Гомоморфное шифрование

(защищенные облачные вычисления)

С.Ф. Кренделев

Лаборатория Параллелс-НГУ

Гомоморфизмы

- Пусть $A< +, \times, 1_A, 0_A >$, $B <\oplus, \otimes, 1_B, 0_B>$
- Два коммутативных кольца с соответствующими сложениями, умножениями, нулем и единицей.
- Гомоморфизмом колец называется всякое отображение $F: A \rightarrow B$ такое, что для всяких x, y из A

1. $F(0_A) = 0_B,\ F(1_A) = 1_B,$
2. $F(x + y)=F(x) \oplus F(y)$
3. $F(x \times y)=F(x) \otimes F(y)$

- Основное приложение гомоморфизма
- Пусть у клиента есть данные $< x_1,\ x_2,\ \ldots.\ , x_n >$
- из кольца A и он хочет вычислить некоторую полиномиальную функцию от этих данных $G(x_1,\ x_2,\ \ldots.\ , x_n)$

  Используя гомоморфное отображение клиент посылает серверу функцию, которую необходимо вычислить и данные

  $< y_1,\ y_2, \ldots.\ , y_n >$, где $y_i = F(x_i)$

Сервер вычисляет значение функции, и возвращает результат. Клиент находит неоходимое значение своей функции.

- В связи с таким приложением гомоморфизмов, требование на них ужесточаются. Гомоморфизм должен шифровать данные, что означает, что их должно быть много, если он единственный то данные клиента можно восстановит. Другое желательное свойство это возможность представлять гомоморфизм в виде открытого ключа. Если удается построить такой гомоморфизм то говорят, что построена система гомоморфного шифрования. Проблема построения гомоморфного шифрования была поставлена в 1978 году Ривестом, Адельманом и Дертузо причем они считали, что такое построение невозможно.

- В 2009 году Крэйг Гентри построил систему которую назвал “Fully Homomorphic Encryption based on Ideal Lattices”. Суть системы в том, что она доказывала возможность построения гомоморфного шифрования, но практически нереализуема . С моей точки зрения здесь нет гомоморфизма.
- Наличие принципиальной возможности построения гомоморфного шифрования, вызвало интерес, особенно с точки зрения приложений в защищенных облачных вычислениях. В частности Defense Advanced Research Projects Agency(DARPA) предложило грант на создание такой системы.

- Проблема построения гомоморфной системы над числами, заключается в том, что в числовых системах гомоморфизмов мало. Поэтому в качестве колец надо брать другие объекты.
- Примеры объектов в которых гомоморфизмов много – функции над некоторым множеством, в этом случае любое преобразование над множеством автоматически влечет построение гомоморфизма для функций. Тем самым необходимо для объектов кольца получить представление в виде функций.

Элементарный пример гомоморфного шифрования

- Пусть кольцо A- это множество многочленов от одной переменной с коэффициентами из кольца целых чисел **Z** , со стандартными операциями в кольце многочленов. Пусть дан некоторый многочлен $a_0 + a_1 x + \ldots + a_n x^n$, выберем произвольное целое число c, и сделаем замену переменных x=y-c. Тогда получается новый многочлен $a_0 + a_1 (y-c) + \ldots + a_n (y-c)^n$ если раскрыть все скобки, то получаем многочлен $b_0 + b_1 y + \ldots + b_n y^n$, тем самым получено отображение множества многочленов в себя. Данное отображение взаимно однозначно и является гомоморфизмом. Тем самым если клиент хочет секретно перемножить два многочлена, то он выбирает секретный параметр c, преобразует многочлены, отправляет серверу который их перемножает и возвращает клиенту, тот восстанавливает то, что ему нужно.

- У данного примера недостаток один - при умножении неконтролируемый рост степени полиномов, что означает, что много раз умножать нецелесообразно. Второй потенциальный недостаток, рост коэффициентов полиномов можно преодолеть рассматривая полиномы с коэффициентами в кольцах вычетов по модулю m.

- Используя этот простейший пример удается построить полною гомоморфное шифрование в котором нет роста многочленов и есть возможность строить систему с открытым ключом.

Что на самом деле сделали в IBM.

- Секретными являются вычисления в поле $\mathbb{Z}_2$ тем самым речь идет о битах. Прежде всего кодируются биты пусть m некоторый бит сопоставим ему число по правилу – выбираем три числа r, k, q, $r \ll k$ (k секретный параметр) и строим кодовое число $c = 2r + m + (2k+1)q$. Заметим, что выполнены следующие условия

$$c \bmod(2) = (m+q) \bmod(2) \quad (1)$$

$$[c \bmod(2k+1)] \bmod(2) = m \quad (2)$$

- второе условие означает, что если известен секретный параметр, то бит однозначно декодируется, первое условие что по известному c , бит не определяется.

Битовые операции в этой схеме

- Пусть даны два бита m_1, m_2, сопоставим им числа

$$c_1 = 2r_1 + m_1 + (2k+1)q_1, c_2 = 2r_2 + m_2 + (2k+1)q_2$$

- тогда имеет место

$$c_1 + c_2 = 2r_1 + m_1 + 2r_2 + m_2 + (2k+1)(q_1 + q_2) =$$

$$2(r_1 + r_2) + m_1 + m_2 + (2k+1)(q_1 + q_2)$$

- Если $2(r_1 + r_2) + m_1 + m_2 < (2k+1)$ то

$$[(c_1 + c_2) \bmod (2k+1)] \bmod (2) = (m_1 + m_2) \bmod (2)$$

- Аналогично обстоит дело с произведением.
- Полученное отображение не является гомоморфизмом, но операции шифрует

- Вся эта конструкция предназначена только для того, что бы скрыть биты с которыми производятся вычисления. Тем самым данный подход эквивалентен следующей конструкции.
- Пусть m_1, m_2 два бита над которыми надо произвести какую-нибудь операцию. При шифровании (кодировании) невозможно определить во что перешел первый бит он мог быть 0 и перейти в 0 или 1, мог быть 1 и перейти в 0 или 1. Аналогично про второй бит. Это следует из свойства кодирования (1). Если посмотреть на множество пар бит, то возможных пар кодирования может быть 4 штуки (0,0), (0,1),(1,0), (1,1). Расположим эти пары в следующем порядке

$$\begin{array}{lcccc} m_1: & 0 & 0 & 1 & 1 \\ m_2: & 0 & 1 & 0 & 1 \\ & 0 & 1 & 2 & 3 \end{array}$$

- Внизу номер возможного состояния из 4. В такой таблице описаны всевозможные состояния пары бит.

- Предположим, что нужно вычислить некоторую функцию $\mathrm{f}(m_1, m_2)$ от двух переменных, тогда делается вычисление по столбцам, другими словами вычисляются значения функции для каждого возможного значения переменных. Возможных значений также 4, которые тоже можно записать в строчку. Если эти значения вернуть клиенту то ему надо выбрать позицию в которой находились реальные результаты и в позиции с тем же номером находится результат вычисления. Номер позиции это секретное число.

- Предположим, что нужно провести вычисления с тремя битами m_1, m_2, m_3 тогда таблица будет следующая

m_1:	0	0	0	0	1	1	1	1
m_2:	0	0	1	1	0	0	1	1
m_3:	0	1	0	1	0	1	0	1
	0	1	2	3	4	5	6	7

- Секретом является число 5.
- Теперь можно вычислять любую функцию от трех переменных $\mathrm{f}(m_1, m_2, m_3)$

- По построению с ростом количества переменных наблюдается экспоненциальный рост для размера строчек таблицы. Однако это легко преодолевается поскольку можно использовать не все возможные состояния битов а только некоторые из них, среди которых есть нужные данные.
- Плюс этой конструкции заключается в том, что одновременно можно получить результат для нескольких данных. Другими словами вычисления осуществляются параллельно

Вот и настоящий гомоморфизм

- Всякую строчку в таблицах можно интерпретировать как функцию целочисленного аргумента со значениями в некотором кольце. Множество таких функций так же образует кольцо. Преобразование, которое переводит функцию в значение этой функции в фиксированной точке это гомоморфизм. Следовательно получена простейшая реализация настоящего гомоморфного шифрования.

- Замечание 1: пример с заменой переменных в начале доклада это просто алгебраическая трактовка вычисления функции в точке.
- Замечание 2: обобщения на случай многих переменных, для вещественных чисел, для расширений полей, для гомоморфного шифрования с открытым ключом и т.п. является делом техники. Все эти обобщения рассматриваются в лаборатории как с точки зрения реализации так и с точки зрения возможных приложений.

Благодарю за внимание

- Вопросы Есть ?