УДК 681.324

Д. Карчмарски, аспирант; М. Плонковски, аспирант; Е. В. Лисица, студентка

МЕТОДЫ И АЛГОРИТМЫ МОДЕЛИРОВАНИЯ СИСТЕМ КРИПТОПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

This article is devoted to a new key-exchange protocol, which is based on mutually learning neural networks. The convergence process of points and improvement ways of this system are considered. Comparison of efficiency of geometric attack has been lead at use of architecture TPM and TPCM. Also the reliability of this protocol is analyzed and the methods of advance the level of its security is considered. The software has been developed for experimental studying neural protocol, allowing estimating its speed, reliability and safety at various parameters (learning rules, amounts of perceptrons K, and amounts of inputs N, synaptic depths of weight).

Введение. В основе нейросетевой криптографии лежит принцип использования искусственных нейронных сетей и их способности синхронизироваться в процессе «обучения». Стороны A и B используют нейронные сети, характеризующиеся одинаковой архитектурой и набором параметров, но обладающие различным для каждой из сторон, случайно заданным, начальным вектором весовых коэффициентов. На каждом шаге «обучения» стороны A и B обмениваются открытой информацией, касающейся состояния их нейронных сетей (значениями входов и выходов), и при необходимости изменяют скрытые значения весовых коэффициентов, согласно выбранному правилу «обучения». Цель нейронной криптографии заключается в возможности синхронизации сторон A и B друг с другом раньше атакующей стороны C. Полученные в результате синхронизации весовые коэффициенты могут быть использованы в качестве ключа в симметричных системах шифрования.

1. Процесс обучения нейронных сетей. Каждая нейронная сеть, применяемая в нейронной криптографии, характеризуется тремя величинами (K, N, L) и имеет архитектуру TPM (tree parity machine), состоящую из двух слоев [1]. Входной слой содержит K персептронов, каждый из которых характеризуется N-элементным вектором входных значений ($[x_{K,1}, x_{K,2}, ..., x_{K,N}]$) и N-элементным вектором весовых коэффициентов ($[w_{K,1}, w_{K,2}, ..., w_{K,N}]$, где $w_{K,N} \in [-L; L]$). Выходы нейронов ($[y_1, y_2, ..., y_K]$) составляют K значений, которые равны скалярному произведению вектора входных значений на вектор весов. Пороговая биполярная функция активации возвращает величину –1 при отрицательных значениях выходных величин и 1 при нулевых и положительных. Выходной слой содержит элемент o, являющийся выходом целой архитектуры TPM. Значение выхода равно произведению величин, полученных во входном слое, т. е. $o = y_1 y_2 ... y_K$.

В начальный момент времени обе стороны A и B инициализируют векторы весов. Затем на каждом шаге обучения t генерируются вектора входных значений и вычисляются значения выхода целой архитектуры TPM. Активизация весов (синхронизация сетей) происходит тогда, когда выходные значения обеих сетей одинаковы ($o^A = o^B$), и активизируются веса только тех нейронов, значение на выходе которых равно значению на выходе целой архитектуры TPM:

а) если $oy_K > 0$, то

$$w_{ij} = w_{ij} - ox_{ij};$$

б) если $|\ w_{ij}\ | > L$, то

$$w_{ij} = sign(w_{ij})L, \qquad (1)$$

где $i = 1, 2, \ldots, K$.

2. Анализ процесса сходимости точек. Пусть w_i является точкой в N-мерном пространстве. Когда точка в процессе «обучения» совершает свободное движение, изменяются все ее координаты. Однако если точка находится на границе N-мерного пространства (т. е. хотя бы одна координата точки w_i равна $\pm L$), то она не имеет возможности свободно перемещаться и, следовательно, изменять все свои координаты. Это означает, что точки с различными координатами в процессе «обучения» могут сойтись, и значения их координат совпадут.

Пример 1. Пусть заданы следующие параметры: $K = 3$, $N = 2$, $L = 3$. В начальный момент времени все пространство заполняется точками. Пусть вход одной вектор на каждом из 5 шагов «обучения» последовательно принимает следующие значения: (1, –1), (1, 1), (1, 1), (–1, –1), (1, –1). Тогда движение точек будет происходить таким образом как показано на рис. 1. Из рис. 1 видно, что после каждого движения образуются «пустые» промежутки, т. е. места, которые можно исключить из множества всех возможных решений. Точки двигаются из середины к границам области.

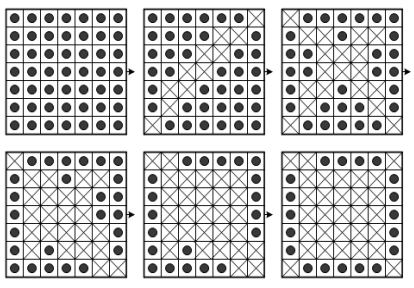

Рис. 1. Сходимость (накладывание) точек и образование «пустых» промежутков в пространстве

После 13 шагов «обучения» пространство может выглядеть следующим образом (рис. 2).

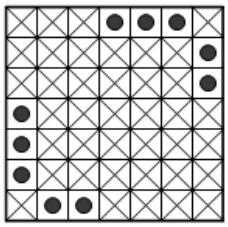

Рис. 2. Сходимость (накладывание) точек и образование «пустых» промежутков в пространстве после 13 шагов

На основании проведенных наблюдений можно сделать вывод, что чем больше шагов совершается в процессе «обучения», тем больше «пустых» промежутков возникает в пространстве. Количество возможных решений r равно произведению количества точек, оставшихся в каждом из рассматриваемых пространств:

$$r = r_1 r_2 \ldots r_K. \quad (2)$$

В пространстве, определенном параметрами (K, N, L), количество возможных решений вычисляется по следующей формуле:

$$(2L+1)^{NK}. \quad (3)$$

Для того чтобы уменьшить количество решений в каждом из K пространств, точкам присваивается следующее значение:

$$\frac{1}{(2L+1)^N}. \quad (4)$$

Значение (4) – это вероятность того, что данная точка находится в определенном месте в пространстве. В начальный момент времени все точки имеют одинаковое значение вероятности. Если точки совершают несвободное движение и накладываются друг на друга, то значения их вероятности суммируются. Благодаря этому возможен более точный выбор потенциальных решений.

Для достижения поставленной цели исследования предположим, что во время синхронизации весов персептронов для обеих сторон A и B создается список, содержащий точки из каждого пространства. Этот список сортируется, согласно величине вероятности этих точек.

Обозначим величиной d_i позицию точек в списке. Тогда значение максимальной глубины $d_{\max}$ равно:

$$d_{\max} = \max(d_1, d_2, \ldots, d_K). \quad (5)$$

В этом случае мы будем в состоянии достичь максимально возможных результатов испытания:

$$d_{\max}{}^K. \quad (6)$$

Пример 2. Пусть заданы следующие параметры нейронной сети: $K = 3$, $N = 3$, $L = 4$. Согласно формуле (3), количество возможных величин весовых коэффициентов составляет 9^{12}. Было проведено 85 испытаний, результаты которых представлены в табл. 1.

Таблица 1

Статистика результатов проведенных испытаний

R	$A(d_{\max})$	R	$A(d_{\max})$
1	34	71–80	1(2)
2	12	81–90	1(2), 1(5)
3	5	101–200	1(1)
4	4	201–300	2(1), 1(3)
6	2	401–500	1(5)
8	4	1 001–2 000	1(2), 1(5)
11–20	3(1), 1(3)	2 001–3 000	1(6)
21–30	1(1), 1(3)	3 001–4 000	1(2)
31–40	2(1)	7 001–8 000	1(5)
41–50	1(2), 1(3)	>60 000	1(5)

Здесь R – количество решений, $A(d_{\max})$ – количество испытаний, проведенных с данным количеством решений (максимальная глубина вычислена по формуле (5)). Для $R < 10$ пропущена величина $d_{\max}$.

На основании полученных данных можно сделать вывод, что, определив величину $d_{\max}$, согласно формуле (5), мы значительно уменьшим количество необходимых решений. Так, например, для испытания, которому соответствовало свыше 60 000 решений, при использовании формулы (6) число решений не превышало значения 5^3.

Изменение величин весовых коэффициентов является детерминированным процессом, что позволяет атакующей стороне подражать этим изменениям. Один из возможных способов улучшения системы – это введение элемента случайности, при котором даже в случае небольшого значения величины N способ определения «пустых» промежутков, а также суммирования соответствующих вероятностей был бы непрактичен.

3. Сравнение эффективности геометрической атаки при использовании архитектур TPM и TPCM. Рассмотрим алгоритм геометрической атаки на процесс синхронизации при использовании архитектур TPM и TPCM и покажем, что архитектура TPCM является более безопасной.

Геометрическая атака при использовании архитектуры TPM. Рассмотрим действия атакующей стороны C в случае простого наблюдения. На каждом шаге синхронизации атакующая сторона пробует синхронизировать свои

скрытые весовые коэффициенты, согласно полученным значениям входов и выходов сетей A и B. В случае, если выходы сетей равны ($o^A = o^B$ и $o^A = o^C$), атакующая сторона обновляет свои веса. Однако если $o^A = o^B$ и $o^A \neq o^C$, то атакующая сторона вынуждена пропустить данный раунд и не обновлять веса, так как направление движения, скорее всего, будет ошибочным.

Предположим, что вектора весовых коэффициентов стороны C и A близки по значениям друг с другом. Это означает, что выход какого-то одного вектора сети C будет отличаться от выхода такого же вектора в сети A. Проблема заключается в поиске этого вектора в сети C. Все выходы векторов сети – это числа, принадлежащие множеству $\{-1, 1\}$ и полученные в результате действия функции активации (знака) $\sigma = sign(w_i x_i)$. Следовательно, вероятность ошибки максимальна для векторов, выходная величина которых близка к 0. Таким образом, для поиска «ошибочного» вектора необходимо найти такой вектор, величина выхода которого имеет наименьшее значение ($y_K = \min$), и заменить эту величину на противоположную по знаку. Эти действия необходимо выполнить в том случае, когда $o^A \neq o^C$. Итак, классический алгоритм геометрической атаки выглядит следующим образом.

1. Если стороны A и B имеют различные значения выхода ($o^A \neq o^B$), то атакующая сторона не обновляет свои весовые коэффициенты и пропускает этот раунд так же, как и стороны A и B.

2. Если стороны A, B и C имеют одинаковые значения выхода ($o^A = o^B = o^C$), то все сети обновляют свои весовые коэффициенты, согласно выбранному правилу «обучения».

3. Если стороны A и B имеют одинаковые значения выхода, но $o^A \neq o^C$, то атакующая сторона выбирает персептрон с наименьшим значением выходной величины и изменяет эту величину на противоположную по знаку. Далее проводится «обучение» сети с учетом произведенной замены.

Рассмотренная атака дает возможность атакующей стороне C синхронизироваться быстрее, чем A и B, на 70% (табл. 2).

Геометрическая атака при использовании TPCM. Рассмотрим эффективность геометрической атаки при применении архитектуры TPCM. Поскольку в данной архитектуре вектор входных значений содержит комплексные числа, то в рассмотренный выше классический алгоритм необходимо внести дополнительные изменения.

В этом алгоритме мы ищем вектор, выходное значение которого находится ближе всего к одной из линий деления плоскости. Причем этот вектор не может быть выбран случайно. В рассмотренном ранее классическом алгоритме было необходимо и достаточно найти вектор с наименьшим значением выходной величины. При использовании архитектуры TPCM важно учитывать, что функция знака возвращает числа, принадлежащие множеству $\{1, i, -1, -i\}$, и, следовательно, область искомых векторов должна ограничиваться только теми векторами, для которых изменение выходной величины гарантированно не нарушит равенство $o^A = o^C$.

Алгоритм геометрической атаки при использовании архитектуры TPCM выглядит следующим образом.

1. Если стороны A и B имеют различные значения выхода ($o^A \neq o^B$), то атакующая сторона не обновляет свои весовые коэффициенты и пропускает этот раунд так же, как и стороны A и B.

2. Если стороны A, B и C имеют одинаковые значения выхода ($o^A = o^B = o^C$), то все сети обновляют свои весовые коэффициенты в соответствии с выбранным правилом «обучения».

3. Если стороны A и B имеют одинаковые значения выхода, но $o^A \neq o^C$, то атакующая сторона выбирает такой вектор w_i, для которого значение $\|\mathrm{Re}(\alpha_K)|-|\mathrm{Im}(\alpha_K)\|$ является наименьшим (или ближайшим к одной из линий деления плоскости). Область поиска вектора ограничивается теми векторами, для которых изменение или перенос выходной величины в соседнюю плоскость не нарушит равенство $o^A = o^C$. Перемещение осуществляется путем умножения выходной величины на i или $-i$ в зависимости от его направления. Далее проводится «обучение» сети с учетом произведенной замены.

В табл. 2 представлен сравнительный анализ количества шагов, необходимых для синхронизации и проведения геометрической атаки при различных архитектурах нейронных сетей.

Таблица 2

Сравнительный анализ количества шагов, необходимых для синхронизации и проведения геометрической атаки при различных архитектурах нейронных сетей

Архитектура	Время синхронизации сети A и B (количество шагов)	Время геометрической атаки на сети A и B (количество шагов)
TPM	222,9	387,6
TPCM	2324,4	1 000 000*

* 1 000 000 – это максимальное число шагов, при котором атакующая сторона C не смогла синхронизироваться со сторонами A и B.

Из полученных данных следует, что геометрическая атака представляет существенную угрозу при использовании обычной архитектуры TPM и является абсолютно неэффективной для TPCM архитектуры построения нейронных сетей.

4. Программная реализация нейросетевого протокола. Для экспериментального изучения нейросетевого протокола было разработано программное средство, позволяющее оценить его быстродействие, надежность и безопасность при различных параметрах (правилах «обучения», количестве персептронов K, количестве входов N, синаптической глубине весовых коэффициентов). На рис. 3 представлено окно программы, которое содержит поля для ввода желаемых значений параметров и выбора правил «обучения».

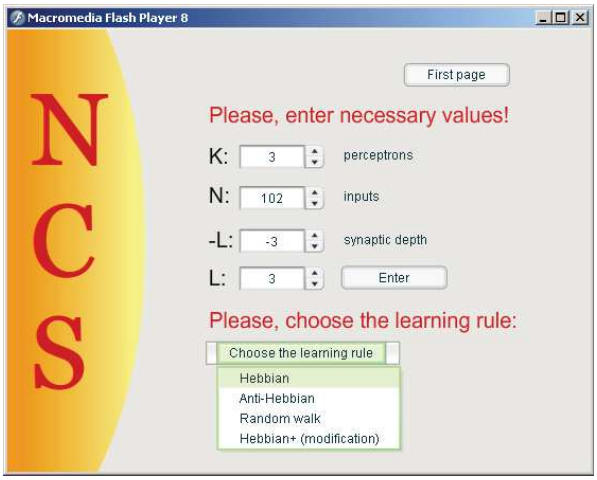

Рис. 3. Интерфейс программы.
Задание входных параметров и правила обучения

Кроме процесса синхронизации двух сетей A и B, в программе также были реализованы два вида атак: простая атака (Simple Attack) и геометрическая атака (Geometric A ttack). Результаты эксперимента показали, что при $K = 3$, $N = 5$, $L = 3$ и использовании правила «Hebbian» геометрическая атака эффективна в 30 случаях из 100. Однако при этом же наборе параметров и использовании правил «Anti-Hebbian» и «Modified H ebbian» эффективность атаки значительно снизилась.

Также результаты эксперимента подтвердили наличие экспоненциальной зависимости вероятности успешной атаки от значения синаптической глубины – с ее увеличением вероятность успешной атаки понижается по экспоненте (рис. 4).

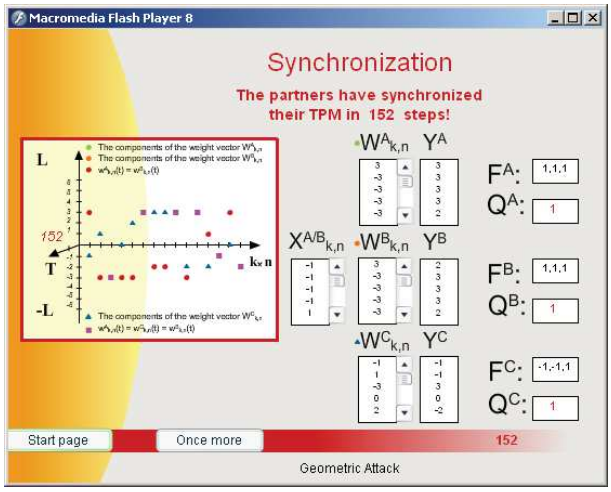

Рис. 4. Интерфейс программы. Результат работы

Заключение. Из анализа полученных данных (табл. 2) можно сделать вывод что при использовании архитектуры TPM геометрическая атака может представлять серьезную угрозу для безопасности нейросетевого протокола. Однако при использовании архитектуры TPCM геометрическая атака бесполезна, что подтверждается очень большим количеством шагов (>1 000 000), необходимых атакующей стороне для синхронизации. Следовательно, нейронные сети TPCM архитектуры характеризуются более высоким уровнем безопасности процесса синхронизации, чем сети, основанные на архитектуре TPM [2].

Литература

1. Klimov, A. Analysis of Neural C ryptography / A. Klimov, A. Shamir, A. Mityaguine // ASIACRYPT. – 2002. – Vol. 2501. – P. 288–298

2. Плонковски, М. Криптографическое преобразование информации на основе нейросетевых технологий / М. Плонковски, П. П. Урбанович // Труды БГТУ. Сер. VI, Физ.-мат. науки и информ. – 2005. – Вып. XIII. – С. 161–164.