

# Некоммутативная криптография

Сергей Николенко

Криптография — АУ РАН, осень 2011

# Outline

- 1 Общие замечания и протоколы
  - Группы для криптографии
  - Протоколы на этих группах
  - Атаки
  
- 2 Криптография в группе кос
  - Группа кос
  - Протоколы

# Введение

- Кругом враги, особенно квантовые.
- Хорошо бы придумать криптографические протоколы, основанные на примитивах, которые даже квантово никто не умеет взламывать.
- Для этого придётся использовать некоммутативные группы, потому что с коммутативными уже всё ясно.
- Рассмотрим группу  $G$ , заданную образующими и соотношениями.
- Какие есть на ней вычислительные задачи?

## Задачи в группе

- Задача равенства слов (word problem): даны два слова  $u, v \in G$ . Верно ли, что  $u = v$ ?
- Это – вообще говоря, очень сложная задача.
- В частности, есть группы, для которых это неразрешимая задача; теорема Новикова.
- Нас интересуют группы с разрешимой задачей равенства слов.

## Задачи в группе

- Задача сопряжённости (conjugacy problem): даны два слова  $u, v \in G$ . Верно ли, что существует  $x \in G$ , для которого  $u^x [= x^{-1}ux] = v$ ?
- Это тоже задача сложная; бывают группы, в которых задача равенства слов разрешима, а задача сопряжённости – нет.
- Поисковая задача сопряжённости (conjugacy search problem): даны два слова  $u, v \in G$ , и известно, что существует  $x \in G$ , для которого  $u^x [= x^{-1}ux] = v$ . Найти такой  $x$ .
- Именно conjugacy search problem – это главный инструмент для классической некоммутативной криптографии.
- Часто есть основания считать функцию  $x \mapsto g^x$  кандидатом в односторонние.

## Некоммутативный протокол согласования ключей

- Простой протокол (Ko, Lee и др.), похож на Диффи-Хеллмана:
  - Публикуется элемент  $w \in G$ .
  - Алиса выбирает  $a \in G$ , посылает Бобу  $w^a$ .
  - Боб выбирает  $b \in G$ , посылает Алисе  $w^b$ .
- Если  $a$  и  $b$  коммутируют, то у Алисы и Боба появится общий ключ  $(w^a)^b = w^{ab} = w^{ba} = (w^b)^a$ .
- Чтобы Алиса и Боб выбирали коммутирующие элементы, обычно публикуются две подгруппы  $A, B \leq G$  (т.е. порождающие их элементы), для которых  $[A, B] = 1$ .

## Как выбирать группу?

- Какой должна быть базовая (platform) группа  $G$ ?
- P0** Группа  $G$  должна быть хорошо известна. Точнее, conjugacy search problem для неё либо давно и безуспешно изучалась, либо может быть сведена к другой хорошо известной задаче.
- P1** Word problem в группе  $G$  должна решаться эффективно (линейно или квадратично). Ещё лучше – если у слов в  $G$  будет какая-нибудь нормальная форма.

## Как выбирать группу?

- P2 Conjugacy search problem должно быть нельзя эффективно решить (тут, конечно, доказательств никаких не ожидается).
- P3 Должно быть возможно «замаскировать» элементы  $G$  так, чтобы нельзя было из  $x^{-1}ix$  найти  $x$ , просто посмотрев на внешний вид. Это как раз хорошо достигается при помощи нормальных форм.
- P4  $G$  должна быть группой сверхполиномиального роста, т.е. количество элементов длины  $n$  в  $G$  растёт быстрее, чем любой полином от  $n$ ; это нужно, чтобы нельзя было просто перебрать.



## Согласование ключей

- Вариант conjugacy search problem – decomposition search problem: даны два элемента  $w, w' \in G$ . Найти два элемента  $x, y \in A \subseteq G$ , для которых  $xwy = w'$ , если известно, что такие существуют.
- Обычно  $A$  – подгруппа (тогда это ещё называется double coset problem).
- Основанный на ней протокол: пусть есть публичная группа  $G$  и два подгруппы  $A, B \leq G$ ,  $[A, B] = 1$ .
  - Выбирается публично известный элемент  $w \in G$ .
  - Алиса случайно выбирает  $a_1, a_2 \in A$ , посылает Бобу  $a_1 w a_2$ .
  - Боб случайно выбирает  $b_1, b_2 \in B$ , посылает Алисе  $b_1 w b_2$ .
  - Алиса и Боб вычисляют ключ  $a_1 b_2 w b_2 a_1 = a_2 b_1 w b_2 a_1$ .

## Перекрёстный протокол

- Более надёжным может оказаться «перекрёстный» протокол: пусть есть публичная группа  $G$  и две подгруппы  $A, B \leq G$ ,  $[A, B] = 1$ .
  - Выбирается публично известный элемент  $w \in G$ .
  - Алиса случайно выбирает  $a_1 \in A$  и  $b_1 \in B$ , посылает Бобу  $a_1 w b_1$ .
  - Боб случайно выбирает  $a_2 \in A$  и  $b_2 \in B$ , посылает Алисе  $b_2 w a_2$ .
  - Алиса и Боб вычисляют ключ  $a_1 b_2 w a_2 b_1 = b_2 a_1 w a_2 b_1$ .

## Anshel-Anshel-Goldfeld

- Протокол Аншель-Аншеля-Голдфельда (ещё называется *Arithmetica key exchange*).
- Отличается тем, что не требует никаких коммутирующих подгрупп.
- Как следствие, работает для любой некоммутативной базовой группы.
- Это – очень серьёзное преимущество.

## Anshel-Anshel-Goldfeld

- Публикуем группу  $G$  и элементы  $a_1, \dots, a_k, b_1, \dots, b_m \in G$ .
- Алиса выбирает  $x \in G$  как слово в  $a_1, \dots, a_k$  и посылает Бобу  $b_1^x, \dots, b_m^x$ .
- Боб выбирает  $y \in G$  как слово в  $b_1, \dots, b_m$  и посылает Алисе  $a_1^y, \dots, a_k^y$ .
- Алиса вычисляет  $x(a_1^y, \dots, a_k^y) = x^y = y^{-1}xy$ , а затем  $[x, y] = xy^{-1}xy$ . Боб вычисляет  $y(b_1^x, \dots, b_m^x) = y^x = x^{-1}yx$ , а затем  $[x, y] = xy^{-1}xy$ . Получается общий ключ  $[x, y]$ .

## Anshel-Anshel-Goldfeld

- Казалось бы, можно было бы решить одновременную conjugacy search problem для  $b_1^x, \dots, b_m^x$  и  $a_1^y, \dots, a_k^y$  в группе  $G$  и получить секретный ключ.
- Но заметим, что чтобы воспроизвести последний шаг, Чарли должен получить  $x$  и  $y$  не просто в генераторах  $G$ , а в генераторах  $a_1, \dots, a_k$  и  $b_1, \dots, b_m$  соответственно.
- То есть ещё и решить задачу membership search: по  $x, a_1, \dots, a_k \in G$  найти выражение  $x$  через  $a_1, \dots, a_k$ .
- Membership decision problem – когда просто решить, есть ли такое – может быть неразрешимой.
- В итоге AAG работает на subgroup-related simultaneous conjugacy search problem: найти  $x \in \langle a_1, \dots, a_k \rangle$ , для которого  $a_1 = b_1^x, \dots, a_k = b_k^x$ .

## Length-based attacks

- Однако не всё так гладко, иногда ещё и атакуют нас.
- Атаки, связанные с понятием *длины* (length-based attacks) — наиболее успешные в общем случае.
- На subgroup-related simultaneous conjugacy search problem — единственные известные.

## Length-based attacks

- Нам дано  $b^x$ ; этот  $b^x$  был получен так:

$$b \rightarrow b^{\alpha_1} \rightarrow b^{\alpha_1\alpha_2} \rightarrow \dots \rightarrow b^{\alpha_1\dots\alpha_L} = b^x.$$

- Мы хотим обратить этот процесс.
- Предположение: для большинства слов  $u, w \in G$ ,  $|u^w| > |u|$ .  
Тогда, может быть,

$$|b| < |b^{\alpha_1}| < \dots < |b^{\alpha_1\dots\alpha_L}| = |b^x|.$$

## Length-based attacks

- Соответственно, вот алгоритм атаки: если мы говорим об одной conjugacy search problem, то чтобы найти  $x \in \langle a_1, \dots, a_k \rangle$ , для которого  $a^x = b$ :
  - найдём генератор  $\alpha \in \langle a_1, \dots, a_k \rangle$ , для которого  $|b| - |b^\alpha|$  максимальна;
  - повторим для  $x' = x\alpha^{-1}$  и  $b' = b^\alpha$ .
- А для simultaneous, чтобы найти  $x \in \langle a_1, \dots, a_k \rangle$ , для которого  $a_i^x = b_i$ ,  $i = 1..k$ :
  - найдём генератор  $\alpha \in \langle a_1, \dots, a_k \rangle$ , для которого  $\sum |b_i| - \sum |b_i^\alpha|$  максимальна;
  - повторим для  $x' = x\alpha^{-1}$  и  $b'_i = b_i^\alpha$ .



## Length-based attacks

- Эта атака точно работает для свободных групп, даже в нестандартном представлении, если мы можем вычислить длину по стандартному.
- Вообще, для этой атаки самое главное — правильно определить длину. Обычно это число образующих в некотором стандартном представлении.
- Но для групп кос — непонятно, работает ли.

## Атаки линейной алгеброй

- Расскажем об одном неудачном протоколе.
- Протокол Stickel: пусть  $G$  – группа,  $a, b \in G$  – некоммутирующие элементы порядков  $N$  и  $M$ :
  - Алиса выбирает два случайных числа  $n < N$ ,  $m < M$ , отправляет Бобу  $u = a^n b^m$ ;
  - Боб выбирает два случайных числа  $r < N$ ,  $s < M$ , отправляет Алисе  $v = a^r b^s$ ;
  - Они вычисляют ключ  $K = a^n v b^m = a^r u b^s$ .

## Атаки линейной алгеброй

- Обобщённый вариант: пусть  $G$  – группа,  $a, b \in G$  – некоммутирующие элементы порядков  $N$  и  $M$ ,  $w \in G$  – известный элемент:
  - Алиса выбирает два случайных числа  $n < N$ ,  $m < M$ , элемент  $c_1 \in C(G)$ , отправляет Бобу  $u = c_1 a^n w b^m$ ;
  - Алиса выбирает два случайных числа  $r < N$ ,  $s < M$ , элемент  $c_2 \in C(G)$ , отправляет Алисе  $v = c_2 a^r w b^s$ ;
  - Они вычисляют ключ  $K = c_1 c_2 a^{n+r} w b^{m+s}$ .

## Атаки линейной алгеброй

- На самом деле протокол работает нормально для многих групп.
- Более того, он может работать для полугрупп, что вовсе замечательно.
- Но Stickel рекомендовал использовать его для групп обратимых матриц  $k \times k$  над конечным полем  $\mathbb{F}_{2^l}$ .
- И это был неудачный выбор.

## Атаки линейной алгеброй

- Заметим, что Чарли достаточно найти такие  $x, y \in G$ , что

$$xa = ax, \quad yb = by, \quad u = xwy.$$

- Если он такие найдёт, он сможет по  $v$  посчитать

$$xvy = xc_2a^rwb_sy = c_2a^rxwyb^s = K.$$

- Первые два уравнения без проблем переводятся в линейные уравнения на  $2k^2$  неизвестных элементов матриц  $x$  и  $y$ .
- Но  $u = xwy$  только в квадратное хочет превращаться, а это гораздо хуже; что делать?

## Атаки линейной алгеброй

- Давайте будем искать не  $x$ , а  $x^{-1}$ ; тогда первое уравнение не изменится, а третье теперь станет линейным:

$$x^{-1}a = ax^{-1}, \quad yb = by, \quad x^{-1}u = wy.$$

- Получается  $3k^2$  уравнений на  $2k^2$  неизвестных, причём мы знаем, что решение есть.
- Это можно решить.
- Чтобы спасти протокол, достаточно использовать необратимые  $a, b, w$  (например, полугруппу всех матриц над кольцом).

# Outline

- 1 Общие замечания и протоколы
  - Группы для криптографии
  - Протоколы на этих группах
  - Атаки
  
- 2 Криптография в группе кос
  - Группа кос
  - Протоколы

# Группа кос

- Основной пример базовой группы для некоммутативной криптографии: группа кос  $B_n$ .
- Коса – это переплетённые ниточки, умножение – прикладывание двух натянутых наборов ниточек друг к другу.
- Её можно задать представлением Артина:

$$B_n = \left\langle \sigma_1, \dots, \sigma_{n-1} \mid \begin{array}{ll} \sigma_i \sigma_j = \sigma_j \sigma_i, & |i - j| \geq 2, \\ \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}, & i = 1..n-2 \end{array} \right\rangle.$$



## Группа кос

- Есть каноническая форма  $\Delta^u A_1 A_2 \dots A_l$ , где  $\Delta$  – фундаментальная коса

$$\Delta = (\sigma_1 \dots \sigma_{n-1})(\sigma_1 \dots \sigma_{n-2}) \dots (\sigma_1 \sigma_2)(\sigma_1),$$

а  $A_i$  – простые косы перестановок.

- Люди умеют делать все нужные операции за время  $O(l^2 n \log n)$ , где  $l$  – число кос перестановок.
- Т.е. есть P0, P1, P3, P4 тоже есть, а P2 предполагается.

## Группа кос

- И есть естественно возникающие коммутирующие подгруппы:

$$LB_n = \langle \sigma_1, \dots, \sigma_{\lfloor \frac{n}{2} \rfloor - 1} \rangle,$$
$$UB_n = \langle \sigma_{\lfloor \frac{n}{2} \rfloor + 1}, \dots, \sigma_n \rangle.$$

## Простейший протокол

- Получается протокол:
  - Публикуется коса  $a \in B_n$ .
  - Алиса выбирает  $x_1, x_2 \in LB_n$ , посылает Бобу  $x_1 a x_2$ .
  - Боб выбирает  $y_1, y_2 \in RB_n$ , посылает Алисе  $y_1 a y_2$ .
  - Они вычисляют общий ключ  $x_1 y_1 a x_2 y_2$ .

# Braid Diffie-Hellman Problem

- Он сводится к BDHP (Braid Diffie-Hellman Problem): по  $a, b_1, b_2 \in B_n$ , где  $b_1 = x_1 a x_2$  и  $b_2 = y_1 a y_2$  для некоторых  $x_1, x_2 \in LB_n, y_1, y_2 \in RB_n$ , найти  $y_1 b_1 y_2 [= x_1 b_2 x_2]$ .
- А на задаче поиска сопряжённого — по двум косам  $a, b$  найти  $x$ , для которого  $b = a^x$  — будет работать Аншель-Аншель-Голдфельд.
- Можно на этом и криптосистему сделать; пусть  $H : \{0, 1\}^M \rightarrow B_n$  — хеш-функция без коллизий.

## Криптосистема на группе кос

- Генерация ключей:
  - выбрать достаточно сложную косу  $a \in B_n$ ;
  - выбрать секретный ключ  $(x_1, x_2) \in LB_n \times LB_n$ ;
  - выдать публичный ключ  $(a, b) = (a, x_1 a x_2)$ .
- Кодирование (сообщения  $m \in \{0, 1\}^M$ ):
  - выбрать  $y_1, y_2 \in UB_n$ ;
  - выдать код  $(c_1, c_2) = (y_1 a y_2, H(y_1 b y_2) \oplus m)$ .
- Декодирование:
  - $m = H(x_1 c_1 x_2) \oplus c_2$ .

## Спасибо за внимание!

- Lecture notes и слайды будут появляться на моей homepage:  
`http://logic.pdmi.ras.ru/~sergey/`
- Присылайте любые замечания, решения упражнений, новые численные примеры и прочее по адресам:  
`sergey@logic.pdmi.ras.ru`, `snikolenko@gmail.com`.