2.2 Szyfry i kryptografia

1. Główne cele kryptografii

Ponieważ szyfr to w pewnym uproszczeniu rodzina kodów indeksowana kluczem, więc w naturalny sposób szyfry wpisują się w zagadnienia związane z kodami i kodowaniem.

Kryptologia bazuje na czterech działach matematyki: teorii liczb, algebrze, teorii algorytmów komputerowych i teorii prawdopodobieństwa. Kryptologia dzieli się na kryptografię (sztukę szyfrowania i tworzenia wyrafinowanych zabezpieczeń systemów komputerowych) oraz kryptoanalizę (sztukę łamania szyfrów i zabezpieczeń). Obecnie często nazywa się kryptologię niezbyt precyzyjnie kryptografią i takie uproszczoną nieco terminologię będziemy stosować w dalszym ciągu rozdziału.

Kryptografia to zespół metod matematycznych i technik tkwiący częściowo w szeroko pojętej elektronice a częściowo w informatyce teoretycznej.

Kryptografia oraz nieco obszerniejsza dziedzina jaką jest ochrona informacji w systemach i sieciach komputerowych odgrywają bardzo ważną rolę nie tylko we współczesnej informatyce, ale również w organizacji życia gospodarczego i administracji państwowej. Handel elektroniczny, bankowość elektroniczna, elektroniczna wymiana dokumentów (słynne EDI – Electronic Documents Interchange) czy ogólnie rzecz biorąc bezpieczne magazynowanie i przesyłanie informacji nie byłyby możliwe bez współczesnych metod kryptograficznych. Takie skróty oznaczające bezpieczne protokoły przesyłania danych jak SSL (Secure Socket Layer), SSH (Secure Shell) czy HTTPS (Secure HTTP czyli Secure HyperText Protocol) znane są każdemu użytkownikowi przeglądarki internetowej.

Główne cele kryptografii to

- zapewnienie poufności wiadomości nazywane też utajnianiem wiadomości (ang. privacy
- lub confidentiality)
- uwierzytelnianie strony czyli identyfikacja strony pragnącej uzyskać dostęp do zasobów systemu komputerowego lub sieci komputerowej (ang. entity authentication lub identification)
- uwierzytelnianie wiadomości lub uwierzytelnianie dokumentu (ang. data origin authentication)
- zapewnienie integralności danych (ang. data integrity)
- niezaprzeczalność (ang. non repudiation)

2. System kryptograficzny

Teraz powiemy co to jest wiadomość jawna, kryptogram (albo szyfrogram) oraz szyfr.

System kryptograficzny lub ***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.

- 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). Często alfabety V_1 i V_2 są tym samym afabetem.
- K jest dowolnym zbiorem jest to tzw. przestrzeń kluczy (ang. *key space*), każdy element $k \in K$ nazywamy kluczem (lub kluczem szyfrującym). Na ogół K jest zbiorem skończonym ale w definicji ogólnej systemu kryptograficznego nie robimy tego założenia.
- E jest funkcją $E : M \times K \to C$, obliczanie wartości funkcji E nazywamy szyfrowaniem, a samą funkcję przekształceniem szyfrującym lub co jest nieco mylące również szyfrem. Wartość $E(k,m)$ nazywamy wiadomością zaszyfrowaną, ***szyfrogramem***, ***kryptogramem*** lub tekstem zaszyfrowanym. Sposób obliczania $E(k,m)$ dla danych k,m nazywamy *algorytmem szyfrowania.*
- D jest funkcją $D : C \times K \to M$ obliczanie wartości funkcji D nazywamy *deszyfrowaniem* a samą funkcję D przekształceniem deszyfrującym. Sposób obliczania $D(k,m)$ dla danych k, m nazywamy *algorytmem deszyfrowania.*

Od systemu kryptograficznego wymagamy by dla każdego *klucza szyfrującego* $k_1 \in K$, istniał *klucz deszyfrujący* $k_2 \in K$ taki, że dla dowolnej wiadomości jawnej $m \in M$ zachodzi

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

Istota rzeczy: Każdą wiadomość zaszyfrowaną musimy umieć rozszyfrować, ale być może służy do tego już inny klucz.

Z powyższej definicji wynika, że dla każdego $k_1 \in K$, odwzorowanie $E(\cdot, k_1) : M \to C$ jest funkcją różnowartościową. Oczywiście podobnie odwzorowanie $D(\cdot, k_2) : E(M, k_1) \to C$, (gdzie $k_2 \in K$ jest kluczem deszyfrującym odpowiadającym kluczowi $k_1 \in K$) jest funkcją różnowarościową.

Z powyższej definicji wynika również, że mówiąc niezbyt precyzyjnie, szyfr to rodzina kodów sparametryzowana kluczem $k \in K$.

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)

Uwaga: 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 wówczas 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. Istnieją jednak w wielu krajach (np. w USA, Francji czy Iraku) pewne ograniczenia o charakterze prawnym np. 128 bitowe klucze to standard w USA i w Polsce do zakupów sieciowych z kartą kredytową i w transakcjach bankowych.

W powszechnym mniemaniu klucze 128 bitowe uważa się za bezpieczne, ale rzecz jasna taka zasada jest niebezpieczną 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.

Ważna uwaga praktyczna o tajności metod i algorytmów kryptograficznych:

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

3. Podział szyfrów

Systemy kryptograficzne, szyfry, algorytmy kryptograficzne dzielimy na

- symetryczne (inaczej z kluczem symetrycznym)
- 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 $k_1 \in K$ takich że z $k_1 \in K$ nie daje się łatwo obliczyć klucz deszyfrujący $k_2 \in K$ i odwrotnie z klucza deszyfrującego $k_2 \in K$ nie daje się łatwo obliczyć klucz szyfrujący $k_1 \in K$) to system kryptograficzny, szyfr czy algorytm kryptograficzny nazywamy systemem, szyfrem czy algorytmem kryptograficznym z kluczem asymetrycznym.

Systemy kryptograficzne z kluczem symetrycznym nazywamy też systemami z kluczem prywatnym. Przykładami takich systemów są DES (Digital Encryption Standard), 3DES (tzw. potrójny DES), DESX (DES z wybielaniem), IDEA, LOKI, Twofish, Blowfish, RC5, AES (Advanced Encryption Standard) czyli Rijndael .

Systemy kryptograficzne z kluczem asymetrycznym nazywamy też systemami z kluczem publicznym. Przykładami takich systemów są szyfr RSA, szyfr Rabina, szyfr ElGamala, szyfr plecakowy Merklego-Hellmanna, szyfr plecakowy Chora-Rivesta, szyfr McEliece'a i szyfr probabilistyczny Bluma-Goldwasser'a.
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).

4. 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*** $f : V_1^n \times K \to V_2^m$. Blokowa funkcja szyfrująca $f : V_1^n \times K \to V_2^m$ musi spełniać warunek: dla każdego $k \in K$ funkcja $f(\cdot, k) : V_1^n \to V_2^m$ jest różnowartościowa.

Z kolei funkcję deszyfrującą $D : C \times K \to M$ wygodnie jest zdefiniować za pomocą ***blokowej funkcji deszyfrującej*** $g : V_2^m \times K \to V_1^n$

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.*

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". Rozwiązanie jest oczywiste. Dopełniamy tekst szyfrowany do wielokrotności długości bloku. Najczęściej jest to zgodne z pewnymi standardami.

5. Funkcja jednokierunkowa

Funkcja $f : X \to Y$, gdzie *X, Y* są dowolnymi niepustymi zbiorami, jest nazywana ***funkcją jednokierunkową*** (ang. one way function) jeśli:

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

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.

- 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^*.
- 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 $x \in Z_n$.
- 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
$$f: Q_n \ni x \to f(x) = x^2 \pmod n \in Q_n$$
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.

6. Szyfry przestawieniowe czyli szyfry permutacyjne

Szyfry permutacyjne (ang. transposition ciphers) należą do klasy szyfrów symetrycznych. Załóżmy, że alfabety V_1 i V_2 są takie, że $V_1 = V_2 \overset{ozn.}{=} V$, a zbiór kluczy

K={zbiór wszystkich permutacji zbioru <1,r>}

gdzie r jest ustaloną liczbą naturalną.

Niech $m = m_1 m_2 ... m_r \in M$ 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 r

$$f: V^r \times K \to V^r$$

a dokładniej

$$f: V^r \times K \ni (m,\pi) \to f(m,\pi) = m_{\pi(1)} m_{\pi(2)} ... m_{\pi(r)} = c \in V^r$$

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

Do zapamiętania:	Kluczem jest permutacja zbioru <1,r>, gdzie r jest długością jednostki tekstu (bloku).

7. Szyfry podstawieniowe

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,. który będzie omówiony dokładniej w dalszym ciągu podrozdziału.

Polialfabetyczny 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.

Przykład: Rozważmy dla ustalenia uwagi 26 literowy alfabet języka angielskiego $V = \{a, b, c, ..., z\}$. Klasycznym ***szyfrem Cezara*** nazywamy szyfr podstawieniowy, w którym $V_1 = V_2 = V$ i funkcją definiującą podstawienia jest permutacja $\pi: V \to V$, gdzie

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

Jeśli utożsamimy litery z liczbami z pierścienia Z_{26} na zasadzie $a \sim 0$, $b \sim 1$, $c \sim 2$, d~3, ..., y~24, $z \sim 25$, to widać, że:

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

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

Ogólnie rzecz biorąc, jeśli alfabet V jest n literowy, to możemy V utożsamić z Z_n i zdefiniować permutację $\pi: V \to V$ wzorem:

$$\pi(x) = (x + r) (\mathrm{mod}\, n)$$

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

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

Tak zdefiniowany szyfr, nazywamy szyfrem Cezara z przesunięciem *r*. Oczywiście *r* jest kluczem szyfrującym i deszyfrującym jednocześnie. Kluczem w szyfrze Cezara jest więc liczba naturalna ze zbioru Z_n

Przykład: Załóżmy, że *m* jest licznością alfabetu jawnego V tzn. $m = cardV$, np. *m*= 26 w przypadku języka angielskiego.

Identyfikujemy a ÷ 1 czyli identyfikujemy kolejne litery z kolejnymi liczbami
b ÷ 2 naturalnymi ze zbioru {1,2,...,26}
...
z ÷ 26

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.

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

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

dla tekstu jawnego $m = m_1 m_2 ... m_k$

$$c = f_{[1]_t}(m_1) f_{[2]_t}(m_2) ... f_{[k]_t}(m_k)$$

Przykład: Okres klucza 5, klucz *k* = OKRES

```
m =   ENCYKLOPEDIA TECHNIKI
k =   OKRESOKRESOKRESOKRESO
------------------------------
c =   TYUCC ZGJWXLRYXRSENCX
```

Oczywiście szyfr Cezara jest szczególnym przypadkiem szyfru Vigenere`a.

8. 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.

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 (ang. 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 ale nie jest to regułą.

9. Szyfry strumieniowe

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

- 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 $(V_1, V_2, \tilde{K}, \tilde{E}, \tilde{D})$ następująco. V_1 i V_2 są wprowadzonymi już alfabetami, $\tilde{K} = \{(k_i)_{i=1}^{\infty}; k_i \in K\}$ zbiorem kluczy, $\tilde{E} : \tilde{K} \times V_1^* \to V_2^*$ funkcją szyfrującą, dokładniej $\tilde{E}$ zadane jest wzorem:

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

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

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

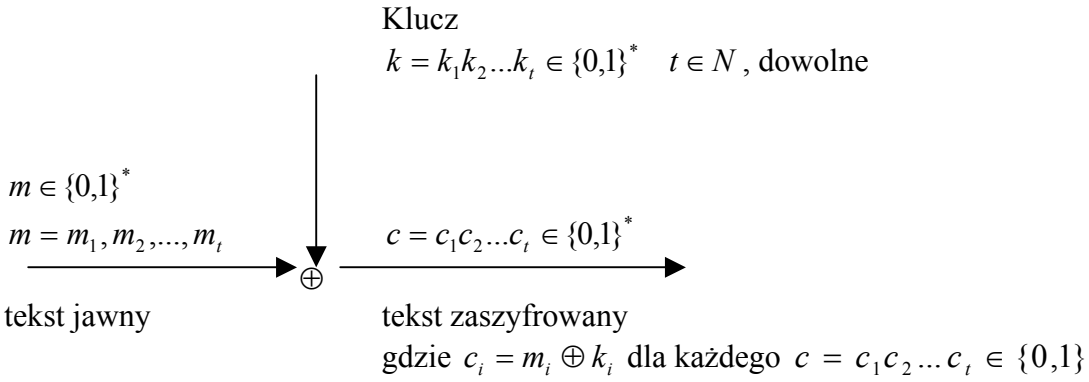
Taki system kryptograficzny nazywamy szyfrem strumieniowym.

10. Szyfry idealne

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

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 przechwyceniu szyfrogramu c o określonej długości n prawdopodobieństwo warunkowe wystąpienia danego tekstu jawnego m (długości n) pod warunkiem odebrania c nie zależy od m.

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.

11. Algorytm szyfrowania szyfrem RSA

Szyfr RSA jest typowym, najczęściej stosowanym szyfrem z kluczem publicznym. Nazwa RSA pochodzi od nazwisk 3 profesorów z MIT (Massachussets Institute of Technology)

R.L.Rivesta, A.Shamira i L.M.Adlemana, którzy w roku 1978 opracowali koncepcję tego szyfru. Szyfr RSA uważany jest za jeden z najbezpieczniejszych, najpewniejszych szyfrów. Używany jest do szyfrowania wiadomości, do tworzenia podpisów cyfrowych oraz dystrybucji kluczy kryptograficznych.

Opiszemy teraz, jak tworzymy system kryptograficzny RSA. Potrzebne nam będzie przy omawianiu szyfru RSA pojęcie funkcji Eulera. Dla danej liczby naturalnej n symbolem $\varphi(n)$ oznaczamy liczbę liczb naturalnych względnie pierwszych z n i mniejszych od n, tzn. $\varphi(n) = card\{m \in N; m \le n, NWD(m,n) = 1\}$. W ten sposób definiujemy funkcję $\varphi : N \to N$, która nazywa się funkcją Eulera. Łatwo można wykazać, że jeśli p jest liczbą pierwszą to $\varphi(p) = p - 1$ i dla każdego $k \in N$, mamy

$$\varphi(p^k) = p^{k-1}(p-1) \tag{1}$$

Jeśli m i n są względnie pierwszymi liczbami naturalnymi to mamy

$$\varphi(mn) = \varphi(m)\varphi(n) \tag{2}$$

Z (1) i (2) dostajemy, że jeśli $n = p_1^{k_1} p_2^{k_2} \cdot \ldots \cdot p_r^{k_r}$, gdzie $p_1, p_2, \ldots, p_r$ są parami różnymi liczbami pierwszymi to

$$\varphi(n) = p_1^{k_1-1}(p_1-1) \cdot p_2^{k_2-1}(p_2-1) \cdot \ldots \cdot p_r^{k_r-1}(p_r-1) \tag{3}$$

Wybieramy teraz dwie duże liczby pierwsze p i q obliczamy iloczyn $n=pq$ oraz wartość funkcji Eulera $\varphi(n) = (p-1)(q-1)$. Wybieramy następnie losowo dowolną taką liczbę $d \in Z_{\varphi(n)}$, dla której istnieje element odwrotny $e = d^{-1}$ w pierścieniu $Z_{\varphi(n)}$ czyli takie $e \in Z_{\varphi(n)}$ by $d \otimes_{\varphi(n)} e = 1$ w pierścieniu $Z_{\varphi(n)}$ (oczywiście jest to równoważne temu by znaleźć takie $e \in Z_{\varphi(n)}$, żeby $d \cdot e \equiv 1 \pmod{\varphi(n)}$). Warunkiem koniecznym i dostatecznym na to by istniał taki element odwrotny jest by $NWD(\varphi(n), d) = 1$. Obliczamy $e = d^{-1}$ i ujawniamy parę uporządkowaną (e,n) jako klucz publiczny. Kluczem prywatnym będzie para uporządkowana (d,n).

Tekst szyfrowany, czyli tekst jawny $m \in V_1^*$ np. "ala ma kota" zamieniamy na liczbę całkowitą x np. 10982398.

Podnosimy tę liczbę do ustalonej potęgi e w pierścieniu Z_n ((e,n) jest znane i stanowi klucz publiczny) lub co na jedno wychodzi podnosimy tę liczbę do potęgi e w pierścieniu Z i bierzemy resztę z dzielenia tej liczby przez liczbę n. Tę resztę wysyłamy do adresata.

$$c = x^e$$

Adresat podnosi (w pierścieniu Z_n) uzyskaną liczbę do pewnej potęgi d będącej kluczem tajnym czyli kluczem prywatnym uzyskując wiadomość jawną x.

$$x = c^d$$

Uwaga: Algorytmy RSA mające klucz o długości do 512 bitów, są stosunkowo proste do złamania. Adi Shamir opracował w 1999 r. specjalizowany równoległy komputer łamiący 512 bitowe RSA w ciągu 2 dni. Obecnie stosuje się już klucze 1024 bitowe a nawet 2048

i 4096 bitowe praktycznie całkowicie bezpieczne (w 2003 r.) tzn. nie do złamania przy współczesnym stanie wiedzy o algorytmach komputerowych i przy współczesnych możliwościach obliczeniowych systemów cyfrowych.

Uwaga: Rivest, Shamir i Adleman w słynnej słynnej zagadce zamieszczonej w 1977 roku w czasopiśmie "Scientific American" zamieścili szyfrogram pewnego tekstu jawnego, proponując czytelnikom odtworzenie tego tekstu z szyfrogramu. Użyli przy tym szyfru nazwanego później RSA-129. Liczba 129 w oznaczeniu bierze się od 129 cyfrowej liczby (cyfry dziesiętne), którą trzeba rozłożyć na czynniki pierwsze by uzyskać klucz i funkcję odwrotną deszyfrującą szyfrogram. Zadanie przed którym stanęli czytelnicy było więc typowym problemem przed którym staje kryptoanalityk. Szyfr RSA-129 został w końcu złamany w 1994 roku dzięki równoległej pracy wielu komputerów w sieci komputerowej i superkomputerów równoległych.

Uwaga*:* Fakt praktyczny: żeby złamać szyfr RSA musimy umieć rozkładać duże liczby na czynniki pierwsze. Do określenia funkcji odwrotnej do funkcji szyfrującej czyli funkcji deszyfrującej potrzebna jest znajomość 2 liczb pierwszych, p i q, których iloczyn daje n.

Wykażemy teraz poprawność RSA. Skorzystamy z następującego lematu będącego konsekwencją chińskiego twierdzenia o resztach.

Lemat*:* Niech $m_1, m_2, ..., m_n \in N$ będą parami względnie pierwsze (tzn. $NWD(m_i, m_j) = 1$ dla każdego $i, j \in <1, n>, i \neq j$) i $m_i \geq 2$ dla każdego i=1,2,...,n oraz niech a, b będą dowolnymi liczbami całkowitymi czyli $a, b \in Z$.

$a \equiv b (\mathrm{mod}\, m_i)$ dla każdego i=1,2,...,n wtedy i tylko wtedy gdy $a \equiv b (\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$

Dowód poprawności algorytmu RSA: Z założenia $ed \equiv 1 \pmod{\varphi(n)}$, zatem istnieje takie $k \in Z$, że

$$ed = 1 + k \cdot \varphi(n) \qquad (*)$$

Załóżmy teraz, że wiadomość jawna $m \in Z_n$ jest względnie pierwsza z p, czyli $NWD(m, p) = 1$. Z małego twierdzenia Fermata mamy wówczas:

$$m^{p-1} \equiv 1 \pmod{p} \qquad (**)$$

Podnosząc obie strony kongruencji (**) do potęgi $k \cdot (q-1)$ i mnożąc przez m dostajemy:

$$m^{1+k(p-1)(q-1)} \equiv m (\mathrm{mod}\, p)$$

a więc korzystając z (*) mamy:

$$m^{ed} \equiv m (\mathrm{mod}\, p).$$

Jeżeli $NWD(m, p) > 1$, to $NWD(m, p) = p$. Zatem również mamy wówczas:

$$m^{ed} \equiv m (\mathrm{mod}\, p) \qquad (***)$$

Istotnie, ponieważ m jest podzielna przez p, więc $m^{ed} \equiv 0(\mathrm{mod}\, p)$ i $m \equiv 0(\mathrm{mod}\, p)$, i kongruencja (***) zachodzi.

Zatem niezależnie od tego czy $NWD(m, p) = 1$, czy $NWD(m, p) > 1$, zawsze mamy

$$m^{ed} \equiv m(\mathrm{mod}\, p).$$

Podobnie możemy rozumować dla liczby pierwszej q otrzymując

$$m^{ed} \equiv m(\mathrm{mod}\, q)$$

Z lematu dostajemy ostatecznie, że $m^{ed} \equiv m(\mathrm{mod}\, pq)$. ■

Probabilistyczne uzasadnienie algorytmu RSA: Warto zwrócić uwagę na to, że nie wymagamy w założeniach powyższego twierdzenia by $NWD(a, n) = 1$, tak jak w założeniach twierdzenia Eulera. Próba uzasadniania kongruencji $m^{ed} \equiv m(\mathrm{mod}\, pq)$ twierdzeniem Eulera byłaby więc zręcznym korzystaniem z nieuwagi Czytelnika.

Jednak przy $n = p_1 \cdot p_2$ (gdzie czynniki są liczbami pierwszymi) i założeniu równomiernego rozkładu prawdopodobieństwa na zbiorze Z_n prawdopodobieństwo tego, że $NWD(a, n) \neq 1$ maleje do 0 wraz ze wzrostem p_1 i p_2 . Istotnie z definicji funkcji Eulera mamy, że:

$$P(\{a \in Z_n; NWD(a, n) \neq 1\}) = \frac{n - \varphi(n)}{n} = \frac{n - (p_1 - 1) \cdot (p_2 - 1)}{n} = \frac{p_1 + p_2 - 1}{p_1 \cdot p_2} = \frac{1}{p_1} + \frac{1}{p_2} - \frac{1}{n}$$

a więc $P(\{a \in Z_n; NWD(a, n) \neq 1\}) \to 0$ jeśli $p_1 \to +\infty$ i $p_2 \to +\infty$

zatem sytuacja, że $NWD(a, n) = 1$ nie zachodzi staje się bardzo mało prawdopodobna.

Uwagi końcowe o algorytmie RSA: Szyfr RSA jako algorytm z kluczem publicznym może być wykorzystywany zarówno do zwykłego szyfrowania jak również do dystrybucji tzw. kluczy sesyjnych oraz realizacji algorytmów podpisów cyfrowych.

Bezpieczeństwo szyfru RSA jest zależne od tego czy istnieje efektywny algorytm rozkładu dużej liczby n na czynniki pierwsze (dotychczas nie jest znany taki efektywny algorytm). Dokładniej, gdyby taki algorytm istniał to moglibyśmy łatwo obliczyć wartość funkcji Eulera $\varphi(n)$, n jest ogólnie znanym parametrem sytemu kryptograficznego. Odtworzenie klucza prywatnego z klucza publicznego sprowadzałoby się w tej sytuacji do znalezienia elementu odwrotnego do klucza publicznego w pierścieniu $Z_{\varphi(n)}$ co sprowadza się z kolei do realizacji rozszerzonego algorytmu Euklidesa (istnieją szybkie efektywne wersje tego algorytmu np. algorytm Stein'a).

Główną wadą algorytmu RSA jest jego znacznie większa złożoność obliczeniowa w porównaniu z innymi algorytmami szyfrowania jak np. algorytmem DES . Okazuje się to specjalnie kłopotliwe, gdy zależy nam na szyfrowaniu w czasie rzeczywistym, np. gdy szyfrujemy próbki sygnału mowy w systemie telekomunikacyjnym.

12. Funkcje skrótu

Funkcje skrótu (ang. hash function) nazywa się również jednokierunkowymi funkcjami skrótu, funkcja skrótu $f : M \to \{0,1\}^n$ (gdzie M jest zbiorem wiadomości a n jest ustalone) powinna być bowiem jednokierunkowa tzn. jeśli $f(x) = y$ i znamy y, to praktycznie powinno być niemożliwe obliczenie x.

Przykład: Funkcja skrótu SHA-1 (Secure Hash Algorithm 1).

Przykład: Funkcja skrótu MD4 (Message Digest 4).

Przykład: Funkcja skrótu MD5 (Message Digest 5).

Przykład: Funkcja skrótu RIPE MD -128

Przykład: Funkcja skrótu RIPE MD -160

Plik, tekst, wiadomość m o dowolnej długości „obłożona" funkcja skrótu daje ciąg bitów o ustalonej długości n (np.: 20 bajtów, czyli 160 bitów).

Jeśli znamy wartość funkcji skrótu $f(m)$ i samą funkcję skrótu i dysponujemy parą $(m', f(m))$, to możemy sprawdzić integralność danych, czyli sprawdzić czy $m = m'$ obliczając dla m' wartość $f(m')$. Jeśli $f(m) = f(m')$, to przyjmujemy, że $m = m'$ (choć tak być nie musi), czyli stwierdzamy, że „integralność danych jest zachowana".

Uwaga: Podpis cyfrowy wiadomości jawnej m jest tworzony z reguły dla funkcji skrótu $f(m)$.

Uwaga: Intuicja funkcji skrótu jest prosta. Dla danej wiadomości jawnej m, wartość $f(m)$ to „streszczenie" tej wiadomości zawarte w słowie o stałej liczbie bitów. Typowe n to 20 bajtów, tzn. 160 bitów.

13. Uwagi końcowe

Realizacja systemów kryptograficznych: Systemy kryptograficzne można realizować programowo, sprzętowo lub w sposób mieszany czyli częściowo sprzętowo częściowo programowo. Na ogół uważa się, że im więcej specjalizowanego sprzętu , tym system jest bezpieczniejszy. Współczesne profesjonalne systemy kryptograficzne są z reguły realizowane sprzętowo, wewnątrz specjalizowanych układów szyfrujących. Są to na ogół układy scalone 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.

Zalety rozwiązań sprzętowych → Szybkość
Zalety rozwiązań sprzętowych → Odporność na atak

Realizując systemy kryptograficzne trzeba zwracać uwagę na zastrzeżenia eksportowe i patenty np. szyfr z kluczem publicznym RSA jest opatentowany w obu wersjach podstawowej i uogólnionej.

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 (alternatywnie np. możemy ustalonej jednostce tekstu przyporządkowywać różne szyfrogramy) albo jest zawarta w samym szyfrogramie, wbudowana w szyfrogram (tak jest np. w systemie kryptograficznym ElGamala).

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

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