INVESTIGATION STUDY OF FEASIBLE PRIME NUMBER TESTING ALGORITHMS

Mohammed M. ASAD, Ibrahim Marouf, Qasem Abu AL-HAIJA
King Faisal University, Department of Electrical Engineering
Al-Ahsa 31982, P.O. Box 380, qalhaja@kfu.edu.sa

Abstract: In this expository paper, we describe three primality-testing algorithms: Miller-Rabin, Fermat and AKS primality tests. The first test is very efficient, but is only capable of proving that a given number is either composite or 'very probably' prime. The second test is also probabilistic with lower probability and higher execution time. The third test is a deterministic unconditional polynomial time algorithm to prove that a given number is either prime or composite; however, it had no practical applications due to the time complexity O ($\log^5$ (n)). Thus, the first primality test is at present one of the most widely used in practice as it run at logarithmic run time complexity O (log (n)).

Keywords: *Prime Number, Fermat Primality Test, Miller-Rabin Primality Test, AKS Primality Test*

I. INTRODUCTION

Recently, the prompt growth in the information and communication technology (ICT) has raised the need to exchange massive information. This led to introduce the technology of Big Data and Internet-Of-Things (IoT) (Gubbi et al. 2013) as well as increased the need for secure communication to provide the privacy and information security from unauthorized users. As a result, several security techniques where used to provide data security and assurance named collectively as Cryptography (Paar and Pelzl 2010). Cryptography is the science of changing information to unreadable form by (i.e. ciphertext) using the encryption process, where only the authorized person can retrieve or modify the original information (i.e. plain text) by using the decryption process.

Based on the mechanism of Encryption/Decryption processes, cryptographic algorithms can be classified into Symmetric Key Cryptography (SKC) in which the same key is used to encrypt the plaintext and to decrypt the ciphertext, and Public Key Cryptography (PKC) in which the two parties (sender and receiver) have two keys; one public shared common key for encryption and one private key for decryption. Modern SKC algorithms such as AES or 3DES are very secure. However, there are several drawbacks associated with symmetric-key scheme such as key distribution problem, number of keys or the lack of protection against cheating (Paar and Pelzl 2010). Thus, PKC have solved many of SKC's related problems. PKC algorithms are used mainly for Key Establishment, Identification and Encryption. RSA is well-known public-key algorithm.

Even though PKC algorithms have resolved many of the SKC issues, PKC algorithms requires significant computation, which based considerably on the use of number theory and modular arithmetic. For instance, RSA crypto-algorithm (Abu Al-Haija et al. 2014a) primarily depends on the modular arithmetic involving the use of large prime numbers with even 1024-bit or more to guarantee an acceptable level of security. Such large numbers cannot be easily selected with modest trial and error methods, instead, distinctive iterative methods are used to test if the number is prime or composite that are collectively called as primality testing.

Primality testing (Gallier 2017) is a pure mathematics problem that concerns of determining whether a given integer is prime. This problem has caught the interest of mathematicians in the 20th century with the advent of cryptographic systems that use large primes, such as RSA, which was the main driving force for the development of fast and reliable methods for primality testing. Many algorithms were proposed by the scientists over the past few years to address the efficient method of testing the primality of the integer number. For instance, in this paper, we will thoroughly review three of them namely; Fermat primality test, Miller-Rabin primality test, and AKS primality test as well as we mention the two other tests namely Solovay–Strassen and Baillie-PSW primality tests for comparison purposes

The remaining of this paper is organized as follows: Section 2 provides a brief background for the topics related to prime numbers or involved in the primality testing algorithms such as divisibility, prime number theorem, Euler theorem and Fermat's Little Theorem. Section 3 discusses three different common and practicable primality-testing algorithms with numerical examples and flowcharts and it mentions two other algorithms as well as provides a cost complexity summery of all algorithms. Finally, Section 4 concludes the paper.

II. MATHMATICAL BACKROUND

In modern crypto-system, messages (data) are represented in numerical values. These values are encrypted and decrypted using mathematical operations that turn input message in numerical value and into unreadable form. Building, analyzing and attacking crypto-systems require mathematical tools and number theory is considered as the most important of these. The number theory (Stein 2008), sometimes-called higher arithmetic, is a branch of

Manuscript received August 11, 2017; revised September 14, 2017

mathematics that is concerned with the properties of positive integers; divisibility, primes and congruence. Number Theory has a reached history of (at least) several centuries And among several mathematicians (Z. I. Borevich and I. R. Shafarevich 1964) (Allan J. Silberger 1967)

❖ **Divisibility**

Definition: Let *a & b* be integers with $a \neq 0$. It is said a divides b, if there is an integer k such that, *b= ak*. This is denoted by $a \mid b$. Another way to express this is that b is a multiple of a.

Example: 3|15 is 3 divides 15 or 15 is a multiple of 3.

Properties: Let a, b, and c represent integers.
1. For every a ≠ 0, a|0 and a|a. Also, 1|b for every b.
2. If a|b and b|c, then a|c.
3. If a|b and a|c, then a| (sb+ tc) for all integers s and t.

❖ **Chebysev Theorem**

Definition: A number p > 1 that is divisible only by one and itself is called a prime number.
Composite number means that the number can be written as the product of two smaller numbers. For example, 15 can be written as 3 times 5. Primes number are infinite, however, in finite groups the number of primes can be estimated using Prime Number Theorem (Trappe and Washington 2002). Let π(x) be the number of primes less than x. Then

$$\pi(x) \approx \frac{x}{\ln(x)} \tag{1}$$

Example: calculate the number of primes in 50 digits only:

$$\pi(x) \approx \pi(50) - \pi(49) = \frac{10^{50}}{ln 10^{50}} - \frac{10^{49}}{\ln(10^{49})}$$

$$\approx 7.80 \times 10^{47}\ number.$$

Prime numbers are the building block for positive integers. All integers can be represented be a product of prime numbers. Moreover, these representations are unique. This uniqueness plays a vital role in cryptographic algorithms. Factorizing any integers will produce a unique product of primes where the factors can be reordered only. For example, factoring the number 504 produces $2^3 \times 3^2 \times 7^1$, ignoring the ordering, there is no other representation for the number 504 (Trappe and Washington 2002).

❖ **Euler Theorem**

Where ∅(n) is Euler's ∅-Function (Trappe and Washington 2002), defined to be the number m such that m < n and *gcd (m, n) =1*.

Definition: if *gcd (a, n) = 1*, then

$$a^{\phi(n)} \equiv 1 (mod\ n) \tag{2}$$

Example: What is the last three digits of 7^{1603}?

$$Since\ \phi(1000) = 1000\left(1-\frac{1}{5}\right)\left(1-\frac{1}{2}\right) = 400$$

$$7^{1606} = \left(7^{400}\right)^4 7^3 \equiv 1^4\, 7^3 \equiv 343 mod\ 1000$$

Thus, by using Euler's Theorem the last three digits are 343.

❖ **Fermat's Little Theorem**

Definition: If p is a prime number and p∤a

$$a^p \equiv a\ mod\ p$$
$$a^{p-1} \equiv 1 mod\ p \tag{3}$$

Example: Evaluate $3^{100,000}$ mod 53

$$3^{53-1} \equiv 1 mod\ 35 \rightarrow 3^{52} \equiv 1 mod\ 53$$
$$\frac{100,000}{52} \rightarrow quetiont = 1923\ , reminder = 4$$
$$3^{100,000} = 3^{999,996}\, 3^4 \equiv \left(3^{52}\right)^{1923} 3^4\ mod\ 53$$
$$\equiv 1^{1923}\, 3^4\ mod\ 53 \equiv 81 mod\ 53$$

In previous example, instead of computing $3^{100,000}$, using Fermat's little theorem makes it easier and faster to calculate. In implementation, Fermat's Little theorem can be used as a primality tester. First, an odd number is n chosen. Then, $2^{(n-1)}$ mod n is computed. If the result is congruent to 1, the chosen number is a prime and if not, choose the next odd number to test. The advantage of this test is faster than factoring n, especially since this procedure eliminate many n quickly. Fermat's Little theorem is a special case of Euler's theorem. Euler's theorem □(n) is used to calculate the relatively prime numbers less than or equal an integer. For example, 10 is co-prime with 1, 3, 7, 9.

Definition: Let □(n) be the number of integers 1 < a < n such that *gcd (a, n) = 1*.

$$\phi(n) = n \coprod_{p|n} \left(1 - \frac{1}{p}\right) \tag{4}$$

where p is prime factor of n.

Example Find the number of integers a≤120 such that *gcd(a,120) =2*.

$$120 = 2^3 \times 3 \times 5 \rightarrow \phi(120) = 120\left(\frac{1}{2}\right)\left(\frac{2}{3}\right)\left(\frac{4}{5}\right) = 32$$

III. PRIMALITY TESTING TECHNIQUES

Generating prime numbers is a very essential part in any crypto-system since they depend heavily on prime's properties. There are two methods to generate a prime (Trappe and Washington 2002). First algorithms are called Prime Sieve. In this kind, primes in a specified range are generated, however, if the need is only for an individual prime, primality testers are more convenient since the first methods is very slow comparing to the second as well as generating many unneeded primes. Primality testers are algorithms used to check if the chosen number is whether a

prime or composite. Unlike integer factorization, primality testers do not generate a prime, but they state whether the input is prime or not. Some of the Primality testers are used to prove a number is prime where some are used to prove a compositeness. Primality testers can be divided into two division; probabilistic and deterministic primality tester.

Basic Principle: Let n be an integer and suppose there exist integers x and y with
$x^2 \equiv y^2 \ mod\ n$, but $x \neq \pm y \ mod\ n$, then n is composite. Moreover, *gcd (x — y, n)* gives a nontrivial factor of n.

❖ Fermat Primality Test

Fermat Primality Test is a probabilistic primality tester based on Fermat's Little Theorem (Agrawal 2006). It is used to check the compositeness of numbers. If a number is declared to be composite, then it is guaranteed to be composite. In contrast, if a number is declared to be prime, it is probably to be prime. However, Fermat Primality Test is quite accurate for large numbers. Moreover, this test can be carried out quickly (Trappe and Washington 2002). The algorithm is shown below in Figure 1.

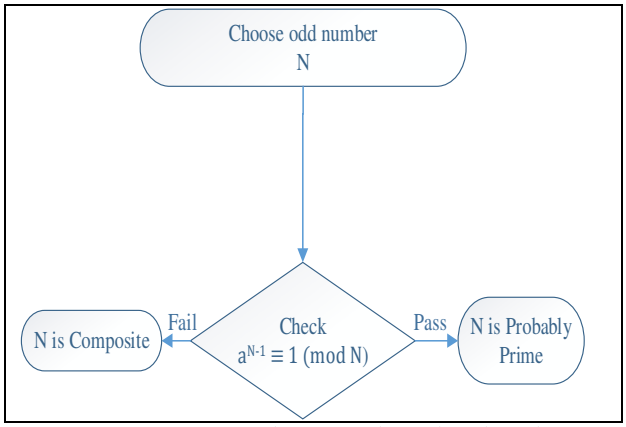

Figure 1. Fermat Primality Test algorithm flow diagram.

Example: show that n=41 is prime with base 2.

$$2^{41-1} \equiv 2^{40}\ mod\ 41 \equiv 1 mod\ 41$$

Thus 41 is most likely a prime number that it is.

Example: check the primality of n=1105 with base 2.

$$2^{1105} \equiv 2^{1104}\ mod\ 1105 \equiv 1 mod\ 1105$$

It declares 1105 to be a prime, however, 1105 is =5×13×17, in other word composite.

In the previous example, Fermat fails to prove the compositeness of a number which is called a Carmichael number. In number theory, a Carmichael number is a composite number in which satisfies the modular arithmetic congruence relation:

$$a^{p-1} \equiv 1 mod\ p \qquad (5)$$

For all *1<b<n* which are relatively prime to n.

Example: check the primality of n=1105 with base 13.

$$13^{1105} \equiv 13^{1104}\ mod\ 1105 \equiv 936 mod\ 1105$$

Thus, composite.

❖ Miller-Rabin Primality Test

Miller-Rabin algorithm (Ishmukhametov and Mubarakov 2013), as demonstrated in Figure 2, is a probabilistic test used to check whether an input number is prime or composite based on the basic principle discussed at the beginning of this chapter. Since it is probabilistic, Miller-Rabin test guarantees the compositeness of a number only and declare a primality with high accuracy. For instance, authors in (Abu Al-Haija 2014b) have used Miller-Rabin test in the FPGA hardware implementation of RSA

Example: check the primality of n=1105 with base 2.

$$n-1 = 560 = 16 \times 35, so\ \ 2^k = 2^4 and\ \ m = 35$$

$$b_0 \equiv 2^{35} \equiv 263\ mod\ 561$$

$$b_1 \equiv b_0^{\ 2} \equiv 166 mod\ 561$$

$$b_2 \equiv b_1^{\ 2} \equiv 67 mod\ 561$$

$$b_3 \equiv b_2^{\ 2} \equiv 1 mod\ 561$$

Thus, 561 is composite. Moreover, *gcd (b_2 - 1, 561) = 33*, which is a nontrivial factor of 561.

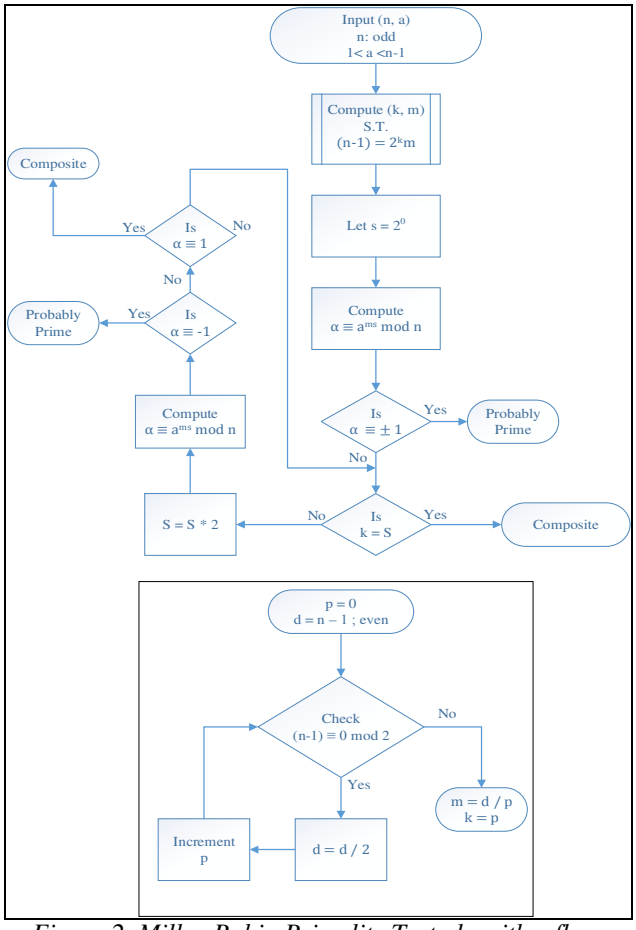

Figure 2. Miller-Rabin Primality Test algorithm flow diagram.

For a given base, strong pseudoprimes are much rarer than pseudoprimes. Up to 10^{10}, there are 455052511 primes; 14884 pseudoprimes for the base 2, and 3291 strong pseudoprimes for the base 2. Therefore, calculating $2^{(n-1)}$ *(mod n)* will fail to recognize a composite in this range with

probability less than 1 out of 30 thousand (Fermat Test), and using the Miller-Rabin test with a = 2 will fail with probability less than 1 out of 100 thousand. It can be shown that the probability that the Miller-Rabin test fails to recognize a composite for a randomly chosen *a* is at most 1/4. In fact, it fails much less frequently than this. If we repeat the test 10 times, say, with randomly chosen values of a, then we expect that the probability of certifying a composite number as prime is at most $(1/4)^{10} = 10^{-6}$

❖ AKS Primality Test

AKS algorithm is the first general unconditional, polynomial, and deterministic primality-proving algorithm. It was published in 2002. It had no practical applications due to the time complexity $O(\log(n)^{12})$ and low performance. However, in 2005 the time complexity was cut into $O(\log(n)^5)$ (Lenstra 2002). However, an improved and modified AKS based tester has been used for RSA design in (Han et al. 2016).
AKS primality test is based on the following theorem:
An integer $n \geq 2$ and $gcd\ (n, a) = 1$, n is prime if and only if:

$$(x + a)^n \equiv x^n + a \bmod n \qquad (6)$$

The complete algorithm is shown below in Figure 3.

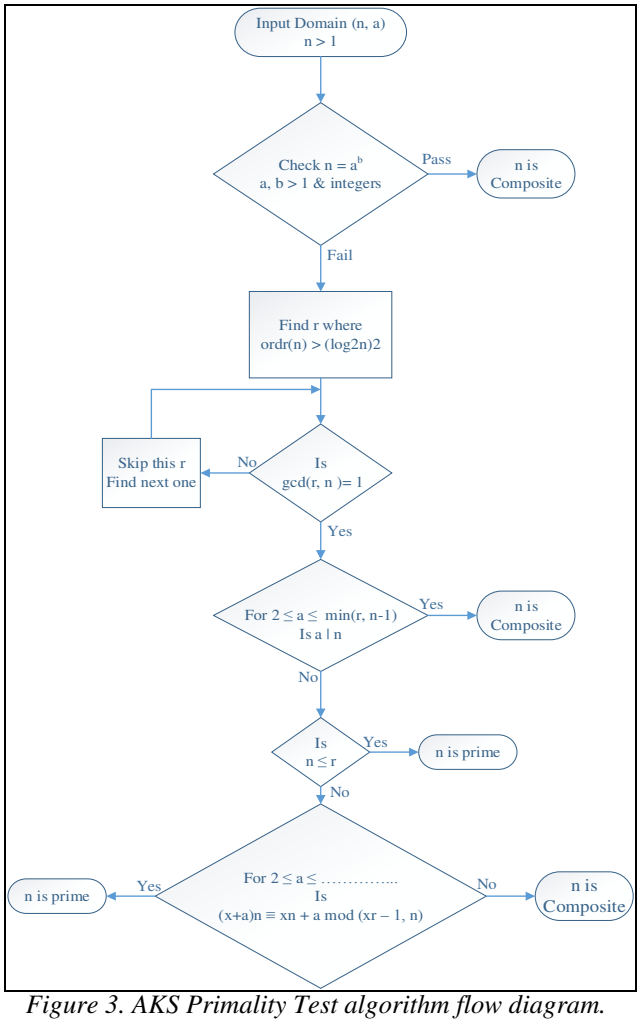

Figure 3. AKS Primality Test algorithm flow diagram.

❖ Run Time Complexity

In the previous sections, we have discussed three well-known primality tests. For comparison purposes, we can briefly report on another two common primality tests such as Solovay–Strassen and Baillie-PSW tests. The Solovay–Strassen primality test (Agrawal 2006) developed by Robert M. Solovay and Volker Strassen, is a probabilistic test to determine if a number is composite or probably prime. It has been largely superseded by the Baillie-PSW primality test and the Miller–Rabin primality test. The Baillie-PSW (BPSW or BSW) primality test (Baillie and Wagstaff 1980) is a compositeness test, in the manner of Fermat's test and the Miller-Rabin test. It is named for Robert Baillie, Carl Pomerance, John L. Selfridge, and Samuel S. Wagstaff, Jr. The algorithm was apparently first conceived by Baillie.
Table 1 below summarizes the run time complexities for five prime test techniques. The table estimates the run time behavior with respect with the number arithmetic operation required by the test to determine whether a given integer is prime or composite. It can be clearly seen that Solovay-Strassen and Miller-Rabin probabilistic primality tests can be used alternatively as they recorded the best run time with logarithmic complexity. However, Solovay-Strassen has been largely superseded by Miller–Rabin primality test (Agrawal 2006) as it facilitates better computation complexity. The worst run time results were belonged to AKS deterministic primality test due to the complex computations involved in the algorithm, which results in lowering the performance results. However, AKS is considered as the most accurate primality test as it depends on non-stochastic computation steps, which result in precise outcomes in every test.

TABLE 1. Fast Primality Testing Algorithms:

Primality Tester	# of arithmetic operations
Fermat Primality Test	O (m log n)
Solovay-Strassen Test	O (log n)
Miller-Rabin	O (log n)
AKS Test	$O(\log^5 n)$
Baillie-PSW primality test	$O((\log n)^3)$

IV. CONCLUSION

Primality test operation is a significant unit of many public key Crypto-processor such as RSA, El-Gamal and Schmidt - Samoa. We reported on different primality testing techniques such as Miller-Rabin, Fermat, AKS, Solovay-Strassen and Baillie-PSW primality tests. We found that implementing the probabilistic Miller-Rabin or Solovay-Strassen test recorded the highest throughput as they minimize the execution time complexity (i.e. logarithmic run time complexity) while deterministic AKS Test required the longest time execution with $O(\log^5 n)$.

REFERENCES

[1] Abu Al-Haija Q., M. Smadi, M. Al-Ja'fari and A. Al-Shua'ibi. 2014. "Efficient FPGA Implementation of RSA Coprocessor Using Scalable Modules". Procedia Computer Science 34: 647–654. Elsevier.
https://doi.org/10.1016/j.procs.2014.07.092
[2] Abu Al-Haija Q., M. Al Tarayrah, H. Al-Qadeeb, and A. Al-Lwaimi. 2014. "A Tiny RSA Cryptosystem based on Arduino

Microcontroller Useful for Small Scale Networks". Procedia Computer Science 34: 639-646.
http://dx.doi.org/10.1016/j.procs.2014.07.091.
[3] Agrawal M. 2006. "Primality Tests Based on Fermat's Little Theorem". Paper presented at International Conference on Distributed Computing and Networking (ICDCN), Lisboa, Portugal, July 4-7.
[4] Baillie R., and S. S. Wagstaff. 1980. "Lucas PseudoPrimes", Mathematics of Computation 35: 1391-1417.
https://doi.org/10.1090/S0025-5718-1980-0583518-6.
[5] Gallier J. 2017. "Notes on Public Key Cryptography and Primality Testing". University of Pennsylvania: Department of Computer and Information Science.
[6] Gubbi J., R. Buyya, S. Marusic, and M. Palaniswami. 2013. "Internet of Things (IoT): A vision, architectural elements, and future directions". Future Generation Computer Systems 29(7):1645-1660.
http://dx.doi.org/10.1016/j.future.2013.01.010.
[7] Han, W. W., Cai, M. L., Hong, L. L., Ding, J., & Yao, X. M. (2016). An RSA scheme based on improved AKS primality testing algorithm. Paper presented at the MATEC Web of Conferences No. 44, 2016. doi: http://dx.doi.org/10.1051/matecconf/20164401032
[8] Ishmukhametov S. and B. Mubarakov.2013. "On practical aspects of the Miller-Rabin Primality Test", Lobachevskii Journal of Mathematics 34(4): 304–312.
[9] Z. I. Borevich and I. R. Shafarevich. 1964. "Number Theory", ACADEMIC Press,
[10] Allan J. Silberger. 1967. "Algebraic Theory of Numbers", Hermann, Paris.