Faktoryzacja kluczy RSA

Mikołaj Koszowski
Hubert Nowakowski

1) BatchGCD

Na kluczach zebranych w poprzednim zadaniu zastosowana została metoda batchGCD. Złożoność obliczeniowa algorytmu okazała się być w przybliżeniu proporcjonalna do kwadratu liczby analizowanych kluczy. Aby przeanalizować zbiór 106447 zebranych unikalnych kluczy w czasie dostępnym w ramach laboratoriów, zaimplementowany został również algorytm równoległego batchGCD.

2) Algorytm równoległego batchGDC

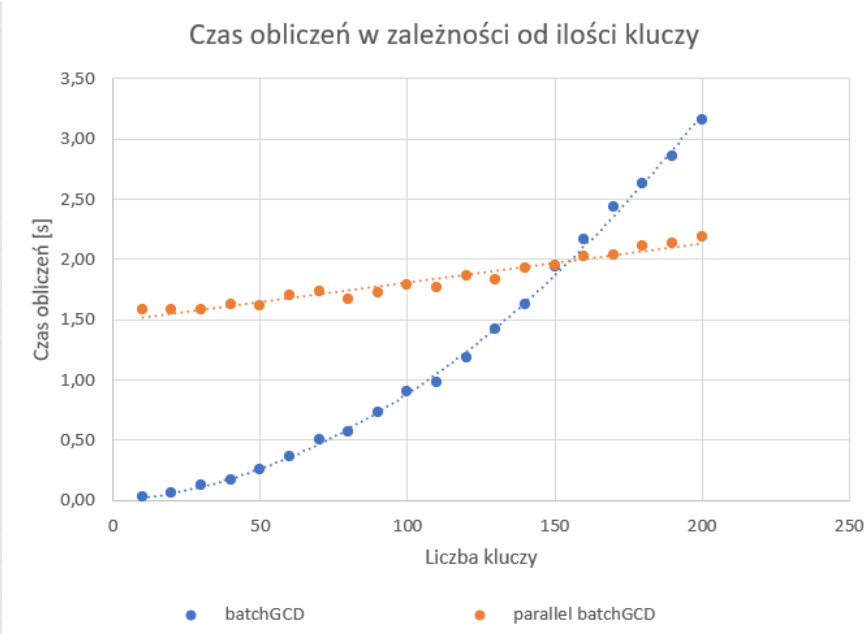
Zaimplementowana wersja równoległego przeprowadzenia algorytmu batchGCD polega na rozdzieleniu zbioru kluczy na *p* losowych podzbiorów. Następnie dla każdego podzbioru stosowana jest metoda batchGCD, co pozwala na równoległe przeprowadzanie obliczeń. Aby zapewnić, że nie pominiemy pary kluczy o wspólnym dzielniku poprzez rozdzielanie początkowego zbioru, proces powtarzany jest k razy. Liczba powtórzeń liczona jest zgodnie ze wzorem:

$$k \approx \frac{log(1-\varepsilon)}{\log m+\log(p-1)-\log(mp-1)},$$

gdzie ε to zadany poziom skuteczności algorytmu, m to liczba węzłów obliczeniowych, a p to liczba podzbiorów.

3) Porównanie złożoności obliczeniowej

Zaimplementowany algorytm równoległego batchGCD zależy liniowo od liczby analizowanych kluczy, co pozwoliło na przeanalizowanie całego zbioru. Przy 85% skuteczności algorytmu nie została znaleziona żadna para kluczy, które miały wspólny dzielnik.



Wykres 1: Porównanie złożoności obliczeniowej algorytmów.

Zródła:

- https://vntkumar8.github.io/docs/iciss_slide.pdf
- Information Systems Security: 13th International Conference, ICISS 2017