DZIAŁ NAUKOWY

MATEMATYKA STOSOWANA 4, 2003

Zbigniew Kotulski (Warszawa)

Budowanie szyfrów blokowych: nowe możliwości

Wstęp. We współczesnej kryptografii wszelkie informacje podlegające ochronie to ciągi binarne, czyli ciągi zer i jedynek. Już to pierwsze zdanie pracy można uznać za mało precyzyjne, istnieje bowiem na przykład kryptografia wizualna, w której obiektami zainteresowania sa obrazy, my jednak w tej pracy będziemy się poruszać w obrębie pewnego modelu, pomijając całe bogactwo obiektów i metod występujących w kryptografii. Tak więc wiadomościami będą ciągi binarne (o ustalonej lub nieograniczonej długości), a ich ochrona będzie polegała na utajnieniu zawartości, czyli sprawieniu, że nikt niepowołany nie będzie mógł poznać treści tych wiadomości (to również uproszczenie, ponieważ poufność ciągu to tylko jedna z wielu jego cech, które można chronić, por. [6], [16]). Mamy zatem ciąg binarny postaci

$$\{01100111010001001\ldots\}, \tag{1}$$

będący tekstem jawnym (odkrytym). Po wykonaniu odpowiedniej operacji szyfrowania nasz ciąg binarny staje się *szyfrogramem*, czyli innym ciągiem bitów mającym tę własność, że na jego podstawie nie można rozpoznać znaczenia tekstu odkrytego. Zatem nasza wiadomość zaszyfrowana (szyfrogram) jest to także ciąg binarny, tym razem postaci (zakładamy tu, że tekst odkryty i szyfrogram są tej samej długości)

$$\{00110010100100110\ldots\}. \tag{2}$$

Jeżeli teraz zapiszemy oba teksty jeden nad drugim w taki sposób, że pod każdym kolejnym bitem tekstu odkrytego (1) znajdzie się dokładnie jeden, kolejny bit szyfrogramu (2), to możemy stwierdzić, że sposób szyfrowania wiadomości jest zadany regułą, która jest także ciągiem bitów o długości równej długości tekstu odkrytego (lub, co oznacza to samo, długości szyfrogramu):

$$\begin{aligned} &\{01100111010001001\ldots\} \text{ — tekst jawny} \\ &\{00110010100100110\ldots\} \text{ — tekst zaszyfrowany} \\ &\{01010101110101111\ldots\} \text{ — strumień klucza} \end{aligned} \tag{3}$$

Powyższa reguła (nazwiemy ją *strumieniem klucza*) jest utworzona w taki sposób, że jedynka oznacza, iż bit tekstu jawnego został zamieniony na przeciwny podczas szyfrowania, natomiast zero oznacza, że pozostał bez zmiany. Opisany tu sposób szyfrowania nosi nazwę *szyfru strumieniowego*.

Formalna definicja szyfru strumieniowego wymaga podania sposobu szyfrowania i sposobu odszyfrowania wiadomości. Zwykle jest ona bardzo podobna do sformułowanego wyżej intuicyjnego opisu. Niech zatem $P = \{p_1, p_2, \ldots\}$ będzie tekstem odkrytym, $K = \{k_1, k_2, \ldots\}$ strumieniem klucza, natomiast $C = \{c_1, c_2, \ldots\}$ — szyfrogramem. Jeżeli zatem chcemy, mając dany tajny klucz K, zaszyfrować (a następnie odszyfrować) wiadomość, musimy wykonać kolejno dla odpowiednich par bitów następujące operacje:

$$\begin{aligned} &c_i = p_i \oplus k_i \text{ — dla szyfrowania,} \\ &p_i = c_i \oplus k_i \text{ — dla odszyfrowywania,} \end{aligned} \qquad i = 1, 2, \ldots \tag{4}$$

Operacja binarna oznaczona symbolem $\oplus$ to logiczna operacja różnicy symetrycznej (w interpretacji bitów jako wartości logicznych), potocznie nazywana XOR. Jej definicję podaje następująca tabela:

$$\begin{aligned} &0 \oplus 0 = 0, \quad 1 \oplus 0 = 1, \\ &0 \oplus 1 = 1, \quad 1 \oplus 1 = 0. \end{aligned} \tag{5}$$

Sam fakt opisania transformacji bitów podany w (4) nie wystarcza, by uzyskać szyfr, a zatem operację utajnienia tekstu odkrytego. Niezbędnym warunkiem jest tu zagwarantowanie odporności na złamanie tego szyfru, czyli nielegalne ujawnienie treści wiadomości lub wartości tajnego klucza na podstawie znajomości bitów szyfrogramu. W przypadku szyfru strumieniowego wymagania gwarantujące jego bezpieczeństwo wynikają z pierwszego twierdzenia informacyjnego Shannona [21]. Wnioskiem z tego twierdzenia jest, że szyfr (nie tylko strumieniowy) jest *informacyjnie bezpieczny* (to znaczy szyfrogram nie dostarcza żadnej informacji niezbędnej do złamania szyfru), gdy entropia klucza jest nie mniejsza od entropii szyfrowanej wiadomości. W szczególnym przypadku naszego szyfru binarnego warunek ten sprowadza się do wymogu, aby ciąg bitów użyty jako strumień klucza był *białym szumem binarnym*, tzn. by kolejne bity klucza z równym prawdopodobieństwem przyjmowały wartości 0 i 1 oraz by były od siebie statystycznie niezależne.

Powyższe sformułowanie wymogu bezpieczeństwa szyfru niesie w sobie dwie informacje. Jedna jest dobra: szyfr strumieniowy (inaczej: szyfr Ver-

nama) może być szyfrem bezpiecznym (jest to jedyny znany szyfr o takiej własności). Druga jest gorsza: ciąg bitów będący kluczem musi być ciągiem czysto losowym, co w praktyce oznacza, że nie istnieje reguła opisująca ten ciąg krótsza od samego ciągu [3], [10]. Zatem w praktyce przed operacjami szyfrowania i odszyfrowania należy taki ciąg dostarczyć (w bezpieczny sposób) do obu komunikujących się stron, co może być równoważne bezpiecznemu przesłaniu samej wiadomości, ponieważ długość klucza jest taka sama, jak długość tekstu odkrytego.

Jak zatem uniknąć tego problemu? W praktyce najczęściej stosowane są dwie metody. Jedna polega na równoczesnemu wytwarzaniu ciągów bitów u nadawcy i u odbiorcy wiadomości z wykorzystaniem odpowiednio zaprojektowanych algorytmów matematycznych, gwarantujących pseudolosowe (czyli nieodróżnialne od losowych) własności tych ciągów. Metoda ta była już przedstawiona w [15]. Inna metoda bezpiecznego przesyłania wiadomości polega na wykorzystaniu szyfrów blokowych. Ta właśnie metoda jest tematem dalszych rozważań niniejszej pracy.

1. Szyfry blokowe. W szyfrach blokowych strumień informacji jest dzielony na bloki o skończonej długości. Zwykle (w różnych praktycznie stosowanych szyfrach) taki blok ma $64, 128, 192$ lub 256 bitów. Szyfrowanie polega teraz na takim przekształceniu bloku bitów (tekstu odkrytego) w inny blok bitów o takiej samej długości (szyfrogram), aby na podstawie znajomości samego szyfrogramu nie można było ani uzyskać poprawnej wartości bitów (całego bloku lub kilku bitów) tekstu odkrytego, ani ustalić sposobu, w jaki dokonano tego przekształcenia. Pozostaje teraz odpowiedzieć na pytanie: Jakie cechy musi mieć takie przekształcenie (szyfr blokowy), aby można uznać, że jest ono bezpieczne? Próba odpowiedzi na to pytanie jest celem tej pracy.

Zacznijmy od sformułowania problemu. *Szyfr blokowy* jest odwzorowaniem postaci

$$F_K(\cdot) : \{0,1\}^l \to \{0,1\}^l, \tag{6}$$

czyli odwzorowującym blok bitów o długości l na blok bitów o takiej samej długości, zależnym od parametru K (tajnego klucza), będącego również ciągiem bitów. Zakładamy przy tym, że odwzorowanie to jest wzajemnie jednoznaczne dla każdego ustalonego K, a zatem posiada odwzorowanie odwrotne $F_K^{-1}(\cdot)$. Jak widać z (6), parametrami szyfru są dwie liczby:

- długość bloku wiadomości/kryptogramu — l bitów;
- długość klucza — k bitów.

W tym miejscu należy wyjaśnić, że współczesne konstrukcje szyfrów (poza zastosowaniami specjalnymi) swoje bezpieczeństwo opierają na za-

sadzie sformułowanej jeszcze w XIX wieku przez Kerckhoffa ([1]), że funkcja $F_{\cdot}(\cdot)$ wprowadzona w (6) jest znana, a bezpieczeństwo szyfru oparte jest na zachowaniu poufności klucza K — znajomość klucza jest warunkiem wykonania zarówno operacji szyfrowania, jak i odszyfrowania. Z kolei upublicznienie postaci odwzorowania szyfrującego sprawia, że staje się ono przedmiotem analizy prowadzonej przez całą społeczność kryptologów, co gwarantuje wykrycie i ujawnienie wszelkich ewentualnych słabości szyfru.

Zanim zastanowimy się nad własnościami, które powinna mieć funkcja szyfrująca, aby móc spełnić wymogi bezpieczeństwa, omówimy sposób wykorzystania tej funkcji do szyfrowania długich tekstów. Zarówno szyfrowanie, jak i odszyfrowywanie przebiega w kilku krokach. W przypadku szyfrowania są to:

- Wybór i uzgodnienie klucza sesyjnego K (klucz jest ustalany dla każdej sesji).
- Podział tekstu jawnego na bloki o długości l (z uzupełnieniem ostatniego bloku do pełnej długości w sposób odróżniający bity uzupełnienia od wiadomości), $\{P_1, \ldots, P_m\}$.
- Szyfrowanie polega na wykonaniu funkcji szyfru $F_K(\cdot)$ dla każdego bloku tekstu jawnego. Operacja ta jest wykonywana w odpowiednim trybie pracy szyfru. Nie będziemy się jednak tutaj zajmować trybami pracy szyfrów blokowych, odsyłając zainteresowanych czytelników do literatury (por. np. [6]).

Odbiorca otrzymuje ciąg bitów, który składa się z zaszyfrowanych bloków $\{C_1, \ldots, C_m\}$. Odszyfrowywanie polega na wykonaniu funkcji odwrotnej $F_K^{-1}(\cdot)$ dla każdego bloku szyfrogramu (w takim samym trybie pracy, jak w trakcie szyfrowania).

Po tym formalnym opisie operacji szyfrowania i odszyfrowywania należy odpowiedzieć sobie na pytanie: Jakie wymagania musi spełniać taki szyfr (czyli funkcja F), aby można go było uznać za bezpieczny? Przypomnijmy jeszcze raz formalne wymagania dotyczące szyfru. Tak więc funkcja F jest powszechnie znana. Dla legalnego użytkownika, algorytm szyfrowania i odszyfrowania (obliczenia wartości funkcji $F_K(\cdot)$ i funkcji odwrotnej $F_K^{-1}(\cdot)$) powinien być łatwy do wykonania, o ile znany jest mu tajny klucz K. Z kolei dla potencjalnych napastników algorytm powinien być trudny do złamania, to znaczy trudne powinno być:

- wykonanie operacji zaszyfrowania lub odszyfrowania, gdy klucz nie jest znany,
- ujawnienie tajnego klucza (na podstawie znajomości pary tekst odkryty — tekst zaszyfrowany),

([1]) Zasady bezpieczeństwa szyfrów sformułowane przez Kerckhoffa były jednym z przewodnich tematów zaproszonego wykładu na EUROCRYPT'2003, por. [22].

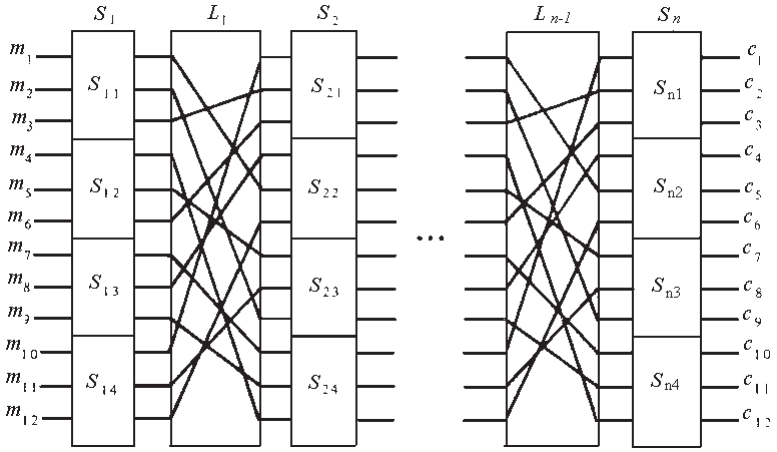
- uzyskanie jakiejkolwiek informacji o zaszyfrowanym tekście lub tajnym kluczu na podstawie znajomości tekstu zaszyfrowanego.

Te ogólnie sformułowane wymagania bezpieczeństwa sa realizowane przez odpowiednie (w domyśle: dostatecznie skomplikowane) skonstruowanie funkcji $F_K(\cdot)$. Według powszechnie przyjętej zasady budowy szyfru blokowego, funkcja F jest n-krotnym złożeniem pewnego przekształcenia $f_{K_i}(\cdot), i = 1, \ldots, n$, nazywanego *funkcją rundową* lub *funkcją cyklu*. W tym oznaczeniu wielkości (parametry odwzorowania) K_i są tak zwanymi *kluczami rundowymi*, otrzymanymi z klucza K w algorytmie generacji kluczy rundowych. Przyjęte jest, że wprowadzona powyżej funkcja $f_{K_i}(\cdot)$, $i = 1, \ldots, n$, działająca odpowiednio na blok wejściowy P_i, $i = 1, \ldots, n$, jest złożeniem szeregu prostych operacji:

- nieliniowych (oznaczmy je przez S_i), to znaczy podstawień, czyli operacji zastępowania ciągów bitów podbloków bloku P_i innymi ciągami bitów,

oraz

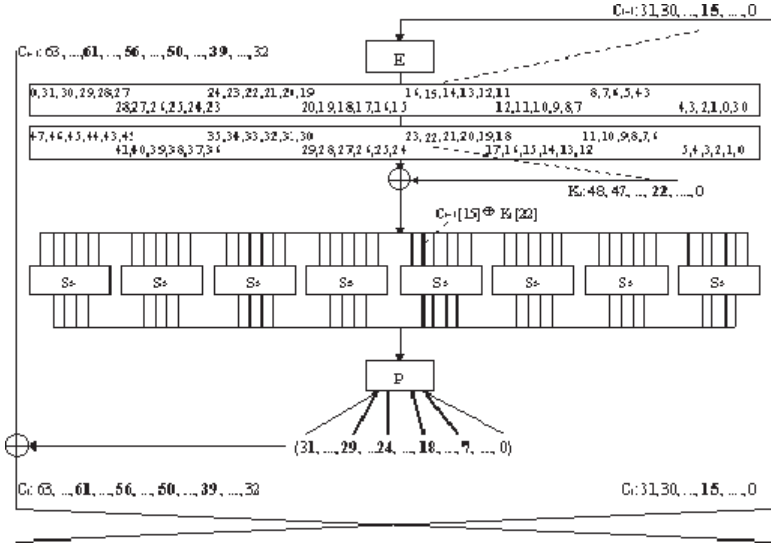
- liniowych (oznaczmy je przez L_i), to znaczy permutacji bitów bloku wejściowego P_i.



Rys. 1. Schemat prostego szyfru

Jak widać z zaprezentowanego opisu, w szyfrach blokowych skomplikowaną funkcję $F_K(\cdot)$ uzyskuje się w wyniku wielokrotnego (m-krotnego) złożenia prostej funkcji $f_{K_i}(\cdot)$, będącej z kolei odpowiednio dobranym złożeniem podstawień i permutacji bitów w blokach. Schemat takiego szyfru blokowego przedstawiono na rysunku 1. W zaprezentowanej konstrukcji dokonano dodatkowego uproszczenia polegającego na podzieleniu w warstwie

operacji podstawienia dużych bloków wejściowych na kilka równoległych operacji podstawienia mniejszych bloków (podbloków). Operacje działające na podblokach oznaczono jako S_{ij}. Zazwyczaj nazywane są one *skrzynkami podstawieniowymi* (S-*box*) ([2]). Powyższa dekompozycja warstwy podstawień jest dokonana w celu ułatwienia możliwości zaprogramowania operacji nieliniowej oraz łatwiejszej analizy jej bezpieczeństwa.



Rys. 2. Schemat rundy DES

Oprócz odpowiedniego doboru rodzaju operacji liniowych i nieliniowych w każdej rundzie szyfru, ważne jest ustalenie liczby wykonywanych rund. Więcej rund to dłuższy czas wykonywania operacji, ale i większe (w zasadzie) bezpieczeństwo. W powszechnie znanych szyfrach liczba ta waha się od $m = 6$ (dla szyfru Hierocrypt-3 o długości bloku 128 bitów) przez $m = 16$ (dla DES) do $m = 32$ (dla algorytmu Serpent) ([3]). O ile zapis iteracyjny szyfru blokowego wydaje sie prosty i łatwy do prześledzenia, to próba określenia zależności między bitami bloku wejściowego całego szyfru a bitami jego bloku wyjściowego jest niezmiernie trudna. Na rysunku 2 (por. [24], [8]) przedstawiono taką zależność, dla ustalonej wartości klucza, dla jednej rundy algorytmu DES. Jak widać, taki schemat szyfru blokowego pozwala śledzić wpływ każdego z bitów na wejściu (rundy, skrzynki podsta-

([2]) Na rysunku 1 bity bloku wejścia pierwszej rundy oznaczono jako m_j, $j = 1, \ldots, 12$ (od słowa *message* — wiadomość), natomiast bity wyjścia ostatniej rundy jako c_j, $j = 1, \ldots, 12$ (od słowa *ciphertext* — kryptogram).

([3]) Wyjątkiem jest tu szyfr blokowy Godsave, który ma tylko jedną rundę z bardzo skomplikowaną definicją przekształcenia rundowego.

wieniowej, innej operacji) na bity wyjścia. Przy uwzględnieniu wielu iteracji (operacji szyfru) jest to jednak niezmiernie trudne, praktycznie niewykonalne.

Jak zatem widzimy, aby szyfr był bezpieczny, powinien on w odpowiedni (czytaj: dostatecznie skomplikowany) sposób przekształcać bity wiadomości (wejścia) na bity szyfrogramu (wyjścia). Niestety, sformułowanie „dostatecznie skomplikowany" nie dostarcza informacji, jak dokładnie zapisać operacje składowe szyfru. Ponadto, w celu zapewnienia odwracalności szyfrowania, przekształcenia bloków bitów wykorzystane do budowy kryptosystemu powinny być wzajemnie jednoznaczne, co znacznie ogranicza swobodę działań konstruktora. W praktyce wymagania dotyczące szyfru, to znaczy rodzaj zastosowanych w nim przekształceń, ich kolejność i liczba są przyjmowane przez autorów algorytmów na podstawie rozważań intuicyjnych. W dalszym ciagu pracy przedstawimy takie intuicyjne zasady bezpieczeństwa szyfrów blokowych.

2. Wymagania konstrukcyjne w stosunku do szyfrów blokowych. Własności, jakie powinny mieć odwzorowania realizujące szyfrowanie blokowe, zazwyczaj podawane są jako pary cech, wzajemnie się uzupełniających. Dlaczego pary? Po pierwsze dlatego, że szyfr jest pewną kompozycją dwóch rodzajów operacji: permutacji i podstawień. Po drugie, również dlatego, że szyfrowanie musi ukryć w kryptogramie informację o dwóch obiektach: bloku tekstu odkrytego i bloku bitów klucza. Omówimy teraz trzy najczęściej wykorzystywane pary własności przekształceń szyfrujących mających zagwarantować ich bezpieczeństwo (por. [18], [19]).

2.1. *Mieszanie i rozpraszanie.* Pierwsza para własności to mieszanie i rozpraszanie. Koncepcję szyfru jako *przekształcenia mieszającego* wprowadził Shannon. Mówimy, że dane przekształcenie ma tę własność, gdy losowo i równomiernie rozprowadza wiadomości zapisane tekstem jawnym po zbiorze wszystkich możliwych wiadomości zaszyfrowanych. Oznacza to na przykład, że jeżeli podzielimy przestrzeń szyfrogramów w dowolny sposób na dwie równoliczne części, to szyfrogramy uzyskane z kolejno losowanych w sposób niezależny tekstów odkrytych będą z równym prawdopodobieństwem należeć do każdego z podzbiorów podziału. Przekształcenia mieszające można zrealizować przez odpowiednio dobrane sekwencje permutacji i podstawień.

Druga z wymienionych własności, *rozpraszanie*, oznacza, że bity króre znajdują się w sąsiedztwie przed wejściem do rundy (na przykład, w jednej skrzynce podstawieniowej), po wyjściu z tej rundy wpływają na bity odległe od siebie (na przykład, należące do różnych skrzynek w następnej rundzie). W wyniku wielokrotnej iteracji takich rund uzyskujemy efekt, że każdy bit

wejścia szyfru (tekstu odkrytego) wpływa na wiele bitów wyjścia tego szyfru (szyfrogramu).

2.2. *Lawinowość i zupełność.* Kolejna para uzupełniających się własności gwarantujących bezpieczeństwo szyfru blokowego to lawinowość i zupełność. Koncepcja *lawinowości* wymaga, żeby zmiana jednego bitu na wejściu rundy wywoływała zmianę co najmniej dwóch bitów na wyjściu tej rundy. Z tego wynika, że zmiana jednego bitu na wejściu szyfru, w wyniku iteracji, powinna wywołać „lawinowe" zmiany na wyjściu całego algorytmu. Precyzując to wymaganie, przyjmujemy, że zmiana jednego bitu na wejściu powinna wywołać, średnio, zmianę połowy bitów na wyjściu szyfru.

Z kolei *zupełność* algorytmu oznacza, że każdy bit bloku wyjściowego jest bardzo skomplikowaną funkcją wszystkich bitów bloku wejściowego. Inaczej sformułowana, ta własność oznacza, że zawsze istnieje taki stan bloku wejściowego, że zmiana dowolnie wybranego bitu wejścia spowoduje zmianę wskazanego bitu wyjścia szyfru.

2.3. *Dyfuzja i konfuzja.* Ostatnia z omawianych tu par własności to dyfuzja i konfuzja. Własność *dyfuzji* oznacza rozmycie wszelkich związków pomiędzy bitami tekstu jawnego lub klucza równomiernie w całym bloku. Ma to na celu uniemożliwienie znalezienia związków statystycznych między tymi bitami na podstawie obserwacji kryptogramu. Dyfuzja jest zazwyczaj realizowana przez permutacje (transpozycje) bitów w bloku, a zatem operacje liniowe.

Konfuzja ma na celu maksymalne wymieszanie bitów bloku klucza z bitami bloku tekstu szyfrowanego i uczynienie ich związku bardzo skomplikowanym. Konfuzja jest osiągana zazwyczaj przez operacje podstawiania, czyli odwzorowania nieliniowe.

Każda runda szyfru powinna wprowadzać do algorytmu zarówno dyfuzję, jak i konfuzję. W przeciwieństwie do poprzednio omawianych własności, dyfuzja i konfuzja mogą być wyrażone ilościowo za pomocą odpowiednio zdefiniowanych prawdopodobieństw aproksymacji różnicowej DP i aproksymacji liniowej LP.

Jak powiedzieliśmy, miarą stopnia dyfuzji jest prawdopodobieństwo aproksymacji różnicowej DP. Aby zdefiniować to prawdopodobieństwo, przedstawmy proces szyfrowania F jako odwzorowanie z przestrzeni tekstów odkrytych P do przestrzeni szyfrogramów C. Zakładamy przy tym, że obie te przestrzenie są jednakowe (oznaczmy je jako X) i każda z nich jest zbiorem ciągów binarnych o długości l. Dla uproszczenia zapisu elementy tej przestrzeni możemy oznaczać jako liczby naturalne z zakresu od 0 do 2^l-1 (każda taka liczba w postaci binarnej, uzupełniona na początku zerami do długości l bitów, jest elementem tej przestrzeni). W przestrzeni tej możemy wprowadzić strukturę przestrzeni liniowej nad ciałem $GF(2)$ z operacją dodawania

$\oplus$ polegającą na wykonaniu dodawania bit po bicie modulo 2 (operacja binarna XOR). Mamy zatem (w odwzorowaniu nie uwzględniamy klucza, co oznacza, że traktujemy go jako wartość stałą w obliczeniach):

$$F : P \to C, \qquad P \equiv C \equiv \mathrm{X} = \{0, 1, 2, \ldots, 2^l - 1\}. \tag{7}$$

Prawdopodobieństwo aproksymacji różnicowej DP jest zdefiniowane jako

$$DP = \max_{\Delta X \neq 0,\, \Delta Y} P(\Delta Y | \Delta X), \tag{8}$$

gdzie

$$P(\Delta Y | \Delta X) = \frac{\#\{X \in \mathrm{X} \mid F(X) \oplus F(X \oplus \Delta X) = \Delta Y\}}{2^l}. \tag{9}$$

W równaniach (8) i (9) ΔY i ΔX oznaczają dowolne bloki l bitów, czyli elementy przestrzeni X, znak # w równaniu (9) oznacza liczebność zbioru elementów (spełniających warunek opisany w nawiasie), natomiast w równaniu (8) maksimum jest obliczane dla wszystkich wartości ΔY i $\Delta X \neq 0$ należących do przestrzeni X.

Zdefiniowane w powyższy sposób prawdopodobieństwo DP ma szereg własności. Oto najważniejsze z nich:

1. Dla wszystkich ΔY i $\Delta X \neq 0$ należących do X spełniony jest warunek

$$\frac{1}{2^{l-1}} \leq P(\Delta Y | \Delta X) \leq 1. \tag{10}$$

2. Warunek

$$P(\Delta Y | \Delta X) = 1 \tag{11}$$

oznacza, że istnieje takie ΔX, które jest przez odwzorowanie F zawsze przekształcane w ΔY.

3. Warunek

$$P(\Delta Y | \Delta X) = \frac{1}{2^{l-1}} \tag{12}$$

oznacza, że dla danego ΔX różnica ΔY ma rozkład równomierny na przestrzeni X.

4. DP jest miarą różnicowej jednorodności odwzorowania F. Jest to maksymalne prawdopodobieństwo tego, że uzyskamy na wyjściu różnicę ΔY, gdy na wejście podamy różnicę ΔX.

5. Jeżeli DP maleje, to rośnie złożoność ataku różnicowego.

W podobny sposób można wprowadzić miarę nieliniowości (konfuzji) przekształcenia, czyli *prawdopodobieństwo aproksymacji liniowej* LP. Jest ono zdefiniowane jako

$$LP = \max_{a,b \neq 0} (2P_{a,b} - 1)^2, \tag{13}$$

gdzie z kolei prawdopodobieństwo $P_{a,b}$ jest zdefiniowane jako

$$P_{a,b} = \frac{\#\{X \in \mathrm{X} \mid X \bullet a = F(X) \bullet b\}}{2^l} \tag{14}$$

dla wszystkich $a, b \in \mathrm{X}\backslash\{0\}$. Elementy przestrzeni X oznaczone przez a i b określają w (14) maski nałożone na bloki odpowiednio X i $F(X)$. Maska taka jest oznaczona symbolem $\bullet$, a zdefiniowana jako

$$X \bullet a = \bigoplus_{1} X_i a_i, \tag{15}$$

to znaczy określa ona, które bity bloku zostają uwzględnione w związku liniowym wykorzystywanym w (14).

Również prawdopodobieństwo LP ma szereg własności.

1. Oznaczmy przez N liczbę tych $X \in \mathrm{X}$, które spełniają związek liniowy

$$X \bullet a = F(X) \bullet b, \tag{16}$$

to znaczy

$$N = \#\{X \in \mathrm{X} \mid X \bullet a = F(X) \bullet b\}. \tag{17}$$

Wówczas

$$0 \le N \le 2^l. \tag{18}$$

2. Oznaczmy:

$$A = a \bullet X, \qquad B = b \bullet F(X). \tag{19}$$

W dwóch skrajnych przypadkach uwzględnionych w ograniczeniu (18) wielkości (19) spełniają implikacje:

$$\begin{aligned} N = 2^l &\Rightarrow A \oplus B = 0 \ \ \forall X \ \in \mathrm{X}, \\ N = 0 \ &\Rightarrow A \oplus B = 1 \ \ \forall X \ \in \mathrm{X}. \end{aligned} \tag{20}$$

W obu tych przypadkach $LP = 1$.

3. Jeśli $N = 2^{l-1}$, to implikacja

$$N = 2^{l-1} \Rightarrow A \oplus B = 0 \tag{21}$$

jest spełniona z prawdopodobieństwem $1/2$. W tym przypadku $LP = 0$.

4. Prawdopodobieństwo liniowej aproksymacji LP jest kwadratem maksymalnego niezrównoważenia następującego zdarzenia: „Parzystość wejściowych bitów wskazanych przez maskę a jest równa parzystości bitów wyjściowych wskazanych przez maskę b".

5. Jeżeli LP maleje, to rośnie złożoność ataku liniowego.

Należałoby jeszcze wyjaśnić, co to są ataki różnicowy i liniowy. Oba ataki polegają na zmniejszeniu liczby sprawdzeń kluczy (dla prawidłowej pary tekst odkryty–tekst zaszyfrowany, w celu dobrania jego właściwej wartości) dzięki wykorzystaniu pewnych własności szyfru. A zatem, *atak różnicowy* korzysta z faktu, że może istnieć wiele par tekstów odkrytych róż-

niących się o zadaną różnicę ΔX (tutaj różnica to XOR tekstów), których kryptogramy różnią się o znaną różnicę ΔY. Mówimy wówczas, że szyfr ma *charakterystykę* z pewnym prawdopodobieństwem (równym proporcji liczby par tekstów mających powyższą własność do liczby wszystkich par tekstów odkrytych o zadanej różnicy). Z kolei *atak liniowy* wykorzystuje fakt, że dla danego odwzorowania F mogą istnieć równania liniowe postaci (16) spełnione z prawdopodobieństwem znacznie różniącym się od 1/2. Jeśli szyfr nie jest odporny na któryś z tych rodzajów ataków, oznacza to, że wykorzystanie związków liniowych lub faktu istnienia charakterystyk mających duże prawdopodobieństwo pozwala znacznie zmniejszyć zakres przeszukiwania przestrzeni klucza w stosunku do ataku wyczerpującego (w ataku wyczerpującym należy średnio przeszukać 2^{k-1} wartości kluczy, aby natrafić na tę właściwą).

2.4. *Inne spojrzenie na bezpieczeństwo szyfrów.* W praktyce od projektowanego szyfru wymaga się, aby był on odporny na szczególne klasy ataków. Najogólniej mówiąc, powinna to być odporność na wszystkie znane klasy ataków. Problem w tym, co się stanie w przypadku pojawienia się nowych metod łamania szyfrów. I tak wymagane jest, aby konstruowany szyfr był odporny na analizę różnicową i analizę liniową. Pod tym kątem był optymalizowany algorytm szyfrowania Rijndael (por. [5]), zwycięzca rozpisanego przez NIST konkursu na nowy standard szyfrowania, następcę DESa. Tymczasem okazało się, że szyfr ten może nie być odporny na nowy atak, nazywany *atakiem algebraicznym*, polegający na zbudowaniu i rozwiązaniu układu równań nadmiarowych wielu zmiennych, wiążących ze sobą bity tekstu odkrytego, szyfrogramu i tajnego klucza (por. [4]). Zatem takie podejście, z pozoru skuteczne, nie gwarantuje wystarczającego bezpieczeństwa tworzonych algorytmów szyfrujących. Jak zatem w praktyce wygląda konstruowanie szyfrów?

Konstruowanie szyfru polega na wykonaniu dwóch kroków. Są to:

1. Budowa algorytmu zgodnie z zasadami podanymi w poprzednich rozdziałach.
2. Badanie działania skonstruowanego algorytmu metodami statystycznymi. Przyjmuje się, że ciąg bitów uzyskanych na wyjściu szyfru nie powinien wykazywać prawidłowości statystycznych pozwalających odróżnić go od szumu binarnego. Odpowiednie metody testowania zostały omówione w [15].

Powyższe rozważania wskazują, że choć szyfry blokowe są powszechnie stosowane i konstruowane są nowe ich rodzaje, nadal nie istnieją ścisłe metody pozwalające udowodnić, że zaprojektowany szyfr blokowy jest bezpieczny.

3. Analogia do układów chaotycznych. Czytelnicy, którzy znają teorię chaotycznych dyskretnych (w czasie) układów dynamicznych, łatwo zauważą, że pojęcia występujące w wymaganiach konstrukcyjnych dotyczących szyfrów blokowych są jakby przeniesione z tej teorii do kryptografii. Gdyby rzeczywiście udało się je tam zastosować, zyskalibyśmy wreszcie precyzyjną z matematycznego punktu widzenia metodę opisu własności szyfrów, a co za tym idzie — narzędzie do konstrukcji nowych algorytmów o z góry zadanej lub możliwej do ocenienia sile szyfrowania. Jednak nasuwającą się na pierwszy rzut oka wadą takiego podejścia jest fakt, że układy takie, choć dyskretne w czasie, operują na ciągłych przestrzeniach stanów, czyli przekształcają obiekty nie będące skończonymi ciągami bitów (jak już zauważyliśmy, szyfry blokowe są operacjami przekształcającymi ciąg bitów w ciąg bitów, oba o tej samej skończonej długości). Z kolei pozytywną wiadomością jest to, że układy dynamiczne, jako obiekty matematyczne, mają dobrze udokumentowane własności, których przydatność do celów kryptograficznych jest w sposób naturalny widoczna. Zanim pokażemy teraz te analogie, wprowadźmy podstawowe definicje i oznaczenia.

Dyskretny układ dynamiczny to para (X, φ), gdzie:

- X jest przestrzenią stanów (ogólnie — przestrzenią metryczną, np. w naszych przykładach — odcinkiem jednostkowym),
- φ jest ciągłym odwzorowaniem z X do X, będącym generatorem półgrupy iteracji.

Trajektorią układu dynamicznego startującą ze stanu x_0 jest ciąg elementów X uzyskanych przez iteracje:

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \ldots, \tag{22}$$

inaczej

$$x_n = \underbrace{\varphi(\varphi(\varphi(\ldots\varphi(x_0))))}_{n\,\text{razy}} = \varphi^n(x_0), \quad n = 1, 2, \ldots \tag{23}$$

Jak widzimy, tak zdefiniowany układ dynamiczny (dla ustalonego n) przypomina funkcję $F(\cdot)$ szyfru blokowego zbudowaną z n rund. Jeżeli jeszcze będzie on miał własności korzystne z punktu widzenia kryptografii, nic nie stoi na przeszkodzie, aby mógł być zastosowany do budowy kryptosystemu.

Najważniejszą własnością układu dynamicznego, która sprawia, że może on znaleźć zastosowanie w kryptografii, jest własność chaosu. Będziemy mówili, że układ dynamiczny jest *chaotyczny*, gdy spełnia trzy warunki (z których żaden nie może być pominięty):

- jego trajektorie są eksponencjalnie wrażliwe na zmiany warunku początkowego (tak zwany efekt motyla: ruch skrzydeł motyla w Amazonii wywołuje burzę w Arizonie);

- układ ma ciągłą (w pewnym przedziale) gęstość widmową;
- układ traci informację (o warunku początkowym) w czasie z prędkością wykładniczą.

Definicja ta, chociaż precyzyjna, jest trudna do zastosowania w praktyce. W literaturze występuje wiele równoważnych warunków (często nazywanych definicjami), których spełnienie gwarantuje, że układ jest chaotyczny, por. np. [2]. Najpopularniejszy jest warunek korzystający z pojęcia wykładników Lapunowa $\lambda_{x,v}$ (co czytamy: wykładnik Lapunowa w punkcie x w kierunku v), zdefiniowanych jako

$$\lambda_{x,v} \equiv \lim_{n\to\infty} \frac{1}{n} \ln \|D\varphi^n(x)(v)\|, \tag{24}$$

gdzie $\|\cdot\|$ jest normą w przestrzeni stycznej w punkcie $x \in X$, v jest elementem tej przestrzeni, natomiast $D\varphi^n(x)(v)$ oznacza pochodną Fŕecheta n-tej iteracji φ w punkcie x w kierunku v. Układ dynamiczny (X, φ) jest chaotyczny w pewnym obszarze (por. [20]), gdy dla prawie wszystkich punktów tego obszaru ma co najmniej jeden dodatni wykładnik Lapunowa (gdy ma ich więcej niż jeden, nazywamy go *hiperchaotycznym*). „Prawie wszędzie" jest tu rozumiane w sensie pewnej miary niezmienniczej równoważnej mierze Lebesgue'a, to znaczy takiej mierze skończonej μ na $\sigma(X)$ (σ-algebrze podzbiorów mierzalnych X), która spełnia warunek φ-niezmienniczości

$$\forall A \in \sigma(X), \qquad \mu(A) = \mu(\varphi^{-1}(A)), \tag{25}$$

oraz dla której istnieje dodatnia ograniczona mierzalna funkcja rzeczywista g na X taka, że

$$\forall A \in \sigma(X), \qquad \mu(A) = \int_A g(x)\, dx.$$

Chaos układu dynamicznego sprawia, że trajektorie są bardzo wrażliwe na zmiany stanu początkowego: startując z dwóch bardzo bliskich punktów początkowych, rozbiegają się wykładniczo. Ta cecha układu przypomina warunki typu rozpraszania, lawinowości lub dyfuzji nakładane przez projektantów na szyfry blokowe.

Drugim warunkiem, który sprawia, że dyskretny w czasie układ dynamiczny może być dobrym kandydatem na algorytm kryptograficzny, jest jego ergodyczność. Mówimy, że układ (X, φ) jest *ergodyczny*, gdy miara każdego φ-niezmienniczego podzbioru B przestrzeni X (tzn. $\varphi(B) \subset B$) jest równa 0 lub $\mu(X)$. Ergodyczność oznacza, że przestrzeń X nie może być nietrywialnie (względem miary μ) podzielona na kilka części. Oznacza to, że trajektoria startująca z dowolnego $x_0 \in X$ nigdy nie lokalizuje się w pewnym podzbiorze przestrzeni X, co sprawia, że w przypadku wszelkich ataków wyczerpujących nie można ograniczać się do przeszukiwania zbiorów mniejszych niż cała przestrzeń X.

Kolejną cechą układu dynamicznego wzmacniającą jego własności kryptograficzne jest mieszanie, własność silniejsza niż ergodyczność. Powiemy, że układ dynamiczny (X, φ) jest *mieszający*, gdy dla dowolnych $A, B \in \sigma(X)$,

$$\lim_{n\to\infty} \mu(\varphi^{-n}(A) \cap B) = \mu(A)\mu(B). \tag{26}$$

W równaniu (26), $\varphi^{-n}(A)$ oznacza przeciwobraz zbioru A pod działaniem n-tej iteracji odwzorowania φ. Zależność (26) oznacza, że wielokrotne działanie operatora φ^{-1} uniezależnia (w sensie probabilistycznym) zbiory (zdarzenia) A i B. Jeżeli dodatkowo założymy, że $\mu(X) = 1$ (miara niezmiennicza μ jest probabilistyczna), to wyrażenie (26) jest równoważne

$$\lim_{n\to\infty} \frac{\mu(\varphi^{-n}(A) \cap B)}{\mu(B)} = \frac{\mu(A)}{\mu(X)}, \tag{27}$$

co wskazuje, że wielokrotne zastosowanie odwzorowania φ^{-1} do dowolnego zbioru A równomiernie rozprzestrzenia ten zbiór po całej przestrzeni stanów X. Innymi słowy, startując z dowolnego $x_0 \in X$, w wyniku iteracji osiągamy dowolny podzbiór X z prawdopodobieństwem proporcjonalnym do rozmiaru tego zbioru w całej przestrzeni stanów (równym ilorazowi miary tego zbioru i miary całej przestrzeni stanów). Oznacza to również, że dla dowolnego stanu końcowego x_n i dużego n, dowolny stan początkowy x_0 jest μ-równie prawdopodobny. Widzimy, że zdefiniowany tu warunek mieszania to odpowiednik mieszania Shannonowskiego wykorzystywanego w teorii szyfrów.

Pozostaje teraz odpowiedzieć na pytanie: jak w sposób efektywny wykorzystać te naturalne własności kryptograficzne chaotycznych układów dynamicznych? Okazuje się, że jest to możliwe. W dodatku możliwe jest wykorzystanie zarówno układów dynamicznych z czasem dyskretnym, jak i z czasem ciągłym, o których nie będziemy mówili szerzej w tej pracy.

4. Szyfr wykorzystujący chaotyczny układ dynamiczny

4.1. *Prosty przykład.* Zacznijmy od przykładu najprostszego odwzorowania chaotycznego i mieszającego (a więc również ergodycznego). Rozważmy *skośne odwzorowanie trójkątne*. Jest to odwzorowanie φ przedziału $[0, 1]$ na siebie następującej postaci (tu zapisane dla kolejnej, $n+1$-szej iteracji):

$$\varphi : \begin{cases} x_{n+1} = \dfrac{x_n}{\alpha} & \text{dla } 0 \le x_n \le \alpha, \\ x_{n+1} = \dfrac{x_n - 1}{\alpha - 1} & \text{dla } \alpha < x_n \le 1, \end{cases} \tag{28}$$

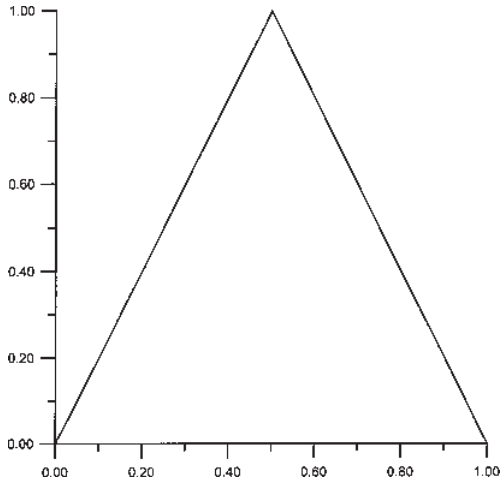
gdzie α jest parametrem określającym położenie wierzchołka trójkąta. Jak już wspomnieliśmy, odwzorowanie takie jest chaotyczne, ergodyczne i mieszające. Miarą niezmienniczą jest tu miara Lebesgue'a na $(0, 1)$. Wykładnik Lapunowa dla danego α jest równy

$$\lambda = -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha). \tag{29}$$

Odwzorowanie odwrotne do odwzorowania trójkątnego, czyli w naszych oznaczeniach φ^{-1}, ma postać (tu zapisaną dla n-tej iteracji)

$$\varphi^{-1} : \begin{cases} x_{n-1} = \alpha x_n \\ \text{lub} \\ x_{n-1} = (\alpha - 1)x_n + 1, \end{cases} \tag{30}$$

jest zatem odwzorowaniem dwuwartościowym.



Odwzorowania φ i φ^{-1} mają następujące własności:

- φ jest odwzorowaniem dwa do jednego, $2:1$,
- φ^{-1} jest odwzorowaniem jeden do dwóch, $1:2$,
- n-ta iteracja φ, tj. φ^n jest odwzorowaniem $2^n:1$,
- n-ta iteracja φ^{-1}, tj. φ^{-n} jest odwzorowaniem $1:2^n$.

Ponadto dla każdego n i dla każdego $x \in X = [0,1]$ spełniona jest równość

$$X = \varphi^n(\varphi^{-n}(X)), \tag{31}$$

czyli odwzorowanie φ^n jest odwrotne do φ^{-n}, niezależnie od sposobu wyboru elementów należących do przeciwobrazu przy każdorazowym obliczaniu φ^{-1}.

Własność (31) odwzorowania trójkątnego pozwoliła na jego podstawie zaproponować prosty szyfr blokowy (por. [9]). W zaproponowanym modelu tajnym kluczem K jest parametr $\alpha \in (0,1)$ odwzorowania. Szyfrowaną wiadomością (blokiem) jest liczba $P \in (0,1)$, czyli punkt startowy iteracji odwzorowania odwrotnego $\varphi^{-1}(\cdot)$. Samo szyfrowanie polega na wykonaniu n iteracji odwrotności odwzorowania trójkątnego z warunkiem początkowym P, czyli operacji (30) dla $x_n = P$. Kryptogram C jest wynikiem działania n-krotnego złożenia tego odwzorowania, to znaczy

$$C = \varphi^{-n}(P) = \varphi^{-1}(\varphi^{-1}(\ldots\varphi^{-1}(P))). \tag{32}$$

Operacja odszyfrowywania polega na znalezieniu obrazu C w wyniku n-tej iteracji φ:

$$P = \varphi^n(C) = \varphi(\varphi(\ldots\varphi(C))). \tag{33}$$

W tak skonstruowanym kryptosystemie wartość kryptogramu C może być wybrana losowo jako jedna z 2^n możliwych wartości należących do przeciwobrazu punktu P uzyskanego w wyniku n-krotnej iteracji φ^{-1}. Dla dowolnie wybranego kryptogramu C (należącego do tego przeciwobrazu) wiadomość P jest odtwarzana jednoznacznie. Bezpieczeństwo szyfru oparte jest na wrażliwości odwzorowania φ na zmiany wartości parametru α.

Zaprezentowany wyżej szyfr chaotyczny sam nie ma wielkiego znaczenia praktycznego, ma jednak podstawową zaletę: jego bezpieczeństwo szyfru oparte jest na ścisłej teorii matematycznej: teorii dyskretnych (w czasie) układów dynamicznych posiadających własność chaosu, ergodyczności i mieszania. Podejście wykorzystywane w tym algorytmie (wiadomość jako warunek początkowy, klucz jako parametr odwzorowania) udało sie później uogólnić (por. [13], [14]) na szerszą klasę problemów, w których kluczem może być również część warunku początkowego. Wpłynęło to na zwiększenie bezpieczeństwa algorytmu w wyniku wykorzystania wrażliwości stanu układu na zmianę warunków początkowych, a nie jedynie na zmianę wartości parametrów.

Wadą zaprezentowanego tutaj szyfru (jak również wszystkich szyfrów bazujących na ciagłej przestrzeni stanów) jest fakt, że działa on na liczbach rzeczywistych, a zatem wynik wszelkich wykonywanych obliczeń zależy od konkretnej implementacji komputerowej (dotyczy to zwłaszcza dalszych cyfr rozwinięcia dziesiętnego liczb). W istocie, możliwość praktycznego odwrócenia operacji zależy od dokładności obliczeń. Działa tutaj swoista reguła nieoznaczoności, wiążąca liczbę iteracji (mającą wpływ na bezpieczeństwo szyfru) z liczbą cyfr znaczących przekształcanych tekstów (ilością informacji zaszyfrowanych w jednym bloku), limitowaną długością słowa komputerowego. Mówiąc w skrócie, ograniczeniem metody są dokładność obliczeń i błędy zaokrągleń operacji arytmetycznych. Pojawia się zatem pytanie, czy można jednak wykorzystać opisane wyżej układy chaotyczne, mające przecież dobre własności kryptograficzne, w praktycznie stosowanych kryptosystemach. Wydaje się, że skutecznym lekarstwem może być odpowiednia dyskretyzacja problemu.

4.2. *Przykład dyskretyzacji — dyskretny (w przestrzeni stanów) szyfr chaotyczny.* Przedstawiony szyfr chaotyczny wykorzystujący odwzorowanie trojkątne może być łatwo zdyskretyzowany, to znaczy możemy go tak przekształcić, że operować on będzie na liczbach całkowitych (por. [17]). Oznaczmy odwzorowanie generujące chaotyczny układ dynamiczny przez

$f_a(\cdot)$, to znaczy:

$$f_a(x) = \begin{cases} \dfrac{x}{a} & \text{dla } 0 < x \leq a \ , \\ \dfrac{x-1}{a-1} & \text{dla } a < x \leq 1. \end{cases} \tag{34}$$

W pierwszym kroku dyskretyzacji wybieramy dowolną liczbę naturalną $M \geq 2$ (im większa liczba, tym bardziej liczna dyskretna przestrzeń stanów), a następnie wprowadzamy przestrzenie tekstów odkrytych P i kryptogramów C w postaci:

$$P = C = \{x = X/M : X \in N,\, 1 \leq X \leq M\} \tag{35}$$

oraz dyskretne odwzorowanie szyfrujące $\hat{f}_a(\cdot)$:

$$\hat{f}_a(x) \equiv \frac{|\{x' \in P : f_a(x') < f_a(x)\}|+1}{M}, \tag{36}$$

gdzie $|\ldots|$ oznacza moc zbioru. Tak zdefiniowana funkcja $\hat{f}_a(x)$ jest odwzorowaniem trójkątnym (skośnym) na skończonej przestrzeni stanów będących liczbami wymiernymi z przedziału $[0, 1]$, czyli zbiorze postaci (35).

Dodatkowo przyjmiemy, że gdy spełniony jest warunek:

$$\hat{f}_a(x_1) = \hat{f}_a(x_2) \quad \text{dla} \quad x_1 < a < x_2, \tag{37}$$

czyli wartości funkcji dla dwóch różnych argumentów są równe, to różnicujemy $\hat{f}_a(x_1)$ i $\hat{f}_a(x_2)$ w taki sposób, że

$$\hat{f}_a(x_1) + \frac{1}{M} = \hat{f}_a(x_2). \tag{38}$$

W efekcie zdyskretyzowane odwzorowanie $\hat{f}_a(x)$ staje się odwzorowaniem różnowartościowym i odwracalnym.

Zauważmy, że funkcja ciągła $f_a(x)$ odwzorowuje przedziały powstałe z podziału odcinka $[0, 1]$ przez punkt (klucz) a w taki sposób, że przekształca $0 \leq x \leq a$ w $0 \leq x \leq 1$ oraz $a < x \leq 1$ w $0 \leq x \leq 1$. W ten sposób powstaje tak zwane jednowymiarowe odwzorowanie piekarza, które najpierw nakłada przedziały na siebie, a później je rozciąga do poprzedniej długości (rozwałkowuje). Natomiast dyskretna funkcja $\hat{f}_a(\cdot)$ miesza punkty wymierne ze zbioru P (zawarte w odcinku $[0, 1]$) w opisany wyżej sposób, zależny od wybranej wartości parametru a.

Odwzorowanie (36)–(38) i odwrotne do niego (teraz już będące zwykłą funkcją różnowartościową) definiują kryptosystem działający na liczbach wymiernych — ułamkach postaci (35). Operacje szyfrowania i odszyfrowywania są w nim zdefiniowane jako, odpowiednio,

$$y = \hat{f}_a^n(x) \quad \text{oraz} \quad x = \hat{f}_a^{-n}(y). \tag{39}$$

Możemy teraz dokonać przejścia w naszym dyskretnym kryptosystemie od liczb wymiernych do liczb całkowitych, wykonując odpowiednie mnoże-

nia:

$$X = M * x, \quad Y = M * y, \quad A = M * a. \tag{40}$$

Nowe przestrzenie tekstów odkrytych i kryptogramów mają postać:

$$C' = P' = \{X : X \in N,\, 1 \le X \le M\}; \tag{41}$$

podobną postać ma również przestrzeń kluczy

$$K' = \{A : A \in N,\, 1 \le A \le M\}. \tag{42}$$

Nowe funkcje stanowiące szyfr są permutacjami (to znaczy odwzorowaniami różnowartościowymi podzbioru liczb naturalnych na siebie). Funkcję szyfrującą i odszyfrowującą (właściwie: funkcję rundy szyfrowania i odszyfrowywania) oznaczymy przez $\tilde{F}_A$ i $\tilde{F}_A^{-1}$. Całkowitoliczbowe odpowiedniki funkcji o wartościach wymiernych, to znaczy (36)–(38) i do niej odwrotnej, mają postać (por. [17])

$$\tilde{F}_A(X) = \begin{cases} \dfrac{M}{A}X & \text{dla } 1 \le X \le A, \\ \dfrac{M}{M-A}(M-X) & \text{dla } A < X \le M, \end{cases} \tag{43}$$

oraz

$$\tilde{F}_A^{-1}(Y) = \begin{cases} X_1, & \text{gdy } m(Y) = Y,\ \dfrac{X_1}{A} > \dfrac{M-X_2}{M-A}, \\ X_2, & \text{gdy } m(Y) = Y,\ \dfrac{X_1}{A} \le \dfrac{M-X_2}{M-A}, \\ X_1, & \text{gdy } m(Y) = Y+1, \end{cases} \tag{44}$$

gdzie

$$X_1 \equiv \lfloor M^{-1}AY \rfloor, \tag{45}$$

$$X_2 \equiv \lceil (M^{-1}A - 1)Y + M \rceil, \tag{46}$$

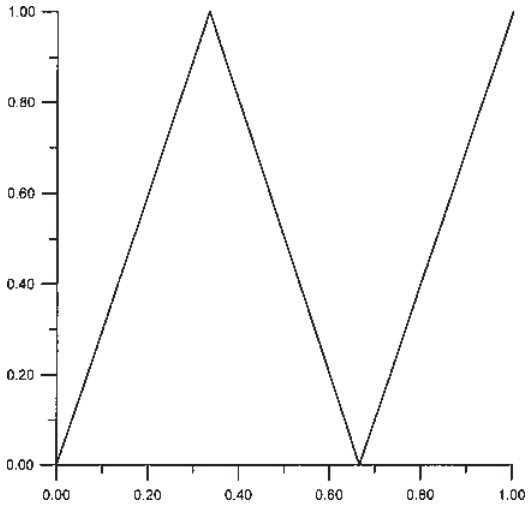
$$m(Y) = \left\lfloor \frac{AY}{M} \right\rfloor - \left\lceil \frac{(A-M)Y}{M} \right\rceil + 1. \tag{47}$$

Jak łatwo się domyślić, odwzorowanie przedstawione wyżej nie jest mocnym (tzn. odpornym na kryptoanalizę) szyfrem i wykorzystanie go do szyfrowania bloków tekstów reprezentowanych w postaci liczb naturalnych jest mało realne, zwłaszcza z powodu dużej złożoności obliczeniowej samych operacji szyfrujących przy dużym M. Wydaje się jednak, że tego rodzaju odwzorowania mogą pełnić rolę składników szyfrów blokowych, na przykład jako element nieliniowy w szyfrze blokowym (funkcja nieliniowa, skrzynka podstawieniowa, permutacja). W takiej sytuacji odwzorowanie działałoby na podblokach pełnego bloku tekstu, czyli na znacznie mniejszych liczbach naturalnych. Ponieważ zazwyczaj w szyfrach blokowych składniki nieliniowe sprawiają najwięcej problemów: ich zapis zajmuje dużo miejsca (trzeba pamiętać całe skrzynki podstawieniowe), a własności są trudne do przewidze-

nia, wykorzystanie chaotycznych układów dynamicznych może być tu cenną alternatywą dla metod stosowanych dotychczas.

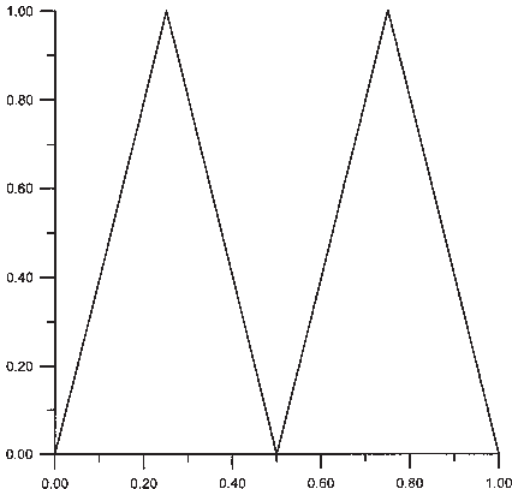
5. Możliwość wykorzystania innych odwzorowań. W poprzednim rozdziale pokazaliśmy przykład chaotycznego układu dynamicznego, który można zdyskretyzować, uzyskując rodzinę funkcji (zależnych od parametru a — tajnego klucza oraz liczby iteracji n), którymi można zastąpić elementy nieliniowe w szyfrach blokowych. Odwzorowanie trójkątne generujące ten układ dynamiczny nie jest oczywiście jedynym, które może być zdyskretyzowane tak, żeby stać się składnikiem szyfru blokowego. Rozważane odwzorowanie trójkątne składa się z jednego daszka. Równie łatwo można zdyskretyzować odwzorowanie składające się z półtora daszka:

$$x_{n+1} = \begin{cases} 3x_n, & 0 \le x_n < 1/3, \\ 2 - 3x_n, & 1/3 \le x_n < 2/3, \\ -2 + 3x_n, & 2/3 \le x_n \le 1, \end{cases} \tag{48}$$



z dwóch pełnych daszków

$$x_{n+1} = \begin{cases} 4x_n, & 0 \le x_n < 1/4, \\ 2(1 - 2x_n), & 1/4 \le x_n < 2/4, \\ 2(2x_n - 1), & 2/4 \le x_n < 3/4, \\ 4(1 - x_n), & 3/4 \le x_n \le 1, \end{cases} \tag{49}$$



i tak dalej. Dyskretyzacja takich odwzorowań (zmodyfikowanych przez wprowadzenie jako parametru zmiennej współrzędnej wierzchołków daszków) prowadzi do nowych definicji permutacji.

Dalsze rodziny odwzorowań można uzyskać z dyskretyzacji odwzorowania logistycznego, czyli odwróconej paraboli, postaci

$$x_{n+1} = 4x_n(1 - x_n) \tag{50}$$

i dalszych odwzorowań z tej rodziny, czyli krzywych wyższego stopnia (o kształcie analogicznym do opisanych daszków, jednak z gładkimi wierzchołkami; por. [12]). Można szukać dalszych generatorów chaotycznych układów dynamicznych (por. np. [2], [12]), wzbogacając możliwości budowy szyfrów blokowych. W tej pracy poprzestaniemy na podanych przykładach, zwrócimy jednak jeszcze uwagę na możliwość, jaką dają tak zwane rozwiązywalne (lub lepiej: konstruowalne) chaotyczne układy dynamiczne.

Zauważmy, że układy dynamiczne generowane przez odwzorowania (28), (48) i (49) mogą być rozwiązane, to znaczy można podać wyrażenie dla n-tego elementu trajektorii startującej z punktu x_0:

$$x_n = \frac{1}{\pi} \arccos(\cos k^n \pi x_0) \tag{51}$$

dla, odpowiednio, $k = 2, 3, 4$. Również dla $k = 5, 6, \ldots$ uzyskany układ dynamiczny jest chaotyczny. Rozwiązanie takie odpowiada generatorowi układu dynamicznego, będącemu odwzorowaniem trójkątnym o odpowiednio większej liczbie daszków. Okazuje się (por. [7]), że własność chaosu mają również układy dynamiczne (trajektorie) postaci

$$x_n = \frac{1}{\pi} \arccos(\cos z^n \pi x_0) \tag{52}$$

z parametrem z będącym ułamkiem

$$z = \frac{p}{q}, \tag{53}$$

gdzie p i q są liczbami naturalnymi względnie pierwszymi. Wykładnik Lapunowa takiego układu jest równy

$$\lambda = \ln z. \tag{54}$$

Trajektorie (52) mają tę własność, że wykres par punktów (x_n, x_{n+1}) jest krzywą Lissajous taką, że każdej wartości x_n odpowiada q wartości x_{n+1}, a każdej wartości x_{n+1} odpowiada p wartości x_n. Zatem w tym przypadku nie tylko odwzorowanie odwrotne jest wielowartościowe (p-wartościowe), jak to było w przypadku naszej wyjściowej funkcji trójkątnej, lecz również samo odwzorowanie przyporządkowujące pierwszej zmiennej z pary punktów drugą zmienną jest wielowartościowe (q-wartościowe).

Odpowiednia dyskretyzacja takiego odwzorowania pozwoli uzyskać funkcje przydatne do budowy skrzynek podstawieniowych w szyfrach blokowych. Można również pokazać, że dyskretyzacja chaotycznego układu dynamicznego prowadzi do kryptosystemu odpornego na kryptoanalizę liniową (por. [1]), zatem jest szansa uzyskania silnych kryptograficznie składowych szyfru blokowego.

6. Zastosowania ciągłych układów dynamicznych. W bezpiecznych systemach komunikacji szeroko wykorzystywane są również ciągłe (w czasie) układy dynamiczne, czyli rozwiązania nieliniowych równań różniczkowych mających własność chaosu. Zauważmy, że w tym wypadku ciągłe są zarówno czas, który w poprzednio omawianych przypadkach był dyskretny i dobrze odpowiadał iteracjom funkcji rundowej szyfrów blokowych, jak i przestrzeń stanów (w tradycyjnej kryptografii bloki bitów). Zanim w tej trudnej sytuacji spróbujemy znaleźć możliwości zastosowania układów dynamicznych do konstrukcji szyfrów blokowych, przedstawimy powszechnie stosowane metody. Najpopularniejsze są dwa systemy bezpiecznej komunikacji: sterowanie chaosu i synchronizacja chaosu (por. np. [11]).

Bezpieczne przesyłanie wiadomości metodą *sterowania chaosu* polega na modulowaniu wybranego parametru ciągłego układu dynamicznego za pomocą impulsów będących zakodowaną wiadomością. Odbiorca, dysponując całą trajektorią z zawartą w niej informacją oraz znając jej punkt startowy i pozostałe parametry układu, może wygenerować nową trajektorię, wplatając w nią metodą prób i błędów modulacje parametru tak, aby wygenerowana przez niego trajektoria była identyczna z tą, która zawiera informacje. Sterowanie, które generuje właściwą trajektorię, jest przesłaną tajną informacją.

Druga metoda, nazywana *synchronizacją chaosu*, polega na potraktowaniu trajektorii ciągłego układu chaotycznego o dużej amplitudzie jako szumu maskującego wiadomość. Odbiorca otrzymujący wiadomość zamaskowaną chaotyczną trajektorią odzyskuje wiadomość, odejmując tę trajektorię od otrzymanego sygnału, odzyskując w ten sposób tajną wiadomość. W obu przypadkach bezpieczna komunikacja możliwa jest dzięki silnej wrażliwości trajektorii chaotycznego układu dynamicznego na zmiany warunków początkowych i zmiany wartości parametrów: tajnym kluczem są tu te właśnie wielkości. Zarówno szyfrujący, jak i odszyfrowujący muszą znać dokładne wartości tych parametrów. Ewentualny napastnik nie znający klucza nie ma szans na skuteczne przeprowadzenie opisanych wyżej operacji odszyfrowywania wiadomości.

Przedstawione metody sterowania chaosu i synchronizacji chaosu korzystają w istotny sposób z ciągłości czasu i ciągłości przestrzeni stanów. Do celów budowy skrzynek podstawieniowych może być wykorzystana inna interpretacja ciągłych układów dynamicznych, a mianowicie dynamika symboliczna. Metoda ta polega na przekształceniu ciągłej trajektorii układu dynamicznego (ciągłej w czasie i w przestrzeni stanów) do postaci dyskretnego ciągu symboli należących do pewnego alfabetu, pojawiających się w kolejności zależnej od kształtu trajektorii. Najprostszym przykładem takiego przekształcenia może być chaotyczny generator bitów (por. [23]), w którym alfabet składa się jedynie z dwóch elementów: bitu 0 i bitu 1. Już to proste przekształcenie ma dobre własności kryptograficzne (jednoznaczną zależność od warunku początkowego, asymptotyczną niezależność wygenerowanych bitów, równoprawdopodobne wygenerowanie 0 i 1). W ogólniejszych przypadkach również można uzyskać dobre przekształcenia. Odpowiednio je dostosowując, można znaleźć przekształcenia, które będą odwzorowywać pewne początkowe ciągi symboli (brane z początku trajektorii układu dynamicznego) na inne ciągi, uzyskane z oddalonego fragmentu trajektorii. Ponadto można pokazać (w sposób podobny do odpowiedniego dowodu z pracy [23]), że oba ciągi będą asymptotycznie statystycznie niezależne, co powinno mieć wpływ na kryptograficzne bezpieczeństwo takiego odwzorowania. Jest też nadzieja na sprzętową realizację takiego odwzorowania, wykorzystującą fizyczne realizacje odpowiednio dobranych, ciągłych w czasie i w przestrzeni stanów chaotycznych układów dynamicznych. Problemem pozostaje jedynie efektywna konstrukcja odwzorowania dostosowanego do zastąpienia skrzynki podstawieniowej w szyfrze blokowym.

7. Podsumowanie. Powtórzymy tu w punktach idee, które przewijały się w całej pracy.

- Dyskretne w czasie, chaotyczne układy dynamiczne są bardzo podobne, ze względu na własność ukrywania związku między stanem

końcowym po n iteracjach i stanem początkowym, do iteracyjnych szyfrów blokowych, które przecież zostały stworzone w tym celu, żeby całkowicie zamaskować swoje dane wejściowe.

- Szyfry blokowe działają na danych, które są blokami bitów skończonej długości, a więc na dyskretnej przestrzeni stanów.
- Chaotyczne układy dynamiczne działają na danych z ciągłej przestrzeni stanów, więc wszystkie obliczenia wykonywane dla nich są zależne od konkretnej implementacji.
- Kryteria bezpieczeństwa wykorzystywane przy budowie szyfrów blokowych mają charakter intuicyjny.
- Korzystne kryptograficznie własności układów dynamicznych, to znaczy chaos, ergodyczność, mieszanie, są precyzyjnie matematycznie zdefiniowane.
- Są możliwości takiej dyskretyzacji chaotycznych układów dynamicznych, by mogły znaleźć zastosowanie jako element składowy szyfrów blokowych.

References

[1] J. M. Amigó, J. Szczepański, *Approximations of dynamical systems and their applications to cryptography*, Internat. J. Bifurcation Chaos 13 (2003), to appear.

[2] R. Brown, L. O. Chua, *Clarifying chaos*: *examples and counterexamples*, Internat. J. Bifurcation Chaos 6 (1996), 219–249.

[3] A. Compagner, *Definitions of randomness*, Am. J. Phys. 59 (1991), 700–705.

[4] N. Courtois, J. Pieprzyk, *Cryptanalysis of block ciphers with overdefined system of equations*, w: Asiacrypt 2002, LNCS 2501, Springer 2002.

[5] J. Daemen, L. R. Knudsen, V. Rijmen, *Linear frameworks for block ciphers*, Designs Codes Cryptography 22 (2001), 65–87.

[6] D. E. Robling Denning, *Kryptografia i ochrona danych*, WNT, Warszawa, 1992.

[7] J. A. Gonzalez, R. Pino, *Chaotic and stochastic functions*, Physica 276A (2000), 425–440.

[8] A. Górska, *Metoda zwiększająca efektywność kryptoanalizy liniowej iteracyjnych szyfrów blokowych*, rozprawa doktorska, WEiTI PW, Warszawa, 2002.

[9] T. Habutsu, Y. Nishio, I. Sasase, S. Mori, *A secret key cryptosystem by iterating a chaotic map*, w: Eurocrypt'91, 1991, 127–140.

[10] M. Kac, *What is random?* Amer. Scientist 71 (1983), 405–406.

[11] T. Kapitaniak, *Controlling Chaos. Theoretical and Practical Methods in Non-linear Dynamics*, Academic Press, London, 1996.

[12] S. Katsura, W. Fukuda, *Exactly solvable models showing chaotic behavior*, Physica 130A (1985), 597–605.

[13] Z. Kotulski, J. Szczepański, *Discrete chaotic cryptography*, Ann. Physik 6 (1997), 381–394.

[14] Z. Kotulski, J. Szczepański, K. Górski, A. Paszkiewicz, A. Zugaj, *The application of discrete chaotic dynamical systems in cryptography — DCC Method*, Internat. J. Bifurcation Chaos 9 (1999), 1121–1135.

[15] Z. Kotulski, *Generatory liczb losowych*: *algorytmy, testowanie*, *zastosowania*, Mat. Stos. 2 (2001), 32–66.
[16] Z. Kotulski, *Nowoczesne technologie informacyjne*: *bezpieczeństwo danych*, w: M. Kleiber (red.), *Nauki techniczne u progu XXI wieku*, IPPT, Warszawa, 2002, 181–210.
[17] N. Masuda, K. Aihara, *Cryptosystems based on space-discretization of chaotic maps*, IEEE Trans. Circuits Systems I Fund. Theory Appl. 49 (2002), 28–40.
[18] A. Menezes, P. van Oorschot, C. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, 1996.
[19] J. Pieprzyk, T. Hardjono, J. Seberry, *Fundamentals of Computer Security*, Springer, Berlin, 2002.
[20] H. G. Schuster, *Chaos deterministyczny,* PWN, Warszawa, 1995.
[21] C. E. Shannon, *A Mathematical Theory of Communication*, Bell System Tech. J. 27 (1948), 379–423 and 623–656.
[22] J. Stern, *Why provable security matters?*, w: Advances in Cryptology — EUROCRYPT'2003, Lecture Notes in Comput. Sci. 2656, Springer, Berlin, 2003, 489–461.
[23] J. Szczepański, Z. Kotulski, *Pseudorandom number generators based on chaotic dynamical systems*, Open Sys. Inf. Dyn. 8 (2001), 137–146.
[24] S. Trznadel, A. Zugaj, K. Górski, A. Paszkiewicz, Z. Kotulski, J. Szczepański, *Przegląd stanu wiedzy na temat kryptoanalizy liniowej ze szczególnym uwzględnieniem algorytmu DES*, w: Poznańskie Warsztaty Telekomunikacyjne, PWT'98, Poznań 1998, 4.20-1 do 4.20-6.
[25] R. Zieliński, *Wytwarzanie losowości*, Wiadomości Mat. 29 (1992), 189–203.

Instytut Podstawowych Problemów Techniki PAN
ul. Świętokrzyska 21, 00-049 Warszawa
E-mail: zkotulsk@ippt.gov.pl