УДК 004.004.04

Студ. В.А. Косс, А.А. Потапчик
Науч. рук. Н.В. Ржеутская
(Кафедра информационных систем и технологий, БГТУ)

ПРИМЕНЕНИЕ РАСШИРЕННОГО АЛГОРИТМА ЕВКЛИДА В АЛГОРИТМЕ ШИФРОВАНИЯ RSA

Шифрование является важным инструментом в обеспечении безопасности информации в современном мире. Оно позволяет защитить данные от несанкционированного доступа, а также обеспечить конфиденциальность и целостность передаваемой информации.

Существует два типа алгоритмов шифрования: симметричное шифрование и асимметричное. При реализации симметричного шифрования используется один и тот же ключ как для зашифрования информации, так и для её расшифрования. В свою очередь, асимметричное шифрование использует два разных ключа: один для зашифрования, другой для расшифрования. Первый еще называют «открытым» ключом, а второй –«закрытым» [1].

RSA–одна из первых криптосистем с открытым ключом, используемая для безопасной передачи данных. Эта криптосистемаиспользуется в самых различных продуктах, на различных платформах и во многих отраслях. В настоящее время она встраивается во многие коммерческие продукты, число которых постоянно увеличивается. Также ее используют операционные системы Microsoft, Apple, Sun и Novell и т. д. В аппаратном исполнении RSA-алгоритм применяется на сетевых платах Ethernet, на смарт-картах, широко используется в криптографическом оборудовании [2]. Окно работы программы представлено на рисунке 1.

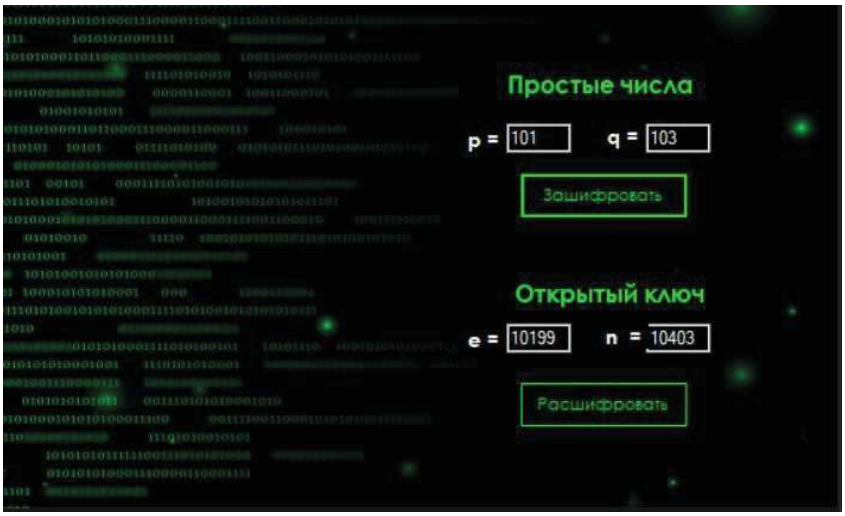

Рисунок 1 – Окно программы

При реализации многих математических и теоретико-числовых алгоритмов и их приложений используется вычисление наибольшего общего делителя (НОД) натуральных чисел. В конце XX века задачи проверки простоты или разложения на множители больших целых чисел, а также операции над классами вычетов по простому модулю стали интенсивно использоваться при формировании криптографических ключей в асимметричных криптосистемах, в том числе и в RSA.

Расширенный алгоритм Евклида в RSA используется для нахождения закрытого ключа d (секретная экспонента), который, в свою очередь, используется для расшифровки зашифрованного сообщения. Сравнение $d \cdot e \equiv 1\ (mod\ \varphi(n))$, где e – целочисленное значение открытого ключа, находящееся в пределах $(1; \varphi(n))$и взаимно простое со значением функции Эйлера, должно быть истинным. Закрытый ключ $\{d, n\}$ остается у отправителя и держится в секрете. Для вычисления закрытого ключа была использована функция, представленная в листинге 1.

```
     privatelong Calculate_d(long _e, long m)//расширенный алго-
ритм Евклида
              {
     long i = m, v = 0, d = 1;
     while (d > 0)
                  {
     long t = i / d, x = d;
                      d = i % x;
                      i = x;
                      x = _e;
                      d = v - t * x;
                      v = x;
                  }
                  v %= m;
     if (v < 0) v = (v + m) % m;
     return v;
     }
```

Листинг 1 – Реализация расширенного алгоритма Евклида

Результат расшифрования, при котором используется значение закрытого числа, подставленное в формулу $m=D(c)\equiv c^d(mod n)$ для расчета исходного числа m, представлен на рисунке 2.

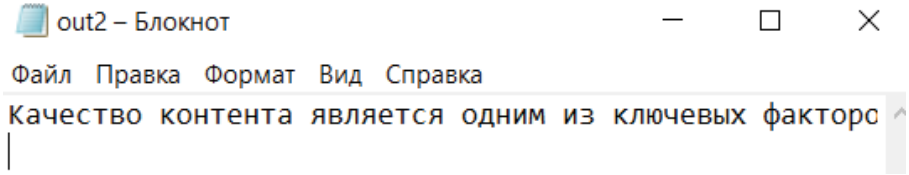

Рисунок 2 – Результат расшифрования

Расширенный алгоритм Евклида также используется для проверки того, что выбранные простые числа являются взаимно простыми. Если они не являются взаимно простыми, то алгоритм RSA не будет работать. Важной характеристикой качества программы является время расшифрования и зашифрования. Время шифрования и расшифрования зависит от размера шифруемого текста, языка написания программы, значения ключа, технических характеристик компьютера и др. Время шифрования и расшифрования текста из 300 символов, шифруемых на языке C#, представлено на рисунке 3 и рисунке 4 соответственно.

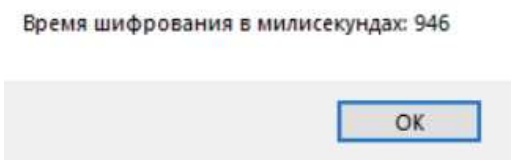

Рисунок 3 – Время зашифрования

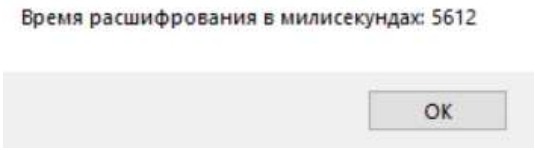

Рисунок 4 – Время расшифрования

В результате процесс расшифрования занимает почти в 6 раз больше времени. Таким образом, алгоритм RSA является классическим примером шифров с открытым ключом и является актуальным и в настоящее время. Этот метод шифрования нельзя назвать самым безопасным, так как он был разработан еще в XX веке. Однако для современных технологий алгоритм RSA используется и сегодня, например, для передачи зашифрованных ключей.

В основе алгоритма RSA лежит использование расширенного алгоритма Евклида.

ЛИТЕРАТУРА

1. Алгоритм шифрования данных RSA: электронный журнал: наука, техника и образование; рубрика 2 «Информационные технологии» / авт. Е. А. Коваленко, О. С. Клочко

2. Преимущества и недостатки алгоритма шифрования RSA [Электронный ресурс]/https://studwood.net/–2019. – Режим доступа: https://studwood.net/1685074/informatika/preimuschestva_nedostatki_algoritma_shifrovaniya– Дата доступа: 29.03.2023.