Об использовании библиотек полностью гомоморфного шифрования

А.А. Гаража, И.Ю. Герасимов, М.В. Николаев, И.В. Чижов

***Аннотация*—Технологии полностью гомоморфного шифрования позволяют выполнять операции над зашифрованными данными, не раскрывая их, благодаря чему имеют огромный потенциал применения в решении задач хранения и обработки персональных данных. Растущий интерес к таким технологиям привел к появлению множества программных средств и библиотек, поддерживающих полностью гомоморфное шифрование. Однако, в силу относительно молодого возраста этой области криптографии, стандарты и рекомендации по использованию схем полностью гомоморфного шифрования все еще находятся в разработке. Таким образом, применение указанных библиотек без уделения внимания вопросам криптографической стойкости используемых схем может иметь существенные риски информационной безопасности. В настоящей статье мы рассматриваем вопросы практического применения схем полностью гомоморфного шифрования, в том числе выбора подходящих библиотек и параметров их инициализации для обеспечения достаточного уровня информационной безопасности.**

***Ключевые слова*—полностью гомоморфное шифрование, удаленные вычисления**

I. Введение

Полностью гомоморфное шифрование (FHE) является обобщением классического шифрования, позволяя не только защищать конфиденциальные данные, но и выполнять их обработку, оперируя только лишь соответствующими шифртекстами (без использования ключа расшифрования). Указанные свойства гомоморфного шифрования открывают широкие возможности по его практическому применению, ярким примером которых являются *удаленные вычисления*: пользователь загружает данные на удаленный сервер в зашифрованном виде, выполняет вычисления над ними, а полученный результат расшифровывает уже локально.

Вопрос существования систем полностью гомоморфного шифрования был сформулирован еще в конце 70-х годов [1], однако долгое время оставался открытым. Ряд последующих результатов показал возможность построения систем *частично гомоморфного шифрования*, позволяющих выполнять лишь отдельные операции над зашифрованными данными (например, криптосистема RSA [2, 1] (гомоморфна относительно умножения), криптосистема Эль-Гамаля[3] (гомоморфна относительно умножения), криптосистема Пэйе [4] (гомоморфна относительно сложения)), а также систем *ограниченно гомоморфного шифрования*, позволяющих выполнять лишь ограниченные по сложности преобразования (например, криптосистема Бонэ-Го-Ниссим [5], допускающая преобразования содержащие произвольное количество сложений и только одно умножение). Первая система полностью гомоморфного шифрования была представлена в 2009 году в диссертационной работе Крейга Джентри [6]. Предложенная криптосистема, основанная на идеальных решетках, не была достаточна эффективна для практических задач, однако привлекла интерес большого круга экспертов к этой научной области. Уже спустя несколько лет появилось множество работ, содержащих более эффективные конструкции полностью гомоморфного шифрования, которые можно условно разделить на три поколения [7].

1) Различные модификации классической системы [6]. Используемые в их основе алгоритмы шифрования обладают свойством накапливания ошибки при выполнении операций над шифртекстами. Чтобы размер такой ошибки оставался в допустимых рамках, Джентри предложил использовать технику «переинициализации» (bootstrapping). Однако в силу вычислительной сложности этой техники, системы первого поколения имели относительно низкую эффективность.
2) Существенно повысить эксплуатационные характеристики систем гомоморфного шифрования удалось за счет уменьшения скорости накапливания ошибок, оптимизации техники переинициализации, «упаковке» нескольких открытых текстов в один шифртекст. Ко второму поколению можно отнести криптосистемы BGV (и ее программную реализацию HELib [8]), FV.
3) Наконец, следующий этап развития систем гомоморфного шифрования связан с идеей использования асиметричных операций умножения [9], что позволяет существенно уменьшать скорость накапливания ошибки.

В настоящее время доступно множество программных реализаций систем полностью гомоморфного шифрования. Некоторые из них носят экспериментальный характер и разработаны в академических целях, другие нацелены на использование широким кругом разработчиков. В рамках настоящей работы нас интересовали популярные библиотеки, предоставляющие возможности полностью гомоморфного шифрования и обладающие открытым ис-

Статья получена: 12.02.2021

Александра Андреевна Гаража, Лаборатория Криптографии АО НПК «Криптонит», (email: a.garazha@kryptonite.ru).

Илья Юрьевич Герасимов, Лаборатория Криптографии АО НПК «Криптонит», (email: i.gerasimov@kryptonite.ru).

Максим Владимирович Николаев, Лаборатория Криптографии АО НПК «Криптонит», (email: m.nikolaev@kryptonite.ru).

Иван Владимирович Чижов, МГУ имени М.В. Ломоносова, Лаборатория Криптографии АО НПК «Криптонит», Федеральный исследовательский центр «Информатика и управление» РАН (email: ichizhov@cs.msu.ru).

ходным кодом (таблица I).

Библиотека	Поддерживаемые системы	Язык программирования
HELib [10]	BGV [11], CKKS [12]	C++
SEAL [13]	BFV [14, 15], CKKS	C++
PALISADE [16]	BGV, BFV, CKKS, FHEW [17], TFHE [18]	C++
TFHE [19]	TFHE	C++
HEAAN [20]	CKKS	C++
$\Lambda \circ \lambda$ [21]	вариант BGV	Haskell
lattigo [22]	BFV, CKKS	Go

Таблица I: популярные библиотеки, предоставляющие возможности полностью гомоморфного шифрования

- HELib – одна из наиболее популярных библиотек, разработана Халеви и Шупом [8, 23], предоставляет возможность тонкой настройки режимов работы схем гомоморфного шифрования.
- Библиотека гомоморфного шифрования SEAL разработана исследователями Microsoft Research, поддерживает операции сложения и умножения над целыми и вещественными числами. Стоит отметить альтернативную JavaScript реализацию библиотеки [24].
- Библиотека криптографических механизмов PALISADE, основанных на целочисленных решетках, в том числе систем полностью гомоморфного шифрования.
- Библиотека разработана авторами одноименной системы полностью гомоморфного шифрования TFHE. В отличие от HELib и SEAL, не поддерживает работу с вещественными числами.
- Библиотека HEAAN разработана авторами системы CKKS, предоставляет возможность выполнения гомоморфных приближенных вычислений над вещественными числами.
- $\Lambda \circ \lambda$ – Haskell-библиотека общего назначения, предоставляющая интерфейс для многих математических операций, используемых в криптографических механизмах, основанных на целочисленных решетках. В том числе в библиотеке реализован модифицированный вариант системы BGV.
- lattigo – реализация на языке Go криптографических механизмов, основанных на целочисленных решетках. Включает классические и пороговые версии систем BFV и CKKS. В настоящее время библиотека носит экспериментальный характер и рекомендуется к использованию только в научных целях.

Работа над некоторыми из других библиотек не ведется уже продолжительное время: libScarab [25] (последний релиз: 26.08.2015), FHEW [26] (последний релиз: 30.05.2017), Krypto [27] (последний релиз: 21.10.2016), FV-NFLlib [28] (последний релиз: 26.07.2016). Отдельно стоит отметить ряд библиотек, использующих графические сопроцессоры для повышения эффективности систем гомоморфного шифрования, но не рассматриваемых нами в рамках настоящей работы: cuHE [29], cuFHE [30], cuYASHE [31], NuFHE [32].

II. Схемы полностью гомоморфного шифрования

В настоящем разделе мы дадим более формальные определения различных видов гомоморфного шифрования. Далее, мы определим две задачи из теории целочисленных решеток (LWE и RLWE), на вычислительной трудности которых базируются обоснования стойкости большинства схем гомоморфного шифрования, в том числе схем BGV, BFV и CKKS, используемых в рассматриваемых библиотеках. Наконец, будет выполнено сравнение некоторых свойств трех указанных схем.

Определение и виды гомоморфного шифрования

Схемой гомоморфного шифрования (HE, Homomorphic Encryption) с открытым ключом называется следующий набор полиномиальных вероятностных алгоритмов $(KeyGen, Enc, Dec, Eval)$.

- $KeyGen(1^\lambda) \to sk, pk$. По параметру стойкости λ выдает секретный и открытый ключи.
- $Enc(pk, m) \to c$. Выполняет шифрование текста m на открытом ключе pk.
- $Dec(sk, c) \to m$. Выполняет расшифрование шифртекста c на секретном ключе sk.
- $Eval(\Sigma, c_1, c_2, \ldots, c_l) \to c$. Выполняет вычисление схемы из функциональных элементов (СФЭ) Σ на входах $c_1, c_2, \ldots, c_l$. Предполагается, что схема выбирается из некоторого допустимого класса $\mathcal{C}$.

Схема гомоморфного шифрования является *корректной*, если она:

1) корректна как криптосистема с открытым ключом,
2) позволяет вместо вычислений с текстами производить вычисления с шифртекстами с последующей расшифровкой. То есть для любой допустимой СФЭ Σ и любых текстов $m_1, \ldots, m_l$ и их шифртекстов $c_1, \ldots, c_l$ выполнено

$$\begin{gathered} Dec(sk, Enc(pk, m)) = m, \\ Dec(sk, Eval(\Sigma, c_1, \ldots, c_l)) = \Sigma(m_1, \ldots, m_l). \end{gathered} \quad (1)$$

Схема называется *компактной*, если существует многочлен p такой, что размер вывода $Eval(\Sigma, c_1, \ldots, c_l)$ ограничен $p(\lambda)$ и не зависит ни от $\Sigma \in \mathcal{C}$, ни от $c_1, \ldots, c_l$. Это условие означает, что во-первых, по размеру вывода нельзя извлечь никакой информации ни о СФЭ, ни о данных. А во-вторых, что время расшифровки не зависит от вычисляемой СФЭ.

Схема гомоморфного шифрования называется схемой *полностью гомоморфного шифрования* (FHE, Fully HE), если она корректна и компактна для класса $\mathcal{C}$, состоящего из всех СФЭ (т.е. гомоморфна относительно операций сложения и умножения).

Крайне важными с практической точки зрения являются параметризованные схемы *уровневого полностью гомоморфного шифрования* (LFHE, Leveled FHE) — корректные и компактные схемы, для которых класс $\mathcal{C}$ состоит из всех СФЭ глубины не больше L. В этом случае на вход алгоритму $KeyGen$ подается еще один дополнительный параметр L. Обычно в качестве глубины рассматривают глубину по умножению.

Вычислительно трудные задачи на решетках

Для построения схем полностью гомоморфного шифрования используются задачи из теории целочисленных решеток, колец многочленов, теории чисел и т.д. Нас же будут интересовать всего две задачи, которые легли в

основу рассматриваемых схем. Это задачи LWE и RLWE из теории целочисленных решеток.

Введем обозначения. Векторы будем обозначать жирным шрифтом: **a**. Через $\langle\cdot|\cdot\rangle$ обозначим скалярное умножение. Кольцо вычетов по модулю q обозначим через $\mathbb{Z}_q$, для целого числа $z \in \mathbb{Z}$ через $[z]_q$ будем обозначать представителя в $\mathbb{Z}_q$ из интервала $\mathbb{Z} \cap (-q/2,\, q/2]$. Положим $R = \mathbb{Z}[x]/(x^d + 1)$ — фактор-кольцо многочленов, где в качестве d обычно берется степень 2, а также положим $R_q = R/qR$. Для вещественных чисел $r \in \mathbb{R}$ округление до ближайшего целого обозначим через $\lfloor r \rceil$. Выбор элемента x в соответствии с распределением D будем обозначать $x \leftarrow D$. Если множество S конечно, то $x \leftarrow S$ обозначает, что x выбрано в соответствии с равномерным распределением на S.

Обучение с ошибками (LWE, learning with errors). Это задача решения переопределенной системы линейных уравнений над кольцом вычетов, в которую внесены небольшие ошибки в соответствии с заданным распределением. Более строго, зафиксируем параметр $n \geq 1$, модуль $q \geq 2$ и распределение вероятности ошибки χ на $\mathbb{Z}_q$. Рассмотрим:

- $\{\mathbf{a_i}\}$ — набор известных случайных векторов из $\mathbb{Z}_q^n$,
- $\{e_i\}$ — набор малых неизвестных ошибок из $\mathbb{Z}_q$, полученных по распределению χ,
- **s** — неизвестный вектор из $\mathbb{Z}_q^n$.

Построим набор $\{b_i\}$, где

$$b_i := \langle \mathbf{a_i} | \mathbf{s} \rangle + e_i, \qquad i = 1 \dots m \tag{2}$$

Задача LWE заключается в нахождении вектора **s** по заданным $(\{\mathbf{a_i}\}, \{b_i\})$. *Распознавательная задача LWE* состоит в том, чтобы по $(\{\mathbf{a_i}\}, \{b_i\})$ определить, получены ли $\{b_i\}$ по формуле 2 или же выбраны случайным равновероятным образом из $\mathbb{Z}_q$.

Обучение с ошибками над кольцом (RLWE, Ring-LWE). Это задача обучения с ошибками, где вместо колец $\mathbb{Z}_q$ и $\mathbb{Z}_q^n$ используется фактор-кольцо многочленов $R_q = \mathbb{Z}_q[x]/(x^d + 1)$, а вместо скалярного умножения — умножение в кольце.

Обе эти задачи считаются вычислительно трудными даже в квантовом случае, на их основе строятся многие постквантовые криптографические механизмы. Задача RLWE отличается от LWE наличием на кольце R_q сильной алгебраической структуры, что имеет свои преимущества и недостатки. С одной стороны, предположение о вычислительной трудности решения задачи RLWE является более сильным по сравнению с предположением о LWE. С другой стороны, для схем, построенных на основе задачи RLWE, длина ключа существенно меньше, что имеет большое значение на практике.

Схемы полностью гомоморфного шифрования

Как можно догадаться, криптосистемы, основанные на задачах LWE/RLWE, предполагают внесение в вырабатываемые шифртексты небольших случайных ошибок, выбираемых из специального распределения. Таким образом, в случае схем гомоморфного шифрования вычисления производятся над зашумленными шифртекстами. Сложность описанного подхода заключается в том, что после некоторого количества вычислений шум вырастает настолько, что шифртекст невозможно расшифровать корректно (причем при сложении шум растет линейно, а при умножении — квадратично). В первых работах (например, [6]) эта проблема была решена с помощью «переинициализации» — выполнения гомоморфного расшифрования [1] сильно зашумленного шифртекста (см. рис. 1). Переинициализация позволяет уменьшать внесенные ошибки и продолжать вычисления над шифртекстом. К сожалению, использование на практике этой операции осложнено высокой вычислительной стоимостью.

Одной из первых схем, позволившей свести к минимуму (а в ряде случаев – отказаться) использование переинициализации, стала схема Бракерски-Джентри-Вайкутанатан, представленная в 2011 году. Вместо переинициализации авторы предложили использовать *смену ключа* и *смену модуля*.

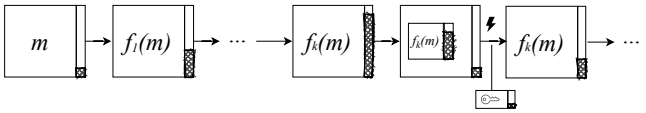

Рис. 1: Схемы с переинициализацией: квадрат вокруг текста обозначает шифрование, а штрихованный прямоугольник показывает уровень шума, который растет в процессе вычислений. Когда шум становится максимально допустимым, происходит переинициализация (отмечена молнией) — повторное шифрование.

Схема BGV (Brakerski-Gentry-Vaikuntanathan) [11]. Пусть множество открытых текстов — это кольцо R_p, а множество шифртекстов — кольцо R_q, где p и q — простые числа, причем $q \equiv 1 \mod p$. Основные преобразования, используемые в алгоритмах схемы $KeyGen$, Enc и Dec, приведены в табл. II. Расшифрование задается формулой $m = [[\langle \mathbf{c}|\mathbf{s}\rangle]_q]_p$. Откуда следует, что для текстов m_1, m_2 и соответствующих шифртекстов c_1, c_2 выполнено:

$$m_1 + m_2 = [[\langle \mathbf{c_1} + \mathbf{c_2} | \mathbf{s}\rangle]_q]_p,$$
$$m_1 m_2 = [[\langle \mathbf{c_1} \otimes \mathbf{c_2} | \mathbf{s} \otimes \mathbf{s}\rangle]_q]_p.$$

	BGV	BFV
KeyGen	$s', e \leftarrow \chi$ $a \leftarrow R_q$ $b \leftarrow -(as' + pe)$ $sk \leftarrow \mathbf{s} = (1, s')$ $pk \leftarrow \mathbf{k} = (b, a)$	$s', e \leftarrow \chi$ $a \leftarrow R_q$ $b \leftarrow [-(as' + e)]_q$ $sk \leftarrow \mathbf{s} = (1, s')$ $pk \leftarrow \mathbf{k} = (b, a)$
Enc	$r \leftarrow R_p$ $\mathbf{e} \leftarrow \chi^2$ $\mathbf{m} \leftarrow (m, 0)$ $\mathbf{c} \leftarrow \mathbf{m} + \mathbf{k}r + p\mathbf{e}$	$r \leftarrow R_p$ $\mathbf{e} \leftarrow \chi^2$ $\mathbf{m} \leftarrow (m, 0)$ $\mathbf{c} \leftarrow \lfloor q/p \rfloor \mathbf{m} + \mathbf{k}r + \mathbf{e}$
Dec	$m \leftarrow [[\langle \mathbf{c}\|\mathbf{s}\rangle]_q]_p$	$m \leftarrow [\lfloor \frac{p}{q} [\langle \mathbf{c}\|\mathbf{s}\rangle]_q \rceil]_p$

Таблица II: Схемы шифрования с открытым ключом BGV и BFV

Таким образом, сложению текстов соответствует сложение шифртекстов, а умножению текстов — тензорное произведение шифртектов и изменение секретного ключа **s** на $\mathbf{s} \otimes \mathbf{s}$. Чтобы избежать чрезмерного роста длин ключей, используют процедуру «смены ключа» (KeySwitch), при которой ключ и шифртекст $\mathbf{s}, \mathbf{c}$ меняются на более

[1] с использованием зашифрованного секретного ключа

короткие $\mathbf{s}', \mathbf{c}'$ так, что $[[\langle \mathbf{c}|\mathbf{s}\rangle]_q]_p = [[\langle \mathbf{c}'|\mathbf{s}'\rangle]_q]_p$. При этом также растет зашумленность шифртекста, для уменьшения которой используют процедуру «смены модуля» (ModulusSwitch), которая пару $\mathbf{c}, q$ заменяет на $\mathbf{c}', q'$ так, что $[[\langle \mathbf{c}|\mathbf{s}\rangle]_q]_p = [[\langle \mathbf{c}'|\mathbf{s}\rangle]_{q'}]_p$, шум при этом уменьшается в q/q' раз. Таким образом, в процессе гомоморфного вычисления СФЭ на каждом уровне по умножению меняются как модули, так и ключи (см. рис. 2). Более точно, модули выражаются по формуле $q_i = p_0 \cdot \ldots \cdot p_i$, где p_i — простые числа с условием $p_i \equiv 1 \mod p$ (т.е. модули уменьшаются в процессе вычислений).

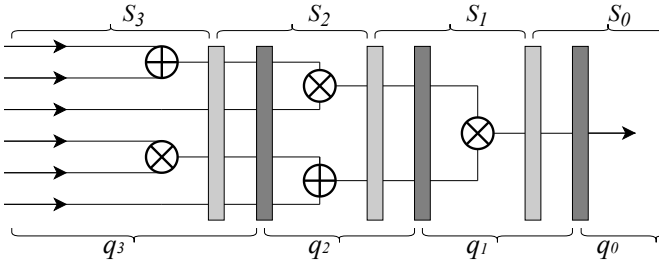

Рис. 2: Схема BGV: после каждого уровня по умножению происходит смена ключа (светло серые прямоугольники) и смена модуля (темно серые прямоугольники).

Схема BFV, Brakerski/Fan-Vercauteren [14, 15]. Эта схема шифрования была представлена в двух работах в 2012 году и во многом схожа с BGV-схемой. Основные преобразования, используемые в алгоритмах схемы $KeyGen$, Enc и Dec, также приведены в табл. II. Умножение на $\lfloor q/p \rfloor$ при вычислении шифртекста позволяет разделить сообщение (смещается к более значимым битам в $\langle \mathbf{c}|\mathbf{s}\rangle$) и добавляемый шум (остается в наименее значимых битам в $\langle \mathbf{c}|\mathbf{s}\rangle$). Благодаря этому отпадает необходимость в операции смены модуля, кроме того, ослаблены ограничения на размер модуля q (нет требования $q \equiv 1 \mod p$).

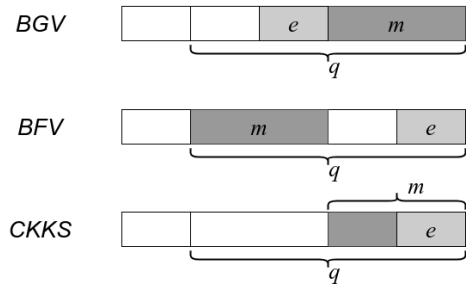

Рис. 3: Относительное расположение текста m и шума e внутри $\langle \mathbf{c}|\mathbf{s}\rangle$ в схемах BGV, BFV и CKKS

Сравнительный анализ схем BGV и BFV может быть найден в работах [33, 34]. В работе [34] сделан вывод, что схемы BGV и BFV имеют лишь незначительные различия в производительности с точки зрения вычислительных возможностей.

Схема CKKS (Cheon-Kim-Kim-Song) [12]. Эта схема была предложена в 2015 году и принципиально отличается от описанных выше: она работает не с целыми числами, а с числами с плавающей запятой. И строго говоря, эта схема не удовлетворяет определению 1 гомоморфного шифрования, т.к. равенства выполняются лишь приближенно. Принципиальным также является взаимное расположение текста и шума в шифртексте: в этой схеме погрешность результата вычисления и вносимый шум объединяются (см. рис 3). Операции смены ключа в данной схеме заменяются на операции «ремасштабирования» (Rescaling). Кроме того, схема позволяет вычислять и трансцендентные функции (экспонента, логарифм и т.д.) с помощью разложения Тейлора.

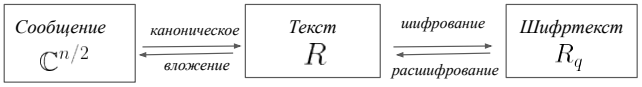

Рис. 4: Схема CKKS для работы с комплексными числами

Чтобы представить вещественное число как элемент кольца R используется каноническое вложение $\varphi : \mathbb{C}^{n/2} \to R$ при котором небольшая погрешность в поле комплексных чисел влечет небольшую погрешность в кольце R (см. рис. 4).

Простейшие свойства всех трех схем представлены в таблице III.

	BGV	BFV	CKKS
Представление текста	$\mathbb{Z}_p$	$\mathbb{Z}_p$	$\mathbb{R}, \mathbb{C}$
Расшифрование	$[[\langle c\|s\rangle]_q]_p$	$[[\lfloor \frac{p}{q}[\langle c\|s\rangle]_q \rceil]_p$	$[\langle c\|s\rangle]_q]$
Релинеаризация	$KeySwitch$, $ModSwitch$	$KeySwitch$	$Rescaling$
Модули шифртекста, $q_i =$	$p_0 \cdot \ldots \cdot p_i$, где $p_k \equiv 1 \mod p$	$q_0 \Delta^i$	$q_0 \Delta^i$
Масштабируемость	—	✓	✓
Трансцендентные функции	—	—	✓

Таблица III: Сравнение общих свойств схем BGV, BFV и CKKS.

III. Методы решения задач LWE и RLWE

В этом разделе мы перечислим наиболее эффективные методы решения задач LWE и RLWE, а также сформулируем требования к параметрам схем полностью гомоморфного шифрования.

Методы решения задачи LWE можно разбить на 3 класса: комбинаторные (BKW), алгебраические (Arora-Ge) и основанные на сведении к задачам теории решеток (BDD, SIS и uSVP). При рассматриваемых рекомендуемых параметрах схемы (в случае, когда размерность d кольца $R = \mathbb{Z}[x]/(x^d + 1)$ является степенью двух, а распределение ошибки χ достаточно «узкое») использование алгебраической структуры кольца RLWE для получения более эффективного решения (по сравнению с LWE), неизвестны [35]. Таким образом, все атаки на задачу RLWE сводятся к атакам на задачу LWE.

Далее содержание настоящего раздела опирается на работу [36], в которой проанализированы различные подходы к решению задачи LWE, а также программное средство *LWE Estimator* [37] для оценки уровня стойкости криптографических механизмов, основанных на задаче LWE. Для компактной записи введем обозначения: матрицу, состоящую из m строк $\mathbf{a_i}$ обозначим через $\mathbf{A}$, вектора, состоящие из чисел b_i и e_i, обозначим через $\mathbf{b}$ и $\mathbf{e}$ соответственно. Тогда уравнения 2 могут быть записаны в следующем виде: $\mathbf{b} = \mathbf{As} + \mathbf{e}$.

Алгоритм BKW. Алгоритм предложен Blum, Kalai, Wasserman в 2003 году для решения задачи LPN (learning parity with noise) и может быть обобщен на случай задачи LWE. Алгоритм является модификацией метода Гаусса, использование которого в случае LWE невозможно из-за быстрого накопления ошибки в процессе вычислений. BKW, как и метод Гаусса, состоит из трех этапов: прямого хода, нахождения значения последней переменной и обратного хода. На первом этапе алгоритма BKW строки подбираются так, чтобы каждый раз при вычитании сразу несколько значений обращалось в ноль. Это возможно, т.к. система уравнений сильно переопределена. Таким образом, за небольшое количество вычитаний получаем блочную верхнетреугольную матрицу. Второй этап, «проверка гипотез», заключается в подборе последних значений таких, чтобы распределение ошибки было наиболее близко к ожидаемому. Обратный ход, как и в алгоритме Гаусса, заключается в последовательной подстановке значений в уравнения.

Алгоритм Arora-Ge. Алгоритм был предложена Arora и Ge в 2011 году [38] и основан на сведении системы линейных уравнений с шумом к системе нелинейных уравнений без шума. Предполагается, что ошибка лежит в интервале $[-t, t]$ для некоторого натурального t и тогда система линейных уравнений $e_i = \langle \mathbf{a}_i | \mathbf{x} \rangle - \mathbf{b_i}$ равносильна системе нелинейных уравнений $P(\langle \mathbf{a}_i | \mathbf{x} \rangle - \mathbf{b_i}) = 0$, где $P(z) = \prod_{i=-t}^{t}(z - i)$ — многочлен с корнями $[-t, t]$. Затем все одночлены объявляются новыми переменными и решается система линейных уравнений относительно этих переменных. Чтобы система была определена, количество уравнений должно быть порядка $O(n^{2t+1})$, но в этом случае высока вероятность того, что шум перестанет попадать в интервал $[-t, t]$. Таким образом, выбор параметров оказывается достаточно тонким моментом. Отметим, что алгоритм может быть модифицирован с помощью применения базиса Гребнера, также дополнительное ускорение можно получить и в случае наличия решения малой нормы [36].

Следующие три метода основаны на сведении к классическим задачам из теории решеток. Для решения этих задач обычно используют *алгоритм приведения базиса BKZ (block Korkin-Zolotarev*, [39]), в результате которого получают более удобный (близкий к ортогональному) базис. Описание этого алгоритма и различные оценки его сложности могут быть найдены в работе [36]. Для оценки длин векторов базиса, получаемого с помощью BKZ, обычно используют *предположение о геометрической прогрессии* (GSA, Geometric Series Assumption [40]).

Метод декодирования. Этот метод заключается в сведении к задаче BDD (Bounded Distance Decoding) — задаче поиска ближайшего вектора решетки Λ к вектору v при условии, что вектор v расположен достаточно близко к Λ. В качестве Λ рассматривается решетка, порожденная столбцами матрицы $\mathbf{A}$, после чего выполняется поиск вектора, ближайшего к $\mathbf{b}$. Вначале базис приводится с помощью алгоритма BKZ, а затем применяется метод ближайших плоскостей [41], который находит вектор решетки такой, что вектор ошибки лежит в фундаментальном параллелепипеде базиса Грама-Шмидта. Сложность заключается в том, что этот фундаментальный параллелепипед оказывается «вытянутым» по предположению GSA. Поэтому Линднер и Пекерт разработали [42] модификацию метода ближайших плоскостей, в процессе которого фундаментальный параллелепипед «утолщается».

Метод на основе дуальной решетки. Этот метод заключается в сведении к задаче SIS (Short Integer Solutions) — поиску целочисленного решения малой нормы системы $\mathbf{xA} \equiv 0 \mod q$, что равносильно поиску короткого вектора в дуальной решетке $q\Lambda^* = \{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{xA} \equiv 0 \mod q\}$. Пусть такой короткий вектор $\mathbf{v}$ найден, рассмотрим $x = \langle \mathbf{v} | \mathbf{e} \rangle = \langle \mathbf{v} | \mathbf{b} \rangle$. Тогда распознавательная задача LWE сводится к определению распределения числа x: если вектор $\mathbf{b}$ распределен равномерно, то и x распределен равномерно над $\mathbb{Z}_q$, а если вектор $\mathbf{b}$ распределен в соответствии с формулой 2, то x распределен согласно гауссовому распределению по модулю q. Заметим, что эффективность этого подхода может быть увеличена в случае использования разреженных ключей или ключей малой нормы[43].

Метод на основе задачи $uSVP$. Этот метод [44] заключается в сведении к задаче uSVP (unique shortest vector problem) — задаче поиска единственного кратчайшего вектора решетки. Для этого рассматривается решетка

$$\Lambda = \{\mathbf{x} \in \mathbb{Z}^{m+n+1} : (\mathbf{A}| - \mathbf{I_m}| - \mathbf{b})\mathbf{x} = 0 \mod q\}$$

с единственным кратчайшим вектором $\mathbf{v} = (\mathbf{s}, \mathbf{e}, 1)$. Этот кратчайший вектор $\mathbf{v}$ может быть найден с помощью BKZ, в случае если проекция $\mathbf{v}$ на векторное пространство, натянутое на последние β векторов базиса Грама-Шмидта достаточно коротка. Используя предположение GSA, можно получить достаточное условие на β — длину блоков в методе BKZ [35].

IV. Рекомендованные параметры безопасности

В настоящем разделе мы представим требования к параметрам схем полностью гомоморфного шифрования, которые позволят получить необходимый уровень стойкости для практического применения. Параметрами задачи RLWE являются: n — размерность кольца $R = \mathbb{Z}[x]/(x^n + 1)$, q — модуль кольца для вычислений с шифртекстами, χ_e — распределение ошибки и χ_s — распределение секретного ключа.

Распределение ошибки χ_e. В большинстве библиотек полностью гомоморфного шифрования распределение, в соответствии с которым выбирается ошибка для зашумления шифртекста, фиксировано — это дискретное гауссово распределение со средним значением $\mu = 0$ и среднеквадратичным отклонением $\sigma = 8/\sqrt{2\pi} \approx 3.2$. Такой выбор параметра σ объясняется тем, что согласно работе [45] это наименьшее из устойчивых значений.

Распределение секретного ключа χ_s. Обычно в качестве χ_s рассматривают одно из следующих распределений [35]: равномерное, распределение ошибки χ_e, тернарное (равномерное на множестве $\{-1, 0, 1\}$) и разреженное распределение. В библиотеках полностью гомоморфного шифрования, рассматриваемых в настоящей статье, используется тернарное распределение секретного ключа.

Параметры n и q. Напомним, что в качестве параметра n мы рассматриваем только степени 2. С помощью программного средства LWE Estimator можно по заданному

набору параметров (n, q, χ_e, χ_s) получить оценку времени работы для различных методов решения LWE. Будем говорить, что схема с заданными параметрами имеет уровень безопасности λ, если наиболее эффективный метод решения требует не менее 2^λ операций. При рассматриваемых уровнях безопасности и размерах ключей наиболее эффективными оказались методы, основанные на теории решеток. А среди них наиболее эффективным – на основе задачи uSVP.

Для достижения необходимого уровня безопасности λ модуль шифртекстов q должен быть не больше, чем указанный в таблице. В случае применения схемы глубины L с модулями $q_{L-1}, \ldots, q_0$ необходимо, чтобы больший из модулей q_{L-1} удовлетворял этому условию. Для вычислений, требующих большей глубины по умножению, применяется переинициализация, которая реализована в обеих библиотеках.

	n					
λ	1024	2048	4096	8192	16384	32768
128	27	54	108	218	438	881
192	19	37	75	151	304	611
256	14	29	58	118	237	476

Таблица IV: Значения $\log_2 q$, соответствующие уровню безопасности $\lambda \in \{128, 192, 256\}$ и размерности кольца $n \in \{2^{10}, \ldots, 2^{15}\}$

V. Об инженерном применении библиотек полностью гомоморфного шифрования

Разработка схем полностью гомоморфного шифрования является многообещающим событием в области защиты и обработки конфиденциальных данных. Технологии, построенные с использованием гомоморфного шифрования, позволяют избежать рисков компрометации обрабатываемых персональных данных. Несмотря на молодой возраст этой области криптографии и относительно невысокую эффективность существующих схем, полностью гомоморфное шифрование уже доказало свою эффективность в ряде практических задач. Далее, следуя [46], мы отметим несколько примеров.

- *Здравоохранение*
  Цифровизация медицинских услуг неразрывно связана со сбором, хранением и обработкой данных о здоровье человека и его фенотипических признаках. Возможность хранения и обмена этими данными между специализированными учреждениями позволяет не только сделать оказание услуг пациентам более удобным, но и повысить эффективность предиктивной медицины. Однако персональные данные, содержащие информацию о физическом или психическом состоянии человека, крайне чувствительны к раскрытию. Использование гомоморфного шифрования позволяет осуществить хранение и обработку таких персональных данных на базе недоверенных облачных платформ (см., например [47]). Другим важным примером являются биологические исследования по полногеномному поиску ассоциаций, позволяющему на основе однонуклеотидных полиморфизмов выявлять возможные заболевания человека. Использование гомоморфного шифрования в этой области (см., например [48]) позволяет сохранить конфиденциальность не только данных пациента, но и интеллектуальной собственности медицинского учреждения (алгоритма выявления заболеваний).
- *Финансовая сфера*
  Как и в случае с медицинскими данными, конфиденциальность информации о финансовом благополучии человека должна бережно охраняться. С другой стороны, деятельность в сфере оказания банковских услуг в настоящее время невозможна без анализа широкого спектра пользовательских данных для оценки платежеспособности, построения качественной рекомендательной системы и т.д. Результаты пилотного проекта [49], проведенного исследователями IBM Research и специалистами бразильской банковской компании Banco Bradesco, показали возможность практического применения библиотек полностью гомоморфного шифрования как для предсказаний по зашифрованным пользовательским данным, так и для непосредственного обучения модели на зашифрованных данных.
- *Городское и государственное управление*
  Еще одним примером является использование полностью гомоморфного шифрования для анализа данных муниципальными и государственными организациями. Централизованная обработка персональных данных граждан или данных городской инфраструктуры (умные города) крайне нежелательна в связи с рисками компрометации. Агрегация и обработка таких данных в зашифрованном виде для получения статистических отчетов не требует больших вычислительных ресурсов и может быть осуществлена с помощью современных схем гомоморфного шифрования. Результаты такого анализа могут использоваться для целей планирования городских/государственных программ, а также выявления аномалий.

Далее мы детально разберем несколько примеров использования библиотек полностью гомоморфного шифрования для решения практических задач, уделяя внимания эксплуатационным характеристикам получаемых решений.

A. Поиск по базе зашифрованных представлений изображений

В то время как конфиденциальность некоторых типов данных может быть обеспечена классическими методами, защита изображений (например, портретных фотографий пользователей) средствами традиционных криптосистем крайне ограничена: дело в том, что такие персональные данные обычно используются в открытом (расшифрованном) виде в качестве входных данных для алгоритмов распознавания и поиска по базе имеющихся изображений. Стоит отметить, что компрометация вектора числовых признаков фотографии (например, используемых для классификации) также может привести к раскрытию сведений о человеке: поле, расе, возрасте и т.п.

Решением этой проблемы может быть использование базы зашифрованных представлений изображений, поддерживающей операцию поиска по входному зашифрованному образцу. В этом случае отпадает необходимость

использования открытых изображений, а все операции производятся только с зашифрованными представлениями. Один из первых результатов по использованию полностью гомоморфного шифрования для этой задачи [50] показал возможность сопоставления зашифрованных изображений с размерностью представления 512. Экспериментальная оценка эффективности разработанного алгоритма (одно сопоставление занимает 12.8 секунд и требует 48.7 MB памяти) не позволяет применять его для поиска, т.е. сопоставления «один-ко-многим».

Эффективную систему поиска такого типа на базе библиотеки SEAL представила группа исследователей из Университета штата Мичиган [51]. Авторами используется масштабирование представления (вектора признаков) с сохранением точности на основе нейронной сети DeepMDS++ [52], а также модификация реализации схемы полностью гомоморфного шифрования BFV из библиотеки SEAL для работы с массивом чисел согласно SIMD принципу вычислений. Используемые оптимизации позволили получить следующие результаты: при использовании платформы Tensorflow [53] для DeepMDS++ и библиотеки SEAL для BFV схемы поиск изображения в базе данных, состоящей из миллиона представления размерности 512 занимает от 16 до 20 минут и использует 22 GB памяти. Для базы данных того же размера, но с представлениями размерности 128 поиск занимает не более 5 минут и требует 5.5 GB памяти. Для сравнения классическое применение полностью гомоморфного шифрования [54] требует до 4 часов и 90 GB памяти независимо от размерности представления. Для проведения экспериментов была использована аппаратная платформа на базе 10-ядерного процессора Intel i9-7900X с частотой 3.30 GHz с однопоточным окружением. В качестве параметров схемы BFV были использованы:

- $n = 4,096$;
- $\log_2 q = 108$;
- $\sigma = 3.20$ — среднеквадратическое отклонение, определено в библиотеке SEAL по умолчанию;
- тернарное распределение секретного ключа.

В [51] авторы также отмечают, что именно использование схемы полностью гомоморфного шифрования позволило получить такую эффективность. Так использование систем частично гомоморфного шифрования (например, криптосистемы Пайэ) не позволяет использовать SIMD принцип вычислений, а применение техник совместных конфиденциальных вычислений (MPC) требует накладных расходов по разделению базы данных между участниками протокола.

В таблице V приведены полученные авторами работы оценки на количество используемых операция сложения и умножения, а также необходимой памяти в зависимости от размерности представления изображений и количества изображений.

Кол-во умножений	Кол-во сложений	Память
$\lceil \frac{m}{n} \rceil d$	$\lceil \frac{m}{n} \rceil (d-1)$	$\mathcal{O}(dn\lceil \frac{m}{n} \rceil)$

Таблица V: Количество операций и асимптотическая оценка количества затрачиваемой памяти относительно размерности представления изображений d, их количества m, при наличии n вычислительных узлов

B. Конфиденциальный полногеномный поиск ассоциаций

Поиск связей между изменениями в ДНК (нуклеотидными полиморфизмами) и заболеваниями человека с целью исследования природы таких заболеваний, причин возникновения и борьбы с ними называется полногеномным поиском ассоциаций (genome-wide association studies, GWAS).

Однако сама структура ДНК представляет собой генетический код человека, позволяющий получить информацию о его здоровье. В результате обучение диагностирующей системы должно выполняться на зашифрованных данных, так же как и её использование при классификации очередной цепочки ДНК. В рамках этой задачи каждую ДНК можно представить вектором с целочисленными элементами, которые, в свою очередь, соответствуют нуклеотидам ДНК.

Приведем формальное описание задачи. Пусть информация о некоторой сущности задается набором значений $(\mathbf{x}, y)$, где $\mathbf{x} \in \mathbb{R}^d$ известны (это может быть не только последовательность нуклеотидов ДНК, но также информация о человеке, данные состояния устройства и пр.), а $y \in \{0, 1\}$ требует предсказания (например, $y = 1$ может означать, что у пользователя есть некоторое заболевание, пользователь сможет погасить кредит или устройство неисправно). Требуется по $\mathbf{x}$ определить вероятность того, что $y = 1$, то есть найти $p_{\mathbf{x}} = Pr[y = 1|\mathbf{x}]$. Для вычисления $p_{\boldsymbol{x}}$ существует множество классических решений, среди которых:

- логистическая регрессия [55];
- тест Пирсона по критерию хи-квадрат [56].

Пусть также значение $\mathbf{x}$ является конфиденциальной информацией и не может быть передана в открытом виде. В таком случае необходимо рассматривать решение на основе уже зашифрованных признаков, выполнение операций над которыми должно соответствовать тем же операциям, что и над самими признаками.

В статье [48] авторы приводят описание решения этой задачи. Для этого была использована имплементация схемы CKKS [12] полностью гомоморфного шифрования в библиотеке PALISADE [16]. Дополнительным преимуществом по эффективности является подключение OpenMP для реализации параллельных вычислений. Для повышения эффективности используется система остаточных классов (Residue Number System, RNS), позволяющая работать с вектором полиномов по модулю и проводить ремасштабирование.

В схеме CKKS использовались следующие параметры:

- Дискретное гауссово распределение ошибки со среднеквадратическим отклонением $\sigma = 3.19$;
- Размерность кольца $n = 2^{13}$;
- $\log_2(q) = 50$;
- Тернарное распределение секретного ключа.

Авторы провели результаты как для метода логистической регрессии, так и для теста хи-квадрат. Было получено, что применение критерия хи-квадрат является лучшим вариантом и на выборке из 15000 векторов длины 2^{14} было получено ускорение в 41 раз. Более того, экстраполяция полученных результатов указывает что FHE решение задачи с применением критерия хи-квадрат эффективнее чем применение протокола конфиденциальных вычислений: для 100000 векторов длины

500000 потребуется 5.6 часов по сравнению с 197 часами MPC. По количеству используемой памяти, хи-квадрат также превосходит логистическую регрессию в 8 раз, что приводит к ускорению реализации параллельных вычислений, узким местом которых является работа с памятью. Сравнение результатов экстраполяции приведено в таблице VI.

Кол-во векторов	ЛР	Хи-квадрат
5 000	1301 с, 162 GB	39 с, 35 GB
7 500	1994 с, 244 GB	54 с, 47 GB
10 000	2688 с, 324 GB	72 с, 59 GB
15 000	4074 с, 485 GB	105 с, 83 GB

Таблица VI: Экстраполяция результатов при длине вектора 2^{14}

C. Гомоморфная свёрточная нейронная сеть

Глубокое обучение доказало свою эффективность во множестве прикладных задач: рекомендательных системах, контентной фильтрации, электронной коммерции. Нередко предобученные модели размещают в облачных хранилищах, предоставляющих быстрый и дешевый способ построения отказоустойчивых распределенных сервисов. В таких случая критичным с точки зрения конфиденциальности становится передача пользовательских персональных данных для выполнения модели.

В работе [57] была представлена гомоморфная свёрточная нейронная сеть (Homomorphic Convolutional Neural Networks), позволяющая выполнять работу с зашифрованными данными пользователей. На первом этапе выполняется обучение сети на незашифрованных изображениях, после чего производится классификацию подаваемых зашифрованных изображений. Для формирования обучающей и тестовой выборок были использованы наборы данных MNIST и CIFAR-10. В качестве библиотеки полностью гомоморфного шифрования были использованы SEAL и A*FV [58] (BFV схема). Проведенные эксперименты показали большую эффективность решения на базе библиотеки A*FV, использующей вычисления на GPU. В таблицах VII, VIII приведены основные характеристики реализации HCNN. Поскольку наиболее тяжеловесной операцией в схемах FHE является умножение, мы ограничимся только указанием результатов относительно этой операции, а также оценкой времени генерации ключей.

	MNIST	CIFAR-10
Количество умножений на открытое сообщение	46 000	6 952 332
Количество умножений шифртекстов	1 520	57 344

Таблица VII: Количество умножений для MNIST и CIFAR-10

Ограничения, связанные с использованием FHE

Важно отметить, что возможности использования полностью гомоморфного шифрования для практических задач имеют несколько ограничений.

- *Относительно низкие эксплуатационные характеристики криптографических схем.*

Операция \ Библиотека	SEAL мс	A*FV мс
Генерация ключей	542.920	21.392
Возведение в квадрат	138.199	2.371
Умножение	173.167	2.769

Таблица VIII: Время выполнения операций с использованием библиотек SEAL и A*FV на CPU Intel Xeon Platinum и GPU NVIDIA Tesla P100

Выполнение операций над зашифрованными данными с использованием современных FHE схем требует существенных накладных расходов, поэтому гомоморфное осуществление ресурсоемких вычислений обычно оказывается экономически необоснованным. Хотя перенос вычислительно тяжелой обработки (например, рендеринг видео, научные экспериментальные расчеты) от персональных компьютеров к облачным провайдерам с сохранением конфиденциальности данных является крайне привлекательной задачей, ее решение исключительно средствами FHE в настоящее время кажется невозможным.

- *Сложность обеспечения многопользовательских режимов работы.*
  Ключевое значение для полностью гомоморфного шифрования имеет принципиальная невозможность сторон, не обладающих секретным ключом, расшифровать обрабатываемые данные. В случае многопользовательской системы агрегируемые данные зашифрованы на различных секретных ключах, что делает их совместную обработку средствами FHE невозможной. Решить эту проблему удается лишь с использованием более общих подходов, таких как мультиключевое гомоморфное шифрование [59].

Накладные расходы, необходимые для использования схем полностью гомоморфного шифрования, также делают нежелательным их использования там, где можно обойтись более традиционными криптографическими механизмами. Внедрение систем электронного голосования призвано повысить прозрачность проведения выборов, ускорить подсчет бюллетеней, снизить затраты на оплату труда персонала и обеспечить доступность для избирателей с ограниченными возможностями. Онлайн-голосование может быть реализовано с помощью методов частично гомоморфного шифрования. Например, в работе [60] приводится пример схемы на основе криптосистемы Пайэ. Более того, международная ассоциация криптологических исследований (IACR) для выборов членов совета директоров использует систему Helios [61], основанную на криптосистеме Эль-Гамаля.

VI. Заключение

Разработка схем полностью гомоморфного шифрования является не только научным прорывом в теоретической криптографии, но и многообещающим событием в области практической защиты и обработки конфиденциальных данных. Технологии, построенные с использованием гомоморфного шифрования, позволяют нивелировать риски компрометации обрабатываемых персональных данных и потенциально могут быть применены во всех сферах человеческой деятельности.

Использование полностью гомоморфного шифрования для выполнения ресурсоемких операций с зашифрованными данными все еще остается затруднительным на практике из-за относительно низких эксплуатационных характеристик основных криптографических схем. При этом применение FHE в задачах, не требующих больших вычислительных затрат, таких как прогнозирование с использованием предварительно обученной модели, возможно уже сейчас. Кроме того, программные библиотеки, предоставляющие возможности полностью гомоморфного шифрования, активно развиваются и оптимизируются, что постоянно расширяет область их применения. Немаловажное значение имеют проекты по разработке аппаратной поддержки типовых арифметических операций, используемых в гомоморфном шифровании, что позволит существенно повысить их общую эффективность.

Использование полностью гомоморфного шифрования открывает возможности для обеспечения надежного хранения и обработки персональных данных, включающих данные о здоровье человека, его финансовом благополучии и других. В связи с крайней чувствительностью таких данных к компрометации, необходима стандартизация криптографических механизмов FHE, а также выработка рекомендаций по их использованию, основанная на научном консенсусе.

Библиография

[1] Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms // Foundations of Secure Computation, Academia Press. — 1978. — P. 169–179.

[2] Rivest R. L., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. — 1978. — Vol. 21, no. 2. — P. 120–126. — URL: https://doi.org/10.1145/359340.359342.

[3] El Gamal Taher. A public key cryptosystem and a signature scheme based on discrete logarithms // Proceedings of CRYPTO 84 on Advances in Cryptology. — Berlin, Heidelberg : Springer-Verlag, 1985. — P. 10–18.

[4] Paillier Pascal. Public-key cryptosystems based on composite degree residuosity classes // Advances in Cryptology — EUROCRYPT '99 / Ed. by Jacques Stern. — Berlin, Heidelberg : Springer Berlin Heidelberg, 1999. — P. 223–238.

[5] Boneh Dan, Goh Eu-Jin, Nissim Kobbi. Evaluating 2-dnf formulas on ciphertexts // Theory of Cryptography / Ed. by Joe Kilian. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. — P. 325–341.

[6] Gentry Craig. A Fully Homomorphic Encryption Scheme : Ph. D. thesis / Craig Gentry. — Stanford, CA, USA : Stanford University, 2009.

[7] Halevi Shai. Homomorphic Encryption // Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich / Ed. by Yehuda Lindell. — Cham : Springer International Publishing, 2017. — P. 219–276. — ISBN: 978-3-319-57048-8. — URL: https://doi.org/10.1007/978-3-319-57048-8_5.

[8] Halevi Shai, Shoup Victor. Bootstrapping for helib. — Cryptology ePrint Archive, Report 2014/873. — 2014. — https://eprint.iacr.org/2014/873.

[9] Gentry Craig, Sahai Amit, Waters Brent. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. — Cryptology ePrint Archive, Report 2013/340. — 2013. — https://eprint.iacr.org/2013/340.

[10] Helib. — https://github.com/shaih/HElib.

[11] Brakerski Zvika, Gentry Craig, Vaikuntanathan Vinod. Fully homomorphic encryption without bootstrapping. — Cryptology ePrint Archive, Report 2011/277. — 2011. — https://eprint.iacr.org/2011/277.

[12] Cheon Jung Hee, Kim Andrey, Kim Miran, Song Yongsoo. Homomorphic encryption for arithmetic of approximate numbers. — Cryptology ePrint Archive, Report 2016/421. — 2016. — https://eprint.iacr.org/2016/421.

[13] Microsoft SEAL (release 3.5). — https://github.com/Microsoft/SEAL. — 2020. — apr. — Microsoft Research, Redmond, WA.

[14] Brakerski Zvika. Fully homomorphic encryption without modulus switching from classical gapsvp. — Cryptology ePrint Archive, Report 2012/078. — 2012. — https://eprint.iacr.org/2012/078.

[15] Fan Junfeng, Vercauteren Frederik. Somewhat practical fully homomorphic encryption. — Cryptology ePrint Archive, Report 2012/144. — 2012. — https://eprint.iacr.org/2012/144.

[16] PALISADE Lattice Cryptography Library (release 1.9.2). — https://palisade-crypto.org/. — 2020. — April.

[17] Ducas Léo, Micciancio Daniele. Fhew: Bootstrapping homomorphic encryption in less than a second. — Cryptology ePrint Archive, Report 2014/816. — 2014. — https://eprint.iacr.org/2014/816.

[18] Chillotti Ilaria, Gama Nicolas, Georgieva Mariya, Izabachène Malika. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. — Cryptology ePrint Archive, Report 2016/870. — 2016. — https://eprint.iacr.org/2016/870.

[19] Tfhe. — https://github.com/tfhe/tfhe.

[20] Heaan. — https://github.com/snucrypto/HEAAN.

[21] Lol. — https://github.com/cpeikert/Lol.

[22] Lattigo 1.3.1. — Online: http://github.com/ldsec/lattigo. — 2020. — feb. — EPFL-LDS.

[23] Halevi Shai, Shoup Victor. Algorithms in helib. — Cryptology ePrint Archive, Report 2014/106. — 2014. — https://eprint.iacr.org/2014/106.

[24] node-seal. — https://github.com/morfix-io/node-seal.

[25] libscarab. — https://github.com/hcrypt-project/libScarab.

[26] Fhew. — https://github.com/lducas/FHEW.

[27] Krypto. — https://github.com/kryptnostic/krypto/tree/develop.

[28] Fv-nfllib. — https://github.com/CryptoExperts/FV-NFLlib.

[29] cuhe: Homomorphic and fast. — https://github.com/vernamlab/cuHE.

[30] cufhe. — https://github.com/vernamlab/cuFHE.

[31] cuyashe. — https://github.com/cuyashe-library/cuyashe.

[32] A gpu implementation of fully homomorphic encryption on torus. — https://github.com/nucypher/nufhe.

[33] Costache Anamaria, Smart Nigel P. Which ring based somewhat homomorphic encryption scheme is best? — Cryptology ePrint Archive, Report 2015/889. — 2015. — https://eprint.iacr.org/2015/889.

[34] Costache Anamaria, Laine Kim, Player Rachel. Evaluating the effectiveness of heuristic worst-case noise analysis in fhe. — Cryptology ePrint Archive, Report 2019/493. — 2019. — https://eprint.iacr.org/2019/493.

[35] Albrecht Martin, Chase Melissa, Chen Hao et al. Homomorphic encryption standard. — Cryptology ePrint Archive, Report 2019/939. — 2019. — https://eprint.iacr.org/2019/939.

[36] Albrecht Martin R., Player Rachel, Scott Sam. On the concrete hardness of learning with errors. — Cryptology ePrint Archive, Report 2015/046. — 2015. — https://eprint.iacr.org/2015/046.

[37] Lwe-estimator. — https://bitbucket.org/malb/lwe-estimator.

[38] Arora Sanjeev, Ge Rong. New algorithms for learning in presence of errors // Proceedings of the 38th International Colloquim Conference on Automata, Languages and Programming - Volume Part I. — ICALP'11. — Berlin, Heidelberg : Springer-Verlag, 2011. — P. 403–415.

[39] Schnorr Claus, Euchner M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems // Mathematical Programming. — 1994. — 08. — Vol. 66. — P. 181–199.

[40] Schnorr Claus Peter. Lattice reduction by random sampling and birthday methods // STACS 2003 / Ed. by Helmut Alt, Michel Habib. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2003. — P. 145–156.

[41] Babai László. On lovász' lattice reduction and the nearest lattice point problem (shortened version) // Proceedings of the 2nd Symposium of Theoretical Aspects of Computer Science. — STACS '85. — Berlin, Heidelberg : Springer-Verlag, 1985. — P. 13–20.

[42] Lindner Richard, Peikert Chris. Better key sizes (and attacks) for lwe-based encryption // Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011. — CT-RSA'11. — Berlin, Heidelberg : Springer-Verlag, 2011. — P. 319–339.

[43] Albrecht Martin R. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. — Cryptology ePrint Archive, Report 2017/047. — 2017. — https://eprint.iacr.org/2017/047.

[44] Albrecht Martin R., Fitzpatrick Robert, öpfert Florian G. On the efficacy of solving lwe by reduction to unique-svp. — Cryptology ePrint Archive, Report 2013/602. — 2013. — https://eprint.iacr.org/2013/602.

[45] Micciancio Daniele, Regev Oded. Lattice-based Cryptography // Post-Quantum Cryptography / Ed. by Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2009. — P. 147–191. — ISBN: 978-3-540-88702-7. — URL: https://doi.org/10.1007/978-3-540-88702-7_5.

[46] Archer David, Chen Lily, Cheon Jung et al. Applications of homomorphic encryption. — 2017. — 07.

[47] Assessment of cloud-based health monitoring using homomorphic encryption / Ovunc Kocabas, Tolga Soyata, Jean-Philippe Couderc et al. — 2013. — 10.

[48] Secure large-scale genome-wide association studies using homomorphic encryption / Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, Shafi Goldwasser // Proceedings of the National Academy of Sciences. — 2020. — Vol. 117, no. 21. — P. 11608–11613.
[49] Masters Oliver, Hunt Hamish, Steffinlongo Enrico et al. Towards a homomorphic machine learning big data pipeline for the financial services sector. — Cryptology ePrint Archive, Report 2019/1113. — 2019. — https://eprint.iacr.org/2019/1113.
[50] Troncoso-Pastoriza Juan Ramón, González-Jiménez Daniel, Pérez-González Fernando. Fully private noninteractive face verification // IEEE Transactions on Information Forensics and Security. — 2013. — Vol. 8, no. 7. — P. 1101–1114.
[51] Engelsma Joshua J, Jain Anil K, Boddeti Vishnu Naresh. Hers: Homomorphically encrypted representation search // arXiv preprint arXiv:2003.12197. — 2020.
[52] Gong Sixue, Boddeti Vishnu Naresh, Jain Anil K. On the intrinsic dimensionality of image representations // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2019. — P. 3987–3996.
[53] Abadi Martín, Agarwal Ashish, Barham Paul et al. TensorFlow: Large-scale machine learning on heterogeneous systems. — 2015. — Software available from tensorflow.org. URL: https://www.tensorflow.org/.
[54] Boddeti Vishnu Naresh. Secure face matching using fully homomorphic encryption // 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS) / IEEE. — 2018. — P. 1–10.
[55] Maalouf Maher. Logistic regression in data analysis: An overview // International Journal of Data Analysis Techniques and Strategies. — 2011. — 07. — Vol. 3. — P. 281–299.
[56] Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles / Xin Jin, Anbang Xu, Rongfang Bie, Ping Guo. — Vol. 3916. — 2006. — 04. — P. 106–115.
[57] The alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus / Ahmad Al Badawi, Jin Chao, Jie Lin et al. // arXiv preprint arXiv:1811.00778. — 2018.
[58] High-performance fv somewhat homomorphic encryption on gpus: An implementation using cuda / Ahmad Al Badawi, Bharadwaj Veeravalli, Chan Fook Mun, Khin Mi Mi Aung // IACR Transactions on Cryptographic Hardware and Embedded Systems. — 2018. — P. 70–95.
[59] Lopez-Alt Adriana, Tromer Eran, Vaikuntanathan Vinod. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. — Cryptology ePrint Archive, Report 2013/094. — 2013. — https://eprint.iacr.org/2013/094.
[60] Sharma Tannishk. E-voting using homomorphic encryption scheme // International Journal of Computer Applications. — 2016. — 05. — Vol. 141. — P. 14–16.
[61] Helios. — https://github.com/benadida/helios-server.

On the Usage of Fully Homomorphic Encryption Libraries

A. Garazha, I. Gerasimov, M. Nikolaev, I. Chizhov

***Abstract*—Fully homomorphic encryption allows computation to be performed on encrypted data without knowing or learning the decryption key. Therefore this technology can be extremely useful for storing and processing personal data. Due to the great interest in this technology, many software tools and libraries are now known to support fully homomorphic encryption. However, this field of cryptography is still relatively young. Standards and guidelines for using fully homomorphic encryption schemes are still under development. Thus, when using these libraries, it is necessary to pay attention to the cryptographic strength of the used schemes to avoid significant information security risks. We consider the issues of the practical application of fully homomorphic encryption schemes, including the choice of suitable libraries and their initialization parameters to ensure a sufficient security level.**

***Keywords*—Fully Homomorphic Encryption, Outsourced Computation**

References

[1] Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms // FoundationsHAHA of Secure Computation, Academia Press. — 1978. — P. 169–179.

[2] Rivest R. L., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. — 1978. — Vol. 21, no. 2. — P. 120–126. — URL: https://doi.org/10.1145/359340.359342.

[3] El Gamal Taher. A public key cryptosystem and a signature scheme based on discrete logarithms // Proceedings of CRYPTO 84 on Advances in Cryptology. — Berlin, Heidelberg : Springer-Verlag, 1985. — P. 10–18.

[4] Paillier Pascal. Public-key cryptosystems based on composite degree residuosity classes // Advances in Cryptology — EUROCRYPT '99 / Ed. by Jacques Stern. — Berlin, Heidelberg : Springer Berlin Heidelberg, 1999. — P. 223–238.

[5] Boneh Dan, Goh Eu-Jin, Nissim Kobbi. Evaluating 2-dnf formulas on ciphertexts // Theory of Cryptography / Ed. by Joe Kilian. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. — P. 325–341.

[6] Gentry Craig. A Fully Homomorphic Encryption Scheme : Ph. D. thesis / Craig Gentry. — Stanford, CA, USA : Stanford University, 2009.

[7] Halevi Shai. Homomorphic Encryption // Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich / Ed. by Yehuda Lindell. — Cham : Springer International Publishing, 2017. — P. 219–276. — ISBN: 978-3-319-57048-8. — URL: https://doi.org/10.1007/978-3-319-57048-8_5.

[8] Halevi Shai, Shoup Victor. Bootstrapping for helib. — Cryptology ePrint Archive, Report 2014/873. — 2014. — https://eprint.iacr.org/2014/873.

[9] Gentry Craig, Sahai Amit, Waters Brent. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. — Cryptology ePrint Archive, Report 2013/340. — 2013. — https://eprint.iacr.org/2013/340.

[10] Helib. — https://github.com/shaih/HElib.

[11] Brakerski Zvika, Gentry Craig, Vaikuntanathan Vinod. Fully homomorphic encryption without bootstrapping. — Cryptology ePrint Archive, Report 2011/277. — 2011. — https://eprint.iacr.org/2011/277.

[12] Cheon Jung Hee, Kim Andrey, Kim Miran, Song Yongsoo. Homomorphic encryption for arithmetic of approximate numbers. — Cryptology ePrint Archive, Report 2016/421. — 2016. — https://eprint.iacr.org/2016/421.

[13] Microsoft SEAL (release 3.5). — https://github.com/Microsoft/SEAL. — 2020. — apr. — Microsoft Research, Redmond, WA.

[14] Brakerski Zvika. Fully homomorphic encryption without modulus switching from classical gapsvp. — Cryptology ePrint Archive, Report 2012/078. — 2012. — https://eprint.iacr.org/2012/078.

[15] Fan Junfeng, Vercauteren Frederik. Somewhat practical fully homomorphic encryption. — Cryptology ePrint Archive, Report 2012/144. — 2012. — https://eprint.iacr.org/2012/144.

[16] PALISADE Lattice Cryptography Library (release 1.9.2). — https://palisade-crypto.org/. — 2020. — April.

[17] Ducas Léo, Micciancio Daniele. Fhew: Bootstrapping homomorphic encryption in less than a second. — Cryptology ePrint Archive, Report 2014/816. — 2014. — https://eprint.iacr.org/2014/816.

[18] Chillotti Ilaria, Gama Nicolas, Georgieva Mariya, Izabachène Malika. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. — Cryptology ePrint Archive, Report 2016/870. — 2016. — https://eprint.iacr.org/2016/870.

[19] Tfhe. — https://github.com/tfhe/tfhe.

[20] Heaan. — https://github.com/snucrypto/HEAAN.

[21] Lol. — https://github.com/cpeikert/Lol.

[22] Lattigo 1.3.1. — Online: http://github.com/ldsec/lattigo. — 2020. — feb. — EPFL-LDS.

[23] Halevi Shai, Shoup Victor. Algorithms in helib. — Cryptology ePrint Archive, Report 2014/106. — 2014. — https://eprint.iacr.org/2014/106.

[24] node-seal. — https://github.com/morfix-io/node-seal.

[25] libscarab. — https://github.com/hcrypt-project/libScarab.

[26] Fhew. — https://github.com/lducas/FHEW.

[27] Krypto. — https://github.com/kryptnostic/krypto/tree/develop.

[28] Fv-nfllib. — https://github.com/CryptoExperts/FV-NFLlib.

[29] cuhe: Homomorphic and fast. — https://github.com/vernamlab/cuHE.

[30] cufhe. — https://github.com/vernamlab/cuFHE.

[31] cuyashe. — https://github.com/cuyashe-library/cuyashe.

[32] A gpu implementation of fully homomorphic encryption on torus. — https://github.com/nucypher/nufhe.

[33] Costache Anamaria, Smart Nigel P. Which ring based somewhat homomorphic encryption scheme is best? — Cryptology ePrint Archive, Report 2015/889. — 2015. — https://eprint.iacr.org/2015/889.

[34] Costache Anamaria, Laine Kim, Player Rachel. Evaluating the effectiveness of heuristic worst-case noise analysis in fhe. — Cryptology ePrint Archive, Report 2019/493. — 2019. — https://eprint.iacr.org/2019/493.

[35] Albrecht Martin, Chase Melissa, Chen Hao et al. Homomorphic encryption standard. — Cryptology ePrint Archive, Report 2019/939. — 2019. — https://eprint.iacr.org/2019/939.

[36] Albrecht Martin R., Player Rachel, Scott Sam. On the concrete hardness of learning with errors. — Cryptology ePrint Archive, Report 2015/046. — 2015. — https://eprint.iacr.org/2015/046.

[37] Lwe-estimator. — https://bitbucket.org/malb/lwe-estimator.

[38] Arora Sanjeev, Ge Rong. New algorithms for learning in presence of errors // Proceedings of the 38th International Colloquim Conference on Automata, Languages and Programming - Volume Part I. — ICALP'11. — Berlin, Heidelberg : Springer-Verlag, 2011. — P. 403–415.

[39] Schnorr Claus, Euchner M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems // Mathematical Programming. — 1994. — 08. — Vol. 66. — P. 181–199.

[40] Schnorr Claus Peter. Lattice reduction by random sampling and birthday methods // STACS 2003 / Ed. by Helmut Alt, Michel Habib. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2003. — P. 145–156.

[41] Babai László. On lovász' lattice reduction and the nearest lattice point problem (shortened version) // Proceedings of the 2nd Symposium of Theoretical Aspects of Computer Science. — STACS '85. — Berlin, Heidelberg : Springer-Verlag, 1985. — P. 13–20.

[42] Lindner Richard, Peikert Chris. Better key sizes (and attacks) for lwe-based encryption // Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011. — CT-RSA'11. — Berlin, Heidelberg : Springer-Verlag, 2011. — P. 319–339.

[43] Albrecht Martin R. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. — Cryptology ePrint Archive, Report 2017/047. — 2017. — https://eprint.iacr.org/2017/047.
[44] Albrecht Martin R., Fitzpatrick Robert, öpfert Florian G. On the efficacy of solving lwe by reduction to unique-svp. — Cryptology ePrint Archive, Report 2013/602. — 2013. — https://eprint.iacr.org/2013/602.
[45] Micciancio Daniele, Regev Oded. Lattice-based Cryptography // Post-Quantum Cryptography / Ed. by Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2009. — P. 147–191. — ISBN: 978-3-540-88702-7. — URL: https://doi.org/10.1007/978-3-540-88702-7_5.
[46] Archer David, Chen Lily, Cheon Jung et al. Applications of homomorphic encryption. — 2017. — 07.
[47] Assessment of cloud-based health monitoring using homomorphic encryption / Ovunc Kocabas, Tolga Soyata, Jean-Philippe Couderc et al. — 2013. — 10.
[48] Secure large-scale genome-wide association studies using homomorphic encryption / Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, Shafi Goldwasser // Proceedings of the National Academy of Sciences. — 2020. — Vol. 117, no. 21. — P. 11608–11613.
[49] Masters Oliver, Hunt Hamish, Steffinlongo Enrico et al. Towards a homomorphic machine learning big data pipeline for the financial services sector. — Cryptology ePrint Archive, Report 2019/1113. — 2019. — https://eprint.iacr.org/2019/1113.
[50] Troncoso-Pastoriza Juan Ramón, González-Jiménez Daniel, Pérez-González Fernando. Fully private noninteractive face verification // IEEE Transactions on Information Forensics and Security. — 2013. — Vol. 8, no. 7. — P. 1101–1114.
[51] Engelsma Joshua J, Jain Anil K, Boddeti Vishnu Naresh. Hers: Homomorphically encrypted representation search // arXiv preprint arXiv:2003.12197. — 2020.
[52] Gong Sixue, Boddeti Vishnu Naresh, Jain Anil K. On the intrinsic dimensionality of image representations // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2019. — P. 3987–3996.
[53] Abadi Martín, Agarwal Ashish, Barham Paul et al. TensorFlow: Large-scale machine learning on heterogeneous systems. — 2015. — Software available from tensorflow.org. URL: https://www.tensorflow.org/.
[54] Boddeti Vishnu Naresh. Secure face matching using fully homomorphic encryption // 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS) / IEEE. — 2018. — P. 1–10.
[55] Maalouf Maher. Logistic regression in data analysis: An overview // International Journal of Data Analysis Techniques and Strategies. — 2011. — 07. — Vol. 3. — P. 281–299.
[56] Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles / Xin Jin, Anbang Xu, Rongfang Bie, Ping Guo. — Vol. 3916. — 2006. — 04. — P. 106–115.
[57] The alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus / Ahmad Al Badawi, Jin Chao, Jie Lin et al. // arXiv preprint arXiv:1811.00778. — 2018.
[58] High-performance fv somewhat homomorphic encryption on gpus: An implementation using cuda / Ahmad Al Badawi, Bharadwaj Veeravalli, Chan Fook Mun, Khin Mi Mi Aung // IACR Transactions on Cryptographic Hardware and Embedded Systems. — 2018. — P. 70–95.
[59] Lopez-Alt Adriana, Tromer Eran, Vaikuntanathan Vinod. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. — Cryptology ePrint Archive, Report 2013/094. — 2013. — https://eprint.iacr.org/2013/094.
[60] Sharma Tannishk. E-voting using homomorphic encryption scheme // International Journal of Computer Applications. — 2016. — 05. — Vol. 141. — P. 14–16.
[61] Helios. — https://github.com/benadida/helios-server.