WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 8, Number 3, Year 2026
DOI: https://doi.org/10.62051/ijcsit.v8n3.06
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Possible Reinforcements of RSA and Some Other Encryption Methods

Huini Xu [1], Qiyuan Sun [2]

[1] Cosumnes Oaks High School, California, 95757, USA
[2] WLSA Shanghai Academy, Shanghai, 200940, China

ABSTRACT

This paper explores potential reinforcements of the RSA algorithm and examines several alternative public-key cryptosystems. Building on the mathematical foundation of integer factorization, RSA has been the cornerstone of secure digital communication, yet it faces vulnerabilities from parameter weaknesses and emerging quantum algorithms. To address efficiency and resilience, this study first analyzes multi-prime RSA, highlighting its advantages in decryption speed through the Chinese Remainder Theorem, while also noting its reduced security margins. In addition, the Goldwasser–Micali cryptosystem is evaluated for its probabilistic encryption mechanism, which enhances semantic security by producing randomized ciphertexts. The LUC encryption scheme, based on Lucas sequences, is then discussed as a variant of RSA with potentially stronger resistance against certain attacks. Finally, an algebraic encryption method utilizing polynomial roots is introduced as an innovative approach, though its practical security remains uncertain. Collectively, these explorations illustrate the trade-offs between efficiency, ciphertext size, and security, and point toward future directions in strengthening public-key cryptography against advancing computational threats.

KEYWORDS

RSA; Multi-prime RSA; Goldwasser–Micali; LUC Cryptosystem; Public-key Cryptography

1. INTRODUCTION

Cryptography is a science dealing with secure information storage and transmission. Essentially, it is studying the methods of how to transform plaintext into un-comprehensible ciphertext with specific algorithms, and the methods of how to decode the ciphertext back into plaintext.

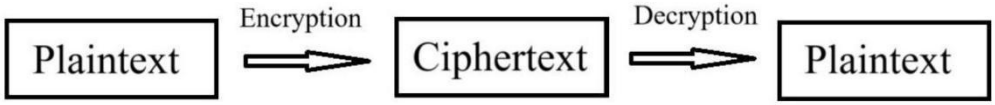

Figure 1. Basic Process of Encryption and Decryption in Cryptographic Systems

The first encryption algorithm appeared over 4000 years ago in ancient Egypt [1]. Those early encryption methods are called classical cryptography, as they use paper and pens or minimal mechanical aids to operate. In the twentieth century, after the mechanical and electronic boom, cipher complexity increased tremendously, with the German Enigma machine being a prominent example. However, computing technology advancements in World War II demonstrated that even highly complex rotor ciphers could be broken after specific algorithms. Nowadays, computers open new possibilities for encryption, enabling mathematically based ciphers rather than mere letter substitution.

Whitfield Diffie and Martin Hellman proposed the concept of public-key cryptography in 1976 [2], where encryption and decryption utilize different but mathematically connected keys. Based on that,

Ron Rivest, Adi Shamir, and Leonard Adleman created the RSA algorithm in 1978 [3]. By exchanging an encryption key (public key) and retaining the associated decryption key (private key), RSA facilitated secure digital communication without pre-exchange of a secret key. Despite its revolutionary impact, RSA is not invulnerable. Poor parameter choices, such as small private exponents d, can lead to low-exponent attacks [4]. RSA security in the future is under attack from quantum computers [5], which can factor large numbers efficiently.

In the following chapter, we will explore techniques for reinforcing the RSA encryption, as well as other encryption methods.

2. THE RSA ENCRYPTION

RSA encryption, introduced by Rivest, Shamir, and Adleman in 1977, is one of the most widely used public-key cryptosystems. It is grounded in the computational difficulty of factoring large integers. The RSA scheme involves two keys: a public key for encryption and a private key for decryption. The modulus N is formed by multiplying two large, distinct primes p and q, i.e.,

$$N = pq$$

The security and correctness of RSA rely on Euler's theorem, which states that for any integer a such that $\gcd(a,n) = 1$:

$$a^{\phi(n)} \equiv 1 \ (\mathrm{mod}\ n)$$

Where $\phi(n)$ is Euler's totient function. In the RSA setup, $\phi(N) = (p-1)(q-1)$. A public exponent e is selected such that $1 < e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$. The private key d is the modular inverse of e modulo $\phi(N)$:

$$ed \equiv 1 \ (\mathrm{mod}\ \phi(N))$$

2.1. Euler's Theorem

Claim: For any integer a such that $\gcd(a,n) = 1$,

$$a^{\phi(n)} \equiv 1 \ (\mathrm{mod}\ n)$$

Proof:

Let n be a positive integer, and let $a \in \mathbb{Z}_n$ with $\gcd(a,n) = 1$. Let $\mathbb{Z}_n^* = \{x \in \mathbb{Z}_n \mid \gcd(x,n) = 1\}$ be the reduced residue system modulo n, containing $\phi(n)$ elements.

Consider:

$$S = \{x_1, x_2, \ldots, x_{\phi(n)}\}$$

Multiplying each element of S by a modulo n:

$$aS = \{ax_1 \bmod n, ax_2 \bmod n, \ldots, ax_{\phi(n)} \bmod n\}$$

Since multiplication by a permutes S (as $\gcd(a,n) = 1$ preserves coprimality), the products are congruent:

$$\prod_{i=1}^{\phi(n)}(ax_i) \equiv a^{\phi(n)} \prod_{i=1}^{\phi(n)} x_i \equiv \prod_{i=1}^{\phi(n)} x_i \ (\bmod n)$$

Since $\gcd(\prod x_i, n) = 1$ (all x_i coprime to n), we have

$$a^{\phi(n)} \equiv 1 \ (\bmod n)$$

This ensures RSA decryption correctness:

$$m^{ed} \equiv m \ (\bmod N), \ \forall m \in \mathbb{Z}_N$$

2.2. Example

Let $p = 61, q = 53$, so $N = 3233$ and $\phi(N) = 3120$. Choose $e = 17$, satisfying $\gcd(17,3120) =$

The private key is $d = 2753$, verified by:

$$17 \times 2753 = 46{,}801 = 15 \times 3120 + 1 \equiv 1 \ (\bmod 3120).$$

Encryption for $m = 65$:

$$c = 65^{17} \bmod 3233 = 2790$$

Decryption:

$$m = 2790^{2753} \bmod 3233 = 65$$

3. RSA ENCRYPTION WITH MORE THAN 2 PRIMES

Multi-prime RSA extends classical RSA by using $r \geq 3$ distinct primes [6] $p_1, \ldots, p_r$ for modulus $N = \prod p_i$. This design improves decryption efficiency via the Chinese Remainder Theorem (CRT), enabling parallel computations on smaller moduli.

When primes are balanced $\left(p_i \approx N^{1/r}\right)$, decryption complexity reduces significantly [7]:

$$\mathcal{O}\left(\frac{3}{2r^3}(\log N)^3\right)$$

Benefiting resource-constrained devices like smart cards.

3.1. Key Generation and Structure

(1) Select $r \geq 3$ distinct primes $p_1, \ldots, p_r$, compute $N = \prod p_i$

(2) Compute $\phi(N) = \prod(p_i - 1)$

(3) Choose e with $\gcd\left(e, \phi(N)\right) = 1$

(4) Compute $d \equiv e^{-1}\left(\bmod \phi(N)\right)$

Encryption/decryption remains:

$$c = m^e \bmod N, \ m = c^d \bmod N$$

3.2. Security Implications

While improving efficiency, multi-prime RSA reduces prime sizes, increasing vulnerability to factoring attacks like ECM.

Table 1. Maximum Safe Number of Primes for Different Modulus Sizes

Modulus Size (bits)	Max Safe Number of Primes
1024	3
2048	4
4096	5

Note: Values assume balanced primes and resistance to ECM attacks

Wiener-type attack bounds weaken [8] to $d < N^{1/(2r)}$, and lattice attacks strengthen with partial key exposure. Implementers must balance speed against security.

Multi-prime RSA significantly accelerates decryption via CRT parallelization. However, designers must:

(1) Adhere to Figure 1 prime counts

(2) Ensure $d > N^{1/(2r)}$

(3) Maintain $p_i \in \left[0.5N^{1/r}, 2N^{1/r}\right]$

These measures preserve security while leveraging efficiency gains.

3.3. Example

Using primes $p_1 = 101, p_2 = 103, p_3 = 107$:

$$N = 101 \times 103 \times 107 = 1{,}111{,}321$$

$$\phi(N) = 100 \times 102 \times 106 = 1{,}081{,}200$$

$$e = 7 \ (\gcd(7{,}1{,}081{,}200) = 1)$$

$$d = 7^{-1} \bmod 1{,}081{,}200 = 463{,}371$$

$$c = 100^7 \bmod 1{,}111{,}321 = 1{,}000{,}000$$

$$m_{\text{dec}} = 1{,}000{,}000^{463{,}371} \bmod 1{,}111{,}321 = 100$$

This demonstrates multi-prime RSA's functional equivalence with improved efficiency.

4. GOLDWASSER-MICALI CRYPTOSYSTEM

The Goldwasser-Micali cryptosystem was introduced by Shafi Goldwasser and Silvio Micali in 1982. Different from the RSA, it introduces randomness to each encryption operation. Encrypting a bit twice yields distinct and independent ciphertexts. Its security relies on the difficulty of the quadratic residuosity problem modulo a chosen composite number (also a product of two primes), where it is hard to distinguish random quadratic residues from non-residues without knowing the chosen composite number.

4.1. Quadratic Residue

For a prime number p, if there exists y such that

$$y^2 \equiv a \ (\mathrm{mod}\ p)$$

We call a a quadratic residue $\mathrm{mod}\ p$.

The Legendre symbol:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p \mid a \\ 1, & \text{if } a \text{ is a quadratic residue mod } p \\ -1, & \text{if } a \text{ is not a quadratic residue mod} \end{cases}$$

Claim (Euler's criterion):

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \ (\mathrm{mod}\ p)$$

Proof:

When $p \mid a$ the problem is trivial. We consider the other two cases.

By Fermat's little theorem,

$$a^{p-1} \equiv 1 \ (\mathrm{mod}\ p)$$

Suppose $a^{\frac{p-1}{2}} \equiv x(\mathrm{mod}\ p)$. Note that $x^2 \equiv 1(\mathrm{mod}\ p) \Rightarrow x^2 - 1 \equiv 0(\mathrm{mod}\ p) \Rightarrow (x+1)(x-1) \equiv 0(\mathrm{mod}\ p) \Rightarrow x \equiv 1(\mathrm{mod}\ p)$ or $x \equiv -1(\mathrm{mod}\ p)$

(i) a is a quadratic residue $\mathrm{mod}\ p \Rightarrow a^{\frac{p-1}{2}} \equiv 1(\mathrm{mod}\ p)$

Since a is a quadratic residue, there exists x such that

$$x^2 \equiv a \ (\mathrm{mod}\ p) \Rightarrow a^{\frac{p-1}{2}} \equiv (x^2)^{\frac{p-1}{2}} \equiv x^{p-1} \equiv 1 \ (\mathrm{mod}\ p)$$

Where $x^{p-1} \equiv 1(\mathrm{mod}\ p)$ by Fermat's little theorem.

(ii) a is a quadratic residue $\mathrm{mod}\ p \Leftarrow a^{\frac{p-1}{2}} \equiv 1(\mathrm{mod}\ p)$

Consider a primitive root $k \bmod p$ and $0 \le i < j \le p-2$. Since k is a primitive root, we have $k^{p-1} \equiv 1(\mathrm{mod}\ p)$ and $k^r \not\equiv 1(\mathrm{mod}\ p)$ for $r < p-1$. If $k^i \equiv k^j(\mathrm{mod}\ p)$, we have $k^{j-i} \equiv 1$ (mod p) where $j-i < p-1$ giving a contradiction. Thus, for any $0 \le i < j \le p-2, k^i \not\equiv k^j$ $(\mathrm{mod}\ p)$. Thus $\{k^0, k^1, \cdots k^{p-2}\}$ forms a reduced residue system $\mathrm{mod}\ p$. Thus, there exits t such that $k^t \equiv a(\mathrm{mod}\ p)$.

$$a^{\frac{p-1}{2}} \equiv 1 \ (\mathrm{mod}\ p) \Rightarrow (k^t)^{\frac{p-1}{2}} \equiv 1 \ (\mathrm{mod}\ p) \Rightarrow k^{\frac{t}{2}\cdot(p-1)} \equiv 1 \ (\mathrm{mod}\ p)$$

Since k is a primitive root, we have $p-1 \mid \frac{t}{2} \cdot (p-1) \Rightarrow \frac{t}{2}$ is an integer. Thus, $a \equiv k^t \equiv \left(k^{\frac{t}{2}}\right)^2$ $(\mathrm{mod}\ p)$ indicating a is a quadratic residue.

Thus, a is a quadratic residue $\mathrm{mod}\ p \Leftrightarrow a^{\frac{p-1}{2}} \equiv 1(\mathrm{mod}\ p)$ and $a^{\frac{p-1}{2}} \equiv \pm 1(\mathrm{mod}\ p)$, so a is a not quadratic residue mod $p \Leftrightarrow a^{\frac{p-1}{2}} \equiv -1(\mathrm{mod}\ p)$ and so the statement is proven.

The Jacobi symbol:

$$\left(\frac{a}{m}\right)=\left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\cdots\left(\frac{a}{p_k}\right)$$

Where $(a,m)=1$ and $m=p_1p_2\cdots p_k, p_i\neq 2.$

For both Legendre symbol and Jacobi symbol, we have

(1) $\left(\frac{x}{m}\right)=\left(\frac{y}{m}\right)$ if $x\equiv y(\bmod m)$

(2) $\left(\frac{x_1x_2\cdots x_n}{m}\right)=\left(\frac{x_1}{m}\right)\left(\frac{x_2}{m}\right)\cdots\left(\frac{x_n}{m}\right)$

4.2. Encryption and Decryption Process

Pick two large primes p and q and compute $N=pq$. Find a number x such that

$$\left(\frac{x}{N}\right)=1 \text{ and } \left(\frac{x}{p}\right)=\left(\frac{x}{q}\right)=-1$$

In other words x is not a quadratic residue both $\bmod p$ and $\bmod q$. Then transfer the plaintext into binary numbers.

For any $b_i\in\{0,1\}$, if $b_i=0$, pick a random $k\in\mathbb{Z}_N^*$ and the ciphertext $c_i=k^2(\bmod N)$, if $b_i=1$, pick a random $k\in\mathbb{Z}_N^*$ and the ciphertext $c_i=x\cdot k^2(\bmod N)$

The decryption process is to compute $\left(\frac{c_i}{p}\right)$ and $\left(\frac{c_i}{q}\right)$ respectively. If $\left(\frac{c_i}{p}\right)=\left(\frac{c_i}{q}\right)=1$, output $b_i=0$, otherwise $b_i=1$.

4.3. Example and Evaluation

Label the letters a to z from 1 to 26. We encrypt the word mathematics. Pick $p=5$ and $q=7$, so $N=35$. Note that $\left(\frac{17}{35}\right)=1$ and $\left(\frac{17}{5}\right)=\left(\frac{17}{7}\right)=-1$, so let $x=17$. (For convenience here ignore the leading zeros)

$$\text{m}\rightarrow 13\rightarrow 1101$$

$$\text{a}\rightarrow 1\rightarrow 1$$

$$\text{t}\rightarrow 20\rightarrow 10100$$

$$\text{h}\rightarrow 8\rightarrow 1000$$

$$\text{e}\rightarrow 5\rightarrow 101$$

$$\text{m}\rightarrow 13\rightarrow 1101$$

$$\text{a}\rightarrow 1\rightarrow 1$$

$$\text{t}\rightarrow 20\rightarrow 10100$$

$$\text{i}\rightarrow 9\rightarrow 1001$$

$$c \rightarrow 3 \rightarrow 11$$

$$s \rightarrow 19 \rightarrow 10011$$

To encrypt m, pick four random k as $\{2,3,4,6\}$, then the ciphertext is $\{17 \cdot 2^2, 17 \cdot 3^2, 4^2, 17 \cdot 6^2\} \Rightarrow \{33,13,16,17\}$. It is also possible to pick four random k as $\{27,29,31,32\}$, then the ciphertext is $\{17 \cdot 27^2, 17 \cdot 29^2, 31^2, 17 \cdot 32^2\} \Rightarrow \{3,17,16,13\}$. Repeat this method and so the ciphertext for mathematics is

$$m \rightarrow 33,13,16,17$$

$$a \rightarrow$$

$$t \rightarrow 12,16,33,29,11$$

$$h \rightarrow 13,9,11,29$$

$$e \rightarrow 33,16,12$$

$$m \rightarrow 3,17,16,13$$

$$a \rightarrow 33$$

$$t \rightarrow 17,4,13,16,1$$

$$i \rightarrow 3,11,16,33$$

$$c \rightarrow 3,12$$

$$s \rightarrow 13,9,11,3,33$$

For decryption,

$$\left(\frac{33}{5}\right) \equiv 33^{\frac{5-1}{2}} \equiv -1 \pmod 5 \quad \left(\frac{33}{7}\right) \equiv 33^{\frac{7-1}{2}} \equiv -1 \pmod 7$$

$$33 \rightarrow 1$$

$$\left(\frac{13}{5}\right) \equiv 13^{\frac{5-1}{2}} \equiv -1 \pmod 5 \quad \left(\frac{13}{7}\right) \equiv 13^{\frac{7-1}{2}} \equiv -1 \pmod 7$$

$$13 \rightarrow 1$$

$$\left(\frac{16}{5}\right) \equiv 16^{\frac{5-1}{2}} \equiv 1 \pmod 5 \quad \left(\frac{16}{7}\right) \equiv 16^{\frac{7-1}{2}} \equiv 1 \pmod 7$$

$$16 \rightarrow 0$$

$$\left(\frac{17}{5}\right) \equiv 17^{\frac{5-1}{2}} \equiv -1 \pmod 5 \quad \left(\frac{17}{7}\right) \equiv 17^{\frac{7-1}{2}} \equiv -1 \pmod 7$$

$$17 \rightarrow 1$$

Thus $\{33, 13, 16, 17\} \rightarrow 1101 \rightarrow 13 \rightarrow$ m.

The Goldwasser-Micali encryption's strength lies in its random nature, where encrypting the same plaintext can yield completely different ciphertexts. This can avoid attackers recover plaintext by focusing on repeated letters. The RSA encryption, for example, can be potentially broken down by collecting highly repeated numbers and assume them to be some statistically most commonly used letters in Endlish vocabulary. The Goldwasser-Micali encryption's security is based on the quadratic residuo problem. However, the privacy of the Goldwasser-Micali encryption totally rely on the factorization of N , making it vulnerable against computers and algorithms that can break down large composites efficiently.

5. THE LUC ENCRYPTION

LUC was introduced by Peter Smith and Chris Skinner in 1994 [9]. It is a public-key system in which the exponentiation step of RSA is replaced by terms of a Lucas sequence. The Lucas sequence is a generalization of the Fibonacci sequence. In LUC encryption system, the encryption process is to compute a specific sequence term modulo the product of two large primes, and the decryption process uses a complementary sequence term chosen so that the two operations undo each other. For some known attacks, it can be proven that LUC is more secure than RSA in some cases [10].

5.1. The Lucas Sequence

For two fixed parameters $P, Q \in \mathbb{Z}$, the Lucas sequence $U_n(P,Q)$ and $V_n(P,Q)$ are defined as:

$$\begin{cases} U_0(P,Q) = 0 \\ U_1(P,Q) = 1 \\ U_{n+2}(P,Q) = P \cdot U_{n+1}(P,Q) - Q \cdot U_n(P,Q) \end{cases}$$

$$\begin{cases} V_0(P,Q) = 2 \\ V_1(P,Q) = P \\ V_{n+2}(P,Q) = P \cdot V_{n+1}(P,Q) - Q \cdot V_n(P,Q) \end{cases}$$

Claim: Let $f(x) = x^2 - Px + Q$, and α and β are roots for $f(x) = 0$.

$$U_n(P,Q) = \frac{\alpha^n - \beta^n}{\alpha - \beta} \quad V_n(P,Q) = \alpha^n + \beta^n$$

Proof:

$$X_{n+2}(P,Q) - kX_{n+1}(P,Q) = (P-k)\left(X_{n+1}(P,Q) - \frac{Q}{P-k} \cdot X_n(P,Q)\right)$$

Let $k = \frac{Q}{P-k}$, we have $k^2 - Pk + Q = 0 \Rightarrow k = \alpha$. By Vieta theorem, $P - \alpha = \beta$. Thus,

$$X_{n+2}(P,Q) - \alpha X_{n+1}(P,Q) = \beta\big(X_{n+1}(P,Q) - \alpha \cdot X_n(P,Q)\big)$$

$$\Rightarrow X_n(P,Q) - \alpha X_{n-1}(P,Q) = \beta^{n-1}\big(X_1(P,Q) - \alpha \cdot X_0(P,Q)\big)$$

Let $M = \frac{X_1(P,Q) - \alpha \cdot X_0(P,Q)}{\alpha - \beta}$, which is a constant.

$$\Rightarrow X_n(P,Q) + M \cdot \beta^n = \alpha(X_{n-1}(P,Q) + M \cdot \beta^{n-1})$$

$$\Rightarrow X_n(P,Q) + M \cdot \beta^n = \alpha^{n-1}(X_1(P,Q) + M \cdot \beta^1)$$

$$\text{For } U_n(P,Q), M = \frac{U_1(P,Q) - \alpha \cdot U_0(P,Q)}{\alpha - \beta} = \frac{1}{\alpha - \beta}$$

$$\Rightarrow U_n(P,Q) + \frac{1}{\alpha - \beta} \cdot \beta^n = \alpha^{n-1}\left(1 + \frac{1}{\alpha - \beta} \cdot \beta\right)$$

$$\Rightarrow U_n(P,Q) = \frac{\alpha^n - \beta^n}{\alpha - \beta}$$

For $V_n(P,Q), M = \frac{V_1(P,Q) - \alpha \cdot V_0(P,Q)}{\alpha - \beta} = \frac{P - 2\alpha}{\alpha - \beta} = -1$

$$\Rightarrow V_n(P,Q) - \beta^n = \alpha^{n-1}(P - \beta)$$

$$\Rightarrow V_n(P,Q) = \alpha^n + \beta^n$$

Claim:

$$U_{m+n}(P,Q) = U_m(P,Q)V_n(P,Q) - Q^n U_{m-n}(P,Q)$$

$$V_{m+n}(P,Q) = V_m(P,Q)V_n(P,Q) - Q^n V_{m-n}(P,Q)$$

Proof:

$$U_{m+n}(P,Q) = \frac{\alpha^{m+n} - \beta^{m+n}}{\alpha - \beta}$$

$$U_m(P,Q)V_n(P,Q) - Q^n U_{m-n}(P,Q) = \frac{\alpha^m - \beta^m}{\alpha - \beta} \cdot (\alpha^n + \beta^n) - (\alpha\beta)^n \frac{(\alpha^{m-n} - \beta^{m-n})}{\alpha - \beta}$$

$$= \frac{\alpha^{m+n} - \beta^{m+n} - \alpha^n\beta^m + \alpha^m\beta^n}{\alpha - \beta} - \frac{\alpha^m\beta^n - \alpha^n\beta^m}{\alpha - \beta}$$

$$= \frac{\alpha^{m+n} - \beta^{m+n}}{\alpha - \beta}$$

$$V_{m+n}(P,Q) = \alpha^{m+n} + \beta^{m+n}$$

$$V_m(P,Q)V_n(P,Q) - Q^n V_{m-n}(P,Q) = (\alpha^m + \beta^m)(\alpha^n + \beta^n) - (\alpha\beta)^n(\alpha^{m-n} + \beta^{m-n})$$

$$\alpha^{m+n} + \beta^{m+n} + \alpha^n\beta^m + \alpha^m\beta^n - \alpha^m\beta^n - \alpha^n\beta^m$$

$$= \alpha^{m+n} + \beta^{m+n}$$

Claim:

$$U_n(P \bmod N, Q \bmod N) \equiv U_n(P,Q) \pmod N$$
$$V_n(P \bmod N, Q \bmod N) \equiv V_n(P,Q) \pmod N$$

Proof:

When $n = 0,1$, the statement is true. Assume that the statement is true for $n = k$ and $n = k + 1$, consider $n = k + 2$.

$$U_{k+2}(P \bmod N, Q \bmod N)$$

$$\equiv (P \bmod N)U_{k+1}(P \bmod N, Q \bmod N) - (Q \bmod N)U_k(P \bmod N, Q \bmod N) \pmod N$$

$$\equiv P \cdot U_{k+1}(P,Q) - Q \cdot U_k(P,Q) \pmod N$$

$$\equiv U_{k+2}(P,Q)$$

By mathematical induction, the statement is true for all non-negative integers n. The case for $V_n(P,Q)$ is identical.

Claim:

$$V_{mn}(P,Q) = V_m(V_n(P,Q), Q^n)$$

Proof:

$$V_{mn}(P,Q) = \alpha^{mn} + \beta^{mn}$$

$$V_m(V_n(P,Q), Q^n) = V_m(\alpha^n + \beta^n, (\alpha\beta)^n) = \alpha_0^m + \beta_0^m$$

Where α_0 and β_0 are roots for $x^2 - (\alpha^n + \beta^n)x + (\alpha\beta)^n = 0$.

$$(x - \alpha^n)(x - \beta^n) = 0 \Rightarrow \alpha_0 = \alpha^n, \beta_0 = \beta^n$$

$$V_m(V_n(P,Q), Q^n) = V_m(\alpha^n + \beta^n, (\alpha\beta)^n) = \alpha^{mn} + \beta^{mn}$$

Claim:

$$2Q^n V_{m-n}(P,Q) = V_m(P,Q)V_n(P,Q) - (\alpha - \beta)^2 U_m(P,Q)U_n(P,Q)$$

Proof:

$$2Q^n V_{m-n}(P,Q) = 2(\alpha\beta)^n(\alpha^{m-n} + \beta^{m-n}) = 2\alpha^m\beta^n + 2\alpha^n\beta^m$$

$$V_m(P,Q)V_n(P,Q) - (\alpha - \beta)^2 U_m(P,Q)U_n(P,Q)$$

$$= (\alpha^m + \beta^m)(\alpha^n + \beta^n) - (\alpha - \beta)^2 \cdot \frac{\alpha^m - \beta^m}{\alpha - \beta} \cdot \frac{\alpha^n - \beta^n}{\alpha - \beta}$$

$$= \alpha^{m+n} + \beta^{m+n} + \alpha^m\beta^n + \alpha^n\beta^m - (\alpha^{m+n} + \beta^{m+n} - \alpha^m\beta^n - \alpha^n\beta^m)$$

$$= 2\alpha^m\beta^n + 2\alpha^n\beta^m$$

Theorem [11]: If p is an odd prime that does not divide $\alpha\beta$ or $\alpha - \beta$, then

$$U_{k\left(p-\left(\frac{D}{p}\right)\right)}(P,Q) \equiv 0 \pmod p$$

$$V_{k\left(p-\left(\frac{D}{p}\right)\right)}(P,Q) \equiv 2Q^{\frac{k\left(1-\left(\frac{D}{p}\right)\right)}{2}} \pmod p$$

Where $\left(\frac{D}{p}\right)$ is the Legendre symbol.

5.2. Encryption and Decryption Process

Pick two large primes p and q and compute $N = pq$. Find a number $r \in \mathbb{Z}_N$ such that r is co-prime to $(p-1)(p+1)(q-1)(q+1)$.

For plaintext $b_i \in \mathbb{Z}_N$, the ciphertext is $c_i = V_r(b_i, 1) \bmod N$

For the decryption process, let $M = \text{lcm}\left(p - \left(\frac{c_i^2-4}{p}\right),\ q - \left(\frac{c_i^2-4}{q}\right)\right)$ and $rs \equiv 1(\bmod M)$.

Then the plaintext can be recovered by $b_i = V_s(c_i, 1) \bmod N$. Utilizing the claims and theorem presented in 4.1, the decryption process can be explained as:

$$V_s(c_i, 1) = V_s(V_r(b_i, 1), 1)$$

$$= V_{sr}(b_i, 1)$$

$$= V_{kM+1}(b_i, 1)$$

$$= V_{kM}(b_i, 1)V_1(b_i, 1) - V_{kM-1}(b_i, 1)$$

$$= b_i V_{kM}(b_i, 1) - \frac{1}{2}\left(V_{kM}(b_i, 1)V_1(b_i, 1) - (\alpha - \beta)^2 U_{kM}(b_i, 1)U_1(b_i, 1)\right)$$

$$= 2b_i - \frac{1}{2}(2b_i - 0) \bmod N$$

$$= b_i$$

5.3. Example and Evaluation

Label the letters a to z from 1 to 26. We encrypt the word math. Pick $p = 17$ and $q = 23$, so $N = 391$. Let $r = 5$, a quick check shows 5 is co-prime to $(17-1)(17+1)(23-1)(23+1)$.

$$\text{m} \to 13 \to V_5(13,1) = 262 \bmod 391$$

$$\text{a} \to 1 \to V_5(1,1) = 1 \bmod 391$$

$$\text{t} \to 20 \to V_5(20,1) = 38 \bmod 391$$

$$\text{h} \to 8 \to V_5(8,1) = 141 \bmod 391$$

So the ciphertext for math is $\{262,1,38,141\}$.

For decryption,

$$M = \text{lcm}\left(17 - \left(\frac{262^2 - 4}{17}\right), 23 - \left(\frac{262^2 - 4}{23}\right)\right) = 198$$

$$5s \equiv 1 \ (\bmod 198) \Rightarrow s = 119$$

$$V_{119}(262,1) = 13 \bmod 391$$

$$262 \rightarrow 13 \rightarrow \mathbf{m}$$

$$M = \text{lcm}\left(17 - \left(\frac{1^2 - 4}{17}\right), 23 - \left(\frac{1^2 - 4}{23}\right)\right) = 72$$

$$5s \equiv 1 \ (\bmod 72) \Rightarrow s = 29$$

$$V_{29}(1,1) = 1 \bmod 391$$

$$1 \rightarrow 1 \rightarrow \mathbf{a}$$

$$M = \text{lcm}\left(17 - \left(\frac{38^2 - 4}{17}\right), 23 - \left(\frac{38^2 - 4}{23}\right)\right) = 72$$

$$5s \equiv 1 \ (\bmod 72) \Rightarrow s = 29$$

$$V_{29}(38,1) = 20 \bmod 391$$

$$38 \rightarrow 20 \rightarrow \mathbf{t}$$

$$M = \text{lcm}\left(17 - \left(\frac{141^2 - 4}{17}\right), 23 - \left(\frac{141^2 - 4}{23}\right)\right) = 48$$

$$5s \equiv 1 \ (\bmod 48) \Rightarrow s = 29$$

$$V_{29}(141,1) = 8 \bmod 391$$

$$141 \rightarrow 8 \rightarrow \mathbf{h}$$

The python code below can be used to calculate Lucas sequence efficiently.

```
def lucas_v(n, P, Q, mod):
V0, V1 = 2 % mod, P % mod
        if n == 0:
                return VO
        elif n == 1:
                return V1
        for _ in range (2, n + 1):
                                V_n = (P ⋆ V1 − Q ⋆ V0)%mod
V0, V1 = V1, Vn
```

return Vn

---

n =

P =

Q =

mod =

---

result = lucas_v(n, P, Q, mod)

print(f"V_{n}({P},{Q}) mod {mod} = {result}")

As shown, the encryption and decryption of the LUC system is more complicated then the RSA. It is believed that LUC requires more computation than RSA [12]. Moreover, the LUC system restricted $Q = 1$, indicating it is not utilizing full potential of the Lucas sequence. It is possible to give Q other values and so construct a more complicated encryption system than LUC.

6. AN ALGEBRAIC ENCRYPTION METHOD

The is an innovative and interesting encryption algorithm that utilizes basic principles of roots of polynomials and calculus. The encryption method is described as below:

For plaintext $b_1 b_2 \cdots b_n$, divide it into non-decreasing sub-sequences. For each sub-sequence $b_i b_{i+1} \cdots b_j$, generate random odd primes $p_i p_{i+1} \cdots p_j$, where $b_m \neq q_n$ for any m, n. Then define

$$f(x) = \prod_{k=i}^{j} (x - b_k)^{p_k} \prod_{l=i}^{j} (x - p_k)^{b_k}$$

$$g(x) = \int f(x) \mathrm{d}x$$

The ciphertext is all roots for $g(x) = 0$, regardless of order. For decryption, $g(x)$ can be recovered by its roots and so recover $f(x)$. The strength of this encryption algorithm rely on difficulty of factorizing $f(x)$ without knowing the sequence of odd primes $p_i p_{i+1} \cdots p_j$, as well as distinguishing the correct roots that are the plaintext wanted.

We encrypt the word REZHI as an example. Label the letters a to z from 1 to 26.

$$\mathrm{R} \rightarrow 18$$

$$\mathrm{E} \rightarrow 5$$

$$\mathrm{Z} \rightarrow 26$$

$$\mathrm{H} \rightarrow 8$$

$$\mathrm{I} \rightarrow 9$$

$$\{18,5,26,8,9\} \to \{18\},\{5,26\},\{8,9\}$$

Generate primes $\{7\},\{11,13\},\{17,19\}$

$$f_1(x) = (x-18)^7(x-7)^{18}$$

$$f_2(x) = (x-5)^{11}(x-11)^5(x-26)^{13}(x-13)^{26}$$

$$f_3(x) = (x-8)^{17}(x-17)^8(x-9)^{19}(x-19)^9$$

Below is the python code that gives the ciphertext when input the sub-sequence and odd primes.

```
import numpy as np
        def encrypt_subsequence(subsequence, primes):
        assert len (subsequence) == len (primes), ""
        f = np.poly1d([1])
        for b_r, p_r in zip(subsequence, primes):
#(x - b_r)p̂_r
term1 = np.poly1d([1, -b_r]) ** p_r
#(x - p_r) b̂_r
term2 = np.poly1d([1, -p_r]) ** b_r
f = np.polymul(f, np.polymul(term1, term2))
g =     f.integ(     )
        roots =     g.r
return roots
                if __name__ == '__main_':
subsec = []
primes = []
roots = encrypt_subsequence(subseq, primes)
print ("Ciphertext roots for given subsequence: ")
for r in roots:
print(r)
```

For $\{18\}$ and $\{7\}$, the ciphertext is

$$(51.22065882518969 \pm 8.267524281832303j)(43.771756577556026 \pm 22.969688916092736j)$$
$$(30.62536090555549 \pm 32.59636684734452j)(14.90522578376057 \pm 35.07510280622937j)$$
$$(0.4588921822106251 \pm 29.972396094144536j)(-8.771451563943296 \pm 18.530107403302054j)$$

$$(-3.051457725980207 + 0j)(1.2104112408800058 \pm 0.3051913383044657j)(0.9208118768458669 \pm 0.8456046538239602j)(0.40543125606442154 \pm 1.177394250122765j)(-0.2566602771348726 \pm 1.056250777595812j)(-0.8610529967776561 \pm 0.8912518299530275j)(-1.063654947217006 \pm 0.22387591496225584\text{j})$$

7. CONCLUSION

In conclusion, standard RSA encryption with two primes is the cornerstone of public-key cryptography. Future developments such us using more than two primes can reinforce this encryption strategy. Utilizing other number theory concepts and theorems, other encryption systems such as Goldwasser-Micali encryption is secure under quadratic residuosity and its ability to generate distinct ciphertext for the same plaintext. However, Goldwasser-Micali encryption is limited by the size of its ciphertexts and so it cannot be efficiently used in bulk data encryption. LUC encryption utilizes Lucas sequences to reach equivalent performance to RSA and can be proved to have better security in most cases. Each encryption systems emphasize on different aspects of efficiency, ciphertext size, and security level. Cryptography will face more challenges in the future, as attacking algorithms and computers rapidly develop.

ACKNOWLEDGEMENTS

The two authors have contributed equally to this work.

REFERENCES

[1] Cypher Research Laboratories (2006) *A brief history of cryptography*. 24 January. Available at: https://www.cypher.com.au/crypto_history.htm (Accessed: 25 July 2025)

[2] Diffie, W. and Hellman, M. (1976) 'New directions in cryptography', *IEEE Transactions on Information Theory*, 22(6), pp. 644–654. Available at: https://ieeexplore.ieee.org/document/1055638 (Accessed: 25 July 2025)

[3] 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), pp. 120–126. Available at: https://dl.acm.org/doi/10.1145/359340.359342 (Accessed: 25 July 2025)

[4] Boneh, D. and Durfee, G. (1999) 'Cryptanalysis of RSA with private key d less than N^0.292', in Stern, J. (ed.) *Advances in Cryptology – EUROCRYPT'99*. Lecture Notes in Computer Science, vol. 1592. Berlin, Heidelberg: Springer, pp. 1–11. Available at: https://doi.org/10.1007/3-540-48910-X_1 (Accessed: 25 July 2025)

[5] Shor, P.W. (1994) 'Algorithms for quantum computation: discrete logarithms and factoring', *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. Available at: https://doi.org/10.1109/SFCS.1994.365700 (Accessed: 25 July 2025)

[6] Hinek, M.J. (2008) 'On the security of multi-prime RSA', *Journal of Mathematical Cryptology*, 2(2), pp. 117–147

[7] Boneh, D. and Shacham, H. (2002) 'Fast variants of RSA', *CryptoBytes*, 5(1), pp. 1–9

[8] Bernstein, D.J., Chang, Y.-A. and Cheng, C.-M. (2007) 'Multi-factor RSA', *IACR Cryptology ePrint Archive*, 2007:227

[9] Smith, P. and Skinner, C. (1995) 'A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms', *Journal of Cryptology*. Available at: https://doi.org/10.1007/BFb0000447 (Accessed: 28 July 2025)

[10] Sarbini, I.N., Wong, T.J., Koo, L.F., Rasedee, A.F.N., Naning, F.H. and Abdul Sathar, M.H. (2023) 'Security Analysis on LUC-type Cryptosystems Using Common Modulus Attack', *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 29(3), pp. 206–213. doi: 10.37934/araset.29.3.206213

[11] Lehmer, D.H. (1930) 'An extended theory of Lucas' functions', *Annals of Mathematics*, 31(3), pp. 419–448. Available at: https://doi.org/10.2307/1968235 (Accessed: 29 July 2025)

[12] Sahu, S.K. and Sahu, R.K. (2010) 'Reduce Computation Steps Can Increase the Efficiency of Computation in Lucas Cryptosystem', *Journal of Computer Science and Security*, 6(4), pp. 1203–1207. Available at: https://thescipub.com/pdf/10.3844/jcssp.2010.1203.1207.pdf (Accessed: 29 July 2025)