УДК 681.382

ПЛОНКОВСКИ М., УРБАНОВИЧ П.П.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ В СИСТЕМАХ КРИПТОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ

Рассмотрены возможности использования нейросетевых технологий для обмена тайными ключами между отправителем и получателем сообщения в системах связи на основе криптографического преобразования информации.

Как известно, безопасная передача данных – это один из важных аспектов, характеризующих как канал передачи, так и само сообщение. Вместе с тем, передача сообщений посредством Интернет-технологий сопряжена с опасностью несанкционированного доступа к передаваемой информации. Потому важную роль в этой области играют криптографические методы, позволяющие шифровать важную информацию и, тем самым, повышать уровень ее защиты.

Классическая криптография основывается на теории чисел. В связи с этим эффективность криптографических систем зависит от трудности разрешения некоторых проблем, связанных с теорией больших чисел (например, проблемы факторизации, дискретного логарифма).

Идея использования нейросетевых технологий для криптографических приложений связана с проблемой обмена ключами через незащищенные каналы передачи. Классический метод согласования ключей и обмена ключами (Diffie-Hellman) основывается на трудности вычисления дискретного логарифма.

Протокол обмена ключами, который используется в нейросетевых криптографических технологиях, опирается на синхронном «обучении» сетей со стороны отправителя и получателя сообщений [1]. Процесс «обучение» двух нейронных сетей с использованием их общих параметров ведётся до появления так называемых идентичных весовых коэффициентов (векторов весов) [2]. Сети обмениваются между собой выходными и входными параметрами; при этом секретными должны оставаться внутренние значения весовых коэффициентов. Интруз (хакер или кракер) не может быстро восстановить эти секретные параметры. Следовательно, значения весовых коэффициентов могут использоваться как секретные ключи в процессе передачи информации по незащищенным каналам.

Проанализируем кратко алгоритм синхронизированного «обучения» нейронных сетей, результатом которого должны быть значения весовых коэффициентов.

Архитектуру сети составляет трехслойный персептрон с одним скрытым слоем, который состоит с K нейронов. Каждый из них имеет N входов. Поэтому размер входного вектора равен $K*N$, а соответствующие входные элементы обозначим как x_{kj} , где k=1, 2, ..., K и j=1, 2, ..., N. Для упрощения полагаем, что каждый элемент вектора X принимает только бинарные величины: $x_{kj} = \pm 1$. Нейроны укрытого слоя обозначаем как $y_1, y_2, ..., y_k$. И, наконец, величины вектора весов – это целые числа, находящиеся в некотором интервале $[-L, L]$ [1].

Сначала каждая из сетей генерирует свой собственный весовой коэффициент. Таким образом, начальные состояния сетей – разные, и начальный коэффициент является величиной секретной. После этого каждая из сетей «обучается» с использованием выходной информации второй сети. Эта информация является открытой и пересылается через канал для синхронизации «обучения» другой сети с использованием архитектуры TPM (Tree Parity Machine) [3].

Выходной параметр каждой сети может быть вычислен с использованием следующего соотношения:

$$O^{A/B} = \prod_{k=1}^{K} y_k^{A/B} = \prod_{k=1}^{K} \sigma\left(\alpha_k^{A/B}\right) = \prod_{k=1}^{K} \sigma\left(\sum_{j=1}^{N} w_{kj}^{A/B} x_{kj}\right) \quad (1)$$

где индекс *A/B* означает, что данный параметр (операция) относится к сетям *A* и *B*; единичный индекс *A* или *B* означает, что операция относится только к одной сети (*A* или *B*); σ-функция – это модифицированная функция знака, которая определяется следующим образом:

$$\sigma\left(\alpha_k^{A/B}\right) = \begin{cases} 1, \alpha_k^{A/B} > 0 \vee \alpha_k^{A} = 0, \\ -1, \alpha_k^{A/B} > 0 \vee \alpha_k^{B} = 0. \end{cases} \quad (2)$$

«Обучение» сетей происходит в соответствии с правилом Hebb'a. Однако весовые коэффициенты обеих сетей модифицируются только тогда, когда исходные входные величины являются одинаковыми. Кроме того, модифицируются веса только тех укрытых нейронов, для которых выходная величина равна выходной величине для всей сети. В соответствии с этим правило «обучения» в формализованном виде может быть описано следующим образом:

$$w_{kj}^{A/B} = \begin{cases} w_{kj}^{A/B} + O^{A/B} x_{kj} & , O^{A} = O^{B} \wedge O^{A/B} y_k^{A/B} > 0 \\ w_{kj}^{A/B} & , \textit{в противном случае.} \end{cases} \quad (3)$$

В конструкцию архитектуры нейронной сети TPM введено дополнительное условие, ограничивающие величины весов в интервале [-*L*, *L*]. Активизация весов требует выполнения следующей операции:

$$w_{kj}^{A/B} = \begin{cases} sign(w_{kj}^{A/B}) L & , \left| w_{kj}^{A/B} \right| \\ w_{kj}^{A/B} & , \textit{в противном случае.} \end{cases} \quad (4)$$

Процесс синхронизированного «обучения» начинается с инициализации векторов весов обеих нейронных сетей. Затем на каждом шаге будет сгенерирован случайный входной вектор *X*. Каждая сеть вычисляет на его основе свою исходную величину (в соответствии с (1) и (2)) и высылает это значение в адрес другой сети. В соответствии с этим происходит «обучение» сетей с использованием выражений (3) и (4).

Состояние синхронизации будет достигнуто тогда, когда обе сети характеризуются одинаковыми выходными параметрами для соответствующих входных сигналов. Это означает практически, что обе сети имеют одинаковые весовые коэффициенты, которые могут быть использованы как тайный ключ.

Согласование тайного ключа с использованием нейронных сетей имеет несколько достоинств. Во-первых, описанный алгоритм достаточно прост и может быть легко реализован. Во-вторых, объем вычислений, необходимых для генерации ключа, сравнительно мал. В-третьих, для каждого высланного сообщения может быть сгенерирован иной ключ, предназначенный для зашифрования этого сообщения [3].

Безопасный обмен ключами должен характеризоваться следующим свойством: оппонент, который знает все детали протокола и содержание сообщений, не должен никоим образом вычислить тайный ключ. Если оппонент (например, *C*) применяет правило «обучения», определенное соотношениями (3) и (4) в отношении выходных величин сетей *A* и *B*, то через некоторый промежуток времени *t* между сетью *C* и сетями *A*

и B должна наступить синхронизация. Однако время t значительно длиннее, чем время входа в синхронизм сетей A и B. Результаты вычислен подтверждают, что в среднем время t синхронизации сети C с сетями A и B в 1000 раз больше, чем время синхронизации между сетями A и B [3]. Отсюда следует, что если процесс синхронизации между сетями A и B завершается на достаточно ранним этапе, то тогда оппонент C не имеет никаких шансов, чтобы вычислить состояние вектора весов сетей A и B. В случае использования других методов криптоанализа (например, генетического или вероятностного [4]) шансы на успех интруза возрастают.

Если «обучение» сетей основывается на методе Oja [5], то (3) может быть преобразовано к следующему виду:

$$w_{kj}^{A/B} = \begin{cases} w_{kj}^{A/B} + O^{A/B}(x_{kj} - w_{kj}^{A/B}) & ,O^A = O^B \wedge O^{A/B} y_k^{A/B} > 0 \\ w_{kj}^{A/B} & ,\text{в противном случае.} \end{cases} \quad (5)$$

Как подтверждают тесты, этой метод обеспечивает увеличение скорости достижения состояния синхронизации. Чтобы получить подтверждение этому, было промоделировано (на основе (5)) функционирование двух архитектур машин TPM (с использованием обоих выше упомянутых методов); при этом K=3 и N=10. Опыт (анализ времени наступления синхронизации) повторялся 100 раз. Предполагалось, что интруз *C1* при этом использует только выходные параметры сети A, интруз *C2* – выходные параметры обеих сетей, *C3* – только одинаковые параметры обеих сетей.

Полученные результаты свидетельствуют о том, что время достижения синхронизации между сетями A и B уменьшилось почти в 22 раза, однако время t сократилось в 1300 раз. Увеличение количества скрытых нейронов до 10 приводит к тому, что время наступления синхронизации оппонента C с сетями A и B около 16 раз больше, чем время входа в синхронизм самих сетей A и B. Следовательно, предложенный подход позволяет достичь синхронизации за достаточно короткое время при большем уровне безопасности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. I.Kanter, W.Kinzel, E.Kanter. Secure exchange of information by synchronization of neural networks// Europhys, 2002. – Lett.57. – pp.141-147 (http://arxiv.org/abs/cond-mat/0202112)

2. W.Kinzel, I.Kanter. Neural Cryptography // 9th Intern. Conf. on Neural Information Processing, Singapore, Nov. 2002 (http://arxiv.org/abs/cond-at/0208453)

3. M.Volkmer, S.Wallner. Tree Parity Machine Rekeying Architectures // Cryptology ePrint Archive: Report 2004/216 (http://eprint.iacr.org/2004/216/)

4. A.Klimov, A.Mityaguine, A.Shamir. Analysis of Neural Cryptography // Proc. of AsiaCrypt, 2002. – Vol.2501 of LNCS. – pp.288-298. – Springer Verlag, 2002. (http://cryptome.org/neuralsub.ps)

5. J.Korbicz, A.Obuchowicz, D.Ucinski. Sztuczne sieci neuronowe. Podstawy i zastosowania. – Warszawa: Akademicka Oficyna Wydawnicza PLJ, 1994

Плонковски Мартин
Магистр
Люблинский университет KUL, г.Люблин, Польша
Тел.: (+10 48) 445-45-50
E-mail: plomyk@kul.lublin.pl