http://dx.doi.org/10.7236/JIIBC.2013.13.3.103
JIIBC 2013-3-14

단계적 소수 판별법

A Step-by-Step Primality Test

이상운*

Sang-Un Lee

요 약 대표적인 소수판별법으로 밀러-라빈 방법이 적용되고 있다. 밀러-라빈 판별법은 카마이클 수 또는 반소수가 합성수임에도 불구하고 소수로 잘못 판별하는 단점이 있어 $m=[2,n-1]$, $(m,n)=1$인 m을 k개 선택하여 소수 여부를 판별한다. 밀러-라빈 방법은 $n-1=2^s d, 0\le r\le s-1$에 대해 $m^d\equiv 1(\mathrm{mod}\, n)$ 또는 $m^{2^r d}\equiv -1(\mathrm{mod}\, n)$로 소수를 판별한다. 본 논문은 $m=2$로 한정시켜 98.9%를 판별할 수 있는 알고리즘을 제안한다. 제안된 방법은 $n=6k\pm 1, n_1\neq 5$로 1차로 합성수 여부를 판별한다. 2차에서는 $2^{2^{s-1}d}\equiv\beta_{s-1}(\mathrm{mod}\, n)$과 $2^d\equiv\beta_0(\mathrm{mod}\, n)$로 판별하였으며, 3차에서는 $\beta_0>1$이면 $1\le r\le s-2$에서 $\beta_r\equiv -1$ 존재 여부로, $\beta_0=1$이면 $m=3,5,7,11,13,17$을 순서대로 적용하였다. 제안된 알고리즘을 $n=[101,1000]$에 적용한 결과 $\beta_0>1$은 26개로 3.0%, $\beta_0=1$은 0.55%만 수행되었으며, 96.55%는 초기에 판별할 수 있었다.

Abstract Miller-Rabin method is the most prevalently used primality test. However, this method mistakenly reports a Carmichael number or semi-prime number as prime (strong lier) although they are composite numbers. To eradicate this problem, it selects k number of m, whose value satisfies the following : $m=[2,n-1]$, $(m,n)=1$. The Miller-Rabin method determines that a given number is prime, given that after the computation of $n-1=2^s d, 0\le r\le s-1$, the outcome satisfies $m^d\equiv 1(\mathrm{mod}\, n)$ or $m^{2^r d}\equiv -1(\mathrm{mod}\, n)$. This paper proposes a step-by-step primality testing algorithm that restricts $m=2$, hence achieving 98.8% probability. The proposed method, as a first step, rejects composite numbers that do not satisfy the equation, $n=6k\pm 1, n_1\neq 5$. Next, it determines prime by computing $2^{2^{s-1}d}\equiv\beta_{s-1}(\mathrm{mod}\, n)$ and $2^d\equiv\beta_0(\mathrm{mod}\, n)$. In the third step, it tests $\beta_r\equiv -1$ in the range of $1\le r\le s-2$ for $\beta_0>1$. In the case of $\beta_0=1$, it retests $m=3,5,7,11,13,17$ sequentially. When applied to $n=[101,1000]$, the proposed algorithm determined 96.55% of prime in the initial stage. The remaining 3% was performed for $\beta_0>1$ and 0.55% for $\beta_0=1$.

Key Words : Prime number, Primality test, Probabilistic primality test, Carmichael number, Semiprime

Ⅰ. 서 론

비대칭 공개키 방식인 RSA의 공개키 n은 합성수(composite number)로 유사하거나 동일한 길이의 2개 소수 (prime number) p,q를 선택하여 곱한 값으로 $n=p\times q$로 결정된다. 또한, 대칭 비밀키 방식인 DES는

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 : 2013년 1월 3일, 수정완료 : 2013년 5월 12일
게재확정일자 : 2013년 6월 14일

Received: 3 January 2013 / Revised: 12 May 2013 / Accepted: 14 June 2013
** Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

소수 p를 사용한다. 따라서 비대칭이나 대칭키 암호를 생성하기 위해서는 선택된 수가 소수인지 여부를 판별하는 소수 판별 (primality test)을 반드시 수행해야 한다.[1,2]

소수 판별법에는 소박한 방법 (naïve PT), 확률적 방법 (probabilistic PT)과 결정론적 방법 (deterministic PT)이 적용되고 있다. 가장 간단한 소박한 방법은 $n \equiv 0 \pmod{\alpha}$, $(2 \le \alpha \le \sqrt{n}$, 홀수$)$의 나눗셈 시행법 (trial division)으로 합성수를 결정한다. 이 방법은 α을 정확히 소수들만으로 구하기 어려워 편의상 홀수만을 대상으로 하고 있다. 확률적 방법은 가장 일반적으로 사용되고 있으며, 페르마 (Fermat), Lucas, Solovay-Strassen, 밀러-라빈 (Miller-Rabin) 방법 등이 있다.[2] Fermat 방법은 다중 소인수로 구성된 카마이클 수 (Carmichael number)[3]를 소수로 잘못 판별하는 단점이 있다. Solovay-Strassen 방법은 2^{-k}, 밀러-라빈 방법은 4^{-k}의 판별 오류 확률을 갖고 있다.

본 논문에서는 $\alpha = 2$로 한정시킨 변형된 밀러-라빈 방법을 제안한다. 2장에서는 주어진 수 n의 소수판별법을 고찰해 본다. 3장에서는 밀러-라빈 방법의 k회 시행을 1회 시행으로 간략화한 소수 판별 알고리즘을 제안하고, 4장에서는 제안된 알고리즘을 합성수, 소수, 카마이클 수에 적용하여 판별력을 검증하여 본다.

II. 소수 판별법

소수 p에 대한 법칙은 "$(n-1)! \equiv -1 \pmod{n}$이면 소수이다."는 윌슨의 정리 (Wilson's theorem)와 "$\alpha^{p-1} \equiv 1 \pmod{p}$이다."는 페르마의 소정리 (Fermat's little theorem)가 있다.

n과 서로소인 정수 개수를 오일러의 totient 함수 $\phi(n)$라 하며, 소수 p의 $\phi(p) = p-1$, 2개 소인수의 곱인 반소수 n의 $\phi(n) = (p-1)(q-1) = (n+1) - (p+q)$이다. 반소수에 대해 $(\alpha, n) = 1$이면 $\alpha^{\phi(n)} \equiv 1 \pmod{n}$이 성립한다.

주어진 수 n의 소수 판별 방법인 소박한 소수 판별 방법 (NPT)은 $n \neq 0 \pmod{\alpha}$으로 소수를 판별하는 방법으로 $\alpha = [2, \sqrt{n}]\,(odd)$ 또는 $\alpha = 6k \pm 1 \le \sqrt{n}\,(odd)$을 적용한다. $\alpha = [2, \sqrt{n}]\,(odd)$ 방법은 9,15, 21,… 등 홀수 합성수를, $\alpha = 6k \pm 1 \le \sqrt{n}\,(odd)$ 방법은 5,7,11,… 소수의 배수를 포함하고 있다. 결국, 소박한 방법은 주어진 수 n 까지 소수를 구하는 고전적인 에라토스테네스의 체 (Sieve of Eratosthenes)[4]를 적용하지 못하고 있다. 소박한 방법으로 최대로 달성할 수 있는 목표는 $2 \le \alpha \le \sqrt{n}$의 소수들을 대상으로 하는 것이며, 더 이상의 성능 개선은 불가능하다.

확률적 방법 (PPT)에는 페르마, Lucas, Solovay-Strassen, 밀러-라빈 방법 등이 있다. 지금 부터는 $\gcd(\alpha, n)$을 줄여서 (α, n)으로, 서로소 (co-prime)를 $(\alpha, n) = 1$로 표기한다. 페르마 방법은 "만약, n이 소수이면 $(\alpha, n) = 1$인 α에 대해 $\alpha^{n-1} \equiv 1 \pmod{n}$이 성립한다."는 Fermat의 소정리[5]에 기반하여 $m = [2, n-1]$을 임의로 k개를 선택하여 판별하는 방법이다. n (홀수)은 소수 또는 합성수이며, 합성수는 반소수 (semi-prime)와 카마이클 수로 구성되어 있다. k를 $1, n$을 제외한 n의 소인수 개수라 하면, 소수의 $k = 0$, 반소수의 $k = 2$, 카마이클의 수는 $k \ge 3$이다. $\alpha = 2$의 경우 반소수도 $2^{n-1} \equiv 1 \pmod{n}$이 되며, $\alpha \ge 3$의 경우 카마이클 수에 대해 $\alpha^{n-1} \equiv 1 \pmod{n}$로 유사소수 (pseudo-prime)로 잘못 판별한다. $\alpha = [2, n-1]$ 대신 $\alpha = 2$로 한정시켜, Poulet number $2^{n-1} \equiv 1 \pmod{n}$으로 판별하는 방법을 Chinese Hypothesis[5]라 한다. 그러나 이 방법은 n이 소수라면 $2^{n-1} \equiv 1 \pmod{n}$은 성립하지만, 역으로 $2^{n-1} \equiv 1 \pmod{n}$이면 n은 소수는 성립하지 않는다. 예로, 반소수 $341 = 11 \times 31$의 합성수는 $2^{340} \equiv 1 \pmod{n}$로 소수로 잘못 판별한다.

Lucas 방법은 $\alpha = [2, n-1]$을 임의로 선택하여 $\alpha^{n-1} \equiv 1 \pmod{n}$과 소인수 $q \in n-1$에 대해 $\alpha^{(n-1)/q} \neq 1 \pmod{n}$ 성립 여부로 판별하며, $n-1$회의 인수 분해를 수행해야 한다. Solovay-Strassen 방법은 홀수 n에 대해 $\alpha = [1, n-1]$에서 임의로 k회 선택하면서 $\alpha^{(n-1)/2} \equiv \left(\frac{\alpha}{p}\right) \pmod{n}$의 Legendre 심볼 $\left(\frac{\alpha}{p}\right)$을 적용하여 소수 여부를 판별한다. 이 알고리즘은 유사소수를 소수로 잘못 판별할 확률이 2^{-k}로 알려져 있으며, $\left(\frac{\alpha}{p}\right)$를 계산하기 쉽지 않다.

소수 p에 대해 p는 홀수로 $p-1$은 짝수이며, $(p-1)/2$인 정수 (홀수 또는 짝수)가 반드시 존재한다. $\alpha^{p-1} \equiv 1 \pmod{p}$와 $\alpha^{(p-1)/2} \equiv \pm 1 \pmod{p}$인 특징이 있다. 밀러-라빈 방법은 이 특징을 적용하여 $\alpha^{(n-1)/2}$까지

만 검증하는 방법이다.

그림 1의 밀러-라빈 방법[6]은 $n-1=2^s d$로 변환시키고, $\alpha \in (Z/nZ)^*$, $d=odd$, $0 \le r \le s-1$에 대해 $\alpha=[2,n-1]$을 k회 선택하면서 $\alpha^d \equiv 1 (\mathrm{mod}\, n)$ 또는 $\alpha^{2^r d} \equiv -1 (\mathrm{mod} n)$의 존재 여부로 소수를 판별한다. 이 알고리즘은 $O(k \cdot \log^2 n)$ 복잡도와 유사소수를 소수로 잘못 판별할 확률이 4^{-k}로 알려져 있다. 그러나 $0 \le r \le s-1$에 대해 모든 값이 1이 되는 소수도 존재한다. 따라서 이를 명확히 하여 "$\alpha^d \equiv \pm 1 (\mathrm{mod}\, n)$ 또는 $1 \le r \le s-1$에 대해 $\alpha^{2^r d} \equiv -1 (\mathrm{mod}\, n)$이 존재하는 경우 n은 소수이다."로 수정되어야만 한다.

n: 소수여부를 검증할 숫자
k: 소수판별법을 몇 회 실행할지 결정하는 인자
$n-1=2^s d.\ d=odd$
loop: repeat k times
  $[2,n-1]$에서 임의의 α 선택.
  $\alpha^d \equiv \beta (\mathrm{mod}\, n)$ 계산.
  if $\beta = \pm 1$ then do next loop
  else if $\beta \neq \pm 1$ then
    for $0 \le r \le s-1$
      /*$\alpha^{2^r d} \equiv \beta_r (\mathrm{mod}\, n)$ 계산.
      $\beta_r \leftarrow (\beta_{r-1})^2 (\mathrm{mod}\, n)$
      if $\beta_r = 1$ then n = **합성수**
      else if $\beta_r = -1 (= n-1)$ then
        do next loop
    return n = **합성수**
  return n = **소수**

그림 1. 밀러-라빈 소수판별법
Fig. 1. Miller-Rabin's Primality Test

$n=341=11\times31$ (반소수)에 대해 밀러-라빈 판별법을 적용할 경우, $340=2^2\times85$로 $s=2, d=85, 0 \le r \le 1$이다. 만약, $m=2$로만 한정시킬 경우, $2^{85} (\mathrm{mod}\, 341)=32$, $2^{170} (\mathrm{mod}\, 341)=1$로 $\alpha^d (\mathrm{mod}\, n) \neq 1$과 $\alpha^{2^r d} (\mathrm{mod}\, n) \neq -1$의 조건을 만족시키지 못해 소수로 판별한다. $\alpha=3$인 경우 $3^{85} (\mathrm{mod}\, 341)=48$, $3^{170} (\mathrm{mod}\, 341)=-1$이 되며, $\alpha=4,5,6,7,\cdots$로 설정하면 $\alpha^d (\mathrm{mod}\, n) \neq 1$로 합성수로 판별한다. 따라서 α을 k회 수행해야 소수 여부를 판별할 수 있다. 따라서 확률적 방법에서 $\alpha=[2,n-1]$에서 임의로 k개를 선택하는 이유는 α 값에 따라 합성수 여부를 판별하지 못할 수도 있기 때문이다.

Pomerance et al[7]과 Jaeschke[8]은 Miller-Rabin 방법의 $\alpha=[2,n-1]$을 대략 $\alpha=[2,17]$로 한정시켜도 15자리 수까지는 판별이 가능함을 보였다.

결정론적 방법에는 AKS, Elliptic 곡선 방법, Pocklington 방법 등이 있다. AKS 방법은 $(x-\alpha)^n \equiv (x^n-\alpha)\ (\mathrm{mod}\ n), (\alpha,n)=1, (0<k<n)$에 기반하여 $n=\alpha^b, (m>0,\ b>1)$, $1<\gcd(\alpha,n)<n$과 $(x+\alpha)^n \neq x^n=\alpha (\mathrm{mod} x^r-1,n)$이면 합성수로 결정하며, 다항시간인 $O(\log^6 n)$ 복잡도로 소수 여부를 판별한다. Elliptic 곡선 방법은 Elliptic 곡선에 기반한 방법으로 특정 공식이 없다. 이 방법은 휴리스틱하게 $O((\log n)^{5+\epsilon}), \epsilon>0$으로 판별할 수 있으며, 최악의 경우 복잡도는 알려져 있지 않다. 이와는 다른 분야인 주어진 합성수 n을 두 개의 소수 p,q로 효율적으로 소인수분해하는 방법에는 Lee와 Choi[8]이 있다.

결론적으로 소수판별법의 성능은 3개 이상의 소인수들로 합성된 카마이클 수 (carmichael number)를 어떻게 합성수로 판별하는가에 달려 있다.

Ⅲ. 단계적 소수 판별 알고리즘

일반적으로 $(\alpha,n)=1$이면 $\alpha^{\phi(n)} \equiv 1 (\mathrm{mod}\, n)$이다. 소수 p의 오일러 함수는 $\phi(p)=p-1$로 $\alpha^{n-1} \equiv 1 (\mathrm{mod}\, n)$이다. 제안 알고리즘은 밀러-라빈 방법과 동일하게 $n-1=2^s d, (d=odd)$, $0 \le r \le s-1$에 대해 $\beta_r=\alpha^{2^r d} (\mathrm{mod}\, n)$의 값에 기반하고 있다. 밀러-라빈 방법은 k개의 α에 대해 $\alpha^d \equiv \beta_0 (\mathrm{mod}\, n)$의 모듈러 지수연산을 수행하고 $1 \le r \le s-1$에 대해서는 $\beta_r=(\beta_{r-1})^2 (\mathrm{mod}\, n)$의 제곱법을 적용한다. 따라서 모듈러 지수연산법을 k회 수행해야 한다.

$\alpha=2$로 한정시키면 어떠한 소수 $p>2$에 대해서도 $(\alpha,p)=1$로 $2^{(p-1)} \equiv 1 (\mathrm{mod}\, p)$, $2^{(p-1)/2} \equiv \pm 1 (\mathrm{mod}\, p)$이 되기 때문에 RSA에서 적용하고 있는 2개 소인수를 갖는 합성수 (반소수) n, $\phi(n)=(n+1)-(p+q)$을 합성수로 판별할 수 있다. 그러나 3개 이상의 소인수로 구성된 합성수인 카마이클 수는 이 공식이 적용되지 않을 수 도 있다.

$2^{(n-1)/2} \equiv 1 (\mathrm{mod}\, n)$은 모든 소수, 카마이클 수와 더불어 일부 반소수에서도 발생한다. 특이하게도 $2^{(n-1)/2} \equiv -1 (\mathrm{mod}\, n)$은 소수에서만 발생한다. 또한, 소수인 경우 $0 \le r \le s-1$에 대해 $\beta_r \equiv -1$이 존재하거나 $\forall_r, \beta_r \equiv 1$이 된다. 이러한 특성을 일부 카마이클 수도 갖고 있다. 본 장에서는 이러한 특성을 판별할 수 있는 알고리즘을 제안한다.

제안된 방법은 사전에 $n \neq 6k \pm 1, (n_1 = 1,3,7,9)$로 많은 합성수를 제거한다. 다음으로, $\alpha = 2$로 한정시켜 $2^d \equiv \beta_0 (\mathrm{mod}\, n)$과 $2^{2^{s-1}d} \equiv \beta_{s-1} (\mathrm{mod}\, n)$의 2개에 대한 모듈러 지수연산을 수행하여 대부분의 합성수를 제거한다. 만약, $\beta_{s-1} = 1$과 $\beta_0 > 1$인 n에 대해서는 $1 \le r \le s-2$에 대해 $\beta_r = (\beta_{r-1})^2 (\mathrm{mod}\, n)$의 제곱법을 적용하여 $\beta_r = -1$이 발생하면 소수로, $\beta_r = -1$이 없으면 합성수로 판별한다. 또한, $\beta_{s-1} = \beta_0 = 1$인 n에 대해서만 $\alpha = [3,17]$의 소수를 반복 적용해 카마이클수를 검출한다. 본장에서 제안하는 방법은 그림 2에 제시되어 있다.

제안된 방법은, 밀러-라빈 방법과 다음과 같은 차이점이 있다. 제안된 알고리즘을 "단계적 소수판별법"이라 하자.

(1) 밀러-라빈 방법은 n에 대해 합성수 여부를 사전 검증을 하지 않는다. 반면에, 제안된 알고리즘은 $n = 6k \pm 1, (n_1 \neq 5)$로 2,3과 5의 배수인 합성수를 사전에 제외시킨다.

(2) 밀러-라빈 방법은 α을 k개 선택하는데 반해, 제안된 방법은 $\beta_{s-1} = \beta_0 = 1$인 경우를 제외하면 $\alpha = 2$로 $k = 1$이다.

(3) 밀러-라빈 방법은 $n-1 = 2^s d,\ 0 \le r \le s-1$에 대해 s회 수행하는데 반해, 제안된 알고리즘은 $\beta_r = -1$을 찾으면 알고리즘을 종료한다.

(4) 밀러-라빈 방법은 $\beta_0 \equiv -1$은 소수, $\beta_{s-1} \neq \pm 1$이면 합성수임에도 불구하고 s회 수행한다. 반면에 제안된 방법은 이 부류를 1회만에 검증할 수 있다.

(5) 제안된 방법은 $s \ge 2$인 $\beta_{s-1} = 1$, $\beta_0 > 1$에 대해서는 $1 \le r \le s-2$ 범위에서 $\beta_r = -1$이 발생하면 소수로 판정한다. 또한, $s-1 \ge 1$인 $\beta_{s-1} = 1$, $\beta_0 = 1, (s-1) \ge 1$에 대해서는 $\alpha = [3,17]$의 소수를 반복 적용해 카마이클수를 검출한다.

Step 1. /* 1차 검증
if $n_1 \neq 1,3,7,9$ (n의 1의 자리수) then
$n =$ 합성수, 알고리즘 종료.
else if $n \neq 6k \pm 1$ then $n =$ 합성수, 알고리즘 종료.
/* $n+1 \equiv 0 (\mathrm{mod}\ 6)$ 또는 $n-1 \equiv 0 (\mathrm{mod}\ 6)$이면 $n = 6k \pm 1$임.
else if $n = 6k \pm 1 (n_1 \neq 5)$ then Step 2 수행.

$n-1 = 2^s d,\ \alpha = 2.$
Step 2. /* $0 \le r \le s-1$의 하한 0와 상한 $s-1$ 검증.
if $s = 0$ then $2^d \equiv \beta_0 (\mathrm{mod}\, n)$ 계산
if $\beta_0 = \pm 1$ then $n =$ 소수, 알고리즘 종료
else if $\beta_0 > 1$ then $n =$ 합성수, 알고리즘 종료
else if $s \ge 1$ then $2^d \equiv \beta_0 (\mathrm{mod}\, n)$, $2^{(n-1)/2} \equiv \beta_{s-1} (\mathrm{mod}\, n)$ 계산
if $\beta_0 = -1$ or $\beta_{s-1} = -1$ then
$n =$ 소수, 알고리즘 종료
else if $\beta_{s-1} > 1$ then $n =$ 합성수, 알고리즘 종료
else if $s = 1$, $\beta_0 = 1$ then $n =$ 소수, 알고리즘 종료
else if $s \ge 2$ and $\beta_0 = 1$ then Step 3 수행
else if $\beta_{s-1} = 1$, $\beta_0 > 1$ then Step 4 수행

Step 3. $\alpha = 3,5,7,11,13,17$을 순서대로 대입, Step 2 수행

Step 4. /* $1 \le r \le s-2$ 범위에서 $\exists \beta_r = -1$ 검증
for $1 \le r \le s-2$
$\beta_r \leftarrow (\beta_{r-1})^2 (\mathrm{mod}\, n)$ 계산
if $\beta_r = -1$ then $n =$ **소수,** exit
else if $\beta_r = 1$ then $n =$ **합성수,** exit
else if $\beta_r > 1$ then $r = r+1$, do loop
end
if $r = s-2$ and $\beta_r > 1$ then $n =$ **합성수.**

그림 2. 단계적 소수판별법
Fig. 2. Step-by-Step Primality Test

Ⅳ. 실험 및 결과 분석

제안 알고리즘의 적합성을 검증하기 위해 $n = [101, 10000]$을 대상으로 비교하여 보자. $n = [101, 1000]$의 900개 수 중에서 소수는 홀수이므로 대상은 450개이다. 홀수 450개 중에서 사전처리 단계로 $n = 6k \pm 1, (n_1 = 1,3,7 \text{ or } 9)$를 만족하는 수는 239개로, 211개는 제외된다. 239개 수 중에서 합성수 97개는 표 1에, 소수 142개는 표 2에 제시되어 있다.

표 1의 합성수 97개 중에서 $2^{2^{s-1}d} \equiv 1 (\mathrm{mod}\, n)$은 $n = 341$의 1개만 존재한다. 따라서, $2^{2^{s-1}d} \neq 1 (\mathrm{mod}\, n)$인 96개 합성수는 1회의 모듈러 지수연산에서 판별될 수 있다. 나머지 $n = 341$에 대해서는 $1 \le r \le s-2$인 170에

서 $32^2 (\mathrm{mod}\, 341) = 1$이 계산되어 $\beta_r \equiv -1 (= n-1)$이 존재하지 않아 합성수로 판별되었다.

표 1. 합성수의 $\alpha = 2$, $2^{2^r d} (\mathrm{mod}\, n)$

Table 1. $\alpha = 2$, $2^{2^r d} (\mathrm{mod}\, n)$ of Composite Number

n	$2^{2^r d} \equiv \beta_r (\mathrm{mod}\, n)$, $2^{2^r} d(\beta_r)$, $s-1 \geq r \geq 0$
119	**59(25)**
121	**60(89)**-30(45)-15(98)
133	**66(106)**-33(50)
143	**71(46)**
161	**80(123)**-40(128)-20(144)-10(58)-5(32)
169	**84(105)**-42(116)-21(31)
187	**93(151)**
203	**101(137)**
209	**104(82)**-52(81)-26(9)-13(41)
217	**108(8)**-54(78)-27(190)
221	**110(30)**-55(128)
247	**123(164)**
253	**126(9)**-63(118)
259	**129(29)**
287	**143(172)**
289	**144(256)**-72(273)-36(152)-18(21)-9(223)
299	**149(110)**
301	**150(78)**-78(204)
319	**159(171)**
323	**161(257)**
329	**164(102)**-82(296)-41(25)
341	**170(1)**-85(32)
343	**171(57)**
361	**180(210)**-90(286)-45(37)
371	**185(151)**
377	**188(139)**-94(270)-47(345)
391	**195(348)**
403	**201(343)**
407	**203(338)**
413	**206(228)**-103(72)
427	**213(358)**
437	**218(213)**-109(173)
451	**225(32)**
469	**234(260)**-117(295)
473	**236(97)**-118(322)-59(94)
481	**240(417)**-120(248)-60(417)-30(233)-15(60)
493	**246(353)**-123(76)
497	**248(221)**-124(450)-62(109)-31(324)
511	**255(8)**
517	**258(267)**-129(28)
527	**263(349)**
529	**264(461)**-132(231)-66(116)-33(323)
533	**266(433)**-133(197)
539	**269(193)**
551	**275(184)**
553	**276(8)**-138(176)-69(526)
559	**279(151)**
581	**290(158)**-145(184)
583	**291(233)**
589	**294(140)**-147(407)
611	**305(487)**
623	**311(186)**
629	**314(225)**-157(15)
637	**318(155)**-159(372)
649	**324(27)**-162(26)-81(519)
667	**333(336)**
671	**335(395)**
679	**339(8)**
689	**344(360)**-172(81)-86(433)-43(50)
697	**348(543)**-174(353)-87(128)
703	**351(265)**
707	**353(396)**
713	**356(591)**-178(349)-89(140)
721	**360(8)**-180(694)-90(442)-45(169)
731	**365(389)**
737	**368(102)**-184(456)-92(257)-46(317)-23(74)
749	**374(634)**-187(366)
763	**381(428)**
767	**383(644)**
779	**389(471)**
781	**390(529)**-195(758)
791	**395(347)**
793	**396(729)**-198(454)-99(281)
799	**99(162)**
803	**401(178)**-204(729)-102(790)-51(297)
817	**408(391)**-208(86)-104(603)-52(730)-26(718)
833	**416(732)**-13(695)
841	**420(436)**-210(202)-105(336)
847	**423(701)**
851	**425(542)**
869	**434(269)**-217(194)
871	**435(606)**
889	**444(8)**-222(540)-111(64)
893	**446(747)**-223(394)
899	**449(698)**
901	**450(327)**-225(427)
913	**456(383)**-228(36)-114(907)-57(117)
917	**458(123)**-229(842)
923	**461(916)**
931	**465(449)**
943	**471(121)**
949	**474(64)**-237(811)
959	**479(830)**
961	**480(94)**-240(528)-120(745)-60(373)-30(187)-15(94)
973	**486(687)**-243(456)
979	**489(655)**
989	**494(403)**-247(469)

표 2의 소수 142개 중에서 $\beta_{s-1} = \beta_0 \equiv -1 (\mathrm{mod}\, n)$이 계산된 (a)의 76개는 1회의 모듈러 지수연산에서 소수로 판별되었다. 나머지 66개 소수에 대해서는 $\beta_{s-1} = \beta_0 \equiv 1 (\mathrm{mod}\, n), (s=1)$인 (b)의 26개는 소수로 판별되었다. 마지막으로 남은 31개 중에서$\beta_{s-1} = 1, \beta_0 > 1 (\mathrm{mod}\, n)$인 (c)의 25개에 대해서는 $2^{2^r d} \equiv \beta_r (\mathrm{mod}\, n), 1 \leq r \leq s-2$로 $\beta_r \equiv -1$ 존재 여부로 소수로 판별되었다. $\beta_{s-1} = \beta_0 = 1, (s > 2)$인 (d)의 5개에 대해서만 $\alpha = 3, 5, 7, 11, 13, 17$로 표 3과 같이 재 수행되었다. $\alpha = 3$에서 $n = 233, 337, 881, 937$이 소수로 판별되었으며, $n = 601$은 $\alpha = 5$에서 소수로 판별되었다.

표 4는 카마이클 수에 대해 제안된 알고리즘을 적용한 사례이다. 15개의 카마이클 수에 제안된 알고리즘을 적용한 결과 $\alpha = 2$에서 14개는 합성수로 판별하였으며, $n = 15841$에 대해서만 $\alpha = 2$에서 판별하지 못하고 $\alpha = 3$에서 판별할 수 있었다.

$n = [101, 1000]$의 900개 수에 제안된 알고리즘을 적용한 결과 Step 1의 $n = 6k \pm 1, (n_1 \neq 5)$를 만족하는 239개에 대해서만 알고리즘이 수행되었으며, Step 2의 $\alpha = 2, r = 0$ and $r = s-1$에서 판별되지 못한 31개 중 26개는 $\alpha = 2, 1 \leq r \leq s-2$에서, 나머지 5개는 $\beta_0 = 1, (s > 2)$로 $\alpha = 3$에서 4개, $\alpha = 5$에서 1개가 판별되었다.

결국, 제안된 알고리즘을 적용할 경우 $\alpha = 2, 1 \leq r \leq s-2$를 수행하는 비율은 $27/900 \times 100 = 3.0\%$이며, $\alpha = 3$은 $4/900 \times 100 = 0.44\%$, $\alpha = 5$는 $1/900 \times$

$100=0.11\%$로 $\alpha=3$ 이상을 수행하는 비율은 0.55%로 미미함을 알 수 있다.

본 장에서는 제안된 알고리즘이 밀러-라빈 알고리즘의 소수로 잘못 판별할 오류율 4^{-k}과 비교하지는 못하였다. 왜냐하면 어떠한 수에 대해 밀러-라빈 알고리즘이 판별 오류를 발생시키는지 알려져 있지 않기 때문에 카마이클 수에 대해 판별을 수행하였다. 또한, 밀러-라빈 방법에 비해 얼마나 빨리 판별을 할 수 있는지 여부에 초점을 두었다. 임의의 수 n을 선택하였을 때 제안된 알고리즘은 $n-1=2^s d$, $\alpha=2, r=0, r=s-1$에 대한 $2^{2^r d}(\bmod n)$ 계산만으로 96.55%를 판별하였으며, $\alpha=2, 1\le r\le s-2$를 수행할 비율은 3.0%, $\alpha=3,5,7,11,13,17$에 대해 불만족 수행할 비율은 0.55%에 불과함을 보였다.

표 2. 소수의 $\alpha=2$, $2^{2^r d}(\bmod n)$

Table 2. $\alpha=2$, $2^{2^r d}(\bmod n)$ of Prime Number

$2^{2^r d}\equiv\beta_r(\bmod n),\ 2^{2^r}d(\beta_r),\ s-1\ge r\ge 0$			
n	$s-1\ge r\ge 0$	n	$s-1\ge r\ge 0$
101	**50(-1)**-25(10)	523	**261(-1)**
107	**53(-1)**	541	**270(-1)**-135(52)
109	**54(-1)**-27(33)	547	**273(-1)**
131	**65(-1)**	557	**278(-1)**-139(118)
139	**69(-1)**	563	**281(-1)**
149	**74(-1)**-37(105)	571	**285(-1)**
157	**78(-1)**-39(129)	587	**293(-1)**
163	**81(-1)**	613	**306(-1)**-153(578)
173	**86(-1)**-43(80)	617	**308(1)**-154(1)-**77(-1)**
179	**89(-1)**	619	**309(-1)**
181	**90(-1)**-45(162)	643	**321(-1)**
197	**98(-1)**-49(183)	653	**326(-1)**-163(149)
211	**105(-1)**	659	**329(-1)**
227	**113(-1)**	661	**330(-1)**-165(555)
229	**114(-1)**-57(122)	677	**338(-1)**-169(26)
251	**125(-1)**	683	**341(-1)**
269	**134(-1)**-67(187)	691	**345(-1)**
277	**138(-1)**-69(60)	701	**350(-1)**-175(566)
281	**140(1)**-70(1)-**35(-1)**	709	**354(-1)**-177(96)
283	**141(-1)**	733	**366(-1)**-183(380)
293	**146(-1)**-73(138)	739	**369(-1)**
307	**153(-1)**	757	**378(-1)**-189(87)
317	**158(-1)**-79(203)	773	**386(-1)**-193(317)
331	**165(-1)**	787	**393(-1)**
347	**173(-1)**	797	**398(-1)**-199(215)
349	**174(-1)**-87(213)	811	**405(-1)**
373	**186(-1)**-93(104)	821	**410(-1)**-205(295)
379	**189(-1)**	827	**413(-1)**
389	**194(-1)**-97(115)	829	**414(-1)**-207(583)
397	**198(-1)**-99(63)	853	**426(-1)**-213(333)
419	**209(-1)**	859	**429(-1)**
421	**210(-1)**-105(29)	877	**438(-1)**-219(151)
443	**221(-1)**	883	**441(-1)**
461	**230(-1)**-115(48)	907	**453(-1)**
467	**233(-1)**	941	**470(-1)**-235(844)
491	**245(-1)**	947	**473(-1)**
499	**249(-1)**	971	**485(-1)**
509	**254(-1)**-127(301)	997	**498(-1)**-249(161)

(a) $\beta_{s-1}=1$ or $\beta_0=-1$

$2^{2^r d}\equiv\beta_r(\bmod n),\ 2^{2^r}d(\beta_r),\ s-1\ge r\ge 0$					
n	β_r	n	β_r	n	β_r
103	**51(1)**	367	**183(1)**	719	**359(1)**
127	**63(1)**	383	**191(1)**	727	**363(1)**
151	**75(1)**	431	**215(1)**	743	**371(1)**
167	**83(1)**	439	**219(1)**	751	**375(1)**
191	**95(1)**	463	**231(1)**	823	**411(1)**
199	**99(1)**	479	**239(1)**	839	**419(1)**
223	**111(1)**	487	**243(1)**	863	**431(1)**
239	**119(1)**	503	**251(1)**	887	**443(1)**
263	**131(1)**	599	**299(1)**	911	**455(1)**
271	**135(1)**	607	**303(1)**	919	**459(1)**
311	**155(1)**	631	**309(1)**	967	**483(1)**
359	**179(1)**	647	**323(1)**	983	**491(1)**

(b) $\beta_0=1, (s=1)$

n	$2^{2^r d}\equiv\beta_r(\bmod n),\ 2^{2^r}d(\beta_r),\ s-1\ge r\ge 0$
113	**56(1)**-28(1)-**14(-1)**-7(15)
137	**68(1)**-**34(-1)**-17(100)
193	**96(1)**-**48(-1)**-24(112)-12(43)-6(64)-3(8)
241	**120(1)**-**60(-1)**-30(64)-15(233)
257	**128(1)**-64(1)-32(1)-16(1)-**8(-1)**-4(16)-2(4)-1(2)
313	**156(1)**-**78(-1)**-39(25)
353	**176(1)**-88(1)-**44(-1)**-22(311)-11(283)
401	**200(1)**-**100(-1)**-50(20)-25(356)
409	**204(1)**-**102(-1)**-51(143)
433	**216(1)**-**108(-1)**-54(254)
449	**224(1)**-**112(-1)**-56(382)-28(357)-14(220)-7(128)
457	**228(1)**-**114(-1)**-57(348)
521	**260(1)**-**130(-1)**-65(286)

n	$2^{2^r d}\equiv\beta_r(\bmod n),\ 2^{2^r}d(\beta_r),\ s-1\ge r\ge 0$
569	**284(1)**-**142(-1)**-71(86)
577	**288(1)**-144(1)-**72(-1)**-36(553)-18(186)-9(512)
593	**296(1)**-148(1)-**74(-1)**-37(516)
641	**320(1)**-**160(-1)**-80(154)-40(385)-20(541)-10(383)-5(32)
673	**336(1)**-**168(-1)**-84(615)-42(326)-21(84)
761	**380(1)**-**190(-1)**-95(722)
769	**384(1)**-**192(-1)**-96(707)-48(173)-24(712)-12(251)-6(64)-3(8)
809	**404(1)**-**202(-1)**-101(491)
857	**428(1)**-**214(-1)**-107(650)
929	**464(1)**-**232(-1)**-116(605)-58(258)-29(883)
953	**476(1)**-**238(-1)**-119(442)
977	**488(1)**-**244(-1)**-122(252)-61(439)

(c) $\beta_0>1, (s>2)$

n	$2^{2^r d}\equiv\beta_r(\bmod n),\ 2^{2^r}d(\beta_r),\ s-1\ge r\ge 0$
233	**116(1)**-58(1)-**29(1)**
337	**168(1)**-84(1)-42(1)-**21(1)**
601	**300(1)**-150(1)-**75(1)**
881	**440(1)**-220(1)-110(1)-**55(1)**
937	**466(1)**-234(1)-**117(1)**

(d) $\beta_0=1, (s>2)$

표 3. $\beta_0=1, (s>2)$의 $\alpha=3,5,7,11,13,17$

Table 3. $\alpha=3,5,7,11,13,17$ of $\beta_0=1, (s>2)$

n	α	$\alpha^{2^r d}\equiv\beta_r(\bmod n),\ \alpha^{2^r}d(\beta_r),\ s-1\ge r\ge 0$
233	**3**	**116(-1)**-58(144)-**29(221)**
337	**3**	**168(1)**-**84(-1)**-42(148)-**21(252)**
601	**3**	**300(1)**-150(1)-**75(1)**
	5	300(1)-**150(-1)**-75(125)
881	**3**	**440(-1)**-220(387)-110(662)-**55(767)**
937	**3**	**468(1)**-234(1)-**117(-1)**

표 4. 카마이클수의 $\alpha=2,\ 2^{2^r d}(\mathrm{mod}\, n)$

Table 4. $\alpha=2,\ 2^{2^r d}(\mathrm{mod}\, n)$ of Carmichael Number

n	α	$\alpha^{2^r d}\equiv\beta_r(\mathrm{mod}\, n),\ \alpha^{2^r}d(\beta_r),\ s-1\geq r\geq 0$
561	2	280(1)-**140(67)**-70(166)-35(263)
1729	2	864(1)-432(1)-216(1)-**108(1)**-54(1065)-27(645)
2821	2	**1410(1520)**-705(2605)
6601	2	3300(1)-**1650(4509)**-825(2738)
8911	2	**4455(6364)**
15841	2	7920(1)-3960(1)-1980(1)-990(1)-**495(1)**
	3	7920(1)-3960(1)-**1980(1)**-990(218)-**495(12802)**
29341	2	**14670(29340)**-7335(26424)
41041	2	20520(1)-**10260(1)**-5130(27182)-2565(27994)
63973	2	**31986(6252)**-15993(512)
75361	2	37680(1)-**18840(1)**-9420(39898)-4710(73657)-2355(15036)
101101	2	**50550(51206)**-25275(38774)
126217	2	63108(1)-**31554(9710)**-15777(512)
188461	2	**94230(52845)**-47115(159566)
340561	2	170280(1)-**85140(140232)**-42570(274299)-21285(244951)
1721081	2	860540(1)-**430270(1)**-215135(1721080)

V. 결론

본 논문은 $\alpha=2$로 한정시킨 변형된 밀러-라빈 소수 판별법을 제안하였다. 제안된 알고리즘은 단계적 판별 과정을 거치는 특징이 있다. 1차 검증 단계에서 $n=6k\pm1, n_1\neq 5$만을 대상으로 한다. 2차 검증에서는 $r=0$와 $r=s-1$의 $2^{2^r d}(\mathrm{mod}\, n)$의 모듈러 지수연산으로 소수인지 합성수인지 여부를 대부분 판별하였다. 3차 검증에서는 $\beta_0>1, \beta_0=2^d(\mathrm{mod} n)$에 대해서는 $1\leq r\leq s-2$에 대해 $\beta_r\equiv -1$ 존재 여부로 소수인지 판별하였으며, $\beta_0=1$에 대해서는 $\alpha=3,5,7,11,13,17$을 순서대로 적용하는 방법이다.

제안된 알고리즘을 $n=[101,1000]$의 900개 수에 적용한 결과 26개는 $\alpha=2, 1\leq r\leq s-2$에서 27개 (3.0%)를, $\beta_0=1,(s>2)$의 5개 (0.55%)중 4개는 $\alpha=3$에서 나머지 1개는 $\alpha=5$에서 판별할 수 있었다. 또한, 카마이클 수 15개에 적용한 결과 α=2에서 14개를, $\alpha=3$에서 1개를 판별할 수 있었다.

Reference

[1] D. Zagier, "Newman's Short Proof of the Prime Number Theorem," American. Mathematical. Monthly, Vol. 104, No. 8, pp. 705 - 708, 1997.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," 2nd Ed., MIT Press and McGraw-Hill. pp. 887 - 896, 2001.

[3] R. D. Carmichael, "On composite numbers P which satisfy the Fermat congruence $a^{P-1}\equiv 1 \mod P$," American Mathematical Monthly, Vol. 19, No. 2, pp. 22-27, 1912.

[4] M. E. O'Neill, "The Genuine Sieve of Eratosthenes," Journal of Functional Programming, Cambridge University Press, 2008.

[5] P. Ribenboim, "The New Book of Prime Number Records (3rd ed.)," New York: Springer-Verlag, 1995.

[6] M. O. Rabin, "Probabilistic algorithm for testing primality," Journal of Number Theory, Vol. 12, No. 1, pp. 128 - 138, 1980.

[7] C. Pomerance, J. L. Selfridge, and S. S. Wagstaff, "The Pseudoprimes to 25·109", Mathematics of Computation, Vol.35, No. 151, pp. 1003-1026, 1980.

[8] G. Jaeschke, "On strong pseudoprimes to several bases," Mathematics of Computation, Vol. 61, No. 204, pp. 915 - 926, 1993.

[9] S. U. Lee and M. B. Choi, "The Integer Factorization Method Based on Congruence of Squares," Journal of Korean Institute of Information Technology, Vol. 12, No. 5, pp. 185-189, Oct. 2012.

저자 소개

이 상 운(정회원)

- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007년 2월 : 국립 원주대학 여성교양과 조교수
- 2007년 3월 ~ 현재 : 강릉원주대학교 멀티미디어공학과 부교수

<관심분야 :소프트웨어 프로젝트 관리, 개발 방법론, 소프트웨어 척도, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 신경망, 뉴로-퍼지, 그래프 알고리즘>

- E-mail : sulee@gwnu.ac.kr