Zajęcia 8: Szyfr RSA

Zasada RSA
Jest to szyfr asymetryczny, czyli inny klucz jest używany do szyfrowania a inny do deszyfrowania. Algorytm wygląda tak:

- wybrać dwie liczby pierwsze: p i q
- policzyć n = p*q
- policzyć φ(n) = (p-1)(q-1)
- wybrać e, takie że 1<e<φ(n), oraz NWD(e,φ(n))=1 – o ten drugi warunek nie musimy się martwić, bo wyjdzie nam automatycznie w następnym kroku
- znaleźć d, takie że (d e) mod φ(n) = 1 – to jest właśnie odwracanie modulo które robiliście Państwo na poprzednich zajęciach. Jeżeli warunek NWD(e,φ(n))=1 nie jest spełniony to funkcja zwróci nam „-1", bo równanie nie ma rozwiązań i musimy powtórzyć ten punkt z innym „e"
- szyfrowanie „m_i": $c_i = m_i^e \bmod n$ – to jest operacja potęgowania modulo z zajęć na których mieliśmy protokół Diffiego-Hellmana
- deszyfrowanie: $m_i = c_i^e \bmod n$ – znów potęgowanie modulo, ponieważ wszystkie operacje są wykonywane mod n to największą liczbę jaką możemy zaszyfrować to n-1

klucz publiczny, który jest używany do szyfrowania składa się z e i n,
klucz prywatny (do deszyfrowania) składa się z d i n
Złamanie szyfru wymaga faktoryzacji n na p i q, co jest operacją złożoną obliczeniowo.

Prosty przykład:
p=3, q=11, n = 33, φ(n) = 20
wybieramy e=7 i wyznaczamy d=3; (jeżeli wybralibyśmy np. e=6 to równanie nie miałoby rozwiązań bo NWD(6,20)=2 != 1 i musielibyśmy losować ponownie)

chcemy zakodować liczbę m=2
$c=2^7 \bmod 33 = 128 \bmod 33 = 29$ – liczba po zakodowaniu
dekodowanie $m = 29^3 \bmod 33 = 24389 \bmod 33 = 2$ - dostaliśmy tę samą liczbę 2 co przed kodowaniem

Dzielenie na bloki
RSA się zwykle implementuje dzieląc plik na bloki danych o stałej długości i szyfrując każdy blok. Typowo te bloki są 1024-2048 bitowe, my zrobimy coś podobnego ale dla małych bloków (kilkanaście - kilkadziesiąt bitów). Rozmiar bloku można policzyć na podstawie n, bo operacje szyfrowania i deszyfrowania liczone są modulo n.
Jeżeli $n<2^8$ to szyfrując nawet pojedynczy kod ASCII możemy wyjść poza zakres (np. jeżeli n=100, a zaszyfrujemy m = 125 to po deszyfrowaniu zamiast 125 dostaniemy 125 mod 100 = 25).
Jeżeli $2^8<=n<2^{16}$ to możemy zaszyfrować jeden znak naraz.
Jeżeli $2^{16}<=n<2^{24}$ to możemy zaszyfrować bloki dwu-literowe.
I tak dalej dla bloków 3, 4, … znakowych.

Uwaga, proszę pamiętać, że operacje potęgowania modulo (i podobnie tez odwracanie modulo) będą wymagały przechowywania w pamięci liczb rzedu n^2, więc trzeba się upewnić, że program używa wystarczająco pojemnych typów dla liczb całkowitych.

Chcąc połączyć kilka liter w blok (który potem jest szyfrowany) najlepiej zrobić to poprzez przesunięcie bitowe.
Np. jeżeli nasz tekst to 'abc' i dzielimy go na bloki dwu literowe 'ab', 'c\0' (brakujący znak

wypełniamy kodem ASCII o wartości 0). Wtedy dla pierwszego bloku 'a' = 97, 'b' =98 (kody ASCII), a cały blok to (97 << 8) + 98 = 24930, a drugi blok = (99<<8) + 0 = 25344

Chcąc odzyskać litery robimy dla pierwszego bloku:
$(24930/2^8) \bmod 2^8 = 97$='a' (dzielenie bez reszty)
$24930 \bmod 2^8 = 98$ ='b'

Jeżeli mamy np. 3 znaki w bloku to łączenie w blok będzie wyglądało:
m=(pierwszy <<16) + (drugi <<8) + trzeci
a ponowne rozdzielanie:
pierwszy = $(m / 2^{16}) \bmod 2^8$
drugi = $(m/2^8) \bmod 28$
…

Zadanie dla państwa:
- zaimplementować szyfr RSA szyfrujący kilkuliterowe bloki tekstu, wynik wyjścia programu ma wyglądać tak:
p = ? // użytkownik podaje p
q = ? // użytkownik podaje q, program sprawdza czy p i q są pierwsze
n = … // liczy n
fi(n) = … // liczy φ(n)
e = … // e jest podawane z klawiatury albo losowane
d = … // wyliczone d
tekst= ? // użytkownik podaje tekst do zaszyfrowania z klawiatury
bloki = … … … // program wypisuje liczby (oddzielone spacją) odpowiadające kodom ASCII tekstu połączonego w bloki
szyfr = … … … // program szyfruje po kolei bloki i wypisuje zaszyfrowane liczby (oddzielone spacjami)
odszyfrowane_bloki // operacja odwrotna deszyfrujemy RSA szyfr, żeby dostać z powrotem bloki
odszyfrowany_tekst = … // dzielimy z powrotem odszyfowane_bloki na kody ascii i wyświetlamy odszyfrowany tekst

Jako tekst do zaszyfrowania proszę wpisać swoję Imie_Nazwisko (bez polskich „ogonków"), szyfrowanie ma się odbywać w blokach 3-znakowych.

Proszę wysłać mi kod oraz wynik wyjścia programu (na którym widać wszystkie liczby wypisane wyżej)