RESEARCH ARTICLE OPEN ACCESS

Generalized RSA: An approach based on multiplicative level lines

Cédric Kabeya Tshiseba [1], Salem Kumpovela [2]

[1] Department of mathematics and computer science, National pedagogical university, Congo-Kinshasa
[2] Department of mathematics and computer science, National pedagogical university, Congo-Kinshasa

ABSTRACT
This paper explores an extension of the RSA encryption system, called generalized RSA, which relies on the multiplicative level lines of integers modulo n. We introduce the line indicator function and analyze its properties. We formalize the generalized RSA by introducing new parameters and demonstrate that this encryption system can achieve maximum generalization for certain generators. We establish relationships between the line indicator and Euler's totient. Finally, we present an algorithm for key generation within the framework of generalized RSA.
Keywords:- Generalized RSA, Euler's totient function, line indicator function, multiplicative level line, public key cryptography.

1. INTRODUCTION

Since its creation by Rivest, Shamir, and Adleman ([1], [4], [5]) in 1978, the RSA encryption system has captured the attention of researchers in mathematical cryptography due to the challenge it poses with one of the most fundamental mathematical problems: the factorization of large integers into prime components ([4], [13]). Indeed, the security of RSA relies on the assumption that factoring a large integer into its prime factors is a problem difficult to solve in polynomial time [10].

To introduce the RSA encryption system, consider a modulus $n = pq$, where p and q are distinct prime numbers [1]. The RSA can be defined by the quintuple (M, K, C, E, D). In this definition:

- M represents the set of plaintext messages m composed of positive integers.
- K is the set of keys used for encryption and decryption, comprising a public key $k_e = (e, n)$, where e is the encryption exponent, and a private key $k_d = (d, n)$ with d as the decryption exponent.
- C is the set of ciphertexts, also positive integers.
- E is the set of encryption functions $\beta : M \times K \to C$ producing a ciphertext $c = \beta(m, k_e) = m^e \pmod n$.
- D is the set of decryption functions $\gamma : C \times K \to M$ producing a plaintext $m = \gamma(c, k_d) = c^d \pmod n$.

The RSA principle relies on the use of two mathematically linked but distinct keys: a public key for encryption and its corresponding private key for decryption ([1], [4]). To establish a fundamental relationship between encryption and decryption operations, RSA relies significantly on Euler's theorem [13].

Theorem 1 (Euler). For any integer n with its Euler's totient $\phi(n)$, if a is coprime to n, then $a^{\phi(n)} \equiv 1 \pmod n$.

Proof. Euler's totient $\phi(n)$ [5] is defined as the number of positive integers less than n that are coprime to n. If a is coprime to n, then $\gcd(a, n) = 1$.

Consider the set of positive integers less than n that are coprime to n. This set forms the multiplicative group of units modulo n, denoted $U(n)$ (see [5]).

The order of an element a in $U(n)$ is the smallest positive integer k such that $a^k \equiv 1 \bmod(n)$ [12].

For any integer a in $U(n)$, the order of a divides Euler's totient $\phi(n)$. This implies that $a^{\phi(n)} \equiv 1 \pmod n$. □

Theorem 2. In an RSA encryption system with modulus n, any message m, public key k_e, and corresponding private key k_d satisfy: $\gamma(\beta(m, k_e), k_d) = m$.

Proof. Let k_e be the public key and k_d the corresponding private key, and m a plaintext message.

Apply the encryption function β to message m using the public key k_e:

$$c = \beta(m, k_e) = m^e \pmod n$$

Now apply the decryption function γ to the ciphertext c using the private key k_d:

$$m' = \gamma(c, k_d) = c^d \pmod n$$

We want to show that $m' = m$. Consider the expression for m':

$$m' = c^d \pmod n = (m^e)^d \pmod n$$

By the property of congruences [4] modulo n, we can write $(m^e)^d$ as $m^{ed} \pmod n$.

Since k_e and k_d are a valid RSA key pair, we know that d is the inverse of e modulo $\phi(n)$. This means $ed - 1$ is a multiple $\phi(n)$.

We can write $ed - 1$ as $\phi(n) \cdot k$ for some k. Thus, we have:

$m' \equiv m^{ed} \equiv m^{\phi(n)k+1} \pmod{n}$

Using Euler's theorem above (and also see [4]), we can simplify the previous expression:

$m' \equiv m^{\phi(n)k+1} \equiv mm^{\phi(n)k} \equiv m \pmod{n}$

Thus, we have shown that $m' \equiv m \pmod{n}$, implying $\gamma(\beta(m, k_e), k_d) = m$. □

2. MULTIPLICATIVE LEVEL LINE AND LINE INDICATOR FUNCTION

We introduce a variant of RSA called generalized RSA, based on the multiplicative level lines [7] of integers modulo n. Consider the function $(u, v) \mapsto u \cdot v \pmod{n}$, where n is a non-zero integer. Let a be an integer modulo n [5], and we seek to determine the solutions to the equation:

(1) $$u \cdot v \equiv a \pmod{n}$$

Define the set $L(a) = \{(u, v); u \cdot v \equiv a \pmod{n}\}$, representing the set of solution pairs to equation (1). This set is also called the multiplicative level line of a (see [7]).

Proposition 1. For any integer a modulo n, the set $L(a)$ is non-empty.

Proof. To prove that $L(a)$ is non-empty for any integer a modulo n, we need to show that there exists at least one pair (u, v) of integers such that $u \cdot v \equiv a \pmod{n}$.

Case 1: $a = 0$. Start with the special case where $a \equiv 0 \pmod{n}$. The equation becomes: $u \cdot v \equiv 0 \pmod{n}$. This means $u \cdot v$ is divisible by n. Choose $u = 0$. Then, for any integer v, we have: $0 \cdot v \equiv 0 \pmod{n}$.

Thus, $(0, v) \in L(0)$ for any integer v. Therefore, $L(0)$ is non-empty.

Case 2: $a \neq 0$. We need to find u and v such that: $u \cdot v \equiv a \pmod{n}$.

Fix v to an arbitrary value, for example, $v = 1$. Then we seek u such that: $u \cdot 1 \equiv a \pmod{n}$.

This reduces to: $u \equiv a \pmod{n}$.

Since a is an integer modulo n, there exists at least one integer u satisfying this equation, namely $u = a$. Thus: $a \cdot 1 \equiv a \pmod{n}$

To complete the proof, consider any integer a modulo n. By Bézout's theorem [5], for any integers a and n, there exist integers u and v such that: $\gcd(a, n) = au + nv$.

If $\gcd(a, n) = d$, then there are solutions to the Diophantine equation [5]: $au + nv = d$.

In particular, when $d = 1$, a is coprime to n, and the equation becomes: $au \equiv 1 \pmod{n}$.

This implies that there exists a multiplicative inverse of a modulo n, denoted a^{-1}, such that: $a \cdot a^{-1} \equiv 1 \pmod{n}$.

Choose $u = a$ and $v = 1$. Then: $a \cdot 1 \equiv a \pmod{n}$.

This proves that $(a, 1) \in L(a)$, so $L(a)$ is non-empty for any integer a modulo n. □

Having established that $L(a)$ is non-empty for any integer a modulo n, we can now examine a symmetric property of this set.

Proposition 2. For any integer a modulo n, if the pair (u, v) belongs to $L(a)$, then its symmetric pair (v, u) also belongs to $L(a)$.

Proof. This follows from the commutative property of multiplication in modular arithmetic (see [5]). □

Having established the non-emptiness of $L(a)$ for any integer a modulo n and the symmetry of this set with respect to its components, we can introduce a crucial arithmetic function associated with this set. We define the determinant of $L(a)$ as an arithmetic function called the line indicator, denoted φ, defined in [7] by: $\varphi(a) = \det(L(a)) = |L(a)|$.

Proposition 3. The line indicator function φ is strictly positive for any integer a modulo n.

Proof. This follows from the non-emptiness of $L(a)$. □

Since we have established the strict positivity of the line indicator function φ for any integer a modulo n, we can now explore an essential property of this function: its multiplicativity under certain conditions.

Proposition 4. The line indicator function φ is multiplicative when $\gcd(m, n) = 1$ for two integers m and n. Specifically, for an integer a modulo mn, we have:

(2) $$\varphi(a) = \varphi(a_m) \cdot \varphi(a_n)$$

where a_m and a_n are the reductions of a modulo m and n respectively.

Proof. Consider two integers m and n such that $\gcd(m, n) = 1$. Let a be an integer modulo mn. Define the level line sets:

$L(a) = \{(u, v) : u \cdot v \equiv a \pmod{mn}\}$

$L(a_m) = \{(u, v) : u \cdot v \equiv a_m \pmod{m}\}$

$L(a_n) = \{(u, v) : u \cdot v \equiv a_n \pmod{n}\}$

where $a_m \equiv a \pmod{m}$ and $a_n \equiv a \pmod{n}$.

The Chinese remainder theorem ([5], [13], [14]) ensures that solutions modulo mn can be decomposed into solutions modulo m and n using the respective reductions a_m and a_n (see [7]).

Let (u,v) be a solution to $u \cdot v \equiv a \pmod{mn}$. Then there exist solutions (u_m, v_m) and (u_n, v_n) such that: $u \equiv u_m \pmod{m}$, $v \equiv v_m \pmod{m}$ and $u_m \cdot v_m \equiv a_m \pmod{m}$ with $u \equiv u_n \pmod{n}$, $v \equiv v_n \pmod{n}$ and $u_n \cdot v_n \equiv a_n \pmod{n}$.

The solutions modulo mn are obtained by combining the solutions modulo m and n. More formally, each pair $(u_m, v_m) \in L(a_m)$ and $(u_n, v_n) \in L(a_n)$ gives a solution $(u,v) \in L(a)$ using the isomorphisms of the Chinese remainder theorem.

The total number of solutions modulo mn is thus the product of the number of solutions modulo m and n.

Since $\varphi(a)$ counts the number of pairs (u,v) in $L(a)$, we have [7]: $\varphi(a) = \det(L(a)) = |L(a)|$.

According to the multiplicativity of solutions induced by the Chinese remainder theorem, we obtain:

$$\varphi(a) = \varphi(a_m) \cdot \varphi(a_n)$$

for any integer a modulo mn. □

Having demonstrated the multiplicativity of the indicator function φ under certain conditions, we will now express $\varphi(a)$ explicitly using the prime factorization of n.

Theorem 3. Let n be an integer decomposed into prime factors $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$. Then, for any integer a modulo n, the indicator function $\varphi(a)$ can be expressed using the following formula:

(3)

$$\varphi(a) = \prod_{i=1}^{r} \left[\left(1 + \log_{p_i}\left(\gcd\left(a, p_i^{\alpha_i}\right)\right)\right) p_i - \left(\xi_1 \cdot \alpha_i + \xi_2 \left(1 + \log_{p_i}\left(\gcd\left(a, p_i^{\alpha_i}\right)\right)\right)\right)\right] p_i^{\alpha_i - 1}$$

where the parameters ξ_1 and ξ_2 are conditionally defined as follows:

- $\xi_1 = 1$ and $\xi_2 = 0$ if and only if $\gcd(a, p_i^{\alpha_i}) > 1$ et $r > 1$,
- $\xi_1 = 0$ and $\xi_2 = 1$ otherwise.

We will not provide a proof for the result stated above, and refer the reader to [7]. However, we note that the proof of the primality result between integers is based on three key ideas:

1. The multiplicative group $U(n)$, the set of units modulo n, decomposes as a direct product of the multiplicative groups $U(p_i^{\alpha_i})$. For each prime factor $p_i^{\alpha_i}$, the indicator function $\varphi(a)$ is influenced by $\gcd(a, p_i^{\alpha_i})$;
2. The function $\log_{p_i}(\gcd(a, p_i^{\alpha_i}))$ provides a logarithmic measure of the divisibility of a with each $p_i^{\alpha_i}$. These results were first observed in [8];
3. It is possible to write weak formulations of the parameters ξ_1 and ξ_2 to adjust for specific cases of divisibility.

These methods are not only useful in formulating the expression for φ, but also provide insights into many number theory models, such as the order function of proper non-trivial subgroups of the additive group $\mathbb{Z}_n$, and we refer the reader to [8] for further details. Finally, we note that, in this development, we can intuitively understand that $\varphi(a)$ strongly depends on how a interacts with each prime factor of n. If a shares a prime factor with n, it reduces the number of integers that are coprime to n and can be multiplied with a to produce a result in the same equivalence class of $U(n)$, and vice versa.

To illustrate this relationship in more detail, we explore a specific case via the following corollary.

Corollary 1. Let n be an integer decomposed as $n = p^{\alpha}$. Then, the indicator function is given by:

(3') $$\varphi(a) = (1 + \log_p(\gcd(a, p^{\alpha}))(p-1)p^{\alpha-1}$$

Proof. The proof is straightforward since $r = 1$, we have $\xi_1 = 0$ and $\xi_2 = 1$. Hence, we easily obtain the given expression.

In detail, let $d = \gcd(a, p^{\alpha})$. Since d divides a, we can write $a = dk$ for some integer k. Since d also divides p^{α}, we can write $p^{\alpha} = dk'$ for some integer k'. Note that $\gcd(k, k') = 1$, otherwise d would not be the greatest common divisor of a and p^{α}.

Now, let u and v be integers modulo n such that $u \equiv k \pmod{n}$ and $v \equiv k' \pmod{n}$. For each integer a modulo n in $L(a)$, we have $u \cdot v \equiv k \cdot k' \equiv a \pmod{n}$. Thus, for each integer a modulo n, we have $\gcd(a, p^{\alpha}) = d$. Therefore, the number of distinct elements in $L(a)$ is $1 + \log_p d$, because there are distinct integers of the form $1, p, \ldots, p^{\log_p d}$ that are all divisors of d [8].

Multiplying this number by $(p-1)p^{\alpha-1}$, which gives the order of the multiplicative group $\mathbb{Z}_{p^{\alpha}}^{*}$, we obtain $(1 + \log_p d)(p-1)p^{\alpha-1}$, which is the value of $\varphi(a)$ for $n = p^{\alpha}$. □

By observing the particular structure of the previous corollary, we can deepen our understanding by examining how the indicator function behaves when compared to Euler's totient function for integers coprime to n. This leads us to the following proposition.

Proposition 5. Let a be an integer modulo n. If a and n are coprime, the indicator function is equal to Euler's totient function.

Proof. Let a be an integer modulo n. We will show that when a is coprime to n, the indicator function $\varphi(a)$ is equal to Euler's totient function $\phi(n)$.

Suppose a is coprime to n. This means that $\gcd(a,n) = 1$. Now, consider an element (u,v) in $L(a)$. By the definition of $L(a)$, this pair satisfies the equation $u \cdot v \equiv a \pmod{n}$. Since a is coprime to n, we will prove that $\gcd(u,n) = \gcd(v,n) = 1$.

Consider an integer u modulo n. If $\gcd(u,n) = d > 1$, it would mean that u and n are not coprime, which would contradict the fact that a is coprime to n. Therefore, we must have $\gcd(u,n) = 1$. Similarly, we can show that $\gcd(v,n) = 1$.

As a result, each pair (u,v) in $L(a)$ is also a pair of integers coprime to n, and they are all distinct. Therefore, the number of elements in $L(a)$, i.e., is equal to $\varphi(a)$, which is the number of positive integers less than or equal to n that are coprime to n. □

With this proposition, we establish a fundamental link between the indicator function and Euler's totient function, showing that $\varphi(a)$ aligns perfectly with $\phi(n)$ when a and n are coprime. We can now generalize this observation to a broader framework with the following theorem.

Theorem 4. The indicator function generalizes Euler's totient function for any integer a modulo n.

Proof. Let φ be the indicator function and ϕ the Euler's totient function. We will show that φ is a generalization of ϕ.

First, consider $n = p^{\alpha}$ and $\alpha > 1$, We want to prove that there exists an integer k such that $\varphi(a) = k \cdot \phi(n)$.

We know that for $n = p^{\alpha}$, Euler's totient function is given by:

$$\phi(n) = (p-1)p^{\alpha-1}$$

and the indicator function by:

$$\varphi(a) = \left(1 + \log_p\left(\gcd(a, p^{\alpha})\right)\right)(p-1)p^{\alpha-1}$$

Next, consider $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$. We want to show that $\varphi(a)$ contains $\phi(n)$. For this, note that it suffices to take $\xi_1 = 0$ and $\xi_2 = 1$ for all $i = 1, \dots, r$. Since $r \geq 1$, this is possible only if $\gcd(a, p_i^{\alpha_i}) = 1$ for all $i = 1, \dots, r$.

In this case, the indicator function $\varphi(a)$ simplifies to:

$$\varphi(a) = \prod_{i}^{r}(p_i - 1)p_i^{\alpha_i - 1} = \phi(n)$$

which shows that $\varphi(a)$ naturally generalizes Euler's totient function $\phi(n)$ for any integer a modulo n. □

Thus, we see that the indicator function φ not only includes ϕ as a special case, but it also extends it by incorporating the arithmetic properties of a and n in more general cases. Euler's totient function, used in the RSA algorithm, finds its counterpart in the indicator function. This correspondence opens new perspectives in the development of cryptographic theories.

3. STRUCTURE OF THE GENERALIZED RSA

Definition 1: Let $g \in \mathbb{Z}$. We say that g is a generator of the RSA encryption system with modulus n if: i) $g \neq 0$, ii) $\gcd(g,n) \geq 1$.

Definition 2: Let $\sigma \in \mathbb{N} - \{0\}$. We will call σ-generator in the RSA encryption system with modulus n, the set: $\{g \in \mathbb{Z}: \gcd(g,n) = \sigma\}$.

Remark: In the Generalized RSA encryption system with modulus $n = pq$, we distinguish four types of σ-generators of the cryptographic configurations, which are: 1-generator, p-generator, q-generator, and pq-generator.

Definition 3: Let g be the generator. Considering $n = pq$ as the RSA modulus, the degree of generalization of RSA, denoted δ, is given by: $(1 + \log_p(\gcd(g,p)))(1 + \log_q(\gcd(g,q)))$.

The degree of generalization δ represents a measure of the influence of the generator g on the structure of the RSA encryption system. This concept introduces a complexity scale, where a higher δ indicates greater unpredictability in the key generation process.

Now, let us analyze the degree of generalization and its implications for the structure of the RSA encryption system. We introduce the following theorem, which specifies the conditions under which an RSA encryption system tends toward maximum generalization.

Theorem 5. Let g be the generator. An RSA encryption system with modulus $n = pq$ achieves maximum generalization if there does not exist a nonzero integer g' such that $\delta(g) < \delta(g')$.

Proof. By definition, $\delta(g)$ is a function that depends on the arithmetic structure of the generator g and its interaction with the modulus n. For $n = pq$, where p and q are distinct prime numbers, the choice of generator g influences the way the keys are generated.

Suppose there exists a generator g' such that $\delta(g) < \delta(g')$. This would imply that g' increases the complexity of the system more than g. However, if g is already optimized to maximize δ, such a g' could not exist. Suppose, for

contradiction, that there exists a g' such that $\delta(g) < \delta(g')$. This would imply that g is not optimal, which contradicts our initial assumption that g maximizes generalization. Therefore, $\delta(g)$ must be maximal among all possible generators.

Thus, the generator g is such that it maximizes δ, and no other generator g' can produce greater unpredictability or complexity. □

This theorem lays the foundation for characterizing RSA encryption systems in terms of maximum generalization. By studying particular cases, we can better understand the practical implications of this theory. This leads us to the following corollary, which defines the conditions for a classical RSA encryption system.

Corollary 2. Let g be the generator. An RSA encryption system with modulus n is said to be classical if for all $g' \in \mathbb{Z}$, $\delta(g) \leq \delta(g')$. In other words, when its degree of generalization $\delta(g) = 1$.

This result establishes the conditions under which an RSA encryption system is said to be classical, based on the degree of generalization δ. We now proceed by introducing a new definition that extends the use of the line indicator in the context of RSA moduli.

Remark 2. For an RSA modulus $n = pq$ and a generator g, the line indicator is expressed as follows:

(4)

$$\varphi(g) = (1 + \log_p(\gcd(p, g)))(q - 1)\ (1 + \log_q(\gcd(q, g)))(p - 1)$$

This definition introduces a detailed form of the line indicator for generators in the context of RSA moduli. We can now establish a crucial theorem that links this indicator to Euler's totient function.

Lemma 1. In an RSA encryption system with modulus n and generator g, the relation between the line indicator and Euler's totient function is given by:

(5)

$$\frac{\varphi(g)}{\phi(n)} = \delta$$

This result provides an understanding of the relationship between the line indicator and Euler's totient function in the RSA context, directly leading to the following corollary that specifies the calculation of the degree of generalization.

Corollary 3. In an RSA encryption system with modulus $n = pq$ and generator g, the degree of generalization is formulated as:

(6) $\delta(g) = (1 + \log_p(\gcd(p, g)))(1 + \log_q(\gcd(q, g)))$

This corollary highlights how the degree of generalization is calculated directly from the arithmetic properties of the generator g. We proceed to a result that explores the conditions for maximum generalization.

Proposition 6. An RSA encryption system with modulus $n = pq$ and generator g tends toward maximum generalization if $\varphi(g) > n$.

Proof. We simply decompose $\varphi(g)$. Indeed:

$$\varphi(g) = \left(1 + \log_p(\gcd(p, g))\right)(q - 1)$$

$$\left(1 + \log_q(\gcd(q, g))\right)(p - 1)$$

$$= \delta \cdot (q - 1)(p - 1)$$

Note that if the degree of generalization is maximal, then $\delta > 1$. In other words, by definition, there does not exist a $g' \in \mathbb{Z}$ such that $\delta(g) < \delta(g')$.

Therefore, δ is the largest factor of $(q - 1)(p - 1)$, large enough such that $\delta(q - 1)(p - 1)$ exceeds pq. which implies that $\varphi(g) > n$. □

Having established the result of generalization in terms of the generator g, let us introduce the invertibility transform κ, which is a strictly positive function defined by the expression:

(7)

$$\kappa(u) = \left[pq + \log_q(\gcd(q, u))(p - u) - u + \log_p(\gcd(p, u))(q - u)\right]$$

for any nonzero integer u.

This function is essential for understanding algebraic properties related to the invertibility of elements in the RSA encryption system structure.

Proposition 7. For any generator g in the RSA encryption system with modulus $n = pq$, there exists a constant $\kappa(g)$ that is coprime with n.

Proof. Suppose $\kappa(g)$ is an integer and prove that it exists for any generator g. Since g is a nonzero integer and $n = pq$ where p and q are distinct primes, we examine the terms involved in $\kappa(g)$.

The terms $(p - g)$ and $(q - g)$ are integers, and the multiplication of these terms with $\log_q(\gcd(q, g))$ and $\log_p(\gcd(p, g))$ is always an integer for any value of g. Hence, $\kappa(g)$ is an integer. This guarantees its existence for any generator g.

We now prove that $\kappa(g)$ is coprime with n, i.e., $\gcd(\kappa(g), n) = 1$. Using Euler's theorem, which states that if $\gcd(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$, we conclude that $\kappa(g)$ is indeed coprime with n for any generator $g \in \mathbb{Z}$, □

regardless of its relationship with $n = pq$.

Following the study of the invertibility transform κ and its properties in the RSA encryption system, we can deepen our analysis by focusing on the necessary criteria for encryption and decryption exponents. The next result aims to establish fundamental propositions regarding these exponents in a context where the modulus n is decomposed as pq and the generator g.

Lemma 2. In an RSA encryption system with modulus $n = pq$ and generator g, a strictly positive integer e is an encryption exponent if and only if: i) $e < \varphi(g)$ and ii) $\gcd(e, \varphi(g)) = \gcd(\kappa(g), n) = 1$.

Having established the necessary criteria for an integer to be an encryption exponent, we now turn to the analysis of the decryption exponent, which is essential in the structure of the RSA encryption system.

Lemma 3. Let $n = pq$ be the modulus and g the generator in an RSA encryption system. The positive integer d is a decryption exponent if and only if there exists $d = e^{-1} \pmod{\varphi(g)}$.

These lemmas on encryption and decryption exponents highlight the importance of the line indicator function $\varphi(g)$ and the invertibility transform κ within the structure of the RSA encryption system.

5. GENERALIZED RSA ALGORITHM

The Generalized RSA Algorithm introduces a significant enhancement to the key generation process while keeping the encryption and decryption procedures of the classical RSA intact [11]. The encryption and decryption steps remain unchanged, preserving compatibility with existing applications.

The improvement primarily lies in the introduction of additional parameters during key generation. The use of a generator g and its invertible transform $\kappa(g)$ provides increased complexity, making attacks more difficult without altering the simplicity of the encryption and decryption processes.

Thus, the implementation of the Generalized RSA Algorithm can be performed without requiring major changes to existing infrastructures. The following algorithm outlines the key generation according to the generalized RSA model, ensuring the creation of more robust public and private keys.

Input: Two prime numbers p, q and a random non-zero number g, the generator.

Output: A public key k_e and its corresponding private key k_d.

1. Compute $n = p \times q$ (the modulus).
2. Compute
$$\varphi(g) = \left(1 + \log_p\left(\gcd(p, g)\right)\right) \times \left(1 + \log_q\left(\gcd(q, g)\right)\right) \times (p - 1) \times (q - 1)$$
.
3. Choose an integer e such that $1 < e < \varphi(g)$ and $\gcd\left(e, \varphi(g)\right) = 1$.
4. Compute d such that $d = e^{-1} \pmod{\varphi(g)}$ (the modular inverse of e).

This algorithm ensures the generation of secure keys by leveraging the generalized RSA model. Additionally, it offers a more robust and complex approach, particularly suited for modern computational environments [10] where security remains a primary concern.

The key element of this advancement lies in replacing Euler's totient function with the line indicator function, which opens new perspectives in the RSA encryption system by providing significantly larger encryption and decryption exponents, thereby increasing the complexity of any attack attempt.

Although the key pairs generated by the Generalized RSA Algorithm, k_e for the public key and k_d for the corresponding private key, remain at the core of the system, they are now complemented by additional parameters.

The Generalized RSA offers several significant advantages over its classical counterpart, thereby enhancing its relevance in contexts where computer security is critical. Notable advantages include:

- Increased attack complexity: Cryptographic attacks have become more sophisticated, requiring appropriate countermeasures. Generalized RSA aims to strengthen security against these new threats.
- Evolution of computation: The computational power of computers has significantly increased since the original design of RSA ([1], [9]). The generalization aims to ensure security even in the face of more powerful computing technologies.
- Uncertainty in parameters: The introduction of the generator g creates uncertainty in the predictability of the generated keys. Attackers are faced with a multitude of potential combinations, making their attempts to compromise the system more uncertain.
- Larger exponents: With the line indicator function, it is possible to have encryption and decryption exponents far greater than the RSA modulus n. This makes prime factorization-based attacks even more difficult.

- New challenges for attackers: The benefits introduced by the line indicator imply that even if an attacker succeeds in finding the prime numbers p and q of the RSA modulus n [1], probing the decryption exponent d becomes extremely complex. Traditional prime factorization-based attacks are no longer sufficient, necessitating new methods and a deeper understanding of the function φ.
- Adaptation to cryptographic advancements: By incorporating the function φ, this approach offers adaptation to new advancements in cryptography [9], which is crucial in a context where attack techniques are constantly evolving.
- Compatibility with existing standards: Although based on a generalized approach, this method remains compatible with existing RSA standards [9], making it easier to integrate into current systems.

Example in MATLAB:

In MATLAB, the RSA key generation process can be implemented with the following code:

```
function [k_e, k_d] = generalizedRSA(p, q, g)
    % Compute Euler's totient function for g
    phi_g = phi(p,q,g);
    % Select a random number e such that 1 < e < phi(g)
    e = randi([2, phi_g-1]);
    while gcd(e, phi_g) ~= 1
        e = randi([2, phi_g-1]);  % Repeat if gcd(e, phi(g)) != 1
    end
    % Compute d such that d = e^(-1) mod phi(g)
    d = modInverse(e, phi_g);
    % Public key is (e, n), private key is (d, n)
    n = p * q;  % RSA modulus
    k_e = [e, n];  % Public key
    k_d = [d, n];  % Private key
end
function inv = modInverse(a, m)
    % Function to find modular inverse of a mod m
    [gcd_val, x, ~] = gcd(a, m);
    if gcd_val == 1
        inv = mod(x, m);  % Return modular inverse
    else
        error('Inverse does not exist');
    end
end
function phi_val = phi(p,q,g)
    % Function to compute the indicator of g
    phi_val = (1 + log(gcd(p, g)) / log(p))*(1 + log(gcd(q, g)) / log(q))*(q-1)*(p-1);
end
```

The following examples illustrate the practical application of both classical and generalized RSA algorithms with specific numerical values for p, q and g, along with the generated public and private keys:

Example 1: classical RSA

Input values:

- $p = 13$
- $q = 17$

Steps:

1. Compute $n = 13 \times 17 = 221$.
2. Compute $\phi(n) = (13 - 1) \times (17 - 1) = 192$.
3. Choose $e = 5$ $(\gcd(5, 192) = 1)$.
4. Compute d such that $d \equiv 5^{-1} (\mathrm{mod}\ 192)$. Using the extended Euclidean algorithm, we find $d = 77$.

Keys:

- Public key: $(e, n) = (5, 221)$
- Private key: $(d, n) = (77, 221)$

Example 2: generalized RSA (with $\delta = 2$)

Input values:

- $p = 13$
- $q = 17$
- $g = -234$

Steps:

1. Compute $n = 13 \times 17 = 221$.
2. Compute
$\varphi(g) = (1 + \log_{13} \gcd(13, -234)) \times (1 + \log_{17} \gcd(17, -234)) \times (13 - 1) \times (17 - 1) = 384$
.
3. Choose $e = 83$ $(\gcd(83, 384) = 1)$.
4. Compute d such that $d \equiv 83^{-1} (\mathrm{mod}\ 384)$. We find $d = 347$.

Keys:

- Public key: $(e, n) = (83, 221)$
- Private key: $(d, n) = (347, 221)$

Example 3: generalized RSA (with $\delta = 4$)

Input values:

- $p = 13$
- $q = 17$

- $g = 663$

Steps:

1. Compute $n = 13 \times 17 = 221$.
2. Compute $\varphi(g) = (1 + \log_{13} \gcd(13, 663)) \times (1 + \log_{17} \gcd(17, 663)) \times (13 - 1) \times (17 - 1) = 768$.
3. Choose $e = 317$ ($\gcd(317, 768) = 1$).
4. Compute d such that $d \equiv 317^{-1} (\mathrm{mod}\ 768)$. We find $d = 533$.

Keys:

- Public key: $(e, n) = (317, 221)$

Private key: $(d, n) = (533, 221)$

Example 4: generalized RSA (with $\delta = 4$)

Input values:

- $p = 13$
- $q = 17$
- $g = 442$

Steps:

1. Compute $n = 13 \times 17 = 221$.
2. Compute $\varphi(g) = (1 + \log_{13} \gcd(13, 442)) \times (1 + \log_{17} \gcd(17, 442)) \times (13 - 1) \times (17 - 1) = 768$.
3. Choose $e = 5$ ($\gcd(5, 768) = 1$).
4. Compute d such that $d \equiv 5^{-1} (\mathrm{mod}\ 768)$. We find $d = 461$.

Keys:

- Public key: $(e, n) = (5, 221)$
- Private key: $(d, n) = (461, 221)$

The examples reveal an important distinction between the classical and generalized RSA algorithms. In the classical RSA, both the encryption exponent e and the decryption exponent d are typically constrained to values less than n. However, in the generalized RSA, it is evident that d can exceed n, as shown in the second, third and fourth examples, and e can also be larger than n (as in the third example). Focusing on examples 1 and 4, we can draw a direct comparison that highlights the advantages of the generalized RSA over the classical RSA. Both examples share the same modulus $n = 221$ and use the same encryption exponent $e = 5$. However, the method of generating the decryption exponent d differs significantly between the classical and generalized versions.

This feature demonstrates its potential for increased complexity and security by introducing additional parameters, making it more difficult for attackers to apply standard factorization and decryption techniques. The capacity to work with larger values of e and d without compromising the encryption and decryption processes highlights a significant advantage in cryptographic resilience, particularly in scenarios where more robust key structures are necessary.

REFERENCES

[1] D. Boneh. Twenty Years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2), 203-213, 1999.

[2] J. Buchmann. *Introduction to Cryptography* (2nd ed.). Springer, 2004. ISBN: 978-0387202293

[3] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644-654, 1976. doi:10.1109/TIT.1976.1055638

[4] S. D. Galbraith and P. Gaudry. The Mathematics of public-key cryptography. *Notices of the AMS*, 63(6), 691-697, 2016.

[5] J. Katz and Y. Lindell. *Introduction to Modern Cryptography* (3rd ed.). CRC Press. ISBN: 978-0367331581, 2020.

[6] N. Koblitz and A. Menezes. The random oracle model: a twenty-year retrospective. *Designs, Codes and Cryptography*, 77(2), 569-599, 2010.

[7] S. Kumpovela and C. Paluku. On the level lines of primary functions of two variables defined on the ring $\mathbb{Z}_n$. *Interdisciplinary Research Journal of the National Pedagogical University*, 93(4), 75-81, 2022.

[8] S. Kumpovela. Two or three things about additive groups $\mathbb{Z}_n$. *Interdisciplinary Research Journal of the National Pedagogical University*, 90(2), 75-83, 2022.

[9] A. Menezes, P. C. van Oorschot and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN: 978-0849385230

[10] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In *Mathematical Algorithms and Complexity*, 89-139. Academic Press, 1982.

[11] R. L. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120-126, 1978. doi:10.1145/359340.359342

[12] A. Shamir. Cryptography in an algebraic eraser. *Communications in Algebra*, 11(13), 1473-1483, 1963.

[13] D. R. Stinson and M. B. Paterson. *Cryptography: Theory and Practice* (4th ed.). CRC Press, 2018. ISBN: 978-1138197015

[14] W. Trappe and L. C. Washington. *Introduction to Cryptography with Coding Theory*. Pearson Prentice Hall, 2006.