ПРИЛОЖЕНИЕ ДЛЯ БЕЗОПАСНЫХ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ГОМОМОРФНОЙ КРИПТОГРАФИИ

М. А. Кадан
Кафедра системного программирования и компьютерной безопсности, Гродненский государственный университет им. Янки Купалы
Гродно, Республика Беларусь
E-mail: kadan.maria@gmail.com

В докладе ставится задача изучения подходов к проведению облачных вычислений над зашифрованными данными методами гомоморфного шифрования и реализация гомоморфной криптосхемы в виде приложения на языке Objective-C для iOS для оценки производительности и пригодности алгоритмов.

Введение

Не смотря на широкое использование облачных вычислений, хранение и обработка конфиденциальных данных в облачной инфраструктуре небезопасны, что обусловленно, согласно [1], следующими рисками нарушения конфиденциальности данных в облаке:

1. Доступ к данным со стороны провайдера
2. Публичное разглашение данных (доступ неограниченного круга лиц)
3. Выемка данных или носителей из датацентра провайдера (органы правопорядка, сотрудники датацентра)
4. Ошибки изоляции среды (доступ одного клиента облака к данным других клиентов)
5. Недостаточное уничтожение данных провайдером при уходе клиента или стирании данных.

Разумным решением проблемы конфиденциальности данных может служить шифрование всех приватных данных перед передачей в облако. Однако, к сожалению, все распространенные в настоящее время криптографические алгоритмы не позволяют производить произвольные вычисления над зашифрованными данными, существенно ограничивая возможности использования облачных ресурсов.

Одной из основных задач криптографии в данном направлении является обеспечение возможности проведения вычислений над зашифрованными данными без их дешифрования. Данным свойством обладает полностью гомоморфное шифрование.

I. Гомоморфное шифрование

Гомоморфное шифрование – форма шифрования, позволяющая производить определённые математические действия с зашифрованным текстом и получать зашифрованный результат, который соответствует результату операций, выполняемых с открытым текстом [2].

Понятие схем шифрования, допускающих нетривиальные вычисления над зашифрованными данными, впервые было предложено еще в 1978 году авторами криптосистемы RSA, которая является гомоморфной относительно умножения.

Пусть $E(m,k)$ – функция шифрования, где m – открытый текст, k — ключ шифрования. Функция E гомоморфна относительно операции op над открытыми текстами, если существует эффективный алгоритм M, который, получив на вход любую пару криптограмм вида $E(m_1,k)$, $E(m_2,k)$, выдает криптограмму, при дешифровании которой будет получен открытый текст $m_1\ op\ m_2$ [3].

Как правило, рассматривается следующий важнейший частный случай гомоморфного шифрования. Для данной функции шифрования E и операции op_1 над открытыми текстами существует операция op_2 над криптограммами такая, что из криптограммы $E(m_1,k) op_2 E(m_2,k)$ при дешифровании извлекается открытый текст $m_1\ op_1\ m_2$:

$$E(m_1\ op_1\ m_2,\ k)\ =\ E(m_1,k)\ op_2\ E(m_2,k)$$

Особый интерес представляет возможность построения полностью гомоморфного шифрования, т.е. шифрования, позволяющего проводить над шифртекстами любые необходимые вычисления. К примеру, такую криптосистему можно было бы получить в случае, если бы она была гомоморфна одновременно и по операции сложения, и по операции умножения:

$$D(E(m_1,k)\ op_1\ E(m_2,k),k)\ =\ m_1 \cdot m_2$$

$$D(E(m_1,k)\ op_2\ E(m_2,k),k)\ =\ m_1\ +\ m_2$$

Здесь – шифртекст, op_1 и op_2 – операции над шифртекстами, соответствующие операциям $\cdot$ и $+$ над открытыми текстами.

Таким образом, для создания защищенного облачного сервиса необходимо шифровать поступающие на него данные с помощью полностью гомоморфной схемы шифрования. В этом случае окажется возможным проводить вычисления над данными непосредственно в зашифрованном виде на стороне сервера. При этом шифрование данных будет проводиться на стороне клиента.

II. Требования к реализации гомоморфного шифрования

Автору не известны реализации гомоморфного шифрования, пригодные для внедрения в реальные программные системы. В то же время, не составляет труда сформулировать, что такая реализация должна удовлетворять, как минимум, следующим требованиям:

1. Множество поддерживаемых математических функций должен покрывать повседневные нужды программистов.
2. Диапазоны значений чисел должны покрывать по крайней мере стандартные типы данных, а вычисления, производимые над зашифрованными данными, соответствующие такому размеру чисел, – иметь приемлемую производительность.
3. Точность и скорость вычислений не должны деградировать в течение вычислений.
4. Количество доступных ключей должно быть достаточно велико, чтобы исключить атаку полным перебором.

Самым сложным является первое требование, и на данный момент имеется лишь приближенное решение этой проблемы с помощью рядов Фурье.

III. Реализация гомоморфного шифрования

Существующие облачные сервисы не являются полностью защищенными. В лучшем случае есть возможность лишь зашифровать данные на стороне пользователя, но это бывает крайне редко. Обычно данные шифруются ключом, который хранится в том же самом облаке.

Из сказанного выше вытекают несколько конкретных требований к безопасному (защищенному) облачному сервису:

1. Данные клиента должны храниться в таком виде, что при их чтении невозможно было бы понять, что это за данные. Причем очевидно, что данные должны поступать на сервер уже зашифрованными. Что означает, что шифрование должно проводиться ещё на стороне клиента.
2. Должна быть возможность обрабатывать эти данные не расшифровывая. Иначе облачный сервер становится всего лишь безопасным хранилищем. А для каждой операции над данными потребуется пересылать их на сторону клиента.

В качестве заключительного этапа исследования была разработано экспериментальное приложение на языке ObjectiveC для оценки производительности и пригодности алгоритмов, реализующее полностью гомоморфную криптосистему, с возможностью сложения и умножения над зашифрованными данными в кольце Z_n. Алгоритм, реализованный в приложении, представляет собой расширение кольца Z_2 на кольцо $Z_{1000000}$. Данное требование означает, что операнды вычислений должны быть неотрицательными числами, не превосходящими 10^7. В то же время, для корректности вычислений, сумма и произведение двух чисел также не должна превосходить данный лимит.

Алгоритм работы программы может быть описан следующим образом:

1. Генерация публичного и приватного ключей
2. Шифрование операндов
3. Применение к операндам одной из функций (сложение или умножение)
4. Дешифрование полученного на предыдущем шаге результата.

Рассмотрим данный процесс более детально. В качестве секретного ключа будем использовать некоторое число p, взаимно простое с порядком группы Z_n. Числа r и q выбираем произвольно, $z = n * r + m$, где m – исходный текст.

Процесс шифрования в данном примере будет иметь вид: $c = z + pq = m + n * r + pq$. Процесс дешифрования: $m = (c \bmod p) \bmod n$.

Ниже представлен пимер шифрования двух целых чисел и выполнения операций сложения и умножения над ними. Данные отображаются в hex-формате и в виде символов таблицы ASCII.

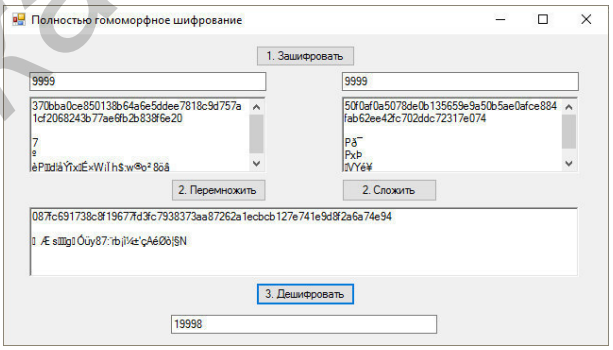

Рис. 1 – Шифрование двух целых чисел и выполнения операций сложения и умножения над ними

IV. Заключение

К сожалению, в настоящее время нет ни одной реализации гомоморфного шифрования, готовой к внедрению в реальные системы. Однако данное направление криптографии является актуальным и интересным с точки зрения защиты информации.

1. Cloud security alliance [Электронный ресурс] / CSA. – Режим доступа: https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf. – Дата доступа:20.04.2016.
2. Homomorphic encryption [Электронный ресурс] / Википедия. – Режим доступа: https://en.wikipedia.org/wiki/Homomorphic_encryption. – Дата доступа: 01.04.2016.
3. Варновский, Н. П. Гомоморфное шифрование / Н. П. Варновский, А. В. Шокуров // Труды Института Системного программирования: Том 12. (под Ред. В. П. Иванникова). – М.:ИСП РАН, 2006, с. 27-36.