A Cryptanalytic Approach to Breaking RSA Encryption Using Public Key, with Proposed Improvement Leveraging Prime Number Distribution Patterns

Abubakar T. U.[1], Ibrahim A. A.[2], Garba A. I.[2], Abubakar S. F.[3], James T. O.[3], Sarki M. N.[3], Mua'azu S. B.[3], Shehu S.[4], and Muhammad A. H.[5]

[1]Department of Mathematics, ShehuShagari College of Education, Sokoto., Sokoto.

[2]Department of Mathematics, UsmanuDanfodio University Sokoto.

[3]Department of Mathematics, Abdullahi Fodio University of Science and Technology, Aliero.

[4]Department of Mathematics, Sokoto State University.

[5]Department of Science, Mathematics Unit, State Collage of Basic & Remedial Studies

DOI: https://dx.doi.org/10.51244/IJRSI.2025.1210000338

Received: 10 November 2025; Accepted: 16 November 2025; Published: 22 November 2025

ABSTRACT

RSA algorithm, a widely used public-key cryptosystem, relies on the difficulty of factoring large composite numbers into their prime factors. However, advancements in computational power and factorization techniques have introduced potential threats to its security. As RSA remains a cornerstone of modern cryptography, the need for improved in its security measures is paramount in the face of evolving computational challenges. This study presents a cryptanalytic examination of the RSA encryption scheme and proposes an improvement that leverages prime number distribution patterns to strengthen data security. Several mathematical methods which involve RSA key generation and its encryption/decryption process, ASCII Table, mapping as well as the Sieve of Eratosthenes were applied in the study. The research analyses how RSA public parameters and encoding methods can reveal structural weaknesses when subjected to mathematical scrutiny. To address these vulnerabilities, a modified scheme is introduced, in which plaintext characters are mapped using prime distribution patterns. This substitution increases ciphertexts randomness and minimizes predictable patterns between plaintexts and ciphertexts. Experimental evaluation demonstrates that the proposed improvement enhances resistance to analytical attacks while maintaining RSA operational compatibility. The study contributes to the on-going development of more secure and efficient public-key cryptographic systems.

Keywords: RSA, Encryption, Decryption, Cryptanalysis, Mapping and Prime numbers.

INTRODUCTION

The advent of public-key cryptography revolutionized data security by introducing a mechanism for secure communication over untrusted networks. Among various asymmetric cryptographic algorithms, the **Rivest–Shamir–Adleman (RSA)** encryption scheme remains one of the most widely adopted due to its mathematical simplicity and proven resistance to brute-force attacks. Since its introduction in 1977 by Rivest, Shamir, and Adleman, RSA has been the cornerstone of digital security in applications such as secure email, digital signatures, and key exchange protocols (Rivest, **Shamir and Adleman**, 1978). Its security fundamentally relies on the computational difficulty of **factoring a large composite integer** $N = pq$, where p and q are large prime numbers. The **public key** in RSA, typically denoted as (N, e), is openly distributed, while the private key d remains secret. Theoretically, knowledge of only the public key should not compromise the private key, (Nitaj, 2016). Nonetheless, recent studies have shown that under specific conditions such as weak key

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

generation, poor randomness in prime selection, or low encryption exponents an attacker can exploit mathematical patterns to deduce partial information about the private key or factor N, (Ariffin & Shehu, 2016; Wiener, 1990). Cryptanalysis of RSA using the encryption exponent involves exploiting possible vulnerabilities when the encryption key e, modulus N, and encoding scheme are public.

This study, therefore, focuses on exploring **cryptanalytic techniques** that utilize only the **public key** to test the boundaries of RSA security. It aims to investigate a cryptanalytic approach to breaking RSA encryption using public key, with proposed improvement leveraging prime number distribution patterns.

The findings are expected to contribute to both **cryptanalysis** and **cryptographic engineering**, providing a clearer understanding of how RSA can be both broken and improved.

LITERATURE REVIEW

Overview of the RSA Algorithm and Its Mathematical Foundation

The **Rivest, Shamir, Adleman (RSA)** algorithm, introduced in 1977, remains one of the earliest and most enduring forms of public key cryptography, (Nadia, Syahril & Sawaluddin, 2020). Its theoretical foundation lies in **modular arithmetic** and the difficulty of factoring a large composite number $N = pq$, where p and q are prime numbers (Rivest, Shamir & Adleman, 1978). RSA operates on the principle of one way functions: while it is computationally straightforward to multiply large primes, reversing the process to retrieve the original factors is considered infeasible with classical computing methods. The security of RSA thus depends on the hardness of the **Integer Factorization Problem** (IFP).

Classical Cryptanalytic Techniques against RSA

Early research into RSA vulnerabilities focused primarily on **factoring algorithms**. The **Fermat factorization method** is one of the oldest techniques, effective when the two primes p and q are close in magnitude (Aminudina & Cahyonoa, 2021). Subsequently, **Pollard's Rho algorithm** improved upon Fermat's approach by using pseudo-random number generation to detect nontrivial factors more efficiently.

Kefa (2006), in a review of integer factorization methods, discussed several algorithms including the Elliptic Curve Algorithm (ECM), Quadratic Sieve (QS), and Number Field Sieve (NFS). Hinek (2009) demonstrated that it is possible to factor the modulus N_i if $d < N^{\delta}$ where $\delta = \frac{k}{2(k+1)} - \varepsilon$, ε is a small constant.

Somsuk and Kasemvilas (2013) introduced the Possible Prime Modified Fermat Factorization (PPMFF) algorithm, which improves the handling of both trivial and non-trivial N values.

Overmars and Venkatraman, (2020), in their paper applied mathematical tools and algorithms designed for factorizing semi-prime numbers using the sum of squares method incorporating mathematical theorems and principles related to prime factorization and semi-prime numbers to support the development and validation of the new method.

Aminudin and Cahyono, (2021), provide a practical analysis of Fermat's method alongside Pollard's rho method, reinforcing the notion that these classical techniques remain vital in the landscape of integer factorization.

Overmars and Venkatraman, (2021), showed that if a Pythagorean quadruple is known and one of its squares represents a Pythagorean triple, then the semi-prime is factorized. They proved that to factor a semi-prime, it is sufficient that only one of these Pythagorean quadruples be known.

Kannan and Mohana, (2023), they delve into the challenges associated with factorization in the context of the RSA cryptosystem. They also discussed the various programming tools utilized to handle the challenge of

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

factoring large numbers and penetrated into the enhancement of the Simple Fermat Primality Test as a means to expedite the primality testing algorithm.

Despite these advances, even GNFS requires exponential time for sufficiently large RSA moduli (greater than 2048 bits), maintaining RSA practical security under classical computation.

Quantum Threats and Post-Quantum Cryptanalysis

A major turning point in cryptographic research occurred with **Shor's algorithm**, which demonstrated that a quantum computer could factor large integers in **polynomial time**, effectively breaking RSA's foundational assumption, (shor, 1994). Although large-scale quantum computers capable of executing Shor's algorithm on RSA-2048 remain theoretical, this revelation has prompted the development of **post-quantum cryptography (PQC)**.

Researchers are now exploring cryptographic schemes based on **lattice problems, multivariate polynomials,** and **hash-based systems**, which are believed to resist quantum attacks. However, before transitioning away from RSA entirely, it remains crucial to understand its **residual vulnerabilities** under classical computation, particularly through mathematical cryptanalysis using **public key inference**.

Prime Number Distribution and Cryptographic Implications

Prime numbers form the backbone of RSA encryption. Their distribution, however, is neither entirely random nor uniform. The **Prime Number Theorem (PNT)** approximates the number of primes less than a given integer x as $\frac{x}{\ln x}$, indicating that primes become sparser as numbers increase (Hardy & Wright, 2008). Despite their apparent irregularity, primes exhibit deep mathematical structures that can be modelled through probabilistic and analytical techniques.

Recent studies (Ameha, 2021) have explored formulas for predicting prime occurrences, revealing patterns in **prime gaps**, **residue sequences**, and **modular symmetries**. Such findings have two major implications:

1. Cryptanalysts may use these patterns to **narrow down prime search spaces**, thus improving factorization attempts.
2. Cryptographers may exploit these same distributions to **generate primes more securely**, by introducing pseudo-random distortions that obscure detectable regularities.

Drawing from these insights, the present research proposes a **cryptanalytic approach** that leverages the **distribution patterns of prime numbers** to infer potential weaknesses in RSA when the public key (N, e) is available. The paper introduces a new **cryptographic improvement** embedding **prime-distribution-based substitution of ASCII characters prior to encryption**. This modification aims to enhance security by obfuscating plaintext patterns before they enter the RSA process, effectively creating a layered defence that combines **number-theoretic complexity** with **symbolic transformation**. In essence, this study extends the scope of RSA research beyond normal factorization-based cryptanalysis to a **pattern-oriented and distribution-aware approach**, offering a new lens through which RSA mathematical foundation can be both examined and strengthened.

METHODOLOGY

This study employed several mathematical methods which involve RSA key generation and its encryption/decryption process, ASCII Table, mapping as well as the Sieve of Eratosthenes.

ASCII Table

ASCII (American Standard Code for Information Interchange) is a character encoding standard that assigns numeric values to characters and symbols. The complete 7-bit ASCII table and corresponding decimal

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

equivalents are presented in the table below:

Table 1: ASCII Table

Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	26	SUB	52	4	78	N	104	H
1	SOH	27	ESC	53	5	79	O	105	I
2	STX	28	FS	54	6	80	P	106	J
3	ETX	29	GS	55	7	81	Q	107	K
4	EOT	30	RS	56	8	82	R	108	L
5	ENQ	31	US	57	9	83	S	109	M
6	ACK	32		58	:	84	T	110	N
7	BEL	33	!	59	;	85	U	111	O
8	BS	34	"	60	<	86	V	112	P
9	HT	35	#	61	=	87	W	113	Q
10	LF	36	$	62	>	88	X	114	R
11	VT	37	%	63	?	89	Y	115	S
12	FF	38	&	64	@	90	Z	116	T
13	CR	39	'	65	A	91	[	117	U
14	SO	40	(	66	B	92	\	118	V
15	SI	41	)	67	C	93	]	119	W
16	DLE	42	*	68	D	94	^	120	X
17	DC1	43	+	69	E	95	_	121	Y
18	DC2	44	,	70	F	96	`	122	Z
19	DC3	45	-	71	G	97	a	123	{
20	DC4	46	.	72	H	98	b	124	|
21	NAK	47	/	73	I	99	c	125	}
22	SYN	48	0	74	J	100	d	126	~
23	ETB	49	1	75	K	101	e	127	DEL
24	CAN	50	2	76	L	102	f		
25	EM	51	3	77	M	103	g		

RSA Encryption/Decryption Process

Key Generation:

RSA key generation involves creating a public key (used for encryption) and a private key (used for decryption).

Step I: Choose two distinct primes p, q with $q < p < 2q$.

Step II: Compute $N = pq$ and $\varphi(N) = (p-1)(q-1)$

Step III: Choose a public key e such that GCD $\left(e,\ \varphi(N)\right) = 1$.

Step IV: Compute the private key d such that $ed - k\varphi(N) = 1$.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

Encryption Process

Compute the ciphertexts C using the public key (e, N): $\mathrm{C} = \mathrm{M}^e \pmod{\mathrm{N}}$

Decryption Process

Decrypt the ciphertexts $\boldsymbol{C}$ using the private key $(\boldsymbol{d}, \boldsymbol{N})$: $\boldsymbol{M = C^d \ (mod\ N)}$

Concept of Mapping

Mapping refers to a fundamental concept that describes the **relationship between two sets**, where each element in one set is associated with one or more elements in another set. Formally, if A and B are sets, a *mapping* or *function* f from A to B is defined as:

$$f: A \to B$$

such that for every element $a \in A$, there exists a unique element $b \in B$ satisfying $f(a) = b$. Here, A is called the **domain**, B is the **codomain**, and the set of all actual images $f(a)$ in B is the **range** of the function (Hardy & Wright, 2008).

Sieve of Eratosthenes Algorithm

The Sieve of Eratosthenes is a classical algorithm for generating prime numbers up to a given limit n, (Ameha, 2021).

Algorithm Steps:

1. Create a list of numbers from 2 to n.
2. Begin with the first prime 2.
3. Mark all multiples of each prime as non-prime.
4. Continue until all numbers are processed.
5. The unmarked numbers are primes.

RESULT AND ANALYSIS

This section presents the results and analysis of the research. The findings demonstrated that RSA is vulnerable to attacks that exploit the public key exponent, and that patterns of prime numbers can be used to enhance the efficiency and security of the RSA algorithm.

Cryptanalysis of RSA Using Encryption Key

Cryptanalysis of RSA using the encryption exponent involves exploiting vulnerabilities when the encryption key e, modulus N, and encoding scheme are public. In such cases, an attacker may recover plaintext by systematically encrypting all possible character codes and comparing them with the ciphertexts. Below is an illustration of the cryptanalysis process:

Let (e, N) be a public key

$S =$ Set of standard codes comprising letters, numerals and symbols.

$P =$ Plaintext, where $P \subset S$

$C =$ Ciphertexts $= p_i^e \ (mod\ N)$, where $p_i \in P,\ i = 1, 2, \ldots$

$A =$ Attack $= s_i^e \ (mod\ N)$, where $s_i \in S,\ i = 1, 2, \ldots, n$, $n =$ number of elements in S

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

To have $A \supset C$.

Mapping the values in C with the corresponding values in A, the original message will be recovered.

Suppose a plaintext message "Mission completed" was encrypted using RSA public key $(e, N) = (17,\ 3233)$, and each character of the message was assigned with its corresponding value in ASCII as:

Plaintext	$\mathbb{Z}_N$	$C \equiv P^e (mod\ N)$ **Encryption**
M	077	$077^{17}\ (mod\ 3233) = 3123$
i	105	$105^{17}\ (mod\ 3233) = 3179$
		⋮
d	100	$100^{17}\ \ (mod\ 3233) = 1773$

To have encrypted message as:

$$C = 312331791230123031792185172419920281218122710612074513130884131317 73$$

An attacker can recover the original message by encrypting all the characters in the ASCII table using the public key $(e, N) = (17,\ 3233)$ to have encrypted ASCII values A as:

NUL ⟶	000	$000^{17}\ (mod\ 3233) = 0000$
SOH ⟶	001	$001^{17}\ (mod\ 3233) = 0001$
		⋮
DEL ⟶	127	$127^{17}\ \ (mod\ 3233) = 2355$

Mapping the values in the ciphertexts C with the corresponding values in A, the original message

"Mission completed" will be recovered.

Proposed Improvement of the RSA Algorithm Leveraging Prime Number Distribution Patterns.

Applying patterns of prime numbers to improve RSA algorithm involves four stages. Firstly, generate pattern of prime numbers; secondly, replace the characters of ASCII with the generated pattern sequentially. Thirdly, express the given plaintext in terms of assigned prime numbers and lastly, encrypt the resulting values using RSA method of encryption.

Encryption Process of the Improved Algorithm

Below is the step by step of the encryption process:

Step I: Generate pattern of prime numbers of specific range.

Step II: Map each value of the generated prime to a corresponding value in ASCII

Step III: Express the given plaintext in terms of assigned prime numbers as P_n

Step IV: Generate RSA public key (N, e) and private key (N, d).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

Step V: Encrypt P_n using the RSA public key to obtain ciphertexts $C = P_n^e (mod\ N)$

Decryption Process of the Improved Algorithm

To decrypt the generated ciphertexts C the process of encryption is reverse using RSA decryption exponent d, as in the steps below:

Step I: Decrypt the ciphertexts C to obtain P_n as $P_n = C^d (mod\ N)$

Step II: Express the values of P_n to the corresponding values in ASCII

Step III: Convert the values to the corresponding characters to obtain the plaintext P.

Pattern of Prime Numbers using Sieve of Eratosthenes Technique

The Sieve of Eratosthenes is an ancient algorithm used to find all prime numbers up to a given limit. By applying this technique, pattern of prime numbers can be generated sequentially. For instance, to generate prime numbers corresponding to the total number of characters in ASCII, ranging from 31 to 800 (that is $31 \leq p_i < 800, 1 \leq i \leq 128$, where p_i is prime). This is illustrated in table 2:

Table 2: Sieve of Eratosthenes for Prime Numbers $31 \leq p_i < 800, 1 \leq i \leq 128$

31	33	37	39		411	413	417	419
41	43	47	49		421	423	427	429
51	53	57	59		431	433	437	439
61	63	67	69		441	443	447	449
71	73	77	79		451	453	457	459
81	83	87	89		461	463	467	469
91	93	97	99		471	473	477	479
101	103	107	109		481	483	487	489
111	113	117	119		491	493	497	499
121	123	127	129		501	503	507	509
131	133	137	139		511	513	517	519
141	143	147	149		521	523	527	529
151	153	157	159		531	533	537	539
[redacted]	163	167	169		541	543	547	549
171	173	177	179		551	553	557	559
181	183	187	189		561	563	567	569
191	193	197	199		571	573	577	579
201	203	207	209		581	583	587	589
211	213	217	219		591	593	597	599
221	223	227	229		601	603	607	609
231	233	237	239		611	613	617	619
241	243	247	249		621	623	627	629
251	253	257	259		631	633	637	639
261	263	267	269		641	643	647	649
271	273	277	279		651	653	657	659
281	283	287	289		661	663	667	669
291	293	297	299		671	673	677	679

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

301	303	307	309		681	683	687	689
311	313	317	319		691	693	697	699
321	323	327	329		701	703	707	709
331	333	337	339		711	713	717	719
341	343	347	349		721	723	727	729
351	353	357	359		731	733	737	739
361	363	367	369		741	743	747	749
371	373	377	379		751	753	757	759
381	383	387	389		761	763	767	769
391	393	397	399		771	773	777	779
401	403	407	409		781	783	787	789

Where multiples of two and that of five are deleted, hence, the required pattern of primes is:

31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787.

Applying Patterns of Prime Numbers to Improve the Security of RSA Algorithm

The generated pattern of primes will be applied to replace letters and characters of a plaintexts derived from ASCII table. Below is the integrated ASCII table with assigned prime numbers:

Table 3: Integrated ASCII Table

Dec	0	1	2	3	4	5	6	7	8	9
Chr	**NUL**	**SOH**	**STX**	**ETX**	**EOT**	**ENQ**	**ACK**	**BEL**	**BS**	**HT**
P_i	31	37	41	43	47	53	59	61	67	71
Dec	10	11	12	13	14	15	16	17	18	19
Chr	**LF**	**BT**	**FF**	**CR**	**SO**	**SI**	**DEL**	**DC1**	**DC2**	**DC3**
P_i	73	79	83	89	97	101	103	107	109	113
Dec	20	21	22	23	24	25	26	27	28	29
Chr	**DC4**	**NAK**	**SYN**	**ETB**	**CAN**	**EM**	**SUB**	**ESC**	**FS**	**GS**
P_i	127	131	137	139	149	151	157	163	167	173
Dec	30	31	32	33	34	35	36	37	38	39
Chr	**RS**	**US**	**SPC**	**!**	**“**	**#**	**$**	**%**	**&**	**‘**
P_i	179	181	191	193	197	199	211	223	227	229
Dec	40	41	42	43	44	45	46	47	48	49
Chr	**(**	**)**	*****	**+**	**,**	**-**	**.**	**/**	**0**	**1**
P_i	233	239	241	251	257	263	269	271	277	281

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

Dec	50	51	52	53	54	55	56	57	58	59
Chr	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**:**	**;**
P_i	283	293	307	311	313	317	331	337	347	349
Dec	60	61	62	63	64	65	66	67	68	69
Chr	<	=	>	**?**	**@**	**A**	**B**	**C**	**D**	**E**
P_i	353	359	367	373	379	383	389	397	401	409
Dec	70	71	72	73	74	75	76	77	78	79
Chr	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
P_i	419	421	431	433	439	443	449	457	461	463
Dec	80	81	82	83	84	85	86	87	88	89
Chr	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**
P_i	467	479	487	491	499	503	509	521	523	541
Dec	90	91	92	93	94	95	96	97	98	99
Chr	**Z**	**[**	\	**]**	^	**_**	**`**	**a**	**b**	**c**
P_i	547	557	563	569	571	577	587	593	599	601
Dec	100	101	102	103	104	105	106	107	108	109
Chr	**d**	**e**	**f**	**g**	**H**	**i**	**j**	**k**	**l**	**m**
P_i	607	613	617	619	631	641	643	647	653	659
Dec	110	111	112	113	114	115	116	117	118	119
Chr	**n**	**o**	**p**	**q**	**R**	**s**	**t**	**u**	**v**	**w**
P_i	661	673	677	683	691	701	709	719	727	733
Dec	120	121	122	123	124	125	126	127		
Chr	**x**	**y**	**z**	**{**	**|**	**}**	**~**	**DEL**		
P_i	739	743	751	757	761	769	773	787		

Utilizing the integrated ASCII table to improve the security of the message encrypted in section 4.1, “Mission completed” using RSA public key $(N, e) = (3233, 17)$:

Plaintext		$C \equiv P^e (mod\ N)$
	P_n	**Encryption**
M	457	$457^{17}\ mod\ 3233 = 1482$
i	641	$641^{17}\ mod\ 3233 = 1019$
		⋮
d	607	$607^{17}\ mod\ 3233 = 0314$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

To have new encrypted message as:

$$C = 1482101906810681101927281893045532132728315015561438230925992309031 4$$

The above ciphertexts will be difficult to attack using public key as in normal RSA. With this improvement the original message can be recovered using decryption key d as illustrated below:

Decryption Process:

Computing the decryption exponent d, such that $ed - k\varphi(N) = 1$, with $e = 17$, $\varphi(N) = 3120$ to have $d = 2753$.

To decrypt the ciphertexts C using $p_i = C^d(mod\ N)$ and taking its corresponding value character from Table 3, as:

$C^d(mod\ N) = p_i \rightarrow$ ASCII Value = Plaintext

$1482^{2753}(mod\ 3233) = 457 \rightarrow 077 =$ M

$1019^{2753}(mod\ 3233) = 641 \rightarrow 105 =$ i

⋮

$0314^{2753}\ (mod\ 3233) = 607 \rightarrow 077 =$ d

Hence, the plaintext message "Mission completed" has been obtained.

SUMMARY AND CONCLUSION

The proposed approach was implemented by assigning prime numbers to each ASCII character, covering the full range of values (0 – 127). Instead of directly encrypting ASCII codes, each character was mapped to a unique prime number within a defined range (31–800). The mapped primes were then encrypted using the RSA scheme with modulus and exponent. Every ASCII character had a distinct prime representative. This mapping introduced an additional layer of substitution before RSA encryption, increasing ciphertexts diversity. Ciphertexts generated from prime values were larger than those generated directly from ASCII codes. The use of primes patterns avoided direct correlation between ciphertexts values and ASCII frequencies. Frequency analysis, which exploits repeated characters in plaintext, became significantly harder.

The results also, highlighted that the integration of prime pattern mapping with RSA improves security in two major ways: normal RSA encrypts small integers (ASCII 0–127), which can be predictable and by replacing ASCII values with pattern of primes, the plaintext space becomes unpredictable. Furthermore, the finding showed vulnerability of RSA through public key-based.

Conclusively, the proposed improved RSA algorithm increases resistance against frequency analysis, ciphertexts pattern recognition, and direct recovery from public key attacks.

REFERENCES

1. Ameha, T. T. (2021). Advanced mathematical formulas to calculate prime numbers. Strategic Planning, Commercial Bank of Ethiopia, Addis Ababa, Ethiopia.
2. Aminudina, A., & Cahyonoa, E. B. (2021). A practical analysis of the Fermat factorization and Pollard Rho method for factoring integers. Lontar Komputer, **12**(1). https://doi.org/10.24843/LKJITI.2021
3. Ariffin, M. R. K., & Shehu, S. (2016). *Cryptanalysis on prime power RSA modulus of the form* $N = p^r q$. *International Journal of Applied Mathematical Research*, **5**(4), 167–175.
4. Hardy, G. H., & Wright, E. M. (2008). An introduction to the theory of numbers (6th ed.). Oxford University Press.

5. Hinek, M. J. (2009). Cryptanalysis of RSA and its variants. CRC Press.
6. Kannan, B., & Mohana, P. P. (2023). A survey of Fermat factorization algorithms for factoring RSA composite numbers. Mathematics and Its Applications in Technology: Multidisciplinary Science Journal. https://doi.org/10.3193/
7. Kefa, R. (2006). Review of methods for integer factorization applied to cryptography. Journal of Applied Science, 6(1), 458–481.
8. Nadia, W. N., Syahril, E., Sawaluddin, (2020). Analysis of RSA variants in securing message. *IOP Conf. Series: Materials Science and Engineering*. DOI:10.1088/1757-899X/725/1/012131.
9. Nitaj, A. (2016). New Attacks on RSA with two or three decryption exponents. In *International Conference on Cryptology in Africa; Springer, Cham, Switzerland,* **8**(3), 112–129.
10. Overmars, A., & Venkatraman, S. (2020). Mathematical attack of RSA by extending the sum of squares of primes to factorize a semi-prime. Mathematical and Computational Applications, 25(4), 63. https://doi.org/10.3390/mca25040063
11. Overmars, A., & Venkatraman, S. (2021). New semi-prime factorization and application in large RSA key attacks. Journal of Cybersecurity and Privacy, 1(4), 660–674. https://doi.org/10.3390/jcp1040033
12. Rivest, R. L., Shamir, A., & Adleman, L. (1978). *A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM*, **21**(2), 120–126.
13. Shor, P. W. (1994). *Algorithms for quantum computation: Discrete logarithms and factoring*. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*, 124–134.
14. Somsuk, K., & Kasemvilas, S. (2013). MFFv2 and MNQsv2: Improved factorization algorithms. https://doi.org/10.1109/icisa.2013.6579415
15. Wiener, M. J. (1990). *Cryptanalysis of short RSA secret exponents. IEEE Transactions on Information Theory*, **36**(3), 553–558.