УДК 004.056

Методы полностью гомоморфного шифрования для СУБД

Неклепаева А.Н neklepaevaan@student.bmstu.ru SPIN-код 1067-1024

Оленев А.А. aolenev@bmstu.ru SPIN-код 4149-5756

Организация - МГТУ им. Н.Э. Баумана

Аннотация: *Гомоморфная криптосистема - комплексная модель, позволяющая проводить вычисления над шифротекстами. Существуют частично и полностью гомоморфные схемы шифрования. Частично гомоморфные схемы обладают свойством гомоморфности относительно только одной операции: сложения или умножения. Частичная гомоморфность достигается в криптосистемах RSA, Эль-Гамаля, Пэйе и других [1]. Полностью гомоморфная схема поддерживает операции и сложения, и умножения над шифротекстами. Идея полностью гомоморфного шифрования возникла в 1978 году у изобретателей алгоритма шифрования с открытым ключом RSA Ривестом и Шамиром совместно с Дертузо. Но попытки создания полностью гомоморфной схемы, проведенные в то время, были не успешны. Только в 2009 году была сформулирована полностью гомоморфная криптосистема Крейгом Джентри. С того момента появилось множество модификаций этого подхода. В данной статье рассматривается полностью гомоморфная криптосистема как один из современных методов защиты персональных данных в облачных платформах. Описываются существующие схемы гомоморфного шифрования, проводится их сравнительный анализ. Осуществляется краткий обзор существующих реализаций полностью гомоморфных криптосистем. Рассматриваются возможности практического применения полностью гомоморфного шифрования (Fully Homomorphic Encryption - FHE).*

Ключевые слова: *гомоморфное шифрование, схемы полностью гомомрфного шифрования, СУБД, защита персональных данных, облачные вычисления, поиск по зашифрованным данным.*

Введение. В последние несколько лет рынок облачных решений растет. Хранение информации переносится в облачные системы, персональные данные доверяются третьим лицам. Каждый день по компьютерным сетям проводятся межбанковские расчеты, переводы заработных плат, покупки в Интернет-магазинах, также сейчас большую ценность представляют из себя медицинские данные, история поиска в различных поисковых системах. Вся эта информация должна быть защищена во избежание последствий утечек персональных данных. Поэтому одной из самых

важных задач современной криптографии является разработка метода проведения вычислений над зашифрованными данными без предварительной расшифровки. Современные СУБД используют механизм индексации для увеличения скорости доступа к данным. Индексация-это способ сортировки нескольких записей по нескольким полям. Для применения этого механизма над зашифрованным значениям требуется криптосистема, поддерживающая операции сравнения. Гомоморфное шифрование - криптографическая система, позволяющая для некоторого рода функции *f(x)* производить операции над зашифрованными данными так, что при расшифровке результат будет равен результату операции над незашифрованными данными. Существуют частично и полностью гомоморфные схемы шифрования. Полностью гомоморфная схема обладает свойством гомоморфности относительно операций сложения и умножения. В настоящий момент ведется активная разработка технологий, использующих гомоморфную криптографию. Однако, до сих пор полностью гомоморфное шифрование не было широко применено на практике.

Первая схема полностью гомоморфного шифрования:схема Крейга Джентри.

Гомоморфное шифрование - форма шифрования, позволяющая производить определённые математические операции с зашифрованным текстом и получать зашифрованный результат, который соответствует результату операций, выполненных с открытым текстом.

Система шифрования является полностью гомоморфной, если она гомоморфна относительно операций умножения и сложения, т.е. если

$$D(E(m_1) \otimes E(m_2)) = m_1 \cdot m_2$$
$$D(E(m_1) \oplus E(m_2)) = m_1 + m_2,$$

где $\otimes$, $\oplus$ - операции умножения и сложения над шифрованными текстами, соответствующие операциям умножения и сложения над открытыми текстами; D - функция расшифровки; E - функция шифрования.

В 2009 году Крейгом Джентри была описана первая схема полностью гомоморфного шифрования, которая базировалась на идеальных решетках. В основу гомоморфной криптографии легли две задачи из теории решеток: задача обучения с ошибками (LWE) и задача обучения с ошибками в кольце (RLWE). Решетка – дискретная аддитивная подгруппа, заданная на множестве $\mathbb{R}^n$, т. е. решетку L можно представить как множество векторов заданных целочисленными линейно независимыми базисными векторами $B = \{\overline{b_1}, \ldots, \overline{b_n}\} \in \mathbb{R}^n$, определенными по модулю некоторого целого числа $x \in \mathbb{Z}^n$. Шифртекст в схеме Джентри содержит компонент случайного «шума»,

уровень которого растет при осуществлении вычислений над шифртекстами (при операции сложения уровень шума растет линейно, при операции умножения - квадратично). Уровень шума может превысить допустимое значение, что приводит к некорректному результату при расшифровке. Крейг Джентри предложил процедуру повторного шифрования, при которой вычисляется новый шифртекст, содержащий значение шума меньше, чем в исходном шифртексте. Данная операция (bootstrapping) позволяет снизить шум в случае, когда он превышает допустимый порог, и осуществить произвольное количество вычислений над зашифрованными данными [2]. Однако такая «очистка» шифртекста требует высоких вычислительных затрат, что делает данную схему непригодной для практического применения. Криптосистема Джентри послужила опорной точкой в создании новых схем полностью гомоморфного шифрования, в которых использование процедуры «очистки» сводится к минимуму или же заменяется на другие подходы.

Схема BGV (Бракерски-Джентри-Вайкутанатан). Обозначим кольцо вычетов по модулю q как $\mathbb{Z}_q$. $[z]_q$ - элемент кольца $\mathbb{Z}_q$, причем$z \in \mathbb{Z}$ и $[z]_q$ лежит в интервале $\mathbb{Z} \cap (-\frac{q}{2}, \frac{q}{2}]$. Пусть $R = \mathbb{Z}[x]/(x^d + 1)$ - фактор-кольцо многочленов, где d - степень 2. Коэффициенты многочленов и все операции над ними будут выполняться в конечном поле, определяемом как $R_q = R/qR$. Тогда пусть множество открытых текстов - это кольцо R_p, множество шифртекстов R_q, где p и q - простые числа, связанные соотношением $q \equiv 1 \ mod \ p$. Алгоритмы, составляющие полностью гомоморфную схему шифрования BGV описаны ниже:

SecretKeyGen: В соответствии с распределением χ выбирается s'. Тогда $sk = \mathbf{s} = (1, s')$ секретный ключ, где $\mathbf{s}[0] = 1$.

PublicKeyGen: Из R_q выбирается полином (вектор) **a**. В соответствии с распределением χ выбирается ошибка e. Строится вектор **b**:

$$\mathbf{b} \leftarrow \mathbf{a}s' + pe$$

Открытый ключ pk представляет собой вектор $\mathbf{k} \leftarrow (a, b)$.

Encryption: Для шифрования открытого текста m вычисляется вектор $\mathbf{m} \leftarrow (m, 0)$, из R_q выбирается элемент **r**, выбирается вектор $\mathbf{e} \leftarrow \chi^2$. Тогда шифртекст вычисляется следующим образом:

$$\mathbf{c} \leftarrow \mathbf{m} + \mathbf{kr} + p\mathbf{e}$$

Decryption: Формула расшифровки имеет следующий вид:

$$m \leftarrow [[<\mathbf{c}|\mathbf{s}>]_q]_p ,$$

где $<\mathbf{c}|\mathbf{s}>$ - скалярное произведение векторов **c** и **s**.

При умножении шифртекстов увеличивается длина ключа. Для предотвращения превышения допустимой длины ключа используется алгоритм *KeySwitch* [3, 6]. В нем осуществляется замена ключей на более короткие, чем исходные. Как и в схеме Джентри, шум в шифртекте растет, что в дальнейшем может привести к некорректной расшифровке данных. Для снижения шума был предложен метод смены модуля (*ModulusSwitch*), согласно которому модуль q заменяется на модуль q', при этом шифртекст **c** также заменяется на **c'**, такая замена осуществляется при выполнении равенства $[[<\mathbf{c}|\mathbf{s}>]_q]_p = [[<\mathbf{c'}|\mathbf{s}>]_{q'}]_p$.

Схема BFV (Бракерски/Фан-Веркаутерен). Схема BFV схожа со схемой BGV. Основным различием между двумя схемами служит умножение на $[q/p]$ на этапе шифрования в схеме BFV (на этапе расшифровки умножение на p/q). Благодаря этому в схеме BFV нет необходимости в использовании процедуры смены модуля (*ModulusSwitch*), так как умножение на $[q/p]$ приводит к тому, что шифруемые данные занимают старшие биты шифртекста, а ошибка - младшие [4].

Схема CKKS (Чон-Ким-Ким-Сонг). Схема CKKS значительно отличается от двух предыдущих схем. Схемы BGV и BFV предназначены для работы с целыми числами, в то время как схема CKKS оперирует числами с плавающей точкой. Результаты операций имеют погрешность, что учитывается вместе с шумом в шифртексте [5].

Существующие реализации FHE схем. На данный момент существует несколько программных реализаций полностью гомоморфного шифрования. В этой предметной области работу ведут многие крупные компании и технологические институты. Ниже рассматриваются 3 популярные библиотеки полностью гомоморфного шифрования. Данные библиотеки написаны на языке программирования C++ и являются проектами с открытым исходным кодом.

	BGV	CKKS	BFV	FHEW	CKKS Bootstrapping	TFHE
IBM HElib	+	+	-	-	-	-
SEAL	-	+	+	-	-	-
PALISADE	+	+	+	+	-	+

HElib - криптографическая библиотека с поддержкой полностью гомоморфного шифрования, разработанная компанией IBM. Она включает в себя реализацию

криптосистемы Бракерски-Джентри-Вайкутанатан (BGV), оптимизированную по быстродействию за счёт эффективного использования техники упаковки зашифрованного текста Смарта-Веркаутерена и оптимизаций Джентри-Халеви-Смарта [6].

SEAL - библиотека полностью гомоморфного шифрования от Microsoft Research, в которой реализованы операции сложения и умножения над целыми и вещественными числами [7].

PALISADE - библиотека решётчатой криптографии и гомоморфных схем шифрования, в разработке которой участвовали Массачусетский институт технологий, Калифорнийский университет Сан-Диего и Технологический институт Нью-Джерси [8].

Осуществление поиска по зашифрованным данным. Частным примером гомоморфного шифрования является поиск по ключевым словам в зашифрованных данных. Рассмотрим базу данных, которая представляет из себя хранилище ключ-значение k, v с уникальными значениями ключей k. Пусть k_q - запрос к базе данных, который является зашифрованным значением ключа. Запрос k_q реплицируется, затем он сравнивается с ключами в базе данных k_i согласно алгоритму, основанному на малой теореме Ферма. Результатом этого сравнения является маска *mask*, состоящая из нулей и единиц. Полученная маска умножается на значения v_i в базе данных. Так как умножение значения на 0 приводит к обнулению значения, а умножение значения на 1 приводит к получение самого значения, то в ходе процесса умножения на полученную маску извлекается значение, соответствующее ключу k_q. Так как ключи в базе данных уникальны, это гарантирует, что будет получено максимум одно соответствие ключу k_q. Тогда так как сложение значения с 0 не меняет самого значения, то результат, полученный в ходе умножения на маску *mask* можно объединить в один шифртекст путем сложения.

$$E(0) + E(0) + \ldots + E(v_q) + E(0) = E(v_q),$$

где v_q - значение, соответствующее ключу k_q.

Таким образом клиенту отправляется только один шифртекст, а не множество шифртекстов в соответствии с количеством записей в базе данных [9, 10].

Заключение. Растущий интерес к облачным вычислениям привел к бурной разработке в области полностью гомоморфной криптографии. Некоторые решения в этой сфере все еще носят лишь научно-исследовательский характер из-за низких

эксплуатационных характеристик полностью гомоморфных схем шифрования. К недостаткам гомоморфного шифрования относятся: низкая скорость вычислений над гомоморфно зашифрованными данными по сравнению с работой с незашифрованными данными (вычисления над гомоморфно зашифрованными данными работают в сотни раз медленнее, чем вычисления над незашифрованными данными), нелинейный рост времени выполнения математических операций над гомоморфно зашифрованными данными, увеличение вычислительной сложности из-за процедуры уменьшения уровня шума в шифртексте. Применение полностью гомоморфного шифрования является решением проблемы защиты персональных данных в облачных системах. В силу востребованности гомоморфной криптосистемы в ближайшем будущем будут проводиться дальнейшие оптимизации программных реализаций различных гомоморфных схем. Для повсеместного использования гомоморфного шифрования на облачных платформах ведется работа над аппаратной составляющей. Компания Intel в марте 2021 объявила, что займётся разработкой интегральной схемы, которая упростит обработку зашифрованных данных [11]. Таким образом, полностью гомоморфная криптография может быть широко применена на практике в ближайшем будущем.

Литература

[1] Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges 2019 https://www.researchgate.net/publication/332927799_Privacy-preserving_cloud_computing_on_sensitive_data_A_survey_of_methods_products_and_challenges

[2] Dissertation «A Fully Homomorphic Encryption scheme» Craig Gentry 2009 https://crypto.stanford.edu/craig/craig-thesis.pdf

[3] Fully homomorphic encryption without bootstrapping. Brakerski Zvika, Gentry Craig, Vaikuntanathan Vinod. 2011 https://eprint.iacr.org/2014/873

[4] Fan Junfeng, Vercauteren Frederik. Somewhat practical fully homomorphic encryption. Fan Junfeng, Vercauteren Frederik. 2012 https://eprint.iacr.org/2012/144.pdf

[5] Homomorphic Encryption for Arithmetic of Approximate Numbers. Cheon Jung Hee, Kim Andrey, Kim Miran, Song Yongsoo. 2016 https://eprint.iacr.org/2016/421.pdf

[6] HElib design principles 2020 https://homenc.github.io/HElib/documentation/Design_Document/HElib-design.pdf

[7] Microsoft SEAL https://www.microsoft.com/en-us/research/project/microsoft-seal/

[8] PALISADE Documentation https://palisade-crypto.org/documentation/

[9] Configurable Private Querying: Lookup and Partial Matching under Homomorphic Encryption 2020 https://eprint.iacr.org/2020/964.pdf

[10] Database Lookup https://github.com/homenc/HElib/blob/master/examples/BGV_country_db_lookup/README.md

[11] Intel to Collaborate with Microsoft on DARPA Program https://www.intel.com/content/www/us/en/newsroom/news/intel-collaborate-microsoft-darpa-program.html

Неклепаева Анастасия Николаевна - студент кафедры “Программное обеспечение ЭВМ и информационные технологии”, МГТУ им. Н.Э. Баумана, Москва, Российская федерация.

Оленев Антон Александрович - старший преподаватель кафедры “Программное обеспечение ЭВМ и информационные технологии”, МГТУ им. Н.Э. Баумана, Москва, Российская федерация.