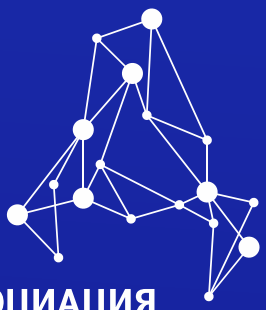


АССОЦИАЦИЯ
БОЛЬШИХ ДАННЫХ

КОНФИДЕНЦИАЛЬНЫЕ ВЫЧИСЛЕНИЯ И ДОВЕРЕННЫЕ СРЕДЫ ИСПОЛНЕНИЯ

Homomorphic Encryption (HE)

Москва
2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 5

ОБЗОР ТЕХНОЛОГИИ 5
История развития 5
Классификация HE-схем 5
Формальное описание 6
Общие принципы гомоморфного шифрования на основе LWE и RLWE 6
Структрура и кодирование шифртекста 7
Гомоморфные операции и рост шума 8
Переход от ограниченного к полному гомоморфному шифрованию .. 8
Пример современной FHE-схемы 8

ПРИМЕРЫ ПРАКТИЧЕСКОГО ПРИМЕНЕНИЯ 11
Примеры применения частичного гомоморфного шифрования 11
Система дистанционного электронного голосования ЦИК РФ на основе схемы EC-Elgamal 11
Схема Пайе в VFL 11
Схема Пайе в HFL 12

Примеры применения полностью гомоморфного шифрования 13
Конфиденциальные вычисления с использованием гомоморфного шифрования 13
Конфиденциальный инференс на недоверенном сервере 15
Обучение моделей на зашифрованных данных: возможности и ограничения FHE 16

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ТЕХНОЛОГИИ 18

МОДЕЛЬ РИСКОВ ГОМОМОРФНОГО ШИФРОВАНИЯ 19
Контекст и поверхность атак 19
Модель рисков для частично гомоморфного шифрования 19
Связь между длиной ключа и уровнем криптостойкости 19
Оценка криптографической сложности 20
Современные возможности атакующих 21
Операционные факторы риска 23
Иллюстративный пример корпоративная система с HSM 23
Расчет совокупного риска 24
Ключевые ограничения предложенного описания 24
Практические рекомендации 24

СОДЕРЖАНИЕ

Модель рисков для FHE 24
От классической криптографии к криптографии на решетках 25
Классификация атак на FHE 25
Операционные риски, специфичные для FHE 25
Управление массивными ключевыми структурами 25
Усиление рисков побочных каналов 25
Сложности корректной реализации 25
Особенности различных схем FHE 27
CKKS: сила и слабость приближенной арифметики 27
TFHE: скорость через специализацию 27

Выводы и рекомендации 27

ЮРИДИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ТЕХНОЛОГИИ 28

ПЕРСПЕКТИВЫ ПРИМЕНИМОСТИ ТЕХНОЛОГИИ HE 29
Общие тезисы о применимости 29
Направления дальнейших исследований и ближайшие задачи 29

ВЫВОДЫ 30

ОБ АВТОРАХ ДОКЛАДА 31

ИСТОЧНИКИ 32

ВВЕДЕНИЕ

Конфиденциальность данных в задачах машинного обучения подразумевает обеспечение безопасности обрабатываемой информации как на этапе обучения моделей, так и в ходе их последующего инференса (использования). Требования к системам, работающим на этих стадиях, могут существенно различаться: если в процессе обучения увеличение вычислительных затрат вполне допустимо (чтобы обеспечить высокий уровень приватности), то на этапе инференса ресурсоемкость вычислений имеет критически важное значение, так же как и длительность задержки ответа. Эти различия приводят к появлению дополнительных требований в отношении механизмов защиты.

Развитие облачных технологий кардинально изменило подход к хранению, обработке и передаче данных: вместо использования локальных серверов и физической инфраструктуры компании могут арендовать вычислительные ресурсы у провайдеров, таких как Yandex Cloud, Amazon Web Services (AWS), Microsoft Azure, Google Cloud и др. Такой подход позволяет быстро и гибко изменять объемы потребляемых ресурсов, получать доступ к данным и приложениям из любой географической точки и при этом экономить средства благодаря отказу от затрат на покупку и обслуживание собственных серверов. Кроме того, за счет применения предлагаемых провайдерами готовых сервисов для работы с искусственным интеллектом (ИИ), большими данными, Интернетом вещей (IoT), ускоряется разработка новых систем и сервисов.

Вместе с тем остается проблема обеспечения конфиденциальности данных: поскольку облачный оператор технически имеет доступ к данным, могут нарушаться коммерческие и регуляторные требования к работе с данными, сформулированные в ФЗ-152, GDPR, HIPAA и других законодательных актах и стандартах. Кроме того, нельзя полностью исключить угрозы неправомерного получения доступа к данным со стороны третьих лиц во время обработки в облаке, когда данные расшифровываются, даже если они хранятся в зашифрованном виде.

Одним из наиболее перспективных методов защиты данных при использовании облачных вычислений является гомоморфное шифрование (Homomorphic Encryption, HE) — технология, позволяющая выполнять вычисления над зашифрованными данными без их расшифрования. Это делает возможным обучение и инференс моделей машинного обучения на стороне провайдера без доступа к исходным данным.

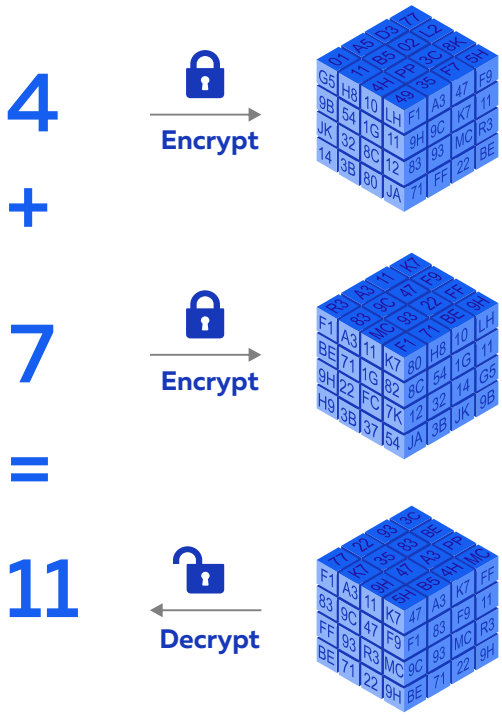
Однако с точки зрения теоретической криптографии большинство HE-схем обеспечивают лишь CPA-устойчивость — защищают от пассивного наблюдателя, имеющего доступ к шифртекстам, но при этом не обладают достаточной стойкостью по модели CCA2, поскольку злоумышленник может модифицировать расшифрование или запрашивать его адаптивно. Следовательно, для применения HE требуется строгий контроль над тем, как используются и обрабатываются зашифрованные данные, особенно в сценариях с активной угрозой.

Как бы то ни было, крупнейшие технологические и исследовательские игроки (IBM, Google, DARPA и др.) рассматривают HE как стратегически важное направление в области машинного обучения с сохранением приватности (Privacy-Preserving Machine Learning, PPML). Это подтверждается как масштабами инвестиций в HE, так и включением его в перспективные государственные программы. Впрочем, HE, так же как и другие технологии конфиденциальных вычислений, такие как MPC и TEE, имеет свои ограничения, узкие области применения и специфические риски.

КОНФИДЕНЦИАЛЬНЫЕ ВЫЧИСЛЕНИЯ И ДОВЕРЕННЫЕ СРЕДЫ ИСПОЛНЕНИЯ
Homomorphic Encryption (HE)

ОБЗОР ТЕХНОЛОГИИ

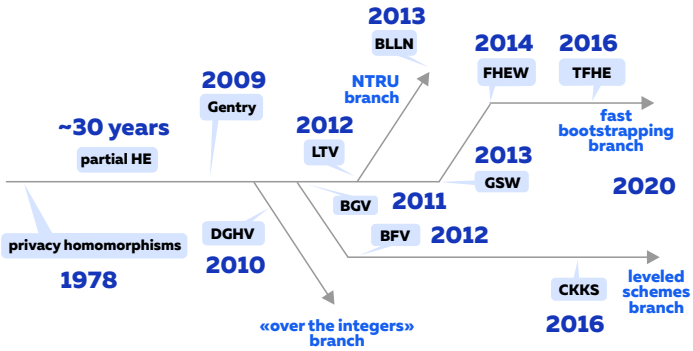
Гомоморфное шифрование — это форма шифрования данных, позволяющая производить некоторые вычислительные операции над зашифрованными данными так, что результат после расшифрования совпадает с результатом операций над исходными незашифрованными данными.



История развития

Термин «гомоморфное шифрование» был введен в 1978 г. [1] авторами алгоритма RSA, отметившими его гомоморфность относительно операции умножения и сформулировавшими смелое предположение о возможности выполнения произвольных операций над зашифрованными данными без их расшифрования. Однако задача создания криптосистемы, гомоморфной относительно и операции сложения, и операции умножения, оставалась нерешенной более 30 лет, несмотря на усилия множества математиков, которым приходилось довольствоваться лишь промежуточными успехами. Так, в 1999 г. была предложена криптосистема [2], гомоморфная относительно сложения, а в 2005 г. — криптосистема [3], основанная на применении билинейных спариваний в группах точек эллиптических кривых, позволявшая выполнять неограниченное количество операций сложения и одну операцию умножения шифрзначений. И только в 2009 г. Крейг Джентри предложил первую схему полностью гомоморфного шифрования [4], основанную на сложных задачах теории решеток.

В дальнейшем на основе работы [4] были предложены новые конструкции полностью гомоморфных схем, которые позволили значительно повысить их эффективность. Например, производительность реализации HE в библиотеке HELib C++, разработанной IBM в 2018 г., в сто миллионов раз выше, чем у оригинального алгоритма Джентри. Тем не менее HELib замедляет обработку шифрованных данных в миллион раз по сравнению с операциями над незашифрованными данными.



Классификация HE-схем

В настоящее время схемы гомоморфного шифрования разделяются на два основных направления:

- Схемы частично гомоморфного шифрования (Partly Homomorphic Encryption, PHE), позволяющие многократно выполнять над шифрзначениями лишь одну из операций умножения (схемы RSA, Elgamal и пр.) и сложения (схемы EC-Elgamal, Пайе, Бенало).

- Схемы полностью гомоморфного шифрования (Fully Homomorphic Encryption, FHE), позволяющие многократно выполнять над шифрзначениями операции сложения и умножения в произвольном порядке (единственным ограничением на количество операций является величина допустимой ошибки). К современным FHE-схемам относятся CKKS, TFHE и пр.

Иногда в отдельную группу выделяют схемы ограниченного гомоморфного шифрования (Somewhat Homomorphic Encryption, SHE), позволяющие выполнять над шифрзначениями обе операции в произвольном порядке, но со строгим ограничением числа применений одной или обеих операций. Примером подобной схемы может служить схема Боне-Го-Ниссима, поддерживающая неограниченное число сложений шифрзначений и одно умножение в произвольном порядке следования операций.

Формальное описание

Пусть k – ключ шифрования, t – открытый текст, $E_k(t)$ – функция зашифрования, $D_k(E_k(t))$ – функция расшифрования. Тогда данная криптосистема является гомоморфной относительно операции $\star \in \{+,\times\}$, если для любых открытых текстов t_1 и t_2 выполняется равенство:

$$t_1 \star t_2 = D_k(E_k(t_1) \circ E_k(t_2)),$$

где $\circ$ – некоторая операция над шифртекстами, результат применения которой к шифртекстам статистически неотличим от шифртекста.

В таком случае система шифрования является:

- **мультипликативно гомоморфной**, если

$$t_1 \cdot t_2 = D_k(E_k(t_1) \otimes E_k(t_2)),$$

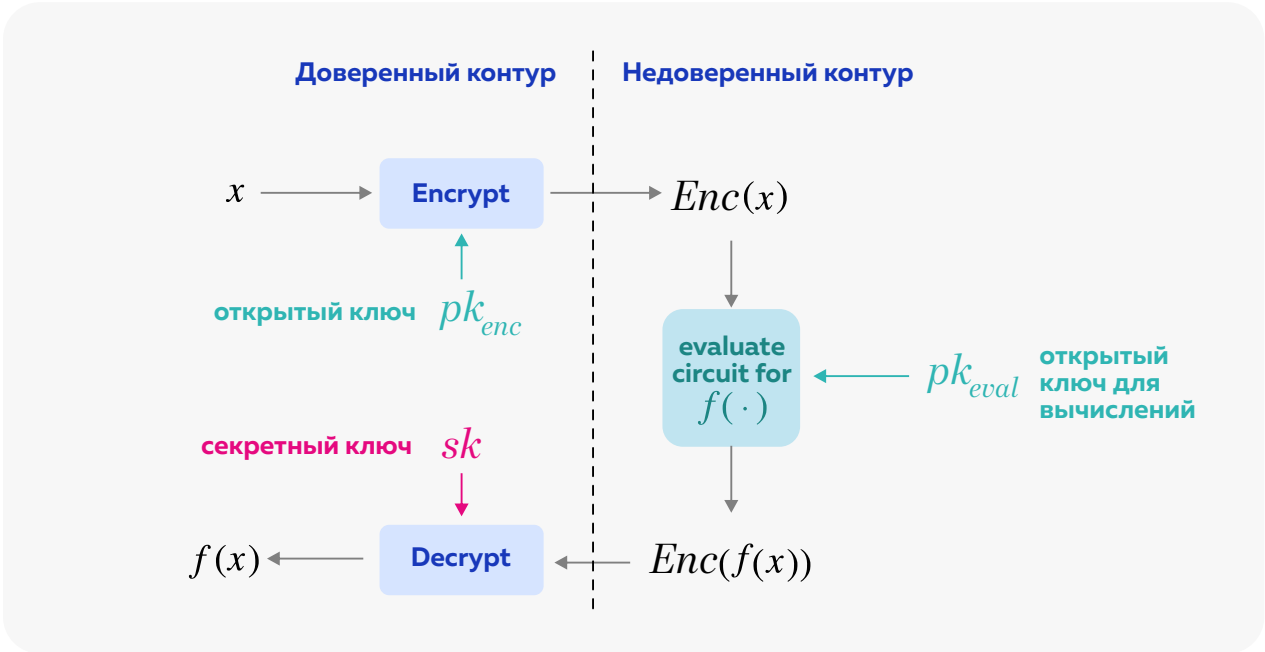
где $\otimes$ – полиномиально вычислимая операция над шифртекстами, результат которой есть корректный шифртекст для значения $t_1 \cdot t_2$;

- **аддитивно гомоморфной**, если

$$t_1 + t_2 = D_k(E_k(t_1) \oplus E_k(t_2)),$$

где $\oplus$ – полиномиально вычислимая операция над шифртекстами, результат которой есть корректный шифртекст для значения $t_1 + t_2$;

- **полностью гомоморфной**, если выполняются оба эти равенства.



FHE-схемы обеспечивают возможность выполнения произвольных вычислений над зашифрованными данными.

Пусть некоторая схема шифрования гомоморфна относительно операций сложения и умножения в поле Галуа $GF(2)$. Тогда, если интерпретировать элементы $GF(2)$ как биты, то:

1) сложение в $GF(2)$ будет эквивалентно операции XOR над входными битами;

2) умножение в $GF(2)$ соответствует логической операции AND.

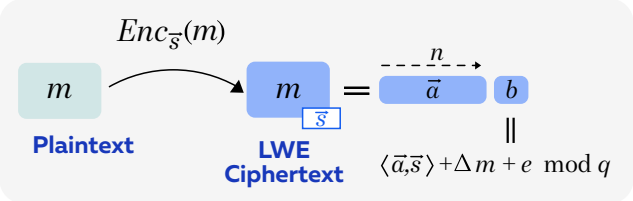
Таким образом, если схема шифрования гомоморфна относительно обеих операций в $GF(2)$, а шифртексты представляют собой зашифрованные биты, то с помощью гомоморфных операций можно вычислять логические функции AND и XOR над этими битами. Из полноты системы {AND, XOR} для класса булевых функций следует, что существует возможность вычисления произвольной булевой функции над зашифрованными входными битами с использованием только гомоморфных операций. Таким образом, достаточно создать схему, гомоморфную относительно сложения и умножения в любом конечном поле $GF(p)$, где p — простое число, поскольку операции в поле $GF(2)$ можно эмулировать через операции в $GF(p)$.

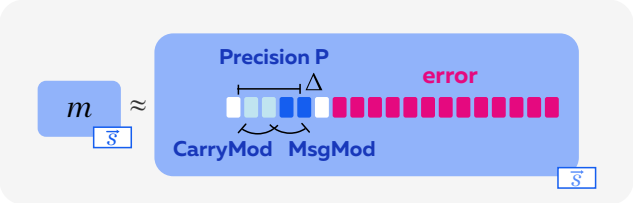
Общие принципы гомоморфного шифрования на основе LWE и RLWE

Одними из наиболее активно применяемых криптографических оснований для построения полностью гомоморфных схем (FHE) являются задача обучения с ошибками (Learning With Errors, LWE) [5] и ее кольцевая модификация Ring-LWE (RLWE) [6]. Эти задачи обладают рядом свойств, обеспечивающих как теоретическую стойкость (включая отсутствие на текущий момент эффективных квантовых алгоритмов их решения), так и удобство реализации арифметических операций над зашифрованными данными. Именно это делает их подходящими для использования в FHE-схемах.

Структура и кодирование шифртекста

Перед шифрованием сообщение m преобразуется в числовую форму, совместимую с модулем q, в котором работает схема. При этом значение масштабируется и размещается в числовом представлении таким образом, чтобы оставить резерв для шума, возникающего в ходе вычислений. Хотя шифртекст формально является элементом кольца (вектором или полиномом), для целей анализа можно интерпретировать его как структуру, состоящую из следующих логических компонентов:





- **полезная нагрузка** (message region): часть значений, соответствующая зашифрованному сообщению (обычно сообщение размещают в старших битах или масштабируют для устойчивости к шуму);
- **шум** (error region): область, в которую при шифровании и последующих гомоморфных операциях вносится гауссовский шум;
- **запас по шуму** (noise margin/headroom): оставшееся пространство между текущим уровнем шума и пределом, при котором расшифрование становится некорректным (эта зона служит для амортизации накопления ошибок округления и арифметических переносов при вычислениях).

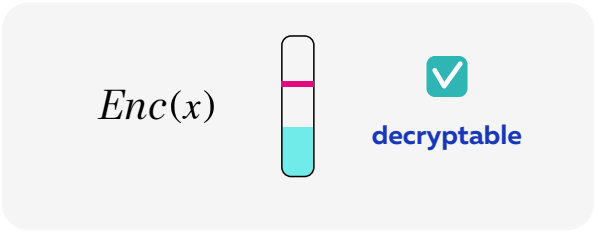
Это разделение не отражается явно в байтовой структуре шифртекста, но используется аналитически — как модель, помогающая оценивать корректность вычислений и проектировать схемы.

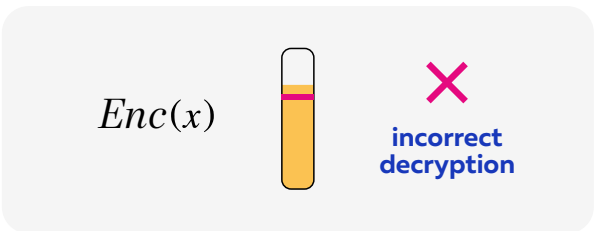
Гомоморфные операции и рост шума

Схемы на основе LWE и RLWE допускают выполнение базовых арифметических операций над зашифрованными данными:

- **сложение шифртекстов:** уровень шума растет аддитивно — суммируются шумы исходных шифртекстов;
- **умножение на константу:** шум масштабируется пропорционально величине множителя;
- **умножение двух шифртекстов** (внешнее произведение): шум возрастает значительно сильнее. В RLWE также увеличивается степень результирующего полинома, поэтому требуется выполнение релинеаризации (relinearization) — процедуры, возвращающей шифртекст к фиксированной размерности.

Все операции сохраняют корректность до тех пор, пока результирующий уровень шума остается в пределах допустимого, то есть существенно меньше модуля или соответствующего кольцевого аналога. Эта особенность влечет ограничение глубины возможных вычислений и необходимость применения механизмов управления шумом.

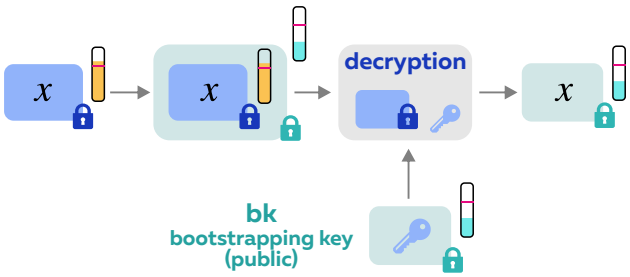




Переход от ограниченного к полному гомоморфному шифрованию

Схемы без дополнительных механизмов управления шумом могут выполнять лишь ограниченное число гомоморфных операций и классифицируются как схемы SHE. Для преодоления этого ограничения применяется метод бутстраппинга (bootstrapping) — ключевой техники, превращающей схему SHE в FHE.

Бутстраппинг — это процедура, в ходе выполнения которой схема гомоморфно расшифровывает шифртекст внутри самой себя, используя зашифрованный секретный ключ, а затем повторно шифрует результат. Это позволяет сбросить накопившийся шум и получить «освеженный» шифртекст, эквивалентный исходному, но пригодный для дальнейших вычислений.



С момента появления первой схемы Джентри идея бутстраппинга стала центральной идеей развития практических реализаций FHE-систем. Благодаря теоретическим и инженерным оптимизациям время выполнения бутстраппинга и объемы используемых ключей значительно сократились, что сделало возможным применение FHE в реальных задачах.

Пример современной FHE-схемы

Одним из наиболее перспективных для применения на практике примеров современных схем полностью гомоморфного шифрования является схема TFHE (Torus Fully Homomorphic Encryption) [7], которая основана на применении задач LWE и GLWE (General Learning With Errors) над дискретизированным тором. Эта схема представляет собой эволюцию схемы FHEW (Fully Homomorphic Encryption Worst-case) [8]. В отличие от других FHE-схем, TFHE была изначально ориентирована на булевы

вычисления, но позже с помощью так называемого **программируемого бутстраппинга** (programmable bootstrapping) была расширена до поддержки целых чисел и даже приближенных действительных значений.

Ключевой особенностью TFHE является использование тора $T = R/Z$, представленного в виде дробей $\frac{a}{q} mod\ 1$, где арифметика сводится к вычислениям по модулю q. Это позволяет эффективно кодировать данные с высокой точностью, сохраняя возможность операций над ними в зашифрованном виде.

Бутстраппинг в TFHE используется как базовый вычислительный примитив: он не только сбрасывает накопленный шум в шифртексте, но и позволяет гомоморфно применять к зашифрованному сообщению произвольную унарную функцию — программируемый бутстраппинг.

Шифрование в TFHE реализуется на основе схем TLWE (Torus LWE) и TGLWE (Torus Galois LWE).

TLWE

Сообщение μ кодируется как элемент из подгруппы $T_p = \left\{0, \frac{1}{p}, ..., \frac{p-1}{p}\right\} \subset T_q$, где $q = 2^{\Omega}$.

Генерируется вектор маски $a \in T_q^n$ и шум $e \leftarrow \chi \subset T_q$, после чего формируется шифртекст:

$$c = (a, b =< a, s > + \mu + e) \in T_q^{n+1}.$$

Расшифрование производится посредством нахождения значения

$$\mu^* = (b - < a, s >) \quad \Rightarrow \quad \mu = round_p(\mu^*),$$

где операция округления выбирает ближайшее значение из T_p при условии, что $|e| < \frac{1}{2p}$.

TGLWE

Для упаковки (packing) нескольких значений используется TGLWE — полиномиальный вариант TLWE, в котором коэффициентами полинома $\mu(x)$ являются элементы T_q, а шифрование производится над кольцом $T_q[X]/(X^N + 1)$.

TFHE поддерживает следующие типы операций: **сложение шифртекстов**, умножение на константу, **булевы операции** (с помощью программируемого бутстраппинга) и **внешнее умножение** (умножение элементов разных алгебраических структур с сохранением гомоморфных свойств шифрования; эта операция реализуется с помощью схемы TGSW и требует контроля роста шума).

Бутстраппинг в TFHE реализуется поэтапно и включает три ключевых шага. На практике он применяется для гомоморфного вычисления функции $f(m)$, заданной в виде таблицы значений (Lookup Table, LUT) — структуры, содержащей выходы функции, заранее вычисленные для всех возможных входов. Такой подход позволяет эффективно применять произвольные унарные функции к зашифрованному сообщению, не раскрывая само значение m.

Этапы бутстраппинга:

- **вращение вслепую** (blind rotation) — гомоморфный сдвиг LUT в зашифрованном виде (фактически выбор значения функции $f(m)$ по зашифрованному входу);
- **отбор образцов** (sample extraction) — извлечение результата из полиномиального представления (GLWE) в формат TLWE;
- **переключение ключей** (key switching) — преобразование полученного шифртекста под целевой секретный ключ для продолжения вычислений или передачи результата.

Возможность гомоморфного применения произвольной унарной функции $f: T_q \to T_q$ к зашифрованному сообщению в процессе бутстраппинга позволяет естественным образом реализовывать в зашифрованной форме вычислительные конструкции, включающие условные переходы, функции активации или логические

преобразования. Данный механизм лежит в основе парадигмы функциональных схем (functional circuits), согласно которой любая многомерная функция (в соответствии с теоремой Колмогорова) может быть выражена через композицию сложений и унарных преобразований. В этой связи TFHE представляется перспективным вариантом для построения вычислительных графов и моделей, работающих над зашифрованными данными.

Результаты оценки производительности TFHE продемонстрированы на задаче инференса нейросетей разной глубины (NN-20, NN-50, NN-100) при 128-битном уровне криптостойкости [9]. В таблице ниже приведено сравнение времени инференса тех же моделей применительно к данным в открытом виде и к зашифрованным данным как на локальной машине (PC), так и в облаке (AWS).

	Открытые данные	Зашифрованные данные	
	PC	PC	AWS
NN-20	0,17 мс	115,52 с	17,96 с
NN-50	0,20 мс	233,55 с	37,69 с
NN-100	0,33 мс	481,61 с	69,32 с

Из таблицы видно, что при переходе к вычислениям над зашифрованными данными время выполнения увеличивается на несколько порядков. Однако даже для глубокой модели (NN-100) полное зашифрованное выполнение на платформе AWS занимает менее 70 секунд — это делает подход практически применимым в задачах приватного инференса. TFHE демонстрирует линейный рост времени по мере увеличения глубины сети, что подтверждает предсказуемую масштабируемость этой схемы.

ПРИМЕРЫ ПРАКТИЧЕСКОГО ПРИМЕНЕНИЯ

Примеры применения частичного гомоморфного шифрования

PHE-схемы находят применение в различных практических задачах. Например, уже невозможно себе представить системы электронного голосования, в которых не использовались бы встроенные частично гомоморфные криптографические механизмы.

Как уже отмечалось в предыдущей статье, для обеспечения конфиденциальности и предотвращения атак, нацеленных на восстановление исходных данных на основе параметров модели, передаваемых в ходе федеративного обучения, протоколы такого рода обучения необходимо комбинировать с другими технологиями конфиденциальных вычислений. В частности, довольно перспективным с точки зрения обеспечения защиты передаваемых параметров как в случае VFL, так и HFL является применение гомоморфного шифрования. Причем для некоторых архитектур моделей бывает достаточно применения частично гомоморфных схем шифрования.

Система дистанционного электронного голосования ЦИК РФ на основе схемы EC-Elgamal

31 августа 2020 г. состоялось публичное тестирование системы дистанционного электронного голосования (ДЭГ) с применением технологии блокчейна, разработанной по заказу ЦИК - РФ. Основные требования, предъявляемые к системе, определены Федеральным законом от 12 июня 2002 г. N67-ФЗ (в ред. от 31 июля 2020 г.) «Об основных гарантиях избирательных прав и права на участие в референдуме граждан Российской Федерации».

Процесс дистанционного голосования включает в себя следующие этапы:

1. Заполнение и отправка бюллетеня:

- **шифрование** бюллетеня по схеме EC-Elgamal, обладающей свойством гомоморфности в отношении операции сложения, что позволяет получить результаты голосования без необходимости расшифрования бюллетеня;
- **доказательство с нулевым разглашением,** использующееся для доказательства корректности содержимого бюллетеня без его расшифрования;
- **электронная подпись** зашифрованных бюллетеней по ГОСТ Р 34.10-2012.

2. Подсчет итогов голосования:

- **гомоморфное сложение** зашифрованных бюллетеней;
- **предварительное частичное расшифрование** итогового суммированного бюллетеня частями закрытого ключа участниками, контролирующими отдельные узлы и серверы подсчета, и получение шифртекстов от каждого участника;
- **сборка закрытого ключа** в Избирательной комиссии и частичное расшифрование итогового суммированного бюллетеня собранным ключом;
- **окончательное суммирование шифртекстов** и получение итогов подсчета;
- **выработка и проверка доказательства с нулевым разглашением,** используемого для доказательства корректности расшифрования итогового суммированного бюллетеня.

3. Аудит:

на данном этапе могут быть выполнены проверки корректности всех этапов протокола.

Схема Пайе в VFL

В качестве примера использования аддитивно-гомоморфной схемы Пайе для обеспечения защиты передаваемых между участниками VFL параметров предлагается рассмотреть обучение модели градиентного бустинга на решающих деревьях [10, 11].

На каждой итерации обучения модель дополняется деревом. Процесс формирования очередного дерева начинается с дерева глубины 0,

состоящего только из корневого узла, с последующим разбиением каждого листа на левого и правого потомков до тех пор, пока не будет достигнута заданная глубина дерева. Для определения правила разбиения текущего узла ищется максимум выражения

$$S = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i+\lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i+\lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i+\lambda}\right] - \gamma, \quad (1)$$

где g_i – первая производная функции потерь (градиенты), h_i – вторая производная функции потерь (гессианы), I_L и I_R – сущности, попавшие, согласно опробуемому правилу разбиения сущностей I узла, в левый и правый потомок соответственно. В результате формирования дерева оптимальный вес листа вычисляется на основе выражения

$$w_j = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i+\lambda},$$

где I_j – сущности, попавшие в лист j.

Из приведенных выражений видно, что определение наилучшего разбиения узла и вычисление оптимального веса листа зависят только от сумм значений градиентов и гессианов. Таким образом, если активная сторона отправит пассивным сторонам градиенты и гессианы в зашифрованном с помощью криптосхемы Пайе виде $\left\{Enc_{Paillier}(g_i),\ Enc_{Paillier}(h_i)\right\}_{i\in I}$, то каждая пассивная сторона для каждого своего варианта разбиения множества I на подмножества I_L и $I_R = I \backslash I_L$ сможет вычислить

$$Enc_{Paillier}(G) = \prod_{i\in I_L} Enc_{Paillier}(g_i),$$

$$Enc_{Paillier}(H) = \prod_{i\in I_L} Enc_{Paillier}(h_i),$$

и отправить полученные шифрзначения активной стороне, которая, расшифровав их, получит множество пар (G, H), на основе которых сможет вычислить максимальное значение выражения (1) для всех участвующих в обучении сторон. Аналогичные действия выполняются для каждого узла каждого дерева совместной модели. В итоге обеспечивается совместное обучение модели градиентного бустинга на решающих деревьях без передачи исходных данных и с гомоморфным шифрованием производной от исходных данных информации, которой стороны обмениваются в процессе обучения модели.

Схема Пайе в HFL

Аддитивно-гомоморфная схема Пайе может использоваться и для защиты параметров в ходе HFL при обучении все той же модели градиентного бустинга на решающих деревьях [12]. В этом случае стороны предварительно согласуют параметры квантильных разбиений по каждому признаку $Q = \left\{\left(q_0^i, q_1^i, \ldots, q_{s_i}^i\right) | i = \underline{1, n}\right\}$, где n – число признаков. Пороговые значения квантильных разбиений могут формироваться, например, с помощью алгоритма DDsketch [13].

Каждая сторона в ходе выполнения вычислений на своих локальных данных выполняет следующие действия:

- формирует гистограммы распределений градиентов и гессианов в соответствии с согласованными границами квантилей;
- вычисляет сумму градиентов и сумму гессианов $G_{b_j} = \sum_{i\in bucket_j} (g_i)$ и $H_{b_j} = \sum_{i\in bucket_j} (H_i)$, попавших в одну корзину гистограммы $bucket_j$;
- зашифровывает с помощью схемы Пайе полученные значения сумм;
- отправляет шифрзначения агрегатору.

Агрегатор, в свою очередь, вычисляет суммы шифрзначений, полученных от всех участников градиентов и гессианов, относящихся к одному

интервалу квантильного разбиения, и отправляет полученные значения обратно всем сторонам. После этого каждая из них расшифровывает итоговые значения сумм градиентов и гессианов и вычисляет с использованием выражения (1) информационный выигрыш для каждого интервала квантильного разбиения, определяя, для какого именно интервала достигается максимум, и формируя разбиение текущего узла на основе признака и порога соответствующего квантильного разбиения. После этого все стороны переходят к анализу следующего узла дерева.

Данный процесс выполняется для каждого узла каждого дерева обучаемой модели до тех пор, пока не будет выполнен какой-либо критерий остановки обучения.

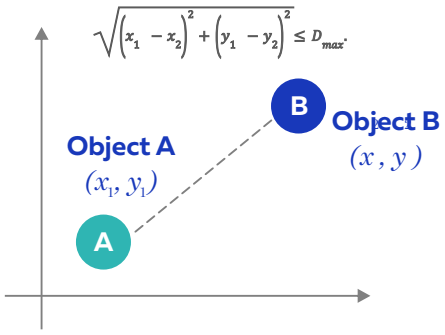
Очевидно, что описанный подход к обучению совместной модели можно применять при наличии как минимум трех сторон. Если же в схеме будут участвовать только две стороны, то каждая из них, зная агрегированные суммы гистограмм, сможет легко восстановить значения гистограмм градиентов и гессианов другой стороны.

Примеры применения полностью гомоморфного шифрования

Конфиденциальные вычисления с использованием гомоморфного шифрования

Пусть имеется два объекта, каждый из которых обладает конфиденциальными координатами в пространстве:

- объект A с координатами (x_1, y_1);
- объект B с координатами (x_2, y_2).



Координаты являются чувствительными данными и не могут быть раскрыты. Однако требуется вычислить евклидово расстояние между объектами и проверить, находится ли оно в пределах допустимого порога D_{max}:

$$d(A,B) = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2} \le D_{max}.$$

Эту задачу можно решить, применив любой из двух способов организации гомоморфных криптосистем.

Вариант использования идентичных ключей основан на использовании схемы CKKS, поддерживающей арифметику с числами с плавающей точкой. Все участники используют один публичный ключ pk, при этом приватный ключ sk хранится у доверенного центра или у получателя. Каждая сторона производит шифрование своих координат на публичном ключе доверенной стороны:

- сторона A: $Enc_{pk}(x_1)$, $Enc_{pk}(y_1)$;
- сторона B: $Enc_{pk}(x_2)$, $Enc_{pk}(y_2)$.

На сервере выполняются следующие вычисления:

$$\Delta x = Enc_{pk}(x_1) - Enc_{pk}(x_2)$$

$$\Delta y = Enc_{pk}(y_1) - Enc_{pk}(y_2)$$

$$Enc_{pk}(d^2) = (\Delta x)^2 + (\Delta y)^2$$

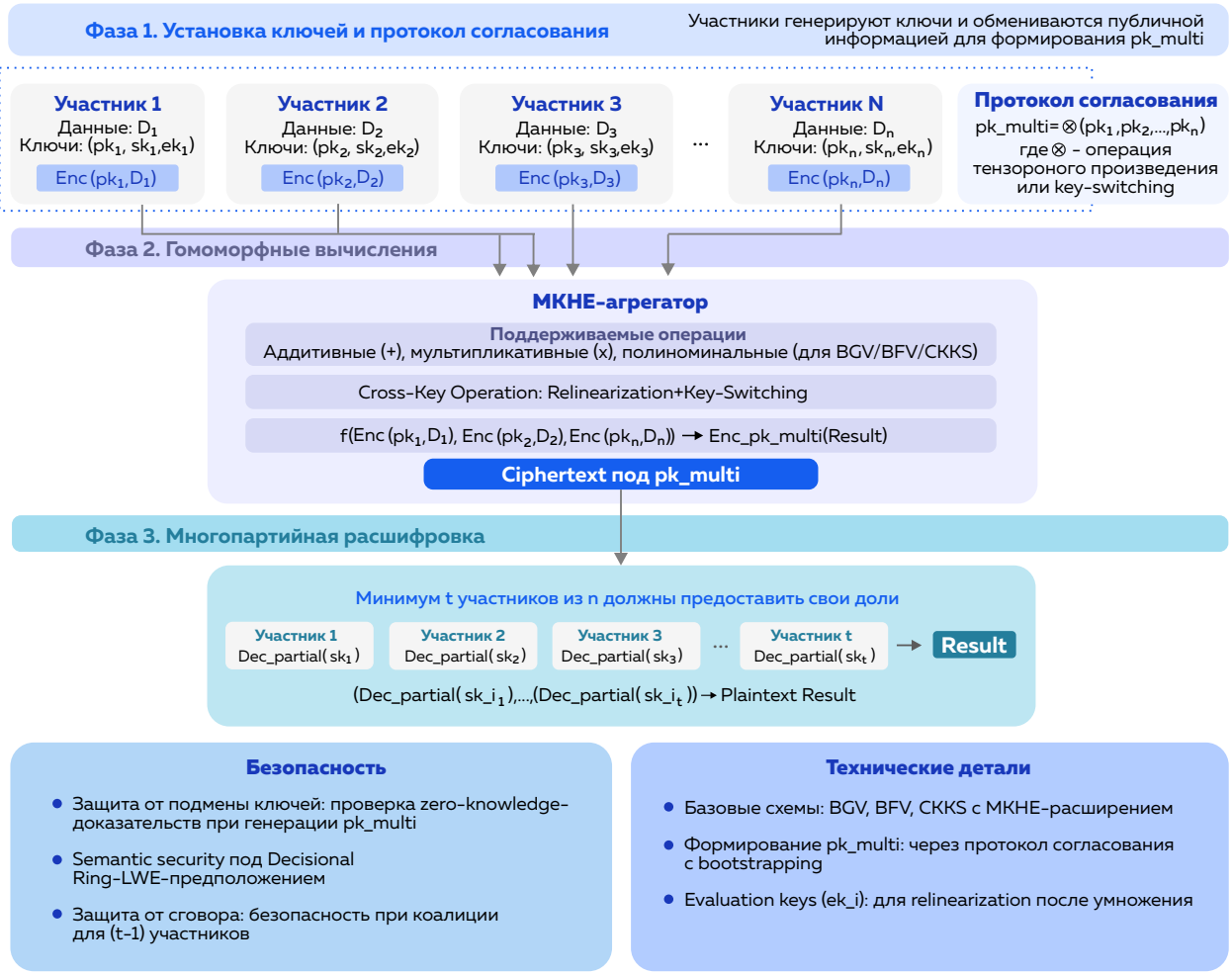
Доверенная сторона получает $Enc_{pk}(d^2)$, расшифровывает и сравнивает результат с $(D_{max})^2$.

При необходимости сторонам направляются предупреждения о выходе за допустимые границы.Главные преимущества этого варианта заключаются в простоте реализации и хорошей поддержке необходимого функционала существующими библиотеками (Microsoft Seal, HEaan), недостатки — в необходимости наличия доверенной стороны и в возможности расшифрования, неподконтрольного участникам.

КОНФИДЕНЦИАЛЬНЫЕ ВЫЧИСЛЕНИЯ И ДОВЕРЕННЫЕ СРЕДЫ ИСПОЛНЕНИЯ
Homomorphic Encryption (HE)

Второй вариант решения предполагает использование мультиключевой схемы гомоморфного шифрования (MKHE) [14], на порядок более сложной в организации и функционировании, с дополнительными операциями (например, MPC) и тесной координацией между участниками с целью расшифрования результатов.

На основе BGV/BFV/CKKS с расширением multykey операций



Предложенные подходы на основе гомоморфного шифрования позволяют эффективно вычислять расстояние между двумя объектами без раскрытия их координат. Это означает, что координаты и маршруты остаются полностью скрытыми, а система обрабатывает только зашифрованные значения и извлекает из них только нужную функцию — квадрат расстояния. Такая схема делает невозможными отслеживание и анализ передвижения объектов.

Подобным образом можно решать широкий класс задач, где необходимо вычислить совместную функцию на основе данных нескольких сторон, не раскрывая при этом сами данные. Таким образом, FHE выступает универсальной технологией для построения вычислений «на доверии без доверия», когда серверу не нужно знать незашифрованные исходные данные, чтобы с ними работать. Это открывает путь к безопасным формам аналитики и кооперации в оперирующих с чувствительными данными отраслях, таких как здравоохранение, финансы, оборона, телеком, транспорт и пр.

КОНФИДЕНЦИАЛЬНЫЕ ВЫЧИСЛЕНИЯ И ДОВЕРЕННЫЕ СРЕДЫ ИСПОЛНЕНИЯ
Homomorphic Encryption (HE)

Примером практического применения FHE может служить южнокорейское приложение Beomdong-I, предназначенное для защиты жертв сталкинга, домашнего насилия и некоторых других форм правонарушений. Система использует гомоморфное шифрование для вычисления расстояния между координатами правонарушителя и жертвы без раскрытия их местоположения. Сервер, работая только с зашифрованными координатами, определяет расстояние, и, если нарушитель приближается на дистанцию ближе допустимой, система автоматически фиксирует инцидент и отправляет предупреждение. Таким образом, FHE может применяться там, где требуется реакция в реальном времени (в том числе в социально значимых задачах) и где необходимы приватность и безопасность без компромиссов.

Конфиденциальный инференс на недоверенном сервере

При решении многих задач (аналитика, статистические вычисления, машинное обучение и пр.) возникает потребность передать чувствительные данные на удаленный сервер для выполнения произвольной обработки. Однако сервер, производящий такую обработку, может не быть доверенным, в связи с чем прямая передача данных на него сопряжена с риском раскрытия персональной или коммерчески значимой информации. Полностью гомоморфное шифрование (FHE) предлагает решение: оно позволяет серверу выполнять вычисления над зашифрованными данными без их расшифрования.

Яркий и практически значимый пример использования FHE — его применение для инференса модели машинного обучения на зашифрованных данных. Такой инференс можно представить в виде последовательности шагов:

1. **Подготовка модели:** модель машинного обучения должна быть адаптирована для работы с FHE. Это означает, что все операции модели должны быть сведены к полиномиальным операциям — сложению и умножению. Нелинейные функции, такие как ReLU или сигмоида, заменяются на их полиномиальные аппроксимации.

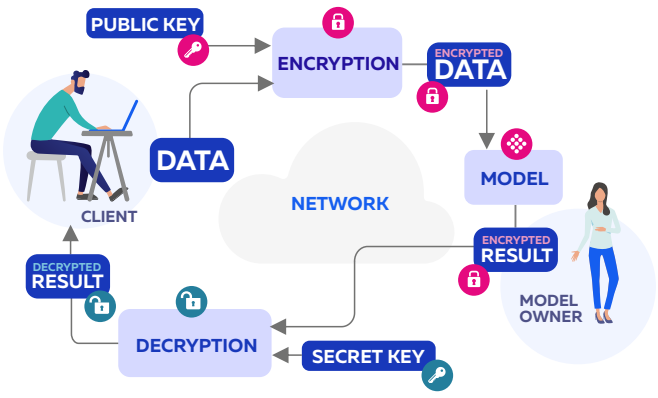
2. **Генерация ключей:** клиент генерирует пару ключей — публичный и приватный. Публичный ключ используется для шифрования данных, а приватный остается у клиента для расшифрования результатов.

3. **Шифрование данных:** клиент шифрует свои данные с использованием публичного ключа и отправляет зашифрованные данные на сервер.

4. **Инференс на сервере:** сервер получает зашифрованные данные и выполняет инференс модели, которая была предварительно адаптирована для работы с FHE. Все вычисления происходят над зашифрованными данными, и сервер не имеет доступа к исходным данным клиента.

5. **Получение результатов:** результат инференса в зашифрованном виде отправляется обратно клиенту.

6. **Расшифрование результатов:** клиент использует свой приватный ключ для расшифрования полученных результатов и получения прогнозов, подготовленных моделью.



Такая архитектура обеспечивает два ключевых преимущества:

- **конфиденциальность данных** — результат инференса сохраняется в зашифрованном виде и только владелец данных может его расшифровать, что исключает раскрытие информации в модели пассивного нарушителя, то есть при отсутствии активного вмешательства в процесс вычислений или расшифрования;

- **безопасность модели** — размещение модели на стороне сервера позволяет ограничить доступ к ее архитектуре и параметрам, что обеспечивает защиту интеллектуальной собственности разработчика.

Такой подход открывает возможность безопасного использования внешних моделей машинного обучения без риска утечки чувствительных данных, однако он сопряжен с рядом технических ограничений:

- **высокая вычислительная нагрузка** — инференс над зашифрованными данными требует существенно больше ресурсов и времени по сравнению с обычными вычислениями;
- **снижение точности** — необходимость замены нелинейных активационных функций на полиномиальные аппроксимации может привести к ухудшению качества предсказаний;
- **сложность внедрения** — подготовка модели и инфраструктуры к работе с FHE требует глубокой модификации вычислительных графов, что невозможно для некоторых архитектур или фреймворков.

Реальный пример защищенного инференса с использованием FHE был реализован в 2019 г. в рамках партнерства IBM и Banco Bradesco — одного из крупнейших банков Бразилии. Проект был направлен на конфиденциальное прогнозирование клиентского поведения на основе чувствительных транзакционных данных. В проекте была использована библиотека HElib, реализующая схему CKKS, что позволило обрабатывать зашифрованные данные с плавающей точкой. Модель линейной регрессии была предварительно обучена и применялась на стороне сервера к зашифрованным входным данным клиента.

- Входной вектор содержал 16 признаков, полученных из набора, включающего 7500 записей и 546 переменных.

  Инференс над зашифрованными данными выполнялся за 5.4 секунды на один запрос при 128-битной криптостойкости.

  Результаты были сопоставимы по точности с результатами модели в открытом виде.

Важно, что на всем протяжении процесса сервер не имел доступа ни к исходным данным, ни к результатам в открытом виде, при этом модель также была защищена от копирования, поскольку оставалась на сервере в зашифрованном виде.

Этот проект продемонстрировал возможность применения FHE-инференса в банковском секторе и заложил основу для создания сервисов аналитики, соответствующих требованиям к защите персональных данных и бизнес-информации.

Обучение моделей на зашифрованных данных: возможности и ограничения FHE

Во многих прикладных сценариях возникает необходимость обучать модели машинного обучения на конфиденциальных пользовательских данных, находящихся в распоряжении клиента или размещенных в распределенной сети. В таких случаях прямая передача данных на внешний сервер или в облако связана с рисками утечки персональной информации, нарушений законодательства и утраты конкурентных преимуществ.

Полностью гомоморфное шифрование (FHE) позволяет проводить обучение модели прямо на зашифрованных данных, не раскрывая при этом данные обучающей выборки. Это направление выходит за рамки классического инференса на зашифрованных данных, приближаясь к более защищенному конфиденциальному машинному обучению (Confidential ML).

Процесс обучения выглядит как последовательность шагов:

1. **Выбор подходящей модели:** обучение на зашифрованных данных пока ограничено следующими простыми интерпретируемыми моделями:

   линейная и логистическая регрессия,

   наивный байесовский классификатор,

   небольшие деревья решений и случайные леса с ограниченной глубиной,

   полиномиальные аппроксимации SVM.

2. Полиномизация модели: все операции обучения должны быть сведены к полиномиальным — сложению и умножению. Нелинейные функции (например, сигмоида) заменяются полиномиальными аппроксимациями.

3. Генерация ключей: клиент генерирует криптографическую пару ключей (публичный и приватный), шифрует обучающую выборку и передает зашифрованный датасет на сервер.

4. Гомоморфное обучение на сервере: сервер выполняет обучение модели без доступа к исходным данным. Используются FHE-библиотеки, поддерживающие гомоморфную арифметику: Zama Concrete ML, Microsoft SEAL, HEaaN, PALISADE и др.

5. Отправка модели клиенту — в зависимости от сценария:

либо клиент получает обученную модель в зашифрованной форме;

либо обученная модель остается на сервере для дальнейшего инференса.

6. Мультиклиентский вариант (Multi-key FHE): если данные распределены между несколькими участниками, то может применяться *multi-key FHE*, позволяющий обучать модели на распределенных шифртекстах без раскрытия данных между участниками.

Показательную реализацию такого подхода продемонстрировала компания CryptoLab, обучившая модель логистической регрессии на зашифрованном финансовом датасете из 400 тыс. записей Корейского кредитного бюро (KCB) с более чем 200 признаками: если в 2019 г. такое обучение требовало около 17 часов вычислений на CPU, то в 2021 г. — менее 20 минут. Сокращение времени было достигнуто благодаря оптимизации функций библиотеки HEaaN, реализующей высокопроизводительные FHE-примитивы.

Этот пример доказывает, что FHE можно применять на практике для полномасштабного обучения линейных моделей на реальных чувствительных данных с сохранением их приватности и без раскрытия их признаков или целевых переменных.

На основе BGV/BFV/CKKS с расширением multykey операций

Модель	Датасет	Обучение (FHE), мин	Точность (FHE)	Снижение качества
Логистическая регрессия	KCB (400K × 200)	~20 (HEaaN)	Близка к открытой	Минимальное
Логистическая регрессия	Breast Cancer	~15–20 (Concrete)	~96,2%	<1%
Линейная регрессия	Boston Housing	~10	RMSE ~6,5	+5–15%

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ТЕХНОЛОГИИ

Преимущества

1. Данные остаются зашифрованными на протяжении всего процесса обработки, что исключает риск утечки в ходе вычислений — это полезно при использовании облаков, в медицине, финансовой и других сферах, где важна приватность.

2. Безопасные аутсорсинговые вычисления: можно передавать зашифрованные данные третьим сторонам (например, облачным провайдерам) без риска раскрытия информации.

3. Помогает соблюдать ФЗ-152, GDPR, HIPAA и другие законы, требующие защиты конфиденциальных данных.

Недостатки

1. Высокая вычислительная сложность: операции на зашифрованных данных требуют на 2–6 порядков больше ресурсов, чем на открытых, что делает HE непрактичным для многих реальных задач.

2. Большие накладные расходы: зашифрованные данные значительно (иногда в 100–1000 раз) больше по объему, чем исходные.

3. Ограниченная функциональность: не все алгоритмы эффективно реализуемы в FHE из-за ограничений на типы операций. Например, нелинейные функции требуют применения сложных обходных путей.

4. Снижение характеристик моделей обучения из-за аппроксимаций нелинейных функций, входящих в архитектуру этих моделей.

5. Сложность реализации: требуются глубокие знания в области криптографии, а ошибки в настройке могут привести к появлению уязвимостей.

6. Требуется специализированное оборудование: для ускорения FHE разрабатываются GPU- и FPGA-ускорители, но до их массового внедрения еще довольно далеко.

МОДЕЛЬ РИСКОВ ГОМОМОРФНОГО ШИФРОВАНИЯ

Безопасность систем HE, как и любых других криптосистем, определяется комплексом факторов: не только математической стойкостью алгоритмов, но и корректностью их реализации в конкретной архитектурной среде. Поверхность атак существенно зависит от контекста применения: одни и те же криптографические примитивы могут демонстрировать разный уровень уязвимости в зависимости от архитектуры системы, сетевой инфраструктуры и модели угроз.

Контекст и поверхность атак

Системы гомоморфного шифрования содержат четыре ключевых компонента с уникальными рисками. Для владельцев данных основная проблема заключается в отсутствии стандартизации параметров HE-схем, что ведет к ошибкам в выборе уровней безопасности и генерации ключей. Особую сложность представляют увеличенные размеры криптоключей по сравнению с используемыми в традиционных системах.

Вычислительные узлы подвержены атакам через побочные каналы (путем анализа времени выполнения и энергопотребления), а также рискам модификации вычислений недобросовестными операторами. Доверенные центры, управляющие распределением ключей, становятся критической точкой отказа: их компрометация через уязвимости аппаратного модуля безопасности (HSM) или протокольные ошибки разрушает всю систему безопасности.

Потребители результатов, проводя их итоговое расшифрование, могут столкнуться с проблемами верификации цепочек вычислений и контроля доступа. Взаимозависимость этих компонентов создает комплексную поверхность атак, где слабое звено снижает уровень защиты всей системы.

Наиболее естественный подход к объединению рисков основан на теории вероятностей для независимых событий: система считается скомпрометированной, если успешна хотя бы одна атака — криптографическая или операционная.

Вероятность того, что произойдет хотя бы одно из событий компрометации, можно оценить по следующей формуле:

$$\Phi_{total} = 1 - (1 - \Phi_{crypto}) \times (1 - \Phi_{oper}) \quad (2)$$

Это выражение, представляющее собой произведение шансов устоять против каждого типа атак, можно интуитивно понимать как общий шанс системы на выживание.

Модель рисков для частично гомоморфного шифрования

Частично гомоморфное шифрование представляет собой более простой и изученный класс криптосистем по сравнению с полностью гомоморфным. Схемы PHE, такие как RSA (гомоморфная по умножению) или Пайе (гомоморфная по сложению), уже десятилетиями применяются на практике, их модели безопасности хорошо изучены и понятны. Ниже приводится эмпирическая количественная модель оценки рисков для PHE, которая послужит основой для сравнения с более сложными схемами FHE.

Связь между длиной ключа и уровнем криптостойкости

При оценке защищенности криптосистемы необходимо разделять два ключевых параметра: длину ключа (L) и уровень криптостойкости (security level) (λ). L — это физический размер ключа в битах (например, 2048 бит для Пайе), тогда как λ характеризует уровень безопасности как количество операций, которое требуется для взлома (2^{λ}). Для схем факторизации (RSA, Пайе) соотношение L и λ нелинейно и определяется оптимальным на текущий момент алгоритмом факторизации и прогрессом вычислительных мощностей. В рекомендациях NIST (стр. 54–55) приводятся следующие соотношения:

Длина ключа L	Уровень криптостойкости λ
1024 бит	~80 бит
2048 бит	~112 бит
3072 бит	~128 бит
7680 бит	~192 бит
15360 бит	~256 бит

Цветом выделена длина ключа, нерекомендуемая к использованию в современных системах криптографической защиты информации.

Оценка криптографической сложности

Φ_{crypto} в формуле (2) количественно выражает вероятность компрометации системы через решение базовой криптографической задачи. В отличие от операционных рисков Φ_{oper}, связанных с практическими аспектами эксплуатации, Φ_{crypto} характеризует вычислительную сложность криптоанализа.

Можно выделить следующие основные векторы криптографических атак:

1. Прямые атаки на математическую основу схемы:

решение базовой задачи (факторизация для RSA/Пайе, LWE/RLWE для FHE);

использование структурных особенностей (например, атаки на Ring-LWE);

квантовые алгоритмы (алгоритм Шора для факторизации, ограниченное применение для решеток).

2. Извлечение информации через криптографические артефакты:

анализ публичных параметров и метаданных;

эксплуатация особенностей схемы (например, утечка через шум в CKKS без LMSS padding);

корреляционные атаки (при многократном использовании).

Теоретическая база оценки Φ_{crypto} опирается на работы [26], [27], [28], [29], [30].

Важное замечание: для новых конструкций (особенно FHE) точная оценка Φ_{crypto} остается открытой проблемой, поэтому применяется консервативный подход с существенным запасом безопасности, например, выбирается $\lambda \geq 128$ бит даже в случае, когда теоретически достаточно λ=112. Мы ограничимся только инженерными аппроксимациями и эвристиками.

В случае PHE безопасность определяется преимущественно длиной ключа, поскольку атаки, как правило, направлены на решение одной конкретной математической проблемы (например, факторизации или дискретного логарифмирования), которая лежит в основе генерации и использования ключа. Таким образом, оценка риска для PHE сводится к определению того, насколько велика эта математическая проблема, чтобы даже самый «грубый» метод атаки — полный перебор (брутфорс) всех возможных ключей — оставался вычислительно невыполнимым в течение приемлемого периода времени, обеспечивая тем самым верхнюю границу при оценке сложности любой потенциальной атаки.

Вероятность успеха для PHE, определяемая брутфорс-атаками (при отсутствии более эффективных атак), определяет верхнюю границу:

$$\Phi_{crypto} \leq P_{brutforce} = 1 - (1 - 2^{-\lambda})^{N} \qquad (3)$$

Здесь:

- λ — уровень криптостойкости;
- C — вычислительная мощность атакующего, измеряемая в количестве проверок ключей-кандидатов в секунду;
- T — временной горизонт атаки в секундах;

- $N = C \times T$ — общее количество ключей, которое атакующий способен проверить за отведенное время.

Данное выражение представляет собой классическую вероятность хотя бы одного успеха в серии независимых испытаний. Для практических оценок формулу (3) можно упростить, принимая допущение, что атакующий способен проверить лишь незначительную долю пространства ключей ($N \ll 2^{\lambda}$). В этом случае разложение в ряд Тейлора дает линейное приближение, смысл которого в том, что вероятность успеха пропорциональна доле исследованного пространства ключей:

$$P_{brutforce} = N / 2^{\lambda} = \frac{C \times T}{2^{\lambda}} \qquad (4)$$

Однако применение (4) имеет существенные ограничения:

- **Предположение о случайном переборе:** модель не учитывает возможные криптоаналитические атаки, способные существенно сократить эффективное пространство поиска.
- **Использование классических вычислений:** появление масштабируемых квантовых компьютеров кардинально изменит ландшафт угроз. Так, алгоритм Гровера позволяет квантовому компьютеру осуществлять поиск в пространстве из 2^{λ} элементов за $O(2^{\lambda/2})$ операций, что эквивалентно уменьшению эффективной длины ключа вдвое.
- **Отсутствие побочных каналов:** реальные системы могут быть уязвимы к атакам по побочным каналам (атаки на основе анализа времени выполнения и энергопотребления), которые обходят криптозащиту.
- **Линейное приближение:** формула (4) дает адекватные оценки только при $N \ll 2^{\lambda}$. Если N приближается к 2^{λ}, то необходимо использовать точную формулу (3).

Приведенные выше методы применимы только для PHE. Они являются инженерным решением и не должны подменять сложные модели, многие из которых пока находятся на стадии исследований.

Современные возможности атакующих

Для получения практической оценки рисков необходимо реалистично оценить вычислительные ресурсы потенциального атакующего. Хотя динамика вычислений и возможности атакующих с каждым годом существенно меняются, а надежных экспериментов мало, можно воспользоваться следующей достаточно грубой оценкой производительности на основе имеющихся сведений.

Рассмотрим криптосхему Пайе с модулем длиной L = 2048 бит, что соответствует криптостойкости $\lambda \approx 112$ бит. Особенность данной схемы заключается в невозможности ее взлома методом полного перебора. Ключ Пайе основан на составном модуле $n = p \times q$, где p и q — большие простые числа длиной по 1024 бита. Так как прямой перебор этих чисел невозможен, атакующий вынужден прибегать к специализированным алгоритмам факторизации.

На сегодняшний день наиболее эффективным является общий метод решета числового поля (General Number Field Sieve, GNFS) — субэкспоненциальный алгоритм, сложность которого растет быстрее любого полинома от $\log n$, но медленнее экспоненты. Алгоритм состоит из четырех фаз:

1. **Выбор полинома** — нахождение пары многочленов с общим корнем по модулю n, подходящих для последующего просеивания.
2. **Просеивание (фаза решета, sieving)** — массовый отбор пар целых чисел («соотношений»), значения которых при подстановке в многочлены полностью раскладываются на малые простые числа. Эта фаза хорошо распараллеливается и может эффективно выполняться на GPU.
3. **Линейная алгебра** — построение и решение гигантской разреженной системы уравнений над полем $\mathbb{Z}_2$. Фаза плохо масштабируется и обычно требует применения CPU-кластеров.
4. **Извлечение квадратного корня** — финальный этап, приводящий к нахождению множителей p и q.

Примечание: в дальнейших оценках принимается упрощающее предположение о том, что фаза просеивания занимает более 80% общего времени. В реальности, когда значения n очень велики, фаза линейной алгебры может стать доминирующей.

Методология оценки производительности

Поскольку метод GNFS принципиально отличается от прямого перебора, введем понятие эквивалентной криптографической операции (ECO) — условной единицы измерения, соответствующей одной полной проверке ключа AES-128 (шифрование блока и сравнение результата). Эта метрика позволяет унифицировать сравнение различных криптоаналитических атак.

Для адаптации ECO к конкретному алгоритму вводится коэффициент *алгоритмической сложности* α, отражающий относительную трудоемкость операций данного алгоритма по сравнению с одной ECO.

Оценка коэффициента сложности для GNFS

Коэффициент α_GNFS определяется тремя ключевыми факторами:

1. Арифметика больших чисел. Для реализации GNFS требуются операции над 2048-битными числами, тогда как GPU оптимизированы под 32-битные инструкции. Одно модульное умножение эквивалентно тысячам базовых операций. Потери производительности достигают 1000–2000 крат.

2. Паттерны доступа к памяти. Для фазы просеивания требуется случайный доступ к массивам объемом в десятки гигабайт. GPU эффективны при последовательном доступе, что ведет к дополнительному снижению производительности еще в 10–50 раз.

3. Ограниченный параллелизм. В отличие от перебора ключей, GNFS содержит фазы с зависимостями по данным, что мешает полной загрузке вычислительных блоков GPU. Связанное с этим снижение эффективности составляет 5–10 раз.

Суммарная оценка такова:

$$\alpha_{GFNS} \approx 1000 \times 10 \times 5 = 50\,000 \rightarrow \text{до } 100\,000$$

Для консервативной оценки принимаем:

$$\alpha_{GNFS} = 10^5$$

Для дальнейших числовых оценок выполним *калибровку производительности по единичной GPU NVIDIA RTX 4090*.[1]

На алгоритме майнинга *KawPow*, сочетающем задачи, интенсивные по вычислениям и объемам памяти, наблюдается производительность *около 65 млн хешей в секунду*. Однако этот хешрейт необходимо свести к введенной метрике ECO:

- Один хеш KawPow ≈ 1000–2000 операций.
- Одна проверка AES-128 ≈ 500–1000 операций.
- Коэффициент пересчета: k ≈ 1,5–2,0.

Базовая производительность:

$$C'_{base} \approx 65 \times 10^6 \times 1,75 \approx 1,1 \times 10^8\ ECO/\text{с}$$

С учетом сложности GNFS получаем:

$$C'_{effective} = C'_{base} / \alpha_{GNFS\_} =$$

$$1,1 \times 10^8 / 10^5 = 1,1 \times 10^3\ ECO/\text{с}$$

Оценка коэффициента сложности для GNFS

Рассчитаем вероятность взлома Пайе-2048 за один год (T = 31 536 000 секунд) с использованием фермы из N видеокарт RTX 4090.

Общее количество операций за год:

$$C_{total} = N \times 1,1 \times 10^3 \times 31536000 \approx$$

$$N \times 3,47 \times 10^{10}\ ECO$$

Поскольку $2^{112} \approx 5,19 \times 10^{33}$, вероятность успешной атаки можно оценить так:

$$P_{success} = C_{total} / 2^{\lambda} \approx$$

$$(N \times 3,47 \times 10^{10}) / 5,19 \times 10^{33} \approx N \times 6,69 \times 10^{-24}$$

[1] kryptex (2025). NVIDIA RTX 4090: Mining performance: hashrate, specs and profitability on popular cryptocurrencies. Получено из https://www.kryptex.com/en/hardware/nvidia-rtx-4090

Практические сценарии с учетом параллельных GPU

Масштаб атаки	Количество GPU	Вероятность за год	Эквивалентное время до достижения успеха
Малая организация	100	$6{,}7 \times 10^{-22}$	$\sim 10^{21}$ лет
Средний корпоративный кластер	10 000	$6{,}7 \times 10^{-20}$	$\sim 10^{19}$ лет
Государственный уровень	1 000 000	$6{,}7 \times 10^{-18}$	$\sim 10^{17}$ лет

Таким образом, даже в самом пессимистичном сценарии криптографический риск остается пренебрежимо малым — разумеется, при правильном выборе параметров.

Операционные факторы риска

В отличие от рисков криптоатак, операционные риски связаны с практическими аспектами эксплуатации системы. Основные категории включают риски управления ключами, риски побочных каналов и риски реализации криптопротоколов.

Для количественной оценки совокупного риска при условии независимости угроз используется стандартная формула [16]:

$$\Phi_{oper} = 1 - \prod_{i=1}^{n} (1 - P_i) \quad (5),$$

где P_i — вероятность осуществления одной угрозы.

Для получения точной количественной оценки операционных рисков требуется обширная статистика инцидентов, которая лишь в редких случаях доступна публично. Приведенные ниже значения являются иллюстративными и демонстрируют применение подхода к оценке. В реальных условиях оценки должны базироваться на отраслевых стандартах (NIST SP 800-30, ISO/IEC 27005), доступной статистике инцидентов и экспертных оценках с учетом специфики системы.

Иллюстративный пример: корпоративная система с HSM

Рассмотрим гипотетический сценарий организации, использующей сертифицированные аппаратные модули безопасности. Для демонстрации методологии оценим три категории операционных рисков:

Риск утечки ключей (P_{leak}): в качестве базовой оценки для системы с HSM уровня FIPS 140-2 Level 3 примем $P_{leak} = 10^{-5}$. Это значение учитывает потенциальные ошибки администрирования, уязвимости в процедурах резервного копирования и крайне редкие случаи компрометации

аппаратного модуля безопасности. Порядок величины соответствует примерно одному инциденту на 100 тыс. систем в год в организациях с высоким уровнем организационной зрелости.

Риск атак через побочные каналы (P_{side}): предположим, что используется защищенная физическая среда дата-центра с контролируемым доступом и HSM с встроенной защитой от анализа энергопотребления и временных характеристик, и применим экспертную оценку $P_{side} = 10^{-6}$. Столь низкая вероятность объясняется тем, что для успеха злоумышленников необходимы наличие инсайдера либо организация сложной многоэтапной атаки на инфраструктуру.

Риск уязвимостей реализации (P_{impl}): оценка $P_{impl} = 10^{-5}$ основана на том факте, что даже в зрелых криптографических библиотеках периодически обнаруживаются критические уязвимости (в среднем выявляются одна-две в год) со средним окном до исправления 30–90 дней. Однако за такой период лишь малая доля уязвимостей подвергается успешной эксплуатации — этому препятствуют наличие дополнительных уровней защиты и ограниченность распространения информации об уязвимости.

Расчет совокупного риска

Применим формулу (2) к иллюстративным значениям:

$$\Phi_{oper} = 1 - (1 - 10^{-5})(1 - 10^{-6})(1 - 10^{-5}) \approx 2,1 \times 10^{-5}$$

Такое значение соответствует ожидаемой частоте инцидентов: она составляет примерно один на 50 тыс. систем в год — это существенно выше криптографических рисков, но в пределах приемлемого для большинства корпоративных применений.

Ключевые ограничения предложенного описания:

- **Вариативность реальных значений.** Приведенные оценки могут отличаться на несколько порядков в зависимости от архитектуры системы, качества процессов компетенции персонала и мотивации атакующих.
- **Корреляция рисков.** Предположение о независимости угроз является упрощением. В реальности слабые практики безопасности часто проявляются одновременно в нескольких областях, что нелинейно увеличивает совокупный риск.
- **Динамический характер угроз.** Операционные риски изменяются быстрее криптографических: появляются новые векторы атак, обнаруживаются уязвимости нулевого дня, меняется ландшафт угроз и пр.

Практические рекомендации

Для адаптации методологии к конкретной организации необходимо:

- вести детальный учет инцидентов безопасности;
- отслеживать и анализировать отраслевые отчеты (Verizon DBIR, отчеты ENISA и др.);
- проводить регулярные оценки рисков и угроз с привлечением внешних экспертов;
- при отсутствии количественных данных использовать качественные методы оценки рисков.

Из предложенных оценок следует, что операционные риски значительно выше криптографических — это необходимо учитывать при реализации.

Модель рисков для FHE

Переход от частично гомоморфного к полностью гомоморфному шифрованию представляет собой качественный сдвиг как в плане функциональных возможностей, так и по сложности анализа безопасности. В отличие от PHE, которое базируется на классических задачах теории чисел с многолетней историей анализа, FHE использует современные криптографические конструкции на основе решеток, что требует принципиально новых методов оценки безопасности.

От классической криптографии к криптографии на решетках

В основе криптостойкости алгоритмов шифрования на решетках лежат две вычислительно сложные задачи — обучение с ошибками (LWE) и поиск самого короткого ненулевого вектора в решетке (Shortest Vector Problem, SVP). Обе проблемы относятся к классу трудно решаемых даже для квантовых компьютеров.

Полностью гомоморфные схемы (FHE) преимущественно используют LWE и ее кольцевые варианты (RLWE). Безопасность FHE определяется комплексом взаимосвязанных параметров:

n — размерность решетки (обычно из интервала $[2^{10}; 2^{16}]$);

q — модуль, определяющий размер коэффициентов (от 2^{30} до 2^{100});

σ — стандартное отклонение распределения ошибок.

Уровень криптостойкости схемы λ для конкретного набора параметров можно оценить с помощью специализированных инструментов, таких как LWE Estimator, учитывающих современные алгоритмы решеточной редукции и дающих консервативные оценки.

Классификация атак на FHE

Угрозы для полностью гомоморфного шифрования можно разделить на три основные категории атак. Каждая из них требует специфических контрмер.

Криптоанализ решеток — атаки на математические основы схем.

- **BKZ-редукция:** оптимизация эвристик перебора для блоков β=250. Практический взлом LWE-параметров при n = 350 за 2^{80} операций [17].
- **Алгоритмы с просеиванием:** алгоритм ListSieve с квантовым ускорением $O(2^{0,292n})$ [18].
- **Атаки на структурированные ключи:** успешное восстановление разреженных секретов с весом Хемминга ≤ 11 для n = 256 [19].

Стоит отметить, что приведенные атаки работают при существенно заниженных параметрах — крайне далеких от значений, рекомендуемых для использования на практике.

Атаки на реализацию — эксплуатация физических особенностей вычислений.

- **Анализ шума в CKKS:** восстановление ключа через статистику ошибок [20].
- **Использование побочных каналов:** показатели энергопотребления (DPA на бутстраппинге) [21];

  временные характеристики (атаки на основе анализа времени выполнения теоретико-числовых преобразований) [21].

Протокольные атаки

Присущая схемам FHE изменчивость (malleability) означает, что любой, кто обладает шифртекстом, может преобразовать его в другой шифртекст, соответствующий применению определенной функции к исходному сообщению, не зная при этом самого сообщения и не расшифровывая данные. Это свойство делает невозможным достижение CCA-безопасности (устойчивости к адаптивной атаке с выбранным шифртекстом) стандартными криптографическими методами.

Аналогичная проблема характерна и для других криптосхем. Например, базовая схема RSA без надстроек также не удовлетворяет требованиям CCA-безопасности [22] и требует специальных конструкций (например, OAEP) для защиты от подобных атак.

Кроме того, в условиях кумулятивного роста шума при последовательных гомоморфных операциях необходим строгий контроль шума. Недооценка этого фактора может привести к невозможности корректного расшифрования.

Операционные риски, специфичные для FHE

Полностью гомоморфное шифрование (FHE) порождает качественно новые операционные риски по сравнению с частично гомоморфными схемами. Эти особенности не просто усиливают известные риски, а формируют новые

классы уязвимостей и векторов атак, требующие пересмотра традиционных подходов к эксплуатации и защите криптосистем.

Управление массивными ключевыми структурами

В отличие от частично гомоморфных схем с относительно компактными ключами, для FHE требуется обеспечить хранение множества объемных криптографических объектов и обращение с ними. Вот примеры таких объектов:

- **Ключи вычислений (evaluation keys)** — достигают десятков гигабайт в схемах, поддерживающих глубокие вычисления.
- **Бутстраппинговые ключи (bootstrapping keys)** — критически важны для обновления зашифрованных данных, объем этих ключей может достигать 1 Гбайт.
- **Ключи релинеаризации (relinearization keys)** — используются после операций умножения для сдерживания роста объемов шифртекста.
- **Ключи Галуа (Galois keys)** — необходимы для выполнения перестановок и поворотов в схемах типа BGV и BFV.

Такой масштаб ключевой информации создает значительные операционные сложности — от потребности в специализированной инфраструктуре хранения до рисков, возникающих при передаче ключей через публичные сети. Необходима фундаментальная адаптация традиционных механизмов управления ключами, разработанных для симметричных и асимметричных алгоритмов с малым объемом ключей, — только тогда их использование в схемах FHE станет безопасным.

Усиление рисков побочных каналов

Вычислительная нагрузка при использовании FHE на порядки превышает нагрузку, создаваемую классическими криптоалгоритмами. Каждая гомоморфная операция сопряжена с интенсивными вычислениями, создающими потенциально эксплуатируемые злоумышленниками побочные проявления:

- В схемах BGV/BFV применяются тысячи операций теоретико-числового преобразования (NTT), формирующих характерные паттерны доступа к памяти и энергопотребления.
- В CKKS используется работа с приближенной арифметикой над кольцами полиномов, где ошибки округления могут непреднамеренно раскрывать информацию.
- В TFHE частые (каждые 5–10 вычислений) операции бутстраппинга создают предсказуемые временные сигнатуры

Перечисленные факторы не только увеличивают поверхность атак для классических каналов (временных, энергопотребления), но и открывают новые, специфичные для гомоморфных вычислений каналы утечки.

Сложности корректной реализации

Алгоритмическая сложность FHE существенно превосходит сложность традиционных криптосистем, что делает их корректную реализацию особенно уязвимой. Можно выделить следующие основные категории рисков:

- **Управление шумом:** неправильный учет накопления шума может привести к ошибкам расшифрования или частичной утечке данных.
- **Арифметика больших чисел:** модули размером в сотни или тысячи бит повышают вероятность ошибок в модульной арифметике и ее оптимизациях на основе китайской теоремы об остатках.
- **Управление памятью:** промежуточные вычисления требуют очистки памяти после использования, так как злоумышленники могут извлечь остаточные данные, сохранившиеся в оперативной памяти.
- **Параметры безопасности:** выбор параметров в FHE особенно чувствителен — необходимо искать баланс между безопасностью, производительностью и функциональностью, а это требует глубокого понимания математических основ схемы.

Такие особенности FHE указывают на необходимость дополнительного внимания к операционной модели безопасности.

- **Инфраструктура:** необходимы специализированные решения для хранения и защищенной передачи структур ключей.
- **Мониторинг:** системы обнаружения аномалий должны учитывать характерные шаблоны вычислений FHE.
- **Валидация:** критически важно обеспечить формальную верификацию кода и автоматизированное тестирование граничных случаев.
- **Стандартизация:** отсутствие устоявшихся стандартов и успешных практик повышает риски ошибок при самостоятельной реализации FHE.

Таким образом, FHE имеет на порядок более высокий операционный риск при сопоставимой с PHE криптостойкости.

Особенности различных схем FHE

Современные FHE-схемы различаются как по производительности, так и по криптографическим свойствам. Выбор схемы напрямую влияет на доступные модели угроз и особенности защиты.

CKKS: сила и слабость приближенной арифметики

Благодаря поддержке операций над зашифрованными вещественными числами, схема CKKS [23] сделала FHE практически применимым для задач машинного обучения и обработки чисел с плавающей точкой. Однако использование приближенной арифметики влечет за собой новые типы рисков, связанных с накоплением ошибок и возможной утечкой информации через статистику шумов.

В работе [2] показано, что без корректного добавления защитного шума (noise padding) CKKS-подобные схемы уязвимы к атакам на секретный ключ, реализуемым на основе анализа ошибок даже в пассивной модели атакующего. Уязвимость была эмпирически подтверждена в работе [25] и устранена предложением схемы LMSS, которая накладывает дополнительные ограничения на параметры схемы и требует расширения шума.

Хотя защита существует и успешно реализуется, она оказывает влияние на точность:

- риск утечки оказывается высоким в условиях слабой защиты;
- при усиленной защите сохраняется безопасность, однако точность численного результата снижается.

Важно подчеркнуть, что речь идет не о взломе CKKS, а о необходимости корректной реализации и настройки криптографических параметров — как и в любых схемах с приближенной арифметикой.

TFHE: скорость через специализацию

Основанная на частом применении процедуры бутстраппинга схема TFHE создает особый профиль рисков:

- высокая частота бутстраппинга повышает экспозицию ко времени выполнения и побочным каналам, особенно в аппаратных реализациях;
- каждая операция в TFHE имеет малую сложность и ограниченную «выделяемую» информацию, что снижает информативность потенциальных утечек.

При наличии надлежащей защиты (например, константное время, очищение памяти, защита от DPA) схема TFHE демонстрирует сопоставимый уровень устойчивости к реализации атак, как и другие FHE-системы.

Таким образом, при правильно выстроенной инженерной защите TFHE остается достаточно безопасной схемой, особенно хорошо подходящей для задач, требующих логической обработки, булевых деревьев и условных ветвлений.

Выводы и рекомендации

Выбор криптоалгоритма с доказанной стойкостью не гарантирует надежности системы в целом. Операционные риски доминируют над криптографическими, что демонстрирует важность инфраструктурных аспектов обеспечения безопасности.

Криптография на основе целочисленных решеток более требовательна к уровню погружения для корректного настраивания параметров. При этом операционные риски более критичны: сложность FHE увеличивает вероятность ошибок реализации и атак через побочные каналы.

Выбор схемы определяет профиль рисков: CKKS требует особого внимания к защите шума, TFHE — к побочным каналам, BGV/BFV — к параметризации.

Выбор между PHE и FHE в первую очередь определяется не уровнем математической стойкости, а задачами и условиями эксплуатации. PHE-схемы оптимальны, когда нужен ограниченный набор гомоморфных операций (аддитивных или мультипликативных), невысокие задержки и простая интеграция в доверенной среде. Эти схемы требуют в сотни раз меньше ресурсов для хранения ключей и шифртекстов, чем схемы FHE, при этом внедрение проходит быстрее и с меньшими затратами. FHE-схемы дают большее разнообразие гомоморфных вычислений над данными без расшифрования, но предъявляют высокие требования к инфраструктуре: необходимы гигабайтные размеры evaluation- и bootstrapping-ключей, миллисекундные операции с шумовой буферизацией (LMSS padding) и контроль целостности. Операционные риски при этом возрастают в 5–10 раз по сравнению с PHE, а сложность конфигурации и их поддержки значительно выше.

Рекомендации для практического внедрения

Для PHE рекомендуется выбирать длину ключа, соответствующую уровню криптостойкости $\lambda \geq 112$ бит (≥ 128 для долгосрочных проектов), и хранить приватный ключ в HSM/TEE при условии ротации и аудита ключей, а также учета аномалий использования.

Для FHE, помимо обеспечения базовой безопасной инфраструктуры PHE, рекомендуется выбирать параметры, соответствующие $\lambda \geq 128$ бит с учетом деградации, и обеспечивать шумовую буферизацию в CKKS.

Гомоморфное шифрование — мощный инструмент для создания систем, изначально спроектированных так, чтобы обеспечивать приватность (privacy-by-design). Однако создание таких систем возможно только при условии тщательного баланса между криптостойкостью, операционной сложностью и архитектурными требованиями.

ЮРИДИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ТЕХНОЛОГИИ

Защита обрабатываемых персональных данных (ПД) в России является одной из обязанностей оператора, а также привлекаемых им обработчиков. Шифрование — одна из самых распространенных технологий защиты, при этом гомоморфное шифрование (HE) позволяет не просто обеспечить конфиденциальность обрабатываемых данных, но и совершать над ними операции без угрозы раскрытия в процессе обработки.

Важно учитывать, что законодательство РФ предъявляет определенные требования к используемым средствам защиты ПД. Так, Федеральный закон 152-ФЗ от 27 июля 2006 г. «О персональных данных» требует, чтобы используемые средства защиты информации прошли в установленном порядке процедуру оценки соответствия (п. 3, ч. 2, ст. 19).

Для технологии HE пока нет официально утвержденных документов нормативно-технического регулирования (стандартов)*, которые можно было бы использовать для прохождения оценки соответствия. Следовательно, HE не может применяться как средство защиты информации, которое требуется по закону. Тем не менее организации могут использовать HE как дополнительную технологию, способствующую защите ПД при их обработке, тем самым укрепляя собственную репутацию как добросовестного и ответственного оператора или обработчика ПД.

С учетом описанных нами рисков использования HE, имеет смысл разделять перспективы

* На момент написания статьи (июль 2025 г.) PHE-схема EC-Elgamal проходит стандартизацию в ТК 26 «Криптографическая защита информации».

полного (FHE) и частичного (PHE) гомоморфного шифрования. Принимая во внимание преимущества PHE для защищенной обработки данных, мы надеемся, что уполномоченные органы проведут необходимую работу для того, чтобы PHE стало не просто дополнительной мерой подтверждения ответственного подхода к защите данных, а одним из основных средств защиты. Что касается FHE, то его пока следует рассматривать, скорее, как экспериментальную технологию, обладающую высоким потенциалом при условии правильного использования.

ПЕРСПЕКТИВЫ ПРИМЕНИМОСТИ ТЕХНОЛОГИИ

Общие тезисы о применимости

Гомоморфное шифрование представляет собой одну из наиболее фундаментальных технологий в контексте приватных вычислений: оно позволяет выполнять операции над зашифрованными данными, не раскрывая их содержания, что делает возможным реализацию систем, обеспечивающих безопасность на уровне архитектуры. Однако для практического внедрения важно учитывать ряд инженерных и архитектурных особенностей.

1. FHE — не универсальный механизм, а инструмент точечного применения

С учетом нынешнего уровня зрелости FHE, его разумно применять не к полным потокам данных, а к их агрегированным представлениям или параметрам моделей — например, к обновлениям моделей в федеративном обучении (Federated Learning), элементам аналитики в облачных системах, инференсу по заранее подготовленным признакам и пр. Такой подход позволяет снизить издержки и одновременно сохранить необходимый уровень конфиденциальности, не прибегая к полному реинжинирингу существующих ИТ-систем.

2. FHE как элемент комбинированных архитектур

Гибридные схемы (FL + HE, HE + DP, HE + TEE) становятся приоритетными направлениями в ходе разработки масштабируемых и защищенных ИИ-решений. В таких системах FHE используется там, где данные нельзя доверить даже на уровне инфраструктуры, а для остальных этапов применяется обработка более производительными и гибкими методами. Это особенно актуально в распределенных сценариях с несколькими юридически обособленными участниками — например, клиниками, банками, операторами IoT.

3. Вычисления над данными — уже сегодня

FHE-системы промышленного уровня, такие как Microsoft SEAL, Zama Concrete, Duality PALISADE, IBM HELib, позволяют выполнять реальную обработку защищенных данных — от простых арифметических операций до инференса CNN и MLP. Хотя время обработки остается в миллисекундном диапазоне, этого достаточно для выполнения ряда задач, не требующих жестких ограничений по временной задержке.

4. Признание и инвестиции лидеров рынка

Публичные инициативы Google, Microsoft, Intel, IBM, NVIDIA, подразделения министерства обороны США DARPA DPRIVE подтверждают стратегическую значимость FHE. Компании инвестируют в разработку ускорителей, языков описания защищенных операций (TFHE DSL, Concrete ML) и в запуск тестовых платформ. Это ускоряет переход от научных прототипов к промышленным внедрениям.

Направления дальнейших исследований и ближайшие задачи

Несмотря на прогресс в области теоретической проработки схем FHE, практическое применение этой технологии остается ограниченным из-за сложности настройки и высоких требований к инфраструктуре.

В ближайшие годы ключевыми направлениями, способными расширить область применимости FHE, будут следующие:

1. Аппаратное ускорение

Одним из наиболее перспективных векторов развития является создание специализированных аппаратных ускорителей FHE. Согласно

отчету Zama за четвертый квартал 2024 г., вендоры ожидают появления первых ASIC-чипов, поддерживающих FHE, уже к 2026 г. Эти чипы сократят время операций на порядки, что сделает возможным использование FHE в задачах с жесткими ограничениями по временной задержке, в том числе в финансовых транзакциях и периферийных вычислениях в здравоохранении.

Цель: достичь производительности в миллионы гомоморфных операций в секунду в типовых рабочих нагрузках (например, в моделях инференса).

2. Унификация стандартов параметризации
В отличие от PHE-схем, таких как Пайе, RSA, которые уже прошли сертификацию NIST, в FHE до сих пор нет единой системы классификации схем, уровней безопасности и параметров шумоподавления. Разработка общепринятых рекомендаций снизит входной порог для индустриальных команд и упростит регуляторные проверки.

Цель: выработка стандартизированного профиля параметров FHE-схем (например, CKKS level 5), пригодных для типовых бизнес-применений.

3. Автоматизация параметризации и верификации
Применение FHE требует глубокой экспертной настройки — выбора уровня шума, схемы буферизации, параметров бутстраппинга и т. д. Снизить риски ошибок конфигурации и упростить внедрение FHE в ИТ-стек поможет встраивание автоматизированных инструментов настройки и верификации (включая ZKP-обертки) в пайплайны CI/CD.

ВЫВОДЫ

Схемы гомоморфного шифрования наряду с протоколами распределенных безопасных вычислений существенно расширяют возможности криптографических методов защиты информации, что дает возможность проводить обработку информации без ее непосредственного расшифрования. В ряде сценариев это позволяет минимизировать или исключить наличие доверенной среды, в которой защищаемая информация циркулирует в открытом виде.

Перспективы внедрения подобных методов связаны в первую очередь с существующими тенденциями использования облачных решений и аутсорсинга, которые в большинстве случаев входят в противоречие с регуляторными требованиями в отношении защиты отдельных классов конфиденциальной информации.

Несмотря на то, что существующие методы гомоморфного шифрования обладают высокой вычислительной ресурсоемкостью и слабо применимы для обработки больших объемов данных, для отдельных прикладных задач в большинстве случаев удается найти сравнительно эффективный вариант их использования.

Следует отметить, что для схем гомоморфного шифрования требуются дополнительные механизмы контроля целостности, такие как неинтерактивные схемы с нулевым разглашением и электронные подписи.

ОБ АВТОРАХ ДОКЛАДА

АЛЕКСЕЙ НЕЙМАН
Исполнительный директор Ассоциации больших данных, руководитель FIT Academy of Russia, Master of Data Science, CDMP, PMP

ОЛЕГ ФАТЮХИН
Технический руководитель проекта, GUARDORA

МИХАИЛ ФАТЮХИН
Ведущий разработчик-исследователь / криптограф, GUARDORA

КИРИЛЛ ГРОШЕНКОВ
Ведущий исследователь, GUARDORA

ВАЛЕРИЙ ХВАТОВ
Специалист по кибербезопасности и распределенным вычислениям, технический директор DGT Network

АЛЕКСАНДР ПАРТИН
Адвокат и партнер Privacy Advocates, соучредитель «РППА.Офис», сопредседатель Privacy & Legal Innovation кластера РАЭК, CIPP/E, CIPM

АЛЕКСЕЙ МУНТЯН
Генеральный директор Privacy Advocates, внешний менеджер по защите данных нескольких транснациональных холдингов, соучредитель Regional Privacy Professionals Association (RPPA.pro), сопредседатель Privacy & Legal Innovation и кластера РАЭК

Авторы выражают благодарность эксперту технического комитета по стандартизации «Криптографическая защита информации» (ТК 26) **ГРИГОРИЮ МАРШАЛКО,** чьи замечания и комментарии значительно помогли при подготовке настоящего материала.

ИСТОЧНИКИ

1. R. Rivest, L. Adleman, M. Dertouzos, On Data Banks and Privacy Homomorphisms, Foundations of Secure Computation, Academia Press, 1978.

2. P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, EUROCRYPT, volume 1592 of Lecture Notes in Computer Science, page 223-238. Springer, 1999.

3. D. Boneh, E. Goh, K. Nissim, Evaluating 2-DNF Formulas on Ciphertexts, Theory of Cryptography Conference, TCC 2005, volume 3378 of Lecture Notes in Computer Science, page 325-341, Springer, 2005.

4. C. Gentry, A Fully Homomorphic Encryption Scheme, Stanford University, 2009.

5. O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," Journal of the ACM, vol. 56, no. 6, pp. 34:1–34:40, 2009. DOI: 10.1145/1568318.1568324. Earlier version appeared in STOC 2005.

6. V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," Journal of the ACM, vol. 60, no. 6, pp. 43:1–43:35, 2013. DOI: 10.1145/2535925. Earlier version appeared in EUROCRYPT 2010.

7. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," Journal of Cryptology, vol. 33, no. 1, pp. 34–91, 2020. DOI: 10.1007/s00145-019-09319-x. Earlier versions appeared in ASIACRYPT 2016 and 2017.

8. Ducas, L., & Micciancio, D. (2015). FHEW: Bootstrapping Homomorphic Encryption in less than a second. Centrum Wiskunde & Informatica, Amsterdam; University of California, San Diego.

9. M. Joye, "Guide to Fully Homomorphic Encryption over the [Discretized] Torus," IACR Cryptology ePrint Archive, Report 2021/1402, 2021. Available: https://ia.cr/2021/1402.

10. K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, Q. Yang, SecureBoost: A Lossless Federated Learning Framework, IEEE Intell. Syst. 36(6):87-98, 2021.

11. T. Fan, W. Chen, G. Ma, Y. Kang, L. Fan, Q. Yang, SecureBoost+: Large Scale and High-Performance Vertical Federated Gradient Boosting Decision Tree, arXiv:2110.10927, 2024.

12. Z. Tian, R. Zhang, X. Hou, L. Lyu, J. Liu, K. Ren, FederBoost: Private Federated Learning for GBDT, arXiv:2011.02796, 2022.

13. C. Masson, J. E. Rim, H. K. Lee, DDSketch: A Fast and Fully-Mergeable Quantile Sketch with Relative-Error Guarantees, arXiv:1908.10693, 2019.

14. R. Zhu, C. Ding, Y. Huang, Practical MPC+FHE with Applications in Secure Multi-PartyNeural Network Evaluation. IACR Cryptol. ePrint Arch. 2020: 550, 2020.

15. Katz J., Lindell Y. Introduction to Modern Cryptography. — 2nd ed. — Boca Raton: Chapman & Hall/CRC, 2014. — (Cryptography and Network Security Series).

16. Modarres, M., Kaminskiy, M. P., & Krivtsov, V. (2022). Reliability engineering and risk analysis: A practical guide (3rd ed.). CRC Press.

17. Chen Y., Nguyen P.Q. BKZ 2.0: Better Lattice Security Estimates // INRIA and École Normale Supérieure. — Paris, 2011.

18. Loyer J., Chailloux A. Classical and quantum 3 and 4-sieves to solve SVP with low memory, INRIA [French Institute for Research in Computer Science and Automation], 2022.

19. Wenger, E., Saxena, E., Malhou, M., Thieu, E., & Lauter, K. (2024). Benchmarking Attacks on Learning with Errors. Meta AI, 2024.

20. Li, B., Micciancio, D. (2021). On the Security of Homomorphic Encryption on Approximate Numbers. In: EUROCRYPT 2021, 2021.

21. Ghaleb, B., Buchanan, W. J. Side Channel Analysis in Homomorphic Encryption. arXiv preprint arXiv: 2505.11058, 2025.

22. Bleichenbacher D., Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. Advances in Cryptology — CRYPTO'98, 1998.

23. Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Advances in Cryptology — ASIACRYPT 2017 (pp. 409–437).

24. Doan, T. V. T., Messai, M.-L., Gavin, G., & Darmont, J., A survey on implementations of homomorphic encryption schemes. The Journal of Supercomputing, 2023.

25. B. Li, D. Micciancio, M. Schultz, and J. Sorrell, "Securing Approximate Homomorphic Encryption Using Differential Privacy," Cryptology ePrint Archive, Report 2022/816, 2022.

26. M.R. Albrecht, Estimate All the {LWE, NTRU} Schemes!. In: Catalano, D., De Prisco, R. (eds) Security and Cryptography for Networks. SCN 2018. Lecture Notes in Computer Science, vol 11035, Springer, 2018.

27. S. Goldwasser, S. Micali, Probabilistic encryption & how to play mental poker keeping secret all partial information, In Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali, pages 173–201. 2019.

28. D. Micciancio, O. Regev, Lattice-based Cryptography, In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds) Post-Quantum Cryptography, Springer, Berlin, Heidelberg, 2009.

29. J. Katz, Y. Lindell, Introduction to Modern Cryptography, Chapman & Hall/Crc Cryptography and Network Security Series, 2007.

30. O. Regev, The learning with errors problem, In Proc. of 25th IEEE Annual Conference on Computational Complexity, pages 191–204, 2010.

Редакторы доклада

ЖАННА ПОКРОВСКАЯ
Руководитель пресс-службы
Ассоциации больших данных

АЛИЯ ТРУНАЕВА
PR-менеджер
Ассоциации больших данных

АССОЦИАЦИЯ БОЛЬШИХ ДАННЫХ

www.rubda.ru

Адрес: Москва,
Пресненская набережная, 10с2

+7 (495) 252-72-60
info@rubda.ru