Analysis of RSA and Shor’s Algorithm for Cryptography: a Quantum Perspective

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Seema Kute1 | Dr. Chitra Desai2 | Dr. Mukti Jadhav3 |
| Dept. of CS & IT | Dept. of CS | Dept. CS |
| Dr. B. A. M. U. | NDA Khadakwasla | Shri Shivaji Science and Arts College |
| Aurangabad, India | Pune, India | Chikhali, India |
| seemakute63@gmail.com | chitragdesai@gmail.com | muktijadhav@gmail.com |
|  |  |  |

ABSTRACT

The complexity of public key algorithms like RSA and variants of Elgamal is based on the involvement of exponentiation in encryption and decryption. RSA is based on integer factorization and variants of Elgamal or based on discrete logarithms and are formed as hard problems. To achieve the perceived level of security, the required key length for public key cryptosystem needs to be determined. Quantum computers are capable of conducting an exponential number of simultaneous computations because of which they can find factors in polynomial time. Shor’s algorithm is popular algorithm for factoring huge integers in polynomial time on quantum computers. This paper presents RSA algorithm followed by various possible attacks for factoring and demonstrates Shor's algorithm for factoring and discuss results for interpretations.

Keywords— Prime Factorization; Pollard rho; Quantum Computer; Shor Algorithm.

#  INTRODUCTION

 The network communication has gained enormous importance in past few years because increased use for social media, e-banking, e-commerce, video-conferencing, etc. underlying to each application for communication is the requirement of assured network security with the advancement in technology which is experienced on day to day basis, ensuring security for the today’s network information may not remain promising for tomorrow. That is the reason when we look upon various cryptographic algorithms we have seen them evolving over period of time. The objective being making it more and more difficult to create with the upcoming technological advancements. Several attempts have been made by researchers to propose new algorithms for encryption and decryption using various mathematical models. These algorithms mainly are based on either integer factorization [18][19],descrete logarithm[20] or elliptical curves[21][22][23]. Also, several variations which make hybrid combination of these to add more and more complexity to ensure harder to break these algorithms and their variants have proposed they are best in their time and has also gained commercial importance in various applications. Just like there are algorithms to ensure secure encryption and decryption which fundamentally works on a simple mathematical aspect of fractions ‘f’ and inverse of the fraction ‘f’, there are algorithms which are designed to break the cryptographic algorithms. For example pollard rho [24][25], baby step gaint step [26][27],etc. The today’s day challenge for various cryptographic algorithms is not limited to sustain the attack of these algorithms but also something greater. That is, these algorithms are designed for classical computation and have proved best. However, when these algorithms which are designed for classical computation systems, are attacked by quantum computations, the things to works.

 The functioning of classical computation is based on bits and that quantum com-puters is based in super positioning of bits, that is qubits. Classical computation works on the basic principle of classical mechanics, whereas, the quantum computer works on the principles of quantum theory. The biggest advantage of quantum computer over classical computer is that it is millions of times faster than the most sophisticated super computers in the world today. That is, quantum computers are so faster than classical computers, that what classical computers will require thousands of years to solve, quantum computers will do it in minutes. That is the biggest threat to the cryptographic algorithms designed for classical computers.

 In 1994, Peter Shor, a scientist working for Bell Labs, came up with a polyno-mial time algorithm for factoring big numbers on quantum computers.[1][2][3][4] This determinations drew appreciable attention to the domain of quantum computing. This paper discusses Pollard Rho classical algorithm and Shor’s algorithm is able to break integer factorization based cryptographic algorithm Rivest–Shamir–Adleman (RSA) Algorithm. And analyze the both pollard Rho classical algorithm and Shor’s quantum algorithm on the basis of integer breaking (factoring) time. In a brief, this work will demonstrate how within less time Shor’s algorithm creates threat to secrecy and authentication services as compared to Pollard Rho.

#  RSA ALGORITHM STEPS

1. Pick any two random prime numbers Prime1 and Prime2
2. Calculate the product of Prime1 and Prime2 random numbers N = Prime1 \* Prime2
3. Calculate O(N) =(Prime1 – 1)\*(Prime2 – 1)
4. Pick integer enc and N are coprime
5. Compute decryption value (dec\*enc)%O(N) = 1
6. Public Key = (enc,N) Secrete Key = (dec,N)

#  METHODOLOGY

 In Pollard Rho algorithm it checks the factoring time of ‘N’ on the basis of its length in Python. Here we consider 2, 3, 5, 10, 15 and 25 length size of ‘N’. This algorithm gives the factor of ‘N’ that means the value of ‘prime1’ and ‘prime2’ which is provid-ed as an input to RSA algorithm. So, the time required for ‘N’ length 2, 5, 10, 15 fac-tored by pollard rho in a specific time but when we provide ‘N’ length of 25 then system does not respond properly. As we are not able to directly access a quantum computer so, IBM gives us an alternative that is “IBM Quantum Lab” an online tool which connects to an actual quantum computer is designed and developed by IBM it make a use of “Jupyter notebooks” with Python. It is publically available for everyone through internet anyone can use it with the help of their PCs running any operating system. We run the code in Python and check the execution time to get factors of a number.

#  LITERATURE REVIEW

 In early cryptography, integer factorization and finite-field discrete logarithms were challenging, but crucial for reliable public key encryption. While other mathe-matical challenges have emerged, basics like RSA and Diffie-Hellman, using these methods, remain popular. Factorization rigidity is seen as a well-documented disad-vantage. Intelligent factoring algorithms were developed early on to benefit various contexts like numeric fields or integers. R. L. presented a new encryption technique. Rivest, Shamir, L. Adleman and col-leagues introduced the 'asymmetric cryptosystem' in 1978, inspired by Diffie and Hellman's theories. Its reliability hinges on the difficulty of factoring large numbers. [28] Improved RSA was proposed in 2015 by Israt Jahan, Mohammad Asif, Liton Jude Rozario, et al. Two separate public keys are used in the Improved RSA algorithm, making it resistant to brute force attacks by making it difficult to find the number's factors. Useful for high security, not for speed.[29] A. L. Manasse studied factoring options in 1988 and provided a helpful answer to the question of maximum integer computation in a month.[31] Silverman proved that a large minicomputer can factor 60-digit numbers in a day in his 1987 paper on the MQP sieve. This method is ideal for parallel implementation due to its characteristics. Factoring RSA-129 was the largest MPQS calculation using inputs from many enthusiasts. Its formula was developed by the RSA inventor to solve a problem in a column by Martin Gardner in the August 1977 issue of Scientific American. RSA-129 issue was the main problem in factoring issues. [32][33][34][35] The RSA-129 study notes challenges in achieving desired outcomes, such as fac-toring 512-bit RSA modules, due to its widespread use. MPQS suggests the objective of immense power is almost attainable, but modern digital field sieves are better suit-ed for the task. Pollard created the Intelligent Digital Field Screen in 1988 for Fermat factorization. He used one-dimensional numbers. Math has arrived. Pollard's tech-nique fascinates many. Using 2128 + 1 to support his approach raises doubts about its usefulness for other numbers. Lenstra et al. discovered the 9th Fermat number in 1989/1990, using an expanded technique based on Pollard's work. It was a major breakthrough in mathematics. Obstacles must be overcome to establish standard measurements, particularly in addressing the matrix phase and the Z-factor ring, which requires higher resolution than Z/2Z. Teamwork helped ease restrictions. Bernstein, Buhler, Couveignes, and Pomerance also contributed to the General Number Field Sieve method for factoring large numbers in 1993. These efforts were detailed in "The Evolution of Number Field Sieve" book [35]. Although still in development, GNFS achieved success in factoring RSA-129. In 1996, the RSA-130 was improved using factor analysis GNFS. The RSA challenge list established this excellent benchmark in 1991. Using GNFS, RSA-130 and other difficult records are calculated sequentially from the RSA list. Record-breaking discrete logarithms were achieved in 2019 for a common prime field of 240 decimal digits and 795 bits, while in 2016 the record for somewhat bigger special-form primes was 308 or 1024 bits. Initial attempts at factoring larger integers (768 bits in 2009 and 1024 bits in 2007) suggest that discrete prime logarithms may be slightly more challenging. However, recent surveys indicate that the difference in difficulty between the two methods is not as significant as believed.[36][37] Certicom ECC Task was created in 1997 to address RSA Problems, including toy problems for different bit curves and various levels of challenges up to 359-bit curves. Problems on finite domains of characteristics two and large main fields were resolved by the end of 1999. Assaults use the van Oorschot and Wiener approach, an extension of Pollard's concepts. Clients disperse via the web to idle devices, communicating with control centres via email or sockets. In 2004, 109-bit difficulties were overcome. No achievement has been reported for the 131-bit problem.[39][40][43][45] If a quantum computer is invented, the previous complex examinations of factori-zation and discrete logarithmic problems will become outdated. In 1994, Shor devel-oped a polynomial time technique that can solve these problems in any finite abelian group. Shor's algorithm needs a quantum computer to work.[4][42] Quantum backup recordings are not as good as traditional computers. In 2001, IBM scientists used a 7-qubit device to calculate the number 15. The new record in 2019 is 4088459, achieved on a 5-qubit CPU. This highlights the potential risks of quantum computers and the emergence of post-quantum technology.

## **Example**

Here, we demonstrate the RSA with the length of Prime1 is 309 digits and length of Prime2 is 154 digits so, the length of N is 308 digits.

|  |  |
| --- | --- |
| Prime1 | 129461334469340170055690030156079553304974769919163799672950780252188707688784051661448960830823803164458088322707530461858388331815875657012811563766386476993109263719109419159296573615014488383364399298059559524141772431837606805196513481543623101513935466576250130549265767937006062063084411574085068041131 |
| Prime2 | 6993109263719109419159296573615014488383364399298059559524141772431837606805196513481543623101513935466576250130549265767937006062063084411574085068041131 |
|  | N= prime1 \* prime2 |
| N | 90533725737098079774827580573993097425674967845784509520071268849503259888628062855846074305151939838085957660677656830522128211054627110478887226695589031380770295464396753527964083426943482897815423891158797187251574768198415613732519287271057833916852093725679680371528831169063595716292543517350520189757 |
|  | Compute phi(N)=(prime1-1) \* (prime2-1) |
| Phi (N) | 90533725737098079774827580573993097425674967845784509520071268849503259888628062855846074305151939838085957660677656830522128211054627110478887226695589011441527584811270328799664494203973664016974032676719270368031777117490039930130839660831351650022600181340597279069216877393224352065642430662109075509980 |
|  | Pick integer enc like 1<enc < phi(N) and enc and N are coprime. |
| Message(M) | 100 |
| enc | 35871371095806322919319387841960725176014304318195508831335991309514301169557069430665779519893133567290697914690004301414057469568297922226501715652479244142486775614698183554356951770817836724112188764744355510626047573710882389538641169234807920390211157124261951742447136823478039190526967270280623822659 |
| dec | 18151376413348997749626376125820267647323553455215887023934038406813691655568165957351561145766148932446083042142863395583516258355664687256904574859125555058267020390826081305573077391626987639863528630509782269840410605953706184060667178463961957409331582262099864669556403740948832767524513701683119915479 |
| Public Key is (enc, N) | (35871371095806322919319387841960725176014304318195508831335991309514301169557069430665779519893133567290697914690004301414057469568297922226501715652479244142486775614698183554356951770817836724112188764744355510626047573710882389538641169234807920390211157124261951742447136823478039190526967270280623822659, 90533725737098079774827580573993097425674967845784509520071268849503259888628062855846074305151939838085957660677656830522128211054627110478887226695589031380770295464396753527964083426943482897815423891158797187251574768198415613732519287271057833916852093725679680371528831169063595716292543517350520189757) |
| Private Key is (dec, N) | (18151376413348997749626376125820267647323553455215887023934038406813691655568165957351561145766148932446083042142863395583516258355664687256904574859125555058267020390826081305573077391626987639863528630509782269840410605953706184060667178463961957409331582262099864669556403740948832767524513701683119915479, 90533725737098079774827580573993097425674967845784509520071268849503259888628062855846074305151939838085957660677656830522128211054627110478887226695589031380770295464396753527964083426943482897815423891158797187251574768198415613732519287271057833916852093725679680371528831169063595716292543517350520189757) |

# TECHNIQUES TO BREAK RSA

There are various different techniques to break RSA like

### **Table 1: Techniques to break RSA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Technique** | **Observations** | **Reference** |
| 1 | Small factors | Efficient When N has Small Factors. | [44] |
| 2 | Fermat factorization | Fermat's approach is efficient even though a and b are quite huge whenever a and b are able to choose adjacent to one another(a/b=1). | [11] |
| 3 | Batch GCD | It is Useful for cloned system, Virtual Machines and Embeded devices with low entropy. | [6][12] |
| 4 | Elliptic Curve Method (ECM) | effective even if  the number is very large  and multipliers are extremely small | [21][22][23] |
| 5 | Weak entropy | Several embedded systems, such as network equipment, gateways, smart television sets, and Internet gadgets, are extremely low entropy sources. | [5] |
| 6 | Fault injection | Whenever something erroneous happens during any of such computing power, we may lose the encryption key at that point. | [8][9] |
| 7 | Small private exponent | An intruder may occasionally breach the key if he predicts or retrieves a part of information from pvalue,qvalue,e, or d. | [13][14] |
| 8 | Known Partial bits | An intruder may occasionally breach the encryption key if he correctly predicts or retrieves some part of information from pvalue, qvalue, e, or d. | [10][15] |
| 9 | p/q near a small fraction | If the approximate result is sufficiently accurate, the MSB of value is right, so we may break number. | [16] |
| 10 | Weakness in signature | Inadequate or incorrect buffering prior to encryption/signature. Using two distinct keys to protect the identical information | [14] |
| 11 | Side Channel attacks | By observing power usage, Emations, or any other potentially altering A factor, the calculation can expose data including the secret key. | [6] |
| 12 | No. Field Sieve(NFS) | Quite complicated, but also quite parallel | [17] |

In this paper we use Pollard Rho and Shor Quantum Algorithm for breaking the RSA in detail.

# POLLARD RHO ALGORITHM STEPS

1. Pick any random value of x and c
2. Pick y = x and f(x) = x2-c
3. While divisor is not obtained

i) Update x to f(x) % n

ii) Update y to f(y) % n

iii) Calculate GCD of x – y and n

1. If GCD ≠1

i) If GCD is n, repeat from step2 with other set of x, y and c

ii) Else GCD is our answer.

A. **Example**

N = 437; a = b = 4; k=1

F(a) = a2 + 1(mod 437)

 = 42 + 1(mod 437)

 = 17 mod 437

 = 17

F(a) = a2 + 1(mod 437)

 = (17)2 + 1(mod 437)

 = 290 mod 437

 = 290

F(a) = a2 + 1(mod 437)

 = (290)2 + 1(mod 437)

 = 841 mod 437

 = 404

Now, Calculate GCD i.e. D

D = gcd (| 290 – 404 |, 437)

 = gcd (114, 437)

 = 19

One factor i.e prime1 = 19 now we find second factor i.e prime2 = 437/19 = 23

Therefore, 437 = 19 \* 23 Pollard Rho had its great success in factoring large numbers also. Like, N = 107753506375013 is also factored using the pollard rho and we got the factors 8686087 \* 12405299.

# QUANTUM COMPUTER AND SHOR’S ALGORITHM

RSA, Pollard Rho, and Lenstra's Elliptic Curve Algorithms were developed for digital computers, with RSA currently being a secure cryptographic method. 2048-bit RSA cannot be broken with current techniques, but a quantum machine could make it vulnerable. The quantum computer uses quantum events and excels in data processing time. In digital and quantum computers, the bit and qubit, respectively, are the smallest units of data processing. A 2-qubit quantum machine can split under four conditions at once. Quantum computers process data faster than digital computers. In 1994, Peter Shor proposed a quantum algorithm for integer factorization. With a quantum computer, Shor's method can find the prime factors of a number in polynomial time. Shor's factoring method has two parts: Section-1 on a digital computer. Assuming the order-finding is resolved, it returns to the two prime factors of a number. Section-2 explains how the quantum computers algorithm from section-1 uses Quantum Fourier Transform for period finding. [7] To break RSA with Shor's algorithm, many quantum gates are required. For example, the 2048-bit RSA algorithm can be secured by using around 10000 qubits 2.23T gates, 1.8T circuit depth. [3][7][46]

# SHOR’S ALGORITHM STEPS

# 1. Pick NUM as a value that will be factorised. If NUM is even, prime, or prime power, quit.

# 2. Pick a value ranging from 1 and NUM at random. Make a 'k' out of this value.

# 3. Locate GCD (NUM,k). You may use Euclid's division Algorithm to compute it. If GCD = 1, then GCD is a factor of NUM, so we are finished.

# 4. If, on the other hand, GCD = 1, therefore go to step four.

# 5. We must determine the lowest positive integer remainder such that if

# 6. F(X) = K\*X % NUM, therefore F(A) = F(A + remainder).

# 7. To discover the remainder, perform the instructions below.

# i) Create a new variable, Q = 1.

# ii) Determine (Q \* K) % NUM.

# 8. If the remaining is one, go to step (iii). If not, set the value of 'Q' to whatever remainder we obtained. Continue this step until the residual equals one while keeping note of how many times you performed the change. Always remember to alter the value of 'Q'.

#  iii) The number of modifications you performed in step (ii) is your 'remainder' value.

# 9. If the value of 'remainder' is odd, return to step 2 and select another value instead of 'K'.

#  10. The NUM factors are factor 1 = GCD (P+1, NUM) and factor 2 = GCD (P-1, NUM).

# A. Example

 NUM = 357

 Choose K = 205 at random.

 gcd (357, 205) = 1

 (1 x 205) % 357 = 205-----------(1)

 (205 x 205) % 357 = 256--------(2)

 (256 x 205) % 357 = 1------------(3)

 Transformation = r = 3

 r = 3 is an odd number, the algorithm chooses an alternate number of K at random.

 K = 152

 gcd(152, 357) = 1

 (1 x 152) % 357 = 152----------(1)

 (152 x 152) % 357 = 256-------(2)

 (256 x 152) % 357 = 356-------(3)

 (356 x 152) % 357 = 205-------(4)

 (205 x 152) % 357 = 101-------(5)

 (101 x 152) % 357 = 1----------(6)

 transformation = r = 6 is even

 remainder = q = in (6/2) that is 3rd transformation.

 q = 356

 if q + 1 = N

 356 + 1 = 357

 Again select different k value which is any random number k = 52

 gcd(357,52) = 1

 (1 x 52) % 357 = 52-----------(1)

 (52 x 52) % 357 = 205--------(2)

 (205 x 52) % 357 = 307-------(3)

 (307 x 52) % 357 = 256-------(4)

 (256 x 52) % 357 = 103-------(5)

 (103 x 52) % 357 = 1----------(6)

 transformation = r = 6 is even

 remainder = q = in (6/2) that is 3rd transformation.

 P = 307

 If P+ 1 = 1

 307 + 1 ≠ 357

 finally,

 First Factor is the greatest common divisor of 308 and 357 is 7

 Second Factor is the greatest common divisor of 306 and 357 is 51

 Factors of 357 are 7 and 51.

# EXPERIMENT AND RESULTS

## **A.Experimental Setup**

The computing device used throughout the study includes 8.00 GB DDR4 RAM, an Intel® Core(TM) i5-8265u CPU, and an operating frequency of 1.60GHz-1.80GHz.

**B. Experiment Result**

### **Table 2: Pollard Rho Algorithm Experiment Result**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Length of the key** | **Number** | **Time(Sec)** |
| 1 | 2 | 15 | 0.316643476 |
| 2 | 2 | 21 | 0.605066061 |
| 3 | 2 | 35 | 0.866878271 |
| 4 | 2 | 55 | 1.198400259 |

### **Table 3: Shor’s Algorithm Experiment Result in python**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Length of the key** | **Number** | **Time(Sec)** |
| **1** | 2 | 15 | 3.4942984580993652 |
| **2** | 2 | 21 | 35.55985164642334 |
| **4** | 2 | 35 | 671.4257118701935 |
| **5** | 2 | 55 | Not Respond |

### **Table 4: Shor’s Algorithm Experiment Result in IBM Quantum Lab**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Length of the key** | **Number** | **Time(Sec)** |
| **1** | 2 | 15 | 32.297845125198364 |
| **2** | 2 | 21 |  59. 23662328720093 |
| **3** | 2 | 35 | 431.0061309337616 |
| **4** | 2 | 55 | 455.6278467178345 |

### **Table 5: Factoring time: Pollard rho Algorithm with increasing number size**

|  |  |  |
| --- | --- | --- |
| **Case** | **Length of the key** | **Time(Sec)** |
| 1 | 3 | 0.07776474952697754 |
| 2 | 5 | 0.17345857620239258 |
| 3 | 10 | 17.693896532058716 |
| 4 | 15 | 41.51395344734192 |

**C. Result and Interpretation**

In this experiment we found that the Pollard rho algorithm gives better results than the Shor’s quantum algorithm in both simple python 3.9.5 and IBM Quantum Lab.



**Figure 1: Factoring Time: Pollard rho Vs Shor’s Algorithm**



**Figure 2: Factoring Time: Pollard rho Algorithm with Increasing Number Size**

# CONCLUSION

As per the above comparison tables and graph, it is observed that the classical cryptographic algorithm pollard rho and Shor’s algorithm both break the RSA classical integer factorization algorithm. On Personal Computer the Pollard rho break RSA faster than Shors algorithm for small integers. Even if hacker cannot break RSA using pollard rho on classical computer for large integers. Shors algorithm on quantum simulator break the RSA for larger numbers than shors algorithm in python 3.9.5. But better factoring time of shors algorithm is obtained using python on personal computer than quantum computer simulator.

# LIMITATIONS AND FUTURE SCOPE

There are some limitations with quantum computation. The cost of a quantum com-puter is very high. Because of that, it is not accessible for open use. The 10-qubits quantum computer works very well. Thereafter, growing qubits such as 70 qubits, the accuracy is not right.[47] Shor's quantum algorithm is probabilistic in nature. With a large number of qubits, when a powerful enough quantum computer is available at that time, we can make best use of Shor's algorithm. Google has a 72-qubit quantum computer but it is not available publicly.[1]

The results obtained from our work open several directions for further research. In the future, we will analyze Shor’s quantum algorithm for bigger numbers on the availability of a sufficiently large number Qubits in Quantum computers.

# REFERENCES

[1] Shaheed Nehal A.,Mubasheer Farhan, Sunad Y. S.,” Quantum Cryptography-Breaking RSA Encryption using Quantum Computing with Shor’s Algorithm”,International Journal For Technological Research in Engineering, Vol. 8, Issue 1, Sep. 2020, ISSN: 2347-4718.

[2] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P.Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt,” Realization of a scalable Shor algorithm”, Science, Vol 351,Issue 6277, pp 1068-1070, 31 Jul 2015, arXiv:1507.08852v1 [quant-ph].

[3] C.H. Ugwuishiwu, U. E. Orji, C. I. Ugwu, C. N. Asogwa,”An Overview of Quantum Cryptography and Shor’s Algorithm”, International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, No.5, Sep-Oct 2020, ISSN 2278-3091.

[4] P.W. Shor, “Algorithm for quantum computation: discrete logarithm and factoring” Proc. 35th IEEE Annual Symp. on Foundations of Computer Science, Santa Fe, NM, pp 24-134, Nov. 1994,

[5] Behrouz Zolfaghari , Khodakhast Bibak , and Takeshi Koshiba ,” The Odyssey of Entropy: Cryptography”, Entropy, Vol. 24, Issue 2, Feb 2022. https://doi.org/10.3390/e24020266

[6] https://courses.csail.mit.edu/6.857/2017/project/11.pdf

[7] Kritsanapong Somsuk,” The Improvement of Elliptic Curve Factorization Method to Recover RSA’s Prime Factors”, Symmetry, Vol 13, Issue 8, July 2021. https://doi.org/10.3390/sym13081314 https://www.mdpi.com/journal/symmetry.

[8] https://rp.os3.nl/2017-2018/p84/report.pdf

[9] Pablo Rauzy — Sylvain Guilley,” A Formal Proof of Countermeasures Against Fault Injection Attacks on CRT-RSA”, Journal of Cryptographic Engineering , Jan 2014. DOI: 10.1007/s13389-013-0065-3 • Source: arXiv [ cs.CR].

[10] Santanu Sarkar, Subhamoy Maitra,” Partial key exposure attacks on rsa and its variant by guessing a few bits of one of the prime factors”, Bulletin of the Korean Mathematical Society, Vol. 46 Issue 4, pp 721-741, July 2009.

[11] Hazem M. Bahig, Mohammed A. Mahdi, Khaled A. Alutaibi, Amer AlGhadhban, Hatem M. Bahig,” Performance Analysis of Fermat Factorization Algorithms”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, Issue. 12, 2020.

[12] Shehanaz, Raju K,” Study of Various Algorithms for Computing GCD of Bulk Pair of Numbers and Parallelization Techniques”, International Journal of Advanced Research in IT and Engineering, Special Issue - NCECP16@NMAMIT, NITTE, 2016, ISSN: 2278-6244.

[13] Hatem M. Bahig , Dieaa I. Nassr , Mohammed A. Mahdi and Hazem M. Bahig ,” Small Private Exponent Attacks on RSA Using Continued Fractions and Multicore Systems”, Symmetry 2022. https://doi.org/10.3390/sym14091897

[14] Dragan Savić, Petar Milić, Borislav Mažinjanin, Petar Spalević,” Cryptanalytic attacks on RSA algorithm and its variants”, Przegląd Elektrotechniczny, 2022, ISSN 0033-2097.

[15] Amir Hamzah Abd Ghafar, Muhammad Rezal Kamel Ariffin, Sharifah Md Yasin and Siti Hasana Sapar,” Partial Key Attack Given MSBs of CRT-RSA Private Keys”, Mathematics,Vol. 8, 2020, doi:10.3390/math8122188

[16] Muhammad Rezal Kamel Ariffin, Saidu Isah Abubakar, Faridah Yunos, Muhammad Asyraf Asbullah ,” New Cryptanalytic Attack on RSA Modulus N = pq Using Small Prime Difference Method”, Cryptography, Vol. 3, Issue 2, Mar 2019. doi:10.3390/cryptography3010002

[17] Steven Byrnes,” The Number Field Sieve”, Math 129, May 18, 2005. math129-finalpaper.pdf (sjbyrnes.com)

[18] Hengki Tamando Sihotang, Syahril Efendi, Elvyawati M Zamzami, Herman Mawengkang,” Design and Implementation of Rivest Shamir Adleman’s (RSA) Cryptography Algorithm in Text File Data Security”, ICAISD 2020, doi:10.1088/1742-6596/1641/1/012042

[19] Israt Jahan, Mohammad Asif, Liton Jude Rozario,” Improved RSA cryptosystem based on the study of number theory and public key cryptosystems”, American Journal of Engineering Research (AJER), Volume-4, Issue-1, pp-143-149, 2015, e-ISSN : 2320-0847 p-ISSN : 2320-0936.

[20] Yoshinori Aono, Sitong Liu, Tomoki Tanaka, Shumpei uno, Rodney Van Meter, Naoyuki Shinohara, Ryo Nojima.”The Present and Future of Discrete Logarithm Problems on Noisy Quantum Computers”, IEEE Transactions on Quantum Engineering, Vol. 3, pp. 1-21, 2022, ISSN: 2689-1808. doi:10.1109/TQE.2022.3183385

[21] Karishma Bhirud, Dipashree Kulkarni, Renuka Pawar, Prachi Patil,” Data Security Using Elliptic Curve Cryptography”, International Journal of Computer Engineering In Research Trends, Vol. 3, Issue 5, pp. 222-225, May 2016.

[22] Jiaxu Bao,”Research on the Security of Elliptic Curve Cryptography”, Advances in Economics, Business and Management Research, Vol. 652, 7th International Conference on Social Sciences and Economic Development(ICSSED 2022).

[23] Sharad Kumar Verma, Dr. D. B. Ojha,”A Discussion on Elliptic Curve Cryptography and it’s Applications”, International Journal of Computer Science Issues, Vol. 9, Issue 1, Jan 2022.

[24] Nagaratna Hegde, P.Deepthi,” Pollard RHO Algorithm for Integer Factorization and Discrete Logarithm Problem”, International Journal of Computer Applications, Vol. 121, Issue 18, Jul 2015, ISSN: 0975 – 8887.

[25] Anjan K Koundinya, Harish G,Srinath N K, Raghavendra G E, Pramod Y V, Sandeep R and Punith Kumar G,” Performance Analysis Of Parallel Pollard’s Rho Factoring Algorithm”, International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, Issue 2, Apr 2013.

[26] Eric Bach, Bryce Sandlund, “Baby-step gaint-step Algorithms for the Symmetric Group”, Journal of Symbolic Computation, Vol 85, pp 55-71, Mar-Apr 2018.

[27] A. Uma Maheswari, Prabha Durairaj, “Modified Baby Step Giant Step Algorithm To Solve Elliptic Curve Discrete Logarithm Problem (Ecdlp)”, International Research Journal of Engineering and Technology (IRJET), Vol. 05 Issue 04, Apr-2018.

[28] Rivest, Ronald L., Shamir, Adi, and Adleman, Leonard M., “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Communications of the Association for Computing Machinery, Vol. 21 Issue 2, pp 120–126, 1978.

[29] Israt Jahan Mohammad Asif, Liton Jude Rozario,” Improved RSA Cryptosystem based on the Study of Number Theory and Public key Cryptosystems”, American Journal of Engineering Research(AJER), Vol. 4, Issue 1, pp 143-149, 2015.

[30] Kalpesh S. Prajapati, Prof. B. V. Buddhdev,” Overview of Improvements and Modifications in RSA Algorithm”, Journal of Emerging Technologies and Innovative Research,Vol. 1, Issue 7, Dec 2014, ISSN 2349-5162.

[31] Lenstra, Arjen K., and Manasse, Mark S., “Factoring by Electronic Mail”, EUROCRYPT’89 LNCS, Vol. 434, Pages 355–371, 1990.

[32] Eric Landquist,” The Quadratic Sieve Factoring Algorithm”, MATH 488: Cryptographic Algorithms, Dec 14, 2001.

[33] Robert D. Silverman,”The multiple polynomial quadratic sieve”, Mathematics of Computation, Vol. 48, pp 329–339, Jan 1987.

[34] Atkins, Derek, Graff, Michael, Lenstra, Arjen K., and Leyland, Paul C., “The Magic Words are Squeamish Ossifrage”, ASIACRYPT’94. LNCS ,Vol.917, Pages 263–277, 1995.

[35] Lenstra, Arjen K., and Lenstra Jr., Hendrik W. , “The development of the number field sieve”, Springer,Lecture Notes in Math., vol. 1554., 1993.

[36] Boudot, Fabrice, Gaudry, Pierrick, Guillevic, Aurore, Heninger, Nadia, Thomé, Emmanuel, and Zimmermann, Paul.,” Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment”, Cryptology,2020. https://eprint.iacr.org/2020/697

[37] Fried, Joshua, Gaudry, Pierrick, Heninger, Nadia, and Thomé, Emmanuel. 2017. ,“A Kilobit Hidden SNFS Discrete Logarithm Computation”, EUROCRYPT, 2017, Part I. LNCS, Vol. 10210, Pages 202–231, 2017.

[38] Aoki, Kazumaro, Franke, Jens, Kleinjung, Thorsten, Lenstra, Arjen K., and Osvik, Dag Arne., “A Kilobit Special Number Field Sieve Factorization”, ASIACRYPT 2007. LNCS, Vol. 4833, Pages 1– 12,2007.

[39] Pollard, John M.,” Monte Carlo algorithms for index computation (mod p)”, Math. Comp., 32, 918–924, 1978.

[40] Pollard, John. M. ,”Kangaroos, Monopoly and Discrete Logarithms”, Journal of Cryptology, Vol.13, 437–447, 2000.

[41] Vandersypen, Leiven M.K., Steffen, Matthias, Breyta, Gregory, Yannoni, Costantino S., Sherwood, Mark H., and Chuang, Isaac L.,”Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance”, Nature, Vol. 414, pp 883–887, 2001, doi: https://doi.org/10.1038/414883a

[42] Dash, Avinash, Sarmah, Deepankar, Behera, Bikash K., and Panigrahi, Prasanta K.,”Exact search algorithm to factorize large biprimes and a triprime on IBM quantum computer”, arXiv:1805.10478v2 [quant-ph], 2018.

[43] Paul C. van Oorschot and Michael J. Wiener,” Parallel Collision Search with Cryptanalytic Applications”, Crypto ’96, Sep 1996.

[44] Abderrahmane Nitaj, Tajjeeddine Rachidi,” Factoring RSA moduli with weak prime factors”,International Conference on Codes, Cryptology, and Information Security C2SI 2015: Codes, Cryptology, and Information Security, pp 361–374, 2015.

[45] https://www.certicom.com/content/dam/certicom/images/pdfs/cert\_ecc\_challenge.pdf

[46]<https://www.itnews.com.au/news/quantum-computers-wont-break-rsa-encryption-any-time-soon> 590115#:~:text=They%20found%20that%20to%20factor,104%20days%20to%20crack%20RSA.

[47] https://www.itrelease.com/2020/10/advantages-and-disadvantages-of-quantum-computers/