УДК: 004.75, 004.41

Конфиденциальное машинное обучение с нулевым разглашением

S.V. Zapechnikov, A.Yu. Shcherbakov

Zero-Knowledge Proofs Privacy-Preserving Machine Learning

Abstract. *The article is devoted to the analysis of a new scientific field – privacy-preserving and verifiable machine learning systems. Such systems allow to generate proof of the correctness for model learning and inference and to verify the proof by the customer of inference task and third parties, which guarantees the integrity of the evaluation process. The main tool for privacy-preserving and verifiable machine computing is zero-knowledge proofs. The article provides an example of a universal purpose zero-knowledge proof system and discusses the main ideas underlying the existing implementations of privacy-preserving and verifiable machine learning systems.*

Keywords: *data mining, machine learning, deep learning, federated learning, confidentiality, secure multi-party computations, secret sharing schemes, homomorphic encryption.*

С.В. Запечников[1]
А.Ю. Щербаков[2]
[1]Доктор технических наук, профессор Института интеллектуальных кибернетических систем, Национальный исследовательский ядерный университет «МИФИ», Вице-президент по научной работе Ассоциации специалистов в области криптовалют и цифровых финансовых активов.
E-mail: SVZapechnikov@mephi.ru
[2]Доктор технических наук, профессор, главный научный сотрудник РАН (ИТМиВТ им.С.А.Лебедева), начальник Центра развития криптовалют и цифровых финансовых активов (ЦРКЦФА) ВИНИТИ РАН.
E-mail: x509@ras.ru

Аннотация. *Статья посвящена анализу положения дел в новой научной области – системах конфиденциального и проверяемого машинного обучения. Такие системы позволяют генерировать доказательства корректности вычислений и обеспечивают возможность проверки доказательства заказчиком вычислений и третьими лицами, что даёт гарантии целостности процесса вычислений. Основным инструментом конфиденциального и проверяемого машинного вычисления являются криптографические доказательства с нулевым разглашением. В статье приводится пример системы доказательства с нулевым разглашением универсального назначения, рассматриваются основные идеи, заложенные в основу существующих реализаций систем конфиденциального и проверяемого машинного обучения.*

Ключевые слова: *интеллектуальный анализ данных, машинное обучение, глубокое обучение, федеративное обучение, конфиденциальность, безопасные многосторонние вычисления, схемы разделения секрета, гомоморфное шифрование.*

ВВЕДЕНИЕ

Конфиденциальное машинное обучение (КМО) – одна из наиболее плодотворных идей в сфере компьютерных наук, появившихся за последние годы. На её реализации в настоящее время сосредоточены усилия множества исследовательских групп в ведущих университетах и IT-корпорациях по всему миру. Это научно-техническое направление возникло на стыке классического машинного обучения и криптографии [1 – 3]. Целью создания систем КМО является реализация новой функциональности – обеспечение возможности решать задачи машинного обучения при условии одновременного обеспечения конфиденциальности данных, предоставляемых для обучения модели, самих обученных моделей, а также данных, передаваемых владельцу обученной модели на этапе её применения для получения решений задач анализа данных. Среди уже имеющихся систем КМО есть конфиденциальные аналоги многих классических методов машинного обучения: метода k средних, решающих деревьев, линейной и логистической регрессий, метода опорных векторов и, конечно, многочисленных вариантов искусственных нейронных сетей (ИНС).

Решение различных задач КМО и развитие технологий в этой области весьма важно для множества областей знания, в первую очередь для аналитических исследований в области конфиденциальных финансовых технологий, когда данные для анализа являются персональными данными, конфиденциальной информацией, либо составляют банковскую тайну. Задачи КМО также необходимо решать

в области цифровых финансовых активов для анализа данных криптобирж и прогнозирования волатильности или трендов развития цифровых активов (криптовалют). Работы в области КМО составляют одну из фундаментальных основ исследований в рамках государственного задания «Исследования в области перспектив развития технологий цифровых финансовых активов (криптовалют) и распределенных реестров (блокчейн) для их применения в сфере цифровой трансформации технологий и экономики Российской Федерации».

Однако с развитием систем КМО вскоре стало очевидно, что только функции обеспечения конфиденциальности данных для многих практических применений недостаточно. Важна функция обеспечения целостности, в особенности на этапе применения модели, позволяющая заказчику вычислений убедиться в том, что обладатель модели сообщил ему корректный результат решения задачи с использованием как сообщённого заказчиком запроса, так и обученной модели. Обладатель модели также в большинстве случаев заинтересован в том, чтобы обезопасить себя от возможных претензий заказчика, связанных с качеством решения задачи. Таким образом, взаимовыгодным для заказчика вычислений и провайдера модели становится такое свойство систем КМО как проверяемость (верифицируемость) вычислений, позволяющая убедиться в целостности информации, которая служит входом и выходом решения задачи. В ряде систем обеспечивается возможность проверки вычислений не только для заказчика, но и для третьих лиц. Обобщая все аспекты описанного свойства вычислений с моделями КМО, мы предлагаем использовать для него термин «проверяемая целостность», а для систем КМО, обладающих свойствами конфиденциальности и проверяемой целостности, – термин «конфиденциальное и проверяемое машинное обучение» (КПМО).

ПОСТАНОВКА ЗАДАЧИ

Напомним общую идею КМО [3]. Предполагается, что для решения задач пользователя, связанных с анализом данных и обнаружением в них закономерностей, используется некоторая модель машинного обучения. При этом процессы обучения и применения модели происходят дистанционно, т.е. владелец данных и получатель результата с одной стороны, а также обладатель модели с другой стороны – разные лица, взаимодействующие по дистанционным каналам. Обе взаимодействующие стороны заинтересованы в сохранении конфиденциальности своей информации: первый из них желает сохранить в секрете данные, передаваемые для обучения модели, либо запросы, подаваемые к уже обученной модели, второй – параметры обученной модели. Схемы взаимодействия участников показаны на рис. 1 и 2.

В случае, если участники взаимодействия ожидают от системы КМО не только конфиденциальности данных, но и обеспечения свойства проверяемой целостности, им

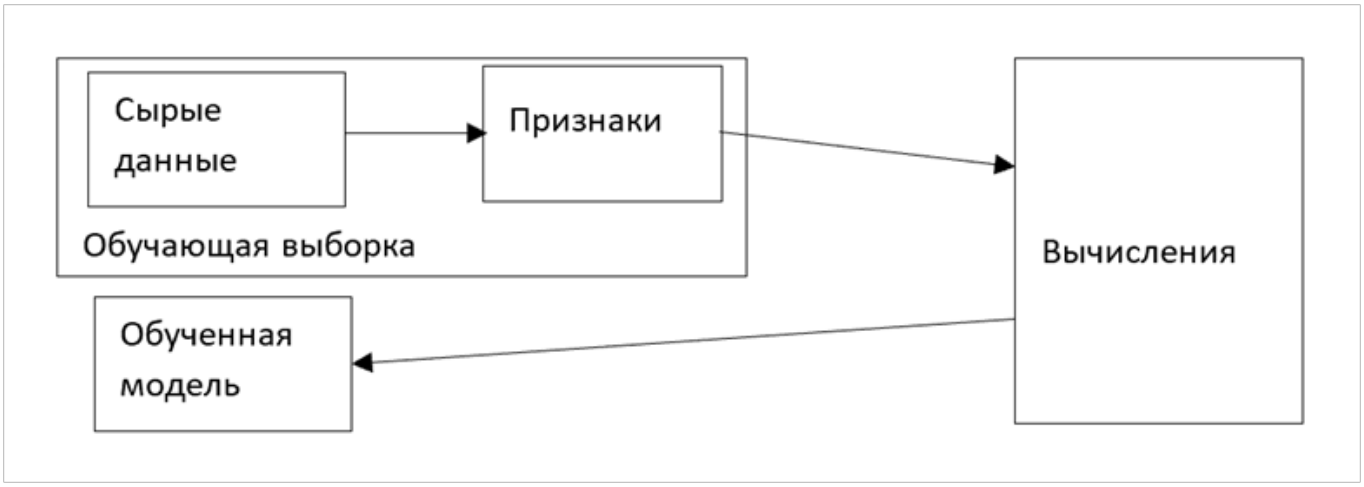

Рис. 1. Взаимодействие участников КМО на этапе обучения

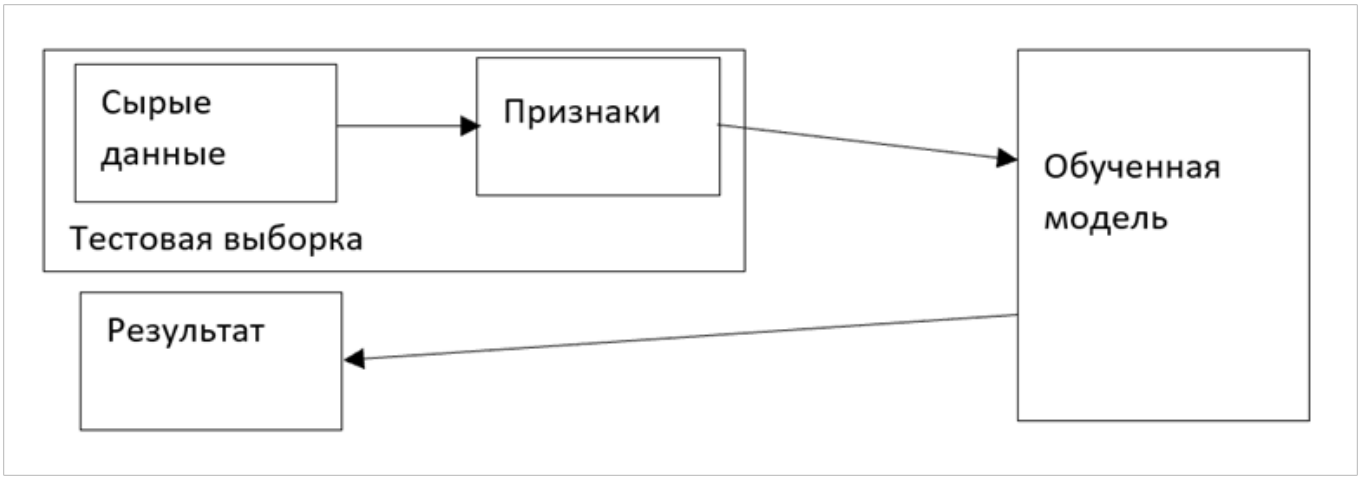

Рис. 2. Взаимодействие участников КМО на этапе применения обученной модели

необходимо дооснастить её дополнительными криптографическими механизмами, чтобы получить систему КПМО. От участников такой системы ожидается следующая функциональность:

- генерация доказательств корректного выполнения алгоритмов обладателем модели.
- прием результатов решения задачи и проверка корректности её решения заказчиком (а если необходимо, то и третьими лицами).

Однако для этого предварительно нужно создать специальный образ модели машинного обучения, называемый коммитментом (commitment). Это специальная структура данных, построенная с использованием криптографических средств, в частности, криптографических функций хэширования или иных однонаправленных функций, которая однозначно связана с исходной конструкцией (например, с ИНС или с решающим деревом), но из которой невозможно восстановить параметры исходной структуры (синаптические веса ИНС или предикаты, ассоциированные с вершинами решающего дерева, соответственно). Коммитменты в большинстве случаев обладают свойствами полноты (completeness), состоятельности (soundness) и статистического сокрытия (hiding) исходных данных.

Обобщая всё сказанное, приведём формальное определение системы КПМО на примере свёрточной ИНС (определение приводится по работе [4].

Пусть W – множество параметров ИНС, $X \in F^{n \times n \times ch}$ – тензор входных данных, где n – размер изображения в пикселях, ch – количество входных каналов, $y = pred(W, X)$ – ответ ИНС на входных данных **X**. Система КПМО для ИНС – это совокупность четырёх алгоритмов:

- алгоритма генерации ключей $KeyGen(1^{\lambda}) \rightarrow pp$, который по заданному параметру безопасности λ генерирует набор общедоступных параметров pp;
- алгоритма генерации коммитмента $Commit(W, pp, r) \rightarrow com_W$, который создаёт коммитмент ИНС с множеством параметров **W**, используя случайность r;
- алгоритма генерации доказательства $Prove(W, X, pp, r) \rightarrow (y, \pi)$, который для заданного тензора **X** вырабатывает при помощи ИНС ответ y и генерирует доказательство целостности π;
- алгоритма проверки доказательства $Verify(com_W, X, y, \pi, pp) \rightarrow \{0,1\}$, который проверяет ответ y, используя коммитмент com_W, доказательство π и входные данные **X**.

Приведенное здесь определение достаточно легко может быть обобщено на КПМО с другими моделями машинного обучения. Дальнейшая задача состоит в конструировании криптосхем, реализующих КПМО. Центральным компонентом таких криптосхем, как следует из определения, являются криптографические системы доказательства с нулевым разглашением (ДНР)

[1] – только они позволяют одновременно обеспечить свойства конфиденциальности и проверяемости вычислений, в связи с чем рассмотрим их подробнее.

СИСТЕМЫ НЕИНТЕРАКТИВНОГО ДОКАЗАТЕЛЬСТВА, ИСПОЛЬЗУЕМЫЕ В КОНФИДЕНЦИАЛЬНОМ МАШИННОМ ОБУЧЕНИИ

Среди всего многообразия ДНР для решения задач КПМО в первую очередь представляют интерес неинтерактивные доказательства, позволяющие доказывать утверждения произвольного вида. Неинтерактивный характер доказательства подразумевает, что они функционируют по аналогии с привычной всем цифровой подписью, т.е. один из участников информационного обмена может сгенерировать доказательство, не привлекая для выполнения этой процедуры никого из других участников, а любое другое заинтересованное лицо, получив такое доказательство, может самостоятельно его проверить и получить однозначное заключение об истинности (или ложности) доказываемого утверждения. Доказательства универсального типа позволяют доказывать утверждения, в которых входные и выходные данные связаны функциями произвольного вида, которые в самом общем случае могут быть выражены арифметическими или, в частном случае, булевыми схемами.

Пусть имеется функция C над некоторым конечным полем F, причем область определения функции C содержит $n+h$ входных переменных из поля F, а область значений функции C содержит l выходных переменных в поле F : $C: F^n \times F^h \to F^l$. Функция C является *арифметической схемой* тогда и только тогда, когда все выходные переменные функции C могут быть определены через входные переменные путем переходов по направленному ациклическому графу, узлы которого ассоциированы с арифметическими операциями сложения и умножения. *Корректным назначением* (valid assignment) для арифметической схемы C будем называть некоторый кортеж $(a_1, \dots, a_N) \in F^N$, где $N = (n+h)+l$ такой, что $C(a_1, \dots, a_{n+h}) = (a_{n+h+1}, \dots, a_N)$.

В наибольшей степени перечисленным условиям соответствует класс криптографических систем доказательства, называемых компактными неинтерактивными доказательствами знания с нулевым разглашением (zk-SNARK – zero-knowledge succinct non-interactive arguments of knowledge). В настоящее время при проектировании систем безопасных многосторонних вычислений, в частности, систем КПМО, используется несколько систем zk-SNARK. Рассмотрим в качестве примера систему, описанную в работе [5].

Для этого рассмотрим центральное этой системы доказательства понятие *квадратичной арифметической программы* (QAP – quadratic arithmetic program). Пусть имеется функция $Q(C) := (\vec{A}, \vec{B}, \vec{C}, Z)$, где $Z(z) \in F[z]$ будем называть целевым полиномом, а тройка векторов $\vec{A}, \vec{B}, \vec{C}$ определяется следующим образом:

$$\vec{A} = \left(A_i(z)\right)_{i=0}^{m}, \vec{B} = \left(B_i(z)\right)_{i=0}^{m}, \vec{C} = \left(C_i(z)\right)_{i=0}^{m}, где\ z \in Fum \geq N.$$

Далее определяется полином $P(z) = \left(A_0(z) + \sum_{i=1}^{m} a_i A_i(z)\right)\left(B_0(z) + \sum_{i=1}^{m} a_i B(z)\right) - \left(C_0(z) + \sum_{i=1}^{m} a_i C(z)\right)$ такой, что $Z(z)$ делит полином $P(z)$ тогда и только тогда, когда кортеж $(a_1, \dots, a_N)$ является корректным назначением для C.

Квадратичная арифметическая программа в zk-SNARK используется для того, чтобы доказывающий **P** сконструировал доказательство π, подтверждающее знание $(a_1, \dots, a_N) \in F^N$ для арифметической схемы C. Имея доказательство π, проверяющий **V** достаточно легко может проверить делимость $P(z)$ на $Z(z)$.

Рассмотрим теперь метод построения QAP для арифметической схемы **C**. Пусть арифметической схеме **C** соответствует такой направленный ациклический граф, концевые вершины которого определены только через операции умножения. Определим **m** как суммарное количество рёбер, входящих в некоторую вершину графа и исходящих из него.

Пусть **M** – подмножество вершин графа, **W** – множество входящих ребер для подмножества вершин **M**. Для узла $g \in M$ введем обозначение

$I_{g,L} \subset W$ для подмножества ребер, входящих слева для узла **g**. Соответственно, $I_{g,R} \subset W$ – подмножество ребер, входящих справа для узла **g**. Далее определим целевой полином $Z(z)$ для вершин схемы **C**: $Z(z) = \prod_{g \subset M}(z - r_g)$.

Наконец, определим полиномы для векторов $\vec{A} u \vec{B}$ следующим образом:

$$A_i(r_g) = c_{g,L,i}, если i \in I_{g,L}, иначе A_i(r_g) = 0$$
$$B_i(r_g) = c_{g,R,i}, если i \in I_{g,R}, иначе B_i(r_g) = 0,$$

где $c_{g,L,i}$ – скалярный множитель, с которым ребро i входит в узел **g** слева. Также введем обозначения $A_0(r_g)$ и $B_0(r_g)$ для констант, входящих в узел g слева и справа соответственно.

Определим полином для вектора $\vec{C}$ следующим образом:

$$C_i(r_g) = 1, если i = g, иначе C_i(r_g) = 0.$$

Таким образом, получаем следующие правила полиномиального представления арифметической схемы:

1) входящее значение для узла g слева:

$$A(r_g) = A_0(r_g) + \sum_{i=1}^{m} a_i A_i(r_g) =$$
$$A_0(r_g) + \sum_{i \in I_{g,L}} a_i c_{g,L,i};$$

2) входящее значение для узла **g** справа:

$$B(r_g) = B_0(r_g) + \sum_{i=1}^{m} a_i B_i(r_g) =$$
$$B_0(r_g) + \sum_{i \in I_{g,R}} a_i c_{g,R,i};$$

3) исходящее значение для узла **g**:
$C(r_g) = C_0(r_g) + \sum_{i=1}^{m} a_i C_i(r_g) = a_g$.

Для всех $g \in M$ это означает, что полином
$P(r_g) = A(r_g) \cdot B(r_g) - C(r_g) = 0$.

Таким образом, кортеж $(a_1, \dots, a_N)$ является корректным назначением для арифметической схемы C, тогда и только тогда, когда $P(z)$ имеет нули для всех r_g что эквивалентно делимости $P(z)$ на $Z(z)$.

Введем дополнительные обозначения. Пусть R_c – пара векторов $(\vec{x}, \vec{w})$ с нулевым выходом, которая образует корректное назначение для арифметической схемы C:

$$R_c = \{(\vec{x}, \vec{w}) | \in F^n \times F^h | C(\vec{x}, \vec{w}) = 0\}$$

Пусть L_C – NP-полный язык для векторов $\vec{x}$, которые могут образовать корректные назначения с нулевым выходом с некоторыми векторами $\vec{w}$:

$$L_c = \{\vec{x} \in F^n \vee \exists \vec{w} \in F^h : C(\vec{x}, \vec{w}) = 0\}$$

В этом случае будем называть $\vec{w}$ свидетельством (witness), которое будет являться секретом доказывающего P.

Практическая реализация такой системы доказательства zk-SNARK основывается на использовании билинейных отображений вида $e: G_1 \times G_2 \to G_T$, где G_1 и G_2 – циклические группы с образующими элементами $P_1 \in G_1$ и $P_2 \in G_2$ соответственно, G_T – группа порядка r.. Отображение e – невырожденное несимметричное (т.е. $G_1 \neq G_2$ билинейное отображение, удовлетворяющее условиям:

$e(n_1 P_1, n_2 P_2) = e(P_1, P_2)^{n_1 n_2}$ и $e(P_1, P_2) \neq 1$.

Рассмотрим свойства zk-SNARK: полноту (completeness), состоятельность (soundness) и нулевое разглашение (zero-knowledge).

Полнота означает, что проверяющий V всегда примет доказательство корректного утверждения. Пусть полиномы QAP были сгенерированы на предыдущем этапе. Для обеспечения полноты доказательства при его генерции выполняются следующие шаги.

Шаг 1. Генерация ключей.

1. Определить арифметическую схему $C: F^n \times F^h \to F^l$.
2. Определить QAP: $Q(C) = (\vec{A}, \vec{B}, \vec{C}, Z)$.
3. Сгенерировать ключи доказательства. Для этого выбрать случайные $\tau, \rho_A, \rho_B \in F$. Пусть $\rho_c := \rho_A \rho_B$, где $pk_A := (A_i(\tau)\rho_A P_1)_{i=1}^m, pk_B := (B_i(\tau)\rho_B P_2)_{i=1}^m, pk_C := (C_i(\tau)\rho_C P_1)_{i=1}^m$, $pk_H := (\tau P_1)_{i=1}^d$.
4. Сгенерировать ключ проверки $vk_{IC} := (A_i(\tau)\rho_A P_1)_{i=1}^n vk_Z := (Z(\tau)\rho_C P_2)$.

В результате выполнения шага 1 создаются ключи доказательства и ключи проверки.

Шаг 2. Создание доказательства.

1. Определить QAP: $Q(C) = (\vec{A}, \vec{B}, \vec{C}, Z)$.
2. Вычислить валидное распределение $(a_1, \dots, a_m = QAPwit(C, \vec{x}, \vec{w})$.
3. Определить коэффициенты $h_i{}_{i=0}^d$ многочлена $H(z) = \frac{A(z)B(z) - C(z)}{Z(z)}$.
4. Вычислить доказательство $\pi := (\pi_A, \pi_B, \pi_C, \pi_H)$, где

$\pi_A := \sum_{i=1}^m a_i pk_{A,i}, \pi_B := pk_{B,0} + \sum_{i=1}^m a_i pk_{B,i}, \pi_C := pk_{C,0} + \sum_{i=1}^m a_i pk_{C,i}$, $\pi_H := pk_{H,0} + \sum_{i=1}^d h_i pk_{H,i}$.

Доказательство π_H кодирует полином

$H(z)$ элементами группы G, т.е. $\pi_H = H(\tau) \cdot \rho_B P_1$. Соответственно, π_B кодирует полином $B(z)$, т.е. $\pi_B = (B_0(\tau) + \sum_{i=1}^{m} a_i B_i(\tau))H(\tau) \cdot \rho_B P_2$.

5. На следующем шаге проверяющий V выполняет деление $P(z)$ на $Z(z)$. Для этого вычислять $vk_{\vec{x}} = vk_{IC,0} + \sum_{i=1}^{n} x_i vk_{IC,i}$.

6. Далее проверить делимость $P(z)$ на $Z(z)$, т.е. тот факт, что $e(vk_{\vec{x}} + \pi_A, \pi_B) = e(\pi_H, vk_z) \cdot e(\pi_C, P_2)$.

Состоятельность означает, что доказывающий сможет убедить проверяющего принять доказательство ложного утверждения лишь с пренебрежимо малой вероятностью.

Нарушитель может использовать $P(z) = Z(z)$ в качестве своего доказательства с $A(z) = 1, B(z) = Z(z) u C(z) = 0$. Такие значения могут быть приняты проверяющим V даже в случае, если доказывающий P не знает корректное назначение $(\vec{x}, \vec{w}) \in R_C$. Таким образом, проверяющий V также должен выполнить две проверки:

- проверку того, что полиномы $A(z), B(z) u C(z)$ являются линейными комбинациями $\vec{A}$, $\vec{B} u \vec{C}$ соответственно;
- проверку того, что линейная комбинация $\vec{A}$, $\vec{B} u \vec{C}$ использует одинаковые коэффициенты a_i.

Такая проверка достигается посредством построения доказывающим второго аналогичного доказательства с использованием отличающихся ключей путем перемножения на некоторый случайный коэффициент α_A: $pk'_{A,i} = \alpha_A pk_{A,i}$. Доказывающий не знает α_A, однако может вычислить $\pi'_A = \alpha_A \pi_A$, для чего должны быть использованы $pk'_{A,i}$ и $pk_{A,i}$.

Таким образом, необходимо выполнить следующие шаги:

Шаг 1. Генерация ключей.

1.1. Генерация ключей для проверки наличия линейных комбинаций:

1) выбрать случайные элементы поля $\alpha_A, \alpha_B, \alpha_C \in F$;

2) вычислить ключи доказывающего P:
$pk'_{A,i} := (\alpha_A A_i(\tau) \rho_A P_1)_{i=1}^{m}$,
$pk'_{B,i} := (\alpha_B B_i(\tau) \rho_B P_1)_{i=1}^{m}$,
$pk'_{C,i} := (\alpha_A C_i(\tau) \rho_C P_1)_{i=1}^{m}$;

3) вычислить ключи проверяющего V:
$vk_A := \alpha_A P_2, vk_B := \alpha_B P_2, vk_C := \alpha_C P_2$.

1.2. Генерация ключей для проверки использования одинаковых коэффициентов:

1) выбрать случайные элементы поля $\beta, \gamma \in F$;

2) вычислить ключи доказывающего P: $pk_K =$;

3) вычислить ключи проверяющего V: $vk_\gamma := \gamma P_2, vk^1_{\beta\gamma} := \gamma\beta P_1, vk^2_{\beta\gamma} := \gamma\beta P_2$.

Шаг 2. Проверка наличия линейных комбинаций.

Доказывающий P добавляет к доказательству π: $\pi'_A := \sum_{i=n+1}^{m} a_i pk'_{A,i}$, $\pi'_B := pk'_{B,0} + \sum_{i=n+1}^{m} a_i pk'_{B,i}$, $\pi'_C := pk'_{C,0} + \sum_{i=n+1}^{m} a_i pk'_{C,i}$.

Проверяющий может использовать это доказательство для того, чтобы проверить равенства:

$e(\pi_A, vk_A) = e(\pi'_A, P_2)$, $e(\pi_B, vk_B) = e(\pi'_B, P_2)$, $e(\pi_C, vk_C) = e(\pi'_C, P_2)$.

Учитывая, что равенства выполняются если и только если $\pi'_A = \alpha_A \pi_A$, можно утверждать, что доказывающий P использовал для вычислений ключи $pk_{A,i}$ и $pk'_{A,i}$, принадлежащие A.

Шаг 3. Проверка наличия коэффициентов.

Доказывающий добавляет к доказательству π выражение $\pi_K := pk_{K,0} + \sum_{i=1}^{m} a_i pk_{K,i}$. Далее проверяющий V выполняет проверку равенства: $e(\pi_K, vk_\gamma) = e(vk_{\vec{x}} + \pi_A + \pi_C, vk^2_{\beta\gamma}) \cdot e(vk^1_{\beta\gamma}, \pi_B)$. Следовательно, при положительном результате проверки доказывающий P должен был использовать одинаковые коэффициенты a_i для вычисления $\pi_A, \pi_B u \pi_C$, чтобы эти равенства выполнялись.

Нулевое разглашение означает, что доказывающий может использовать $\sigma_1, \sigma_2 u \sigma_3 \in F$ и изменить полиномы в QAP на следующие:
$A(z) = A_0(z) + \sum_{i=1}^{m} a_i A_i(z) + \sigma_1 Z(z)$,
$B(z) = B_0(z) + \sum_{i=1}^{m} a_i B(z) + \sigma_2 Z(z)$,
$C(z) = C_0(z) + \sum_{i=1}^{m} a_i C(z) + \sigma_3 Z(z)$.

В этом случае информация о корректном назначении $(a_1, \dots, a_m)$ остается скрытой, при этом сохраняется свойство делимости полинома $P(z)$ на $Z(z)$.

Имеется целый ряд эффективных реализаций систем zk-SNARK, которые отличаются между собой главным образом методикой использования билинейных отображений при генерации доказательства и, соответственно, алгоритмами проверки таких доказательств. Самыми

известными и эффективными из них являются системы Грота [6] и Грота – Маллер [7], которые выбираются в большинстве случаев создателями систем КПМО.

РЕАЛИЗАЦИЯ СИСТЕМ КОНФИДЕНЦИАЛЬНОГО И ПРОВЕРЯЕМОГО МАШИННОГО ОБУЧЕНИЯ

Будем рассматривать реализации систем КПМО согласно хронологии их появления. Все решения относятся к периоду 2019 – 2021 гг. Большинство известных реализаций систем КПМО направлено на обеспечение свойств конфиденциальности и проверяемой целостности для ИНС. Это легко объяснимо, поскольку ИНС – самый популярный и распространенный инструмент машинного обучения. Однако будут встречаться и исключения из этого правила.

Система VeriML [8] является, пожалуй, самой ранней попыткой обеспечить одновременно конфиденциальность и проверяемость ИНС при помощи SNARK (но без свойства нулевого разглашения). Идея схемы состоит в том, что доказывающий создает коммитменты всех слоёв ИНС, а проверяющий случайным образом выбирает какой-то один слой для проверки и проверяет корректность его вычисления при помощи криптографического доказательства. Очевидно, что такая схема не способна обеспечивать свойство состоятельности доказательства. Она слишком сложна и малопроизводительна для того, чтобы с её помощью можно было проверять вычисления ответа на запрос к ИНС целиком. В связи с этим мы оставим эту систему за рамками дальнейшего рассмотрения.

Система vCNN [9] явилась одной из первых попыток создать систему КПМО. Ключевая идея авторов системы состоит в построении оптимизированных арифметических схем для выполнения операций свёртки, которые занимают до 90% времени выполнения операций при применении обученной свёрточной ИНС. Предложенный способ пригоден для свёрточных ИНС с нелинейными функциями активации ReLU и слоями пулинга. В качестве системы ДНР в ней используется конструкция Грота [6]. Авторы протестировали свою разработку на ИНС VGG16 и получили условно приемлемое время генерации доказательства – 8 часов при объёме общей ссылочной строки доказывающего и проверяющего (CRS – common reference string) около 80 ГБ. Главное значение этой работы состоит в том, что она была первой, в которой был указан путь снижения сложности процедур генерации и проверки ДНР для таких сложных схем как ИНС с «астрономической», находящейся далеко за гранью возможностей современной вычислительной техники, до условно практически приемлемой. На текущий момент система vCNN уступает более новым решениям по производительности и стойкости. Вслед за ней довольно быстро появились другие системы, которые превосходят её по скорости генерации доказательств и их объёму на несколько порядков величины. Далее рассмотрим эти решения более подробно.

Единственной в своём роде системой КПМО для важнейшего из неградиентных методов машинного обучения – метода решающих деревьев является система, описанная в работе [10]. Она предназначена для использования на этапе применения обученного решающего дерева, обеспечивая конфиденциальность как данных, подаваемых на вход модели, так и предикатов, ассоциированных с узлами дерева. Идея построенная конфиденциальной и проверяемой модели на основе решающего дерева заключается в следующем.

На фазе инициализации системы создается коммитмент решающего дерева. Эта процедура выполняется за время, линейно зависящее от размера решающего дерева. Для этого используется специальная структура данных – аутентифицированное решающее дерево, которое однозначно связано с исходным решающим деревом, но из которого невозможно восстановить его параметры за счёт хэширования данных, ассоциированных с листьями, промежуточными вершинами и «подмешивания» случайной величины к хэш-коду корня.

На этапе применения решающего дерева для получения ответа на запрос процедура будет иной: обладатель модели выступает в роли доказывающего, инициатор запроса (он же заказчик ответа) – в роли проверяющего. Доказательство, прилагаемое вычислителем к ответу, является неинтерактивным. Для его построения

используются как те данные, которые доступны обеим сторонам, так и те данные, которые известны только доказывающему и которые он не намерен раскрывать проверяющему. К первым относятся запрос, ответ и коммитмент дерева. Ко вторым – путь, пройденный по дереву от корня к одному из листьев при вычислении ответа, а также случайность, подмешанная к корню дерева при создании коммитмента. Для повышения производительности схемы при генерации доказательства может также добавляться вектор-перестановка вектора входных признаков и вектор хэш-кодов вершин дерева, соседних с вершинами, составляющими путь, пройденный при получении ответа (siblings). Окончательно доказательство получается как совокупность выражений, подтверждающих, что при получении ответа был полностью использован вектор входных признаков (точнее, некоторая перестановка компонентов этого вектора), путь по дереву пройден полностью от корня до одного из листьев и при вычислении ответа использовано в точности то же самое дерево, для которого на фазе инициализации был получен коммитмент.

Авторы работы также предлагают специальную схему доказательства, при помощи которой заказчик может убедиться, что предлагаемая ему для предсказания ответа модель соответствует заявленной (или ожидаемой) точности (accuracy) на тестовой выборке.

В качестве неинтерактивной системы ДНР для описанной конструкции выбрана система Aurora [11]. Обладающая постквантовой стойкостью. Схема может быть распространена на часто используемые разновидности моделей машинного обучения, производные от решающих деревьев: дерево для решения задачи регрессии и случайный лес.

В работе [12] предложена и реализована система КПМО для свёрточных ИНС. Основная идея системы ZEN заключается в построении цепочки алгоритмов (toolchain), которая позволяет построить верифицируемую ИНС, обрабатывающую числа с плавающей запятой за практически приемлемое время. Основные звенья этой цепочки – это:

- способ квантования чисел с плавающей запятой – превращения их в целые числа без знака, удобный для обработки их в ИНС с функциями конфиденциальности и проверяемости вычислений;
- способ кодирования векторов целых чисел без знака, обеспечивающий удобство их параллельной обработки при вычислении скалярных произведений и матричном умножении;

При этом авторами одновременно преодолён целый ряд препятствий, ранее не позволявших получить практически приемлемые показатели функционирования таких систем.

Первая из решённых в этой работе проблем заключается в том, что существующие ИНС обрабатывают числа с плавающей запятой, в то время как все существующие системы ДНР предполагают арифметические вычисления над конечными полями. Таким образом, ИНС оказываются несовместимы с ДНР. Существующие алгоритмы квантования чисел с плавающей запятой не подходят, так как требуют операций деления и оставляют часть данных числами с плавающей запятой. В связи с этим авторы предложили новый способ квантования, который позволяет построить ИНС, обрабатывающую только целые числа без знака, которые уже можно интерпретировать как элементы конечного поля. Предложенный способ квантования позволяет вычислять нелинейные функции активации слоёв ИНС типа ReLU и “average pool” (пулинг по среднему значению).

Вторая проблема состоит в том, что большинство известных квантованных ИНС способны обрабатывать лишь 8-битные целые числа, но большинство ДНР используют эллиптические кривые над полем порядка $\approx 2^{254}$, что приводит к неэффективной реализации. В связи с этим авторы предложили новый метод кодирования данных для обработки в ИНС – так называемое скрученное кодирование (stranded encoding), которое позволяет упаковывать множество переменных в один вектор и выполнять множество операций умножения элементов конечного поля в одном пакете, тем самым многократно ускоряя матричные операции.

В качестве системы ДНР в ZEN используется широко известная система Грота [6], основанная на билинейных отображениях над группами точек эллиптических кривых.

Авторы реализовали свою разработку в виде пакета программ с открытым исходным кодом. Для тестирования ими были взяты нейронные сети ShallowNet, LeNet-5 и LeNet-Face (последняя оптимизирована для решения задач распознавания лиц). В качестве тестовых массивов данных брались широко известные датасеты MNIST, CIFAR-10 и ORL. Созданные программы позволяют построить системы конфиденциальной и проверяемой классификации и распознавания. Под классификацией здесь понимается классическая задача позиционирования объекта, детектированного на изображении, как относящегося к одному из нескольких предопределённых классов. Под распознаванием – сравнение двух изображений и вынесение решения о том, относятся или нет они к одному и тому же классу (в частности, распознавание двух изображений лица как принадлежащих одному человеку). Система ZEN предназначена для использования только на этапе применения обученной ИНС (inference), но не на этапе обучения.

Обе схемы, как показано авторами, обладают свойствами полноты, состоятельности и нулевого разглашения, хотя определения этих свойств для двух систем несколько различаются. Показано, что предложенные методы не влияют существенным образом на точность (accuracy) решения задач по сравнению с обычными ИНС, не обладающими свойствами конфиденциальности и проверяемой целостности.

В качестве ограничений системы ZEN следует указать, прежде всего, ограниченную масштабируемость (испытания проводились на далеко не самых сложных из современных свёрточных ИНС и не самых обширных массивах данных), отсутствие исследований возможности её применения к иным типам ИНС, помимо свёрточных, отсутствие возможности сокрытия от постороннего наблюдателя архитектуры ИНС.

В работе [4] предложена ещё одна система КПМО для свёрточной ИНС, названная авторами zkCNN. Ключевая особенность этой системы – использование быстрого преобразования Фурье (БПФ) для вычисления свёрток, в связи с чем для эффективного вычисления и проверки криптографических доказательств авторы предлагают новый алгоритм вычисления контрольных сумм (sumcheck) для БПФ. Алгоритм позволяет добиться логарифмического времени генерации и проверки доказательств.

В zkCNN используется обобщение неинтерактивной системы доказательства GKR (Goldwasser – Kalai – Rothblum) [13], оптимизированное для свёрточных ИНС. zkCNN поддерживает вычисление нелинейных функций активации ReLU и пулинг по максимальному значению.

Тестирование системы zkCNN, проведённое на ИНС VGG16 с 15 миллионами параметров на датасете CIFAR-10 показало среднее время генерации доказательств 163 с и среднее время их проверки 172 мс при объёме доказательства 230 КБ, что, по утверждению авторов, на три порядка быстрее лучшей из ранее известных схем.

В работе [14] представлена система Mystique, в которой на единой платформе предложено решение сразу несколько задач, остающихся актуальными для КПМО. Разработка позволяет переключаться между арифметическими и булевыми схемами (что удобно и необходимо при вычислении линейных преобразований ИНС), числами с плавающей и с фиксированной запятой (что актуально при вычислении нелинейных преобразований), коммитментами и конфиденциальными данными (что позволяет настраивать степень конфиденциальности системы под требования заказчика), интегрируя их все в вычислительную схему, для которой генерируется доказательство. Таким образом, эта система позволяет, в том числе, скрывать от посторонних лиц архитектуру модели машинного обучения. Авторами также представлен оптимизированный протокол ДНР для матричного умножения, который даёт 7-кратный рост производительности по сравнению с лучшими из известных алгоритмов. Все перечисленные свойства достигаются за счёт несущественного (около 0,2%) снижения точности моделей по сравнению с исходными. Система Mystique интегрирована в фреймворк Rosetta, предназначенный для реализации КМО и основанный на библиотеке TensorFlow.

ЗАКЛЮЧЕНИЕ

В статье проведен обзор всех известных (по состоянию на июнь 2021 г.) систем КПМО. Выявлено, что основным криптографическим инструментом для конструирования таких систем служат компактные неинтерактивные доказательства знания с нулевым разглашением (zk-SNARK).

Нет сомнений в том, что системы КМО и КПМО будут активно востребованы в будущем. Рассмотренные в статье системы, скорее всего, в недалеком будущем станут считаться лишь первыми шагами по пути их практической реализации. Все описанные системы пока являются экспериментальными образцами, для их широкого практического применения необходимо решить целый ряд проблем.

Среди основных нерешённых проблем можно отметить следующие:

- относительно низкая производительность систем КПМО, ограничивающая сферу их применения лишь высокопроизводительными вычислительными устройствами;
- весьма ограниченная применимость всех известных методов на этапе обучения моделей;
- невозможность существующими средствами обеспечить конфиденциальность архитектуры ИНС – все известные системы КПМО скрывают лишь параметры ИНС, считая все элементы её архитектуры (количество слоёв, каналов, нейронов, конфигурацию связей между слоями и пр.) общеизвестными.

Представляется, что дальнейшее развитие КМО будет в значительной мере связано с решением перечисленных научно-практических задач, что позволит широко применить их в области цифровых финансовых активов, когда необходимо анализировать данные криптобирж, прогнозировать волатильность или тренды развития цифровых активов (криптовалют).

СПИСОК ЛИТЕРАТУРЫ

1. Запечников С.В. Криптографическая защита процессов обработки информации в недоверенной среде: достижения, проблемы, перспективы // Вестник современных цифровых технологий. 2019. №1. С. 6 – 18.

2. Запечников С.В. Модели и алгоритмы конфиденциального машинного обучения // Безопасность информационных технологий. 2020. Т. 27, Вып. 1. С. 51–67.

3. Запечников С.В. Доказательства с нулевым разглашением и их применения при обработке информации в недоверенных средах // Вестник современных цифровых технологий. 2021. №6. С. 11 – 22.

4. Liu T., Xie X., Zhang Y. zkCNN Zero knowledge proofs for convolutional neural network predictions and accuracy. URL: https://eprint.iacr.org/2021/673 (дата обращения: 07.06.2021).

5. Parno B., Howell J., Gentry C., Raykova M. Pinocchio: Nearly Practical Verifiable Computation // IEEE Symposium on Security and Privacy. 2013. Pp. 238-252. doi: 10.1109/SP.2013.47.

6. Groth J. On the Size of Pairing-Based Non-interactive Arguments // Advances in Cryptology – EUROCRYPT 2016. Lecture Notes in Computer Science. Vol. 9666. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-49896-5_11

7. Groth J., Maller M. Snarky Signatures: Minimal Signatures of Knowledge from Simulation-Extractable SNARKs // Advances in Cryptology – CRYPTO 2017. Lecture Notes in Computer Science. Vol. 10402. Springer, Cham. https://doi.org/10.1007/978-3-319-63715-0_20

8. Zhao L., Wang Q., Wang C., et al. VeriML: Enabling Integrity Assurances and Fair Payments for Machine Learning as a Service. URL: https://arxiv.org/pdf/1909.06961v1.pdf

9. Lee S., Ko H., Kim J., Oh H. vCNN: Verifiable convolutional neural network based on zk-SNARKs. URL: https://eprint.iacr.org/2020/584 (дата обращения: 07.06.2021).

10. Zhang J., Fang Z., Zhang Y., Song D. Zero knowledge proofs for decision tree predictions and accuracy // CCS '20: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security,

2020. P. 2039–2053. Doi: 10.1145/3372297.3417278.

11. Ben-Sasson E., Chiesa A., Riabzev M. et al. Aurora: Transparent Succinct Arguments for R1CS // Advances in Cryptology – EUROCRYPT 2019. Lecture Notes in Computer Science. Springer. Vol. 11476. P. 103-121. Doi: 10.1007/978-3-030-17653-2_4.

12. Feng B., Qin L., Zhang Z. et al. ZEN: An optimizing compiler for verifiable, zero-knowledge neural network inferences. URL: https://eprint.iacr.org/2021/087 (дата обращения: 07.06.2021).

13. Goldwasser S., Kalai Y., Rothblum G. Delegating Computation: Interactive Proofs for Muggles. Journal of ACM. Vol. 62, No. 4. Article 27 (Sept. 2015). Pp. 1-64.

14. Weng C., Yang K., Xie X. et al. Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning. URL: https://eprint.iacr.org/2021/730 (дата обращения: 07.06.2021).