Selected Topics of Embedded Software Development 2

WS-2021/22

Prof. Dr. Martin Schramm

Testing and generating Prime Numbers and Safe Primes using CryptoCore

Group 2 – Team 4
Rashed Al-Lahaseh – 00821573
Vikas Gunti - 12100861

Supriya Kajla – 12100592

Srijith Krishnan – 22107597

Wannakuwa Nadeesh – 22109097

Prime Numbers

A whole positive integer which are larger than or equal to 2 that can only be divided by 1 and themselves are known as prime numbers.

There are three reasons why prime numbers are extremely important:
- They are regarded the basic components of natural numbers in number theory.
- They constitute the foundation for many clever mathematical concepts.
- They are extremely useful in current cryptography (for example: public key cryptography).

Safe Prime Numbers

A safe prime is a prime number p of the form $p = 2q + 1$ where q is also prime. In such a case, q is called a Sophie Germain prime.

Safe primes are used in some implementations of the Diffie–Hellman key exchange protocol, for example, to protect against certain types of attacks.

Note:
Diffie–Hellman (DH) key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography.

Usage of Prime Numbers and Safe Prime Numbers

Many cryptographic techniques necessitate the usage of prime number testers and prime number generators, which necessitate the use of "large" random primes.

There are two types of prime number testers:

- Deterministic approaches, which offer a 100% response whether a number is prime or not.
- Probabilistic tests, which can only give a probability of whether a number is prime or not.

But generally probabilistic tests are more time-efficient compared to deterministic testers.

Miller–Rabin Test (MRT)

Is a probabilistic prime tester and algorithm from the mathematical subfield of number theory, particularly algorithmic number theory. In addition, the Miller-Rabin Test is the most well-known and widely used probabilistic prime tester.

Theory

Steps:

1. Find $n - 1 = 2^k m$
2. Choose a where $1 < a < n - 1$
3. Compute $b_0 = a^m (mod\ n),\ b_i = {b_{i-1}}^2$

Example:
Is 53 a prime number?

1. $n - 1 = 2^k m$, where k and m are whole numbers

   $53 - 1 = 2^k m$

   $52 = 2^k m$

   $\frac{52}{2^1} = 24 \Big| \frac{52}{2^2} = 13 \Big| \frac{52}{2^3} = 6.5$

   So, we do not consider the last one because it is not whole number and we pick the one before.

   $52 = 2^k m$
   $52 = 2^2 * 13$

2. Choose a where $1 < a < n - 1$ and we are going to pick a = 2 for example

3. Compute $b_0 = a^m (mod\ n)$

   $b_0 = 2^{13}(mod\ 53) = 30\ mod\ 53$

   $If\ b_0 = \begin{cases} +1 \\ -1 \end{cases}$ then the number we picked is prime

   $b_0 = 30\ then\ we\ move\ to\ b_1$

   $b_1 = 30^2\ mod\ 53 = -1\ mode\ 53$
   Note: If $a\ mod\ b = b - 1$ then it's equivalent to saying a congruent to $-1\ mod\ b$

   $If\ b_1 = \begin{cases} +1\ then\ it\ is\ Composite \\ -1\ then\ it\ is\ Prime \end{cases}$, then 53 is prime number

Algorithm

Pseudocode

The algorithm can be written in pseudocode as follows.

```
$ First Input: n > 3, and odd integer to be tested
$ Second Input: k, number of testing rounds
$ Output: Type is ‘Composite’ or ‘Prime’
```

Figure 1: MRT Pseudocode Terminal Window

```
Write n as $2^n * d$ with $d$ odd and that’s by factoring powers of 2 from $n - 1$
Loop: repeat k times:
    pick random integer a in the range of [2, n - 2]
    x = $a^d \mod n$
    if x = 1 or x = n – 1 then
        continue Loop
    else
        repeat r – 1 time:
            x = $x^2 \mod n$
            if x = n – 1 then
                continue Loop
        return “Type is Composite”
    return “Type is Prime (probably)”
```

Figure 2: MRT Pseudocode

So as a conclusion here the parameter **k** determines the accuracy of the test where the greater the number of rounds, the more accurate is the result.

Sample Code for research purposes:

- GeeksForGeeks - Primality Test using Miller Rabin C++

Complexity

Using repeated squaring, the running time of this algorithm is $\boldsymbol{O}(\boldsymbol{k}\ \boldsymbol{log3n})$, where $\boldsymbol{n}$ is the number tested for primality, and $\boldsymbol{k}$ is the number of rounds performed; thus, this is an efficient, polynomial-time algorithm. FFT-based multiplication can push the running time down to $\boldsymbol{O}(\boldsymbol{k}\ \boldsymbol{log2n})$.

SageMath (Sage)

SageMath (Sage) is the free, open-source competitor to Maple, Mathematica, Magma, and MATLAB.

It is a computer-algebra system ideally suited to students of mathematics, and all other STEM fields, vastly more sophisticated and advanced than any graphing calculator.

Moreover, Sage is based on the popular programming language Python. That means if you use Sage in a mathematics, statistics, physics, or data-science class and a lot of other STEM fields are using Sage too, such as chemistry and geology.

Sage is great for calculus, differential equations, linear algebra, data science, abstract algebra, discrete mathematics, graph theory, physics, numerical computing, machine learning, and many other courses. Sage also includes a full copy of R for those who use statistics or those who want to learn data science, and other open-source systems like Maxima, Pari, and GAP.

Resources can be used:

- Manual - Sage for Undergraduates by Gregory V. Bard | University of Wisconsin

- SageCell - Server-side version of SageMath

Usage

We will be using SageMath to implement the MRT algorithm in two different domains

- Normal (ordinary) domain
- Montgomery Domain

Where the generated function should check whether the generated random number is prime or not. If it’s not prime then the generate prime function will generate another true random number and checks again with the MRT and this will continue until MRT function returns a flag that a prime number is detected.

References

- ENCYCLOPEDIA OF CRYPTOGRAPHY AND SECURITY by Henk C.A. van Tilborg | Eindhoven University of Technology The Netherlands

- Prime and Safe Prime Numbers

- Sage Math - Manual by Gregory V. Bard | University of Wisconsin
- Wiki: Miller–Rabin primality test
- SPOJ - Prime or Not problem
- Miller Rabin Algo Implementation in C++
- GeeksForGeeks - Primality Test using Miller Rabin C++
- Safe primes, Sylow theorems, and Cryptography
- TRNG in RSA Algorithm
- Diffie–Hellman key exchange