ЗАЩИТА ИНФОРМАЦИИ

УДК 004.056:004.62

А.И. Трубей

ГОМОМОРФНОЕ ШИФРОВАНИЕ: БЕЗОПАСНОСТЬ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ И ДРУГИЕ ПРИЛОЖЕНИЯ (ОБЗОР)

Представляются основные понятия гомоморфного шифрования и различные алгоритмы шифрования в соответствии с основополагающими свойствами гомоморфного шифрования. Приводятся практические примеры различных принципов и свойств гомоморфного шифрования, некоторые алгоритмы гомоморфного шифрования, использующие асимметричные ключевые системы, такие как RSA, Эль-Гамаля, Пэйе, а также различные схемы гомоморфного шифрования. Рассматриваются перспективы применения гомоморфного шифрования в области безопасных облачных вычислений, электронного голосования, поиска в зашифрованном тексте, фильтрации шифрованной почты, мобильного шифрования, защищенных систем с обратной связью.

Введение

Наряду с облегченной криптографией [1] значительный интерес для анализа и практического использования представляет гомоморфная криптография, или гомоморфное шифрование. Гомоморфное шифрование – это форма шифрования, позволяющая осуществлять определенные типы вычислений на зашифрованном тексте и получать зашифрованные результаты вычислений, которые при расшифровании соответствуют результатам операций, выполняемых на открытом тексте. Теоретические и практические аспекты гомоморфного шифрования тесно взаимосвязаны с проблемой обеспечения безопасности облачных вычислений. Идея облачных вычислений появилась еще в 1960 г., когда Джон Маккарти высказал предположение, что когда-нибудь компьютерные вычисления будут производиться с помощью «общенародных утилит». Считается, что идеология облачных вычислений обрела популярность с 2007 г. благодаря быстрому развитию каналов связи и стремительно растущим потребностям пользователей.

Облачные вычисления – одно из наиболее быстро развивающихся направлений в современных информационных технологиях (ИТ), приложения для использования в облачных сервисах являются приоритетными проектами ведущих мировых ИТ-компаний. Этот подход позволяет организовать динамическое предоставление услуг, что дает возможность пользователям производить оплату по факту использования ресурсов и регулировать их объем в зависимости от реальных потребностей без долгосрочных обязательств.

Рост мирового рынка облачных технологий оценивается в 20–25 % в год, в то время как рынок в целом увеличивается на 5–10 %. В России рынок ИТ в целом стагнирует, но динамика выручки в его облачном сегменте, по разным подсчетам, приближается к 40–60 %. Такие высокие темпы обоснованы тем, что рост с нуля всегда велик [2]. В Беларуси облачные серверы пока используются в основном для хостинга сайтов с высокой нагрузкой и обеспечения работы интернет-магазинов. Однако в последние два года наметился устойчивый тренд на размещение в облаке комплексных инфраструктурных проектов. Например, в 2013 г. выручка белорусского офиса группы компаний ActiveCloud, одного из крупнейших на постсоветском пространстве разработчиков ИТ-решений для размещения информационных ресурсов и систем в облаке, по услуге IaaS (инфраструктура как сервис) удвоилась по сравнению с 2012 г. [3].

Национальный институт стандартов и технологий NIST (National Institute of Standards and Technology, USA) определяет следующие характеристики облаков [4]:

– возможность высокоавтоматизированного самообслуживания системы со стороны провайдера;

– наличие системы Broad Network Access;

– сосредоточенность ресурсов на отдельных площадках для эффективного распределения;

– быструю масштабируемость (ресурсы могут неограниченно выделяться и высвобождаться с большой скоростью в зависимости от потребностей);

– наличие управляемого сервиса (система управления облаком автоматически контролирует и оптимизирует выделение ресурсов).

Таким образом, облачные вычисления – это способ обеспечения удобного сетевого доступа к разделяемому пулу реконфигурируемых вычислительных ресурсов (например, к сетям, серверам, устройствам памяти, приложениям и сервисам), которые могут быть быстро подобраны и предоставлены с минимальными усилиями для взаимодействия с поставщиком услуг.

Крупные вычислительные облака состоят из тысяч серверов, размещенных в центрах обработки данных. Они обеспечивают ресурсами десятки тысяч приложений, которые одновременно используют миллионы пользователей. Облачные технологии являются удобным инструментом для предприятий, которым слишком дорого содержать собственные ERP, CRM или другие серверы, требующие приобретения и настройки дополнительного оборудования. По модели развертывания облака разделяют на частные, общедоступные и гибридные [5].

Частные (корпоративные) облака – это внутренние облачные инфраструктура и службы организации. Все данные и приложения пользователя остаются внутри организации. Недостатком является то, что только крупные организации могут себе позволить создание частного вычислительного облака, поскольку его инфраструктура может быть достаточно дорогой и требовать высококвалифицированных администраторов.

Общедоступные (публичные) облака – это облачные сервисы, предоставляемые поставщиком. Пользователи данных облаков не имеют возможности управлять облаком или обслуживать его, вся ответственность возложена на владельца облака. Поставщик облачных услуг принимает на себя обязанности по установке, управлению, предоставлению и обслуживанию программного обеспечения, инфраструктуры приложений или физической инфраструктуры.

Гибридные облака представляют собой такое внедрение облачных вычислений, при котором часть системы размещается в публичном облаке, т. е. на базе центров данных облачного провайдера, часть – в частном облаке, т. е. на серверах самой организации. По сути, гибридное облако не является самостоятельным типом облачных вычислений, а лишь указывает на тесную интеграцию публичных и частных облачных систем.

Причины возрастающей популярности облачных технологий очевидны, возможности их применения очень разнообразны. Пользователь экономит как на обслуживании и персонале, так и на инфраструктуре. Нет необходимости приобретать лицензии на программное обеспечение, организацию и обслуживание собственных серверов, нанимать опытных администраторов и т. д. Все эти проблемы перекладываются на провайдера услуг. Кроме того, данный подход позволяет стандартизировать программное обеспечение, даже если на компьютерах предприятия установлены разные операционные системы (Windows, Linux, MacOS и т. д.).

Проблемы защиты информации в облачных технологиях стали активно анализироваться достаточно поздно, когда облака были уже фактически реализованной технологией. Практика применения облачных вычислений показала, что для защиты информации недостаточно уже имеющихся криптографических средств. Поясним это на следующем примере. Предположим, что в облаке S содержится множество пользователей (клиентов) $p_1, ..., p_i, ..., p_l$. У пользователя p_i имеются конфиденциальные данные x_i, хранящиеся в облаке. Такая облачная услуга называется Storage aaS (хранилище как сервис). Пользователь p_i может обратиться к облаку с запросом на вычисление значения некоторой функции F, зависящей от конфиденциальных данных. Запрос должен состоять из описания функции F, идентификатора пользователя и его открытого ключа pk_i. Облако должно проверить полномочия пользователя p_i на вычисление $F(x_i)$. Такая проверка может быть реализована с помощью стандартной процедуры электронной цифровой подписи (ЭЦП). Если пользователь подтвердил свои права на вычисление функции F, то облако должно вычислить значение $E(pk_i, F(x_i))$ и отправить его пользователю. В качестве E можно взять функции шифрования некоторой криптосистемы с открытым ключом.

Пользователь, который размещает в хранилище свои конфиденциальные данные и дает запрос на вычисление функции F, не доверяет облаку и должен принимать соответствующие меры и предъявлять требования по обеспечению их безопасности. Очевидно, что было бы гораздо безопаснее передавать данные в таком виде, чтобы во время операций, которые производятся над ними, никоим образом не распространялась информация об этих данных. Поэтому, во-первых, данные необходимо шифровать, причем они должны поступать на сервер уже в шифрованном виде. Это означает, что шифрование должно осуществляться еще пользователем. Во-вторых, необходимо обрабатывать эти данные без расшифровки, так как для передачи и хранения секретного ключа необходимо соблюдение определенных процедур, особенно сложных, если информация обрабатывается в недоверенной среде.

Оказалось, что защита информации в облачных вычислениях намного сложнее тех задач защиты информации, которые решаются известными криптографическими средствами. Криптосистемы с открытым ключом для решения данной проблемы не всегда подходят. В 1978 г. авторы известного алгоритма с открытым ключом RSA Майкл Дертузос, Рональд Риверст и Леонард Адлеман впервые обосновали, что методом, позволяющим успешно проводить операции над зашифрованными данными, не искажая и не расшифровывая их, является так называемое гомоморфное шифрование [6]. В своей работе они описали концепцию гомоморфного шифрования, а также задались вопросами, возможно ли такое шифрование в принципе и для каких алгебраических систем такой гомоморфизм существует.

В статье на основании изучения большого фактического материала приводятся основные понятия и определения в области гомоморфного шифрования, теоретические и практические проблемы по разработке данных систем шифрования, структура, свойства и операции соответствующих криптоалгоритмов. Осуществлен также краткий обзор существующих и перспективных систем гомоморфного шифрования и их практического применения.

1. Теоретические основы гомоморфного шифрования

Гомоморфное шифрование является формой шифрования, позволяющей осуществить определенную алгебраическую операцию над открытым текстом посредством выполнения алгебраической операции над зашифрованным текстом. Пусть $E(k, m)$ – функция шифрования, где k – ключ шифрования, а m – открытый текст. Функция E называется гомоморфной относительно операции * над открытыми текстами, если существует эффективный алгоритм M (требующий полиномиального числа ресурсов и работающий за полиномиальное время), который, получив на вход любую пару шифрованных текстов вида $E(k, m_1)$, $E(k, m_2)$, выдает шифрованный текст $c = M(E(k,m_1), E(k,m_2))$, такой, что при расшифровании c будет получен открытый текст $m_1 * m_2$ [7].

Как правило, рассматривается следующий частный случай гомоморфного шифрования. Для данной функции шифрования E и операции $*_1$ над открытыми текстами существует операция $*_2$ над шифрованными текстами, такая, что из шифрованного текста $E(k,m_1) *_2 E(k,m_2)$ при расшифровании извлекается открытый текст $m_1 *_1 m_2$. При этом требуется, чтобы по заданным c, $E(k, m_1)$, $E(k, m_2)$, но при неизвестном ключе было бы невозможно эффективно проверить, что шифрованный текст c получен из $E(k, m_1)$ и $E(k, m_2)$.

Любую стандартную систему шифрования можно описать в виде трех операций: генерации ключей (KeyGen), шифрования (Encypt) и расшифрования (Decrypt).

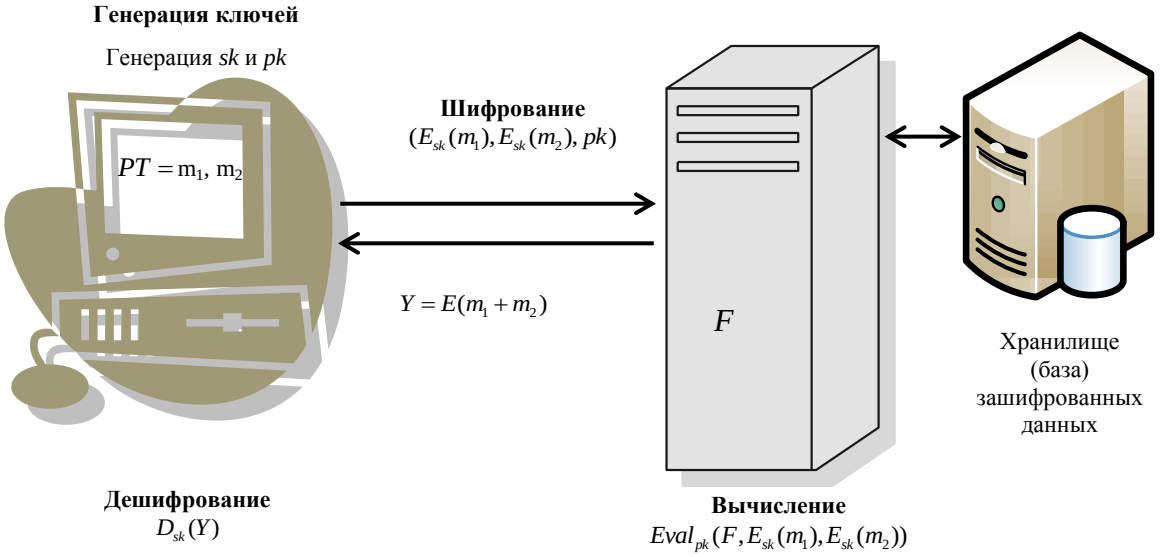
Гомоморфная система шифрования, кроме трех перечисленных выше операций, включает в себя операцию вычислений (Eval). Таким образом, гомоморфное шифрование является последовательностью из четырех операций: генерации ключей, шифрования, вычисления, расшифрования (рисунок) [8]:

– генерация ключей: клиент генерирует открытый ключ pk (public key) и секретный ключ sk (secret key) для шифрования открытого текста;

– шифрование: используя секретный ключ sk, клиент шифрует открытый текст PT (plain text), создает $E_{sk}(PT)$ и вместе с открытым ключом pk отправляет шифрованный текст CT (cipher text) на сервер;

– вычисление: сервер получает функцию F для проведения вычислений над шифрованным текстом CT и выполняет их в соответствии с требованиями данной функции, используя pk;

– расшифрование: для получения искомого результата значение $Eval(F(PT))$, полученное в ходе вычислений, расшифровывается клиентом с использованием своего секретного ключа sk.



Результат $D_{sk}(Eval_{pk}(F, E_{sk}(m_1), E_{sk}(m_2)))$

Операции гомоморфного шифрования

Система шифрования является гомоморфной относительно операции умножения, если

$$D(E(m_1) \otimes E(m_2)) = m_1 \cdot m_2.$$

Система шифрования является гомоморфной относительно операции сложения, если

$$D(E(m_1) \oplus E(m_2)) = m_1 + m_2.$$

Система шифрования является гомоморфной относительно операций умножения и сложения, т. е. полностью гомоморфной, 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 расшифрования; E – функция шифрования.

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

Свыше 30 лет оставалась нерешенной задача полностью гомоморфного шифрования − создания системы, гомоморфной относительно операций сложения и умножения одновременно. Только в 2009 г. аспирант Стэндфордского университета и стажер IBM Крейг Джентри теоретически обосновал принципиальную возможность создания такой системы шифрования. В схеме Джентри [9] выполняются свойства гомоморфизма как относительно умножения, так и сложения, т. е. она является алгебраической гомоморфной системой. Предложенная сис-

тема может использоваться для обеспечения конфиденциальности данных при любых видах их обработки в недоверенной среде, например при облачных или распределенных вычислениях. Однако модель Джентри оказалась слишком непрактичной. С увеличением количества операций, производимых над зашифрованным текстом, сложность и размер шифрованного текста увеличиваются с невероятной скоростью. Несмотря на то что за последние годы было проведено множество улучшений данной схемы, она все еще остается скорее теоретической моделью, которая пока не применима на практике.

2. Краткий обзор систем гомоморфного шифрования

В настоящем разделе на примерах конкретных алгоритмов и схем описываются мультипликативные, аддитивные [10, 11] и смешанные [9, 12] свойства гомоморфного шифрования. Приведенные алгоритмы и схемы являются общедоступными, поэтому дадим детальное описание только некоторых из них, в отношении других ограничимся кратким перечислением их основных свойств и областей применения.

Криптосистема RSA. Метод шифрования RSA (аббревиатура от фамилий создателей – Rivest, Shamir, Adleman) предложен в 1977 г. как реализация идеи основоположников криптографии с открытым ключом Диффи и Хеллмана.

Предположим, что открытый текст представлен числом m, таким, что $0 < m < N$. Пользователь B желает, чтобы ему передали секретное сообщение. Для этого он делает общедоступными два числа N (составной модуль) и e (открытый ключ), которые удовлетворяют следующим условиям: $N = pq$, где p, q – большие простые числа, которые B держит в секрете; $p, q \geq 2^{256}$; e выбирается взаимно простым с $\varphi(N) = (p-1)(q-1)$.

Пользователь A, отправивший сообщение m, шифрует его следующим образом:

$$E(m) = m^e \pmod N.$$

Это и есть шифрованный текст, который получает B.

Для расшифрования B находит число d, такое, что $1 \leq d \leq N-1$ и $ed = 1(\mathrm{mod}\,\varphi(N))$. Данное сравнение разрешимо единственным образом, так как $(e, \varphi(N)) = 1$. Для решения уравнения $ed = 1(\mathrm{mod}\,\varphi(N))$ пользователь B должен вычислить $\varphi(N)$, что для него не составляет труда, так как $\varphi(N) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$. Любой другой пользователь, который знает только N, вынужден находить p и q, т. е. разлагать число N на простые сомножители, а эта задача при больших p и q имеет значительную вычислительную сложность.

Далее, имея в распоряжении $y = E(m) = m^e \pmod N$, пользователь B вычисляет величину $D(y) = y^d \pmod N$, которая является открытым текстом m. Действительно, применяя теорему Эйлера, получаем

$$D(y) = y^d = m^{ed} = m^{\varphi(N)k+1} = (m^{\varphi(N)})^k m = m(\mathrm{mod}\,N).$$

Криптосистема RSA гомоморфна относительно операции умножения открытых текстов. Для любых двух открытых текстов m_1, m_2

$$E(m_1)E(m_2) = m_1^e m_2^e \pmod N = E(m_1 m_2).$$

Криптосистема Эль-Гамаля. Пусть G – циклическая группа порядка p и g – порождающий элемент группы. В качестве секретного ключа выбирается случайный элемент d группы Z_{p-1}. Соответствующий открытый ключ e вычисляется по формуле $e = g^d$.

Функция шифрования для сообщения m выглядит следующим образом:

$$E(e, m) = (e^r m, g^r),$$

где r – случайный элемент группы Z_{p-1}.

Расшифрование криптограммы (c_1, c_2) выполняется следующим образом. Вычисляется

$$c_2^d = g^{rd},$$

откуда

$$m = c_1 / c_2.$$

Криптосистема Эль-Гамаля гомоморфна относительно операции умножения открытых текстов. Если $E(e, m_1) = (e^{r_1} m_1, g^{r_1})$ и $E(e, m_2) = (e^{r_2} m_2, g^{r_2})$, то

$$E(e, m_1 m_2) = (e^{r_1} e^{r_2} m_1 m_2, g^{r_1} g^{r_2}) = E(e, m_1) E(e, m_2).$$

Криптосистема Пэйе. Криптосистема базируется на алгоритме вероятностного асимметричного преобразования и применяется в криптографических протоколах с отрытым ключом.

Пусть p и q – два простых числа, $n = pq$, $\lambda = \text{НОК}(p-1, q-1)$. Выберем случайное число g из $Z_{n^2}^*$ и вычислим $\mu = (L(g^\lambda \bmod n^2))^{-1} (\bmod n)$, где $L(u) = (u-1)/u$.

Открытым ключом является пара (n, g), а закрытым ключом – пара (λ, μ).

Для шифрования открытого текста $m \in Z_n$ выбирается случайное число $r \in Z_{n^2}^*$ и вычисляется $E(m) = c = g^m r^n (\bmod n^2)$.

Расшифрование выполняется по формуле $m = L(c^\lambda (\bmod n^2))\mu (\bmod n)$.

Свойство гомоморфизма выглядит следующим образом:

$$E(m_1) E(m_2) = (\mathrm{g}^{m_1} r_1^n)(\mathrm{g}^{m_2} r_2^n) = \mathrm{g}^{m_1 + m_2} (r_1 r_2)^n = E(m_1 + m_2) (\bmod n).$$

Схема Джентри полностью гомоморфного шифрования. Рассмотрим предложенную схему Джентри на примере вычислений в Z_2.

1. Генерация ключей. Выбирается произвольное нечетное целое число $p = 2k+1$. Данное число p является секретным ключом.

2. Шифрование. Пусть требуется зашифровать бит $m \in (0,1)$. Для этого сгенерируем число $z = 2r + m$, где r – произвольное целое число. Это означает, что $z = m (\bmod 2)$.

Шифрование заключается в том, что всякому числу m ставится в соответствие число $c = pq + z$, где q – произвольное целое число. Следовательно, $E(m) = c = 2r + m + (2k+1)q = 2(r + kq) + m + q$.

Вычислениям подвергается именно это число c.

3. Расшифрование. Пусть даны числа c, p, q, где p, q известны. Проведем расшифрование с помощью секретного ключа p:

$$c (\bmod p) = (z + pq) (\bmod p) = z (\bmod p) + pq (\bmod p) = z (\bmod p) =$$

$$= (2r + m) (\bmod p) = 2(r (\bmod p)) + m (\bmod p).$$

Далее вычисляем

$$(c (\bmod p)) (\bmod 2) = (2(r (\bmod p)) (\bmod 2) = m (\bmod 2) = m.$$

Шифрование является гомоморфным относительно операций сложения и умножения. Рассмотрим пару битов $m_1, m_2 \in (0,1)$. Сопоставим им $z_1 = 2r_1 + m_1$, $z_2 = 2r_2 + m_2$. Выберем сек-

ретный ключ $p = 2k + 1$. Тогда $E(m_1) = c_1 = z_1 + pq_1$, $E(m_2) = c_2 = z_2 + pq_2$ – шифрованные тексты для m_1 и m_2 соответственно.

Операция сложения над зашифрованными числами будет иметь вид

$$E(m_1) + E(m_2) = c_1 + c_2 = z_1 + z_2 + p(q_1 + q_2) = 2(r_1 + r_2) + m_1 + m_2 + p(q_1 + q_2).$$

Операция умножения над зашифрованными числами будет иметь вид

$$E(m_1)E(m_2) = c_1c_2 = z_1z_2 + p(z_1q_2 + z_2q_1) + p^2q_1q_2 =$$
$$= (2r_1 + m_1)(2r_2 + m_2) + \mathrm{p}(\mathrm{z}_1 q_2 + z_2 q_1) + p^2q_1q_2 =$$
$$= 4r_1r_2 + 2(r_1m_2 + r_2m_1) + m_1m_2 + p(z_1q_2 + z_2q_1) + p^2q_1q_2.$$

При расшифровании

$$D(E(m_1) + E(m_2)) = ((c_1 + c_2)(\mathrm{mod}\, p))(\mathrm{mod}\, 2) = m_1 + m_2;$$

$$D(E(m_1)E(m_2)) = ((c_1c_2)(\mathrm{mod}\, p))(\mathrm{mod}\, 2) = m_1m_2.$$

Существенным недостатком данной схемы является то, что выполнение вычислений приводит к накоплению ошибки и, после того как она превышает *p*, расшифровать сообщение становится невозможным. Одним из вариантов решения данной проблемы является перешифровка данных после некоторого количества операций, однако такой вариант снижает производительность вычислений и требует постоянного доступа к секретному ключу. Стойкость схемы Джентри на основе идеальных решеток (решеток со свойствами идеала на некотором кольце чисел) сводится к *NP*-полной задаче нахождения кратчайшего вектора. Появилось немало работ, направленных на развитие предложенных в ней идей и устранение недостатков. В частности, была предложена схема BGV (аббревиатура от фамилий создателей – Brakerski, Gentry, Vaikuntanathan). Авторы представили альтернативный вариант полностью гомоморфного шифрования на основании LWE (Learning With Errors) [13], который позволил уменьшить сложность построения криптосистемы, однако унаследовал основные недостатки схемы Джентри:

– наличие возрастающей ошибки в шифрованном тексте;

– рост размера шифрованного текста.

В зависимости от обстоятельств свойство гомоморфизма может рассматриваться как в качестве достоинства, так и в качестве недостатка криптосистемы. Это относится, например, к криптосистеме RSA, в которой функция расшифрования используется в схеме ЭЦП. Подпись сообщения *m* вычисляется по формуле $s = m^d (\mathrm{mod}\, N)$, где d – секретный компонент ключа. Очевидно, что и это преобразование гомоморфно относительно операции умножения. Следовательно, можно предложить следующий способ подделки подписей. Если известны подписи s_1, s_2 сообщений m_1, m_2, то ЭЦП сообщения m_1m_2 будет являться соответственно s_1s_2. Однако на практике такая уязвимость не представляет угрозы стойкости схемы ЭЦП, так как подписываются не сами сообщения, а значения хэш-функций сообщений. Тем не менее гомоморфизм функции генерации подписей накладывает на хэш-функцию дополнительное требование, которое, вообще говоря, не следует из стандартных определений криптографической хэш-функции. Основные параметры гомоморфных систем шифрования приведены в таблице [8, 14–17].

Большинство алгоритмов гомоморфного шифрования, стойкость которых базируется на сложности дискретного логарифмирования в конечном поле, достаточно легко переносятся на случай эллиптических кривых. Криптосистемы на основе эллиптических кривых превосходят другие системы с открытым ключом по двум важным параметрам: степени защищенности в расчете на каждый бит ключа и быстродействию при программной реализации. Это объясняется тем, что для вычисления обратных функций на эллиптических кривых известны только алгоритмы с экспоненциальным ростом трудоемкости, тогда как для обычных систем предложены также субэкспоненциальные методы. При обеспечении одной и той же стойкости криптографических протоколов вычисления в группе точек эллиптической кривой выполняются примерно

на 20 % быстрее, чем для групп конечного поля [18]. В работе [16] приводятся сравнительные характеристики быстродействия и стойкости алгоритмов на эллиптических кривых.

Сравнительные характеристики некоторых гомоморфных систем шифрования

Система (схема)	Год	Гомоморфизм			Применение
		по сложению	по умножению	полный	
RSA	1977	–	✓	–	Обеспечение безопасности Интернета, банковских транзакций, транзакций по кредитным картам
Гольдвассер – Микали (Goldwasser – Micali)	1982	–	✓	–	Первая вероятностная криптосистема с открытым ключом. Премия Тьюринга за 2012 г.
Эль-Гамаль (ElGamal)	1985	–	✓	–	Гибридные облачные системы. Модификацией данной схемы является схема Шнорра, которая положена в основу СТБ 34.101.45–2013
Бенало (Benaloh)	1988	✓	–	–	Системы электронного голосования
Накаче – Штерн (Naccache Stern)	1998	✓	–	–	Является усовершенствованием схемы Бенало
Окамото – Очияма **(**Okamoto – Uchiyama)	1998	✓	–	–	При проектировании операционной системы EPOC (Electronic Piece Of Cheese*)*
Пэйе (Paillier)	1999	✓	–	–	Системы электронного голосования, пороговые схемы
Дамгард – Джарик (Damgard – Jurik)	2001	✓	–	–	Обобщение схемы Пэйе для больших модулей с целью расширения области применения
Бракерски – Джентри – Вайкунтанатан (Brakerski – Gentry – Vaikuntanathan, BGV)	2012	–	–	✓	Обеспечение безопасности целочисленных многочленов
Неитерактивная экспоненциальная гомоморфная система шифрования (Non-interactive Exponential Homomorphic Encryption Scheme, NEHE)	2012	–	–	✓	Активные сети, электронная коммерция на основе мобильных агентов, грид-вычисления
Алгебраическая гомоморфная система на базе модифицированной схемы Эль-Гамаля (Algebra Homomorphic Encryption Scheme Based On Updated ElGamal, AHEE)	2012	–	–	✓	Протокол конфиденциальных вычислений, электронное голосование и мобильные шифры
Усовершенствованная криптосистема Горти (Gorti's Enhanced Homomorphic Cryptosystem, EHC)	2013	–	–	✓	Обеспечение безопасности передачи сообщений в беспроводных децентрализованных самоорганизующихся сетях (MANET)

На основании вышеизложенного можно утверждать, что для применения гомоморфного шифрования на практике разработанные криптосистемы должны удовлетворять, по крайней мере, следующим требованиям:

– набор поддерживаемых математических функций должен покрывать повседневные нужды программистов;

– точность и скорость вычислений не должны деградировать в течение вычислений;

– стойкость алгоритма должна исключить атаку полным перебором.

3. Области применения гомоморфного шифрования

Применение гомоморфного шифрования может представлять значительный интерес в следующих областях:

Облачные вычисления. Как уже было отмечено выше, гомоморфное шифрование открывает новые возможности по сохранению целостности, доступности и конфиденциальности данных при их обработке в облачных системах. В облачных вычислениях, где производительность является главным приоритетом, следует применять разные алгоритмы, каждый из которых лучше всего справляется с поставленной задачей. Например, для операций умножения зашифрованных данных целесообразно использовать алгоритмы RSA или Эль-Гамаля, а для сложения – Пэйе. Возможно применение комбинированных систем данных алгоритмов. Для операций сравнения и сортировки необходимо использовать другие схемы.

Для практического применения полностью гомоморфной системы шифрования следует ограничивать количество операций, которые можно производить над данными без риска выйти за границу критических пределов ошибки вычислений. При этом предпочтительным является использование гомоморфного шифрования в гибридных облачных системах, так как вычисления, которые не могут быть вынесены в публичное облако в силу законодательных или иных ограничений, могут производиться во внутренней сети.

Электронное голосование. Электронное голосование – еще одна перспективная сфера применения гомоморфного шифрования. Система сможет зашифровать голоса избирателей и провести расчеты над зашифрованными данными, сохраняя анонимность избирателей. Например, в схеме электронного голосования Бенало процесс голосования включает следующие этапы [19]:

– каждый голосующий участник схемы разделяет свой голос (секрет) на составляющие (частичные секреты) по соответствующей ему схеме разделения секрета со свойством гомоморфности по сложению и посылает частичные секреты выборным представителям;

– представители складывают полученные голоса; по свойству гомоморфности (по сложению) суммы голосов являются частичными секретами соответствующего итога выборов, а значит, суммы голосов могут быть вычислены без нарушения конфиденциальности схемы;

– главное доверительное лицо вычисляет конечный итог голосования, используя набор частичных сумм голосов, переданный ему выборными представителями.

Предположим, поставлена задача выбора лучших сотрудников Объединенного института проблем информатики НАН Беларуси. Имеется набор из n кандидатов, из которых формируется список, включаемый в бюллетень. Дирекция, которая обладает криптосистемой, гомоморфной относительно операции сложения, распространяет среди участников тайного голосования бюллетень как вектор $(p_1, ..., p_i, ..., p_n)$, где p_i – фамилия i-го кандидата. Кроме того, голосующим передается открытый ключ системы гомоморфного шифрования pk. Каждый из избирателей составляет вектор предпочтений $(v_1, ..., v_i, ..., v_n)$, где $v_i \in 0,1$. После этого с помощью открытого ключа pk он поэлементно шифрует вектор и отправляет представителю дирекции. Для подведения итогов голосования тот производит вычисления над соответствующими элементами полученных векторов предпочтений и производит расшифрование с помощью секретного ключа sk. Так как криптосистема гомоморфна относительно операции сложения, индексы наибольших элементов результирующего вектора и будут индексами победивших кандидатов. В качестве системы гомоморфного шифрования при тайном голосовании, включая тайное голосование с весами, может использоваться криптосистема Пэйе.

Защищенный поиск информации. Гомоморфное шифрование может предоставить пользователям возможность извлечения информации из поисковых систем с сохранением конфиденциальности: сервисы смогут получать и обрабатывать запросы, а также выдавать результаты обработки, не анализируя и не фиксируя их реальное содержание. Например, метод извлечения записей из базы данных по их индексам можно представить следующим образом.

Пусть $v_1, v_2, ..., v_j, ..., v_n; v_j \in 0,1$ – индекс записи, которую нужно извлечь; $c_1, c_2, ..., c_i, ..., c_{2^n}$ – все проиндексированные записи из базы данных.

Тогда, для того чтобы выбрать требуемую запись, необходимо вычислить следующую функцию F:

$$F(v_1, v_2, ..., v_j, ..., v_n; c_1, c_2, ..., c_i, ..., c_{2^n}) =$$

$$= c_1 \cdot ((v_1 \oplus 1) \otimes (v_2 \oplus 1) \otimes ... \otimes (v_n \oplus 1)) +$$

$$+ c_2 \cdot ((v_1 \oplus 1) \otimes (v_2 \oplus 1) \otimes ... \otimes (v_{n-1} \oplus 1) \otimes v_n) +$$

$$+ c_3 \cdot ((v_1 \oplus 1) \otimes (v_2 \oplus 1) \otimes ... \otimes v_{n-1} \otimes (v_n \oplus 1)) + ... +$$

$$+ c_{2^n} \cdot (v_1 \otimes v_2 \otimes ... \otimes v_n).$$

Если все c_i зашифрованы с помощью гомоморфной криптосистемы, F можно вычислить гомоморфно над шифрованными текстами. Для этого клиенту достаточно побитно зашифровать индекс $v_1, v_2, ..., v_j, ..., v_n$ нужной ему записи и отправить на сервер. Результат гомоморфного вычисления функции F над шифрованными текстами будет искомым шифрованным значением записи c_i, запрашиваемой клиентом. Очевидно, что криптосистема должна обладать как мультипликативными, так и аддитивными гомоморфными свойствами.

Защита беспроводных децентрализованных сетей связи. Беспроводные децентрализованные самоорганизующиеся сети (MANET) – это сети, состоящие из мобильных устройств. Каждое такое устройство может независимо передвигаться в любых направлениях и, как следствие, часто разрывать и устанавливать соединения с соседями. Одной из основных проблем при построении MANET является обеспечение безопасности передаваемых данных. Для решения этой проблемы может применяться гомоморфное шифрование [8], которое встраивается в протоколы маршрутизации для повышения безопасности. В этом случае операции над шифрованными текстами могут безопасно выполняться промежуточными узлами. В частности, для нахождения оптимального пути между двумя узлами необходимо осуществлять линейные операции над зашифрованными данными без их расшифрования. Наличие гомоморфного шифрования не позволяет злоумышленнику найти связь между сообщениями, входящими в узел и выходящими из узла. Поэтому невозможно отследить путь передачи сообщения с помощью анализа трафика [20].

Аутсорсинговые услуги для смарт-карт. В настоящее время наблюдается тенденция к разработке универсальных карт с собственной операционной системой, которая может выполнять разнообразные функции и взаимодействовать с несколькими поставщиками услуг. Высказываются предположения, что некоторые приложения могут работать вне карты на гомоморфно зашифрованных данных. Особо ресурсоемкие приложения, например приложения сервис-провайдеров, а также биометрические проверки (распознавание голоса, отпечатков пальцев или почерка), которым, как правило, требуется значительный объем хранения и большое количество сравнительно простых операций, могут использовать внешние устройства хранения и внешние процессоры, более мощные, чем на карте.

Системы с обратной связью. Гомоморфное шифрование может использоваться, например, в так называемых безопасных гомоморфных системах с обратной связью (secure feedback system) [14], когда необходимо сохранить анонимность пользователя и скрыть промежуточные результаты вычислений. Системы помогают осуществлять анонимный сбор отзывов (комментариев) студентов либо преподавателей об их работе. Полученные таким образом отзывы шифруются и сохраняются для последующих вычислений. Системы с обратной связью могут быть использованы для повышения осведомленности о состоянии дел и улучшения показателей работы.

Установлено, что достоверная обратная связь любой системы или процесса может быть обеспечена только в случаях сохранения анонимности пользователя, неизменности данных, собранных в процессе обратной связи, обеспечения безопасности внутренних операций для анализа данных.

Обфускация для защиты программных продуктов. Впервые о применении обфускации в криптографии было упомянуто в работе Диффи и Хеллмана [21]. В ней предложено использовать для построения асимметричной криптосистемы сложность задачи, заключающейся в анализе программ на низкоуровневом языке программирования (ассемблере, байт-коде). Основной целью обфускации является затруднение понимания функционирования программы [22]. Поскольку все традиционные компьютерные архитектуры используют двоичные строки, применяя полностью гомоморфное шифрование над битами, можно вычислить любую функцию. Следовательно, можно гомоморфно зашифровать целиком всю программу так, что она сохранит свою функциональность [7, 23].

Кроме того, гомоморфные свойства различных криптосистем в перспективе могут быть использованы для архивации и хранения медицинских записей без угрозы их утечки, фильтрации зашифрованной электронной почты, создания стойких к коллизиям хэш-функций.

Заключение

На основании вышеизложенного можно утверждать, что в ближайшей перспективе средства и методы гомоморфного шифрования будут оказывать существенное влияние на рынок облачных услуг и в той или иной степени на облик современных информационных технологий. Однако пока не созданы эффективные алгоритмы полностью гомоморфного шифрования, обеспечивающие уровень производительности, пригодный для практического применения, а тем более для применения в системах реального времени. Все предлагаемые схемы не реализуемы на практике и не готовы к внедрению в реальные системы, так как приводят к накоплению ошибок и быстрому увеличению шифрованных текстов. При этом частично гомоморфные системы (относительно операций сложения или умножения) успешно применяются в облачных вычислениях, электронном голосовании, защищенном поиске информации, системах с обратной связью и т. д.

Следует учитывать, что некоторые гомоморфные криптосистемы могут поддаваться преднамеренным внешним воздействиям (например, принципиально уязвимы к атаке с адаптивно подобранным шифрованным текстом) и поэтому не всегда подходят для безопасной передачи данных. Оценка криптостойкости гомоморфных систем требует отдельного исследования.

В отличие от облегченной криптографии для гомоморфного шифрования пока не разработаны соответствующие международные стандарты, однако активно продолжаются работы по созданию приемлемых решений, позволяющих безопасно обрабатывать конфиденциальные данные в облаках и других приложениях.

Список литературы

1. Поляков, А.С. Анализ возможностей алгоритмов международного стандарта «Облегченная криптография» – ISO/IEC 29192-2:2012 / А.С. Поляков, В.Е. Самсонов // Информатика. – 2014. – № 3. – С. 107–112.

2. Облачные технологии: новые задачи [Электронный ресурс]. – Режим доступа : http://events.cnews.ru/events/oblachnye_tehnologii__novye_zadachi.shtml. – Дата доступа : 03.02.2015.

3. Как ActiveCloud собирается заработать больше, чем обещает рынок облачных технологий [Электронный ресурс]. – Режим доступа : http://probusiness.by/tech/205.html. – Дата доступа : 03.02.2015.

4. Батура, Т.В. Облачные технологии: основные понятия, задачи и тенденции развития / Т.В. Батура, Ф.А. Мурзин, Д.Ф. Семич // Программные продукты и системы. – 2014. – № 3. – С. 64–72.

5. Афанасьев, С.В. Облачные сервисы, онтологическое моделирование таксономии / С.В. Афанасьев // Труды СПИИРАН. – 2012. – № 23. – С. 392–399.

6. Rivest, R.L. On data banks and privacy homomorphisms / R.L. Rivest, L. Adleman, M.L. Dertouzos // Foundations of secure computation. – 1978. – Vol. 32, no. 4. – P. 169–178.

7. Варновский, Н.П. Гомоморфное шифрование / Н.П. Варновский, А.В Шокуров // Труды Ин-та системного программирования РАН. – 2006. – Т. 12. – С. 27–36.

8. Survey of various homomorphic encryption algorithms and schemes / P.V. Parmar [et al.] // Intern. J. of Computer Applications. – 2014. – Vol. 91, no. 8. – P. 26–32.

9. Gentry, C. A Fully homomorphic encryption using ideal lattices / C. Gentry // Symposium on the Theory of Computing (STOC). – Bethesda, USA, 2009. – P. 169–178.

10. Математические и компьютерные основы криптологии : учеб. пособие / Ю.С. Харин [и др.]. – Минск : Новое знание, 2003. – 381 с.

11. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes / P. Paillier // Advances in cryptology – EUROCRYPT’99. – Berlin, Heidelberg : Springer, 1999. – P. 223–238.

12. Жиров, А.О. Безопасные облачные вычисления с помощью гомоморфной криптографии / А.О. Жиров, О.В. Жирова, С.Ф. Кренделев // Безопасность информационных технологий. – 2013. – Т.1. – С. 6–12.

13. Gentry, C. Homomorphic Encryption from Learning With Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based / C. Gentry, A. Sahai, B. Waters // Advances in cryptology – CRYPTO-2013, 33rd Annual Cryptology Conf. – Santa Barbara, CA, USA, 2013. – Part 1. – P. 73–93.

14. Ahmad, I. Survey: homomorphic encryption schemes / I. Ahmad, D. Adiga // Proc. of 9th IRF Intern. Conf. – Pune, India, 2014. – P. 89–94.

15. Jain, N. Implementation and analysis of homomorphic encryption schemes / N. Jain, S.K. Pal, D.K. Upadhyay // Intern. J. on Cryptography and Information Security (IJCIS). – 2012. – Vol. 2, no. 2. – P. 27–44.

16. Patel, S.J. Comparative Evaluation of Elliptic Curve Cryptography Based Homomorphic Encryption Schemes for a Novel Secure Multiparty Computation / S.J. Patel, A. Chouhan, D.C. Jinwala // J. of Information Security. – 2014. – № 5. – P. 12–18.

17. Batch fully homomorphic encryption over the integers / J.H. Cheon [et al.] // Advances in Cryptology – EUROCRYPT'2013 (LNCS). – 2013. – Vol. 7881. – P. 315–335

18. Степанян, А.Б. Актуальные вопросы обеспечения надежности и безопасности программного обеспечения систем дистанционного банковского обслуживания / А.Б. Степанян, А.И. Трубей, В.В. Анищенко // Электроника инфо. – 2014. – № 12. – С. 35–40.

19. Шенец, Н.Н. Модулярное разделение секрета и системы электронного голосования / Н.Н. Шенец // Вестник БГУ. Сер. 1. – 2011. – № 1. – С. 101–104.

20. Габидулин, Э.М. Защита информации в телекоммуникационных сетях / Э.М. Габидулин, Н.И. Пилипчук, О.В. Трушина // Труды МФТИ. – 2013. – Т. 5, № 3. – С. 97–111.

21. Diffie, W. New directions in cryptography / W. Diffie, M. Hellman // IEEE Trans. Inf. Theory. – 1976. – Vol. 22, no. 11. – P. 644–654.

22. Сергейчик, В.В. Особенности обфускации VHDL-описаний и методы оценки ее сложности / В.В. Сергейчик, А.А. Иванюк // Информатика. – 2014. – № 1. – С. 116–125.

23. On the relationship between functional encryption, obfuscation, and fully homomorphic encryption / J. Alwen [et al.] // Cryptography and Coding – 14th IMA Intern. Conf., IMACC-2013. – Oxford, UK, 2013. – P. 65–84.

Поступила 21.01.2015

Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: trubeia@newman.bas-net.by

A.I. Trubei

HOMOMORPHIC ENCRYPTION: CLOUD COMPUTING SECURITY AND OTHER APPLICATIONS (A SURVEY)

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and to obtain an encrypted result which matches the result of operations performed on the plain text. The article presents a basic concept of the homomorphic encryption and various encryption algorithms in accordance with the fundamental properties of the homomorphic encryption. The examples of various principles and properties of homomorphic encryption, some homomorphic algorithms using asymmetric key systems such as RSA, ElGamal, Paillier algorithms as well as various homomorphic encryption schemes are given. Prospects of homomorphic encryption application in the field of secure cloud computing, electronic voting, cipher text searching, encrypted mail filtering, mobile cipher and secure feedback systems are considered.