

An evaluation of the RSA private keys and the presence of weak keys

Mahmoud M. Almazari *

Eyad Taqieddin

Department of Network Engineering

Jordan University of Science and Technology

Irbid 21110

Jordan

Ahmed S. Shatnawi

Zakarea Al-Shara

Department of Software Engineering

Jordan University of Science and Technology

Irbid 21110

Jordan

Abstract

Numerous applications that rely on asymmetric cryptography use the RSA algorithm. It can be applied to digital signatures and the encryption of sensitive data. The secure storage of the private key is essential for the algorithm's strength. Finding ways to use factorization or other heuristics to determine the value of the private key, was the goal of several academic efforts. Both the Euler totient and the Carmichael functions are used in this study to analyze the private key properties and demonstrate the existence of many private keys for the same public key. We further demonstrate that a universal private key that complies with the FIPS standard exists. Moreover, by taking advantage of a condition imposed by FIPS recommendations, we present a new method for attacking the RSA modulus (N).

Subject Classification: (2010) 68P25, 94A60, 94A62.

Keywords: Cryptanalysis, RSA attacks, Two private keys, Universal RSA key, RSA factorization, Weak RSA keys, RSA FIPS conditions.

* E-mail: mmalmazari16@cit.just.edu.jo (Corresponding Author)

1. Introduction

The RSA encryption algorithm was invented by Ron Rivest, Adi Shamir, and Len Adleman as an asymmetric cipher to deliver privacy and authenticity of messages [1]. It is widely used for electronic payments, secure e-mail, and other web traffic requiring secure data transfer.

The algorithm is based on number theory in its core with a straightforward procedure as listed below:

- (1) Two large prime numbers, p and q , are chosen to compute the RSA modulus $N = p \cdot q$
- (2) Compute the Euler totient function, which represents the order of the multiplicative group Z_N^* , as $\phi(n) = (p-1) \cdot (q-1)$
- (3) Choose a value of e , the encryption exponent, that is relatively prime to $\phi(N)$ and use it to compute d , the decryption exponent, such that $e \cdot d \equiv 1 \pmod{\phi(N)}$

Note that any integer d which is co-prime with the Euler totient function and satisfies the condition $\frac{de-1}{\phi(n)} = k$, where k is an integer, is considered a private key. The pair (N, e) is used as the public key and the value of d must be secretly stored. To encrypt a message, M , we use the public key to compute the cipher message, C , as $C = M^e \pmod{N}$. Decryption is done in the reverse by computing $M = C^d \pmod{N}$. One possible attack against RSA is to factorize N to p and q and then run the steps above. However, factorization itself was shown to run in exponential or sub-exponential time.

This paper covers multiple contributions. We first present an analysis of the private key and prove the existence of more than just one key [2]. This is shown to apply to implementations of the Euler totient function as well as those using the Charmichael function. We also show that RSA has a universal key as well as a set of weak keys. Finally, we present an attack to factor N when specific conditions are satisfied.

2. Related Work

Many RSA cryptanalytic techniques were proposed in the literature. However, no one yet competes for a total break of the large keys. Some of these techniques are based on mathematical algorithms like factorization attacks [3]. Most of these techniques are based on relaxed models by

using partial information, weak exponents (public/private) or other considerations [3].

Attacks on the RSA aim to identify the two prime factors (p and q) of the RSA modulus N . One such attack can be used to break RSA with short secret exponents was proposed in [4]. This attack uses a continued fraction algorithm in polynomial time when $e < p \cdot q, \text{GCD}(p-1, q-1)$ is small, p and q have approximately the same number of bits, and $d < \frac{1}{\sqrt[4]{18}} N^{0.25}$. However, the continued fraction algorithm is not guaranteed to work if a large $\text{GCD}(p-1, q-1)$ is chosen or if $e > (pq)^{1.5}$.

After that, the authors of [5] used the continued fraction algorithm and Coppersmith's method to identify large weak exponents used for factorizing N in polynomial time. Consequently, breaking the RSA. The work in [6], proposed three new attacks on RSA using continued fractions with the modulus $N = p^2q$. These attacks could be used to factorize N in polynomial time. As complementary to the Boneh's Durfee attack [7], another continued fraction technique attack was proposed in [8]. This attack can be used when e is much smaller than N with the running time of $O(t \log(N))$. In [9], one of the best attacks on RSA is proposed using the lattice technique with a short RSA secret key. However, this lattice technique attack is not efficient if $d \geq N^{0.292}$. Another work [10] proposed an attack on some special case when $d > e$ of Boneh and Durfee's attack method. Using lattice basis reduction method, the approach proved that its shortest polynomial vector to find the weak keys.

A tool that can efficiently and completely identify any weak keys was presented in [11]. The tool uses NVIDIA's GPU and the CUDA massively-parallel programming model to compare a large number of 1024-bit RSA public keys. The tool speeds up the comparison by 27.5 times compared to a sequential CPU implementation. Another approach was given in [12] with a proof of concept of the GPGPU-accelerated system to identify weak keys. The experiments showed that the fast Euclidean algorithm has a good performance by using GPU automatic caching rather than a simpler GPU code.

Many research proposed techniques to analyze the fraction of encrypted connections between cryptographic keys and cryptographic libraries. [13] Trained more than 150 million public and private key pairs on a Bayesian model. The pair keys are collected from 70 cryptographic libraries, cryptographic smartcards and hardware security modules. The model classified the pair keys based on the chosen cryptographic libraries that are responsible for producing GCD-factorable keys. The

result achieved by using Private and public keys significantly increases the classification performance compared to the previous works used only public keys.

3. The First Set Of Private Keys

Referring to equation (1), if we take d_1 as the private key for some $k_1 = \alpha$, we can find another private key, d_2 using a second integer $k_2 = \alpha + e$ for some integer $e \in \mathbb{N}$. In other words, the second private key in the first set of the private keys will be

$$d_2 = \frac{(\alpha + e) \cdot \phi(n) + 1}{e} \quad (1)$$

Corollary 3.1 : *The first set of the private keys, denoted as d_x , for a given public key (n, e) , is:*

$$d_x = \frac{(\alpha + X \cdot e) \cdot \phi(n) + 1}{e}$$

Proof:

$$d_x = \frac{\alpha \cdot \phi(n) + X \cdot e \cdot \phi(n) + 1}{e} = \frac{X \cdot e \cdot \phi(n)}{e} + \frac{\alpha \cdot \phi(n) + 1}{e} = X \cdot \phi(n) + \frac{\alpha \cdot \phi(n) + 1}{e}$$

The second part of the sum is d_1 , thus $d_x = X \cdot \phi(n) + d_1, \forall X \in \mathbb{N}$. Accordingly, the distance between any two successive keys in the first set is $\phi(n)$. Next, we show that the key d_x is a valid key that can be used to decrypt the ciphertext C to yield the plaintext (M) .

$$\begin{aligned} C^{d_x} \bmod N &= C^{(X \cdot \phi(n) + d_1)} \bmod N \\ ((C^{X \cdot \phi(n)} \bmod N) * (C^{d_1} \bmod N)) \bmod N \\ ((1) * (C^{d_1} \bmod N)) \bmod N &= M \end{aligned}$$

4. The Existence of A Second Private Key

The Carmichael function, as opposed to the Euler totient function, is used in present-day implementations of the PKCS#1 standard. The Carmichael function allows for the generation of a private key that is both unique and of the shortest possible size. In this section, we demonstrate the impact of the Carmichael function and the conditions in which a

second private key that is not part of the first set of private keys will exist. In addition, we'll talk about the conditions under which a second private key will exist.

Theorem 4.1 : *Euler's totient theorem states that if $\gcd(n, a) = 1$, n and a are positive integers, then $a^{\phi(n)} \equiv 1 \pmod{n}$. Also, using the reduced totient function (Carmichael function) $a^{\lambda(n)} \equiv 1 \pmod{n}$*

Theorem 4.2 : *The unique factorization theorem states that every positive integer ($n \geq 2$) can be represented in exactly and only one way as a product of prime powers:*

$$n = \prod_{i=1}^k p_i^{n_i}$$

where $p_1 < p_2 < \dots < p_k$ are primes and the n_i are positive integers.

Referring to Theorem 4.2, $\lambda(n) = \alpha_1^{\beta_1} \cdot \alpha_2^{\beta_2} \cdot \alpha_3^{\beta_3} \dots$. Let δ be an integer such that $\delta = \gamma_1^{v_1} \cdot \gamma_2^{v_2} \cdot \gamma_3^{v_3} \dots$, where γ_x could be equal to α_x , then:

$$\begin{aligned} a^{\delta \cdot \lambda(n)} &\equiv 1 \pmod{n} \\ (a^{\lambda(n)} \pmod{n})^\delta &= 1^\delta = 1 \end{aligned}$$

The Carmichael function is equal to the least common multiple of $(p-1)$ and $(q-1)$, due to the fact that p and q are odd numbers, then one of the common factors of $p-1$ and $q-1$ is 2 (i.e. $\text{GCD}(p-1, q-1) \geq 2$, and $\delta \geq 2$). As a consequence $\phi(n) \neq \lambda(n)$. However, for a private key computed as $d_1 = e^{-1} \pmod{\phi(n)}$ and another private key $d_2 = e^{-1} \pmod{\lambda(n)}$, where $\lambda(n) = \phi(n) / \gcd(p-1, q-1)$, then there is a possibility that $d_1 \neq d_2$ which yields generating a new set of private keys, d_y and the existence of a second private key. However, there are some conditions where some keys in the set d_x will be equal to a key in the set d_y , as we present next.

Theorem 4.3 : *Based on Euler theorem and RSA:*

$$d_x \cdot e - \alpha \cdot \phi(n) = 1$$

and the second set will be

$$d_y \cdot e - \beta \cdot \frac{\phi(n)}{\gcd(p-1, q-1)} = 1$$

Now, if $\gcd(p-1, q-1) \mid \beta$ then $d_x \equiv d_y$. In other words, if β and $\gcd(p-1, q-1)$ are relatively prime, then $d_x \neq d_y$.

Corollary 4.4 : *The second set of the private keys will be*

$$d_{y_x} = \frac{(\beta + X.e).\lambda(n)+1}{e}, \text{ and } \beta = \frac{d_{y_0} \cdot e - 1}{\lambda(n)} \quad (1)$$

With a proof similar to that in section 1, it can be shown that the distance between any two successive keys in the second set is $\lambda(n)$. One interesting observation is that there is no strict relation between α in the first set of the private keys and β in the second. However, if $d_x \equiv d_y$ then

$$\beta = \gcd(p-1, q-1) \cdot \alpha \quad (3)$$

where α is calculated based on totient function, and β is calculated based on the reduced totient function, as shown earlier. We illustrate this with a couple of examples.

Example 1: $q = 100049$, $p = 465947$, $e = 1303$ Then, $\lambda(n) = 1792960208$ and $\phi(n) = 46616965408$.

The private key using $\phi(n)$, $d = 24471223591$ and the private key using $\lambda(n)$, $d = 1162740887$.

Example 2: $q = 100019$, $p = 465989$, $e = 1303$ Then, $\lambda(n) = 23303593892$ and $\phi(n) = 46607187784$.

The private key using $\phi(n)$, $d = 3219222487$ and the private key using $\lambda(n)$, $d = 3219222487$.

In the first example, it is clear that the two keys are not equal. This is further supported by the fact that the values of $\alpha = 684$ and $\beta = 845$ with the $\gcd(p-1, q-1) = 26$ do not hold in equation (3). Nevertheless, in the second example, the two keys are equal. Note that $\alpha = 90$ and $\beta = 180$ with the $\gcd(p-1, q-1) = 2$ which satisfy equation (3).

The equality between the keys in the two sets is not necessarily dependent on their position in the set. A key in the set d_x at position i can equal a key in d_y at position j , where $i \neq j$, according to equation (3). The question becomes, how do we determine the values of i and j to relate these two set together? Considering the distance between the first keys in both sets, (d_x) and (d_y) ,

$$d_{x_0} - d_{y_0} = \frac{\alpha \cdot \gcd(p-1, q-1) \cdot \lambda(n) + 1}{e} - \frac{\beta \cdot \lambda(n) + 1}{e}$$

$$d_{x_0} - d_{y_0} = \frac{\lambda(n)}{e} \cdot (\alpha \cdot \gcd(p-1, q-1) - \beta)$$

$$dx_i - dy_i = \lambda(n) \cdot w$$

Because $\lambda(n)$ and e are relatively prime and $d_{x_0} - d_{y_0}$ is integer, then $e \mid (\alpha \cdot \gcd(p-1, q-1) - \beta)$, which yields $w = (\alpha \cdot \gcd(p-1, q-1) - \beta) / e$ is some positive integer value that represents the number of keys needed to be passed in the second set to find a key that is equal to the first key in the first set. In other words, $dy_w = dx_0$, which can be seen in the following example 3.

Example 3: $q = 100213$, $p = 465781$, $e = 1303$. We compute $\lambda(n) = 555675540$, $\phi(n) = 46676745360$, $\alpha = 700$, $\beta = 165$, and $\gcd(p-1, q-1) = 84$. Thus, $w_0 = 45$. The private key calculations can be seen in Table 1 where you may notice that $dx_0 = dy_{45}$.

Also, the distance between two keys at any position v is directly proportional to the $(\gcd(p-1, q-1) - 1)$, as explained earlier:

$$w_0 = (d_{x_0} - d_{y_0}) / 555675540 = 0 \cdot (84 - 1) + 45 = 045$$

$$w_1 = (d_{x_1} - d_{y_1}) / 555675540 = 1 \cdot (84 - 1) + 45 = 128$$

$$w_2 = (d_{x_2} - d_{y_2}) / 555675540 = 2 \cdot (84 - 1) + 45 = 211$$

$$w_3 = (d_{x_3} - d_{y_3}) / 555675540 = 3 \cdot (84 - 1) + 45 = 294$$

$$w_4 = (d_{x_4} - d_{y_4}) / 555675540 = 4 \cdot (84 - 1) + 45 = 377$$

Table 1

Example 3 - Distance between private keys sets.

$dx_0 = 25075764967$	$dy_0 = 70365667$
$dx_1 = 71752510327$	$dy_1 = 626041207$
$dx_2 = 118429255687$	$dy_2 = 1181716747$
$dx_3 = 165106001047$	$dy_3 = 1737392287$
$\vdots$	$\vdots$
$dx_{42} = 1985499070087$	$dy_{42} = 23408738347$
$dx_{43} = 2032175815447$	$dy_{43} = 23964413887$
$dx_{44} = 2078852560807$	$dy_{44} = 24520089427$
$dx_{45} = 2125529306167$	$dy_{45} = 25075764967$

5. The Existence of A Multi Private Key

In the previous section, we showed that any private key can be extended to set of private keys. Going one step beyond that, we demonstrate in this section the existence of multi-private keys from which further sets can be found.

When generating the keys in the RSA cryptosystem, we seek to find the e modular inverse of $\lambda(n)$. In that process, we aim to find the public key's modular inverse of any integer that contains the λ factors within it. By examining the above equation where $\delta = \gamma_1^{\nu_1} \cdot \gamma_2^{\nu_2} \cdot \gamma_3^{\nu_3} \cdots$, it presents that there are multi sets of private keys. Consider $\phi(pq) = \lambda(pq) \cdot \gcd(p-1, q-1)$, and from the previous sections we know that Euler's totient function is replaced by Carmichael function. As a result, we can express that as $\delta = \gcd(p-1, q-1)$.

To prove that such keys are valid, we consider the decryption of the ciphertext to find the plaintext.

Proof:

$$\begin{aligned} c^d \bmod N &= c^{\left(\frac{k \cdot \phi(n)+1}{e}\right)} \bmod N = M^{(k \cdot \phi(n)+1)} \bmod N \\ &= ((M^{k \cdot \phi(n)} \bmod N) \cdot (M^1 \bmod N)) \bmod N \\ &= (M^{k \cdot \lambda(n) \cdot \gcd(p-1, q-1)} \bmod N) \cdot (M^1 \bmod N) \\ &= (M^{\lambda(n)} \bmod N)^{k \cdot \delta} \cdot (M^1 \bmod N) \\ &= (1)^{k \cdot \delta} \cdot (M^1 \bmod N) = M \end{aligned}$$

6. RSA Universal Private Key And A New Set Of Weak Keys

In the previous section, we demonstrated that the private key d is any number that is $d = e^{-1} \bmod s$, where $s = k \cdot \lambda(n)$, k is any integer such that $\gcd(s, e) = 1$. In other words, there exists a key that can decrypt any RSA modulus (a universal key) without knowing the factorization of N . Consider n to be the number of bits of N then,

$$d_{uni} = e^{-1} \bmod \frac{(2^{n/2})!}{\gcd((2^{n/2})!, e^v)} \quad (4)$$

where v is a huge exponent to remove e values from the factorial. Such a key will be huge and is computationally infeasible to be generated. For example, with RSA-2048, the size of the key will be around $2.14 \cdot 10^{301}$

GiB. In other words, the factorial will have a size of gp^{g^2} , where gp is the googolplex number and g is the googol number.

Therefore, to overcome this limitation, we use a flaw to define a new set of weak RSA keys that can satisfy the FIPS recommendation. The factorial is a massive increasing function, despite the length and size that is needed to calculate it, we have managed to compute the factorial of the prime numbers up to 2^{24} , which does not require a huge computation power. The private key size after computation was 0.274 MiB, such a key can decrypt any message encrypted with the same e if and only if the biggest factor of $(p-1, q-1)$ is lower than 2^{24} . We used $e = 2^{16} + 1$ but it can be easily changed to a different value, the code and the universal private key can be found at GitHub in [almazariGitHub](#).

Theorem 6.1: *If $p-1 = \alpha_1^{\beta_1} \cdot \alpha_2^{\beta_2} \cdot \alpha_3^{\beta_3} \dots$ and $q-1 = \gamma_1^{\nu_1} \cdot \gamma_2^{\nu_2} \cdot \gamma_3^{\nu_3} \dots$. Then N is considered a broken and weak key if and only if $\max(\alpha_1, \alpha_2, \alpha_3, \gamma_1, \gamma_2, \gamma_3 \dots) \leq 2^{(r)}$, where 2^r depends on the computational power to compute the factorial of prime numbers from 1 up to 2^r .*

If we want a factorial of the prime numbers up to 2^{40} then such a key size will be $2^{\frac{2^{40}}{\ln(2^{40})} \cdot 40/3}$ bits, where $(40/3)$ is the average size of each prime from 1 to 2^{40} , by rough estimation, the maximum size of such a private key is 62 GiB.

7. A New Attack on RSA To Factor N

In the previous section, we discussed the existence of a universal key that can break a wide range of RSA pairs that satisfy the FIPS recommendations. In this section, we discuss a new attack and a new set of weak RSA keys by exploiting a vulnerability of the factors of $p-1, q-1$ that will lead to the factorization of N . Unfortunately, FIPS recommendations require the private key size to be bigger than $2^{n/2}$, where n is the modulus size, we show that if the private key is a little bigger than $2^{n/2}$, we can factorize N .

Theorem 7.1: *Suppose that $q-1 = A \cdot C$, $p = B \cdot C$, where $C = \alpha_1^{\beta_1} \cdot \alpha_2^{\beta_2} \cdot \alpha_3^{\beta_3} \dots$, then we can factorize N efficiently in polynomial time if and only if $A \cdot B$ can be brute forced.*

Since $\lambda(n)$ and $\gcd(p-1, q-1)$ are inversely proportional, a huge GCD implies small $\lambda(n)$ which results in a small private key size which can be used to reduce computational time and power. Nevertheless, the private key will be still bigger than $2^{n/2}$ when e is small which what FIPS recommends. Based on Theorem 7.1 where $C = \gcd(p-1, q-1)$, we exploit this GCD in order to factorize N as follows:

$$\begin{aligned} q-1 &= A.C, \text{ and } p-1 = B.C \\ N = pq &= (A.C+1)(B.C+1) = A.B.C^2 + (A+B).C + 1 \\ A.B.C^2 &+ (A+B).C + (1-N) = 0 \end{aligned} \quad (5)$$

If the size of A and B is small, we can use brute force and solve the quadratic equation in polynomial time. This attack will succeed due to the small values of A and B . One interesting observation is that the countermeasures of the Wiener Attack recommend increasing $\gcd(p-1, q-1)$ [4]. However, as shown in this section, such a measure can be exploited in this attack. The code and RSA-2048 example that satisfies FIPS recommendations can be found at GitHub in [almazariGitHub](#).

8. Extending The Attack on RSA To Factor N

In the previous section, a new attack was introduced on RSA that targets a weakness in the FIPS recommendation. The attack also, can be extended such that if there is a big common factor between $q-\alpha$ and $p-\beta$ where $\alpha, \beta \in \mathbb{Z}$ then:

$$\begin{aligned} q-\alpha &= A.C, \text{ and } p-\beta = B.C \\ N = pq &= (A.C+\alpha)(B.C+\beta) = A.B.C^2 + (\beta A + \alpha B).C + \alpha\beta \\ A.B.C^2 &+ (\beta A + \alpha B).C + (\alpha\beta - N) = 0 \end{aligned} \quad (6)$$

Notice that $\alpha, \beta \in \mathbb{Z}$, which means that assuming $q+\alpha$ and $p+\beta$ is also possible. Nevertheless, by assuming the size of α, β, A and B to be n_α, n_β, n_A and n_B bits, respectively, the previous equation can be solved and factor N in $O(2^{(n_\alpha + n_\beta + n_A + n_B)})$.

9. Conclusion

In this paper, we presented a detailed analysis related to the existence of multiple sets of private keys (rather than a single key). We further showed a relationship between the keys generated using the Euler totient function and those created using the Charmichael function. In

our analysis, we proved the existence of universal key where we define a weak RSA keys that generate ciphertexts that can be decrypted using the universal key without factoring the modulus N where these keys meet the FIPS recommendations. Finally, we illustrated a new attack on RSA that would yield the factors used in computing N by exploiting a condition set by FIPS recommendations based on the value of the private key being greater than $2^{n/2}$.

References

- [1] Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 21(2) : 120–126 (1978).
- [2] Ibrishimova M. Proving the existence of a second private key that decrypts a message encrypted with the RSA encryption algorithm [<https://marinaibrishimova.net/docs/otherkeys.pdf>]; 2017. (Accessed on 03/19/2022).
- [3] Mumtaz M, Ping L. Forty years of attacks on the RSA cryptosystem: A brief survey. *Journal of Discrete Mathematical Sciences and Cryptography*. 22(1) : 9–29 (2019).
- [4] Wiener MJ. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*. 36(3) : 553–558 (1990).
- [5] Meng X, Bi J. Weak keys in RSA with primes sharing least significant bits. In: *International Conference on Information Security and Cryptology*; Springer. p. 278–287 (2009).
- [6] Abd Rahman NN, Ariffin MRK. New practical attack on RSA modulus of type $n = p_2q$. *International Journal of Pure and Applied Mathematics*. 110(4) : 587–607 (2016).
- [7] Boneh D, Durfee G, Frankel Y. An attack on RSA given a small fraction of the private key bits. In: *International Conference on the Theory and Application of Cryptology and Information Security*; Springer. p. 25–34 (1998).
- [8] Susilo W, Tonien J, Yang G. Divide and capture: An improved cryptanalysis of the encryption standard algorithm RSA. *Computer Standards & Interfaces*. 74 : 103470 (2021).
- [9] Boneh D, Durfee G. Cryptanalysis of RSA with private key d less than $n/\sup 0.292$. *IEEE transactions on Information Theory*. 46(4) : 1339–1349 (2000).

- [10] Mumtaz M, Ping L. Cryptanalysis of a special case of RSA large decryption exponent using lattice basis reduction method. In: 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS); IEEE. p. 714–720 (2021).
- [11] Scharfglass K, Weng D, White J, et al. Breaking weak 1024-bit RSA keys with cuda. In: 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies; IEEE. p. 207–212 (2012).
- [12] Russek P, Karbownik P, Wiatr K. Weak RSA key discovery on gpgpu. *International Journal of Electronics and Telecommunications*. 65 (2019).
- [13] Janovsky A, Nemec M, Svenda P, et al. Biased RSA private keys: Origin attribution of gcdfactorable keys. In: European Symposium on Research in Computer Security; Springer. p. 505–524 (2020).
- [14] Almazari MM. Universal key calculation and factorization attack [Github repository] (2022). Available from: <https://github.com/lcsig/RSALWeakKeys>.

Received April, 2022

Revised August, 2022