Лабораторная работа № 1 по курсу Криптография

Выполнил студент группы М8О-306Б-17 Ветренко Полина.

Задание

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

Вариант 2

 $\begin{array}{l} n1\!=\!1197606395839410537256528037313284196976497391762438410219156212\\ 42807618608591 \end{array}$

 $\begin{array}{l} n_2 \! = \! 1916242087180680156861712994509728052535159091128844805658679025\\ 2967165594044346648117256191866527259013257746490175941447883606374\\ 0717847693631691522075814453568196437131165707175097041470721811222\\ 2280453951875213591639735019844579642622014874212594838041457800464\\ 9211823451274964608882500841718155403512117458135421929696241085675\\ 0448190529031735941575253507798593150790972216736431298009983402302\\ 3021212767107040301344392783417575981002593796696074442689507301\\ \end{array}$

Введение

Факторизация больших целых чисел является нетривиальной задачей. На вычислительной сложности задачи факторизации больших целых чисел основывается криптографический алгоритм с открытым ключом RSA, который стал первой системой, пригодной для шифрования и цифровой подписи. Трудоемкость алгоритмов, работающих с большими числами, оценивается количеством битовых операций. Зная количество операций, выполняемых компьютером за 1 секунду, можно оценить машинное время, необходимое для выполнения алгоритма. Оценка трудоемкости наиболее быстрых алгоритмов факторизации натуральных чисел имеет вид $O(\exp\sqrt{\log n} * \log\log n)$.

Метод решения

Процесс разложения числа на простые множетели называется факторизацией. Для решения этой задачи существует множество алгоритмов, позволяющих находить множетели, используя свойства простых чисел.

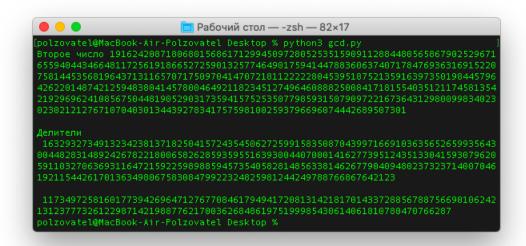
Для решения задачи был использован алгоритм ρ - Полларда, как один из наиболее простых и эффективных. Сложность алгоритма оценивается как $O(N^{1/4})$. Изначально ищем все простые делители исходного числа, затем, получаем все существующие делители перемножением простых делителей. Простые делители ищутся следующим способом: случайным образом выбирается число x, на каждой итерации вычисляется значение функции $f^i(x)$. В качестве функции взята $f(x) = ax^2 + b$ для случайных a и b. На i-ом

шаге получаем значения $x_i = f^i(x_0)(modn), y_i = x_{2i} = f^{2i}(x_0)(modn).$ gcd(abs(x-y), n) даст нетривиальный сомножитель n. Функция выбирается каждый раз, когда количество итераций превышает $O(\sqrt{n})$. Проверка простоты числа осуществляется с помощью теста Миллера-Рабина, который позволяет выполнять проверку быстрее.

Результаты работы

```
Рабочий стол — -zsh — 80×26

[polzovatel@MacBook-Air-Polzovatel Desktop % python3 factor.py ]
Первое число 1197606395839410537256528037313284196976497391762438410219156212428
076186085911
Простые делители
7
17
49
119
509
833
3563
8653
24941
60571
423997
2824563371531898898474583634585348945809751936363791277341953392189275363
19771943600723292289322085442097442620668263554546538941393673745324927541
48017577316042281274067921787950932078765782918184451714813207667217681171
138403605205063046025254598094682098344677844881825772589755716217274492787
336123041212295968918475452515656524551360480427291162003692453670523768197
14377027561097365393235630700033942613417163735609169760167054276624341159767
23528612884860717824293281676095956718595233629910381340558471769366377379
100639192927681557752649414900275982939201461492641883216937993637038818369
24440946853865521168500572190067024428091783505355885922839922702613799716039
70447435049377090426854590430193188057441023044849318248185659554592716828583
polzovatel@MacBook-Air-Polzovatel Desktop %
```



Выводы

В данной лабораторной работе я познакомилась с новой для меня областью на стыке математики и программирования - криптографией. В частности, изучила алгоритм шифрования RSA и различные алгоритмы факторизации больших чисел. В настоящее время задача факторизации некоторых больших чисел можно считать неразрешимой.