## От автора

Эта книга - краткое введение в криптографию. С одной стороны, здесь изложен материал, который отвечает на многие вопросы, которые возникают у тех кто делает на ниве этой науке первые шаг, с другой стороны здесь есть тот минимум информации, который достаточен для того чтобы самостоятельно оценивать любые реальные криптосистемы или даже создавать свои собственные.

Язык книги делался по возможности доступным, но не освобождает Читателя от необходимости владения элементарными основами математики, в частности алгебры и теории групп и полей.

Многие вопросы к сожалению остались за обложками этой книги. В частности после долгих сомнений Автор решил отказаться от рассмотрения DES, ввиду его крайней непрактичности и неуживчивости на российской почве[[1]](#footnote-1).

Массу полезной информации можно найти на сервере ftp.rsa.com. В faq5.doc Вы если и не найдете ответ на любой вопрос по криптографии, то обнаружите большое количество ссылок на другие источники.

Автор будет признателен за любые замечания и вопросы, которые проще всего направить по адресу: bar@glasnet.ru

*Баричев Сергей*

## Введение

Про­бле­ма за­щи­ты ин­фор­ма­ции пу­тем ее пре­об­ра­зо­ва­ния, исключающего ее про­чте­ние по­сто­рон­ним ли­цом вол­но­ва­ла че­ло­ве­че­ский ум с дав­них вре­мен. История криптографии - ровесница истории человеческого языка. Более того, первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные. Священные книги Древ­него Егип­та, Древ­ней Индии тому примеры.

С широким распространением письменности криптография стала формироваться как самостоятельная наука. Первые криптосистемы встречаются уже в начале нашей эры. Так, Цезарь в своей переписке использовал уже более менее систематический шифр, получивший его имя.

Бурное раз­ви­тие крип­то­гра­фи­че­ские сис­те­мы по­лу­чи­ли в го­ды пер­вой и вто­рой ми­ро­вых войн. Начиная с послевоенного времени и по нынешний день появление вычислительных средств ускорило разработку и совершенствование криптографических методов.

По­че­му про­бле­ма ис­поль­зо­ва­ния крип­то­гра­фи­че­ских ме­то­дов в информационных системах (ИС) ста­ла в на­стоя­щий мо­мент осо­бо ак­ту­аль­на?

С од­ной сто­ро­ны, рас­ши­ри­лось ис­поль­зо­ва­ние ком­пь­ю­тер­ных се­тей, в частности глобальной сети Интернет, по ко­то­рым пе­ре­да­ют­ся боль­шие объ­е­мы ин­фор­ма­ции го­су­дар­ст­вен­но­го, во­ен­но­го, ком­мер­че­ско­го и ча­ст­но­го ха­рак­те­ра, не до­пус­каю­ще­го воз­мож­ность дос­ту­па к ней по­сто­рон­них лиц.

С дру­гой сто­ро­ны, по­яв­ле­ние но­вых мощ­ных ком­пь­ю­те­ров, тех­но­ло­гий се­те­вых и ней­рон­ных вы­чис­ле­ний сде­ла­ло воз­мож­ным дис­кре­ди­та­цию криптографических сис­тем еще не­дав­но счи­тав­ших­ся прак­ти­че­ски не раскрываемыми.

Про­бле­мой защиты информации путем ее преобразования за­ни­ма­ет­ся *крип­то­ло­гия* (*kr*y*p*to*s* - тай­ный, lo*g*o*s* - нау­ка). Криптология раз­де­ля­ет­ся на два на­прав­ле­ния - *крип­то­гра­фию* и *крип­тоа­на­лиз*. Це­ли этих на­прав­ле­ний прямо про­ти­во­по­лож­ны.

*Крип­то­гра­фия* за­ни­ма­ет­ся по­ис­ком и ис­сле­до­ва­ни­ем ма­те­ма­ти­че­ских ме­то­дов пре­об­ра­зо­ва­ния ин­фор­ма­ции.

Сфе­ра ин­те­ре­сов *криптоанализа* - ис­сле­до­ва­ние воз­мож­но­сти рас­шиф­ро­вы­ва­ния ин­фор­ма­ции без зна­ния клю­чей.

В этой книге ос­нов­ное вни­ма­ние бу­дет уде­ле­но крип­то­гра­фи­че­ским ме­то­дам.

Современная криптография включает в себя четыре крупных раздела:

1. Симметричные криптосистемы.
2. Криптосистемы с открытым ключом.
3. Системы электронной подписи.
4. Управление ключами.

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

### Тер­ми­но­ло­гия

Итак, криптография дает возможность преобразовать информацию таким образом, что ее прочтение (восстановление) возможно только при знании ключа.

В качестве информации, подлежащей шифрованию и дешифрованию, будут рассматриваться *тексты*, построенные на некотором *алфавите*. Под этими терминами понимается следующее.

*Алфавит* - конечное множество используемых для кодирования информации знаков.

*Текст* - упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных ИС можно привести следующие:

1. алфавит Z33 - 32 буквы русского алфавита и пробел;
2. алфавит Z256 - символы, входящие в стандартные коды ASCII и КОИ-8;
3. бинарный алфавит - Z2 = {0,1};
4. восьмеричный алфавит или шестнадцатеричный алфавит;

*Шиф­ро­ва­ние* - пре­об­ра­зо­ва­тель­ный про­цесс: *ис­ход­ный текст*, ко­то­рый но­сит так­же на­зва­ние *от­кры­то­го тек­ста*, за­ме­ня­ет­ся *шиф­ро­ван­ным тек­стом*.

Криптографическая система

исходный

текст

шифрованный

текст

КЛЮЧ

*Дешифрование* - обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный.

шифрованный

текст

исходный

текст

Криптографическая система

КЛЮЧ

*Ключ -* ин­фор­ма­ция, не­об­хо­ди­мая для бес­пре­пят­ст­вен­но­го шиф­ро­ва­ния и де­шиф­ро­ва­ния тек­стов.

*Крип­то­гра­фи­че­ская сис­те­ма* пред­став­ля­ет со­бой се­мей­ст­во **T** пре­об­ра­зо­ва­ний от­кры­то­го тек­ста. Чле­ны это­го се­мей­ст­ва ин­дек­си­ру­ют­ся, или обо­зна­ча­ют­ся сим­во­лом *k*; па­ра­метр *k* яв­ля­ет­ся *клю­чом*. Про­стран­ст­во клю­чей *K* - это на­бор воз­мож­ных зна­че­ний клю­ча. Обыч­но ключ пред­став­ля­ет со­бой по­сле­до­ва­тель­ный ряд букв ал­фа­ви­та.

Криптосистемы разделяются на *симметричные* и *с открытым ключом*.

В *симметричных криптосистемах* и для шифрования, и для дешифрования используется *один и тот же ключ*.

В *системах с открытым ключом* используются два ключа - *открытый* и *закрытый*, которые математически связаны друг с другом. Информация шифруется с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения.

Тер­ми­ны *рас­пре­де­ле­ние клю­чей* и *управ­ле­ние клю­ча­ми* от­но­сят­ся к про­цес­сам сис­те­мы об­ра­бот­ки ин­фор­ма­ции, со­дер­жа­ни­ем ко­то­рых яв­ля­ет­ся со­став­ле­ние и рас­пре­де­ле­ние клю­чей ме­ж­ду поль­зо­ва­те­ля­ми.

*Электронной (цифровой) подписью* называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

*Крип­то­стой­ко­стью* на­зы­ва­ет­ся ха­рак­те­ри­сти­ка шиф­ра, оп­ре­де­ляю­щая его стой­кость к де­шиф­ро­ва­нию без зна­ния клю­ча (т.е. крип­тоа­на­ли­зу). Имеется несколько показателей криптостойкости, среди которых:

1. количество всех возможных ключей;
2. среднее время, необходимое для криптоанализа.

Пре­об­ра­зо­ва­ние T*k* оп­ре­де­ля­ет­ся со­от­вет­ст­вую­щим ал­го­рит­мом и зна­че­ни­ем па­ра­мет­ра *k*. Эф­фек­тив­ность шиф­ро­ва­ния с це­лью за­щи­ты ин­фор­ма­ции за­ви­сит от со­хра­не­ния тай­ны клю­ча и криптостойкости шифра.

### Требования к криптосистемам

Про­цесс крип­то­гра­фи­че­ско­го за­кры­тия данных мо­жет осу­ще­ст­в­лять­ся как про­грамм­но, так и аппаратно. Ап­па­рат­ная реа­ли­за­ция от­ли­ча­ет­ся су­ще­ст­вен­но боль­шей стои­мо­стью, од­на­ко ей при­су­щи и пре­иму­ще­ст­ва: вы­со­кая про­из­во­ди­тель­ность, про­сто­та, за­щи­щен­ность и т.д. Про­грамм­ная реа­ли­за­ция бо­лее прак­тич­на, до­пус­ка­ет из­вест­ную гиб­кость в ис­поль­зо­ва­нии.

Для со­вре­мен­ных крип­то­гра­фи­че­ских сис­тем за­щи­ты ин­фор­ма­ции сфор­му­ли­ро­ва­ны сле­дую­щие об­ще­при­ня­тые тре­бо­ва­ния:

1. за­шиф­ро­ван­ное сообщение дол­жно под­да­вать­ся чте­нию толь­ко при на­ли­чии клю­ча;
2. чис­ло опе­ра­ций, не­об­хо­ди­мых для оп­ре­де­ле­ния ис­поль­зо­ван­но­го клю­ча шиф­ро­ва­ния по фраг­мен­ту шиф­ро­ван­но­го сообщения и со­от­вет­ст­вую­ще­го ему от­кры­то­го тек­ста, долж­но быть не мень­ше об­ще­го чис­ла воз­мож­ных клю­чей;
3. чис­ло опе­ра­ций, не­об­хо­ди­мых для рас­шиф­ро­вы­ва­ния ин­фор­ма­ции пу­тем пе­ре­бо­ра все­воз­мож­ных ключей долж­но иметь стро­гую ниж­нюю оцен­ку и вы­хо­дить за пре­де­лы воз­мож­но­стей со­вре­мен­ных ком­пь­ю­те­ров (с учетом возможности использования сетевых вычислений);
4. зна­ние ал­го­рит­ма шиф­ро­ва­ния не долж­но вли­ять на на­деж­ность за­щи­ты;
5. не­зна­чи­тель­ное из­ме­не­ние клю­ча долж­но при­во­дить к су­ще­ст­вен­но­му из­ме­не­нию ви­да за­шиф­ро­ван­но­го сообщения да­же при ис­поль­зо­ва­нии од­но­го и то­го же клю­ча;
6. струк­тур­ные эле­мен­ты ал­го­рит­ма шиф­ро­ва­ния долж­ны быть не­из­мен­ны­ми;
7. до­пол­ни­тель­ные би­ты, вво­ди­мые в сообщение в про­цес­се шиф­ро­ва­ния, должен быть пол­но­стью и на­деж­но скры­ты в шиф­ро­ван­ном тек­сте;
8. дли­на шиф­ро­ван­но­го тек­ста долж­на быть рав­ной дли­не ис­ход­но­го тек­ста;
9. не долж­но быть про­стых и лег­ко ус­та­нав­ли­вае­мых зависимостью ме­ж­ду клю­ча­ми, по­сле­до­ва­тель­но ис­поль­зуе­мы­ми в про­цес­се шиф­ро­ва­ния;
10. лю­бой ключ из мно­же­ст­ва возможных дол­жен обес­пе­чи­вать на­деж­ную за­щи­ту ин­фор­ма­ции;
11. ал­го­ритм должен до­пус­кать как про­грамм­ную, так и ап­па­рат­ную реа­ли­за­цию, при этом из­ме­не­ние длины к­лю­ча не долж­но вес­ти к ка­че­ст­вен­но­му ухуд­ше­нию ал­го­рит­ма шифрования.

## Симметричные криптосистемы

Все мно­го­об­ра­зие су­ще­ст­вую­щих крип­то­гра­фи­че­ских ме­то­дов мож­но све­сти к сле­дую­щим клас­сам пре­об­ра­зо­ва­ний:

*Гаммирование*

*Подстановки*

Симметричные

криптосистемы

*Блочные шифры*

*Перестановки*

*Мо­но- и мно­го­ал­фа­вит­ные под­ста­нов­ки.*

Наи­бо­лее про­стой вид пре­об­ра­зо­ва­ний, за­клю­чаю­щий­ся в за­ме­не сим­во­лов ис­ход­но­го тек­ста на другие (того же алфавита) по бо­лее или ме­нее слож­но­му пра­ви­лу. Для обес­пе­че­ния вы­со­кой крип­то­стой­ко­сти тре­бу­ет­ся ис­поль­зо­ва­ние боль­ших клю­чей.

*Пе­ре­ста­нов­ки.*

Так­же не­слож­ный ме­тод крип­то­гра­фи­че­ско­го пре­об­ра­зо­ва­ния. Ис­поль­зу­ет­ся как пра­ви­ло в со­че­та­нии с дру­ги­ми ме­то­да­ми.

*Гам­ми­ро­ва­ние.*

Этот ме­тод за­клю­ча­ет­ся в на­ло­же­нии на ис­ход­ный текст не­ко­то­рой псев­до­слу­чай­ной по­сле­до­ва­тель­но­сти, ге­не­ри­руе­мой на ос­но­ве клю­ча.

*Блочные шифры.*

Пред­став­ля­ют со­бой по­сле­до­ва­тель­ность (с воз­мож­ным по­вто­ре­ни­ем и че­ре­до­ва­ни­ем) ос­нов­ных ме­то­дов пре­об­ра­зо­ва­ния, при­ме­няе­мую к блоку (части) шиф­руе­мого­ тек­ста. Блочные шифры на прак­ти­ке встре­ча­ют­ся ча­ще, чем “чис­тые” пре­об­ра­зо­ва­ния то­го или ино­го клас­са в си­лу их бо­лее вы­со­кой крип­то­стой­ко­сти. Рос­сий­ский и аме­ри­кан­ский стан­дар­ты шиф­ро­ва­ния ос­но­ва­ны имен­но на этом классе шифров.

### Перестановки

*Перестановкой* σ набора целых чисел (0,1,...,N-1) называется его переупорядочение. Для того чтобы показать, что целое i пере­мещено из позиции i в позицию σ(i), где 0 ≤ (i) < *n*, будем использовать запись

σ=(σ(0), σ(1),..., σ(N-1)).

Число перестановок из (0,1,...,N-1) равно *n*!=1\*2\*...\*(N-1)\*N. Введем обозначение σ для взаимно-однозначного отображения (гомо­морфизма) набора S={*s*0,*s*1, ...,*s*N-1}, состоящего из *n* элементов, на себя.

σ: S → S

σ: *s*i → *s*σ(i), 0 ≤ i < *n*

Будем говорить, что в этом смысле σ является *перестановкой элементов* S. И, наоборот, автоморфизм S соответствует пере­становке целых чисел (0,1,2,.., *n*-1).

*Криптографическим преобразованием* T для алфавита Zm называется последовательность автоморфизмов: T={T(n):1≤n<∞}

T(n): Zm,n→Zm,n, 1≤n<∞

Каждое T(n) является, таким образом, перестановкой *n*-грамм из Zm,n.

Поскольку T(i) и T(j) могут быть определены независимо при i≠j, число криптографических преобразований исходного текста размерности *n* равно (mn)![[2]](#footnote-2). Оно возрастает непропорционально при увеличении *m* и *n*: так, при *m*=33 и *n*=2 число различных криптографических преобразований равно 1089!. Отсюда следует, что потенциально существует большое число отображений исходного текста в шифрованный.

Практическая реализация криптогра­фических систем требует, чтобы преобразо­вания {T*k*: *k*∈*K*} были определены алгоритмами, зависящими от относительно небольшого числа параметров (ключей).

### Сис­те­мы под­ста­но­вок

*Определение* *Подстановкой* π на алфавите Zm называется автоморфизм Zm, при котором буквы исходного текста t замещены буквами шифрованного текста π(t):

Zm 🡪 Zm; π: t 🡪 π(t).

Набор всех подстановок называется симметрической группой Zm è будет в дальнейшем обозначаться как SYM(Zm).

*Утверждение* SYM(Zm) c операцией произведения является группой, т.е. операцией, обладающей следующими свойствами:

1. *Замкнутость*: произведение подстановок π1π2 является подста­новкой:

π: t🡪π1(π2(t)).

1. *Ассоциативность*: результат произведения π1π2π3 не зависит от порядка расстановки скобок:

(π1π2)π3=π1(π2π3)

1. *Существование нейтрального элемента*: постановка i, опре­деляемая как i(t)=t, 0≤t<m, является нейтральным элементом SYM(Zm) по операции умножения: iπ=πi для ∀π∈SYM(Zm).
2. *Существование обратного*: для любой подстановки π существует единственная обратная подстановка π-1, удовлетворя­ющая условию

ππ‑1=π‑1π=i.

Число возможных подстановок в симметрической группе Zm называется *порядком* SYM(Zm) и равно *m*! .

*Определение. Ключом* подстановки *k* для Zm называется последовательность элементов симметрической группы Zm:

*k*=(*p*0,*p*1,...,*p*n-1,...), *p*n∈SYM(Zm), 0≤n<∞

Подстановка, определяемая ключом *k*, является крипто­гра­фи­ческим преобразованием T*k*, при помощи которого осуществляется преоб­разование *n*-граммы исходного текста (x0 ,x1 ,..,xn-1) в *n*-грамму шифрованного текста (y0 ,y1 ,...,yn-1):

yi=*p*(xi), 0≤i<n

где *n* – произвольное (n=1,2,..). T*k* называется моноалфавитной под­ста­новкой, если *p* неизменно при любом i, i=0,1,..., в противном случае T*k* называется многоалфавитной подстановкой.

*Примечание*. К наиболее существенным особенностям подста­новки T*k* относятся следующие:

*1. Исходный текст шифруется посимвольно*. Шифрования *n*-граммы (x0 ,x1 ,..,xn-1) и ее префикса (x0 ,x1 ,..,x*s*-1) связаны соотношениями

T*k*(x0 ,x1 ,..,xn-1)=(y0 ,y1 ,...,yn-1)

T*k*(x0 ,x1 ,..,x*s*-1)=(y0 ,y1 ,...,y*s*-1)

2. *Буква шифрованного текста yi является функцией только i-й компоненты ключа pi и i-й буквы исходного текста x*i.

#### Подстановка Цезаря

Подстановка Цезаря является самым простым вариантом подстановки. Она относится к группе *моноалфавитных подстановок*.

*Определение*. Подмножество Cm={C*k*: 0≤*k*<m} симметрической группы SYM(Zm), содержащее *m* подстановок

C*k*: j→(j+*k*) (mod *m*), 0≤*k* < *m*,

называется подстановкой Цезаря.

Умножение коммутативно, C*k*Cj=CjC*k*=Cj+*k*, C0 – идентичная подстановка, а обратной к Cк является C*k*-1=Cm-*k*, где 0<*k*<m. Семейство подстановок Цезаря названо по имени римского императора Гая Юлия Цезаря, который поручал Марку Туллию Цицерону составлять послания с использованием 50-буквенного алфавита и подстановки C3.

Подстановка определяется по таблице замещения, содержащей пары соответствующих букв “исходный текст – шифрованный текст”. Для C3 подстановки приведены в Табл. 1. Стрелка (🡪) означает, что буква исходного текста (слева) шифруется при помощи C3 в букву шифрованного текста (справа).

*Определение***.** *Системой Цезаря* называется моноалфа­витная подстановка, преобразующая *n*-грамму исходного текста (x0, *x*1 ,..,xn-1) в *n*‑грамму шифрованного текста (y0 ,y1 ,...,yn-1) в соответствии с правилом

yi=C*k*(xi), 0≤i<n.

Например, ВЫШЛИТЕ\_НОВЫЕ\_УКАЗАНИЯ посредством подстановки C3 преобразуется в еюыолхиврсеюивцнгкгрлб.

*Таблица 1*.

|  |  |  |  |
| --- | --- | --- | --- |
| А🡪г | Й🡪м | Т🡪х | Ы🡪ю |
| Б🡪д | К🡪н | У🡪ц | Ь🡪я |
| В🡪е | Л🡪о | Ф🡪ч | Э🡪\_ |
| Г🡪ж | М🡪п | Х🡪ш | Ю🡪а |
| Д🡪з | Н🡪р | Ц🡪щ | Я🡪б |
| Е🡪и | О🡪с | Ч🡪ъ | \_🡪в |
| Ж🡪й | П🡪т | Ш🡪ы |  |
| З🡪к | Р🡪у | Щ🡪ь |  |
| И🡪л | С🡪ф | Ъ🡪э |  |

При своей несложности система легко уязвима. Если злоумышленник имеет

1) шифрованный и соответ­ствующий исходный текст или

2) шифрованный текст выбранного злоумыш­ленником исходного текста,

то определение ключа и дешифрование исходного текста тривиальны.

Более эффективны обобщения подстановки Цезаря - *шифр Хилла* и *шифр Плэйфера*. Они основаны на подстановке не отдельных символов, а 2-грамм (шифр Плэйфера) или *n*-грамм[[3]](#footnote-3) (шифр Хилла). При более высокой криптостойкости они значительно сложнее для реализации и требуют достаточно большого количества ключевой информации.

#### Многоалфавитные системы. Системы одноразового использования.

Слабая криптостойкость моноалфавитных подстановок преодолевается с применением подстановок многоалфавитных.

*Многоалфавитная подстановка* определяется ключом π=(π1,
π2, ...), содержащим не менее двух различных подстановок. В начале рассмотрим многоалфавитные системы подстановок с нулевым начальным смещением.

Пусть {*K*i: 0≤i<n} - независимые случайные переменные с одинаковым распределением вероятностей, принимающие значения на множестве Zm

*P*кл{(*K*0, *K*1, ..., *K*n-1)=(*k*0, *k*1, ..., *k*n-1)}=(1/m)n

*Система одноразового использования* преобразует исходный текст

X=(X0, *x*1, ..., *x*n-1)

в шифрованный текст

Y=(Y0, *y*1, ..., *y*n-1)

при помощи подстановки Цезаря

Yi=C*K*i(xi)=(*K*i+Xi) (mod *m*) i=0...n-1 (1)

Для такой системы подстановки используют также термин “одноразовая лента” и “одноразовый блокнот”. Пространство ключей К системы одноразовой подстановки является вектором рангов (*K*0, *K*1, ..., *K*n-1) и содержит *m*n точек.

Рассмотрим небольшой пример шифрования с бесконечным ключом. В качестве ключа примем текст

 “БЕСКОНЕЧНЫЙ\_КЛЮЧ....”.

Зашифруем с его помощью текст “ШИФР\_НЕРАСКРЫВАЕМ”. Шифрование оформим в таблицу:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ШИФРУЕМЫЙ\_ТЕКСТ | 24 | 8 | 20 | 16 | 19 | 5 | 12 | 27 | 9 | 32 | 18 | 5 | 10 | 17 | 18 |
| БЕСКОНЕЧНЫЙ\_КЛЮЧ | 1 | 5 | 17 | 10 | 14 | 13 | 5 | 23 | 13 | 27 | 9 | 32 | 10 | 11 | 30 |
| ЩРДЪАТТССЦЪЫДФЬП | 25 | 13 | 4 | 26 | 0 | 18 | 17 | 17 | 22 | 26 | 27 | 4 | 20 | 28 | 15 |

Исходный текст невозможно восстановить без ключа.

Наложение белого шума в виде бесконечного ключа на исходный текст меняет статистические характеристики языка источника. Системы одноразового использования *теоретически не расшифруемы[[4]](#footnote-4)*, так как не содержат достаточной информации для восстановления текста.

Почему же эти системы неприменимы для обеспечения секретности при обработке информации? Ответ простой - они непрактичны, так как требуют независимого выбора значения ключа для каждой буквы исходного текста. Хотя такое требование может быть и не слишком трудным при передаче по прямому кабелю Москва - Нью-Йорк, но для информационных оно непосильно, поскольку там придется шифровать многие миллионы знаков.

Посмотрим, что получится, если ослабить требование шифровать каждую букву исходного текста отдельным значением ключа.

#### Системы шифрования Вижинера

Начнем с конечной последовательности ключа

 *k* = (*k*0 ,*k*1 ,...,*k*n),

которая называется *ключом пользователя*, и продлим ее до бесконечной последовательности, повторяя цепочку. Таким образом, получим *рабочий ключ*

*k* = (*k*0 ,*k*1 ,...,*k*n), *k*j = *k*(jmod *r*, 0 ≤ j < ∞ .

Например, при *r* = ∞ и ключе пользователя 15 8 2 10 11 4 18 рабочий ключ будет периодической последовательностью:

15 8 2 10 11 4 18 15 8 2 10 11 4 18 15 8 2 10 11 4 18 ...

*Определение. Подстановка Вижинера* VI*Gk* определяется как

 VI*Gk* : (x0, *x*1, ..., *x*n-1) → (y0, *y*1, ..., *y*n-1) = (x0+*k*, *x*1+*k*,. .., *x*n-1+*k*).

Таким образом:

1) исходный текст *x* делится на *r* *фрагментов*

 *x*i = (xi , *x*i+*r* , ..., *x*i+*r*(n-1)), 0 ≤ i < *r*;

2) i-й фрагмент исходного текста *x*i шифруется при помощи подстановки Цезаря C*k* :

(xi , *x*i+*r* , ..., *x*i+*r*(n-1)) → (yi , *y*i+*r* , ..., *y*i+*r*(n-1)),

Вариант системы подстановок Вижинера при *m*=2 называется *системой Вернама (1917 г)*.

В то время ключ *k*=(*k*0 ,*k*1 ,...,*k*к-1) записывался на бумажной ленте. Каждая буква исходного текста в алфавите, расширенном некоторыми дополнительными знаками, сначала переводилась с использованием *кода Бодо* в пятибитовый символ. К исходному тексту Бодо добавлялся ключ (по модулю 2). Старинный телетайп фирмы AT&T со считывающим устройством Вернама и оборудованием для шифрования, использовался корпусом связи армии США.

Очень распространена плохая с точки зрения секретности *практика использовать слово или фразу в качестве ключа* для того, чтобы *k*=(*k*0 ,*k*1 ,...,*k*к-1) было легко запомнить. В ИС для обеспечения безопасности информации это недопустимо. Для получения ключей должны использоваться программные или аппаратные средства случайной генерации ключей.

*Пример. Преобразование текста с помощью подстановки Вижинера (r=4)*

Исходный текст (ИТ1):

НЕ\_СЛЕДУЕТ\_ВЫБИРАТЬ\_НЕСЛУЧАЙНЫЙ\_КЛЮЧ

Ключ: КЛЮЧ

Разобьем исходный текст на блоки по 4 символа:

НЕ\_С ЛЕДУ ЕТ\_В ЫБИР АТЬ\_ НЕСЛ УЧАЙ НЫЙ\_ КЛЮЧ

и наложим на них ключ (используя таблицу Вижинера):

H+К=Ч, Е+Л=Р и т.д.

Получаем зашифрованный (ЗТ1) текст:

ЧРЭЗ ХРБЙ ПЭЭЩ ДМЕЖ КЭЩЦ ЧРОБ ЭБЮ\_ ЧЕЖЦ ФЦЫН

Можно выдвинуть и обобщенную систему Вижинера. ЕЕ можно сформулировать не только при помощи подстановки Цезаря.

Пусть *x* - подмножество симметрической группы SYM(Zm).

*Определение*. *r-многоалфавитный ключ* шифрования есть *r*-набор π = (π0, π1, ..., π*r*-1) с элементами в *x*.

*Обобщенная система Вижинера* преобразует исходный текст (x0, *x*1 ,..., *x*n-1) в шифрованный текст (y0 ,y1 ,...,yn-1) при помощи ключа π = (π0, π1, ..., π*r*-1) по правилу

VI*Gk* : (x0 ,x1 ,...,xn-1) → (y0 ,y1 ,...,yn-1) = (π0(х0), π1(х1), ..., πn-1(xn-1)),

где используется условие πi = πimod *r* .

Следует признать, что и многоалфавитные подстановки в принципе доступны криптоаналитическому исследованию. Криптостойкость многоалфавитных систем резко убывает с уменьшением длины ключа.

Тем не менее такая система как шифр Вижинера допускает несложную аппаратную или программную реализацию и при достаточно большой длине ключа может быть использован в современных ИС.

### Гам­ми­ро­ва­ние

Гам­ми­ро­ва­ние яв­ля­ет­ся так­же ши­ро­ко при­ме­няе­мым крип­то­гра­фи­че­ским пре­об­ра­зо­ва­ни­ем. На са­мом де­ле гра­ни­ца ме­ж­ду гам­ми­ро­ва­ни­ем и ис­поль­зо­ва­ни­ем бес­ко­неч­ных клю­чей и шиф­ров Ви­жи­не­ра, о ко­то­рых речь шла вы­ше, весь­ма ус­лов­ная.

Прин­цип *шифрования* гам­ми­ро­ва­ни­ем за­клю­ча­ет­ся в ге­не­ра­ции гам­мы шиф­ра с по­мо­щью дат­чи­ка псев­до­слу­чай­ных чи­сел и на­ло­же­нии по­лу­чен­ной гам­мы на от­кры­тые дан­ные об­ра­ти­мым об­ра­зом (на­при­мер, ис­поль­зуя сло­же­ние по мо­ду­лю 2).

Про­цесс *дешифрования* дан­ных сво­дит­ся к по­втор­ной ге­не­ра­ции гам­мы шиф­ра при из­вест­ном клю­че и на­ло­же­нии та­кой гам­мы на за­шиф­ро­ван­ные дан­ные.

По­лу­чен­ный за­шиф­ро­ван­ный текст яв­ля­ет­ся дос­та­точ­но труд­ным для рас­кры­тия в том слу­чае, ес­ли гам­ма шиф­ра не со­дер­жит по­вто­ряю­щих­ся би­то­вых по­сле­до­ва­тель­ностей. По су­ти де­ла гам­ма шиф­ра долж­на из­ме­нять­ся слу­чай­ным об­ра­зом для ка­ж­до­го шиф­руе­мо­го сло­ва. Фак­ти­че­ски же, ес­ли пе­ри­од гам­мы пре­вы­ша­ет дли­ну все­го за­шиф­ро­ван­но­го тек­ста и не­из­вест­на ни­ка­кая часть ис­ход­но­го тек­ста, то шифр мож­но рас­крыть толь­ко пря­мым пе­ре­бо­ром (про­бой на ключ). Криптостойкость в этом слу­чае оп­ре­де­ля­ет­ся раз­ме­ром клю­ча.

Ме­тод гам­ми­ро­ва­ния ста­но­вит­ся бес­силь­ным, ес­ли зло­умыш­лен­ни­ку ста­но­вит­ся из­вес­тен фраг­мент ис­ход­но­го тек­ста и со­от­вет­ст­вую­щая ему шиф­ро­грам­ма. Про­стым вы­чи­та­ни­ем по мо­ду­лю по­лу­ча­ет­ся от­ре­зок ПСП и по не­му вос­ста­нав­ли­ва­ет­ся вся по­сле­до­ва­тель­ность. Зло­умыш­лен­ни­ки мо­жет сде­лать это на ос­но­ве до­га­док о со­дер­жа­нии ис­ход­но­го тек­ста. Так, ес­ли боль­шин­ст­во по­сы­лае­мых со­об­ще­ний на­чи­на­ет­ся со слов “СОВ.СЕК­РЕТ­НО”, то крип­тоа­на­лиз все­го тек­ста зна­чи­тель­но об­лег­ча­ет­ся. Это сле­ду­ет учи­ты­вать при соз­да­нии ре­аль­ных сис­тем ин­фор­ма­ци­он­ной безо­пас­но­сти.

Ниже рассматриваются наиболее распространенные методы генерации гамм, которые могут быть использованы на практике.

### Датчики ПСЧ

Что­бы по­лу­чить ли­ней­ные по­сле­до­ва­тель­но­сти эле­мен­тов гам­мы, дли­на ко­то­рых пре­вы­ша­ет раз­мер шиф­руе­мых дан­ных, ис­поль­зу­ют­ся *дат­чи­ки ПСЧ*. На ос­но­ве тео­рии групп бы­ло раз­ра­бо­та­но не­сколь­ко ти­пов та­ких дат­чи­ков.

#### Конгруэнтные датчики

В на­стоя­щее вре­мя наи­бо­лее дос­туп­ны­ми и эф­фек­тив­ны­ми яв­ля­ют­ся *кон­гру­энт­ные* ге­не­ра­то­ры ПСП. Для это­го клас­са ге­не­ра­то­ров мож­но сде­лать ма­те­ма­ти­че­ски стро­гое за­клю­че­ние о том, ка­ки­ми свой­ст­ва­ми об­ла­да­ют вы­ход­ные сиг­на­лы этих ге­не­ра­то­ров с точ­ки зре­ния пе­рио­дич­но­сти и слу­чай­но­сти.

Од­ним из хо­ро­ших кон­гру­энт­ных ге­не­ра­то­ров яв­ля­ет­ся ли­ней­ный кон­гру­энт­ный дат­чик ПСЧ. Он вы­ра­ба­ты­ва­ет по­сле­до­ва­тель­но­сти псев­до­слу­чай­ных чи­сел T(i), опи­сы­вае­мые со­от­но­ше­ни­ем

T(i+1) = (A\*T(i)+C)mod *m*,

где А и С - кон­стан­ты, Т(0) - ис­ход­ная ве­ли­чи­на, вы­бран­ная в ка­че­ст­ве по­ро­ж­даю­ще­го чис­ла. Оче­вид­но, что эти три ве­ли­чи­ны и об­ра­зу­ют ключ.

Та­кой дат­чик ПСЧ ге­не­ри­ру­ет псев­до­слу­чай­ные чис­ла с оп­ре­де­лен­ным пе­рио­дом по­вто­ре­ния, за­ви­ся­щим от вы­бран­ных зна­че­ний А и С. Зна­че­ние *m* обыч­но ус­та­нав­ли­ва­ет­ся рав­ным 2n , где *n* - дли­на машинного сло­ва в би­тах. Дат­чик име­ет мак­си­маль­ный пе­ри­од М до то­го, как ге­не­ри­руе­мая по­сле­до­ва­тель­ность нач­нет по­вто­рять­ся. По при­чи­не, от­ме­чен­ной ра­нее, не­об­хо­ди­мо вы­би­рать чис­ла А и С та­кие, что­бы пе­ри­од М был мак­си­маль­ным. Как по­ка­за­но Д. Кну­том, ли­ней­ный кон­гру­энт­ный дат­чик ПСЧ име­ет мак­си­маль­ную дли­ну М то­гда и толь­ко то­гда, ко­гда С - не­чет­ное, и Аmod 4 = 1.

Для шиф­ро­ва­ния дан­ных с по­мо­щью дат­чи­ка ПСЧ мо­жет быть вы­бран ключ лю­бо­го раз­ме­ра. На­при­мер, пусть ключ со­сто­ит из на­бо­ра чи­сел *x*(j) раз­мер­но­стью b, где j=1, 2, ..., *n*. То­гда соз­да­вае­мую гам­му шиф­ра *G* мож­но пред­ста­вить как объ­е­ди­не­ние не­пе­ре­се­каю­щих­ся мно­жеств H(j).

#### Датчики М-последовательностей[[5]](#footnote-5)

М-последовательности также популярны, благодаря относительной легкости их реализации.

М-последовательности представляют собой линейные рекуррентные последовательности максимального периода, формируемые *k*-разрядными генераторами на основе регистров сдвига. На каждом такте поступивший бит сдвигает k предыдущих и к нему добавляется их сумма по модулю 2. Вытесняемый бит добавляется к гамме.

Строго это можно представить в виде следующих отношений:

r1:=r0 r2:=r1 ... rk-1:=rk-2

r0:=a0 r1 ⊕ a1 r2 ⊕ ... ⊕ ak-2 rk-1

Гi:=rk-

Здесь r0 r1 ... rk-1 - k однобитных регистров, a0 a1 ... ak-1 - коэффициенты неприводимого двоичного полинома степени k-1. Гi - i-е значение выходной гаммы.

*Период* М-последовательности исходя из ее свойств равен 2*k*-1.

Другим важным свойством М-последовательности является *объем ансамбля*, т.е. количество различных М-последовательностей для заданного *k*. Эта характеристика приведена в таблице:

|  |  |
| --- | --- |
| *k* | Объем ансамбля |
| 5 | 6 |
| 6 | 8 |
| 7 | 18 |
| 8 | 16 |
| 9 | 48 |
| 10 | 60 |
| 16 | 2048 |

Очевидно, что такие объемы ансамблей последовательности неприемлемы.

Поэтому на практике часто используют последовательности Голда, образующиеся суммированием нескольких М-последовательно­стей. Объем ансамблей этих последовательностей на несколько порядков превосходят объемы ансамблей порождающих М-последовательностей. Так при *k*=10 ансамбль увеличивается от 1023 (М-последовательности) до 388000.

Также перспективными представляются нелинейные датчики ПСП (например сдвиговые регистры с элементом И в цепи обратной связи), однако их свойства еще недостаточно изучены.

 Воз­мож­ны и дру­гие, бо­лее слож­ные ва­ри­ан­ты вы­бо­ра по­ро­ж­даю­щих чи­сел для гам­мы шиф­ра.

Шиф­ро­ва­ние с по­мо­щью дат­чи­ка ПСЧ яв­ля­ет­ся до­воль­но рас­про­стра­нен­ным крип­то­гра­фи­че­ским ме­то­дом. Во мно­гом ка­че­ст­во шиф­ра, по­стро­ен­но­го на ос­но­ве дат­чи­ка ПСЧ, оп­ре­де­ля­ет­ся не толь­ко и не столь­ко ха­рак­те­ри­сти­ка­ми дат­чи­ка, сколь­ко ал­го­рит­мом по­лу­че­ния гам­мы. Один из фун­да­мен­таль­ных прин­ци­пов крип­то­ло­ги­че­ской прак­ти­ки гла­сит, да­же слож­ные шиф­ры мо­гут быть очень чув­ст­ви­тель­ны к про­стым воз­дей­ст­ви­ям.

### Стан­дарт шиф­ро­ва­ния дан­ных ГОСТ 28147-89[[6]](#footnote-6)

Важ­ной за­да­чей в обес­пе­че­нии га­ран­ти­ро­ван­ной безо­пас­но­сти ин­фор­ма­ции в ИС яв­ля­ет­ся раз­ра­бот­ка и ис­поль­зо­ва­ния стан­дарт­ных ал­го­рит­мов шиф­ро­ва­ния дан­ных. Пер­вым сре­ди по­доб­ных стан­дар­тов стал аме­ри­кан­ский DES, пред­став­ляю­щий со­бой по­сле­до­ва­тель­ное ис­поль­зо­ва­ние за­мен и пе­ре­ста­но­вок. В на­стоя­щее вре­мя все ча­ще го­во­рят о не­оп­рав­дан­ной слож­но­сти и не­вы­со­кой крип­то­стой­ко­сти. На прак­ти­ке при­хо­дит­ся ис­поль­зо­вать его мо­ди­фи­ка­ции.

Бо­лее эф­фек­тив­ным яв­ля­ет­ся оте­че­ст­вен­ный стан­дарт шиф­ро­ва­ния дан­ных.

Он ре­ко­мен­до­ван к ис­поль­зо­ва­нию для за­щи­ты лю­бых дан­ных, пред­став­лен­ных в ви­де дво­ич­но­го ко­да, хо­тя не ис­клю­ча­ют­ся и дру­гие ме­то­ды шиф­ро­ва­ния. Дан­ный стан­дарт фор­ми­ро­вал­ся с уче­том ми­ро­во­го опы­та, и в ча­ст­но­сти, бы­ли при­ня­ты во вни­ма­ние не­дос­тат­ки и не­реа­ли­зо­ван­ные воз­мож­но­сти ал­го­рит­ма DES, по­это­му ис­поль­зо­ва­ние стан­дар­та ГОСТ пред­поч­ти­тель­нее. Ал­го­ритм дос­та­точ­но сло­жен и ни­же бу­дет опи­са­на в ос­нов­ном его кон­цеп­ция.

Вве­дем ас­со­циа­тив­ную опе­ра­цию кон­ка­те­на­ции, ис­поль­зуя для нее муль­ти­п­ли­ка­тив­ную за­пись. Кро­ме то­го бу­дем ис­поль­зо­вать сле­дую­щие опе­ра­ции сло­же­ния:

1. A⊕B - побитовое сложение по модулю 2;
2. A[+]B - сложение по модулю 232;
3. A{+}B - сложение по модулю 232-1;.

Алгоритм криптографического преобразования предусматривает несколько режимов работы. Во всех режимах используется ключ W длиной 256 бит, представляемый в виде восьми 32-разрядных чисел *x*(i).

W=X(7)X(6)X(5)X(4)X(3)X(2)X(1)X(0)

Для дешифрования используется тот же ключ, но процесс дешифрования является инверсным по отношению к исходному.

Самый простой из возможных режимов - *замена.*

Пусть открытые блоки разбиты на блоки по 64 бит в каждом, которые обозначим как T(j).

Очередная последовательность бит T(j) разделяется на две последовательности B(0) и A(0) по 32 бита (правый и левый блоки). Далее выполняется итеративный процесс шифрования описываемый следующими формулами, вид который зависит от :i:

1. Для i=1, 2, ..., 24, j=(i-1)mod 8;

A(i) = *f*(A(i-1) [+] *x*(j)) ⊕ B(i-1)

B(i) = A(i-1)

1. Для i=25, 26, ..., 31, j=32-i;

A(i) = *f*(A(i-1) [+] *x*(j)) ⊕ B(i-1)

B(i) = A(i-1)

1. Для i=32

A(32) = A(31)

B(32) = *f*(A(31) [+] *x*(0)) ⊕ B(31).

Здесь i обозначает номер итерации. Функция *f* – функция шифрования.

Функция шифрования включает две операции над 32-разрядным аргументом.

Первая операция является подстановкой *K*. Блок подстановки К состоит из 8 узлов замены К(1)...К(8) с памятью 64 бита каждый. Поступающий на блок подстановки 32-разрядный вектор разбивается на 8 последовательно идущих 4-разрядных вектора, каждый из который преобразуется в 4-разрядный вектор соответствующим узлом замены, представляющим из себя таблицу из 16 целых чисел в диапазоне 0...15. Входной вектор определяет адрес строки в таблице, число из которой является выходным вектором. Затем 4-разрядные векторы последовательно объединяются в 32-разрядный выходной.

Вторая операция - циклический сдвиг влево 32-разрядного вектора, полученного в результате подстановки К. 64-разрядный блок зашифрованных данных Т представляется в виде

Т=А(32)В(32).

Остальные блоки открытых данных в режиме простой замены зашифровываются аналогично.

Следует учитывать, что данный режим шифрования обладает ограниченной криптостойкостью.

Другой режим шифрования называется *режимом гаммирования*.

Открытые данные, разбитые на 64-разрядные блоки T(i) (i=1,2,...,m) *(m* определяется объемом шифруемых данных), зашифровываются в режиме гаммирования путем поразрядного сложения по модулю 2 с гаммой шифра Гш, которая вырабатывается блоками по 64 бит, т.е.

Гш=(Г(1),Г(2),....,Г(m)).

Уравнение шифрования данных в режиме гаммирования может быть представлено в следующем виде:

Ш(i)=A(Y(i-1) ⊕ C2, Z(i-1)) {+} C(1) ⊕ T(i)=Г(i) ⊕ T(i)

В этом уравнении Ш(i) обозначает 64-разрядный блок зашифрованного текста, А - функцию шифрования в режиме простой замены (аргументами этой функции являются два 32-разрядных числа). С1 и С2 - константы, заданные в ГОСТ 28147-89. Величины *y*(i) и Z(i) определяются итерационно по мере формирования гаммы следующим образом:

(Y(0),Z(0))=A(S), S - 64-разрядная двоичная последовательность

(Y(i),Z(i))=(Y(i-1) [+] C2, Z(i-1) {+} C(1)), i=1, 2, ..., *m*.

64-разрядная последовательность, называемая синхропосылкой, не является секретным элементом шифра, но ее наличие необходимо как на передающей стороне, так и на приемной.

*Режим гаммирования с обратной связью* очень похож на режим гаммирования. Как и в режиме гаммирования открытые данные, разбитые на 64-разрядные блоки T(i), зашифровываются путем поразрядного сложения по модулю 2 с гаммой шифра Гш, которая вырабатывается блоками по 64 бит:

Гш=(Г(1), Г(2), ..., Г(m)).

Уравнение шифрования данных в режиме гаммирования с обратной связью выглядят следующим образом:

Ш(1)=A(S)⊕T(1)=Г(1)⊕T(1),

Ш(i)=A(Ш(i-1)⊕T(i)=Г(i)⊕T(i), i=2, 3, ..., *m*.

В ГОСТ 28147-89 определяется процесс выработки имито­вставки, который единообразен для всех режимов шифрования. Имитовставка - это блок из *р* бит (имитовставка Ир), который вырабатывается либо перед шифрованием всего сообщения. либо параллельно с шифрованием по блокам. Параметр *р* выбирается в соответствии с необходимым уровнем имитозащищенности.

Для по­лу­че­ния ими­тов­став­ки от­кры­тые дан­ные пред­став­ля­ют­ся так­же в ви­де бло­ков по 64 бит. Пер­вый блок от­кры­тых дан­ных Т(1) под­вер­га­ет­ся пре­об­ра­зо­ва­нию, со­от­вет­ст­вую­ще­му пер­вым 16 цик­лам ал­го­рит­ма ре­жи­ма про­стой за­ме­ны. При­чем в ка­че­ст­ве клю­ча ис­поль­зу­ет­ся тот же ключ, что и для шиф­ро­ва­ния дан­ных. По­лу­чен­ное 64-раз­ряд­но чис­ло сум­ми­ру­ет­ся с от­кры­тым бло­ком Т(2) и сум­ма вновь под­вер­га­ет­ся 16 цик­лам шиф­ро­ва­ния для ре­жи­ма про­стой за­ме­ны. Дан­ная про­це­ду­ра по­вто­рят­ся для всех *m* бло­ков со­об­ще­ния. Из по­лу­чен­но­го 64-раз­ряд­но­го чис­ла вы­би­ра­ет­ся от­ре­зок Ир дли­ной *р* бит.

Ими­тов­став­ка пе­ре­да­ет­ся по ка­на­лу свя­зи по­сле за­шиф­ро­ван­ных дан­ных. На при­ем­ной сто­ро­не ана­ло­гич­ным об­ра­зом из при­ня­то­го со­об­ще­ния выделяется? ими­тов­став­ка и срав­ни­ва­ет­ся с по­лу­чен­ной откуда?. В слу­чае не­сов­па­де­ния ими­тов­ста­вок со­об­ще­ние счи­та­ет­ся лож­ным.

## Сис­те­мы с от­кры­тым клю­чом

Как бы ни бы­ли слож­ны и на­деж­ны крип­то­гра­фи­че­ские сис­те­мы - их сла­бое ме­ст при прак­ти­че­ской реа­ли­за­ции - про­блема *рас­пре­де­ле­ния клю­чей*. Для то­го, что­бы был воз­мо­жен об­мен кон­фи­ден­ци­аль­ной ин­фор­ма­ци­ей ме­ж­ду дву­мя субъ­ек­та­ми ИС, ключ дол­жен быть сге­не­ри­ро­ван од­ним из них, а за­тем ка­ким-то об­ра­зом опять же в кон­фи­ден­ци­аль­ном по­ряд­ке пе­ре­дан дру­го­му. Т.е. в об­щем слу­чае для пе­ре­да­чи клю­ча опять же тре­бу­ет­ся ис­поль­зо­ва­ние ка­кой-то крип­то­си­сте­мы.

Для ре­ше­ния этой про­бле­мы на ос­но­ве ре­зуль­та­тов, по­лу­чен­ных классической и со­вре­мен­ной ал­геб­рой, бы­ли пред­ло­же­ны *сис­те­мы с от­кры­тым клю­чом.*

Суть их со­сто­ит в том, что ка­ж­дым ад­ре­са­том ИС ге­не­ри­ру­ют­ся два клю­ча, свя­зан­ные ме­ж­ду со­бой по оп­ре­де­лен­но­му пра­ви­лу. Один ключ объ­яв­ля­ет­ся *от­кры­тым*, а дру­гой *за­кры­тым*. От­кры­тый ключ пуб­ли­ку­ет­ся и дос­ту­пен лю­бо­му, кто же­ла­ет по­слать со­об­ще­ние ад­ре­са­ту. Секретный ключ сохраняется в тайне.

Ис­ход­ный текст шиф­ру­ет­ся от­кры­тым клю­чом адресата и пе­ре­да­ет­ся ему. За­шиф­ро­ван­ный текст в прин­ци­пе не мо­жет быть рас­шиф­ро­ван тем же от­кры­тым клю­чом. Де­шиф­ро­ва­ние со­об­ще­ние воз­мож­но толь­ко с ис­поль­зо­ва­ни­ем за­кры­то­го клю­ча, ко­то­рый из­вес­тен толь­ко са­мо­му ад­ре­са­ту.

Отправитель

Адресат

исходный

текст

шифрованный

текст

исходный

текст

Система

с открытым ключом

Система

с открытым ключом

Закрытый ключ

Открытый ключ

Крип­то­гра­фи­че­ские сис­те­мы с от­кры­тым клю­чом ис­поль­зу­ют так называемые  *не­об­ра­ти­мые или од­но­сто­рон­ние функ­ции*, ко­то­рые об­ла­да­ют сле­дую­щим свой­ст­вом: при за­дан­ном зна­че­нии *x* от­но­си­тель­но про­сто вы­чис­лить зна­че­ние *f(x),* од­на­ко ес­ли *y*=*f(x*), то нет про­сто­го пу­ти для вы­чис­ле­ния зна­че­ния *x.*

Мно­же­ст­во клас­сов не­об­ра­ти­мых функ­ций и по­ро­ж­да­ет все раз­но­об­ра­зие сис­тем с от­кры­тым клю­чом. Од­на­ко не вся­кая не­об­ра­ти­мая функ­ция го­дит­ся для ис­поль­зо­ва­ния в ре­аль­ных ИС.

В са­мом оп­ре­де­ле­нии не­об­ра­ти­мо­сти при­сут­ст­ву­ет не­оп­ре­де­лен­ность. Под *необратимостью* понимается не теоретическая необратимость, а практическая невозможность вычислить обратное значение используя современные вычислительные средства за обозримый интервал времени.

По­это­му что­бы га­ран­ти­ро­вать на­деж­ную за­щи­ту ин­фор­ма­ции, к сис­те­мам с от­кры­тым клю­чом (СОК) предъ­яв­ля­ют­ся два важ­ных и оче­вид­ных тре­бо­ва­ния:

1. Пре­об­ра­зо­ва­ние ис­ход­но­го тек­ста долж­но быть не­об­ра­ти­мым и ис­клю­чать его вос­ста­нов­ле­ние на ос­но­ве от­кры­то­го клю­ча.

2. Оп­ре­де­ле­ние за­кры­то­го клю­ча на ос­но­ве от­кры­то­го так­же долж­но быть не­воз­мож­ным на со­вре­мен­ном тех­но­ло­ги­че­ском уров­не. При этом же­ла­тель­на точ­ная ниж­няя оцен­ка сложности (ко­ли­че­ст­ва опе­ра­ций) рас­кры­тия шиф­ра.

Ал­го­рит­мы шиф­ро­ва­ния с от­кры­тым клю­чом по­лу­чи­ли ши­ро­кое рас­про­стра­не­ние в со­вре­мен­ных ин­фор­ма­ци­он­ных сис­те­мах. Так, ал­го­ритм RSA стал ми­ро­вым стан­дар­том де-фак­то для от­кры­тых сис­тем и ре­ко­мен­до­ван МККТТ.

Вообще же все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих типов необратимых преобразований:

1. Разложение больших чисел ан простые множители.
2. Вычисление логарифма в конечном поле.
3. Вычисление корней алгебраических уравнений.

Здесь же сле­ду­ет от­ме­тить, что ал­го­рит­мы криптосистемы с открытым ключом (СОК) мож­но ис­поль­зо­вать в трех на­зна­че­ни­ях.

1. Как *са­мо­стоя­тель­ные сред­ст­ва за­щи­ты* пе­ре­да­вае­мых и хра­ни­мых дан­ных.

2. Как *сред­ст­ва для рас­пре­де­ле­ния клю­чей*. Ал­го­рит­мы СОК бо­лее тру­до­ем­ки, чем тра­ди­ци­он­ные крип­то­си­сте­мы. По­это­му час­то на прак­ти­ке ра­цио­наль­но с по­мо­щью СОК рас­пре­де­лять клю­чи, объ­ем ко­то­рых как ин­фор­ма­ции не­зна­чи­те­лен. А по­том с по­мо­щью обыч­ных ал­го­рит­мов осу­ще­ст­в­лять об­мен боль­ши­ми ин­фор­ма­ци­он­ны­ми по­то­ка­ми.

1. *Сред­ст­ва ау­тен­ти­фи­ка­ции поль­зо­ва­те­лей*. Об этом бу­дет рас­ска­за­но в главе «Электронная подпись».

Ниже рассматриваются наиболее распространенные системы с открытым ключом.

### Ал­го­ритм RSA

Не­смот­ря на до­воль­но боль­шое чис­ло раз­лич­ных СОК, наиболее популярна - криптосистема RSA, разработанная в 1977 году и по­лу­чив­шая на­зва­ние в честь ее соз­да­те­лей: Рона Ри­ве­ста[[7]](#footnote-7), Ади Ша­ми­ра и Леонарда Эй­дель­ма­на.

Они вос­поль­зо­ва­лись тем фак­том, что на­хо­ж­де­ние боль­ших про­стых чи­сел в вы­чис­ли­тель­ном от­но­ше­нии осу­ще­ст­в­ля­ет­ся лег­ко, но раз­ло­же­ние на мно­жи­те­ли про­из­ве­де­ния двух та­ких чи­сел прак­ти­че­ски не­вы­пол­ни­мо. До­ка­за­но (тео­ре­ма Ра­би­на), что рас­кры­тие шиф­ра RSA эк­ви­ва­лент­но та­ко­му раз­ло­же­нию. По­это­му для лю­бой дли­ны клю­ча мож­но дать ниж­нюю оцен­ку чис­ла опе­ра­ций для рас­кры­тия шиф­ра, а с уче­том про­из­во­ди­тель­но­сти со­вре­мен­ных ком­пь­ю­те­ров оце­нить и не­об­хо­ди­мое на это вре­мя.

Воз­мож­ность га­ран­ти­ро­ван­но оце­нить за­щи­щен­ность ал­го­рит­ма RSA ста­ла од­ной из при­чин по­пу­ляр­но­сти этой СОК на фо­не де­сят­ков дру­гих схем. По­это­му ал­го­ритм RSA ис­поль­зу­ет­ся в бан­ков­ских ком­пь­ю­тер­ных се­тях, осо­бен­но для ра­бо­ты с уда­лен­ны­ми кли­ен­та­ми (об­слу­жи­ва­ние кре­дит­ных кар­то­чек).

В настоящее время алгоритм RSA используется во многих стандартах, среди которых SSL, S-HHT*P*, S-MIME, S/WAN, STT и *P*CT.

Рас­смот­рим ма­те­ма­ти­че­ские ре­зуль­та­ты, по­ло­жен­ные в ос­но­ву это­го ал­го­рит­ма.

*Теорема 1. (Малая теорема Ферма.)*

Если р - простое число, то

*xp*-1 = 1 (mod *p*) (1)

для любого *х*, простого относительно р, и

*xp*= *х* (mod *p*) (2)

для любого *х.*

Доказательство. Достаточно доказать справедливость уравнений (1) и (2) для *х*∈Z*p*. Проведем доказательство методом индукции.

Очевидно, что уравнение (8.2.2) выполняется при х=0 и 1. Далее

*xp*=(*x*-1+1)*p*= ∑ C(*p*,j)(*x*-1)j=(*x*-1)*p*+1 (mod *p*),

 0≤j≤*p*

так как C(*p*,j)=0(mod *p*) при 0<j<*p*. С учетом этого неравенства и предложений метода доказательства по индукции теорема доказана.

*Определение. Функцией Эйлера* ϕ(n) называется число положительных целых, меньших *n* и простых относительно *n*.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| ϕ(n) | 1 | 2 | 2 | 3 | 2 | 6 | 4 | 6 | 4 | 10 | 4 |

*Теорема 2.* Если *n*=*pq*, (*p* и *q* - отличные друг от друга простые числа), то

ϕ(n)=(*p*-1)(*q*-1).

*Теорема 3.* Если *n*=*pq*, (*p* и *q* - отличные друг от друга простые числа) и х - простое относительно р и *q*, то

*x*ϕ(n) = 1 (mod *n*).

*Следствие .* Если *n*=*pq*, (*p* и *q* - отличные друг от друга простые числа) и е простое относительно ϕ(n), то отображение

Еe,n: *x*→*x*e (mod *n*)

является взаимно однозначным на **Z**n.

Очевиден и тот факт, что если е - простое относительно ϕ(n), то существует целое d, такое, что

ed = 1 (mod ϕ(n)) (3)

На этих математических фактах и основан популярный алгоритм RSA.

Пусть *n*=*pq*, где *p* и *q* - различные простые числа. Если e и d удовлетворяют уравнению (8.2.3), то отображения Еe,n и Еd,n являются инверсиями на Zn. Как Еe,n, так и Еd,n легко рассчитываются, когда известны e, d, *p*, *q*. Если известны e и *n*, но *p* и *q* неизвестны, то Еe,n представляет собой одностороннюю функцию; нахождение Еd,n по заданному *n* равносильно разложению *n*. Если *p* и *q* - достаточно большие простые, то разложение *n* практически не осуществимо. Это и заложено в основу системы шифрования RSA.

*Пользователь i* выбирает пару различных простых *p*i и *q*i и рассчитывает пару целых (ei, di), которые являются простыми относительно ϕ(ni), где *n*i=*p*i *q*i . Справочная таблица содержит публичные ключи {(ei ,ni)}.

Предположим, что исходный текст

 *x* =(*x*0, *x*1, ..., *x*n-1), *x*∈Zn , 0 ≤ i < *n*,

сначала представлен по основанию *n*i :

N = c0+ci *n*i+....

Пользователь i зашифровывает текст при передаче его пользователю j, применяя к *n* отображение Edi,ni :

N → Edi,ni *n* = *n*’.

Пользователь j производит дешифрование *n*’, применяя Eei,ni :

N’ → Eei,ni *n*’= Eei,ni Edi,ni *n* = *n* .

Очевидно, для того чтобы найти инверсию Edi,ni по отношению к Eei,ni, требуется знание множителей *n*=*p*i *q*i. Время выполнения наилучших из известных алгоритмов разложения при *n*=10100 на сегодняшний день выходит за пределы современных технологических возможностей.

Рассмотрим небольшой пример, иллюстрирующий применение алгоритма RSA.

*Пример* Зашифруем сообщение “САВ”. Для простоты будем использовать маленькие числа (на практике применяются гораздо большие).

1. Выберем *p*=3 и *q*=11.
2. Определим *n*=3\*11=33.
3. Найдем (*p*-1)(*q*-1)=20. Следовательно, в качестве *d*, взаимно простое с 20, например, *d*=3.
4. Выберем число е. В качестве такого числа может быть взято любое число, для которого удовлетворяется соотношение (е\*3) (mod 20) = 1, например 7.
5. Представим шифруемое сообщение как последовательность целых чисел с помощью отображения: А→1, В→2, С→3. Тогда сообщение принимает вид (3,1,2). Зашифруем сообщение с помощью ключа {7,33}.

ШТ1 = (37) (mod 33) = 2187 (mod 33) = 9,

ШТ2 = (17) (mod 33) = 1 (mod 33) = 1,

ШТ3 = (27) (mod 33) = 128 (mod 33) = 29.

1. Расшифруем полученное зашифрованное сообщение (9,1,29) на основе закрытого ключа {3,33}:

ИТ1 = (93) (mod 33) = 729 (mod 33) = 3,

ИТ2= (13) (mod 33) = 1 (mod 33) = 1,

ИТ3 = (293) (mod 33) = 24389 (mod 33) = 2.

Итак, в реальных системах алгоритм RSA реализуется следующим образом: каждый пользователь выбирает два больших простых числа, и в соответствии с описанным выше алгоритмом выбирает два простых числа *e* и *d*. Как результат умножения первых двух чисел (*p* и *q*) устанавливается *n*.

*{e,n}* образует открытый ключ, а *{d,n}* - закрытый (хотя можно взять и наоборот).

От­кры­тый ключ пуб­ли­ку­ет­ся и дос­ту­пен ка­ж­до­му, кто же­ла­ет по­слать вла­дель­цу клю­ча со­об­ще­ние, ко­то­рое за­шиф­ро­вы­ва­ет­ся ука­зан­ным ал­го­рит­мом. По­сле шифрования, со­об­ще­ние не­воз­мож­но рас­крыть с по­мо­щью от­кры­то­го клю­ча. Вла­де­лец же за­кры­то­го клю­ча без тру­да мо­жет рас­шиф­ро­вать при­ня­тое со­об­ще­ние.

#### Практическая реализация RSA

В настоящее время алгоритм RSA активно реализуется как в виде самостоятельных криптографических продуктов[[8]](#footnote-8), так и в качестве встроенных средств в популярных приложениях[[9]](#footnote-9).

Важная проблема практической реализации - *генерация больших простых чисел*. Решение задачи «в лоб» - генерация случайного большого числа *n* (нечетного) и проверка его делимости на множители от 3 вплоть до *n0.5*. В случае неуспеха следует взять n+2 и так далее.[[10]](#footnote-10)

В принципе в качестве p и q можно использовать «почти» простые числа, то есть числа для которых вероятность того, что они простые, стремится к 1. Но в случае, если использовано составное число, а не простое, криптостойкость RSA падает. Имеются неплохие алгоритмы, которые позволяют генерировать «почти» простые числа с уровнем доверия 2-100.

Другая проблема - *ключи какой длины следует использовать*?

Для прак­ти­че­ской реа­ли­за­ции ал­го­рит­мов RSA по­лез­но знать оцен­ки тру­до­ем­ко­сти раз­ло­же­ния про­стых чи­сел раз­лич­ной дли­ны, сде­лан­ные Шроппелем.

|  |  |  |
| --- | --- | --- |
| lo*g*10 *n* | Число операций | Примечания |
| 50 | 1.4\*1010 | Раскрываем на суперкомпьютерах |
| 100 | 2.3\*1015 | На пределе современных технологий |
| 200 | 1.2\*1023 | За пре­де­ла­ми со­вре­мен­ных тех­но­ло­гий |
| 400 | 2.7\*1034 | Тре­бу­ет су­ще­ст­вен­ных из­ме­не­ний в тех­но­ло­гии |
| 800 | 1.3\*1051 | Не раскрываем |

В кон­це 1995 го­да уда­лось прак­ти­че­ски реа­ли­зо­вать рас­кры­тие шиф­ра RSA для 500-знач­но­го клю­ча. Для это­го с по­мо­щью се­ти Ин­тер­нет бы­ло за­дей­ст­во­ва­но 1600 ком­пь­ю­те­ров.

Сами авторы RSA рекомендуют использовать следующие размеры модуля n:

1. 768 бит - для частных лиц;
2. 1024 бит - для коммерческой информации;
3. 2048 бит - для особо секретной информации.[[11]](#footnote-11)

Третий немаловажный аспект реализации RSA - *вычислительный*. Ведь приходится использовать аппарат длинной арифметики. Если используется ключ длиной *k* бит, то для операций по открытому ключу требуется *О(k2)* операций, по закрытому ключу - *О(k3)* операций, а для генерации новых ключей требуется *О(k4)* операций.

Криптографический пакет BSAFE 3.0 (RSA D.S.) на компьютере Pentium-90 осуществляет шифрование со скоростью 21.6 Кбит/c для 512-битного ключа и со скоростью 7.4 Кбит/c для 1024 битного. Самая «быстрая» аппаратная реализация обеспечивает скорости в 60 раз больше.

По сравнению с тем же алгоритмом DES, RSA требует в тысячи и десятки тысяч раз большее время.

### Криптосистема Эль-Гамаля

Данная система является альтернативой RSA и при равном значении ключа обеспечивает ту же криптостойкость[[12]](#footnote-12).

В отличие от RSA метод Эль-Гамаля основан на проблеме дискретного логарифма. Этим он похож на *алгоритм Диффи-Хелмана*. Если возводить число в степень в конечном поле достаточно легко, то восстановить аргумент по значению (то есть найти логарифм) довольно трудно.

Основу системы составляют параметры *p* и *g* - числа, первое из которых - простое, а второе - целое.

Александр генерирует секретный ключ *а* и вычисляет открытый ключ *y* = *gа* mod *p.* Если Борис хочет послать Александру сообщение *m*, то он выбирает случайное число *k*, меньшее p и вычисляет

*y1*= *gk* mod *p* и

*y2*= *m* ⊕ *yk,*

где ⊕ означает побитовое сложение по модулю 2. Затем Борис посылает *(y1,y2)* Александру.

Александр, получив зашифрованное сообщение, восстанавливает его:

*m* = *(y1a* mod *p)* ⊕ *y2.*

Алгоритм цифровой подписи DSA, разработанный NIST (National In*s*titute of Standa*r*d and Technolo*g*y) и являющийся частью стандарта DSS частично опирается на рассмотренный метод.

### Криптосистемы на основе эллиптических уравнений

Эллиптические кривые - математический объект, который может определен над любым полем (конечным, действительным, рациональным или комплексным). В криптографии обычно используются конечные поля. Эллиптическая кривая есть множество точек *(x,y),* удовлетворяющее следующему уравнению:

*y2 = x3 + ax + b,*

а также бесконечно удаленная точка. Для точек на кривой довольно легко вводится операция сложения, которая играет ту же роль, что и операция умножения в криптосистемах RSA и Эль-Гамаля.

В реальных криптосистемах на базе эллиптических уравнений используется уравнение

*y2 = x3 + ax + b* mod *p,*

где р - простое.

Проблема дискретного логарифма на эллиптической кривой состоит в следующем: дана точка G на эллиптической кривой порядка r (количество точек на кривой) и другая точка Y на этой же кривой. Нужно найти единственную точку *x* такую, что Y = *x*G, то есть Y есть *х*-я степень G.

## Электронная подпись

В чем со­сто­ит про­бле­ма ау­тен­ти­фи­ка­ции дан­ных?

В кон­це обыч­но­го пись­ма или до­ку­мен­та ис­пол­ни­тель или от­вет­ст­вен­ное ли­цо обыч­но ста­вит свою под­пись. По­доб­ное дей­ст­вие обыч­но пре­сле­ду­ет две це­ли. Во-пер­вых, по­лу­ча­тель име­ет воз­мож­ность убе­дить­ся в ис­тин­но­сти пись­ма, сли­чив под­пись с имею­щим­ся у не­го об­раз­цом. Во-вто­рых, лич­ная под­пись яв­ля­ет­ся юри­ди­че­ским га­ран­том ав­тор­ст­ва до­ку­мен­та. По­след­ний ас­пект осо­бен­но ва­жен при за­клю­че­нии раз­но­го ро­да тор­го­вых сде­лок, со­став­ле­нии до­ве­рен­но­стей, обя­за­тельств и т.д.

Ес­ли под­де­лать под­пись че­ло­ве­ка на бу­ма­ге весь­ма не­про­сто, а ус­та­но­вить ав­тор­ст­во под­пи­си со­вре­мен­ны­ми кри­ми­на­ли­сти­че­ски­ми ме­то­да­ми - тех­ни­че­ская де­таль, то с под­пи­сью элек­трон­ной де­ло об­сто­ит ина­че. Под­де­лать це­поч­ку би­тов, про­сто ее ско­пи­ро­вав, или не­за­мет­но вне­сти не­ле­галь­ные ис­прав­ле­ния в до­ку­мент смо­жет лю­бой поль­зо­ва­тель.

С ши­ро­ким рас­про­стра­не­ни­ем в со­вре­мен­ном ми­ре элек­трон­ных форм до­ку­мен­тов (в том чис­ле и кон­фи­ден­ци­аль­ных) и средств их об­ра­бот­ки осо­бо ак­ту­аль­ной ста­ла про­бле­ма ус­та­нов­ле­ния под­лин­но­сти и ав­тор­ст­ва без­бу­маж­ной до­ку­мен­та­ции.

В раз­де­ле крип­то­гра­фи­че­ских сис­тем с от­кры­тым клю­чом бы­ло по­ка­за­но, что при всех пре­иму­ще­ст­вах со­вре­мен­ных сис­тем шиф­ро­ва­ния они не по­зво­ля­ют обес­пе­чить ау­тен­ти­фи­ка­цию дан­ных. По­это­му сред­ст­ва ау­тен­ти­фи­ка­ции долж­ны ис­поль­зо­вать­ся в ком­плек­се и крип­то­гра­фи­че­ски­ми ал­го­рит­ма­ми.

Итак, пусть име­ют­ся два поль­зо­ва­те­ля Александр и Борис. От ка­ких на­ру­ше­ний и дей­ст­вий зло­умыш­лен­ни­ка долж­на за­щи­щать сис­те­ма ау­тен­ти­фи­ка­ции.

*От­каз (ре­не­гат­ст­во).*

Александр за­яв­ля­ет, что он не по­сы­лал со­об­ще­ние Борису, хо­тя на са­мом де­ле он все-та­ки по­сы­лал.

Для ис­клю­че­ния это­го на­ру­ше­ния ис­поль­зу­ет­ся *элек­трон­ная* (или *циф­ро­вая*) *под­пись*.

*Мо­ди­фи­ка­ция (пе­ре­дел­ка).*

Борис из­ме­ня­ет со­об­ще­ние и ут­вер­жда­ет, что дан­ное (из­ме­нен­ное) со­об­ще­ние по­слал ему Александр.

*Под­дел­ка.*

Борис фор­ми­ру­ет со­об­ще­ние и ут­вер­жда­ет, что дан­ное (из­ме­нен­ное) со­об­ще­ние по­слал ему Александр.

*Ак­тив­ный пе­ре­хват.*

Владимир пе­ре­хва­ты­ва­ет со­об­ще­ния ме­ж­ду Александром и Борисом с це­лью их скры­той мо­ди­фи­ка­ции.

Для за­щи­ты от мо­ди­фи­ка­ции, под­дел­ки и мас­ки­ров­ки ис­поль­зу­ют­ся *циф­ро­вые сиг­на­ту­ры*.

*Мас­ки­ров­ка (ими­та­ция).*

Владимир по­сы­ла­ет Борису со­об­ще­ние от име­ни Александра .

В этом случае для за­щи­ты так­же ис­поль­зу­ет­ся элек­трон­ная под­пись.

*По­втор.*

Владимир по­вто­ря­ет ра­нее пе­ре­дан­ное со­об­ще­ние, ко­то­рое Александра по­сы­лал ра­нее Борису . Не­смот­ря на то, что при­ни­ма­ют­ся все­воз­мож­ные ме­ры за­щи­ты от по­вто­ров, имен­но на этот ме­тод при­хо­дит­ся боль­шин­ст­во слу­ча­ев не­за­кон­но­го сня­тия и тра­ты де­нег в сис­те­мах элек­трон­ных пла­те­жей.

Наи­бо­лее дей­ст­вен­ным ме­то­дом за­щи­ты от по­вто­ра яв­ля­ют­ся

1. ис­поль­зо­ва­ние *ими­тов­ста­вок*,
2. учет вхо­дя­щих со­об­ще­ний.

*подделка*

*модификация*

*отказ*

Пользователь В

Пользователь А

сообщение

Нарушитель

*перехват*

*повтор*

*маскировка*

*Возможные нарушения защиты сообщений,. посылаемых пользователем А пользователю В.*

### Электронная подпись на основе алгоритма RSA

Наи­бо­лее про­стым и рас­про­стра­нен­ным ин­ст­ру­мен­том элек­трон­ной под­пи­си яв­ля­ет­ся уже зна­ко­мый ал­го­ритм RSA. Ни­же оно бу­дет рас­смот­ре­на в ка­че­ст­ве при­ме­ра. Кро­ме это­го су­ще­ст­ву­ют еще десятки других схем цифровой подписи.

Предположим, что

 *d*,*p*,*q* - секретные, а *е*, *n*=*pq* - открытые.

*Замечания.*

1. Разложение по *n* дает*: ϕ(n)=(p-1)(q-1);* зная *ϕ(n)* и *e*, можно найти *d*.

2. Из *e* и *d* можно найти кратность *ϕ(n);* кратность *ϕ(n)* позволяет определить делители *n*.

Пусть DATA - передаваемое Александром Борису сообщение.

Александр подписывает DATA для Бориса при передаче :

EeB,nB {EdA,nA {DATA}}.

При этом он использует:

1. закрытый ключ EdA,nA  Александра,
2. открытый ключ EeB,nB  Бориса.

Борис может читать это подписанное сообщение сначала при помощи закрытого ключа EdВ,nВ Бориса с целью получения

EdA,nA {DATA} = EdB,nB {EeB,nB {EdA,nA {DATA}}}

и затем - открытого ключа EeA,nA Александра для получения

DATA = EeA,nA { EdA,nA {DATA}}.

Таким образом, у Бориса появляется сообщение DATA, посланное ему Александром.

Очевидно, что данная схема позволяет защититься от нескольких видов нарушений.

Александр не может отказаться от своего сообщения, если он признает, что секретный ключ известен только ему.

Нарушитель без знания секретного ключа не может ни сформировать, ни сделать осмысленное изменение сообщения, передаваемого по линии связи.

Данная схема позволяет при решении многих конфликтных ситуаций обходиться без посредников.

Иногда нет необходимости зашифровывать передаваемое сообщение, но нужно его скрепить электронной подписью. В этом случае текст шифруется закрытым ключом отправителя и полученная цепочка символов прикрепляется к документу. Получатель с помощью открытого ключа отправителя расшифровывает подпись и сверяет ее с текстом.

*передача*

*проверка*

*подпись*

Исходный текст

Исходный текст

Исходный текст

Подпись

Исходный текст ’

*Открытый ключ А*

*Закрытый ключ А*

####

В 1991 г. Национальный институт стандартов и технологии (NIST)предложил для появившегося тогда алгоритма цифровой подписи DSA (Di*g*ital Si*g*natu*r*e Al*g*o*r*ithm) стандарт DSS (Di*g*ital Si*g*natu*r*e Standa*r*d), в основу которого положены алгоритмы Эль-Гамаля и RSA. [[13]](#footnote-13)

### Цифровая сигнатура

Часто возникают ситуации, когда получатель должен уметь доказать подлинность сообщения внешнему лицу. Чтобы иметь такую возможность, к передаваемым сообщениям должны быть приписаны так называемые цифровые сигнатуры.

*Цифровая сигнатура* - это строка символов, зависящая как от идентификатора отправителя, так и содержания сообщения.

сообщение

*сигнатура*

Пользователь А

*Цифровая сигнатура*

Ни­кто при этом кро­ме поль­зо­ва­те­ля А не мо­жет вы­чис­лить циф­ро­вую сиг­на­ту­ру А для кон­крет­но­го со­об­ще­ния. Ни­кто, да­же сам поль­зо­ва­тель не мо­жет из­ме­нить по­слан­но­го со­об­ще­ния так, что­бы сиг­на­ту­ра ос­та­лась не­из­мен­ной. Хо­тя по­лу­ча­тель дол­жен иметь воз­мож­ность про­ве­рить яв­ля­ет­ся ли циф­ро­вая сиг­на­ту­ра со­об­ще­ния под­лин­ной. Что­бы про­ве­рить циф­ро­вую сиг­на­ту­ру, поль­зо­ва­тель В дол­жен пред­ста­вить по­сред­ни­ку С ин­фор­ма­цию, ко­то­рую он сам ис­поль­зо­вал для ве­ри­фи­ка­ции сиг­на­ту­ры.

Ес­ли по­ме­чен­ное сиг­на­ту­рой со­об­ще­ние пе­ре­да­ет­ся не­по­сред­ст­вен­но от от­пра­ви­те­ля к по­лу­ча­те­лю, ми­нуя про­ме­жу­точ­ное зве­но, то в этом слу­чае идет речь об *ис­тин­ной циф­ро­вой сиг­на­ту­ре*.

Рас­смот­рим ти­пич­ную схе­му циф­ро­вой сиг­на­ту­ры.

Пусть Е - функция симметричного шифрования и *f* - функция отображения некоторого множества сообщений на подмножество мощности *р* из последовательности {1, ..., *n*}.

Например р=3 и *n*=9. Если *m* - сообщение , то в качестве *f* можно взять функцию *f*(*m*) = {2, 5, 7}.

Для каждого сообщения пользователь А выбирает некоторое множество ключей *K*=[*K*1, ..., *K*n} и параметров V={v1, ...,vn} для использования в качестве пометок сообщения, которое будет послано В. Множества V и V’={E(v1,*K*1) ..., E(vn,*K*n)} посылаются пользователю В и заранее выбранному посреднику С.

Пусть *m* - сообщение и idm - объединение идентификационных номеров отправителя, получателя и номера сообщения. Если *f(*{idm, *m*}), то цифровая сигнатура *m* есть множество *K*’=[*K*i, ..., *K*j}. Сообщение *m*, идентификационный номер idm и цифровая сигнатура К’ посылаются В.

V, V'

сообщение, *K*'

Пользователь В

Пользователь А

Посредник С

V, V'

Получатель В проверяет сигнатуру следующим образом. Он вычисляет функцию *f(*{idm, *m*}) и проверяет ее равенство К’. Затем он проверяет, что подмножество {vi, ...,vj} правильно зашифровано в виде подмножества {E(vi,*K*i) ..., E(vj,*K*j)} множества V’.

В кон­фликт­ной си­туа­ции В по­сы­ла­ет С сообщение *m*, иден­ти­фи­ка­ци­он­ный но­мер idm и мно­же­ст­во клю­чей *K*’, ко­то­рое В объ­яв­ля­ет сиг­на­ту­рой *m*. То­гда по­сред­ник С так же, как и В, бу­дет спо­со­бен про­ве­рить сиг­на­ту­ру. Ве­ро­ят­ность рас­кры­тия двух со­об­ще­ний с од­ним и тем же зна­че­ни­ем функ­ции *f* долж­на быть очень ма­ла. Что­бы га­ран­ти­ро­вать это, чис­ло  *n* долж­но быть дос­та­точ­но боль­шим, а чис­ло *р* долж­но быть боль­ше 1, но мень­ше *n*.

Ряд не­дос­тат­ков этой мо­де­ли оче­ви­ден:

1. долж­но быть третье ли­цо - по­сред­ник, ко­то­ро­му до­ве­ря­ют как по­лу­ча­тель, так и от­пра­ви­тель;
2. по­лу­ча­тель, от­пра­ви­тель и по­сред­ник долж­ны об­ме­нять­ся су­ще­ст­вен­ным объ­е­мом ин­фор­ма­ции, пре­ж­де чем бу­дет пе­ре­да­но ре­аль­ное со­об­ще­ние;
3. пе­ре­да­ча этой ин­фор­ма­ции долж­на осу­ще­ст­в­лять­ся в за­кры­том ви­де;
4. эта ин­фор­ма­ция ис­поль­зу­ет­ся край­не не­эф­фек­тив­но, по­сколь­ку мно­же­ст­ва *K*, V, V’ ис­поль­зу­ют­ся толь­ко один раз.

Тем не ме­нее да­же та­кая схе­ма циф­ро­вой сиг­на­ту­ры мо­жет ис­поль­зо­вать­ся в ин­фор­ма­ци­он­ных сис­те­мах, в ко­то­рых не­об­хо­ди­мо обес­пе­чить ау­тен­ти­фи­ка­цию и за­щи­ту пе­ре­да­вае­мых со­об­ще­ний.

#### Хэш-функции

Использование цифровой сигнатуры предполагает использование некоторых функций шифрования:

*S = H(k, T)*,

где *S* - сигнатура, *k* - ключ, *T* - исходный текст.

Функция *H(k, T)* - является хэш-функцией, если она удовлетворяет следующим условиям:

1. исходный текст может быть произвольной длины;
2. само значение *H(k, T)* имеет фиксированную длину;
3. значение функции *H(k, T)* легко вычисляется для любого аргумента;
4. восстановить аргумент по значению с вычислительной точки зрения - практически невозможно;
5. функция *H(k, T) -* однозначна[[14]](#footnote-14).

Из определения следует, что для любой хэш-функции есть тексты-близнецы - имеющие одинаковое значение хэш-функции, так как мощность множества аргументов неограниченно больше мощности множества значений. Такой факт получил название «эффект дня рождения».[[15]](#footnote-15)

Наиболее известные из хэш-функций - MD2, MD4, MD5 и SHA.

Три алгоритма серии MD разработаны Ривестом в 1989-м, 90-м и 91-м году соответственно. Все они преобразуют текст произвольной длины в 128-битную сигнатуру.

Алгоритм MD2 предполагает:

1. дополнение текста до длины, кратной 128 битам;
2. вычисление 16-битной контрольной суммы (старшие разряды отбрасываются);
3. добавление контрольной суммы к тексту;
4. повторное вычисление контрольной суммы.

Алгоритм MD4 предусматривает:

1. дополнение текста до длины, равной 448 бит по модулю 512;
2. добавляется длина текста в 64-битном представлении;
3. 512-битные блоки подвергаются процедуре Damgard-Merkle[[16]](#footnote-16), причем каждый блок участвует в трех разных циклах.

В алгоритме MD4 довольно быстро были найдены «дыры», поэтому он был заменен алгоритмом MD5, в котором каждый блок участвует не в трех, а в четырех различных циклах.

Алгоритм SHA (Secure Hash Algorithm) разработан NIST (National In*s*titute of Standa*r*d and Technolo*g*y) и повторяет идеи серии MD. В SHA используются тексты более 264 бит, которые закрываются сигнатурой длиной 160 бит. Данный алгоритм предполагается использовать в программе Capstone[[17]](#footnote-17).

## Управ­ле­ние клю­ча­ми

Кро­ме вы­бо­ра под­хо­дя­щей для кон­крет­ной ИС крип­то­гра­фи­че­ской сис­те­мы, важ­ная про­бле­ма - управ­ле­ние клю­ча­ми. Как бы ни бы­ла слож­на и на­деж­на са­ма крип­то­си­сте­ма, она ос­но­ва­на на ис­поль­зо­ва­нии клю­чей. Ес­ли для обес­пе­че­ния кон­фи­ден­ци­аль­но­го об­ме­на ин­фор­ма­ци­ей ме­ж­ду дву­мя поль­зо­ва­те­ля­ми про­цесс об­ме­на клю­ча­ми три­виа­лен, то в ИС, где ко­ли­че­ст­во поль­зо­ва­те­лей со­став­ля­ет де­сят­ки и сот­ни управ­ле­ние клю­ча­ми - серь­ез­ная про­бле­ма.

Под *клю­че­вой ин­фор­ма­ци­ей* по­ни­ма­ет­ся со­во­куп­ность всех дей­ст­вую­щих в ИС клю­чей. Ес­ли не обес­пе­че­но дос­та­точ­но на­деж­ное управ­ле­ние клю­че­вой ин­фор­ма­ци­ей, то за­вла­дев ею, зло­умыш­лен­ник по­лу­ча­ет не­ог­ра­ни­чен­ный дос­туп ко всей ин­фор­ма­ции.

*Управ­ле­ние клю­ча­ми* - ин­фор­ма­ци­он­ный про­цесс, вклю­чаю­щий в се­бя три эле­мен­та:

1. ге­не­ра­цию клю­чей;
2. на­ко­п­ле­ние клю­чей;
3. рас­пре­де­ле­ние клю­чей.

Рас­смот­рим, как они долж­ны быть реа­ли­зо­ва­ны для то­го, что­бы обес­пе­чить безо­пас­ность клю­че­вой ин­фор­ма­ции в ИС.

### Ге­не­ра­ция клю­чей

В са­мом на­ча­ле раз­го­во­ра о крип­то­гра­фи­че­ских ме­то­дах бы­ло ска­за­но, что не сто­ит ис­поль­зо­вать не­слу­чай­ные клю­чи с це­лью лег­ко­сти их за­по­ми­на­ния. В серь­ез­ных ИС ис­поль­зу­ют­ся спе­ци­аль­ные ап­па­рат­ные и про­грамм­ные ме­то­ды ге­не­ра­ции слу­чай­ных клю­чей. Как пра­ви­ло ис­поль­зу­ют дат­чи­ки ПСЧ. Од­на­ко сте­пень слу­чай­но­сти их ге­не­ра­ции долж­на быть дос­та­точ­но вы­со­ким. Иде­аль­ным ге­не­ра­то­ра­ми яв­ля­ют­ся уст­рой­ст­ва на ос­но­ве “на­ту­раль­ных” слу­чай­ных про­цес­сов. На­при­мер, поя­ви­лись се­рий­ные об­раз­цы ге­не­ра­ции клю­чей на ос­но­ве *бе­ло­го ра­дио­шу­ма*. Дру­гим слу­чай­ным ма­те­ма­ти­че­ским объ­ек­том яв­ля­ют­ся де­ся­тич­ные зна­ки иррациональных чисел, например π или *е*, которые вычисляются с помощью стандартных математических методов.

 В ИС со средними требованиями защищенности вполне приемлемы программные генераторы ключей, которые вычисляют ПСЧ как сложную функцию от текущего времени и (или) числа, введенного пользователем.

### Накопление ключей

Под *на­ко­п­ле­ни­ем клю­чей* по­ни­ма­ет­ся ор­га­ни­за­ция их хра­не­ния, уче­та и уда­ле­ния.

По­сколь­ку ключ яв­ля­ет­ся са­мым при­вле­ка­тель­ным для зло­умыш­лен­ни­ка объ­ек­том, от­кры­ваю­щим ему путь к кон­фи­ден­ци­аль­ной ин­фор­ма­ции, то во­про­сам на­ко­п­ле­ния клю­чей сле­ду­ет уде­лять осо­бое вни­ма­ние.

*Сек­рет­ные клю­чи ни­ко­гда не долж­ны за­пи­сы­вать­ся в яв­ном ви­де на но­си­те­ле, ко­то­рый мо­жет быть счи­тан или ско­пи­ро­ван.*

В дос­та­точ­но слож­ной ИС один поль­зо­ва­тель мо­жет ра­бо­тать с боль­шим объ­е­мом клю­че­вой ин­фор­ма­ции, и ино­гда да­же воз­ни­ка­ет не­об­хо­ди­мость ор­га­ни­за­ции ми­ни-баз дан­ных по клю­че­вой ин­фор­ма­ции. Та­кие ба­зы дан­ных от­ве­ча­ют за при­ня­тие, хра­не­ние, учет и уда­ле­ние ис­поль­зуе­мых клю­чей.

Итак, ка­ж­дая ин­фор­ма­ция об ис­поль­зуе­мых клю­чах долж­на хра­нить­ся в за­шиф­ро­ван­ном ви­де. Клю­чи, за­шиф­ро­вы­ваю­щие клю­че­вую ин­фор­ма­цию на­зы­ва­ют­ся *мас­тер-клю­ча­ми*. Же­ла­тель­но, что­бы мас­тер-клю­чи ка­ж­дый поль­зо­ва­тель знал наи­зусть, и не хра­нил их во­об­ще на ка­ких-ли­бо ма­те­ри­аль­ных но­си­те­лях.

Очень важ­ным ус­ло­ви­ем безо­пас­но­сти ин­фор­ма­ции яв­ля­ет­ся пе­рио­ди­че­ское об­нов­ле­ние клю­че­вой ин­фор­ма­ции в ИС. При этом пе­ре­на­зна­чать­ся долж­ны как обыч­ные клю­чи, так и мас­тер-клю­чи. В осо­бо от­вет­ст­вен­ных ИС об­нов­ле­ние клю­че­вой ин­фор­ма­ции же­ла­тель­но де­лать еже­днев­но.

Во­прос об­нов­ле­ния клю­че­вой ин­фор­ма­ции свя­зан и с треть­им эле­мен­том управ­ле­ния клю­ча­ми - рас­пре­де­ле­ни­ем клю­чей.

### Рас­пре­де­ле­ние клю­чей

Рас­пре­де­ле­ние клю­чей - са­мый от­вет­ст­вен­ный про­цесс в управ­ле­нии клю­ча­ми. К не­му предъ­яв­ля­ют­ся два тре­бо­ва­ния:

1. Опе­ра­тив­ность и точ­ность рас­пре­де­ле­ния
2. Скрыт­ность рас­пре­де­ляе­мых клю­чей.

В по­след­нее вре­мя за­ме­тен сдвиг в сто­ро­ну ис­поль­зо­ва­ния крип­то­си­стем с от­кры­тым клю­чом, в ко­то­рых про­бле­ма рас­пре­де­ле­ния клю­чей от­па­да­ет. Тем не ме­нее рас­пре­де­ле­ние клю­че­вой ин­фор­ма­ции в ИС тре­бу­ет но­вых эф­фек­тив­ных ре­ше­ний.

Рас­пре­де­ле­ние клю­чей ме­ж­ду поль­зо­ва­те­ля­ми реа­ли­зу­ют­ся дву­мя раз­ны­ми под­хо­да­ми:

1. *Пу­тем соз­да­ния од­но­го ли не­сколь­ких цен­тров рас­пре­де­ле­ния клю­чей.* Не­дос­та­ток та­ко­го под­хо­да со­сто­ит в том, что в цен­тре рас­пре­де­ле­ния из­вест­но, ко­му и ка­кие клю­чи на­зна­че­ны и это по­зво­ля­ет чи­тать все со­об­ще­ния, цир­ку­ли­рую­щие в ИС. Воз­мож­ные зло­упот­реб­ле­ния су­ще­ст­вен­но влия­ют на за­щи­ту.

2. *Пря­мой об­мен клю­ча­ми* ме­ж­ду поль­зо­ва­те­ля­ми ин­фор­ма­ци­он­ной сис­те­мы. В этом слу­чае про­бле­ма со­сто­ит в том, что­бы на­деж­но удо­сто­ве­рить под­лин­ность субъ­ек­тов.

В обо­их слу­ча­ях долж­на быть га­ран­ти­ро­ва­на под­лин­ность се­ан­са свя­зи. Это мож­но обес­пе­чить дву­мя спо­со­ба­ми:

1. *Ме­ха­низм за­про­са-от­ве­та*, ко­то­рый со­сто­ит в сле­дую­щем. Ес­ли поль­зо­ва­тель А же­ла­ет быть уве­рен­ным, что со­об­ще­ния ко­то­рый он по­лу­ча­ет от В, не яв­ля­ют­ся лож­ны­ми, он вклю­ча­ет в по­сы­лае­мое для В со­об­ще­ние не­пред­ска­зуе­мый эле­мент (за­прос). При от­ве­те поль­зо­ва­тель В дол­жен вы­пол­нить не­ко­то­рую опе­ра­цию над этим эле­мен­том (на­при­мер, до­ба­вить 1). Это не­воз­мож­но осу­ще­ст­вить за­ра­нее, так как не из­вест­но, ка­кое слу­чай­ное чис­ло при­дет в за­про­се. По­сле по­лу­че­ния от­ве­та с ре­зуль­та­та­ми дей­ст­вий поль­зо­ва­тель А мо­жет быть уве­рен, что се­анс яв­ля­ет­ся под­лин­ным. Не­дос­тат­ком это­го ме­то­да яв­ля­ет­ся воз­мож­ность ус­та­нов­ле­ния хо­тя и слож­ной за­ко­но­мер­но­сти ме­ж­ду за­про­сом и от­ве­том.

2. *Ме­ха­низм от­мет­ки вре­ме­ни (“вре­мен­ной штем­пель”).* Он под­ра­зу­ме­ва­ет фик­са­цию вре­ме­ни для ка­ж­до­го со­об­ще­ния. В этом слу­чае ка­ж­дый поль­зо­ва­тель ИС мо­жет знать, на­сколь­ко “ста­рым” яв­ля­ет­ся при­шед­шее со­об­ще­ние.

В обо­их слу­ча­ях сле­ду­ет ис­поль­зо­вать шиф­ро­ва­ние, что­бы быть уве­рен­ным, что от­вет по­слан не зло­умыш­лен­ни­ком и штем­пель от­мет­ки вре­ме­ни не из­ме­нен.

При ис­поль­зо­ва­нии от­ме­ток вре­ме­ни вста­ет про­бле­ма до­пус­ти­мо­го вре­мен­но­го ин­тер­ва­ла за­держ­ки для под­твер­жде­ния под­лин­но­сти се­ан­са. Ведь со­об­ще­ние с “вре­мен­ным штем­пе­лем” в прин­ци­пе не мо­жет быть пе­ре­да­но мгно­вен­но. Кро­ме это­го ком­пь­ю­тер­ные ча­сы по­лу­ча­те­ля и от­пра­ви­те­ля не мо­гут быть аб­со­лют­но син­хро­ни­зи­ро­ва­ны. Ка­кое за­паз­ды­ва­ние “штем­пе­ля” счи­тать по­доз­ри­тель­ным.

По­этому в ре­аль­ных ИС, на­при­мер в сис­те­мах оп­ла­ты кре­дит­ных кар­то­чек ис­поль­зу­ет­ся имен­но вто­рой ме­ха­низм ус­та­нов­ле­ния под­лин­но­сти и за­щи­ты от под­де­лок. Ис­поль­зуе­мый ин­тер­вал со­став­ля­ет от од­ной до не­сколь­ких ми­нут. Боль­шое чис­ло из­вест­ных спо­со­бов кра­жи элек­трон­ных де­нег, ос­но­ва­но на “вкли­ни­ва­нии” в этот про­ме­жу­ток с под­лож­ны­ми за­про­са­ми на сня­тии де­нег.

Для обмена ключами можно использовать криптосистемы с открытым ключом, используя тот же алгоритм RSA.

Но весьма эффективным оказался алгоритм Диффи-Хелмана, позволяющий двум пользователям без посредников обменяться ключом, который может быть использован затем для симметричного шифрования.

#### Алгоритм Диф­фи-Хелл­ма­на

Диффи и Хелман пред­ло­жи­ли для соз­да­ния крип­то­гра­фи­че­ских сис­тем с от­кры­тым клю­чом *функ­цию дис­крет­но­го воз­ве­де­ния в сте­пень*.

Не­об­ра­ти­мость пре­об­ра­зо­ва­ния в этом слу­чае обес­пе­чи­ва­ет­ся тем, что дос­та­точ­но лег­ко вы­чис­лить по­ка­за­тель­ную функ­цию в ко­неч­ном по­ле Га­луа со­стоя­щим из *p* эле­мен­тов. (*p* - ли­бо про­стое число, либо простое в любой степени). Вычисление же логарифмов в таких полях - значительно более трудоемкая операция.

Если *y*=α*x*,, 1<*x*<*p*-1, где - фиксированный элемент поля *GF(p)*, то *x*=lo*g*α *y* над *GF(p)*. Имея *x*, легко вычислить *y*. Для этого потребуется 2 ln(*x*+*y*) операций умножения.

Обратная задача вычисления *x* из *y* будет достаточно сложной. Если *p* выбрано достаточно правильно, то извлечение логарифма потребует вычислений, пропорциональных

*L(p)* = *exp* { (ln *p* ln ln *p*)0.5 }

Для обмена информацией первый пользователь выбирает случайное число *x*1, равновероятное из целых 1...*p*-1. Это число он держит в секрете, а другому пользователю посылает число

*y*1 = α*x* mod *p*

Аналогично поступает и второй пользователь, генерируя *x*2 и вычислив *y*2, отправляя его первому пользователю. В результате этого они могут вычислять *k*12 = α*x*1*x*2mod *p*.

Для того, чтобы вычислить *k*12, первый пользователь возводит *y*2 в степень *x*1. То же делает и второй пользователь. Таким образом, у обоих пользователей оказывается общий ключ *k*12, который можно использовать для шифрования информации обычными алгоритмами. В отличие от алгоритма RSA, данный алгоритм не позволяет шифровать собственно информацию.

Не зная *x*1 и *x*2, злоумышленник может попытаться вычислить *k*12, зная только перехваченные *y*1 и *y*2. Эквивалентность этой проблемы проблеме вычисления дискретного логарифма есть главный и открытый вопрос в системах с открытым ключом. Простого решения до настоящего времени не найдено. Так, если для прямого преобразования 1000-битных простых чисел требуется 2000 операций, то для обратного преобразования (вычисления логарифма в поле Галуа) - потребуется около 1030 операций.

Как видно, при всей простоте алгоритма Диффи-Хелмана, вторым его недостатком по сравнению с системой RSA является отсутствие гарантированной нижней оценки трудоемкости раскрытия ключа.

Кроме того, хотя описанный алгоритм позволяет обойти проблему скрытой передачи ключа, необходимость аутентификации остается. Без дополнительных средств, один из пользователей не может быть уверен, что он обменялся ключами именно с тем пользователем, который ему нужен. Опасность имитации в этом случае остается.

В ка­че­ст­ве обоб­ще­ния ска­зан­но­го о рас­пре­де­ле­нии клю­чей сле­ду­ет ска­зать сле­дую­щее. За­да­ча управ­ле­ния клю­ча­ми сво­дит­ся к по­ис­ку та­ко­го про­то­ко­ла рас­пре­де­ле­ния клю­чей, ко­то­рый обес­пе­чи­вал бы:

1. воз­мож­ность от­ка­за от цен­тра рас­пре­де­ле­ния клю­чей;
2. вза­им­ное под­твер­жде­ние под­лин­но­сти уча­ст­ни­ков се­ан­са;
3. под­твер­жде­ние дос­то­вер­но­сти се­ан­са ме­ха­низ­мом за­про­са-от­ве­та, ис­поль­зо­ва­ние для это­го про­грамм­ных или ап­па­рат­ных средств;
4. ис­поль­зо­ва­ние при об­ме­не клю­ча­ми ми­ни­маль­но­го чис­ла со­об­ще­ний.

## Про­бле­мы и пер­спек­ти­вы крип­то­гра­фи­че­ских сис­тем

### Шиф­ро­ва­ние боль­ших со­об­ще­ний и по­то­ков дан­ных

Эта про­бле­ма поя­ви­лась срав­ни­тель­но не­дав­но с по­яв­ле­ни­ем средств *муль­ти­ме­диа* и се­тей с вы­со­кой про­пу­ск­ной спо­соб­но­стью, обес­пе­чи­ваю­щих пе­ре­да­чу муль­ти­ме­дий­ных дан­ных.

До сих пор го­во­ри­лось о за­щи­те со­об­ще­ний. При этом под ни­ми под­ра­зу­ме­ва­лась ско­рее некоторая тек­сто­вая или сим­во­ли­че­ская ин­фор­ма­ция. Од­на­ко в со­вре­мен­ных ИС и ин­фор­ма­ци­он­ных сис­те­мах на­чи­на­ют при­ме­нять­ся тех­но­ло­гии, ко­то­рые тре­бу­ют пе­ре­да­чи су­ще­ст­вен­но боль­ших объ­е­мов дан­ных. Сре­ди та­ких тех­но­ло­гий:

1. фак­си­миль­ная, ви­део и ре­че­вая связь;
2. го­ло­со­вая поч­та;
3. сис­те­мы ви­део­кон­фе­рен­ций.

Объ­ем пе­ре­да­вае­мой ин­фор­ма­ции раз­ных ти­пов мож­но пред­ста­вить на ус­лов­ной диа­грам­ме.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Объем видеоинформации |  |  |  |  |  |  |  |  |  |
| графика |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| звуктекст |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Так как пе­ре­да­ча оциф­ро­ван­ной зву­ко­вой, гра­фи­че­ской и ви­деоин­фор­ма­ции во мно­гих слу­ча­ях тре­бу­ет кон­фи­ден­ци­аль­но­сти, то воз­ни­ка­ет про­бле­ма шиф­ро­ва­ния ог­ром­ных ин­фор­ма­ци­он­ных мас­си­вов. Для ин­те­рак­тив­ных сис­тем ти­па те­ле­кон­фе­рен­ций, ве­де­ния ау­дио или ви­деосвя­зи, та­кое шиф­ро­ва­ние долж­но осу­ще­ст­в­лять­ся в ре­аль­ном мас­шта­бе вре­ме­ни и по воз­мож­но­сти быть “про­зрач­ным” для поль­зо­ва­те­лей.

Это немыс­ли­мо без ис­поль­зо­ва­ния со­вре­мен­ных тех­но­ло­гий шиф­ро­ва­ния.

Наи­бо­лее рас­про­стра­нен­ным яв­ля­ет­ся *по­то­ко­вое шиф­ро­ва­ние* дан­ных. Ес­ли в опи­сан­ных ра­нее крип­то­си­сте­мах пред­по­ла­га­лось, что на вхо­де име­ет­ся не­ко­то­рое ко­неч­ное со­об­ще­ние, к ко­то­ро­му и при­ме­ня­ет­ся крип­то­гра­фи­че­ский ал­го­ритм, то в сис­те­мах с по­то­ко­вым шиф­ро­ва­ни­ем прин­цип дру­гой.

Сис­те­ма за­щи­ты не ждет, ко­гда за­кон­чит­ся пе­ре­да­вае­мое со­об­ще­ние, а сра­зу же осу­ще­ст­в­ля­ет его шиф­ро­ва­ние и пе­ре­да­чу.

Пользователь А

шифрованный

поток

сообщение

(поток)

шифратор

КЛЮЧ

(ПСП)

*Потоковое шифрование данных*

Наи­бо­лее оче­вид­ным яв­ля­ет­ся по­би­то­вое сло­же­ние вхо­дя­щей по­сле­до­ва­тель­но­сти (со­об­ще­ния) с не­ко­то­рым бес­ко­неч­ным или пе­рио­ди­че­ским клю­чом, по­лу­чае­мым на­при­мер от ге­не­ра­то­ра ПСП[[18]](#footnote-18). Примером стандарта потокового шифрования является RC4, разработанный Ривестом. Однако, технические подробности этого алгоритма держатся в секрете[[19]](#footnote-19).

Дру­гим, ино­гда бо­лее эф­фек­тив­ным ме­то­дом по­то­ко­во­го шиф­ро­ва­ния яв­ля­ет­ся *шиф­ро­ва­ние бло­ка­ми*. Т.е. на­ка­п­ли­ва­ет­ся фик­си­ро­ван­ный объ­ем ин­фор­ма­ции (блок), а за­тем пре­об­ра­зо­ван­ный не­ко­то­рым крип­то­гра­фи­че­ским ме­то­дом пе­ре­да­ет­ся в ка­нал свя­зи.

### Ис­поль­зо­ва­ние “блу­ж­даю­щих клю­чей”

Как бы­ло не­од­но­крат­но от­ме­че­но, про­бле­ма рас­пре­де­ле­ния клю­чей яв­ля­ет­ся наи­бо­лее ост­рой в круп­ных ин­фор­ма­ци­он­ных сис­те­мах. От­час­ти эта про­бле­ма ре­ша­ет­ся (а точ­нее сни­ма­ет­ся) за счет ис­поль­зо­ва­ния от­кры­тых клю­чей. Но наи­бо­лее на­деж­ные крип­то­си­сте­мы с от­кры­тым клю­чом ти­па RSA дос­та­точ­но тру­до­ем­ки, а для шиф­ро­ва­ния муль­ти­ме­дий­ных дан­ных и во­все не при­год­ны.

Ори­ги­наль­ные ре­ше­ния про­бле­мы “ блу­ж­даю­щих клю­чей” ак­тив­но раз­ра­ба­ты­ва­ют­ся спе­циа­ли­ста­ми. Эти сис­те­мы яв­ля­ют­ся не­ко­то­рым ком­про­мис­сом ме­ж­ду сис­те­ма­ми с от­кры­ты­ми клю­ча­ми и обыч­ны­ми ал­го­рит­ма­ми, для ко­то­рых тре­бу­ет­ся на­ли­чие од­но­го и то­го же клю­ча у от­пра­ви­те­ля и по­лу­ча­те­ля.

Идея ме­то­да дос­та­точ­но про­ста.

По­сле то­го, как ключ ис­поль­зо­ван в од­ном се­ан­се по не­ко­то­ро­му пра­ви­лу он сме­ня­ет­ся дру­гим. Это пра­ви­ло долж­но быть из­вест­но и от­пра­ви­те­лю, и по­лу­ча­те­лю. Зная пра­ви­ло, по­сле по­лу­че­ния оче­ред­но­го со­об­ще­ния по­лу­ча­тель то­же ме­ня­ет ключ. Ес­ли пра­ви­ло сме­ны клю­чей ак­ку­рат­но со­блю­да­ет­ся и от­пра­ви­те­лем и по­лу­ча­те­лем, то в ка­ж­дый мо­мент вре­ме­ни они име­ют оди­на­ко­вый ключ. По­сто­ян­ная сме­на клю­ча за­труд­ня­ет рас­кры­тие ин­фор­ма­ции зло­умыш­лен­ни­ком.

Ос­нов­ная за­да­ча в реа­ли­за­ции это­го ме­то­да - вы­бор эф­фек­тив­но­го пра­ви­ла сме­ны клю­чей. Наи­бо­лее про­стой путь - ге­не­ра­ция слу­чай­но­го спи­ска клю­чей. Сме­на клю­чей осу­ще­ст­в­ля­ет­ся в по­ряд­ке спи­ска. Од­на­ко, оче­вид­но спи­сок при­дет­ся ка­ким-то об­ра­зом пе­ре­да­вать.

Дру­гой ва­ри­ант - ис­поль­зо­ва­ние ма­те­ма­ти­че­ских ал­го­рит­мов, ос­но­ван­ных на так на­зы­вае­мых *пе­ре­би­раю­щих по­сле­до­ва­тель­но­стях*. На мно­же­ст­ве клю­чей пу­тем од­ной и той же опе­ра­ции над эле­мен­том по­лу­ча­ет­ся дру­гой эле­мент. По­сле­до­ва­тель­ность этих опе­ра­ций по­зво­ля­ет пе­ре­хо­дить от од­но­го эле­мен­та к дру­го­му, по­ка не бу­дет пе­ре­бра­но все мно­же­ст­во.

Наи­бо­лее дос­туп­ным яв­ля­ет­ся ис­поль­зо­ва­ние по­лей Га­луа. За счет воз­ве­де­ния в сте­пень по­ро­ж­даю­ще­го эле­мен­та мож­но по­сле­до­ва­тель­но пе­ре­хо­дить от од­но­го чис­ла к дру­го­му. Эти чис­ла при­ни­ма­ют­ся в ка­че­ст­ве клю­чей.

Клю­че­вой ин­фор­ма­ци­ей в дан­ном слу­чае яв­ля­ет­ся ис­ход­ный эле­мент, ко­то­рый пе­ред на­ча­лом свя­зи дол­жен быть из­вес­тен и от­пра­ви­те­лю и по­лу­ча­те­лю.

На­деж­ность та­ких ме­то­дов долж­на быть обес­пе­че­на с уче­том из­вест­нос­ти зло­умыш­лен­ни­ку ис­поль­зуе­мо­го пра­ви­ла сме­ны клю­чей.

Ин­те­рес­ной и пер­спек­тив­ной за­да­чей яв­ля­ет­ся реа­ли­за­ция ме­то­да “блу­ж­даю­щих клю­чей” не для двух або­нен­тов, а для дос­та­точ­но боль­шой се­ти, ко­гда со­об­ще­ния пересылаются ме­ж­ду все­ми уча­ст­ни­ка­ми.

### Шиф­ро­ва­ние, ко­ди­ро­ва­ние и сжа­тие ин­фор­ма­ции

Эти три ви­да пре­об­ра­зо­ва­ния ин­фор­ма­ции ис­поль­зу­ют­ся в раз­ных це­лях, что мож­но пред­ста­вить в таб­ли­це.

|  |  |  |
| --- | --- | --- |
| Вид преобразования  | Цель | Изменение объема информации после преобразования. |
| *Шифрование* | 1. пе­ре­да­ча кон­фи­ден­ци­аль­ной ин­фор­ма­ции;
2. обес­пе­че­ние ау­тен­ти­фи­ка­ции и за­щи­ты от пред­на­ме­рен­ных из­ме­не­ний;
 | обыч­но не из­ме­ня­ет­ся, уве­ли­чи­ва­ет­ся лишь в циф­ро­вых сиг­на­ту­рах и под­пи­сях |
| *По­ме­хо­устой­чи­вое ко­ди­ро­ва­ние* | 1. за­щи­та от ис­ка­же­ния по­ме­ха­ми в ка­на­лах свя­зи
 | уве­ли­чи­ва­ет­ся |
| *Сжа­тие (ком­прес­сия)* | 1. со­кра­ще­ние объ­е­ма передаваемых или хра­ни­мых дан­ных
 | умень­ша­ет­ся |

Как видно эти три вида преобразования информации отчасти дополняют друг друга и их комплексное использование поможет эф­фек­тив­но ис­поль­зо­вать ка­на­лы свя­зи для на­деж­ной за­щи­ты пре­да­вае­мой ин­фор­ма­ции.

Осо­бен­но ин­те­рес­ным пред­став­ля­ет­ся *воз­мож­ность объ­е­ди­не­ния ме­то­дов ко­ди­ро­ва­ния и шиф­ро­ва­ния*. Мож­но ут­вер­ждать, что по су­ти ко­ди­ро­ва­ние - это эле­мен­тар­ное шиф­ро­ва­ние, а шиф­ро­ва­ние - это эле­мен­тар­ное по­ме­хо­устой­чи­вое ко­ди­ро­ва­ние.

Другая возможность - *комбинирование алгоритмов шифрования и сжатия информации*. Задача сжатия состоит в том, чтобы преобразовать сообщение в пределах одного и того же алфавита таким образом, чтобы его длина (количество букв алфавита) стала меньше, но при этом сообщение можно было восстановить без использования какой-то дополнительной информации. Наиболее популярные алгоритмы сжатия - RLE, коды Хаффмана, алгоритм Лемпеля-Зива. Для сжатия графической и видеоинформации используются алгоритмы JPEG и MPEG.

Главное достоинство алгоритмов сжатия с точки зрения криптографии состоит в том, что они изменяют статистику входного текста в сторону ее выравнивания[[20]](#footnote-20). Так, в обычном тексте, сжатом с помощью эффективного алгоритма все символы имеют одинаковые частотные характеристики и даже использование простых системы шифрования сделают текст недоступным для криптоанализа.

Раз­ра­бот­ка и реа­ли­за­ция та­ких уни­вер­саль­ных ме­то­дов - пер­спек­ти­ва со­вре­мен­ных информационных сис­тем[[21]](#footnote-21).

### Реализация криптографических методов

Про­бле­ма реа­ли­за­ции ме­то­дов за­щи­ты ин­фор­ма­ции име­ет два ас­пек­та:

1. раз­ра­бот­ку средств, реа­ли­зую­щих крип­то­гра­фи­че­ские ал­го­рит­мы,
2. ме­то­ди­ку использования этих средств.

Ка­ж­дый из рас­смот­рен­ных крип­то­гра­фи­че­ских ме­то­дов могут быть реализованы ли­бо *про­грамм­ным*, ли­бо *ап­па­рат­ным* спо­со­бом.

Воз­мож­ность про­грамм­ной реа­ли­за­ции обу­слав­ли­ва­ет­ся тем, что все ме­то­ды крип­то­гра­фи­че­ско­го преобразования фор­маль­ны и мо­гут быть пред­став­ле­ны в виде ко­неч­ной алгоритмической про­це­ду­ры.

При ап­па­рат­ной реа­ли­за­ции все про­це­ду­ры шиф­ро­ва­ния и де­шиф­ро­ва­ния вы­пол­ня­ют­ся спе­ци­аль­ны­ми элек­трон­ны­ми схе­ма­ми. Наи­боль­шее распространение по­лу­чи­ли мо­ду­ли, реа­ли­зую­щие ком­би­ни­ро­ван­ные ме­то­ды.

 При этом не­пре­мен­ным ком­по­нен­тов всех ап­па­рат­но реализуемых ме­то­дов яв­ля­ет­ся *гам­ми­ро­ва­ние*. Это объ­яс­ня­ет­ся тем, что ме­тод гам­ми­ро­ва­ния удач­но со­че­та­ет в се­бе вы­со­кую криптостойкость и про­сто­ту реа­ли­за­ции.

Наи­бо­лее час­то в ка­че­ст­ве ге­не­ра­то­ра ис­поль­зу­ет­ся ши­ро­ко известный ре­гистр сдви­га с об­рат­ны­ми свя­зя­ми (ли­ней­ны­ми или не­ли­ней­ны­ми). Ми­ни­маль­ный пе­ри­од по­ро­ж­дае­мой по­сле­до­ва­тель­но­сти ра­вен 2N-1 бит. Для по­вы­ше­ния ка­че­ст­ва ге­не­ри­руе­мой по­сле­до­ва­тель­но­сти мож­но пре­ду­смот­реть спе­ци­аль­ный блок управ­ле­ния ра­бо­той ре­ги­ст­ра сдви­га. Та­кое управ­ле­ние мо­жет за­клю­чать­ся, на­при­мер, в том, что по­сле шифрования оп­ре­де­лен­но­го объ­е­ма ин­фор­ма­ции со­дер­жи­мое ре­ги­ст­ра сдви­га цик­ли­че­ски из­ме­ня­ет­ся.

Другая воз­мож­ность улуч­ше­ния ка­че­ст­ва гам­ми­ро­ва­ния за­клю­ча­ет­ся в использовании не­ли­ней­ных об­рат­ных свя­зей. При этом улуч­ше­ние дос­ти­га­ет­ся не за счет уве­ли­че­ния дли­ны гам­мы, а за счет ус­лож­не­ния закона ее фор­ми­ро­ва­ния, что су­ще­ст­вен­но ус­лож­ня­ет крип­тоа­на­лиз.

Боль­шин­ст­во за­ру­беж­ных се­рий­ных средств шиф­ро­ва­ния ос­но­ва­но на аме­ри­кан­ском стан­дар­те DES. Оте­че­ст­вен­ные же раз­ра­бот­ки, та­кие как, на­при­мер, уст­рой­ст­во КРИП­ТОН, ис­поль­зу­ет оте­че­ст­вен­ный стан­дарт шиф­ро­ва­ния.

Ос­нов­ным дос­то­ин­ст­вом про­грамм­ных ме­то­дов реа­ли­за­ции за­щи­ты яв­ля­ет­ся их гиб­кость, т.е. воз­мож­ность бы­ст­ро­го из­ме­не­ния ал­го­рит­мов шиф­ро­ва­ния.

Ос­нов­ным же не­дос­тат­ком про­грамм­ной реа­ли­за­ции яв­ля­ет­ся су­ще­ст­вен­но мень­шее бы­ст­ро­дей­ст­вие по срав­не­нию с ап­па­рат­ны­ми сред­ст­ва­ми (при­мер­но в 10 раз).

В по­след­нее вре­мя ста­ли по­яв­лять­ся ком­би­ни­ро­ван­ные сред­ст­ва шифрования, так на­зы­вае­мые про­грамм­но-ап­па­рат­ные сред­ст­ва. В этом слу­чае в ком­пь­ю­те­ре ис­поль­зу­ет­ся свое­об­раз­ный “крип­то­гра­фи­че­ский со­про­цес­сор”[[22]](#footnote-22) - вы­чис­ли­тель­ное уст­рой­ст­во, ори­ен­ти­ро­ван­ное на вы­пол­не­ние крип­то­гра­фи­че­ских опе­ра­ций (сло­же­ние по мо­ду­лю, сдвиг и т.д.). Ме­няя про­грамм­ное обес­пе­че­ния для та­ко­го уст­рой­ст­ва, мож­но вы­би­рать тот или иной ме­тод шиф­ро­ва­ния. Та­кой ме­тод объ­е­ди­ня­ет в се­бе дос­то­ин­ст­ва про­грамм­ных и ап­па­рат­ных ме­то­дов.

Та­ким об­ра­зом, вы­бор ти­па реа­ли­за­ции крип­то­за­щи­ты для кон­крет­ной ИС в су­ще­ст­вен­ной ме­ре за­ви­сит от ее осо­бен­но­стей и дол­жен опи­рать­ся на все­сто­рон­ний ана­лиз тре­бо­ва­ний, предъ­яв­ляе­мых к сис­те­ме за­щи­ты ин­фор­ма­ции.

## За­клю­че­ние

В книге сде­лан об­зор наи­бо­лее рас­про­стра­нен­ных в на­стоя­щее вре­мя ме­то­дов крип­то­гра­фи­че­ской за­щи­ты ин­фор­ма­ции.

 Выбор для кон­крет­ных ИС дол­жен быть ос­но­ван на глу­бо­ком ана­ли­зе сла­бых и силь­ных сто­рон тех или иных ме­то­дов за­щи­ты. Обос­но­ван­ный вы­бор той или иной сис­те­мы за­щи­ты в об­щем-то дол­жен опи­рать­ся на ка­кие-то *кри­те­рии эф­фек­тив­но­сти*. К со­жа­ле­нию, до сих пор не раз­ра­бо­та­ны под­хо­дя­щие ме­то­ди­ки оцен­ки эф­фек­тив­но­сти крип­то­гра­фи­че­ских сис­тем.

Наи­бо­лее про­стой кри­те­рий та­кой эф­фек­тив­но­сти *- ве­ро­ят­ность рас­кры­тия клю­ча* или *мощ­ность мно­же­ст­ва клю­чей (М).* По сути это то же самое, что и *криптостойкость*. Для ее численной оценки можно использовать также и сложность раскрытия шифра путем перебора всех ключей.

Од­на­ко, этот кри­те­рий не учи­ты­ва­ет других важных *требований к криптосистемам*:

1. невоз­мож­ность рас­кры­тия или ос­мыс­лен­ной мо­ди­фи­ка­ции ин­фор­ма­ции на ос­но­ве ана­ли­за ее струк­ту­ры,
2. со­вер­шен­ст­во ис­поль­зуе­мых про­то­ко­лов за­щи­ты,
3. минимальный объ­ем ис­поль­зуе­мой клю­че­вой ин­фор­ма­ции,
4. минимальная слож­ность реа­ли­за­ции (в ко­ли­че­ст­ве ма­шин­ных опе­ра­ций), ее стои­мость,
5. высокая опе­ра­тив­ность.

Же­ла­тель­но ко­неч­но ис­поль­зо­ва­ние не­ко­то­рых ин­те­граль­ных по­ка­за­те­лей, учи­ты­ваю­щих ука­зан­ные фак­то­ры.

Для уче­та стои­мо­сти, тру­до­ем­ко­сти и объ­е­ма клю­че­вой ин­фор­ма­ции мож­но ис­поль­зо­вать удель­ные по­ка­за­те­ли - от­но­ше­ние ука­зан­ных па­ра­мет­ров к мощ­но­сти мно­же­ст­ва клю­чей шиф­ра.

Час­то бо­лее эф­фек­тив­ным при вы­бо­ре и оцен­ке крип­то­гра­фи­че­ской сис­те­мы яв­ля­ет­ся ис­поль­зо­ва­ние экс­перт­ных оце­нок и ими­та­ци­он­ное мо­де­ли­ро­ва­ние.

В лю­бом слу­чае вы­бран­ный ком­плекс крип­то­гра­фи­че­ских ме­то­дов дол­жен со­че­тать как удоб­ст­во, гиб­кость и опе­ра­тив­ность ис­поль­зо­ва­ния, так и на­деж­ную за­щи­ту от зло­умыш­лен­ни­ков цир­ку­ли­рую­щей в ИС ин­фор­ма­ции.

*Содержание*

От автора 1

Введение 2

Терминология 3

Требования к криптосистемам 4

Симметричные криптосистемы 6

Перестановки 7

Системы подстановок 7

Гаммирование 13

Датчики ПСЧ 13

Стандарт шифрования данных ГОСТ 28147-89 15

Системы с открытым ключом 19

Алгоритм RSA 20

Криптосистема Эль-Гамаля 24

Криптосистемы на основе эллиптических уравнений 25

Электронная подпись 26

Электронная подпись на основе алгоритма RSA 27

Цифровая сигнатура 29

Управление ключами 32

Генерация ключей 32

Накопление ключей 32

Распределение ключей 33

Проблемы и перспективы криптографических систем 36

Шифрование больших сообщений и потоков данных 36

Использование “блуждающих ключей” 37

Шифрование, кодирование и сжатие информации 39

Реализация криптографических методов 40

Заключение 42

1. IMHO [↑](#footnote-ref-1)
2. Здесь и далее **m** - объем используемого алфавита. [↑](#footnote-ref-2)
3. *n*-граммой называется последовательность из *n* символов алфавита. [↑](#footnote-ref-3)
4. К вопросу о том, существует ил не существует абсолютно надежная криптосистема. [↑](#footnote-ref-4)
5. Материал предоставлен Ю. Г. Писаревым [↑](#footnote-ref-5)
6. ГОСТ 28147-89 закрыт грифом ДСП поэтому дальнейшее изложение сделано по изданию Спесивцев А.В. и др. «Защита информации в персональных ЭВМ», М., Радио и связь, 1992. [↑](#footnote-ref-6)
7. В настоящее время он возглавляет компанию RSA Data Security [↑](#footnote-ref-7)
8. Например, в нашумевшей программе PGP [↑](#footnote-ref-8)
9. В браузерах Интернет от Microsoft и Netscape [↑](#footnote-ref-9)
10. В теории чисел показано, что вероятность того, что число порядка n будет простым составляет 1/ln n [↑](#footnote-ref-10)
11. Данные оценки сделаны с учетом развития вычислительной техники вплоть до 2004 года. [↑](#footnote-ref-11)
12. Однако общего мнения по поводу предпочтительности того или иного метода нет. [↑](#footnote-ref-12)
13. В РФ принятые стандарты цифровой подписи Р38 и Р39, также как и ГОСТ 28147-89 имеют гриф ДСП [↑](#footnote-ref-13)
14. При этом разделяют *слабую* и *сильную* однозначность. При слабой однозначности для заданного значения T практически невозможно отыскать другой текст Т’, для которого *H(k, T) = H(k, T’). При сильной однозначности для любого текста T невозможно найти другой подходящий текст, имеющий то же значение хэш-функции.*  [↑](#footnote-ref-14)
15. Факт теории вероятностей: в группе из 23 человек с вероятностью больше 0.5 два и более человека родились в одно и то же число. [↑](#footnote-ref-15)
16. В отличие от хэш-функции - этот класс преобразований предполагает вычисление для аргументов *фиксированной длины* также фиксированных по длине значений. [↑](#footnote-ref-16)
17. Государственная программа США, предполагающая централизованное хранение всех ключей, используемых организациями а частными лицами. [↑](#footnote-ref-17)
18. Отчасти это метод похож на гаммирование и информацию о способах генерации ПСП можно почерпнуть из соответствующей главы. Но важным отличием потокового шифрования является то, что шифрованию подвергаются не символы сообщения, а отдельные биты. [↑](#footnote-ref-18)
19. Данный алгоритм является собственностью RSA Data Security, и на его экспорт правительством США наложены серьезные ограничения. [↑](#footnote-ref-19)
20. Принципиально важно с точки зрения криптостойкости, чтобы сначала осуществлялось сжатие информации а потом шифрование, но не наоборот. [↑](#footnote-ref-20)
21. Так, в криптографическом пакете PGP перед шифрованием информации происходит ее сжатие по алгоритму, лицензированному у PKWARE. [↑](#footnote-ref-21)
22. А то и просто специализированный шифровальный микропроцессор как, например, Clipper/ [↑](#footnote-ref-22)