РЕАЛИЗАЦИЯ МОДУЛЯ ДЛЯ БЕЗОПАСНЫХ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ГОМОМОРФНОЙ КРИПТОГРАФИИ НА ПЛАТФОРМЕ IOS

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

В докладе представлена реализация программного модуля для безопасных облачных вычислений с использованием гомоморфной криптографии, встраиваемого в приложения на платформе iOS. Модуль, помимо функций шифрования и дешифрования, обеспечивает функцию генерации и распределения ключей. Использование модуля позволит защитить конфиденциальные данные как от злоумышленников, так и от сомнительных либо незаконных действий администраторов или владельцев облачного хранилища.

Введение

Хранение и обработка конфиденциальных данных в облачной инфраструктуре небезопасны. Согласно [1], основные угрозы нарушения конфиденциальности данных в облаке:

1. Доступ к данным со стороны провайдера
2. Публичное разглашение данных
3. Угроза выемки данных или носителей из датацентра провайдера
4. Ошибки изоляции среды
5. Неполное уничтожение данных при уходе клиента или стирании данных.

В то же время, распространенные криптографические алгоритмы не позволяют производить произвольные вычисления над зашифрованными данными, требуя их предварительного дешифрования, что существенно снижает уровень безопасности использования облачных ресурсов.

Возможность выполнять операции над зашифрованными данными обеспечивает полностью гомоморфное шифрование [2].

I. Гомоморфное шифрование

Гомоморфное шифрование – вид шифрования, позволяющий производить определённые математические действия с зашифрованным текстом и получать зашифрованный результат, который соответствует результату операций, выполняемых с открытым текстом [3].

Особый интерес представляет возможность построения полностью гомоморфного шифрования, т.е. шифрования, позволяющего проводить над шифртекстами любые необходимые вычисления [2].

К примеру, если данные являются элементами кольца Z_n, то можно построить такую криптосистему, которая была бы 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 m_1 и m_2 – открытые тексты, k – ключ шифрования, E и D – функции шифрования и дешифрования, op_1 и op_2 – операции над шифртекстами, соответствующие операциям $\cdot$ и $+$ над открытыми текстами.

Если криптосистема с такими свойствами сможет надежно зашифровать два бита, то поскольку над битами операции сложения и умножения формируют полный базис, становится возможным вычислить любую булеву (а следовательно, и вообще любую вычислимую) функцию.

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 облачному сервису

Существующие облачные сервисы в лучшем случае обеспечивают возможность лишь зашифровать данные на стороне облака. При этом данные шифруются ключом, который хранится в том же самом облаке.

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 [4]:

1. Данные клиента должны храниться и поступать на сервер в зашифрованном виде, т.е. шифрование должно проводиться ещё на стороне клиента.
2. Данные должны обрабатыватся на сервере без их предварительного дешифрования. Иначе облачный сервер становится всего лишь безопасным хранилищем, а для каждой операции над данными потребуется пересылать их на сторону клиента.

Библиотека БГУИР

III. Основные требования к программному модулю

Было принято решение спроектировать и разработать программный модуль, встраиваемый в исходный код приложений, который обеспечивал бы возможность шифрования и дешифрования целочисленных и строковых данных.

Модуль может быть использован в роли промежуточного звена между клиентским приложением, оперирующим открытыми данными, и серверной (облачной) средой, оперирующей исключительно зашифрованными данными.

Основные требования к модулю:

1. Использование алгоритмов гомоморфной криптографии
2. Шифрование/дешифрование целочисленных данных.
3. Преобразование строк в валидный для шифрования (целочисленный) формат.
4. Управление ключами шифрования.
5. Множество поддерживаемых математических функций и диапазоны чисел – достаточны для экономических задач.
6. Точность и скорость вычислений не должны деградировать в течение вычислений.
7. Количество доступных ключей должно быть достаточно велико, чтобы исключить атаку полным перебором.

IV. Средства разработки модуля и мобильного приложения

Модуль представляет собой функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом, предназначенный для использования в приложениях для платформы iOS. Для написания исходного кода модуля использован язык программирования Swift, являющийся основным языком разработки нативных приложений для iOS.

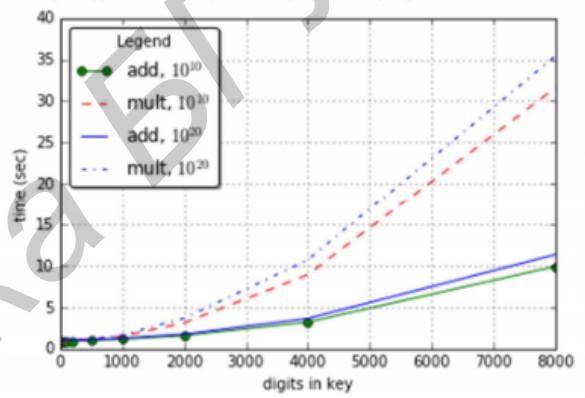
Клиентская часть мобильного приложения для операционной системы iOS, использующая модуль, была также разработана на Swift с учетом известных атак на уязвимости мобильных устройств и приложений. Для разработки клиентской части мобильного приложения, а также непосредственно модуля, была выбрана IDE XCode, предоставляемая компанией Apple, для написания программного обеспечения на языках Swift и Objective-C.

В качестве платформы для разработки серверной части мобильного приложения была выбрана платформа Firebase, разработанная компанией Google. В контексте данной работы Firebase будет рассматриваться в качестве облачной услуги типа BaaS (Backend-as-a-service).

V. Оценка эффективности применения гомоморфного шифрования

Эффективность реализации гомоморфного шифрования была подтверждена (см. рисунок) результатами эксперимента с параметрами:

- уровень защищенности – характеристика, определяющая значение модуля группы и длину используемых ключей шифрования. Использованы модуль и ключи с длиной от 5 до 8000 десятичных цифр;
- типы выполняемых операций – шифрование и дешифрование операндов, умножение и сложение операндов;
- максимальная длина операндов – использованы 10- и 20-значные десятичные числа;
- время работы – оценки получены при 1000-кратном повторении вычислений с заданными параметрами на компьютере класса Intel(R) CORE (TM) i3-2728 CPU @ 2.20 GHz RAM 4 GB.



Заключение

Разработанный модуль является Open Source продуктом. Предусмотрена возможность его использования в качестве библиотеки другими пользователями. Возможна доработка функциональность модуля под личные задачи или под архитектуру существующего приложения.

1. Cloud security alliance [Electronic resource] / CSA. – Mode of access: https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf. – Date of access:20.08.2017.
2. Craig Gentry, Fully homomorphic encryption using ideal lattices, Symposium on the Theory of Computing (STOC), 2009, pp. 169-178.
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.
4. Кадан, М. А. Безопасные вычисления с использованием гомоморфной криптографии для облачных хранилищ данных / М. А. Кадан, М. А. Макарычев // Современные информационные технологии и ИТ-образование. Т. 12 (№ 3), часть 1, 2016. – С.43–49.

Библиотека БГУИР