Malaysian Journal of Mathematical Sciences 13(S) August: 111–125 (2019)
Special Issue: The 6th International Cryptology and Information Security Conference (CRYPTOLOGY2018)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

A New Attack on Special-Structured RSA Primes

Ghafar, A.H.A. *[1], Ariffin, M.R.K.[1,2], and Asbullah, M.A.[1,3]

[1] *Laboratory of Cryptography, Analysis and Structure, Institute for Mathematical Research, Universiti Putra Malaysia, Malaysia*
[2] *Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Malaysia*
[3] *Centre of Foundation Studies for Agricultural Science, Universiti Putra Malaysia, Malaysia*

E-mail: amirghafar87@gmail.com
**Corresponding author*

ABSTRACT

RSA cryptosystem has withstand a number of cryptanalysis over the years on its mathematical structures. The cryptanalysis provides the users of the cryptosystem some particular cases where the RSA private keys can be exposed hence diminishes its security elements. In this paper, we discusses a general case of our previous attack on RSA primes. Our attack corresponds to the special-structured RSA primes namely the primes are relatively close to their nearest squared numbers. We also count the number of primes that are vulnerable to our attack. Finally, we present the countermeasure that can be implemented in the RSA key generation algorithm to avoid our attack.

Keywords: RSA cryptosystem, cryptanalysis, RSA primes.

1. Introduction

The growth in numbers of digital applications marked the importance of a secure cryptosystem. It is to ensure the communications between the applications to be confidential while maintaining its integrity. One of the main cryptosystem in use today is RSA cryptosystem which was introduced by Rivest et al. (1978). The cryptosystem utilizes the integer factorization problem (IFP) as one of its security features. The hard mathematical problem depends on the hardness of finding the prime factors of a very large integer which in general is still an infeasible problem since it can only be solved by factoring algorithm in sub-exponential time(Crandall and Pomerance, 2006). However, there are many factoring algorithms that focus on the special instances of primes. These algorithms are able to solve the factorization of an integer in polynomial time if the prime factors of the integer exhibit certain structures that can be manipulated mathematically. This situation consequently poses a danger on RSA cryptosystem if no proper countermeasure is introduced. For that, FIPS (2013) has provided a standard guideline to avoid the usage of such vulnerable primes.

In this paper, we introduce another instances of primes that can lead to a disastrous impact on RSA. The special-structured primes in this paper can be retrieved in a polynomial time if they are used as the RSA primes. We also count the number of these vulnerable primes in terms of n-bit size to show that there are possibilities for these primes to be unknowingly chosen as the RSA primes. Finally, we present a suitable countermeasure to avoid the usage of such primes since there is no method in the standard guideline of RSA to avoid the primes.

1.1 RSA Cryptosystem

A brief on the workings RSA key generation algorithm is discussed in this section. We omit the details of RSA encryption and decryption algorithms since our attack is not related to the algorithms. First, the RSA key generation algorithm generates two non-trivial primes of n-bit sizes, p and q to form a parameter called RSA modulus, N where $N = pq$. Then, the key generation algorithm chooses a suitable e such that $\gcd(e, \phi(N)) = 1$ where $\phi(N)$ is the Euler's phi function of N. Then d is computed such that $ed \equiv 1 \pmod{\phi(N)}$. The parameters (N, e) are called RSA public keys while $(p, q, \phi(N), d)$ are called RSA private keys.

Our attack in this paper describes an effort to factor N in polynomial time. In general, we focus on the structures of p and q. We show that if p and q are

both having a special structure introduced in this paper, N can be factored by the adversary in polynomial time hence exposes the private keys p and q.

1.2 Outline of This Paper

The paper is organized as follows. In Section 2, we describe some significant previous works that have been the motivations and guidelines of our result. Next, we present our attack in Section 3 and describe in Section 4 the method to count the vulnerable primes affected by our attack in the previous section. Then, in Section 5, we introduce the countermeasure to avoid using the vulnerable primes before we conclude our paper in Section 6.

2. Previous Works

There are numbers of factoring algorithms over the years since IFP fascinates mathematicians. One of the earliest algorithm of this type is called Euler's factorization algorithm (Riesel, 2012). It depends on the statement that the product of two sums of two squares is a sum of two squares. If there are primes that satisfy the conditions of the statement then the product of the primes can be factored. This work is almost similar to Fermat's factorization method. The Fermat's method focuses on finding the values of odd integers v and w to factor u such that $u = v^2 - w^2 = (v - w)(v + w)$ (Lehman, 1974). While the method is sometimes less efficient than the trial division method which is basically the simplest strategy in factoring an integer, but the combination of both methods may work on certain instances of a composite number. The strategy used in Fermat gives a motivation to the fastest general-purpose factoring algorithm which is general number field sieve algorithm Lenstra et al. (1993).

In another hand, there are also special-purpose factoring algorithms that specializes on the certain instances of primes. For example, an algorithm by Pollard (1974) can solve the factorization of a number, N which has prime factors, $p_1 \cdot p_2 \cdot \ldots \cdot p_n$ in polynomial time if for $i = 1, 2, \ldots, n$ there exists $p_i - 1$ with small prime factors. That is, $p - 1$ can be broken completely into small prime factors that are less than an integer, L. Another algorithm called elliptic curve factoring algorithm was introduced by Lenstra Jr (1987). It replaces the multiplicative group used in Pollard's $p - 1$ algorithm to the group of points in a random elliptic curve.

2.1 Our Motivation

In our previous work, we investigated the impact of using $p = a^m + 1$ and $b^m + 1$ where $m = 2^i$ with $i = 1, 2, \ldots$ (Ghafar et al., 2018). In the paper, we showed that such $N = pq$ can be factored in polynomial time by computing the value of $\left(\lfloor\sqrt{N}\rfloor - i\right)^m$ where i is a small integer. The work was motivated by a result by Friedlander and Iwaniec (1997) that states there are infinitely many primes in the form of $a^2 + 1$. This shows the significance of our attack since primes in this form are affected by our result. In this paper, we generalize the form of p and q. Particularly, we investigate the result of using $p = a^m + r_p$ and $b^m + r_q$ where $m = 2^i$ with $i = 1, 2, \ldots$ and r_p, r_q are sufficiently small integers.

3. The New Attack

In this section, we discuss our new attack. The next lemma shows the equality of $\sqrt{a^m + r}$ to its integer and decimal forms.

Lemma 3.1. *Let $a, r \in \mathbb{Z}^+$ and $m \geq 2$ be a power of 2. If $\sqrt{a^m + r} = a^{m/2} + \epsilon$ then $\epsilon < \frac{r}{2a^{m/2}}$.*

Proof. Let $a^m + r$ be an integer where $a \in \mathbb{Z}^+$. Then

$$\sqrt{a^m + r} < \sqrt{a^m + \frac{r^2}{4}a^{-m} + r} = \sqrt{(a^{m/2} + \frac{r}{2}a^{-m/2})^2} = a^{m/2} + \frac{r}{2}a^{-m/2}$$

Since $\sqrt{a^m + r} = a^{m/2} + \epsilon$ then $\epsilon < \frac{r}{2a^{m/2}}$. This terminates the proof. □

With result from Lemma 3.1, we can find the lower and upper bounds of $N^{1/2} - (ab)^{m/2}$ in the following lemma.

Lemma 3.2. *Let $a, b \in \mathbb{Z}^+$ and $m \geq 2$ be a power of 2 such that $a < b < 2a$. Suppose $N = (a^m + r_p)(b^m + r_q)$ where $r_p \leq r_q < N^\gamma$. If $r_p < 2a^{m/2}$ and $r_q < 2b^{m/2}$ then $(r_p r_q)^{1/2} < N^{1/2} - (ab)^{m/2} < \frac{r_q}{2} + 2^{\frac{m}{2}-1}r_p + 1$.*

Proof. To prove the lower bound, first we need to show that $a^m r_q + b^m r_p > 2(ab)^{m/2}(r_p r_q)^{1/2}$. Observe that

$$\left(a^{m/2}r_q^{1/2} - b^{m/2}r_p^{1/2}\right)^2 = a^m r_q + b^m r_p - 2(ab)^{m/2}(r_p r_q)^{1/2}.$$

Since $\left(a^{m/2}r_q^{1/2} - b^{m/2}r_p^{1/2}\right)^2$ will always be positive value, it implies that $a^m r_q + b^m r_p > 2(ab)^{m/2}(r_p r_q)^{1/2}$. Then

$$\begin{aligned}\sqrt{(a^m + r_p)(b^m + r_q)} &= \sqrt{(ab)^m + a^m r_q + b^m r_p + r_p r_q} \\ &> \sqrt{(ab)^m + 2(ab)^{m/2}(r_p r_q)^{1/2} + r_p r_q} \\ &= \sqrt{\left(ab^{m/2} + (r_p r_q)^{1/2}\right)^2} \\ &= (ab)^{m/2} + (r_p r_q)^{1/2}\end{aligned}$$

Thus, $\sqrt{(a^m + r_p)(b^m + r_q)} - (ab)^{m/2} = N^{1/2} - (ab)^{m/2} > (r_p r_q)^{1/2}$. To prove the upper bound, since $\sqrt{a^m + r_p} = a^{m/2} + \epsilon_1$ and $\sqrt{b^m + r_q} = b^{m/2} + \epsilon_2$. Then, based on Lemma 3.1,

$$\begin{aligned}N^{1/2} &= \sqrt{(a^m + r_p)(b^m + r_q)} = \sqrt{(a^m + r_p)}\sqrt{(b^m + r_q)} \\ &= (a^{m/2} + \epsilon_1)(b^{m/2} + \epsilon_2) = (ab)^{m/2} + a^{m/2}\epsilon_2 + b^{m/2}\epsilon_1 + \epsilon_1\epsilon_2 \\ &< (ab)^{m/2} + a^{m/2}\frac{r_q}{2b^{m/2}} + b^{m/2}\frac{r_p}{2a^{m/2}} + \frac{r_p}{2a^{m/2}}\frac{r_q}{2b^{m/2}} \qquad (1)\end{aligned}$$

If $r_p < 2a^{m/2}$ and $r_q < 2b^{m/2}$ then

$$\begin{aligned}\frac{r_p}{2a^{m/2}}\frac{r_q}{2b^{m/2}} &= \frac{r_p r_q}{4(ab)^{m/2}} < \frac{4(ab)^{m/2}}{4(ab)^{m/2}} \\ &= 1. \qquad (2)\end{aligned}$$

If $a < b < 2a$, (1) then will become

$$\begin{aligned}N^{1/2} - (ab)^{m/2} &< a^{m/2}\frac{r_q}{2b^{m/2}} + b^{m/2}\frac{r_p}{2a^{m/2}} + 1 \\ &= \left(\frac{a}{b}\right)^{m/2}\frac{r_q}{2} + \left(\frac{b}{a}\right)^{m/2}\frac{r_p}{2} + 1 \\ &< (1)^{m/2}\frac{r_q}{2} + (2)^{m/2}\frac{r_p}{2} + 1 \\ &= \frac{r_q}{2} + 2^{\frac{m}{2}-1}r_p + 1.\end{aligned}$$

This terminates the proof. □

By obtaining the lower and upper bounds of $N^{1/2} - (ab)^{m/2}$ in Lemma 3.2, we proceed with the following theorem to factor $N = pq$ in polynomial time.

Theorem 3.1. *Let $a, b \in \mathbb{Z}^+$ and $m \geq 2$ be a power of 2 such that $a < b < 2a$. Suppose $N = (a^m + r_p)(b^m + r_q)$ be a valid RSA modulus. Let $r_p < 2a^{m/2}$ and $r_q < 2b^{m/2}$ where $\max\{r_p, r_q\} = N^\gamma$. If N^γ is sufficiently small then N can be factored in polynomial time.*

Proof. From Lemma 3.2 we can see that $(r_p r_q)^{1/2} < N^{1/2} - (ab)^{m/2} < \frac{r_q}{2} + 2^{\frac{m}{2}-1} r_p + 1$. Thus,

$$N^{1/2} - \left(\frac{r_q}{2} + 2^{\frac{m}{2}-1} r_p + 1\right) < (ab)^{m/2} < N^{1/2} - (r_p r_q)^{1/2}. \tag{3}$$

Suppose $r_p = N^{\gamma_1}$ and $r_q = N^{\gamma_2}$ are known. Then the difference between the upper and lower bounds of (3) will be

$$\begin{aligned} & N^{1/2} - (r_p r_q)^{1/2} - N^{1/2} + \frac{r_q}{2} + 2^{\frac{m}{2}-1} r_p + 1 \\ & < N^\gamma \left(2^{\frac{m}{2}-1} + \frac{1}{2}\right) - \left((\min\{r_p, r_q\})^2\right)^{1/2} + 1 \\ & = N^\gamma \left(\frac{2^{\frac{m}{2}} + 1}{2}\right) - \min\{r_p, r_q\} + 1 \end{aligned}$$

which is the size for set of numbers to find $(ab)^{m/2}$. If N^γ is sufficiently small, then we can find $(ab)^{m/2}$ in polynomial time. By computing $\left((ab)^{m/2}\right)^2$, we find $(ab)^m$. Next, we can see that

$$\begin{aligned} N - r_p r_q &\equiv (a^m + r_p)(b^m + r_q) - r_p r_q \\ &\equiv (ab)^m + a^m r_q + b^m r_p \\ &\equiv a^m r_q + b^m r_p \pmod{(ab)^m} \end{aligned}$$

By finding the roots of the following quadratic equation

$$X^2 - (a^m r_q + b^m r_p)X + ((ab)^m r_p r_q),$$

we find $x_1 = a^m r_q$ and $x_2 = b^m r_p$. Since r_p and r_q are known, we can can obtain

$$a^m = \frac{x_1}{r_q} \quad \text{and} \quad b^m = \frac{x_2}{r_p}.$$

Thus we can factor N by calculating

$$\frac{N}{b^m + r_q} = a^m + r_p.$$

□

Remark 3.1. *Throughout this paper, we use the term 'sufficiently small' to indicate the size of numbers that are computationally feasible to be brute-forced by current computing machine. This is related to a suggestion from NIST in 2010 that a key space with less than* 2^{80} *elements may be feasible to be brute-forced by the computing machine in the nearest future (Barker et al., 2012). Hence an integer space with less than* 2^{80} *is to be sufficiently small in our case.*

The algorithm to factor $N = pq$ via Theorem 3.1 is as follows:

Algorithm 1 Factoring $N = pq = (a^m + r_p)(b^m + r_q)$ via Theorem 3.1

Require: N, r_p, r_q, m
Ensure: p, q
Set $i = \lceil (r_p r_q)^{1/2} \rceil$.
while $i < \left\lfloor \frac{r_q}{2} + 2^{\frac{m}{2}-1} r_p + 1 \right\rfloor$ **do**
    Set $\sigma = \left(\left\lceil \sqrt{N} \right\rceil - i \right)^2$
    Calculate $z \equiv N - r_p r_q \pmod{\sigma}$
    Solve $X^2 - zX + \sigma r_p r_q = 0$
    Set $x_1 = X_1$ and $x_2 = X_2$
    if $\frac{N}{\frac{x_1}{r_q} + r_p}$ or $\frac{N}{\frac{x_2}{r_p} + r_q} \neq$ integer **then**
        $i++$
    else
    end if
end while
Output $p = x_1$ and $q = x_2$

The following is an example to illustrate Algorithm 1.

Example 3.1. *We use RSA-2048 modulus in this example. Specifically, we*

are given

$$\begin{aligned} N = {} & 193913383192480660613387699697687168706860965376306090932444 8 \\ & 866851480063582681574438048057706400797136513466618366960109 5 \\ & 815578348184279072815340847911939976260386127814159348358831 8 \\ & 065819600683619012858178980402062206103633907115435868094206 3 \\ & 256540418940505568127291793667612093108100283288847882382037 3 \\ & 594731337928412771946828301938628563293346305927440947130188 8 \\ & 976697542902048312445267988574678148456607659510000792603507 6 \\ & 967693003253550321429143142767707366266857511273221104482265 2 \\ & 038629904439346898175153518026147497585149159763034439743562 7 \\ & 051678166446294195271747338407003033269268808148343449770148 5 \\ & 3137639. \end{aligned}$$

If $r_p = 900535$ *and* $r_q = 801217$ *are known, then we set*

$$\begin{aligned} i &= \left\lceil (r_p r_q)^{1/2} \right\rceil \\ &= 849426. \end{aligned}$$

Then we calculate

$$\sigma = \left(\left\lceil \sqrt{N} \right\rceil - i \right)^2 \quad \text{and} \quad z \equiv N - (r_p r_q) \pmod{\sigma} \tag{4}$$

and solve the equation

$$x_{1,2} = X^2 - zX + \sigma r_p r_q = 0 \tag{5}$$

We find that neither $\frac{x_1}{r_q} + r_p$ *nor* $\frac{x_2}{r_p} + r_q$ *are integers. This means* x_1 *and* x_2 *are not our final solutions. It also means* $\sigma \neq (ab)^m$ *at this point. To find the correct* σ*, we continue to search for them by iterating equations (4) and (5) using iterated values of* i*. This search can be done in polynomial time as* i *should be less than* $\frac{r_q}{2} + 2^{\frac{m}{2}-1} r_p + 1 = 1301144$ *as stated in Lemma 3.2. That means operations in (4) and (5) must be repeated at most* $1301144 - 849426 + 1 =$

451719 *times. In this case, when* $i = 851797$ *(*2371^{st} *iteration), we find*

$$\begin{aligned}\sigma &= \left(\left\lfloor\sqrt{N}\right\rfloor - i\right)^2 \\ &= 193913383192480660613387699697687168706860965376306090932444886685148006358268157443804805770640079713651346661836696010958155783481842790728153408479119399762603861278141593483588318065819600683619012858178980402062206103633907115435868094206325654041894050556812729179366761209310810028328888478823820136364422624735950947962275020662781504470936651653105865417319746049233522254247303533290322490869136239193537645247272985605591555129796787303628030327922665738872798367891162749149287861005696959713685738636593354486355902828982494686426996172772753237351210746944843083904775198693703624742535499763659125809936\end{aligned}$$

and

$$\begin{aligned}z &= N - (r_p r_q) \pmod{\sigma} \\ &= 237230308713192461823984553281275847058822409654274335081712869151620520767950583940993508534976945709421572413354831987365164085374902738715910663401099754407923795776744841048295673364177624207479633212436516924671661141682320177268348012747389932414579293418672582331649935925104633223256072298845212634203376608\end{aligned}$$

produces

$$\begin{aligned}\frac{N}{\frac{x_1}{r_q} + r_p} &= 137003867614795364022261360584496271633209962439732321473343457124962291495213754038469819203702161030210516848787964345485387738830063329770436775997607191136323419341789781052736048214528016366278890765702876850890464219455932122991129761048431317671185555880032806666277637011643622625530512124168946150939\end{aligned}$$

which is an integer and

$$\begin{aligned}\frac{N}{\frac{x_2}{r_p}+r_q} = {} & 14153861972546204716991607515293408672586498379096795090 \\ & 45234715633813327222483363462306688824357868992092397689 \\ & 36782644059806867509531431679202497456095447203331524817 \\ & 99603351626050627890990413314169820141621929500991751698 \\ & 98966830625839073322513143453891474436200004141038009977 \\ & 91940162343931070147127595301\end{aligned}$$

which is also an integer. Hence, N has been successfully factored in polynomial time.

Remark 3.2. *Observe that N in Example 3.1 does not exhibit any noticeable structures (such as long adjacent 0's or 1's) in its value. As such, users of RSA have a possibility to have generated such RSA modulus. Thus, Algorithm 1 is valuable for RSA users to preempt usage of such RSA modulus.*

4. The Number of Vulnerable Primes to the New Attack

In this section, we calculate the number of primes having the structures as discussed in Section 3. First, we determine the number of squared numbers that share the same bit size.

Lemma 4.1. *If n is any large positive integer then there are at least $\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor$ squared numbers between 2^{n-1} and 2^n-1.*

Proof. Let $X=\{x_i^2\}_{i=1}^k$ be the set of all squared numbers between 2^{n-1} and 2^n-1. Particularly,

$$2^{n-1}<x_i^2<2^n-1.$$

Then

$$2^{\frac{1}{2}(n-1)}<x_i<(2^n-1)^{\frac{1}{2}} \Rightarrow 2^{\frac{1}{2}(n-1)}<x_i<\left(\left(2^{\frac{n}{2}}-1\right)\left(2^{\frac{n}{2}}+1\right)\right)^{\frac{1}{2}}. \qquad (6)$$

Next, compute the difference between the upper bound and the lower bound

of (6) in integer form. That is,

$$\begin{aligned}\left\lfloor \left(\left(2^{\frac{n}{2}}-1\right)\left(2^{\frac{n}{2}}+1\right)\right)^{\frac{1}{2}}-2^{\frac{1}{2}(n-1)}\right\rfloor &> \left\lfloor \left(\left(2^{\frac{n}{2}}-1\right)\left(2^{\frac{n}{2}}-1\right)\right)^{\frac{1}{2}}-2^{\frac{1}{2}(n-1)}\right\rfloor \\ &= \left\lfloor \left(\left(2^{\frac{n}{2}}-1\right)^{2}\right)^{\frac{1}{2}}-2^{\frac{1}{2}(n-1)}\right\rfloor \\ &= \left\lfloor 2^{\frac{n}{2}}-1-2^{\frac{1}{2}(n-1)}\right\rfloor . \\ &= \left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)-1\right\rfloor .\end{aligned}$$

If n is any large positive integer then

$$\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)-1\right\rfloor \approx \left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor .$$

This terminates the proof. □

Theorem 4.1. *Let $\pi(x)$ be the prime-counting function that gives the number of primes less than or equal to x, for any real number x. Then*

$$\pi(x) \sim \frac{x}{\log x}.$$

Proof. See (Jameson, 2003) □

With the results from Lemma 4.1, we can determine the number of weak primes that are affected by our attack.

Theorem 4.2. *Let a, b, r_p, r_q be integers greater > 0. Let m be a power of 2. Suppose $r_p < 2a^{m/2}$ and $r_q < 2b^{m/2}$ where $\max\{r_p, r_q\} = N^{\gamma}$. Let $x > 0$ be an integer where x^2 is the smallest squared number with n-bit size then the numbers of primes affected by our attack, $\pi_2(x)$ is asymptotic to*

$$\pi_2(x) \sim \frac{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor}{2}\left(\frac{N^{\gamma}}{\log(x)^2}+\frac{N^{\gamma}}{\log\left(x+\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor\right)^2}\right).$$

Proof. First, we recall Prime Number Theorem in Theorem 4.1 that states for any real number x, $\pi(x)$ is asymptotic to

$$\pi(x) \sim \frac{x}{\log x}.$$

Using this value, given two real numbers x_0 and x_1 where $x_0 < x_1 < 2x_0$, we can count the number of primes between the two numbers. That is,

$$\begin{aligned} \frac{x_1}{\log x_1} - \frac{x_0}{\log x_0} &\approx \frac{x_1}{\log x_0} - \frac{x_0}{\log x_0} \\ &= \frac{x_1 - x_0}{\log x_0}. \end{aligned} \tag{7}$$

Let $x > 0$ be an integer where x^2 is the smallest squared number with n-bit. Let $\pi_1(x)$ be the prime-counting function between x^2 and $x^2 + \max\{r_p, r_q\}$. Similar to (7),

$$\begin{aligned} \pi_1(x) = \frac{x^2 + \max\{r_p, r_q\}}{\log\left(x^2 + \max\{r_p, r_q\}\right)} - \frac{x^2}{\log x^2} &\approx \frac{x^2 + \max\{r_p, r_q\}}{\log x^2} - \frac{x^2}{\log x^2} \\ &= \frac{x^2 + \max\{r_p, r_q\} - x^2}{\log x^2} = \frac{\max\{r_p, r_q\}}{\log x^2} \\ &= \frac{N^\gamma}{\log x^2}. \end{aligned}$$

From Lemma 4.1, we know there are approximately $\left\lfloor 2^{\frac{n}{2}}\left(1 - 2^{-\frac{1}{2}}\right)\right\rfloor$ squared numbers with n-bit size where n is a large integer suitably used in RSA. Thus, $\pi_1(x)$ for the consecutive squared numbers are as follows:

$$\begin{aligned} \pi_1(x) &= \frac{N^\gamma}{\log(x)^2} \\ \pi_1(x+1) &= \frac{N^\gamma}{\log(x+1)^2} \\ \pi_1(x+2) &= \frac{N^\gamma}{\log(x+2)^2} \\ &\vdots \\ &\vdots \\ \pi_1\left(x + \left\lfloor 2^{\frac{n}{2}}\left(1 - 2^{-\frac{1}{2}}\right)\right\rfloor\right) &= \frac{N^\gamma}{\log\left(x + \left\lfloor 2^{\frac{n}{2}}\left(1 - 2^{-\frac{1}{2}}\right)\right\rfloor\right)^2}. \end{aligned} \tag{8}$$

The summation of (8) can be represented in the sum of arithmetic progression formula where the number of i terms is multiplied by the sum of the first and

last number in the progression and dividing by 2. That is,

$$\begin{aligned}\pi_2 &= \sum_{i=0}^{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)-1\right\rfloor} \frac{N^{\gamma}}{\log(x+i)^2} \\ &= \frac{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor}{2}\left(\pi_1(x)+\pi_1\left(x+\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor\right)\right) \\ &= \frac{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor}{2}\left(\frac{N^{\gamma}}{\log(x)^2}+\frac{N^{\gamma}}{\log\left(x+\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor\right)^2}\right) \quad (9)\end{aligned}$$

This terminates the proof. □

The following is an example to illustrate the result from Theorem 4.2.

Example 4.1. *In this example we proceed to compute the number of primes used in RSA-2048 that are vulnerable to our attack. From Theorem 4.2, we need to compute*

$$\frac{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor}{2}\left(\frac{N^{\gamma}}{\log(x)^2}+\frac{N^{\gamma}}{\log\left(x+\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor\right)^2}\right).$$

Following Example 3.1, we have $n = 1024$ *and* $\max\{r_p, r_q\} = N^{0.009661\ldots}$ *which implies* $\gamma = 0.009661$. *Observe that* $x = 2^{n-\frac{1}{2}}$ *since* x^2 *is the smallest squared number with n-bit size. Substituting these values into (9), we obtain*

$$\begin{aligned}\pi_2(x) &\sim \frac{\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor}{2}\left(\frac{N^{\gamma}}{\log(x)^2}+\frac{N^{\gamma}}{\log\left(x+\left\lfloor 2^{\frac{n}{2}}\left(1-2^{-\frac{1}{2}}\right)\right\rfloor\right)^2}\right) \\ &\approx 7.0265327\ldots\times 10^{153}. \quad (10)\end{aligned}$$

Thus, there are approximately $7.0265327\ldots\times 10^{153}$ *primes that are susceptible to our attack if RSA-2048 is used.*

5. Countermeasure of the Attack

In this section, we present a countermeasure to prevent using the vulnerable primes discussed in Section 3. The countermeasure is depicted in Figure 1.

Given N, p and q, if

$$\left\lceil N^{1/2} - \left\lfloor p^{1/2} \right\rfloor \cdot \left\lfloor q^{1/2} \right\rfloor \right\rfloor \tag{11}$$

is a sufficiently small integer, then RSA key generation algorithm must find new p or q.

Figure 1: Countermeasure of the attacks shown in Section 3.

Since the computation is minimal, the prevention of the attack can be applied in the real-world RSA implementation.

6. Conclusion

Our new method can successfully factor N in polynomial time given that it satisfies certain conditions as in Theorem 3.1. We also show in Theorem 4.2 that the number of primes which are susceptible to our attack is large and depends on the size of p and q. Our attack includes primes that can be generated by current standard RSA implementation namely RSA-2048 as in Example 3.1. Thus, a new countermeasure should be introduced to the existing guidelines in preventing such attack to occur.

Acknowledgements

The research was supported by Ministry of Education of Malaysia with Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UPM/02/08).

References

Barker, E., Barker, W., Burr, W., Polk, W., and Smid, M. (2012). NIST Special Publication 800-57 Recommendation for Key.

Crandall, R. and Pomerance, C. B. (2006). *Prime numbers: a computational perspective*, volume 182. Springer Science & Business Media.

FIPS, P. (2013). 186-4: Federal information processing standards publication. Digital Signature Standard (DSS). *Information Technology Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD*, pages 20899–8900.

Friedlander, J. and Iwaniec, H. (1997). Using a parity-sensitive sieve to count prime values of a polynomial. *Proceedings of the National Academy of Sciences*, 94(4):1054–1058.

Ghafar, A. H. A., Ariffin, M. R. K., and Asbullah, M. A. (2018). Extending Pollard Class of Factorable RSA Modulus. In *Cryptology and Information Security Conference*, page 103.

Jameson, G. J. O. (2003). *The prime number theorem*, volume 53. Cambridge University Press.

Lehman, R. S. (1974). Factoring large integers. *Mathematics of Computation*, 28(126):637–646.

Lenstra, A. K., Lenstra, H. W., Manasse, M. S., and Pollard, J. M. (1993). The number field sieve. In *The development of the number field sieve*, pages 11–42. Springer.

Lenstra Jr, H. W. (1987). Factoring integers with elliptic curves. *Annals of mathematics*, pages 649–673.

Pollard, J. M. (1974). Theorems on factorization and primality testing. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 76, pages 521–528. Cambridge University Press.

Riesel, H. (2012). *Prime numbers and computer methods for factorization*, volume 126. Springer Science & Business Media.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.