РАЗДЕЛЕНИЕ СЕКРЕТА

Speaker notes

Доклад будет о разделении секрета.

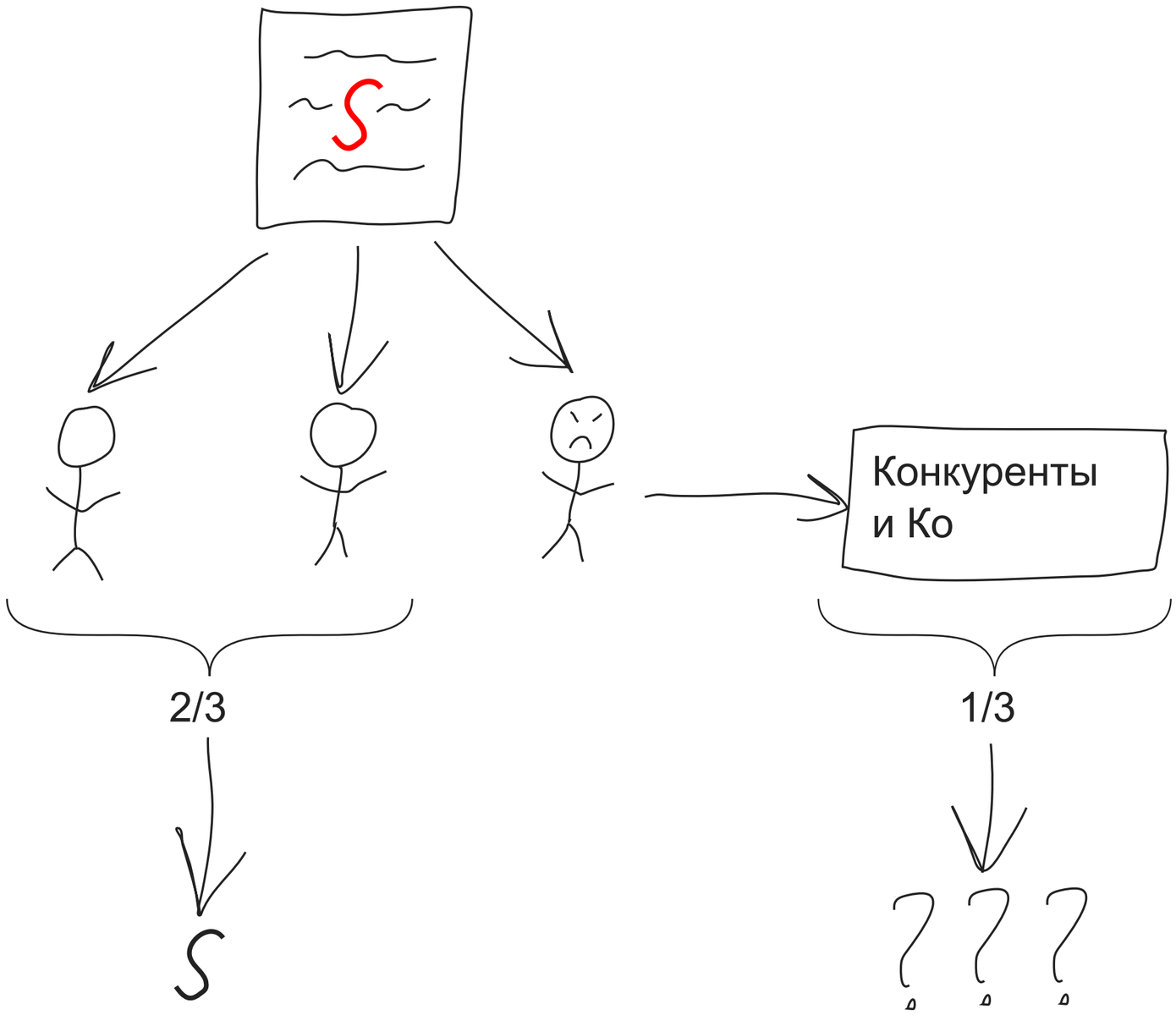
ПЛАН

1. Введение
2. (n, n)-схема
3. Схема Блэкли
4. Многочлен Лагранжа (★)
5. Схема Шамира
6. Атаки (★)
7. Реализация простых структур доступа
8. Связь с теорией матроидов
9. Реализация произвольных структур доступа (★)

Speaker notes

Рассказать нужно много.

ЗАЧЕМ ЭТО НАДО?



Speaker notes

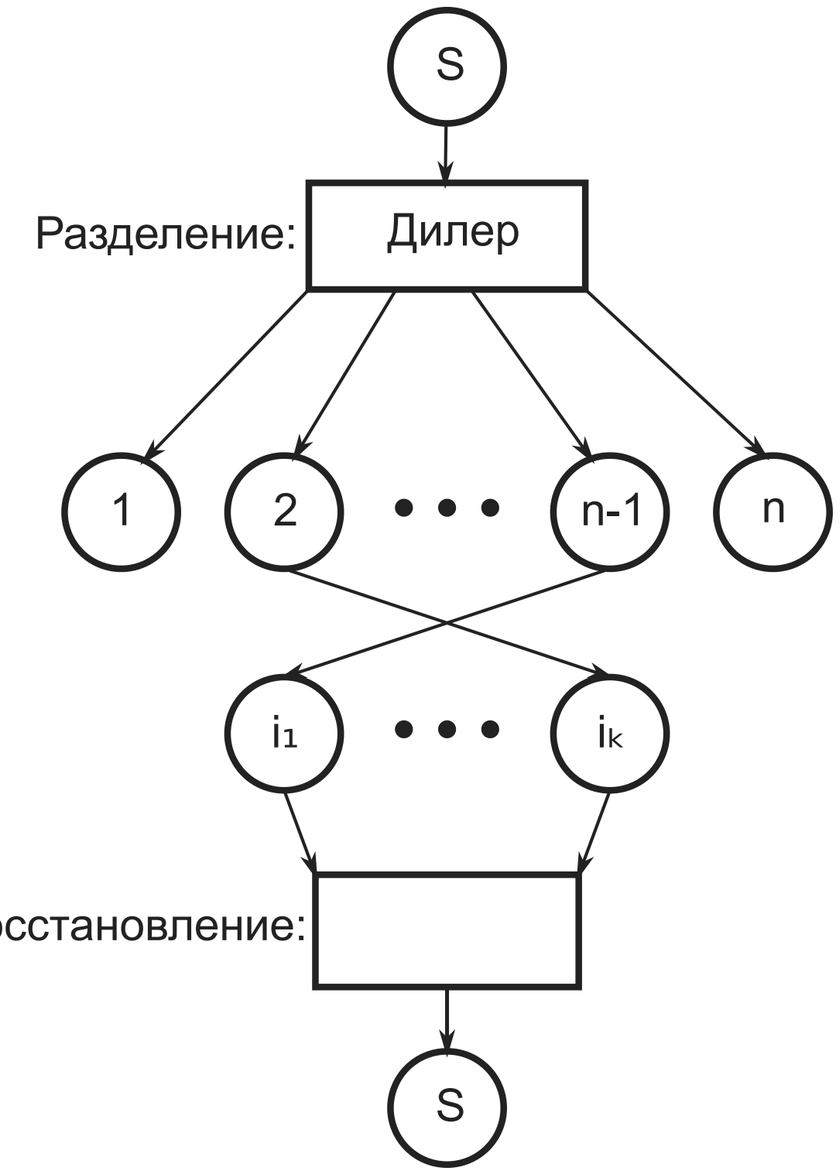
1. Есть важный секрет. *(клик)* Например, приватный ключ. Некоторые компании весь бизнес строят на том, что есть ключ и репутация.
2. Конечно, его ни в коем случае нельзя случайно потерять. Если на главу компании упадёт кирпич, то это не должно быть катастрофой. Поэтому давайте его разделим между разными людьми *(клик)*
3. *(клик)* С другой стороны, конкуренты тоже не против получить ключ и что-то сделать. Что, если один из сотрудников отдаст ключ? Ну или хакеры там украдут?
4. *(клик)* Поэтому хочется сделать так, чтобы мы могли восстановить секрет, а враги — нет.

СХЕМА РАЗДЕЛЕНИЯ СЕКРЕТА

Схема разделения секрета (СРС) разделяет секрет s между n участниками.

1. Дилер берёт секрет и делит его на доли
2. Они раздаются участникам — n штук
3. Затем какое-то подмножество участников собираются вместе
4. И они вместе восстанавливают секрет обратно

Две важные фазы: разделение и восстановление



Speaker notes

Итак, схемы разделения секрета описывают, как нужно разделять и восстанавливать секрет.

В общем, секрет всегда передаётся какому-то дилеру, который этот секрет затем разделяет. Дилером может выступать либо человек, который придумал или уже знает секрет, либо какой-то компьютер, в котором мы уверены. Дилеру мы всегда полностью доверяем.

Так вот, он секрет разделяет на несколько долей, которые затем раздаются участникам. На этом фаза разделения завершается.

Потом происходит восстановление секрета. Для этого некоторое подмножество участников собираются вместе — может и вообще все — которые вместе восстанавливают секрет из своих долей. Конечно же, восстановленный должен совпадать с исходным.

КРАТКО ОБ ОПРЕДЕЛЕНИЯХ

1. Разрешенное множество — те участники, которые могут восстановить секрет.
2. Структура доступа — совокупность всех разрешенных множеств. Должна быть монотонной.
3. Секрет разделяется на **доли**.
4. Схема называется **совершенной**, если недостаточное число долей (не входящие ни в одно разрешенное множество) не дают **никакой** информации о секрете.
5. Схема называется **идеальной**, если каждая доля содержит не больше информации, чем содержится в секрете.

Speaker notes

Давайте сразу кратко пробегусь по определениям. Более подробно будет позднее, когда появятся хорошие примеры.

1. Если группа участников могут восстановить секрет, то эта группа называется разрешенным подмножеством.

2. Структура доступа — совокупность всех разрешенных множеств. Монотонное всегда подразумевается, и означает что добавление участника в группу не может лишать их возможности восстановить секрет. Об этом будем говорить уже ближе к концу доклада.

3. Долями называем то, что получают участники

4. Совершенность. Мы хотим, чтобы, скажем, конкуренты, вообще ничего не знали о секрете по тем долям, которые у них есть. Т.е. если группа участников не образует разрешенное множество, то они совсем ничего не знают о секрете. Нам интересны именно такие схемы.

5. Идеальность. Если каждая доля содержит не больше информации, чем секрет, то схема будет идеальной.

Лучше стараться обращать на эти свойства внимание.

ПОРОГОВАЯ (n, k)-СХЕМА

Это такая схема разделения доступа, что любые k участников из n могут восстановить секрет.

Другими словами, чтобы получить секрет, нужно хотя бы k участников.

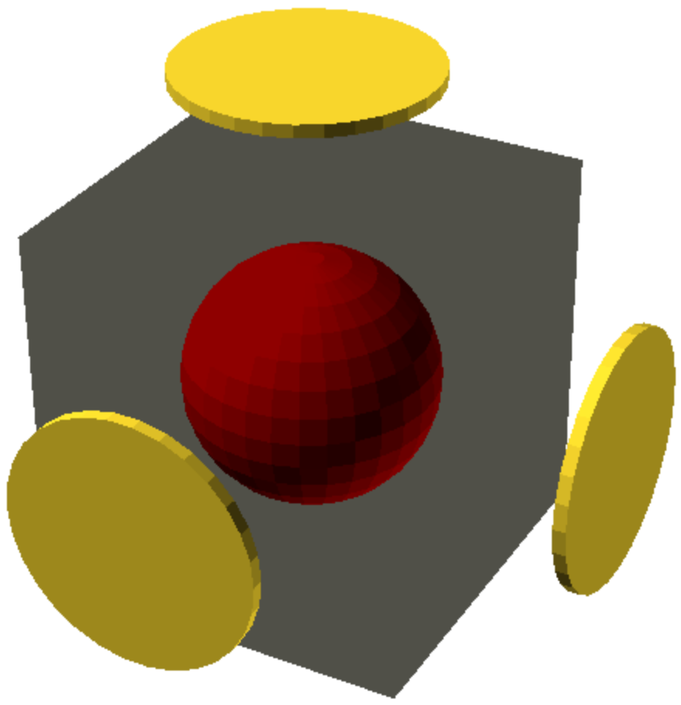
Множество разрешенное, если в нём не меньше k участников.

Speaker notes

Таким схемам посвящена практически половина доклада. Очень простые — если имеется больше участников, то секрет восстановить можно. Более сложные будут сильно позже.

ПРИМЕР

Разделим между **тремя** участниками, чтобы только вместе они могли восстановить



- **Секрет** $\in$ { Шар, Куб, Цилиндр }
- **Доля** $\in$ { Круг, Квадрат }

Любых двух проекций недостаточно, чтобы восстановить секрет

Это $(3, 3)$-схема. $n = k = 3$

Speaker notes

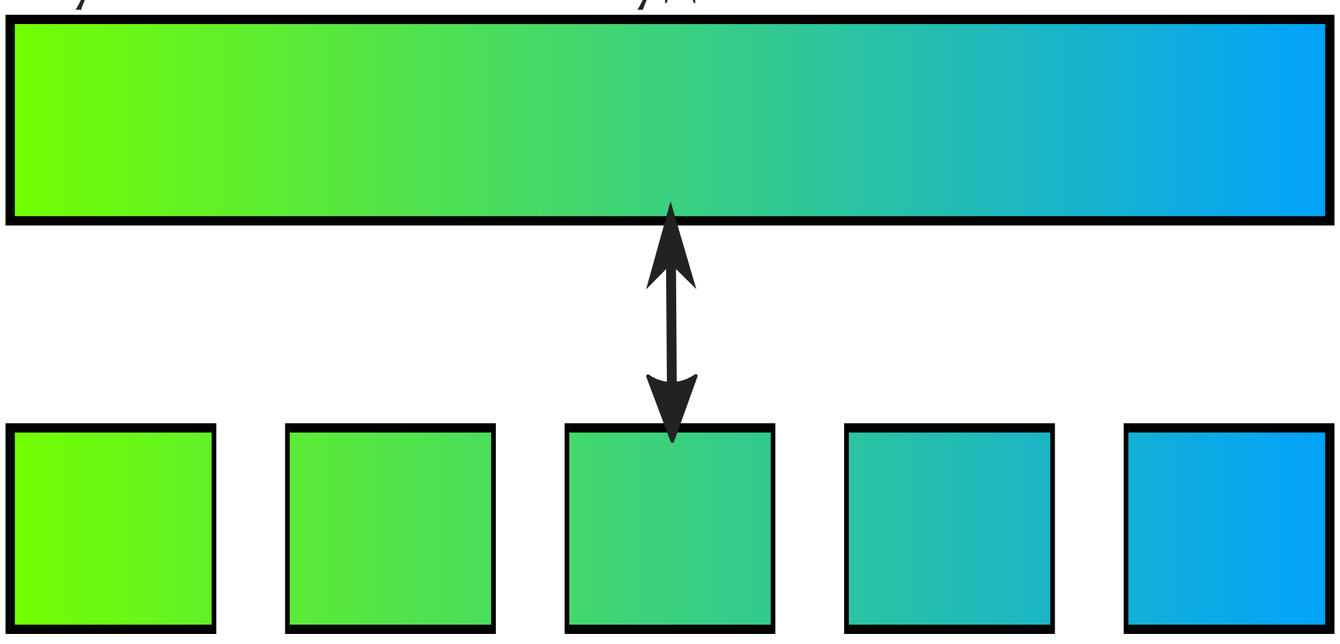
Доклад начинается с простых -схем, где только все участники вместе могут восстановить секрет.

Вот тут очень простой геометрический пример. Мы прячем секрет в коробку и даём трём участникам три проекции секрета на грани коробки. Здесь так получается, что если только два участника попытаются восстановить секрет, то они никогда не смогут сделать это с полной уверенностью. Всегда будет два подходящих варианта. Но когда соберутся трое — они с полной уверенностью смогут восстановить секрет. Разделили ровно так, как и нужно.

(n, n)-СХЕМА

$k = n$. То есть n участников только **все вместе** могут получить секрет.

- Идея 1: нарезать секрет на доли.
- Недостаток: конкуренты легко взломают перебором
- Никогда такую схему использовать не будем



Speaker notes

Начнём с построения простейшей схемы. Это не -схема, а попроще.

Первая идея очень простая. Есть какой-то длинный секрет, мы его можем просто разрезать на доли и раздать участникам. Тогда если они объединятся, то смогут с легкостью восстановить его. Казалось бы, всё хорошо и просто.

Однако здесь есть большой недостаток: конкуренты, если обладают долями — на одну меньше нужного — запросто могут перебрать все значения оставшейся неизвестной. Их не так уж и много, особенно с ростом числа долей.

Мы этого **совсем** не хотим, поэтому давайте строго определим, чего именно мы не хотим.

СОВЕРШЕННОСТЬ

Хотим, чтобы конкуренты ничего не знали о секрете. Даже если знают $k-1$ долю.

По-другому: знание $k-1$ долей ничего не даёт

- s — секрет
- v_i — доля секрета ($i = 0 \dots n$)
- $H(s \mid v_{i_1}, v_{i_2}, \dots, v_{i_m})$ — сколько энтропии в s, если знаем эти доли
- $H(s)$ — сколько энтропии в секрете, если мы совсем ничего про него не знаем

Хотим:

$$H(s \mid v_{i_1}, v_{i_2}, \dots, v_{i_m}) = H(s), \quad \text{при } m < k$$
$$H(s \mid v_{i_1}, v_{i_2}, \dots, v_{i_m}) = 0, \quad \text{при } m \geq k$$

Speaker notes

Итак, “хорошие” схемы, в которых конкуренты совершенно ничего не знают о секрете, называются совершенными.

Так и запишем: *(клик)* знание долей ничего не даёт

Теперь давайте попробуем это записать более строго. Для этого надо ввести пару обозначений *(2 клика)*.

Переходим к сложной часть. *(клик)*.

Здесь упоминается энтропия, причём условная. Это мера неопределённости — как много мы ещё не знаем о значении при таких-то условиях. Грубо говоря, вот мы знаем столько-то долей секрета. Как много надо угадывать, чтобы узнать его весь? Вспоминая старую плохую схему, если мы знаем много долей, то нам остается угадать только один маленький. Очевидно, это гораздо быстрее чем угадать весь секрет.

(клик). Следующее — количество энтропии в секрете, когда мы вообще ничего не знаем. Грубо, как сложно угадать весь секрет целиком.

Вообще, запросто могу переписать при помощи условных вероятностей, если кому-то не очень понятно.

(клик)

И теперь можем записать что именно мы хотим.

- Если мы знаем меньше долей, то мы знаем о секрете ровно столько же, как если бы не знали совсем ничего
- А если мы знаем достаточно долей, то знаем и весь секрет полностью. Никаких сомнений в его значении нету.

НАСТОЯЩАЯ (n, n)-СХЕМА

Идея: $s = v_1 + v_2 + \ldots + v_n$

Алгоритм:
0. Пусть $s \in \mathbb{F}$
1. Генерируем совершенно случайные $v_2, \ldots, v_n$ тоже из $\mathbb{F}$.
2. Находим $v_1 = s - v_2 - \ldots - v_n$
3. Раздаём эти доли участникам

Почему она совершенная?

Допустим: знаем $v_1, \ldots, v_{n-1}$ — все кроме (без потери общности) последнего.

Тогда:
1. Есть $|\mathbb{F}|$ вариантов для последней доли.
2. Каждый вариант даёт свой уникальный s.
3. Угадать значение доли из $|\mathbb{F}|$ также сложно, как угадать секрет (тоже из $\mathbb{F}$).

Speaker notes

Теперь реализуем подходящую схему. Хотим секрет разделить между участниками. Идея очень проста: пусть все доли в сумме дают этот самый секрет.

(клик) Во-первых, договоримся что секрет взялся из какого-то поля . Совершенно не важно из какого.

(клик) Для реализации достаточно сгенерировать все числа кроме одного. Они должны быть из того же поля.

(клик) Затем остаётся единственным образом определить одно оставшееся.

(клик) И всё, доли готовы.

(клик) Следующий шаг: покажем что она действительно совершенна.

(клик) Допустим самое худшее — знаем все доли кроме одного.

(клик) Тогда мы можем попытаться найти последний подбором.

(клик) Но их ровно столько, сколько и возможных секретов.

(клик) Получается, что ровно с тем же успехом мы могли бы пытаться угадать секрет. Видно, что знания долей никак не помогают. А значит схема совершенна.

Теперь и секрет, и доли из одного поля, т.е. они содержат одинаково информации (если они выбраны случайно). Это интересно.

ТЕОРЕМА

Пусть схема совершенная, а $v_i \in V$ — доля секрета $s \in S$.
Тогда $H(v_i) \geq H(s)$.

Следствие: Если секрет и доли выбираются случайно, то $|V| \geq |S|$.

Доказательство следствия

1. Известно, что знание $k-1$ доли не даёт никакой информации о секрете
2. Также известно, что зная k долей можно единственным образом восстановить секрет.
3. Если $|V| < |S|$, то зная $k-1$ долей, мы можем получить лишь $|V|$ значений секрета, по одному на каждое возможное значение доли.
4. Но ведь секрет — случайно выбран из $|S|$ возможных вариантов! Пришли к противоречию: такая схема не совершенна.

Speaker notes

Заметили, что размер доли вырос? В плохой и неправильной схеме он был в раз меньше секрета, а теперь совпадает с ним?

Так вот, существует подходящая теорема *(клик)*. Она говорит, что у совершенной схемы каждая доля не может содержать меньше информации, чем целый секрет.

У неё есть довольно интересное следствие *(клик)*. А именно, мощность множества с долью обычно не может быть меньше, чем множество с секретом. На практике это означает, что если у вас секрет занимает столько-то бит, то и каждая доля будет не меньше.

Теперь давайте это докажем. Будем доказывать только следствие, потому что доказательство самой теоремы слишком сложно для этого доклада, да и не нужно. Будем считать, что числа у нас всегда случайные, это обычно правда.

Если в двух словах, то когда мы знаем долей, то, чтобы восстановить секрет, достаточно перебрать все значения оставшейся доли. И если этих значений меньше, чем значений секрета, то это противоречит тому, что знание долей ничего не даёт.

РАЗДЕЛЕНИЕ ДЛИННЫХ СЕКРЕТОВ

Есть строка: `I like secret sharing systems`. Она длинная (29 символов).

Можем её рассмотреть, как набор байтов

```
49 20 6c 69 6b 65 20 73 65 63 72 65 74 20 73 68 61 72 69 6e 67 20 73 79 73 74 65 6d 73
```

И каждый байт разделять отдельно, используя группу $\mathbb{Z}_{256}$:

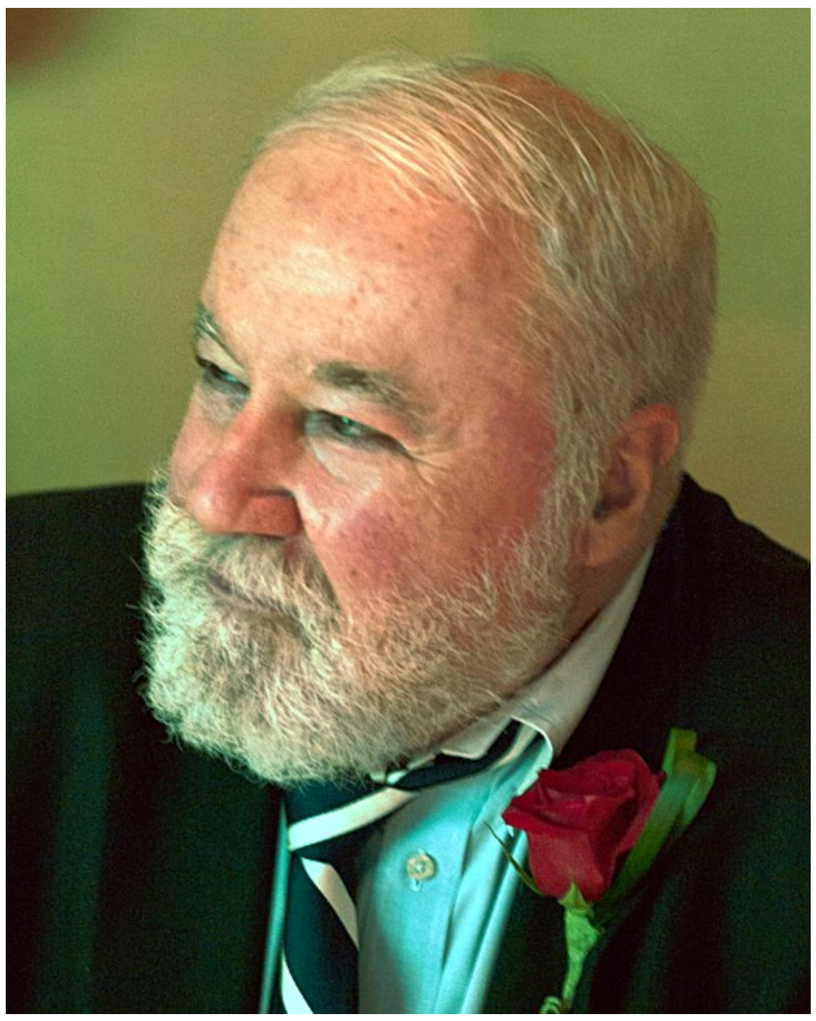
- $49_{16} = 73 = 10 + 63$
- $20_{16} = 32 = 50 + 238$
- $6c_{16} = 108 = \dots$

Такое разделение всё равно останется совершенным.

СХЕМА БЛЭКЛИ

Это (n, k)-схема. Из n участников достаточно только k.

Описана Джорджем Блэкли в начале июня 1979 года.

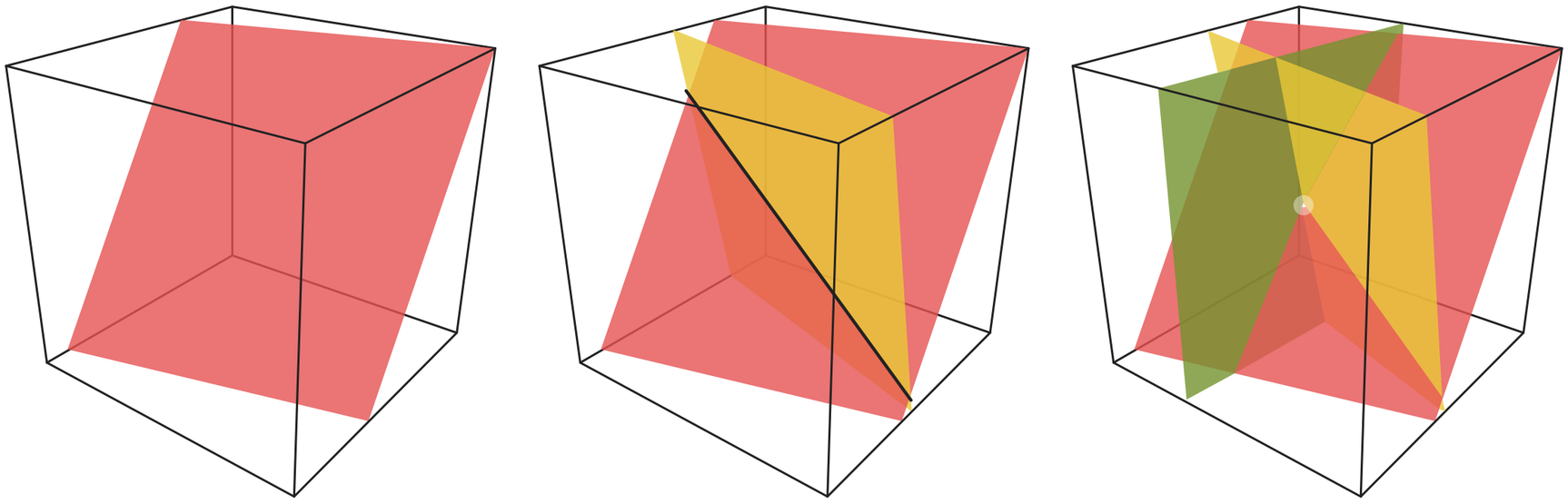


Speaker notes

Мы реализовали простейшую -схему, но она не даёт права потерять никакую долю. Обязательно все должны присутствовать. На самом деле, с её помощью можно делать удивительные вещи, но это не очень эффективно и будет в самом конце доклада. А пока что поговорим про схему Блэкли.

СХЕМА БЛЭКЛИ

Идея: k гиперплоскостей пересекаются в k-мерном пространстве в точке.



Алгоритм:

0. Секрет $s \in \mathbb{F}$. Возьмём пространство $\mathbb{F}^k$
1. Выберем случайные и независимые $x_2, x_3, \ldots, x_k$
2. Секретная точка: $(s, x_2, x_3, \ldots, x_k)$
3. Провести через неё n случайных гиперплоскостей и раздать их участникам.

Speaker notes

Идея этой схемы заключается в том, что гиперплоскости пересекаются только в одной точке. На примере трёхмерного пространства как раз показано, что две плоскости пересекаются по прямой, а три — уже в точке.

(клик) Теперь пройдёмся по алгоритму. Как и раньше, возьмём секрет из поля . *(клик)* К нему сразу добавляем -мерное пространство, в котором и будет всё происходить.

(клик) Дальше снова генерируем случайные точку. Удивительно, но это продолжает совпадать с простейшей схемой.

(клик) И вот теперь начались отличия. Как мы помним, плоскости пересекаются в точке. Значит какая-то точка пространства должна кодировать секрет. Ну и вот мы её выбрали. Одна её координата содержит секрет, а остальные выбраны совершенно случайно.

(клик) Дальше мы должны провести через эту точку произвольных плоскостей. Вот и всё.

ПРИМЕР

- Секрет из поля $\mathbb{Z}_{23}$ будет $s = 18$
- Действуем в пространстве $\mathbb{Z}_{23}^3$ — три участника могут восстановить
- Выберем случайную точку $(\mathbf{18}, 7, 20)$ — секрет в первой координате
- Проведём плоскости через эту точку:
  - $21x_1 + x_2 + 15x_3 = 18$
  - $11x_1 + 9x_2 + x_3 = 5$
  - $3x_1 + 6x_2 + 8x_3 = 3$

  Именно этой информацией обладают участники — каждый знает ровно одну плоскость.
- Пересечём их:

$$\begin{pmatrix} 21 & 1 & 15 \\ 11 & 9 & 1 \\ 3 & 6 & 8 \end{pmatrix} \vec{x} = \begin{pmatrix} 18 \\ 5 \\ 3 \end{pmatrix}$$

- Решение СЛАУ: $\vec{x} = \begin{pmatrix} \mathbf{18} \\ 7 \\ 20 \end{pmatrix}$; отсюда секрет $s = 18$

Speaker notes

Берём какой-то секрет — здесь он из поля — и хотим его разделить. Восстановить могут любые три участника, поэтому пространство трёхмерное.

Для этого помещаем его в какую-то случайную точки и проводим через неё три случайных плоскости — по одной на каждого участника. Эти плоскости и будем раздавать участникам.

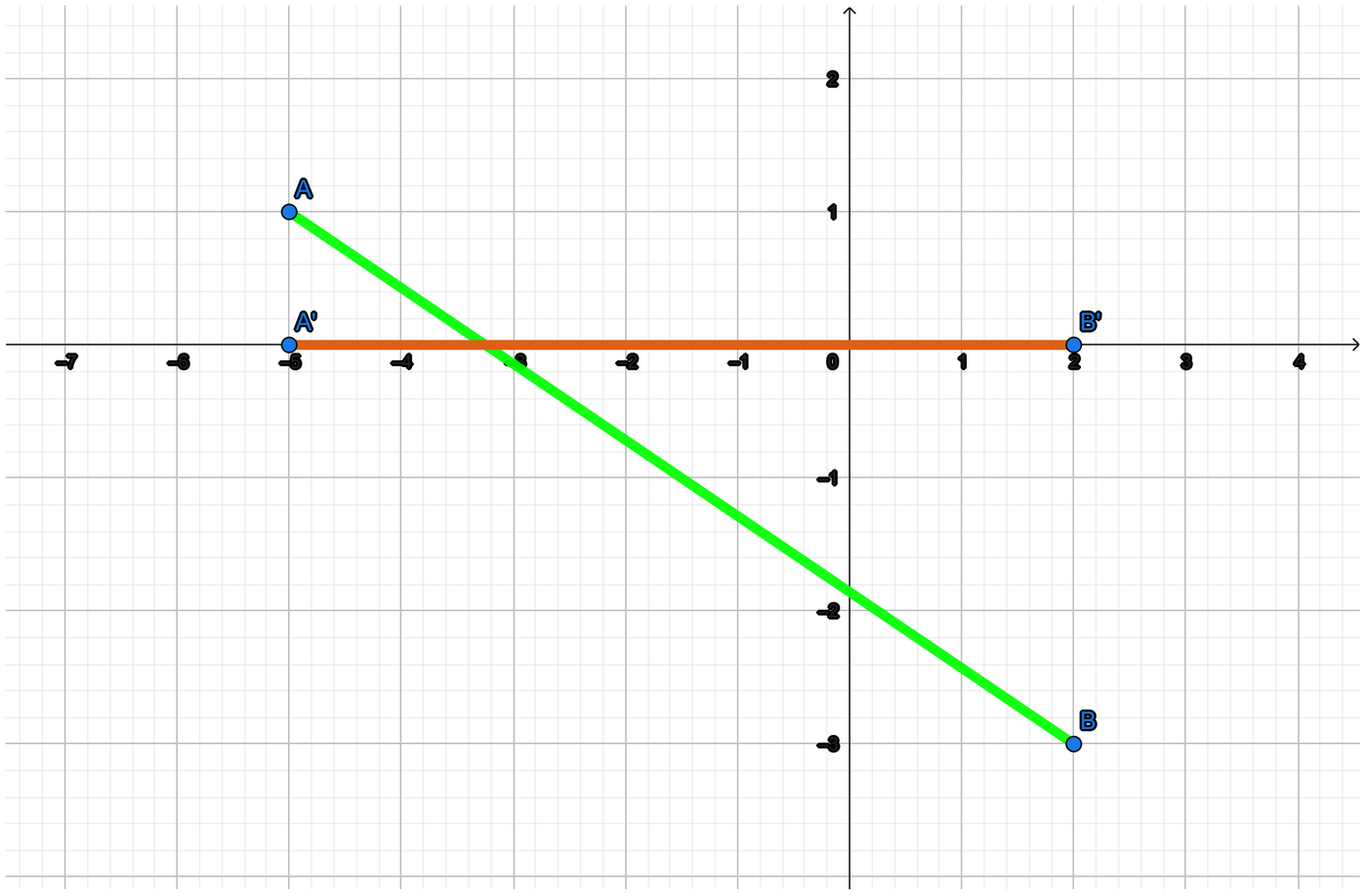
На этом с разделением всё, теперь давайте восстановим обратно *(клик)*.

У нас есть ьтри плоскости, надо найти их общую точку пересечения. Для этого достаточно будет составить вот такую СЛАУ на точку и решить её любым удобным способом. Поскольку мы действуем в поле, то какое-нибудь решение найдется.

И вот решение – как раз та точка, которые мы сгенерировали раньше. Берём первую её координату и получаем секрет обратно.

СХЕМА БЛЭКЛИ: СОВЕРШЕННОСТЬ

- Дана $k - 1$ плоскость, пересекаются по прямой ($\cong \mathbb{F}$)
- Но нас интересует только одна координата



- На прямой столько же точек, сколько и значений секрета!

Speaker notes

Давайте покажем, почему эта схема совершенна. То есть что даже зная плоскость, угадывать секрет не становится легче.

С одной стороны, вполне известно, что секретная точка лежит на прямой. Казалось бы, это облегчает задачу. И это было бы так, если бы были важны все координаты точки.

Поскольку сам секрет находится в первой координате точки, то можно рассмотреть только эту координату у прямой.

И тогда будет легко видеть, что прямая содержит столько же точек, сколько и ось координат. То есть столько же точек, сколько и значений секрета.

СХЕМА БЛЭКЛИ

Секрет: $s \in \mathbb{F}$.

Доля: k-мерная плоскость

$$A_1 x_1 + A_2 x_2 + \ldots + A_k x_k + A_0 = 0$$

Плоскость: набор $(A_1, A_2, \ldots, A_k) \in \mathbb{F}^{k+1}$

Но секрет-то только из $\mathbb{F}$! Доля в $k+1$ раз больше.

Speaker notes

Теперь посмотрим что же у нас получилось. А получились плоскости. Каждую из них можно записать таким образом: *(клик)*. Таким образом, каждая плоскость записывается при помощи ровно коэффициента, каждый из нашего поля: *(клик)*.

В этой схеме так получилось, что размер доли в раз больше, чем размер самого секрета.

ИДЕАЛЬНОСТЬ

Секрет $s \in S$, доля $v_i \in V$.

- Из теоремы: $H(v_i) \geq H(s)$
- Если $H(v_i) = H(s)$, то схема идеальна

Схема идеальна тогда, когда доля содержит ровно столько же информации, сколько и секрет.

Speaker notes

Теперь давайте определим новое свойство.

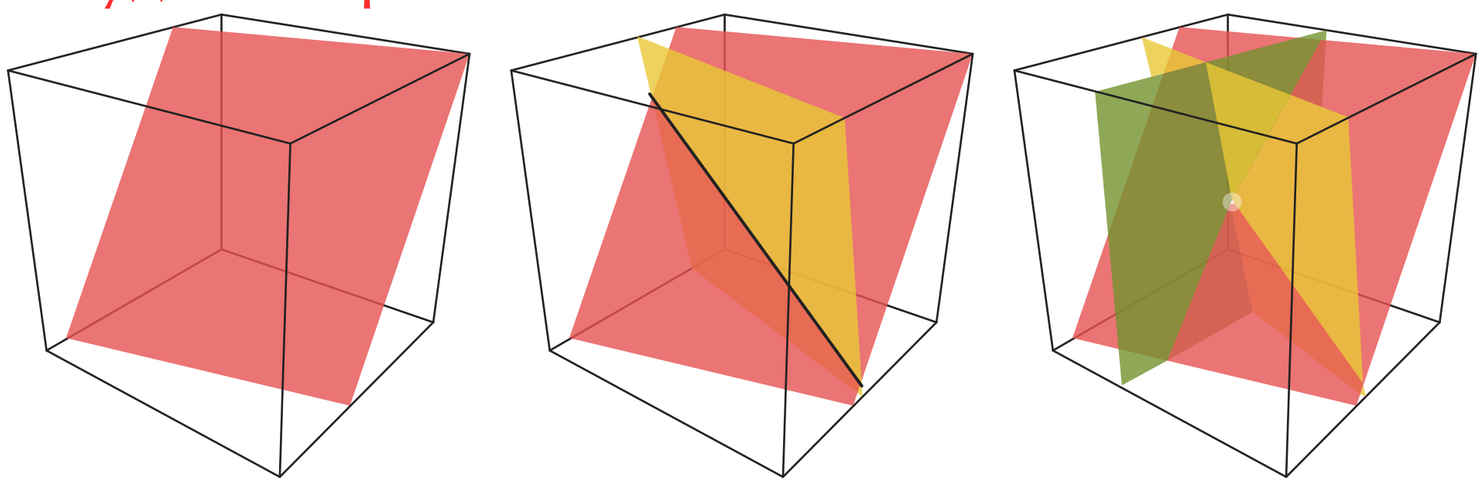
Идеальность — когда размер доли ровно такой же, как и размер секрета. Если схема совершенна, то меньше он быть и не может, мы это доказали раньше.

Поскольку в схеме Блэкли размер доли гораздо больше, то она не идеальна. Совершенна, но не идеальна.

НЕПРАВИЛЬНАЯ СХЕМА

Хотим сделать схему Блэкли её идеальной.

- Секрет хранится только в одной координате.
  - $s \in \mathbb{F}$
- А доля гораздо больше
  - $v_i \in \mathbb{F}^k$
- Идея: распределить секрет по всем координатам: $s \in \mathbb{F}^k$
  - Нельзя: на прямой всего лишь $|\mathbb{F}|$ точек, хотя вариантов секрета $|\mathbb{F}|^k$
  - Такая схема не будет совершенной!



Speaker notes

Теперь давайте рассмотрим вариацию схемы, которая является идеальной.

(клик) Во-первых, секрет хранится только в одной координате. *(клик)* Но вот размер доли гораздо больше, и нам это не нравится.

(клик) Что будет, если распределить секрет по всем координатам точки? Ведь тогда как раз размеры сравняются. Как думаете, так можно сделать?

(клик) Но тогда плоскость пересекается по прямой. И на этой прямой уже гораздо меньше точек. А значит секрет становится гораздо проще найти, даже если не знать все доли. *(клик)* Такая схема будет несовершенной.

Именно такой вариант схемы Блэкли почему-то описан на русской википедии и какой-то криптовики. Эта схема аналогична самой первой, где мы просто нарезали секрет на доли. Конечно, это нам не подходит.

Ни в коем случае так не делайте. Я это показываю во многом из-за того, что почему-то на некоторых русскоязычных сайтах приводится именно такая реализация схемы. Она неправильная.

МНОГОЧЛЕН ЛАГРАНЖА

Хотим: провести многочлен не более $n-1$ степени через n точек

Например, **прямая** — многочлен первой степени — легко строится по **двум** точкам.

Дано: точки $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Очень простая идея:

Можно составить СЛАУ на c_i, подставив точки в $c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1}$.

Здесь n неизвестных, n уравнений, а следовательно решение единственно.

Но есть явная формула: $f(x) = \sum_{j=1}^{n} y_j \prod_{\substack{i=0 \\ i \neq j}} \frac{x - x_i}{x_j - x_i}$ — многочлен Лагранжа

Speaker notes

Теперь немного поговорим про многочлен Лагранжа. Он нам потом пригодится. *(клик)*

Вообще, довольно известный факт, что через точек можно построить многочлен степени.

(клик) Для этого достаточно составить СЛАУ и найти этот самый многочлен.

(клик) Но это не так эффективно — иногда интересно узнать значение только в одной точке — так что давайте всё-таки рассмотрим многочлен Лагранжа. Он тоже очень простой.

ОЧЕВИДНОЕ РЕШЕНИЕ

Просто скажем следующее:

$$f(x) = \begin{cases} y_1, & x = x_1 \\ y_2, & x = x_2 \\ \dots & \\ y_n, & x = x_n \\ \text{что-нибудь}, & x \notin \{x_1, x_2, \dots, x_n\} \end{cases}$$

И если бы у нас была такая функция:

$$\ell_j(x) = \begin{cases} 1, & x = x_i, i = j \\ 0, & x = x_i, i \neq j \\ \text{что-нибудь} & x \notin \{x_1, x_2, \dots, x_n\} \end{cases}$$

Тогда бы мы разбили на сумму:

$$f(x) = y_1\ell_1(x) + y_2\ell_2(x) + \ldots + y_n\ell_n(x)$$

Speaker notes

Итак, во-первых, мы хотим как-то построить вот такую функцию. Важно лишь то, чтобы она проходила через некоторые точки, а всё остальное — как получится.

(клик) Но строить такую большую функцию довольно сложно, поэтому давайте упростим себе жизнь и сделаем функцию , она поменьше. В одной из нужных точек она равна единице, в других нужных — равна нулю, а во всех остальных нас значение не волнует.

(клик) И если мы вдруг сможем построить такую функцию , то первую большую можно будет легко разбить на вот такую сумму. Здесь довольно очевидно, что она будет проходить через все нужные точки. Для каждой интересующей нас точки только одна из будет равна единице, а все остальные равны нулю. И эту единственную единицу мы умножаем на требуемое значение функции в нужной нам точке. Вот и всё.

ТАКАЯ ФУНКЦИЯ ЕСТЬ!

Напомню: $\ell_j(x) = \begin{cases} 1, & x = x_j \\ 0, & x = x_i, i \neq j \\ \text{что-нибудь}, & x \notin \{x_1, x_2, \ldots, x_n\} \end{cases}$

Выглядит она как произведение дробей:

$$\ell_j(x) = \prod_{\substack{i=0 \\ i \neq j}} \frac{x - x_i}{x_j - x_i}$$

- Если $x = x_j$, то каждая дробь равна 1, и вся функция тоже.
- Если $x = x_i, i \neq j$, то найдётся i, такой что $x_i = x$. А значит один из числителей будет таким: $x_i - x_i = 0$. И всё произведение обнулится.
- Все точки различны и $j \neq i$, а значит мы не делим на ноль.

Speaker notes

К счастью, такая маленькая функция существует, и сейчас покажу как она работает.

(клик) Выглядит она как вот такое сложное произведение дробей. Здесь из произведения пробегает по всем, кроме .

(клик) Смотрим на первое требование. Функция при должна быть равна единице. Здесь это вполне выполняется. Каждая дробь будет иметь один и тот же числитель и знаменатель, поэтому все они равны единице.

(клик) Двигаемся дальше. Надо чтобы во всех остальных точках функция была равна нулю. Здесь это обеспечивается за счёт того, что мы перебираем все эти остальные точки и вычитаем из аргумента. Тогда в этих точках хотя бы один числитель обнулится, а вместе с ним обнулится и всё произведение.

(клик) Ну и наконец, функция везде определена. Поскольку предполагается, что все иксы различны, то мы на ноль никогда не делим.

ИТОГ

Интерполяционный многочлен Лагранжа:

$$f(x) = \sum_{j=1}^{n} y_j \prod_{\substack{i=0 \\ i \neq j}} \frac{x - x_i}{x_j - x_i}$$

- Проходит через точки $(x_1, y_1), \ldots, (x_n, y_n)$
- Степень не больше n

Speaker notes

Ну и вот итог. Интерполяционный многочлен Лагранжа выглядит вот таким вот образом. Как мы показали, он проходит через все нужные точки (если они различны).

Поскольку в каждом произведении не больше линейного множителя, то и весь многочлен тоже не больше степени.

Остается только разобраться с единственностью.

ЕДИНСТВЕННОСТЬ

> *Через n точек проходит единственный многочлен степени не больше $n-1$.*

Доказательство:

- Пусть $f(x)$ и $q(x)$ — два многочлена, оба проходят через одинаковые n точек
- Тогда $f(x) - q(x)$ имеет не меньше n нулей — в точках через которые они проходят.
- А ещё $f(x) - q(x)$ тоже степени не больше $n-1$.
- Но $f(x) - q(x)$ не может иметь n нулей! Это же больше его степени.
- А значит $f(x) - q(x) = 0$ ∎

Speaker notes

Доказательство очень простое.

Действуем от противного. Пусть есть два многочлена, которые оба проходят через одни и те же точек.

Тогда рассмотрим их разность. Она будет тоже многочленом степени не больше , как и исходные два.

Но по основной теореме алгебры такой многочлен не может иметь более чем нуль. А здесь у нас их , поскольку и точно совпадают в точках.

Получаем что равно нулю, то есть , что и требовалось доказать.

СХЕМА ШАМИРА

Опубликована в ноябре 1979 года, всего через полгода после схемы Блэкли.

Идея: через k точек можно провести единственный многочлен. Пусть $s = f(0)$.

Доли — точки на многочлене. Секрет — его значение в нуле.

Speaker notes

Итак, переходим к практически полезной схеме. Схема Шамира была опубликована в ноябре 1979 года криптографом Ади Шамиром, через примерно полгода после схемы Блэкли. Та была представлена в начале июня. Шамир тот же самый, который участвовал в создании RSA, и в схеме нулевого разглашения.

Идея схемы Шамира очень проста. По долей мы всегда можем построить единственный многочлен степени . И секрет тогда будет равен его значению в нуле. А вот по долей уже нельзя ничего говорить о том, какой может быть многочлен и где находится секрет.

РЕАЛИЗАЦИЯ

1. Выбрать достаточно большое поле (секрет должен поместится).
2. Сгенерировать случайный многочлен степени $k - 1$
$$f(x) = c_1 x^{k-1} + c_2 x^{k-2} + \ldots + c_{k-2} x^2 + c_{k-1} x + s$$
Свободный член равен s, благодаря чему $f(0) = s$
3. Посчитать значения в точках $1, 2, 3, \ldots, n$ и раздать их. То есть $v_i = f(i)$.
Координата x точек — публичная информация.
4. По k любых из этих n точек можно построить f обратно. Тогда $s = f(0)$.

Speaker notes

Подробная схема такова.

1. Сначала надо выбрать какое-то поле. Чаще всего используется поле вычетов по большому простому модулю. Но вообще-то говоря, сгодится любое. Просто чем меньше поле, тем меньше значений для секрета и тем легче взломать перебором. Хотя обычно на размер секрета влиять нельзя. Также в маленьком поле будет мало точек, которые можно раздать. Например, в никак нельзя выбрать 10 разных точек на многочлене.

2. Затем генерируем случайные коэффициенты для многочлена. Здесь они обозначены как . Свободный же член делаем равным , за счёт чего можно очень легко сделать так, чтобы .

   Вообще-то говоря, не обязательно использовать именно для секрета, просто иначе генерация многочлена становится немного менее тривиальной. Можно, например, сгенерировать точку и интерполировать многочлен через них и точку с секретом. Но мы так делать не будем, у нас всегда секрет в нуле.

3. Дальше надо выбрать любые точек на многочлене и посчитать их. Обычно выбирают просто по порядку и т.д., но вообще-то говоря это совершенно не важно, ведь здесь эти координаты — публичная информация.

4. Восстановление секрета происходит очень просто. Нужно всего лишь выбрать любые точек и построить многочлен обратно. Ну а дальше секрет легко находится. Вообще-то говоря, лучше не искать многочлен отдельно, а просто подставить d многочлен Лагранжа и посчитать значение в этой точке.

ПРИМЕР

- Из поля $\mathbb{Z}_{13}$ выбрали секрет $s = 4$.
- Три участника могут восстановить секрет. Используем многочлен второй степени.
- Сгенерируем такой: $f(x) = 5x^2 + 11x + 4$; здесь $s = f(0)$
- Посчитаем точки (над полем!):

  1. $f(1) = 7$
  2. $f(2) = 7$
  3. $f(3) = 4$

  Именно этой информацией обладают участники.

- Восстановим многочлен (вычисления по модулю 13):

$$f(x) = y_1 \frac{x - x_2}{x_1 - x_2} \frac{x - x_3}{x_1 - x_3} + y_2 \frac{x - x_1}{x_2 - x_1} \frac{x - x_3}{x_2 - x_3} + y_3 \frac{x - x_1}{x_3 - x_1} \frac{x - x_2}{x_3 - x_2} =$$

$$= 7 \frac{x - 2}{1 - 2} \frac{x - 3}{1 - 3} + 7 \frac{x - 1}{2 - 1} \frac{x - 3}{2 - 3} + 4 \frac{x - 1}{3 - 1} \frac{x - 2}{3 - 2} =$$

$$= \frac{7}{2}(x - 2)(x - 3) + \frac{7}{-1}(x - 1)(x - 3) + \frac{4}{2}(x - 1)(x - 2) =$$

$$= 10\left(x^2 - 5x + 6\right) - 7\left(x^2 - 4x + 3\right) + 2\left(x^2 - 3x + 2\right) =$$

$$= 5x^2 - 28x + 43 \equiv 5x^2 + 11x + 4$$

- И тогда секрет: $f(0) = 4$

Speaker notes

Итак, вот у нас есть поле и из него выбрали секрет . Будем его делить на трёх участников.

Для этого придумываем случайный многочлен, который в нуле равен секрету. Достаточно правильно подобрать свободный член.

И теперь считаем его значения в трёх разных точках. Интересно, что пара совпали. Само значение многочлена секретная информация. Каждое знает только один участник. Но вот список точек, в которых считали значения, уже публичный.

Теперь приступим к восстановлению секрета обратно *(клик)*.

В целом, ничего интересного. Беру формулу Лагранжа для трёх точек и начинаю аккуратно всё это дело считать. Здесь я иду долгим путём и вычисляю весь многочлен. Гораздо лучше и быстрее было бы просто подставить и не считать те ужасные скобки.

Но в конце-концов получаем ровно тот же самый многочлен, и запросто восстанавливаем секрет.

Заметка:

СОВЕРШЕННОСТЬ

> *For example, a polynomial of degree b can be reconstructed from its values at $b+1$ points. But already its values at any b points tell a lot about it. It can also be reconstructed from the values of its 0th through bth Taylor coefficients at a point. But already the values of any b of these $b+1$ numbers tell a lot about it.*

— Блэкли о многочленах

Speaker notes

Перед тем как мы двинемся дальше, вот тут вот цитата из работы Блэкли по его схеме.

Вот он говорит, что многочлен степени может быть восстановлен по , но даже только уже дают очень много информации о многочлене. Как вы думаете, он прав?

Нет, в общем случае мы не получаем никакой дополнительной информации о многочлене.

СОВЕРШЕННОСТЬ

Мы знаем $k-1$ долей. Хотим узнать какой может быть секрет.

- Не используя знания долей, можем перебрать все секреты: их $|\mathbb{F}|$ штук.
- Перебор вариантов последнего долей не легче: тоже $|\mathbb{F}|$ вариантов.
- Правда ли, что все секреты возможны?
- Можем ли построить многочлен степени $k-1$, проходящий через всякий секрет и через известные $k-1$ точек?
- Конечно можем! Даны k точек, надо многочлен степени $k-1$. Это легко и всегда возможно.

Значит схема совершенна!

Speaker notes

Хотим показать совершенность. Напомню: это когда знание долей не даёт совершенно никакой информации о том, какой может быть секрет.

Вообще, мы всегда можем перебрать всевозможные секреты. Даже когда вообще ничего не знаем. Надо как-то показать, что если знаем много точек, то этот перебор никак нельзя облегчить.

(клик) Например, выкинув заведомо невозможные секреты.

(клик) То есть надо показать, что через каждый из секретов можно провести какой-то многочлен, который к тому же будет проходить через уже известные нам точки.

(клик) Всё сводится к тому, что у нас есть точка из долей и есть ещё одна точка из секрета. Как я показывал раньше, через точек всегда можно провести многочлен степени . Именно такой и нужен по схеме.

Значит любой из секретов одинаково возможен, как было бы если мы вообще ничего не знали. Схема совершенна!

И кстати, каждый из секретов соответствует ровно одному значению последней доли. Это легко доказать, ведь интерполированные многочлены не могут пересекаться в других точках.

ИДЕАЛЬНОСТЬ

— Размер доли равен размеру секрета?

— Конечно! $f(0) \in \mathbb{F}$ и $f(i) \in \mathbb{F}$

— Значит схема идеальна.

Speaker notes

Ну здесь всё очень просто. Секрет – значение многочлена в некоторой точке. Доля — тоже значение, но уже в другой точке. Конечно же они содержат одно и то же количество информации. А значит схема идеальна.

О СОВЕРШЕННОСТИ

Совершенна ли следующая схема?

1. Дан секрет $0 \leq s \leq 100$
2. Генерируем $k-1$ случайный коэффициент $c_i \in \mathbb{Z}_{100}$ и многочлен
$$f(x) = c_1 x^{k-1} + c_2 x^{k-2} + \ldots + c_{k-2} x^2 + c_{k-1} x + s$$
3. ...

Правильная и совершенная схема Шамира:

1. Выбрать достаточно большое поле (секрет должен поместится).
2. Сгенерировать случайный многочлен степени $k-1$
$$f(x) = c_1 x^{k-1} + c_2 x^{k-2} + \ldots + c_{k-2} x^2 + c_{k-1} x + s$$
3. ...

Speaker notes

Вернёмся чуть-чуть назад, к совершенности. Вопрос — так уж ли надёжна предложенная на слайде схема?

…

(клик) Вот сравнение со схемой, которая точно идеальна, мы это доказали

(клик) Здесь проблема заключается в том, что мы взяли не поле.

АТАКА НА НЕ-ПОЛЕ

Известно кольцо: $\mathbb{Z}_{30}$.

В нём многочлен: $y = c_0 + c_1 x + c_2 x^2 + c_3 x^3$

Знаем три точки (из четырёх):

1. $(1, 18)$
2. $(2, 24)$
3. $(3, 10)$
4. $(4, y_4)$

Что мы знаем о секрете?

Speaker notes

Давайте сначала рассмотрим пример. Вот кто-то разделил секрет на четырёх участников, и мы как-то узнали три точки. Ну и знаем, что четвертая точка находится в , это не секретная информация. Её значения не знаем

Вопрос в том, как много мы теперь знаем о секрете. Конечно, схема Шамира совершенна, но здесь выбрано не поле, а какое-то кольцо вычетов по модулю 30. 30 — не простое число.

Подставим точки: $(1, 18), (2, 24), (3, 10), (4, y_4)$ в уравнение $y = c_0 + c_1 x + c_2 x^2 + c_3 x^3$, получим СЛАУ на коэффициенты:

$$\begin{cases} 18 = c_0 + c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1 \\ 24 = c_0 + c_1 \cdot 2 + c_2 \cdot 4 + c_3 \cdot 8 \\ 10 = c_0 + c_1 \cdot 3 + c_2 \cdot 9 + c_3 \cdot 27 \\ y_4 = c_0 + c_1 \cdot 4 + c_2 \cdot 16 + c_3 \cdot 64 \end{cases}$$

Решение:

$$c_0 = -y_4 - 32; \quad c_1 = \frac{11}{6} y_4 + 80; \quad c_2 = -y_4 - 34; \quad c_3 = \frac{1}{6} y_4 + 4$$

В кольце $\mathbb{Z}_{30}$ **можем** делить на: $1, 7, 11, 13, 17, 19, 23, 29$.

На 6 **не можем**.

Значит y_4 делится на 6. Значит $y_4 \in \{0, 6, 12, \mathbf{18}, 24\}$.

Всего 5 значений вместо 30!

Speaker notes

Итак, давайте посмотрим. Конечно, все эти точки лежат на каком-то многочлене. Давайте попробуем его найти. Для этого достаточно просто подставить их в уравнение и получить СЛАУ. А из неё уже легко найти коэффициенты.

Эту систему можно решить любым удобным способом. Способов решения СЛАУ очень много, не будем на них останавливаться. *(клик)* Решение будет каким-то таким.

Замечу, что решали мы не в , а в каком-то нормальном поле вещественных чисел. Из решения мы видим, какие могут быть коэффициенты вообще.

Если бы для шифрования использовали **поле**, то мы, перебирая все , получили бы всевозможные , а значит и всевозможные секреты. Ведь секрет обычно хранится в свободном члене.

(клик) Но здесь у нас **не поле**, у нас просто кольцо. А значит не для всякого числа существует обратное, чтобы их произведение давало единицу. Строго говоря, “делить” мы можем только на взаимно простые с модулем кольца. На простые числа например, вон они написаны на слайде.

Теперь посмотрим на решение внимательнее. Все коэффициенты многочлена обязательно лежат в кольце. Его так генерировали. Но ведь в решении СЛАУ нам необходимо делить на 6! *(клик)* Вон там дроби видны.

Поскольку какой-нибудь “одной шестой” в кольце не существует, получается, что необходимо чтобы делилось на 6. Только тогда дробь нормально сократится и всё будет хорошо.

Собственно, вот и всё. Зная только лишь 3 точки, мы смогли восстановить очень много информации о секрете — уменьшили диапазон возможных значений в 6 раз.

А значит конкретно эта схема, с кольцом вместо поля, не совершенна.

ЕЩЁ ПРО НЕ-ПОЛЕ

Интересный пример:

Используется $\mathbb{Z}_{15}$. Тогда:

1. $f(x) = x^2 - 1$
2. $g(x) = x^2 - 5x + 4$

У них разные секреты: $f(0) = -1$, но $g(0) = 4$.

Для восстановления обоих достаточно трёх участников. Выберем $x = 1$, $x = 4$, $x = 7$:

1. $f(1) = 0, g(1) \equiv 0$
2. $f(4) = 0, g(4) = 0$
3. $f(7) \equiv 3, g(7) \equiv 3$

Получается, что **конкретно эти три** участника не способны восстановить секрет!

Speaker notes

Давайте ещё рассмотрим такой интересный пример. Вот снова используется какое-то кольцо вместо поля, в нём для двух разных секретов сгенерировали два разных многочлена. Оба второй степени, поэтому для восстановления достаточно любых трёх участников.

(клик) Пусть из многих остались только те участники, которые владеют точками, соответствующими , и .

И видим, что оба эти многочлена одинаково хорошо подходят под их точки. Получается, что именно эта тройка участников (и не только эта) не может восстановить секрет. А ведь схемы делаются для того, чтобы **любые** три могли восстановить.

АТАКИ

С внешними врагами разобрались, как делать схемы совершенными теперь знаем.

Но схемы разделения применяются когда нет доверия даже участникам!

Участники могут назвать вместо настоящей доли что-то своё. И никто не узнает!

А если сговорилось $k - 1$ участников...

Speaker notes

Итак, переходим ко второй части. Как говорилось в самом начале, участники схемы запросто могут предать. И ничего им не может помешать сказать вместо своей настоящей доли что-то другое. В целом, участники больше ничего не могут сделать.

Вообще-то говоря, самый худший случай — почти все участники участвуют в заговоре и хотят как-то обмануть последнего. Но мы будем рассматривать только простой случай со схемой Шамира.

АТАКА НА СХЕМУ ШАМИРА

Имеется: один участник — заговорщик

Хочется: узнать секрет одному, так чтобы другие его не знали

Можем: назвать другое число в качестве своей доли

Знаем: координаты x других участников

Как?

Идея: сдвинуть свою долю так, чтобы секрет от этого **предсказуемо** сдвинулся

Speaker notes

Теперь буду показывать, как действительно провести настоящую атаку. Достаточно одного заговорщика.

Цель такой атаки заключается в том, чтобы обмануть остальных участников — заставить их поверить в ложный секрет. А заодно и узнать настоящий секрет самим.

При этом мы можем влиять только на свою долю, а знаем только лишь его и публичную информацию: где примерно находятся точки остальных участников. То есть какие они имеют, значений не знаем.

(клик)

Оказывается, есть способ посчитать, как надо сдвинуть свою долю, чтобы при этом восстановленный секрет изменился некоторым конкретным способом.

РЕАЛИЗАЦИЯ АТАКИ

Обозначения:

- (x_1, y_1) — «своя» точка (без потери общности первая).
- t — насколько мы хотим изменить секрет. $s' = s + t$
- f — «настоящий» многочлен, проходит через $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$.
- y_1' — поправленное значение «своей» доли. Хотим его найти.
- f' — «ложный» многочлен, проходит через ложную точку (x_1, y_1').
- $s = f(0)$ и $s' = f'(0) = s + t$ — восстановленные секреты: «настоящий» и «ложный» соответственно

Атака: Проводим многочлен $g(x)$ через $(0, t), (x_2, 0), (x_3, 0), \ldots, (x_k, 0)$. Тогда $y_1' = y_1 + g(x_1)$.

То есть многочлен в нуле (где секрет) равен t, а в точках других участников равен нулю. Тогда $y_1' = y_1 + g(x_1) = f(x_1) + g(x_1)$

Speaker notes

Итак, мы хотим как-то модифицировать своё значение таким образом, чтобы от этого предсказуемо изменилось значение секрета.

Что у нас есть:

1. Нам выдали точку, мы её знаем. Без потери общности будем говорить, что она первая.
2. И ещё знаем, на сколько мы собираемся изменять секрет. Хотим чтобы в результате восстановления получился , который отличается от настоящего ровно на .
3. Есть какой-то многочлен, мы его вообще не знаем, но он есть
4. Хотим как-то изменить наш , чтобы от этого секрет сдвинулся ровно на
5. После того как мы назовём ложное значение, в результате восстановления получится какой-то другой многочлен. Назовём , никакого отношения к производной не имеет.
6. Как всегда, секрет равен значению в нуле.

Атака выглядит следующим образом: Сначала мы проводим вспомогательный многочлен , такой что в “чужих” точках он равен нулю, а в точке с секретом – в нуле – он равен . Здесь мы никак не фиксируем его значение в «своей» точке . И когда мы его провели, то уже довольно легко можем подделать значение. Для этого достаточно прибавить к значение вспомогательного многочлена в «своей» точке.

На этом с атакой всё, но я хочу ещё показать почему это вообще работает.

$g(x)$ проходит через $(0, t), (x_2, 0), (x_3, 0), \ldots, (x_k, 0)$. Тогда $y'_1 = y_1 + g(x_1)$

Используем формулу Лагранжа: $g(x) = \sum_{j=1}^{n} y_j \prod_{\substack{i=1 \\ i \neq j}} \frac{x - x_i}{x_j - x_i}$

Здесь из-за $y_j = 0$ обнулится всё кроме $g(x) = t \prod_{i=2}^{k} \frac{x - x_i}{0 - x_i}$

Тогда $g(x_1) = t \prod_{i=2}^{k} \frac{x_1 - x_i}{0 - x_i}$

Теперь в $f(x) = \sum_{j=1}^{n} y_j \prod_{\substack{i=1 \\ i \neq j}} \frac{x - x_i}{x_j - x_i}$ вместо y_1 поставим $y'_1 = y_1 + g(x_1)$

$$s' = f'(0) = \underbrace{\left(\sum_{j=1}^{n} y_j \prod_{\substack{i=1 \\ i \neq j}} \frac{0 - x_i}{x_j - x_i} \right)}_{f(0)=s} + \underbrace{\overbrace{\left(t \prod_{i=2}^{k} \frac{x_1 - x_i}{0 - x_i} \right)}^{g(x_1)} \left(\prod_{i=2}^{k} \frac{0 - x_i}{x_1 - x_i} \right)}_{t}$$

Speaker notes

Собственно, ради этого момента я и рассказывал про многочлен Лагранжа.

Сначала мы проводим многочлен через точки как было описано раньше. Для этого используем формулу Лагранжа.

Поскольку у нас там везде нули, то от суммы останется только одно слагаемое, соответствует точке — как мы меняем секрет.

Дальше нас интересует только . Мы не требовали какого-то конкретного значения в точке.

Это мы прибавляем к старому значению доли, . Тем самым получаем — поддельное значение.

И теперь хотим посчитать что за секрет получится, если в схеме использовать такую неправильную долю. Для этого восстанавливаем многочлен при помощи всё той же формулы Лагранжа. Нас интересует только значение в нуле.

Если раскрыть скобки, то получаем, что написано внизу. А там получился как раз настоящий секрет, плюс какая-то штука, которая сокращается ровно в .

То есть мы сумели прибавить к значению секрета. Не удивительно ли?

ЗАЩИТА ОТ ЭТОЙ АТАКИ

Очень просто:

1. Выбирать x_i для долей случайно среди всех возможных.
2. Сделать эту информацию секретной.

Тогда доля содержит всю пару (x, y) и схема больше не идеальна.

Speaker notes

Защитится от этой атаки довольно просто. Надо всего лишь засекретить информацию о том, где находятся доли. Тогда атакующий не сможет предсказуемо изменить секрет и атака не удастся.

Строго говоря, он всё также может обмануть участников, но уже не столь предсказуемо.

P.S. Авторы работы говорят, что эта защита не идеальна и позволяет лишь заметить атаку, но атакующий всё равно узнает секрет. Однако там написано что-то очень странное, я не понял.

СТРУКТУРЫ ДОСТУПА

- Множество всех долей: $P = \{p_1, p_2, \ldots, p_n\}$
  - Число долей может не совпадать с числом участников
- Структура доступа: $\Gamma \subseteq \mathcal{P}(P)$
- Секрет могут получить только те подмножества, которые лежат в Γ.
- Пороговая схема — любые k могут получить секрет
- Γ — монотонная структура: если $A \in \Gamma$ и $A \subset B$, то $B \in \Gamma$
  - Эквивалентно, $A \in \Gamma \to \forall i (A \cup \{p_i\}) \in \Gamma$

Speaker notes

Для начала, пара новых, но важных обозначений.

Во-первых, давайте пронумеруем все доли — их штук. Потом пригодится. Иногда оно совпадает с множеством участников, но в более сложных случаях — уже нет.

Структура доступа — множество подмножеств долей. Если у нас есть какая-то группа людей, и они хотят получить секрет, то они это могут сделать только тогда, когда их доли находится в структуре доступа. Иначе они о секрете ничего не должны знать.

Раньше мы занимались только пороговыми схемами — когда любые участников могут получить доступ. Кстати, неплохо записать как выглядит множество для них.

И ещё небольшое замечание: — монотонная структура. Если какая-то группа может получить доступ, то даже если добавить к ним ещё участников, то они всё равно смогут получить доступ. Было бы очень странно пытаться сделать иначе.

ПРОСТЫЕ СТРУКТУРЫ ДОСТУПА

Задача: разделить секрет так, чтобы:

1. CEO и CTO компании могли вдвоём восстановить секрет
2. ... или три бухгалтера
3. ... или же пять обычных сотрудников

Сделаем так:

- CEO и CTO получат по 15 долей
- Каждый бухгалтер получит 10 долей
- Каждый сотрудник получит 6 долей

И тогда получим пороговую схему с $k = 30$

Откуда числа?

- $k = \text{НОК}(2, 3, 5) = 30$
- CEO и CTO: $\frac{30}{2} = 15$
- Бухгалтеры: $\frac{30}{3} = 10$
- Сотрудники: $\frac{30}{5} = 6$

Speaker notes

Вот есть такая задача как выше. Это уже не очень похоже на пороговую схему.

Но она легко реализуется, если начать раздавать одному участнику сразу по несколько долей. Причём в зависимости от статуса разное.

Тогда можно эту структуру реализовать таким образом: *(клик)*. Самым главным раздать по 15 долей, бухгалтерам по 10, а сотрудникам по 6. В этом случае у каждой группы получается ровно по 30 долей, чего вполне достаточно для восстановления секрета при пороге равным 30.

Для того чтобы посчитать число долей можно воспользоваться вот такой несложной формулой. Здесь мы считаем НОД от числа участников в каждой группе — 2 президента, 3 бухгалтера и 5 сотрудников — получим 30, это будет как раз будет порог схемы.

Затем просто распределяем эти доли между участниками групп.

Конечно, не обязательно считать НОК, можно и просто перемножить, лишь бы оно хорошо делилось. Но НОК будет самым оптимальным.

МАТРОИДЫ

Определение: это пара из двух множеств – $\langle X, I \rangle$:

- X — «носитель матроида», произвольное множество
- I — «независимые подмножества» X. Т.е. $I \subseteq \mathcal{P}(X)$

Такие что:

1. $\varnothing \in I$ (т.е. I не пусто)
2. Если $A \in I$ и $B \subset A$, то $B \in I$
3. Если $A, B \in I$ и $|B| < |A|$, то $\exists x \in A \setminus B : A \cup \{x\} \in I$
   - Можем расширить B новым элементом из A

Соглашение: Если $A \in I$, то говорим, что A независимо (и наоборот).

Speaker notes

Я должен рассказать о связи с теорией матроидов. А для этого надо рассказать о матроидах. Определяются они как пара из двух множеств.

Первое — называется носитель матроида — на самом деле не очень важно, но нужно для задания второго.

Второе — множество независимых подмножеств — собственно задаёт структуру матроида. Оно состоит из подмножеств . Powerset — множество всех подмножеств — если кто помнит.

Но имеются следующие ограничения, чтобы эта пара действительно называлась матроидом:

1. Пустое (под)множество всегда независимо.

2. Если независимо, то и всякое его подмножество тоже независимо.

3. Самое сложное и непонятное. Лучше разбирать на примерах, но я прочитаю.

   Если какие-то и являются независимыми, то мы можем расширить элементом из , и при этом получить снова независимое множество.

УНИВЕРСАЛЬНЫЙ МАТРОИД

- X — произвольное множество
- V — все подмножества, мощности не больше некоторого $k \in \mathbb{N}$

1. Поскольку $|\varnothing| \leq k$, то $\varnothing \in V$
2. Если $A \in I$, то $|A| \leq k$. Если $B \subset A$, то $|B| < k$, а значит $B \in I$.
3. Если $A \in I$, то $|A| \leq k$. Если $|B| < |A|$, то $|B| < k$. Отсюда $|B \cup \{x\}| \leq k$, а значит $B \cup \{x\} \in I$

Матроид это... такие $\langle X, I \rangle$, что:

1. $\varnothing \in I$
2. Если $A \in I$ и $B \subset A$, то $B \in I$
3. Если $A, B \in I$ и $|B| < |A|$, то $\exists x \in A \setminus B : A \cup \{x\} \in I$

Speaker notes

Пожалуй, самый простой матроид. Давайте его разберём. Снизу я выписал определение матроида.

1. Пустое множество определенно имеет мощность не больше
2. Если имеет мощность не больше , то ещё меньшее множество тем более.
3. Практически аналогично второму пункту. Если у нас есть множество, большее чем – здесь это —, и оно является независимым, то мы можем расширить каким-то элементом из него. Вообще, определение матроида требует чтобы было из , но здесь мы доказали немного более сильное утверждение.

ЦВЕТНОЙ МАТРОИД

- X – элементы, каждый раскрашен в какой-то цвет
- V — все подмножества X, в которых все элементы разных цветов

Тогда:

1. В пустом множестве все элементы разных цветов (vacuous truth)
2. Если в A элементы разных цветов, то и во всяком его подмножестве B они тоже разных цветов.
3. И в A, и в B все элементы разных цветов. Т.к. $|A| > |B|$, то в A больше цветов, чем в B. Значит найдётся $x \in A \setminus B$ такого цвета, которого нет в B. И тогда $B \cup x \in I$

Матроид это...
такие $\langle X, I \rangle$, что:

1. $\varnothing \in I$
2. Если $A \in I$ и $B \subset A$, то $B \in I$
3. Если $A, B \in I$ и $|B| < |A|$, то $\exists x \in A \setminus B : A \cup \{x\} \in I$

Speaker notes

Теперь немного сложнее. Подготовка к следующему

Первое выполняется очень просто. В пустом множестве нет двух элементов с одинаковыми цветами.

Второе тоже довольно очевидно. Если мы выкинем элемент из , то не может внезапно появится двух одноцветных элементов.

Теперь непонятное третье. Итак, вот у нас есть множества и , в обоих все элементы разных цветов. Но в элементов — а значит и цветов — больше. Вопрос в том, может ли быть такое, что все цвета из уже содержаться в , если их больше. Конечно же нет. Значит что в мы можем добавить какой-то новый элемент с каким-то новым цветом.

Таким образом вот такая пара вполне может называться матроидом.

ЛИНЕЙНЫЙ МАТРОИД

- X – вектора из какого-то линейного пространства
- V — все подмножества X, в которых вектора линейно независимы

Тогда:

1. В пустом множестве все вектора л.н.з. (vacuous truth)
2. Если в A вектора л.н.з., то и во всяком его подмножестве B они тоже раных цветов.
3. Вектора в B л.н.з., значит $\dim L(B) = |B|$. Поскольку $|A| > |B|$, то в A найдется вектор x, не входящий в линейную оболочку $L(B)$. Тогда он будет л.н.з. с векторами из B. Т.е. $B \cup x \in I$.

Матроид это... такие $\langle X, I \rangle$, что:

1. $\varnothing \in I$
2. Если $A \in I$ и $B \subset A$, то $B \in I$
3. Если $A, B \in I$ и $|B| < |A|$, то $\exists x \in A \setminus B : A \cup \{x\} \in I$

Speaker notes

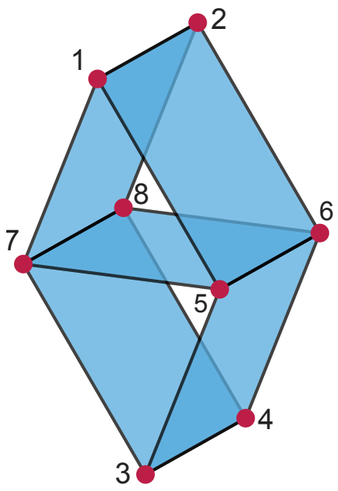
Линейный матроид очень похож на цветной. Просто вместо одноцветности здесь линейная независимость.

Первые два пункта совершенно аналогичны и просты, так что перейдём к третьему.

Давайте действовать от противного. Пусть мы не можем добавить в ни один вектор, чтобы сохранить линейную независимость. То есть всякий вектор выражается через линейную комбинацию векторов из . Но ведь в векторов больше, и все они линейно независимы. Но в линейном пространстве они никак не могут все одновременно выражаться через меньшее количество векторов. Здесь мы приходим к противоречию, более формально доказательство даётся через линейную оболочку.

МАТРОИД ВАМОСА

- $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$
- Подмножество A **независимо**, если и только если $A \leq 4$ и его нет на картинке.



Известно, что **этот матроид не является линейным**.

Другое (но равносильное) определение:

> *Положим* $a = \{1, 2\}, b = \{3, 4\}, c = \{5, 6\}$ *и* $d = \{7, 8\}$.
> *Матроид Вамоса определяется как матроид, в котором множества* $a \cup c, a \cup d, b \cup c, b \cup d, c \cup d$, *а также все подмножества из пяти или более элементов являются зависимыми.*

Например, $\{1, 2, 5, 6\} \notin V$ — оно зависимо, а $\{1, 2, 3, 4\} \in V$ — независимо.

Speaker notes

Ещё есть такой вот матроид. У меня не хватило места для доказательства того, что это матроид, но он точно является матроидом.

Чем он интересен, так это тем, что про него достоверно доказано, что он совсем не является линейным. Не все матроиды линейны — вот пример такого.

Я привожу сразу два определения. Одно по картинке, другое текстом — оба совершенно одинаковые.

СВЯЗЬ СХЕМ РАЗДЕЛЕНИЯ И МАТРОИДОВ

- Носитель матроида (X) — участники схемы
- Зависимые подмножества ($\mathcal{P}(X) \setminus V$) — подмножества участников, могущие восстановить секрет

Например, универсальный матроид (где $A \in V \Leftrightarrow |A| < k$) соответствует (n, k) схеме

Тогда верны следующие утверждения:

1. Не любой матроид может быть реализован как **идеальная** СРС (например, матроид Вамоса)
2. Но всякий **линейный** матроид (над полем) реализуется как **идеальная** СРС
3. Любой идеальной СРС соответствует матроид (не обязательно линейный)
4. Не всякая структура доступа может быть реализована идеально
5. (★) Однако для любой структуры доступа можно построить **совершенную** СРС
6. Идеальных СРС больше, чем линейных матроидов (и меньше, чем всех матроидов)

**СРС — схема разделения секрета*

Speaker notes

Итак, наконец связываем матроиды и схемы разделения вместе.

Известно, что линейные матроиды — когда использовали линейную независимость — могут быть реализованы как идеальные схемы разделения секрета. Как именно не говорится.

В целом, изучение этой связи дало миру есть несколько интересных утверждений. Доказывать я их не буду.

Хотя не любой матроид реализуется как идеальная СРС, но любой линейный — вполне. Причём любой идеальной СРС соответсвует матроид, и не обязательно линейный. Более того, не всякая структура доступа может быть реализована идеально, но всякая может быть реализована совершенно.

Дальше я хочу поговорить о пункте 5 — реализации произвольных структур доступа. Реализации не идеальной, но очень простой и часто даже вполне эффективной.

СЛОЖНЫЕ СТРУКТУРЫ ДОСТУПА

Участники:

- Алиса (a) и Берта (b)
- Степан (c) и Денис (d)

Хотим:

- Алиса и Берта вместе могут восстановить секрет
- Степан и Денис тоже могут
- Но чтобы никак по-другому было нельзя!
  Например, Алиса и Степан не могли восстановить секрет.

Структура доступа: $\Gamma = \{\{a, b\}, \{c, d\}, \ldots\}$

Проблема: пороговая схема это не позволяет.

Доказательство:

1. Участники a, b, c, d получат каждый по w_a, w_b, w_c, w_d долей, соответственно.
2. a и b могут восстановить секрет: $w_a + w_b \geq k$
3. Скажем, что $w_a \geq w_b$ (и $w_c \geq w_d$)
4. Тогда $w_a \geq w_b \implies w_a + w_a \geq w_a + w_b \geq k \implies w_a \geq k/2$
5. Аналогично с другой парой: $w_c + w_d \geq k$ и $w_c \geq k/2$
6. Теперь рассмотрим пару a и c. Получаем $w_a + w_c \geq k/2 + k/2 = k$

Значит a и c смогут восстановить секрет. Противоречие.

Speaker notes

Давайте рассмотрим вот такую ситуацию. Хотим, чтобы секрет могли восстановить только строги вот эти две пары. И никак по-другому нельзя было. Вопрос такой — можно ли это реализовать при помощи пороговых схем.

Вспоминая прошлый пример, иногда они вполне способны на что-то похожее. Но, вообще-то, тогда мы могли заменять любого сотрудника на бухгалтера или директора. А здесь что-то подобное хотим запретить.

Кстати, я там пишу многоточие, потому что мы договорились, что должно быть монотонным, а значит там ещё есть кучка подмножеств, но они нам не очень-то и интересны на самом деле.

(клик) Так вот, простые пороговые схемы такое не позволяют. Совсем никак. Давайте это докажем.

(клик) Доказательство очень простое. Допустим, что вот каждый участник получит по столько-то долей, а порог схемы равен .

Рассмотрим сначала первую пару: Алиса и Берта. Без потери общности будем считать, что у Алисы долей не меньше чем у Берты. Тогда видим, что Алиса должна получить больше, чем пополам долей.

Совершенно аналогично поступим с второй парой — там Степан должен получить больше долей.

И теперь приходим к противоречию. Алиса и Степан могут восстановить секрет, а мы этого не хотим.

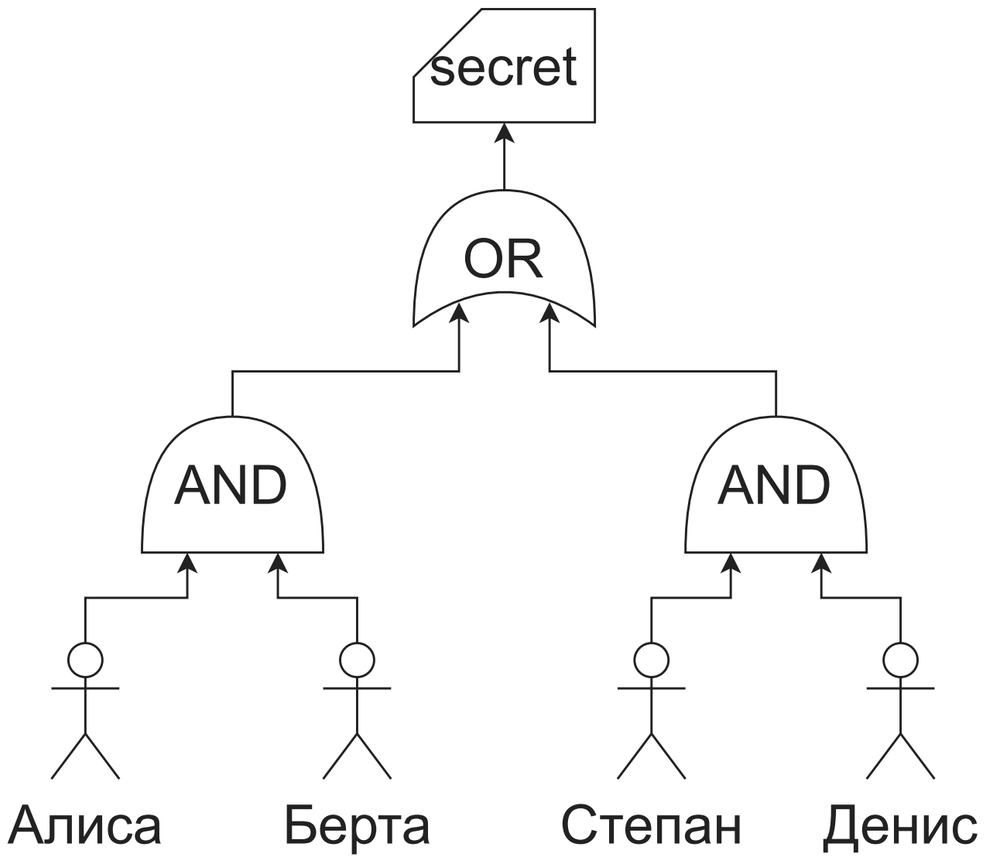
Таким образом, пороговые схемы не всегда позволяют реализовать любую структуру доступа.

БУЛЕВЫ ФУНКЦИИ

Структура доступа: $\Gamma = \{\{a, b\}, \{c, d\}, \ldots\}$

Соответствующая функция: $f(a, b, c, d) = (a \wedge b) \vee (c \wedge d)$

Хотим: $f(v_1, v_2, \ldots, v_k) \leftrightarrow (\{v_1, v_2, \ldots, v_k\} \in \Gamma)$ для всяких $v_1, \ldots, v_k$



Speaker notes

Перед тем как двинуться дальше, давайте поймем как можно представлять структуру доступа — множество подмножеств — как булеву функцию. Это очень облегчит жизнь.

Вот всё то же пример. Есть какая-то структура доступа, заданная как множество. Её можно переписать в виде вот такой булевой формулы. А ещё её можно представить картинкой.

Здесь мы требуем, чтобы для восстановления секреты присутствовали **либо** Алиса **и** Берта, **либо** Степан **и** Денис.

Причём каждая монотонная функция соответствует какому-то множеству, и наоборот. Думаю, это понятно.

РЕШЕНИЕ ПРИМЕРА

Как разделить секрет в соответствии с $f(a, b, c, d) = (a \wedge b) \vee (c \wedge d)$?

1. Разделить секрет между a и b
2. Снова разделить тот же секрет между c и d

Получим две раздельные схемы для одного секрета.

Speaker notes

Теперь наконец дам решение этого примера. Чтобы разделить секрет между двумя такими независимыми группами, нужно дважды независимо его разделить. Сначала между участниками первой группы, затем между участниками второй.

В ОБЩЕМ СЛУЧАЕ

Хотим разделить секрет s в соответствии с *формулой* F.
Обозначим это как $\$(s; F)$, где $\$(s; F)$ — множество пар: какой участник какой долей владеет.

Пусть $v_1, \ldots, v_n$ — участники. Тогда зададим $\$(s; F)$ рекуррентно:

1. $\$(s; v_i) = \{(v_i, s)\}$ — просто передать участнику эту долю (или весь секрет)
2. $\$(s; f \vee g) = \$(s; f) \cup \$(s; g)$ — если надо чтобы и f, и g могли восстановить секрет, то разделить его между ними по отдельности
3. $\$(s; f \wedge g) = \$(s_1; f) \cup \$(s_2; g)$, где $s = s_1 + s_2$ — если надо чтобы f и g только вместе могли восстановить секрет, то разделить его на сумму и раздать слагаемые (вспомните самую первую (n, n)-схему)

Speaker notes

А вот и способ, как реализовывать совершенно произвольные структуры доступа. Не самый эффективный, правда, но рабочий.

Вот у нас есть секрет и какая-то булева формула. Не функция, а вполне конкретная формула, кстати. Мы хотим разделить секрет в соответствии с той структурой доступа, которую задаёт эта формула.

Для этого нам будет достаточно разделить секрет на какие-то числа — доли — и как-то распределить их между участниками схемы. Каждый участник может иметь сразу несколько чисел. Вот функция и будет задавать это распределение. Будем задавать как множество пар: такой-то участник владеет таким-то числом.

(клик). Итак, функция у нас будет вот такая. Будем поэтапно разбирать формулу и потихоньку строить соответствие. Пример будет на следующем слайде.

1. *(клик)* Во-первых, если нам нужно разделить секрет между одним участником, то просто отдадим его ему целиком.

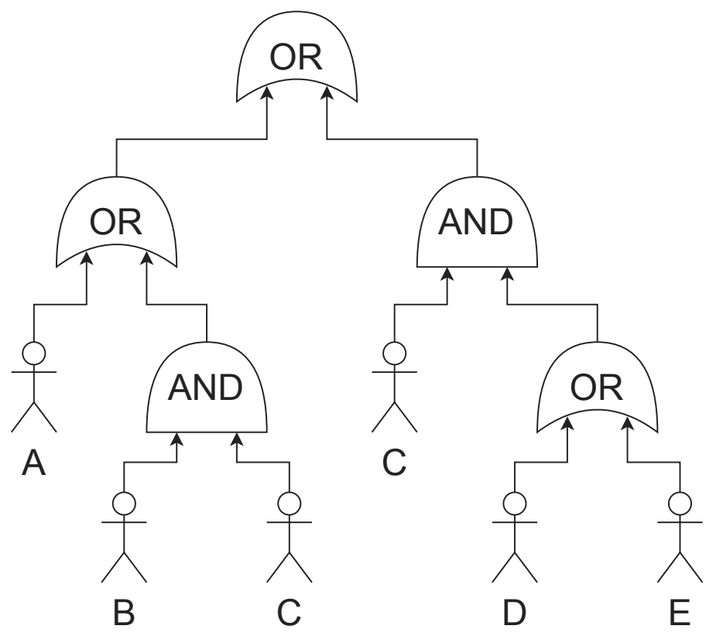
2. *(клик)* Если хотим, чтобы либо одна группа могла получить секрет, либо другая, то просто независимо разделяем этот же секрет сначала между одной группой, а потом между другой.

3. *(клик)* И последний случай. Хотим чтобы эти два участника — или группы посложнее — могли восстановить секрет только вместе. Тогда вопользуемся старой -схемой из самого начала доклада. Разобьем секрет на сумму и будем делить каждое слагаемое между этими двумя группами — или участниками.

P.S. Правила очень пригодятся для следующего слайда, но они не влезли на него.

ПРИМЕР СЛОЖНОГО РАЗДЕЛЕНИЯ

Формула: $F(a, b, c, d, e) = (a \vee (b \wedge c)) \vee (c \wedge (d \vee e))$. Надо разделить секрет s.



1. $\$(s; F) = \$(s; a \vee (b \wedge c)) \cup \$(s; c \wedge (d \vee e))$
2. $\$(s; \underline{a \vee (b \wedge c)}) = \$(s; a) \cup \$(s; b \wedge c)$
3. $\$(s; \underline{a}) = \{(a, s)\}$
4. $\$(s; \underline{b \wedge c}) = \$(s_1; b) \cup \$(s_2; c)$,

   где $s = s_1 + s_2$
5. $\$(s_1; \underline{b}) = \{(b, s_1)\}$ и $\$(s_2; \underline{c}) = \{(c, s_2)\}$
6. $\$(s; \underline{c \wedge (d \vee e)}) = \$(s_3; c) \cup \$(s_4; d \vee e)$,

   где $s = s_3 + s_4$
7. $\$(s_3; \underline{c}) = \{(c, s_3)\}$
8. $\$(s_4; \underline{d \vee e}) = \$(s_4; d) \cup \$(s_4; e)$
9. $\$(s_4; \underline{d}) = \{(d, s_4)\}$ и $\$(s_4; \underline{e}) = \{(e, s_4)\}$

Speaker notes

Вот есть такая функция. То есть здесь самый главный, затем и вместе могут восстановить секрет, либо же вместе с или — не важно с кем. Я нарисовал вот такую диаграмму для этой формулы.

1. *(клик)* Теперь давайте начнём её разбирать. Во-первых, навешиваем на всю формулу наш доллар и видим, что нужно использовать правило для “ИЛИ”. То есть мы должны разделить всё тот же между зелененьким и оранжевым.

2. *(клик)* Начнём с зелененького. Это снова ИЛИ, всё аналогично.

3. *(клик)* Теперь мы должны разделить секрет между одним участником . Это легко — отдаём секрет ему.

4. *(клик)* Продолжаем. Видим, что b и c только вместе должны восстановить секрет. Делим секрет на сумму и .

5. *(клик)* И затем сразу раздаём эти слагаемые этим двум участникам: b и c.

6. *(клик)* Переходим к правой части большой формулы. Здесь снова “И”, поэтому делим на и .

7. *(клик)* Одно из слагаемых сразу отдаем участнику c. У него теперь сразу два значения, кстати. Это означает, что схема не идеальна.

8. Возвращаемся немного назад. Здесь видим ИЛИ, поэтому разделяем между этими двумя.

9. Ну и вот мы разделили секрет ровно так, как и хотели. Причём для этого нужно разделить на две суммы, как написано внизу, и раздать числа.

Хотим разделить секрет, чтобы любые 3 из 5 могли его восстановить.

- Легко при помощи пороговых (n, k) схем $(n = 5, k = 3)$
- Сложно через функции: $F(a, b, c, d, e) = (a \wedge b \wedge c) \vee (a \wedge b \wedge e) \vee \ldots \vee (c \wedge d \wedge e)$ — всего $C_n^k = 10$ слагаемых

Это плохо.

Дополнительно к $\wedge$ и $\vee$ добавим оператор: $\texttt{THRESHOLD}_k(F_1, F_2, \ldots, F_n)$ — это выражение истинно тогда и только тогда, когда не меньше k аргументов истинны.

Тогда: $\$(s; \texttt{THRESHOLD}_k(F_1, \ldots, F_n)) = \bigcup_{1 \le i \le n} \$(s_i; F_i)$, где s_i — доли, полученные при помощи эффективной пороговой схемы.

Т.е. чтобы вычислить $\$(s; \texttt{THRESHOLD}_k(F_1, \ldots, F_n))$ нужно

1. Разделить s при помощи пороговой схемы и получить $s_1, s_2, \ldots, s_n$
2. Каждую долю распределить между соответствующей группой: $\$(s_1; F_1)$

Speaker notes

Мы получили замечательную схему, которая позволяет реализовывать любые структуры доступа, если только их записать в виде функции. Так зачем же я так долго рассказывал про пороговые схемы?

Проблема в том, что при помощи функций очень сложно и неприятно разделять секрет порогово. Получится так, что у каждого участника долей, а это очень много. К счастью, довольно легко воспользоваться преимуществами эффективных пороговых схем внутри функций.

(клик) Для этого давайте добавим новый логический оператор. Он будет истинным только тогда, когда хотя бы его аргументов истинны.

Теперь надо расширить определение функции доллар, чтобы добавить в неё поддержку этого оператора. Записывается это немного страшно, но суть такая: мы число разделяем при помощи эффективной схемы и получаем долей. Каждый из этих долей мы распределяем между — той группы, которая задаётся этим параметром.

Таким образом, мы можем достаточно эффективно реализовывать сложные структуры доступа. Не идеально, но эффективно.

ИСТОЧНИКИ

1. Введение в криптографию. Под редакцией В.В.Ященко
2. “Safeguarding cryptographic keys”, G. R. Blakley, 1979
3. “On secret sharing systems”, E. Karnin, J. Greene and M. Hellman, 1983
4. “Generalized secret sharing and monotone functions.”, Benaloh, Josh, and Jerry Leichter, 1988
5. “How to share a secret with cheaters.”, Tompa, Martin, and Heather Woll., 1989

Speaker notes

1. В “Введении” очень ёмко описываются все основные понятия и не только. А ещё она на русском языке
2. В оригинальной статье Блэкли хорошее введение — зачем схемы разделения секрета нужны. А вот сама схема там описана очень сложно и не совсем хорошо. Для этого лучше взять более поздние источники.
3. В “On secret sharing systems” очень просто и понятно описываются основы, очень рекомендую.
4. Из четвертой статьи я и взял реализацию сложных схем доступа при помощи булевых функций.
5. В “How to share a secret with cheaters.” описывается атака на схему Шамира изнутри, когда один из участников врёт.

СПАСИБО ЗА ВНИМАНИЕ

- Презентация с комментариями
- Варианты ДЗ

https://secret-sharing.sldr.xyz

Speaker notes

На сайте вы найдёте презентацию со всеми заметками и дз.