Многосторонние секркткные вычисления

Ю. Лифшиц*

11 декабря 2005 г.

План лекции

1. Постановка задачи
2. Любопытные участники
3. Нечестные участники

1 Постановка задачи

1.1 Неформальная постановка

Что такое многосторонние вычисления? У нас есть n участников и у каждого свои входные данные. И есть функция, которую нужно посчитать. Это, так сказать, многосторонние вычисления без слова "секретные".

Для достижения цели у них есть два типа связи:

- Часные каналы

  Непрослушиаемые каналы между каждыми двумя участниками

- Общий канал(broadcast)

Частных может не быть, но они могут быть заменены криптосистемами с открытым ключом.

Задача: вычислить функцию f с двумя свойствами:

- Корректность

  Получено верное значение f.

- Секретность

  Каждый участник i не узнает ничего, кроме своего входа и значения f.

*Законспектировал Н.Афанасенко.

Но напрямую корректности и секретности достичь невозможно. Возможно это сделать только с помощью криптографии и секретность будет не абсолютной, а криптографической. Например, возможно ли узнать чье число больше, не раскрывая чисел? Ответ: возможно, если есть третий человек, которому все верят, но такого человека нет. Кстати, есть другая постановка задачи о многосторонних вычислениях: есть несколько функций, результаты вычислений каждой из которых должен узнать только один человек из n. Эта задача очевидно сводится к поставленой выше при помощи использования криптосистемы с открытым ключом. Т.о. если мы научимся решать задачу для одной функции f, которую должны знать все, то мы сможем научиться решать задачу для набора функций.

Пример многосторонних секретных вычислений - электронные выборы. Входные данные каждого участника - это его голос, а функция выдает ответ большинства.

1.2 План действий

Итак, сегодня мы будем рассматривать ситуацию, когда есть честные участники, которые полносью следуют протоколу и есть нарушители, которые могут обьединяться в группы, но не могут взламывать криптосистемы, причем нарушителей не более чем половина от общего числа участников. Также всем известны commitment'ы для контроля нарушений, связанных с отсылкой разных данных разным участникам. Они публикуются в зашифрованном виде и поэтому не могут ничем помочь взломщикам.

Мы будем строить протокол для любой полиномиально вычислимой функции f, которая, в свою очередь, может выдавать не только свое значение, но и имя нарушителя. В таком случае f надо будет запускать уже без этого участника. Далее, любая группа, состоящая из меньше чем половины участников, не может ничего вычислить кроме того, что она могла бы вычислить, зная значение f и свои входные данные, после завершения протокола.

Решение будет в два этапа. Сначала определим такое понятие, как "получестный участник". "Получестный участник это такой участник, который пытаается вычислить чужие входные данные, но при этом следует протоколу. Важны три момента: он посылает действительно случайные биты, посылает именно то сообщение, которое должен и не слушает чужие каналы. И мы посороим сначала протокол для "получестных"участников, а затем покажем как можно заставить участников посылать действительно случайные биты и верные сообщения.

2 Любопытные участники

Поговорим сначала о том, как мы будем представлять функцию f. Мы будем ее вычислять как будто бы она является логической схемой от двух операций: NOT и AND. Основная идея - считать по этой схеме. Т.е. для

каждого узла по очереди считать значения. Каждый участник знает значение в своей вершинке. И дальше, если мы будем хранить значения во всех узлах в зашифрованном виде, то так сможем потихоньку продвигаться по схеме.

Замечательный факт: любую полиномиально вычислимую функцию можно представить в виде полиномиального размера логической схемы. Причем XOR и OR можно представить с помощью AND и NOT.

Итак, как мы будем поступать? Вот, я участник P_i и у меня есть бит b и я делаю такое распределение битов для каждого участника, чтобы они в XOR'е давали бит b. И каждому участнику посылаю его бит. Т.о. мы сделали такое разбиение для первого уровня логической схемы.

Чтобы сделать NOT b нужно сделать отрицание у бита, скажем, первого участника. Получим нужное распределение, причем никто ничего лишнего не узнал.

Гораздо хитрее с AND'ом. Нужно построить c AND d, причем у нас есть распределение для c и для d. Будем считать, что мы находимся в поле остатков mod 2, т.е. AND это то же самое, что и умножение, а XOR это то же самое, что и просто сложение. Итак, мне надо произведение c и d представить в виде суммы n слагаемых, т.е. хотим сосчитать произведение суммы c_i и суммы d_i.

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

Т.о. все сводится к построению b_{ij} и b_{ji} таких, что $b_{ij} + b_{ji} = c_i \cdot d_j + c_j \cdot d_i$
Тогда $b_i = c_i \cdot d_i + sum_{j,\, j \neq i} b_{ij}$

Мы свели задачу о вычислении многосторонненго AND'а к перевычислению другой функции, кода у одного участника есть a_1, a_2, а у другого b_1, b_2 и нужно получить $a_1 \cdot b_1 + a_2 \cdot b_2$. И воспользуемся мы здесь передачей данных вслепую "1-из-4".

Скажем, я участник у которого есть a_1, a_2. Я выбираю случайный бит c_1 и для каждой пары b_1, b_2 формирую особым образом s, которая и является искомым значением для разных b: $s_{00} \leftarrow c_1$ $s_{01} \leftarrow c_1 + a_2$ $s_{10} \leftarrow c_1 + a_1$ $s_{11} \leftarrow c_1 + a_1 + a_2$

Что делает второй участник? Он забирает значение, соответствующее его битам. Теперь сложив наши результаты мы получим искомый бит.

Вот мы построили протокол для честных участников. Еще раз: основная идея состоит в представлении функции в виде лог.схемы и разделение секрета для всех промежуточных результатов и считать и в неявном виде.

3 Нечестные участники

Для начала вспомним про проверяемое разделение секрета и введем доп. требование: если раздающий частички секрета нарушает протокол, то участники смогут это обнаружить. Такой протокол назовем VSS-схемой.

Итак, что мы делаем? Мы разделяем секрет как и для честных участников, но не по XOR'у, а по VSS-схеме. Дальше начинаем друг другу создавать случайные биты,т.е. идея в том, что каждый участник не сам создает свои

биты, а другие участники ему их вежливо подсунут. Каждый участник i готовит для каждого участника j строчку r_{ij} и распределяет по VSS-схеме. Потом раскрываем все r_{ij}, кроме r_{ii}. И случайные биты каждого участника считаются XOR'ом всех приготовленных ему битов плюс своих.

Что этим достигается? Случайные биты саждого участника от него не зависят, поскольку нечестных участников у нас меньше чем честных. Причем, если все честные участники соберутся вместе, он смогут собрать строчку любого участника.

Теперь каждый шаг алгоритма отределен как функция от случайных битов, от предыдущих сообщений и от входных данных участника. Правильно? Да. И теперь, что вы думаете мы будем делать? Теперь каждый участник будет посылать свое сообщение и доказывать с нулевым разглашением, что он воспользовался теми случайными битами, которые согласованы с первыми слагаемыми, объявленными в r_i и тем разделением секрета, которое тоже объявлено. Если он это докажет, то он убедит всех в том, что он использовал правильные биты и при этом он ничего лишнего не расскажет.

Все. На этом общий рассказ о протоколе закончен.

Итоги

Если не запомните ничего другого:

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

Литература

Список литературы использованной при подготовке к лекции

http://www.wisdom.weizmann.ac.il/~oded/LN89/lect14.ps