Załącznik nr 3

do wniosku o przeprowadzenie postępowania habilitacyjnego

Autoreferat w języku polskim

1. Imię i nazwisko: dr Agnieszka Jakóbik

2. Posiadane dyplomy, stopnie naukowe/ artystyczne - z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej.

2007, **stopień doktora nauk technicznych**, dyscyplina mechanika, Politechnika Krakowska, tytuł pracy doktorskiej: *Analiza wybranych zagadnień mechaniki konstrukcji i materiałów za pomocą sztucznych sieci neuronowych i filtrów Kalmana*, promotor w przewodzie doktorskim Prof. dr hab. inż. Zenon Waszczyszyn
2003, **dyplom magisterski, matematyka,** Uniwersytet Jagielloński, Kraków, praca magisterska z zastosowania szeregów czasowych w prognozowaniu danych, promotor dr hab. Leon Antoni Dawidowicz

3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych.

Od 2008, **adiunkt**, Instytut Informatyki, Wydział Fizyki, Matematyki i Informatyki, Politechnika Krakowska
2007-2008, **asystent naukowo-dydaktyczny**, Instytut Informatyki, Uniwersytet Jagielloński, Kraków,
2003-2007, zajęcia dydaktyczne prowadzone na PK w ramach **Studium Doktoranckiego Politechniki Krakowskiej**,

4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. nr 65, poz. 595 ze zm.):

a. Tytuł osiągnięcia naukowego:

Rozwój wybranych algorytmów poprawy poziomu bezpieczeństwa przetwarzania zadań i energooszczędności środowisk obliczeniowych opartych na szeregowaniu zadań.

b. Publikacje w recenzowanych czasopismach i materiałach konferencyjnych dotyczące bezpośrednio osiągnięcia:

1. *Stackelberg games for modeling defense scenarios against cloud security threats.* Agnieszka Jakóbik, Francesco Palmieri, Joanna Kołodziej. Network and Computer Applications 110: 99-107 (2018), Elsevier, IF **3.991**

3. *Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security.* Daniel Grzonka, Agnieszka Jakóbik, Joanna Kołodziej, Sabri Pllana, Future Generation Comp. Syst. 86: 1106-1117 (2018), Elsevier, IF **4.639**

4. *Non-deterministic security driven meta scheduler for distributed cloud organizations. Simulation Modelling Practice and Theory,* Agnieszka Jakóbik, Daniel Grzonka, Francesco Palmieri: 76: 67-81 (2017), Elsevier, **IF 2.092**

5. *Security supportive energy-aware scheduling and energy policies for cloud environments. Damián Fernández-Cerero,* Agnieszka Jakóbik, Daniel Grzonka, Joanna Kołodziej, Alejandro Fernández-Montes: J. Parallel Distrib. Comput. 119: 191-202 (2018), Elsevier, **IF 1.815**

6. *SCORE: Simulator for cloud optimization of resources and energy consumption.* Damián Fernández-Cerero, Alejandro Fernández-Montes, Agnieszka Jakóbik, Joanna Kołodziej, Miguel Toro, Simulation Modelling Practice and Theory 82: 160-173 (2018), Elsevier, **IF 2.092**

7. *GAME-SCORE: Game-based energy-aware cloud scheduler and simulator for computational clouds,* Damián Fernández Cerero; Agnieszka Jakóbik; Alejandro Fernández-Montes; Joanna Kołodziej, Journal: Simulation Modelling Practice and Theory 2018, https://doi.org/10.1016/j.simpat.2018.09.001, Elsevier, **IF 2.092**

c) omówienie celu naukowego ww. pracy/prac i osiągniętych wyników wraz z omówieniem ich obecnego i planowanego wykorzystania.

Motywacja:

Celem podjętej pracy naukowej jest rozwój wybranych algorytmów poprawy poziomu bezpieczeństwa przetwarzania zadań i energooszczędności środowisk obliczeniowych, opartych o szeregowanie zadań, w szczególności systemów typu Chmury Obliczeniowe (Computational Clouds).

Problem ten jest jednym z bardziej aktualnych problemów współczesnego społeczeństwa informacyjnego. Systemy te od kilku lat są w powszechnym użyciu, obejmują coraz większe obszary zastosowań [35, 36]. Ze względu na ich złożoność, wykorzystywaną ideę wirtualizacji, a także elastyczną możliwość konfiguracji środowiska obliczeniowego, istnieje wiele (często niezidentyfikowanych) zagrożeń dotyczących bezpieczeństwa elementów oraz procesów zachodzących w takich systemach (w szczególności procesów przesyłania oraz przetwarzania danych) [29, 30, 31, 32, 33, 34].
Przy tak dużej popularności, znaczącej mocy obliczeniowej oraz dostępności usług przez całą dobę, drugim ważnym problemem w środowiskach tego typu jest optymalizacja energii zużywanej przez te systemy [37].

Prezentowany cykl artykułów koncentruje się na tych dwóch aspektach, w odniesieniu do systemów chmurowych, które wyposażone są w mechanizmy szeregowania zadań. Mechanizmy te (*ang. Task Schedulers*) umożliwiają przydział konkretnych zadań obliczeniowych do określonych jednostek obliczeniowych, dostępnych w ramach środowiska, wg. założonych z góry kryteriów. Użytkownik ma do dyspozycji wbudowane algorytmy szeregujące, oraz możliwość implementacji własnych rozwiązań w tym zakresie. Dostępne są one w każdym dużym systemie tego typu, np. Amazon [38], Rackspace [39], Google Cloud [40], IBM Cloud [41] i wielu innych. Środowiska obliczeniowe, oparte o szeregowanie zadań, stanowią więc większość spośród działających obecnie systemów chmurowych.

Ze względu na dużą liczbę użytkowników, zasobów oraz zadań, najbardziej efektywne algorytmy, to te które, umożliwiają podejmowanie kluczowych decyzji w sposób zautomatyzowany [42]. Do takich rozwiązań należą w szczególności usługi automatycznego skalowania zdolności obliczeniowej oraz

innych, wybranych parametrów jednostek obliczeniowych [43, 44]. Aby usługi te zostały efektywnie zrealizowane niezbędne są odpowiednie algorytmy, wspierające proces podejmowania decyzji.

Cele naukowe:

Celem naukowym było zaproponowanie algorytmów, umożliwiających podejmowanie decyzji podnoszących poziom bezpieczeństwa zwiększając oszczędność energetyczną przetwarzania danych w systemach chmurowych w elastyczny, automatyczny sposób, w zależności od aktualnego obciążenia systemu zadaniami oraz wobec najczęściej identyfikowanych zagrożeń, zmieniających się w czasie.

Realizację tego celu podzielić można na rozwiązania wykorzystywane w ramach następujących obszarów:
1. wykorzystywane podczas zbierania zadań,
2. używane podczas ich szeregowania oraz przetwarzania,
3. bezpiecznego przechowywania wyników obliczeń,
4. algorytmy doboru metod bezpośredniej ochrony środowiska obliczeniowego, niezależne od aktualnej fazy, w jakiej znajduje się dany pakiet zadań.

Przykładem pierwszego z rozwiązań może być zastosowanie szczególnych protokołów kryptograficznych, podczas wykonywania zadań które wiążą się z przetwarzaniem danych wrażliwych, np. zadań dotyczących analizy danych medycznych konkretnych pacjentów. Przykładem drugiego rozwiązania jest zastosowanie mechanizmu, który w razie wykrycia ataku na maszynę wirtualną, przenosi obliczenia na inną maszynę - zaatakowaną maszynę wyłącza, a następnie odtwarza obraz maszyny sprzed ataku.

Krótkie omówienie tematyki poszczególnych prac:

Dwie pierwsze prace z cyklu prezentują modele automatyzacji podejmowania decyzji, dotyczących wyboru konkretnych metod zabezpieczania systemów chmurowych przed atakiem. Metody te opierają się na założeniu, że zarówno obrońca, jak też atakujący posiada własne, jasno zdefiniowane cele i postępują tak, by zmaksymalizować założony przez siebie zysk (lub zminimalizować straty).

Praca [1] prezentuje zastosowanie prostszego modelu w którym funkcja wypłaty dla atakującego ma postać tabeli rankingowej, obrazującej atrakcyjność atakowanych celów. Zaproponowany model to gra o sumie niezerowej. Decyzje obrońcy podejmowane są w odpowiedzi na akcje atakującego, gra rozgrywana jest wielokrotnie. Przedstawiony model pozwala na automatyczny wybór jednej ze skonczonej ilosci, z góry zadanych akcji, których prawdopodobieństwo wyboru określone jest poprzez strategię gry.

Model zakłada dwa etapy gry: etap pasywny, kiedy zbierane są statystyczne dane, na podstawie których układana jest tabela atrakcyjności celów ataku, oraz etap aktywny. Podczas fazy aktywnej model funkcjonuje w trybie predyktor - korektor. Najpierw przewidywana jest przyszła aktywność atakującego, w celu zastosowania odpowiednich środków ochrony, następnie, jest ona korygowana na podstawie wystąpienia rzeczywistych ataków. W modelu przyjęto założenie racjonalności postępowania obu graczy, tzn. założenie, że faktycznie postępują oni tak, by zmaksymalizować lub zminimalizować przyjęte funkcje celu. W artykule przedstawiono wyniki symulacji ataku, który przeprowadzono wg. listy 7 największych zagrożeń bezpieczeństwa dla systemów chmurowych, opracowanych przez Security Alliance (por. [26]). Są to:

- wstrzyknięcie nieautoryzowanego zadania;

- atak typu blokada usługi (DoS), w postaci wysłania bardzo dużej liczby zadań, które powodują wydłużenie czasu harmonogramowania ich na tyle, aby moduł harmonogramujacy działał zbyt długo, aby system mógł rozpocząć działanie;
- atak z wewnątrz systemu, nieautoryzowana modyfikacja zadania podczas przesyłania z jednej części systemu chmurowego do innej;
- rozproszony atak DDoS w postaci zbyt wielu żądań przyjęcia zadania do kolejnego pakietu, powodujący niezdolność elementu gromadzącego zadania do prawidłowego funkcjonowania;
- utrata danych w postaci zaginięcia zadania;
- atak długofalowy na zasoby energetyczne systemu w postaci rosnącej liczby i częstotliwości zgłaszanych do przetworzenia zadań;
- atak nieznanego typu na wybraną maszynę wirtualną w postaci zablokowania możliwości obliczeniowej.

Strategia obrony przewidywała zastosowanie w każdej turze gry jednego z 6 środków obronnych, spośród umieszczonych w wytycznych Cloud Controls Matrix [27]:
- zastosowanie podpisu cyfrowego do weryfikacji integralności zadania oraz autentykacji nadawcy zadania;
- zastosowanie dodatkowego procesu działającego w tle w formie antywirusa;
- zastosowanie dodatkowego procesu działającego w tle w formie zapory ogniowej (firewall)
- tzw. strategie ucieczki (ang. Escaping Routines), polegające na każdorazowym zamknięciu zaatakowanej maszyny wirtualnej i odtworzeniu jej z zachowanego obrazu;
- weryfikacja integralności zadań z użyciem funkcji skrótu SHA-2 w zastosowaniu do całego pakietu zadań;
- odcięcie energetyczne - skalowanie możliwości obliczeniowych maszyny wirtualnej na niższe, w razie przekroczenia określonej z góry ilości pobieranej przez nią energii (ang. Energy Cupping).

W zaproponowanym modelu uwzględniono koszty zastosowania każdego ze środków w postaci wydatku energii skonsumowanej przez każdy z nich. Założono także, że w przypadku skutecznego ataku, stosowne zadania obliczeniowe trzeba procesować jeszcze raz, pomimo wydatku energetycznego na ochronę. Przykładem może być sytuacja, gdy cała pula zadań w pakiecie jest uszkodzona. Energia zużyta do obliczeń została zmarnowana, a zadania należy przetworzyć ponownie. Jeśli atak był nieskuteczny, koszt energetyczny jest pomniejszany tylko o wydatek na ochronę systemu.

Funkcję wypłaty dla atakującego modelowano w postaci punktów, których wartości były wprost proporcjonalne do zdolności obliczeniowych atakowanej jednostki.

Pracy [3] rozszerzono zaproponowany w [8] model agentowy, wzbogacając go o możliwość weryfikowania integralności zadań, poprzez użycie funkcji skrótu oraz sieć neuronową, wykrywającą nieprawidłowości w przesyłanych do systemu zadaniach.
Dodatkowym rozszerzeniem możliwości systemu było umożliwienie przygotowywania harmonogramów z uwzględnieniem podziału zadań, oraz jednostek obliczeniowych na grupy, zależnie od wymaganego poziomu (dostępne 4 możliwe) zabezpieczeń kryptograficznych jednostki, na której zadania zostaną uruchomione. W ramach pracy zaimplementowano ewolucyjny algorytm do wyznaczania harmonogramów z takimi dodatkowymi ograniczeniami. 4 możliwe poziomy oferowanych usług w zakresie bezpieczeństwa zostały dostosowane do wymogów międzynarodowych standardów FIPS [22] oraz ISO/IEC 19790 Standard of security requirements for cryptographic module [23].

W pracy [4] zaprezentowano uwzględnienie podczas harmonogramowania - nakładu obliczeniowego, niezbędnego do wykonania dodatkowych operacji kryptograficznych. Podejście to pozwala na planowanie dobory zastosowanych algorytmów i procedur kryptograficznych, zwiększających

bezpieczeństwo przetwarzania zadań. Przykładowe z nich, to szyfrowanie wyników, otrzymanych podczas procesu przetwarzania zadania, weryfikacja tożsamości jednostki odbierającej wyniki, poprzez zastosowanie procedury podpisu cyfrowego, czy dedykowanych algorytmów autentykacji. Do testów wykorzystano najczęściej używane procedury, takie jak: SHA-1, SHA-256, SHA-512, procedurę szyfrowania RSA z kluczem o długości 1024 bitów (RSA-1024), RSA-2048 RSA-2048, podpis cyfrowy: RSA DS/SHA: 1024/256 1024/512 2048/512.
Takie podejście pozwala także na wyznaczanie harmonogramów uwzględniających nakład obliczeniowy na te dodatkowe operacje, oraz monitorowania wpływu określonych usług kryptograficznych na szybkość przetwarzania pakietów zadań. Stanowi także podstawę do konstrukcji algorytmów automatyzujących dobór ochrony kryptograficznej w sposób automatyczny zob. [9].

W pracy [5] zaprezentowano testy 4 scenariuszy harmonogramowania zadań, z uwzględnieniem dodatkowego kryterium minimalizacji energii:
1.Harmonogramowanie początkowo wg. kryterium, najkrótszego czasu wykonania całego pakietu zadań. Kryterium minimalizacji energii jest stosowane dopiero w sytuacji wyboru między dwoma harmonogramami o tym samym czasie wykonania zadań.
2.Harmonogramowanie początkowo wg. kryterium, najmniejszego wydatku energetycznego na realizację całego pakietu zadań. Kryterium minimalizacji czasu wykonania jest stosowane dopiero w sytuacji wyboru między dwoma harmonogramami równoważnymi energetycznie.
3.Scenariusz przewidujący harmonogramowanie z ograniczeniem czasowym. Nowe rozwiązania są generowane tak długo, aż znaleziony jest harmonogram spełniający zadane ograniczenie czasowe. Z uwagi na przyjętą reprezentację harmonogramu, populacja rozwiązań jest następnie przeszukiwana w celu znalezienia harmonogramu, którego wykonanie wiąże się z najmniejszym wydatkiem energetycznym.
4.Scenariusz przewidujący harmonogramowanie z ograniczeniem energetycznym. Nowe rozwiązania są generowane tak długo, aż znaleziony jest harmonogram spełniający zadane ograniczenie energetyczne. Następnie otrzymana populacja rozwiązań jest przeszukiwana w celu znalezienia harmonogramu, którego wykonanie jest najszybsze.

Modelem referencyjnym jest harmonogramowanie oparte o kryterium najkrótszego czasu wykonania pakietu zadań.

Scenariusze te testowano, uwzględniając realistyczne obciążenie systemu obliczeniowego w postaci tzw. Google Traces. Przetestowano zarówno przypadek harmonogramowania zadań w trybie pakietowym, jak też przy dodatkowym założeniu wykonywania przez system zadań także bez czasu zakończenia (tzw. ang. Service Jobs). Dodatkowo przetestowano również skuteczność użycia dwóch wybranych polityk hibernacji maszyn wirtualnych, w celu dodatkowego oszczędzania energii w czasie oczekiwania na zadania z kolejnego pakietu. Rozważano przypadki pojedynczej jednostki harmonogramującej, oraz kilku działających równolegle.

Przedstawione wyniki otrzymano na podstawie symulacji obciążenia systemu chmurowego przez 7 dni z rzędu, dla systemu obliczeniowego przetwarzającego rzeczywiste obciążenie udostępnione przez Google, składającego się z 1000 maszyn wirtualnych. Otrzymane rezultaty potwierdzają skuteczność zaproponowanych metod harmonogramowania. Symulacje pozwalają także na wybór efektywnej polityki energetycznej maszyn wirtualnych w zakresie hibernacji.

W pracy [6] przedstawiono rozwój symulatora do testowania metod harmonogramowania oraz doboru polityk energetycznych maszyn wirtualnych, które zostało zbudowane na bazie doświadczeń pracy [5]. Wszystkie omawiane powyżej algorytmy weszły w skład prezentowanego symulatora. Symulator

został udostępniony jako oprogramowanie open source, https://github.com/DamianUS, do użytku przez badaczy zajmujących się podobnymi zagadnieniami.

Praca [7] prezentuje, rozszerzoną o moduł przeprowadzania gier Stackelberga, wersję symulatora opisanego w pracy [6]. W ramach symulatora zaimplementowano metody oparte o gry Stackelberga do automatyzacji doboru polityk energetycznych maszyn wirtualnych, por. [9], [11]. Zaproponowano grę wieloetapową pomiędzy jednostką generującą harmonogramy, a menedżerem wydajności energetycznej systemu chmurowego. W celu przeprowadzenia gry zaproponowano stosowne funkcje wypłat dla graczy oraz kolejność podejmowania decyzji w grze. W zaproponowanym modelu: lider gry wybiera harmonogram, który spełnia założone przez niego kryterium, gracz drugi, w odpowiedzi na przesłany harmonogram, decyduje o wyborze jednej z kilku polityk energetycznych. W pracy przedstawiono wyniki dwóch symulacji, w których gracz drugi ma do wyboru dwie spośród możliwych decyzji.
Na potrzeby prezentacji możliwości narzędzia, zaimplementowano kilka bardziej zaawansowanych testów, przedstawiono ich rezultaty oraz podsumowanie. Podobnie jak w przypadku poprzedniej wersji oprogramowania, jest ono dostępne dla środowiska naukowego w postaci archiwum GitHub: https://github.com/DamianUS/game-score.

Mój wkład w każdą z tych prac z osobna, w kontekście wkładu współautorów, został przedstawiony w załączniku nr. 8.

Krótkie omówienie pozostałych opublikowanych prac, w kontekście cyklu:

Celem pracy [8] była prezentacja możliwości zwiększenia poziomu bezpieczeństwa formowania i szeregowania pakietów zadań. Prezentowane rozwiązanie pomaga przeciwdziałać praktyce wstrzykiwania zadań. Na atak tego typu mogą być narażone środowiska chmurowe, w których operacje wykonywane są w ściśle określonych momentach czasowych. Przykładowo usługa monitorująca Amazon CloudWatch, jednego z głównych dostawców usług chmurowych, udostępnia możliwość sprawdzania wybranych cech środowiska obliczeniowego co 60, 300, 3600 sek., w równych odstępach czasowych, natomiast The CloudWatch Logs Agent zbiera i wysyła dane z rekordów monitoringu co 5 sekund [21]. Wstrzykiwanie zadania jest to nieautoryzowane umieszczenie zadania w pakiecie zadań do wykonania oraz wykonanie je, zwykle na koszt użytkownika, któremu zostało wstrzyknięte [17]. Niesie to za sobą jego straty finansowe, gdyż płatność za usługi obliczeniowe wzrasta oraz marnowana jest energia.
W pracy zaproponowano generowanie losowych momentów czasowych opartych o dwa różne mechanizmy generujących ciągi bitów. Pierwszy z nich oparty był na skrótach binarnych, otrzymywanych z procedury SHA-2, rekomendowanej przez National Security Agency (NSA) [20]. Drugi, na generatorze liczb pseudolosowych Blum-Blum-Shub [18], którego własności są dobrze udokumentowane [19]. Takie podejście utrudnia bardzo wyznaczenie momentu w którym nastąpi koleje sprawdzenia zbioru zadań, w celu wykrycia ewentualnych zadań wstrzykiwanych, lub innych nieprawidłowości.
Dodatkowo, wraz z możliwością skracania podstawowej jednostki czasu, która otrzymują znacznik "0" lub "1", otrzymujemy narzędzie do zagęszczania ilości pomiarów w czasie.
Celem przeprowadzonej w pracy analizy numerycznej było także użycie ww. algorytmów do wyznaczania czasu generowania harmonogramu oraz wykrycie ewentualnych opóźnień lub wąskich gardeł procedury. Dodatkowo, sprawdzano wpływ nierównych momentów rozpoczynania harmonogramowania na płynność przepływu pakietów zadań.

W ramach pracy [8] zaimplementowano także prosty system agentowy, do kontroli opisanej w pracy symulacji.

W pracy [9] zaproponowano algorytm automatyzacji doboru poziomów zaufania kryptograficznego dla maszyn wirtualnych oraz ich pojemności obliczeniowej. W celu realizacji tego zadania, zaprojektowano grę Stackelberga z procedurą dodatkowej gry wewnętrznej, rozgrywanej na każdym etapie gry głównej.
Algorytm zaprojektowano uwzględniając możliwość skalowania pojemności maszyn wirtualnych oraz biorąc pod uwagę kryterium kosztu czasu pracy jednostek obliczeniowych.
Praca ta została poprzedzona pracą przeglądową [13], dotyczącą gier Stackelberga oraz ich wykorzystania przy wyznaczaniu strategii postępowania podczas planowania procedur bezpieczeństwa obiektów, takich jak lotniska.

W kolejnej pracy [11], przedstawiono rozszerzenie modelu harmonogramowania o kryterium minimalizacji energii.

W pracy [12] przeanalizowano dwa algorytmy wspomagające bezpieczne przetwarzanie zadań, które można wykorzystać do budowy mniej typowych systemów kryptograficznych, niż wspomniane wyżej. W pracy przedstawiono także aspekty użycia tych procedur w systemach wymagających obsługi zadań dla dużej liczby użytkowników. Do analizy wybrano ślepe szyfrowanie algorytmem RSA oraz weryfikację integralności zadań z wykorzystaniem procedury dzielenia sekretu. Przeanalizowane algorytmy zostały wybrane pod kątem możliwości przeprowadzenia weryfikacji poprawności obliczeń zespołowych na całym pakiecie zadań bez ujawniania zawartości przesyłanych danych. W pracy przeprowadzono analizę obciążenia systemu chmurowego podczas użycia obu algorytmów, umożliwiającą wybór pomiędzy różnymi parametrami obu algorytmów, w zależności od priorytetu: czasu wykonywania obliczeń lub bezpieczeństwa kryptograficznego.

Praca przeglądowa [14] stanowi wstęp do pracy [1]. Prezentuje wybrane zagadnienia dotyczące zapewnienia bezpieczeństwa w systemach typu Big Data, czyli dla systemów w których przetwarzane są bardzo duże ilości danych, napływających bardzo szybko, o różnej strukturze wewnętrznej. Taki sposób przetwarzania danych jest najczęściej spotykanym w obrębie środowisk chmur obliczeniowych.

Praca przeglądowa [16], podsumowuje podstawowe pojęcia związane z zarządzaniem energią w systemach chmurowych.

Prace [24] oraz [25] opisują efekty mojej współpracy z zespołem naukowym w składzie: Marco Gribaudo i Mauro Iacono, który zajmuje się projektowaniem dedykowanych systemów chmurowych. Przedstawione w omawianych pracach rozwiązania spotkały się z zainteresowaniem ze strony tego zespołu. Algorytm do harmonogramowania został przetestowany w symulacji CloudSim przedstawionej w pracy [24]. W pracy [25] opisano wieloelementowy system chmurowy do przetwarzania danych podczas interwencji służb specjalnych (policja, straż pożarna). Mój wkład w omawiany system dotyczył optymalizacji wersji zaprezentowanej w przytoczonej publikacji.

Obecnie prowadzone prace badawcze:

- zastosowanie ww. algorytmów do podniesienia bezpieczeństwa przetwarzania zadań w systemie opisanym w pracy [25],

- testy algorytmu do dynamicznego doboru algorytmów kryptograficznych w ramach oferowanych poziomów zaufania elementów systemu chmurowego, oparte o lokalne harmonogramowanie, wykorzystujące czas bezczynności maszyn wirtualnych (automatyczny dobór parametru b, występującego w modelu opisanego np. w pracy [10]),

- opracowywanie pracy omawiającej testy dedykowanej funkcji skrótu, o zmiennej długości generowanego skrótu, jako elementu elastycznego systemu podnoszącego bezpieczeństwo przetwarzania zadań,

- metody opisane w pracy [1] spotkały się z zainteresowaniem, ze strony Cloud Italia. oraz Prof. Francesco Palmieri, który zajmuje problematyką sie min. modelowania długofalowych ataków na Chmury obliczeniowe. Do przetestowania ww. algorytmów na rzeczywistym systemie chmurowym zbierane są obecnie odpowiednie dane.

Efekty końcowe i osiągnięcia:

Zaadaptowanie metodologii modelowania strategii wobec zagrożeń bezpieczeństwa obiektów takich jak lotniska, na przypadek zagrożeń cyberbezpieczeństwa.

Opracowanie metodologii przewidywania możliwej strategii atakującego, wyboru strategii obrońcy, adaptacja funkcji wypłat do przypadku ataku środowiska chmurowego.

Zaproponowanie modelu, w którym doszło do złamania zasad gry, oraz przecieków informacji pomiędzy poszczególnymi rundami gry. Opracowanie metodologii wyznaczania strategii atakującego bez znajomości możliwej funkcji wypłaty, przy wykorzystaniu sztucznych sieci neuronowych.

Innowacyjna metoda wyznaczenia kolejnych momentów czasowych, monitorowania i harmonogramowania.

Innowacyjna metoda dynamicznego doboru poziomu usług kryptograficznych przez maszyny wirtualne, w zależności od wymagań określonych przez zadania obliczeniowe.

Opracowanie algorytmu harmonogramowania zadań przetwarzanych w trybie pakietowym, z uwzględnieniem zapotrzebowania na operacje kryptograficzne. Umożliwiającego harmonogramowanie wg. wskazanego poziomu usług, uwzględniające kryterium jak najkrótszego czasu wykonania zadań oraz jak najmniejszej ilości zużytej energii. Opracowanie algorytmu uwzględniającego 4 kryteria mieszane, w sytuacji gdy priorytetem jest jedno z kryteriów, lecz poszukujemy również rozwiązania spełniającego częściowo drugie z kryteriów.

Opracowanie algorytmu do automatyzacji doboru polityk energetycznych maszyn wirtualnych w dużych centrach obliczeniowych.

Opracowanie metodologii testowania własności funkcji skrótu, w odniesieniu do certyfikowanych przez instytucje międzynarodowe funkcji skrótu, jako narzędzia do skutecznej weryfikacji nowo budowanych funkcji skrótu.
Opracowanie metodologii testowania wpływu wykorzystanych algorytmów kryptograficznych na czas przetwarzania całych pakietów zadań, z uwzględnieniem procesu ich wcześniejszego harmonogramowania.

Opracowanie metodologii skalowania zdolności obliczeniowych maszyn wirtualnych, na podstawie znajomości przydzielonych im przez moduł harmonogramowania zadań.

Znacząca modyfikacja symulatora środowisk chmurowych Cloudsim oraz SCORE I GAME-SCORE, uwzględniająca możliwość przeprowadzenia zaawansowanych symulacji oraz testowanie przedstawionych algorytmów.

W wyniku przeprowadzonych analiz, testów i eksperymentów numerycznych został zbudowany spójny system który uwzględnia:

- moduł managera gier, który obsługuje gry Stackelberga, w zakresie definicji funkcji wypłat dla graczy, definicji możliwych strategii, oraz rozwiązania numerycznego stosownych problemów optymalizacyjnych, którego kopie można umieścić w jednostce harmonogramującej, instancji maszyny wirtualnej, lub centralnym systemie zarządzania środowiskiem chmurowym.
- moduł managera polityk energetycznych, którego kopie można umieścić w instancji maszyny wirtualnej, centralnym systemie zarządzania środowiskiem chmurowym, który odpowiada za stosowne skalowanie maszyn wirtualnych oraz narzucanie polityk energetycznych dotyczących np. hibernacji maszyn wirtualnych,
- moduł szeregujący zadania o znanym zapotrzebowaniu obliczeniowym i określonym zapotrzebowaniu na usługi kryptograficzne, oparty o algorytm ewolucyjny,
- moduł do analizy raportów bezpieczeństwa przy pomocy sztucznych sieci neuronowych,
- system agentowy nadzorujący przeprowadzane operacje, oraz zbierający dane niezbędne do poprawnego zastosowania zaproponowanych algorytmów.

Zaproponowane algorytmy zostały przetestowane numerycznie w szeregu symulacji. Efekty ich zastosowania omówiono w każdej z prac, zawierających liczne przykłady numeryczne oraz przykłady możliwych zastosowań w procesach konfiguracji środowisk chmurowych.

Uzyskane wyniki i wykorzystanie w praktyce:

Przedstawione osiągnięcie składa się z cyklu publikacji, w których zaprezentowano kolejne etapy rozwoju algorytmów służących automatyzacji podejmowania decyzji zwiększenia bezpieczeństwa oraz energooszczędności środowisk chmurowych.

Zaproponowane algorytmy zostały wykorzystane do testowania rozwijanych przez zespół: D. Fernández-Cerero i A. Fernández-Montes, polityk energetycznych, projektowanych do zastosowania w dużych centrach obliczeniowych.

Zaproponowane algorytmy zostały wykorzystane także do harmonogramowania zadań opisanych w systemie budowanym pod kierunkiem: M. Iacono.

Wszystkie przedstawione rozwiązania mają zastosowanie praktyczne, gdyż zaprezentowane rozwiązania powstały w odpowiedzi na analizę konkretnych problemów, powstających podczas przetwarzania zadań w rozważanych środowiskach.

Wnioski płynące z przeprowadzonych eksperymentów numerycznych stanowią wartościowy punkt odniesienia dla rozwoju metod automatycznego podejmowania decyzji, wspierających podniesienie bezpieczeństwa oraz energooszczędności środowisk, wyposażonych w moduły do harmonogramowania.

Symulator SCORE, oraz GAME-SCORE jest gotowym do użycia, przez badaczy tematu oraz osoby konfigurujące środowisko typu Cloud, narzędziem.

Szczegóły implementacyjne :

W ramach przedstawionych prac, powstały następujące pakiety oprogramowania: m-funkcje w środowisku Matlab - do prototypowania algorytmów, funkcje dołączane do symulatora środowisk chmurowych CloudSim - zaimplementowane w języku Java, oprogramowanie w języku C++ oparte o framework FastFlow, moduły symulatora SCORE oraz GAME-SCORE - zaimplementowane w

języku Scala. Oprogramowanie to jako najbardziej kompletne i gotowe do użycia dostępne jest na licencji Open Source.

Wpływ na dyscyplinę naukową - opracowanie tytułowej metodyki obliczeń stworzyło podstawy do rozwoju dyscypliny informatyka w następujących kierunkach:

- Algorytmy metod automatyzacji podejmowania decyzji podnoszących poziom bezpieczeństwa przetwarzania zadań w środowiskach obliczeniowych Cloud, w tym:
- algorytmy doboru środków obrony przed atakiem na cyberbezpieczeństwo;
- algorytmy doboru procedur kryptograficznych stosowanych do zadań harmonogramowanych i przetwarzanych w trybie pakietowym.
- Rozwój metodologii przeprowadzania symulacji numerycznych środowisk chmurowych.
- Wspomaganie procesu konstrukcji rozwiązań dotyczących omawianych problemów, poprzez dostarczenie odpowiednich narzędzi do testowania nowo proponowanych metod podnoszenia bezpieczeństwa oraz energooszczędności środowisk chmurowych.

Bibliografia:

8. *Towards secure Non-Deterministic Meta-Scheduling for clouds*, Agnieszka Jakóbik, Daniel Grzonka, Joanna Kołodziej, Horacio González-Vélez, ECMS 2016, pp. 596-602 , http://www.scs-europe.net/dlib/2016/ecms2016acceptedpapers/0596-dis_ECMS_0084.pdf
9. *Using Polymatrix Extensive Stackelberg Games in Security-Aware Resource Allocation and Task Scheduling in Computational Clouds*, Agnieszka Jakóbik, Andrzej Wilczynski, Journal of Telecommunications and Information Technology, 71-80, 2017
10. *Security supportive energy aware scheduling and scaling for Cloud environments,* Agnieszka Jakóbik, Daniel Grzonka, Joanna Kołodziej, Proceedings 31st European Conference on Modelling and Simulation, ECMS 2017, pp. 583-590
11. *Stackelberg Game-Based Models In Energy-Aware Cloud Scheduling.* Damián Fernández-Cerero, Alejandro Fernández-Montes, Agnieszka Jakóbik, Joanna Kołodziej, ECMS 2018: 460-467, https://doi.org/10.7148/2018-0460
12. *Analysis of Selected Cryptographic Services for Processing Batch Tasks in Cloud Computing Systems*, Agnieszka Jakóbik, Jacek Tchórzewski, Modeling and Simulation in HPC and Cloud Systems. Studies in Big Data, Springer, vol. 36, pp. 135-155, 2018
13. *Stackelberg security games: models, applications and computational aspects,* Andrzej Wilczyński, Agnieszka Jakóbik, Joanna Kołodziej , Journal of Telecommunications and Information Technology : JTIT – 2016, 3, s. 70-79 http://www.itl.waw.pl/czasopisma/JTIT/2016/3/70.pdf
14. *Big data security,* Agnieszka Jakóbik, Resource management for big data platforms: algorithms, modelling, and high-performance computing techniques , eds. Florin Pop, Joanna Kołodziej, Beniamino Di Martino: Springer, 2016, pp. 241-261
15. *Theoretical and Experimental Analysis of Cryptographic Hash Functions,* Jacek, Tchórzewski, Agnieszka Jakóbik, Journal of Telecommunications and Information Technology (w druku)
16. *Energy Efficient Scheduling Methods for Computational Grids and Clouds*, A. Jakóbik, D. Grzonka, J. Kołodziej, A.E. Chis, H. González-Vélez, Journal of Telecommunications and Information Technology, 56, 2017
17 F. Zhou , M. Goel , P. Desnoyers , R. Sundaram, *Scheduler vulnerabilities and coordinated attacks in cloud computing*, J. Comput. Secur. 21 (4) (2013) , 533–559
18 L. Blum, M. Blum, and M. Shub, *A Simple Unpredictable Pseudo-Random Number Generator*, SIAM Journal on Computing, vol. 15, p. 364-383, May 1986
19. Cryptography and Coding: 10th IMA International Conference, Lecture Notes in Computer Science 3796 (2005) 355–375. Springer-Verlag. *Concrete Security of the Blum-Blum-Shub Pseudorandom Generator,* Andrey Sidorenko and Berry Schoenmakers
20. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
21. https://aws.amazon.com/cloudwatch
22. Security Requirements for Cryptographic Modules. FIPS PUB 140–2, Technical Report, 2001.
23. ISO/IEC 19790 Security requirements for cryptographic modules (2012).
24. *Exploiting CloudSim in a multiformalism modeling approach for cloud based systems*, Enrico Barbierato, Marco Gribaudo, Mauro Iacono, Agnieszka Jakóbik, Journal: Simulation Modelling Practice and Theory 2018, https://doi.org/10.1016/j.simpat.2018.09.018, Elsevier,
25. *Performance Optimisation Of Edge Computing Homeland Security Support Applications*, Marco Gribaudo, Mauro Iacono, Agnieszka Jakóbik, Joanna Kołodziej, ECMS 2018: 440-446, https://doi.org/10.7148/2018-0440,
26. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf
27. https://cloudsecurityalliance.org › Research › Cloud Controls Matrix
28. https://www.first.org/cvss/user-guide

29. https://downloads.cloudsecurityalliance.org/initiatives/ccm/CSA_CCM_v3.0.xlsx
30. https://www.owasp.org/index.php/Category:OWASP_Cloud_□_10_Project
31.The Security Guidance for Critical Areas of Focus in Cloud Computing v4.0 ("Guidance v4.0") is licensed by the Cloud Security Alliance under a Creative Commons Attribution-NonCommercial- ShareAlike 4.0 International License (CC-BY-NC-SA 4.0)., https://cloudsecurityalliance.org/artifacts/security-guidance-v4/
32. https://www.ibm.com/blogs/cloud-computing/2016/04/01/12-biggest-cloud-computing-security-threats/
33.https://www.enisa.europa.eu/topics/cloud-and-big-data?tab=articles
34.https://www.nist.gov/sites/default/files/documents/itl/cloud/NIST_SP-500-291_Version-2_2013_June18_FINAL.pdf
35. https://www.forbes.com/sites/louiscolumbus/2018/08/30/state-of-enterprise-cloud-computing-2018/#4157a8e3265e
36.https://www.forbes.com/sites/louiscolumbus/2018/09/23/roundup-of-cloud-computing-forecasts-and-market-estimates-2018/#4ce52f51507b
37. A survey of computing strategies for green cloud, G. Rubyga ; Ponsy R. K. SathiaBhama, 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM), Year: 2016, Page s: 141 - 145
38. https://docs.aws.amazon.com/batch/latest/userguide/job_scheduling.html
39. https://support.rackspace.com/how-to/cloud-load-balancer-scheduling-algorithms/
40.https://cloud.google.com/scheduler/
41.https://console.bluemix.net/catalog/services/workload-scheduler
42.https://www.ibm.com/blogs/cloud-computing/2018/11/07/automating-decision-management/
43.https://aws.amazon.com/autoscaling/
44.https://console.bluemix.net/catalog/services/auto-scaling

Aparicio Jacobi