Сложностная криптография

Эдуард Алексеевич Гирш

http://logic.pdmi.ras.ru/~hirsch

ПОМИ РАН

30 марта 2008 г.

Bit commitment

Alice:

Bob:

α,  ,

Bit commitment

Alice:

Bob:

α,  ,

$\boxed{\alpha}$ 

commitment

$\boxed{\alpha}$ 

Bit commitment

Alice: 

Bob:

α,  ,

$\boxed{\alpha}$ 

$\boxed{\alpha}$ 

. . . жизнь. . .

Bit commitment

Alice: 

Bob:

α,  ,

$\boxed{\alpha}$ 

commitment

$\boxed{\alpha}$ 

. . . жизнь. . .

α

Определение (Bit commitment)

. . . это протокол общения двух полиномиально ограниченных участников, для которого

- вход участника A — бит α,
  вход обоих участников A, B — параметр надёжности 1^n;
- по окончании протокола выход B — бит α'' либо “ошибка”;
- после некоторого раунда протокола ситуация такова:
  имеется значение α', такое, что
  - для “честного” A итоговый ответ B будет $\alpha'' = \alpha' = \alpha$;
  - для любого A' (вместо A) вероятность $\alpha'' \neq \alpha'$ мала $(< \frac{1}{n^k})$;
  - никакой B' (вместо B) ещё не может выдать α со сколь-нибудь существенной вероятностью $(\frac{1}{2} + \frac{1}{n^k})$;

  информация, полученная B к этому моменту, называется привязкой (commitment).

Протоколы: (A, A) — неинтерактивный, $(AB..., AB...)$ — интерактивный.

Неинтерактивная привязка на основе owp

(A, A)-протокол

Пусть $f : \{0,1\}^n \mapsto \{0,1\}^n$ — owp, B — её трудный бит.

Привязка $(f(s), B(s) \oplus \alpha)$, где случайное $s \in \{0,1\}^n$, надёжна: после её отправки

- Боб не может узнать α: такого Боба можно просить найти $B(s)$;
- узнав s *потом*, Боб найдёт α;
- Алиса не может дать другое s', для которого $f(s) = f(s')$ (его нет!).

Неинтерактивная привязка на основе owp

(A, A)-протокол

Пусть $f : \{0,1\}^n \mapsto \{0,1\}^n$ — owp, B — её трудный бит.

Привязка $(f(s), B(s) \oplus \alpha)$, где случайное $s \in \{0,1\}^n$, надёжна: после её отправки

- Боб не может узнать α: такого Боба можно просить найти $B(s)$;
- узнав s *потом*, Боб найдёт α;
- Алиса не может дать другое s', для которого $f(s) = f(s')$ (его нет!).

Упражнение

Можно было обойтись и инъективной owf f, определённой не на $\{0,1\}^n$, а на строках, выдаваемых samplerом. Какое дополнительное свойство от owf понадобилось бы?

Неинтерактивная привязка на основе owp

(A, A)-протокол

Пусть $f : \{0,1\}^n \mapsto \{0,1\}^n$ — owp, B — её трудный бит.

Привязка $(f(s), B(s) \oplus \alpha)$, где случайное $s \in \{0,1\}^n$, надёжна: после её отправки

- Боб не может узнать α: такого Боба можно просить найти $B(s)$;
- узнав s *потом*, Боб найдёт α;
- Алиса не может дать другое s', для которого $f(s) = f(s')$ (его нет!).

Упражнение

Можно было обойтись и инъективной owf f, определённой не на $\{0,1\}^n$, а на строках, выдаваемых samplerом. Какое дополнительное свойство от owf понадобилось бы?

Ответ:

Неинтерактивная привязка на основе owp

(A, A)-протокол

Пусть $f : \{0,1\}^n \mapsto \{0,1\}^n$ — owp, B — её трудный бит.

Привязка $(f(s), B(s) \oplus \alpha)$, где случайное $s \in \{0,1\}^n$, надёжна: после её отправки

- Боб не может узнать α: такого Боба можно просить найти $B(s)$;
- узнав s *потом*, Боб найдёт α;
- Алиса не может дать другое s', для которого $f(s) = f(s')$ (его нет!).

Упражнение

Можно было обойтись и инъективной owf f, определённой не на $\{0,1\}^n$, а на строках, выдаваемых samplerом. Какое дополнительное свойство от owf понадобилось бы?

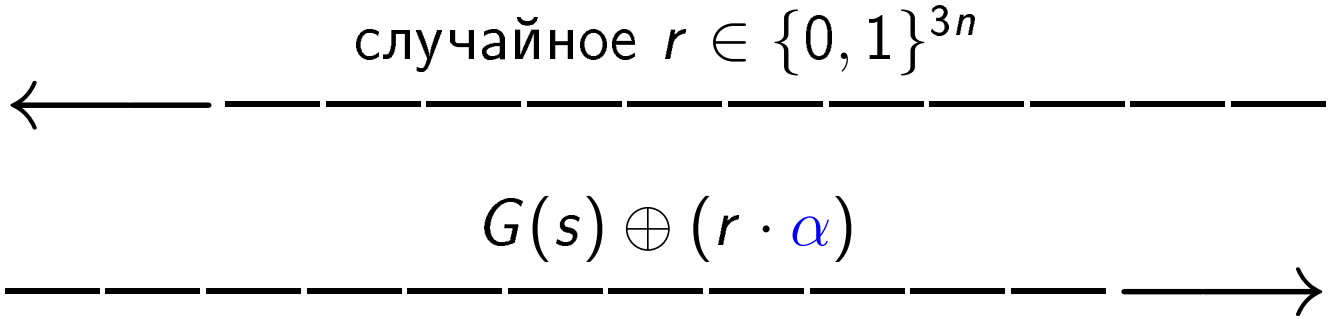
Ответ: полиномиальная разрешимость области определения f.

Интерактивная привязка на основе PRG

(BA, A)-протокол

Пусть G — $3n$–генератор.

Alice: α | Bob: r



Интерактивная привязка на основе PRG

(BA, A)-протокол

Пусть G — $3n$–генератор.

Alice: α | Bob: r

$\leftarrow$ случайное $r \in \{0,1\}^{3n}$

$G(s) \oplus (r \cdot \alpha) \rightarrow$

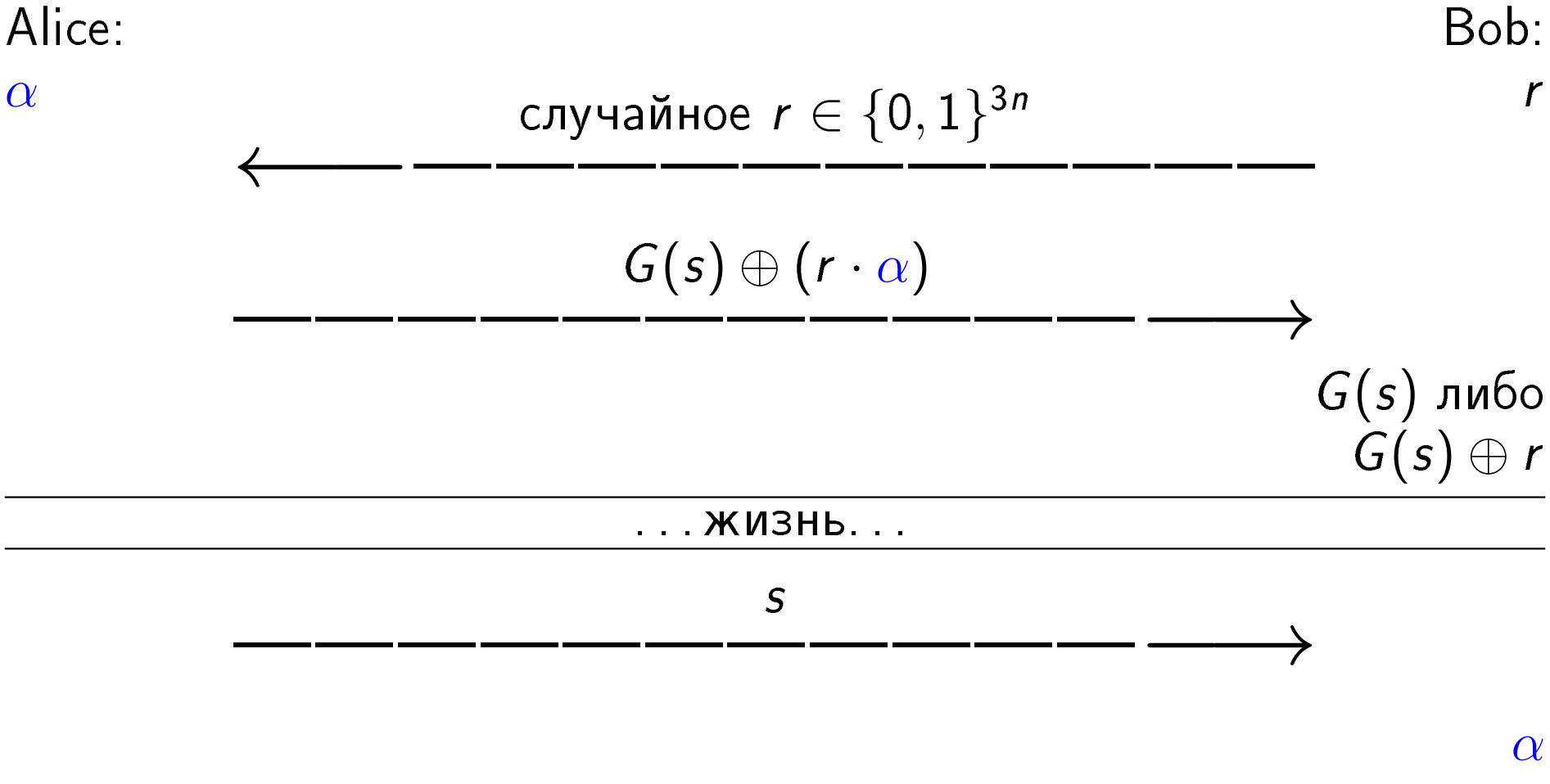
$G(s)$ либо $G(s) \oplus r$

. . . жизнь. . .

Интерактивная привязка на основе PRG

(BA, A)-протокол

Пусть G — $3n$–генератор.



Интерактивная привязка на основе PRG

Надёжность (BA, A)-протокола

1. Боб (даже выбиравший r!) не может отличить $G(s)$ от $G(s) \oplus r$:

$G(U_n)$ похоже на U_{3n} похоже на $U_{3n} \oplus r$ похоже на $G(U_n) \oplus r$.

Интерактивная привязка на основе PRG

Надёжность (BA, A)-протокола

1. Боб (даже выбиравший r!) не может отличить $G(s)$ от $G(s) \oplus r$:

$G(U_n)$ похоже на U_{3n} похоже на $U_{3n} \oplus r$ похоже на $G(U_n) \oplus r$.

2. Алиса не может подменить α:

$G(s_1) = G(s_2) \oplus r$ означает $r = G(s_1) \oplus G(s_2)$.

Таких пар (s_1, s_2) имеется 2^{2n}, и для каждой из них одно r.
А возможных r имеется 2^{3n}.

Вероятность, что Боб попадёт в плохое r — менее $\frac{2^{2n}}{2^{3n}} = \frac{1}{2^n}$.

1-out-of-2 Oblivious Transfer

Передача одного бита из двух возможных

Алиса отдаёт один из двух предметов (сама не знает, какой!).

Боб получает только один из них (ничего не знает о другом!).

Физическая реализация:

1-out-of-2 Oblivious Transfer

Передача одного бита из двух возможных

Алиса отдаёт один из двух предметов (сама не знает, какой!).

Боб получает только один из них (ничего не знает о другом!).

Физическая реализация:

- Взять два предмета, перемешать с закрытыми глазами.
- В левой руке или в правой? Тоже с закрытыми глазами.
- Оставшееся выбрасываем.

Надёжность, конечно, хромает... (кто проверит Алису?). К тому же, Боб не может выбрать того предмета, который ему нужен, а вынужден на самом деле брать случайный.

Определение ((1,2)–Oblivious Transfer, (1,2)–OT)

. . . это протокол общения двух полиномиально ограниченных участников, для которого. . .

- Вход участника A — два бита α_0, α_1,
  вход участника B — индекс $i \in \{0, 1\}$,
  вход обоих участников A, B — параметр надёжности 1^n.
- Выход по окончании протокола:
  - выход B — пара[1] битов (β_0, β_1);
  - выход A — индекс[2] j.
- Функциональность: для честных $\beta_0 = \alpha_i$.
- Надёжность:
  - для любого B' (вместо B) вероятность $\beta_1 = \alpha_{1-i}$ мала $(< \frac{1}{2} + \frac{1}{n^k})$;
  - никакой A' (вместо A) ещё не может выдать $j = i$ со сколь-нибудь существенной вероятностью $(\frac{1}{2} + \frac{1}{n^k})$.

[1] Честный B выдаёт только один бит.

[2] Честный A ничего не выдаёт.

Протокол для (1,2)–OT из расширенного tdpf

Расширенное tdpf (e, s, s', d) с трудным битом B:
есть дополнительный sampler s' по образу:

- $s'(r')$ распределено похоже на $e(s(r))$ даже для того, кто знает d,

Протокол для (1,2)–OT из расширенного tdpf

Расширенное tdpf (e, s, s', d) с трудным битом B:
есть дополнительный sampler s' по образу:

- $s'(r')$ распределено похоже на $e(s(r))$ даже для того, кто знает d,
- но $d(s'(r'))$ трудно найти без d, даже зная r'.

Протокол для (1,2)–OT из расширенного tdpf

Расширенное tdpf (e, s, s', d) с трудным битом B:
есть дополнительный sampler s' по образу:

- $s'(r')$ распределено похоже на $e(s(r))$ даже для того, кто знает d,
- но $d(s'(r'))$ трудно найти без d, даже зная r'.

Предположим, что участники *пассивно честны (semi-honest)*: следуют протоколу, но могут вычислять что-то лишнее на основе увиденного.

Протокол:

1. Алиса генерирует (e, s, s', d) и посылает (e, s, s') Бобу.
2. Боб вычисляет $a_i = e(s(r))$ и $a_{1-i} = s'(r')$ и отправляет Алисе.
3. Алиса вычисляет $\forall k \quad c_k = b_k \oplus B(d(a_k))$, посылает Бобу (c_0, c_1).
4. Боб вычисляет $b_i = B(s(r)) \oplus c_i$ и выдает его.

Упражнения

Упражнение

Написать формально доказательство надёжности (BA,A)-протокола. Указание: с помощью возможного противника можно взломать либо трудный бит, либо одно из свойств расширенного tdpf.

Упражнение

Извлечь owf из протокола bit commitment.

Упражнение

А что можно извлечь из протокола (1,2)-OT?

Secure Function Evaluation (SFE)

Алиса и Боб имеют по половине аргументов функции $c = f(a_1, \ldots, a_m, b_1, \ldots, b_m)$ и хотят её вычислить, сохранив свои аргументы в тайне.

Пассивно-честная Алиса не может вычислить ничего, кроме полиномиально вычислимой функции от $a_1, \ldots, a_m$ и c:

$\forall g\ \forall k\ \forall$ полин. $A'\ \exists$ полин. A''
$\Pr\{A'(\text{что видела Алиса}) = g(\vec{a}, \vec{b})\} \leq \Pr\{A''(\vec{a}, f(\vec{a}, \vec{b})) = g(\vec{a}, \vec{b})\} + \frac{1}{n^k}$.

То же и Боб.

SFE: алгоритм Yao

Алиса кодирует функцию f (булеву схему): таблица истинности каждого гейта кодируется случайными строчками, результаты шифруются:

0	0	1	u_0	v_0	$E_{u_0}(E_{v_0}(w_1))$
0	1	0	u_0	v_1	$E_{u_0}(E_{v_1}(w_0))$
1	0	1	u_1	v_0	$E_{u_1}(E_{v_0}(w_1))$
1	1	0	u_1	v_1	$E_{u_1}(E_{v_1}(w_0))$

$\longmapsto$

Боб всё вычисляет, для этого получает

- зашифрованную схему,
- коды входов Алисы,
- коды своих входов при помощи (1,2)–OT: что-то из v_0 и v_1,
- после вычисления — ключ для расшифровки ответа.

SFE: алгоритм Yao

Алиса кодирует функцию f (булеву схему): таблица истинности каждого гейта кодируется случайными строчками, результаты шифруются:

$$\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \longmapsto \left| \begin{array}{l} E_{u_0}(E_{v_0}(w_1)) \\ E_{u_0}(E_{v_1}(w_0)) \\ E_{u_1}(E_{v_0}(w_1)) \\ E_{u_1}(E_{v_1}(w_0)) \end{array} \right\} \text{переставить}$$

Боб всё вычисляет, для этого получает

- зашифрованную схему,
- коды входов Алисы,
- коды своих входов при помощи (1,2)–OT: что-то из v_0 и v_1,
- после вычисления — ключ для расшифровки ответа.

Готовимся к экзамену —

— решаем упражнения —

— и копим вопросы!