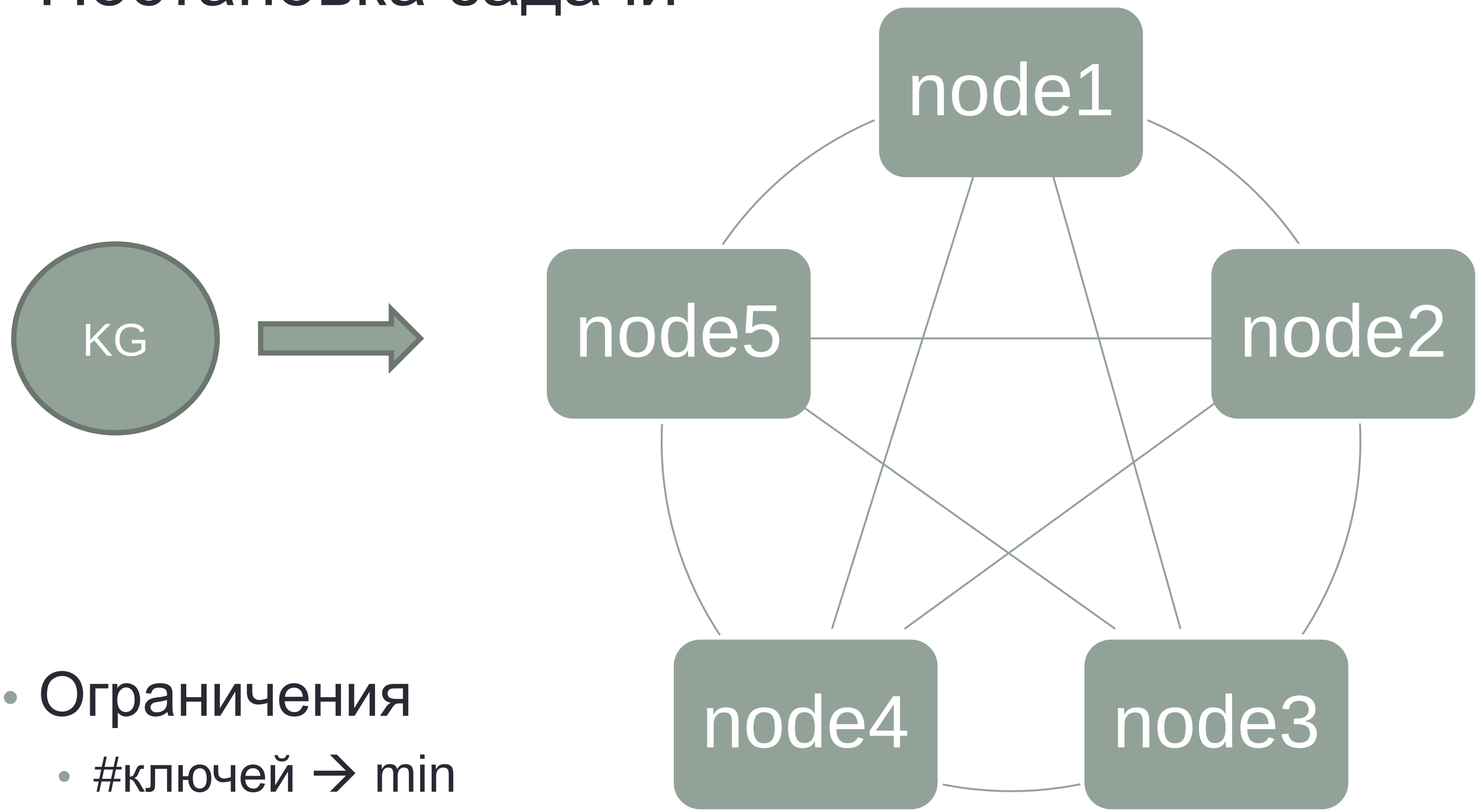
УПРАВЛЕНИЕ КЛЮЧАМИ-2

Распределение ключей

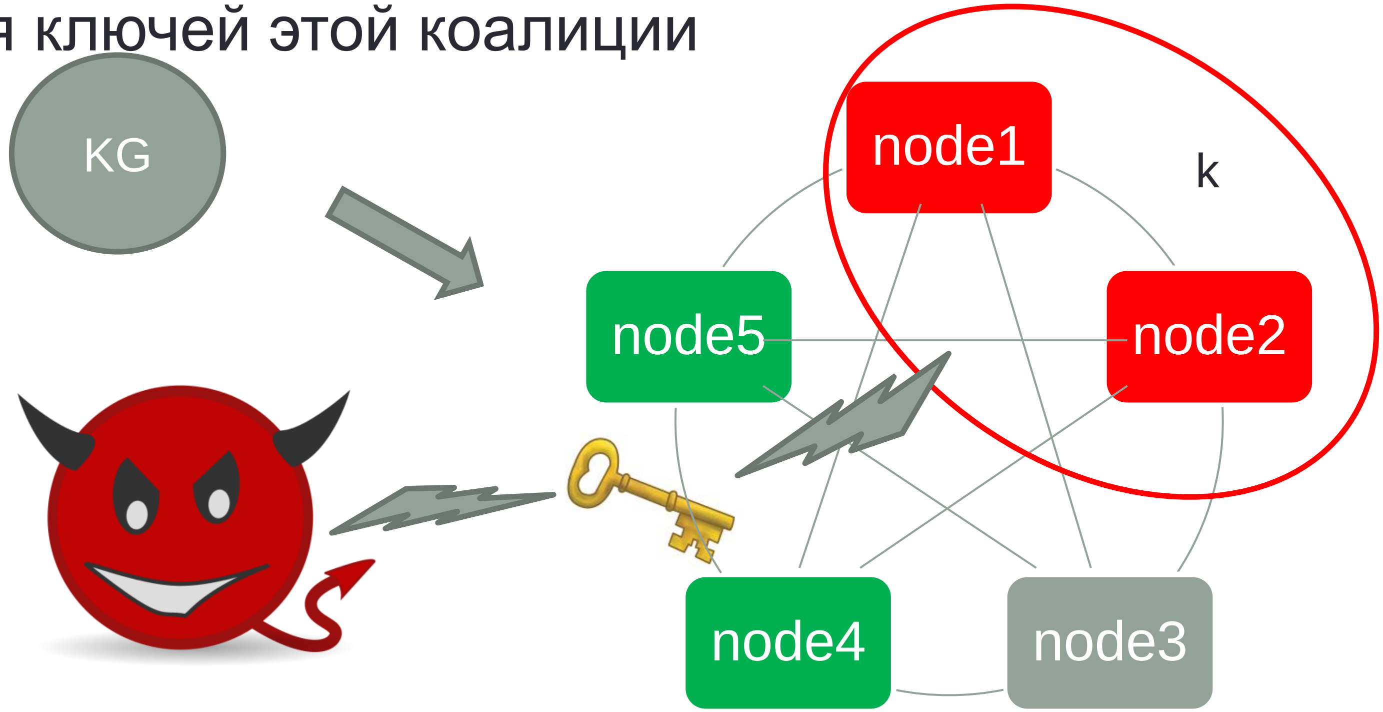
- Постановка задачи



- Ограничения
  - #ключей → min
  - Нельзя вычислить чужой ключ

k-надежность схемы распределения ключей

- Система распределения ключей k-надежна, если никакая коалиция из k и менее пользователей не может угадать ключ какой-либо пары пользователей, не входящих в коалицию, лучше, чем алгоритм без знания ключей этой коалиции



K-надежность

$K=\{k_1, k_2, \dots, k_s\}$ – пул ключей

- Каждый участник *i* получает K_i подмножество K

- Система распределения ключей k-надежна, если

- Для любой коалиции С: $|C| < k$ и стратегии $F(\cup(K_i, i \in C)) = x$, существует алгоритм A, которому неизвестно ни одно K_i:

- $|\Pr\{F(\bigcup_{i \in C} K_i) = k_j : k_j \in K \setminus ((\bigcup_{i \in C} K_i))\} - \Pr\{A(\quad) = k_j : k_j \in K \setminus ((\bigcup_{i \in C} K_i))\}|$ -- negligible

Способы построения k-надежных схем

- Основанные на комбинаторных объектах(SIS: Set intersection scheme, блок-схемы, CFF: Cover free families, проективные плоскости)
  - Объекты существуют только для небольшого набора параметров

- Вероятностные подходы

- Схема Блума (Blom).

Вероятностный подход

- P :размер пула ключей, k = кол-во ключей у одного пользователя

- Pr[Два пользователя получат один общий ключ]
- = 1 - Pr[множества ключей 2-х пользователей не пересекаются]
- = 1 - (C(P, k) / C(P, k)) × (C(k, 0) × C(P-k, k) / C(P, k))
- = $1 - \frac{k!(P-k)!(P-k)!}{P!k!(P-2k)!}$

- По формуле Стирлинга получим:

- Pr = $1 - \frac{(1-\frac{k}{P})^{2(P-k+\frac{1}{2})}}{(1-\frac{2k}{P})^{(P-2k+\frac{1}{2})}}$

Примеры параметров вероятностной схемы

- Example1:
- P=1000 , k=100
- Pr = $1 - \dfrac{\left(1 - \frac{100}{1000}\right)^{2\left(1000 - 100 + \frac{1}{2}\right)}}{\left(1 - \frac{200}{1000}\right)^{\left(1000 - 200 + \frac{1}{2}\right)}}$ = 1 - 3.8972×e^{-83}/ 2.6517×e^{-78} = 1
- Example2:
- P=1000 , k=10
- Pr = $1 - \dfrac{\left(1 - \frac{10}{1000}\right)^{2\left(1000 - 10 + \frac{1}{2}\right)}}{\left(1 - \frac{20}{1000}\right)^{\left(1000 - 20 + \frac{1}{2}\right)}}$ = 1 - 2.2559×e^{-9} / 2.4955×e^{-9} =
- 1 - 0.9039 = 0.0961

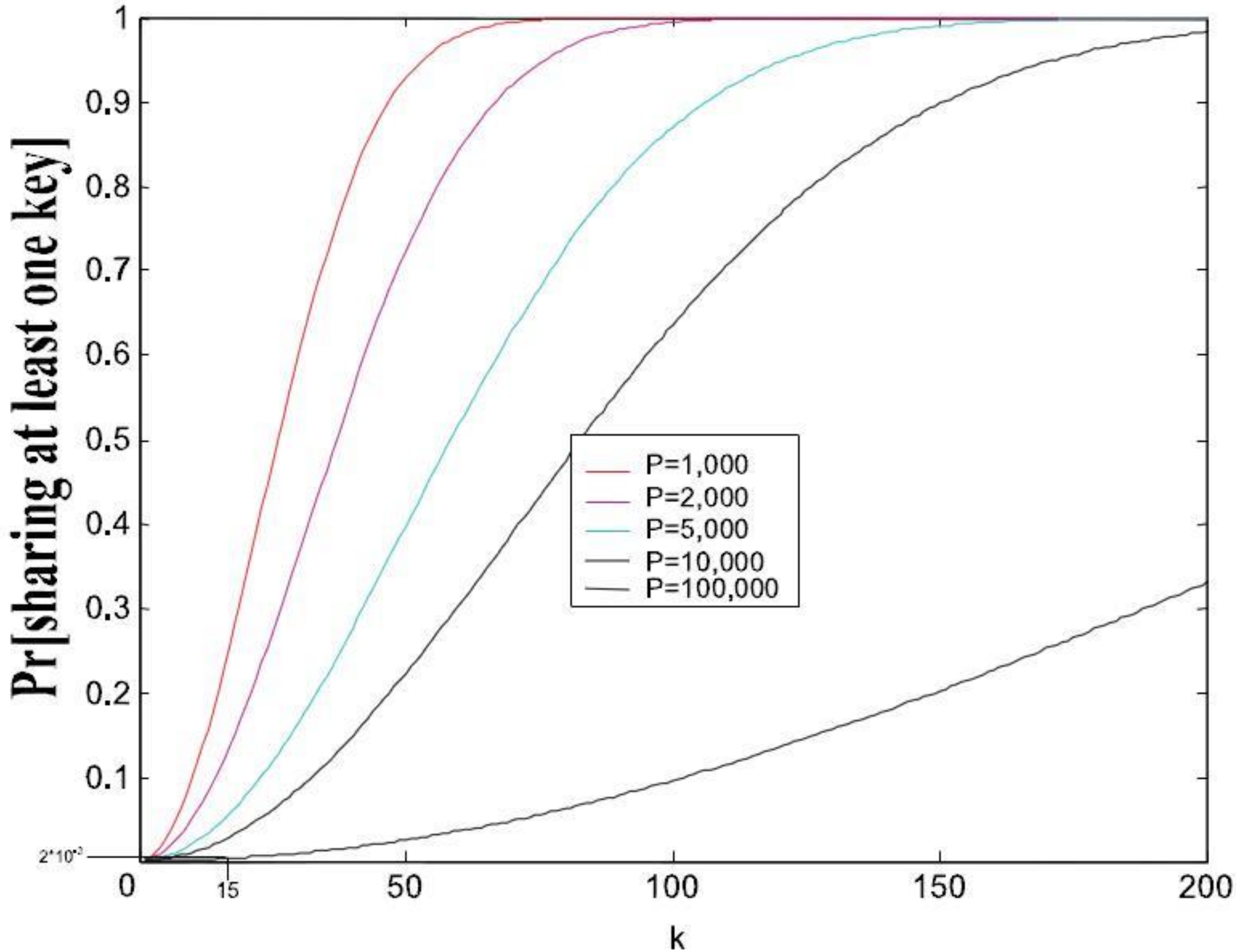

Pr[sharing at least one key]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
15
50
100
150
200
k
P=1,000
P=2,000
P=5,000
P=10,000
P=100,000

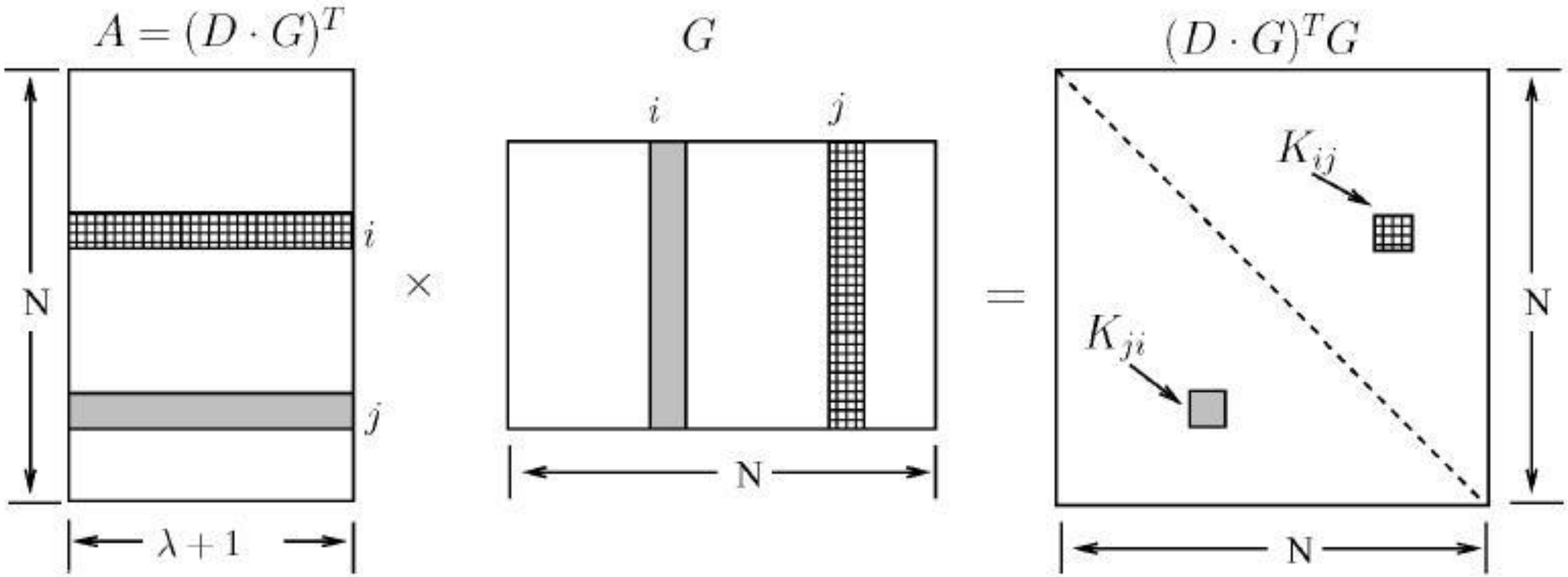
Схема Блума

- Нижняя оценка (теоретико-информационная): Для каждой k-надежной схемы, раздающей ключи по m бит, у каждого клиента должно быть не меньше (k+1)m бит секретной информации.

- Оптимальность следует из границы существования MDS кодов (кодов, лежащих на границе Синглтона)

- Граница Синглтона: Дан код длины n, размерности k, тогда расстояние этого кода $d<=n-k+1$

Описание схемы



D : Симметричная матрица над полем GF(q) $(\lambda+1)\times(\lambda+1)$

G : матрица Вандермонда $(\lambda+1)\times N$
Строка i: ключ i-того пользователя
$K_{ij} = K_{ji}$ – парный ключ i и j

$$G = \begin{bmatrix} s & s^2 & s^3 & \cdots & s^N \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^N)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & \cdots & (s^N)^\lambda \end{bmatrix}$$

Пример работы системы

Параметры системы: N=2 , λ=2 , GF(7)

$$D \cdot G = \begin{bmatrix} 1 & 6 & 2 \\ 6 & 3 & 5 \\ 2 & 5 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 3 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 4 & 4 \\ 0 & 6 \end{bmatrix} \mod 7$$

$$A = (D \cdot G)^T = \begin{bmatrix} 2 & 0 \\ 4 & 4 \\ 0 & 6 \end{bmatrix}^T = \begin{bmatrix} 2 & 4 & 0 \\ 0 & 4 & 6 \end{bmatrix}$$

$$A \cdot G = \begin{bmatrix} 2 & 4 & 0 \\ 0 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 3 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 3 & 4 \end{bmatrix} \mod 7$$

$K_{12} = K_{21} = 3$

Разделение секрета

- Мотивация:
  - Надежное и безопасное хранение секретных ключей
  - Совместный доступ к секрету

- Постановка задачи:
- Есть n – пользователей системы, которые хотят представить общий секрет S в виде набора значений SS_i, $0<i<n$, таким образом, чтобы $S = F(SS_1, \ldots, SS_n)$
- При этом никое подмножество SS_i не дает никакой информации о S.

Простейшая схема

- Аддитивное разделение секрета
- Пусть SS_i, 0<i<n-1 – случайные числа из заданного диапазона, а SS_n = S XOR SS_1 … XOR Ss_{n-1}
- Тогда n проекций гарантируют восстановление, а любое их подмножество не несет никакой информации
- Недостаток: Потеря любой части – потеря всего секрета

Пороговое разделение секрета

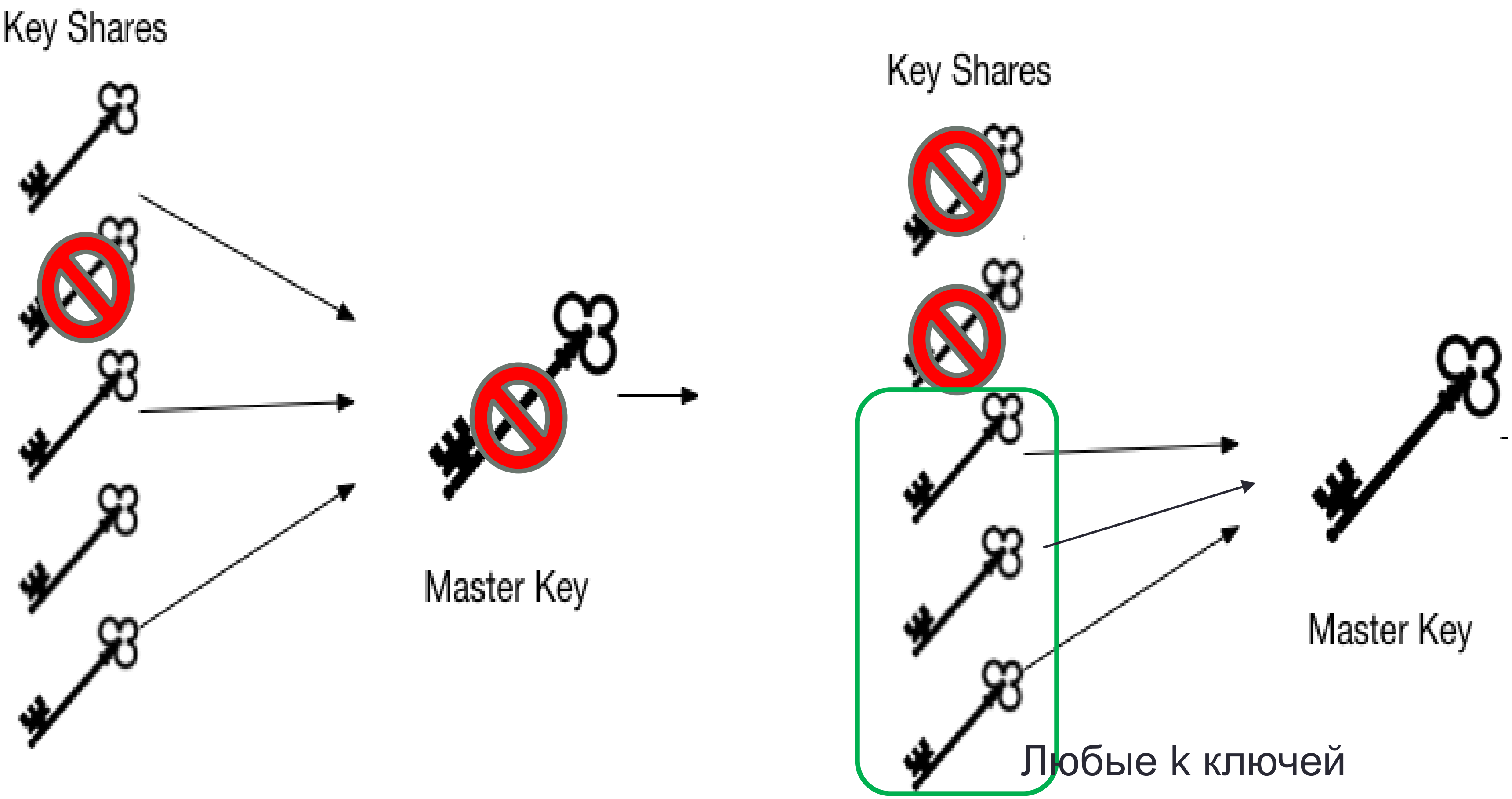


Схема Шамира

- Секрет — это число a_0.
- Центр выбирает случайные числа $a_1, \ldots, a_{k-1}$ и определяет многочлен

$$f = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}.$$

- Центр раздаёт участникам числа $f(1), f(2), \ldots, f(n)$ (или значения в любых других точках).
- Любые k участников теперь могут воспользоваться интерполяцией по Лагранжу, а любые $k-1$ не могут.

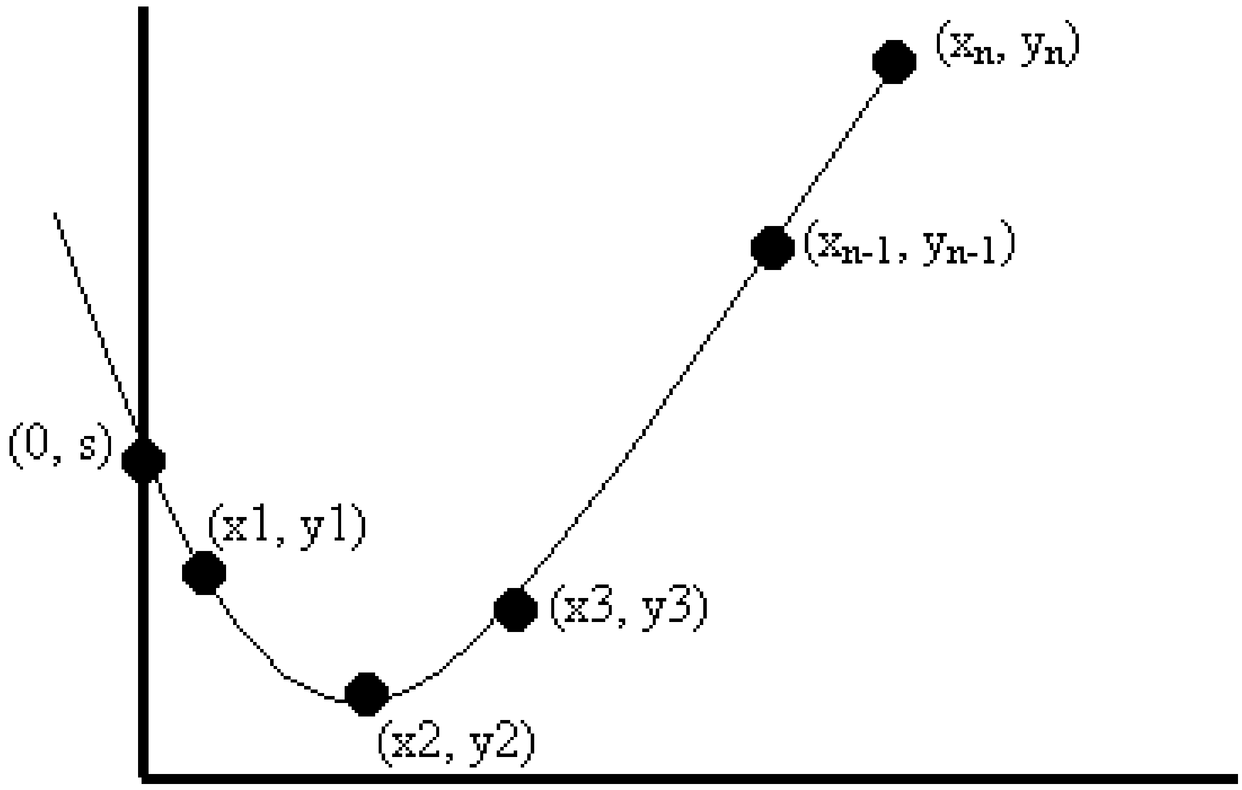
- Использование интерполяции Лагранжа для восстановления секрета

$$P(x) = \sum_{i=1}^{k} y_i \prod_{j=1,\, j\neq i}^{k} \frac{x - x_j}{x_i - x_j}$$

$$P(x) = y_1 \frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + y_2 \frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)}$$

$$+\cdots+ y_k \frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}$$

Пример 1



Пример 2: $(3,5)$-пороговая схема

$$n = 5$$
$$k = 3$$
$$S = 7$$
$$a_0 = S$$
$$a_1 = 3$$
$$a_2 = 5$$
$$p = 11$$

$$\Rightarrow$$

$$q(x) = 5x^2 + 3x + 7 \pmod{11}$$

$$S_1 = q(1) = 5(1)^2 + 3(1) + 7 \pmod{11} \equiv 4$$
$$S_2 = q(2) = 5(2)^2 + 3(2) + 7 \pmod{11} \equiv 0$$
$$S_3 = q(3) = 5(3)^2 + 3(3) + 7 \pmod{11} \equiv 6$$
$$S_4 = q(4) = 5(4)^2 + 3(4) + 7 \pmod{11} \equiv 2$$
$$S_5 = q(5) = 5(5)^2 + 3(5) + 7 \pmod{11} \equiv 4$$

- Каждому из n участников выдается S_i

Пример 2 (продолжение)

- Предположим пользователи с проекциями $S_1 = 4, S_2 = 0$ и $S_5 = 4$ хотят восстановить секрет

$$P(x) = [4\frac{(x-2)(x-5)}{(1-2)(1-5)} + 0\frac{(x-1)(x-5)}{(2-1)(2-5)} + 4\frac{(x-1)(x-2)}{(5-1)(5-2)}] \pmod{11}$$

$$P(x) = [(x-2)(x-5) + 4(x-1)(x-2)] \pmod{11} = 5x^2 + 3x + 7 \pmod{11}$$

$$S = P(0) = 7 \quad \checkmark$$

Достоинства схемы

- Если выбрать $n = 2k - 1$ атакующий будет вынужден захватить $\left\lfloor \frac{n}{2} \right\rfloor = k - 1$ проекций
- Решение эффективно по памяти: размер проекции равен размеру секрета
- Легко создавать и удалять проекции не меняя секрет
- Можно вводить иерархии ключей

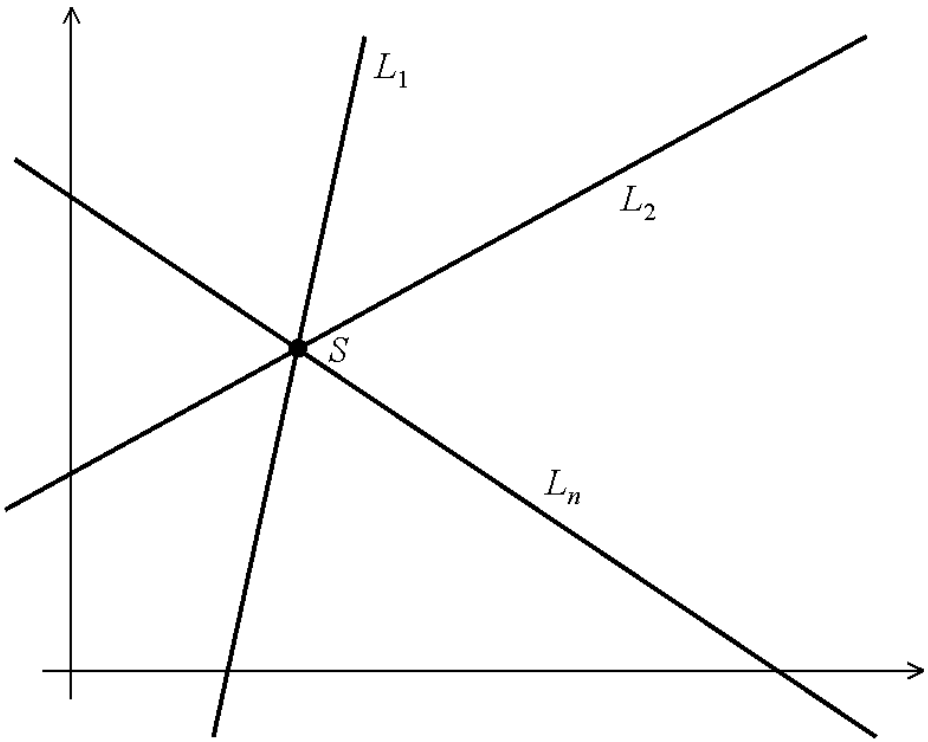
Схема Блекли

- Схема предложена George Blakley в 1979 (первая)
- Основывается на любых k гиперплоскостях размерности k пересекающихся в одной точке

- Проекция – гиперплоскость

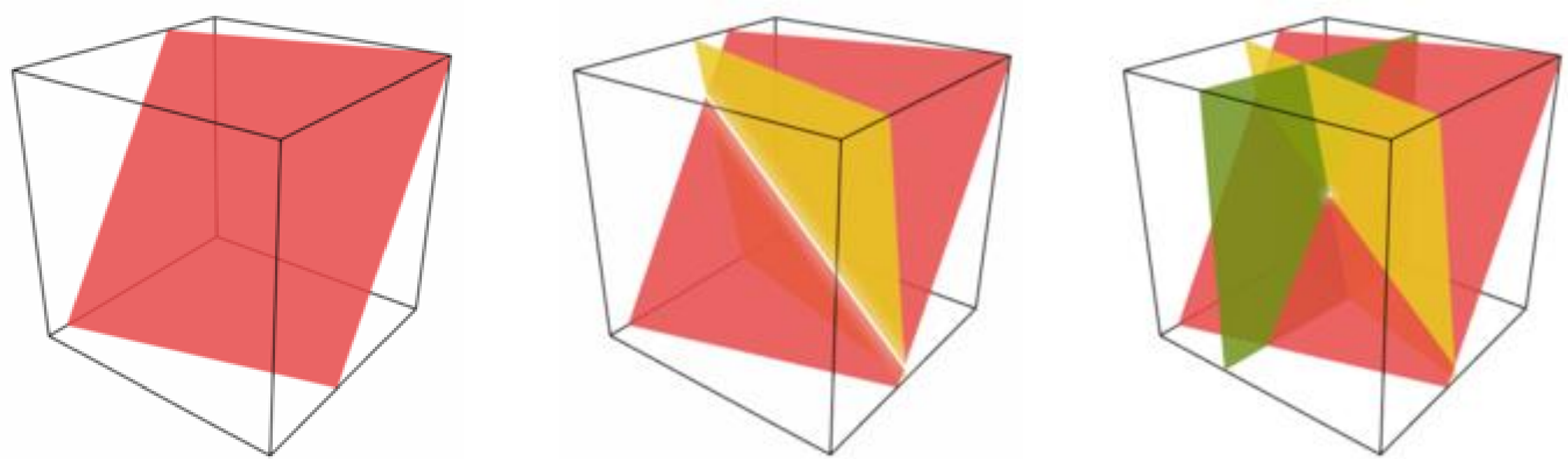
- Секрет – точка пересечения плоскостей

Пример 1: $k = 2$



- Секрет – точка пересечения любой пары прямых

Пример 2: $k = 3$



- Секрет – пересечение 3 плоскостей
- А что нам дают любые 2?

Особенности схемы

- Менее эффективна по памяти: проекция в k раз больше секрета
- Не очень удобна в работе
- Впервые было предложено решение порогового разделения секрета

Схема разделения секрета Asmuth-Bloom

- Основывается на китайской тереме об остатках
- (t, n)-Asmuth-Bloom схема:
  - Выбрать последовательность публичных целых взаимнопростых чисел

    $m_0 < m_1 < \ldots < m_n$ таким образом, чтобы выполнялось неравенство:

$$m_0 \prod_{i=1}^{t-1} m_{n-i+1} < \prod_{i=1}^{t} m_i$$

Asmuth-Bloom's получение проекций секрета

- (t, n) – разделение секрета:
  - Секрет $d \in Z_{m_0}$
  - $y = d + Am_0$

  A – случайное положительное целое, такое что $y < M$

$$\text{Let } M = \prod_{i=1}^{t} m_i.$$

  - Проекции вычисляют по формуле $y_i = y \bmod m_i$ для всех $1 \le i \le n$

Asmuth-Bloom's: восстановление секрета

- Схема восстановление секрета:
  - *y* получается как единственное решение системы уравнений по модулю *M*

$$
\begin{aligned}
y &\equiv y_{i_1} \pmod{m_{i_1}} \\
&\vdots \\
y &\equiv y_{i_t} \pmod{m_{i_t}}.
\end{aligned}
$$

  - Секрет равен $d = y \bmod m_0$

Пороговая криптография

- Асимметричная криптография + пороговое разделение секрета

⇓

- Пороговая криптография:
  - Стандартное шифрование на публичном ключе
  - Разделение секретного ключа на проекции
  - Распределенный алгоритм дешифрования или подписи
    - Нужно не менее k участников
    - Секретный ключ не собирается в процессе вычислений

Дополнительные требования

- Проверка промежуточных вычислений (VSS)
- Обновление проекций без участия дилера
- Выдача новых проекций ключа без участия дилера

Пороговая подпись RSA

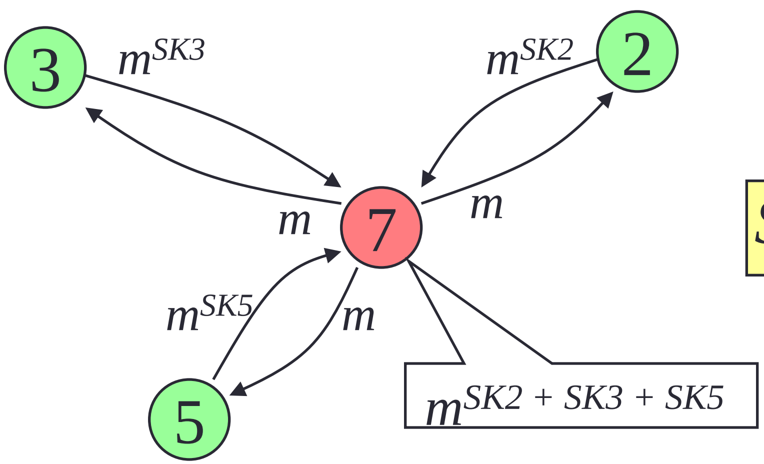
- Генерация ключей
- Разделение секретного ключа
- Генерация частичных подписей
- Сбор подписи

Threshold RSA (TS-RSA)

- J.Kong, et al. [ICNP'01, ISCC'02, WCMC'02]
- Setup
  - Генерация RSA парных ключей: d, e, N
  - Построение случайного полинома *f(x)* степени *t-1*

$$f(x) = \mathbf{d} + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1} \quad \underline{\mathbf{(mod\ N)}}$$

- Генерация подписи

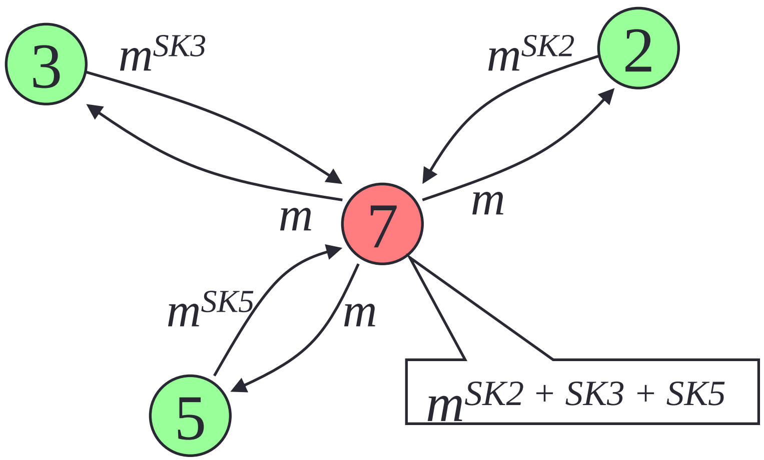


$$SK_i = ss_i l_i(0) \pmod N$$

$$SK_2 + SK_3 + SK_5 \equiv d \pmod N$$

$$m^d \equiv \prod_{i=1}^{t} m^{SK_i} \pmod N$$

TS-RSA: t-ограничение на откручивание



$$SK_2 + SK_3 + SK_5 \equiv d \pmod N$$

$$SK_2 + SK_3 + SK_5 = tN + d$$

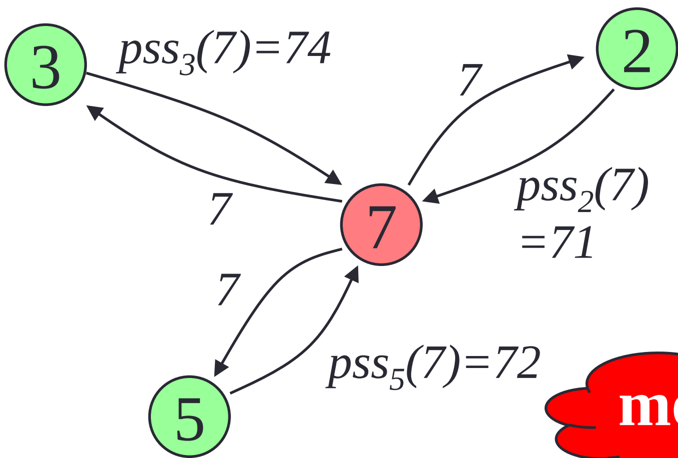
$$m^{SK2+SK3+SK5} = m^{tN+d} = m^{tN} m^{d}$$

```
Y = m^(tN+d);
for (i=0; i <= t;   {
      Y = Y * m^(-N) mod N;
      if (Y^e = m  mod N)
            break;   }
return Y  (= m^d  mod N)
```

22 msec

TS-RSA: VSS отсутствует

- Пример: $f(x) = 77 + 2x + 5x^2 \pmod{119}$, $g=3$
- Свидетели: $w_0=3^{77}=12$, $w_1=3^2=9$, $w_2=3^5=5 \pmod{119}$



$$pss_i(id_j) = ss_i l_i(id_j) \bmod N$$

$$ss_7 = pss_2(7) + pss_3(7) + pss_5(7) = 98$$

$$g^{ss_7} = 3^{98} = 9$$

$$(w_0)(w_1)^7(w_2)^{7^2} = 1$$

mod $\Phi(N)$

$$g^{ss_7} \ \underline{(\text{mod } 119)} \neq \prod_{i=0}^{2} (w_i)^{7^i} \pmod{119}$$

Невозможно проверить корректность ss_i

TS-DSA: Setup

- Схема предполагает самоинициализацию
  - Используется Joint Secret Sharing (JSS), Pedersen [Eurocrypt'91]

$$f(x) = \sum \begin{cases} f_1(x) = f_{10} + f_{11}x + \cdots + f_{1,t-1}x^{t-1} \pmod q \rightarrow User_1 \\ f_2(x) = f_{20} + f_{21}x + \cdots + f_{2,t-1}x^{t-1} \pmod q \rightarrow User_2 \\ \vdots \\ f_n(x) = f_{n0} + f_{n1}x + \cdots + f_{n,t-1}x^{t-1} \pmod q \rightarrow User_n \end{cases}$$

$$S = f(0) = \sum f_i(0) = \sum f_{i0}$$

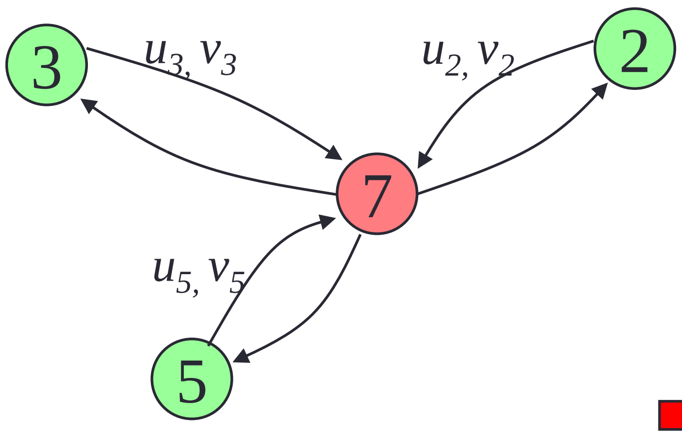
- Каждый пользователь вычисляет $f_i(j)$ (j=1..n, j != i), и отсылает его остальным.
- На своей стороне каждый получатель вычисляет свою проекцию ключа

$$ss_i = f(id_i) = \sum_{j=1}^{n} f_j(id_i)$$

TS-DSA: Генерация подписи

- DSA подпись: (r, s)

O(t²) comm.



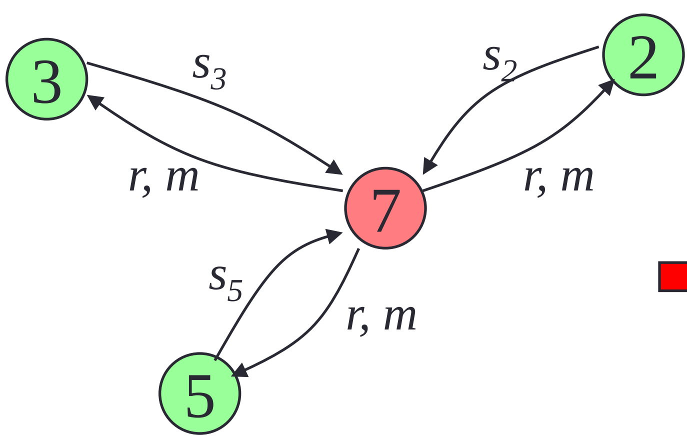
$$u_j = k_j a_j \bmod q \quad v_j = g^{a_j} \bmod p$$

$$U = \sum_{j=1}^{t} u_j l_j(0) \bmod q \quad [= ka \bmod q]$$

$$V = \prod_{j=1}^{t} (v_j)^{l_j(0)} \bmod p \quad [= g^a \bmod p]$$

$$r = (V^{U^{-1}} \bmod p) \bmod q \quad [= g^{k^{-1}} \bmod p \bmod q]$$

(t+1)/2-secure

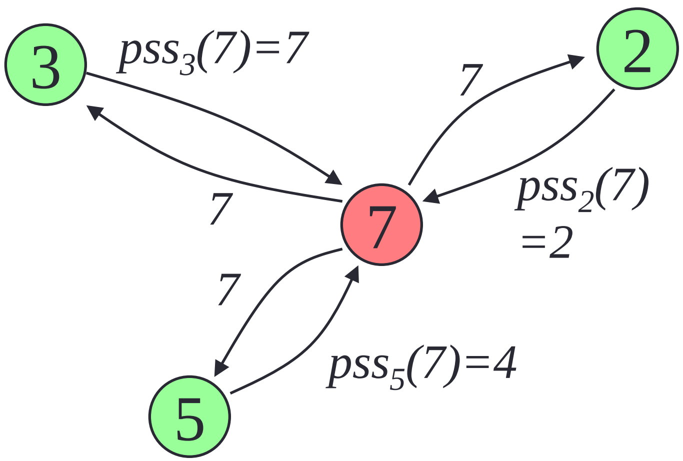


$$s_j = k_j(m + x_j r) \bmod q$$

$$s = \sum_{j=1}^{t} s_j l_j(0) \bmod q$$

TS-DSA: VSS holds

- Пример: $f(x) = 7 + 2x + 5x^2 \pmod{11}$, $g=9$, $q=11$, $p=23$
- Свидетели: $w_0=9^7=4$, $w_1=9^2=12$, $w_2=9^5=8 \pmod{23}$



$$pss_i(j) = ss_i l_i(j) \bmod p$$

$$ss_7 = pss_2(7) + pss_3(7) + pss_5(7) = 2$$

$$g^{ss_7} = 9^2 = 12$$

$$(w_0)(w_1)^7(w_2)^{7^2} = 12$$

$$g^{ss_7 \pmod{11}} = \prod_{i=0}^{2} (w_i)^{7^i} \pmod{23}$$

TS-DSA: Summary

- Достоинства:
  - VSS гарантированы
  - Полностью распределенная процедура генерации ключей
- Ограничения:
  - Стойка только при наличии $\lfloor (t+1)/2 \rfloor$ недобросовестных пользователей
  - Дополнительные $O(t^2)$ сообщения между участниками для генерации секретного ключа *k*