УДК 519.17:512

В. В. Смелов, кандидат технических наук, доцент (БГТУ)

ИССЛЕДОВАНИЕ ПРОЦЕССА СХОДИМОСТИ TPM-МАШИН ПРИ ИХ ВЗАИМНОМ ОБУЧЕНИИ

В работе рассмотрен процесс синхронизации двух искусственных нейронных сетей TPM-архитектуры при взаимном обучении. В результате исследования выявлена зависимость продолжительности синхронизации от архитектуры TPM-сети, предложен критерий завершения процесса синхронизации, а также сделана оценка вероятности вычисления секретного ключа, сгенерированного двумя TPM-сетями в процессе взаимного обучения.

We studied the synchronization of two artificial neural networks TPM-architecture in mutual learning. The study revealed the dependence of the duration of the synchronization of the TPM-architecture network, the criterion of completion of the synchronization process, and estimated the probability of computing the secret key generated by the two TPM-networks in the process of mutual learning.

Введение. Одним из новых развивающихся направлений современной криптографии является ***нейрокриптография***. В [1] предлагается, а в [2] развивается идея применения искусственных нейронных сетей архитектуры TPM (tree parity machines – древовидные машины четности) для генерации общего секретного ключа для двух корреспондентов, использующих незащищенный канал связи для обмена данными. Генерация такого ключа основывается на свойстве двух TPM-сетей приходить в синхронное состояние при взаимном обучении. При этом в качестве общего секретного ключа используют весовые коэффициенты входов синхронизировавшихся TPM-сетей. Данная работа посвящена исследованию этого процесса методом имитационного моделирования.

Архитектура TPM-сети. На рис. 1 представлена TPM-сеть, состоящая из m скрытых нейронов, каждый из которых имеет n входов. На вход TPM-сети подается вектор X, состоящий из $m \times n$ элементов $x_{ij} \in \{-1, 1\}$, $i = \overline{1, m}$, $j = \overline{1, n}$. На вход каждого нейрона подается вектор $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $i = \overline{1, m}$, который является частью вектора X. Для каждого входа нейрона применяется коэффициент $w_{ij} \in [-L, L]$, $i = \overline{1, m}$, $j = \overline{1, n}$, где $L > 0$ – параметр TPM-сети. Выходом внутреннего слоя сети является вектор $\Sigma = (\sigma_1, \sigma_2, ..., \sigma_m)$, где σ_i – выход каждого нейрона, вычисляемый по формуле

$$\sigma_i = sign\left(\sum_{j=1}^{m} x_{ij} w_{ij}\right), \quad i = \overline{1, m},$$

где

$$sign(x) = \begin{cases} -1, & x \le 0, \\ 1, & x > 0. \end{cases}$$

Выход TPM-сети O рассчитывается в блоке, обозначенном на рис. 1 символом П, как произведение значений выходов всех нейронов $O = \prod_{i=1}^{m} \sigma_i$. Очевидно, что значение выхода TPM-сети $O \in \{-1, 1\}$.

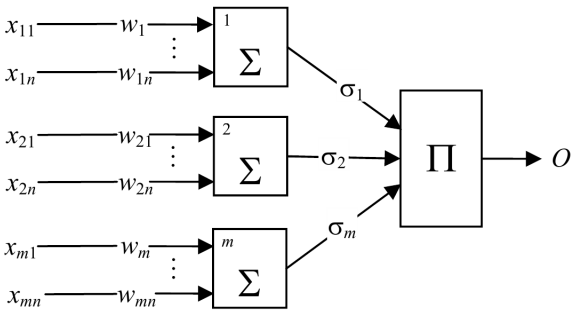

Рис. 1. TPM-сеть, состоящая из m нейронов, каждый из которых имеет n входов

Архитектура TPM-сети описывается тройкой $<m, n, L>$, а текущее ее состояние – двойкой $<X, W>$, где $W = (w_{11}, w_{12}, ..., w_{m \times n})$ – вектор весов.

Процесс взаимного обучения. В общем случае суть процесса обучения искусственной нейронной сети заключается в итеративной настройке ее вектора весов W таким образом, чтобы после завершения процедуры обучения, сеть стала обладать целевыми вычислительными свойствами.

Будем обозначать W_k^A, W_k^B – векторы весов соответственно сетей TPM-A и TPM-B на шаге k обучения. Начальные значения W_0^A и W_0^B устанавливаются случайным образом. Процесс обучения представляет собой последовательность пар:

$$<W_0^A, W_0^B>, \ldots, <W_s^A, W_s^B>, \ldots, \quad (1)$$

где $W_k^A \neq W_k^B$ при $k < s$ и $W_k^A = W_k^B$ при $k \ge s$. Число s называется ***продолжительностью***

обучения. На каждом шаге обучения: $W^A_{k+1} = W^A_k + \Delta W^A_{k+1}$, $W^B_{k+1} = W^B_k + \Delta W^B_{k+1}$, где ΔW^A_{k+1} и ΔW^B_{k+1} вычисляются в зависимости от метода обучения.

Как правило, при обучении используется метод Хебба. При этом компоненты Δw^A_{ij}, Δw^B_{ij}, $i = \overline{1, m}$, $j = \overline{1, n}$, векторов ΔW^A_k и ΔW^B_k рассчитываются по формулам

$$\begin{aligned} \Delta w^A_{ij} &= \sigma_i x_{ij} \theta(\sigma_i, O^A) \theta(O^A, O^B), \\ \Delta w^B_{ij} &= \sigma_i x_{ij} \theta(\sigma_i, O^B) \theta(O^A, O^B), \end{aligned} \qquad (2)$$

где

$$\theta(x, y) = \begin{cases} 0, x \neq y, \\ 1, x = y. \end{cases}$$

Если при этом новое значение $|w_{ij}| > L$, то ему присваивается $sign(w_{ij})L$.

Схема процесса взаимного обучения сетей TPM-A и TPM-B показана на рис. 2. Рисунок состоит из двух за небольшим исключением (в правой части нет стрелки Random-TPM-B) симметричных частей, которые будем условно называть корреспондентами A (на рис. 2 слева) и B (справа). Каждый из корреспондентов состоит из нескольких блоков, которые на рис. 2 обозначены прямоугольниками. Движение информации указывается направленными линиями, причем штриховыми линиями обозначается однократное (на шаге обучения 0) перемещение информации, тонкими линиями – периодическое движение информации внутри корреспондента и утолщенными – периодическая передача информации между корреспондентами.

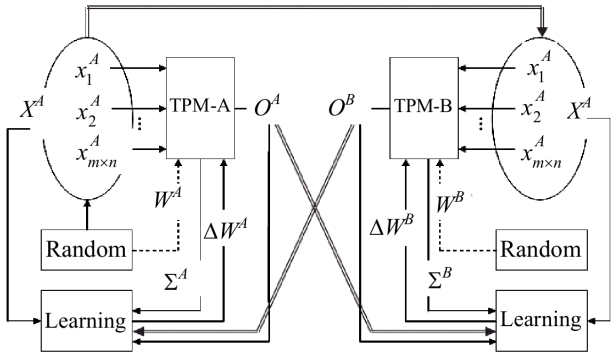

Рис. 2. Схема взаимного обучения двух TPM-сетей

При взаимном обучении двух TMP-сетей одной из них назначается роль ***учителя***, другой – ***ученика***. На рис. 2 роль учителя выполняет сеть TPM-A, а роль ученика – TPM-B. На каждом шаге учитель генерирует обучающую пару $<X^A_k, O^A_k>$ для ученика и, если ученик прошел обучение, осуществляет самообучение.

Кроме собственно TPM-сети в процессе обучения используются еще блоки Random и Learning. Блоки Random применяются для генерации случайных векторов W^A и X^A на стороне учителя и W^B на стороне ученика. Причем векторы W^A и W^B генерируются один раз на шаге $k = 0$, а вектор X^A генерируется для каждого шага на стороне учителя. Будем использовать символ X^A_k для обозначения входного вектора TPM-A-сети, сгенерированного на шаге k.

Блоки Learning на каждом шаге обучения в соответствии с формулами (2) формируют векторы ΔW^A и ΔW^B. Очевидно, что изменение весов w_{ij} сетей осуществляется только в том случае, если $O^A = O^B$ и только для тех нейронов, для которых $\sigma_i = O$.

Имитационная модель TPMCrypt. Для исследования свойств TPM-сетей и процесса их взаимного обучения разработана имитационная модель TPMCrypt, позволяющая создавать модели систем TPM-сетей различной архитектуры. В таблице представлены параметры модели TPMCrypt и их значения.

Параметры модели TPMCrypt

Параметр	Семантика параметра	Значение
loop	Количество шагов обучения	2500
cycle	Количество опытов	200
TPM:L	Ограничение значений $\|w_{ij}\| < L$	4, 6
TPM:m	Количество нейронов внутреннего слоя	2, 3, 4
TPM:n	Количество входов каждого нейрона	2–10, 15, 20

Оценка продолжительности взаимного обучения TPM-сетей. Одной из важнейших характеристик любой системы генерации общего секретного ключа является продолжительность процесса генерации.

На рис. 3 представлены результаты исследования продолжительности процесса синхронизации двух TPM-сетей в зависимости от параметров m (количество нейронов скрытого слоя) и n (количество входов каждого нейрона) архитектуры при постоянном параметре $L = 4$. Все данные на рис. 3 получены с помощью модели TPMCrypt с параметрами loop = 2500, cycle = 200.

На рис. 3, *а* показана зависимость частоты достижения синхронизации за шагов $s \leq$ loop обучения от параметров m и n. Например, при $m = 3$, $n = 5$ частота достижения синхронизации $\gamma \approx 0{,}9$. Наибольший интерес для построе-

ния системы генерации общего секретного ключа представляют TPM-сети архитектуры $m \times n > 8$. На графике видно, что частота достижения синхронного состояния для TPM-сетей растет при увеличении параметров m, и при $m \times n > 24$ на отрезке [0, 2500] шагов синхронизация достигается практически всегда.

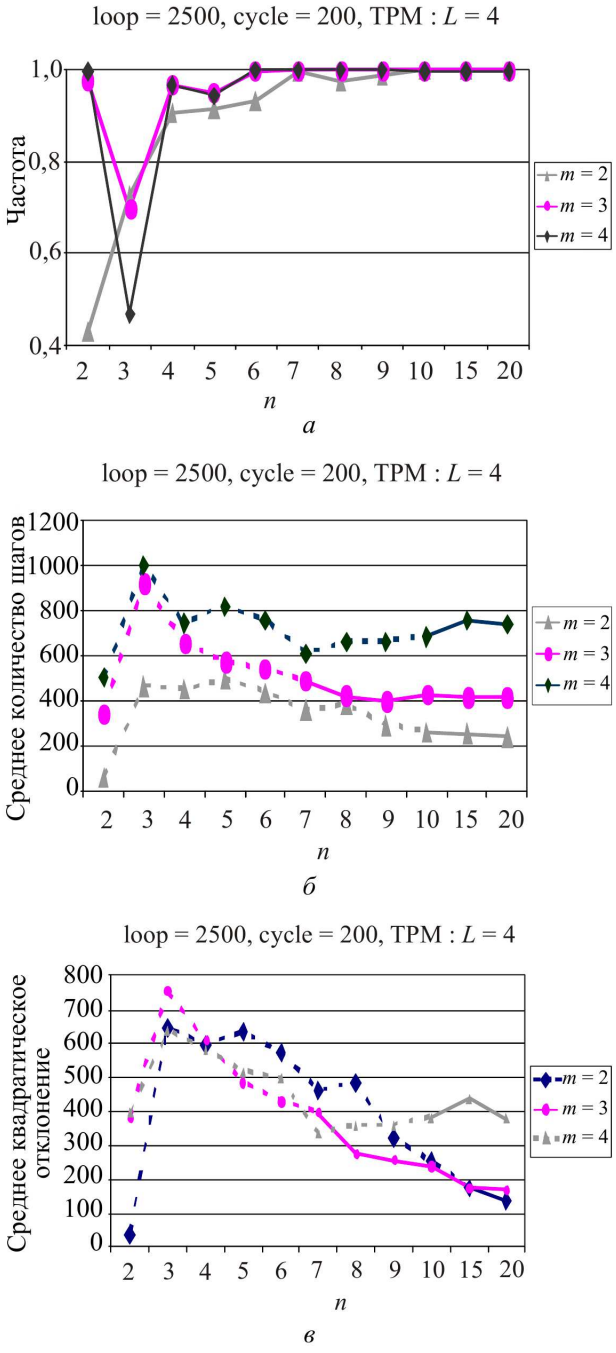

Рис. 3. Оценка зависимости продолжительности процесса синхронизации от архитектуры TPM-сети: *а* – частота достижения синхронного состояния; *б* – среднее количество шагов обучения для достижения синхронного состояния; *в* – среднее квадратическое отклонение количества шагов обучения для достижения синхронного состояния

На рис. 3, *б* приведена зависимость среднего количества шагов обучения для достижения синхронизации от параметров m и n.

На графиках штриховой линией изображены значения, соответствующие архитектуре TPM-сети, при которой не всегда достигается синхронизация за $s \le 2500$ шагов. Поэтому приведенные средние значения являются заниженными, так как при расчете среднего значения учитывались только значения $s \in [0, 2500]$.

На рис. 3, *в* представлена зависимость среднего квадратического отклонения (СКО) количества шагов обучения для достижения синхронизации от параметров m и n. Анализ рис. 3 позволяет сделать вывод, что с ростом значения $m \times n$ при постоянном L среднее количество шагов, необходимое для синхронизации двух сетей, уменьшается и становится более предсказуемым.

На рис. 4 представлена зависимость частоты достижения синхронного состояния за $s \le 2500$ шагов двух TPM-сетей с параметрами $L = 6$, $m = 2$ от параметра n. Сравнение графика, представленного на рис. 4, и соответствующего графика, показанного на рис. 3, подтверждает понятный и ожидаемый результат: увеличение значения L приводит к росту среднего значения шагов обучения, необходимых для синхронизации двух сетей.

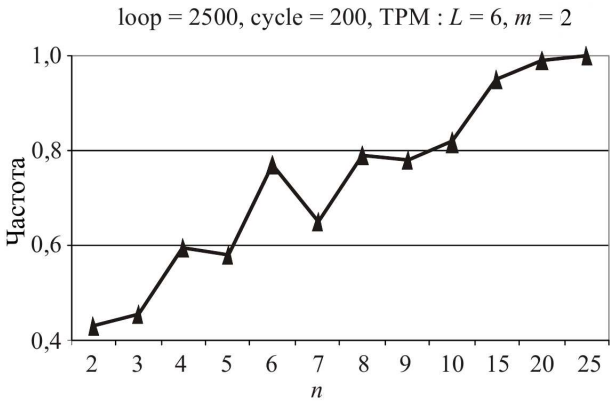

Рис. 4. Зависимость частоты достижения синхронного состояния от количества входов нейронов

Критерий достижения синхронного состояния двух сетей. Остался неисследованным вопрос: каким образом корреспонденты A и B, не пересылая друг другу информацию о состоянии векторов весовых коэффициентов W_k^A или W_k^B, могут определить, что синхронизация TPM-сетей уже осуществилась.

После s шагов обучения последовательность (1) принимает следующий вид: $<W_s^A, W_s^B>$, $<W_{s+1}^A, W_{s+2}^B>$, ..., $<W_{s+r}^A, W_{s+r}^B>$, ..., где $W_{s+i}^A = W_{s+i}^B$, $i \ge 0$, но в общем случае $W_{s+i}^A \ne W_{s+i-1}^B$. Отсюда следует, что для $k \ge s$ всегда $O_k^A \equiv O_k^B$. Поэтому

естественно предположить, что критерием достижения синхронного состояния может служить серия r шагов $k, k+1, k+2, ..., k+r$, для которых $O_k^A = O_k^B$.

Дополнительным критерием, очевидно, может стать общее количество шагов, которое прошло сначала обучения.

На рис. 5 приведена оценка предложенных критериев для TPM-сетей с параметрами $L = 4$, $m = 3$, $n = 8$.

Гистограмма на рис. 5, *а* демонстрирует зависимость частоты достижения синхронного состояния за s шагов обучения. Анализ гистограммы показывает, что в более чем 95% случаев синхронизация таких TPM-сетей наступает менее за 1000 шагов. Частота появления серии $O^A = O^B$ длины r до момента синхронизации представлена на рис. 5, *б*. Очевидно, что если корреспонденты A и B в процессе своего обучения фиксируют серию $r \geq 20$, то можно достаточно точно утверждать, что сети синхронизировались.

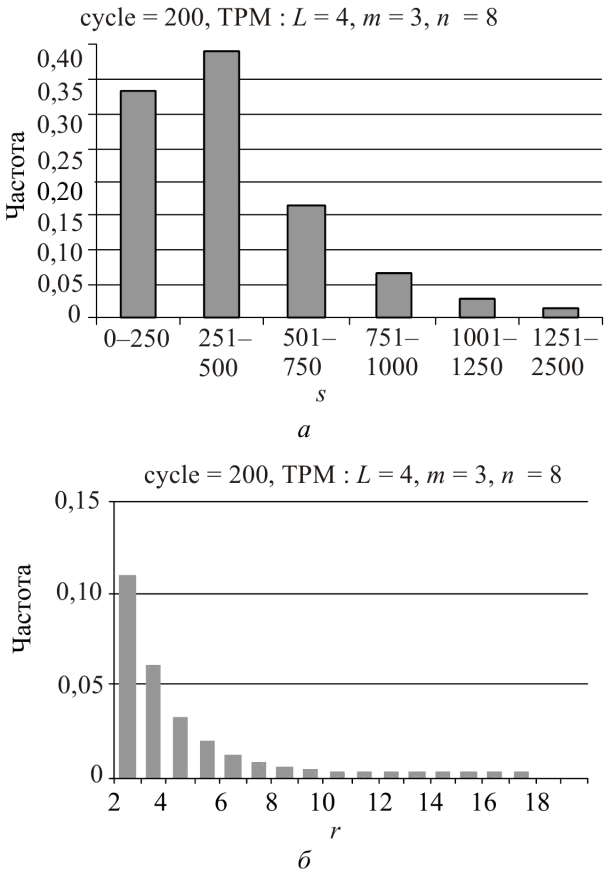

Рис. 5. Частота достижения синхронного состояния TPM-сетей с параметрами $L = 4$, $m = 3$, $n = 8$:
а – частота достижения синхронизации за s шагов;
б – частота серий $O^A = O^B$ длины r до момента синхронизации

Анализ уязвимости системы генерации секретного ключа на основе взаимного обучения TPM-сетей. Будем предполагать, что обмен данными между корреспондентами A (учитель) и B (ученик) происходит по открытому каналу, который прослушивается криптоаналитиком I. При этом считаем, что криптоаналитику известны все параметры архитектуры TPM-сетей.

Процесс синхронизации является случайным. Количество шагов, необходимое для достижения синхронного состояния между сетями TPM-A и TPM-B (рис. 2), зависит от расстояния (например, евклидового) между векторами W_0^A и W_0^B. Разница в средней скорости сходимости векторов W_0^A и W_0^B по отношению к сходимости вектора W_0^C (TPM-сети криптоаналитика) к $W_{0_B}^B$ возникает из-за того, что векторы W_k^A и W_k^B в процессе обучения сходятся к друг другу одновременно, а вектор W_k^C в большинстве случаев «догоняет» W_k^B. Сходимость W_k^C к W_k^B может быть быстрее в том случае, если на одном из шагов вектор W_k^C «пересечется» с траекторией сближения W_k^A и W_k^B.

С точки зрения криптоаналитика синхронизация W_k^C и W_k^B эффективна, если она произошла за $k \leq s + r$ шагов, т. е. до того, как синхронизировались сети TPM-A и TPM-B. Если предположения, сделанные выше, верны, то увеличение количества TPM-сетей на стороне криптоаналитика должно привести к росту частоты достижения эффективной синхронизации.

На рис. 6 изображена модель взаимодействия двух корреспондентов A, B и прослушивающего открытый канал криптоаналитика I.

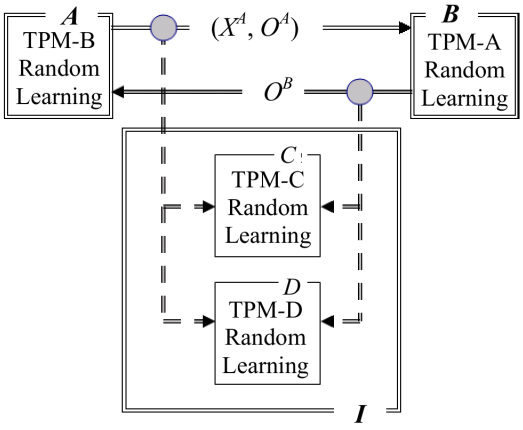

Рис. 6. Модель для исследования уязвимости процесса генерации общего секретного ключа на основе взаимного обучения TMP-сетей A (учитель) и B (ученик)

Криптотаналитик, выполняя роль ученика, пытается синхронизироваться с B. Для синхронизации криптоаналитиком применяются две TPM-сети: C и D. Будем называть мощностью криптоаналитика – количество TPM-сетей, которое он использует для синхронизации с корреспондентом.

На рис. 7 представлена зависимость, позволяющая сделать предположения относительно уязвимости системы генерации общего секретного ключа на основе взаимного обучения двух TPM-сетей.

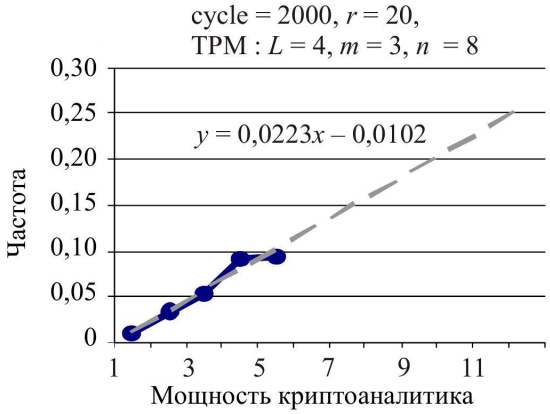

Рис. 7. Зависимость частоты достижения синхронизации хотя бы одной из TPM-сетей криптоаналитика от его мощности

Заключение. Идея применения нейронных сетей архитектуры TPM для генерации общего секретного ключа в той постановке, которая рассматривалась здесь, по мнению автора, требует дополнительного изучения. Проведенные здесь исследования позволили выявить три явных ее недостатка:

1) увеличение мощности криптоаналитика значительно повышает его шансы вычислить общий секретный ключ;

2) процесс синхронизации TPM-сетей может потребовать много шагов обучения, что приведет к большим затратам на передачу информации по сети;

3) для подтверждения завершения процесса синхронизации требуется специальная процедура, которая может быть потенциально опасна для сохранения секрета.

Литература

1. Kanter, I. Secure exchange of information by synchronization of neural networks / I. Kanter, W. Kinzel, E. Katner // Europhys. Lett. 57. – 2002. – P. 141–147.

2. Плонковски, М. Криптографическое преобразование информации на основе нейросетевых технологий / М. Плонковски, П. П. Урбанович // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика. – 2005. – Вып. XIII. – С. 165–167.

Поступила 28.02.2011