SKRYPT
WSTĘP DO KRYPTOLOGII

Spis treści:

1. Systemy i protokoły kryptograficzne

1.1

System kryptograficzny jest tak zwaną piątka spełniającą następujące warunki:

1. **P** - przestrzeń wiadomości jawnych (Plane Text)
2. **C -** przestrzeń wiadomości zaszyfrowanych
3. **K -** przestrzeń kluczy K={(k , k')}
4. **E -** algorytm szyfrujący
5. **D -** algorytm deszyfrujący

1.2

Funkcja szyfrująca jest **E** o następującym wzorze: $P \times K = C$
Z kolei funkcją deszyfrującą jest **D** o wzorze: $C \times K = P$

$E(k,D(k',c)) = c$

Jeżeli na tekst zaszyfrowany c kluczem k' zadziałamy algorytmem E przy pomocy klucza k, w wyniku otrzymamy tekst c.

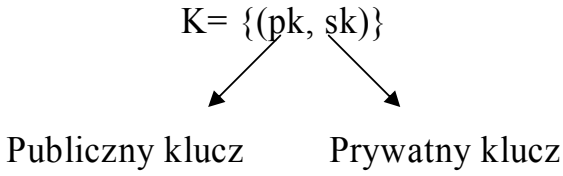
$D(k',E(k, p)) = p$

Jeżeli na tekst p zaszyfrowany algorytmem E przy pomocy klucza k zadziałamy algorytmem D z kluczem k' ,w wyniku otrzymamy tekst p.

Wniosek !
Szyfrowanie i deszyfrowanie są wzajemnie odwrotnymi procesami.

1.3

K- przestrzeń kluczy. Wyrażona wzorem K = {(k, K)} k – szyfruje, K - deszyfruje



Kryptografia z kluczem publicznym stosowana jest w systemach niesymetrycznych.
Kryptografia z kluczem prywatnym stosowana jest w systemach symetrycznych.
Jeden klucz można obliczyć z drugiego i na odwrót.

1.4

System kryptograficzny symetryczny- to taki gdzie ten sam klucz służy do szyfrowania i deszyfrowania lub są dwa klucze – jeden do szyfrowania, drugi do deszyfrowania ale pod warunkiem, że jeden da się wyliczyć z drugiego i na odwrót.

System kryptograficzny niesymetryczny – to taki, który nie jest symetryczny - przestrzeń kluczy jest dwuparametrowa.

1.5

Kryptologia jest to nauka obejmująca kryptografię i kryptoanalizę. Jest sumą tych dwóch dziedzin.

Kryptografia to dziedzina zajmująca się budowaniem systemów.

Kryptoanaliza - zajmuje się metodami łamania szyfrogramu lub systemu. (wskazuje na jego słabości)

Wymagania stawiane przed systemem kryptograficznym:

- poufność komunikacji
- identyfikacja podmiotu
- integralność danych
- niezaprzeczalność (niewypieralność)

Uwaga:

Identyfikacja to nie to samo co niezaprzeczalność !

Modele systemów buduje się w oparciu o następujące pojęcia pierwotne:

- algorytm szyfrujący
- algorytm deszyfrujący
- kryptograficzna funkcja haszująca
- generator liczb pseudolosowych

Bardziej złożone struktury natomiast wymagają dodatkowego pojęcia:

- protokołu kryptograficznego

1.6

Protokół kryptograficzny – to dobrze zdefiniowany ciąg kroków, w którym mamy do czynienia z co najmniej dwoma różnymi podmiotami, i który ma ściśle określony cel, np. protokół identyfikacji (uwierzytelnienia)

Własności:

1. Pełność opisu (znajomość wszystkich kroków przez użytkownika)
2. Użytkownik musi zgodzić się na jego stosowanie.
3. Protokół powinien być dobrze zdefiniowany i nie może wystąpić jakakolwiek szansa na nieporozumienie.
4. Protokół musi być kompletny: dla każdej możliwej sytuacji musi być podany odpowiedni sposób postępowania.

Przykładowe cele stosowania protokołów:

1. Potwierdzanie tożsamości
2. Autoryzacja użytkownika systemu
3. Podzielenie się tajemnicą tak, by nikt inny nie miał do niej dostępu.
4. Wymiana informacji w sposób bezpieczny.
5. Zabezpieczenie wiadomości email podpisem cyfrowym
6. Wymiana jakiejś liczby losowej w celu generowania klucza.

2. Systemy kryptograficzne klasyczne

W tym wykładzie będziemy stosować uproszczone oznaczenie systemu kryptograficznego.

2.1

Definicja:

Systemem kryptograficznym klasycznym nazywamy taką trójkę

(*K, C, P)*

gdzie:

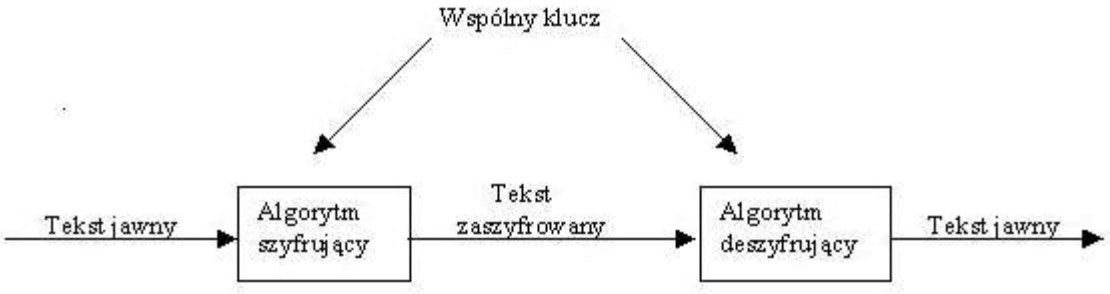
P - zbiór jednostek tekstu jawnego

C - zbiór jednostek tekstu zaszyfrowanego

K - zbiór kluczy

Dla każdego $k \in K$ istnieje reguła szyfrowania $e_k \in E$ i odpowiadająca jej reguła deszyfrowania $d_k \in E$. Wtedy e_k: $P \rightarrow C$ i d_k: $C \rightarrow P$ są funkcjami takimi, że $d_k(e_k(x)) = x$ dla każdego $x \in P$. Funkcje e_k, d_k musza być wzajemnie jednoznaczne:

$x_1 \neq x_2 \Rightarrow e_k(x_1) \neq e_k(x_2)$, i podobnie dla d_k



2.2

Kryptosystem symetryczny -polega na tym, że tekst jawny, czyli oryginalny komunikat, ulega przekształceniu, za pomocą algorytmu szyfrującego oraz klucza, na postać zwaną tekstem zaszyfrowanym. Następnie tekst zaszyfrowany jest przekazywany do odbiorcy, i w momencie odbioru może zostać przekształcony z powrotem w tekst jawny za pomocą klucza oraz algorytmu deszyfrującego. Zarówno w przypadku szyfrowania jak i deszyfracji jest używany ten sam klucz bądź z klucza szyfrującego da się wyliczyć klucz deszyfrujący lub odwrotnie z deszyfrującego da się wyliczyć klucz szyfrujący.

2.3

Szyfr Afiniczny - Jest on szczególnym przypadkiem szyfru podstawieniowego, który zawiera 26 permutacji z liczby 26 wszystkich permutacji 26 –literowego alfabetu. Funkcja szyfrująca ma postać: $e_k(x) = (ax + b) \pmod{26}$, gdzie klucz szyfrujący $k = \{a, b\}$ jest pewną parą liczb z Z_{26}. Dla $a = 1$ otrzymujemy szyfr przesuwający z parametrem b. Liczba $b \in Z_{26}$ może być w szyfrze afinicznym dowolna, natomiast $a \neq 0$ musi spełniać pewien warunek w celu otrzymania jednoznacznej funkcji deszyfrującej. Jeśli oznaczymy $y=(ax + b) \bmod 26$ to warunkiem jednoznaczności funkcji deszyfrującej jest istnienie jedynego rozwiązania x powyższego równania przy zadanym y. Przekształcenie deszyfrujące ma postać :

$x = d_k(y)=(a^{-1} y - a^{-1} b) \bmod 26$ dla a względnie pierwszych z 26. Klucz deszyfrujący jest jednoznacznie wyznaczony przez k i może być łatwo obliczony stosując algorytm Euklidesa do obliczania a^{-1} . Jest to kryptosystem symetryczny. Możliwymi wartościami dla a, tzn. takimi, że $(a, 26)=1$ są: $a=1,3,5,7,9,11,15,17,19,21,23,25$. Szyfr afiniczny ma łącznie $12 \times 26=312$ możliwych kluczy.

Przykład:

Szyfr afiniczny z kluczem szyfrującym k={7,3} . Mamy (7,26)=1 i znajdujemy 7^{-1} mod 26 = 15. Funkcja szyfrująca ma postać: $e_k(x) = 7x + 3$ natomiast funkcja deszyfrująca $d_k(y) = 15y - 19 = 15y+7$ gdzie równości rozumiane są modulo 26. Korzystając z tych wzorów szyfrujemy tekst jawny: kryptografia. Po wykonaniu obliczeń otrzymujemy szyfrogram: VSPEGXTSDAHD.

2.4

Szyfr Cezara -należy on do grupy szyfrów monoalfabetycznych, które polegają na tym, że zamieniają one każdy znak tekstu jawnego na odpowiedni znak kryptogramu, przy czym w całej wiadomości do zamiany każdego znaku jawnego na zaszyfrowany stosuje się odwzorowanie typu *jeden do jednego*.

Szyfr Cezara polega on na przyporządkowaniu każdej literze alfabetu łacińskiego odpowiedniego numeru identyfikacyjnego (np. A=0, B=1 itd.) i dokonaniu przesunięcia numeru każdej litery tekstu jawnego o $k = 3$ pozycje (ma tu miejsce tzw. *przewijanie* - gdy kończy się alfabet przesuwamy się do jego początku). Zakres szyfrowania można oczywiście rozszerzyć na zbiór znaków ASCI (numery od 0-255) lub jakiś inny skończony zbiór *n* znaków.

Funkcja szyfrująca będzie się wówczas wyrażała wzorem:

$F(a)=(a + k) \bmod 26 \rightarrow F(a)=(a+3) \bmod 26$

Funkcja deszyfrująca wygląda następująco:

$F(a)=(a - k) \bmod 26 \rightarrow F(a)=(a-3) \bmod 26$

2.5

Szyfr Vigenere'a -Szyfr Vigenere'a poszczególne litery tekstu jawnego mogą być przekształcone na różne litery alfabetu szyfrogramu. Ten kryptosystem należy do kategorii polialfabetycznych. Liczba możliwych kluczy w szyfrze

Vigenere'a jest bardzo duża, równa jest 26^m. Dla m.=5 przestrzeń klucza jest większa niż $1{,}1 \times 10^7$. Sprawdzenie takiej ilości kluczy jest zadaniem dla komputera, jednak istnieją metody kryptoanalizy, które umożliwiają złamanie szyfru Vigenere'a w czasie szybszym niż przeszukiwanie całej przestrzeni klucza.
Jeśli w szyfrze Vigenere'a długość użytego klucza równa jest długości tekstu jawnego to nazywamy go *szyfrem z kluczem bieżącym*. Jeśli dodatkowo klucz ten jest losowym ciągiem liter lub bitów oraz klucz jest użyty tylko jeden raz to jest to *szyfr z kluczem jednokrotnym (one time pad).*

Szyfrowanie wiadomości przebiega tu na podstawie dowolnie wybranego słowa kluczowego (hasła). W przypadku znaków ASCI może to być dowolny ich ciąg. Do numeru każdego kolejnego znaku tekstu jawnego dodajemy numer odpowiadającego mu znaku słowa kluczowego i uzyskujemy znak kryptogramu. Gdy słowo kluczowe się skończy, bierzemy je kolejny raz od początku. Dla znaków ASCI szyfr Vigenere'a można przedstawić za pomocą poniższej funkcji:

$F_i(a) = (a + k_i) \bmod 255$
Funkcja deszyfrująca będzie oczywiście wyglądała tak:
$G_i(x) = (x-k_i) \bmod 255$
W szyfrze Viegenera im dłuższe i bardziej skomplikowane jest hasło, tym trudniej odszyfrować tekst utajniony. Z kolei równie łatwo jest zauważyć, że gdy hasło będzie jednoznakowe otrzymamy prosty szyfr monoalfabetyczny. Dla określonego zbioru znaków, będącego dziedziną działań na tekstach możemy stworzyć tzw. tablicę Vigenere'a, która określa nam przesunięcia dla dowolnej kombinacji znaków.

Przykład:

Zilustrujmy działanie szyfru na przykładzie zdania "JUTRO JEST SOBOTA". Dla uproszczenia bierzemy pod uwagę tylko litery alfabetu łacińskiego (bez polskich liter).

Przyporządkowujemy im kolejno liczby od 0 do 25. Hasłem będzie słowo "KOT". Tekst jawny dzielimy na 3 literowe grupy (pomijając oczywiście spacje) i przesuwamy pierwszą literę każdej grupy o 10 znaków, drugą o 14, a trzecią o 19. Wygląda to mniej więcej tak:

J	**U**	**T**	**R**	**O**	**J**	**E**	**S**	**T**	**S**	**O**	**B**	**O**	**T**	**A**
K	O	T	K	O	T	K	O	T	K	O	T	K	O	T
T	I	M	B	C	C	O	G	M	C	C	U	Y	H	T

Otrzymany kryptogram to następujący ciąg liczb: "TIMBCCOGMCCUYHT".

2.6

Kryptoanaliza – zajmuje się łamaniem szyfrów

Podział:

- mamy dany tylko kryptogram (the cipertext only)
- znany kryptogram i odpowiadający mu tekst jawny (known plaintext)
- atakujący uzyskuje dostęp do algorytmu szyfrującego (chosen plaintext)
- mamy dostęp do algorytmu deszyfrującego (chosen cipertext)

Cel: znalezienie klucza szyfrującego

Litery najczęściej występujące w języku angielskim
E – 13%
T – 10%
A – 8%
Q – 0,2%
Z – 0,1%
W języku polskim nie ma już tak znaczącego rozrzutu pomiędzy częstościami poszczególnych liter alfabetu. Najczęściej pojawiają się samogłoski A, E, I, O (w ok. 7-8% przypadków)

Analiza częstości – przykład standardowy

Metody kryptoanalizy szyfru afinicznego:

1. Podany jest szyfrogram składający się z kilkunastu liter
2. Otrzymujemy tablicę częstości występowania liter w szyfrogramie
3. Sprawdzamy jakie litery najczęściej występują w szyfrogramie
4. Zakładamy pierwsze przypuszczenie , że dana litera odpowiada e i druga litera odpowiada t, gdzie $e_k(x) = ax + b$ jest funkcją szyfrującą o nieznanych parametrach a i b. Rozwiązujemy układ równań. Po rozwiązaniu układu równań sprawdzamy, czy dane przypuszczenie jest spełnione itd.
5. Jeżeli przypuszczenie jest spełnione prowadzi do funkcji deszyfrującej, która daje tekst jawny

Metody kryptoanalizy szyfru Cezara:

Można przeszukać wszystkie 26 kluczy ale jest to mało efektywne (atak brutalny). Stosujemy metodę analizy częstości – nie wszystkie litery występują z jednakową częstością, wyliczamy experymentalnie procentowe występowanie liter, następnie podstawiając do funkcji sprawdzamy, która jest prawidłowa.

Metody kryptoanalizy szyfru Viegenera:

1. Metoda Kasiskiego
2. Metoda indeksu koincydencji Friedmana

I. Metoda Kasiskiego

Pierwszym etapem jest określenie długości klucza m. Punktem wyjścia jest obserwacja, że dwa identyczne bloki tekstu jawnego są szyfrowane na te same bloki szyfrogramu, jeśli odległość między początkami tych segmentów jest równa *d*, gdzie *d* jest podzielne przez m. Odwrotnie, jeśli zaobserwujemy dwa

identyczne segmenty szyfrogramu (o długości minimum 3 litery – zakładamy, że używamy kluczy o takiej minimalnej długości), wtedy jest duże prawdopodobieństwo, że odpowiadają one identycznym fragmentom tekstu jawnego. Powyższa metoda realizowana jest w następujący sposób: w szyfrogramie szukamy par identycznych fragmentów tekstu i określamy odległości między początkami tych segmentów.
Jeśli znajdziemy kilka takich par i ich odległości równe są odpowiednio d_1, d_2, ..., wtedy możemy postawić hipotezę, że długość klucza m dzieli największy wspólny dzielnik liczby d_i.

II. **Metoda Friedmana** (ustalenie okresu dla szyfru VIGENERE'A)

$p_1p_2p_3$
$k_1k_2k_3$

$c_1c_2c_3$

$c_1=p_1+k_1(\text{mod } 26)$
$c_2=p_2+k_2(\text{mod } 26)$
$c_3=p_3+k_3(\text{mod } 26)$

$$WZ = \sum_{i=1}^{26} Fi(Fi-1)(M(M-1))^{-1}$$

wskaźnik zgodności
gdzie F_i – ilość wystąpień i-tego znaku alfabetu w kryptogramie długości M.

Wykorzystujemy tabelę oczekiwanych wartości wskaźnika zgodności jako funkcji długości okresu szyfru VIGENERE'A

E(d)					
E(1)	E(2)	E(3)	…	E(k)	$k \to \infty$
0,066	0,052	0,047		0,038	

Załóżmy, że z analizy WZ długość okresu wynosi 2. W takim przypadku przeprowadzamy analizę częstości dla liter stojących na parzystych i nieparzystych miejscach w kryptogramie. Załóżmy przykładowo, że X i D są najczęściej występującymi literami na parzystych i nieparzystych miejscach odpowiednio. Postulujemy, że każda z nich szyfruje literę E. Niech l_1 i l_2 oznaczają litery słowa-klucza. Ponieważ #E=4, #X=23, #D=3 więc

$\#\ l_1+4=23 \pmod{26}$
$\#\ l_2+4=3 \pmod{26}$

Stąd
$\#\ l_1=19 \rightarrow\ l_1=\mathrm{T}$
$\#\ l_2=25 \rightarrow\ l_2=\mathrm{Z}$
Klucz Vigenere'a to: TZ

2.7
Cele i podstawowe typy ataków kryptograficznych:

1. Znany tylko tekst zaszyfrowany (cipertext - only). Celem jest odtworzenie tekstu jawnego lub użytego klucza.
2. Znany jest tekst jawny (known plaintext). Przeciwnik zna pewne pary wiadomości – jawną x i odpowiadającą jej zaszyfrowaną wiadomość y. Celem jest znalezienie odpowiadającego klucza, który mógłby użyty do deszyfrowania innych wiadomości.
3. Wybrany tekst jawny (chosen plaintext). Przeciwnik ma możliwość szyfrowania wybranych przez siebie wiadomości i uzyskania odpowiadających szyfrogramów. Może się to odbywać w ten sposób, że przeciwnik ma czasowy dostęp do urządzenia szyfrującego lub zleci komuś zaszyfrowanie danej wiadomości. Celem ataku jest uzyskanie klucza szyfrującego.
4. Wybrany tekst zaszyfrowany (chosen ciphertext). Przeciwnik ma okresowy dostęp do urządzenia deszyfrującego. Może wybrać szyfrogram i uzyskać odpowiadający mu tekst jawny. Celem jest uzyskanie klucza szyfrującego.

3. Algorytmy probabilistyczne i elementy teorii prawdopodobieństwa.

3.1
Algorytmy probabilistyczne

Def .
Algorytm A nazywamy deterministycznym jeśli wartość jego wyjścia y jest jednoznacznie określona przez wartość wejścia x.

A: X -> Y

Przykładem może być deterministyczna maszyna Turinga.

W przypadku algorytmu probabilistycznego wyjście y zależy od x oraz skończonej liczby eksperymentów (losowych) tzw. niepowtarzalność.

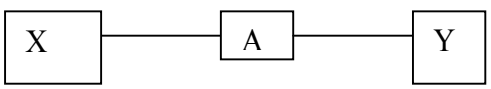
Def.
Algorytm probabilistyczny (zrandomizowany) to algorytm, którego wyjście y zależy od wejścia x i skończonej liczby t_x kroków, w których każdy wykonywany jest niezależnie losowy wybór bitu 0 lub 1 (z prawdopodobieństwem 0,5) i kolejny krok algorytmu może zależeć od wybranych uprzednio bitów.

Def.
Deterministyczne rozszerzenie algorytmu probabilistycznego A nazywamy algorytm deterministyczny A_D, którego wejściem jest para (x, r) gdzie

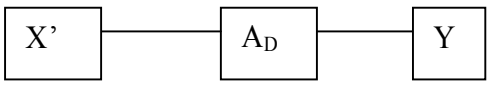
$x \in X, r \in \{0,1\}^{t_x}$ a wyjście $y \in Y$ (zależy od x i r)

Zatem modelem może być deterministyczna maszyna Turinga z rozszerzonym wejściem (x, r) zamiast x.

Schematyczna różnica dwóch typów algorytmów:



Algorytm deterministyczny



Algorytm probabilistyczny

Przykład.
Niech $M = C = \{0,1\}^n = K_0$ Zrandomizowany algorytm szyfrujący $E : m \to E_k(m) = k \oplus m$, gdzie $k \in \{0,1\}^n$ nazywamy szyfrem VERNAMA (One Time Pad)

Uwaga!
Klucz losowy jest długości wiadomości m.
(W zastosowaniach losowy klucz k jest zastępowany przez generowanie ciągów pseudolosowych)

Wynikiem algorytmu probabilistycznego A na wejściu x ⟵ X jest $y \in Y$ z prawdopodobieństwem:

$$pr(A(x) = y) = \frac{|\{r : A_D(x,r) = y\}|}{2^{t_x}}$$

3.2
Elementy teorii prawdopodobieństwa.

Def.

p = (p_1,p_2,...,p_n) taki, że $\sum_{i=1}^{r} Pi = 1$, $0 \le Pi \le 1$

Def.
Przestrzeń probabilistyczna to taka para (X, P_x), gdzie $P_x : X \to [0,1]$
P_x jest rozkładem prawdopodobieństwa, X – skończony zbiór $X = \{x_1,...,x_n\}$,
$Px = (p_1,...,p_n)$, $p(x_i) = p_i$

Def.
Zdarzeniem losowym nazywamy dowolny podzbiór $E \subset X$
Jego prawdopodobieństwo określone jest wzorem:

$$P_x(E) = \sum_{xi \in E} P_x(x_i)$$

Oznaczenia:
pr (E.E') := pr (E ∩ E')
pr(x, E) := pr ({x} ∩ E)

Własności:

1) pr(X) = 1, pr(ϕ) = 0
2) pr(A + B) = pr(A) + pr(B), A $\cap$ B = α
3) pr(X\ A) = 1 – pr(A)

Niech S: (X,Px) $\longrightarrow$ Y, wtedy S oraz P_x indukują rozkład S(Px) na Y zadany wzorem:

$$S(Px)(y) = Px(S^{-1}(y)) = Px(\{x \in X : S(x)=y\})$$

Nazywamy go rozkładem indukowanym Px przez przekształcenie S.

Def.
Przyporządkowanie S:(X, Px) $\longrightarrow$ Y nazywamy zmienną losową o wartościach w zbiorze Y.
Rozkładem zmiennej losowej S nazywamy rozkład indukowany PS(obraz S(Px)) zadany wzorem: PS(y) = S(Px)(y)

Zmienna losowa jest rzeczywista (predykat Boolowski), jeśli y $\in$ Y (Y = {0,1}) odpowiednio.

Uwaga!
Zmienną losowa S: X $\longrightarrow$ Y zadaje przestrzeń probabilistyczną na zbiorze Y.
Odwrotnie, mając przestrzeń probabilistyczną (X, P_x) możemy traktować ją jako zadaną przez pewną zmienną losową, a mianowicie:

$$Sx: (\Omega, P_\Omega) \longrightarrow X, \text{ gdzie } S(P_\Omega) = Px$$

Def.
Niech (X, P_x) przestrzeń probabilistyczna, A, B $\subseteq$ X : P_x(B)>0
Prawdopodobieństwo warunkowe Px(A | B) jest definiowane następująco:

$$Px(A|B) = \frac{P_x(A,B)}{Px(B)}$$

Prawdopodobieństwo warunkowe P_x (x|B), x $\in$ X zadaje rozkład prawdopodobieństwa na przestrzeni X.

Oznaczenia:
P_x(A | B, C) := P_x(A | B $\cap$ C)

Def.
A, B $\in$ X nazywamy niezależnym wtedy i tylko wtedy gdy pr(A, B) = pr(A)pr(B)

Wniosek!
Jeśli pr(B) > 0 to A i B są niezależne wtedy i tylko wtedy gdy pr(A | B) = pr(A

Def.
Przestrzeń probabilistyczna (X, Px) jest przestrzenią łączną o składowych $(X_1,P_1),...,(X_r, P_r)$ (oznaczanie skrótowe $(X_1,X_2,...,X_r)$) wtedy i tylko wtedy gdy

1) $X= X_1 \times X_2 \times ... \times X_r$
2) rozkład Pi jest obrazem $\Pi_i : X \to X_i$

Przestrzenie (x_i, p_i) są niezależne wtedy i tylko wtedy gdy

$$P_x(x_1,...,x_r) = \prod_{i=1}^{r} P_i(x_i)$$

dla dowolnego $(x_1,...,x_r) \in X$ to oznacza, że zbiory x^{-1} (xi) są zdarzeniami niezależnymi w przestrzeni x_i, i = 1,2,...,r . Wtedy X jest nazywany produktem prostym przestrzeni (x_i, p_i), oznaczamy $X = x_1 \times ... \times x_r$

Niech
$S_1 : (x_1,p_1) \to Y_1$
...
$S_r : (x_r, p_r) \to Y_r$
Z przestrzenią łączną $X = X_1,...,X_r$ można związać rozkład łączny $S = S_1,...,S_r$, gdzie $S_i : (xi, pi) \to Xi$
Wtedy $P_x(x_1,...,x_r) = P_x(S1=x1,...,S_r = x_r)$ jest rozkładem łącznym o składowych S_i , i=1,2...,r
Def.
Powiemy, że rozkłady $S_1,...,S_r$ są niezależne wtedy i tylko wtedy gdy

$pr(S_1=x_1,...,S_r = x_r) = \prod_{i=1}^{r} pr(S_i = x_i)$ dla dowolnego $x = (x_1,...,x_r) \in X_1 \times ... \times X_r$

Uwaga!
Przestrzeń łączna jest zgodna z rozkładem łącznym w sensie odpowiedniości między odpowiednimi składowymi.

Notacja.
Niech (XY, P_{XY}) będzie przestrzenią łączną o składowych $(X, P_x)(Y, P_y)$, $x \in X$, $y \in Y$ wtedy $pr(y \mid x) := P_{XY}((x, y)|\{x\} \times Y)$ jest prawdopodobieństwem warunkowym zdarzenia (x, y) przy założeniu, że pierwszą współrzędną jest x

(tzn. jest prawdopodobieństwo zdarzenia, że y pojawia się jako druga współrzędna, jeśli pierwszą jest x)

3.3
Probabilistyka w algorytmach zrandomizowanych.

(X, P_x) – przestrzeń probabilistyczna
A – algorytm zrandomizowany

Eksperyment:
- wybieramy losowo $x \in X$
- obliczamy y = A(x)

Jeżeli $y \in Y$, to powyższy eksperyment można modelować w przestrzeni łącznej (XY, P_{XY}).Rzuty monetą w A(x) są niezależne od x. Zatem mówimy, że $pr(x, A(x) = y) = P_{XY}(x, y) = Px(x)pr(y \mid x) = Px(x)pr(A(x) = y)$ (rozkład łączny jest zadany przez prawdopodobieństwo $pr(y \mid x)$), $x \in X$, $y \in Y$

Uwaga!
Algorytm A nie stanowi cześć eksperymentu w logarytmie A' ani odwrotnie (rzuty monetą w jednym i drugim algorytmie są niezależne) to

$$Pr(A(x), A'(x') = y') = pr(A(x) = y)pr(A(x') = y')$$

W przestrzeni łącznej XY operator rzutowania Π_2 pozwalający na obliczenie rozkładu prawdopodobieństwa na przestrzeni Y indukowanego przez Π_2

$$P(y) = P_{XY}(\Pi_2^{-1}(y)) = \sum_{x \in X} P_{XY}(x|y) = \sum_{x \in X} Px(x)pr(A(x) = y)$$

P_Y możemy traktować jako rozkład indukowany (obraz przy A_D) rozkładu łącznego dla X i losowych rzutów monetą tj.

$$P_Y(y) = pr(\Pi_2^{-1}(y)) = pr(\{(x,r) : A_D(x,r)=y\} = \sum_{x \in X, r \in \{0,1\}^{t_x} : A_D(x,r)=y} Px(x)pr(r)$$

Uwaga!
Rozkład indukowany P_Y bywa oznaczany jako {A(x) : x<- X}

Rozważmy na koniec algorytm A: z wejściem X i wyjściem Y
Niech k: X -> Z będzie pewną własnością elementu $x \in X$ (np. ostatni bit)
Niech B – algorytm probabilistyczny.
Zakładamy, że B ma odczytywać własność elementu x na podstawie y = A(x) i losowego rzucania monetą.

Eksperyment:
- wybieramy losowo $x \in X$
- obliczamy y = A(x) oraz B(y) niezależnie od wyboru x i siebie nawzajem
- sprawdzamy czy B(y) = k(x)

Używając modelowania w przestrzeni łącznej XYZ wiemy, że
Pr(x,A(x) = y, B(y) = k(x)) = pr(x)pr(A(x)) = pr(B(y) = k(x))

4. Rozkłady prawdopodobieństw i ich entropia

4.1

Rozkład indukowany – przestrzeń łączna ($X \times W, P_{X,W}$)

Dane: (X, P_X), Y, A- algorytm probabilistyczny

A_D- rozmieszczenie deterministyczne A

Def.

R- rzuty monetą w algorytmie SA niezależne od wyników X i można je traktować jako zmienną losową $W :\to W_x$ i mamy $A_D : X \times W \to Y$
Gdzie $X \times W = \{(x,w) : x \in X, w \in W_x\} = \underset{x \in X}{W} \{x\} = W_x$

W_X- nazywamy włóknem nad X
Ponieważ wynik zmiennej W_X nie zależy od wyniku X otrzymujemy, że rozkład łączny jest zadany przez iloczyn odpowiednich.

Uzupełniamy ewentualnie zerami wartości prawdopodobieństw zdarzeń (x, w) możemy zakładać, że $X \times W$ jest pod przestrzenią przestrzeni $X \times W$.
Odwrotnie mając rozkład $P_{X,W}(X)$ na przestrzeni $X \times W$ otrzymujemy rozkład indukowany na X

$$\text{Zadany wzorem: } P_X(X) = \sum_{w \in Wx} P_{xw}(X,W)$$

Z drugiej strony rozkład indukowany(warunkowy) -na W_x zadany jest według wzoru:

$$P_{Wx}(w) = P_{XW}((x,w) | \{X\} \times Wx) = \frac{P_{xw}(x,w)}{P_x(x)}$$

Ogólnie rozkład $P_{X_1X_2}$ przestrzeni X_1X_2 można ekstrapolować. Niech $X_1=(X_1,p_1)$.Definiujemy rekurencyjnie rodzinę przestrzeni probabilistycznych na $z \le j \le r$

$$Xj = (X_{j,x}, P_{j,x})x \in X_1 \times .... \times X_{j-1}$$

przez dołączanie kolejnych włókien otrzymujemy w efekcie przestrzeń łączoną oznaczoną przez $(X_1 \times X_2 \times ... \times X_r, P_{x1x2...xn})$

Niech $X=(X, P_X)$

$B : X \to \{0,1\}$ - predykat boolowski

Wtedy:

$$pr(B(x) = 1 : x \xleftarrow{Px} X) = Px(\{x \in X, B(x) = 1\})$$

Def.

Jeśli $Y \subset X$ i mamy rozkład indukowany na przestrzeni Y to można (uzupełniając zerami) rozszerzyć go do rozkładu przestrzeni X(z rozkładem indukowanym przez P_x)

Niech $S : X \to Y$ zmienna losową.

Rozkład indukowany $S(P_X)(y)$ oznaczamy często jako $\{S(x), x \xleftarrow{Px} X\}$-rozkład skoncentrowany na wartościach y= S(x)

Def.

Niech $(X \mid \times \mid W, Pxw)$ będzie rozkładem łącznym wtedy $pr(B(x,w) = 1 : x \xleftarrow{Px} X, w \xleftarrow{Pwx} Wx) := Pxw(\{(x,w) \mid B(x,w) = 1\})$ (najpierw wybieramy $x \in X$ potem $w \in Wx$)

$$pr(B(x_1,x_2,x_3) = 1 : x_1 \in X_1, x_2 \in X_2X_1, x_3 \in X_3X_1X_2) =$$
$$= \sum_{(x_1,x_2,x_3)} pr(x_1) * pr(x_2 \mid x_1) pr(x_3 \mid x_2x_1)$$

z definicji rozkładu łączonego.

Przykład:

X = I_k , W =(Z_n)$_{n \in Ik}$, I_k (n = pq, |p|=(q)=k) p, q- liczby pierwsze

E: $X : W \to Y$

$$E : (n,w) = w^2 (\mathrm{mod}\, n) \in W_n = Z_n,\ n \in I_k$$

4.2
Elementy teorii informacji

X- przestrzeń probabilistyczna(zmienna losowa)

$$H(X) := \sum_{\substack{x \in X \\ pr(x)>0}} pr(x) \ln_2 \frac{1}{pr(x)} == \sum_{\substack{x \in X \\ pr(x)>0}} pr(x) \ln_2 (pr(x))$$ -entropia rozkładu Px

Def.

Obserwacja zdarzenia X niesie informacje. Jeżeli zdarzenie jest mało prawdopodobne to porcje informacji jaką niesie jest duża, a niepewność związana z jego zaistnieniem duże.
Odwrotnie jeśli zdarzenie bardzo prawdopodobne to informacja jaką niesie jego zaistnienie jest mała, a niepewność związana z jego zaistnieniem mała.
Mierzona w bitach na $\log_2 \frac{1}{pr(x)}$

Entropia to zatem średnia ilość informacji wynikająca z przeprowadzenia eksperymentu. Wiąże się ze średnia długości kodu dla potencjalnych wyników zdarzeń

Przykład:

1. W rzucie kostką H(x)=2,6 BITU
2. dla rzutu moneta $\left(\frac{1}{4}, \frac{3}{4}\right) \rightarrow 0{,}8$ BITU

4.3
Probabilistyka w algorytmach zrandomizowanych.

(X, p_X) –przestrzeń probabilistyczna,
p_X- jednostajny,
A-algorytm zrandomizowany

Eksperyment:

- Wybieramy losowo $x \in X$
- Obliczamy y= A(x)

Jeśli $y \in Y$ to powyższy eksperyment można modelować w przestrzeni łącznej (XY, p_{XY}. Rzuty moneta w A(x) są niezależne od x.

Zatem mamy, że pr(x, A(x) =y)= p_{XY} (x, y)= p_X(x)* pr(y | x)= p_X(x) *pr(A(x)=y (Rozkład łączny jest zrandomiozowany przez prawdopodobieństwo warunkowe pr(y | x), $x \in X$, $y \in Y$)

Uwaga:

Jak algorytm A nie jest pod ścieżką algorytmu A' ani odwrotnie(rzuty monetą w jednym o drugim algorytmie są niezależne) to pr{A(x)=y, A'(x')=y'}=pr(A(x) = y) pr(A'(x') = y')

W przestrzeni łącznej XY operator rzutowania $\prod_2$ pozwala na obliczeniowe rozkłady prawdopodobieństwa na przestrzeni Y indukowanego przez $\prod_2$

P_Y= P_{XY}($\prod_2^{-1}$(y))

$$\sum_{x \in X} P_{XY}(x, y) = \sum_{x \in X} p_X(x)\, pr(A(x) = y)$$

Przechodząc do rozszerzenia det. A_D algorytmu A możemy traktować p_Y jako rozkład indukowany (obraz przy A_D) rozkładu łacznego dla X i losowych rzutach monety. tj.

P_Y= P_{XY}($\prod_2^{-1}$(y))=pr({(x,r): A_D(x,r)=y}= $\sum_{x \in X, r \in \{0,1\}^{t_x}} p_X(x) * pr(r)$

Uwaga:

Rozkład indukowany p_Y bywa oznaczany jak {A(x):x $\leftarrow$ X}

Rozważmy na koniec algorytm A: U z wejściem X i wyjściem T.

Niech h: $X \to Z$, h(x) będzie pewnym element $x \in X$ (np. ostatni bit)
Niech B algorytm probabilistyczny który dla wejścia $y \in Y$ odpowiada $B(y) \in Z$.
Zakładając ,że b ma odgrywać własność elementu x na przestrzeni y = A(x) i losowego rzucania monety.

Eksperyment:

- wyliczamy losowo $x \in X$
- obliczamy y = A(x) oraz B(y) nie zależnie od wyboru x i siebie nawzajem
- sprawdzamy według B(y)=według(x).

Używając modelowania w przestrzeni łącznej XYZ mamy, że pr(x,A(x)=y, B(y)=h(x))= pr(x)*pr(A(x)=y'*pr(B(y)=h(x))

5. Rodziny funkcji jednokierunkowych

5.1
Elementy teorii Shanona

Def:
Szyfr E nazywamy doskonale poufnym (perfect secrecy) jeśli rozkład prawdopodobieństwa dla MC jest równy iloczynowi rozkładów dla M i C.

$$E: M \times K \rightarrow C$$

Wniosek:
To znaczy, że $pr(m,c) = pr(m) \times pr(c)$ dla dowolnych $m \in M$ i $c \in C$

Tw. Shanona
Niech M = C i $K = \{0,1\}^n$ i niech E będzie szyfrem Vernama , tj. $m=(m_1, m_2, ..., m_n)$, $k=(k_1, k_2, ..., k_n)$ – losowy klucz niezależny od wiadomości m.

$$E(m) = m \oplus k = (m_1 \oplus k_1, ..., m_n \oplus k_n)$$

Wtedy E jest doskonale poufny wtedy i tylko wtedy gdy K jest przestrzenią kluczy o rozkładzie jednostajnym.

Dowód Tw.

$$pr_{MC} = pr_{MK}(m , m \oplus k) = pr_M(m) * pr_K(m \oplus c)$$

Jeśli M i C są niezależne to:

$$pr_M(m) * pr_C = pr_{MC}(m,c) = pr_M (m) * pr_K(m \oplus c)$$

Zatem:

$$pr_K(m \oplus c) = pr_C(c) \text{ dla dowolnej wiadomości } m \in M$$

Więc K ma rozkład jednostajny.
Implikacja udowodniona.
Odwrotnie, załóżmy, że K ma rozkład jednostajny wtedy:

$$pr_C(c) = \sum_{m \in M} pr_M(m) * \frac{1}{2^n} = \frac{1}{2^n}$$

Zatem C ma rozkład jednostajny i mamy:

$$pr_{MC}(m,c) = pr_{MK}(m , m \oplus k) = pr_M (m) * pr_K(m \oplus c) = pr_M (m) * \frac{1}{2^n} = pr_M (m) * pr_C(c)$$

A więc E jest doskonale poufne czego należało dowieść.

5.2
Funkcje jednokierunkowe

Def. 1
Niech P będzie pewnym problemem obliczeniowym.
A nazywamy algorytmem Monte Carlo dla problemu P wtedy i tylko wtedy gdy A jest algorytmem probabilistycznym o czasie działania wielomianowym, który daje poprawną odpowiedź dla problemu P z prawdopodobieństwem $\geq \frac{2}{3}$

Tzn.

$time_A(x) \leq Q(|x|)$ oraz pr(A(x) jest poprawną odpowiedzią dla problemu P) $\geq \frac{2}{3}$

gdzie $Q(x) \in Z[x]$ i jest wielomianem dodatnim o współczynnikach całkowitych.

Def 2.
Algorytm probabilistyczny A nazywamy algorytmem LAS VEGAS dla problemu P wtedy i tylko wtedy gdy zawsze odpowiada poprawnie (daje poprawną odpowiedź) dla P oraz gdy oczekiwany czas działania algorytmu jest ograniczony przez pewien dodatni wielomian $Q \in Z[x]$, tzn.

$$E(time_A(x)) = \sum_{t=1}^{\infty} t * pr(time_A(x) = t) < Q(|x|) \text{ dla dowolnego x.}$$

5.3
Funkcje wykładnicze

Niech I = {p,q: p-liczba pierwsza, q – generator Z_p^*}

$E_{xp} := \left\{E_{xp}(p,q) : Z_{p-1} \to Z_p^* \mapsto q^* (\text{mod } p)\right\}_{(p,q)\in I}$ nazywamy dyskretną rodziną wykładniczą.

Uwaga.
Ponieważ $E_{xp}(p, q)$ jest funkcją wzajemnie jednoznaczną i izomorfizmem do poszczególnych grup więc jest izomorfizmem grup Z_{p-1} i $Z^*_{p.}$
Izomorfizm odwrotny będziemy oznaczali przez Log (p, q), a rodzinę wszystkich takich izomorfizmów rodziną logarytmu dyskretnego (logarytmią dyskretną),
tzn. $Log := (Log_{(p,q)} : Z^*_p \to Z_{p-1})_{(p,q)\leftarrow I}$

Założenie logarytmu dyskretnego:

Niech $I_k=\{(p ,q)\in I,|p|=k\}$ gdzie $k\in K$ i niech $Q(x) \in Z[x]$ będzie wielomianem dodatnim.
Niech A(p, q, y) będzie algorytmem probabilistycznym wtedy istnieje $K_0 \in N$ takie, że:

$$Pr(A(p, q, y) = Log_{(p,q)}(y): (p, q) \xleftarrow{u} I_k, y \xleftarrow{u} Z^*_p) < \frac{1}{Q[k]}$$

Żaden algorytm wielomianowy nie obliczy logarytmu z dużym prawdopodobieństwem.

5.4

I = {(n, e), n = pq, $p \neq q$ – liczby pierwsze oraz $0<e<\vartheta(n)$ takie, że NWD(e, n) = 1 gdzie $\varphi(n)$ jest funkcją Eulera}

1. Rodzina RSA
   $RSA = (RSA_{n,e}, Z^*_n \rightarrow Z^*_n, x \mapsto x^e \bmod n)_{(n,e)\in I}$
2. e=2 Rodzina kwadratowa (Robina)
   I = {n: n=pq, $p \neq q$ – liczby pierwsze o tej samej długości $\Rightarrow |p|=|q|$}
   $S_q := \{S_q n: Z^*_n \rightarrow Z^*_n,\ x \mapsto x^2 \bmod n)_{n \in I}$

Uwaga!
Elementy rodziny S_q nie są funkcjami różnowartościowymi ani „na”.

Def.
Założenie trudności faktoryzacji.
Niech $I_k=\{n\in I, n = pq, |p| = |q| = k\}$ oraz $Q(x)\in Z[x]$ będzie dowolnym wielomianem dodatnim.
Niech A_n będzie algorytmem probabilistycznym.
Istnieje stała $k_0 \in N$ taka, że

$$Pr(A(n) = p, n \xleftarrow{u} I_k) \leq \frac{1}{Q(k)}$$ dla wszystkich $k \geq k_0$ wielomianowym

6.Schematy podpisu cyfrowego

Schematem podpisu nazywamy trójkę algorytmów (K, S, V) gdzie K jest algorytmem generacji kluczy, S algorytmem podpisywania i V algorytmem wersyfikacji podpisu.

K:

wejście: 1^k gdzie k jest parametrem bezpieczeństwa
wyjście (pk, sk) – klucz publiczny i prywatny

S:

wejście: sk, m gdzie m jest wiadomością
wyjście: σ czyli podpis wiadomości m

V:

wejście: T(pk , m, σ)
wyjście: {0,1} gdzie 1 – podpis akceptowany a 0 podpis odrzucony

Bezpieczeństwo schematu (podpisu) związane jest ze środkami i celem (poziomem sukcesu) potencjalnego przeciwnika.

Środki:

1. Atak tylko przy znajomości klucza publicznego podpisującego. Adwersan zna tylko klucz publiczny podpisującego.
2. Atak ze znanym podpisem. Przeciwnik posiada klucz publiczny oraz podpis pod znaną wiadomością.
3. Atak z wybraną wiadomością. Przeciwnik wybiera (na początku) zbiór wiadomości (m_1,...,m_k) i poznaje podpisy pod tymi wiadomościami.
4. Atak adaptacyjny. Przeciwnik przygotowuje kolejne wiadomości do podpisu obserwując wartości wcześniejszych podpisu.

Ze względu na poziom sukcesu przeciwnika wyróżniamy następujące cele (fałszerstwa):

1. Fałszerstwo egzystencjalne – przeciwnik osiąga podpis pod pewną wiadomością (fałszuje) niekoniecznie przez niego wygenerowaną.
2. Fałszerstwo selektywne – przeciwnik osiąga sukces i potrafi sfałszować podpis pod wybraną przez niego wiadomością.
3. Fałszerstwo uniwersalne – przeciwnik potrafi sfałszować podpis pod dowolną wiadomością.
4. Złamanie klucza prywatnego – przeciwnik potrafi obliczyć klucz prywatny podpisującego.

W tych kategoriach najbardziej bezpieczny schemat to taki, który nie pozwala osiągnąć fałszerstwa egzystencjalnego nawet przy ataku adaptacyjnym.

Def.
Niech D = (K, S, V) będzie schematem podpisu cyfrowego. Fałszerzem egzystencjalnym F dla schematu D nazywamy algorytm probabilistyczny, który mając na wejściu klucz publiczny podpisującego otrzymuje na wyjściu parę (m, σ) - wiadomość wraz z podpisem.

F:

wejście: pk
wyjście: (m, σ)

F osiąga sukces dla pk wtedy i tylko wtedy gdy σ jest poprawnym podpisem pod wiadomością m to znaczy, że V((pk,F(pk))=1. F dokonuje ataku adaptacyjnego jeśli obliczając F(pk) może kolejno generować wiadomości $\tilde{m}$ otrzymując (od wyroczni) poprawne podpisy pod nimi.

Fałszerz egzystencjalny – dokonujący ataku adaptacyjnego z wybraną wiadomością.

Przykład.
Schemat EL GAMALA

K: generujemy (p, g, y) – klucz publiczny, (p, g, x) – klucz prywatny gdzie p to liczba pierwsza, g to generator grupy Z_p^* oraz $y=g^x \pmod p$

S: 1. wybieramy losowe $k \geq 1$ i $k \leq p-2$ takie, że NWK(k,p-1) = 1
2.obliczamy $r = g^k$ $s = k^{-1}(m - rx) \pmod{p-1}$
3. C = (r, s) jest podpisem pod wiadomością m

V: 1.sprawdzamy czy $1 \leq r \leq p-1$, jeśli nie to podpis odrzucamy, czyli $V(pk, m, \sigma) = 0$
2.obliczamy $v = g^m \pmod p$ i $w = y^r r^s \pmod p$
3. jeśli v = w to podpis akceptujemy $V(pk,k^{-1}) = 1$ w pozostałych przypadkach odrzucamy podpis tj. $V(pk, m, \sigma)=0$

7. Podpisy RSA, El Gamala i rabina

Podpis El Gamala

Def.
Schemat podpisu cyfrowego nazywamy odpornym ze względu na atak adaptacyjny (z wybraną wiadomością) wtedy i tylko wtedy gdy dla dowolnego fałszerza egzystencjalnego F, który wykonuje atak adaptacyjny oraz dowolnego wielomianu P dodatniego takiego, że istnieje stała $k_0 \in N$ taką, że dla wartości parametru bezpieczeństwa k>=K_0 prawdopodobieństwo sukcesu fałszerza F jest mniejsze niż $\frac{1}{P(k)}$.

Podpis El Gamala:

(p, g, y) – klucz publiczny
(p, g, x) – klucz prywatny
gdzie $y = g^x \pmod p$

Podpisywanie:
1) Wybieramy k losowe <= p-2 i >=1 takie, że k $\perp$ p-1
2) Obliczamy $r=g^k \pmod p$
   $s=k^{-1}(n - rx) \pmod{p-1}$
3) (m, r, s) – podpis pod m

Uwaga!
!) Jeśli przeciwnik pozna wartość K to znajdzie (obliczy) klucz prywatny podpisującego (x). Przeciwnik mając podpis (m, r, s) i k będzie chciał obliczyć x w następujący sposób:
Oblicza
$sk = m-rx \pmod{p-1}$
$X = (m-sk)r^{-1} \pmod{p-1}$

Uwaga 2!
Jak podpisujemy użyte dwukrotnie tej samej wartości k do podpisu pod różnymi wiadomościami to skompromituje swój klucz prywatny.

Na mocy uwagi 1 wystarczy wykazać, że przeciwnik jest w stanie obliczyć k.
Przeciwnik wie, że (m,r,s) i m' $\neq$ m oraz (m', s', r') – odpowiednie podpisy.

Z definicji mamy, że r'= $r = g^k \pmod p$
Zatem
$s=k^{-1}(m-rx) \pmod{p-1}$

$s'=k^{-1}(m'-rx)(\text{mod } p-1)$
z tego wynika: $(s-s')=k^{-1}(m-m')(\text{mod } p-1)$

Zatem przeciwnik oblicza k ze wzoru:

$$k=(m-m')(s-s')^{-1}(\text{mod } p-1)$$

Jeśli okaże się, że (m-m') ma nietrywialny wspólny dzielnik d z p-1 to jest on też dzielnikiem (s-s'). Zatem, biorą obie strony tej równości $k(s-s')=(m-m')(\text{mod } p-1)$ przez wspólny dzielnik otrzymany kongruencję, z której k można wyliczyć (mod $\frac{p-1}{d}$) oraz (mod d), a zatem ostatecznie (mod p-1).

Podpis Rabina

m=pq gdzie p i q to liczby pierwsze.
Niech M – przestrzeń wiadomości jawnych, h-funkcja hashująca bezkolizyjna taka, że:

$$B: M \times \{0,1\}^k \longrightarrow Z_m$$
$$(m, x) \longrightarrow h(m, x)$$

x-losowy parametr taki, że $h(m, x) \in QR_n$ (reszta kwadratowa mod n)

Podpis pod wiadomością m jest trójka (m, x, y) gdzie $y = \sqrt[2]{h(m,x)(\text{mod} n)}$

<u>Weryfikacja:</u>
Polega na sprawdzaniu czy $h(m,x)=y^2(\text{mod } n)$

Uwaga!
Informacja zapadkowa (p, q) można traktować jako klucz prywatny podpisującego, natomiast n wraz z funkcją haszującą h oraz generatorem liczb pseudolosowych jako klucz publiczny.

System RSA

Podpisywanie i szyfrowanie są dualne w tym systemie.

<u>Generacja kluczy:</u> wybieramy losowo 2 duże liczby pierwsze p i q.
Definiujemy m = pq. Następnie losujemy e takie, że
NWD(e,(p-1)(q-1))=1 ($e \perp (p-1)(q-1)$) i 1<e<=n-1 oraz obliczamy d:$ed=1(\text{mod } (p-1)(q-1))$

Klucz publiczny to para (e, n)
Klucz prywatny to para (d, n)

Podpisywanie:
Dla danej wiadomości m < n podpisem jest para
(m, σ):m^d(mod n)= σ(m)

Weryfikacja:
Dla dowolnego podpisu (m, σ(m)) i klucza publicznego (e, n) sprawdzamy, czy $\sigma(m^e)$=m (mod n)

W praktyce:
Stosujemy do podpisu wartość funkcji haszującej h na wiadomości m tzn. podpisujemy h(m) zamiast m ze względów bezpieczeństwa.

8. Testy pierwszości

Jak stwierdzić czy dana liczba jest pierwsza?

8.1

Założenie:

m- losowa liczba z przedziału $\left(\frac{x}{2}, x\right)$, gdzie x- odpowiednio duży parametr

m- me- liczba pierwsza to m→m+1
liczby nieparzyste to m→m+2
Za każdym razem testujemy na pierwszość.

1) $m \in \left[\frac{x}{2}, x\right]$- losowa nieparzysta
2) TP(a) jeśli odp. Tak koniec wpp. Itd. To 3)
3) TP (m+2) itd.

Klasyfikacja testów → Deterministyczne
Klasyfikacja testów → Probabilistyczne

Wymagania dla algorytmu testowania na pierwszość:

1. Efektywność obliczania(szybkość)- warunek najważniejszy.
2. Poprawność.

3. Uniwersalność(czy my tak naprawdę testujemy wszystkie liczby czy ich podzbiory).

8.2

TEST MILLERA- RABINA:

1) Efektywny.
2) Uniwersalny.

Własności:

Jeśli odp. NIE – to prawda

Jeśli odp. TAK to prawdopodobieństwo pomyłki jest < $\left(\frac{1}{4}\right)$

8.3

ALGORYTM MILLERA- RABINA:

1) n- testowana liczba
   Wybieramy losowe z przedziału $1 \le a \le n$ (zwane świadkiem)
   Sprawdzamy $a^{n-1} = 1(\mathrm{mod}\ p)$ - warunek FERMATA

   Jeśli NIE to odp. Algorytmu NIE (odnosi się do warunku)
   Jeśli TAK (mod =) to przejdź do 2)
2) $n-1 = 2^R * s$, s- nieparzysta, jeśli istnieje takie $0 \le r \le R$, takie że

   $a^{2^r * s} = -1(\mathrm{mod}\ p)$ lub $a^s = -1(\mathrm{mod}\ p)$

   To odp. Algorytmu NIE wpp. TAK

Weryfikacja:

Algorytm MILLERA- RABINA dla danej wejściowej pierwszej daje zawsze poprawną odpowiedz, warunek wynika z twierdzenia FERMATA.

Wiemy, że $a^{n-1} = a^{2^R * s} = -1(\mathrm{mod}\ p)$ gdzie p- liczba pierwsza to zachodzi dla r= R.

Rozważmy:

$$a^{2^{R-1}*s} = (a^{2^R*s})^{\frac{1}{2}} (\mathrm{mod}\, p) = \sqrt[2]{a^{n-1}} = \sqrt[2]{1(\mathrm{mod}\, p)}$$

Zatem pytamy o rozwiązanie równania $x^2 = 1(\mathrm{mod}\, p)$

$p \mid: x^2 - 1 = (x+1)(x-1) \Rightarrow p \mid: x-1 \,\mathrm{lub}\, p \mid: x+1 \Rightarrow x = 1(\mathrm{mod}\, p) \,\mathrm{lub}\, x = -1(\mathrm{mod}\, p)$

Jeśli $a^{2^{R-1}*s} = -1(\mathrm{mod}\, p)$ to odp. TAK jest to algorytm poprawny

Jeśli $a^{2^{R-1}*s} = 1(\mathrm{mod}\, p)$ to obliczam $a^{2^{R-1}*s} = \pm 1(\mathrm{mod}\, p)$ i powtarza procedurę ostatecznie odpowiedz algorytmu będzie TAK.

PODSUMOWANIE:

1) Algorytm efektywny.
2) Może się pomylić.
3) Jak odpowie ,że liczba jest złożona to jest i nigdy nie powie pierwsza.

8.3
Algorytm Lehmana

Twierdzenie(kryterium pierwszości Lehmana – efektywne obliczeniowo):
Liczba $n \in N$ jest pierwsza wtedy i tylko wtedy, gdy zbiór

$$\left\{ a^{\frac{n-1}{2}} (\mathrm{mod}\, n), a \in Z_n^* \right\} = \{-1, 1\} \qquad (*)$$

„⇒” Jeśli *n* jest pierwsze to $a^{n-1} = 1 \pmod n$ na mocy twierdzenia Fermata. Co więcej jest ciałem *n* - elementowym. Więc równanie $x^2 - 1$ na tym ciele ma dokładnie dwa rozwiązania: $x = \pm 1 \pmod n$.

Kładąc otrzymujemy, że $x^2 = 1 \pmod n$ zatem $a^{\frac{n-1}{2}} \in \{-1, 1\}$ dla dowolnego

$$\left| a \in Z_n^* \text{ tzn } \left\{ a^{\frac{n-1}{2}},\ a \in Z_n \right\} \in \{-1,\ 1\} \right.$$

Ale mamy, że $1^{\frac{n-1}{2}} = 1(\mathrm{mod}\, 1)$. Z drugiej strony: ponieważ Z_n^* jest cykliczna to biorąc *a* równe generatorowi Z_n^* otrzymujemy, że najmniejszy wykładnik *k* taki, że $a^k = 1 \pmod n$ jest równy $n-1$ i wtedy $a^{\frac{n-1}{2}} \neq 1(\mathrm{mod}\, n)$ a więc $1^{\frac{n-1}{2}} = -1(\mathrm{mod}\, 1)$.

„⇐” Załóżmy, że $n \in P$, gdzie P to zbiór liczb pierwszych.
Pokażmy, że nie zachodzi następująca równość (*)

I) $n \notin n1\ n2$, $n2 \perp n1$, (n1 , n2 >1)

Zakładamy że nie zachodzi (*)

Istnieje $a \in Z_n^*$ takie, że $a^{\frac{n-1}{2}} = -1(\mathrm{mod}\,1)$

Niech $b \in Z_n^*$ będzie rozwiązaniem układu konkurencji.

$$\begin{aligned} b &= a(\mathrm{mod}\, n_1) \\ b &= 1(\mathrm{mod}\, n_2) \end{aligned}$$

(także b istnieje na mocy twierdzenia chińskiego o resztach gdyż $n_1 \perp n_2$)
Wtedy

$$\begin{aligned} b^{\frac{n-1}{2}} &= a^{\frac{n-1}{2}} = -1(\mathrm{mod}\, n_1) \\ b^{\frac{n-1}{2}} &= a^{\frac{n-1}{2}} = 1(\mathrm{mod}\, n_2) \end{aligned}$$

Zatem $b^{\frac{n-1}{2}} \neq +-1(\mathrm{mod}\, n)$ wbrew założeniu (*)

II) $n = p^a$, p - liczba pierwsza

Gdy $n = p^a$ gdzie $a \geq 2$, to w grupie Z_p^* której rząd jest równy

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

istnieje element a rzędu p. Niech będzie tym elementem, wtedy:

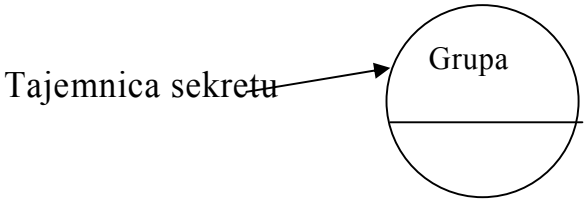
$$\left.\begin{aligned} a^{n-1} &= 1(\mathrm{mod}\, n) \\ a^p &= 1(\mathrm{mod}\, n) \end{aligned}\right\} \text{ ponieważ } p \mid n \text{ więc } \mathrm{rzad}\, a = 1 \text{ co niemożliwe.}$$

Otrzymana sprzeczność dowodzi implikacji „⇐”. c.k.d.

9. Schematy progowe

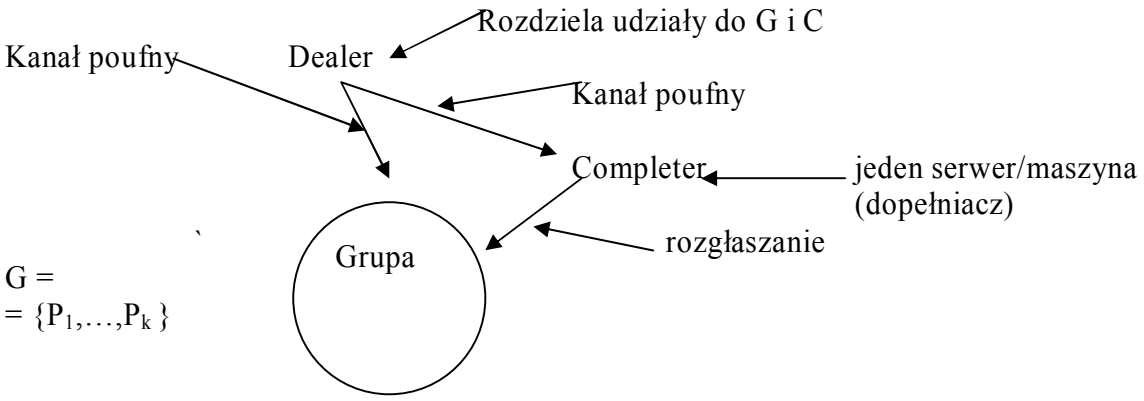
9.1
SCHEMATY PROGOWE

Quorum/próg → Podpisy progowe(autoryzacja wiadomości)
Quorum/próg → Deszyfrowanie progowe



Jeśli co najmniej połowa grupy wymieni informacje, to tajemnica zostanie zrekonstruowana, jeżeli mniej niż połowa to nie.

9.2
OGÓLNY SCHEMAT:



Uczestnicy:(D, G, C, A)

9.3
KOMUNIKACJA:

1) Rozgłaszanie(kanały publiczne)
2) Transmisja poufna (1-1) 2 węzły poufny kanał

Dealer rozdziela udziały tajemnicy, w postaci f(1),f(2),…f(k) dla grupy G oraz f(k+1),…f(2k) dla Completera, gdzie stopień f=K-1 (wielomian stopnia K-1 potrzebuje, co najmniej K punktów = (węzłów do jego rekonstrukcji).

9.4
SCHEMAT PODPISU I DZIELENIE SEKRETU:

(K, S, V)-wybrany schemat podpisu (w naszym przypadku zakładamy, że podpis jest indukowany, przez homomorfizm jednokierunkowy fi (duże)) wtedy podpis (m,σ) jest weryfikowany, sprawdzeniem czy $fi(\sigma)$=m

Def:

Podziałem sekretu s typu (d, l) nazywamy zbiór $(s_1,\ldots,s_l)$, taki ,że dla dowolnych d elementów S_i można zrekompensować sekret s, ale z żadnej

mniejszej ilości nie uzyskamy żadnej informacji na temat s. Będziemy oznaczać taki schemat następująco

$$(s_1,\ldots s_l) \xrightarrow{(d,l)} s$$

(S_i- nazywamy udziałem sekretu s)

9.5
KLUCZE KRYPTOGRAFICZNE I ALGORYTMY:

1) (pk_G, sk_G) - dla grupy

   Public securit

2) (pk_i, sk_i) i ≤ K & i ≥ 1 – dla członków grupy
3) Klucze weryfikujące (pary) (r, rk_i) członków G (1 ≤ i ≤ k)
4) Algorytmy:
   1) Generacja kluczy autoryzujących autoryzujących-globalna informacja publiczna
      (sk, pk) ← (I) gdzie sk pod przyrodnym kluczem członka P_i (1 ≤ i ≤ k) gdzie pk pod przyrodnym kluczem członka P_i (1 ≤ i ≤ k)
   2) Algorytm (d,2K) progowego generowania klucza.
      $(si)_{i \le K} \leftarrow (I, pk_G)$

      Takie, że $(s_1,\ldots s_{2k}) \xrightarrow{(d,2k)} sk_G$ gdzie d=K-1 i (pk_G, sk_G)jest parą kluczy dla G otrzymana algorytmem (K, S, V)
   3) Algorytm generacji udziału podpisu

      $$\sigma i = \sigma i(s_i) \leftarrow (I, m, (s_i), (r, rk_i))$$

      wyjście wejście
   4) Algorytm weryfikacji udziałów podpisu
      TAK

      $$\longleftarrow \quad (I, pk, (r, rk_i), \sigma_i)$$

      NIE
   5) Algorytm składający(łączący) udziały podpisu

      $\sigma = \sigma(m) \leftarrow (I, \sigma_1 \ldots \sigma_d)$ Taki, że weryfikacja $pk(m, \sigma) = akceptacja(TAK)$
   6) Algorytm weryfikacji podpisu
      TAK lub NIE $\leftarrow (I, m, \sigma)$

10. Pełnomocnictwo cyfrowe

Twórcami tego schematu są (Chin-Chen-Chan, Hui-Feng-Huang). Bazuje on na protokole podziału tajemnicy pochodzącym od Shamira (używa systemu RSA). Nowością jest to, że w tym podpisie wszyscy użytkownicy są łatwo rozpoznawani, (co wcześniej nie było osiągalne). Ponadto jest szybki obliczeniowo.
P0, P1,..., Pn – pełnomocnicy.

10.1

FAZA WSTĘPNA

Zaufany serwer (trusted authority) generuje klucze RSA: (ei, Ni)- publiczny, di - prywatny, gdzie Ni = pi qi (Ni jest iloczynem dwóch liczb pierwszych) przekazuje wartości kluczy odpowiednim pełnomocnikom:

(e_0, N_0), d_0 φ (N_0) dla P_0
(e_i, N_i), d_i φ (N_i) dla P_i, gdzie $i \neq 0$.

10.2

GENEROWANIE KLUCZA PEŁNOMOCNICTWA

1)P_0 tworzy informację m_0 N , □ $m_0 \neq 1$ (zawierającą identyfikator P_0 i czas udzielanego pełnomocnictwa) i oblicza

$$\left.\begin{aligned} D &= d_0^{m_0} \bmod(\phi(N_0)) \\ e &= e_0^{m_0} \bmod(\phi(N_0)) \end{aligned}\right\} \rightarrow ED = 1(\bmod \phi(N_0))$$

$$(d_0 e_0)^{m_0} = 1^{m_0} = 1(\bmod \phi(N_0))$$

Gdzie D - klucz tajny pełnomocnictwa, E - klucz jawny pełnomocnictwa

2) P_0 publikuje

$\left[m_0, E, \sigma_0(m_0) = m_0^{d_0} (\mathrm{mod}\, N_0), \sigma_0(E) = E^{d_n} (\mathrm{mod}\, N_0)\right]$ oraz
bezpieczną funkcję haszującą h .
3) Każdy użytkownik może zweryfikować wiarygodność P_0 obliczając przy użyciu e_0 wartości:

$$\sigma_0(m_0)^{e_0} = m_0 (\mathrm{mod}\, N_0)$$
$$\sigma_0(E)^{e_0} = E (\mathrm{mod}\, N_0)$$

10.3

PODZIAŁ TAJEMNICY

1) Wybór wielomianu podziału tajemnicy.
• P0 wybiera losowe liczby całkowite a_i [0, □ N0) i definiuje

$$f(x) - D + a_1 x_1 + \ldots + a_{i-1} x_{i-1} \in Z[x]$$

wtedy f ma postać:

$$f(x) = \sum f(b) g_b(x), \qquad \text{gdzie} \qquad g_b(x) \in Q[x]$$

• P0 definiuje wielomian pomocniczy f(x) = n! skąd

$$H(x) = \sum_{b \in B} f(b) G_b(x) \text{ ,gdzie } \quad G_b(x) = n!, g_b(x) \in Z[x]$$

2) Wyznaczanie udziałów S pomiędzy użytkowników P_1,..., P_n
P_0 oblicza udziały dla b□B wzorem:

$$S_b = f(b) G_b(0) = f(b) g_b(0) n! \in Z_i$$

Wysyła kryptogramy postaci $(\sigma_0(S_b))^{e_0} (\mathrm{mod}(N_b))$ do wszystkich b □ P .
3) Sprawdzenie wiarygodności udziałów. P_b oblicza

$((\sigma_0(S_b))^{e_0})^{db} = \sigma_0(S_b) (\mathrm{mod}\, N_b)$, a następnie

$(\sigma_0(S_b))^{e_0} = (S_b) (\mathrm{mod}\, N_b)$ (jest to weryfikacja podpisu dealera jego kluczem publicznym).

10.4

TWORZENIE PODPISU PEŁNOMOCNICTWA

1) Wzajemne uwierzytelnianie grupy B. Każdy P_0 oblicza
$y_b = h(b, M)d_b (\mathrm{mod}\, N_b)$ (jest to wartość publicznie znanej funkcji haszujacej dla danej wiadomości z zawartą informacja o adresacie) i przesyła wyniki do pozostałych udziałowców.
Każdy z nich weryfikuje autentyczność otrzymanego kryptogramu obliczając
$y_b^{e_b} = h(b, M)d_b (\mathrm{mod}\, N_b)$.
2) Składanie częściowego podpisu. Użytkownik P_b oblicza wartość

$$\sigma_{\mathrm{b}}(\mathrm{M}) = \mathrm{h}\left(\left[\sum_{\mathrm{b}\in\mathrm{B}} h(b,M)^{d_b}\right], M\right)^{f(b)G_b(0)} \quad (\mathrm{mod}\, N_0)$$

$f(b)G_b(0)$ - to otrzymane od Dealera udziały każdego użytkownika) i wysyła parę $\left[\sigma_b(M), \sigma_b(M)^{d_b}\right]$ do każdego pozostałych użytkowników.

3) Weryfikacja poprawności „częściowych" podpisów.

• P_i sprawdza, czy $\left(\sigma_b(M)^{d_b}\right)^{e_b} = \sigma_b(m)(\mathrm{mod}\, N_0)$
• Jeśli tak, to oblicza wartość:

$$\prod_{b\in B}\sigma_b(M) = h(y_B^{(d)}, M)\sum_{b\in B} f(b)G_b(0) = h\left(y_B^{(d)}, M\right)^{Dn!}$$

gdzie $y_B^{(d)} = \sum_{b\in B} h(b, M)^{d_b}$

4) Obliczanie podpisu z pełnomocnictwa.

• Każdy P_i oblicza wspólną wartość $\sigma_b(M)$ jak następuje

$$\sigma_b(M) = \prod_{b\in B}\sigma_b(M)^{\frac{1}{n!}} (\mathrm{mod}\, N_0)$$

Podpisem pełnomocnictwa pod wiadomością M jest trójka

$(\sigma_b(M), R, B)$, gdzie R jest rozwiązaniem układu równań:

$R = h(b, M)^{d_b} (\text{mod } N_b)$, b□B . (równań takich będzie tyle, ilu jest użytkowników – czyli b).

10.5

WERYFIKACJA PODPISU

1) Sprawdzenie wiarygodności pełnomocnictwa.

• Odbiorca weryfikuje prawdziwość $m_0(Id(P_0, t))$, t□ (gdzie Id -identyfikator

P_0, t – czas ważności pełnomocnictwa) oraz E
(gdzie E klucz publiczny pełnomocnictwa) obliczając

$\sigma_0(m_0)e_0(= m_0), \sigma_0(E)e_0(= E)$

2) Weryfikacja wiarygodności grupy na podstawie podpisu

$(\sigma_b(M), R, B)$

• Odbiorca oblicza $R(\text{mod } N_b) = h(b, M)^{d_b} (\text{mod } N_0)$.

• Sprawdza czy $\left(\sigma_b(M)^{d_b}\right)^{e_b} = \sigma_b(m)(\text{mod} N_0)$

• Jeśli tak, to oblicza $y_B^{(d)} = \sum_{b \in B} h(b, M)^{d_b}$ i sprawdza czy

$\sigma_B(m)^E = h(y_B, M)(\text{mod } N_0)$ Jeśli tak, to $(\sigma_b(M), R, B)$ uznaje za prawdziwy.

11. Założenia podpisu Cramera - Shoupa

11.1
SCHEMAT CYFROWY CRAMERA-SHOUPA

Założenia:

1.Silne założenie RSA tzn. Niech I {n= pq, p, q- liczby pierwsze o tej samej długości}. Ik={neI: |p|= |q| =k}. Dla dowolnego

I_k wielomianu Q: probabilistycznego algorytmu A o wejściu $(n, y) \in I \times Z^* n$ i wyjściu $(e, x) \in Z_{>1} * Z^* n$
Istnieje, $k_0 \in N$ takie że

$$pr(x^e = y) : n \xleftarrow{u} I_k, y \xleftarrow{u} Z^* n : (e, x) \leftarrow A(n, y) \leq \frac{1}{Q(k)} \text{ dla } k \geq k_0$$

Założenie to jest silniejsze niż RSA, tzn. że wynika z niego klasyczne założenie RSA.

Def:

Liczba pierwsza p nazywamy liczbę Sophie Germain $\Leftrightarrow$ gdy 2p+1 jest tez liczba pierwsza. Heurystycznie mamy ,że istnieje wielomian Q taki że $\#\{p \in P_{SGK}\} \geq \frac{2^k}{Q(k)}$

Wniosek:

Załóżmy, że $\#\{p \in P_{SGK}\} \geq \frac{2^k}{Q(k)}$. Wtedy silne założenie RSA implikuje, że dla dowolnego wielomianu dodatniego R i algorytmu probabilistycznego A o wejściu(n, y) i wyjściu (x, y) istnieje, $k_0 \in N$ takie, że

$$pr(x^e = y), n \in I_{SGK}, I \in Z^* n, (e, x) \leftarrow A(n, y) \leq \frac{1}{p(k)}$$

Dla $k \geq k_0$ gdzie I_{SGK}- zbiór $\{n = pq : n \in I_k, p, q \in P_{SG}\}$

11.2
SCHEMAT CRAMERA-SHOUPA

Nazywany jest w nim algorytm probabilistyczny Germ Prime o wejściu (1^k)
W fazach generacji kluczy i podpisywaniu wiadomości o własności.

1. Wejście 1^k Germ Prime
   Wyjście k- bitowa liczba pierwsza $\in P_{SGK}$
2. szansa na powtórzenie tego samego wyjścia w algorytmie Germ Prime , w którym z R(k) przebiegów algorytmu jest $\leq \frac{1}{p(k)}$ dla dowolnego wielomianu dodatniego P. Dokładniej:

$$pr(e_{j1} = e_{j2} \_ dla \_ j_1 \neq j_2, e_j \in Gen\,\mathrm{Pr}\,ime(1^k), 1 \leq j \leq p(k)) \leq \frac{1}{p(k)}$$

<u>*Dowód:*</u>

Liczb pierwszych p długości k jest asymptotycznie $\frac{2^k}{k}$

Zatem ilość par to $\left(\frac{2^k}{k}\right)^2$ .Zatem prawdopodobieństwo, że $e_{j1} = e_{j2}$ wynosi $\frac{k}{2^k}$. Stąd szansa na wybranie pary j_1, j_2 takiej, że $e_{j1} = e_{j2}, 1 \leq j_1, j_2 \leq R(k)$ nie przekracza $R(k)^2 * \frac{k}{2^k} = \frac{r(k)^2 k}{2^k}$ dla k≥k_0

<u>*Parametry bezpieczeństwa:*</u> schematu C-S

Wybieramy 2 liczby k, R oraz ustalamy $N > 0 : k^{\frac{1}{2}} < R + 1 < k - 1$
wybieramy funkcje haszującą odporna na kolizje $H:\{0,1\}^* \rightarrow \{0,1\}^R$
(wyjście h można identyfikować z liczbą ze zbioru $\{0,1,2,\ldots 2^R\text{-}1\}$. W dalszym ciągu zakładamy, że {k, R, n} są ustalone.

11.3
GENERACJA KLUCZY

1. $n \xleftarrow{n} I_{SG,2k}$

   a) $\tilde{p}, \tilde{q} \in P_{SG,k-1}$

   b) n= pq gdzie $p = 2\tilde{p}+1, q = 2\tilde{q}+1$

2. Losuje element $g \xleftarrow{u} QR_n, x \xleftarrow{u} QR_n \quad QRn \subset Z^* n$ oraz $\tilde{e} \leftarrow Gen\,\mathrm{Pr}ime(1^{k+1})$

3. $\left(n, g, x, \tilde{e}\right)$- klucz publiczny
(p, g)- klucz prywatny
UWAGA:
QR_n jest podgrupą cykliczną Z_n^*

4. Podpisywanie:

$m \in \{0,1\}^*$ losujemy $e \leftarrow Gern\,\mathrm{Pr}ime(1^{k+1})$ i $\tilde{y} \leftarrow QR_n$

Obliczamy: $\tilde{x} := (\tilde{y})^{\tilde{e}} * g^{-k(m)}$ oraz $y := (x * g^{h(\tilde{x})})^{\frac{1}{e}}$ w grupie Zn*

Podpisem pod m jest $\sigma = (e, y, \tilde{y})$

UWAGA:

Mamy $| e |= l + 1 < k + 1 =| \tilde{p} |=| \tilde{q} |$

Zatem e nie jest dzielnikiem 4 * $\tilde{p}\,\tilde{q}$

Podpis obliczamy łatwo wykorzystując algorytm Euklidesa – odrzucanie $e^{-1} (\mathrm{mod}\, \tilde{p})$ oraz $e^{-1} (\mathrm{mod}\, \tilde{q})$ i złożenie rozwiązań do $e^{-1} (\mathrm{mod}\, \tilde{p}\,\tilde{q})$

Podpisywanie jest probabilistyczne e i $\tilde{y}$ są wybierane losowo

5.Weryfikacja

Mając wiadomość m i podpis $\sigma = (e, y, \tilde{y})$
Wykonujemy:

1. Sprawdzania czy e jest liczba l+1 -bitową i czy $\tilde{e}$ nie jest dzielnikiem e
2. Obliczamy $\tilde{x} = \tilde{y}^{\tilde{e}} * g^{h(m)}$

Sprawdzamy czy $x = y^{e} * g^{-h(m)}$

UWAGA:

Podpisujący sprawdza czy $e = \tilde{e}$
Jeśli TAK to losuje nowe e

Twierdzenie

Załóżmy, że zachodzą następujące warunki:
1.Silne założenie RSA

2.Istnieje dostatecznie dużo liczb pierwszych S_G
3.Istnieje funkcja haszującą h odporna na kolizje wtedy:
Schemat C-S jest bezpieczny ze względu na atak adaptacyjny z wybraną wiadomością

12. Bezpieczeństwo schematu Cramera - Shoupa

12.1

LEMAT

Istnieje algorytm deterministyczny wielomianowy A taki, że dla dowolnego $k, n \in I_{SG,k}$ i czwórki (e, v, f, r): |e|<k-1, $u, r, v \in Zn*$ takie, że $(*) v^e = r^f$ oblicza wartość r- tego pierwiastka z r gdzie

$$r = \frac{e}{d}, d = NWD(e, f)$$

Dowód:

Równość (*) jest równoważna równości

$$(**) v^{\frac{e}{d}} = r^{\frac{f}{d}} \Leftrightarrow v^r = r^s \text{ gdzie } r = \frac{e}{d}, s = \frac{f}{d}$$

Celem jest obliczenie $r^{\frac{1}{r}} (\mathrm{mod}\, n)$ o ile istnieje. Istnienie wynika z założeń, że $\varphi(n) = \varphi(pq) = (p-1)(q-1) = 2\tilde{p} * 2\tilde{q} = 4\tilde{p}\tilde{q}$
Zatem NWD $(e, \varphi(n)) = 1$. Z tego wynika, że
NWD, $(r, \varphi(n)) = 1 \Rightarrow r^{-1} (\mathrm{mod}\, \varphi(n))$ Zatem
$r^{\frac{1}{r}} = r^{r-1(\mathrm{mod}\, \varphi(n))} (\mathrm{mod}\, n)$ (Tw.Eulera)

Teraz pokażemy, że $r^{\frac{1}{r}} (\mathrm{mod}\, n)$ można policzyć efektywnie znając tylko n (a nie $\varphi(n)$)

Problem jest następujący. Mając daną czwórkę (u, v, r, s) taką, że $v^{r} = r^{s}$ oraz gdzie NWD(r, s)=1 pokażemy jak obliczyć $v^{\frac{1}{r}}$

UWAGA:

Zauważmy, że $v^{\frac{1}{r}} = r^{\frac{1}{s}}$ wyciągając pierwiastek stopnia rs od obu stron tej równości(**).Ale $v^{\frac{r}{s}} = r$, zatem problem redukuje się do znalezienia $v^{\frac{1}{s}}$ mając dane u, oraz $v^{\frac{r}{s}}$ pokażemy, że $v^{\frac{1}{s}}$ można znlezć w postaci $v^{\alpha} * (v^{\frac{r}{s}})^{\beta}$ gdzie α i β są liczbami całkowitymi, które można efektywnie wyznaczać. To znaczy:

$v^{\frac{1}{r}} = v^{\alpha} * (v^{\frac{r}{s}})^{\beta}$ Wtedy α i β można wyznaczyć rozwiązując równanie:

$\frac{1}{s} = \alpha + (\frac{r}{s})^{\beta}$ /* s

$1 = \alpha s + \beta r$ Korzystając z Algorytmu Euklidesa.

To kończy dowód LEMATU

Dowód: twierdzenia

Fałszerz: ma do dyspozycji klucz publiczny podpisującego $g, x, \in QRn$ oraz $\tilde{e} \in Ger\,\mathrm{Pr}\,ime(1^{n+1})$

$h \subset Hl$ - rodzina funkcji laszujących odpornych na kolizje.

$$n = pq = (2\tilde{p}+1)(2\tilde{q}+1)$$

Celem fałszerza F jest podrobienie podpisu $\sigma = (e, y, \tilde{y})$

Pod wybraną wiadomością m, gdzie $y = (xg^{h(\tilde{x})})^{\frac{1}{e}}$

$\tilde{y}$ -losowe, oraz $\tilde{x} := \tilde{y}^{\tilde{e}} * g^{-h(m)}$

Formalnie będziemy identyfikować fałszerza F z pewnym algorytmem probabilistycznym wielomianowym, R, który na wejściu dostaje podpisy $\sigma_i(mi)$ pod wybranymi wiadomościami gdzie y= R(k), a na wyjściu poprawny podpis pod m z prawdopodobieństwem(nie zaniedbywanym) $> \frac{1}{q(k)}$ dla nieskończenie wielu parametrów bezpieczeństwa (k, l)

Dowód:

Jeśli istnieje fałszerz to istnieje algorytm przeciwnika A, który dla wejścia, $n \in I_{SG}$ i ,że Zn* efektywnie oblicza $\sqrt[n]{Z} (\mathrm{mod}\, n)$ bez znajomości rozkładu n. To będzie przeczyć silnym założeniom RSA.

Algorytm przeciwnika A jest następujący:

1. A szacuje losowo parametr bezpieczeństwa l oraz funkcje $n \in Hl$
2. Sprytnie generuje pozostałe parametry klucza publicznego czyli $(n, g, x, \tilde{e})$
3. Po interakcji z fałszerzem F otrzymuje sfałszowany podpis $[m, \sigma] = [m.(e, y, \tilde{y})]$ odpowiadającej kluczowi publicznemu $(n, g, x, \tilde{e})$
4. Używając sfałszowanego podpisu $[m, \sigma]$ atakując oblicza $\sqrt[n]{Z} (\mathrm{mod}\, n)$ dal pewnego r>1

13. Zakończenie dowodu bezpieczeństwa podpisu Cramera - Shoupa

Działanie fałszerza F.

Niech m_i (i <= t) są wybranymi przez F wiadomościami do podpisu oraz σ_i = (e_i, y, $\tilde{y}$) – podpisy otrzymane przy użyciu symulatora.

Niech [m, σ] – wyjście algorytmu fałszerza , gdzie $m \neq m_i$ (i <= t) oraz σ = (e_i,y, $\tilde{y}$).
Niech $\tilde{x}_i = y_i^{\tilde{e}_i} * g^{-h(mi)}$

Założenia:
Niech $e_i \nmid \tilde{e}$ i=1,2,...,t oraz e jest parametrem (niekoniecznie pierwszym).
Przeciwnik A po wybraniu losowego l i h $\in H_e$ wybiera wartości $e_i \longleftarrow$
GermPrime(1^{l+1}), (1<=i<t) oraz g: $Z^{2\tilde{e}*\prod_{i=1}^{t} e_i}$

Niech a $\xleftarrow{u}$ {1,2,...,n^2} x := g^a

A łatwo oblicza wartości σ_i = (e_i,y, $\tilde{y}$) żądane przez fałszerza F,
Mianowicie:

$$\tilde{y}_i \longleftarrow QR_n$$

$$\tilde{x}_i = y_i^{\tilde{e}_i} * g^{-h(m_i)}$$

$$y_i := \left(x * g^{f(\tilde{e})}\right)^{\frac{1}{e_i}}$$

$$x * g^{h(\tilde{x}_i)} := g^a * g^{h(\tilde{x}_i)} = (z)^{2\tilde{e}*\prod_{i=1}^{t} e_i (a+h(\tilde{x}_i))}$$

Zatem:

$$y := \left(z^{2\tilde{e}*\prod_{i=1}^{t} e_i (a+h(\tilde{x},1)}\right)^{\frac{1}{e_i}} = z^{2\tilde{e}*\prod_{j\neq i} e_j (a+h(\tilde{x}_j))}$$

Następnie F daje na wyjściu fałszowany podpis σ=(e,y, $\tilde{y}$) taki, że m$\neq$ m_i.
Zatem mamy, że $y^e = x g^{h(\tilde{x})} = z^f$ dla pewnego f gdzie
f=2$\tilde{e}$ $\prod e_j (a + h(\tilde{x}))$ oraz $\tilde{x} = \tilde{y}^{\tilde{e}} * g^{-h(m)}$

Zatem:

$$y^{\frac{e}{d}} = z^{\frac{f}{d}}$$

gdzie d=NWD(e,f)
Jesli d<e (l_j, e | f) to nie mamy LEMATU.
Potrafimy obliczyć $\sqrt[r]{z} (\mathrm{mod}\, n)$ gdzie r = $\frac{e}{d} > 1$ i schemat RSA został złamany.

Pokażemy, że jeśli obliczył poprawny podpis to z prawdopodobieństwem >=$\frac{2}{3}$ zachodzi warunek e | f. Ustalmy (e, y, $\tilde{y}$),(h, n, g, x, $\tilde{e}$) oraz wiadomość z podpisem [m, σ].

Niech a = b $\tilde{p}\tilde{q}$ +c gdzie c< $\tilde{p}\tilde{q}$ i >=0 oraz

$2^{k-2} <= \tilde{p}\tilde{q} < 2^{k-1}$

$2^{2k-2} <= n^2 <= 2^{2k}$ wtedy e | f w związku z tym istnieje liczba pierwsza nieparzysta s | f zatem:

$$f = (2\tilde{e} * \prod_{i=1}^{t} e_i)(a + h(\tilde{x})) = 0 (\mathrm{mod}\, s)$$

i dla ustalonego c dostajemy:

$$L(b) = b\tilde{p}\tilde{q} + c + h(\tilde{x}) = 0 (\mathrm{mod}\, s)$$

Zatem ilość (liczba) b spełniających tą kongruencję jest rzędu $\frac{n^2}{\tilde{p}\tilde{q}} * \frac{1}{s}$ więc stąd dostajemy, że prawdopodobieństwo takiego zdarzenia nie przekracza <= $\frac{1}{s}$ .A

więc prawdopodobieństwo zdarzenia takiego, że e | f jest <= $\frac{1}{s}$. Stąd

prawdopodobieństwo zdarzenia takiego, że e | f jest >=1-$\frac{1}{s}$>=$\frac{2}{3}$ czego należało dowieść.

Do zakończenia dowodu pozostaje pokazać, że odpowiednie rozkłady prawdopodobieństw dla oryginalnego podpisującego i symulatora są wielomianowo bliskie,

Def.

Rozkłady (p_j) i ($\tilde{p}_j$) na przestrzeni x_j gdzie $j \in J$ są wielomianowo bliskie wtedy i tylko wtedy gdy:

$$\forall p \exists K_0 \forall k >= K_0 \forall j \in J_k$$

zachodzi: $dist(p_j, \tilde{p}_j) := \frac{1}{2} \sum_{x \in x_j} | p(x) - \tilde{p}(x) | \leq \frac{I}{P(k)}$

Pokażemy teraz, że rozkłady $x = g^a \pmod n$ gdzie $a \xleftarrow{u} \{1,2,...,n^2\}$, że rozkład jest wielomianowo bliski do rozkładu jednostajnego w grupie Z_n^*.

Dowód.

Niech $a = b\tilde{p}\tilde{q} + c$ wtedy c ma rozkład jednostajny na odcinku $[1, \tilde{p}\tilde{q}]$.

Ustalone c jest realizowane z prawdopodobieństwem około $\frac{1}{\tilde{p}\tilde{q}}$

Zatem jeśli g jest generatorem grup OR_n rzędu $\tilde{p}\tilde{q}$ odpowiedni rozkład (warunek) jest wielomianowo bliski do rozkładu jednostajnego w $Z_{\tilde{p}\tilde{q}}$ a więc i Z_n^*.

14. Test pierwszości AKS

Wymyślony w 2002 roku przez Agravala, Keyala i Saxenę.
Algorytm AKS jest algorytmem deterministycznym i działa w czasie wielomianowym(!!!) od rozmiaru liczby.

Lemat 1

Jeśli n jest liczbą pierwszą nieparzystą, to
$(x-a)^n=x^n-a\,(mod\ n)$, gdzie $a \perp n$

Dowód

Dwumian Newtona dla $(x-a)^n$ ma postać:

$$\sum x^k (-a)^{n-k} \binom{n}{k}$$

Z twierdzenia Fermata wynika, że $a^n \equiv a\,(mod\ n)$. Tylko dwa wyrazy, pierwszy i ostatni są różne od $0\,(mod\ n)$. Wszystkie pozostałe są równe $0\,(mod\ n)$ bo

$$n \left| \frac{n!}{k!(n-k)!} \right. .$$

Lemat 2

Jeśli zachodzi równość, że $(x-a)^n=x^n-a\,(mod\ n)$ (gdzie n jest nieparzyste), to jest n liczbą pierwszą.

Dowód

Niech q^k będzie najwyższą potęgą liczby pierwszej q, która dzieli n.
Jeśli: $n=q^k l$, gdzie $k \geq 1 \; i \; l \perp q$,

to wtedy:

$$\binom{n}{q}=\binom{q^k l}{q!(q^k l-q)!}=\frac{(q^k l)(q^k l-1)\cdot\ldots\cdot(q^k l-q-1)}{q!}=q^{k-1}l\,.$$

tylko ten czynnik dzieli się przez q

ten i następne już nie bo q nie dzieli $q-m$

$$\left.\begin{array}{l} q \mid q^k l-(q-m) \\ q \mid q^k l \end{array}\right\} \Rightarrow q \mid (q-m)$$

sprzeczność!!!

$$(x-a)^n-(x^n-a) \;=\sum_{i=1}^{n-1}\binom{n}{i}(-a)^{n-i}x^i$$

Weźmy teraz $i=q$. Wtedy $\binom{n}{i}$ nie jest wielokrotnością q^k , więc także nie jest wielokrotnością n. Zatem różnica $(x-a)^n-(x^n-a)\neq 0\,(mod\ n)$. Stąd n jest liczbą pierwszą (na mocy dowodu nie wprost).

Twierdzenie AKS

Załóżmy że $n\in N$, q r są liczbami pierwszymi, gdzie $q|r-1$ oraz spełnione są cztery poniższe założenia:

1) $n^{\frac{r-1}{q}}\ (mod\ r)\notin\{0,1\}$
2) $\boldsymbol{S}$ - skończony zbiór liczb całkowitych taki, że
   $\forall\, b\neq b\in S$ zachodzi: $b-b'\perp n$
3) $\binom{q+|S|-1}{|S|}>n^{2\sqrt{(r)}}$
4) $\forall\, b\in S \quad (x+b)^n=x^n+b\,(mod\ x^r-1,n)$ tzn.
   $(x+b)^n-x^n+b=f(x)(x^r-1)+ng(x)$ Gdzie, $f\,,g\in Z[x]$

Wtedy n jest potęgą liczby pierwszej.

Przykład

$n\,g(x)=0\,(mod\ n)$

$x^{2r}-1=0\,(mod\ x^r-1,n)$, gdyż $x^{r-1}|x^{2r}-1=(x^r+1)(x^r-1)$

Szkic dowodu twierdzenia AKS.

I. Warunki 1, 2, 3 można traktować jako koszt spełnienia założenia z

lematu 2.

II. Jeśli zachodzi $(x+b)^n=(x^n-b)(mod\ x^r-1,n)$ to zachodzi to także dla liczb m w postaci $m=n^i p^j$, gdzie p jest dzielnikiem pierwszym liczby n, tzn. $(x+b)^m=x^m+b(mod\ x^r-1,n)$ (jest to propagacja równości z warunku 4 na liczby postaci).

a) Bierzemy $m=n^i$. Otrzymamy wtedy $(x+b)^{n^i}=x^{n^i}+b(mod\ x^r-1,n)$

Dla dowodu zauważmy, że:

$$(x+b)^{n^i}=[(x+b)^n]^{n-1}=(x^n+b)^{n^{i-1}}=[(x^n+b)^n]^{n^{i-2}}=(x^{n^2}+b)^{n^{i-2}}=...=$$
$$=(x^{n^i}+b)^0=x^{n^i}+b$$

b) Bierzemy $n=p^j$ tzn. $(x+b)^{p^j}=x^{p^j}+b(mod\ x^r-1,n)$ Zauważmy teraz, że

$(x+b)^n=(x^n+b)(\mathbf{mod}\,x^{r-1},n)$, **więc także** $(x+b)^n=(x^n+b)(\mathbf{mod}\,x^{r-1},p)$, **gdyż** $p\mid n$.

Mamy $(x+b)^P=x^P+b^P(mod\ p)=x^P+b(mod\ p)$

Pozostałe współczynniki znikną, gdyż są podzielne przez p.

Na mocy twierdzenia Fermata mamy:
Jeśli $b\perp p$, to $b^{p-1}=1(mod\ p)\Rightarrow b^P=b(mod\ p)$
Jeśli $p|b$, to $b^P=b(mod\ p)$, gdyż $p|b^P-b$
Zatem:

$$(x+b)^{p^j} = ((x+b)^p)^{p^{j-1}} = (x^p+b)^{p^{j-1}} = [(x^p+b)^p]^{p^{j-2}} = (x^{p^2}+b)^{p^{j-2}} = \ldots =$$
$$= x^{p^j} + b(\mathrm{mod}\, p).$$

Udowodniliśmy więc, że zachodzi $(x+b)^{p^j} = x^{p^j} + b\,(mod\ x^r-1, n)$

Jest to o słabszy warunek niż $(\mathrm{mod}\, n)$, więc jeśli równanie jest prawdziwe $(\mathrm{mod}\, n)$, to jest też prawdziwe dla $(\mathrm{mod}\, x^r-1, n)$.

Połączenie kroków a) i b).

$$\begin{cases} (x+b)^{n^i} = x^{n^i} + b(\mathrm{mod}\, x^r-1, n) \\ (x+b)^{p^j} = x^{p^j} + b(\mathrm{mod}\, x^r-1, n) \end{cases} \qquad m = n^i p^j$$

$$(x+b)^{n^i p^j} = [(x+b)^{n^i}]^{p^j} = [x^{n^i}]^{p^j} + b = x^{n^i p^j} + b\,(mod\ x^r-1, n)$$

krok a) krok b)

III. Załóżmy, że $b<a$. Jeżeli $a=b\,(modr)$, to $x^a = x^b\,(mod\, x^r-1, n)$.

Dowód:

$x^a - x^b = x^b(x^{a-b}-1)$. Pokażemy, że $x^b(x^{a-b}-1)$ jest wielokrotnością x^r-1.

Połóżmy $a-b=lr$. Wtedy

$$x^{lr}-1 = (x^r)^l - 1 = y^l - 1 = (y-1)(y+y^2+\ldots+y^{l-1}) =$$
$$= (x^r-1)(1+x^r+x^{2r}+\ldots+x^{r(l-1)}). \qquad \text{C.N.D.}$$

IV. Zasada szufladkowa Dirichleta.

Jeśli $0 \le i, j \le |\sqrt{(r)}|$, to wtedy istnieją pary $(i,j) \neq (i',j')$ takie, że $n^i p^j = n^{i'} p^{j'} (mod\, r)$

(Są to liczby dające tą samą resztę. Dzieje się tak dlatego, gdyż możliwości wyboru par (ij) jest o jeden więcej niż istnieje reszt.)

V. Ekstrapolacja kroku II.

Krok II można zapisać

że dla $q_1(x) = x+b_1$ mamy równość $q_1(x)^m = q_1(x^m)(mod\, x^r-1, n)$,

a dla $q_2(x)x+b_2$ mamy $q_2(x)^m = q_2(x^m)(mod\, x^r-1, n)$.

Wtedy równość $q(x^m) = q(x)^m (mod\, x^r-1, n)$ zachodzi dla $q = q_1 q_2$

Dowód.

$$(*)\begin{cases} q_1(x^m) = q_1(x)^m \bmod(x^r-1,n) \\ q_2(x^m) = q_2(x)^m \bmod(x^r-1,n) \end{cases}$$

$$q_1q_2(x^m) \stackrel{?}{=} q_1(x)^m q_2(x)^m$$

Ale lewa strona powyższego równania jest równa $q_1(x^m)q_2(x^m)=q_1(x)^m q_2(x)^m$ ma mocy (*). C.K.D.

VI. Konstrukcja elementu dużego rzedu w $Z[x]/x^r-1$.

Rozważmy grupę *G* generowaną przez dwumiany $x+b$, gdzie $b\in S$, w ciele $Z_P[x]/h(x)$, gdzie $b(x)$ jest wielomianem nierozkładalnym w $Z_P[x]$ dzielącym $\frac{x^r-1}{x-1}$

Zatem:

$$|G| \geq \left| \{ \prod_{b\in S} (x+b)^{a_b}, \sum_{b\in S} \alpha_b \leq q-1, \alpha_b \geq 0 \} \right|$$

Grupa multiplikatywna ciała $Z_P[x]/h(x)$ jest cykliczna, zatem grupa *G* też jest cykliczna.

Jeśli *g* jest jej generatorem, to rząd $g=|G| \geq \left(\frac{q+|S|-1}{|S|} \right)$

VII. Zakończenie dowodu.

Niech *g* - generator grupy *G*, oraz m, m' będą elementami zbioru $\{n^i p^j : 0 \leq i, j \leq |\sqrt{(n)}|\}$ takimi, że $m \equiv m'(mod\, r)$ wtedy prawdziwe jest:

$(*)$ $q(x^m)=q(x^{m'})(mod\, x^r-1,n)$

Jeśli q_1, q_2 spełniają (*) ⇒ $q_1 * q_2$ też spełniają (*)

Zatem równanie (*) zachodzi dla generatora *g* w grupie *G*, tj.

$q(x^m)=p(x)^m mod\,(x^r-1,n)$

Ponieważ $m=m'(mod\, r)$ więc $(x+b)^m=(x+b)^{m'}(mod\, x^r-1)$ i na mocy multiplikatywności $(\prod_{b\in S}(x+b)^{a_b})^m=(\prod_{b\in S}(x+b)^{a_b})^{m'}(mod\, x^r-1,n)$

Czyli $q(x)^m=q(x)^{m'}$ w grupie G. Wtedy mamy

$$|G| \leq |m-m'| \leq n^{2\lfloor\sqrt{n}\rfloor} < \left| \begin{matrix} q+|S|-1 \\ |S| \end{matrix} \right| \leq |G|$$

Otrzymaliśmy sprzeczność z wnioskiem płynącym z zasady szufladkowej Dirichleta, czyli sprzeczność z $m \neq m'$. Sprzeczne więc jest $n^i p^j \neq n^{i'} p^{j'}$,

$$n^i p^j = n^{i'} p^{j'}$$
$$n^{i-i'} = n^{j-j'} \Rightarrow n = p^{\alpha}$$

C.N.D

Algorytm AKS:

1) Znajdź liczbę pierwszą r rzędu $\mathcal{L}\hat{}c(n)$. Znajdź q – największy dzielnik pierwszy liczby $r-1$ taki, że $n^{\frac{r-1}{q}} (mod\, r) \notin \{0,1\}$
2) Znajdź , takie że $\binom{q+s-1}{s} \geq n^{2\lfloor\sqrt{(r)}\rfloor}$
3) Sprawdź, czy n ma dzielnik pierwszy w zbiorze {2, 3, ..., s}
4) Jeśli nie, to sprawdź następującą kongruencję:
   $(x+b)^n = x^n + b\,(mod\ x^r - 1, n)$
5) Sprawdź jaką potęgą liczby pierwszej jest n

Zadania

Test Lehmana
n – nieparzyste > 1
l – świadków -> prawdopodobieństwo błędu < 0.5

$a_1, a_2, ..., a_l$ - losowe elementy $\in \{2,3,...,n-2\}$
$NWD(a_l, n) = 1$

Przebieg algorytmu

1. Obliczamy $a_i \frac{n-2}{2} (mod\, n) = r \quad i = \{1,2,...,l\}$
2. Jeśli $\{r_1, r_2, ..., r_l\} = \{-1\ (mod\, n), 1\ (mod\, n)\}$ to odpowiedź testu : TAK, wpw NIE

Zadanie 1.

n =13
l =2
$a_1 = 2$, $a_2 = 3$
$r_1 = 2^6 = -1\ (mod\, 13)$
$r_2 = 3^6 = 1\ (mod\, 13)$
$\{r_1, r_2\} = \{1, -1\}$
Odpowiedz TAK.

Zadanie 2.

n =15

l =3

$a_1 = 2$, $a_2 = 3$, $a_3 = 4$

$r_1 = 2^7 = 8 \pmod{15}$

$r_2 = 3^7 = 12 \pmod{15}$

$r_1 \neq -1$ wiec można dalej nie przeprowadzać testu, gdyż liczba nie będzie pierwsza

Test Miller`a – Rabin`a

l – świadków -> prawdopodobieństwo błędu $\leqslant \frac{1}{4^l}$

1. Test Fermata $a^{n-1} = 1 \pmod{n}$
    1. Jeżeli "NIE" to koniec testu
    2. Jeżeli "TAK" to przedstawiamy $n-1 = 2^R \cdot s$, gdzie s – nieparzyste
        1. Jeżeli "NIE" to koniec testu
        2. Jeżeli "TAK" to odpowiedź testu jest pozytywna albo
            $a^s = 1 \pmod{n}$ lub $0 \leqslant r \leqslant R-1$ *takie że* $a^{2^r \cdot s} = -1 \pmod{n}$
        3. Wpw odpowiedź testu jest "NIE"

Zadanie 3.

n=13

$13-1=12=2^2\cdot 3$

s = 3

R = 2

a = 3

1. $3^{12}=1 \pmod{13}$
2. $3^3=1 \pmod{13}$ $\quad r=0 \rightarrow 3^{2^0\cdot 3}=1 \pmod{13}$

Odpowiedź “TAK”

Zadanie 4.

n=15

$15-1=14=2^1\cdot 7$

s = 7

R = 1

a = 4

1. $4^{14}=1 \pmod{15}$
2. $4^7=4 \pmod{15}$ $\quad 4^{2^0\cdot 7}=4 \pmod{13}$

Odpowiedź “NIE”

Zadanie 5.

n=17

$17-1=16=2^4\cdot 1$

s = 1

R = 4

$a_1=3$, $a_2=4$

Dla $a_1=3$

1. $3^{16}=1 \pmod{17}$
2. $3^1=3 \pmod{17}\neq 1$

$3^{2^0\cdot 1}=3 \pmod{17}\neq -1$

$3^{2^1\cdot 1}=9 \pmod{17}\neq -1$

$3^{2^2\cdot 1}=-4 \pmod{17}\neq -1$

$3^{2^3\cdot 1}=16 \pmod{17}\neq -1$

Odpowiedź “TAK”

Dla $a_2=4$

1. $4^{16}=1 \pmod{17}$
2. $4^{2^3\cdot 1}=1 \pmod{17}$

$4^{2^2\cdot 1}=1 \pmod{17}$

$4^{2^1\cdot 1}=-1 \pmod{17}$

Odpowiedź “TAK”

15. Struktury niesymetryczne, Diffie – Hellman

Definicja

Systemem kryptograficznym niesymetrycznym nazywamy piątkę:

(P, C, K, Φ, ψ) gdzie

$\phi : P \times K \to C$

$\psi : C \times K \to P$

$\forall_{k \in K} \quad \exists_{k' \in K} \phi(\cdot, k) \circ \psi(\cdot, k') = Id$

ϕ – zadaję regułę szyfrowania (z kluczem k)

ψ – zadaje regułę deszyfrowania (z kluczem k')

Komentarz

Dotychczas rozważane systemy były symetryczne gdyz klucz deszyfrujący k' łatwo uzyskuje się z klucza szyfrującego k.

Teraz (P, C, K, Φ, ψ) - niesymetryczny tzn. K' nie można obliczyć mając k. Przydzielając odbiorcy klucz $k' = k'(k)$, każdy użytkownik posiadający klucz k (zwany publicznym) może szyfrować za jego pomocą wiadomość m, a mianowicie obliczając wartość $\phi_k(m) = \phi(m, k)$. Tylko odbiorca posiadający klucz k' może weryfikować kryptogram $c = \phi_k(m)$ obliczając $\psi_{k'}(c) = \psi_{k'}(\phi_k(m)) = m$.

Diffie i Hellman jako pierwsi pokazali jak wykorzystać jednokierunkowość funkcji $k = g^k (mod\ p)$ w kryptografii niesymetrycznej.

Ogólnie niech G będzie grupą cykliczną, skończoną i niech $f : G \to G$ będzie funkcją jednokierunkową spełniającą warunek (1)

$k_{AB}(f(x), y) = k_{AB}(f(y), x)$.

To oznacza, że jeśli stronie (podmiotowi) A przyporządkujemy klucz $(x_A, f(x_A))$, a stronie B klucz $(x_B, f(x_B))$ oraz wartość $f(x_A)$ i $f(x_B)$ podamy do wiadomości publicznej to klucz $k_{AB} = f(x_A)^{x_B} = f(x_B)^{x}_{A}$ będzie wspólnym kluczem wymiany Diffie-Hellamana.

Jeśli przyjąć, że $k_{AB}(t, u) = t^u$, to równanie (1) można wyrazić następująco:

$y \log f(x) = x \log f(x)$

$$\frac{x}{y}=\frac{\log f(x)}{\log f(y)}$$

a to jak wiadomo łatwo jest spełnione dla funkcji wykładniczej $f(x)=g^x$.

Wymiana klucza D-H rozwiązuje klasyczny problem KURIERA.

D-H

1. $A \quad \overset{g^{x_A}}{\rightarrow} \quad B$
2. $B \quad \overset{g^{x_B}}{\rightarrow} \quad A$
3. A oblicza $k_{AB}=(g_B^x)_A^x$, B oblicza $k_{BA}=(g_A^x)_B^x$
4. Wspólne dla A i B to $k_{AB}=k_{BA}=g^{x_A x_B}$

Założenie D-H (hipoteza)

Obliczenie k_{AB} tylko na podstawie wartości g^{x_A}, g^{x_B} jest tak trudne jak problem logarytmu dyskretnego w grupie G.

Szyfrowanie El Gamala

Jeśli potrafimy bezpiecznie uzgodnić klucz to w dalszym ciągu możemy go wykorzystać do szyfrowania. Ta idea prowadzi do kryptogramu ElGamala.

Niech G - grupa (skończona i cykliczna) $r, x \in G$, g – generator G

$\overset{A}{(r, g^r)}$ $\overset{B}{(x, g^x)}$

1) B przekazuje A swój klucz publiczny $X = g^x$.
2) A losuje r i oblicza klucz wymiany $k_{AB} = X^r$ a następnie tworzy szyfrogram wiadomości m w postaci $\Phi_{(X,r)}(m) = (R, mX^r)$ gdzie $R = g^r$.
3) B na podstawie R oblicza klucz na wymiany $X^r = R^x$ i dzieląc drugą współrzędną kryptogramu przez X^r otrzymuje wartość m.

Rivest, Shemir, Adelman stworzyli alternatywny system kryptograficzny (P, C, K, Φ, Ψ).

Podobnie $k, k' \in Z_{\lambda(n)}$,
gdzie:
$\lambda(n) = NWW\,\phi(P^{\alpha_p})$
Klucze: k - klucz publiczny

k' – klucz prywatny
spełnia warunki $k, k' \perp \varphi(n), k \cdot k' = 1 (\mathrm{mod}\, \lambda(n))$.

Np.
$\lambda(12) = NWW(\varphi(2^2), \varphi(3)) = NWW(2,2) = 2$

Trudność obliczeniowa k' na podstawie k wynika stąd, że funkcja $n \to \varphi(n)$ jest trudno obliczalna, jeśli nie znamy rozkładu n na czynniki pierwsze.

Szyfrowanie :
$\Phi_k(m) = m^k (\mathrm{mod}\, n) = c$
Deszyfrowanie :
$\Phi_{k'}(c) = c^{k'} (\mathrm{mod}\, n)$
gdzie k i k' spełniają $k, k' \perp \varphi(n), k \cdot k' = 1 (\mathrm{mod}\, \lambda(n))$

<u>Podpisy cyfrowe</u>

Niech $G = Z_p^*$
Podpis El Gamala :
$\overset{A}{(x, X)}$ $\overset{B}{}$
(r, R)
$\Phi_{X,r}(n) = \sigma(m) = (R, s)$
gdzie:

$s = (m + xR)r^{-1} (\mathrm{mod}\, p^{-1})$

Weryfikacja podpisu:

$\sigma_{x,r}(m)$ - przy pomocy klucza publicznego X jest następująca :

1)obliczenie klucza wymiany

$k = X^r = R^x$

Mamy:

$X^r = R^x \Leftrightarrow X^R = \left(x^r\right)^{\frac{R}{r}} = \left(R^x\right)^{\frac{R}{r}}$

i obliczamy:

$R^s X^{-R} = R^{\frac{m+R+r}{r}} \cdot R^{-x\frac{R}{r}} = R^{\frac{m}{r}} = g^m$

Dla RSA podpis pod wiadomością m jest postaci

$\sigma_{k'}(m) = \Psi_{k'}(m) = m^k (\mathrm{mod}\, n) = c$

Weryfikacja to podniesienie $\sigma_{k'}(m)$ do potęgo będącej kluczem publicznym strony A i sprawdzenie, że w wyniku otrzymujemy wiadomość m gdyż:

$\Phi_k(\Psi_{k'}(m)) = \left(m^{k'}\right)^k = m^{k'k} = m (\mathrm{mod}\, n)$, ponieważ $kk' = 1 \, mod \, \lambda(n)$