ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 19-02-15

j.inst.Korean.electr.electron.eng.Vol.23,No.2,438~447,June 2019
http://dx.doi.org/10.7471/ikeee.2019.23.2.438


# 오일러체를 적용한 소수와 안전소수의 생성법 제안과 분석

# Proposal and Analysis of Primality and Safe Primality test using Sieve of Euler

조 호 성*, 이 지 호**, 박 희 진**★

Hosung Jo*, Jiho Lee**, Heejin Park**★

## Abstract

As the IoT-based hyper-connected society grows, public-key cryptosystem such as RSA is frequently used for encryption, authentication, and digital signature. Public-key cryptosystem use very large (safe) prime numbers to ensure security against malicious attacks. Even though the performance of the device has greatly improved, the generation of a large (safe)prime is time-consuming or memory-intensive. In this paper, we propose ET-MR and ET-MR-MR using Euler sieve so it runs faster while using less memory. We present a running time prediction model by probabilistic analysis and compare time and memory of our method with conventional methods. Experimental results show that the difference between the expected running time and the measured running time is less than 4%. In addition, the fastest running time of ET-MR is 36% faster than that of TD-MR, 8.5% faster than that of DT-MR and the fastest running time of ET-MR-MR is 65.3% faster than that of TD-MR-MR and similar to that of DT-MR-MR. When $k$=12,381, the memory usage of ET-MR is 2.7 times more than that of DT-MR but 98.5% less than that of TD-MR and when $k$=65,536, the memory usage of ET-MR-MR is 98.48% less than that of TD-MR-MR and 92.8% less than that of DT-MR-MR.

## 요 약

IoT 기반의 초연결사회가 되어감에 따라 암호, 인증, 전자서명 등을 위해 RSA와 같은 공개키암호시스템이 빈번하게 사용되고 있다. 공개키암호시스템은 악의적인 공격으로부터 보안성을 확보하기 위해 크기가 매우 큰 (안전)소수를 사용하는데 기기의 성능이 크게 발전하였음에도 불구하고 크기가 큰 (안전)소수생성은 수행시간이 오래 걸리거나 메모리를 많이 요구하는 작업이다. 본 논문에서는 수행시간과 사용공간의 효율을 높이기 위해 오일러체(Euler sieve)를 사용하는 ET-MR 소수검사법과 ET-MR-MR 안전소수검사법을 제안한다. 제안한 검사법을 확률적으로 분석한 수행시간 예측 모델을 제안하고 기존 방법들과 수행시간, 메모리 사용량을 비교하였다. 실험결과, 이론적 예측시간과 실제 수행시간의 차이는 거의 없었으며(4%미만) 각 알고리즘이 가장 빠를 때의 수행시간을 비교하면 ET-MR이 TD-MR보다 34.5%, DT-MR보다 8.5% 더 빨랐으며, ET-MR-MR이 TD-MR-MR보다 65.3% 더 빨랐고, DT-MR-MR과는 비슷하였다. 공간의 경우 $k$=12,381일 때 ET-MR이 DT-MR보다 약 2.7배 더 사용했지만 TD-MR보다 98.5% 더 적게 사용하였고 $k$=65,536일 때 ET-MR-MR이 TD-MR-MR보다 98.4%, DT-MR-MR보다 92.8% 더 적게 사용하였다.

*Key words : Security, RSA, Prime, Safe-prime, Probabilistic analysis, Sieve of Euler*

* Institute of software convergence, Hanyang University
** Department of Computer Science, Hanyang University
★ Corresponding author
E-mail : hjpark@hanyang.ac.kr, Tel : +82-2-2220-1986
※ Acknowledgement
This work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.
Manuscript received Jun. 6, 2019; revised Jun. 17, 2019; accepted Jun. 21, 2019.

## Ⅰ. 서론

스마트시티, 스마트홈, 5G, AI, IoT 등 최신 트렌드와 기술을 기반으로 한 초연결사회로 진입함에 따라 네트워크에 연결된 수많은 객체들은 상호간의 암복호(encryption/decryption), 인증(authentication), 전자서명(digital signature) 등을 빈번히 사용하고 있다.[1] 특히 IoT, 스마트 모바일 디바이스 등이 널리 사용됨에 따라 개인정보의 활용과 보호에 대한 관심이 높아지면서 기존의 ID/PW방식의 정보기반 인증 외에도 안면인식, 지문인식 등의 생체인증(biometrics)이나 기지국, AP의 정보를 이용하는 위치인증(Location-based authentication) 등을 사용하거나 이들을 2개 이상 사용하는 멀티팩터 인증(Multi factor authentication)이 이용되고 있다.[2]

따라서 이종 간 장치들이나 객체들 간의 보안통신을 위해 사용되는 보안키의 개수가 급격히 증가하고 있다. 이에 따라 보안키 유출이나 해킹에 의한 보안 문제가 지속적으로 발생하고 있으며 이를 해결하기 위해 공개키암호시스템의 사용빈도는 점점 더 높아지고 있다.

공개키암호시스템[3][4]은 악의적인 공격으로부터 보안성을 확보하기 위해 크기가 매우 큰 (안전)소수를 사용하는데 하드웨어와 소프트웨어의 성능이 크게 발전하였음에도 불구하고 크기가 큰 (안전)소수를 생성하는 것은 오랜 시간이 걸린다. 이에 따라 (안전)소수생성에서 수행시간이 가장 오래 걸리는 (안전)소수검사의 시간을 줄이기 위한 연구가 진행되고 있다.

본 논문에서는 기존의 (안전)소수생성방법을 개선한 ET-MR 소수검사법과 ET-MR-MR 안전소수검사법을 제안하였다. 이 방법들은 오일러체를 이용하여 나눗셈의 횟수와 중복되는 메모리 접근 횟수를 줄여 기존 방법보다 효율적으로 작동할 수 있다. 본 논문의 구성은 다음과 같다. 먼저 2장에서는 (안전)소수 생성 관련 연구를 소개하고 3장에서는 오일러체를 이용하는 제안방법과 수행시간의 확률분석을 소개한다. 4장에서는 실험을 통해 확률분석의 모델의 정확성을 보인 다음, 기존방법과 제안방법의 성능비교를 수행한다. 5장에서 결론을 맺고 향후 연구방향을 제안한다.

## Ⅱ. 본론

### 1. 기존 연구

가. 소수검사법과 안전소수검사법

공개키암호시스템에서 가장 널리 사용되는 RSA 암호 알고리즘은 두 개의 (안전)소수를 곱하여 공개키와 개인키로 사용한다. (안전)소수를 이용하여 공개키와 개인키를 생성하면 악의적인 공격자가 공개키를 소인수분해를 하여 어떤 수가 사용되었는지 알 수는 있다. 하지만, 크기가 매우 큰 수를 소인수분해하는 것은 매우 오랜 시간이 걸리기 때문에 RSA암호 알고리즘에 사용된 (안전)소수를 알아내는 것은 매우 어렵다.

소수를 생성하기 위해서는 $n$-bit 난수 $r$을 생성한 다음 소수검사법을 수행하여 검사를 통과하면 $r$을 소수로 판단한다. 안전소수를 생성하기 위해서는 $n$-bit 난수 $r$을 생성한 다음 $r$과 $(r-1)/2$를 소수검사법으로 검사한다. $r$과 $(r-1)/2$가 모두 소수라고 확인되면 $r$을 안전소수로 판단한다.

이 과정에서 $n$-bit 난수 $r$을 생성하는데 걸리는 난수생성시간보다 생성된 난수가 소수인지를 검사하는 소수검사시간이 매우 크기 때문에 (안전)소수를 빠르게 생성하기 위해서는 효율적인 소수검사법에 대한 연구가 필요하다.

난수 $r$이 소수인지 판단하는 대표적인 검사방법으로는 Trial Division, Miller-Rabin test, Fermat test, 나눗셈 테이블(division table)[5][6][7][8] 등이 있다. 일반적으로 소수검사의 속도를 향상시키기 위해 두 가지 이상의 검사법을 조합해서 사용한다. 대표적인 조합소수검사법에는 Trial Division과 Miller-Rabin test방식을 조합한 TD-MR검사법과 GCD함수와 Miller-Rabin test를 조합한 GCD-MR검사법, 그리고 나눗셈테이블 방식과 Miller-Rabin test를 조합한 DT-MR검사법등이 있다.[9]

나. TD-MR검사법과 TD-MR-MR검사법

TD-MR검사법은 trial division(이하 TD)과 Miller-Rabin test(이하 MR test)를 조합하여 소수를 검사하는 조합소수검사방법으로 상세한 수행과정은 다음과 같다.

1. 난수 생성
   - $n$-bit 난수 $r$을 무작위로 생성
2. Trial Division 수행
   - $r$을 $k$개의 작은 소수로 나눔
   - 한 번이라도 나누어진다면 1로 이동하고 아니라면 3으로 이동
3. MR test 수행
   - $r$을 Miller-Rabin test로 검사
   - 검사를 통과하면 $r$을 소수로 반환하고, 아니라면 1로 이동

Maurer[10]는 $k$개의 소수를 사용하는 TD-MR검사법의 수행시간을 확률적으로 분석하였다. $k$개의 소수를 사용하는 TD-MR검사법의 수행시간은 난수생성시간인 $T_{rnd}$, $TD$의 수행시간인 $T_{TD}$, MRtest의 수행시간인 $T_{MR}$의 합으로 다음과 같다.

$$T = N(T_{rnd} + T_{TD} + T_{MR}) \tag{1}$$

먼저, $N$은 $n$-bit 소수를 한 개를 생성할 때까지 생성해야하는 난수 $r$의 개수로 $n\ln 2/2$이고 $T_{rnd}$는 난수생성시간으로 $r$을 한 개 생성하는데 걸리는 시간이다.

$TD$는 $r$을 소수 $p_j$ ($p_1 < p_2 < ... < p_j < ... < p_k$, $p_1 = 3$)로 나누어보고 나누어지지 않으면 그 다음 소수인 $p_{j+1}$로 나누어보는 작업을 모든 $j$에 대해 수행한다. $T_d$를 나눗셈을 한번 수행하는데 걸리는 시간이라고 하면 $TD$의 수행시간은 다음과 같다.

$$T_{TD} = T_d\left\{1 + \sum_{i=1}^{k}\prod_{j=1}^{i}\left(1 - \frac{1}{p_j}\right)\right\} \tag{2}$$

수식(2)에서 $\prod_{j=1}^{k}\left(1 - \frac{1}{p_j}\right)$은 난수 $r$이 $i$개 소수에 대해 나누어지지 않을 확률이다. $T_{mr}$을 MR test를 한 번 수행하는 데 걸리는 시간이라 하면 $T_{MR}$는 다음과 같다.

$$T_{MR} = T_{mr}\prod_{j=1}^{k}\left(1 - \frac{1}{p_j}\right) \tag{3}$$

따라서 TD-MR의 수행시간은 다음과 같다.

$$T = \frac{n\ln 2}{2}\left[\begin{array}{l} T_{rnd} + T_d\left\{1 + \sum_{i=1}^{k}\prod_{j=1}^{j}\left(1 - \frac{1}{p_j}\right)\right\} \\ + T_{mr}\prod_{j=1}^{k}\left(1 - \frac{1}{p_j}\right) \end{array}\right] \tag{4}$$

TD-MR-MR검사법은 TD-MR 수행 후 $(r-1)/2$를 MR test로 수행하며 상세과정은 다음과 같다.

1. 난수 생성
   - $n$-bit 홀수 난수 $r$을 생성
2. Trial Division 수행
   - 난수 $r$을 $k$개의 소수로 나눔
   - 나머지가 0이나 1이 나오면 1로 이동하고, 아니라면 3으로 이동
3. MR test 수행
   - $r$이 MR test를 통과하면 4로 이동하고 아니라면 1로 이동
4. MR test 수행
   - $(r-1)/2$가 MR test를 통과하면 안전소수로 반환하고, 아니라면 1로 이동

Park and Kim [11]은 TD-MR-MR검사법의 수행시간을 확률적으로 분석하였으며 상세 내용은 다음과 같다.

$$T = N(T_{rnd} + T_{TD} + T_{MR1} + T_{MR2}) \tag{5}$$

이 식에서 $N$은 안전소수를 찾을 때까지 난수를 생성해야 하는 횟수로 $0.66(n\ln 2/2)^2$이다. $T_{rnd}$는 이전과 동일하며 $T_{MR1}$과 $T_{MR2}$는 $r$과 $(r-1)/2$에 대한 MR test의 수행시간이다. $T_{TD}$는 $r$과 $(r-1)/2$이 모두 $p_j$에 의해 나누어떨어지지 않아야 하므로 다음과 같이 정리된다.

$$T_{TD} = T_d\left\{1 + \sum_{i=1}^{k}\prod_{j=1}^{i}\left(1 - \frac{2}{p_j}\right)\right\} \tag{6}$$

$T_{MR1}$은 $r$과 $(r-1)/2$이 모두 $k$개의 소수로 나누어지지 않아야 하고, $T_{MR2}$는 난수 $r$이 소수이면서 $(r-1)/2$이 $k$개의 소수와 서로소여야 하므로 전체 수행시간은 다음과 같이 정리된다.

$$T_{mr}\left\{\prod_{j=1}^{k}\left(1 - \frac{2}{p_j}\right) + \frac{2}{n\ln 2}\prod_{j=1}^{k}\left(1 - \frac{1}{p_j - 1}\right)\right\} \tag{7}$$

따라서 $k$개의 소수를 이용한 TD-MR-MR의 수행시간은 다음과 같다.

$$\begin{aligned} T = {} & \frac{1}{0.66}\left(\frac{n\ln 2}{2}\right)^2\left[T_{rnd} + T_d\left\{1 + \sum_{i=1}^{k}\prod_{j=1}^{i-1}\left(1 - \frac{2}{p_j}\right)\right\}\right. \\ & \left. + T_{mr}\left\{\prod_{j=1}^{k}\left(1 - \frac{2}{p_j}\right) + \frac{2}{n\ln 2}\prod_{j=1}^{k}\left(1 - \frac{1}{p_j - 1}\right)\right\}\right]. \end{aligned} \tag{8}$$

다. DT-MR검사법과 DT-MR-MR검사법

나눗셈테이블(Division Table, 이하 DT)은 기존의 TD를 개선한 것으로 순차난수 생성법과 나눗셈테이블 $S[i]$를 사용한다. [12] 다음은 DT-MR검사법의 과정을 설명하고 있다.

1. 난수 생성
   - $n$-bit 난수 $r_0$을 생성
2. DT 수행
   1) $S[i]=0$로 초기화, $r_0$을 $p_j$로 나누어 $R$을 계산
   2) $R$로 $p_j$로 나누어지는 $r_i$를 찾아 $S[i]=1$을 저장
3. Miller-Rabin test 수행
   - $S[i]=0$인 $r_i$에 대해 차례대로 MR test 수행
   - 검사를 통과한 $r_i$을 소수로 반환하고 종료
   - MR test를 통과한 $r_i$가 없다면 1로 이동

DT는 $p_j$로 나눌 수 있는 난수들을 쉽게 찾을 수 있는데 왜냐하면 테이블에서 $p_j$에 의해 나누어지는 난수들은 서로 $p_j$의 배수만큼 떨어져 있기 때문에 실제로 나누어보지 않고 테이블의 인덱스를 이동하면서 후보에서 제외할 수 있다. 나눗셈 테이블 $S[i]$는 길이가 $s$인 비트벡터 테이블이다.

$k$개의 소수를 이용하는 DT-MR검사법의 수행시간은 다음과 같다.

$$T = N(T_{RND} + T_{TD} + T_{TS}) + T_{MR} \tag{9}$$

$N$은 상수로서 소수를 찾을 때까지 만들어야 하는 테이블의 개수로 아래와 같다.

$$N = \frac{1}{1-\left(1-\frac{2}{n\ln 2}\right)^s} \tag{10}$$

$T_{TD}$는 $r_0$를 $k$개의 소수로 나누어보는데 $T_d$를 나눗셈 시간이라고 하면 $T_{TD} = kT_d$이다. $T_{TS}$는 $S[i]$를 0으로 초기화하고 $p_j$로 나누어지는 $r_i$를 찾은 다음 해당되는 $S[i]$에 1을 저장하는 시간이다. 메모리 접근시간을 $T_m$이라 할 때, 초기화시간은 $sT_m$이고 $S[i]$에 1을 저장할 확률은 각 $p_j$에 대해 $s/p_j$이다. 따라서 메모리에 접근하는 횟수는 $\left(\sum_{i=1}^{k}\frac{s}{p_j}\right)$이다.

$T_{MR}$은 $S[i]$=0인 $r_i$에 대해 MR test를 수행한다. $S[i]$=0을 찾는데 걸리는 시간은 $NsT_m$이고 $r_i$가 $k$개의 소수에 대해 나누어지지 않을 확률은 $\prod_{j=1}^{k}\left(1-\frac{1}{p_j}\right)$이고 소수를 찾을 때까지 반복해야하므로 $T_{MR}$는 다음과 같다.

$$T_{MR} = NsT_m + T_{mr}\frac{n\ln 2}{2}\prod_{j=1}^{k}\left(1-\frac{1}{p_j}\right) \tag{11}$$

따라서 나눗셈 테이블을 이용한 순차 난수 생성법의 예측시간은 다음과 같이 정리할 수 있다.

$$T = \frac{1}{1-\left(1-\frac{2}{n\ln 2}\right)^s}\left\{\begin{matrix} T_{rnd}+kT_d \\ +sT_m\left(2+\sum_{j=1}^{k}\frac{1}{p_j}\right)\end{matrix}\right\} + T_{mr}\frac{nln2}{2}\prod_{j=1}^{k}\left(1-\frac{1}{p_j}\right) \tag{12}$$

DT-MR-MR검사법에서 난수 $r$은 동일하게 생성하고 $r_i$ 또는 $(r_i-1)/2$가 $p_j$로 나누어지면 $S[i]$=1이고 아니면 $S[i]$=0이다.

1. 난수 생성
   - $n$-bit 난수 $r_0$를 생성
2. DT 수행
   1) $S[i]$=0로 초기화, $r_0$을 $p_j$로 나누어 $R$을 계산
   2) $R$로 $p_j$로 나누어지는 $r_i$를 찾아 $S[i]=1$을 저장
3. MR test 수행
   - $S[i]$=0인 $r_i$에 대해 MR test를 수행
   - 통과하면 4로 이동, 아니라면 $S[i]$=0인 다음 $r_i$에 3을 수행, $S[i]$=0인 $i$가 없을 경우 1로 이동
4. MR test 수행
   - $S[i]$=0인 $(r_i-1)/2$에 대해 MR test를 수행
   - 통과하면 $r_i$를 안전소수로 반환 후 종료,
   - 아니라면 $S[i]$=0인 다음 $r_i$에 대해 3으로 이동

DT-MR-MR검사법의 수행시간을 확률적으로 분석한 전체수행시간은 다음과 같다.

$$T = N(T_{RND} + T_{TD} + T_{TS}) + T_{MR1} + T_{MR2} \tag{13}$$

$T_{RND}$와 $T_{TD}$는 DT-MR검사법과 동일하다. $S[i]$에 1을 저장할 확률은 각 $p_j$에 대해 $s/p_j$이고 $r_i$와 $(r_i-1)/2$에 대해 수행하므로 $2s/p_j$라는 점이 다르다. 따라서 $T_{TS}$는 다음과 같다.

$$T_{TS} = sT_m\left(1+\sum_{j=1}^{k}\frac{2}{p_j}\right) \tag{13}$$

$T_{MR1}$과 $T_{MR2}$ $S[i]$=0인 $r_i$에 대해 MR test를 수행하는데 걸리는 시간과 $(r_i-1)/2$에 대해서 MR test

를 수행하는 데 걸리는 시간으로 다음과 같다.

$$T_{MR1}+T_{MR2}= T_{mr}\left\{\frac{1}{0.66}\left(\frac{n\ln 2}{2}\right)^2\prod_{j=1}^{k}\left(1-\frac{2}{p_j}\right)\right\} \left\{+\frac{2}{nln2}\prod_{j=1}^{k}\left(1-\frac{1}{p_j-1}\right)\right\} \quad (14)$$

전체 수행시간을 예측해보면 다음과 같다.

$$T=\frac{1}{1-\left\{1-0.66\left(\frac{2}{n\ln 2}\right)^2\right\}^s}\left[T_{rnd}+kT_{div} +2sT_m\left\{1+\sum_{j=1}^{k}\left(\frac{1}{p_j}\right)\right\}\right] + T_{mr}\left\{\frac{1}{0.66}\left(\frac{nln2}{2}\right)^2\prod_{j=1}^{k}\left(1-\frac{2}{p_j}\right)\right\} \left\{+\frac{2}{nln2}\prod_{j=1}^{k}\left(1-\frac{1}{p_j-1}\right)\right\} \quad (15)$$

## 2. 연구내용

가. 오일러체를 이용한 제안방법

TD는 난수를 $k$개의 소수로 나누어보기 때문에 나눗셈의 횟수가 많으며 DT는 나눗셈의 횟수는 줄였지만 합성수를 여러 번 검사하기 때문에 비효율적이다. 다음 표는 DT에서 중복판단을 하는 예시이다. X는 최초로 합성수임을 판단한 위치이며 D는 X 이후로 합성수로 중복판단한 위치이다. 이와 같이 이미 합성수로 판단한 난수에서도 수차례 중복으로 소수인지 판단하는 과정을 거치게 된다.

Table 1. An example of duplicate operations in DT.
표 1. DT에서 중복판단하는 예시

| | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 |
|---|---|---|---|---|---|---|---|---|---|
| 51 | X | D | | | | D | | | |
| 55 | X | D | | D | | | | | |
| 59 | | | | | | | | | X |
| 63 | X | | D | | | | | | |
| 67 | X | | | D | | | | | |
| 71 | | X | D | | | | | | |
| 75 | X | D | | | | | | | |
| 79 | X | | | | D | | | | |
| 83 | | | | | | | | | |
| 87 | X | | | | | | | | D |
| 91 | X | D | D | | D | | | | |
| 95 | | X | | | | | D | | |
| 99 | X | | D | D | | | | | |
| 103 | X | | | | | D | | | |
| 107 | | | | | | | | | |
| 111 | X | D | | D | | | | | |

다음은 오일러체를 이용한 소수검사방법이다.

1. $n$개의 자연수를 크기순으로 배열한다.
2. 판단되지 않은 숫자 중에 가장 작은 수를 선택하고 그 숫자를 소수로 판단한다.
3. 2번에서 소수로 판단된 숫자의 모든 배수를 합성수로 판단하고 삭제한다.
4. 모든 수를 판단할 때 까지 2~3번을 반복한다.

오일러체를 적용하면 이미 합성수로 판단한 난수를 중복검사하지 않아 성능이 향상될 수 있다. 뿐만 아니라 소수 $p_i$ 리스트를 가지고 있지 않아도 되므로 메모리공간도 개선할 수 있다.

나. ET-MR 소수검사법

오일러테이블(Euler Table, 이하 ET)은 DT의 테이블에 오일러체를 적용하여 개선한 방법이다.

ET-MR 소수검사법의 수행순서는 사전계산단계와 소수검사단계로 나누어진다. 사전계산의 상세내용은 다음과 같다.

사전계산

1. $\prod_{j=1}^{k}p_j$ 계산($k$는 $n$-bit를 고려하여 사용자가 설정)
2. $n$-bit 홀수 난수 $r_{init}$ 생성
3. 난수 인덱스 리스트 $R_n$ 생성
4. $R_n$에서 $k$개의 소수로 나누어지는 난수의 인덱스를 모두 삭제

사전계산은 TD 혹은 DT를 대체하는 과정으로 $k$개의 소수로 나누어지지 않는 $n$-bit 난수 후보들을 찾는다. $R_n$은 순차난수생성방법으로 $r_{init}$부터 시작해서 2씩 차이가 나는 난수들 중에 $k$개의 소수로 나누어지지 않는 난수들의 인덱스만 저장하는 리스트이다. $R_n$의 길이는 $\prod_{j=1}^{k}p_j$이고 $j$번째 원소의 값은 인덱스값인 $j$이며 이에 해당하는 난수는 $r_{init}+2j$이다.

소수 검사

1. $r_0$를 생성($r_0=r_{init}+u\cdot\prod_{j=1}^{k}p_j$, $u$는 임의의 정수)
2. $r_i$를 MR test 수행 ($r_i = r_{init} + 2R_n[i]$ $(0\le i\le t)$ )
3. $r_i$가 MR test를 통과하면 소수로 반환, 아니라면 $i$를 1증가하고 2로 이동, $i$가 최댓값이 되어도 소수가 없다면 1로 이동

소수검사에서는 $R_n$의 인덱스를 0부터 1씩 증가하면서 각 난수를 생성한 뒤 MR test를 수행한다. MR test를 통과한다면 소수로 반환하고 아니라면 다음 난수를 MR test한다.

수행시간을 확률적으로 분석해보면, $N$은 소수를 찾을 때까지 후보난수를 생성하는 횟수, $T_{rng}$는 $r_i$를 생성하는데 걸리는 시간, 리스트에 저장되어 있는 $r_i$의 개수 $t$, $T_{MR}$은 MR test를 수행하는 시간으로 정의하면 전체시간은 다음과 같다.

$$T = N \times t \times T_{rng} + T_{MR} \tag{16}$$

$N$과 $T_{MR}$은 DT-MR의 수행횟수와 시간이 동일하나 $T_{rng}$는 다음과 같다.

$$T_{rng} = T_{mul} + T_{add} \tag{17}$$

$T_{rng}$는 $r_i$를 하나 생성하는데 걸리는 시간이고, $r_i = r_{init} + 2R_n[i] (0 \le i \le t)$이므로 곱셈과 덧셈을 한번 하는데 걸리는 시간과 같다. 따라서 전체 수행시간은 다음과 같다.

$$T = \frac{1}{1-\left(1-\frac{2}{n\ln 2}\right)^s}\{t(T_{mul}+T_{add})\} + T_{mr}\frac{nln2}{2}\prod_{j=1}^{k}\left(1-\frac{1}{p_j}\right) \tag{18}$$

다. ET-MR-MR 안전소수검사법

ET-MR-MR 안전소수검사법도 사전계산단계와 소수검사단계로 나누어진다. 사전계산의 상세 내용은 다음과 같다.

사전계산
1. $\prod_{j=1}^{k} p_j$ 계산($k$는 $n$-bit를 고려하여 사용자가 설정)
2. $n$-bit 홀수 난수 $r_{init}$ 생성(단, $r_{init}\%4=3$을 만족)
3. 난수 인덱스 리스트 $R_n$ 생성
4. $R_n$에서 $k$개의 소수로 나누어지거나 나머지가 1인 난수의 인덱스를 모두 삭제

사전계산은 ET-MR 소수검사법과 2가지 차이점이 있다. 하나는 2번 과정에서 $r_{init}$를 생성할 때 홀수일 뿐만 아니라 $r_{init}\%4=3$을 만족해야 하는데 이는 $(r-1)/2$도 홀수이면서 $p_j$를 나누어지지 않아야하기 때문이다. 또 다른 하나는 4번 과정에서 $k$개의 소수로 나누었을 때 나머지가 0 혹은 1이 되는 모든 수를 제외해야 하는데 그 이유는 $r$과 $(r-1)/2$이 모두 $p_j$에 의해 나누어지지 않아야하기 때문이다.

소수 검사
1. $r_0$를 생성($r_0 = r_{init} + w \cdot \prod_{j=1}^{k} p_j$, $w$는 임의의 정수)
2. $r_i$를 MR test 수행 ($r_i = r_{init} + 4R_n[i]$ $(0 \le i \le t)$)
3. $r_i$가 MR test를 통과하면 4로 이동, 아니라면 $i$를 1 증가하고 2로 이동, $i$가 최댓값이 되어도 소수가 없다면 1로 이동
4. $(r_i-1)/2$를 MR test 수행
5. $(r_i-1)/2$가 MR test를 통과하면 안전소수로 반환, 아니라면 i를 1 증가하고 2로 이동

수행시간을 확률적으로 분석해보면, $N$은 안전소수를 찾을 때까지 후보난수를 생성하는 횟수, $T_{rng}$는 $r_i$를 생성하는데 걸리는 시간, 리스트에 저장되어 있는 $r_i$의 개수 $t$, $T_{MR}$은 $r$과 $(r-1)/2$에 대해 MR test를 수행하는 시간으로 다음과 같다.

$$T = N \times t \times T_{rng} + T_{MR} \tag{19}$$

$N$과 $T_{MR}$은 DT-MR-MR과 동일하고 $T_{rng}$는 ET-MR과 동일하다.

$$T_{rng} = T_{mul} + T_{add} \tag{20}$$

오일러체를 이용한 안전수소검사의 수행시간 예측수식은 다음과 같다.

$$T = \frac{1}{1-\left\{1-0.66\left(\frac{2}{nln2}\right)^2\right\}^s} t(T_{mul}+T_{add}) + T_{mr}\left\{\frac{1}{0.66}\left(\frac{nln2}{2}\right)^2\left[\prod_{j=1}^{k}\left(1-\frac{2}{p_j}\right) + \frac{2}{nln2}\prod_{j=1}^{k}\left(1-\frac{1}{p_j-1}\right)\right]\right\} \tag{21}$$

## 3. 실험 및 분석

PC의 실험환경은 Microsoft Windows 10 Home, 3.2GHz CPU, 8.00GB RAM에서 Microsoft Visual Studio 2017을 사용하였다.

가. 이론모델 검증 실험

먼저 확률적으로 분석하여 수립한 ET-MR 소수

검사법과 ET-MR-MR 안전소수검사법의 수행시간 예측모델이 정확한지를 검증하였다. 이를 위해, 임의의 $k$와 $n$에 대해 예측수행시간을 계산하고 실제 수행시간을 측정하여 두 시간을 비교하였다.

ET-MR 소수검사법의 실험은 $n$=128bit이고 $k$=8일 때와 $n$=512bit이고 $k$=64일 때, 소수 10,000개를 생성하는데 걸리는 시간을 측정하고 평균을 구하였다. 다음 표는 ET-MR의 실제측정시간과 예측실행시간을 비교한 것이다.

Table 2. Comparison of the theoretical time and measured time of ET-MR.

표 2. ET-MR의 이론시간과 실제시간의 비교

| $n$ | 128bit ($ms$) | 512bit ($ms$) |
|---|---|---|
| Theoretical time | 2.708 | 108.448 |
| Measured time | 2.643 | 111.933 |
| error rate(%) | 2.41 | 3.21 |

비교결과, 오차율이 각각 2.41%와 3.21%로 나와 ET-MR의 수행시간 예측모델이 확률적으로 잘 분석되었음을 확인하였다.

ET-MR-MR 안전소수검사법의 실험은 $n$=128bit이고 $k$=8일 때와 $n$=512bit이고 $k$=64일 때, 각각 안전소수를 생성하여 수행시간을 평균을 구하였다. 다음 표는 ET-MR-MR의 실제측정시간과 수행시간 예측모델에 의한 예측실행시간을 비교한 것이다.

Table 3. Comparison of the theoretical time and measured time of ET-MR-MR.

표 3. ET-MR-MR의 이론시간과 실제시간의 비교

| $n$ | 128bit ($ms$) | 512bit ($ms$) |
|---|---|---|
| Theoretical time | 4,236 | 3,827 |
| Measured time | 4,317 | 3,879 |
| error rate((%) | 1.91 | 1.38 |

비교결과 오차율이 각각 1.91%와 1.38%로 나와 ET-MR-MR이 수행시간 예측모델이 잘 수립되었음을 확인하였다.

나. 기존 조합소수와의 성능비교

기존의 대표적인 조합소수검사법인 TD-MR, DT-MR과 ET-MR의 수행시간을 이론모델을 이용하여 비교하였다.

Table 4. The running time comparison of TD-MR, DT-MR, ET-MR.

표 4. TD-MR, DT-MR, ET-MR의 수행시간비교

| $k$ | TD-MR ($ms$) | DT-MR ($ms$) | ET-MR ($ms$) |
|---|---|---|---|
| 16 | 151.56 | 151.54 | 150.31 |
| 64 | 110.44 | 109.76 | 108.44 |
| 256 | 89.18 | 86.29 | 84.90 |
| 512 | 83.57 | 78.09 | 76.68 |
| 991 | 81.60 | 71.69 | 70.25 |
| 1,229 | 81.84 | 69.84 | 68.40 |
| 2,048 | 84.72 | 65.84 | 64.38 |
| 4,096 | 96.09 | 61.14 | 59.66 |
| 12,381 | 148.84 | 58.46 | 53.47 |

$n$=512bit에서 실험결과, $k$가 64보다 작을 때는 TD-MR, DT-MR, ET-MR의 수행시간이 비슷하였으나 $k$가 64보다 커지면 TD-MR이 3개의 방법 중에 가장 느리고, ET-MR이 3개의 방법 중에 가장 빨랐다. TD-MR이 가장 빨랐을 때는 $k$=991일 때, 81$ms$였으며 ET-MR이 가장 빨랐을 때는 $k$=12,381일 때, 53$ms$였다. ET-MR의 수행시간이 가장 빨랐을 때와 TD-MR의 수행시간이 가장 빠른 때를 비교하면, ET-MR이 TD-MR보다 34.5% 빨랐으며, DT-MR보다 8.5% 빨랐다.

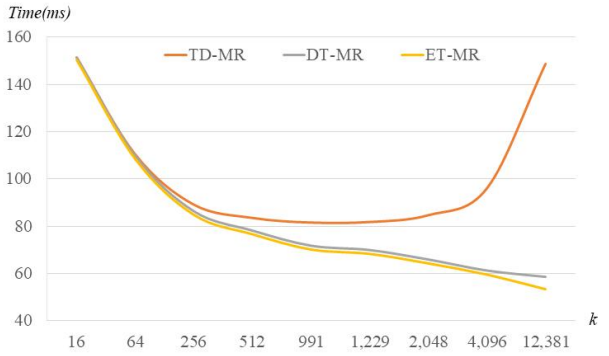


Fig. 1. Comparison of the running time of TD-MR, DT-MR and ET-MR.

그림 1. TD-MR, DT-MR, ET-MR의 수행시간 비교

$n$=512bit에서의 TD-MR, DT-MR, ET-MR의 메모리 사용량을 비교하였는데, $k$개 소수로 나누어지지 않는 난수를 알기 위해 필요한 메모리공간을 계산하였다. 즉 TD-MR은 $k$개의 소수를 저장하는 메모리공간을 계산하였고, DT-MR과 ET-MR은 $k$개

의 소수로 나누어지지 않는 난수들을 저장하는 데 필요한 공간을 계산하였다.

Table 5. Comparison of memory usage of TD-MR, DT-MR, ET-MR.
표 5. TD-MR, DT-MR, ET-MR의 메모리 사용량 비교

| $k$ | TD-MR(bit) | DT-MR(bit) | ET-MR(bit) |
|---|---|---|---|
| 16 | 256 | 817 | 8,725 |
| 64 | 1,024 | 817 | 6,295 |
| 256 | 4,096 | 817 | 4,928 |
| 512 | 8,192 | 817 | 4,451 |
| 991 | 15,856 | 817 | 4,078 |
| 1,229 | 19,664 | 817 | 3,970 |
| 2,048 | 32,768 | 817 | 3,737 |
| 4,096 | 65,536 | 817 | 3,463 |
| 12,381 | 198,096 | 817 | 3,104 |

TD-MR은 $k$가 커질수록 더 많은 메모리를 사용하였으며, DT-MR은 고정크기만큼만 필요하고, ET-MR은 $k$가 커질수록 더 적은 메모리를 사용하였다.

$k$=12,381일 때, TD-MR은 198,096bit를 사용하였고, DT-MR은 817bit를 사용하였다. ET-MR은 3,104bit를 사용하여 TD-MR의 사용량 대비 1.5% 정도만 사용하여 98.5%가 축소되었다. 하지만 DT-MR에 비해 공간을 약 2.7배 더 사용하였다.

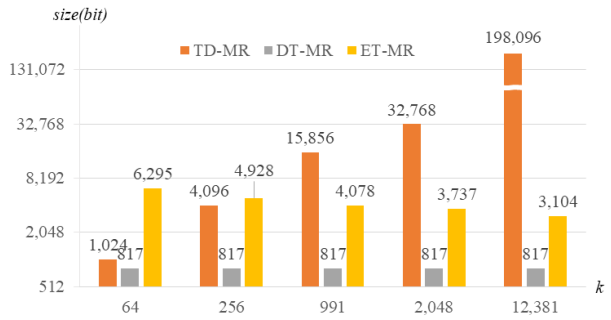


Fig. 2. Comparison of the memory usage of TD-MR, DT-MR and ET-MR.
그림 2. TD-MR, DT-MR, ET-MR의 메모리사용량 비교

다음으로 ET-MR-MR 안전소수검사법의 성능 비교를 위해 TD-MR-MR과 DT-MR-MR의 수행시간과 비교하였다.

Table 6. The running time comparison of TD-MR-MR, DT-MR-MR, ET-MR-MR.
표 6. TD-MR-MR, DT-MR-MR, ET-MR-MR의 수행시간비교

| $k$ | TD-MR-MR ($ms$) | DT-MR-MR ($ms$) | ET-MR-MR ($ms$) |
|---|---|---|---|
| 16 | 7,570 | 7,313 | 7,312 |
| 64 | 4,114 | 3,827 | 3,826 |
| 256 | 2,718 | 2,364 | 2,363 |
| 512 | 2,356 | 1,935 | 1,934 |
| 1,872 | 2,087 | 1,403 | 1,402 |
| 4,096 | 2,212 | 1,187 | 1,184 |
| 16,384 | 3,428 | 918 | 909 |
| 65,536 | 7,887 | 752 | 724 |

$n$=512bit에서 실험결과, $k$가 16보다 작을 때는 TD-MR-MR, DT-MR-MR, ET-MR-MR의 수행시간이 비슷하였으나 $k$가 16보다 커지면, 3개의 방법 중에 TD-MR-MR이 가장 느리고, ET-MR-MR이 가장 빨랐다. TD-MR-MR이 가장 빨랐던 때는 $k$=1,872일 때, 2,087$ms$였으며 ET-MR-MR이 가장 빨랐던 때는 $k$=65,536일 때, 724$ms$였다. ET-MR-MR과 TD-MR-MR의 수행시간 중 가장 빨랐던 때를 비교하면, ET-MR-MR이 TD-MR-MR대비 65.3% 성능이 향상되었다.

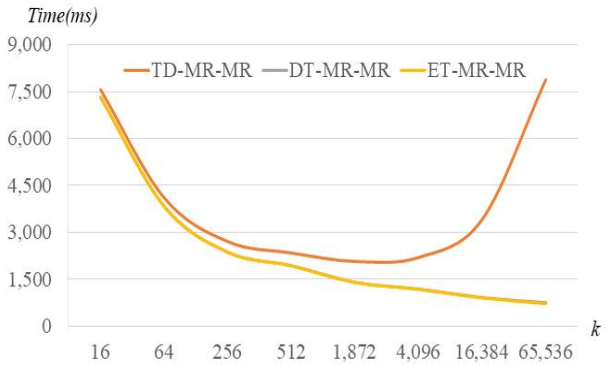


Fig. 3. Comparison of the running time of TD-MR-MR, DT-MR-MR and ET-MR-MR.
그림 3. TD-MR-MR, DT-MR-MR, ET-MR-MR의 메모리 사용량 비교

$n$=512bit에서의 TD-MR-MR, DT-MR-MR, ET-MR-MR의 메모리 사용량을 동일한 방법으로 비교하였다.

Table 7. Comparison of memory usage of TD-MR-MR, DT-MR-MR, ET-MR-MR.

표 7. TD-MR-MR, DT-MR-MR, ET-MR-MR의 메모리 사용량 비교

| $k$ | TD-MR-MR (bit) | DT-MR-MR (bit) | ET-MR-MR (bit) |
|---|---|---|---|
| 16 | 256 | 220,174 | 167,037 |
| 64 | 1,024 | 220,174 | 86,701 |
| 256 | 4,096 | 220,174 | 53,122 |
| 512 | 8,192 | 220,174 | 43,305 |
| 1,872 | 29,952 | 220,174 | 31,186 |
| 4,096 | 65,536 | 220,174 | 26,232 |
| 16,384 | 262,144 | 220,174 | 20,009 |
| 65,536 | 1,048,576 | 220,174 | 15,807 |

TD-MR-MR 또한 $k$가 커질수록 더 많은 메모리를 사용하였으며, DT-MR-MR은 고정크기만큼만 필요하고, ET-MR-MR은 $k$가 커질수록 더 적은 메모리를 사용하였다.

$k$=65,536일 때, TD-MR-MR은 1,048,576bit를 사용하였고, DT-MR-MR은 220,174bit를 사용하였다. ET-MR-MR은 15,807bit를 사용하여 TD-MR-MR의 사용량 대비 1.5%정도만 사용하여 98.5%가 축소되었다.

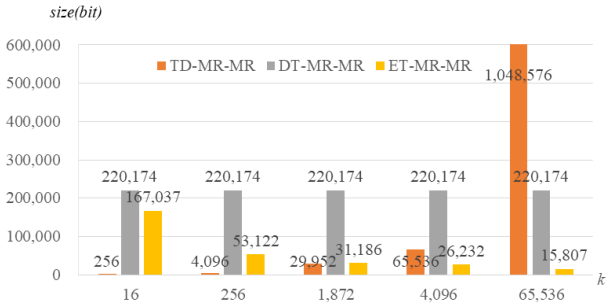


Fig. 4. Comparison of the memory usage of TD-MR-MR, DT-MR-MR and ET-MR-MR

그림 4. TD-MR-MR, DT-MR-MR, ET-MR-MR의 메모리 사용량 비교

## III. 결론

본 논문에서는 오일러체를 적용하여 (안전)소수를 생성하는 ET-MR 소수검사법과 ET-MR-MR 안전소수검사법을 제안하였다. 또한 이들을 확률적으로 분석하여 수행시간을 예측할 수 있는 수행시간 예측모델을 제안하였다. 실험 결과, ET-MR 소수검사와 ET-MR-MR 안전소수검사는 오차율이 각각 4%미만과 2%미만으로 확률적 분석이 잘 수행되었음을 확인하였다.

또한 기존의 (안전)소수검사법인 TD-MR, TD-MR-MR, DT-MR, DT-MR-MR과 성능비교를 하였을 때, 본 논문에서 제안하는 방법이 속도와 공간에서 더 효율적으로 나타났다. 각 알고리즘이 가장 빠를 때를 기준으로 수행시간들을 비교해보면, ET-MR은 TD-MR보다 34.5% 더 빨랐으며, DT-MR에 비해 8.5% 더 빨랐다. ET-MR-MR은 TD-MR-MR보다 65.3% 더 빨랐고, DT-MR-MR과는 비슷하였다. 사용하는 공간의 경우 $k$=12,381일 때 ET-MR이 TD-MR보다 98.5% 더 적게 사용하지만, DT-MR보다는 더 사용하였다. $k$=65,536일 때 ET-MR-MR이 TD-MR-MR보다 98.4% 더 적게 사용하였으며 DT-MR-MR보다 92.8% 더 적게 사용하였다.

## References

[1] Changhun Jung, Daehong Min, DaeHun Nyang and KyungHee Lee, “A Proposal of One-Time Password Authentication Protocol using DigitalSeal,” *The Journal of Korean Institute of Next Generation Computing*, Vol.14, No.4, pp.45-57, 2018.

[2] Jin Bok Kim, Tae Youn Han and Mun-Kyu Lee, “Authentication of smart phone users based on composite stage information,” *The Journal of Korean Institute of Next Generation Computing*, Vol.7, No.5, pp.4-12, 2011.

[3] Rivest, R. L. and Shamir, A. and Adleman, L., “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, Vol.21, No.2, pp.120-126, 1978. DOI: 10.1145/359340.359342

[4] ElGamal, T., “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” *IEEE Transactions on information Theory*, Vol. IT-31, No.4, pp.469-472, 1985. DOI: 10.1007/3-540-39568-7_2

[5] *Stallings, W., Cryptography and Network Security Principles and Practices*, 4th ed, Prentice Hall. 2005.

[6] *Cormen T. H., Leiserson C. E. and Rivest R. L. and Stein, C., Introduction to Algorithms*, 3rd

ed, MIT press. 2009.
[7] Miller, G. L., "Riemann's Hypothesis and Tests for Primality," *Journal of Computer Systems Science*, Vol.13. No.3. pp.300-317. 1976.
DOI: 10.1016/S0022-0000(76)80043-8
[8] *Menezes, A. J. and van Oorschot, P. C. and Vanstone, S. A., Handbook of Applied Cryptography*, CRC Press. 1996.
[9] Hosung Jo, Heejin Park, "Probabilistic Analysis of Incremental Random Number Generation using Division Tables," *2013 Fall Korea Software Congress,* pp.1330-1332, 2013.
[10] Maurer, U. M., "Fast generation of prime numbers and secure public-key cryptographic parameters," *Journal of Cryptology*, Vol.8. No.3. pp.123-155. 1995. DOI: 10.1007/BF00202269
[11] Heejin Park and Dongkyu Kim, "Probabilistic Analysis on the Optimal Combination of Trial Division and Probabilistic Primality Tests for Safe Prime Generation," *IEICE TRANSACTIONS on Information and Systems*, Vol.E94-D. No.6. pp.1210-1215. 2011. DOI: 10.1587/transinf.E94
[12] Changgi Kim, Hosung Jo, Heejin Park "Probabilistic Analysis of Prime and Safe-prime Generation with Division Table," *The Journal of Korean Institute of Next Generation Computing*, Vol.12, Issue6, pp.82-91, 2016.

## BIOGRAPHY

**Hosung Jo** (Member)

2005 : BS degree in Electronics and Computer Engineering, Dankook University.
2007 : MS degree in Electronics and Computer Engineering, Hanyang University.
2015 : PhD degree in Electronics and Computer Engineering, Hanyang University.
2015~2017 : Assistant research professor, Embedded software research center, Hanyang University
2017~ : Adjunct professor, Institute of software convergence, Hanyang University

**Jiho Lee** (Member)

2018 : BS degree in Computer Science, Hanyang University
2018~ : MS student, Department of Computer Science

**HeejinPark** (Member)

1994 : BS degree in Computer Engineering, Seoul National University.
1996 : MS degree in Computer Engineering, Seoul National University.
2001 : PhD degree in Computer Engineering, Seoul National University.
2001~2002 : post-doctoral researcher, Department of Computer Engineering, Seoul National University
2003~2003 : Research professor, Ewha Womens University
2003~ : Professor, Department of Computer Science, Hanyang University