WZK - Egzamin

Dariusz Max Adamski
Nr indeksu 136674

1. Wykazać, że założenie bezkwadratowości liczby N (gdy p=q) w RSA jest istotne. Przeanalizuj przypadek kiedy p, q =5, e=3 i m_1=2, m_2=3, m_3=5.

Gdy p = q, po zaszyfrowaniu i odszyfrowaniu wiadomości nie otrzymamy oryginalnej wiadomości!

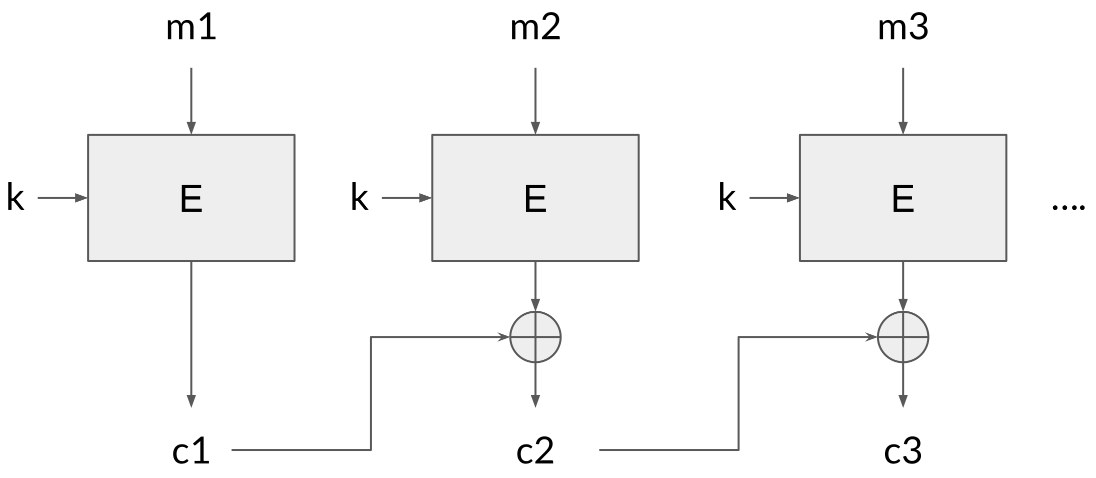
n = p*q = 25; phi = (p-1)(q-1) = 16; d = pow(e, -1, phi) = pow(3, -1, 16) = 11
C1 = m1^e mod n = 2^3 mod 25 = 8
C2 = 3^3 mod 25 = 2
C3 = 5^3 mod 25 = 0

D1 = c1^d mod n = 8^11 mod 25 = 17 != m1
D2 = 2^11 mod 25 = 23 != m2
D3 = 0^11 mod 25 = 0 != m3

Pow(n, -1, mod) - modular inverse

2. W trybie ECB szyfrowanie wykonuje się niezależnie dla każdego komunikatu m_i, dla i=1, 2, … Sekwencja kryptogramów jest przedmiotem wielu ataków wykorzystujących brak powiązań pomiędzy kolejnymi kryptogramami. Rozważyć schemat z powiązaniami w którym $c_i=E_k(m_i)\oplus c_{i-1}$ dla i=1, 2, .. Czy ten schemat jest lepszy niż tryb ECB? Odpowiedź uzasadnić.

Rozważany schemat nie jest lepszy (szybkość, bezpieczeństwo) niż tryb ECB.
Schemat jest wolniejszy, bo wykorzystuje obliczenia sekwencyjne - trzeba XORować poprzedni szyfrogram, żeby otrzymać następny.
Bezpieczeństwo też nie jest lepsze - powinniśmy przepuszać c_i XOR m_{i+1} przez E. Aktualnie atakujący bez problemu może robić XOR c_i i wyjścia z E_{i+1}, sprowadzając ten schemat do ECB.



3. Alicja chce przesłać super tajną duuużą wiadomość M do Bolka, zaproponuj protokół przesłania tej wiadomości, przez kanał narażony na podsłuch, tak aby oboje byli pewni, że zostały spełnione kryteria poufności, integralności i uwierzytelnienia.

Alicja i Bolek ustalają klucz sesji algorytmem Diffiego-Hellmana.

Alicja szyfruje wiadomość algorytmem AES. Jako tryb wybiera CBC, ponieważ zapewnia większe bezpieczeństwo niż ECB, CTR i OFB i jest zdecydowanie szybszy od CFB

Aby zagwarantować integralność Alicja oblicza wartość funkcji skrótu MAC dla swojej wiadomości i wysyła ją Bolkowi. Bolek po odszyfrowaniu weryfikuje zgodność odszyfrowanej wiadomości z hashem.

4. Co to oznacza, ze funkcja skrótu jest jednokierunkowa i jakie to ma znaczenie, po co nam w ogóle taka funkcja? Jakie cechy powinna posiadać dobra funkcja skrótu?

Wartość funkcji jednokierunkowej jest łatwo (w czasie wielomianowym) obliczyć z argumentów, ale obliczenie argumentu z wartości jest trudne (nie ma algorytmu na obliczenie w czasie wielomianowym). Oznacza to, że jednokierunkowa f-a skrótu jest praktycznie nieodwracalna.

Nieodwracalne f-e skrótu są bardzo przydatne np. do weryfikacji, że zawartość wiadomości/pliku jest taka sama - ściągamy instalator ze strony; na stronie jest podany hash; po ściągnięciu .exe weryfikujemy, że jego hash zgadza się z hashem ze strony; jesteśmy pewni, że zawartość pliku jest prawidłowa. Inny przykład - zamiast przechowywać hasła w bazie danych aplikacji składujemy ich solone hashe; aplikacja może łatwo porównać hashe przy próbach logowania z tymi w bazie, ale jak ktoś niepowołany uzyska dostęp do bazy, to nie odczyta haseł użytkowników.

Dobra funkcja skrótu powinna mieć własności kompresji (kompresuje dowolny ciąg do skrótu o określonej długości), łatwości obliczeń (mając x, łatwo obliczyć h(x)), jednokierunkowości i odporności na kolizje (bardzo trudno znaleźć dwa argumenty dające te same wartości)

5. Załóżmy, ze chcemy stworzyć system RSA z modułem N=p1*p2*p3 (wszystkie p1, p2, p3 są liczbami pierwszymi). Czy jest to możliwe? Jeśli tak, to jaka jest różnica pomiędzy taką modyfikacją, a oryginalnym systemem RSA? Wyprowadź wyrażenia potrzebne do szyfrowania, deszyfrowania i na klucze.

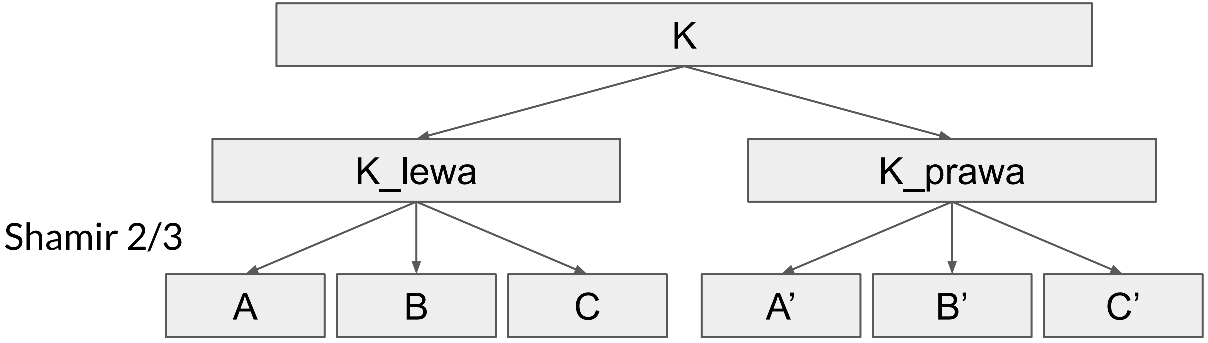
Jest to możliwe. Praktycznie różnica jest tylko w obliczaniu modułu i phi.

n = p1*p2*p3
phi = (p1-1)*(p2-1)*(p3-1)
e: wygenerowana liczba względnie pierwsza z phi
d: wygenerowana liczba gdzie e*d przystaje do 1 modulo phi
(e,n) - klucz publiczny; (d, n) - klucz prywatny
Szyfrowanie wiadomości m:
c = m^e mod n
Deszyfrowanie wiadomości c:
m = c^d mod n

6. Jak podzielić klucz na części A, B, C, A’, B’, C’ tak aby rekonstrukcja klucza możliwa była na podstawie co najmniej 2 z części A, B, C oraz co najmniej dwóch części A’, B’, C’.

Dzielimy klucz K w połowie (lewa i prawa strona).

Każdą połówkę dzielimy na 3 części, korzystając z podziału sekretu Shamira z wymaganą liczbą 2 udziałów.



7. BBS – niech p=7, q=11. Wygenerować ciąg bitów dla losowo wybranego ziarna. Jaki jest okres takiego ciągu? (Przeanalizuj, czy zależy on od wybranego ziarna, czy ziarno powinno spełniać jakieś warunki?)

p = 7, q = 11, n = p*q = 77

Warunki ziarna: x0 != 0, x0 != 1, gcd(n, x0) == 1

Okres ciągu zależy od ziarna (okres jako liczba iteracji do powtórzenia x1):

Dla x0 = 17 okres = 4:
x1 = 17^2 mod 77 = 58 -> 0
x2 = 58^2 mod 77 = 53 -> 1
x3 = 53^2 mod 77 = 37 -> 1
x4 = 37^2 mod 77 = 60 -> 0
x5 = 60^2 mod 77 = 58 -> 0
x6 = 58^2 mod 77 = 53 -> 1
x7 = 53^2 mod 77 = 37 -> 1
x8 = 37^2 mod 77 = 60 -> 0

Dla x0 = 12 okres = 2:
x1 = 12^2 mod 77 = 67 -> 1
x2 = 67^2 mod 77 = 23 -> 1
x3 = 23^2 mod 77 = 67 -> 1
x4 = 67^2 mod 77 = 23 -> 1
x5 = 23^2 mod 77 = 67 -> 1
x6 = 67^2 mod 77 = 23 -> 1
x7 = 23^2 mod 77 = 67 -> 1
x8 = 67^2 mod 77 = 23 -> 1