Laboratorium ochrony danych

Ćwiczenie nr 1

Temat ćwiczenia: Ciała skończone proste

Cel dydaktyczny: Poznanie metod generowania ciał skończonych prostych oraz zasad rachowania w tych systemach algebraicznych, badanie właściwości ciał i wielomianów nad ciałami, sekwencje okresowe i pseudolosowe.

Wprowadzenie teoretyczne

W kryptografii oraz w technice kodowania stosuje się alfabety o skończonej liczbie elementów. Z reguły liczba elementów stosowanego alfabetu równa jest albo liczbie pierwszej, albo potędze liczby pierwszej. Dzięki temu zastosowany alfabet można uważać za strukturę algebraiczną, która nazywa się ciałem skończonym lub inaczej ciałem Galois.

Ciało skończone liczbowe *GF*(***p***) jest to system algebraiczny złożony ze zbioru liczb A={0,1,...,*p*-1} oraz z operacji dodawania i mnożenia modulo *p*, które można wykonywać na tych liczbach (istnieją też ciała skończone nieliczbowe, np. zawierające elementy w postaci wektorów, macierzy oraz odpowiednio zdefiniowanych operacjach dodawania i mnożenia elementów; zbiór A zawiera co najmniej dwa elementy). Taki system spełnia wszystkie aksjomaty ciał, tzn.:

- zbiór {0,1,...,*p*-1} wraz z operacją dodawania modulo *p* jest przemienną grupą addytywną, z elementem neutralnym 0,
- zbiór {0,1,...,*p*-1} wraz z operacją mnożenia modulo *p* jest przemienną grupą multyplikatywną, z elementem neutralnym 1,
- mnożenie jest rozdzielne względem dodawania, czyli dla każdego *a, b* i *c* należących do {0,1,..., *p*-1} spełniona jest zależnośc $\forall a,b,c \in A \quad a \cdot (b+c) = (b+c) \cdot a = a \cdot b + a \cdot c$.

Skończone ciała proste można skonstruować dla zbiorów liczbowych o liczbie elementów równej **liczbie pierwszej** *p*. Ciała takie oznaczamy symbolem *GF*(*p*). Elementami ciała prostego są liczby: 0, 1, 2, ..., $p-1$. Działania w ciałach prostych są takie same jak działania arytmetyczne z operacją modulo *p*. Ciało proste jest więc ciałem reszt modulo *p*. Sumę *S* i iloczyn *P* dwóch elementów ciała prostego *a* i *b* określają zależności:

$$S \equiv a + b \pmod{p},$$

$$P \equiv a \cdot b \pmod{p}.$$

W *GF*(*p*) każdy element ma element do siebie przeciwny, a każdy element niezerowy ma multyplikatywną odwrotność, dzięki czemu w ciele *p*-elementowym można też odejmować, dzielić, potęgować i wyciągać pierwiastki. Wobec tego nad ciałem *GF*(*p*) mają sens takie obliczenia, jak rozwiązywanie równań liniowych i nieliniowych, dodawanie, mnożenie i odwracanie macierzy, wszystkie operacje na wielomianach, itp.

Element przeciwny ciała obliczamy za pomocą aksjomatu:

$$\forall a \in A \quad \exists b \in A \quad a + b = b + a = 0,$$

a element odwrotny *b* dla niezerowego elementu *a* ciała obliczamy za pomocą aksjomatu:

$$\forall a \in A \quad \exists b \in A \quad a \cdot b = b \cdot a = 1.$$

Element odwrotny do *a* oznaczamy jako a^{-1}.

Jako przykład wieloelementowego, skończonego ciała prostego przyjmijmy ciało $GF(7)$. Elementami ciała $GF(7)$ są liczby: 0, 1, 2, 3, 4, 5, 6.

Elementy przeciwne do elementów ciała *GF*(7) wyznaczamy z tabelki dodawania. Element przeciwny do 0 to 0, do 1 to 6 (suma 1 + 6 = 0; liczone mod 7), do 2 to 5, do 3 to 4, do 4 to 3, do 5 to 2 i do 6 to 1. Z kolei elementy odwrotne do elementów ciała *GF*(7) wyznaczamy z tabelki mnożenia. Element odwrotny do 1 to 1 (iloczyn 1 x 1 = 1; liczone mod 7), do 2 to 4, do 3 to 5, do 4 to 2, do 5 to 3 i do 6 to 6.

Tabliczki dodawania i mnożenia ciała $GF(7)$

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

.	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Do badania ciał skończonych używa się funkcji Eulera. Funkcja Eulera $\varphi(n)$ określa liczbę liczb naturalnych w zbiorze {1, 2, ..., *n*–1} względnie pierwszych z *n*. Na przykład $\varphi(8) = 4$, gdyż w zbiorze liczb mniejszych od 8 tylko 1, 3, 5 i 7 są względnie pierwsze z 8. Liczby względnie pierwsze nie mają żadnego wspólnego podzielnika oprócz 1. Funkcja Eulera dla liczby pierwszej *p* jest równa $p - 1$, gdyż wszystkie liczby mniejsze od *p* są względnie pierwsze z *p*, czyli $\varphi(p) = p-1$.

Aby znaleźć wartość funkcji Eulera liczby złożonej *n* rozkładamy ją na iloczyn potęg liczb pierwszych

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdot \ldots \cdot p_m^{e_m}.$$

Wartość funkcji Eulera dla takiej liczby złożonej wylicza się ze wzoru

$$\varphi(n) = \prod_{i=1}^{m} p_i^{e_i - 1}(p_i - 1).$$

P r z y k ł a d .

Obliczenie funkcji Eulera liczby złożonej: $n = 2646 = 2 \cdot 3^3 \cdot 7^2$; $\varphi(2646) = 1 \cdot 3^2 \cdot 2 \cdot 7 \cdot 6 = 756$.

Można też skorzystać z definicji funkcji Eulera i analizować największe wspólne dzielniki liczb {1, 2, …, *n*-1} z liczbą *n*.

Niezerowe elementy ciała $GF(p)$ charakteryzuje rząd multyplikatywny. *Rzędem multyplikatywnym* dowolnego elementu ciała *a* jest najmniejsza liczba całkowita *e* taka, że

$$a^e = 1 \pmod{p}$$

Na przykład rzędem multyplikatywnym elementu 5 ciała $GF(7)$ jest 6, ponieważ $5^6 = 1 \pmod 7$. Rząd multyplikatywny elementu ciała $GF(p)$ jest dzielnikiem $p-1$.

Elementy ciała $GF(7)$ mają następujące rzędy multyplikatywne:
element 1 – rząd multyplikatywny 1;
elementy 2 i 4 – rząd multyplikatywny 3;
elementy 3 i 5 – rząd multyplikatywny 6; (3 i 5 to elementy pierwotne ciała *GF*(7))
element 6 – rząd multyplikatywny 2.

Elementy ciała $GF(p)$ mające rząd multyplikatywny równy $p-1$ nazywamy ***elementami pierwotnymi*** ciała. Liczbę elementów pierwotnych *n* ciała *GF*(*p*) można określić z zależności

$$n = \varphi(p-1),$$

gdzie φ jest funkcją Eulera.

Każdy element niezerowy ciała generuje grupę cykliczną. Element pierwotny ciała generuje grupę multyplikatywną ciała. W tak utworzonej grupie będą wszystkie niezerowe elementy ciała. Elementy grupy multyplikatywnej o rzędzie multyplikatywnym większym od 1 i mniejszym od $p-1$ generują podgrupy multyplikatywne. Taka podgrupa zachowuje działania grupy.

Grupę cykliczną generowaną przez dowolny element ciała skończonego otrzymamy, biorąc kolejne potęgi tego elementu. Na przykład element pierwotny 5 ciała $GF(7)$ pozwala wygenerować całą grupę multyplikatywną: 5, 4, 6, 2, 3, 1, gdyż kolejne potęgi elementu 5, tj. 5^i *mod p*, dla *i*=1,2,3,4,5,6, wynoszą: 5, 5·5=4, 4·5=6, 6·5=2, 2·5=3, 3·5=1 (elementy pierwotne często nazywane są *generatorami grupy*). Podobnie element 2 generuje podgrupę trzyelementową: 2, 4, 1. Podczas obliczeń w ciele $GF(p)$ wykorzystujemy redukcję modularną w postaci: $a^2 \bmod p = ((a \bmod p) \cdot a) \bmod p$.

W teorii kodowania są szeroko wykorzystywane wielomiany nad ciałami skończonymi, a w kryptografii sekwencje okresowe pseudolosowe stowarzyszone z wielomianami pierwotnymi nad ciałami prostymi. Wielomian stopnia *m* nad ciałem *GF*(*q*) ma ogólną postać (współczynniki a_i należą do ciała *GF*(*q*)):

$$p(x) = x^m + a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0 \quad \text{nad } GF(q).$$

Przyrównując ten wielomian do zera, otrzymamy

$$x^m = -a_{m-1}x^{m-1} - a_{m-2}x^{m-2} - \ldots - a_1 x - a_0.$$

Zależność rekurencyjna stowarzyszona z tym wielomianem będzie

$$s_{j+m} = -a_{m-1}s_{j+m-1} - a_{m-2}s_{j+m-2} - \ldots - a_1 s_{j+1} - a_0 s_j, \; j = 0,1,2,3,\ldots$$

Działania należy tu wykonywać zgodnie z zasadami rachowania w ciele $GF(q)$.
W przypadku ciała liczbowego *GF*(*p*) jest $(-a_{m-1}) = (p - a_{m-1})$, a obliczenia wykonujemy modulo *p*.

$$s_{j+m} = (p - a_{m-1})s_{j+m-1} + (p - a_{m-2})s_{j+m-2} + \cdots + (p - a_1)s_{j+1} + (p - a_0)s_j.$$

Gdy założymy ciąg początkowy o długości *m* elementów: $s_0, s_1, s_2, \ldots, s_{m-1}$, np. 1, 0, 0, …, 0, wówczas dla kolejnych wartości j można obliczyć z powyższej zależności elementy sekwencji okresowej (ciąg *m* początkowych wartości sekwencji nie może składać się z samych zer). Okres wygenerowanej sekwencji okresowej zależy od typu wielomianu. W przypadku wielomianów *pierwotnych* sekwencja osiąga okres maksymalny.

Okres maksymalny *T* dla wielomianu stopnia *m* nad ciałem $GF(q)$ wynosi

$$T = q^m - 1.$$

Wielomiany niepierwotne generują sekwencje o okresie mniejszym od *T*.

Aby znaleźć okres każdej sekwencji cyklicznej generowanej na podstawie wielomianu należy wyznaczyć maksymalnie $T + m = q^m - 1 + m$ jej elementów.

Niech wielomianem generującym sekwencję okresową będzie wielomian stopnia trzeciego (m=3; q=2) nad ciałem $GF(2)$

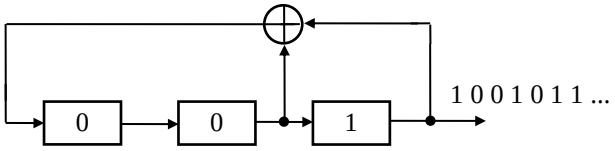
$$x^3 + x + 1 = 0.$$

Zależność rekurencyjna stowarzyszona z tym wielomianem ma postać

$$s_{j+3} = s_j + s_{j+1}, \qquad j = 0,1,2,3,\ldots$$

a wygenerowana sekwencja dla ciągu początkowego 1,0,0 to: 1001011 1001011 ... W ciele tym wszystkie obliczenia wykonujemy modulo 2. Okres sekwencji wynosi T = 7, czyli spełniona jest zależność $T = q^m - 1 = 2^3 - 1 = 7$. Analizowany wielomian jest więc wielomianem pierwotnym.

Realizację zależności rekurencyjnej za pomocą układów logicznych pokazano na rysunku.



Generator binarnej sekwencji okresowej

Dla wielomianu nad *GF*(3) postaci $x^2 + x + 2$ zależność rekurencyjna stowarzyszona z tym wielomianem ma postać $s_{j+2} = 2s_{j+1} + s_j$, gdzie $j = 0, 1, 2, 3 \ldots$. Wygenerowana sekwencja dla ciągu początkowego 1,0 ma postać: 10122021 10122021 … W ciele tym wszystkie obliczenia wykonujemy modulo 3. Okres sekwencji wynosi T = 8, czyli spełniona jest zależność $T = q^m - 1 = 3^2 - 1 = 8$ (m=2; q=3). Analizowany wielomian jest więc wielomianem pierwotnym.

Z kolei wielomian nad *GF*(3) postaci $x^2 + 2x + 1$ nie jest wielomianem pierwotnym, gdyż wygenerowana sekwencja okresowa dla ciągu początkowego 1,0 ma postać: 102201 102201 …, a jej okres wynosi T = 6. Nie jest więc w tym przypadku spełniony warunek $T = q^m - 1 = 3^2 - 1 = 8$ (m=2; q=3).

Sekwencje pseudolosowe

Sekwencje okresowe, generowane przez wielomiany pierwotne stopnia m nad $GF(q)$, mają maksymalny okres definiowany wzorem $T = q^m - 1$ i nazywają się *sekwencjami pseudolosowymi*.

Sekwencje pseudolosowe mają właściwości zbliżone do ciągów losowych, chociaż nie są w pełni ciągami losowymi. Mogą one jednak być generowane w komputerach i znajdują liczne zastosowania w kryptografii, np. do konstrukcji haseł lub tajnych kluczy szyfrujących w szyfrach blokowych oraz strumieniowych.

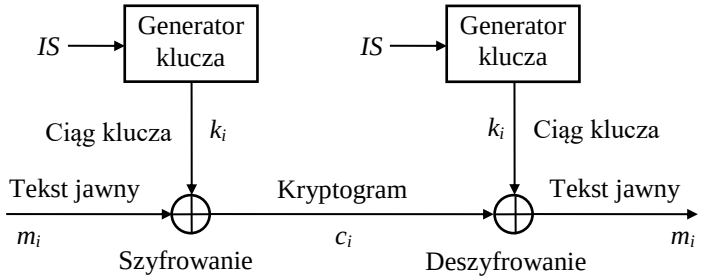
W szyfrowaniu strumieniowym lub potokowym przetwarzaną jednostką jest bit lub bajt (znak), a w szyfrowaniu blokowym – blok, zawierający najczęściej od kilku do kilkunastu bajtów (znaków). Obie te metody szyfrowania mogą być używane zarówno do zabezpieczania transmisji informacji, jak i do bezpiecznego przechowywania danych w pamięciach komputerów lub w plikach dyskowych. Przyjęta technika szyfrowania wynika najczęściej z założonego algorytmu kryptograficznego i ma wpływ na konstrukcję układu szyfratora i deszyfratora.

Szyfry blokowe wymagają podziału informacji na bloki o jednakowym rozmiarze oraz szyfrowaniu każdego bloku tajnym kluczem kryptograficznym, takim samym dla każdego bloku (np. algorytmy DES, AES). W przypadku szyfrów blokowych klucz k-bitowy tworzymy z k kolejnych, wybranych bitów sekwencji pseudolosowej.

Szyfr strumieniowy dzieli wiadomość M na znaki lub bity $m_1, m_2, \ldots,$ a następnie szyfruje każdy element m_i za pomocą elementu klucza k_i, należącego do strumienia klucza $K = k_1, k_2, \ldots$

$$E_K(M) = E_{k1}(m_1)\,E_{k2}(m_2)\ldots$$

Szyfr strumieniowy jest okresowy, jeśli powtarza się strumień klucza. Do algorytmów strumieniowych okresowych należą szyfry podstawieniowe, szyfry Vigenère'a i szyfry realizowane w maszynach rotorowych. Szyfr Vernama jest szyfrem strumieniowym nieokresowym.



Strumieniowy szyfr synchroniczny

Klucz w urządzeniu realizującym szyfr strumieniowy może być umieszczony w pamięci lub generowany na bieżąco. Pamiętanie klucza w przypadku długich strumieni kluczy jest niepraktyczne. Implementację synchronicznego szyfru strumieniowego z generatorem klucza pokazano na rysunku.

Gdy urządzenie przetwarza ciągi bitów, wtedy szyfrowanie i deszyfrowanie odbywa się zgodnie z zasadami rachowania w ciele $GF(2)$, a szyfrator i deszyfrator jest bramką Ex-OR. Zatem algorytm szyfrowania ma postać:

$$c_i = E_{k_i}(m_i) = m_i + k_i,$$

gdzie każdy element jest bitem, a dodawanie jest dodawaniem modulo dwa.

Do obliczenia elementu tekstu jawnego z kryptogramu używa się następującego algorytmu ($2k_i \bmod 2 = 0$):

$$D_{k_i}(c_i) = c_i + k_i = (m_i + k_i) + k_i = m_i.$$

Szyfry strumieniowe należą do systemów z kluczem tajnym, gdzie bezpieczeństwo systemu zależy od klucza. Algorytm generowania klucza musi być algorytmem deterministycznym, aby klucz mógł być łatwo odtworzony po stronie odbiorczej. Klucze najczęściej są ciągami okresowymi, ale najlepszym rozwiązaniem jest klucz jednorazowy.

Generator klucza synchronizuje się za pomocą impulsów synchronizujących *IS*. System kryptograficzny działa poprawnie, jeśli oba generatory, po stronie nadawczej i odbiorczej, pracują synchronicznie. Gdy generatory stracą synchronizm, musi on być przywrócony. Sygnały synchronizacji blokowej w systemie zapewniają protokoły komunikacyjne, a sygnały synchronizacji bitowej – urządzenia transmisyjne modemy.

Przykładowe operacje szyfrowania i deszyfrowania ciągów binarnych:

Szyfrowanie								
Tekst jawny	1	0	0	1	1	1	0	1
Klucz	1	1	0	1	0	0	1	0
Kryptogram	0	1	0	0	1	1	1	1

Deszyfrowanie								
Kryptogram	0	1	0	0	1	1	1	1
Klucz	1	1	0	1	0	0	1	0
Tekst jawny	1	0	0	1	1	1	0	1

System z szyfrem strumieniowym nie powoduje propagacji błędów transmisyjnych. Błąd transmisji jednego bitu (bajtu, znaku) nie wpływa na następne bity (bajty, znaki).

Binarną sekwencję pseudolosową można wykorzystac do szyfrowania strumieniowego bitów informacji zapisanych w pliku dyskowym. W tym celu pobieramy z dysku kolejne bajty (znaki) informacji i wykonujemy na nich operację XOR z kolejnymi 8 bitami pobranymi z binarnej sekwencji pseudolosowej. Wynik zapisujemy do pliku zaszyfrowanego. Istotne jest aby liczba wygenerowanych bitów sekwencji była większa lub równa liczbie bitów danych zapisanych w pliku (może to oznaczać konieczność wygenerowania kilku okresów sekwencji). W tym przypadku deszyfrowanie polega na pobieraniu kolejnych bajtów z pliku zaszyfrowanego i wykonywaniu na nich operacji XOR z kolejnymi bitami pobranymi z binarnej sekwencji pseudolosowej.

Przebieg ćwiczenia

1. Wyznaczyć: tabliczkę mnożenia i dodawania ciała $GF(p)$, elementy przeciwne do elementów ciała $GF(p)$, elementy odwrotne niezerowych elementów ciała $GF(p)$, dla 3 wybranych liczb pierwszych $p \leq 19$.
2. Wyznaczyć rząd multyplikatywny elementów ciała $GF(p)$, dla 3 wybranych liczb pierwszych $p \leq 19$.
3. Znaleźć elementy pierwotne ciała $GF(p)$, dla 3 wybranych liczb pierwszych $p \leq 19$.
4. Wygenerować sekwencję okresową dla wielomianu stopnia m nad ciałem $GF(p)$. Wykorzystać zależność rekurencyjną stowarzyszoną z wielomianem. Przetestować działanie programu dla wielomianów stopnia m o współczynnikach $a\_0, a\_1, \ldots, a\_(m-1)$ i sekwencji początkowej $s\_0, s\_1, \ldots, s\_(m-1)$ zadawanych przez użytkownika

(wprowadzanych jako dane do programu lub zadawanych jako stałe w programie), np. dla $x^4 + x + 1$ nad *GF*(**2**), $x^2 + x + 2$ nad *GF*(**3**), lub innych. Wyznaczyć okres sekwencji. Określić, czy wykorzystywany wielomian jest pierwotny.

5. Opracować program realizujący szyfrowanie i deszyfrowanie strumieniowe dowolnego pliku dyskowego z wykorzystaniem generatora binarnej sekwencji pseudolosowej stowarzyszonej z wielomianem pierwotnym $x^{10} + x^3 + 1$ nad *GF*(**2**). Kolejne bajty odczytane z pliku szyfrować kolejnymi bitami pobranymi z sekwencji pseudolosowej, wykonując operację XOR na bitach (8-bitów z bajtu z pliku XOR z 8 bitami z sekwencji). Po wykonaniu operacji deszyfrowania sprawdzić, czy odszyfrowany plik ma zawartość oraz długość w bajtach identyczną z oryginałem.

Zadania rozwiązać w dowolnym języku programowania. Zademonstrować działanie programów podczas laboratorium. Przysłać krótkie sprawozdanie zawierające wyniki testów funkcjonalnych opracowanych programów.