1. Kryptografia - pojęcia podstawowe

1.1 Cele i środki kryptografii

Kryptologia i kryptografia

Kryptologia jest nauką o szyfrach czyli mówiąc niezbyt precyzyjnie specjalnych metodach, które stosujemy w celu zapewnienie poufności i autentyczności przesyłanej lub zapamiętywanej informacji.

Początki kryptologii sięgają starożytności. Obecnie traktuje się kryptologię z jednej strony jako dział informatyki teoretycznej (a więc matematyki) z drugiej zaś jako fragment szeroko pojętej elektroniki (praktyczne aspekty kryptografii, hardware kryptograficzny, kryptograficzne układy scalone, karty kryptograficzne).

Jednocześnie kryptologię można traktować jako dział obszerniejszej dziedziny jaką jest "bezpieczeństwo i ochrona danych w systemach cyfrowych". Ochrona i bezpieczeństwo danych obejmuje również m.in. ochronę przed wirusami komputerowymi, politykę bezpieczeństwa i techniki podsłuchu.

Nie trzeba Czytelnikowi wyjaśniać jak istotna może być bezpieczeństwo i ochrona danych w zastosowaniach militarnych, służbach specjalnych, dyplomacji, zastosowaniach administracyjnych, handlowych czy bankowych.

Dla bezpieczeństwa systemu komputerowego lub sieci komputerowej ważne są ogólnie rzecz biorąc:

- stosowane metody, algorytmy i protokoły kryptograficzne
- hardware kryptograficzny wspomagający bezpieczeństwo
- polityka bezpieczeństwa i sposoby przeprowadzania tzw. audytu bezpieczeństwa
- mechanizmy bezpieczeństwa wykorzystywanych systemów operacyjnych
- w końcu sama architektura systemu cyfrowego

Dobrze jest zdawać sobie sprawę z tego co z naszą wiadomością może się stać złego jeśli ją np. przesyłamy. Na drodze NADAWCA - ODBIORCA. wiadomość może być:

- przechwycona i odczytana przez osoby niepowołane

- zmieniona, zmodyfikowana (w całości lub częściowo) i przedstawiona jako oryginalna odbiorcy

- pochodzić od kogoś innego niż sądzi odbiorca wiadomości (sprawa tożsamości nadawcy)

Kryptologia bazuje na czterech działach matematyki:

1. teorii liczb
2. algebrze
3. teorii algorytmów komputerowych
4. teorii prawdopodobieństwa.

Kryptologię zaś dzieli się na ogół na dwa obszerne działy: kryptografię i kryptoanalizę, czyli

$$Kryptologia = Kryptografia + Kryptoanaliza.$$

Kryptografia zajmuje się głównie szyfrowaniem i deszyfrowaniem informacji. Kryptoanaliza zajmuje się natomiast odczytywaniem zaszyfrowanych danych, gdy nie jest znany klucz deszyfrujący mówimy wówczas o tzw. ataku (na szyfr) lub łamaniu szyfru. Przystąpienie do odczytywania zaszyfrowanych danych może się przy tym odbywać z różną informacją wstępną. Najoczywistszą metodą ataku jest przeglądanie przestrzeni K wszystkich kluczy i sprawdzanie czy otrzymujemy po deszyfracji sensowny tekst (być może otrzymamy ich wiele). Mówimy wówczas o tzw. ataku brutalnym.

Złamanie danego szyfru to odkrycie klucza używanego do szyfrowania w danym konkretnym przypadku lub opracowanie ogólnej metody, czy algorytmu łamania tego szyfru. W kryptoanalizie zakładamy z reguły, że ogólnie znane są zasady konstrukcji szyfru a nawet jego parametry. Jest to założenie praktycznie zawsze pełnione unika się bowiem stosowania szyfrów nie przebadanych od strony matematycznej.

Często obecnie nazywa się kryptologię niezbyt precyzyjnie kryptografią.

Główne cele kryptografii

Omówimy teraz krótko zadania, cele i środki techniczne kryptografii czyli "cryptographic objectives and primitives".

cele = objectivies (czyli o co chodzi)

środki = primitivies (to “cegiełki” za pomocą których realizujemy cel). Dokładniej są to elementarne metody, algorytmy i protokoły.

Główne cele kryptografii to:

1. Zapewnienie poufności lub jak mówimy prywatności (informacji, danych, wiadomości, dokumentu, tekstu) (ang. privacy lub confidentiality).

Sens: osoby niepowołane nie powinny mieć wglądu w nasze dane

Środki: szyfry i szyfrowanie (ang. encryption)

2. Zapewnienie integralności danych (ang. data integrity)

Sens: Dane będące ciągiem znaków nie mogą być w jakikolwiek sposób zmienione jeśli sobie tego nie życzymy. Jeśli są zmienione np. w wyniku przekłamań podczas transmisji czy celowego działania to powinniśmy móc to zawsze (lub z prawdopodoboeństwem bliskim 1) wykryć.

Środki: funkcje skrótu (ang. hash functions) a dokładniej funkcje skrótu bez klucza i funkcje skrótu z kluczem (ang. keyed hash functions)

3. Uwierzytelnianie wiadomości (uwierzytelnienie dokumentu) (ang. message authentication, document authentication).

Sens: Uwierzytelnianie wiadomości czy dokumentu pełni taką samą funkcję jaką pełni odręczny podpis w życiu codziennym. Dowiadujemy się od kogo pochodzi wiadomość lub dokument. Dla odbiorcy danych oprócz zachowania tajności przekazu bardzo istotna jest również na ogół informacja o autentyczności źródła wiadomości.

Środki: algorytmy podpisów cyfrowych (ang. digital signatures)

4. Uwierzytelnianie strony lub identyfikacja (ang. entity authentication lub entity identification)

Sens: Jest to identyfikacja strony pragnącej uzyskać dostęp do zasobów systemu komputerowego lub sieci komputerowej. Stroną może być np. osoba, komputer w sieci, karta kredytowa.

Środki: Środkami są tu odpowiednie metody uwierzytelniania np. metoda haseł, metoda pytanie-odpowiedź i specjalne protokoły uwierzytelniania.

5. Niezaprzeczalność (ang. non-repudation)

Sens: Istnieją metody kryptograficzne pozwalające zapobiec takiej sytuacji, że strona zaprzecza, że dokonała jakiś czynności (np. zaprzecza, że podjęła jakieś zobowiązania). W przypadku zaistnienia kontrowersji jesteśmy w stanie publicznie udowodnić prawdę.

Środki: Np. podpisy niezaprzeczalne

6. Certyfikacja (ang. certification)

Sens: Potwierdzenie, zaaprobowanie informacji przez zaufaną trzecia stronę np. specjalny urząd certifikacji lub osobę do której mamy zaufanie.

Środki: podpisy cyfrowe, infrastruktura klucza publicznego (PKI od ang. public key infrastructure)

7. Stemplowanie dokumentu czasem (ang. time stamping)

Sens: Wiązanie z dokumentem zapisu mówiącego kiedy ten dokument został utworzony (a dokładniej przeszedł przez serwer świadczący usługę stemplowania czasem).

Środki: podpisy cyfrowe, sieć dystrybucji jednolitego czasu

W każdym współczesnym systemie operacyjnym takim jak np. Unix, Windows 2003 czy w programach uzupełniających system operacyjny takich jak Norton Utilities mamy specjalne polecenia do szyfrowania plików lub systemu plików a ponadto oparty na kryptografii system haseł zapewniający dostęp do określonych zasobów systemu komputerowego (np. twardego dysku, drukarki, plików) tylko osobom upoważnionym.

Również wiele programów aplikacyjnych (np. MS Word 2003, MS Excel 2003) ma wbudowane własne systemy kryptograficzne.

W systemy kryptograficzne wyposażone są również z reguły przeglądarki (np. Internet Explorer i Mozilla) i programy pocztowe.

1.2 System kryptograficzny

Kilka pojęć podstawowych.

Zanim powiemy co to jest wiadomość jawna, kryptogram, szyfr i co oznaczają dokładnie wymienione wyżej cele kryptografii musimy wprowadzić kilka elementarnych pojęć lingwistyki matematycznej: pojęcie alfabetu, słowa nad alfabetem, języka itd.

Alfabet Alfabet to dowolny niepusty zbiór skończony (niekiedy dodajemy zbiór symboli lub zbiór liter). Alfabet tak jak zbiór oznaczamy najczęściej dużymi literami alfabetu łacińskiego np. V.

Np. zbiory $V = \{0,1\}$, $V = \{0,1,2,...,9\}$, $V = \{a,b,c,....,x,y,z\}$ są alfabetami.

Istota rzeczy: Właściwie pojęcie alfabetu jest intuicyjnie jasne a rola dokładnie taka sama jaką pełni alfabet w języku naturalnym. Alfabet służy do zapisywania słów. Warto zwrócić uwagę, na fakt, że to co nazywamy symbolem w formalnej definicji alfabetu jest oczywiście kwestią umowy.

Słowo nad alfabetem V Słowo nad alfabetem V to dowolny ciąg skończony o wartościach w zbiorze V. Czasami słowo nad ustalonym alfabetem nazywamy też *tekstem, napisem* lub *stringiem.* Ilość wyrazów ciągu nazywamy długością słowa np. słowo *abccd* nad alfabetem $V = \{a,b,c,d\}$ ma długość 5 a słowo *ala* długość 3.

Istota rzeczy: Formalne pojęcie "słowo nad alfabetem" z powyższej definicji odpowiada słowu języka naturalnego, jednak z tak rozumianym bardzo formalnie słowem nie musimy wiązać żadnego znaczenia.

Również wiadomość w szczególności tzw. wiadomość jawna i wiadomość zaszyfrowana będą dla nas słowami nad ustalonym alfabetem. Wiadomość zaszyfrowana nazywa się też kryptogramem lub szyfrogramem.

Język. Zbiór wszystkich słów nad alfabetem V oznaczamy symbolem V^* a dowolny niepusty podzbiór tego zbioru nazywamy *językiem.* Do zbioru V^* zaliczamy również słowo puste (oznaczane na ogół symbolem ε). Słowo puste ma długość 0. Zbiór V^* jest oczywiście nieskończony ale przeliczalny mamy bowiem $V^* = \{\varepsilon\} \cup V \cup V^2 \cup ... \cup V^n \cup ....$ W zbiorze V^* definiujemy działanie dwuargumentowe $\circ : V^* \times V^* \to V^*$ tzw. konkatenację (ang. concatenation). Jeśli $\alpha, \beta \in V^*$ i $\alpha = a_1 a_2 ....a_n$ $\beta = b_1 b_2 ....b_n$ to z definicji mamy

$$\alpha \circ \beta = a_1 a_2 ....a_n \circ b_1 b_2 ...b_n \overset{df}{=} a_1 a_2 ...a_n b_1 b_2 ...b_n .$$

Jak widać konkatenacja jest zestawianiem słów np. *ala* konkatenowane z *makota* daje *alamakota* podobnie *kot* z *let* daje *kotlet.*

Jak łatwo sprawdzić działanie konkatenacji jest łączne (tzn. dla każdego $\alpha, \beta, \gamma \in V^*$ mamy $(\alpha \circ \beta) \circ \gamma = \alpha \circ (\beta \circ \gamma)$) . Mamy ponadto dla każdego $\alpha \in V^*$; $\alpha \circ \varepsilon = \varepsilon \circ \alpha = \alpha$ zatem słowo puste ε jest jedynką konkatenacji. Zbiór V^* z działaniem konkatenacji $\circ : V^* \times V^* \to V^*$ jest więc monoidem (czyli półgrupą z jednością).

Istota rzeczy: Jeśli weźmiemy kilka słów a może nieskończoną liczbę słów to mamy język. Możemy 2 słowa zestawiać razem tworząc nowe słowo. Nazywa się to konkatenacją.

Kody. Niech V_1 będzie zbiorem obiektów kodowanych a V_2 dowolnym alfabetem. Kod najogólniej rzecz biorąc to relacja dwuargumentowa $k \subseteq V_1 \times V_2^*$ spełniająca warunek: dla każdego $x_1 \in V_1$ czyli dla każdego elementu ze zbioru obiektów kodowanych istnieje takie $x_2 \in V_2^*$, czyli takie słowo nad alfabetem V_2, że $(x_1, x_2) \in k$. Elementy V_2^* nazywamy słowami kodowymi. Jeśli $V_2 = \{0,1\}$ to kod nazywamy kodem binarnym lub dwójkowym.

W szczególności dowolne odwzorowanie $f : V_1 \to V_2^*$ nazywamy kodem, żądamy przy tym najczęściej by funkcja f była różnowartościowa i tak najczęściej rozumiane jest pojęcie kodu. W dalszym ciągu pod pojęciem kodu będziemy rozumieć tę najbardziej restryktywną definicję.

Jeśli istnieje takie $n \in N$, że dla każdego $x \in V_1$ słowo kodowe $f(x) \in V_2^*$ ma stałą długość *n* to kod nazywamy kodem o stałej długości lub dokładniej kodem o stałej długości słowa kodowego. Oczywiście mamy wówczas $f : V_1 \to V_2^n$. Jeśli dodatkowo $f(V_1)$ jest właściwym podzbiorem zbioru V_2^n to kod nazywamy kodem redundancyjnym lub nadmiarowym

Istota rzeczy: Obiekty chcemy "etykietować" , opisywać za pomocą słów nad ustalonym alfabetem i czynimy to za pomocą kodu. Jest to wygodne i naturalne zwłaszcza jeśli chcemy reprezentować obiekty różnego typu w systemie komputerowym. Z każdym obiektem wiążemy dokładnie jedno słowo.

System kryptograficzny lub szyfr.

System kryptograficzny inaczej szyfr to piątka uporządkowana (V_1, V_2, K, E, D), gdzie:

- V_1 jest alfabetem, w którym zapisujemy wiadomości jawne, $V_1^* \overset{df}{=} M$ czyli zbiór wszystkich słów nad alfabetem V_1 jest tzw. przestrzenią wiadomości jawnych (M od ang. message) a każdy element M nazywamy wiadomością jawną lub tekstem jawnym i oznaczamy symbolem *m* lub *P* (od ang. plain text).

- V_2 jest alfabetem, w którym zapisujemy kryptogramy (szyfrogramy) $V_2^* \overset{df}{=} C$ czyli zbiór wszystkich słów nad alfabetem V_2 jest tzw. przestrzenią wiadomości zaszyfrowanych czyli jak mówimy szyfrogramów (szyfrogram nazywamy też kryptogramem, wiadomością zaszyfrowaną, lub tekstem zaszyfrowanym). Najczęściej alfabety V_1 i V_2 są tym samym afabetem czyli $V_1 = V_2$ ale tak być nie musi.

- K jest dowolnym zbiorem jest to tzw. przestrzeń kluczy (ang. *key space*), każdy element $k \in K$ nazywamy kluczem (ang. key) (lub kluczem szyfrującym). Na ogół K jest zbiorem skończonym ale w definicji ogólnej systemu kryptograficznego nie robimy tego założenia.

- E jest funkcją

$$E : M \times K \ni (m,k) \to E(m,k) \in C ,$$

obliczanie wartości funkcji E nazywamy szyfrowaniem a samą funkcję przekształceniem szyfrującym lub co jest nieco mylące również szyfrem. Wartość $E(m,k) \in C$ nazywamy wiadomością zaszyfrowaną, szyfrogramem, kryptogramem lub tekstem zaszyfrowanym. Sposób obliczania $E(m,k)$ dla danych k, m nazywamy *algorytmem szyfrowania.* Klucz k nazywamy *kluczem szyfrującym* i często oznaczamy go też symbolem e (od ang. encryption czyli szyfrowanie)

- D jest funkcją

$$D : C \times K \ni (c,k) \to D(c,k) \in M$$

obliczanie wartości funkcji D nazywamy *deszyfrowaniem* a samą funkcję D przekształceniem deszyfrującym. Sposób obliczania $D(c,k) \in M$ dla danych k, m nazywamy *algorytmem deszyfrowania a* klucz k nazywamy *kluczem deszyfrującym* i często oznaczamy go symbolem d (od ang. decryption czyli deszyfrowanie)

UWAGA. Od każdego systemu kryptograficznego wymagamy by:

dla każdego *klucza szyfrującego* $k_1 \in K$ istniał *klucz deszyfrujący* $k_2 \in K$ taki, żeby dla każdej wiadomości jawnej $m \in M$ zachodziła równość:

$$D(E(m,k_1),k_2) = m .$$

Wniosek. Z powyższego warunku wynika, że dla każdego klucza $k_1 \in K$ funkcja szyfrująca $E(\cdot, k_1) : M \to C$ jest różnowartościowa (choć nie musi być bijekcją czyli funkcją różnowartościową i na). Dwie różne wiadomości jawne mają więc różne kryptogramy. Ponadto jeśli $k_2 \in K$ jest kluczem rozszyfrowującym dla ustalonego klucza $k_1 \in K$ to funkcja $D(\cdot, k_2) : E(M, k_1) \to M$ jest różnowartościowa.

Z definicji systemu kryptograficznego wynika, że szyfr (czyli system kryptograficzny) to mówiąc niezbyt precyzyjnie dwie rodziny kodów $(E(\cdot, k))_{k \in K}$ i $(D(\cdot, k))_{k \in K}$ sparametryzowane kluczem spełniające pewne dodatkowe warunki.

Istota rzeczy:

Szyfr służy do zastąpienia wiadomości jawnej *m* kryptogramem *c*. Celem jest takie przekształcenie postaci wiadomości jawnej by uzyskać postać nieczytelną dla osób nie dysponujących kluczem deszyfrującym. Realizujemy w ten sposób jeden z zasadniczych celów kryptografii tzw. poufność (ang. confidentiality).

Każdą wiadomość zaszyfrowaną musimy umieć jednoznacznie rozszyfrować ale być może służy do tego już inny klucz.

Mówiąc niezbyt precyzyjnie szyfr to 2 rodziny kodów sparametryzowane kluczem.

Krótki komentarz na temat długości klucza.

Pewnego komentarza wymaga pojęcie klucza. W definicji systemu kryptograficznego wyróżniliśmy pewien zbiór *K* zwany przestrzenią kluczy (ogólnie rzecz biorąc zbiór *K* jest dowolny), którego elementy są nazywane kluczami i parametryzują odwzorowanie szyfrujące. Na ogół im więcej elementów ma zbiór *K* tym szyfr jest bezpieczniejszy trudniej bowiem dopasować wtedy klucz *k* do przechwyconego kryptogramu *c* tak by było $D(c,k) = m$ metodą przeglądania wszystkich kluczy *k* czyli za pomocą tzw. ataku brutalnego.

Klucz w praktyce to słowo nad jakimś ustalonym alfabetem, na ogół alfabetem w którym zapisujemy wiadomość jawną, w szczególności może to być alfabet binarny $\{0,1\}$ i wówczas z reguły $K = \{0,1\}^n$ dla pewnego $n \in N$ lub $K \subseteq \{0,1\}^*$. Jeśli klucz jest słowem nad ustalonym alfabetem to możemy mówić o długości klucza.

Na ogół im dłuższy jest klucz tym bezpieczniejszy jest system kryptograficzny. Oczywiście technicznie rzecz biorąc klucz może mieć dowolną długość jednak im dłuższy klucz tym na ogół dłuższy czas szyfrowania choć oczywiście zależy to również od algorytmu szyfrowania. Ponadto dłuższy klucz jest bardziej kłopotliwy. Nie trzeba tego wyjaśniać osobom, które robiąc zakupy kartą nie mogą sobie przypomnieć przy kasie krótkiego przecież PIN'u.

W wielu krajach (np. w USA czy we Francji) istnieją pewne ograniczenia o charakterze prawnym na długość stosowanego klucza.

Klucze 128 bitowe to standard w USA i w Polsce do zakupów sieciowych z kartą kredytową i w tranzakcjach bankowych. W powszechnym mniemaniu klucze 128 bitowe uważa się za bezpieczne ale rzecz jasna taka zasada jest półprawdą, bowiem klucza nie można rozpatrywać w oderwaniu od systemu kryptograficznego np. 128 bitowy czy nawet 512 bitowy klucz w przypadku szyfru RSA nie jest kluczem bezpiecznym. Przyjmuje się, że dla RSA bezpieczna długość klucza to 1024 bity lub więcej. ■

Uwaga *Algorytmy kryptograficzne* (czyli algorytmy szyfrowania i deszyfrowania a także cały szereg innych algorytmów wykorzystywanych w kryptografii jak np. obliczanie tzw. funkcji skrótu czy tworzenie podpisów cyfrowych) można realizować:

1. programowo (pisząc odpowiednie programy szyfrujące w jakimś języku programowania np. w asemblerze, Pascalu, C, C++, czy Javie.

2. za pomocą specjalizowanych układów szyfrujących (z reguły są to pojedyncze układy scalone ASIC'i takie jak np. układ do szyfrowania danych o nazwie "Clipper" lub specjalizowane karty rozszerzeń do komputera klasy PC)

3. za pomocą metod mieszanych łącząc odpowiednio metody programowe i sprzętowe.

Realizacje wykorzystujące głównie hardware uważane są za rozwiązania najbardziej bezpieczne, najbardziej profesjonalne.

Zasadniczy podział sytemów kryptograficznych.

Systemy kryptograficzne, szyfry, algorytmy kryptograficzne dzielimy na

1) symetryczne (inaczej z kluczem symetrycznym)

2) asymetryczne (inaczej z kluczem asymetrycznym).

Jeśli do szyfrowania i rozszyfrowania używamy tego samego klucza (lub nieco ogólniej pary kluczy $k_1, k_2 \in K$ takich że z klucza szyfrującego $k_1 \in K$ daje się łatwo obliczyć klucz deszyfrujący $k_2 \in K$ i odwrotnie z klucza deszyfrującego $k_2 \in K$ daje się łatwo obliczyć klucz szyfrujący $k_1 \in K$) to system kryptograficzny, szyfr czy algorytm kryptograficzny nazywamy systemem kryptograficznym, szyfrem czy algorytmem kryptograficznym z kluczem symetrycznym.

Jeśli szyfr jest taki, że do szyfrowania i rozszyfrowania używamy różnych kluczy (kluczy $k_1, k_2 \in K$) takich że z klucza szyfrującego $k_1 \in K$ nie daje się łatwo obliczyć klucza deszyfrującego $k_2 \in K$ i odwrotnie z klucza deszyfrującego $k_2 \in K$ nie daje się łatwo obliczyć klucza szyfrującego $k_1 \in K$ to system kryptograficzny, szyfr czy algorytm algorytm

kryptograficzny nazywamy systemem kryptograficznym, szyfrem czy algorytmem kryptograficznym z kluczem asymetrycznym.

Systemy z kluczem prywatnym i systemy z kluczem publicznym.

Systemy kryptograficzne z kluczem symetrycznym nazywamy też systemami z kluczem prywatnym. Przykłady takich szyfrów to DES, 3DES, DESX, AES, IDEA, LOKI, Twofish i Blowfish.

Systemy z kluczem asymetrycznym nazywamy też systemami z kluczem publicznym Przykłady takich szyfrów szyfr RSA, szyfr Rabina, szyfr ElGamala, probabilistyczny szyfr z kluczem publicznym, szyfry plecakowe np. szyfr plecakowy Merklego-Hellmanna i szyfr plecakowy Chora-Rivesta .

Określenie: "silna kryptografia" oznacza szyfry trudne do złamania np. stosuje się w tym celu długie klucze. Pod pojęciem silnej kryptografii rozumie się na ogół algorytm kryptograficzny z kluczem dłuższym od 128 bitów (w przypadku algorytmów z kluczem prywatnym) i z kluczem dłuższym od 2048 bitów (w przypadku algorytmów z kluczem publicznym).

Bloki, blokowa funkcja szyfrująca i szyfry blokowe.

Funkcję szyfrującą $E : M \times K \to C$ wygodnie jest zdefiniować za pomocą różnowartościowej tzw. *blokowej funkcji szyfrującej* $f : V_1^n \times K \to V_1^m$. Blokowa funkcja szyfrująca $f : V_1^n \times K \to V_1^m$ musi spełniać warunek: dla każdego $k \in K$ funkcja $f(\cdot\ ,k) : V_1^n \to V_1^m$ jest różnowartościowa.

Z kolei funkcję deszyfrującą $D : C \times K \to M$ wygodnie jest zdefiniować za pomocą *blokowej funkcji deszyfrującej* $g : V_2^m \times K \to V_2^n$

Jeśli $V_1 = V_2$ to musimy mieć oczywiście $m \geq n$ by zachować różnowartościowość funkcji $f(\cdot\ ,k) : V_1^n \to V_1^m$. Najczęściej jednak w praktyce $V_1 = V_2$ i $m = n$.

Blokową funkcję szyfrującą $f : V_1^n \times K \to V_1^m$ nazywamy też *przekształceniem szyfrującym dla tekstów jawnych o stałej długości.*

Blokową funkcję szyfrującą wraz z blokową funkcją deszyfrującą nazywamy szyfrem blokowym.

Standardowym postępowaniem przy szyfrowaniu długich tekstów jawnych jest podział takiego tekstu na tzw. *jednostki tekstu* lub *bloki* czyli słowa o stałej długości *n,* które skonkatenowane dają wiadomość jawną *m*. Każdą jednostkę tekstu możemy szyfrować wówczas niezależnie za pomocą blokowej funkcji szyfrującej $f : V_1^n \times K \to V_2^m$. Jednostkom tekstu (lub jak mówimy czasem blokom) o długości *n* blokowa funkcja szyfrująca przyporządkowuje fragment szyfrogramu o długości *m*.

Dzielenie długiego tekstu wiadomości jawnej na krótsze jednostki tekstu jest bardzo wygodne ale wymaga niekiedy przedłużenia wiadomości jawnej m tak by ta długość była równa $r \cdot n$ dla pewnego $r \in N$.

Istota rzeczy: Szyfrujemy długie teksty jawne "po kawałku". Ten "kawałek" nazywamy blokiem lub jednostką tekstu. Powstaje rzecz jasna od razu problem a jeśli to się nie da podzielić na jednakowe „kawałki" to co. Rozwiązanie jest oczywiste. Dopełniamy tekst szyfrowany do wielokrotności długości bloku. Najczęściej jest to zgodne z pewnymi standardami.

Szyfry strumieniowe (ang. stream ciphers) są ważna klasa szyfrów z kluczem symetrycznym. Należą do klasy szyfrów blokowych, (z długością bloku =1, a więc oddzielnie szyfrujemy każdą literę tekstu jawnego). Istotą szyfru strumieniowego jest to, że przekształcenie szyfrujące może zmieniać się dla każdego szyfrowanego symbolu.

Szyfry strumieniowe są użyteczne w sytuacji gdy,

1) wysokie jest prawdopodobieństwo błędów transmisji, ponieważ w szyfrach strumieniowych nie ma propagacji błędu;

2) dane muszą być przesyłane symbol po symbolu (np. szyfrator lub deszyfrator nie ma pamięci).

Zdefiniujemy teraz szyfr strumieniowy formalnie. Niech V_1 i V_2 będą alfabetami. Punktem wyjścia jest blokowa funkcja szyfrująca (szyfrująca bloki o długości 1) $E : K \times V_1 \to V_2$ i blokowa funkcja deszyfrująca (deszyfrująca bloki o długości 1) $D : K \times V_2 \to V_1$, spełniające warunek:

$$\forall_{m \in V_1} \; \forall_{k_1 \in K} \; \exists_{k_2 \in K} \; D(k_2, E(k_1, m)) = m .$$

Tworzymy ciąg $(k_i)_{i=1}^{\infty}$, gdzie $k_i \in K$ dla każdego $i \in N$. Nazywamy ten ciąg strumieniem kluczy. Definiujemy nowy system kryptograficzny $(V_1, V_2, \widetilde{K}, \widetilde{E}, \widetilde{D})$ następująco. V_1 i V_2 są wprowadzonymi już alfabetami, $\widetilde{K} = \{(k_i)_{i=1}^{\infty}; k_i \in K\}$ zbiorem kluczy, $\widetilde{E} : \widetilde{K} \times V_1^* \to V_2^*$ funkcją szyfrującą, dokładniej $\widetilde{E}$ zadane jest wzorem:

$$\widetilde{E}((k_i)_{i=1}^{\infty}, \; m_1 m_2 ... m_r) = E(k_1, m_1) E(k_2, m_2) ... E(k_r, m_r) = c_1 c_2 ... c_r$$

$\widetilde{D} : \widetilde{K} \times V_2^* \to V_1^*$ funkcją deszyfrującą, dokładniej $\widetilde{D}$ zadane jest wzorem:

$$\tilde{D}((k_i)_{i=1}^{\infty},\ c_1c_2...c_r) = D(k_1,c_1)D(k_2,c_2)...D(k_r,c_r) = m_1m_2...m_r$$

Taki system kryptograficzny nazywamy szyfrem strumieniowym. W gruncie rzeczy powyższa definicja jest bardzo podobna do definicji szyfru polialfabetowego.

Probabilistyczne systemy kryptograficzne

Jak uwzględnić losowość w szyfrowaniu. W pewnych algorytmach kryptograficznych takich jak np. algorytm z kluczem publicznym ElGamela (z uwagi na losowość „wmontowaną" w algorytm), ustalonej jednostce tekstu jawnego mogą odpowiadać różne szyfrogramy. Tak więc jeśli G jest zbiorem skończonym lub przeliczalnym a $(G,2^G,P)$ przestrzenią probabilistyczną to blokowe przekształcenie szyfrujące definiujemy jako $f: V_1^{l_1} \times G \to V_2^{l_2}$ dodając warunek, że

$$\underset{g\in G}{\forall} f(\cdot,g): V_1^{l_1} \to V_2^{l_2} \text{ jest funkcją różnowartościową}$$

Informacja o wybranym w losowy sposób $g \in G$ albo
- nie jest istotna przy deszyfrowaniu np. alternatywnie możemy ustalonej jednostce tekstu (w szczególności literze) przyporządkowywać zależnie od g różne szyfrogramy (tak jest w tzw. szyfrze homofonicznym) albo

- jest zawarta i przesyłana w samym szyfrogramie, wbudowana w szyfrogram (tak jest np. w systemie kryptograficznym ElGamela).

Ogólnie rzecz biorąc funkcja szyfrująca E z definicji systemu kryptograficznego jest w tej sytuacji funkcją $E: M \times K \times H \to C$, gdzie H jest dowolnym skończonym zbiorem a $(H,2^H,P)$ jest pewną przestrzenią probabilistyczną a funkcja deszyfrująca $D: C \times K \times H \to M$.

Istota rzeczy: Fakt losowej zmienności kryptogramu dla tej samej wiadomości jawnej bez wątpienia utrudnia kryptoanalizę.

Przykład. Tzw. szyfry homofoniczne (a ściślej: szyfry podstawieniowe homofoniczne) stanowią przykład szyfru probabilistycznego. Również szyfr ElGamala jest przykładem szyfru probabilistycznego.■

Funkcje jednokierunkowe (ang. one way function).

Funkcja jednokierunkowa to inaczej funkcja zapadkowa.

Funkcja $f : X \to Y$, gdzie *X, Y* są dowolnymi niepustymi zbiorami, jest nazywana funkcją jednokierunkową jeśli:

1) Dla każdego $x \in X$ łatwo potrafimy obliczyć wartość *f(x)*.

2) Dla każdego $y \in f(X)$, znalezienie takiego $x \in X$, że $y = f(x)$ jest praktycznie nierealizowalne (z uwagi na złożoność obliczeniową problemu).

Nie znamy żadnego dowodu na to, że jakaś funkcja jednokierunkowa w ogóle istnieje. Jesteśmy jednak przekonani, że funkcje jednokierunkowe w przyrodzie są. Na ogół dąży się do tego by zdefiniować funkcję f bardzo prosto i tak by obliczanie wartości tej funkcji było łatwe do implementacji software'owej i hardware'owej.

A oto trzy kandydatki na funkcje jednokierunkowe. Są to permutacje podzbioru liczb całkowitych Z_p. Ogólnie rzecz biorąc jednak funkcja jednokierunkowa nie musi być różnowartościowa.

1. Funkcja wykładnicza modulo p (ang. exponentation modulo *p*), gdzie p jest liczbą pierwszą. Niech p będzie liczbą pierwszą i niech α będzie generatorem grupy multiplikatywnej Z_p^*. Funkcja wykładnicza modulo p $f : Z_p^* \to Z_p^*$ jest zdefiniowana dla każdego $x \in Z_p^*$ wzorem

$$f(x) = \alpha^x \pmod p,$$

lub wzorem $f(x) = \alpha^x$, jeśli mnożenie traktujemy jako mnożenie w Z_p. Odwrócenie funkcji f jest dokładnie problemem znalezienia wartości logarytmu dyskretnego w grupie multiplikatywnej Z_p^* (jeśli $y = f(x)$, czyli $y = \alpha^x$, to $x = \log_\alpha y$). Nie są znane efektywne obliczeniowo algorytmy obliczania wartości logarytmu dyskretnego w grupie Z_p^*.

2. Funkcja RSA (ang. RSA function). Niech p, q będą różnymi nieparzystymi liczbami pierwszymi i niech $n = p \cdot q$. Niech ponadto $NWD(e,(p-1)(q-1)) = 1$. Funkcja RSA zdefiniowana jest tak $f : Z_n \to Z_n, f(x) = x^e \pmod n$ dla każdego $x \in Z_n$.

3. Funkcja Rabina. Niech $n = p \cdot q$ gdzie p i q są różnymi liczbami pierwszymi takimi, że $p \equiv 3 \pmod 4$ i $q \equiv 3 \pmod 4$ (*n* jest tzw. liczbą Bluma). Funkcja Rabina f zdefiniowana jest następująco

$$f : Q_n \ni x \to f(x) = x^2 (\mathrm{mod}\, n) \in Q_n$$

gdzie $Q_n \subset Z_n$ jest tzw. zbiorem reszt kwadratowych modulo n. Można pokazać, że f jest permutacją zbioru Q_n. Odwracanie funkcji f, czyli obliczanie pierwiastka kwadratowego w pierścieniu Z_n jest dla dużych n praktycznie nierealizowalne, tzn. nie są znane żadne efektywne obliczeniowo algorytmy rozwiązujące ten problem, chyba, ze potrafimy rozłożyć liczbę n na czynniki pierwsze.

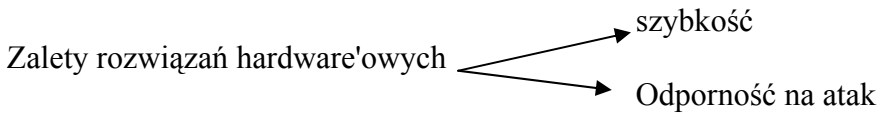
4. Podnoszenie do ustalonej potęgi $n \in Z$ w grupie skończonej G (w szczególności grupie multiplikatywnej ciała skończonego) jest proste z obliczeniowego punktu widzenia. Przyjmijmy więc

$$f : G \ni a \to b = a^n \in G \ .$$

Okazuje się, że obliczenie relacji odwrotnej do f sprowadzające się do obliczenia pierwiastka n-tego stopnia z b jest na ogół trudne obliczeniowo.

Realizacja systemów kryptograficznych.

Systemy kryptograficzne można realizować software'owo, hardware'owo lub w sposób mieszany czyli częściowo hardware'owo częściowo software'owo. Na ogół uważa się, że im więcej specjalizowanego hardware'u tym system jest bezpieczniejszy. Współczesne profesjonalne systemy kryptograficzne są z reguły realizowane hardware`owo, wewnątrz specjalizowanych układów szyfrujących są to na ogół – chipy - układy typu ASIC (Application Specific Integrated Circuit) lub układy typu PLD (Programmable Logic Design) . Zaszycie algorytmu kryptograficznego w układzie scalonym typu ASIC lub PLD nie stanowi współcześnie większego problemu.



Realizując systemy kryptograficzne trzeba zwracać uwagę na zastrzeżenia eksportowe i patenty.

1.3 Rodzaje szyfrów

Omówimy w dalszym ciągu szyfry przestawieniowe, podstawieniowe i przestawieniowo-podstawieniowe

Szyfry przestawieniowe czyli szyfry permutacyjne.

Szyfry permutacyjne (ang. transposition ciphers) należą do klasy blokowych szyfrów symetrycznych.

Niech $t \in N$ oznacza ustaloną długość bloku (czyli jednostki tekstu). Kluczem jest w szyfrach przestawieniowych permutacja $g :< 1, t > \rightarrow < 1, t >$ a zbiór wszystkich takich permutacji jest zbiorem kluczy *K*.

Warto zauważyć, że $card(K) = t!$ szybko rośnie wraz ze wzrostem *t*. Korzystając ze znanego wzoru Stirlinga (*)

$$\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}\, n^n e^{-n}} = 1 \qquad (*)$$

można z dobrym przybliżeniem oszacować wartość $t!$. Mamy mianowicie

$$t! \cong \sqrt{2\pi t}\, t^t e^{-t} = \sqrt{2\pi t} \left(\frac{\mathrm{t}}{\mathrm{e}} \right)^t$$

co np. dla *t*=128 daje $t! \cong \sqrt{2\pi 100}\, (128/\mathrm{e})^{128} \geq 20 \cdot (40)^{128}$.

Załóżmy, że alfabety V_1 i V_2 są takie, że $V_1 = V_2 \overset{ozn.}{=} V$ czyli alfabet wiadomości jawnych i alfabet wiadomości zaszyfrowanych są tym samym alfabetem.

Niech $m = m_1 m_2 ... m_t \in M$ będzie dowolną wiadomością jawną. Dla $i = 1,2,...,r, m_i \in V$. Oznaczmy przez f blokową funkcję szyfrującą bloki (czyli jednostki tekstu) o długości *t*.

$$f : V^t \times K \rightarrow V^t$$

Blokowa funkcja szyfrująca zdefiniowana jest wzorem

$$f : V^t \times K \ni (m_1 m_2 ... m_t, g) \rightarrow f(m_1 m_2 ... m_t, g) = m_{g(1)} m_{g(2)} ... m_{g(t)} \in V^t$$

Zatem wiadomości jawnej m o długości t przyporządkowujemy kryptogram o długości t ustawiając w pewnej kolejności (wyznaczonej przez permutację g) wyrazy ciągu stanowiącego wiadomość jawną.

W celu zdefiniowania funkcji $E: V_1^* \times K \to V_1^*$ czyli rozszerzenia blokowej funkcji szyfrującej f zakładamy wstępnie, że tekst jawny m ma długość $k \cdot t$, gdzie $k \in N$ (czyli długość tekstu jawnego jest wielokrotnością długości jednostki tekstu) i definiujemy

$$E(m_1 m_2 ... m_t m_{1+t} m_{2+t} ... m_{2t} .... m_{(k-1)t+1} m_{(k-1)t+2} ... m_{kt}, g) =$$
$$= m_{g(1)} m_{g(2)} ... m_{g(t)} m_{g(1)+t} m_{g(2)+t} ... m_{g(t)+t} .... m_{(k-1)t+g(1)} m_{(k-1)t+g(2)} ... m_{(k-1)t+g(t)}$$

Jeśli tekst jawny m ma długość różną od $k \cdot t$ to wydłużamy go do długości kt dodając do tekstu pewną liczbę ustalonych znaków z alfabetu V_1 np. spacji, tak by nie zdeformować treści informacyjnej m. Podobnie postępujemy zresztą dla wszystkich szyfrów blokowych.

Istota rzeczy: Kluczem jest permutacja zbioru <1,*t*>, gdzie *t* jest długością jednostki tekstu (bloku).

Szyfry podstawieniowe

Prosty szyfr podstawieniowy. Prosty szyfr podstawieniowy (ang. simple substitution cipher) to szyfr blokowy o długości bloku 1. Niech V_1 będzie ustalonym alfabetem a $f: V_1 \to V_2$ funkcją różnowartościową (z reguły $V_1 = V_2$). Jeśli wiadomość jawna $m = m_1 m_2 ... m_r$ to szyfrogram $c = f(m_1) f(m_2) ... f(m_r)$. Funkcja f jest tu kluczem szyfrującym a f^{-1} kluczem deszyfrującym.

Klasycznym przykładem takiego szyfru jest szyfr Cezara. Szyfr Cezara będzie on omówiony dokładniej w dalszym ciągu rozdziału.

Polialfabetowy szyfr podstawieniowy.

Polialfabetowy szyfr podstawieniowy (ang. polyalphabetic substitution cipher) to szyfr blokowy o długości bloku *t*. Niech V_1 będzie ustalonym alfabetem a $f_i: V_1 \to V_2$, funkcją różnowartościową dla $i = 1,2,...,t$ (z reguły $V_1 = V_2$). Jeśli wiadomość jawna $m = m_1 m_2 ... m_t$ to szyfrogram

$$c = f_1(m_1) f_2(m_2) ... f_t(m_t).$$

Funkcja $f_1 \times f_2 \times ... \times f_t$ jest tu kluczem szyfrującym a $f_1^{-1} \times f_2^{-1} \times ... \times f_t^{-1}$ kluczem deszyfrującym. Liczbę t nazywamy okresem klucza szyfru polialfabetowego.

Szyfr produktowy lub złożeniowy.

Produktem lub *złożeniem 2 systemów kryptograficznych* o blokowych przekształceniach szyfrujących $f_1 : V_1^n \times K_1 \to V_2^m$ i $f_2 : V_2^m \times K_2 \to V_3^r$ nazywamy system kryptograficzny zdefiniowane przez blokowe przekształcenie szyfrujące $g : V_1^n \times (K_1 \times K_2) \to V_3^r$ zadane wzorem $g(m,(k_1,k_2)) = f_2(f_1(m,k_1),k_2)$, gdzie $m \in V_1^n$ oraz $k_1 \in K_1$, $k_2 \in K_2$. Szyfr zdefiniowany przez blokowe przekształcenie szyfrujące $g : V_1^n \times (K_1 \times K_2) \to V_3^r$ nazywa się szyfrem produktowym (ang. product cipher) lub szyfrem złożeniowym. Najczęściej w praktyce $n = m = r$ czyli operujemy stałą długością bloku.

Runda. Typowy szyfr złożeniowy tworzymy składając szyfr permutacyjny z szyfrem podstawieniowym co tworzy tzw. rundę. Składanie rund z kolei jest ogólnym często stosowanym pomysłem na tworzenie szyfrów blokowych z kluczem symetrycznym. Składając rundy tworzymy nowy szyfr zwany siecią permutacyjno - podstawieniową (substitution-permutation network).

Składając odpowiednio dobrane rundy tworzymy bardzo mocne szyfry. Tak skonstruowane są m.in. szyfry DES, IDEA i AES (Rijndael).

Uwaga. Nie zawsze (jakby się wydawało na pierwszy rzut oka) złożenie dwu systemów kryptograficznych prowadzi do istotnie nowego czy lepszego systemu kryptograficznego

Istota rzeczy: Szyfr produktowy tworzymy "stosując jeden po drugim" dwa lub większą ilość szyfrów. Na ogół prowadzi to do znacznie mocniejszych szyfrów.

Ważna uwaga o tajności metod i algorytmów kryptograficznych

Nigdy nie budujemy bezpieczeństwa systemu na ukrywaniu metody czy algorytmu. Wprost przeciwnie algorytm publicznie znany o którym wiadomo, że był atakowany bez powodzenia można uznać za pewny i bezpieczny.

1.4 Szyfry klasyczne

Omówienie wszystkich znanych szyfrów klasycznych wymagałoby osobnej książki. Poniżej zostanie podanych jedynie kilka bardziej znanych szyfrów: szyfr Cezara, szyfr Vigenere'a., szyfr Vernama (tzw. szyfr idealny), szyfr Hilla a również szyfry podstawieniowo-przestawieniowe stanowiące punkt wyjścia dla blokowych szyfrów z kluczem prywatnym. Parę słów poświęcimy również maszynie szyfrującej Enigma.

1. Szyfr Cezara

Szyfr Cezara jest typowym szyfrem podstawieniowym. Jest to szyfr z kluczem symetrycznym tzn. ten sam klucz będzie służył do szyfrowania i deszyfrowania.

Rozważmy dla ustalenia uwagi 26 literowy alfabet języka angielskiego $V = \{a, b, c, ..., z\}$. Możemy utożsamić litery z liczbami z pierścienia Z_{26} na zasadzie $a \sim 0$, b~1, c~2, d~3, ..., y~24, z~25 i przyjąć, że $V = \{a, b, c, ..., z\} = Z_{26}$. Weźmy teraz $V_1 = V_2 = V = \{0,1,...,26\}$ oraz przyjmijmy jako przestrzeń kluczy $K = \{0,1,...,25\}$. Kluczem w szyfrze Cezara jest więc liczba naturalna ze zbioru $\{0,...,25\}$.

Jeśli $k = 0$, to jak się za chwilę okaże, nie mamy żadnego utajnienia informacji. Użyteczne mogą być więc tylko klucze $k = 1,2,...,25$.

Wprowadźmy permutację $\pi_k : V \to V$ wzorem $\pi_k(x) = (x + r)(\mathrm{mod}\, 26)$

Oznaczmy wiadomość jawną przez $m = m_1 m_2 m_3 ... m_r$. Definiujmy odwzorowanie szyfrujące następująco:

$$E : M \times K \ni (m_1 m_2 ... m_r, k) \to c_1 c_2 ... c_r \in C$$

gdzie $c_i = \pi_k(m_i) = m_i \oplus_{26} k$, oraz $M = V^*$, $C = V^*$.

Kryptogram wiadomości jawnej $m = m_1 m_2 m_3 ... m_r$ otrzymujemy więc jako

$$\pi(m_1)\pi(m_2)\pi(m_3)...\pi(m_r)$$

Odwzorowanie deszyfrujące definiujemy wzorem

$$D : C \times K \ni (c_1 c_2 ... c_r, k) \to c_1 c_2 ... c_r \in C$$

gdzie $m_i = \pi_k^{-1}(m_i) = c_i \oplus_{26} (-k)$ (czyli $m_i = c_i \oplus_{26} (26 - k)$ lub $m_i = c_i -_{26} k$)

Tak zdefiniowany szyfr, nazywamy szyfrem Cezara z przesunięciem *k*. Oczywiście *k* jest kluczem szyfrującym i deszyfrującym jednocześnie.

Przykład. Jeśli mamy tekst jawny *abc* to dla klucza *k*=2 dostaniemy kryptogram *cde* ■

Ogólnie rzecz biorąc, jeśli alfabet V jest n literowy, to możemy V utożsamić z pierścieniem $Z_n = \{0,1,...,n-1\}$ i permutację $\pi_k : V \to V$ definiujemy wówczas wzorem

$$\pi_k(x) = (x+k)(\mathrm{mod}\, n)$$

Klasycznym szyfrem Cezara nazywamy przekształcenie szyfrujące i deszyfrujące jakie uzyskujemy dla szyfru Cezara jeśli przyjmiemy klucz *k*=3 (nie jest to więc szyfr w zwykłym rozumieniu tego słowa a właściwie pewien kod).

Funkcją definiującą podstawienia jest dla klasycznego szyfru Cezara permutacja $\pi_3 : V \to V$, czyli *a* zastępujemy przez *d* , *b* przez *e* , *c* przez *f* itd.

$$\pi_3 = \begin{pmatrix} a\,b\,c\,d\,e & \cdots & y\,z \\ d\,e\,f\,g\,h & \cdots & b\,c \end{pmatrix}$$

co pamiętając o utożsamianiu liter i liczb z pierścienia Z_{26} można zapisać jako:

$$\pi_3(x) = x + 3(\mathrm{mod}\, 26)$$

co odpowiada braniu jako wartości permutacji π_3 liczby – litery znajdującej się o 3 pozycje w prawo (modulo 26) w stosunku do liczby – litery argumentu.

Tzw. szyfr ROT13 jest to mówiąc nieprecyzyjnie szyfr Cezara z przesunięciem 13. Szyfr ten używany jest w Internecie do szyfrowania poczty elektronicznej a ściślej treści uznanych w niej za mniej przyzwoite (ROT jest skrótem od "rotation" lub "rot" zgniły, zepsuty) .

Metoda ataku na szyfr Cezara.. Szyfr Cezara nie jest bezpieczny ponieważ własności statystyczne tekstu jawnego przenoszą się na tekst zaszyfrowany. Wystarczy więc sporządzić statystykę dla liter szyfrogramu i znać statystykę występowania liter dla tekstu jawnego by porównując częstości występowania liter odtworzyć funkcję π_k co oczywiście łamie szyfr.

2.Szyfr Playfaire'a

Szyfr Playfaire'a wynalazł w roku 1854 sir Charles Wheatstone i nazwał go szyfrem Playfaire'a na cześć swego przyjaciela Lyona Playfaire'a. Warto przypomnieć, że Wheatstone jest także wynalazcą układu służącego m.in. do pomiaru oporu elektrycznego zwanego obecnie mostkiem Wheatstone'a. Szyfr Playfaire'a był stosowany przez Anglików w czasie pierwszej wojny światowej.

Szyfr Playfaire'a jest blokowym szyfrem z kluczem prywatnym. Jednostkami tekstu (blokami) w szyfrze Playfaire'a są pary liter.

Nie rozróżniamy liter *I* oraz *J*, zawsze piszemy *I*. Litery takiego uproszczonego 25 literowego alfabetu zapisujemy w dowolnej kolejności w tabelce o pięciu , wierszach i pięciu kolumnach np:

R V M H Y
F A S U Q
P Z D N K
T G L B C
E I O W X

Kluczem jest tu właśnie powyższa tabelka. kluczy jest tyle ile różnych tabelek 5x5.

Szyfrowanie. Pary liter szyfrujemy następująco: jeśli obie znajdują się w jednym wierszu, jak np. FU, to zamiast każdej bierzemy następną literę z tego samego wiersza, zamiast FU weźmiemy więc AQ. Oczywiście, zamiast ostatniej litery bierzemy pierwszą. Jeśli obie znajdują się w tej samej kolumnie, to bierzemy następne litery z tej samej kolumny: zamiast DO - LM.

Wreszcie, jeśli obie litery znajdują się w różnych wierszach i różnych kolumnach, np. FH, to zamiast pierwszej litery F bierzemy literę z tego wiersza co F i z tej kolumny co H, a więc U, a zamiast drugiej litery H bierzemy literę z tego wiersza co H i tej kolumny co F, czyli R. Parę FH szyfrujemy więc jako UR. Ten system szyfrowania nazywany jest szyfrem Playfaira.

Deszyfrowanie. W oczywisty sposób skonstruowane dla danej tabelki przekształcenie dla par liter jest odwracalne.

Metoda ataku. Korzystając z tablic częstości występowania par liter w danym języku (często również takie tablice uwzględniają tematykę tekstu jawnego) szyfr Playfaira można złamać metodami statystycznymi podobnie jak szyfr Cezara.

3. Szyfr Hilla i szyfr afiniczny

Szyfr Hilla to szyfr blokowy z kluczem prywatnym. Szyfr Hilla pochodzi z roku 1929. Niech p będzie liczbą pierwszą. Punktem wyjścia jest izomorfizm przestrzeni liniowych Z_p^r nad ciałem Z_p, $f : Z_p^r \to Z_p^r$, (gdzie $r \in N$). Izomorfizm ten jest zadany macierzą kwadratową nieosobliwą $r \times r$; $A \in Z_p^{(r,r)}$.

$$A = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1r} \\ k_{21} & k_{22} & \dots & k_{2r} \\ \dots & \dots & \dots & \dots \\ k_{r1} & k_{r2} & \dots & k_{rr} \end{bmatrix}$$

Ściślej, n-literowy alfabet V w którym zapisujemy wiadomości jawne i szyfrogramy jak zwykle utożsamiamy z liczbami 0,1,..., p-1 czyli elementami ciała Z_p.

Tekst jawny $m = m_1 m_2 ... m_t$ o długości t dzielimy jak zwykle na jednostki tekstu o długości r, ewentualnie wydłużając ostatnią jednostkę tak, by miała długość r.

Szyfrowanie.Blok (jednostkę tekstu) $m_1 m_2 ... m_r$ zastępujemy przez $c_1 c_2 ... c_r$, gdzie $c_i \in Z_p$ dla każdego i oraz

$$\begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_r \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1r} \\ k_{21} & k_{22} & \dots & k_{2r} \\ & \dots & & \\ k_{r1} & k_{r2} & \dots & k_{rr} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ . \\ . \\ . \\ m_r \end{bmatrix} = f(m_1, m_2, ..., m_r) \qquad (*)$$

Działania mnożenia i dodawania we wzorze (*) są działaniami modulo n w ciele Z_p.

Deszyfrowanie. Deszyfrowanie polega na pomnożeniu szyfrogramu przez macierz odwrotną do A.

Metoda ataku.. Metodą ataku na szyfr Hilla jest tzw. kryptoanaliza liniowa.

4. Szyfr Vigenere`a.

Szyfr Vigenere'a jest szyfrem z kluczem prywatnym. Jest to tzw. szyfr polialfabetowy Załóżmy, że m jest licznością alfabetu jawnego V tzn. $m = cardV$, np. m= 26 w przypadku języka angielskiego.

Identyfikujemy a ÷ 0
b ÷ 1
c ÷ 2 ...
z ÷ 25

czyli identyfikujemy kolejne litery z kolejnymi liczbami naturalnymi ze zbioru {0,2,...,25}

W przypadku alfabetu polskiego liter jest trochę więcej, bo mamy ą, ć, ę, ł, ń, ś , ó, ź. Klucz $k = k_1 k_2 ... k_t \in V^t$, $t \in N$ powtarzamy, jeśli trzeba, okresowo. Liczba t jest tzw. okresem klucza szyfru Vigenere'a.

Szyfr Vigenere`a zadajemy tak. Definiujemy odwzorowane $f_i : V \to V$ wzorem:

$$f_i(a) \overset{df}{=} a \oplus_m k_i$$

Szyfrowanie Dla tekstu jawnego $m = m_1 m_2 ... m_r$ szyfrogram $c = c_1 c_2 ... c_r$ obliczamy następująco

$$c = f_{1(\mathrm{mod}\, t)}(m_1) f_{2(\mathrm{mod}\, t)}(m_2) ... f_{r(\mathrm{mod}\, t)}(m_r)$$

Przykład. Okres klucza 5, klucz k = OKRES

m = ENCYKLOPEDI ATECHN IKI
k = OKRESOKRESOKRESOKRESO
c = TYUCCZGJWXLRYXRSENCX

Oczywiście szyfr Cezara jest szczególnym przypadkiem szyfru Vigenere`a.

De szyfrowanie Dla szyfrogramu $c = c_1 c_2 ... c_r$ wiadomość jawną szyfrogram obliczamy następująco:

$$m = f^{-1}_{1(\mathrm{mod}\, t)}(c_1) f^{-1}_{2(\mathrm{mod}\, t)}(c_2) ... f^{-1}_{r(\mathrm{mod}\, t)}(c_r)$$

gdzie $f^{-1}_{k_i}(a) = a \oplus_n (-k_i (\mathrm{mod}\, n))$

Metoda ataku. Najpierw ustalamy okres klucza a następnie stosujemy metodę częstotliwościową tak jak w przypadku szyfru Cezara.

5. Maszyna szyfrująca Enigma

Maszyna szyfrująca Enigma to przykład realizacji szyfru polialfabetowego. Enigma jest tzw. wirnikową maszyną szyfrującą. Enigma była wykorzystywana podczas II wojny światowej przez Wermacht.

Enigma składa się z zestawu niezależnych wirników o regulowanych położeniach kątowych. Pozycja kątowa wirników (każdy wirnik ma własną zmienną pozycję kątową) definiuje permutację (funkcję podstawienia z definicji szyfru polialfabetowego) przyporządkowującą danej literze literę szyfrogramu. Po zaszyfrowaniu danej litery zmienia się położenie układu wirników zmienia się więc permutacja.

Sposób zmiany położenia wirników i ich położenie początkowe a również same wirniki (były wymienne) stanowią klucz Enigmy. Sposób zmiany położenia wirników został dobrany tak by okres klucza szyfru polialfabetowego był bardzo duży.

6. Szyfry podstawieniowo-przestawieniowe

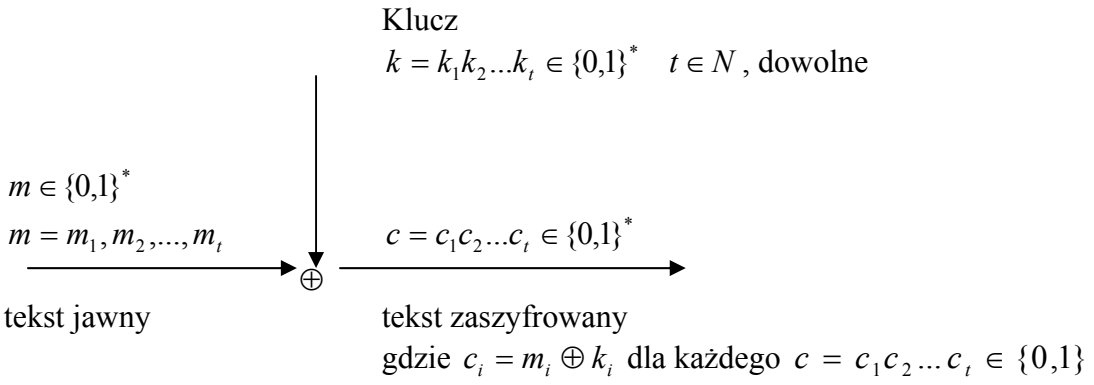
Szyfr podstawieniowo-przestawieniowy to produkt szyfru podstawieniowego i przestawieniowego. Okazuje się, że takie złożenie szyfrów jest bardzo dobrym pomysłem prowadzacym do koncepcji rundy, sieci permutacyjno-podstawieniowych i stanowiącym punkt wyjścia dla współczesnych blokowych szyfrów z kluczem prywatnym.

7. *Szyfr Vernama*.

Szyfr Vermana to inaczej szyfr idealny lub szyfr z kluczem jednokrotnym (ang one pad cipher). Szyfr Vernama jest szyfrem strumieniowym.

Idealnym, z kryptograficznego punktu widzenia, jest szyfr z jednokrotnym kluczem losowym generowanym w ciągu prób Bernoulliego. Szyfr z kluczem jednokrotnym charakteryzuje się bezpieczeństwem idealnym, ponieważ przy ustalonej długości wiadomości szyfrowanej i ustalonej wiadomości o tej długości, prawdopodobieństwo wystąpienia każdego tekstu zaszyfrowanego jest jednakowe.

Szczególnym przypadkiem szyfru jednokrotnego jest szyfr Vernama, przekształcający ciągi zerojedynkowe na ciągi zerojedynkowe. Szyfr ten został wynaleziony w 1917 roku przez dwóch Amerykanów: Gilberta S. Vernama (pracującego dla American Telephone and Telegraph Company, w skrócie AT&T) i Josepha O. Mauborgne (z US Army Signal Corps.)



Rys.1 Szyfr Vernama. Szyfrowanie: $c_i = m_i \oplus k_i$. Deszyfrowanie: $m_i = c_i \oplus k_i$

Algorytm deszyfrowania jest następujący $m_i = c_i \oplus k_i$. Jego poprawność wynika z łączności sumy modulo 2 i faktu, że $k_i \oplus k_i = 0$.

Wadą szyfru Vernama jest to, że klucz jest tak długi, jak wiadomość jawna. Jednak w pewnych zastosowaniach użycie klucza jednokrotnego jest uzasadnione (dyplomacja, wojsko).

W praktyce często zastępuje się ciąg losowy $k = k_1 k_2 ... k_t \in \{0,1\}^*$ ciągiem pseudolosowym. Np. mamy do zaszyfrowania tekst jawny $(m_i)_{i=1}^{N_0}$ o długości N_0, generujemy pseudolosowy ciąg bitów $(k_i)_{i=1}^{N_0}$ i szyfrujemy tekst jawny jako $(c_i)_{i=1}^{N_0}$, gdzie $c_i = k_i \oplus a_i$ dla każdego i=1,2,..., N_0.

Szyfr Vernama jest szyfrem strumieniowym. Można go uogólnić z przypadku binarnego na dowolny alfabet skończony zastępując sumę modulo 2 sumą modulo *n*.

Literatura

[1] A. Menezes, P. Oorschot, S. Vanstone; Handbook of Applled Cryptography; CRC Press Inc., 1997. (treść jest na stronie: http://cacr.math.uwaterloo.ca/hac)

[2] J.Stokłosa, T.Bilski,T.Pankowski; Bezpieczeństwo danych w systemach informatycznych; PWN, Warszawa 2001.

[3] N.Koblitz; Algebraiczne aspekty kryptografii; WNT, Warszawa 2000.

[4] N.Koblitz; A Course in Number Theory and Cryptography; Springer Verlag, New York 1994. (jest przekład polski p.t. Wykład z teorii liczb i kryptografii; WNT, Warszawa 1995.)

[5] M.Kutyłowski, W.Strothmann; Kryptografia, teoria i praktyka zabezpieczania systemów komputerowych; Oficyna Wydawnicza Read Me, Warszawa 2001.

[6] B.Schneier; Kryptografia dla praktyków; WNT, 2002.

[7] W.Stallings; Ochrona danych w sieci i intersieci; WNT; Warszawa 1997.

Strony www - użyteczne linki do „stron kryptograficznych"

www.certicom.ca - strona amerykańskiej firmy kryptograficznej

www.kerberos.pl - strona polskiej firmy kryptograficznej Kerberos

www.enigma.com.pl -strona polskiej firmy kryptograficznej Enigma

http://cacr.math.uwaterloo.ca/hac - strona www z podręcznikiem Menezesa w formacie pdf i ps.

www.openssl.org – strona poświęcona bibliotece open ssl

www.insecure.org/nmap - warto zajrzeć

www.shoup.net/ntb - strona www z podręcznikiem Victora Shoupa „A computational Introduction to Number Theory and Algebra", b. ładnie napisana książka do ściągnięcia

Zadania

Zadanie 1

Załóżmy, że używamy alfabetu angielskiego i umówmy się, że używamy tylko małych liter: *a,b,c,...,z* (w sumie mamy więc 26 liter). Oznaczmy ten alfabet przez *V*. W kryptografii często utożsamiamy tekst szyfrowany (czyli pewne słowo nad alfabetem *V)* z liczbą traktując słowo nad alfabetem *V* jako zapis liczb naturalnych w systemie pozycyjnym (ściślej jako zapis liczb ze zbioru $N \cup \{0\}$).

1) Jaka jest podstawa *W* tego zapisu pozycyjnego
2) Jaką maksymalnie liczbę naturalną możemy zapisać słowem o długości 100.
3) Zapisać w rozważanym zapisie liczbę *n*=10293 jeśli przyjmiemy, że a odpowiada 0, b odpowiada 1, c odpowiada 2,itd. ...,a z odpowiada 25.
4) Ile to jest: *abrakadabra* ?

Rozwiązanie

1. Podstawa *W* zapisu pozycyjnego jest równa liczbie liter alfabetu *V* a więc *W*=26.

1. Jeśli przyjmiemy, że *a* odpowiada 0, *b* odpowiada 1, *c* odpowiada 2,...,a *z* odpowiada 25 to maksymalna liczba naturalna jaką możemy zapisać słowem o długości 100 to liczba $\underbrace{zzz...zz}_{100} = 26^{100} - 1$
2. Stosując algorytm przedstawiania liczby ze zbioru $N \cup \{0\}$ w zapisie wagowym o podstawie (wadze) *W* dostajemy

$10293 : 26 = 395 \cdot 26 + 23$, $395 : 26 = 15 \cdot 26 + 5$, $15 : 26 = 0 \cdot 26 + 15$

zatem 10293 (w zapisie dziesiętnym) = *pfx* (w zapisie przy podstawie *W*=26)

4. $abrakadabra = 1 \cdot W^9 + 17 \cdot W^8 + 10 \cdot W^6 + 3 \cdot W^4 + 1 \cdot W^2 + 17 \cdot W^1 =$

$= 1 \cdot 26^9 + 17 \cdot 26^8 + 10 \cdot 26^6 + 3 \cdot 26^4 + 1 \cdot 26^2 + 17 \cdot 26^1$ ■

Zadanie 2

Załóżmy, że wiadomość jawna i szyfrogram zapisujemy w 26 literowym alfabecie angielskim złożonym z 26 liter. Wiemy, że *gsgtgrvzg* jest tekstem zaszyfrowanym za pomocą szyfru Cezara ale nie znamy klucza *k* (*k* jest liczbą pozycji o które następuje przesunięcie w prawo w alfabecie). Znaleźć klucz *k* i tekst jawny wiedząc, że tekst jawny odpowiadający powyższemu tekstowi zaszyfrowanemu jest sensownym zdaniem dotyczącym zwierząt.

Rozwiązanie

Przestrzeń kluczy dla szyfru Cezara jest bardzo mała. Mamy tylko 25 kluczy $\{1,2,3,...,25\}$. Zastosujemy tzw. atak brutalny czyli przeglądanie po kolei zbioru kluczy. Sprawdzamy czy

dla klucza k=1,2,...,25 uzyskamy wiadomość sensowną. Już dla k=6 uzyskujemy wiadomość sensowną dotyczącą zwierząt: *alamakota* . Oczywiście dla długich tekstów procedurę przeglądania przestrzeni kluczy i rozszyfrowywania możemy zautomatyzować pisząc odpowiedni program. □

Zadanie 3

W typowym kryptograficznym dzielimy tekst jawny (czyli z matematycznego punktu widzenia pewne słowo m nad ustalonym alfabetem V) na tzw. jednostki tekstu, czyli słowa o ustalonej długości, po czym przyporządkowujemy każdej jednostce tekstu liczbę i następnie dopiero przekształcamy ją funkcją szyfrującą. Do przekształcania jednostki tekstu o długości $r \geq 1$ w liczbę, można użyć dowolnej funkcji różnowartościowej $f : V^r \to N \cup \{0\}$ np.: można potraktować napis $a_{r-1}, a_{r-2}...a_0 \in V^r$ jako liczbę w zapisie wagowym o wadze $W = \text{card}\, V$ i umawiamy się, że taką funkcję zastosujemy w zadaniu;

a) przyporządkować jednostce tekstu „pies" liczbę $n \in N \cup \{0\}$ przyjmując, że tekst jawny zapisywany jest w 26 literowym alfabecie angielskim złożonym z samych małych liter, do którego dołączono spację (w sumie mamy więc 27 symboli). Przyjmujemy również następującą umowę:

spacja = 0
a =1
b =2
c =3
.
.
.
z =26

Nasz alfabet jest więc taki $V = \{spacja, a, b, c, ..., z\}$ i $\text{card}\, V = 27$

b) przyporządkować liczbie 8444 (w zapisie dziesiętnym) odpowiadające jej słowo nad alfabetem V.

c) Przyjmijmy, ze kryptosystem działa na liczbach z ciała F_q (ciało skończone o q elementach). Znaleźć największą dopuszczalną długość jednostki tekstu.

Rozwiązanie

1.Obliczamy szukaną liczbę n traktując słowo „pies" jako naturalny zapis wagowy liczby n (z wagą W = 27).

$$\text{pies} = 19 + 5 \cdot 27 + 9 \cdot 27^2 + 16 \cdot 27^3 = 321643$$

Przy dłuższych słowach warto zastosować do powyższych obliczeń schemat Homera będący sposobem obliczenia wartości wielomianu w punkcie, oparty na wzorze:

$$a_r x^r + a_{r-1} x^{r-1} + ... + a_0 = (...(((a_r x + a_{r-1})x + a_{r-2})x + + a_{r-3})...)x + a_0$$

2.Stosując zwykły algorytm zapisu liczby w zapisie naturalnym wagowym z wagą W uzyskamy 8444 (zapis dziesiętny) = kot (zapis z wagą $W = 27$). Poszukiwanym słowem jest więc słowo „kot”.

Uwaga 1. Algorytm konwersji liczby $n \in N \cup \{0\}$ ma zapis naturalny wagowy z wagą W tzn. sposób obliczenia słowa $a_{r-1}a_r...a_0 \in V^r$ polega na wykonaniu wielokrotnego dzielenia liczby n przez wagę W i braniu jako kolejnych cyfr reszt z dzielenia. Dokładniej:

Wejście: n i W
Wyjście: $a_{r-1}a_{r-2}...a_0 \in V^r$
$m_0 = n;$
for $i = 0$ to $i = r-1$ **do begin**
podziel m_1 przez W uzyskując m_{i+1} i resztę a_i, czyli przedstaw m_1 w postaci $m_1 = m_{i+1} \cdot W + a_i$, gdzie $0 \le a_i < W$
end

Jest to algorytm dla przypadku, gdy wiemy, że liczba n zmieści się na r pozycjach. Jeśli nie mamy tej informacji, to algorytm konwersji musi mieć regułą stopu. Najprościej ją zrealizować badając czy $m_1 = 0$, jeśli tak zatrzymujemy obliczenia.

Uwaga 2. Możemy też zdefiniować funkcję $f : V^r \to N \cup \{0\}$ postępując tak. Ciąg r znaków ASCII (8 bitów na znak) zestawiamy w słowo binarne $8 \cdot r$ bitowe i traktujemy to słowo jako liczbę w kodzie NKB (naturalny kod binarny). Metoda taka jest prostsza, ale z reguły ten sam tekst prowadzi do liczby o większej liczbie bitów.

3.Jeśli jednostka tekstu ma długość r, to maksymalna liczba jaką możemy zapisać w naturalnym kodzie wagowym z wagą W jest równa $W^r - 1$. Oczywiście musi być przy tym spełniona nierówność $W^r - 1 \le q$, zatem maksymalna dopuszczalna długość jednostki tekstu jest równa $\max \{r \in N; W^r - 1 \le q\}$ lub inaczej $r_{\max} = \lfloor \log_W (q-1) \rfloor$. ■

Zadanie 4

Załóżmy, że zapisujemy wiadomości jawne w W literowym alfabecie. Jaka jest maksymalna długość jednostki tekstu, jeśli jednostkę tekstu zamieniamy w przyjętym systemie kryptograficznym na:

1. liczbę z grupy multiplikatywnej F_q^* ciała F_q (tak jest np. w systemie kryptograficznym EL Gamala).
2. liczbę z pierścienia Z_n, gdzie $n = p \cdot q$ i p, q są różnymi liczbami pierwszymi (tak jest w systemie kryptograficznym RSA).

Rozwiązanie

1. Ilość różnych słów o długości k nad alfabetem W literowym V jest równa W^k (jest to ilość k elementowych wariacji z powtórzeniami ze zbioru W elementowego lub co na jedno wychodzi, liczba elementów zbioru V^k). Odwzorowanie szyfrujące $f : V^k \to F_q^*$ przyporządkowujące jednostce tekstu liczbę, musi być różnowartościowe, zatem musimy mieć $W^k \le q-1$, ponieważ liczba elementów grupy multiplikatywnej F_q^* jest równa q-1. Ostatecznie więc $k \le \log_W (q-1)$. Jeśli np. liczba q elementów ciała jest równa 2^{400}, a $W = 32$, to mamy $\log_{32} 2^{400} = \log_{32} 32^{80} = 80$.

2. W przypadku pierścienia Z_n rozumujemy analogicznie otrzymując warunek $W^k \le n$, co daje $k \le \log_W n$. ■

Zadanie 5

W systemie operacyjnym Linux (w dystrybucji Linuxa o nazwie Red Hat 6.0 Hewig) można stosować hasła użytkowników systemu o długości do 256 znaków. Ile różnych haseł można używać w tym systemie.

Rozwiązanie

Załóżmy, że hasła tworzymy jako słowa nad alfabetem V zawierającym W symboli. Istnieje dokładnie W^k różnych haseł o długości k (czyli słów) k literowych nad alfabetem V zatem liczba wszystkich możliwych haseł łącznie z hasłem pustym wynosi

$$1 + W + W^2 + W^3 + \ldots + W^{256} = \frac{W^{257} - 1}{W - 1}$$

■

Zadanie 6

Niech A będzie zbiorem $< 0, pq - 1 >$, gdzie p i q są różnymi liczbami pierwszymi takimi, że liczby p-1 i q-1 nie są podzielne przez 3. Pokazać, że funkcja zadana wzorem

$$f(x) = x^3 \,(\mathrm{mod}\, pq)$$

czyli podnoszenie do trzeciej potęgi w pierścieniu Z_n, (gdzie $n = pq$) jest permutacją.

Funkcja f jest klasycznym przykładem tzw. zapadkowej funkcji jednokierunkowej (ang. trapdoor one-way function). Pokazać, co stanowi informację zapadkową (ang. the trapdoor information) dla funkcji f.

Uwaga 1. W praktyce p i q są liczbami pierwszymi mającymi po około 100-150 cyfr dziesiętnych.

Uwaga 2. Nie ma również efektywnych algorytmów obliczenia pierwiastka trzeciego stopnia w pierścieniu Z_n, czyli inaczej algorytmów odwracania funkcji f. Dla dużych p i q odwracanie f jest praktycznie nierealizowalne.

Rozwiązanie:

1. Niech p będzie ustaloną liczbą pierwszą i $a \in Z_p$. Jeśli $NWD(a, p-1)$ z 1, czyli liczby a i p-1 są względnie pierwsze, to istnieje element odwrotny $a^{-1} \in Z_{p-1}$ i funkcja

$$h : Z_{p-1} \ni j \to a \otimes_{p-1} j \in Z_{p-1}$$

jest różnowartościowa i „na". Istotnie, wystarczy wykazać, że funkcja h jest „na", bo dziedzina i przeciwdziedzina funkcji są równolicznymi zbiorami skończonymi. Z kolei to, że h jest „na", wynika z rozwiązalności dla każdego $b \in Z_{p-1}$ równania (1) względem x.

$$a \otimes_{p-1} x = b \tag{1}$$

Mnożąc obie strony równania (1) przez $a^{-1} \in Z_{p-1}$ dostajemy:

$$x = a^{-1} \otimes_{p-1} b \tag{2}$$

Zatem dla dowolnego $b \in Z_{p-1}$ równanie (1) ma rozwiązanie (2). Biorąc jako a liczbę 3 i korzystając z założenia z treści zadania $NWD(3, p-1) = 1$ dostajemy, że przekształcenie

$$Z_{p-1} \ni j \to 3 \otimes_{p-1} j \in Z_{p-1}$$

jest różnowartościowe i „na".

Zauważmy jeszcze, że każdy element grupy multiplikatywnej Z_p^* (jest to grupa cykliczna o p-1 elementach) daje się jednoznaczne przedstawić w postaci g^j dla pewnego $j = 1,2,..., p-1$, gdzie g jest generatorem grupy Z_p^*.

2. By wykazać, że $f : Z_n \to Z_n$ jest permutacją, wystarczy tak jak w przypadku funkcji h z punktu 1 wykazać, że funkcja f jest „na", tzn. rozwiązalne jest dla każdego $b \in Z_n$ równanie (3) w pierścieniu Z_n.

$$x^3 = b \tag{3}$$

Równość (3) jest równoważna kongruencji (4):

$$x^3 \equiv b(\mathrm{mod}\, n) \tag{4}.$$

Z kolei (ponieważ $n = p \cdot q$ i liczby pierwsze p i q są różne) kongruencja (4) jest równoważna (por. zadanie xx) układowi dwu kongruencji

$$x^3 \equiv b(\mathrm{mod}\, p) \tag{5}$$
$$x^3 \equiv b(\mathrm{mod}\, q) \tag{6}$$

Wystarczy więc pokazać, że dla każdego $b' \in Z_p$ (dla liczby pierwszej q rozumujemy analogicznie) rozwiązalne jest w ciele Z_p równanie

$$x^3 \equiv b' \tag{7}$$

Jeśli $b' = 0$, to $x = 0$, zatem rozwiązanie równania (7) istnieje. Załóżmy teraz, że $b' \neq 0$ a dokładniej $b' \in Z_p^*$, wówczas oczywiście rozwiązanie x równania (7), o ile istnieje, jest $\neq 0$.

Jeśli g oznacza generator grupy multiplikatywnej Z_p^*, a rozwiązanie x równania (7) istnieje, to istnieją takie liczby $j,k \in Z_{p-1}$, że $x = g^j$ i $b' = g^k$.

Równanie (7) w ciele Z_p można zapisać teraz tak

$$g^{3j} = g^k \tag{8}$$

i jest to równanie względem j.

Biorąc pod uwagę fakt, że dla każdego $x \in Z_p^*$ mamy w Z_p $x^{p-1} = 1$ (wniosek z małego twierdzenia Fermata lub ogólniejszego faktu z teorii grup, ze rząd elementu jest dzielnikiem rzędu grupy , a grupa Z_p^* ma p-1 elementów) równanie (8) można zapisać jako:

$$g^{3 \otimes_{p-1} j} = g^k \tag{9}$$

co jest równoważne równości wykładników potęg w równaniu (9) modulo p-1, czyli równoważne równaniu (10) w pierścieniu Z_{p-1}

$$3 \otimes_{p-1} j = k \tag{10}$$

skąd

$$j = k \otimes_{p-1} 3^{-1} \tag{11}$$

gdzie 3^{-1} jest odwrotnością w Z_{p-1}. Zatem poszukiwane rozwiązanie x równania (7) jest równe:

$$x = g^j = g^{k \otimes_{p-1} 3^{-1}}$$

co dowodzi rozwiązalności równania (3) w pierścieniu Z_n, czyli równania $x^3 = b$ i ostatecznie tego, że funkcja $f : Z_n \to Z_n$, przy przyjętych założeniach jest różnowartościowa i „na". Funkcja f jest więc permutacją zbioru Z_n, czego mieliśmy dowieść.

3. Zauważmy, że znajomości rozkładu liczby $n = p \cdot q$ na czynniki pierwsze, czyli znajomość liczby p i q umożliwia efektywne rozwiązywanie równania (3). Pewnym problemem może wydawać się znalezienie generatora g grupy multiplikatywnej Z_p^*. Istnieją jednak efektywne algorytmy znajdujące generator grupy Z_p^*.

Zatem znajomość rozkładu liczby n na czynniki pierwsze, czyli znajomość p i q stanowi informację zapadkową (trapdoor information).

Uwaga 1. Analogicznie jak w powyższym zadaniu rozumowanie można przeprowadzić dla każdej funkcji $f : Z_n \to Z_n$, zadanej wzorem

$$f : Z_n \ni x \to x^m \in Z_n$$

przy założeniach, że $n = p_1 \cdot p_2 \cdot \ldots \cdot p_r$, gdzie p_i dla $i = 1,2,...,r$ są parami różnymi liczbami pierwszymi, oraz dla każdego $i = 1,2,...,r$ mamy:

$$NWD(m, p_i - 1) = 1.$$

W tej ogólniejszej sytuacji informacją zapadkową będzie rozkład liczby $n = p_1 \cdot p_2 \cdot \ldots \cdot p_r$ na czynniki pierwsze, czyli znajomość liczb pierwszych $p_1, p_2, ..., p_r$.

Uwaga 2. Zastosowana technika rozwiązywania równania (3) polega w gruncie rzeczy na sprowadzeniu tego równania do dwóch równań w pierścieniu Z_p i Z_q.

$$x_1^3 = [b]_p$$
$$x_2^3 = [b]_q$$

rozwiązaniu tych równań, uzyskaniu $x_1 \in Z_p$ i $x_2 \in Z_q$, a następnie odtworzeniu $x \in Z_n$ z wartości $x_1 \in Z_p$ i $x_2 \in Z_q$. Odtworzenie sprowadza się do skorzystania z tezy chińskiego twierdzenia o resztach. ■

Zadanie 7
Wiadomo, że Alicja i Bob posługują się szyfrem Vernama. Przechwycono wiadomość jawną $m = m_1 m_2 ... m_r$, gdzie $m_i \in \{0,1\}$ i szyfrogram $c = c_1 c_2 ... c_r$, gdzie $c_i \in \{0,1\}$, znaleźć klucz $k = k_1 k_2 ... k_r$; $k_i \in \{0,1\}$ jakim posłużyli się Alicja i Bob.

Rozwiązanie
1. Szyfrogram $c = c_1 c_2 ... c_r$ tworzony jest tak, że dla każdego $i = 1,2,...r$ przyjmujemy

$$c_i = m_i \oplus k_i \qquad (1)$$

2. Mnożąc obie strony równości (1) przez m_i i korzystając z łączności działania sumy modulo 2 w Z_2, dostajemy

$$m_i \oplus c_i = m_i \oplus (m_i \oplus k_i)$$
$$m_i \oplus c_i = (m_i \oplus m_i) \oplus k_i$$
$$m_i \oplus c_i = 0 \oplus k_i$$
$$m_i \oplus c_i = k_i$$

Uwaga. Warto zauważyć, że suma modulo 2 w Z_2 definiowana dla $a, b \in Z_2$ jako reszta z dzielenia $a+b$ przez 2, czyli $a \oplus b \overset{df}{=} (a+b) \pmod 2$, jest tym samym działaniem co suma modulo 2 zdefiniowana równościami

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

■

Zadanie 8
Ile różnych działań 2 argumentowych można określić w ustalonym zbiorze A
a) zawierającym tylko 1 element
b) zawierającym n -elementów
c) jeśli A jest nieskończony

Rozwiązanie
1. Pytanie postawione w zadaniu jest pytaniem o liczbę różnych funkcji postaci $f : A \times A \to A$. W przypadku zbioru 1 elementowego iloczyn kartezjański $A \times A$ zawiera tylko 1 element, zatem możemy zdefiniować tylko jedno działanie.

2. W przypadku zbioru n –elementowego, iloczyn kartezjański $A \times A$ zawiera n^2 elementów. W tym przypadku liczba wszystkich różnych funkcji postaci $f : A \times A \to A$ jest równa liczbie n^2 elementowych wariacji z powtórzeniami ze zbioru n –elementowego, a więc $(n^2)^n = n^{2n}$. Na przykład, jeśli n=2 (tak jest w dwuelementowej algebrze Boole`a), to mamy $2^4 = 16$ różnych działań. W przypadku skończonego zbioru A, wygodnie jest działania definiować za pomocą tabelki.

3. Jeśli A jest zbiorem nieskończonym, to mamy nieskończenie wiele działań. Jeśli $\mathfrak{a}$ jest liczbą kardynalną zbioru A, co zapisujemy jako $\mathfrak{a} = cardA$, to liczba kardynalna zbioru wszystkich działań jest równa $\mathfrak{a}^{\mathfrak{a}}$, (bo dla zbiorów nieskończonych, zbiór $A \times A$ ma taką samą moc jak A). ■

Zadanie 9
Zaszyfrować klasycznym szyfrem Cezara wiadomość jawną

m= this cipher is certainly not secure

zapisaną w 26 literowym alfabecie angielskim złożonym z małych liter.

Rozwiązanie
Szyfr Cezara jest szyfrem podstawieniowym. Zgodnie z definicją klasycznego szyfru Cezara każdą literę wiadomości jawnej m zastępujemy literą położoną o 3 pozycje w prawo (modulo 26) przy zwykłym uporządkowaniu 26 literowego alfabetu (por. wstęp teoretyczny); zatem kryptogram c wiadomości m będzie następujący:

c=wklvf lskhu lvfhu wdlqo bqrwv hfxuh

■