УДК 681.324

М. Плонковски, аспирант; Д. П. Урбанович, аспирант

СИНХРОНИЗАЦИЯ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ И В СИСТЕМАХ КРИПТОПРЕОБРАЗОВАНИЯ НА ОСНОВЕ XML

In article the analysis of safety of the report of an exchange by keys in a context of mutual training two neural networks is submitted, and also some aspects of use of cryptographic transformation of the information in XML environment are analyzed.

В [1] предложен новый протокол обмена ключами, использующий взаимное обучение нейронных сетей и базирующийся на известной архитектуре TPM (tree parity machine).

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

В [2] проанализирована возможность расширения упомянутого протокола за счет использования комплексных чисел в процессе обучения сетей. Ниже будут даны новые результаты в этом направлении, а также будет проанализирована иная возможность использования криптографических технологий: в критических системах реализации транзакций.

Как известно, взаимное обучение двух сетей ведет к синхронизации их векторов весов (весовых коэффициентов). Протокол обмена ключами, который используется в нейросетевых криптографических технологиях, опирается на синхронном «обучении» сетей со стороны отправителя и получателя сообщений. Процесс «обучения» двух нейронных сетей с использованием их общих параметров ведется до появления так называемых идентичных весовых коэффициентов (векторов весов). Сети обмениваются между собой выходными и входными параметрами; при этом секретными должны оставаться внутренние значения весовых коэффициентов. Следовательно, значения весовых коэффициентов могут использоваться как секретные ключи в процессе передачи информации по незащищенным каналам.

Архитектура TPM состоит из двух уровней. Первый составляет K независимых персептронов, из которых каждый характеризуется N-элементным вектором весов ($[w_{k,1}, w_{k,2}, ..., w_{k,N}]$, где $1 \le k \le K$). Коэффициенты этих векторов – это целые числа с интервала $[-L, L]$. Входы персептронов составляет K N-элементных векторов ($[x_{k,1}, x_{k,2}, ..., x_{k,N}]$; k часто отождествляется с одним $N*K$-элементным вектором $[x_1, x_2, ..., x_{kN}]$), выбираемым из двухэлементного массива целых чисел $\{-1, 1\}$. Выходы нейронов – это также целые числа (относятся к тому же массиву $\{-1, 1\}$), обозначим через $y_1, y_2, ..., y_K$. Выход O архитектуры TPM – tree parity machine – (сети A или B) вычисляется как произведение выходов персептронов в соответствии со следующей формулой [2]:

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

где σ – это модифицированная функция знака, которая определяется следующим образом: $\sigma(\alpha_k^{A/B}) = 1$, при $\alpha_k^{A/B}$ положительном; если же $\sigma(\alpha_k^{A/B}) = -1$, то $\alpha_k^{A/B}$ должно быть отрицательным.

Активизация весов (синхронизация сетей) происходит только тогда, когда выходные значения обеих сетей одинаковы ($O^A = O^B$). Кроме того, активизируются только весы тех нейронов, значение на выходе которых равно значению на выходе целой архитектуры TPM. Они могут быть описаны следующими соотношениями:

$$\omega_{kj}^{A/B} = \omega_{kj}^{A/B} + O^{A/B} x_{kj},$$

если $O^A = O^B \cup O^{A/B} = y_k^{A/B}$

и $\omega_{kj}^{A/B} = \omega_{kj}^{A/B}$ – в противном случае.

Архитектура TPM налагает ограничение на значения весовых коэффициентов следующим образом:

$\omega_{kj}^{A/B} = \omega_{kj}^{A/B}$, если $\left|\omega_{kj}^{A/B}\right| > L$

и $\omega_{kj}^{A/B} = \text{sign}\,(\omega_{kj}^{A/B})L$ – в противном случае.

Процесс взаимного обучения сетей начинается с инициализации векторов весов. Потом на каждом шаге такого обучения на основе случайно выбранных входных векторов вычисляются значения O^A и O^B.

Архитектуру TPM, основанную на использовании комплексных чисел, условно назовем TPCM (tree parity complex machine).

Архитектура TPCM состоит из двух уровней. Элементами первого уровня являются персептроны, имеющие также N-элементные векторы весов ($[w_{k,1}, w_{k,2}, ..., w_{k,N}]$, где $1 \le k \le K$), коэффициентами которых являются, в том чис-

ле, комплексные величины, ограниченные интервалом $[-L, L]\times[-L, L]$. Входы персептронов отождествляются также часто с одним $N*K$-элементным вектором $[x_1, x_2, ..., x_{kN}]$) комплексных чисел из массивов $\{(1, 1), (-1, 1), (-1, -1), (1, -1)\}$. Выходы же нейронов – это комплексные числа, принадлежащие массивам $\{(1, 0), (0, 1), (-1, 0), (0, -1)\}$, обозначенным далее через $y_1, y_2, ..., y_K$. Выход O архитектуры TPCM вычисляется в основном аналогично способу, описанному выше. Различие состоит в модификации функции знака σ:

$$\sigma\left(\alpha_k^{A/B}\right)=\begin{cases}(1,0), \dfrac{7\pi}{4} < \arg(\alpha_k^{A/B}) \le \dfrac{\pi}{4};\\ (0,1), \dfrac{\pi}{4} < \arg(\alpha_k^{A/B}) \le \dfrac{3\pi}{4};\\ (-1,0), \dfrac{3\pi}{4} < \arg(\alpha_k^{A/B}) \le \dfrac{5\pi}{4};\\ (0,-1), \dfrac{5\pi}{4} < \arg(\alpha_k^{A/B}) \le \dfrac{7\pi}{4}.\end{cases}$$

Модифицируется тоже правило активизации с целью ограничения величин векторов весов (отдельно для каждого элемента комплексного числа):

$Re(\omega_{kj}^{A/B}) = \text{sign}\,(Re(\omega_{kj}^{A/B}))L$, если

$(\omega_{kj}^{A/B}) > L$ и $Re(\omega_{kj}^{A/B}) = Re(\omega_{kj}^{A/B})$ – в противном случае. Вместе с тем $Im(\omega_{kj}^{A/B}) = \text{sign}\,(Im(\omega_{kj}^{A/B}))L$, если $|\omega_{kj}^{A/B}| > L$ и $Im(\omega_{kj}^{A/B}) = Im(\omega_{kj}^{A/B})$ – в противном случае.

Процесс обучения происходит по тому же алгоритму, что и для архитектуры TPM.

Коэффициенты векторов весов – это комплексные числа, ограниченные квадратом: $[-L, L]\times[-L, L]$.

Выходные параметры архитектуры TPCM как произведение выходных параметров персептронов принадлежат также к области значений, представленной на рисунке.

В архитектуру TPCM количество комбинаций внутренних выходных величин персептронов, обеспечивающих одинаковое значение выходной величины, взрастает квадратично. Например, для $K = 3$ существует 16 комбинаций, при которых $O = 1$: (1, 1, 1), (–1, –1, 1), (1, –1, –1), (–1, 1, –1), (i, i, –1), (i, –1, i), (-1, i, i), (–i, i, 1), (–i, 1, i), (i, 1, –i), (i, –i, 1), (1, i, –i), (1, –i, i), (–i, –i, –1), (–i, –1, –i), (–1, –i, –i)). Поэтому можно предположить, что системы обмена ключами, основанные на архитектуре TPCM, будут характеризоваться большим уровнем безопасности чем архитектуры TPM.

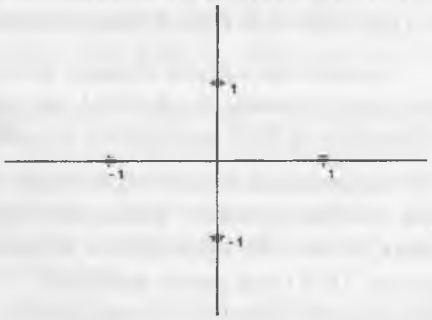

Рисунок. Точки, которые используются как выходные величины персептронов

Для упрощения считаем, что архитектуру TPM составляют 2 ($K = 2$) персептрона. Определяем $p_k = P[y_k^A = y_k^B]$ и для упрощения предполагаем, что $p_1 = p_2 = p$. Имеем четыре возможности: $(y_1^A = y_1^B, y_2^A = y_2^B)$, $(y_1^A \ne y_1^B, y_2^A = y_2^B)$, $(y_1^A = y_1^B, y_2^A \ne y_2^B)$, $(y_1^A \ne y_1^B, y_2^A \ne y_2^B)$, которым соответствуют следующие вероятности: p^2, $p(1–p)$, $(1–p)p$ и $(1–p)^2$. Пары 1, 2 и 3 из числа вышеперечисленных означают, что $O^A \ne O^B$. Итак, вероятность того, что весовые коэффициенты (векторы весов) сетей A и B совпадут, составляет

$$\frac{p^2}{p^2+\frac{1}{3}(1-p)^2}.$$

В то же время вероятность такого совпадения между сетями B и C составляет p. Так как $0 < p < 1$, легко можно доказать, что

$$\frac{p^2}{p^2+(1-p)^2} > p.$$

Несколько иначе ситуация представляется в сетях архитектуры TPCM. Основные допущения примем аналогичными, т. е. как и в предыдущем рассмотрении для упрощения полагаем $K = 2$. Определим, что $p_k = P[y_k^A = y_k^B]$ и $p_1 = p_2 = p$. Имеем четыре ситуации: $(y_1^A = y_1^B, y_2^A = y_2^B)$, $(y_1^A \ne y_1^B, y_2^A = y_2^B)$, $(y_1^A = y_1^B, y_2^A \ne y_2^B)$, $(y_1^A \ne y_1^B, y_2^A \ne y_2^B)$, которым соответствуют следующие вероятности: p^2, $p(1–p)$, $(1–p)p$ и $(1–p)^2$. Паре 1 соответствует ситуация $O^A = = O^B$. Парам 2 и 3 – $O^A \ne O^B$. Однако 1/3 случаев, соответствующих паре 4, приводит к тому, что $O^A = O^B$. Итак, вероятность того, что весовые коэффициенты сетей A и B будут синхронизированы, составляет

$$\frac{p^2}{p^2+\frac{1}{3}(1-p)^2},$$

в то же время для сетей *B* и *C* вероятность такого же события составит *p*. Легко показать, что

$$\frac{p^2}{p^2+\frac{1}{3}(1-p)^2} > \frac{p^2}{p^2+(1-p)^2}.$$

Отсюда следует, что сети *A* и *B* архитектуры TPCM быстрее достигнут синхронизирован-ного состояния в сравнении с оппонентом, чем сети *A*, *B* и *C*, базирующиеся на архитектуре TPM. Теоретические результаты подтверждены на основе моделирования ситуаций, реализованных программно. Установлено, что соотношение времен наступления синхронизации между сетями *A* и *B*, а *B* и *C* архитектуры TPM составляет 0,2, а для архитектуры TPCM – 0,06. Это означает, что сети, базирующиеся на архитектуре TPCM, характеризируются более высоким уровнем безопасности чем сети архитектуры TPM.

Существует известный круг задач, при котором все запросы к какой-либо информационной системе выполняются в одном месте, а затем единственный (частично зашифрованный) XML-документ, предписывая все политики, опубликовывается. Пользователи нуждаются в надлежащих ключах, чтобы обратиться к ограниченным данным. Документ может быть загружен, скопирован, и его доступные части частично обрабатываются соответствующим пользователем прежде чем быть отправленными другому с иными правами доступа. Ключи поддерживаются и управляются владельцем данных, который предоставляет их определенным пользователям. Передача ключей может быть описана в сценарии передачи данных или же ключи могут использовать интерактивные протоколы их передачи для предписания сложной политики управления доступом.

Ситуация может характеризоваться на практике следующими особенностями: во время сканирования web-сайтов наступает обмен заданиями, генерируемыми браузером, а также ответами на эти задания, генерируемыми сетевым сервером.

Для того чтобы предпринимаемые превентивные меры были эффективными, следует обеспечить безопасную передачу информации между всеми элементами системы.

Каждая транзакция между приложением WWW и Пользователем начинается с обращения, сгенерированного поисковой системой в адрес сервера сети Web. Если искомым объектом будет скрипт, то сервер передает его обработку соответствующему механизму, который обслуживает скрипты (предположим, на основе PHP). Скрипт может быть предназначен для считывания информации с диска либо записи информации на диск. Кроме того, скрипт может содержать инструкции для считывания внешних фрагментов кода (функции `include();` или `require()`, а также присоединения кода либо активизации базы данных (БД).

Например, в системе электронной торговли проблему безопасности можно рассматривать в трех аспектах: компьютер Пользователя (Клиента), сеть Интернет, система Интернет-магазина.

На наш взгляд, следует позаботиться, прежде всего, чтобы сама система Интернет-магазина в наибольшей степени отвечала требованиям безопасности. Такая система состоит из трех компонент: сервера WWW, на котором хранятся сайты и скрипты; технологии / языка программирования (PHP); системы управления БД (например, MySQL или PostgreSQL).

Как показывает практика, уже во время инсталляции появляются проблемы. Например, в случае использования PHP следует иметь в виду, что больший уровень безопасности системы и также более высокая производительность будут достигнуты, если в качестве отдельного модуля (а не отдельного скрипта CGI) будет инсталлирован SAPI (*Server Application Programming Interface*) сервера WWW.

В отношении безопасности данных, пересылаемых между компонентами, например, Интернет-магазина, хорошим решением является инсталлирование сервера БД на том же компьютере, что и сервера WWW. В противном случае – механизм PHP, подсоединяясь к такому серверу, пользуется протоколом TCP/IP и, как следствие, передает данные в незакодированном виде.

В Интернет-магазине протокол SSL (*Secure Socket Layer*) может быть использован в механизме идентификации (пароль и идентификатор), а также в конечной фазе реализации покупки (во время инициализации скрипта, «отвечающего» за генерацию протокола покупки (накладной)). Большинство серверов WWW имеют встроенные функции обслуживания этого протокола. Поисковые системы Internet Explorer и Netscape Navigator обслуживают протокол SSL одной из последних версий.

Существующие компьютерные системы состоят из большого количества объектов данных, к которым имеют доступ многие пользователи, взаимодействуют со средой на основе разработанной политики безопасности системы. Права

доступа в такой среде являются или слишком свободными и поэтому потенциально опасными (подобно системам DAC [3, 4]), или слишком ограниченными для большинства применений (подобно системам с принудительным контролем доступа [5]). Именно поэтому получили развитие модели управления доступом на основе иерархий (подобно RBAC [6]).

Такая модель может быть с успехом использована, в частности, при формализации процедур доступа к БД, являющейся одним из элементов Интернет-магазина. При таком подходе право единственного доступа заменено контролем совокупностей прав, названных ролями. Каждая роль представляет собой набор прав, необходимых для того, чтобы выполнить конкретные задачи (считать, записать, модифицировать, удалить информацию или выполнить другие операции). Права пользователей определяются при этом таким образом, чтобы пользователи могли решать определенные задачи через назначение им соответствующего набора ролей. В системах, базирующихся на основе таких взаимосвязей, имеется недостаток, связанный именно с иерархией ролей и их администрированием. Упрощение модели может быть достигнуто посредством группирования объектов.

Кроме вышепроанализированных, существуют два традиционных подхода к управляемому доступу к данным. Первый сохраняет данные на безопасном сервере, который аутентифицирует пользователей и предписывает политику доступа, не размещая данные для свободного доступа. При другом подходе владелец данных должен опубликовать множественные представления данных, одно для каждого пользователя или класса пользователей. Этот подход к управляемой публикации данных имеет несколько недостатков. Один – это то, что количество представлений может стать очень большим. Другой заключается в том, что данный подход препятствует дальнейшему распространению данных: пользователи не могут далее публиковать данные, которые они получают от владельца.

Описанная модель является расширением известной модели RBAC. В случае, когда для каждого объекта определяется свой домен, модель соответствует классической модели RBAC систем.

В описанной модели нет никакого различия между простыми и административными ролями, потому что каждая роль может иметь административные права. Описанная модель достаточно проста в практических применениях. Следует также подчеркнуть, что данный механизм контроля за доступом к объектам может быть компактно представлен в XML [5].

Литература

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

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

3. Урбанович Д. П., Стеенвег Т. Управление доступом к XML-документу, размещенному в Internet // Материалы конференции «Организационно-техническое управление в межотраслевых комплексах», 28–29 октября 2004 г., Минск. – Мн.: БГТУ, 2004. – С. 323–331.

4. G. S. Graham, P. J. Denning. Protection – principles and practice, Proc. Spring Jt. Computer Conf. – Vol. 40, AFIPS Press, Montvale, N. J. – 1972.

5. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler (eds.) Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, October, 2000.

6. R. Sandhu, E. J. Coyne, H. L. Feinstein, Ch. E. Youman. Role-Based Access Control Models, IEEE Computer. – Vol. 29, N 2, February, 1996.