ОБЕСПЕЧЕНИЕ КОНФИДЕНЦИАЛЬНОСТИ ПРИМЕНЕНИЯ ПРЕДВАРИТЕЛЬНО ОБУЧЕННЫХ ГРАФОВЫХ НЕЙРОННЫХ СЕТЕЙ С МЕХАНИЗМОМ ВНИМАНИЯ

Шевченко В. А.[1], Запечников С. В.[2]

DOI: 10.21681/2311-3456-2024-5-18-27

Аннотация. *В статье предлагается комплекс криптографических протоколов для реализации системы конфиденциального машинного обучения на основе графовых нейронных сетей с механизмом внимания (далее – системы «КонфГраф»). Приведена классификация искусственных нейронных сетей, лежащих в основе глубокого обучения. Выделены основные задачи обеспечения конфиденциальности, возникающие при обучении и применении моделей машинного обучения на основе искусственных нейронных сетей. Приведено описание основных криптографических примитивов, необходимых для реализации протоколов безопасных многосторонних вычислений, а именно схем разделения секрета и протокола передачи с забыванием. В статье приводится краткая характеристика методологии доказательства безопасности криптографических протоколов, в том числе протоколов безопасных многосторонних вычислений, называемой универсальной компонуемостью (UC-security). Описываются и анализируются основные и вспомогательные протоколы, лежащие в основе системы «КонфГраф»: «коррелированный» протокол передачи с забыванием, а также протоколы конфиденциального умножения матриц, вычисления значений функций активации ReLU и LeakyReLU, приводятся доказательства их безопасности. Остальные протоколы, применяющиеся в «КонфГраф», перечислены в статье с кратким описанием их входных и выходных данных. Безопасность предлагаемых протоколов системы «КонфГраф» доказывается на основе методологии универсальной компонуемости.*

Ключевые слова: *криптография, информационная безопасность, конфиденциальное машинное обучение, безопасные многосторонние вычисления, графовые нейронные сети с механизмом внимания, схемы разделения секрета, протокол передачи с забыванием.*

Введение

Одной из ключевых технологических тенденций XXI века, несомненно, является искусственный интеллект (ИИ). Его внедрение в различные области может оказать существенное влияние как на бизнес, так и на повседневную жизнь человека. По оценкам Министерства экономического развития РФ Россия входит в первую десятку стран по объему совокупных вычислительных мощностей, используемых для реализации функций ИИ.

Реализация технологий ИИ стала возможной благодаря бурному развитию машинного обучения (МО). В настоящий момент существуют десятки видов моделей МО, решающих различные задачи от прогнозирования численных значений до создания изображений или иных типов данных. Особенно высоких результатов в той или иной сфере применения ИИ можно достигнуть посредством глубокого обучения, которое, по сути, представляет собой МО с применением искусственных нейронных сетей (НС).

Выделяют следующие основные классы НС: полносвязные, рекуррентные, сверточные и графовые. В современных глубоких НС используются преимущественно три последние типа архитектур [1]. Рекуррентные НС представляют собой совокупность слоёв нейронов, накапливающих предыдущее состояние НС и циклически обрабатывающих вновь поступающие данные, что делает их инструментом обработки линейно упорядоченных данных. В основе сверточных НС, как понятно из названия, лежит математическая операция кросс-корреляции (свёртки), что обуславливает их высокую эффективность при обработке многомерных регулярных массивов данных, в том числе, при решении задач распознавания образов. Граф является мощным инструментом представления и обработки сложноструктурированных данных, благодаря чему графовые НС (ГНС) преимущественно используются для решения задачи анализа данных, не имеющих регулярной структуры [2].

1. Графовые нейронные сети

Простейшим, но и наиболее востребованным в прикладных задачах видом ГНС являются сверточные ГНС. Пусть задан некоторый граф $G = (V,E)$,

1 Шевченко Вячеслав Андреевич, соискатель, Национальный исследовательский ядерный университет «МИФИ», Москва, Россия. E-mail sheff-slava@mail.ru
2 Запечников Сергей Владимирович, доктор технических наук, доцент, профессор Национального исследовательского ядерного университета «МИФИ», Москва. Россия. E-mail: svzapechnikov@mephi.ru

где V – множество вершин, E – множество рёбер. Пусть каждой вершине v_i сопоставлен вектор атрибутов x_i. Совокупность векторов атрибутов вершин графа образует матрицу X. ГНС позволяет решать задачи классификации на множестве вершин графа, подграфах графа и графе в целом. Идея свёрточной ГНС основана на том, что в такой сети эмбеддинги (векторные представления в признаковом пространстве) каждой из вершин графа вычисляются на основе атрибутов как самой вершины, так и соседних с ней вершин. Глубина распространения влияния атрибутов на соседние вершины графа определяется количеством слоёв ГНС.

Пусть x_i – атрибут вершины графа v_i, X – матрица атрибутов вершин, W – матрица весов модели МО, тогда состояние вершины графа v_i в сверточной ГНС будет обновляться в соответствии с выражением [3]:

$$h_i = \sum_{j \in N(v_i)} x_j W^T, \quad (1)$$

что в матричной форме можно представить так

$$H = \tilde{A}^T X W^T, \quad (2)$$

где A – матрица смежности графа, $\tilde{A} = A + I$, I – единичная матрица.

Дальнейшее развитие ГНС привело к появлению ГНС с механизмом внимания (attention mechanism), отличительной чертой которых является наличие весов внимания, которые, в буквальном смысле, увеличивают влияние весов одних вершин, а других – уменьшают. В таких НС состояние вершины графа v_i будет обновляться в соответствии с выражениями в линейной форме

$$h_i = \sum_{j \in N(v_i)} a_{i,j} W x_j, \quad (3)$$

и в матричной форме

$$H = \tilde{A}^T W_\alpha X W^T, \quad (4)$$

где $a_{i,j}$ – коэффициент внимания вершины v_i по отношению к вершине v_j, W_α – матрица коэффициентов (весов) внимания.

2. Конфиденциальное машинное обучение

Однако при создании, развертывании и применении моделей МО, в том числе ГНС с механизмом внимания, не всегда уделяется должное внимание конфиденциальности. Риски нарушения конфиденциальности информации возникают как при обучении, так и при применении моделей МО, развернутых в недоверенной среде, например, в облаке. Требования к конфиденциальности особенно актуальны при обработке персональных данных, а также информации, содержащей охраняемые законом виды тайны: коммерческую, банковскую, врачебную и др. Кроме того, сама модель МО может являться интеллектуальной собственностью её владельца, поэтому даже при известной архитектуре НС её параметры тоже требуют защиты.

Перечисленные факторы привели к появлению новой области исследований на стыке МО и криптографии – конфиденциального машинного обучения (КМО). Предметом КМО является разработка методов и алгоритмов, позволяющих обучать и применять модели МО в условиях взаимного недоверия между владельцем модели МО, провайдером вычислительных ресурсов и клиентом, желающим направить некоторый запрос к модели (или даже обучить модель) [4]. КМО стало возможно благодаря развитию таких методов криптографии, как гомоморфное шифрование и безопасные многосторонние вычисления (БМВ).

Основная цель протоколов БМВ состоит в вычислении несколькими участниками общей функции без раскрытия друг другу своих исходных данных, которые являются конфиденциальными. Это достигается посредством применения ряда криптографических примитивов [5].

1. Схемы разделения секрета. В БМВ используются два основных вида СРС: пороговые и арифметические. (t,n)-пороговой схемой разделения секрета называют схему, позволяющую разделить секрет x на n долей, причем обладание любыми $t-1$ долями не позволяет получить никакой информации об x, тогда как обладание t долями позволяет однозначно восстановить x. Более строго схемы разделения секрета можно описать следующим образом. Пусть X – множество секретов, X_1 – множество долей секретов, $Shr: X \longrightarrow X_1^n$ – алгоритм разделения секрета, $Rec: X_1^n \longrightarrow X$ – алгоритм восстановления секрета. Тогда (t,n)-пороговой схемой разделения секрета называется пара алгоритмов (Shr,Rec), обладающих двумя свойствами [6]:

1) корректностью: если $Shr(x) = (x_1,x_2,...,x_n)$, то $Pr[\forall k \geq t, Rec(x_{i_1},x_{i_2},...,x_{i_k}) = x] = 1$;
2) совершенной секретностью: если a, b – секреты, а $v = v_1,v_2,...,v_k$ – вектор любых возможных долей секрета, причем $k < t$, то $Pr[Shr(a)|_k = v] = Pr[Shr(b)|_k = v]$, где $|_k$ – проекция на множество из k элементов.

В настоящей работе нашла применение арифметическая схема разделения секрета (АСРС). Основные операции АСРС можно описать так:

- Shr: S_i случайно выбирает $r \in_{\mathbb{R}} \mathbb{Z}_q$, где q – большое простое число, известное обоим участникам, отправляет его S_{1-i}, вычисляет $\langle x \rangle_i^A = x - r \bmod q$, S_{1-i} принимает, что $\langle x \rangle_{1-i}^A = x$;
- Rec: S_{1-i} отправляет S_i имеющееся у него значение $\langle x \rangle_{1-i}^A$, S_i рассчитывает $x = \langle x \rangle_i^A + \langle x \rangle_{1-i}^A \bmod q$.

2. Протокол передачи с забыванием (oblivious transfer). Пусть взаимодействуют два участника: отправитель, хранящий секреты x_0 и x_1, и получатель, выбирающий один из только один из двух секретов путем генерации бита выбора $b \in \{0,1\}$. Протокол передачи с забыванием позволяет получателю узнать выбранный секрет x_b, при этом не получая никакой информации о значении другого секрета x_{1-b}. В то же время, отправитель не получает информации о выборе, сделанном получателем [7].

3. Протоколы системы конфиденциального машинного обучения на основе ГНС

Предлагаемая система КМО на основе ГНС с механизмом внимания, названная нами «КонфГраф», поддерживает конфиденциальное применение предварительно обученной модели МО и подразумевает наличие следующих участников протокола:

1) клиента, обладающего матрицей атрибутов вершин X некоторого графа G и желающего сохранить ее в тайне;
2) владельца обученной ГНС с механизмом внимания, обладающей следующими параметрами: матрицей смежности графа A, матрицей весов W и матрицей весов линейного преобразования для вычисления коэффициентов внимания W_{att} – желающего сохранить перечисленные параметры модели в тайне;
3) двух независимых серверов S_i, $i \in \{0,1\}$ (например, принадлежащих разным облачным провайдерам).

Клиенту требуется получить ответ от модели МО на его данных, представленных в форме матрицы X. Владелец модели МО обладает предварительно обученной моделью на основе ГНС с механизмом внимания, способной дать желаемый клиентом результат, но не имеет достаточных вычислительных ресурсов для этого, поэтому вынужден прибегать к помощи двух независимых облачных провайдеров. Однако ни клиент, ни владелец модели не доверяют ни друг другу, ни облачным провайдерам, поэтому для сохранения конфиденциальности информации клиента и владельца модели требуется применение механизмов КМО.

В первом приближении принцип работы предлагаемой системы «КонфГраф» можно описать следующим образом.

1. Все участники протокола согласуют ряд значений общедоступных величин:
   1) большого простого числа q;
   2) длины чисел ℓ в битах;
   3) параметра безопасности κ;
   4) длины дробной части чисел f, измеряющейся в битах (предполагается использование чисел с фиксированной точкой).
2. Клиент и владелец модели подготавливают исходные данные.
3. Сервер S_i вычисляет разделенную по АСРС матрицу коэффициентов внимания $\langle W_\alpha \rangle_i^A$ с применением протоколов БМВ следующим образом:
   3.1. Транспонирование $\left(\langle W \rangle_i^A\right)^T$.
   3.2. Конфиденциальное вычисление произведения матриц $\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T$.
   3.3. Составление матрицы
   $$\left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)[\langle X \rangle_i^A[0]],$$
   для чего при помощи протокола конфиденциального доступа к элементам массива выбираются те строки матрицы $\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T$, номера которых соответствуют нулевой строке матрицы C.
   3.4. Составление матрицы
   $$\left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)[\langle X \rangle_i^A[1]],$$
   по аналогии с п. 3.3.
   3.5. Построчная конкатенация матриц
   $$\left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle X \rangle_i^A[0]\right] \text{ и } \left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle X \rangle_i^A[1]\right]$$
   (т.е. запись указанных матриц «друг рядом с другом»).
   3.6. Транспонирование результата п. 3.5:
   $$\left(\left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle X \rangle_i^A[0]\right] \parallel \left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle X \rangle_i^A[1]\right]\right)^T. \quad (5)$$
   3.7. Конфиденциальное вычисление произведения $\langle W_{att} \rangle_i^A$ и матрицы, полученной в п. 3.6, т.е.
   $$\langle W_{att} \rangle_i^A \left(\left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle X \rangle_i^A[0]\right] \parallel \parallel \left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle X \rangle_i^A[1]\right]\right)^T. \quad (6)$$
   3.8. Поэлементное конфиденциальное вычисление значения функции LeakyReLU от матрицы, полученной в п. 3.7, т.е.
   $$\text{LeakyReLu}\left(\left(\langle W_{att} \rangle_i^A \left(\left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle C \rangle_i^A[0]\right] \parallel \parallel \left(\langle X \rangle_i^A \cdot \left(\langle W \rangle_i^A\right)^T\right)\left[\langle C \rangle_i^A[1]\right]\right)^T\right)_j\right), \; i \in \{0,1\}, \; j \in \{0, R-1\} \quad (7)$$
   3.9. Создание матрицы $\langle E \rangle_i^A$, размерность которой совпадает с размерностью $\langle A \rangle_i^A$, и заполнение $\langle E \rangle_i^A$ нулями.
   3.10. Заполнение матрицы $\langle E \rangle_i^A$ посредством протокола конфиденциальной записи элемента массива (т.е. элементы матрицы, полученной в п. 3.8, будут записаны в E в соответствии с координатами, указанными в матрице C, а результат будет получен в виде $\langle E \rangle_i^A$).
   3.11. Конфиденциальное вычисление значения функции Softmax от каждой строки матрицы $\langle E \rangle_i^A$:
   $$\langle W_\alpha \rangle_i^A = \text{softmax}\left(\left(\langle E \rangle_i^A\right)_j\right), \; j \in \{0, V-1\}. \quad (8)$$

4. Сервер S_i вычисляет разделенную по АСРС матрицу вложений $\langle H\rangle_i^A$ с применением протокола конфиденциального умножения матриц:

$$\langle H\rangle_i^A = \left(\langle \tilde{A}\rangle_i^A\right)^T \cdot \langle W_\alpha\rangle_i^A \cdot \langle X\rangle_i^A \cdot \left(\langle W\rangle_i^A\right)^T. \quad (9)$$

5. Если в рассматриваемой ГНС с механизмом внимания применяется механизм многомерного внимания (multi-head attention), то серверы выполняют перечисленные операции необходимое количество раз с разными матрицами W и W_{att}, а затем локально усредняют полученные результаты.

4. Модель нарушителя и методология доказательства безопасности

Система КМО «КонфГраф» допускает присутствие пассивного нарушителя, который следует протоколу, но может собирать данные для их анализа с целью нарушения конфиденциальности информации (такого нарушителя нередко называют получестным). Кроме того, предполагается, что каждый из облачных провайдеров заинтересован в сохранении своей репутации и не будет вступать в сговор с другим с целью нарушения конфиденциальности обрабатываемых им данных. К защищаемой информации относятся матрицы X, A, C, W и W_{att}. Допускается, что нарушителю могут стать известны размерности указанных матриц, что не приведет к нарушению конфиденциальности информации. Между всеми участниками протокола существуют защищенные каналы связи с гарантированной криптографической стойкостью механизмов шифрования и аутентификации, ключи для которых распределены заранее и неизвестны нарушителю.

Для доказательства безопасности протоколов БМВ, как правило, прибегают к методологии универсальной компонуемости (UC – universal composability), в соответствии с которой сравниваются два режима работы криптографического протокола: выполнение его в реальном мире и в идеальном. Под реальным миром понимают такие условия, при которых выполнение протокола БМВ происходит привычным способом: существуют несколько участников, взаимодействующих друг с другом, а также нарушитель и некоторое окружение, предоставляющее участникам исходные данные и получающее от них результат (рис. 1). В идеальном мире все вычисления выполняет третья сторона, пользующаяся неограниченным доверием других участников (в реальности такого быть не может), а действия нарушителя моделирует специальный алгоритм – симулятор. Основная идея сравнения реального и идеального миров состоит в том, что в случае безопасного протокола БМВ нарушитель в реальном мире может добыть информации не более, чем симулятор в идеальном мире. Верно и обратное: если существует такой симулятор, который может смоделировать действия нарушителя так, что результат протокола и все промежуточные вычисления для окружения будут статистически неразличимы от соответствующих в реальном мире, то такой протокол БМВ является безопасным. Иными словами, окружение не сможет «отличить» реальный мир от идеального.

Рис. 1. Схематичное представление реального (а) и идеального (б) миров

Таким образом, применительно к предлагаемому в настоящей работе комплекту протоколов системы КМО «КонфГраф» требуется доказать следующую теорему:

Теорема 1. Система «КонфГраф» безопасно реализует применение ГНС с механизмом внимания посредством протоколов БМВ в предположении о наличии пассивного нарушителя, одновременно компрометирующего не более одного участника протокола. Доказательство проведем ниже, в п. 6, путем демонстрации существования симулятора, отвечающего условиям безопасности протокола БМВ в парадигме реального / идеального миров.

5. Протоколы системы «КонфГраф»

Приведём описания протоколов системы КМО «КонфГраф».

Протокол подготовки данных. Перед началом вычислений владельцу модели требуется преобразовать матрицу смежности графа A в форму списка координат, например:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad (10)$$

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 4 & 4 & 5 & 5 \\ 1 & 2 & 3 & 0 & 2 & 0 & 1 & 4 & 5 & 0 & 2 & 5 & 2 & 4 \end{pmatrix}$$

Затем клиент и владелец модели применяют АСРС для представления защищаемых матриц в виде двух частей секрета:

$$X = (\langle X\rangle_0^A + \langle X\rangle_1^A) \bmod q,$$
$$A = (\langle A\rangle_0^A + \langle A\rangle_1^A) \bmod q,$$
$$C = (\langle C\rangle_0^A + \langle C\rangle_1^A) \bmod q,$$
$$W = (\langle W\rangle_0^A + \langle W\rangle_1^A) \bmod q,$$
$$W_{att} = (\langle W_{att}\rangle_0^A + \langle W_{att}\rangle_1^A) \bmod q. \quad (11)$$

После подготовки исходных данных для работы «КонфГраф» клиент и владелец модели передают серверам S_i, $i \in \{0,1\}$ $\langle X\rangle_i^A$ и $\langle A\rangle_i^A$, $\langle C\rangle_i^A$, $\langle W\rangle_i^A$ и $\langle W_{att}\rangle_i^A$ соответственно.

Коррелированный протокол передачи с забыванием [8] – специальный вариант протокола передачи с забыванием, в котором участники на выходе получают доли произведения секретов, которые они передают на вход протокола.

Входные данные: отправитель S хранит ℓ-битное число a, получатель R хранит ℓ-битное число b, g – образующий элемент циклической группы $\mathbb{Z}_q$, q – большое простое число, κ – параметр безопасности.

Результат: S вычисляет значение $\langle c\rangle_S^A$, не получая при этом никакой информации о значении b, а R вычисляет значение $\langle c\rangle_R^A$, не получая при этом никакой информации о значении b, причем $\left(\langle c\rangle_S^A + \langle c\rangle_R^A\right) \bmod 2^\ell = a \cdot b$.

Протокол:

1. S случайно выбирает κ бит $s = [s_0,\ldots,s_{\kappa-1}]$.
2. R случайно выбирает κ пар κ-битных чисел (k_i^0, k_i^1), $i \in \{0,\kappa-1\}$.
3. Посредством выполнения κ протоколов передачи с забыванием, в которых при $i \in \{0,\kappa-1\}$ S играет роль получателя с битом выбора s_i, а R играет роль отправителя с парой сообщений (k_i^0, k_i^1), S выбирает κ чисел $k_i^{s_i}$.
4. R вычисляет $t_i = G(k_i^0)$ и $u_i = t_i \oplus G(k_i^1) \oplus b$, где $G(x)$ – любой генератор псевдослучайных последовательностей (ГПСП), расширяющий входную κ-битную последовательность в ℓ-битную, t_i – i-й столбец матрицы T размерности $\ell \times \kappa$, и отправляет S все полученные u_i, $i \in \{0,\kappa-1\}$.
5. S рассчитывает $q^i = (s_i \cdot u^i) \oplus G(k_i^{s_i})$, где q^i – i-й столбец матрицы Q размерности $\ell \times \kappa$.
6. S для $j \in \{0,\ell-1\}$ рассчитывает $y_j = f_j(H(q_j)) \oplus H(q_j \oplus s)$, где $f_j(x) = (a \cdot 2^j + x) \bmod 2^\ell$ – функция корреляции, $H(x)$ – корреляционно стойкая криптографическая хэш-функция (в данной работе применяется хэш-функция SHA-3), q_j – j-я строка матрицы Q, и отправляет полученные значения R.
7. R определяет результат, как $\langle c\rangle_R^A = \sum_{j=0}^{\ell-1}(y_j \cdot \sigma_j) \oplus H(t_j)$, а S – $\langle c\rangle_S^A = \sum_{j=0}^{\ell-1} - H(q_j)$.

Конфиденциальное умножение матриц – протокол, позволяющий двум участникам получить доли элементов матрицы, равной произведению двух матриц, доли которых они передают на вход протокола.

Входные данные: разделенные по АСРС матрицы $\langle X\rangle_i^A$, $\langle Y\rangle_i^A$ размерностей $(r_0;r_1)$ и $(r_1;r_2)$ соответственно, ℓ – размерность чисел в битах, q – большое простое число, κ – статистический параметр безопасности.

Результат: матрица $\langle Z\rangle_i^A$ такая, что $Z = \sum_i \langle Z\rangle_i^A$ и $Z = X \cdot Y$.

Фаза предварительных вычислений:

1. S_i случайно выбирают 2 матрицы $\langle A\rangle_i^A$ и $\langle B\rangle_i^A$, состоящие из целых чисел в интервале $[0;2^\ell-1]$. Размерности указанных матриц совпадают с размерностями $\langle X\rangle_i^A$ и $\langle Y\rangle_i^A$ соответственно.
2. S_0 и S_1 выполняют коррелированный протокол «забывчивой передачи», в котором S_0 – отправитель, а S_1 – получатель, а в результате серверы получают разделенную по АСРС матрицу $\langle U\rangle_i^A$. Элемент матрицы $(\langle U\rangle_i^A)_{j,k}$, $j \in \{0,r_0-1\}$, $k \in \{0,r_2-1\}$ рассчитывается следующим образом:

$$(\langle U\rangle_i^A)_{j,k} = \sum_{l=0}^{r1-1} \text{C–OT}_\ell^\ell\left((\langle U\rangle_0^A)_{j,l}, (\langle B\rangle_1^A)_{l,k}, q, \kappa\right) = \sum_{l=0}^{r1-1}\sum_{i=0}^{l-1} s_i \bmod 2^\ell,\ s_i \in_R \mathbb{Z}_{2\ell},$$

$$(\langle U\rangle_1^A)_{j,k} = \sum_{l=0}^{r1-1} \text{C–OT}_\ell^\ell\left((\langle U\rangle_0^A)_{j,l}, (\langle B\rangle_1^A)_{l,k}, q, \kappa\right) = \sum_{l=0}^{r1-1}\sum_{i=0}^{l-1}\left((\langle B\rangle_1^A)_{l,k}[i] \cdot (\langle A\rangle_0^A)_{j,l} \cdot 2^i - s_i\right) \bmod 2^\ell. \quad (12)$$

3. S_0 и S_1 выполняют коррелированный протокол «забывчивой передачи», в котором S_1 – отправитель, а S_0 – получатель, а в результате серверы получают разделенную по АСРС матрицу $\langle V\rangle_i^A$. Элемент матрицы $(\langle V\rangle_i^A)_{j,k}$, $j \in \{0,r_0-1\}$, $k \in \{0,r_2-1\}$ рассчитывается следующим образом:

$$(\langle V\rangle_0^A)_{j,k} = \sum_{l=0}^{r1-1} \text{C–OT}_\ell^\ell\left((\langle A\rangle_1^A)_{j,l}, (\langle B\rangle_0^A)_{l,k}, q, \kappa\right) = \sum_{l=0}^{r1-1}\sum_{i=0}^{l-1}\left((\langle B\rangle_0^A)_{l,k}[i] \cdot (\langle A\rangle_1^A)_{j,l} \cdot 2^i - s_i\right) \bmod 2^\ell,$$

$$(\langle V\rangle_0^A)_{j,k} = \sum_{l=0}^{r1-1} \text{C–OT}_\ell^\ell\left((\langle A\rangle_1^A)_{j,l}, (\langle B\rangle_0^A)_{l,k}, q, \kappa\right) = \sum_{l=0}^{r1-1}\sum_{i=0}^{l-1} s_i \bmod 2^\ell,\ s_i \in_R \mathbb{Z}_{2\ell}. \quad (13)$$

4. Расчет матрицы $\langle C\rangle_i^A$ производится серверами следующим образом:

$$\langle C\rangle_i^A = \langle A\rangle_i^A \cdot \langle B\rangle_i^A + \langle U\rangle_i^A + \langle V\rangle_i^A. \quad (14)$$

Протокол:

1. S_i вычисляет разности $\langle X\rangle_i^A - \langle A\rangle_i^A$, $\langle Y\rangle_i^A - \langle B\rangle_i^A$ и отправляет результат серверу S_{1-i}.
2. S_i восстанавливает значения разностей:

$$X - A = (\langle X\rangle_i^A - \langle A\rangle_i^A) + (\langle X\rangle_{1-i}^A - \langle A\rangle_{1-i}^A),$$
$$Y - B = (\langle Y\rangle_i^A - \langle B\rangle_i^A) + (\langle Y\rangle_{1-i}^A - \langle B\rangle_{1-i}^A). \quad (15)$$

3. S_i вычисляет результат:

$$\langle Z\rangle_i^A = -i\cdot(X-A)(Y-B) + \langle A\rangle_i^A \cdot \cdot (Y-B) + \langle B\rangle_i^A \cdot (X-A) + \langle C\rangle_i^A. \quad (16)$$

Покажем безопасность этого протокола. Для этого приведем описание симулятора $S_{\text{MatrixMult}}$:

1. Симулятор $S_{\text{MatrixMult}}$ случайно выбирает $\langle A\rangle_t^A$, $\langle A\rangle_{1-t}^A$ и $\langle B\rangle_t^A$, $\langle B\rangle_{1-t}^A$, где n – индекс скомпрометированного участника.
2. Дважды используя симулятор $S_{\text{C-OT}}$ «коррелированного» протокола передачи с забыванием, $S_{\text{MatrixMult}}$ получает ансамбль величин, статистически неразличимых от получаемых в результате шага 2 и 3 фазы предварительных вычислений протокола.
3. Симулятор $S_{\text{MatrixMult}}$ рассчитывает $\langle C\rangle_t^A$ и $\langle C\rangle_{1-t}^A$ в соответствии с выражением (14).
4. Симулятор $S_{\text{MatrixMult}}$ рассчитывает $\langle X\rangle_t^A - \langle A\rangle_t^A$, $\langle Y\rangle_t^A - \langle B\rangle_t^A$ и $\langle X\rangle_{1-t}^A - \langle A\rangle_{1-t}^A$, $\langle Y\rangle_{1-t}^A - \langle B\rangle_{1-t}^A$ и восстанавливает указанные разности.
5. Симулятор $S_{\text{MatrixMult}}$ вычисляет результат в соответствии с выражение (16).

Конфиденциальное вычисление значения функции ReLU – протокол, позволяющий двум участникам получить доли значения функции ReLU, широко используемой в качестве функции активации нейронов в ГНС, подав на вход доли некоторого значения аргумента этой функции.

Входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, q – большое простое число, ℓ – размерность чисел в битах.

Результат: серверы S_i рассчитывают разделенное по АСРС число $\langle s\rangle_i^A$, где

$$s = \begin{cases} x, x \geq 0, \\ 0, x < 0. \end{cases}.$$

Протокол:

1. Серверы S_i применяют протокол конфиденциальной проверки числа на положительность:

$$\langle s'\rangle_i^A = \text{GTZ}\left(\langle a\rangle_i^A\right). \quad (17)$$

2. Серверы S_i вычисляют результат при помощи протокола конфиденциального умножения чисел:

$$\langle s\rangle_i^A = \langle a\rangle_i^A \cdot \langle s'\rangle_i^A. \quad (18)$$

Покажем безопасность этого протокола. Для этого приведем описание симулятора S_{ReLU}:

1. Для моделирования п. 1 протокола S_{ReLU} применяет S_{GTZ}, который описан в [9].
2. Затем S_{ReLU} применяет S_{Mult}, который описан в [10], для моделирования п. 2 протокола.

Конфиденциальное вычисление значения функции активации LeakyReLU – протокол, функция которого полностью аналогичная предыдущему, за исключением того, что вычисляется значение функции LeakyReLU.

Входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, α – отрицательный уклон (англ. negative slope), q – большое простое число, ℓ – размерность чисел в битах.

Результат: серверы S_i рассчитывают разделенное по АСРС число $\langle s\rangle_i^A$, где

$$s = \begin{cases} x, x \geq 0, \\ \alpha x, x < 0. \end{cases}.$$

Протокол:

1. Серверы S_i конфиденциально вычисляют значение функции ReLU:

$$\langle s\rangle_i^A = \text{ReLU}\left(\langle a\rangle_i^A\right). \quad (19)$$

2. Серверы S_i локально преобразуют α к виду числа с фиксированной точкой и длиной дробной части f, представленному в поле $\mathbb{Z}_q$:

$$\alpha' = [2^f \cdot \alpha] \bmod q. \quad (20)$$

3. Серверы S_i вычисляют результат посредством протокола конфиденциального усечения числа:

$$\langle s'\rangle_i^A = \text{Trunc}\left(\langle s\rangle_i^A \cdot (2^f - \alpha') + \alpha' \cdot \langle s\rangle_i^A, \ell, f\right). \quad (21)$$

Покажем безопасность этого протокола. Для этого приведем описание симулятора $S_{\text{LeakyReLU}}$:

1. $S_{\text{LeakyReLU}}$ применяет описанный выше S_{ReLU} для моделирования 1 шага протокола.
2. Симулятор $S_{\text{LeakyReLU}}$ преобразует α к α' в соответствии с выражением, указанным в п. 2 протокола.
3. Последним $S_{\text{LeakyReLU}}$ применяет S_{Trunc}, который описан в [9], для моделирования п. 3 протокола.

Другие протоколы. Кроме перечисленных выше, в системе «КонфГраф» используются в качестве примитивов следующие готовые протоколы БМВ:

1. Конфиденциальное согласование случайного бита [9]: входные данные: q – большое простое число, результат: серверы S_i, $i \in \{0,1\}$ согласуют случайный бит $\langle d\rangle_i^A$, разделенный по АСРС, т.е. $(\langle d\rangle_0^A + \langle d\rangle_1^A) \bmod q \in \{0,1\}$.
2. Конфиденциальное согласование случайных величин [9]: входные данные: q – большое простое число, κ – статистический параметр безопасности, α – количество выходных случайных бит, k – разрядность выходного числа, результат: серверы S_i, $i \in \{0,1\}$ согласуют α случайных бит $\langle r_0\rangle_i^A,\ldots,\langle r_{\alpha-1}\rangle_i^A$, разделенных по АСРС, а также α-битное число $\langle r\rangle_i^A = \sum_{j=0}^{\alpha-1} 2^j \cdot \langle r_j\rangle_i^A$ и $(k+\kappa-\alpha)$-битное $\langle r'\rangle_i^A$.
3. Конфиденциальное префиксное умножение чисел [9]: входные данные: $(\langle a\rangle_i^A)_0,\ldots,(\langle a\rangle_i^A)_{l-1}$ – массив входных ненулевых чисел, разделенных по АСРС, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ рассчитывают разделенный по АСРС массив значений $(\langle p\rangle_i^A)_j = \prod_{k=0}^{j} (\langle a\rangle_i^A)_k$, $j \in \{0,\ell-1\}$.

4. Конфиденциальное вычисление остатка от деления на 2 [9]: входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ рассчитывают разделенное по АСРС число $\langle a_0\rangle_i^A = \langle a \bmod 2\rangle_i^A$.
5. Протокол конфиденциального сравнения с известным значением [11]: входные данные: a – число, известное обоим серверам, $(\langle b\rangle_i^A)_0,\ldots,(\langle b\rangle_i^A)_{l-1}$ – входное число, представленное в форме l-битной последовательности, каждый бит которой разделён с использованием АСРС, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ вычисляют разделенный по АСРС бит $\langle u\rangle_i^A$, где $u = (a<b)?1:0$.
6. Конфиденциальное вычисление остатка от деления на 2^m [11]: входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, m – общеизвестное целое число, $m \in \{1,\ell-1\}$, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ вычисляют разделенное по АСРС число $\langle a'\rangle_i^A = \langle a \bmod 2^m\rangle_i^A$.
7. Конфиденциальное усечение числа [9]: входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, m – общеизвестное целое число, $m \in \{1,\ell-1\}$, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ вычисляют разделенное по АСРС число $\langle d\rangle_i^A = \langle[\frac{a}{2^m}]\rangle_i^A$.
8. Конфиденциальное вероятностное усечение числа [9]: входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, m – общеизвестное целое число, $m \in \{1,\ell-1\}$, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ вычисляют разделенное по АСРС число $\langle d\rangle_i^A = \langle[\frac{a}{2^m}] + u\rangle_i^A$, где $u \in \{0,1\}$.
9. Протокол передачи с забыванием (oblivious transfer) [12]: входные данные: отправитель S хранит 2 сообщения (m_0,m_1), получатель R – бит выбора $\sigma \in \{0,1\}$, g – генератор циклической группы $\mathbb{Z}_q$, q – большое простое число, результат: R определяет значение m_σ, причем R не получает никакой информации о значении $m_{1-\sigma}$, а S – о значении бита выбора σ.
10. Конфиденциальное умножение чисел [10]: входные данные: разделенные по АСРС числа $\langle x\rangle_i^A$, $\langle y\rangle_i^A$, $i \in \{0,1\}$, ℓ – размерность чисел в битах, q – большое простое число, κ – статистический параметр безопасности, результат: число $\langle z\rangle_i^A$, $i \in \{0,1\}$, причем $z = \sum_i \langle z\rangle_i^A$ и $z = x \cdot y$.
11. Конфиденциальное умножение чисел с фиксированной точкой [13]: входные данные: разделенные по АСРС числа $\langle x\rangle_i^A,\langle y\rangle_i^A$, $i \in \{0,1\}$, f – длина дробной части в битах, ℓ – полная длина чисел в битах, q – большое простое число, κ – статистический параметр безопасности, результат: число $\langle z\rangle_i^A$, $i \in \{0,1\}$, причем $z = \sum_i \langle z\rangle_i^A$ и $z = x \cdot y$.
12. $\mathrm{AllOr}\left((\langle d\rangle_i^A)_{k-1},\ldots,(\langle d\rangle_i^A)_0\right)$ [14]: входные данные: серверы S_i, $i \in \{0,1\}$ хранят массив $(\langle d\rangle_i^A)_0,\ldots,(\langle d\rangle_i^A)_{k-1}$ длиной k, разделенный по АСРС, причем $\left((\langle d\rangle_0^A)_j + (\langle d\rangle_1^A)_j\right) \bmod q \in \{0,1\}$, q – большое простое число, результат: S_i, $i \in \{0,1\}$ согласуют разделенный по АСРС массив $(\langle b\rangle_i^A)_0,\ldots,(\langle b\rangle_i^A)_{2^k-1}$ длиной 2^k и состоящий из результата логического «ИЛИ» от всех возможных комбинаций входных бит, разделенных по АСРС.
13. Конфиденциальный доступ к элементам массива [14]: входные данные: массив a, состоящий из m элементов и разделенный по АСРС между S_i, $i \in \{0,1\}$, т.е. $\left[(\langle a\rangle_i^A)_0,(\langle a\rangle_i^A)_1,\ldots,(\langle a\rangle_i^A)_{m-1}\right]$, индекс j, также разделенный по АСРС, – $\langle j\rangle_i^A$, q – большое простое число, результат: элемент b, разделенный по АСРС между S_i, $i \in \{0,1\}$, причем $b = a_j$.
14. Конфиденциальная запись элемента массива [14]: входные данные: массив a, состоящий из m элементов и разделенный по АСРС между S_i, $i \in \{0,1\}$, т.е. $\left[(\langle a\rangle_i^A)_0,(\langle a\rangle_i^A)_1,\ldots,(\langle a\rangle_i^A)_{m-1}\right]$, $i \in \{0,1\}$, индекс j, также разделенный по АСРС, – $\langle j\rangle_i^A$, $\langle w\rangle_i^A$ – записываемое значение, q – большое простое число, результат: массив d, состоящий из m элементов и разделенный по АСРС между S_i, $i \in \{0,1\}$, причем
$$d_k = \begin{cases} a_k, k \neq j, \\ w, k = j. \end{cases}$$
15. Конфиденциальная проверка числа на отрицательность [9]: входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ рассчитывают разделенное по АСРС число $\langle s'\rangle_i^A$, где $s' = a < 0? 1:0$.
16. Конфиденциальная проверка числа на положительность [9]: входные данные: $\langle a\rangle_i^A$ – входное число, разделенное по АСРС, q – большое простое число, ℓ – размерность чисел в битах, результат: серверы S_i, $i \in \{0,1\}$ рассчитывают разделенное по АСРС число $\langle s'\rangle_i^A$, где $s' = a > 0? 1:0$.
17. Конфиденциальное вычисление значения функции Softmax [15]: входные данные: массив x, состоящий из m элементов и разделенный по АСРС между участниками протокола $\left[(\langle x\rangle_i^A)_0,(\langle x\rangle_i^A)_1,\ldots,(\langle x\rangle_i^A)_{m-1}\right]$, $i \in \{0,1\}$, $r = 2^{l-f}$ – количество итераций протокола, f – длина дробной части в битах, ℓ – размерность чисел в битах, результат: массив g, состоящий из m элементов и разделенный по АСРС между участниками протокола $\left[(\langle g\rangle_i^A)_0,(\langle g\rangle_i^A)_1,\ldots,(\langle g\rangle_i^A)_{m-1}\right]$, где
$$g_j \approx \frac{e^{x_j}}{\sum_{k=0}^{m-1} e^{x_k}}, j \in \{0,m-1\}.$$

Безопасность перечисленных протоколов в рассматриваемой модели нарушителя доказана в соответствующих источниках, поэтому далее они используются в качестве готовых структурных элементов с доказанными свойствами безопасности.

6. Анализ безопасности протоколов системы «КонфГраф»

Для доказательства безопасности протоколов системы КМО «КонфГраф» в целом воспользуемся методологией универсальной компонуемости: окружение не отличит реальный мир от идеального, если протокол безопасен. При условии безопасности протоколов, лежащих в основе системы, доказательство ее безопасности оказывается достаточно простым. Для демонстрации безопасности протоколов системы «КонфГраф» опишем алгоритм работы симулятора в модели получестного противника. Пусть $n \in \{0,1\}$ – номер скомпрометированного нарушителем сервера. Тогда:

1. Симулятор $S_{\text{КонфГраф}}$ локально вычисляет $\left(\langle W\rangle_n^A\right)^T$.
2. Симулятор $S_{\text{КонфГраф}}$ пользуется симулятором $S_{\text{MatrixMult}}$ для моделирования протокола конфиденциального умножения матриц $\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T$.
3. Симулятор $S_{\text{КонфГраф}}$ использует $S_{\text{ArrayAccess}}$ для моделирования протокола конфиденциального доступа к элементам массива для составления матриц

$$\left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[0]\right]$$
$$\text{и } \left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[1]\right].$$

4. Симулятор $S_{\text{КонфГраф}}$ производит построчную конкатенацию матриц $\left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[0]\right]$ и $\left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[1]\right]$, а затем транспонирование результата

$$\left(\left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[0]\right] \parallel \right.$$
$$\left. \parallel \left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[1]\right]\right)^T.$$

5. Симулятор $S_{\text{КонфГраф}}$ использует $S_{\text{MatrixMult}}$ для моделирования протокола конфиденциального умножения матриц для вычисления:

$$\langle W_{att}\rangle_n^A \cdot \left(\left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[0]\right] \parallel \right.$$
$$\left. \parallel \left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[1]\right]\right)^T. \quad (22)$$

6. Симулятор $S_{\text{КонфГраф}}$ использует $S_{\text{LeakyReLU}}$ для моделирования протокола конфиденциального вычисления значения функции LeakyReLU:

$$\text{LeakyReLU}\left(\left(\langle W_{att}\rangle_n^A \cdot \left(\left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[0]\right] \parallel \right.\right.\right.$$
$$\left.\left.\left. \parallel \left(\langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T\right)\left[\langle C\rangle_n^A[1]\right]\right)^T\right)_j\right), j \in \{0,R-1\}. \quad (23)$$

7. Симулятор $S_{\text{КонфГраф}}$ создает нулевую матрицу $\langle E\rangle_n^A$ размерностью, совпадающей с $\langle A\rangle_n^A$.
8. Симулятор $S_{\text{КонфГраф}}$ использует $S_{\text{ArrayWrite}}$ для моделирования протокола конфиденциальной записи элементов массива для заполнения матрицы $\langle E\rangle_n^A$ значениями, полученными от симулятора $S_{\text{LeakyReLU}}$ (см. п. 6), в соответствии с координатами в матрице C.
9. Симулятор $S_{\text{КонфГраф}}$ использует S_{softmax} для моделирования протокола конфиденциального вычисления значения функции Softmax:

$$\langle W_\alpha\rangle_n^A = \text{softmax}\left(\left(\langle E\rangle_n^A\right)_j\right), j \in \{0,V-1\}. \quad (24)$$

10. Симулятор $S_{\text{КонфГраф}}$ трижды пользуется симулятором $S_{\text{MatrixMult}}$ для моделирования протокола конфиденциального умножения матриц

$$\langle H\rangle_n^A = \left(\langle A\rangle_n^A\right)^T \cdot \langle W_\alpha\rangle_n^A \cdot \langle X\rangle_n^A \cdot \left(\langle W\rangle_n^A\right)^T. \quad (25)$$

В итоге, результатом каждого из шагов работы симулятора $S_{\text{КонфГраф}}$ будет ансамбль случайных величин, которые статистически неразличимы от величин, получаемых в реальном мире, что говорит о том, что окружение не сможет «отличить» реальный мир от идеального, следовательно, систему КМО «КонфГраф» можно считать безопасной в рассматриваемой модели угроз.

7. Заключение

В работе предложен комплекс криптографических протоколов для реализации системы конфиденциального машинного обучения на основе графовых нейронных сетей с механизмом внимания, названной нами «КонфГраф», и доказана безопасность как отдельных протоколов, так и системы в целом.

Система «КонфГраф» позволяет осуществлять конфиденциальное применение графовых нейронных сетей с механизмом внимания при компрометации одного из двух провайдеров облачных вычислений. Система основана на использовании арифметической схемы разделения секрета, однако в ее составе нашли применение и другие базовые для БМВ криптографические примитивы, например, протокол передачи с забыванием. Благодаря использованию таких примитивов стала возможна разработка протоколов для конфиденциального вычисления более сложных математических функций, например, LeakuReLU и Softmax, являющихся базой современных моделей искусственных нейронных сетей. Собрав все протоколы в единую систему и согласовав их работу, удалось создать систему для конфиденциального вычисления функции, задаваемой уравнением (4), и доказать ее безопасность в рассматриваемой модели угроз.

Литература

1. *Younes L. Introduction to Machine Learning. arXiv, 2024. – 649 p. DOI: https://doi.org/10.48550/arXiv.2409.02668.*
2. *Brody S., Alon U., Yahav E. How Attentive are Graph Attention Networks? arXiv, 2022. – 26 p. DOI: https://doi.org/10.48550/arXiv.2105.14491.*
3. *Liao Y. Zhang X. Ferrie C. Graph Neural Networks on Quantum Computers. arXiv, 2024. – 50 p. DOI: https://doi.org/10.48550/arXiv.2405.17060.*
4. *Xu R., Baracaldo N., Joshi J. Privacy-Preserving Machine Learning: Methods, Challenges and Directions. aXiv, 2021. – 40 p. DOI: http://dx.doi.org/10.48550/arXiv.2108.04417*
5. *Запечников С. В. Модели и алгоритмы конфиденциального машинного обучения // Безопасность информационных технологий. Т. 27. № 1. 2020. С. 51–67. DOI: https://doi.org/10.26583/bit.2020.1.05.*
6. *Запечников С. В. Конфиденциальное машинное обучение на основе четырехсторонних протоколов безопасных вычислений // Безопасность информационных технологий. Т. 29. № 2. 2022. С. 46–56. DOI: http://dx.doi.org/10.26583/bit.2022.2.04.*
7. *Mishra P., Lehmkuhl R., Srinivasan A., Zheng W., Popa R. A. Delphi: A cryptographic inference service for neural networks. Proc. of USENIX Security 2020 (USENIX Security Symposium). URL: https://eprint.iacr.org/2020/050.pdf.*
8. *Liu X., Wu B., Yuan X., Yi X. Leia: A Lightweight Cryptographic Neural Network Inference System at the Edge. IEEE Transactions on Information Forensics and Security, vol. 17, 2022, pp. 237–252. DOI: https://doi.org/10.1109/TIFS.2021.3138611.*
9. *Catrina O., de Hoogh S. Improved Primitives for Secure Multiparty Integer Computation. Security and Cryptography for Networks. Lecture Notes in Computer Science, vol 6280, Springer, 2010, pp 182–199. DOI: https://doi.org/10.1007/978-3-642-15317-4_13.*
10. *Patra A., Schneider T., Suresh A., Yalame H. ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation. Cryptology ePrint Archive, 2020. – 29 p. DOI: https://eprint.iacr.org/2020/1225.*
11. *Catrina O. Round-Efficient Protocols for Secure Multiparty Fixed-Point Arithmetic. Proc. of 12th IEEE International Conference on Communications, 2018, pp 431–436. DOI: https://doi.org/10.1109/ICComm.2018.8484794.*
12. *Yadav V., Andola N., Verma S, Venkatesan S. A Survey of Oblivious Transfer Protocol. ACM Comput. Surv., 2022. – 37 p. DOI: https://doi.org/10.1145/3503045.*
13. *Catrina O., Saxena A. Secure Computation With Fixed-Point Numbers. Lecture Notes in Computer Science, vol. 6052, Springer, 2010, pp 35–50. DOI: https://doi.org/10.1007/978-3-642-14577-3_6.*
14. *Blanton M., Kang A., Yuan C. Improved Building Blocks for Secure Multi-Party Computation based on Secret Sharing with Honest Majority. Cryptology ePrint Archive, 2019. – 26 p. URL: https://eprint.iacr.org/2019/718.*
15. *Zheng Y., Zhang Q., Chow S., Peng Y., Tan S., Li L., Yin S. Secure Softmax/Sigmoid for Machine-Learning Computation. Proc. of the 39th Annual Computer Security Applications Conference, 2023, pp 463–476. DOI: https://doi.org/10.1145/3627106.3627175.*

PRIVACY-PRESERVING INFERENCE OF PRE-TRAINED GRAPH NEURAL NETWORKS WITH AN ATTENTION MECHANISM

Shevchenko V. A.[3], Zapechnikov S. V.[4]

Abstract. *The article proposes a set of cryptographic protocols for privacy-preserving machine learning (PPML) system based on graph neural networks with an attention mechanism. The classification of artificial neural networks underlying deep learning is given. The main tasks of ensuring privacy that arise during the training and inference of machine learning models based on artificial neural networks are highlighted. The main cryptographic primitives underlying secure multi-party computations are described, namely secret sharing schemes, an oblivious transfer protocol. It is provided a brief description of the methodology for proving the security of cryptographic protocols, including protocols for secure multi-party computations, known as universal composability (UC-security). The main and auxiliary protocols underlying the PPML system are described and analyzed: the correlated oblivious transfer, as well as protocols for private matrix multiplication, private ReLU and LeakyReLU functions computation, and the proof of their security is provided. The rest of the protocols used in the PPML system are listed in the article with a brief description of their input and output data. The security of the PPML system as a whole is proved based on the universal composability paradigm.*

Keywords: *cryptography, information security, confidential machine learning, secure multi-party computing, graph neural networks with an attention mechanism, secret sharing schemes, transmission protocol with forgetting.*

3 Vyacheslav A. Shevchenko, Applicant, National Research Nuclear University MEPhI, Moscow, Russia. E-mail sheff-slava@mail.ru

4 Sergey V. Zapechnikov, Doctor of Technical Sciences, Associate Professor, Professor, National Research Nuclear University MEPhI, Moscow, Russia. E-mail: svzapechnikov@mephi.ru

References

1. *Younes L. Introduction to Machine Learning. arXiv, 2024. – 649 p. DOI: https://doi.org/10.48550/arXiv.2409.02668.*
2. *Brody S., Alon U., Yahav E. How Attentive are Graph Attention Networks? arXiv, 2022. – 26 p. DOI: https://doi.org/10.48550/arXiv.2105.14491.*
3. *Liao Y. Zhang X. Ferrie C. Graph Neural Networks on Quantum Computers. arXiv, 2024. – 50 p. DOI: https://doi.org/10.48550/arXiv.2405.17060.*
4. *Xu R., Baracaldo N., Joshi J. Privacy-Preserving Machine Learning: Methods, Challenges and Directions. aXiv, 2021. – 40 p. DOI: http://dx.doi.org/10.48550/arXiv.2108.04417*
5. *Zapechnikov S. V. Modeli i algoritmy konfidencial'nogo mashinnogo obuchenija // Bezopasnost' informacionnyh tehnologij. T. 27. № 1. 2020. S. 51–67. DOI: https://doi.org/10.26583/bit.2020.1.05.*
6. *Zapechnikov S. V. Konfidencial'noe mashinnoe obuchenie na osnove chetyrehstoronnih protokolov bezopasnyh vychislenij // Bezopasnost' informacionnyh tehnologij. T. 29. № 2. 2022. S. 46–56. DOI: http://dx.doi.org/10.26583/bit.2022.2.04.*
7. *Mishra P., Lehmkuhl R., Srinivasan A., Zheng W., Popa R. A. Delphi: A cryptographic inference service for neural networks. Proc. of USENIX Security 2020 (USENIX Security Symposium). URL: https://eprint.iacr.org/2020/050.pdf.*
8. *Liu X., Wu B., Yuan X., Yi X. Leia: A Lightweight Cryptographic Neural Network Inference System at the Edge. IEEE Transactions on Information Forensics and Security, vol. 17, 2022, pp. 237-252. DOI: https://doi.org/10.1109/TIFS.2021.3138611.*
9. *Catrina O., de Hoogh S. Improved Primitives for Secure Multiparty Integer Computation. Security and Cryptography for Networks. Lecture Notes in Computer Science, vol 6280, Springer, 2010, pp 182–199. DOI: https://doi.org/10.1007/978-3-642-15317-4_13.*
10. *Patra A., Schneider T., Suresh A., Yalame H. ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation. Cryptology ePrint Archive, 2020. – 29 p. DOI: https://eprint.iacr.org/2020/1225.*
11. *Catrina O. Round-Efficient Protocols for Secure Multiparty Fixed-Point Arithmetic. Proc. of 12th IEEE International Conference on Communications, 2018, pp 431-436. DOI: https://doi.org/10.1109/ICComm.2018.8484794.*
12. *Yadav V., Andola N., Verma S, Venkatesan S. A Survey of Oblivious Transfer Protocol. ACM Comput. Surv., 2022. – 37 p. DOI: https://doi.org/10.1145/3503045.*
13. *Catrina O., Saxena A. Secure Computation With Fixed-Point Numbers. Lecture Notes in Computer Science, vol. 6052, Springer, 2010, pp 35–50. DOI: https://doi.org/10.1007/978-3-642-14577-3_6.*
14. *Blanton M., Kang A., Yuan C. Improved Building Blocks for Secure Multi-Party Computation based on Secret Sharing with Honest Majority. Cryptology ePrint Archive, 2019. – 26 p. URL: https://eprint.iacr.org/2019/718.*
15. *Zheng Y., Zhang Q., Chow S., Peng Y., Tan S., Li L., Yin S. Secure Softmax/Sigmoid for Machine-Learning Computation. Proc. of the 39th Annual Computer Security Applications Conference, 2023, pp 463-476. DOI: https://doi.org/10.1145/3627106.3627175.*