http://dx.doi.org/10.7236/JIIBC.2013.13.6.221
JIIBC 2013-6-28

암호해독을 위한 소인수분해

Integer Factorization for Decryption

이상운*, 최명복**

Sang-Un Lee, Myeong-Bok Choi

요 약 큰 반소수 $n=pq$의 소인수 p,q를 나눗셈 시행법으로 직접 찾는 것은 현실적으로 거의 불가능하다. 따라서 대부분의 소인수분해 알고리즘은 $a^2 \equiv b^2 \pmod n$의 제곱합동을 찾아 $p=GCD(a-b,n), q=GCD(a+b,n)$의 소인수를 찾는 간접 방법을 적용하고 있다. $n=pq$에 대해 p와 q를 선택한 영역은 $l(p)=l(q)=l(\sqrt{n})=0.5l(n)$의 $[10\cdots01, 99\cdots9]$ 범위에서 $\sqrt{n}$을 기준으로 $10\cdots00<p<\sqrt{n}$과 $\sqrt{n}<q<99\cdots9$에 존재한다는 사실만이 밝혀졌다. 본 논문은 n으로부터 획득한 정보를 이용하여 p의 범위를 보다 축소시키는 방법을 제안한다. 제안 방법은 $n=n_{LR}+n_{RL}, l(n_{LR})=l(n_{RL})=l(\sqrt{n})$으로 분할하여 $p_{\min}=n_{LR}, q_{\min}=n_{RL}$로 설정하는 방법을 적용하였다. 본 논문에서 제안한 n의 정보로 p의 범위를 축소하는 방법은 $\sqrt{n}$의 정보로 p의 범위 축소 방법에 비해 최소 17.79%에서 최대 90.17%의 범위 축소 효과를 얻었다.

Abstract It is impossible directly to find a prime number p,q of a large semiprime $n=pq$ using Trial Division method. So the most of the factorization algorithms use the indirection method which finds a prime number of $p=GCD(a-b,n), q=GCD(a+b,n)$; get with a congruence of squares of $a^2 \equiv b^2 \pmod n$. It is just known the fact which the area that selects p and q about $n=pq$ is between $10\cdots00<p<\sqrt{n}$ and $\sqrt{n}<q<99\cdots9$ based on $\sqrt{n}$ in the range, $[10\cdots01, 99\cdots9]$ of $l(q)=l(\sqrt{n})=0.5l(n)$. This paper proposes the method that reduces the range of p using information obtained from n. The proposed method uses the method that sets to $p_{\min}=n_{LR}, q_{\min}=n_{RL}$; divide into $n=n_{LR}+n_{RL}, l(n_{LR})=l(n_{RL})=l(\sqrt{n})$. The proposed method is more effective from minimum 17.79% to maxmimum 90.17% than the method that reduces using $\sqrt{n}$ information.

Key Words : Prime number, Semiprime, Composite number, Sieve, Trial Division, Congruence of squares

Ⅰ. 서 론

RSA 암호의 공개키 n은 동일하거나 유사한 자리수에서 임의로 2개 소수 (prime number) p와 q를 추출하여 곱한 합성수 (composite number)로 반소수 (semiprime)라고도 한다. 여기서 임의로 선택한 p,q가 소수인지 여부는 소수 판별법 (primality test, PT)을 적용한다. 또한, RSA 암호 해독은 반소수 n을 p,q로 소인수분해 (factorization)하여야 한다.[1-4]

임의로 선택한 2개 소수 곱 $p\times q$로 반소수 n을 생성하는 것은 쉬운 반면에, 역으로 n이 주어졌을 때 이를 2개의 소인수 p,q로 인수분해하는 것은 어렵다. 소인수분

*정회원, 강릉원주대학교 멀티미디어공학과
**종신회원, 강릉원주대학교 멀티미디어공학과
접수일자 2013년 4월 19일, 수정완료 2013년 11월 19일
게재확정일자 2013년 12월 13일

Received: 19 March, 2013 / Revised: 19 November, 2013
Accepted: 13 December, 2013
**Corresponding Author: cmb5859@gmail.com
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

해 방법에는 나눗셈 시행 (Trial Division), Pollard의 rho 알고리즘, Pollad의 p-1 알고리즘, William의 p+1 알고리즘, Lenstra elliptic curve 인수분해, 페르마 (Fermat) 인수분해법, 오일러 (Euler) 인수분해법, 2차 체 (quadratic Sieve), MPQS (multiple-polynomial quadratic sieve), NFS (Number field sieve), GNFS (General number field), Dixon, CFRAC (continued fraction factorization), SQUFOF (Shanks' square forms factorization)와 양자 컴퓨터를 활용한 Shor 알고리즘 등 다양한 방법들이 있다.[5]

나눗셈 시행법은 $2 \le p < \sqrt{n}$의 소수에 대해 $n \equiv 0 \pmod p$ 또는 $\frac{n}{p} = q$(정수)로 p를 직접 찾는 방식이다.[6] n이 큰 수인 경우 나눗셈 시행법은 비현실적으로 과다한 시간이 소요된다. 따라서 일반적으로 간접 방법인 체 (Sieve) 방법이 사용되고 있다. 체 방법은 $a^2 - b^2 \equiv 0 \pmod n$의 제곱 합동 (congruence of squares)인 a,b를 결정하고 $p = GCD(a-b, n)$, $q = GCD(a+b, n)$을 구하는 간접 방식이다.[7-11] 그러나 a,b를 결정하는 과정도 쉽지 않다. 왜냐하면, RSA Lab.에서 1991년에 RSA Factoring Challenge를 결성하여 총 $687,563의 상금을 걸었지만 지금까지 RSA-100부터 RSA-200의 일부 데이터만 소인수분해에 성공하여 $82,563의 상금을 지급하였고, $605,000의 상금이 철회된 상태로 2007년에 해체되었다.[12] 이후, 2009년에 T. Kleinjung et al.이 $50,000 상금이 걸린 RSA-768을 소인수분해에 성공하였으나 상금은 수령하지 못하였다. 결국, RSA 수의 소인수분해 문제는 현재 진행형이며, 다항시간 알고리즘을 찾지 못할 수도 있다.

지금까지는 $n = pq$이며, p와 q를 선택한 영역은 $l(p) = l(q) = l(\sqrt{n}) = \frac{l(n)}{2}$의 $[10\cdots01, 99\cdots9]$ 범위에서 $\sqrt{n}$을 기준으로 $10\cdots00 < p < \sqrt{n}$과 $\sqrt{n} < q < 99\cdots9$에 존재한다는 사실만이 밝혀졌다.

만약, 소수 p와 q를 선택한 범위를 $l(p) = l(q) = l(\sqrt{n})$의 $10\cdots00 < p < \sqrt{n}$과 $\sqrt{n} < q < 99\cdots9$보다 축소시킬 수 있다면 n을 소인수분해하는 것이 보다 쉬울 것이다. 본 논문은 이 주제에 초점을 맞춘다.

2장에서는 반소소의 소인수 범위를 고찰해 본다. 3장에서는 반소소의 소인수 범위를 보다 축소시켜 소인수분해하는 방법을 제안한다.

II. 반소수의 소인수 범위

RSA 보안체계에 적용되는 합성수 n은 홀수 반소수로 반드시 2개의 소인수 p,q만을 가진다. 표 1의 실험 데이터를 대상으로 고찰해 보자.

표 1. 반소수의 소인수분해 실험 데이터
Table 1. Factorization experiment data of semiprime

n	$p \times q$	$a^2 \equiv b^2 \pmod n$
84,923	$p=163, q=521$	$a=342, b=179$
112,729	$p=139, q=811$	$a=475, b=336$
6,012,707	$p=2,357, q=2,551$	$a=2,454, b=97$
18,206,927	$p=1,933, q=9,419$	$a=5,676$, $b=3,743$
2,352,854,039	$p=42,013$, $q=56,003$	$a=49,008$, $b=6,995$
137,085,519,229	$p=229,423$ $q=597,523$	$a=413,473$ $b=184,050$
63,209,496,861,097	$p=7,368,511$ $q=8,578,327$	$a=7,973,419$ $b=604,908$
502,636,025,594,071	$p=15,488,197$ $q=32,452,843$	$a=23,970,520$ $b=8,482,323$
164,554,085,186,383,429	$p=217,645,199$ $q=756,065,771$	$a=486,855,485$ $b=269,210,286$
8,229,944,909,131,434,961	$p=2,352,854,041$ $q=3,497,856,121$	$a=2,925,355,081$ $b=572,501,040$
982,301,348,481,615,682,763,349,396,546,115,836,409	$p=20,989,897,656,489,026,809$ $q=46,798,767,890,987,654,401$	$a=33,894,332,773,738,340,605$ $b=1,904,435,117,249,313,796$

주어진 수 n으로부터 소인수 p,q를 직접 구하는 방법은 나눗셈 시행법[7, 13-14]이 있다. 나눗셈 시행법은 $n/p = q$ 공식에 의거 $2 \le p < \sqrt{n}$의 범위에서 소수들만을 대상으로 $n/p = q$ 또는 $n = 0 \pmod p$를 찾아야 한다. 결국, $p = [2, \sqrt{n}]$의 범위에서 무작위로 선택된 소수이다. 그러나 소인수분해된 RSA 수들을 고찰해본 결과 반소수 특징은 RSA-129를 제외한 대부분은 $l(p) = l(q) = l(\sqrt{n}) = l(n)/2$에서 $\sqrt{n}$을 기준으로 양쪽에서 1개씩의 소수를 무작위로 선택한 값이다. $\sqrt{n}$의 정보만을 이용할 경우 $p = [100\cdots01, \sqrt{n}]$, $q = [\sqrt{n}, 99\cdots99]$의 범위로 축소되며, 더 이상의 범위 축소는 불가하다. 이는 표 2에서 확인할 수 있다. $n = pq$의 특징은 n,p,q는 모두 홀수이다. 왜냐하면 소수(홀수)×소수(홀수)=합성수(홀수)가 되기 때문이다. 또한, n,p,q의 1의 자리수는 1, 3, 7 또는 9이다. 만약, $a^2 \equiv b^2 \pmod n$으로 제곱합동 a,b를 얻고, $p = GCD(a-b, n), q = GCD(a+b, n)$으로 p,q를 얻는 간접방법을 적용할 경우, $p = a-b, q = a+b$, $a = \frac{p+q}{2}, b = \frac{q-p}{2}$ 관계를 갖는다.

Ⅲ. 암호해독 소인수분해법

2장에서는 주어진 반소수 $n(p\times q)$으로부터 활용할 수 있는 $\sqrt{n}$으로부터 2개 소인수의 범위는 $p=[100\cdots01,\sqrt{n}]$, $q=[\sqrt{n},99\cdots99]$로 결정할 수 있었다. 이를 $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$로 표기하자.

본 장에서는 p,q의 범위를 보다 더 축소시키는 방법을 제안한다. 제안된 방법은 n으로부터 $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$ 정보를 추출한다.

[Step 1] p와 q의 초기치는 표 2의 결과로 다음과 같다.

$$p_{\min}=10\cdots01, l(\sqrt{n}), p_{\max}=\lfloor\sqrt{n}\rfloor$$

$$q_{\min}=\lceil\sqrt{n}\rceil, q_{\max}=99\cdots9, l(\sqrt{n}).$$

[Step 2] $n=n_{LR}+n_{RL}$로 분할한다. 여기서 LR은 MSB인 좌측(left)에서 LSB인 우측(right)으로의 방향을, RL은 반대 방향을 의미한다. $l(n_{LR})=l(n_{RL})=l(\sqrt{n})$이다.

표 2. 반소수의 p,q 범위 초기치

Table 2. Range initial values of p,q of semiprime

n	$[p_{\min},p_{\max}]$	$\lfloor\sqrt{n}\rfloor$	$[q_{\min},q_{\max}]$
84,923	[101,291]	291	[292,999]
112,729	[101,335]	335	[336,999]
6,012,707	[1001,2452]	2452	[2453,9999]
18,206,927	[1001,4266]	4266	[4267,9999]
2,352,854,039	[10001,48506]	48,506	[48507,99999]
137,085,519,229	[100001,370250]	370,250	[370251,999999]
63,209,496,861,097	[1000001,7950440]	7,950,440	[7950441,9999999]
502,636,025,594,071	[10000001,22419545]	22,419,545	[22419546,99999999]
164,554,085,186,383,429	[100000001,405652665]	405,652,665	[405652666,999999999]
8,229,944,909,131,434,961	[1000000001,2868788055]	2,868,788,055	[2868788056,9999999999]
982,301,348,481,615,682,763,349,336,546,115,836,409	[10000000000000000001,3134168707140857868]	31,341,687,071,400,857,868	[31341687071400857869,99999999999999999999]

표 2의 데이터에 대해 Step 2에 따라 $n=n_{LR}+n_{RL}$로 분할한 결과는 표 3과 같다.

[Step 3] if $n_{LR}<\sqrt{n}<n_{RL}$ then $p_{\min}=n_{LR}$, $q_{\min}=n_{RL}$ /* $\sqrt{n}=Median$: [1]

else if $\sqrt{n}<n_{LR}<n_{RL}$ or $\sqrt{n}<n_{RL}<n_{NR}$ then

$q_{\max}=\lfloor 2\sqrt{n}\rfloor$ /* $\sqrt{n}=Min$: [2]

else if $n_{LR}<n_{RL}<\sqrt{n}$ or $n_{RL}<n_{NR}<\sqrt{n}$ then

$p_{\min}=\max\{n_{LR},n_{RL}\}$,

$q_{\max}=\left\lfloor\frac{n}{p_{\min}}\right\rfloor$.

/* $\sqrt{n}=Max$: [3]

표 3. 반소수의 $n=n_{LR}+n_{RL}$

Table 3. $n=n_{LR}+n_{RL}$ of semiprime

n	n_{LR} $[p_{\min},p_{\max}]$	$\lfloor\sqrt{n}\rfloor$	n_{RL}, n_{RL}^{-1} $[q_{\min},q_{\max}]$
84,923	849 [101,291]	291	923, 329 [292,999]
112,729	112 [101,335]	335	729, 927 [336,999]
6,012,707	6,012 [1001,2452]	2452	2,707, 7072 [2453,9999]
18,206,927	1,820 [1001,4266]	4,266	6927, 7296 [4267,9999]
2,352,854,039	23,528 [10001,48506]	48,506	54039, 93045 [48507,99999]
137,085,519,229	137,085 [100001,370250]	370,250	519229, 922915 [370251,999999]
63,209,496,861,097	6,320,949 [1000001,7950440]	7,950,440	6861097, 7301686 [7950441,9999999]
502,636,025,594,071	50,263,602 [10000001,22419545]	22,419,545	2559071, 17049552 [22419546,99999999]
164,554,085,186,383,429	164,554,085 [100000001,405652665]	405,652,665	186383429, 924383681 [405652666,999999999]
8,229,944,909,131,434,961	8,229,944,909 [1000000001,2868788055]	2,868,788,055	9131434961, 16943413 19 [2868788056,9999999999]
982,301,348,481,615,682,763,349,336,546,115,836,409	98,230,134,848,161,568,276 [100000000000000000001,3134168707140857868]	31,341,687,071,400,857,868	6334033654611536409, 90463851164563394336 [31341687071400857869,99999999999999999999]

Step 3에 따라 $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$을 조정한 결과는 표 4와 같다.

표 4. n_{LR},n_{RL}에 따른 $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$ 조정

Table 4. $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$ of n_{LR}, n_{RL}

n	조건	$[p_{\min},p_{\max}]$	$[q_{\min},q_{\max}]$
84,923	2	[101,291]	[292,999] → [292,582]
112,729	1	[101,335] → [112,335]	[336,999] → [729,999]
6,012,707	2	[1001,2452]	[2453,9999] → [2453,4904]
18,206,927	1	[1001,4266] → [1820,4266]	[4267,9999] → [6927,9999]
2,352,854,039	1	[10001,48506] → [23528,48506]	[48507,99999] → [54039,99999]
137,085,519,229	1	[100001,370250] → [137085,370250]	[370251,999999] → [519229,999999]
63,209,496,861,097	3	[1000001,7950440] → [6861097,7950440]	[7950441,9999999] → [7950441,9212739]
502,636,025,594,071	2	[10000001,22419545]	[22419546,99999999] → [22419546,44839090]
164,554,085,186,383,429	3	[100000001,405652665] → [186383429,405652665]	[405652666,999999999] → [405652666,882879374]
8,229,944,909,131,434,961	2	[1000000001,2868788055]	[2868788056,9999999999] → [2868788056,5737576110]
982,301,348,481,615,682,763,349,336,546,115,836,409	2	[100000000000000000001, 3134168707140857868]	[31341687071400857869, 999999999999999999999] → [31341687071400857869, 626833741428017157 36]

[Step 4] if $q_{\min}\neq\lceil\sqrt{n}\rceil\cap q_{\max}=99\cdots9$ then

$$p_{\max}=\left\lfloor\frac{n}{q_{\min}}\right\rfloor-[1]$$

else if $q_{\min}=\lceil\sqrt{n}\rceil\cap q_{\max}\neq99\cdots9$ then

$$p_{\min}=\left\lfloor\frac{n}{q_{\max}}\right\rfloor-[2]$$

else if $q_{\max}=99\cdots9\cap(90\cdots<n_{RL}^{-1}<95\cdots)$ then

$$q_{\max}=n_{RL}^{-1}, p_{\min}=\left\lceil\frac{n}{q_{\max}}\right\rceil-[3].$$

Step 4에 따라 $p_{\min}$ 또는 $p_{\max}$의 변경된 결과로 얻은 p,q,a의 범위는 표 5에 제시되어 있다. 표 5의 $[a_{\min},a_{\max}]$는 $a_{\min}=\left\lceil \frac{p_{\min}+q_{\min}}{2} \right\rceil$, $a_{\max}=\left\lfloor \frac{p_{\max}+q_{\max}}{2} \right\rfloor$로 결정되었다. 지금까지는 소인수 p의 범위를 축소시키는 방법을 고찰하였다. 축소된 소인수 범위에 대해 어떤 탐색 방법을 적용하면 효율적인가에 대해 제안한다.

가장 단순한 방법은 표 5의 $[p_{\min},p_{\max}]$에 대해 오름차순과 내림차순으로 동시에 탐색할 수 있다. 두 번째 방법은 $[p_{\min},p_{\max}]$값에 대해 MSB 2자리 수의 구간으로 분할한 구간들에 대해 나눗셈 시행법을 수행할 수 있다. 예를 들면, $n=6012707$의 $[1326,2452]$에 대해 [1326,1399], [1401,1499],…,[2301,2399], [2401,2452] 구간의 1의 자리가 1,3,7,9인 값을 동시에 나눗셈 하여 p를 얻을 수 있다. 본 장에서는 Step 5의 방법을 제안한다.

표 5. $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$, $a=[a_{\min},a_{\max}]$
Table 5. $p=[p_{\min},p_{\max}]$, $q=[q_{\min},q_{\max}]$, $a=[a_{\min},a_{\max}]$

n	조건	p $[p_{\min},p_{\max}]$	q $[q_{\min},q_{\max}]$	a $[a_{\min},a_{\max}]$
84,923	2	163 [101,291] → [146,291]	521 [292,582]	342 [292,364]
112,729	1,3	139 [112,335] → [122,155]	811 [729,999]→[729,927]	475 [442,555]
6,012,707	2	2,357 [1001,2452]→[1226,2452]	2,551 [2453,4904]	2,454 [2453,3065]
18,206,927	1	1,933 [1820,4266]→[1820,2628]	9,419 [6927,9999]	5,676 [4778,5909]
2,352,854,039	1	42,013 [23528,48506]→ [25288,43540]	56,003 [54039,99999]→ [54039,93045]	49,008 [48790,61763]
137,085,519,229	1	229,423 [137085,370250]→ [148536,264017]	597,523 [519229,999999]→ [519229,922915]	413,473 [391623,568542]
63,209,496,861,097	2	7,368,511 [6861097,7950440]	8,578,327 [7950441,9212707]	7,973,419 [7950441,8096902]
502,636,025,594,071	2	15,488,197 [10000001,22419545]→ [11209773,22419545]	32,452,843 [22419546,44839090]	23,970,520 [22419546,28024431]
164,554,085,186,383,429	2	217,645,199 [186383429,406652665]	756,065,771 [406652666,889276374]	486,855,485 [406652666,534631401]
8,229,944,909,131,434,961	2	2,352,854,041 [1000000001,2868788055]→ [1434394028,2868788055]	3,497,856,121 [2868788056,5737576110]	2,925,355,081 [2868788056,3585985069]
982,301,348,481,615,682,763,349,336,546,115,896,409	2	20,989,897,656,489,026,809 [10000000000000000001,3134168707140085786 8] → [15670843535700428834,3134168707140085786 8]	46,798,767,890,987,654,401 [3134168707140085786 9,6268337414280171573 6]	33,894,332,773,738,340,605 [3134168707140085786 9,4701253060710128680 2]

[Step 5] $p_{\min}$에 대해 MSB 2번째 자리수 +1로 설정한다. 소인수 p의 탐색 시작점 p_0를 구한다. p_0는 $[p_{\min},p_{\max}]$에 존재하는 모든 값을 n에서 좌에서 우로, 우에서 좌로 획득한다. 또한, $p_{\max}$에 대해 MSB 2번째 자리수 −1을 한 값을 추가한다.

예로, $n=84923$의 경우 $p=[146,281]$이 [156,281]로 변경되고, 1xx과 2xx의 값 156, 238, 294가 추출되고 [156,281]에 속한 156과 238이 선택된다. 또한, 271이 추가된다. 따라서 $p_0=156,238,271$이 된다.

Step 5에 따라 p_0와 "탐색 범위 $[p_{\min},p_{\max}]$ 재설정" 결과는 표 6에 제시되어 있다. 각 "탐색 범위 $[p_{\min},p_{\max}]$ 재설정" 값에 대해 동시에 1의 자리가 1,3,7,9인 값을 오름차순과 내림차순으로 대입하면서 $n/p=$정수 또는 $n=0\pmod p$를 p로 결정한다.

간접 방법을 적용할 경우 표 5의 $[a_{\min},a_{\max}]$에 대해 ±1 증분으로 오름차순과 내림차순으로 동시에 탐색하면서 $a^2\equiv b^2\pmod n$을 구할 수 도 있다.

제안된 방법인 n으로부터 얻을 수 있는 축소된 p의 범위를 기존의 $\sqrt{n}$으로 얻은 p의 범위와 비교한 결과는 표 7과 같다. 제안된 방법은 최소 17.79%에서 최대 85.90%까지의 p 범위 축소 효과를 얻었다.

제안된 방법을 RSA 번호에 적용하여 보자. RSA 번호 중 RSA-129를 제외한 소인수분해가 된 값들은 표 8에 제시되어 있다. 여기서의 특징은 $q_{\max}\neq99\cdots$가 아닌 $q_{\max}=80\cdots$이다. RSA-100에 대해 적용해 보면, $n=$ 1522605027922533360535618378132637429718068114961380688657908494580122963258952897654000350692006139, $p=$ 3797522793694367392280887275544562785456553 6638199, $q=$ 4009469095092088103068373529276146838922 14899724061, $\sqrt{n}=390205\cdots$으로 $15226050\cdots<p<390205$이며, $q_{\max}=80000\cdots$을 적용하면 $p_{\min}=1903256\cdots$을 얻었다. 따라서 $1903256\cdots<p<390205\cdots$로 수정된다. n에서 $19\cdots,2\cdots,38\cdots$을 선택하면 $226050\cdots$, $260502,\cdots$등 다수의 p_0를 얻는다. 이들 p_0에서 [3742971806811496138 0688657908494580122963258952897, 3806886579084945801 2296325895289765400035069200612]에서 1의 자리가 1,3,7,9인 값을 오름차순 보다는 내림차순으로 탐색하면 $p=$ 3797522793694367392280887275544562785456553663819 9을 보다 빠르게 얻을 수 있다.

$\lfloor\sqrt{n}\rfloor-p$와 $q-\lfloor\sqrt{n}\rfloor$의 거리가 동일하면 p,q를 쉽게 구할 수 있기 때문에 p,q를 선택시 주의해야 한다. 예를 들면, $p=q=[11,97], p\neq q$의 소수의 곱 n에 대해 표 9에 제시하였다. 여기서 $\hat{p}=\left\lfloor \lceil\sqrt{n}\rceil-\sqrt{\lceil\sqrt{n}\rceil^2-n} \right\rfloor$으로 구한 결과 210개의 데이터 중에서 $\hat{p}=p$로 74개를 얻어 35.24%는 p값을 즉시 얻는다. 또한, p는 $\hat{p}$와의 차

표 6. 탐색 시작 점 p_0와 탐색 범위
Table 6. Search start point p_0 and search range

n	p	$[p_{\min}, p_{\max}]$ 조정	탐색 시작점 p_0 $p_{\min} \le p_0 < p_{\max}$, $p_{\max}\,(MSB_2 - 1)$	탐색 범위 $[p_{\min}, p_{\max}]$ 재설정
84,923	163	[146,291]→[156,291]	156, 238, 281	→[156,237], [238,280], [281,291]
112,729	139	[122,155]→[132,155]	132, 145	→[132,144], [145,335]
6,012,707	2,357	[1226,2452]→[1326,2452]	1326, 2106, 2352	[1326,2105], [2106,2351], [2352,2452]→
18,206,927	1,933	[1820,2628]→[1920,2628]	1920, 2069, 2528	→[1920,2068], [2069,2527], [2528,4266]
2,352,854,039	42,013	[25288,43540]→ [26288,43540]	26288, 28540, 40392, 43540	[26288,28539], [28540,40391], [40392,43539]← [42540, 48506]
137,085,519,229	229,423	[148536,264017]→ [158536,264017]	158536, 192291, 229137, 229155, 254017	[158536,192290], [192291,229136],[229137,229154], →[229155,254016], [254017, 370250]
63,209,496,861, 097	7,368,511	[6861097,7950440]→ [6961097,7950440]	6961097, 7632094, 7901686	[6961097,7632093]←, [7632094,7901685], [7901686,7950440]
502,636,025,594, 071	15,488,197	[11209773,22419545]→ [12209773,22419545]	12209773, 15026360, 17049552, 20517049, 20636205, 21419545	[12209773,15026359], →[1502360,17049551], [17049552,20517048], [20517049,20636204], [20636205,21419544], [22419545,22419545]
164,554,085,186, 383,429	217,645,199	[186383429,405652665]→ [196383429,405652665]	196383429,243836815,291645540, 368158045,383429164,395652665	→[196383429,243836814], [243836815,291645539], [291645540,368158044], [368158045,383429163], [383429164,395652664], [395652665,405652665]
8,229,944,909, 131,434,961	2,352,854,041	[1434394028,2868788055]→ [1534394028,2868788055]	1534394028, 1694341319, 2299449091, 2768788055	[1534394028,1694341318], [1694341319,2299449090], →[2299449091,2768788054], [2768788055, 2868788055]
982,301,348,481, 615,682,763,349, 336,546,115,836, 409	20,989,897,656, 489,026,809	[1567084353570042893 4, 3134168707140085786 8]→ [1667084353570042893 4, 3134168707140085786 8]	1667084353570042893 4 1843103289904638511 6 1848431032899046385 1 2301348481615682763 3 2763349336546115836 4 2865161848431932899 0 2899046385116456339 4 3013484816156827633 4 3034168707140085786 8	[1667084353570042893 4,1843103289904638511 5] [1843103289904638511 6,1848431032899046385 0] →[1848431032899046385 1,2301348481615682763 2] [2301348481615682763 3,2763349336546115836 3] [2763349336546115836 4,2865161848431932899 0] [2865161848431932899 0,2899046385116456339 3] [2899046385116456339 4,3013484816156827633 3] [3013484816156827633 4,3034168707140085786 7] [3034168707140085786 8,3134168707140085786 8]

이가 커야 한다. 만약, 차이가 작은 경우, $\hat{p}$이 짝수이면 $\hat{p}=\hat{p}-1$로 설정하고, "-2"씩 감소시키면서 나눗셈시행법으로 p를 쉽게 구할 수 있다. 마찬가지로, $\hat{q}=\lfloor \lceil \sqrt{n} \rceil + \sqrt{\lceil \sqrt{n} \rceil^2 - n} \rfloor$로 구한다. 만약, $\hat{p}-p$의 차이가 크다면 제안된 알고리즘을 적용하여 구할 수 있다.

결국, p,q를 선택시 기존의 방법은 동일하거나 유사한 길이의 소수를 임의로 선택하는 기준만을 적용하였다. 이 기준은 "동일하거나 유사한 길이의 소수를 선택하면서 $\lceil \sqrt{n} \rceil - p \neq q - \lceil \sqrt{n} \rceil$ 이어야 하며, $\hat{p}-p$와 $\hat{q}-q$가 커야만 한다."로 수정되어야 한다.

표 7. $\sqrt{n}$과 n에서 추출된 p 범위 비교
Table 7. Range comparison of extracted from $\sqrt{n}$ and n

n	$[p_{\min}, p_{\max}]$		축소율 (%)
	$\sqrt{n}$ 에서 추출된 p 범위	n에서 추출된 p 범위	
84,923	[101,291]	[156,291]	28.95
112,729	[101,335]	[132,155]	90.17
6,012,707	[1001,2452]	[1326,2452]	23.09
18,206,927	[1001,4266]	[1920,2628]	78.32
2,352,854,039	[10001,48506]	[26228,43540]	55.04
137,085,519,229	[100001,370250]	[158536,264017]	60.97
63,209,496,861,097	[1000001,7950440]	[6961097,7950440]	85.77
502,636,025,594,071	[10000001,22419545]	[12209773,22419545]	17.79
164,554,085,186,383, 429	[100000001,405652665]	[196383429,405652665]	28.26
8,229,944,909,131, 434,961	[1000000001,2868788055]	[1534394028,2868788055]	23.24
982,301,348,481,615, 682,763,349,336,546, 115,836,409	[1000000000000000000001,3134168707140085786 8]	[1667084353570042893 4,3134168707140085786 8]	26.57

표 8. RSA 번호의 $q_{\max}=80\cdots$일 때의 $p_{\min}$ 범위

Table 8. $p_{\min}$ range of $q_{\max}=80\cdots$

RSA 번호	n	n_{LR}	n_{RL}	n_{RL}^{-1}	$p_{\min}$	p	$\sqrt{n}$	q
RSA-100	1522605027922533605···	1,522,605,027	8,068,865,790	3,169,411,860	1,908,256,285	3,797,522,793···	3,902,057,186···	4,009,469,095···
RSA-110	3594234179725868774···	3,579,423,417	3,532,908,190	**7,668,657,130**	4,474,279,272	5,846,418,214···	5,982,828,276···	6,122,421,090···
RSA-120	1143816257578888676···	2,270,104,812	2,582,470,091	9,748,456,920	2,837,631,016	3,274,145,556···	4,764,561,693···	6,933,426,671···
RSA-130	1807082088687404805···	1,807,082,088	5,005,666,254	**7,550,884,123**	2,258,852,611	3,968,399,945···	4,250,978,815···	4,553,449,864···
RSA-140	2129024631825875754···	2,129,024,631	9,984,056,495	1,746,391,443	2,661,280,790	3,398,717,423···	4,614,135,490···	6,264,200,187···
RSA-150	1550898124783484050···	1,550,898,124	4,634,659,543	3,864,935,970	1,938,622,656	3,480,098,671···	3,998,144,391···	4,456,477,449···
RSA-155	1094173864157052742 1···	1,094,173,864	9,347,847,179	**7,983,334,533**	1,000,000,001	1,026,395,928···	1,046,027,659···	1,066,034,883···
RSA-160	2152741102718897018···	2,152,741,102	3,567,301,105	3,577,040,757	2,680,926,378	4,542,789,285···	4,639,764,113···	4,738,809,060···
RSA-170	2606262368413984921···	2,606,262,368	2,839,191,451	9,575,451,170	3,257,827,961	3,586,420,730···	5,105,156,578···	7,267,029,064···
RSA-576	1881988129206079638 3···	1,881,988,129	6,060,650,474	9,507,520,567	2,352,485,162	3,980,750,864···	4,338,188,711···	4,727,721,461···
RSA-180	1911478277189866096 8···	1,911,479,277	3,651,351,612	1,401,241,587	2,389,349,096	4,007,800,823···	4,372,046,749···	4,769,396,887···
RSA-640	3107418240490043721 3···	3,107,418,240	3,286,782,437	9,066,432,567	1,000,000,001	1,634,733,645···	1,762,787,066···	1,900,871,281···
RSA-200	2799783391122132787 0···	2,799,783,391	1,822,351,910	3,893,281,229	3,449,729,239	3,532,461,934···	5,291,297,942···	7,925,869,954···
RSA-768	1230186684530117755 1···	1,230,186,684	6,384,592,519	3,143,412,096	1,537,733,356	3,347,807,169···	3,507,401,723···	3,674,604,366···

표 9. $\lceil\sqrt{n}\rceil-p$와 $q-\lceil\sqrt{n}\rceil$

Table 9. $\lceil\sqrt{n}\rceil-p$ and $q-\lceil\sqrt{n}\rceil$

p	q	n	$\sqrt{n}$	$\lceil\sqrt{n}\rceil$	$\hat{p}$	$\hat{p}-p$	$\lceil\sqrt{n}\rceil-p$	$q-\lceil\sqrt{n}\rceil$
11	13	143	11.96	12	11	0	1	1
11	17	187	13.67	14	11	0	3	3
11	19	209	14.46	15	11	0	4	4
13	17	221	14.87	15	13	0	2	2
13	19	247	15.72	16	13	0	3	3
11	23	253	15.91	16	14	3	5	7
13	23	299	17.29	18	13	0	5	5
11	29	319	17.86	18	15	4	7	11
17	19	323	17.97	18	17	0	1	1
11	31	341	18.47	19	14	3	8	12
13	29	377	19.42	20	15	2	7	9
17	23	391	19.77	20	17	0	3	3
13	31	403	20.07	21	14	1	8	10
11	37	407	20.17	21	15	4	10	16
19	23	437	20.90	21	19	0	2	2
11	41	451	21.24	22	16	5	11	19
11	43	473	21.75	22	18	7	11	21
13	37	481	21.93	22	20	7	9	15
17	29	493	22.20	23	17	0	6	6
11	47	517	22.74	23	19	8	12	24
17	31	527	22.96	23	21	4	6	8
13	41	533	23.09	24	17	4	11	17
19	29	551	23.47	24	19	0	5	5
13	43	559	23.64	24	19	6	11	19
11	53	583	24.15	25	18	7	14	28
19	31	589	24.27	25	19	0	6	6
13	47	611	24.72	25	21	8	12	22
17	37	629	25.08	26	19	2	9	11
11	59	649	25.48	26	20	9	15	33
23	29	667	25.83	26	23	0	3	3
11	61	671	25.90	26	23	12	15	35
13	53	689	26.25	27	20	7	14	26
17	41	697	26.40	27	21	4	10	14
19	37	703	26.51	27	21	2	8	10
23	31	713	26.70	27	23	0	4	4
17	43	731	27.04	28	20	3	11	15
11	67	737	27.15	28	21	10	17	39
13	59	767	27.69	28	23	10	15	31
19	41	779	27.91	28	25	6	9	13
11	71	781	27.95	28	26	15	17	43
13	61	793	28.16	29	22	9	16	32
17	47	799	28.27	29	22	5	12	18
11	73	803	28.34	29	22	11	18	44
19	43	817	28.58	29	24	5	10	14
23	37	851	29.17	30	23	0	7	7
11	79	869	29.48	30	24	13	19	49
13	67	871	29.51	30	24	11	17	37
19	47	893	29.88	30	27	8	11	17
29	31	899	29.98	30	29	0	1	1
17	53	901	30.02	31	23	6	14	22
11	83	913	30.22	31	24	13	20	52
13	71	923	30.38	31	24	11	18	40
23	41	943	30.71	31	26	3	8	10
13	73	949	30.81	31	27	14	18	42
11	89	979	31.29	32	25	14	21	57
23	43	989	31.45	32	26	3	9	11
17	59	1003	31.67	32	27	10	15	27
19	53	1007	31.73	32	27	8	13	21
13	79	1027	32.05	33	25	12	20	46
17	61	1037	32.20	33	25	8	16	28
11	97	1067	32.66	33	28	17	22	64
29	37	1073	32.76	33	29	0	4	4
13	83	1079	32.85	33	29	16	20	50
23	47	1081	32.88	33	30	7	10	14
19	59	1121	33.48	34	28	9	15	25
17	67	1139	33.75	34	29	12	17	33
31	37	1147	33.87	34	31	0	3	3
13	89	1157	34.01	35	26	13	22	54
19	61	1159	34.04	35	26	7	16	26
29	41	1189	34.48	35	29	0	6	6
17	71	1207	34.74	35	30	13	18	36
23	53	1219	34.91	35	32	9	12	18
17	73	1241	35.23	36	28	11	19	37
29	43	1247	35.31	36	29	0	7	7
13	97	1261	35.51	36	30	17	23	61
31	41	1271	35.65	36	31	0	5	5
19	67	1273	35.68	36	31	12	17	31
31	43	1333	36.51	37	31	0	6	6
17	79	1343	36.65	37	31	14	20	42
19	71	1349	36.73	37	32	13	18	34
23	59	1357	36.84	37	33	10	14	22
29	47	1363	36.92	37	34	5	8	10
19	73	1387	37.24	38	30	11	19	35
23	61	1403	37.46	38	31	8	15	23
17	83	1411	37.56	38	32	15	21	45
31	47	1457	38.17	39	31	0	8	8
19	79	1501	38.74	39	34	15	20	40
17	89	1513	38.90	39	36	19	22	50
37	41	1517	38.95	39	37	0	2	2
29	53	1537	39.20	40	32	3	11	13
23	67	1541	39.26	40	32	9	17	27
19	83	1577	39.71	40	35	16	21	43
37	43	1591	39.89	40	37	0	3	3
23	71	1633	40.41	41	34	11	18	30
31	53	1643	40.53	41	34	3	10	12
17	97	1649	40.61	41	35	18	24	56
23	73	1679	40.98	41	39	16	18	32
19	89	1691	41.12	42	33	14	23	47
29	59	1711	41.36	42	34	5	13	17
37	47	1739	41.70	42	37	0	5	5
41	43	1763	41.99	42	41	0	1	1
29	61	1769	42.06	43	34	5	14	18
23	79	1817	42.63	43	37	14	20	36
31	59	1829	42.77	43	38	7	12	16
19	97	1843	42.93	43	39	20	24	54
31	61	1891	43.49	44	37	6	13	17
23	83	1909	43.69	44	38	15	21	39
41	47	1927	43.90	44	41	0	3	3
29	67	1943	44.08	45	35	6	16	22
37	53	1961	44.28	45	37	0	8	8
43	47	2021	44.96	45	43	0	2	2
23	89	2047	45.24	46	37	14	23	43
29	71	2059	45.38	46	38	9	17	25
31	67	2077	45.57	46	39	8	15	21
29	73	2117	46.01	47	37	8	18	26
41	53	2173	46.62	47	41	0	6	6
37	59	2183	46.72	47	41	4	10	12
31	71	2201	46.91	47	44	13	16	24
23	97	2231	47.23	48	39	16	25	49

37	61	2257	47.51	48	41	4	11	13
31	73	2263	47.57	48	41	10	17	25
43	53	2279	47.74	48	43	0	5	5
29	79	2291	47.86	48	44	15	19	31
29	83	2407	49.06	50	40	11	21	33
41	59	2419	49.18	50	41	0	9	9
31	79	2449	49.49	50	42	11	19	29
37	67	2479	49.79	50	45	8	13	17
47	53	2491	49.91	50	47	0	3	3
41	61	2501	50.01	51	41	0	10	10
43	59	2537	50.37	51	43	0	8	8
31	83	2573	50.72	51	45	14	20	32
29	89	2581	50.80	51	46	17	22	38
43	61	2623	51.22	52	43	0	9	9
37	71	2627	51.25	52	43	6	15	19
37	73	2701	51.97	52	50	13	15	21
41	67	2747	52.41	53	45	4	12	14
31	89	2759	52.53	53	45	14	22	36
47	59	2773	52.66	53	47	0	6	6
29	97	2813	53.04	54	43	14	25	43
47	61	2867	53.54	54	47	0	7	7
43	67	2881	53.67	54	48	5	11	13
41	71	2911	53.95	54	51	10	13	17
37	79	2923	54.06	55	44	7	18	24
41	73	2993	54.71	55	49	8	14	18
31	97	3007	54.84	55	50	19	24	42
43	71	3053	55.25	56	46	3	13	15
37	83	3071	55.42	56	47	10	19	27
53	59	3127	55.92	56	53	0	3	3
43	73	3139	56.03	57	46	3	14	16
47	67	3149	56.12	57	47	0	10	10
53	61	3233	56.86	57	53	0	4	4
41	79	3239	56.91	57	53	12	16	22
37	89	3293	57.38	58	49	12	21	31
47	71	3337	57.77	58	52	5	11	13
43	79	3397	58.28	59	49	6	16	20
41	83	3403	58.34	59	50	9	18	24
47	73	3431	58.57	59	51	4	12	14
53	67	3551	59.59	60	53	0	7	7
43	83	3569	59.74	60	54	11	17	23
37	97	3589	59.91	60	56	19	23	37
59	61	3599	59.99	60	59	0	1	1
41	89	3649	60.41	61	52	11	20	28
47	79	3713	60.93	61	58	11	14	18
53	71	3763	61.34	62	53	0	9	9
43	89	3827	61.86	62	57	14	19	27
53	73	3869	62.20	63	53	0	10	10
47	83	3901	62.46	63	54	7	16	20
59	67	3953	62.87	63	59	0	4	4
41	97	3977	63.06	64	53	12	23	33
61	67	4087	63.93	64	61	0	3	3
43	97	4171	64.58	65	57	14	22	32
47	89	4183	64.68	65	58	11	18	24
53	79	4187	64.71	65	58	5	12	14
59	71	4189	64.72	65	59	0	6	6
59	73	4307	65.63	66	59	0	7	7
61	71	4331	65.81	66	61	0	5	5
53	83	4399	66.32	67	57	4	14	16
61	73	4453	66.73	67	61	0	6	6
47	97	4559	67.52	68	59	12	21	29
59	79	4661	68.27	69	59	0	10	10
53	89	4717	68.68	69	62	9	16	20
67	71	4757	68.97	69	67	0	2	2
61	79	4819	69.42	70	61	0	9	9
67	73	4891	69.94	70	67	0	3	3
59	83	4897	69.98	70	68	9	11	13
61	83	5063	71.15	72	61	0	11	11
53	97	5141	71.70	72	65	12	19	25
71	73	5183	71.99	72	71	0	1	1
59	89	5251	72.46	73	64	5	14	16
67	79	5293	72.75	73	67	0	6	6
61	89	5429	73.68	74	67	6	13	15
67	83	5561	74.57	75	67	0	8	8
71	79	5609	74.89	75	71	0	4	4
59	97	5723	75.65	76	68	9	17	21
73	79	5767	75.94	76	73	0	3	3
71	83	5893	76.77	77	71	0	6	6
61	97	5917	76.92	77	73	12	16	20
67	89	5963	77.22	78	67	0	11	11
73	83	6059	77.84	78	73	0	5	5
71	89	6319	79.49	80	71	0	9	9
73	89	6497	80.60	81	73	0	8	8
67	97	6499	80.62	81	73	6	14	16
79	83	6557	80.98	81	79	0	2	2
71	97	6887	82.99	83	81	10	12	14
79	89	7031	83.85	84	79	0	5	5
73	97	7081	84.15	85	73	0	12	12
83	89	7387	85.95	86	83	0	3	3
79	97	7663	87.54	88	79	0	9	9
83	97	8051	89.73	90	83	0	7	7
89	97	8633	92.91	93	89	0	4	4

Ⅳ. 결 론

본 논문은 RSA 암호를 효율적으로 해독하기 위한 소인수 분해법을 연구하였다. 제안된 방법은 $n=pq$의 소인수 p의 범위를 보다 축소시키는 방법이다. 지금까지는 $\sqrt{n}$ 정보로부터 $l(p)=l(q)=l(\sqrt{n})=0.5l(n)$의 $[10\cdots01, 99\cdots9]$ 범위에서 $10\cdots00<p<\sqrt{n}$과 $\sqrt{n}<q<99\cdots9$에 존재한다는 사실만이 밝혀졌다. 본 논문은 n의 정보를 이용하여 p의 범위를 보다 축소시키는 방법을 제안하였다. 제안 방법은 $n=n_{LR}+n_{RL}$으로 분할하여 $p_{\min}=n_{LR}, q_{\min}=n_{RL}$로 설정하는 방법을 적용하였다. 또한, $[q_{\min},q_{\max}]$ 값의 변경에 따라 $[p_{\min},p_{\max}]$의 범위가 조정되었다. 또한, 보다 효율적인 탐색 방법도 제안하였다.

본 논문에서 제안한 n의 정보로 p의 범위를 축소하는 방법은 $\sqrt{n}$의 정보로 p의 범위 축소 방법에 비해 최소 17.79%에서 최대 90.17%의 범위 축소 효과를 얻었다.

References

[1] C. Richard and P. Carl, "*Prime Numbers: A Computational Perspective* (2nd ed.)", Berlin, New York: Springer-Verlag, 2005.

[2] D. John, "The Prime Number Theorem", *Prime Obsession: Bernhard Riemann and the Greatest Unsolved Problem in Mathematics*, Washington, D.C.: Joseph Henry Press, 2003.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "*Introduction to Algorithms* (2nd ed.)," MIT Press and McGraw-Hill, 2001.

[4] Wikipedia, "RSA," http://en.wikipedia.org/wiki/Rsa, 2010.

[5] R. P. Brent, "Recent Progress and Prospects for Integer Factorisation Algorithms", Computing and Combinatorics," pp. 3-22, 2000

[6] Wikipedia, "Trial Division," http://en.wikipedia.org/wiki/Trial_Division, 2010.

[7] J. McKee, "Speeding Fermat's Factoring Method," *Mathematics of Computation*, Vol. 68, pp. 1729-1737, 1999.

[8] J. D. Dixon, "Asymptotically fast factorization of integers", *Mathematics of Computation*, Vol. 36, pp. 255 - 260, 1981.

[9] C. Pomerance, "Analysis and Comparison of Some Integer Factoring Algorithms, in Computational Methods in Number Theory," Math. Centre Tract 154, pp. 89-139, Amsterdam, 1982.

[10] K. Thorsten, "On Polynomial Selection for the General Number Field Sieve", *Mathematics of Computation* , Vol.75, No. 256, pp. 2037 - 2047, 2006.

[11] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, *"The Factorization of the Ninth Fermat Number,"* Math. Comp. Vol. 61, pp. 319-349, 1993.

[12] Wikipedia, "RSA Factoring Challenge," http://en.wikipedia.org/wiki/RSA_Factoring_challenge, 2010.

[13] Myeong-Bok Choi, Sang-Un Lee, "The n+1 Integer Factorization Algorithm," The Institute of Internet, Broadcasting and Communication (IIBC), pp.107~112, vol. 11. no.2, April, 2011.

[14] Myeong-Bok Choi, Sang-Un Lee, "The k-Fermat's Integer Factorization Algorithm," The Institute of Internet, Broadcasting and Communication (IIBC), pp.157~164, vol. 11. no.4, August, 2011.

저자 소개

이 상 운(정회원)

- 2007.3 ~ 현재 : 강릉원주대학교 멀티미디어공학과 부교수

<주관심분야 : 소프트웨어 척도, 분석과 설계 방법론, 소프트웨어 신뢰성, 그래프 알고리즘>

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

최 명 복(종신회원)

- 1997~현재 : 강릉원주대학교 멀티미디어공학과 교수
- 2004. 1~현재 : 한국인터넷방송통신학회 이사

<주관심분야 : 지능형 정보검색, 소프트웨어 공학, 알고리즘>

- E-mail :cmb5859@gmail.com