ĐỒ ÁN 01 – HỆ MÃ HÓA RSA

October 16, 2023

1 (4 điểm) Kiểm tra số nguyên tố

1.1 Định nghĩa số nguyên tố

Cho một số nguyên $n \in \mathbb{Z}$ lớn hơn 1. Khi đó, n được gọi là số nguyên tố khi và chỉ khi n thỏa tính chất n chỉ có hai ước dương là 1 và chính nó n.

$$\forall a \in \mathbb{Z}^+, a \mid n \iff a = 1 \vee a = n$$

Trong đó:

- $\mathbb{Z}$ là tập các số nguyên.
- $\mathbb{Z}^+$ là tập các số nguyên dương.
- $a \mid n$ có nghĩa a là ước của n, tức là $\exists k \in \mathbb{Z}, n = ka$.

1.2 Giới thiệu một số cách kiểm tra số nguyên tố

1. Phương pháp vét cạn: Thử lần lượt các số nguyên từ 1 đến n. Nếu chỉ có 1 và n là ước của n thì n là số nguyên tố.

   Cải tiến: Thay vì thử các số nguyên từ 1 đến n, ta thử các số nguyên dương $a \leq \sqrt{n}$.

2. Thuật toán Miller–Rabin: Đây là thuật toán ngẫu nhiên dạng Monte Carlo, tức kết quả trả ra không phải lúc nào cũng chính xác, nhưng có thể cải thiện xác suất chính xác bằng cách thực hiện thuật toán với các dữ liệu đầu vào khác nhau.

   Tham khảo: `https://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test`.

3. Thuật toán AKS: Không giống với thuật toán Miller-Rabin, đây là thuật toán tất định, tức kết quả trả ra luôn chính xác.

   Tham khảo: `https://en.wikipedia.org/wiki/AKS_primality_test`.

1.3 Yêu cầu

Trong bài này, sinh viên viết chương trình dùng để kiểm tra một số nguyên lớn hơn 1 có phải là số nguyên tố hay không. Chương trình thỏa các yêu cầu sau:

- Ngôn ngữ lập trình là `C++`.
- Toàn bộ mã nguồn được lưu thành một file duy nhất là `main.cpp`.
- Ngoại trừ các tính năng từ ngôn ngữ lập trình `C++` (phiên bản `C++17` trở xuống) và thư viện chuẩn của `C++` (`C++` Standard Library) là được cho phép sử dụng, toàn bộ các thư viện khác bị cấm.

- Sau khi biên dịch mã nguồn thành file `main`, chương trình được chạy như sau.

```
$ .\main test.inp test.out
```

  Trong đó:

  - `test.inp` là file text chứa dữ liệu đầu vào của chương trình, gồm có 1 dòng chứa số nguyên cần kiểm tra được viết bằng các chữ số thập lục phân in hoa dưới dạng little endian.
  - `test.out` là file text chứa dữ liệu đầu ra của chương trình. Trong file này, chương trình trả ra 1 dòng chứa số 0 nếu số cần kiểm tra không phải là số nguyên tố hoặc số 1 nếu số cần kiểm tra là số nguyên tố.
  - **Lưu ý:** File text chứa dữ liệu đầu vào và đầu ra không nhất thiết phải có tên `test.inp` và `test.out`.

- Thời gian chạy tối đa của chương trình là **60 giây** cho mỗi test. Khi quá thời gian quy định, chương trình sẽ bị ngắt.

1.4 Test mẫu

Phần này chỉ liệt kê một số test mẫu. Đối với các test mẫu còn lại, sinh viên xem trong thư mục chứa các test cho bài này.

- Test 01:

test.inp	test.out
F1	1

  Giải thích: `0x1F = 31` là số nguyên tố nên kết quả trả ra sẽ là `1`.

- Test 02:

test.inp	test.out
B5	0

  Giải thích: `0x5B = 91 = 7 * 13` không phải là số nguyên tố nên kết quả trả ra sẽ là `0`.

1.5 Cơ cấu các test và cách tính điểm

Có tất cả 100 test, mỗi test sẽ tương ứng với 1% điểm của bài này. Trong đó:

- Có 10/100 test mẫu (có đủ cả `test.inp` và `test.out`).
- Có 10/100 test công khai (chỉ có `test.inp`).
- Có 80/100 test bí mật (không được công bố như test mẫu và test công khai, mà chỉ dùng riêng cho việc tính điểm).

Các test mẫu và test công khai sẽ được cung cấp trong thư mục chứa các test trong bài này. Xét về độ khó, cơ cấu các test như sau:

- Kiểm tra số nguyên từ trên 256 đến 512 bit: 10/100 test.
- Kiểm tra số nguyên từ trên 128 đến 256 bit: 20/100 test.
- Kiểm tra số nguyên từ trên 64 đến 128 bit: 30/100 test.
- Kiểm tra số nguyên từ 64 bit trở xuống: 40/100 test.

2 (3 điểm) Mã hóa RSA - Sinh khóa

2.1 Cơ chế sinh khóa của hệ mã RSA

Cho số nguyên dương $N = pq$ với p và q là các số nguyên tố. Khi đó, với cặp số nguyên dương (e, d) thỏa mãn $\varphi(N) = (p-1)(q-1)$ là ước của $ed - 1$, tức là $ed = 1 \pmod{\varphi(N)}$, ta có cặp khóa của hệ mã RSA như sau:

- Khóa công khai: Cặp số nguyên (N, e).
- Khóa bí mật: Cặp số nguyên (N, d).

2.2 Yêu cầu

Trong bài này, sinh viên viết chương trình dùng để sinh khóa cho hệ mã RSA. Chương trình thỏa các yêu cầu sau:

- Ngôn ngữ lập trình là `C++`.
- Toàn bộ mã nguồn được lưu thành một file duy nhất là `main.cpp`.
- Ngoại trừ các tính năng từ ngôn ngữ lập trình `C++` (phiên bản `C++17` trở xuống) và thư viện chuẩn của `C++` (`C++` Standard Library) là được cho phép sử dụng, toàn bộ các thư viện khác bị cấm.
- Sau khi biên dịch mã nguồn thành file `main`, chương trình được chạy như sau.

  ```
  $ .\main test.inp test.out
  ```

  Trong đó:

  - `test.inp` là file text chứa dữ liệu đầu vào của chương trình, gồm có 3 dòng:
    - Dòng 01 chứa số nguyên tố p.
    - Dòng 02 chứa số nguyên tố q.
    - Dòng 03 chứa số nguyên dương e.
    - **Lưu ý:** Các số nguyên được viết bằng các chữ số thập lục phân in hoa dưới dạng little endian.
  - `test.out` là file text chứa dữ liệu đầu ra của chương trình. Trong file này, chương trình trả ra 1 dòng chứa số nguyên dương d nhỏ nhất được viết bằng các chữ số thập lục phân in hoa dưới dạng little endian sao cho (pq, e) và (pq, d) là một cặp khóa hợp lệ theo cơ chế sinh khóa của hệ mã RSA được mô tả ở trên, hoặc trả ra -1 nếu không thể nào tìm ra được một cặp như vậy.
  - **Lưu ý:** File text chứa dữ liệu đầu vào và đầu ra không nhất thiết phải có tên `test.inp` và `test.out`.
- Thời gian chạy tối đa của chương trình là **60 giây** cho mỗi test. Khi quá thời gian quy định, chương trình sẽ bị ngắt.

2.3 Test mẫu

Phần này chỉ liệt kê một số test mẫu. Đối với các test mẫu còn lại, sinh viên xem trong thư mục chứa các test cho bài này.

- Test 01:

test.inp	test.out
D1 B3 B	BB1

Giải thích: Do `p = 0x1D = 29` và `q = 0x3B = 59` nên $\varphi(N) = \varphi(pq) = (p-1)(q-1) = 1624$. Suy ra, với `e = 0xB = 11`, câu trả lời đúng sẽ là `d = 443 = 0x1BB` do $ed - 1 = 4872 = 1624 \times 3$ và d là số nguyên dương nhỏ nhất thỏa yêu cầu.

- Test 02:

test.inp	test.out
56 71 12	-1

Giải thích: Do `p = 0x65 = 101` và `q = 0x17 = 23` nên $\varphi(N) = \varphi(pq) = (p-1)(q-1) = 2200$. Suy ra, với `e = 0x21 = 33`, câu trả lời đúng sẽ là `-1` do ước chung nhỏ nhất của e và $\varphi(N)$ là $gcd(e, \varphi(N)) = 11 \neq 1$ nên sẽ không tồn tại số nguyên dương d thỏa yêu cầu.

2.4 Cơ cấu các test và cách tính điểm

Có tất cả 100 test, mỗi test sẽ tương ứng với 1% điểm của bài này. Trong đó:

- Có 10/100 test mẫu (có đủ cả `test.inp` và `test.out`).
- Có 10/100 test công khai (chỉ có `test.inp`).
- Có 80/100 test bí mật (không được công bố như test mẫu và test công khai, mà chỉ dùng riêng cho việc tính điểm).

Các test mẫu và test công khai sẽ được cung cấp trong thư mục chứa các test trong bài này. Xét về độ khó, cơ cấu các test như sau:

- Tính d với mỗi p và q từ trên 256 đến 512 bit: 10/100 test.
- Tính d với mỗi p và q từ trên 128 đến 256 bit: 20/100 test.
- Tính d với mỗi p và q từ trên 64 đến 128 bit: 30/100 test.
- Tính d với mỗi p và q từ 64 bit trở xuống: 40/100 test.
- **Lưu ý:** Tất cả 100/100 test đều có e thỏa $e < \min(p, q)$.

3 (3 điểm) Mã hóa RSA - Mã hóa và Giải mã

3.1 Cơ chế mã hóa và giải mã của hệ mã RSA

1. Cơ chế mã hóa: Cho cặp khóa công khai (N, e). Khi đó, với thông điệp là số nguyên dương m sao cho $m < N$ và ước chung lớn nhất của m và N là $gcd(m, N) = 1$, bản mã của m lúc này là số nguyên dương c nhỏ nhất thỏa $c = m^e \pmod{N}$.

2. Cơ chế giải mã: Cho cặp khóa bí mật (N, d). Khi đó, với bản mã là số nguyên dương c sao cho $c < N$ và ước chung lớn nhất của c và N là $gcd(c, N) = 1$, thông điệp của c lúc này là số nguyên dương m nhỏ nhất thỏa $m = c^d \pmod N$.

Như vậy, về cơ bản, cơ chế mã hóa và giải mã đều giống nhau: Cho cặp số nguyên dương (N, k). Khi đó, với số nguyên dương x thỏa $x < N$ và ước chung lớn nhất của x và N là $gcd(x, N) = 1$, cần tìm số nguyên dương y nhỏ nhất thỏa $y = x^k \pmod N$.

3.2 Yêu cầu

Trong bài này, sinh viên viết chương trình dùng để thực hiện các cơ chế mã hóa và giải mã. Chương trình thỏa các yêu cầu sau:

- Ngôn ngữ lập trình là `C++`.
- Toàn bộ mã nguồn được lưu thành một file duy nhất là `main.cpp`.
- Ngoại trừ các tính năng từ ngôn ngữ lập trình `C++` (phiên bản `C++17` trở xuống) và thư viện chuẩn của `C++` (`C++` Standard Library) là được cho phép sử dụng, toàn bộ các thư viện khác bị cấm.
- Sau khi biên dịch mã nguồn thành file `main`, chương trình được chạy như sau.

  ```
  $ .\main test.inp test.out
  ```

  Trong đó:

  - `test.inp` là file text chứa dữ liệu đầu vào của chương trình, gồm có 3 dòng:
    - Dòng 01 chứa số nguyên dương N.
    - Dòng 02 chứa số nguyên dương k.
    - Dòng 03 chứa số nguyên dương x.
    - **Lưu ý:** Các số nguyên được viết bằng các chữ số thập lục phân in hoa dưới dạng little endian.
  - `test.out` là file text chứa dữ liệu đầu ra của chương trình. Trong file này, chương trình trả ra 1 dòng chứa số nguyên dương y nhỏ nhất được viết bằng các chữ số thập lục phân in hoa dưới dạng little endian sao cho $y = x^k \pmod N$.
  - **Lưu ý:** File text chứa dữ liệu đầu vào và đầu ra không nhất thiết phải có tên `test.inp` và `test.out`.
- Thời gian chạy tối đa của chương trình là **60 giây** cho mỗi test. Khi quá thời gian quy định, chương trình sẽ bị ngắt.

3.3 Test mẫu

Phần này chỉ liệt kê một số test mẫu. Đối với các test mẫu còn lại, sinh viên xem trong thư mục chứa các test cho bài này.

- Test 01:

test.inp	test.out
FA6 B F1	B53

Giải thích: Do `N = 0x6AF = 1711`, `k = 0xB = 11`, và `x = 0x1F = 31` nên $y = 31^{11} = 859 \pmod{1711}$. Suy ra, câu trả lời đúng sẽ là `y = 859 = 0x35B` do y là số nguyên dương nhỏ nhất thỏa yêu cầu.

- Test 02:

test.inp	test.out
FA6 BB1 B53	F1

Giải thích: Do `p = 0x6AF = 1711`, `k = 0x1BB = 443`, và `x = 0x35B = 859` nên $y = 859^{443} = 31 \pmod{1711}$. Suy ra, câu trả lời đúng sẽ là `y = 31 = 0x1F` do y là số nguyên dương nhỏ nhất thỏa yêu cầu.

3.4 Cơ cấu các test và cách tính điểm

Có tất cả 100 test, mỗi test sẽ tương ứng với 1% điểm của bài này. Trong đó:

- Có 10/100 test mẫu (có đủ cả `test.inp` và `test.out`).
- Có 10/100 test công khai (chỉ có `test.inp`).
- Có 80/100 test bí mật (không được công bố như test mẫu và test công khai, mà chỉ dùng riêng cho việc tính điểm).

Các test mẫu và test công khai sẽ được cung cấp trong thư mục chứa các test trong bài này. Xét về độ khó, cơ cấu các test như sau:

- Tính y với N từ trên 512 đến 1024 bit: 10/100 test.
- Tính y với N từ trên 256 đến 512 bit: 20/100 test.
- Tính y với N từ trên 128 đến 256 bit: 30/100 test.
- Tính y với N từ 128 bit trở xuống: 40/100 test.
- **Lưu ý:** Tất cả 100/100 test đều có k và x thỏa $k, x < N$.

4 Các quy định khác về đồ án

- Đồ án được thực hiện cá nhân.
- Thời gian thực hiện là 3 tuần tính từ lúc đồ án được công bố chính thức bằng thông báo.
- Cấu trúc bài làm như sau:

```
<MSSV>.zip
├── project_01_01
│   └── main.cpp
├── project_01_02
│   └── main.cpp
└── project_01_03
    └── main.cpp
```

Trong đó:

  - `<MSSV>` là mã số sinh viên của người nộp.
  - `.zip` là định dạng nén cho bài làm (định dạng ZIP).
  - `project_01_01` là nơi chứa mã nguồn cho phần Kiểm tra số nguyên tố.
  - `project_01_02` là nơi chứa mã nguồn cho phần Sinh khóa của hệ mã RSA.
  - `project_01_03` là nơi chứa mã nguồn cho phần Mã hóa và Giải mã của hệ mã RSA.

  - `main.cpp` là các file chứa toàn bộ mã nguồn tương ứng với từng phần nêu trên.

- Trường hợp chương trình của phần nào bị lỗi không biên dịch được thì phần đó không có điểm.
- Trường hợp chương trình biên dịch được nhưng không trả ra được kết quả (do gặp lỗi khi chạy hoặc quá thời gian quy định) ở test nào thì test đó không có điểm.
- Thư viện chuẩn của `C++` (`C++` Standard Library) là các thư viện được liệt kê ở đây: `https://en.cppreference.com/w/cpp/standard_library`

  Cần lưu ý về phiên bản `C++` để lựa chọn thư viện thích hợp.
- Mọi thắc mắc vui lòng gửi về email: `nvqhuy@fit.hcmus.edu.vn`