Prosopon
Europejskie Studia Społeczno-Humanistyczne

Wydawca / Publisher:
Instytut Studiów
Międzynarodowych
i Edukacji HUMANUM
www.humanum.org.pl

12 (3) 2015
ISSN 1730–0266

Michał Stachura
Rzeszów, Poland

Przedstawienie metody szyfrowania informacji ZT-UNITAKOD / *Presentation of the ZT-UNITAKOD course method*

Abstract
The Ministry of Foreign Affairs, which in 1993 had 45 diplomatic missions, stated unequivocally that modern technology is so advanced that it is best to assume the existence of eavesdropping in all settings at all times. In 1992 the Ministry of Foreign Affairs assigned 38,000 secret encrypted messages. Secret letters from the same ministry are transported on 160 different routes by couriers, ie specially trained officers for this purpose. In the sixties crashed the plane, which carried very secret messages, then killed bags with secret diplomatic mail, the rest survived.
The situation I described above clearly shows that the best and only secure means of transport can be mailed electronically in encrypted form. The cipher used to encrypt such important state information must be unbreakable. This cipher must be based on mathematical models that exclude human (human factor) from the encryption and decryption process. Only such a cipher can give us one hundred percent effective protection of information.

Key words: **safety, information, zt-unitakod.**

Ministerstwo Spraw Zagranicznych, które w roku 1993 posiadało 45 placówek dyplomatycznych stwierdza jednoznacznie, iż nowoczesna technika jest tak zaawansowana, że najlepiej jest założyć fakt istnienia podsłuchu we wszystkich placówkach przez cały czas. W roku 1992 MSZ nadało 38 tysięcy tajnych zaszyfrowanych depesz. Tajne listy z tegoż ministerstwa przewożone są na 160 różnych trasach za pomocą kurierów, czyli specjalnie przeszkolonych do tego celu oficerów. W latach sześćdziesiątych rozbił się samolot, który przewoził bardzo tajne depesze, zginęły wówczas worki z tajną pocztą dyplomatyczną, reszta ocalała.

Sytuacja, którą opisałem powyżej pokazuje wyraźnie, że najlepszym i jedynym bezpiecznym środkiem transportu może być poczta przesyłana drogą elektronicz-

ną w zaszyfrowanej postaci. Szyfr użyty do szyfrowania tak ważnych informacji państwowych musi być niemożliwy do złamania. Szyfr ten musi opierać się na modelach matematycznych, które wykluczą człowieka (czynnik ludzki) z procesu szyfrowania i deszyfracji. Tylko taki szyfr może dać nam stu procentową skuteczność ochrony informacji.

Jest to tylko jeden z przykładów, dla których szyfrowanie jest tak ważne w życiu społeczeństw. Metoda szyfrowania, która wyeliminowała człowieka z procesu szyfrowania i deszyfracji opatentowana jest pod nazwą ZT-UNITAKOD. Prace nad ta metodą rozpoczęły się w roku 1985. W roku 1988 metoda przyjęła powyższą nazwę i została bardzo pozytywnie przyjęta przez zespoły naukowe zachodu.

Do powstania metody ZT-UNITAKOD przyczynił się sam człowiek a ściśle mówiąc jego decydująca rola w szyfrowaniu informacji. Po przeanalizowaniu poszczególnych statycznych metod szyfrowania łatwo dojść do wniosku, że najsłabszym ogniwem w tej układance jest sam człowiek oraz klucz. Człowiek opracowuje dany klucz, przechowuje, przesyła, wymienia i ochrania. Nawet jeżeli sama metoda była by nie do złamania to jej wartość rzeczywista staje się zerowa. Metoda komputerowego zabezpieczenia poufności informacji musi być metodą niezależną od człowieka i posiadać klucz zmieniający się w czasie, bez ingerencji człowieka, który zależy tylko od zmieniającego się czasu przetwarzania.

1.1 PODSTAWOWE INFORMACJE

Metoda ZT-UNITAKOD posiada w swojej budowie elementy zmienne i stałe:

1. Elementy stałe – STANDARD
2. Elementy zmienne - SUPLEMENT

W skład STANDARDU wchodzą:

a) Model matematyczny szyfru.
b) Model matematyczny deszyfracji
c) Tablice kryptograficzne A1, A2, A3, A4, i A
d) Pięć rodzajów szyfrów generowanych jednocześnie
e) Wymagania metody dotyczące generatorów permutacji oraz dynamiki kryptosystemów

W skład SUPLEMENTU wchodzą:

a) Generatory permutacji
b) TIME i sposób jego generowania
c) Zasada tworzenia tablic kryptograficznych przy wykorzystaniu TIME, po jego podziale
d) Zasada szyfrowania strumienia danych i deszyfracji szyfrogramu

W metodzie ZT-UNITAKOD zastosowano sumę dwóch elementów modulowaną liczbą N = 256, a tworzenie klucza szyfrującego w postaci tablicy A uzależniono od dwóch generatorów permutacji i układu elementów dynamicznych w postaci

daty i czasu. Szyfr ten jest elementem zmiennym uzależnionym od czasu szyfrowania. Podczas szyfrowania można wygenerować pięć różnych rodzajów szyfru. Informacje sterujące są szyfrowane inaczej nić przesyłany meldunek. Metoda ta nie wymaga tak zwanego transportu tajnych kluczy, ani specjalnego utajniania informacji sterującej.

Model matematyczny szyfru metody ZT-UNITAKOD:

Szyfr = (A + B) mod N,

gdzie:

A – tablica kryptograficzna – klucz szyfrujący utworzony z 65 536 bajtów,

B – przesyłana informacja jawna

N – liczba znaków alfabetu – dla kodu ASCII wartość N = 256

Model matematyczny deszyfracji:

$$B = S - A \qquad \text{dla: } S - A > 0$$

$$B = (S - A) + N \qquad \text{dla: } S - A \leq 0$$

1.2 ZASTOSOWANE GENERATORY PERMUTACJI

1. Generator multiplikatywny G_1

$G_1 = (cx_i) \bmod n$, gdzie c – liczby nieparzyste z przedziału od 3 do 255

Są to następujące liczby: 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213, 215, 217, 219, 221, 223, 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255.

Razem jest to 127 liczb. x_i – ciąg liczb całkowitych od 1 do N

1. Generator mieszany G_2

$G_2 = (ax_i + b) \bmod n$, gdzie a – liczby nieparzyste spełniające równanie: $a = 1 \pmod 4$.

Są to liczby: 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89, 93, 97, 101, 105, 109, 113, 117, 121, 125, 129, 133, 137, 141, 145, 149, 153, 157, 161, 165, 169, 173, 177, 181, 185, 189, 193, 197, 201, 205, 209, 213, 217, 221, 225, 229, 233, 237, 241, 245, 249, 253.

Razem 63 liczby, b – liczby nieparzyste z przedziału od 1 do 255. Liczby te są równe iloczynom „c" z dodatkiem liczby 1(jeden), x_i – ciąg liczb całkowitych od 1 do N.

Pary a/b stosowane w G_2 tworzone są z liczb „a" oraz „b". Liczby łączymy ze sobą według następującej zasady: pierwsza liczbę „a", w naszym przypadku „5" łączymy

ze wszystkimi liczbami „b" i otrzymujemy ciąg par a/b: 5/1, 5/3, 5/5, 5/7,….,5/255. Następnie bierzemy kolejną liczbę „a", która również łączymy ze wszystkimi liczbami „b" i otrzymujemy kolejny ciąg par. Tak postępujemy ze wszystkimi liczbami „a" i „b".

Ponieważ ilość liczb a = 63 i liczb b = 128, stad liczba par wynosi: 63 * 128 = 8064.

1.3 OGÓLNY ALGORYTM METODY ZT-UNITAKOD

Algorytm ZT-UNITAKOD generuje nieokresowy podwójny szyfr strumieniowy. Strumień kluczy $k=k_1k_2k_3$ … generowany jest synchronicznie ze strumieniem tekstu jawnego. Generowanie to odbywa się deterministyczną metoda losową z użyciem szeregu jedno i dwu wymiarowych tablic kryptograficznych, funkcji i generatorów liczb pseudolosowych. Poza tym w metodzie zastosowano szyfrowanie danych inicjujących proces generowania tablic i działania generatorów liczb pseudolosowych oraz umieszczanie ich w kryptogramie w postaci zaszyfrowanej. Proces kodowania źródłowej postaci tekstu jawnego jak i danych inicjujących oparty jest o zastosowanie sumy arytmetycznej dwóch liczb całkowitych, co znacząco zwiększa odporności szyfru na złamanie. Dla przykładu kod występujący w kryptogramie może być wynikiem dodawania wielu możliwości np. kod 120 można otrzymać dodając do 119 + 1 jak i dodając do 110 + 10. Kryptoanalityk staje więc przed problemem rozwiązania pojedynczego równania z dwoma niewiadomymi, co już w pewien sposób ogranicza zakres jego możliwości. Generator strumienia klucza jest inicjowany i pracuje w oparciu o zbiór zarówno stałych jak i zmiennych elementów kryptosystemu.

Szyfrowanie za pomocą metody ZT-UNITAKOD odbywa się w 5 etapach. Na początku tworzona jest początkowa tablica kryptograficzna A0. Powstaje ona w oparciu o zestaw stałych liczb T i zawartość wektora R0. Tablica A0 ma wymiary NxN. W drugim etapie w oparciu o zawartość tej tablicy powstaje wartość zewnętrznych parametrów dynamicznych (unikalnych dla każdego kryptogramu poprzez swoją tymczasowość np. dokładny czas rozpoczęcia momentu szyfrowania, identyfikatory nadawcy i odbiorcy kryptogramu i inne) Bezpośredni proces kodowania tych wartości odbywa się w oparciu o wzór

$$D'[l] = (A0[i,j] + D[k]) \bmod N_A$$

gdzie,

D[k] - to kolejne elementy wektora zawierającego wartość parametrów dynamicznych w postaci jawnej,

A0[i,j] - elementy początkowej tablicy kryptograficznej,

D'[l] - jest elementem wektora zawierającego zaszyfrowaną postać wartości parametrów dynamicznych.

N_A jest liczbą elementów alfabetu przyjętego w kryptogramie standardu służącego do cyfrowego odwzorowania danych w kryptosystemie np. dla ASCII, $N_A = 256$.

W trzecim etapie z tablicy A0, elementów wektora D i elementów zbioru T tworzy się końcową tablicę kryptograficzną A, która służy następnie do ostatecznego i bezpośredniego kodowania źródłowego tekstu jawnego.

W czwartym etapie odbywa się bezpośrednie kodowanie źródłowego tekstu jawnego w oparciu o wzór:

$S[m] = (A[i,j] + B[m]) \bmod N_A$

gdzie,

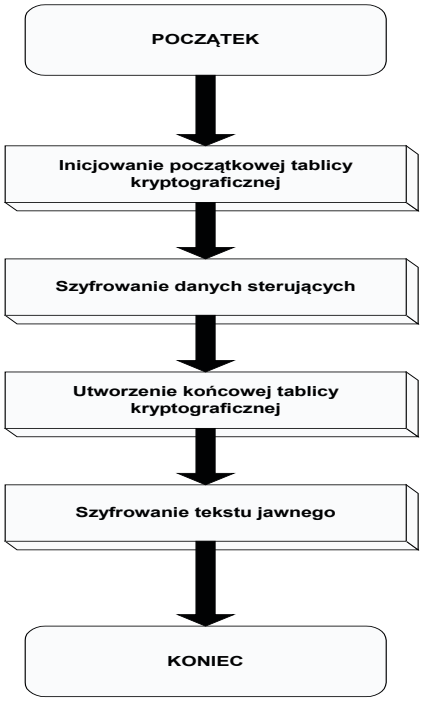
S[m] - to kolejny element zaszyfrowanej postaci tekstu jawnego,

A[i,j] - to element końcowej tablicy kryptograficznej a B[m] kodowany element tekstu jawnego.

Wyboru elementów tablicy A służących do szyfrowania kolejnych porcji (np. bajtów) tekstu jawnego dokonuje się w oparciu o funkcję będącą iloczynem kartezjańskim elementów D, T, R0 i A.

Ostatnim piątym etapem działania algorytmu jest scalenie wektora D' i zaszyfrowanych wartości tymczasowych inicjujących parametrów dynamicznych oraz wektora S zawierającego zaszyfrowaną postać źródłową tekstu jawnego. To połączenie realizowane jest w oparciu o funkcję c,, która dodatkowo może jeszcze mieszając elementy D' i S umieszczając je w sposób pseudolosowy w wektorze C zawierającym ostateczną postać kryptogramu.

Ogólny algorytm szyfrowania:

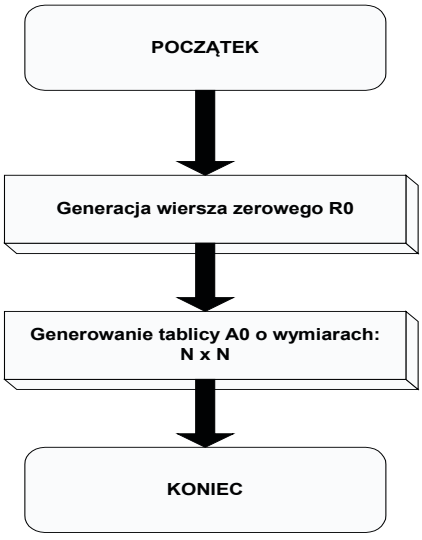


a. Inicjowanie początkowej tablicy kryptograficznej

R0 jest N - wymiarowym wektorem liczbowym. Zawiera on zestaw N dowolnych liczb. Jego postać musi być identyczna zarówno u nadawcy jak i odbiorcy zaszyfrowanej wiadomości. W kryptosystemie opartym na tym algorytmie spełnia on role dynamicznego elementu identyfikacyjnego wymienianej między nadawca i odbiorcą informacji. Może być również wbudowany na stałe w kryptosystem z możliwością lub bez możliwości modyfikacji jego postaci, w postaci jawnej lub zaszyfrowanej w oparciu o zestaw stałych liczb T, własne funkcje i generatory liczb pseudolosowych.

Wektor R0 służy do utworzenia początkowej tablicy A0. Tablica ta zawiera dokładnie N wierszy po N elementów w każdym. Wiersze zawierają wartości liczbowe, które są produktem iloczynu kartezjańskiego wektora R0 i stałych wartości liczbowych T. W najprostszym tworzenie tych wierszy może odbywać się poprzez cykliczne przesuwanie w lewo wektora R0 o ilość pozycji zgodną z kolejnymi liczbami pobranymi z T. W praktyce jednak funkcja odwzorowująca jest bardziej skomplikowana.

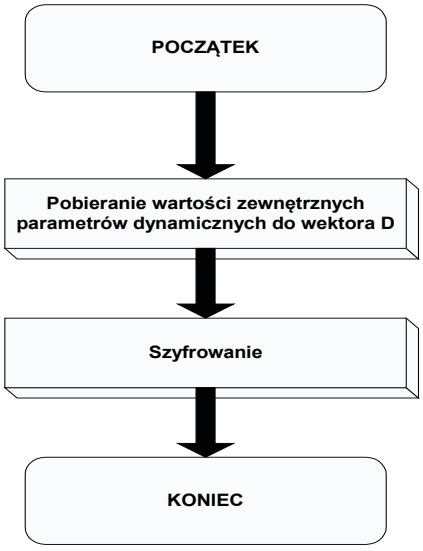
Algorytm inicjowania początkowej tablicy kryptograficznej A0;



b. Algorytm szyfrowania danych sterujących:

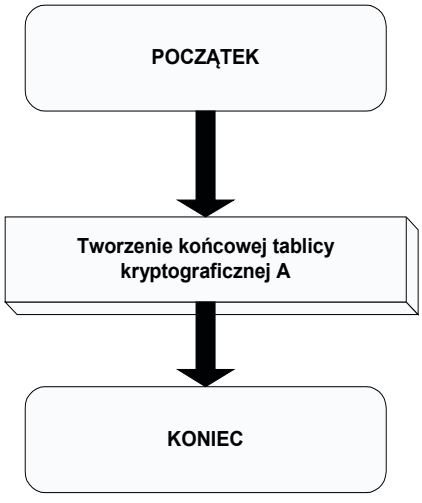
Dynamiczne wartości inicjujące pobiera się do systemu w postaci cyfrowej. Następnie kody tych wartości umieszcza się w wektorze D.

W tym etapie działania algorytmu współrzędne kolejnych elementów tablicy A0 służące do bezpośredniego kodowania elementów wektora D pobiera się w oparciu o zestaw wartości liczbowych T i przyjęte funkcje i generatory pseudolosowe.



c. Algorytm tworzenia końcowej tablicy kryptograficznej A.

Na tym etapie działania algorytmu odbywa się tworzenie końcowej tablicy kryptograficznej A, której elementy są sumowane z kolejnymi elementami tekstu jawnego. Elementy tablicy są generowane na podstawie zawartości A0, D i T. W prostej wersji może to być np. przestawianie elementów kolejnych wierszy A0 zgodnie z adresami pobranymi z T na podstawie wartości z wektora D. W praktyce funkcja f zawiera jeszcze dodatkowe elementy związane z działaniem generatorów liczb pseudolosowych.



d. Algorytm szyfrowania informacji jawnej.

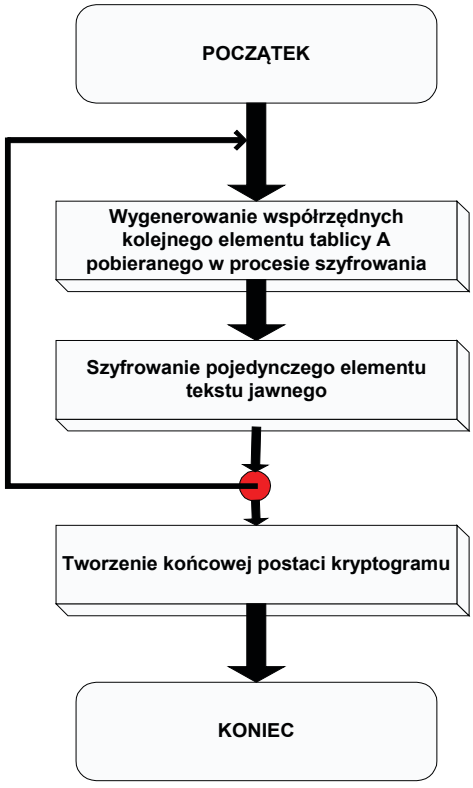
Ten etap działania algorytmu szyfrowania zawiera proces generowania współrzędnych kolejnych elementów tablicy A. Elementy te są z kolei dodawane bezpośrednio do kodów wartości elementów tekstu źródłowego. Generowanie współrzęd-

nych odbywa się w oparciu o funkcję deterministyczną w odwzorowującą iloczyn kartezjański wszystkich dotychczas wygenerowanych elementów kryptosystemu (R0, A0, T, D) w dwuwymiarowy wektor zawierający numer wiersza i kolumny w tablicy A. Wartości funkcji w są z zakresu <N ; N> dlatego elementy kolejno pobierane do szyfrowania pochodzą z dowolnego miejsca tablicy A (np. dla N = 256 jest 2^{16} elementów tablicy A).

Szyfrowanie pojedynczego elementu tekstu źródłowego odbywa się w oparciu o sumę arytmetyczną (tak jak dla dynamicznych parametrów inicjujących). Własność tej metody kodowania opisano krótko w komentarzu do ogólnego algorytmu szyfrowania.

Funkcja c powoduje przemieszanie elementów zaszyfrowanej postaci tekstu źródłowego z elementami wektora zawierającego zaszyfrowane inicjujące parametry dynamiczne. Funkcja ta musi być odwracalna.

Najprostszym rozwiązaniem implementacyjnym jest umieszczenie elementów D' na początku lub na końcu kryptogramu. Długość D' jest zmienna, więc nawet w takim przypadku wydzielenie D' i S przez kryptoanalityk jest trudne. Ostateczna postać kryptogramu może zawierać także elementy kontroli parzystości lub inne.



e. Algorytm deszyfrowania informacji.

Algorytm deszyfrujący zawiera pięć etapów działania. Na początku rozdziela się kryptogram na cześć S zawierającą zaszyfrowaną postać tekstu źródłowego i wektora D' z zaszyfrowanymi wartościami inicjujących parametrów dynamicznych. Z kolei generowanie wiersza zerowego odbywa się identycznie jak dla procesu szyfrowania. W trzecim etapie deszyfruje się wartości parametrów dynamicznych z wektora D' w oparciu o wzór:

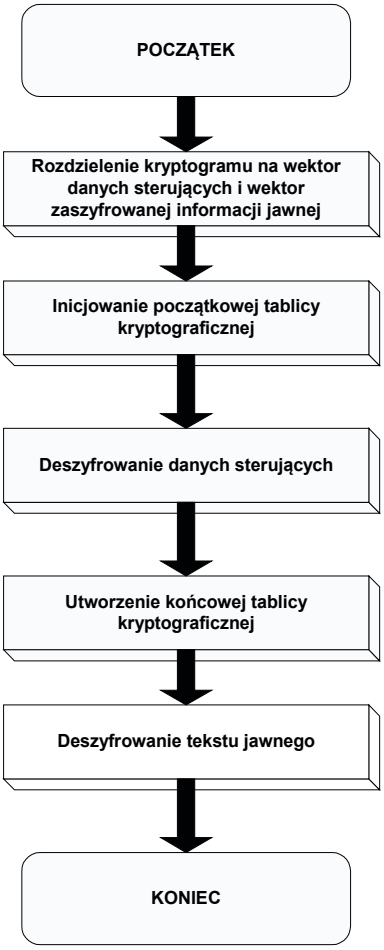
$$D[l] = (A0[i,j] - D'[k]) \bmod N_A$$

Oznaczenia i funkcja wyboru kolejnych elementów pozostaje taka sama jak dla procesu szyfrowania.

Tworzenie kryptograficznej tablicy A odbywa się identycznie jak dla szyfrowania (parametry dynamiczne są już w postaci jawnej w wektorze D). Deszyfracja elementów wektora S odbywa się w oparciu o wzór:

$$B[m] = (a[i,j] - S'[m]) \bmod N_A$$

Ostatecznie deszyfrowana postać tekstu źródłowego znajduje się w wektorze B.



Klucze dynamiczne stosowane w metodzie ZT-UNITAKOD:

1. DATA i CZAS
2. Seed lub TIME
3. liczby dla generatorów
   - pięć liczb,
   - 31 par liczb a/b
   - 256 par liczb a/b

Funkcje kluczy dynamicznych:

- wszystkie funkcje są czasowo zmienne,
- klucze dynamiczne nie biorą bezpośredniego udziału w szyfrowaniu informacji, czyli nie są częścią składową szyfru
- klucze dynamiczne inicjują tworzenie tablicy kryptograficznej, której układ zmienia się co jedną sekundę.

1.4 JAK POWSTAJE KLUCZ DYNAMICZNY

Klucz pierwszy DATA i CZAS tworzą jednorazowy układ szyfrujący.

Przykład:

DATA = 7 grudnia 2004 r. co zapisujemy: 07.12.2004.

CZAS = godzina czternasta, 32 minuty i 18 sekund, co zapisujemy 14.32.18.

Stąd DATA i CZAS utworzyły zapis: 07122004143218.

Taki sam zapis utworzony z DATY i CZASU nie powstanie już nigdy, chociaż sam zapis zmienia się co jedną sekundę. Mówiąc inaczej, szyfrogram przyjmuje ściśle określoną i jednorazową postać.

Klucz drugi to Seed, który wykorzystując wyniki DATY i CZASU tworzy tak zwany wiersz zerowy „R0" składający się z 256 znaków kodu ASCII. Liczba permutacji R0 = 256 !.

Kolejny klucz o nazwie TIME zastępuje Seed, a jego liczba permutacji wynosi:

$L_p = (256 !)^{256}$

Kolejne trzy klucze dynamiczne to układy liczb dla generatorów permutacji.

To wszystko składa się na metodę ZT-UNITAKOD. Przeprowadzenie próby łamania tej metody omówię w kolejnych rozdziałach mojej pracy.

1.5 WYMAGANIA METODY ZT-UNITAKOD

Wymagania tej metody są bardzo ściśle związane z wymaganiami generatorów permutacji. Generatory permutacji zastosowane w tej metodzie należą do SUPLEMENTU, więc mogą być inne niż te obecnie tu zastosowane. Zmiana generatorów

w żaden sposób nie wpływa ujemnie na moc kryptograficzną metody ZT-UNI-TAKOD.

Wymagania dla generatora G_1

1. $(cx_i) \bmod n \neq 0$ dla $i = 1, 2, \ldots, n - 1$
2. $(cx_i) \bmod n \neq 0$ dla $i = n$
3. $(cx_i) \bmod n \neq (cx_j) \bmod n$ dla $i \neq j$

Wymagania dla generatora G_2

1. $(ax_i + b) \bmod n \neq 0$ dla $i = 1,2, .., n.$
2. $(ax_i + b) \bmod n \neq (ax_j + b) \bmod n$ dla $i = j.$

Rodzaje tablic kryptograficzny i par a/b:

Litera „A" przedstawiona w matematycznym wzorze szyfrowania informacji zawiera trzy rodzaje tablic kryptograficznych: T1, T2, TA.

Opis poszczególnych tablic:

1. T1 – jest to tablica początkowa posiadająca przesunięty wiersz zerowy
2. T2 – jest to tablica posiadająca zmieniony układ znaków, wygenerowana za pomocą generatora G_2, po zastosowaniu jednego z 31 układów par a/b
3. TA – tablica końcowa, wygenerowana za pomocą generatora G_2 po zastosowaniu jednego z 256 układów par a/b.

Wygląd tablicy kryptograficznej:

1	2	…	255	256
2	3	…	256	1
. . .		Tablica kryptograficzna		
255	256	…	253	254
256	1	…	254	255

Pary a/b są grupowane dwojako:

1. Układ 256 par a/b
2. Układ 31 par a/b

Z par liczb „a" i „b" tworzone są tak zwane pary a/b dla generatora G_2.

Liczba wszystkich par a/b wynosi M = 8064.

Liczba permutacji dla układu 256 par a/b wynosi zatem: wpisz ze wzoru str. 38

Liczba permutacji dla układu 31 par a/b wynosi: wpisz ze str. 38

Układ liczb nieparzystych zastosowanych w tej metodzie składa się z pięciu liczb, Metoda ZT-UNITAKOD zawiera 100 takich układów. Ogólna liczbę układów po 5 liczb możemy wyrazić wzorem: wstaw ze str. 38

Całkowita liczba permutacji;

Całkowitą liczbę permutacji tablicy „A" z uwzględnieniem zmiennej pozycji początku tablicy, możemy wyrazić wzorem:

$L_p = (N!)^{2N}$,

Gdzie N = 256 – dla kodu ASCII.

5.6 GENERATORY WYKORZYSTANE W METODZIE ZT-UNITAKOD

Teorią prawdopodobieństwa w sposób ścisły są związane generatory. Rozróżnia się dwa typy generatorów:

1. generatory programowe
2. generatory fizyczne

Z generatorów fizycznych w rzeczywistości warto wyróżnić dwa szczególne typy. Pierwszy to moneta. Rzut monetą daje możliwość losowego otrzymania orła lub reszki. Prawdopodobieństwo otrzymania jednego z nich wynosi ½. Zmienna losowa ma tutaj rozkład dwupunktowy przyjmując wartości albo 0 albo 1. Drugim typem jest urna z ponumerowanymi kartkami lub kulkami. Urnę taką będziemy nazywali generatorem liczb losowych o rozkładzie równomiernym. Użycie generatorów fizycznych, całkowicie losowych było by najlepszą metodą na kryptoanalityków, ponieważ większość ataków odbywa się właśnie na generatory i to one powinny być szczególnie bezpieczne. Jednak zastosowanie generatorów fizycznych nie jest możliwe z powodu tego, że nie udałoby się odszyfrować tak zaszyfrowanej wiadomośći.

Generatory programowe odgrywają dominującą role, chociaż w świecie rzeczywistym można znaleźć bardzo wiele generatorów fizycznych, których problem łączy się z brakiem stabilności. Generatory programowe są programami wykonanymi dla maszyn cyfrowych. Ten typ generatorów jest szczególnie często wykorzystywany w kryptografii szczególnie w szyfrowaniu informacji z powodu ich stabilności. Generatory programowe przyjmują następująca postać:

$$x_{n-1} = a_0 x_n + a_1 x_{n-1} + \ldots + a_k x_{n-k} + b \pmod{M},$$

wszystkie wartości są tutaj liczbami całkowitymi z przedziału (0 – M), są to tak zwane generatory liniowe. Szczególny przykładem generatorów liniowych, są generatory multiplikatywne, które wyrażają się wzorem:

$x_{n+1} = cx_n \pmod{M}$,

a generatora mieszanego:

$x_{n+1} = ax_n + b \pmod{M}$.

Każda liczba w ciągu (x_n) powtarza się dokładnie jeden raz, a ciąg x_n jest zbudowany ze skończonej liczby różnych znaków (liczb) n = 0,1,2,…. Długość takiego ciągu jest określona za pomocą liczby - K, która nazywa się okresem tego ciągu. Jeżeli w danym ciągu spełniony jest warunek K = M, to każda liczba ciągu pojawia się w określonej kolejności i ciąg ten staje się permutacją liczb (0, 1, 2, … , M – 1). Jeżeli K < M, to pojawiają się tylko niektóre liczby ciągu. Koniecznym przypadkiem jest zjawisko, gdzie K = M spełnienie tego warunku stało się koniecznością w czasie szyfrowania informacji, bo wtedy otrzymujemy określona permutacja ciągu.

ZASTOSOWANIE GENERATORA MULTIPLIKATYWNEGO

W metodzie ZT-UNITAKOD wymagana jest możliwość generowania permutacji ciągów (x_n), dla K = M, w których każda liczba (znak) pojawi się wyłącznie tylko jeden jedyny raz. W ZT-UNITAKOD zastosowano tablice kodów kryptograficznych, która ma możliwość szyfrowania informacji w oparciu o trzy rodzaje tablic:

- zerową,
- wyjściową,
- szyfrową.

Każda z tych tablic wymaga utworzenia odpowiedniej generacji za pomocą określonego generatora. Tablica kodów kryptograficznych może mieć rozmiary:

- 256 znaków stanowiących maksymalną liczbę możliwych kombinacji kodu ASCII.
- 145 znaków stanowiących wersję międzynarodowego kodu KOI-8.
- 60 znaków stosowanych w maszynach cyfrowych.
- 34 znaki, w tym 24 litery alfabetu łacińskiego oraz 10 cyfr.

Wartość liczbowa N mieści się w przedziale (34 – 256).

Ponadto w metodzie ZT-UNITAKOD jest spełniony warunek generowania odpowiedniej liczby permutacji wyrażonej wzorem:

$(W_z!)^W$

gdzie,

W_z - jest liczbą znaków z przedziału (34 – 256) zastosowaną w tej metodzie,

W - jest liczbą wierszy w tablicy.

Dla $W_z = W = 256$,otrzymujemy maksymalna liczbę permutacji równą $P = (256!)^{256}$. Ten warunek jest podstawowym warunkiem decydującym o wyborze określonego generatora permutacji.

Generator permutacji zastosowany w metodzie ZT-UNITAKOD opiera się o zasady generatora multiplikatywnego, dla którego dokonano określonych zmian:

Stała wartość „c" jest bardzo istotnym elementem wzoru generatora multiplikatywnego, multiplikatywnego jej wartość jest praktycznie obojętna, ale musza to być liczby całkowite z przedziału od 0 do M.

Przykładem może być rozpatrzenie wyników otrzymanych dla c = 4 i ciągu liczb: 1, 2, 3, …, 10. Wartość N = 10. po podstawieniu tych danych do wzoru otrzymujemy:

$$x_1 = 4 * 1 \pmod{10} = 4$$
$$x_2 = 4 * 2 \pmod{10} = 8$$
$$x_3 = 4 * 3 \pmod{10} = 2$$
$$x_4 = 4 * 4 \pmod{10} = 6$$
$$x_5 = 4 * 5 \pmod{10} = 0$$
$$x_6 = 4 * 6 \pmod{10} = 4$$
$$x_7 = 4 * 7 \pmod{10} = 8$$
$$x_8 = 4 * 8 \pmod{10} = 2$$
$$x_9 = 4 * 9 \pmod{10} = 6$$
$$x_{10} = 4 * 10 \pmod{10} = 0.$$

wynik:

1. $x_1 = x_6 = 4$,
2. $x_2 = x_7 = 8$,
3. $x_3 = x_8 = 2$,
4. $x_4 = x_9 = 6$,
5. $x_5 = x_{10} = 0$.

Nie otrzymaliśmy natomiast liczb 1, 3, 4, 7, 9, chociaż wystepują one w ciągu x_n.

Podobnie jest dla c = 6 itd. Można więc wyciągnąć wnioski:

- Otrzymujemy liczby parzyste zakończone okresem.
- Nie otrzymujemy liczb nieparzystych
- Nie został spełniony warunek pojawienia się każdej liczby tylko jeden jedyny raz

Analiza ta jednoznacznie wskazuje nam na to, że tak zaprogramowany generator nie może być stosowany dla parzystej wartości „c".

Rozpatrując wartości nieparzyste dla liczb „c" np. 9 otrzymujemy:

9, 8, 7, 5, 4, 3, 2, 1, 0. Jest to układ malejący w kolejności od 9 do 0.

Dla wartości c = 15 otrzymujemy:

5, 0, 5, 0, 5, 0, 5, 0, 5, 0. Jeżeli teraz za M podstawimy 20 a wartości c pozostawimy bez zmian w wyniku otrzymamy: 15, 10, 5, 0, 15, 10, 5, 0, 15, 10.

We wszystkich tych układach otrzymaliśmy liczby zmieniające się w okresach. Jednak i tutaj nie otrzymaliśmy satysfakcjonującego nas wyniku, liczby dalej się powtarzają.

Rozpatrujemy, więc działanie generatora dla liczb pierwszych. Za wartość „c" podstawiamy kolejne liczby pierwsze; 3, 7, 11, 13, 17 i dokonujemy obliczeń dla M = 10, 20 oraz np. 24.

Przykład 1.

Ciąg x_n = 1, 2, 3, …, 10. M = 10, c = 3 otrzymujemy:

$$x_1 = 3 * 1 \pmod{10} = 3$$
$$x_2 = 3 * 2 \pmod{10} = 6$$
$$x_3 = 3 * 3 \pmod{10} = 9$$
$$x_4 = 3 * 4 \pmod{10} = 2$$
$$x_5 = 3 * 5 \pmod{10} = 5$$
$$x_6 = 3 * 6 \pmod{10} = 8$$
$$x_7 = 3 * 7 \pmod{10} = 1$$
$$x_8 = 3 * 8 \pmod{10} = 4$$
$$x_9 = 3 * 9 \pmod{10} = 7$$
$$x_{10} = 3 * 10 \pmod{10} = 0.$$

Ciąg x_n = 1, 2, 3, …, 10. M = 10, c = 7 otrzymujemy:

$$x_1 = 7 * 1 \pmod{10} = 7$$
$$x_2 = 7 * 2 \pmod{10} = 4$$
$$x_3 = 7 * 3 \pmod{10} = 1$$
$$x_4 = 7 * 4 \pmod{10} = 8$$
$$x_5 = 7 * 5 \pmod{10} = 5$$
$$x_6 = 7 * 6 \pmod{10} = 2$$
$$x_7 = 7 * 7 \pmod{10} = 9$$
$$x_8 = 7 * 8 \pmod{10} = 6$$
$$x_9 = 7 * 9 \pmod{10} = 3$$
$$x_{10} = 7 * 10 \pmod{10} = 0.$$

Ciąg x_n = 1, 2, 3, …, 10. M = 10, c = 11 otrzymujemy:

$$x_1 = 11 * 1 \pmod{10} = 1$$
$$x_2 = 11 * 2 \pmod{10} = 2$$
$$x_3 = 11 * 3 \pmod{10} = 3$$
$$x_4 = 11 * 4 \pmod{10} = 4$$

$$x_5 = 11 * 5 \pmod{10} = 5$$
$$x_6 = 11 * 6 \pmod{10} = 6$$
$$x_7 = 11 * 7 \pmod{10} = 7$$
$$x_8 = 11 * 8 \pmod{10} = 8$$
$$x_9 = 11 * 9 \pmod{10} = 9$$
$$x_{10} = 11 * 10 \pmod{10} = 0.$$

Ciąg $x_n = 1, 2, 3, \ldots, 10$. M = 10, c = 13 otrzymujemy:

$$x_1 = 13 * 1 \pmod{10} = 3$$
$$x_2 = 13 * 2 \pmod{10} = 6$$
$$x_3 = 13 * 3 \pmod{10} = 9$$
$$x_4 = 13 * 4 \pmod{10} = 2$$
$$x_5 = 13 * 5 \pmod{10} = 5$$
$$x_6 = 13 * 6 \pmod{10} = 8$$
$$x_7 = 13 * 7 \pmod{10} = 1$$
$$x_8 = 13 * 8 \pmod{10} = 4$$
$$x_9 = 13 * 9 \pmod{10} = 7$$
$$x_{10} = 13 * 10 \pmod{10} = 0.$$

Ciąg $x_n = 1, 2, 3, \ldots, 10$. M = 10, c = 17 otrzymujemy:

$$x_1 = 17 * 1 \pmod{10} = 7$$
$$x_2 = 17 * 2 \pmod{10} = 4$$
$$x_3 = 17 * 3 \pmod{10} = 1$$
$$x_4 = 17 * 4 \pmod{10} = 8$$
$$x_5 = 17 * 5 \pmod{10} = 5$$
$$x_6 = 17 * 6 \pmod{10} = 2$$
$$x_7 = 17 * 7 \pmod{10} = 9$$
$$x_8 = 17 * 8 \pmod{10} = 6$$
$$x_9 = 17 * 9 \pmod{10} = 3$$
$$x_{10} = 17 * 10 \pmod{10} = 0.$$

Analizując otrzymane wynik stwierdzamy, że wszystkie założenia zostały spełnione. Ciąg x_n przyjmuje zróżnicowana postać uzależnioną od wartości „c". Każda liczba ciągu jest prezentowana jeden raz a cały ciąg kończy się wartością zerowa.

Przykład 2.

Ciąg $x_n = 1, 2, 3, \ldots, 10$. M = 20, wartości c = 3, 7, 11, 13, 17. W tym przypadku otrzymujemy wartości ciągu, które zostały zaprezentowane tylko jeden raz dla wybranej wartości „c". Każdy ciąg kończy się zerem.

Przykład 3.

Ciąg x_n = 1, 2, 3, …, 10. M = 24, wartości c = 3, 7, 11, 13, 17.

Po przeanalizowaniu wyników dochodzimy do wniosku, że dla wartości c =3 generator nie spełnia początkowych warunków, obliczenia dla M = 60 potwierdzają to samo zjawisko, dlatego ten generator nie nadaje się do zastosowania w metodzie ZT-UNITAKOD. Metoda ZT-UNITAKOD postawiła nieco odmienne wymagania na generator permutacji, którego postać można wyrazić następującym zapisem matematycznym:

$P = CX_n \pmod M$

gdzie:

C – jest to określona liczba pierwsza

M – określona liczba znaków zastosowana w tablicy kodów

X_n – ciąg liczb całkowitych

Wymagania postawione generatorowi permutacji zastosowanemu w metodzie ZT-UNITAKOD można przedstawić w dwóch punktach:

1. Liczba M musi zawierać się w przedziale od 24 do 256 i musi być określoną wartością dla danego szyfru. Najczęściej są to wielkości związane z międzynarodowym kodem, mianowicie:
   - 24 – liczba znaków alfabetu
   - 34 – liczba znaków alfabetu i cyfr
   - 60 – minimalna liczba znaków zastosowanych w komputerach
   - 145 – międzynarodowa wersja kodu ośmiobitowego
   - 256 – maksymalna liczba możliwych kombinacji w kodzie międzynarodowym
2. Liczba C musi być określoną liczbą pierwszą spełniająca poniższe trzy warunki:
   - $(C * X_i) \bmod M \neq 0$ dla i = 1, 2, … , M -1
   - $(C * X_i) \bmod M = 0$ dla i = M
   - $(C * X_i) \bmod M \neq (C * X_j) \bmod M$, jeżeli $X_i \neq X_j$

Całość można opisać następującym wzorem:

$S_{wk} = (a_{xz} + b_{ij}) \bmod N$,

gdzie:

a_{xz} – tablica kodów kryptograficznych tworzona za pomocą generatora permutacji: $P = CX_n \pmod M$

N – liczba znaków zastosowanych w określonych zasadach szyfrowania.

Tylko tak opisany generator miał prawo pojawić się w metodzie ZT-UNITAKOD, ponieważ spełnia on wszystkie warunki założone w fazie projektowania metody.

Co prawda generatory programowe nie wybierają poszczególnych liczb w sposób całkowicie losowy, lecz pseudolosowy, ale tylko taki sposób wybierania liczb daje nam możliwość odszyfrowania ciągu szyfrowego.

1.7 ZASTOSOWANIE METODY JEDNORAZOWEGO KLUCZA DYNAMICZNEGO

Metody szyfrowania mogą być stosowana dla następujących przypadków;

1. Przy szyfrowaniu autonomicznym - na jednym komputerze
2. Przy szyfrowaniu na kierunku (jeden nadawca i jeden adresat)
3. Przy szyfrowaniu w sieci (jeden nadawca wielu adresatów)
4. Szyfrowanie „w locie" (możliwość szyfrowania baz danych w dowolnym czasie i wyszukiwania informacji zaszyfrowanych różnymi rodzajami szyfrów, czyli szyfrowanych w różnych czasach)
5. Sprzętowa realizacja szyfrowa

Metoda ZT-UNITAKOD może być stosowana:

1. Szyfrowaniu bezpośrednim
2. Jako szyfrująca nakładka programowa
3. Do szyfrowania „w locie"
4. Hardware

2. WYBÓR KRYTERIÓW I METOD OCENY JAKOŚCI SZYFROWANIA.

Wybór kryteriów do oceny metod szyfrowania zawsze rodzi pewne kontrowersje. Dla części użytkowników najważniejsza jest np. szybkość danej metody, dla kogoś innego może być to na przykład niezawodność, jaką ona gwarantuje a dla jeszcze kogoś innego może to być chociażby cena lub prostota w obsłudze. Wybór jest dość trudną sprawą i indywidualną danego analityka takiej metody.

Z powodu tego, że metody statyczne w szczególności metoda RSA, którą zaprezentowałem w mojej pracy jest zupełnie inna niż metoda ZT-UNITAKOD, nie sposób jest jednoznacznie podać wszystkie najlepsze kryteria, jakie powinny obie te metody spełniać. Dla mnie najważniejsze jest to by poszczególne metody cechowały się następującymi właściwościami:

1. Szybkość metody,
2. Bezpieczeństwo,
3. Prostota obsługi,
4. Niezawodność,
5. Niezależność.

Uważam, że tych sześć kryteriów, jakie wybrałem do analizy i porównania metod szyfrowania statycznego z nowatorską metodą dynamiczną w zupełności zaspokoi każdego chętnego sięgnąć i przeczytać moją pracę.

Rozumienie poszczególnych kryteriów:

a) Szybkość metody – pojęcie to jest dość względne, dla jednej osoby metoda szybka w sowim działaniu to ta, której czas pozwalający na zaszyfrowanie np. jednego megabajta informacji wynosi dwie sekundy a dla kogoś innego np. 0,003 sekundy. Dla mnie szybkość metody charakteryzuje czas rzędu setnych sekundy. Bardzo ważną rzeczą jest by prezentowane metody szyfrowania informacji były możliwe np. do szyfrowania mowy ludzkiej w telefonii cyfrowej.

b) Bezpieczeństwo – pod tym pojęciem rozumiemy wytrzymałość na złamanie przechwyconego szyfrogramu, lub samej metody. W metodach statycznych bezpieczeństwo metody ściśle było związane z mocą kryptograficzną danej metody, co w szczególności wiązało się z mocą kryptograficzną klucza zastosowanego w danej metodzie. Dynamiczna metoda ZT-UNITAKOD zmieniła sposób patrzenia na bezpieczeństwo danej metody, ponieważ nie stosuje się tu żadnych kluczy statycznych, co jednoznacznie zmienia sposób podejścia do bezpieczeństwa gwarantowanego w tej metodzie.

c) Prostota – to trzecie bardzo ważne kryterium oceny poszczególnych metod. Prostota w dzisiejszym świecie jest jedną z najważniejszych rzeczy, ponieważ gwarantuje ona w pewien sposób to, że ludzie mający możliwość korzystania z danej metody będą w rzeczywistości korzystać z niej. Na rynku istnieje wiele nakładek na systemy operacyjne poprawiających bezpieczeństwo lub „łatających" pewne niedopatrzenia wynikłe podczas używania tych metod. Mogą to być niedopatrzenia powstałe już w fazie powstawania projektów jak i te, które popełniają programiści. Jednak bardzo nieliczna grupa ludzi stosuje tego typu rozwiązania, ponieważ często wiążą się one z pewnym poświęceniem swojego czasu na instalacje, później na aktualizowanie itd. Prostota danej metody gwarantuje nam przyjęcie jej przez szerokie grono ludzi korzystających ze sprzętu informatycznego.

d) Niezawodność – niezawodność jest tym ważniejsza, że podczas szyfrowania informacji nie może być mowy o możliwości np. opuszczenia części szyfrowanego tekstu jawnego (jednego z jego bloków), lub „wykrzaczenia" się metody podczas przesyłania na przykład uwierzytelnienie do banku.

e) Niezależność metody - ten aspekt staje się podstawą w szyfrowaniu informacji. Metoda nie powinna zależeć od decydującej roli człowieka. Metoda powinna szyfrować dowolny język świata. Powinna też móc szyfrować schematy, obrazy, zdjęcia satelitarne, bazy danych itd. To wszystko wskazuje na niezależność metody. Metoda powinna być też tak skonstruowana, żeby nawet sam autor nie mając szyfrogram i część tekstu jawnego nie umiał rozszyfrować takiej metody. Spełnienie wszystkich tych zależności daje nam podstawy do sądzenia, że dana metoda jest naprawdę niezależna.