**Marcin PŁONKOWSKI[1], Paweł URBANOWICZ[1], Elena LISICA[2]**

Katolicki Uniwersytet Lubelski Jana Pawła II, Lublin, Polska (1), Białoruski Państwowy Uniwersytet Technologiczny, Mińsk, Białoruś (2)

# Wykorzystanie kwaternionów w protokole uzgadniania klucza kryptograficznego, opartym na architekturach sieci neuronowych TPQM

***Streszczenie.*** *W niniejszym artykule został przedstawiony protokół uzgadniania klucza kryptograficznego, oparty na architekturach sieci neuronowych typu TPQM (wykorzystujących algebrę kwaternionów). Kwaterniony jako kolejne rozszerzenie ciała liczb rzeczywistych mogą być wykorzystane w sieciach neuronowych, gwarantując poprawność prowadzonych operacji matematycznych wykorzystywanych w całym procesie uzgadniania klucza. Dodatkowo zaproponowana nowa architektura pozwala uzyskać wyższy poziom bezpieczeństwa, niż klasyczna architektura TPM oparta na algebrze liczb rzeczywistych.*

***Abstract.*** *This article presents the cryptographic key agreement protocol based on the neural networks architectures of the TPQM type (using the algebra of quaternions). Quaternions, as a subsequent extension of real numbers, can be applied in neural networks, guaranteeing the correctness of the mathematical operations used in the whole process of the cryptographic key agreement. Furthermore, the new proposed architecture ensures a higher level of security than the standard TPM architecture based on the algebra of real numbers (**The use of quaternions in the cryptographic key agreement protocol based on the architectures of the TPQM neural networks**).*

**Słowa kluczowe:** sieci neuronowe, kryptografia.
**Keywords:** neural network, cryptography.

## Wprowadzenie

Uczenie wzajemne dwóch sieci neuronowych prowadzi do synchronizacji ich wektorów wag. Ten fakt pozwala na wykorzystanie go w procesie uzgadniania klucza wykorzystując niezabezpieczony kanał transmisji. Architekturą wykorzystywaną w tym procesie jest TPM (tree parity machine) [1]. Charakteryzuje się ona dość niskim poziomem bezpieczeństwa, pozwalającym osobie atakującej uzyskać stan synchronizacji z obserwowanymi sieciami, a tym samym uzyskać tajny klucz. Dodatkowo w [2] został zaproponowany atak geometryczny, który jest niezwykle groźny w odniesieniu do architektury TPM.

Istotną modyfikacją klasycznej architektury TPM, było wprowadzenie do architektury sieci neuronowej algebry liczb zespolonych. Przyjęła ona nazwę TPCM (tree parity complex machine) [3]. Zaproponowana modyfikacja pozwoliła podnieść poziom bezpieczeństwa systemu uzgadniania klucza.

## 1. Wykorzystanie kolejnych rozszerzeń ciała liczb rzeczywistych w architekturze TPM

Wspomniana we wstępie architektura TPCM, oparta na algebrze liczb zespolonych, charakteryzuje się zwiększonym poziomem bezpieczeństwa (w stosunku do klasycznej architektury TPM). Zalety wprowadzenia tego typu modyfikacji zostały potwierdzone wieloma testami symulacyjnymi [3,4]. Z drugiej strony oprócz zaprezentowanych zalet (zwiększone bezpieczeństwo kryptosystemu) istnieje też istotna wada, związana z wydłużeniem procesu synchronizacji. Związane jest to ze specyfiką operacji na liczbach zespolonych, a także z większym skomplikowaniem samej procedury uczenia wzajemnego. Jednakże bezpieczeństwo architektury TPCM jest aspektem o wiele ważniejszym niż wydłużenie czasu synchronizacji, gdyż tego typu architektura opiera się najbardziej groźnemu atakowi geometrycznemu, będącym bardzo skutecznym mechanizmem kryptoanalizy TPM.

Zasadne jest pytanie czy dalsze rozszerzanie struktur matematycznych, na których opierają się architektury TPM, pozwoli na istotne zwiększenie poziomu bezpieczeństwa.

Liczby zespolone (wykorzystywane w architekturze TPCM) są rozszerzeniem ciała liczb rzeczywistych. Natomiast rozszerzeniem ciała liczb zespolonych jest struktura algebraiczna zwana kwaternionami. W ściśle matematycznym aspekcie kwaterniony są nieprzemiennym pierścieniem z operacją dzielenia. Pierścień kwaternionów jest zaś czterowymiarową algebrą nad swoim centrum, które jest izomorficzne z liczbami rzeczywistymi. Struktura kwaternionów zajmuje ważne miejsce w algebrze, gdyż zgodnie z twierdzeniem Frobeniusa jest jednym z trzech (obok liczb rzeczywistych i liczb zespolonych) skończenie wymiarowych pierścieni z dzieleniem zawierającym liczby rzeczywistej jako podpierścień.

## 2. Architektura TPQM

Sam algorytm działania TPQM (tree parity quaternion machine) jest analogiczny do poprzednich architektur (TPM i TPCM). Modyfikacji jednak będą wymagały parametry konstrukcyjne, które muszą być dostosowane do specyfiki kwaternionów.

Mianowicie architektura TPQM składa się z dwóch warstw. Pierwszy poziom stanowią perceptrony zwierające $N$ – elementowe wektory wag -([$w_{k,1}$, $w_{k,2}$, ..., $w_{k,N}$], gdzie $1 \le k \le K$), których wartości są kwaternionami. Oczywiście tak jak w poprzednich modelach, wagi powinny podlegać pewnym ograniczeniom. W klasycznym modelu TPM był to przedział [$-L$, $L$], który w naturalny sposób może zostać przeniesiony na grunt kwaternionów. Ograniczenie to będzie, więc przedziałem [$-L$, $L$]x[$-L$, $L$]x[$-L$, $L$]x[$-L$, $L$]. Taki sposób reprezentacji jest możliwy ze względu na fakt, iż pomiędzy kwaternionami a punktami przestrzenie $R^4$ istnieje jednoznaczna odpowiedniość. Wejściem perceptronów pierwszej warstwy jest $K$ $N$ – elementowych wektorów ([$x_{k,1}$, $x_{k,2}$, ..., $x_{k,N}$], gdzie $1 \le k \le K$, które w praktycznej implementacji są reprezentowane jako jeden $N$·$K$ – elementowy wektor [$x_1$, $x_2$, ..., $x_{kN}$]). Wyjścia perceptronów muszą stanowić ze względu na operację mnożenie zbiór zamknięty. Stąd będzie to 8 elementów należących do zbioru {(1, 0, 0, 0), (-1, 0, 0, 0), (0, 1, 0, 0), (0, -1, 0, 0), (0, 0, 1, 0), (0, 0, -1, 0), (0, 0, 0, 1), (0, 0, 0, -1)}. Czyli w innej (kanonicznej) notacji będą to elementy: 1, -1, $i$, $-i$, $j$, $-j$, $k$, $-k$. Wyjścia perceptronów oznaczone będą jako $y_1$, $y_2$, ..., $y_K$. Tak więc wyjście architektury $O$ może być obliczone w sposób analogiczny jak w przypadku architektury TPM:

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

Jak zostało już wspomniane wcześniej kwaterniony wraz z operacjami dodawania i mnożenia (+, *) tworzą pierścień, a zatem wszelkie operacje matematyczne zachowują swoją poprawność i sens tak jak w przypadku liczb rzeczywistych (z wyjątkiem przemienności mnożenia). Istotnej modyfikacji będzie podlegała funkcja znaku. W przypadku architektury TPM, dzieliła ona przestrzeń na dwa rozłączne podzbiory, zaś w architekturze TPCM, na cztery. W odniesieniu do TPQM, podział będzie jeszcze bardziej zaawansowany, a mianowicie będzie dotyczył ośmiu rozłącznych podzbiorów przestrzeni $R^4$.

W przypadku funkcji znaku TPM wykorzystana była klasyczna funkcja znaku. Natomiast definicja funkcji znaku architektury TPCM (w celu zwiększenia jej przejrzystości) wykorzystywała argument liczby zespolonej, który w naturalny sposób mógł być wykorzystany do podziału płaszczyzny na cztery ćwiartki. W przypadku kwaternionów i przestrzeni $R^4$ nie istnieje możliwość przedstawienia tego typu rozwiązań w formie graficznej. Stąd wzór funkcji znaku σ przyjmie tylko czysto matematyczna postać:

$$(2)\quad \sigma(q)=\begin{cases}(1,0,0,0), & a_1=\max(\{a_1,a_2,a_3,a_4\})\wedge a_1\geq 0)\\ (-1,0,0,0), & a_1=\max(\{a_1,a_2,a_3,a_4\})\wedge a_1<0)\\ (0,1,0,0), & a_2=\max(\{a_1,a_2,a_3,a_4\})\wedge a_2\geq 0)\\ (0,-1,0,0), & a_2=\max(\{a_1,a_2,a_3,a_4\})\wedge a_2<0)\\ (0,0,1,0), & a_3=\max(\{a_1,a_2,a_3,a_4\})\wedge a_3\geq 0)\\ (0,0,-1,0), & a_3=\max(\{a_1,a_2,a_3,a_4\})\wedge a_3<0)\\ (0,0,0,1), & a_4=\max(\{a_1,a_2,a_3,a_4\})\wedge a_4\geq 0)\\ (0,0,0,-1), & a_4=\max(\{a_1,a_2,a_3,a_4\})\wedge a_4<0)\end{cases},$$

gdzie $q = (a_1, a_2, a_3, a_4)$.

Uczenie sieci TPQM odbywa się zgodnie z regułą uczenia Hebba. Jednakże w tym przypadku nie możemy wprost wykorzystać ograniczeń nakładanych na wartości wektora wag (wykorzystywanych w architekturze TPM). Jak zostało powiedziane wcześniej, ograniczeniem dla wartości wektora wag architektury TPQM będzie przedział [-*L*, *L*]x[-*L*, *L*]x[-*L*, *L*]x[-*L*, *L*]. Stąd formalnie wzory ograniczające wzrost wartości wag w TPQM przyjmą następująca postać. Mianowicie dla dowolnej współrzędnej *a* liczby *q*:

$$(3)\quad a=\begin{cases}sign(a)L & ,|a|>L\\ a & ,w\ przeciwnym\ przypadku\end{cases}$$

Oczywiście jest to najprostszy model ograniczenia wzrostu wartości wektora wag. Jednakże jak zostało pokazane w przypadku architektury TPCM, możliwe są zmiany kształtu ograniczenia. Jednakże kształty w przestrzeni $R^4$ nie mogą być przedstawione w żadnej formie graficznej. Stąd analiza tego typu metod może być prowadzona, albo na podstawie analogii z kształtami w $R^2$, albo z wykorzystaniem aparatu ściśle matematycznego.

## 3. Analiza bezpieczeństwa protokołu uzgadniania klucza w oparciu o architekturę TPQM

Testy potwierdzające bezpieczeństwo architektury TPQM zostaną przeprowadzone w oparciu o trzy egzemplarze dla każdego modelu. Sieci A i B reprezentować będą dwóch użytkowników, chcących dokonać synchronizacji swoich wektorów wag (uzgodnienia klucza). Trzecia sieć C, będzie oponentem chcącym w sposób nieuprawiony dokonać synchronizacji swojego wektora wag z obserwowanymi sieciami. Czas synchronizacji, liczony będzie ilością kroków uczenia. Wyniki testów prezentuje tabela 1. Dla porównania zostaną też podane czasy synchronizacji architektury TPM.

Tabela 1. Porównanie czasów synchronizacji architektur TPM, TPQM

| Architektura | Czas synchronizacji sieci A i B mierzony w ilości kroków | Czas synchronizacji sieci A i C mierzony w ilości kroków | Stosunek czasów synchronizacji sieci A i B oraz A i C |
|---|---|---|---|
| TPM | 254,6 | 758,8 | 0,33553 |
| TPQM | 3705,2 | 310486,8 | 0,011934 |

Z przedstawiony powyżej wyników widać wyraźnie, że rozszerzenie architektury TPM z wykorzystaniem kwaternionów, zapewnia wyższy poziom bezpieczeństwa. Wyniki przedstawione w tabeli wymagają jeszcze dwóch ważnych komentarzy. Po pierwsze ilość kroków przedstawiona w tabeli jest nie tylko miarą bezpieczeństwa procesu synchronizacji, ale także miarą efektywności. Widać, więc ze rozwiązanie charakteryzujące się większym współczynnikiem bezpieczeństwa wymaga większej liczby kroków. Dodatkowym elementem wpływającym na szybkość wykonywanych operacji jest stopień skomplikowania operacji w przypadku dalszych rozszerzeń ciała liczb rzeczywistych. Podsumowując, za znacznie wyższy poziom bezpieczeństwa płacimy spadkiem efektywności (dłuższym czasem synchronizacji). Druga ważna uwaga dotyczy momentu zakończenia procesu synchronizacji. Mianowicie ważnym elementem systemu jest zakończenie procesu synchronizacji, który również istotnie wpływa na poziom bezpieczeństwa. Aby gwarantować odpowiedni poziom ochrony, proces powinien być zakończony możliwe szybko po osiągnięciu stanu synchronizacji. Oczywiście w przypadku architektur z niższym stosunkiem czasów (podana jako 4 kolumna tabeli 1), mamy dużo większy margines bezpieczeństwa.

## Podsumowanie

W niniejszym artykule została zaproponowana i przeanalizowana modyfikacja architektury TPM, wykorzystująca w swojej konstrukcji kwaterniony. Architektura TPQM charakteryzuje się dużo wyższym poziomem bezpieczeństwa i elastycznością przy wyborze parametrów ograniczających wzrost współczynników wektora wag. Niedostatkiem przedstawionego modelu będzie wydłużenie czasu synchronizacji. Jednakże z uwagi na priorytetowe usytuowanie aspektu bezpieczeństwa, jego efektywność ma dużo mniejsze znaczenie.

## LITERATURA


[1] Kanter I., Kinzel W., Vanstone S.A., Secure exchange of information by synchronization of neural networks, *Europhys. Lett.* 57 (2002), 141-147

[2] Klimov A., Mityaguine A., Shamir A., Analysis of Neural Cryptography, Advances in Cryptology, ASIACRYPT, (2002), 823-828

[3] Płonkowski M., Urbanowicz P., Криптографическое преобразование информации на основе нейросетевых технологии, Труды БГТУ. Серия VI. Минск: БГТУ, (2005), 161-164

[4] Płonkowski M., Analiza funkcji chaosu w funkcjach skrótu opartych na sieciach neuronowych, *Przegląd Elektrotechniczny*,3 (2008), 102-104



***Autorzy:***

*mgr Marcin Płonkowski, prof. dr hab. Paweł Urbanowicz, Katolicki Uniwersytet Lubelski Jana Pawła II, Instytut Matematyki, Katedra Systemów Operacyjnych i Sieciowych, ul. Konstantynów 1H, 20-708 Lublin, E-mail: marcin.plonkowski@kul.lublin.pl,*

*mgr Elena Lisica, Białoruski Państwowy Uniwersytet Technologiczny, ul. Swerdłowa 14a, 220000 Mińsk*