответ на эту потребность в 1985 году Гольдвассер, Микали и Реккоф разработали концепцию под названием доказательство с нулевым разглашением, целью которой является доказательство аутентичности своей карты партнёру (кредитной организации) без разглашения секретной информации самой карты (например, десятичного случайного числа из более чем сотни цифр, используемого в качестве пароля). Хотя, на первый взгляд, может показаться странным, когда кто-то просит поверить ему на слово, что он знает некое секретное число, однако, используя магию чисел, основанную на строгой криптографической математике, эту концепцию можно воплотить в жизнь.

Здесь мы расскажем о методе реализации доказательства с нулевым разглашением, разделив его на подготовительный этап и этап выполнения.

● Подготовительный этап

Самым первым шагом в реализации числовой магии доказательства с нулевым разглашением является создание заслуживающего доверия центра (проверяющего лица). Опишем это на конкретном примере.

① Центр опубликовывает составное число *N*

Центр выбирает два простых числа (p, q) и, перемножив их, получает составное число N, которое затем публикует.

$$N = pq \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

Числа p и q центр держит в секрете. Хотя на практике используются огромные простые числа, состоящие из порядка 80 знаков, здесь, чтобы не усложнять пример, мы используем $p = 13$ и $q = 19$.

$N = 13 \times 19 = 247$

Таким образом, мы получили трёхзначное составное число. (На практике используемые простые числа настолько большие, что вычислительная сложность не позволяет разложить N на простые множители даже на самом мощном компьютере.)

② Центр регистрирует идентификаторы (ID) всех пользователей

Идентификатор (ID) – это число, публикуемое каждым из пользователей (соответствует открытому ключу пользователя), причём каждый пользователь име-

ет только один ID (соответствует открытому ключу пользователя). Другими словами, ID позволяет идентифицировать каждого из пользователей. Идентификаторы всех пользователей зарегистрированы в центре.

③ Центр вычисляет секретные ключи и передаёт их всем пользователям

Центр вычисляет квадратный корень по модулю N из каждого зарегистрированного ID. Извлечение квадратного корня на множестве рациональных чисел – очень простая операция, однако на множестве целых чисел эта операция будет простой только в том случае, если известны простые множители p и q числа N, и числовая магия под названием доказательство с нулевым разглашением основана именно на вычислительной сложности этой операции.

В нашем случае, так как простые числа – 13 и 19, – известны только центру, другими словами, только он может вычислить квадратные корни из зарегистрированных ID всех пользователей, поэтому утечки секретной личной информации не произойдёт. Положим, что идентификатор пользователя **A** (ID_A) равен 101. Тогда квадратный корень будет равен 71.

$$\sqrt{101} \pmod{247} \equiv 71$$

Естественно, обратная операция – возведение числа 71 в квадрат, – даст нам число 101: $71^2 \pmod{247} \equiv 101$. Число 71 – это секретный ключ S_A пользователя **A**, который центр тайком передаёт пользователю **A**. (На практике это – число из не менее чем 100 знаков, не предназначенное для запоминания пользователем **A**.) Идентификатор (ID_A) и секретный ключ (S_A) пользователя **A** связаны между собой следующими соотношениями:

$$\sqrt{ID_A} \pmod{N} \equiv S_A \quad \cdots\cdots (2)$$

$$(S_A)^2 \pmod{N} \equiv ID_A \quad \cdots\cdots (3)$$

Так как назначение секретного ключа S_A – не идентификация самого пользователя **A**, а аутентификация его банковской карты, пользователю не нужно запоминать число S_A, в отличие, например, от пин-кода карты. Аналогичным образом центр раздаёт секретные ключи всем зарегистрированным пользователям.

Этап выполнения (порядок аутентификации)

Опишем порядок процедуры аутенификации в том случае, когда пользователь **A** хочет доказать пользователю **B** свою аутентичность (другими слвами, подлинность своей карты).

Шаг 1: Запрос на аутентификацию от пользователя A пользователю B (1)

Сначала пользователь **A** должен выбрать случайное число r_A, возвести его в квадрат и поделить результат на составное число N, чтобы найти остаток. Этот остаток, то есть число y_A, удовлетворяющее следующему сравнению:

$$y_A \equiv (r_A)^2 \pmod{N} \quad \cdots\cdots (4)$$

пользователь **A** отправляет пользователю **B**. Положим, что в качестве случайного числа r_A пользователь **A** выбрал 50. Тогда, решив сравнение:

$$y_A = 50^2 = 2500 \equiv 30 \pmod{247}$$

пользователь **A** отправит пользователю **B** число 30.

Шаг 2: Запрос на аутентификацию от пользователя A пользователю B (2)

Теперь пользователь **A** умножает по модулю составного числа N свой секретный ключ, полученный от центра, на случайное число r_A, выбранное на шаге 1. Другими словами, пользователь **A** решает следующее сравнение:

$$z_A \equiv S_A r_A \pmod{N} \quad \cdots\cdots (5)$$

и отправляет пользователю **B** результат z_A. В нашем случае, так как выбрано 50:

$$z_A = 71 \times 50 \equiv 92 \pmod{247}$$

пользователь **A** отправляет пользователю **B** число 92.

Шаг 3: Аутентификация пользователя A пользователем B (1)

Пользователь **B** возводит число z_A, полученное от пользователя **A**, в квадрат по модулю составного числа N, или, говоря другими словами, решает следующее сравнение:

$$v_A \equiv (z_A)^2 \pmod{N} \quad \cdots\cdots (6)$$
$$\equiv (S_A r_A)^2 \pmod{N} \quad \cdots\cdots (7)$$

Так как, в нашем случае $z_A = 92$, он получит следующий результат:

$$v_A = 92^2 = 8464 \equiv 66 \pmod{247}.$$

Шаг 4: Аутентификация пользователя A пользователем B (2)

Теперь пользователь **B** находит результат ω_A деления числа υ_A, найденного на шаге 3, на число y_A, полученное от пользователя **A** на шаге 1, по модулю составного числа N, то есть, решает следующее сравнение:

$$w_A \equiv \frac{v_A}{y_A} \pmod{N} \quad \cdots\cdots (8)$$
$$\equiv v_A \times (y_A^{-1}) \pmod{N} \quad \cdots\cdots (9)$$

Естественно, так как все вычисления производятся по модулю N, для выполнения деления на y_A необходимо сначала найти обратный элемент (обратное число) y_A^{-1} к числу y_A. Другими словами, число y_A^{-1} должно удовлетворять следующему сравнению:

$$y_A \times (y_A^{-1}) \equiv 1 \pmod{N} \quad \cdots\cdots (10)$$

В нашем примере, так как $\upsilon_A = 66$, $y_A = 30$, $y_A^{-1} \equiv 30^{-1} \pmod{247} \equiv 140$:

$$w_A \equiv \frac{66}{30} \pmod{247} \equiv 66 \times 30^{-1} \pmod{247} \equiv 66 \times 140 \pmod{247} \equiv 101.$$

Таким образом, пользователь **B** нашёл число 101, равное идентификатору пользователя **A** (ID_A).

Как мы смогли увидеть, вышеописанная процедура по шагам 1–4 позволяет пользователю **B** убедиться в аутентичности пользователя **A** в том случае, если отправителем является подлинный пользователь **A**. Причина заключается в том, что секретный ключ S_A пользователя **A** при возведении в квадрат даёт идентификатор пользователя **A** (ID_A). Другими словами, на основе сравнения (8) с учётом сравнений (3), (4) и (5) мы можем записать:

$$\omega_A = \frac{\{(\text{Секретный ключ } S_A \text{ пользователя } \mathbf{A}) \times (\text{Случайное число } r_A)\}^2}{(\text{Случайное число } r_A)^2}$$

$$= \frac{(S_A r_A)^2}{(r_A)^2} = (S_A)^2 = ID_A = \text{Идентификатор пользователя } \mathbf{A} \qquad (12)$$

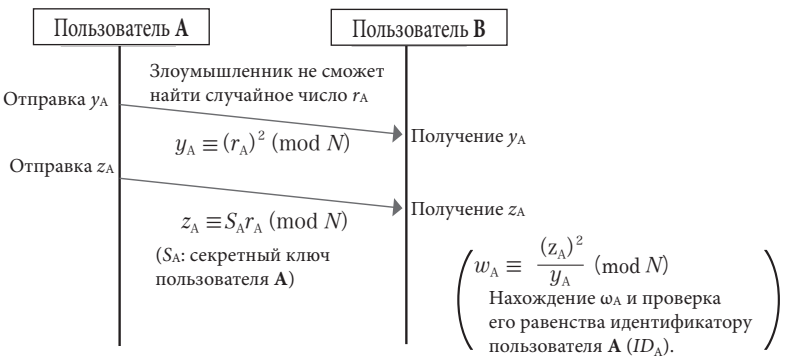


Рис. 4.10. Метод аутентификации на основе доказательства с нулевым разглашением

Метод спуфинга

Теперь рассмотрим способ, с помощью которого злоумышленник **X**, используя процедуру аутентификации, показанную на рис. 4.10, может выдать себя за пользователя **A** с идентификатором (ID_A) 101. Разумеется, злоумышленник не имеет ни малейшего представления о секретном ключе S_A пользователя **A**.

Однако, выбрав два числа *e* и *f*, удовлетворяющие следующему сравнению:

$$e^2 \equiv ID_A \times f \pmod{247} \quad \cdots\cdots (13)$$

злоумышленник **X** на шаге 1 процедуры аутентификации отправляет число f, а затем, на шаге 2, число e.

Пусть это будут, к примеру, числа $e = 25$ и $f = 82$, удовлетворяющие сравнению (13). Покажем, что при выполнении шагов 3 и 4 с этими числами будет вычислен идентификатор пользователя **A** ($ID_A = 101$).

Используя эту пару чисел e и f, произвольно выбранную злоумышленником **X**, вместо числа z_A сравнения (5) и числа y_A сравнения (4), соответственно, вычислим сравнения (6) и (9) и убедимся, что полученный результат будет равен идентификатору, опубликованному пользователем A ($ID_A = 101$).

Все вычисления производятся по модулю составного числа N (в нашем примере $N = 13 \times 19 = 247$).

Шаг 3

$$v_A = e^2 = 25^2 \equiv 131 \pmod{247}.$$

Шаг 4

$$w_A = \frac{e^2}{f} = e^2 \times f^{-1} = v_A \times 82^{-1}$$

Здесь, так как $82^{-1} \pmod{247} \equiv 244$: $\omega_A \equiv 131 \times 244 \pmod{247} \equiv 101$.

Мы показали, что, не обладая знанием секретного ключа S_A и случайного числа r_A, злоумышленник **X** способен воссоздать идентификатор $ID_A = 101$, опубликованный пользователем **A**, и, таким образом, совершать недобросовестные действия, выдавая себя за пользователя **A** (рис. 4.11).

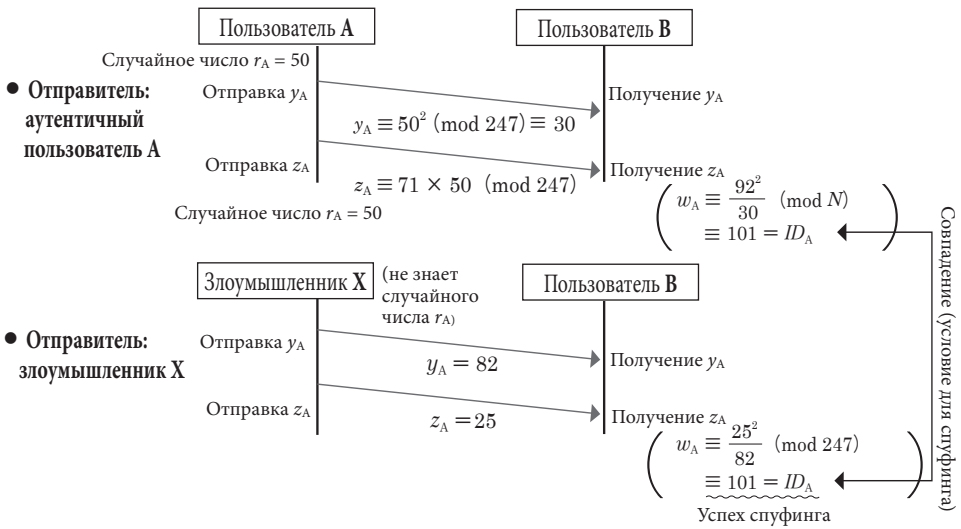


Рис. 4.11. Пример спуфинга

Таким образом, для прохождения процедуры аутентфикации, показанной на рис. 4.10, достаточно на основе данных ID, публикуемых каждым из пользователей, выбрать числа, удовлетворяющие сравнению (13). Конкретно говоря, злоумышленник **X**, не знающий секретного ключа S_A пользователя **A**, сначала произвольно выбирает число e и возводит его в квадрат, а затем, поделив результат на ID_A, то есть решив нижеприведённое сравнение (14), получает число f:

$$f \equiv \frac{e^2}{ID_A} \pmod N \quad \cdots\cdots\cdots\cdots \quad (14)$$

после чего сначала отправляет число f, а затем – число e. Таким образом, злоумышленник **X**, даже не зная случайного числа r_A, известного только пользователю **A**, может легко пройти процедуру аутентификации, осуществляемую пользователем **B** (шаги 3 и 4).

Защита от спуфинга, основанная на доказательстве без разглашения

Итак, как же можно защититься от такого спуфинга? Для этого необходимо усложнить процедуру аутентификации, как показано на рис. 4.12. Пользователь **B** после получения от отправителя каждого из чисел y_A и z_A, отправляет бит запроса (challenge bit), равный 0 или 1, проверяя значения, присылаемые в ответ. Это даёт возможность проверить, правильно ли отправитель выполняет процедуру аутентификации.

Данная методика гарантирует доказательство без разглашения и аутентичность, предотвращает спуфинг, конечно, если злоумышленнику не известны первоначальные простые числа.

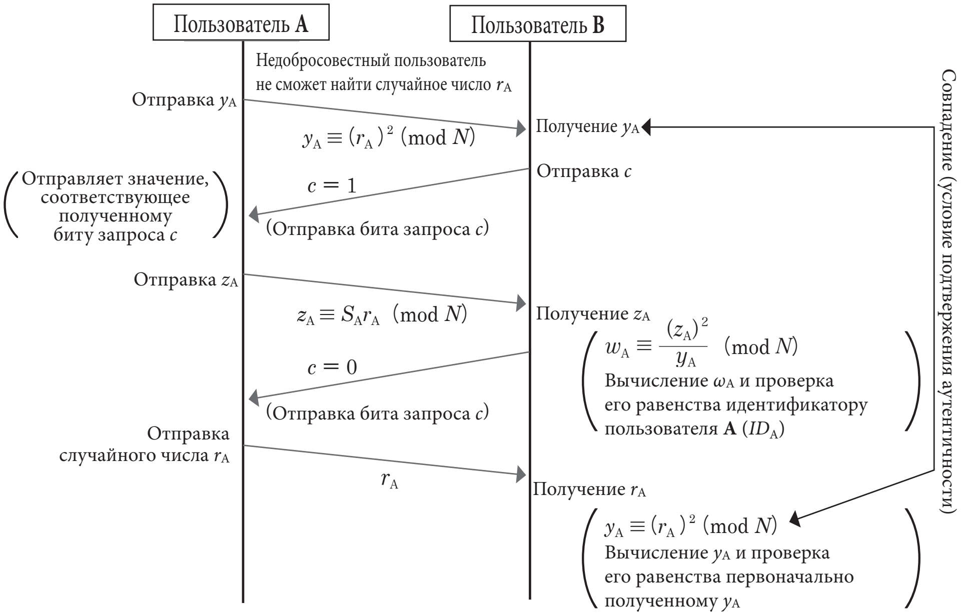


Рис. 4.12. Процедура аутентификации с защитой от спуфинга

Разъяснение некоторых терминов

◎ Псевдослучайные числа и стойкость шифра

Случайные числа, то есть беспорядочные последовательности чисел, – это одна из фундаментальных технологий, обеспечивающих стойкость шифра. Например, ключи в криптосистемах с открытым ключом, используемые для шифрования и расшифрования информации, генерируются с использованием случайных чисел. Так как повторное использование одних и тех же ключей повышает вероятность вскрытия шифра, в целях повышения криптостойкости ключи каждый раз генерируются заново при возникновении необходимости использовать открытый ключ. И это является необходимой мерой, ведь если злоумышленник узнает, какое случайное число используется, то это может привести к финансовым потерям, утечке личной информации. Таким образом, для того чтобы шифр был невскрываем, активно используют такое свойство случайных чисел, как непредсказуемость (невозможность на основе предудущей последовательности предсказать следующее число).

Существуют псевдослучайные числа и истинные случайные числа. Псевдослучайные числа генерируются на основе определённых формул и не обладают таким свойством непредсказуемости, как истинно случайные числа: последовательность псевдослучайных чисел характеризуется периодичностью, в ней могут присутствовать определённые закономерности, что повышает риск вычисления псевдослучайного числа злоумышленником и снижает стойкость шифра. Наиболее типичными методами генерации псевдослучайных чисел являются линейный конгруэнтный метод, метод середины квадрата, М-последовательность, алгоритм Блюм-Блюма-Шуба (англ. BBS: Blum-Blum-Shub); методы с использованием односторонних функций, методы с использованеим шифров и т. п.

С другой стороны, генераторы истинных случайных чисел, основанные на использовании физических явлений, способны всё время выдавать беспорядочные, то есть действительно случайные, последовательности чисел. Для генерации истинно случайных чисел может использоваться, например, шум, создаваемый электрическим током в полупроводниковых элементах. Считается, что в будущем истинно случайные числа будут использоваться в криптографии более широко, что позволит создать прочный фундамент, поддерживающий информационную безопасность.

◎ PGP

PGP – это широко используемая компьютерная криптографическая программа, созданная в 1991 году Филиппом Циммерманом. Её название – PGP, – образовано из первых букв фразы Pretty Good Privacy, что дословно переводится как «весьма надёжная конфиденциальность».

PGP поддерживает практически все функции, необходимые для современного криптографического программного обеспечения. Это, другими словами, одноключевые шифры (AES, 3-DES и др.), шифры с открытым ключом (RSA, шифр Эль-Гамаля и др.), односторонние хеш-функции (MD5, SHA-1, RIPEMD-160 и др.), генерирование сертификатов и т. д.

◎ SSL/TLS

SSL/TLS – это семейство криптографических протоколов безопасной связи, используемых, например, в интернет-шоппинге. В этих протоколах для проверки аутентичности и целостности передаваемых сообщений используется имитовставка (MAC: код аутентификации сообщения). Протоколы SSL (Secure Socket Layer:) или TLS (Transport Layer Security), выполняющие шифрование сообщений, используются, например, в веб-браузерах для таких операций, как отправка номера кредитной карты, что помогает предотвратить перехват этой важной личной информации. Кстати, адреса URL интернет-ресурсов, использующих эти протоколы связи, начинаются не с «http://», а с «https://».

Кроме того, SSL/TLS можно использовать для криптографической защиты и таких протоколов связи, как SMTP (Simple Mail Transfer Protocol), используемый для отправки сообщений электронной почты, POP3 (Post Office Protocol), используемый для получения сообщений электронной почты.

◎ Квантовый шифр

Считается совершенно стойким шифром. При использовании обычной оптоволоконной связи в одном световом импульсе, который соответствует 1 биту информации, содержится более 10 тысяч минимальных порций (квантов) световой энергии (называемых фотонами), однако в квантовом шифре каждый фотон модулируется одним битом информации путём изменения состояния его поляризации (то есть направления векторов электрической и магнитной составляющих электромагнитной волны). Так как фотон является неделимой элементарной частицей, в случае перехвата информации состояние поляризации фотонов изменится, а значит, факт компрометации информации скрыть не удастся. Таким образом, методы квантовой механики могут обеспечить принципиальную невозможность незаметного перехвата информации. Если добавить к этому свойству квантового шифра ещё и принципиальную невозможность расшифрования, основанную на использовании одноразовых ключей шифрования из шифровального блокнота (one time pad), то можно будет создать шифр, обладающий совершенной стойкостью, и в настоящее время ведутся активные работы в этом направлении.

◎ Биометрическая аутентификация

Для биометрической аутентификации используются такие специфические личные данные, как отпечатки пальцев, рисунок вен, черты лица, радужная оболочка глаза, форма ладони, ДНК (гены) и т. п. В качестве знакомых всем примеров можно привести банкоматы, а также системы контроля и управления доступом (СКУД), устанавливаемые на входе в помещения, в которых аутентификация личности производится по рисунку вен, считываемому с пальца или ладони.

Список использованной литературы

- Митани Масааки, «Индустриальная математика, помогающая начать сначала – Информация, связь и анализ сигналов – Шифрование, код коррекции ошибок, интегральное преобразование», изд. CQ, 2000 г.
- Юки Хироси, «Введение в криптографию», изд. SB Creative, 2003 г.
- Цудзии Сигэо, «Криптография и информационное общество», изд. Bungeishunju Ltd., 1999 г.
- Окамото Тацуаки, Ямамото Хиросукэ, «Современная криптография», изд. San-to, 1997 г.
- Ито Масафуми, «Теория криптографии», изд. Natsumesha CO. LTD., 2003 г.
- Вакабаяси Хироси, «Основы и устройство современных шифров простым языком», изд. SHUWA SYSTEM CO., LTD., 2005 г.
- Simon Singh, The code book, пер. Аоки Каору, изд. SHINCHOSHA, 2001 г.
- Sarah Flannery, David Flannery, «16-летняя Сара бросает вызов самому стойкому в мире шифру», пер.Камэи Ёсико, изд. NHK, 2001 г.
- Цудзии Сигэо, «Шифры – информационная безопасность в постмодернистском обществе», изд. Kodansha, 1996 г.
- Хитоцумацу Син, «Математика шифров», изд. Kodansha, 2005 г.
- Joseph Silverman, Friendly Introduction to Number Theory, пер. Судзуки Дзиро, изд. Pearson Education, 2001 г.
- Benedict Gross, Joe Harris, The Magic of Numbers, пер. Судзуки Дзиро, изд. Pearson Education, 2005 г.
- Цуру Кодзи, «Изучаем криптографические технологии с помощью Excel», изд. Ohmsha, 2006 г.
- Иота Дзэми, ред. Дзимбо Масакадзу, «Шифры – это просто!», изд. Ohmsha, 2004 г.

Предметный указатель

▪ Об авторах

Митани Масааки

Доктор технических наук. Родился в р-не Сэтода-тё г. Омити преф. Хиросима. В 1948 г. закончил кафедру электроники технологического факультета токийского технологического института. После работы научным сотрудником технологического факультета токийского технологического института вступил в должность профессора кафедры информации и связи технологического факультета токийского университета электротехники. Специализируется в области обработки сигналов, связи, а также образовательных технологий.

<Основные работы>

«Вводный курс обработки цифровых сигналов», изд. Ohmsha
«Изучаем обработку цифровых сигналов в Scilab», изд. CQ
«Азы электронных схем простым языком (I): диоды, транизисторы», изд. Micronet
«Математические основы обработки сигналов, помогающие начать сначала», изд. CQ
«Используем преобразование Фурье начиная с сегодняшего дня!», изд. Kodansha
и многие другие.

Сато Синъити

Родился в г. Датэ преф. Фукусима. В 1990 г. окончил магистратуру токийского университета электротехники по специальности «Электротехника». Занимался проектированием аппаратуры обработки изображений для частных компаний. В качестве научного сотрудника медицинского факультета частного университета принимал участие в исследованиях, связанных с разработкой биомедицинской техники. В настоящее время является научным сотрудником кафедры информации и связи технологического факультета токийского университета электротехники. Специализируется в области обработки цифровых сигналов, а также образовательных технологий.

▪ **Оформление манги/** Verte Corp.

▪ **Редактирование манги/** Эндо Цугуми

▪ **DTP/** Араи Сатоси

Сложив по модулю 2
двоичные числа подсказки:

00001011 00000110 00000110 00000001 00010111 00000111 00001010

со словом liberty, означающим свободу:

01101100 01101001 01100010 01100101 01110010 01110100 01111001

получим ряд двоичных чисел:

01100111 01101111 01100100 01100100 01100101 01110011 01110011

который в кодировке JIS записывается словом goddess, то есть «богиня».
(Статуя Свободы дословно переводится с японского как «Богиня Свободы»).

Книги издательства «ДМК Пресс» можно заказать
в торгово-издательском холдинге «Планета Альянс» наложенным платежом,
выслав открытку или письмо по почтовому адресу:
115487, г. Москва, 2-й Нагатинский пр-д, д. 6А.
При оформлении заказа следует указать адрес (полностью), по которому
должны быть высланы книги; фамилию, имя и отчество получателя.
Желательно также указать свой телефон и электронный адрес.
Эти книги вы можете заказать и в интернет-магазине: **www.a-planeta.ru**.
Оптовые закупки: тел. **(499) 782-38-89**.
Электронный адрес: **books@alians-kniga.ru**.

Митани Масааки, Сато Синъити (авторы), Хиноки Идэро (художник)

Занимательная информатика
Криптография
Манга

Главный редактор *Д. А. Мовчан*
dmkpress@gmail.com
Перевод с японского *А. Б. Клионский*
Научный редактор *Д. М. Белявский*
Корректор *Г. И. Синяева*
Верстальщик *А. Р. Арифулин*

Формат 70×100 1/16.
Гарнитура Anime Ace. Печать офсетная.
Усл. п. л. 19,5. Тираж 500 экз.

Веб-сайт издательства ДМК Пресс: www.dmkpress.com