ПРЕГЛЕДНИ ЧЛАНЦИ
ОБОЗОРНЫЕ СТАТЬИ
REVIEW PAPERS

Protić, D., Neural Cryptography, pp. 483–495

NEURAL CRYPTOGRAPHY

Danijela D. Protić
General Staff of Serbian Army Forces, Department for Telecommunication and Informatics (J-6), Centre for Applied Mathematics and Electronics, Belgrade, Republic of Serbia
e-mail: adanijela@ptt.rs,
ORCID iD: http://orcid.org/0000-0003-0827-2863

DOI: 10.5937/vojtehg64-8877

FIELD: Telecommunications, Information Security
ARTICLE TYPE: Review Paper
ARTICLE LANGUAGE: English

Abstract:

Neural cryptography based on the tree parity machine (TPM) is presented in this paper. A mutual learning-based synchronization of two networks is studied. The training of the TPM based on the Hebbian, anti-Hebbian and random walk as well as on the secure key generation protocol is described. The most important attacks on the key generation process are shown.

Key words: *Tree parity machine; Neural networks; Cryptography.*

Introduction

Information security (IS) is a set of processes, methodologies and procedures for protecting the information and information systems from unauthorized access, use, modification, or destruction. Protecting information in potentially hostile environments is a crucial factor in the growth of the information-based processes in the industry, business, and administration. Cryptography is a key technology for achieving IS in communications and computer systems (Basin and Peterson, 2015) defined as the exchange data into a mingle code that can be deciphered and sent across public and private networks (El-Zoghabi et al., 2013). Cryptography provides four services: privacy, authentication, data

integrity and non-repudiation. Privacy ensures that communication between two parties remains secret; authentication is required to ensure that information is exchanged with a legitimate party; data integrity refers to overall completeness, accuracy and consistency of data while non-repudiation assures that parties in communication cannot deny the authenticity of their signatures on a document or sending a message they generated (Rouse, 2015).

In recent years, cryptography as the application of artificial neural networks (ANNs) has become more attractive and more widely studied (Zhou et al. 2004), (Klein et al., 2004), (El-Zoghabi et al., 2013). Initially inspired by neuroscience to imitate neural structures of the brain, ANNs have been used to solve problems where analytical solutions have failed. They are powerful tools for finding the decision automatically by calculating appropriate parameters (weights) to make the compatibility of one system to another (Abed, 2012).

Neural networks learn from examples. A "teacher" network is presented by input/output pairs of data and a "student" network is being trained on this data. After training, the "student's" weights overlap with the "teacher's". As a consequence, the "student" can classify an input pattern that does not belong to the training set. It was shown that two randomly initialized feedforward neural networks (FNNs), with one layer of hidden units (Protić, 2015), which are trained on their mutual output, can synchronize identical time-dependent weight vectors. Synchronization can be used for a creation of a secure secret key as well as to construct a key exchange protocol for a secure transmission of secret data using a public channel (Rosen-Zvi et al., 2002). In comparison with traditional supervised learning, there is no fixed target function in mutual learning, because each of communicating parties acts as a "teacher" and a "student" simultaneously. Both of the two parties' statuses are chaotic, driven by a random input (Zhou et al., 2004).

The advantage of neural cryptography is that the algorithm used in generating a secret key can be a simple perception of the Tree Parity Machine (TPM). Synchronization of TPMs by mutual learning only works if both machines receive a common sequence of (random) input vectors. For that purpose, each communication party uses a separate, but identical pseudo-random number generator (PRNG). By mutual learning, two parties synchronize their networks without transmitting inputs over a public channel. Having reached full synchronization, parties can authenticate each other by knowing weight vectors which are identical, and represent a secret key.

This article presents the mutual learning of two TPMs based on the Hebbian, anti-Hebbian and random walk learning rules. The following paragraph presents the structure of the TPM. The secret key generation

and the attacks on TPMs are described in Chapter 3. The analytical and statistical results are presented in Chapter 4. The last chapter concludes the paper.

Tree parity machine

The parity machine (PM) is a neural network applied in cryptography to generate a secret key. It is also used for a key exchange protocol. The TPM network is in fact an FNN that has input layer neurons constructed in the McCulloh-Pitts model (Protić, 2015), (Dolecki and Kozera, 2015). In the second layer, the network has neurons with specific activation functions. The outcome of the output neuron is the results of the entire PM network. Each PM network is described by three parameters: the number of hidden neurons - K, the number of input neurons connected to each hidden neuron - N, and the maximum value for weight {-L, ... L}. A PM consists of KN random input elements x_{ji} = ±1, j = 1…N, K binary hidden units σ_i = ±1, i = 1,…, K, and one binary output unit $\tau=\prod_i\sigma_i$, where σ_i is determined via the function $\sigma_i = \text{sign}(\sum_j w_{ji}x_{ji})$. A PM that has three neurons in the hidden layer (K=3) is called the three parity machine, shown in Figure 1.

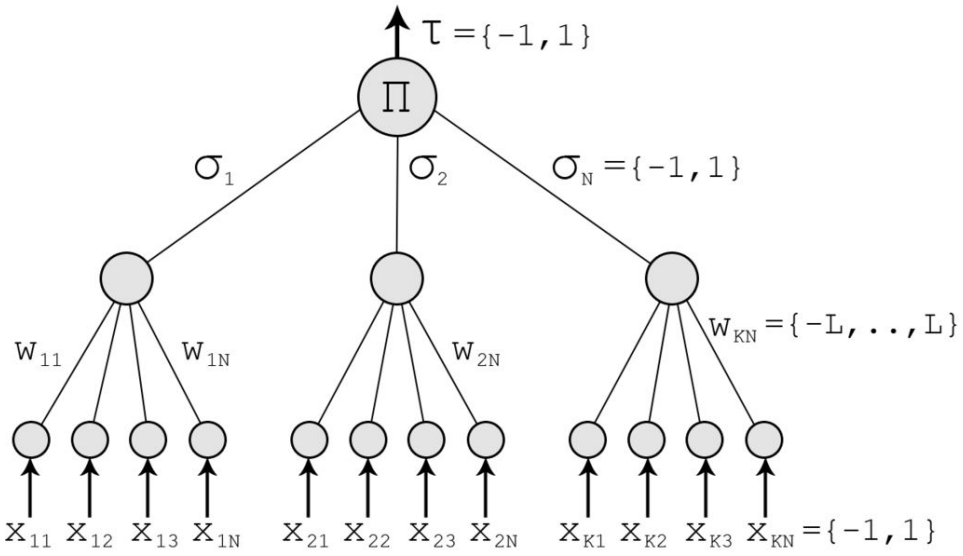

Figure 1 – Tree parity machine
Slika 1 – Tree parity mašina
Рис.1 – Tree parity машина

Function sign(x) is given with the formula

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

To calculate the output value, the following method is used (1)

$$\tau=\prod_{i=1}^{K} sign\left(\sum_{j=1}^{N} w_{ij}x_{ij}\right) \quad (1)$$

Training the tree parity machine

Both communication parties, A and B, are using a TPM with K hidden units (Kinzel and Kanter, 2002). In this example, each network consists of three σ units:

$$\sigma_i^A = sign\left(w_i^A x_i\right),\ \sigma_i^B = sign\left(w_i^B x_i\right)$$

$$w_{i,j}^{A/B} \in \left\{-L,\ -L+1,\ \ldots\ L-1,\ L\right\},\ x_{i,j} \in \left\{-1,\ 1\right\}$$

Three hidden bits σ are combined to an output bit τ of each TMP output bit

$$\tau^A = \sigma_1^A\sigma_2^A\sigma_3^A,\ \tau^B = \sigma_1^B\sigma_2^B\sigma_3^B$$

The two output bits are used for a mutual training process. At each training step, A and B receive identical input vectors $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and the corresponding output bit of their partner. The training algorithm goes in this way (Ruttor et al., 2004):

1) if the output bits are different, $\tau^A \neq \tau^B$, nothing is changed

2) if $\tau^A = \tau^B = \tau$, only hidden units which have an output bit identical to the common output $\sigma^{A/B} = \tau^{A/B}$ are trained

3) to adjust the weights, consider three learning rules

a. Hebbian learning rule

$$w_k^+ = g\left(w_k + \varpi x_k \Theta(\sigma_k\tau)\Theta(\tau_A\tau_B)\right)$$

b. Anti-Hebbian learning rule

$$w_k^+ = g\left(w_k - \varpi x_k \Theta(\sigma_k\tau)\Theta(\tau_A\tau_B)\right)$$

c. Random walk

$$w_k^+ = g\left(w_k + x_k \Theta(\sigma_k\tau)\Theta(\tau_A\tau_B)\right)$$

Θ is a special function:

$$\Theta(a,b)=\begin{cases}0, & a<b\\ 1, & a=b\\ 0, & a>b\end{cases}$$

Only weights are changed by these learning rules, which are connected to hidden units with $\sigma_i = \tau$. By doing so, it is impossible to tell which weights are updated without knowing the internal representation (σ_1, σ_2 … σ_K). This feature is especially needed for the cryptographic application of neural synchronization. The learning rules have to assure that the weights stay in the allowed range between -L and +L. If any weight moves outside this region, it is reset to the nearest boundary value ±L (Ruttor, 2006). This is achieved by the function $g(w)$ in each learning rule:

$$g(w)=\begin{cases}sign\,(w)L, & |w|>L\\ w, & otherwise\end{cases}$$

Afterwards, the current synchronization step is finished. This process can be repeated until the corresponding weights in A's and B's TPM have equal values $w_i^A = w_i^B$. Further applications of the learning rule are unable to destroy this synchronization, because the movements of the weights depend only on the inputs and weights, which are then identical in A's and B's networks.

Secret key encryption

Compared to other algorithms based on the number theory, the neural algorithm has several advantages. Its simplicity is the most important of them. Also, it is easy to implement in hardware. Moreover, the number of calculations is low. For every communication, a new secret key can be generated. In this way, no secret information has to be stored for a long time. The safety of algorithm is based on short time to train the TPM. Attackers who know all details of the protocol and the information exchanged over the public channel should not have the computational power to calculate a secret key.

Secret key generation

After a relatively small number of training steps, all pairs of the weight vectors are identical, $w_i^A = w_i^B$. The two multilayer networks have identical synaptic weights. Weights can be used as a key for encryption, a seed for random bit generators (RBGs) or a key in other encryption algorithms (AES, DES, 3DES).

Attacks

Security of neural synchronization is put at risk if an attacker E is capable of synchronizing with any of the two parties during the training of TPMs. The learning time of the attacker that is trying to imitate one of parties in communication is reported to be much longer than synchronization time (Abed, 2012).

Assume that E knows the algorithm, the sequence of input vectors and the corresponding output bits. In time t=0, E could start from all of the $(2L+1)^{3N}$ initial weight vectors and calculate the one that is consistent with the input/output sequence. It has been shown that all of these initial states move toward the same final weight vector (Kinzel and Kanter, 2002). This task is, however, computationally infeasible.

A given protocol can be broken by geometric, probabilistic, and genetic attacks (Kilmov et al., 2002), as well as under regular and majority flipping attacks (Zhou et al., 2004) and as such it is not entirely secure. One of the attack strategies involves a large number of cooperating attackers that succeeds in revealing the secret key. However, in each time step, every attacker is multiplied to cover 2^{K-1} possible interpretations of σ_i.

Analytical and statistical results

Analytical calculations show that the attacker E can synchronize with A and B after some learning time which is about 1000 times slower than synchronization (Kanter et al., 2002). Why is that? The answer is simple: The difference between the communication parties and the attacker is that partners can influence each other's network, while the attacker only listens. Some attacks are presented here.

1. Klimov et al. (2002) proposed the*Genetic attack* in which a large population of attackers is trained, and every new time step each attacker is multiplied to cover 2^{K-1} possible representations of $\{\sigma_i\}$ for the current output. As dynamics proceeds, successful attackers stay while others are rejected.

2. The same group of authors has described the *Probabilistic attack* in which the attacker tries to follow the probability of every weight element by calculating the distribution of the local field of every input and using the output.

3. In the *Regular flipping attack* (RFA), the attacker imitates one of the parties. In a step in which his output disagrees with the imitated party, he alters (negates) the sign of one of his hidden units. While the synchronization time increases with L^2 the probability of finding a successful flipping-attacker decreases exponentially with L.

4. The strategy of the *Majority flipping attack* (MFA) is to use attackers as a group rather than as individuals who cooperate. All weights are updated in accordance with the majority's results. Wang (2015) defines the attacker to be successful if he is able to guess 98% of the weights.

5. Up to now, the most successful attack on neural cryptography is the *Geometric attack*. The attacker uses the same TPM and training step as A and B. Only for $\tau^A = \tau^B$, E performs the training step. When τ^E agrees with the two parties that communicate, E trains hidden units which agree with the common output. For $\tau^E \neq \tau^{A/B}$ the attacker first inverts the output bit σ_i of the hidden unit with the smallest absolute value of the internal field and then performs the training step. In this case, the probability that an attacker can synchronize with A or B is non-zero. Consequently, if the attacker uses an ensemble of sufficient TPMs, there is a good chance that he can find a secret key (Mislovaty et al., 2002).

Conclusion

In the last decade, mutual learning based on the parity machine has become popular to be used for cryptography. In this paper, a three parity machine that is a special type of a feedforward neural network with three artificial neurons in the hidden layer, one output neuron and KN input elements is presented. Inputs to the network are binary numbers, while the weights between inputs and hidden neurons take predefined values. The output of the hidden neuron is calculated as a weighted sum of all multiplications of inputs and weights. If the scalar product is zero, the output of the hidden neuron is mapped to -1, otherwise it is mapped to 1, in order to ensure a binary output from the network. The output bit of the network is a product of the three bits of the hidden units.

Mutual learning is used for synchronization between two parties that communicate across a public or private channel. During the synchronization, both communication parties use the same tree parity machines, receive common binary inputs generated randomly from the same random number generator, and exchange output bits. Adjusting the weights according to the learning rules leads to full synchronization in a finite number of steps. Networks trained on their mutual inputs synchronize to an identical time dependant weight vectors. This phenomenon is used to generate a secret key. The learning rules presented in this paper are the Hebbian learning rule, the Anti-Hebbian learning rule, and the Random walk.

An attacker who knows the input/output pairs can derive weights, by learning with his own tree parity machine. Still, even if he eavesdrops a message between the two parties, he cannot change it. The communication parties benefit from mutual interaction so the passive

attacker is usually unable to learn the secret key in time. This essential mechanism allows synchronization but prohibits learning. However, it is shown that some attacks such as the genetic attack, the probabilistic attack, the regular and majority flipping attacks, as well as the geometric attack can crack the tree parity machine protocol. Therefore, some additional cryptographic functions can be combined with the tree parity machine protocol to enhance security. Recent research presents the results on combining a neural network and chaos synchronization (Klein et al., 2004), splitting the mutual information and the training process against flipping attacks (Zhou et al., 2004), adding feedback (Ruttor and Kinzel, 2004) using tree parity machines of various sizes (depths), applying delayed chaotic neural networks (Yu and Cao, 2006). Prabakaran et al. (2008) used different learning rules with different units. Additional research shows that various techniques for software development and hardware implementation can also improve the security of the tree parity machine (CyberTrone, 2009), (Godhavari, 2010). Nevertheless, the specificity of the presented theme suggests that further development of this topic is needed.

References

Abed, S., 2012, Cryptography Using Artificial Neural Network, Al-dananeer, 14(1), pp.378-402.

Basin, D., Peterson, K., 2015, Information Security and Cryptography, Springer.

CyberTrone 2004, [Internet] Neural Cryptography, Dostupno na Code Project web-site: www.codeproject.com, Preuzeto: 19.08.2015. godine.

Dolecki, M., and Kozera, R., 2015, Distance of the initial weights of tree parity machine drawn from different distributions, Advances in Science and Technology Research Journal, 9(26), pp.137-142.

El-Zoghabi, A., Yassin, A. H., Hussein H. H., 2013, Survey Report on Cryptography Based on Neural Networks, International Journal of Emerging Technology and Advanced Engineering, 3(12), pp.456-462.

Godhavari, T., and Alamelu, Dr. N. R., 2010, Cryptography Using Neural Network, The Technology World Quarterly Journal, 2(4), pp.49-54.

Kanter, I., Kinzel, W., & Kanter, E. 2002. Secure exchange of information by synchronization of neural networks. *Europhysics Letters (EPL),* 57(1), str.141-147. doi:10.1209/epl/i2002-00552-9

Kilmov, A., Mityaguine, A., Shamir, A., 2002, Analysis of Neural Cryptography, AsiaCrypt 2002. pp.288-298.

Kinzel, W., and Kanter, I., 2002, [Internet] Neural Cryptography, Cornel University Library, Dostupno na http://arxiv.org/abs/cond-mat/0208453v1 Preuzeto: 20.08.2015. godine.

Klein, E., Mislovaty, R., Kanter, R., Ruttor, A., and Kinzel, W., 2004, Synchronization of neural networks by mutual learning and its application to cryptography, Advances in Neural Information Processing Systems, pp.689-696.

Mislovaty, R., Perchenok, Y., Kinzel, W., & Kanter, I. 2002. . *Phys. Rev,* . E 66, 066102. doi:10.1103/PhysRevE.66.066102

Prabakaran, N., Loganathan, P., and Vivekanandan, P., 2008, Neural Cryptography with Multiple Transfer Functions and Multiple Learning Rule, International Journal of Soft Computing, 3(3), pp.177-181.

Protić, D., 2015, Feedforward neural networks: The Levenberg-Marquardt optimization and the optimal brain surgeon pruning, Vojnotehnički glasnik/Military Technical Courier, 63(3), pp.11-28.

Rouse, M., [Internet], Nonrepudiation, TechTarget, Dostupno na http://searchsecurity.techtarget.com/definition/nonrepudiation, Preuzeto: 14.08.2015. godine.

Rosen-Zvi, M., Kanter, I., & Kinzel, W. 2002. Cryptography based on neural networks analytical results. *Journal of Physics A: Mathematical and General,* 35(47). 35(47): L707-L713. doi:10.1088/0305-4470/35/47/104

Ruttor, A., 2006, Neural Synchronization and Cryptography, Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Bayerischen Julius-Maximilians-Universitat. Fakultat fur Phyisik und Astronomie, Wurzburg.

Ruttor, A., Kinzel, W., Shacham, L., and Kanter, I., 2004 [Internet], Neural cryptography with feedback, Cornel University Library, Dostupno na: http://arxiv.org/abs/cond-mat/0311607v2 Preuzeto: 17.08.2015. godine.

Wang, D., 2015, Neural Synchronization Using Genetic Algorithm for Secure Key Establishment, Journal of Engineering Science and Technology Review, 8(2), pp.152-156.

Yu, W., and Cao, J., 2006, Cryptography based on delayed chaotic neural network, Physics Letters A, 356(4), Elsevier, pp.333-338.

Zhou, J., Xu, Q., Pei, W. and He, Y., 2004, Step to Improve Neural Cryptography Against Flipping Attacks, International Journal of Neural Systems, 14(6), pp.393-405.

НЕЙРОКРИПТОГРАФИЯ

Даниела Д. Протич
Генеральный штаб Вооруженных сил Республики Сербия,
Управление информатики и телекоммуникаций (J-6), Центр прикладной математики и электроники, Белград, Республика Сербия

ОБЛАСТЬ: телекоммуникации, информационная безопасность
ТИП СТАТЬИ: обзорная статья
ЯЗЫК СТАТЬИ: английский

В работе представлена нейрокриптография, основанная синхронизации на двух древовидных машин четности (TPM, tree parity machines). Представлен анализ синхронизации методом двунаправленного обучения. Описан TPM тренинг, основанный на: правиле Хебба, анти-правиле Хебба и случайном блуждании, а также на протоколе генерации скрытого ключа. В работе приведены самые значимые атаки на процесс генерации ключа.

Ключевые слова: *tree parity машина, анализ речи, нейронные сети, криптография*

VOJNOTEHNIČKI GLASNIK / MILITARY TECHNICAL COURIER, 2016., Vol 64, No 2

NEURONSKA KRIPTOGRAFIJA

Danijela D. Protić
Generalštab Vojske Srbije, Uprava za telekomunikacije i informatiku (J-6),
Centar za primenjenu matematiku i elektroniku, Beograd, Republika Srbija

OBLAST: telekomunikacije, informaciona bezbednost
VRSTA ČLANKA: pregledni članak
JEZIK ČLANKA: engleski

Sažetak:

U ovom radu prikazana je neuronska kriptografija, zasnovana na tree parity mašini (TPM). Proučavana je sinhronizacija dve mreže bazirana na međusobnom obučavanju. Opisan je trening TPM zasnovan na Hebbovom, anti-Hebbovom i random walk protokolu, kao i na protokolu za generisanje tajnog ključa. Prikazani su najznačajniji napadi na proces generisanja ključa.

Uvod

Informaciona bezbednost je skup procesa, metodologija i procedura za zaštitu informacija i informacionih sistema od neautorizovanog pristupa, korišćenja, modifikacije ili uništenja. Zaštita podataka u potencijalno neprijateljskim okruženjima osnovni je faktor u napretku na informacijama zasnovanih procesa u industriji, poslovanju i administraciji. Kriptografija je ključna tehnologija za postizanje informacione bezbednosti u komunikaciji i računarskim sistemima, definisana kao razmena kodovanih podataka koji mogu da budu dešifrovani i preneseni kroz javnu ili privatnu mrežu.

Kriptografija obezbeđuje četiri servisa: privatnost, autentifikaciju, integritet podataka i nemogućnost pobijanja. Privatnost obezbeđuje da komunikacija između strana ostane tajna, autentikacija je zahtevana da bi bilo osigurano da informaciju dele legitimne strane, integritet podataka odnosi se na celovitost, tačnost i konzistentnost podataka, a nepobijanje obezbeđuje da strane u komunikaciji ne mogu da poreknu autentičnost njihovih potpisa na dokumentu ili prenos poruke koju su generisale.

Poslednjih godina kriptografija je, u smislu primene veštačkih neuronskih mreža, postala atraktivnija i široko proučavana. Početno inspirisane imitacijom neuronskih struktura mozga, veštačke neuronske mreže bile su korišćene za rešavanje problema tamo gde nisu bila moguća klasična analitička rešenja. One su snažni alati za nalaženje rešenja automatski, proračunavajući parametre (težine) da bi se postigla kompatibilnost jednog i drugog sistema. Neuronske mreže uče iz primera. „Učitelj" je predstavljen parom ulazno-izlaznih podataka, a „učenik" je treniran na tim podacima. Nakon treninga učenikovi parametri odgovaraju učiteljevim, pa učenik može da klasifikuje ulazne paterne koji ne pripadaju obučavajućem skupu.

Dokazano je da dve slučajno inicijalizovane feedforward neuronske mreže sa jednim skrivenim slojem, koje su trenirane zajedničkim izlazom, mogu da sinhronizuju istovetne vremenski zavisne težinske vektore. Sinhronizacija može biti korišćena za generisanje tajnog ključa, kao i za formiranje protokola za razmenu ključeva za zaštićeni prenos tajnih podataka korišćenjem javnog kanala. U poređenju sa tradicionalnim supervised obučavanjem, u međusobnom obučavanju nema fiksirane ciljne prenosne funkcije, jer se svaka strana u komunikaciji ponaša i kao učitelj i kao učenik.

Prednost neuronske kriptografije jeste da algoritam korišćen u generisanju tajnog ključa može biti rezultat TPM-a. Sinhronizacija TPM-ova međusobnim obučavanjem radi isključivo ukoliko TPM-ovi dobiju istovetnu sekvencu ulaznih vektora. U tu svrhu, svaka strana u komunikaciji koristi svoj, istovetan generator pseudoslučajnih brojeva. Međusobnim obučavanjem dve strane sinhronizuju svoje mreže tako da ne prenose ulaze javnim kanalom. Dostizanjem pune sinhronizacije strane mogu da se identifikuju, jer znaju težinske vektore koji su identični i predstavljaju tajni ključ.

Tree parity mašina

Parity mašina (PM) jeste neuronska mreža koja se primenjuje u kriptografiji za generisanje tajnog ključa. Takođe, koristi se za protokol razmene ključa. To je, u stvari, feedforward neuronska mreža koja u prvom sloju sadrži neurone generisane po McCulloh-Pitts ovom modelu. U drugom sloju nalaze se neuroni sa specifičnom aktivacionom funkcijom. Izlazna vrednost izlaznog neurona je istovremeno izlazna vrednost PM-a. Svaku PM opisuju tri parametra: K – broj neurona skrivenog sloja, N – broj neurona ulaznog sloja povezanih sa svakim neuronom skrivenog sloja i maksimalna vrednost težina. Parity mašina koja ima tri neurona u skrivenom sloju naziva se tree parity mašina.

Trening tree parity mašine odvija se na sledeći način: ako su izlazni bitovi različiti ništa se ne menja; ako su isti menja se vrednost skrivenim neuronima skrivenog sloja i to isključivo onima koji imaju istovetne vrednosti sa svojim izlazima. Za podešavanje vrednosti težinskih parametara koristi se jedno od sledećih pravila obučavanja: Hebovo, anti-Hebovo ili random walk. U svakom koraku ulazi u obe TPM su isti. Nakon što je izvedena sinhronizacija, parametri TPM-ova (težine) služe za generisanje tajnog ključa. Obučavanje mora da obezbedi da težine ostanu u granicama dozvoljenih vrednosti.

Kriptozaštita bazirana na tajnom ključu

U poređenju sa drugim algoritmima koji su zasnovani na teoriji brojeva, neuronski algoritam ima nekoliko prednosti. Jednostavnost je najbitnija od njih. Takođe, laka je implementacija u hardver. Pored toga, broj kalkulacija je mali. Na kraju, za svaku komunikaciju moguće je generisati novi tajni ključ. Na taj način ni jedna tajna informacija ne mora da bude pohranjena dug period. Sigurnost algoritma zasnovana je

na kratkom vremenu za obučavanje TPM-ova. Napadač koji zna sve detalje protokola i informacija koje će biti razmenjene javnim kanalom nema dovoljnu računarsku snagu da izračuna tajni ključ.

Nakon malog broja koraka za obučavanje dve mreže imaju iste parametre koji mogu biti iskorišćeni za ključeve za šifrovanje, generatore slučajnih brojeva ili ključeve za druge kriptografske algoritme. Bezbednost može biti narušena ukoliko napadač može da se sinhronizuje sa makar jednom od strana u komunikaciji. Međutim, vreme obučavanja je znatno duže. Čak i ako se pretpostavi da napadač zna algoritam i sve parove slučajnih ulaznih sekvenci i odgovarajućih izlaznih bitova potreban mu je znatno veći broj kalkulacija, pa je sinhronizacija teška. Međutim, dokazano je da su mogući napadi na TPM-ove. Dati protokol moguće je napasti geometrijskim, propabilističkim i genetskim napadima, kao i klasičnim i većinskim flipping napadom. Nova strategija napada podrazumeva da se uključi veći broj udruženih napadača koji mogu da uspeju da otkriju tajni ključ.

Analitički i statistički rezultati

Analitički rezultati pokazuju da napadač može da sinhronizuje sa A i B nakon određenog vremena obučavanja koje je oko 1000 puta duže od vremena sinhronizacije. To je zbog toga što partneri u komunikaciji mogu da utiču jedan na drugog dok napadač samo sluša. U radu su prezentovani sledeći napadi: genetički napad koji je zasnovan na velikom broju multiplikacija; probabilistički napad koji je baziran na praćenju verovatnoće pogađanja svake težine; RFA gde napadač imitira jednu od strana; MFA koji koristi više napadača kao grupu; geometrijski napad koji koristi praćenje izlaza obe strane u komunikaciji i izvodi trening alternacijom izlaznog bita skrivenog sloja sa najmanjom apsolutnom vrednošću.

Zaključak

U poslednjoj deceniji međusobno obučavanje zasnovano na tree parity mašini postaje popularno za korišćenje u kriptografiji. U ovom radu prikazana je sinhornizacija dve neuronske mreže sa tri elementa skrivenog sloja i jednim izlaznim neuronom. Ulazi u mrežu su binarni brojevi, dok su težine brojevi predefinisanog skupa. Izlaz iz mreže je proizvod izlaza iz neurona skrivenog sloja, koji su računati kao suma proizvoda pojedinih težina sa odgovarajućim ulazima. Ukoliko je skalarni proizvod nula izlazu se dodeljuje vrednost -1, u suprotnom dodeljuje se vrednost 1. Međusobno obučavanje koristi se za sinhronizaciju između dve strane koje komuniciraju javnim ili privatnim kanalom. Strane koriste iste tree parity mašine, primaju istovetne ulazne bitove i razmenjuju izlazne bitove. Podešavanje težina po pravilima sinhronizuje sistem na identične vremenski zavisne težine u konačnom broju koraka. Ovaj fenomen koristi se za generisanje tajnog ključa. Pravila obučavanja koja su prikazana u radu su Hebbovo, anti-Hebbovo i the Random walk.

Napadač koji zna izlazno-ulazne parove može da odredi težine obučavajući tree parity mašinu. Ipak, i ako presretne poruku, on ne može da je menja. Komunikacija između dve strane obezbeđena je međusobnom interakcijom tako da pasivan napadač obično nije u mogućnosti da odredi tajni ključ. Međutim, postoje napadi na ovakav protokol. To su genetski napad, probabilistični napad, regularni i većinski flipping napadi i geometrijski napad. Zbog toga je moguće kombinovati protokol tree parity mašine sa drugim kriptografskim funkcijama, kao što su haotična sinhronizacija, deljenje međusobne informacije u procesu obučavanja protiv flipping napada, dodavanje povratne sprege, odlaganje, korišćenje mašine različitih veličina (dubine), korišćenje različitih pravila obučavanja, itd. Dodatno, istraživanja pokazuju da različite softverske tehnike i hardverske implementacije takođe doprinose bezbednosti tree parity mašine. U svakom slučaju, specifičnost prezentovane teme ukazuje da je istraživanje u ovom pravcu i dalje potrebno.

Ključne reči: *tree parity mašina, neuronske mreže, kriptografija.*

Paper received on / Дата получения работы / Datum prijema članka: 21. 08. 2015.
Manuscript corrections submitted on / Дата получения исправленной версии работы / Datum dostavljanja ispravki rukopisa: 20. 10. 2015.
Paper accepted for publishing on / Дата окончательного согласования работы / Datum konačnog prihvatanja članka za objavljivanje: 22. 10. 2015.