Многосторонние секретные вычисления

Лекция N 8 курса
“Современные задачи криптографии”
СПбГУ — SPRINT Lab

Юрий Лифшиц
yura@logic.pdmi.ras.ru

Лаборатория мат. логики ПОМИ РАН

Осень’2005

План лекции

1. Постановка задачи
2. Участники: “честные, но любопытные”
3. Нечестные участники
4. Задача

План лекции

1. **Постановка задачи**
2. Участники: “честные, но любопытные”
3. Нечестные участники
4. Задача

Неформальная постановка

Вычислительная задача:

Есть n участников

У каждого свой вход x_i

Нужно вычислить $f(x_1, \ldots, x_n)$

Неформальная постановка

Вычислительная задача:

- Есть n участников
- У каждого свой вход x_i
- Нужно вычислить $f(x_1, \ldots, x_n)$

Инфраструктура:

- Общий канал (broadcast)
- Частные каналы (например, с помощью RSA)

Неформальная постановка

Вычислительная задача:

Есть n участников

У каждого свой вход x_i

Нужно вычислить $f(x_1, \ldots, x_n)$

Инфраструктура:

Общий канал (broadcast)

Частные каналы (например, с помощью RSA)

Требования:

Корректность: получено верное значение f

Секретность: Каждый участник i не узнал ничего, кроме x_i и значения f

Пример: два миллионера

Данные

Два участника A и B

Состояние A — $a\$$, состояние B — $b\$$

Хотят узнать, кто богаче, не раскрывая никакой другой информации

Формулировка теоремы

Пусть

среди участников не более $t < n/2$ нарушителей

всем известны commitment'ы входных данных

Формулировка теоремы

Пусть

среди участников не более $t < n/2$ нарушителей

всем известны commitment'ы входных данных

Тогда

для любой полиномиально-вычислимой f

существует протокол π такой, что

Формулировка теоремы

Пусть

среди участников не более $t < n/2$ нарушителей

всем известны commitment'ы входных данных

Тогда

для любой полиномиально-вычислимой f

существует протокол π такой, что

Выполнены:

Корректность: получено верное значение f или были обнаружены нарушители

Секретность: Все, что любая группа из $t < n/2$ участников могла вычислить после выполнения протокола, она могла бы вычислить, зная только f и свои x_i

Порядок доказательства

“Получестный участник”:

- Использует действительно случайные биты
- Посылает именно то сообщение, которое должен по протоколу
- Не подслушивает сообщений между другими участниками

Порядок доказательства

“Получестный участник”:

- Использует действительно случайные биты
- Посылает именно то сообщение, которое должен по протоколу
- Не подслушивает сообщений между другими участниками

План доказательства:

- Построить протокол для получестных участников
- Заставить участников быть получестными

План лекции

1. Постановка задачи
2. **Участники: “честные, но любопытные”**
3. Нечестные участники
4. Задача

Логическая схема

Наша задача:

Вычислить f

Мы знаем, что f — полиномиально вычислима

Логическая схема

Наша задача:

Вычислить f

Мы знаем, что f — полиномиально вычислима

Факт:

Вычисление f можно представить в виде логической схемы из $\neg$ и $\wedge$ полиномиального размера

Логическая схема

Наша задача:

Вычислить f

Мы знаем, что f — полиномиально вычислима

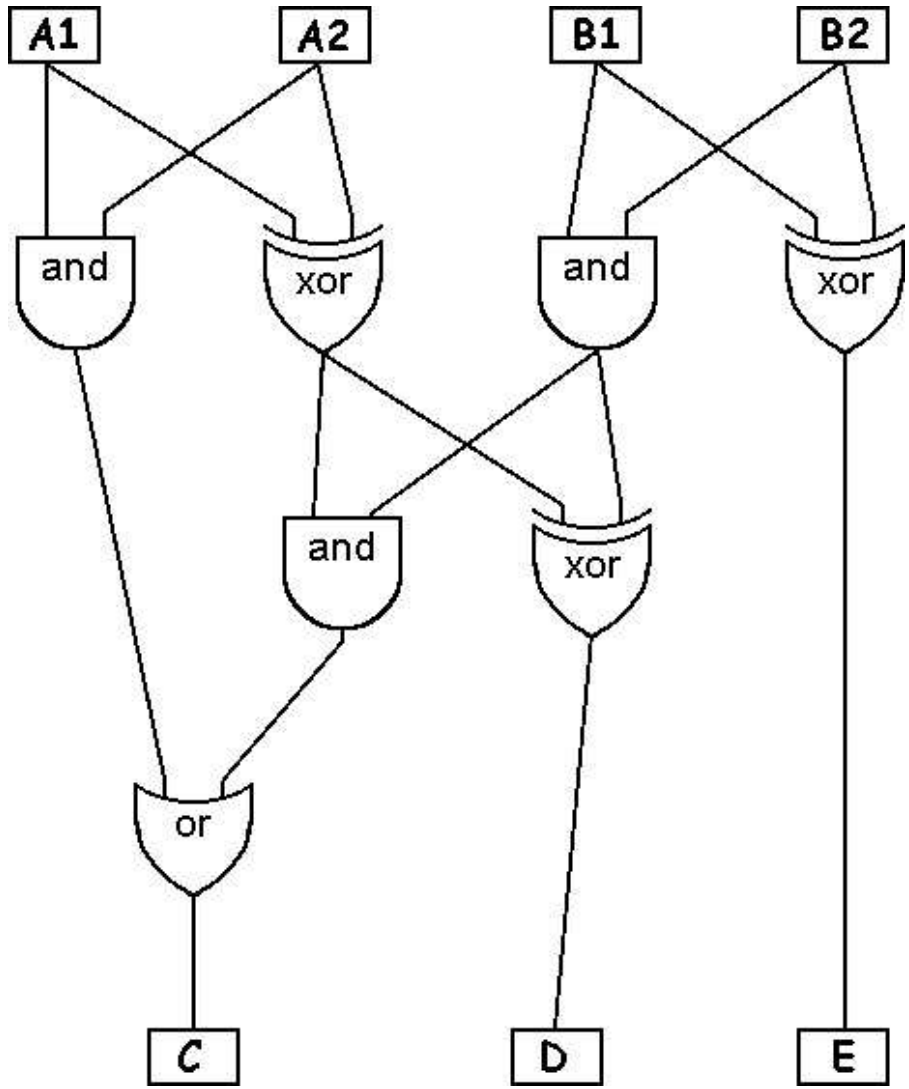
Факт:

Вычисление f можно представить в виде логической схемы из $\neg$ и $\wedge$ полиномиального размера

Идея:

Вычислять в неявном виде все значения в узлах схемы

Логическая схема



Распределение входных данных

Каждый участник P_i

для каждого своего бита b

выбирает случайно n битов, чтобы $a_1 \oplus \cdots \oplus a_n = b$

и для каждого j посылает бит a_j участнику P_j

Распределение входных данных

Каждый участник P_i

для каждого своего бита b
выбирает случайно n битов, чтобы $a_1 \oplus \cdots \oplus a_n = b$
и для каждого j посылает бит a_j участнику P_j

Наша цель

для каждого узла логической схемы
распределить n битов среди участников так,
чтобы их XOR давал значение в узле

Вычисление NOT

Как сделать разделение $\neg b$, когда есть разделение b?

Вычисление NOT

Как сделать разделение $\neg b$, когда есть разделение b?

NOT-конструкция:

Просто делаем отрицание у бита первого участника!

Вычисление AND

Что у нас есть:

Распределение $c = c_1 \oplus \cdots \oplus c_n$

Распределение $d = d_1 \oplus \cdots \oplus d_n$

Хотим построить $c \wedge d = c \cdot d = b_1 \oplus \cdots \oplus b_n$

Вычисление AND

Что у нас есть:

Распределение $c = c_1 \oplus \cdots \oplus c_n$

Распределение $d = d_1 \oplus \cdots \oplus d_n$

Хотим построить $c \wedge d = c \cdot d = b_1 \oplus \cdots \oplus b_n$

Начинаем выкладки:

$$\sum c_i \cdot \sum d_i = \sum c_i \cdot d_i + sum_{i \neq j}(c_i \cdot d_j + c_j \cdot d_i)$$

Вычисление AND

Что у нас есть:

Распределение $c = c_1 \oplus \cdots \oplus c_n$

Распределение $d = d_1 \oplus \cdots \oplus d_n$

Хотим построить $c \wedge d = c \cdot d = b_1 \oplus \cdots \oplus b_n$

Начинаем выкладки:

$$\sum c_i \cdot \sum d_i = \sum c_i \cdot d_i + sum_{i \neq j}(c_i \cdot d_j + c_j \cdot d_i)$$

Мечта:

построить b_{ij} и b_{ji} такие, что

$b_{ij} + b_{ji} = c_i \cdot d_j + c_j \cdot d_i$

Вычисление AND

Что у нас есть:

Распределение $c = c_1 \oplus \cdots \oplus c_n$

Распределение $d = d_1 \oplus \cdots \oplus d_n$

Хотим построить $c \wedge d = c \cdot d = b_1 \oplus \cdots \oplus b_n$

Начинаем выкладки:

$$\sum c_i \cdot \sum d_i = \sum c_i \cdot d_i + sum_{i \neq j}(c_i \cdot d_j + c_j \cdot d_i)$$

Мечта:

построить b_{ij} и b_{ji} такие, что

$b_{ij} + b_{ji} = c_i \cdot d_j + c_j \cdot d_i$

Тогда

$b_i = c_i \cdot d_i + sum_{j, j \neq i} b_{ij}$ — то, что нужно!

Вычисление AND II

Нужно решит задачу:

Table 1: **TPIP protocol specification**

	party A	party B
input	a_1, a_2	b_1, b_2
output	c_1	c_2
	s.t. $c_1+c_2=a_1 \cdot b_1+a_2 \cdot b_2$.	

Вычисление AND II

Нужно решит задачу:

Table 1: **TPIP protocol specification**

	party A	party B
input	a_1, a_2	b_1, b_2
output	c_1	c_2
	s.t. $c_1+c_2=a_1 \cdot b_1+a_2 \cdot b_2$.	

Идея: воспользуемся передачей данных вслепую “1-из-4”

Вычисление AND III

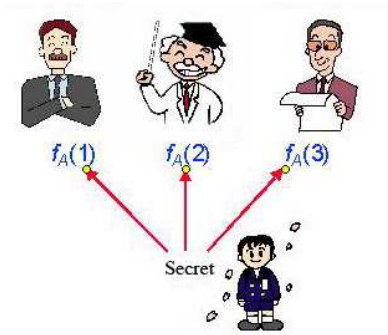
Table 3: **TPIP protocol using** OT_1^4

	party A	party B
input	a_1, a_2	b_1, b_2
The "reduction" part	chooses $c_1 \in_R \{0,1\}$. computes $s_{00} \leftarrow c_1$ $s_{01} \leftarrow c_1 + a_2$ $s_{10} \leftarrow c_1 + a_1$ $s_{11} \leftarrow c_1 + a_1 + a_2$.	computes $i \leftarrow b_1 \circ b_2$
Applying OT_1^4	-	s_i
output	c_1	$c_2 \leftarrow s_i$

План лекции

1. Постановка задачи
2. Участники: “честные, но любопытные”
3. **Нечестные участники**
4. Задача

Проверяемое разделение секрета



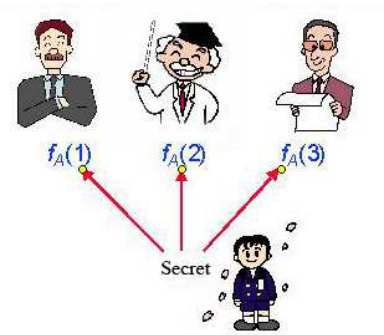
Формализация:

Разделить секрет $m \in [1..N]$ между n участниками

Любые $\lceil n/2 \rceil$ из них могут восстановить m

Любые $\lfloor n/2 \rfloor$ из них НИЧЕГО не могут узнать про m

Проверяемое разделение секрета



Формализация:

Разделить секрет $m \in [1..N]$ между n участниками
Любые $\lceil n/2 \rceil$ из них могут восстановить m
Любые $\lfloor n/2 \rfloor$ из них НИЧЕГО не могут узнать про m

Дополнительное требование:

если раздающий нарушает протокол, честные участники смогут это обнаружить

Такой протокол будем называть VSS-схемой

Сертифицированные случайные биты

1. Каждый участник распределяет по VSS-схеме свои входные данные
2. Каждый участник i выбирает для каждого j случайно r_{ij} и распределяет эти значения по VSS-схеме
3. Участники открывают r_{ij} для всех пар $i \neq j$
4. Случайные биты участника i считаются $r_i = r_{1i} \oplus \cdots \oplus r_{ni}$

Сертифицированные случайные биты

1. Каждый участник распределяет по VSS-схеме свои входные данные
2. Каждый участник i выбирает для каждого j случайно r_{ij} и распределяет эти значения по VSS-схеме
3. Участники открывают r_{ij} для всех пар $i \neq j$
4. Случайные биты участника i считаются $r_i = r_{1i} \oplus \cdots \oplus r_{ni}$

Наблюдения:

Случайные биты каждого участника от него не зависят

Большинство честных участников может восстановить случайные биты и входные данные любого нарушителя

Исполнение протокола

Каждый шаг протокола определен как функция от входных данных, случайных битов и предыдущих сообщений, полученных участником.

Исполнение протокола

Каждый шаг протокола определен как функция от входных данных, случайных битов и предыдущих сообщений, полученных участником.

Теперь каждый шаг будет:

1. Послать само сообщение
2. Доказать с нулевым разглашением, что

Существует строка r, которая могла быть порождена на предыдущем этапе, и такое значение входных данных, не противоречащее распределению на первом этапе, что при применении к ним функции протокола получилось то сообщение, которое и было послано

План лекции

1. Постановка задачи
2. Участники: “честные, но любопытные”
3. Нечестные участники
4. **Задача**

Задача

Как успехи с разрезом графа степени 3 (задача из предыдущей лекции)?

Задача

Как успехи с разрезом графа степени 3 (задача из предыдущей лекции)?

Постройте протокол для передачи данных вслепую “1-из-4”

Последний слайд

Если не запомните ничего другого:

- Многосторонние секретные вычисления: получить общий результат, не раскрывая своих данных

Последний слайд

Если не запомните ничего другого:

- Многосторонние секретные вычисления: получить общий результат, не раскрывая своих данных
- Доказательство в два этапа: протокол для получестных участников + система контроля

Последний слайд

Если не запомните ничего другого:

- Многосторонние секретные вычисления: получить общий результат, не раскрывая своих данных
- Доказательство в два этапа: протокол для получестных участников + система контроля
- Используемые примитивы: разделение секрета, передача данных вслепую, нулевое разглашение

Последний слайд

Если не запомните ничего другого:

- Многосторонние секретные вычисления: получить общий результат, не раскрывая своих данных
- Доказательство в два этапа: протокол для получестных участников + система контроля
- Используемые примитивы: разделение секрета, передача данных вслепую, нулевое разглашение

Последний слайд

Если не запомните ничего другого:

- Многосторонние секретные вычисления: получить общий результат, не раскрывая своих данных
- Доказательство в два этапа: протокол для получестных участников + система контроля
- Используемые примитивы: разделение секрета, передача данных вслепую, нулевое разглашение

Вопросы?