

소수 판별법

이상운*, 최명복**

The Primality Test

Sang-Un, Lee *, Myeong-Bok, Choi **

요약

대표적인 소수판별법으로 밀러-라빈 방법이 적용되고 있다. 밀러-라빈 판별법은 $m = [2, n-1]$ 에서 m 을 k 개 선택하여 $n-1 = 2^s d, 0 \leq r \leq s-1$ 에 대해 $m^d \equiv 1 \pmod{n}$ 또는 $m^{2^r d} \equiv -1 \pmod{n}$ 로 소수를 판별하여 $k \times r$ 회를 수행한다. 본 논문은 $c = p^{\frac{n-1}{2}} \pmod{n}$ 을 계산하여 $c = -1$ 이면 소수로 판별하여 k 회 수행하였다. 제안된 판별법은 밀러-라빈 판별법의 $k \times r$ 회를 k 회로 감소시켰다.

▶ 키워드 : 소수, 합성수, 소수 판별, 확률적 판별법, 카마이클 수, 반소수

Abstract

Generally, Miller-Rabin method has been the most popular primality test. This method arbitrary selects m at k -times from $m = [2, n-1]$ range and $(m, n) = 1$. Miller-Rabin method performs $k \times r$ times and reports prime as $m^d \equiv 1 \pmod{n}$ or $m^{2^r d} \equiv -1 \pmod{n}$ such that $n-1 = 2^s d, 0 \leq r \leq s-1$. This paper suggests more simple primality test than Miller-Rabin method. This test method computes $c = p^{\frac{n-1}{2}} \pmod{n}$ for k times and reports prime as $c = -1$. The proposed primality test method reduces $k \times s$ times of Miller-Rabin method to k times.

▶ Keywords : prime number, composite number, primality test, probabilistic primality test

• 제1저자 : 이상운*, 교신저자 : 최명복**

• 투고일 : 2011. 03. 11, 심사일 : 2011. 04. 11, 게재확정일 : 2011. 05. 05.

* ** 강릉원주대학교 멀티미디어공학과 (Dept. of Multimedia Science, Gangneung-Wonju National University)

I. 서론

RSA 암호체계 (cryptograph)의 암호키 (encryption key, e)와 해독키 (decryption key, d)는 유사하거나 동일한 길이의 2개 소수 (prime number) p, q 를 선택하여 곱한 합성수 (composite number)로 $n = p \times q$ 로부터 생성된다. 이 경우, 선택한 수 p, q 는 반드시 소수이어야 하며, 소수인지 여부를 결정하는 방법이 소수 판별 (primality test)이다[1-5].

선택한 수 p, q 가 소수인지 여부를 판별하는 소수 판별법에는 소박한 방법 (naive PT), 확률적 방법 (probabilistic PT)과 결정론적 방법 (deterministic PT)의 있으며, 확률적 방법이 일반적으로 적용되고 있다[5].

확률적 방법에는 페르마 (Fermat), Lucas, Solovay-Strassen, 밀러-라빈 (Miller-Rabin) 방법 등이 있다[2]. 페르마 방법은 $m = [2, n-1]$, $(m, n) = 1$ 에 대해 m 을 k 개 선택하여 $m^{n-1} \equiv 1 \pmod{n}$ 로 소수를 판별하며, 밀러-라빈 방법은 $n-1 = 2^s d$ ($d = \text{odd}$), $0 \leq r \leq s-1$ 에 대해 $m = [2, n-1]$ 을 k 회 선택하면서 $m^d \equiv 1 \pmod{n}$ 또는 $m^{2^r d} \equiv -1 \pmod{n}$ 로 소수를 판별한다. 밀러-라빈 방법은 $k \times r$ 회를 수행함에도 불구하고 4^{-k} 의 판별 오류 확률을 갖고 있다.

본 논문에서는 $m = 2, 3$ 을 선택하고, $r = s-1$ 로 한정시킨 변형된 밀러-라빈 방법을 제안한다. 따라서 제안된 방법은 밀러-라빈 방법의 $k \times r$ 회를 k 회로 단축시킬 수 있다. 2장에서는 확률적 소수판별법을 고찰해 본다. 3장에서는 $m = 2, 3$ 을 선택하고, $r = s-1$ 로 한정시킨 간단한 소수 판별 알고리즘을 제안한다. 4장에서는 합성수 (반소수, 카마이클 수)와 소수에 적용하여 적합성을 검증한다.

II. 관련연구

두 수 m 과 n 의 최대공약수 $\text{gcd}(m, n)$ 을 (m, n) 으로, 서로소 (co-prime)를 $(m, n) = 1$ 로 표기하자.

n 과 서로소인 정수의 개수를 오일러의 totient 함수 $\phi(n)$ 이라 한다. 이에 따르면 소수 p 의 $\phi(p) = p-1$, 반소수 n 의 $\phi(n) = (p-1)(q-1)$ 이다.

어떤 선택된 수가 소수가 되기 위해서는 소수 p 에 대한 페르마의 소정리 (Fermat's little theorem)인 $m^{p-1} \equiv 1 \pmod{p}$ 을 만족해야 한다. 따라서 $(m, n) = 1$ 이면 $m^{\phi(n)} \equiv 1 \pmod{n}$ 이다.

예로, 소수 $n = 19$ 의 $\phi(19) = 18$ 로, $m = [1, 18]$ 범위

의 어떤 수를 적용하여도 $m^{18} \equiv 1 \pmod{19}$ 이다. $7^{222} \pmod{10}$ 을 구하기 위해서는 합성수 $n = 10(2 \times 5)$, $(7, 10) = 1$ 로 $7^{\phi(10)} \equiv 1 \pmod{10}$, $\phi(10) = (2-1)(5-1) = \{1, 3, 7, 9\} = 4$ 을 적용하면, $7^{222} = 7^{4 \times 55 + 2} = (1)^{55} (7^2)$ 로 $7^2 \pmod{10} = 9$ 를 얻는다.

그러나 RSA와 같이 큰 자리수 공개키 (반소수) $n = p \times q$ 에 대해 선택된 p, q 는 $\frac{n}{2}$ 자리수의 큰 수이므로 p, q 가 소수인지 여부를 검증하는 것 또한 어렵다. 예로, RSA-100 (10진수 100자리수)의 경우 p 와 q 는 각각 10진수 50자리수이다[6].

페르마 방법은 “만약, n 이 소수이면 $(n, m) = 1$ 인 m 에 대해 $m^{n-1} \equiv 1 \pmod{n}$ 이 성립한다.”는 페르마의 소정리에 기반하여 $[2, n-1]$ 범위에서 m 을 임의로 k 개를 선택하여 판별하는 방법이다.[7] 예로, $m = 2, n = 341(11 \times 31)$ 인 경우, $(m, n) = 1, 2^{340} \equiv 1 \pmod{n}$ 이 되어 소수로 잘못 판별한다.

Solovay-Strassen 방법은 홀수 n 에 대해 $[1, n-1]$ 범위에서 m 을 임의로 k 회 선택하면서 $m^{(n-1)/2} \equiv (m/n) \pmod{n}$ 으로 소수 여부를 판별한다. 이 알고리즘은 유사소수를 소수로 잘못 판별할 확률이 2^{-k} 로 알려져 있다.

그림 1의 밀러-라빈 방법은 $n-1 = 2^s d$ ($d = \text{odd}$)로 변환시키고, $m = [2, n-1]$ 에서 n 에 대한 곱셈군 $m \in (Z/nZ)^*$ 을 k 회 선택하면서 $0 \leq r \leq s-1$ 에 대해 $m^d \equiv 1 \pmod{n}$ 또는 $m^{2^r d} \equiv -1 \pmod{n}$ 의 존재 여부로 소수를 판별한다[4]. 이 알고리즘은 $O(k \cdot \log^2 n)$ 복잡도와 유사소수를 소수로 잘못 판별할 확률이 4^{-k} 로 알려져 있다. 즉, $r = 0, m^d \equiv \pm 1 \pmod{n}$ 또는 $1 \leq r \leq s-1, m^{2^r d} \equiv -1 \pmod{n}$ 이 존재하는 경우 n 은 소수이다.

```

k: 소수판별법 실행횟수
n-1 = 2^s d로 변환. d = odd
for i = 1 to k
    [2, n-1]에서 임의의 m 선택.
    0 ≤ r ≤ s-1의 모든 r에 대해 m^{2^r d} (mod n) 계산.
    if m^d (mod n) ≠ 1 and m^{2^r d} (mod n) ≠ -1 then
        n = 합성수
    else n = 소수.
    
```

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

만약, $m^{n-1} \equiv 1 \pmod{n}$ 이면 $m^{(n-1)/2} \equiv 1 \pmod{n}$ 도

성립하는 특징이 있다. 왜냐하면 n 은 홀수 합성수이기 때문에 $n-1$ 은 짝수로 $(n-1)/2$ 도 존재하기 때문이다. 밀러-라빈은 이 특징을 적용하여 페르마 방법의 m^{n-1} 은 생략하고 $m^{(n-1)/2}$ 까지만 검증하는 방법을 적용하였다. $n=561$ 인 경우 밀러-라빈 방법을 적용하면, $n-1=560(2^4 \times 35)$ 로 $s=4, d=35$ 이다. $0 \leq r \leq s-1$ 이므로 $r=0,1,2,3$ 이 되어 $m^{35}, m^{70}, m^{140}, m^{280}$ 을 검증한다.

$n=341(11 \times 31)$ (반소수)에 대해 밀러-라빈 판별법을 적용할 경우, $340=2^2 \times 85$ 로 $s=2, d=85, 0 \leq r \leq 1$ 이다. 만약, $m=2$ 로 한정시킬 경우, $s=2$ 회인 $2^{85} \pmod{341} = 32, 2^{170} \pmod{341} = 1$ 로 $m^d \pmod{n} \neq 1$ 은 만족시키지만 $m^{2^r d} \pmod{n} \neq -1$ 을 만족시키지 못해 합성수임에도 불구하고 소수로 판별한다. 따라서, m 을 k 회 선택한다.

III. 소수 판별법

일반적으로 공개키 n 을 결정하기 위해, 첫 번째로 p, q 는 소수일 것 같은 유사소수 (pseudo-prime)를 선택한다. 이들 값은 $p=6k \pm 1, (k=1, 2, \dots)$ 형태이며, 1의 자리수 p_1 은 1, 3, 7 또는 9를 취하고 있다. 즉, 2, 3과 5의 배수는 제외된다.

지금부터는 선택된 수 p, q 를 소수인지 여부를 검증하므로 n 으로 표기한다. 따라서 사전에 선택된 수가 $p=6k \pm 1, (p_1 \neq 5)$ 인지 검증한다.

두 번째로, 페르마의 소정리에 따르면 $(m, n) = 1$ 이면 $m^{\phi(n)} \equiv 1 \pmod{n}$ 이다. 만약, n 이 소수라면 오일러 함수는 $\phi(n) = n-1$ 로 $m^{n-1} \equiv 1 \pmod{n}$ 이다. 또한, $n-1=2^s d, (d \text{ odd}), r=s-1$ 인 $m^{2^r d} \equiv \pm 1 \pmod{n}$ 이 성립한다. 따라서 밀러-라빈의 $0 \leq r \leq s-1$ 대신 $r=s-1$ 인 $m^{\frac{n-1}{2}} \pmod{n}$ 만을 적용한다.

세 번째로, n 이 합성수 또는 소수와 상관없이 $(m, n) = 1$ 이 되려면 m 은 소수이어야 한다. 밀러-라빈 방법에서 주어진 수 n 에 대해 곱셈군 $m \in (Z/nZ)^*$ 을 정확히 선택하기 위해 오일러 함수 $\phi(n) = (p-1)(q-1)$ 또는 $\phi(n) = n-1$ 을 고려해야 하며, 이는 추가적으로 소인수분해를 해야만 한다. 여기에서는 밀러-라빈 방법의 $m = [2, n-1]$ 의 곱셈군 $m \in (Z/nZ)^*$ 을 대상으로 하지 않고 소수로 한정시킨다.

즉, m 대신 p 를 사용하여 $p^{\frac{n-1}{2}} \pmod{n}$ 를 계산한다.

제안된 방법은 그림 2에 제시되어 있으며, “소수에 의한 소수 판별법”이라 칭하자.

```

n = 6k ± 1, (n1 = 1, 3, 7 or 9)
c = 0.
while c = -1 or c > 1
    c = p^(n-1/2) (mod n), (p = 2, 3, 5, ...) 계산
    if c > 1 then n = 합성수
    else if c = -1 then n = 소수
    else if c = 1 then Next p 선택.
    
```

그림 2 소수에 의한 소수판별법
Fig. 2 Primality Test by Prime Number

제안된 방법은 밀러-라빈 방법과 다음과 같은 차이점이 있다.

- (1) 밀러-라빈 방법은 n 에 대해 합성수 여부를 사전에 검증하지 않는다. 반면에 제안 알고리즘은 $n=6k \pm 1, (n_1 = 1, 3, 7 \text{ or } 9)$ 로 2, 3과 5의 배수를 사전에 제거한다. 이는 1의 자리수가 5가 아니면서 $n \pm 1 \equiv 0 \pmod{6}$ 인 수들이다.
- (2) 밀러-라빈 방법은 m 을 n 에 대한 곱셈군 $m \in (Z/nZ)^*$ 에서 k 개 선택하는데 반해, 제안된 방법은 $p=2, 3, 5, \dots$ 의 소수들을 적용한다.
- (3) 밀러-라빈 방법은 하나의 m 에 대해 $n-1=2^s d, 0 \leq r \leq s-1$ 에 대해 r 회의 $m^{2^r d} \pmod{n}$ 을 계산하는데 반해, 제안된 알고리즘은 $r=s-1$ 인 $p^{\frac{n-1}{2}} \pmod{n}$ 의 1회만 계산한다.

IV. 실험 및 결과 분석

제안된 알고리즘의 적합성을 검증하기 위해 $n=[101, 1000]$ 을 대상으로 검증하여 보자. 실험 데이터 900개 중에서 $n=6k \pm 1, (n_1 \neq 5)$ 을 만족하는 합성수 97개에 제안된 판별법을 적용한 결과는 표 1에, 소수 142개에 대해서는 표 2에 제시되어 있다.

합성수 97개에 대해 $p=2$ 인 경우 96개는 합성수로 판별되어 $p=3$ 이 수행되지 않았으며, 단지 $n=341$ 만이 $p=3$ 이 수행되고 $c > 1$ 로 합성수로 판별되었다. 소수 142개 중에서 $c=-1$ 이 발생한 현황을 보면 $p=2$ 에서 74개, $p=3$ 에서 37개, $p=5$ 에서 18개, $p=7$ 에서 8개, $p=11$ 에서 4개, $p=13$ 에서 1개가 발생하였다. 결국, 합성수는 98.97%가 1회 수행으로 판별되며, 소수는 1회에서는 52.11%, 2회까지는

78.17%, 3회까지는 90.84%, 4회까지는 96.48%, 5회까지는 99.29%가 판별된다.

표 1. 합성수의 $c = p^{(n-1)/2} \pmod n$

Table 1. $c = p^{(n-1)/2} \pmod n$ for Composite numbers

| n | p | | n | p | |
|-----|-----|----|-----|-----|---|
| | 2 | 3 | | 2 | 3 |
| 119 | 25 | | 589 | 140 | |
| 121 | 89 | | 611 | 487 | |
| 133 | 106 | | 623 | 186 | |
| 143 | 46 | | 629 | 225 | |
| 161 | 123 | | 637 | 155 | |
| 169 | 105 | | 649 | 27 | |
| 187 | 151 | | 667 | 336 | |
| 203 | 137 | | 671 | 395 | |
| 209 | 82 | | 679 | 8 | |
| 217 | 8 | | 689 | 360 | |
| 221 | 30 | | 697 | 543 | |
| 247 | 164 | | 703 | 265 | |
| 253 | 9 | | 707 | 396 | |
| 259 | 29 | | 713 | 591 | |
| 287 | 172 | | 721 | 8 | |
| 289 | 256 | | 731 | 389 | |
| 299 | 110 | | 737 | 102 | |
| 301 | 78 | | 749 | 634 | |
| 319 | 171 | | 763 | 428 | |
| 323 | 257 | | 767 | 644 | |
| 329 | 102 | | 779 | 471 | |
| 341 | 1 | 67 | 781 | 529 | |
| 343 | 57 | | 791 | 347 | |
| 361 | 210 | | 793 | 729 | |
| 371 | 151 | | 799 | 162 | |
| 377 | 139 | | 803 | 178 | |
| 391 | 348 | | 817 | 391 | |
| 403 | 343 | | 833 | 732 | |
| 407 | 338 | | 841 | 436 | |
| 413 | 228 | | 847 | 701 | |
| 427 | 358 | | 851 | 542 | |
| 437 | 213 | | 869 | 269 | |
| 451 | 32 | | 871 | 606 | |
| 469 | 260 | | 889 | 8 | |
| 473 | 97 | | 893 | 747 | |
| 481 | 417 | | 899 | 698 | |
| 493 | 353 | | 901 | 327 | |
| 497 | 221 | | 913 | 383 | |
| 511 | 8 | | 917 | 123 | |
| 517 | 267 | | 923 | 916 | |
| 527 | 349 | | 931 | 449 | |
| 529 | 461 | | 943 | 121 | |
| 533 | 433 | | 949 | 64 | |
| 539 | 193 | | 959 | 830 | |
| 551 | 184 | | 961 | 94 | |
| 553 | 8 | | 973 | 687 | |
| 559 | 151 | | 979 | 655 | |
| 581 | 158 | | 989 | 403 | |
| 583 | 233 | | | | |

표 2. 소수의 $c = p^{(n-1)/2} \pmod n$

Table 2. $c = p^{(n-1)/2} \pmod n$ for Prime numbers

| n | p | | | n | p | | |
|-----|----|---|---|-----|----|---|---|
| | 2 | 3 | 5 | | 2 | 3 | 5 |
| 101 | -1 | | | 523 | -1 | | |
| 107 | -1 | | | 541 | -1 | | |
| 109 | -1 | | | 547 | -1 | | |
| 131 | -1 | | | 557 | -1 | | |
| 139 | -1 | | | 563 | -1 | | |
| 149 | -1 | | | 571 | -1 | | |
| 157 | -1 | | | 587 | -1 | | |
| 163 | -1 | | | 613 | -1 | | |
| 173 | -1 | | | 619 | -1 | | |
| 179 | -1 | | | 643 | -1 | | |
| 181 | -1 | | | 653 | -1 | | |
| 197 | -1 | | | 659 | -1 | | |
| 211 | -1 | | | 661 | -1 | | |
| 227 | -1 | | | 677 | -1 | | |
| 229 | -1 | | | 683 | -1 | | |
| 251 | -1 | | | 691 | -1 | | |
| 269 | -1 | | | 701 | -1 | | |
| 277 | -1 | | | 709 | -1 | | |
| 283 | -1 | | | 733 | -1 | | |
| 283 | -1 | | | 739 | -1 | | |
| 307 | -1 | | | 757 | -1 | | |
| 317 | -1 | | | 773 | -1 | | |
| 331 | -1 | | | 787 | -1 | | |
| 347 | -1 | | | 797 | -1 | | |
| 349 | -1 | | | 811 | -1 | | |
| 373 | -1 | | | 821 | -1 | | |
| 379 | -1 | | | 827 | -1 | | |
| 389 | -1 | | | 829 | -1 | | |
| 397 | -1 | | | 853 | -1 | | |
| 419 | -1 | | | 859 | -1 | | |
| 421 | -1 | | | 877 | -1 | | |
| 443 | -1 | | | 883 | -1 | | |
| 461 | -1 | | | 907 | -1 | | |
| 467 | -1 | | | 941 | -1 | | |
| 491 | -1 | | | 947 | -1 | | |
| 499 | -1 | | | 971 | -1 | | |
| 509 | -1 | | | 997 | -1 | | |

| n | p | | | | | |
|-----|---|----|----|----|----|----|
| | 2 | 3 | 5 | 7 | 11 | 13 |
| 103 | 1 | -1 | | | | |
| 113 | 1 | -1 | | | | |
| 127 | 1 | -1 | | | | |
| 137 | 1 | -1 | | | | |
| 151 | 1 | -1 | | | | |
| 167 | 1 | 1 | -1 | | | |
| 191 | 1 | 1 | 1 | -1 | | |
| 193 | 1 | 1 | -1 | | | |
| 199 | 1 | -1 | | | | |
| 223 | 1 | -1 | | | | |
| 233 | 1 | -1 | | | | |
| 239 | 1 | 1 | 1 | 1 | -1 | |
| 241 | 1 | 1 | -1 | | | |
| 257 | 1 | -1 | | | | |
| 263 | 1 | 1 | -1 | | | |
| 271 | 1 | -1 | | | | |
| 281 | 1 | -1 | | | | |
| 311 | 1 | 1 | 1 | 1 | -1 | |
| 313 | 1 | 1 | -1 | | | |
| 337 | 1 | 1 | -1 | | | |
| 353 | 1 | -1 | | | | |
| 359 | 1 | 1 | 1 | -1 | | |
| 367 | 1 | -1 | | | | |
| 383 | 1 | 1 | -1 | | | |
| 401 | 1 | -1 | | | | |
| 409 | 1 | 1 | 1 | -1 | | |
| 431 | 1 | 1 | 1 | -1 | | |
| 433 | 1 | 1 | -1 | | | |
| 439 | 1 | -1 | | | | |
| 449 | 1 | -1 | | | | |
| 457 | 1 | 1 | -1 | | | |
| 463 | 1 | -1 | | | | |

| n | p | | | | | |
|-----|---|----|----|----|----|----|
| | 2 | 3 | 5 | 7 | 11 | 13 |
| 479 | 1 | 1 | 1 | 1 | 1 | -1 |
| 487 | 1 | -1 | | | | |
| 503 | 1 | 1 | -1 | | | |
| 521 | 1 | -1 | | | | |
| 569 | 1 | -1 | | | | |
| 577 | 1 | 1 | -1 | | | |
| 593 | 1 | -1 | | | | |
| 599 | 1 | 1 | 1 | -1 | | |
| 601 | 1 | 1 | 1 | -1 | | |
| 607 | 1 | -1 | | | | |
| 617 | 1 | -1 | | | | |
| 631 | 1 | -1 | | | | |
| 641 | 1 | -1 | | | | |
| 647 | 1 | 1 | -1 | | | |
| 673 | 1 | 1 | -1 | | | |
| 719 | 1 | 1 | 1 | 1 | -1 | |
| 727 | 1 | -1 | | | | |
| 743 | 1 | 1 | -1 | | | |
| 751 | 1 | -1 | | | | |
| 761 | 1 | -1 | | | | |
| 769 | 1 | 1 | 1 | -1 | | |
| 809 | 1 | -1 | | | | |
| 823 | 1 | -1 | | | | |
| 839 | 1 | 1 | 1 | 1 | -1 | |
| 857 | 1 | -1 | | | | |
| 863 | 1 | 1 | -1 | | | |
| 881 | 1 | -1 | | | | |
| 887 | 1 | 1 | -1 | -1 | | |
| 911 | 1 | 1 | 1 | -1 | | |
| 919 | 1 | -1 | | | | |
| 929 | 1 | -1 | | | | |
| 937 | 1 | 1 | -1 | | | |
| 953 | 1 | -1 | | | | |
| 967 | 1 | -1 | | | | |
| 977 | 1 | -1 | | | | |
| 983 | 1 | 1 | -1 | | | |

추가적으로 3개 이상의 소수들 곱으로 구성된 합성수인 카마이클 수 (Carmichael number)에 대해 제안된 알고리즘을 적용하여 보자. 카마이클 수는 561, 1105, 1729, 2465, 2821, 6601, 8911, 41041, ... 등이 있다. 예로 $561 = 3 \times 11 \times 17$, $41041 = 7 \times 11 \times 13 \times 41$ 이다. 이들 수 중 [561, 8911]에서 $6k \pm 1, (n_1 \neq 5)$ 인 1729, 2821, 6601, 8911에 적용한 결과 표 3과 같이 $c = -1$ 이 존재하지 않아 모두 합성수로 판별된다.

표 3. 카마이클 수의 $c = p^{(n-1)/2} \pmod n$

Table 3. $c = p^{(n-1)/2} \pmod n$ for Carmichael Number

| n | p | | | | | |
|------|------|---|---|------|----|----|
| | 2 | 3 | 5 | 7 | 11 | 13 |
| 1729 | 1 | 1 | 1 | 742 | | |
| 2821 | 1520 | | | | | |
| 6601 | 1 | 1 | 1 | 1680 | | |
| 8911 | 6364 | | | | | |

소수에 의한 판별법과 밀러-라빈 판별법을 비교한 결과는 표 4에 제시되어 있다. 밀러-라빈 판별법과의 가장 큰 차이점은 m 대신 p 를 적용하는 점과 $0 \leq r \leq s-1$ 대신 $r = s-1$ 만을 계산한다는 점이다.

표 4. 알고리즘 비교

Table 4. Comparison of Algorithms

| 구분 | 밀러-라빈 판별법 | 소수에 의한 소수 판별법 |
|-------|---------------------------------|---|
| $n-1$ | $n-1 = 2^s d$ | $(n-1)/2$ |
| m | $[2, n-1]$ (k 개) | $p = 2, 3, 5, \dots$ |
| r | $0 \leq r \leq s-1$ (s 개) | $r = s-1$ (1 개) |
| 수행횟수 | $k \times s$ 회 | k 회 |
| 사전검증 | 없음 | $n = 6k \pm 1, (n_1 = 1, 3, 7 \text{ or } 9)$ 검증 |

V. 결론

대표적인 소수판별법인 밀러-라빈 방법은 $n-1 = 2^s d$ ($d = \text{odd}$)로 변환시키고, $m = [2, n-1]$ 에서 n 에 대한 곱셈군 $m \in (Z/nZ)^*$ 을 k 회 선택하면서 $0 \leq r \leq s-1$ 에 대해 $m^d \equiv 1 \pmod n$ 또는 $m^{2^r d} \equiv -1 \pmod n$ 의 존재 여부로 소수를 판별한다.

본 논문은 $p^{\frac{n-1}{2}} \pmod n$ 법을 제안하였다. 제안 방법은 사전 검증 단계에서 $n = 6k \pm 1, (n_1 \neq 5)$ 를 만족하는 수인

지 검증한다. 사전 검증단계를 통과한 수에 대해 $c = p^{\frac{n-1}{2}} \pmod n$ 를 계산한다. 여기서 $p = 2, 3, 5, \dots$ 의 소수이다. 만약, $c > 1$ 이면 n 은 합성수, $c = -1$ 이면 n 은 소수로 판별한다. 만약, $c = 1$ 이면 $c > 1$ 또는 $c = -1$ 이 나올 때까지 p 를 증가시킨다.

밀러-라빈 판별법은 $k \times s$ 회를 수행하는데 반해 제안된 알고리즘은 k 회로 단축시킬 수 있었다.

참고문헌

- [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] Wikipedia, "RSA," <http://en.wikipedia.org/wiki/Rsa>, 2010.
- [4] M. O. Rabin, "Probabilistic algorithm for testing primality," Journal of Number Theory, Vol. 12, No. 1, pp. 128 - 138, 1980.
- [5] N. Kayal and N. Saxena, "Towards a Deterministic Polynom

ial-Time Test.” Technical Report. Kanpur, India: Indian Institute of Technology, 2002.

[6] Wikipedia, “RSA number,” http://en.wikipedia.org/wiki/Rsa_number, 2010.



최 명 복 (Myeong-Bok, Choi)

1992년 : 호서대학교 전자계산학과 (학사)

1994년 : 아주대학교 컴퓨터공학과 (석사)

2001년 : 아주대학교 컴퓨터공학과 (박사)

1997~현재 : 강릉원주대학교 멀티 미디어공학과 (교수)

2004. 1~현재 : 한국인터넷방송통신 학회 이사

관심분야 : 지능형 정보검색, 퍼지응용, 지식표현, 신경망, 지능형 교통제어, 소프트웨어 공학, 알고리즘

e-mail : cmb5859@gmail.com, cmb1@gwnu.ac.kr

저 자 소 개



이 상 운 (Sang-Un, Lee)

1983년~1987년 : 한국항공대학교 항공 전자공학과 (학사)

1995년 ~ 1997년 : 경상대학교 컴퓨터공학과 (석사)

1998년 ~ 2001년 : 경상대학교 컴퓨터공학과 (박사)

2003년 : 강원도립대학 컴퓨터응용과 전임강사

2004년 ~ 2007.2 : 국립 원주대학 여성 교양과 조교수

2007.3 ~ 현재 : 강릉원주대학교 과학기술대학 멀티미디어공학과 부교수

관심분야 : 소프트웨어 프로젝트 관리, 소프트웨어 개발 방법론, 소프트웨어 척도 (소프트웨어 규모, 개발노력, 개발기간, 팀 규모), 분석과 설계 방법론, 소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성, 신경망, 뉴로-퍼지, 그래프 알고리즘

e-mail : sulce@gwnu.ac.kr