Algorytm Rivesta-Shamira-Adlemana (RSA) –najpopularniejszy asymetryczny algorytm kryptograficzny z kluczem publicznym. Bezpieczeństwo szyfrowania opiera się na trudności faktoryzacji dużych liczb złożonych.

**Jak zgenerować kluczy?**

W celu wygenerowania pary kluczy (prywatnego i publicznego) należy posłużyć się algorytmem:

* Wybieramy losowo dwie duże liczby pierwsze p i q.
  + W celu bezpieczeństwa całe liczby p i q muszą być wybrane losowo i muszą być zbliżone długością w bitach, ale jednocześnie muszą różnić się długością znaczenia o kilku cyfr, żeby było trudno robić faktoryzację tych liczb.
* Obliczamy wartość n = p \* q.
* Obliczamy wartość funkcji Eulera dla n:
* Wybieramy liczbę e (1 < e < φ(n)) względnie pierwszą z φ(n).
* Znajdujemy liczbę d, gdzie jej różnica z odwrotnością modularną liczby e jest podzielna przez φ(n):

Klucz publiczny jest definiowany jako para liczb (n, e), natomiast kluczem prywatnym jest para (n, d).

### Szyfrowanie

Zanim zaszyfrujemy wiadomość, dzielimy ją na bloki {\displaystyle m}o wartości liczbowej nie większej niż *n*, a następnie każdy z bloków szyfrujemy według poniższego wzoru:

### Deszyfrowanie

Zaszyfrowana wiadomość będzie się składać z kolejnych bloków *c.* Tak stworzony szyfrogram przekształcamy na tekst jawny, odszyfrowując kolejne blok *c* według wzoru:

**Weryfikacja I podpisywanie**

Analogicznie, RSA może zostać użyte do przeprowadzenia operacji podpisu. W takim przypadku szyfruje się zazwyczaj skrót wiadomości za pomocą klucza prywatnego i propaguje taki szyfrogram wraz z oryginalną wiadomością. Odbiorca posiadający klucz publiczny deszyfruje otrzymaną z wiadomością, zaszyfrowaną wartość funkcji skrótu, następnie oblicza wartość tejże funkcji z otrzymanej wiadomości. Jeśli obie wartości się zgadzają, to przyjmuje się, że wiadomość została podpisana poprawnie.

**Własności operacji szyfrowania i deszyfrowania**

Niech CK1, DK1, CK2, DK2 będą kolejno szyfrowaniem i deszyfrowaniem kluczami K1 i K2. Wtedy zachodzi:

* CK1 (CK2(M)) = CK2 (CK1(M)) – przemienność operacji szyfrowania,
* DK1 (DK2(M)) = DK2 (DK1(M)) – przemienność operacji deszyfrowania.

Ze względów bezpieczeństwa nie powinno się stosować więcej niż 2 zagnieżdżone szyfrowania ze względu na ataki oparte na chińskim twierdzeniu o resztach.

**Przykład użycia:**

1. Niech p = 61 I q = 53

* Liczymy n = p \* q, n = 61 \* 53 = 3233
* Liczymy λ(n) = NWW(p -1, q – 1) = λ(3233) = NWW(60, 52) = 780
  + \*NWW - Najmniejsza wspólna wielokrotność
* Wybieramy losową liczbę 1 < e < 780. Niech e = 17.
* Liczymy d = e (mod (λ(n)))

d \* e = 1 mod λ(n)

413 \* 17 = 1 mod 780

Klucz publiczny jest (n = 3233, e = 17). Dla szyfrowania funkcja jest taka:

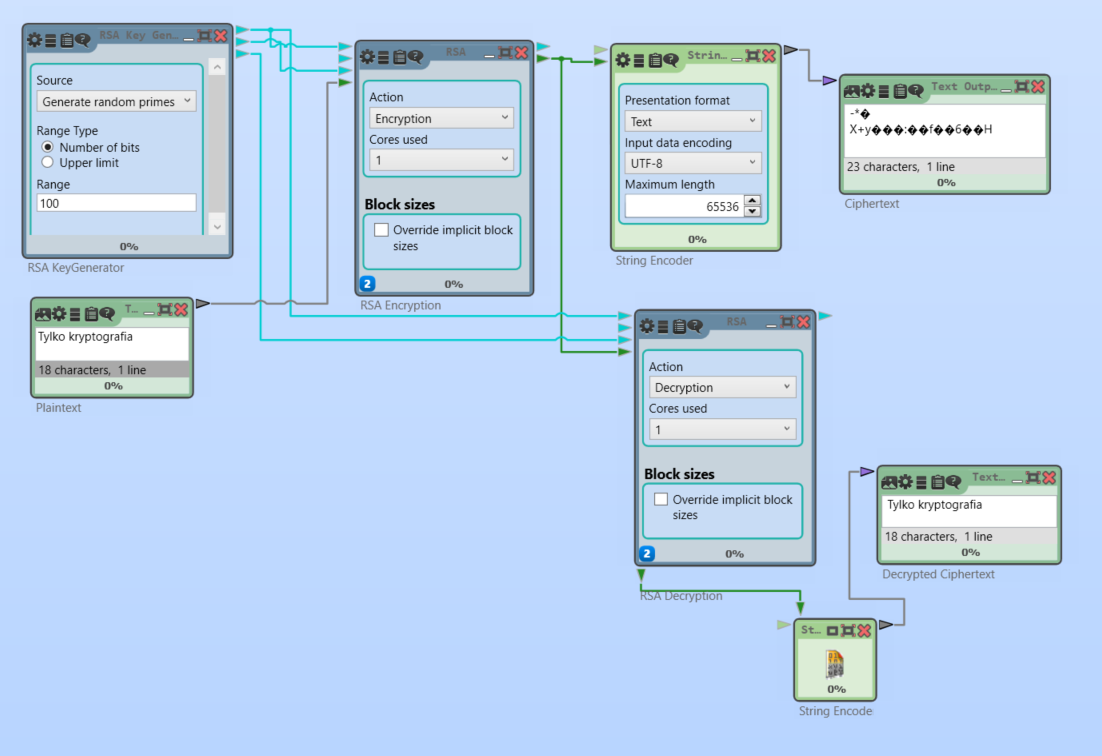
c(m) = m^17 mod 3233

Klucz prywatny jest (n = 3233, d = 413). Dla odszyfrowania używana jest taka funkcja:

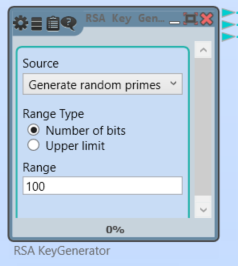
m(c) = c^413 mod 3233

**Przykład Cryptools**

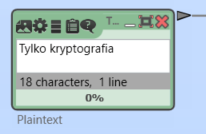
Cały projekt wygląda w taki sposób:

****

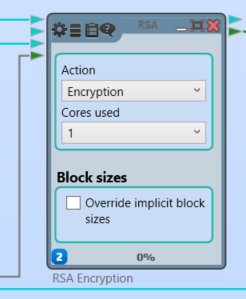
Dla szyfrowania najpierw używamy generator kluczy oparty na liczbach pierwszych:



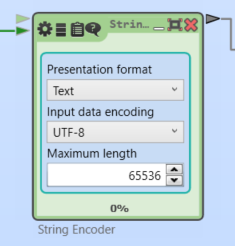
Dalej mamy tekst? Który chcemy zaszyfrować



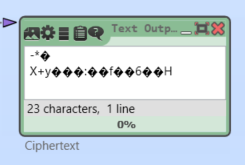
Podłączamy do RSA algorytmu dane wejściowe. Wybieramy opcję szyfrowania.



Po skończeniu operacji otrzymujemy ciąg, który musimy przetworzyć do postaci tekstu:

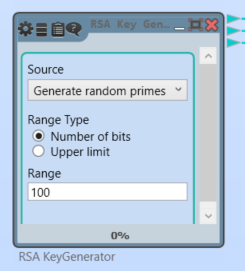


Po skończeniu przetwarzania otrzymujemy zaszyfrowany tekst:

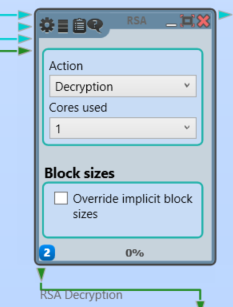


**Dla odszyfrowania:**

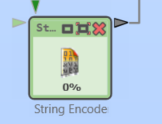
Używamy te same kluczy:

****

Podłączmy szyfrogram i kluczy jak wejścia do naszego algorytmu deszyfrowania RSA. Obieramy opcję Decryption.



Otrzymany ciąg danych przetwarzamy do postaci tekstu z kodowaniem UTF-8 (z który kodowaliśmy).



Na końcu otrzymujemy odszyfrowany tekst:

