ALGORYTMY INSPIROWANE NATURĄ W KRYPTOANALIZIE

Iwona Polak, Mariusz Boryczka

Uniwersytet Śląski, Instytut Informatyki
[iwona.polak, mariusz.boryczka]@us.edu.pl

Streszczenie. W dzisiejszych czasach ochrona informacji jest niezwykle istotna, a jednym z elementów zapewniających ową ochronę jest kryptografia. Tu z kolei ważną rolę odgrywa kryptoanaliza, która pozwala badać bezpieczeństwo używanych szyfrów. Oprócz typowo analitycznego podejścia do łamania szyfrów (jak kryptoanaliza różnicowa, kryptoanaliza liniowa czy analiza statystyczna) od kilkunastu lat do tego celu zaprzęga się różnego rodzaju niedeterministyczne systemy inspirowane naturą. Użycie takich technik nie jest do końca intuicyjne – w kryptoanalizie często ważne jest znalezienie jednego konkretnego klucza (rozwiązania optymalnego), a każde inne rozwiązanie daje kiepskie rezultaty, nawet jeśli jest blisko optimum globalnego.
Słowa kluczowe: kryptoanaliza, metaheurystyka, algorytmy optymalizacyjne, kryptografia.

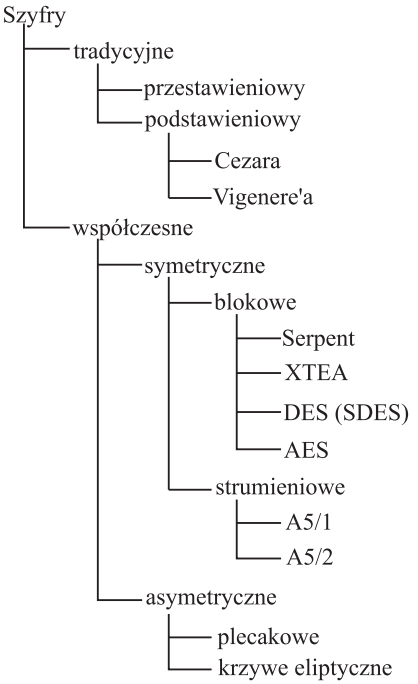
1. Wprowadzenie

Pierwszymi szyframi, które poddano kryptoanalizie metodami heurystycznymi były szyfry klasyczne. W 2003 roku John A. Clark zachęcał do sprawdzenia technik inspirowanych naturą w kryptoanalizie szyfrów współczesnych, zamiast wiary w to, iż one i tak się tam nie sprawdzą [7]. Co prawda szyfry współczesne są dużo bardziej skomplikowane niż szyfry klasyczne, jednak w ogólności opierają się na tych samych zasadach, czyli przestawianiu i podstawianiu. Od tamtej pory próbowano wielu ataków przy wykorzystaniu różnorodnych technik na różne algorytmy szyfrujące. Pojawiły się również modele hybrydowe, łączące kilka różnych podejść. Klasyfikacja szyfrów omówionych w niniejszej pracy została przedstawiona na rys. 1.

Na rzecz kryptografii także można zastosować algorymy heurystyczne. W literaturze ukazały się propozycje szyfru opartego na algorytmach genetycznych. Można również metod inspirowanych naturą użyć w celu zaprojektowania przydatnych w kryptografii silnie nieliniowych zrównoważonych funkcji boolowskich.

2. Kryptoanaliza szyfrów klasycznych

Szyfry klasyczne to szyfry przestawieniowe i podstawieniowe, wszystkie są szyframi symetrycznymi. Ich działanie polega odpowiednio na za-



Rysunek 1. Klasyfikacja szyfrów opisanych w pracy

mianie liter miejscami lub zmianie liter na inny znak lub zestaw znaków (inne litery, liczby, symbole).

W pracy [32] autorzy prezentują oryginalne podejście do kryptoanalizy szyfru przestawieniowego przy użyciu algorytmów mrówkowych [8]. Poszczególne kolumny tekstu przedstawiono w formie grafu. W fazie wstępnej wszystkie wierzchołki były połączone ze wszystkimi innymi. W każdej iteracji algorytmu każda cyfrowa mrówka budowała swoją ścieżkę. Dla danej ścieżki liczona była jej wartość na podstawie wartości bigramów (tj. sąsiadujących dwóch liter) zgodnie z naturalną częstością występowania tych bigramów w języku, w którym została zakodowana wiadomość. Po kilku iteracjach niektóre ścieżki otrzymywały wyższe wartości feromonu i były bardziej prawdopodobnymi kandydatami na klucz deszyfrujący. Autorzy opisali udaną kryptoanalizę przy maksymalnie 5000 iteracjach, przy kluczu do długości 40.

Przekrojową pracą na ten temat jest praca [4]. Autorka sprawdziła tam wiele szyfrów klasycznych: przestawieniowy, Cezara, Vigenere'a i inne wieloalfabetowe szyfry podstawieniowe. Podjęła również próbę kryptoanalizy szyfrów współczesnych: DES i AES. Prezentowane wyniki pokazują, że

algorytmy genetyczne [20] bardzo dobrze sprawdziły się przy kryptoanalizie szyfrów klasycznych, lecz nie poradziły sobie zszyframi współczesnymi. Brak efektów również przy kryptoanalizie szyfrów współczesnych za pomocą optymalizacji stadnej cząsteczek (PSO) [15].

3. Kryptoanaliza szyfrów współczesnych

Na przestrzeni kilkunastu ostatnich lat ciekawie przedstawia się kryptoanaliza szyfrów współczesnych z wykorzystaniem technik inspirowanych naturą. Wykorzystano bardzo wiele różnych technik heurystycznych do kryptoanalizy bardzo różnorodnych szyfrów współczesnych.

3.1. Szyfry asymetryczne

Osobliwym przypadkiem są kryptosystemy oparte na problemie plecakowym, ponieważ z jednej strony nie są liczone do szyfrów klasycznych, gdyż powstały dopiero w dobie komputerów, z drugiej zaś strony nie zostały nigdy wykorzystane do stworzenia współczesnego kryptosystemu [28]. Opierają się one na problemie pakowania plecaka, który zaliczany jest do klasy problemów NP-zupełnych, a klasa owa to doskonałe pole do popisu dla wszelkiego rodzaju algorytmów optymalizacyjnych. Autorzy pracy [39] poddali takie kryptosystemy analizie korzystając z algorytmów genetycznych. Znalezienie właściwego klucza wymagało przeszukania 0,09-4% przestrzeni poszukiwań (dla problemów plecakowych o rozmiarze do 25 elementów). Podobnie, w pracy [9] wykorzystano algorytmy genetyczne do kryptoanalizy szyfrowania opartego na problemie plecakowym. Dla zbioru 8-elementowego średnio po 115 populacjach otrzymywano właściwy klucz, tj. ciąg w postaci superrosnącej, ze średnim przeszukaniem poniżej 50% przestrzeni rozwiązań (czyli wszystkich możliwych kluczy).

Ukazała się również praca dotycząca kryptoanalizy systemu kryptograficznego dotyczącego krzywych eliptycznych [18]. Autorzy wykorzystali sztuczne sieci neuronowe (ANN) [21] do kryptoanalizy owego kryptosystemu dla krzywych eliptycznych o rozmiarze $p = 14, 20, 32$. Wyliczenie najmniej znaczącego bitu (lsb) logarytmu dyskretnego nad krzywą eliptyczną uzyskano ze średnią dokładnością 57% na zbiorze testowym (czyli trochę więcej niż wybór losowy) oraz 90% na zbiorze treningowym. Nie są to najlepsze wyniki, niemniej jednak jako efekt uboczny owej pracy autorzy sugerują możliwość kompresji zbioru stosując tę metodę.

3.2. Szyfry symetryczne blokowe

Interesujące podejście do szyfru Serpent opisano w pracy [3]. Szyfr został przedstawiony w formie grafu, w którym przy wykorzystaniu sieci neuronowych jest szukana najkrótsza ścieżka. Serpent jest symetrycznym szyfrem blokowym i operuje na blokach o rozmiarach 128 bitów oraz na kluczu o długości: 128, 192 lub 256 bitów. Korzysta on z 32 rund; każda runda składa się z przekształcenia XOR względem klucza rundy, użycia 128-bitowej funkcji mieszającej i zastosowania 32 4-bitowych S-Boksów. W wymienionej pracy kryptoanalizie poddano jego wersję 4-, 5-, 6- i 7-rundową. Wraz ze zwiększeniem liczby rund obserwuje się spadek skuteczności algorytmu kryptoanalitycznego.

Istnieje wiele prac dotyczących kryptoanalizy algorytmu DES. Co prawda wszystkie one dotyczą jedynie osłabionej lub ćwiczebnej wersji szyfru (SDES), jednak wiele z nich dokonuje próby porównania różnych technik adaptacyjnych, co może być przydatne do określenia kierunku dalszych badań mogących przynieść obiecujące rezultaty i porzucenia technik nie przynoszących efektów.

SDES (*Simplified DES*) został opracowany w 1996 roku jako narzędzie edukacyjne. Na wejściu pobierany jest 8-bitowy blok tekstu jawnego oraz 10-bitowy klucz, a na wyjściu produkowany jest 8-bitowy blok kryptogramu (tekstu zaszyfrowanego). Deszyfrowanie działa w odwrotnej kolejności. Zasada działania SDES jest analogiczna z zasadą działania używanego w praktyce algorytmu DES. W pracy [10] przedstawiono kryptoanalizę algorytmu SDES wykorzystując do tego celu algorytmy genetyczne oraz algorytmy memetyczne [24]. Algorytm genetyczny odwzorowuje mechanizmy rządzące światem genów i ewolucji; najważniejsze operatory to operator krzyżowania i mutacji. Z kolei algorytm memetyczny opiera się na ewolucji kulturowej i jednostką doboru jest mem, czyli jednostka informacji kulturowej. W tych badaniach osiągnięto następujące wyniki: algorytm genetyczny odtworzył od 4 do 8 bitów 10-bitowego klucza, a algorytm memetyczny – od 5 do 9 bitów, zatem algorytm memetyczny pozwolił osiągnąć trochę lepsze rezultaty. W pracy [26] próba porównania kilku technik, takich jak symulowane wyżarzanie (SA) [6], [16], optymalizacja stadna cząsteczek (PSO), algorytm genetyczny (GA) oraz przeszukiwanie tabu (TS) [11], [12], dała następujący wynik: najlepsze rezultaty osiągnięto dla przeszukiwania tabu, najgorsze dla algorytmów genetycznych.

W pracach [27], [34] badano zarówno SDES, jak i osłabioną wersję oryginalnego DES. W przypadku SDES najlepsze wyniki zostały osiągnięte dla metody TS: 344 deszyfrowania w celu znalezienia prawidłowego klucza

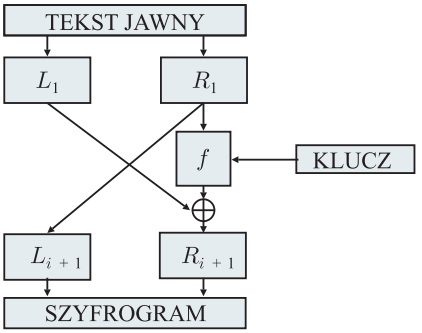
(algorytm siłowy – średnio 512). Odpowiednie wartości dla PSO to 400 deszyfrowań, a dla adaptacyjnego algorytmu genetycznego (AGA) [35] – 460 deszyfrowań. Podstawowa wersja algorytmu genetycznego (GA) nie znalazła rozwiązania nawet w 1024 iteracjach. Do badań autorzy użyli również wersji DES z liczbą rund zredukowaną z 16 do 6 oraz z kluczem zmniejszonym z 56 do 16 bitów, a także z usuniętymi S-Boksami. Dla DES najlepsze wyniki dało użycie technik: PSO, AGA oraz GA (podano w kolejności od najlepszej) [27]. W [34] przetestowano całą gamę metod heurystycznych: GA, PSO, symulowane wyżarzanie (SA), TS, algorytm pszczeli (BA) [30], optymalizacja żerowania bakterii (BFO) [29], systemy kukułcze (CS) [38]. Kryptoanalizie poddano ponownie SDES oraz DES z 16- i 32-bitowym kluczem. Najlepiej poradziło sobie PSO, najgorzej GA oraz BFO.

Na DES (wersja z okrojoną liczbą rund, bez fazy inicjalizacyjnej oraz końcowej) przeprowadzono jeszcze jeden atak [36], wykorzystując metodę ataku z wybranym tekstem jawnym. Użyto do tego celu algorytmu genetycznego. Uzyskano dobre rezultaty dla liczby rund 6 lub mniej. Autorzy twierdzą, że przy odpowiednim zmodyfikowaniu funkcji przystosowania, ich algorytm można zastosować również do innych współczesnych szyfrów blokowych.

Gdy metody heurystyczne nie były w stanie podołać kryptoanalizie samodzielnie, tworzono modele hybrydowe łączące wybraną metodę heurystyczną z bardziej „tradycyjną" metodą analityczną.

W pracy [17] wykorzystano pomocniczo PSO oraz ewolucję różnicową (DE) [37] w kryptoanalizie różnicowej. Kryptoanaliza różnicowa to metoda ataku kryptologicznego polegająca na porównaniu dwóch szyfrogramów, które powstały w wyniku zaszyfrowania tym samym kluczem dwóch odmiennych od siebie tekstów jawnych. Teksty jawne są dobrane w pewien szczególny sposób, zależny od atakowanego systemu i pozwalający na analityczne wyliczenie występujących zależności, a na tej podstawie – użytego klucza. W opisywanym przypadku atakowi został poddany szyfr DES w wersji 4- i 6-rundowej. Kryptoanaliza różnicowa stanowi pierwszą fazę działania całego algorytmu i zapewnia 42 bity klucza, natomiast w drugiej fazie techniki adaptacyjne mają znaleźć pozostałe 14 bitów, co udaje się w 93-100%, ze średnią ok. 99%. Autorzy sugerują, że ich metoda jest możliwa do wykorzystania w przypadku innych szyfrów opartych na sieci Feistela (rys. 2).

Podobnie kryptoanalizę różnicową, lecz tym razem wspartą przez algorytm genetyczny, zastosowano w [14] w celu złamania blokowego szyfru XTEA. Uzyskano bardzo dobre wyniki do 13 z 32 rund tego algorytmu – prawidłowo określono od 27 do 32 bitów 32-bitowego klucza. Podobnie jak



Rysunek 2. Schemat sieci Feistela

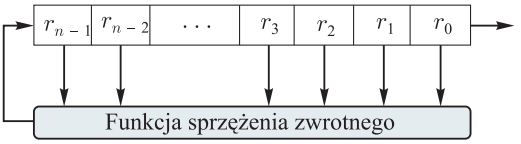
powyżej, i tutaj autorzy sugerują, że to podejście można rozszerzyć na inne szyfry oparte na sieci Feistela.

Z kolei przy kryptoanalizie algorytmu AES, w wersji z jednym S-Boksem [13] wykorzystano uczenie maszynowe [23] wmetodzie SCA (*side channel analysis*), czyli analizie poboru mocy. Analiza poboru mocy jest to metoda bazująca najczęściej na konkretnej sprzętowej bądź programowej implementacji algorytmu. W trakcie takiej analizy sprawdza się pobór mocy podczas wykonywania poszczególnych faz algorytmu i na tej podstawie można uzyskać fragmenty lub poszczególne bity klucza. W artykule porównano różne wariacje metody; co w przypadku tego typu metod dość oczywiste, wyciągnięto wniosek, że dobór odpowiednich parametrów ma bardzo duże znaczenie w jakości otrzymywanych wyników.

3.3. Szyfry symetryczne strumieniowe

W pracy skupiono się na aproksymacji strumieni bitów poprzez liniowe rejestry przesuwne ze sprzężeniem zwrotnym LFSR (*Linear Feedback Shift Register*), rys. 3. Podobne podejście zaprezentowano w pracy [1], lecz na znacznie krótszych ciągach bitów i rejestrach (do $n = 8$). W [31] przedstawiono początek prac w tym kierunku – próby znalezienia rejestru LFSR mając dany ciąg generowany przez rejestr LFSR (którego konstrukcja nie jest znana). W celu znalezienia odpowiedniego rejestru LFSR użyto algorytmów genetycznych. Dla rejestrów o długości do $n = 32$ uzyskano od 68% do 100% zgodności z zadanym strumieniem, ze średnią na poziomie 84%. Najlepsze rezultaty osiągnięto dla krótszych rejestrów, z mniejszą liczbą zaczepów. Wyniki dla losowego stanu początkowego przedstawino w tabeli 1. Jednak, ponieważ dla takiego przypadku znana jest analityczna metoda Berlekampa-Massey'a [19] dokładnego odtworzenia rejestru LFSR

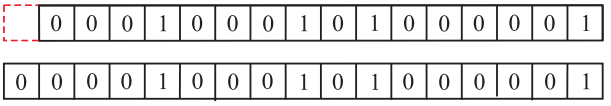
o długości n mając do dyspozycji długość wyjścia $2n$, w dalszych badaniach skupiono się na aproksymacji strumieni bitów generowanych przez inne struktury.



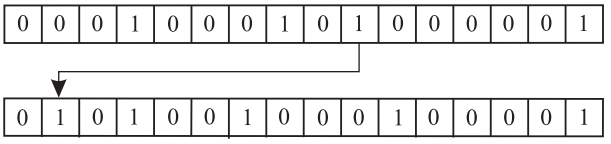
Rysunek 3. Rejestr przesuwny ze sprzężeniem zwrotnym – FSR. W przypadku LFSR funkcją sprzężenia zwrotnego jest alternatywa wykluczająca (XOR) wybranych bitów

Autorzy niniejszej pracy podjęli próby aproksymacji generatora liczb pseudolosowych BBS (Blum-Blum-Shub) [5] oraz szyfrów strumieniowych A5/1 i A5/2 będących standardem szyfrowania rozmów w telefonii komórkowej GSM. Dla generatorów BBS z okresem o długości do 328 osiągnięto przy różnego rodzaju mutacjach wyniki z przedziału od 69% do 75% aproksymacji atakowanego strumienia bitów. Zbiorcze wyniki dla różnych rodzajów mutacji podano w tabeli 2. Na tej podstawie zdecydowano się w dalszym toku badań testować tylko najbardziej obiecujące rodzaje mutacji, tj.:

1. rozciągającą – rejestr zostaje wydłużony o jedno pole (rys. 4),
2. losowe przesunięcie – losowany jest jeden z zaczepów, a następnie przenoszony na inne, również losowo wybrane, miejsce (rys. 5).



Rysunek 4. Mutacja rozciągająca



Rysunek 5. Mutacja losowe przesunięcie

Wyniki dla LFSR z losowym stanem początkowym

Tabela 1

Szukany wielomian	Stan początkowy (losowy)	Maksymalne dopasowanie	Znaleziony wielomian	Pokolenie
$x^{22} + x^{21} + 1$ (A5/1)	00101100101 00000111010	1	$x^{22} + x^{21} + 1$	16
$x^{16} + x^5 + x^4 + x^3 + 1$ (USB 3.0)	0111101111100110	0.679	$x^{22} + x^7 + 1$	13
$x^7 + x^3 + 1$ (CRC-7, MMC)	0001100	1	$x^7 + x^3 + 1$	5
$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11}$ $+x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$ (CRC-32-MPEG2, CRC-32-IEEE 802.3)	1110111010101110 0000000000000000		$x^{26} + x^{25} + x^{24} + x^{23} + x^{22} + x^{21} + x^{20}$ $+x^{18} + x^{17} + x^{16} + x^{13} + x^{12} + x^7 + x^6$ $+x^4 + x^3 + x^2 + x^1 + 1$	16 16
$x^{23} + x^{21} + x^{20} + x^{18} + x^{17} + x^{16}$ $+x^{11} + x^{10} + x^9 + x^7 + x^3 + 1$ (wygenerowany losowo)	110010010110 11100000000	0.710	$x^{31} + x^{28} + x^{23} + x^{20} + x^{18} + x^{16} + x^{14}$ $+x^9 + x^3 + x^2 + x^1 +$	29

Tabela 2

Mutacja posortowana względem najlepszego przystosowania wszystkich testowanych generatorów BBS

Mutacja	Średnie najlepsze przystosowanie	Średnia generacja
rozciągająca	0,754	25
losowe przesunięcie	0,724	15
zmiana na przeciwny	0,715	23
sąsiednie przesunięcie	0,71	15
zwężająca	0,696	8
dodanie zaczepu	0,695	7
usunięcie zaczepu	0,688	5

W przypadku szyfrów A5/1 i słabszej odmiany A5/2 wyniki również są obiecujące. Algorytmy te składają się odpowiednio z trzech oraz czterech sprzężonych ze sobą rejestrów LFSR. Podczas badań przy aproksymacji z wykorzystaniem pojedynczego rejestru LFSR o długości z przedziału 20–30 uzyskano zgodność bitów generowanych przez ów znaleziony rejestr z bitami zadanego wyjścia na poziomie średnim 70%, a dla najlepszych przypadków do 77%. Oznacza to, że korzystając tylko z jednego rejestru LFSR można odszyfrować znaczącą część zaszyfrowanej wiadomości. Autorzy niniejszej pracy będą kontynuować kierunek swoich badań.

4. Algorytmy inspirowane naturą na rzecz kryptografii

Warto również wspomnieć, że opublikowano pierwsze próby stworzenia algorytmów kryptograficznych opartych na metodach heurystycznych czy też wykorzystano te techniki pomocniczo przy generowaniu składników wykorzystywanych w kryptografii.

Autorzy pracy [22] korzystając z algorytmów genetycznych projektują silnie nieliniowe, zrównoważone funkcje boolowskie, które wykorzystuje się w kryptografii do projektowania S-Boksów odpornych na kryptoanalizę liniową i różnicową. Taka pseudolosowa metoda zapewnia odpowiednią alternatywę dla systematycznych metod budowy kryptograficznie silnych funkcji boolowskich.

Z kolei w pracy [25] zaproponowano kryptosystem oparty na algorytmie genetycznym z deterministycznymi operatorami krzyżowania i mutacji. Jest to o tyle ciekawe, że wszystkie omawiane do tej pory techniki heurystyczne mają to do siebie, iż dużą rolę gra w nich przypadek. Są one ze swojej natury losowe i przy każdym wywołaniu algorytmu można otrzymać odmienny wynik. Kryptosystem z kolei wymaga tego, aby zaszyfrowany tekst miał postać taką, by dało się go bez problemu rozszyfrować i otrzymać w wyniku tego sensowne dane. Najczęściej kryptosystemy buduje się w taki sposób, że dany tekst jawny po zaszyfrowaniu danym kluczem za każdym razem da taki sam szyfrogram. Jak widać założenia algorytmu genetycznego i algorytmu szyfrującego są ze sobą sprzeczne. Niemniej jednak autorom omawianej pracy udało się stworzyć modyfikację GA z determistycznymi operatorami krzyżowania i mutacji, która to modyfikacja dla danego tekstu jawnego i klucza wygeneruje za każdym razem ten sam szyfrogram. Co prawda wadą jest to, że szyfrogram zajmuje dwa razy więcej miejsca niż tekst jawny, jednakże sama idea takiego kryptosystemu jest dość oryginalna. Podobnie technikę optymalizacyjną wykorzystano w [2] do stworzenia kryptosystemu opartego na algorytmach genetycznych z pseudolosową

sekwencją. W użytej wersji algorytmu z dwóch operatorów genetycznych wybrano tylko krzyżowanie. Stworzony szyfr jest szyfrem strumieniowym opartym na NLFSR (*Non-Linear Feedback Shift Register*). W swoich badaniach eksperymentalnych autorzy wykazali przepustowość wystarczająco dobrą dla rzeczywistych zastosowań.

5. Podsumowanie

W pracy została przedstawiona cała gama algorytmów inspirowanych naturą w atakach na wiele kryptosystemów. Co prawda bardzo dobre wyniki są uzyskiwane jak dotąd tylko dla szyfrów klasycznych, niemniej jednak powolny, acz systematyczny postęp w łamaniu współczesnych obecnie używanych kryptosystemów pozwala domniemywać, że algorytmy genetyczne, mrowiskowe czy sieci neuronowe nie powiedziały jeszcze ostatniego słowa w tej kwestii. Z jednej strony cieszy, że współczesne kryptosystemy opierają się nowym technikom ataku, z drugiej strony trzeba się strzec i mieć na uwadze, że kryptoanaliza przy wykorzystaniu systemów inspirowanych naturą daje satysfakcjonujące rezultaty dla coraz silniejszych wersji alorytmów szyfrujących i być może dopracowanie tych technik pozostaje tylko kwestią czasu.

Literatura

[1] A. A. Abd, H. A. Younis, W. S., Awad, it Attacking of stream Cipher Systems Using a Genetic Algorithm, Journal of the University of Thi Qar, Volume 6, pp.1–6, 2011.

[2] A. Almarimi, A. Kumar, I. Almerhag, N. Elzoghbi, *A new approach for data encryption using genetic algorithms*, 2006.

[3] A. G. Bafghi, R. Safabakhsh, B. Sadeghiyan, *Finding the differential characteristics of block ciphers with neural networks*, Information Sciences, Vol.178, No 15, pp. 3118–3132, 2008.

[4] K. P. Bergmann, *Cryptanalysis Using Nature-Inspired Optimization Algorithms (master's thesis)*, 2007.

[5] L. Blum, M. Blum, M. Shub, *A Simple Unpredictable Pseudo Random Number Generator*, SIAM J. Comput. 15, 2, pp. 364–383, 1986.

[6] V. Cerny, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications, pp. 41–51, 1985.

[7] J. A. Clark, *Invited paper. Nature-inspired cryptography: Past, present and future*, Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, pp. 1647–1654, 2003.
[8] M. Dorigo, D. Di Caro, L. M. Gambardella, *Ant Algorithms for Discrete Optimization*, Artificial Life, pp. 137–172, 1999.
[9] P. Garg, A. Shastri, *An Improved Cryptanalytic Attack on Knapsack Cipher using Genetic Algorithm*, World Academy of Science, Engineering and Technology, pp.553–560, 2007.
[10] p. Garg, *A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm*, International Journal of Network Security & Its Applications (IJNSA), Vol.1, No 1, pp.34–42, April 2009.
[11] F. Glover, *Future Paths for Integer Programming and Links to Artificial Intelligence*, Comput. Oper. Res., pp. 533–549, 1986.
[12] F. Glover, *Tabu SearchPart I*, ORSA Journal on Computing, pp. 190–206, 1989.
[13] G. Hospodar, B. Gierlichs, E. D. Mulder, I. Verbauwhede, J. Vandewalle, *Machine learning in side-channel analysis: a first study*, J Cryptogr Eng (2011), pp.293–302, 2011.
[14] P. Itaima, M. C. Riffa, *Applying Differential Cryptanalysis for XTEA using a Genetic Algorithm*, 2008.
[15] J. Kennedy, R. Eberhart, *Particle swarm optimization*, IEEE International Conference on Neural Networks, Proceedings, pp. 1942–1948, 1995.
[16] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *Optimization by Simulated Annealing*, Science, Number 4598, pp. 671–680, 1983 .
[17] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, M. N. Vrahatis, *Applying evolutionary computation methods for the cryptanalysis of Feistel ciphers*, Applied Mathematics and Computation 184(1), pp. 63–72, 2007.
[18] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, D. K. Tasoulis, M. N. Vrahatis, *Assessing the effectiveness of artificial neural networks on problems related to elliptic curve cryptography*, Mathematical and Computer Modelling, Vol.46, No 12, pp.174–179, 2007.
[19] J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Transactions on Information Theory, Vol.15, No 1, pp.122–127, 1969.
[20] Z. Michalewicz, *Algorytmy genetyczne + struktuy danych = programy ewolucyjne*, Warszawa, Wydawnictwa Naukowo-Techniczne, 2003.
[21] Z. Michalewicz, D. B. Fogel, *Jak to rozwiązać czyli Nowoczesna heurystyka*, Warszawa, Wydawnictwa Naukowo-Techniczne, 2006.

[22] W. Millan, A. Clark, E. Dawson, *Heuristic design of cryptographically strong balanced Boolean functions*, Advances in Cryptology EUROCRYPT'98, Lecture Notes in Computer Science, pp. 489–499, 1998.

[23] T. M. Mitchell, *Machine Learning*, McGraw-Hill Inc., New York, NY, USA, 1997.

[24] P. Moscato, *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*, California Institute of Technology, 1989.

[25] N. Nalini, G. Raghavendra Rao, *A new encryption and decryption algorithm combining the features of genetic algorithm (GA) and cryptography*, 1999.

[26] N. Nalini, G. Raghavendra Rao, *Cryptanalysis of Simplified Data Encryption Standard via Optimisation Heuristics*, 2006.

[27] N. Nalini, G. Raghavendra Rao, *Experiments on Cryptanalysing Block Ciphers via Evolutionary Computation Paradigms*, 2006.

[28] A. M. Odlyzko, *The rise and fall of knapsack cryptosystems*, Cryptology and Computational Number Theory, pp. 75–88, 1990.

[29] K. M. Passino, *Bacterial Foraging Optimization*, Int. J. Swarm. Intell. Res., vol. 1, no 1, pp. 1–16, 2010.

[30] D. T. Pham i inni, *The bees algorithm*, Technical report, Manufacturing Engineering Centre, Cardiff University, UK, 2005.

[31] I. Polak, M. Boryczka, *Breaking LFSR Using Genetic Algorithm*, Computational Collective Intelligence, Technologies and Applications, pp. 731–738, 2013.

[32] M. D. Russell, J. A. Clark, S. Stepney, *Making the Most of Two Heuristics: Breaking Transposition Ciphers with Ants*, Evolutionary Computation, CEC '03, Vol. 4, pp. 2653–2658, 2003.

[33] B. Schneier, *Kryptografia dla praktyków*, Wyd. 2, Warszawa, Wydawnictwa Naukowo-Techniczne, 2002.

[34] G. Selvi, T. Purusothaman, *Cryptanalysis of Simple Block Ciphers using Extensive Heuristic Attacks*, European Journal of Scientific Research, Vol.78 No.2 (2012), pp.198–221, 2012.

[35] M. Srinivas, L. M. Patnaik, *Adaptive probabilities of crossover and mutation in genetic algorithms*, IEEE Transactions on Systems, Man, and Cybernetics, 4, pp. 656–667, 1994.

[36] J. Song, H. Zhang, Q. Meng, Z. Wang, *Cryptanalysis of Four-Round DES Based on Genetic Algorithm International Conference on Wireless Communications*, Networking and Mobile Computing. WiCom. pp. 2326–2329, 2007.

[37] R. Storn, K. Price, *Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, J. of Global Optimization, vol. 11, no 4, pp. 341–359, 1997.

[38] X. -S. Yang, S. Deb, *Cuckoo Search via Lvy Flights*, NaBIC, IEEE, pp. 210–214, 2009.

[39] I. F. T. Yaseen, H. V. Sahasrabuddhe, *Breaking multiplicative knapsack ciphers using a genetic algorithm*, Proceedings of the International Conference on Knowledge Based Computer Systems, p. 129–139, 1998.

CRYPTANALYSIS USING NATURE-INSPIRED ALGORITHMS

Abstract. Nowadays protection of information is very crucial and cryptography is a significant part of keeping information secure. Here in turn cryptanalysis plays an important role by examining the safety of ciphers used. Besides the analytical approach to ciphers breaking (eg. differential cryptanalysis, linear cryptanalysis, statistical analysis) for this purpose there are several kinds of non-deterministic, inspired by nature systems applied. It is not intuitive - as in cryptanalysis often it is important to find the exact key used (optimal solution) and every other solution is giving poor results, even if it is near global optimum.

Keywords: cryptanalysis, metaheuristic, optimization algorithms, cryptology.