Deciphering[1] RSA Encryption

HCHS Radicals, Spring 2024 Edition

By **Dexter Theisen**, 9th grade

I. History: What is RSA encryption, and why was it invented?

In the early 1950s, mathematical education in the United States transitioned from a purely rote system to a more practical one that favored understanding concepts and applying them to the real world. The US math community's expansion, partly inspired by the technological advancements of World War II, led to growth in fields such as cryptography, which had important military and commercial applications. During this era (the late 20th century), some of the best number theorists and cryptanalysts worked together and challenged each other to develop new systems of encryption that were efficient and secure. Over the subsequent decades, the most prevalent system of encryption became using *keys*[2] to encode and decode every message sent. Only the intended senders and recipients would have this unique key, making the data sent useless to any third-party observer. The fundamental issue with this system was the secure establishment of keys. An encrypted key could not be sent because the recipient would need that key to decrypt the message, and a *plaintext*[3] key could not be sent due to the possibility of it being intercepted by an unintended bad actor and used to decrypt the following messages from the same sender (without their knowledge). The most secure solution at the time was an in-person exchange of keys, which, of course, was severely limiting or even at times self-defeating.

This issue gave rise to a new way of encryption called RSA, an *asymmetric*[4] encryption algorithm created in 1977, that is now used globally. The system was developed by Ron Rivest, Adi Shamir, and Leonard Adleman (hence RSA – their last initials), and relies on two asymmetric but linked keys: a public key used to encrypt and a private key to decrypt. Their method was inspired by a 1976 paper co-authored by Whitfield Diffie and Martin Hellman, in which they outlined an idea for a public-key cryptosystem. The RSA inventors noted that “Diffie and Hellman’s article motivated [their] research, since they presented the concept but not any

---

[1] To be clear, deciphering refers to using a cipher (which shifts letters and maintains ratios) to decode a message, whereas decrypting refers to using an algorithm (which may not maintain the ratio of letters) to decode a message. The verb, decode, can simply be used in reference to simplifying or reducing a previously obfuscated message into understandable text. I chose to use the word “deciphering,” although technically incorrect in this context, in the title, as I felt its synonymy with the word “understanding” better fit the theme of this article.

[2] A key, in cryptography, is a string of numbers or characters used within an encryption algorithm to alter data so that it appears random.

[3] Plaintext, in cryptography, refers to an unencrypted message that can be read without a decryption key or device.

[4] Asymmetric encryption is a cryptographic system that uses a pair of keys – one public and one private – to secure communication. The public key is used to encrypt data or messages, while the private key is used to decrypt them. This type of system, as the article explains, allows for secure communication between parties without the need to share a secret key.

practical implementation of such a system," leaving the challenge up to Rivest, Shamir, and Adleman.

A public-private key system works as follows: each party has its own private key (consisting of two large primes), which no one else has. A public key is generated by multiplying those two large primes together. The public key can be used to encrypt a message, but the original two primes are required to decrypt the message. RSA encryption is considered secure because it utilizes very large prime numbers (around 30 digits long), making it extremely difficult to factor the public number. Guessing the original two primes used, on average, takes millions of years. Because of its high level of security, RSA is now used extensively in various important functions, such as transmitting military intelligence and safeguarding financial information.

II. Framework: How does RSA encryption work?

As explained above, RSA encryption is centered around two prime numbers, commonly referred to as p and q.

Every person has a public key in the form of two variables, (n, e),

and a private key, in the form of two variables, (p, q).

To demonstrate the full encryption and decryption mechanism, I will assign values to every to demonstrate the full encryption and decryption mechanism variable below. The corresponding justifications for each step will be explained and proved after the example.

We shall start by letting $p = 11$ and $q = 19$.
We shall define a larger number, called n, which is equal to $p * q$.
Therefore $n = 209$ and n is a ***semiprime***[5].

We also define a ***totient***[6], called phi, which is equal to $(p - 1) * (q - 1)$ as given by ***Euler's Totient Theorem***[7].
Therefore $phi = 180$.

Finally, we shall define an integer e, ensuring $1 <= e < phi$ and e is ***coprime***[8] to phi.

[5] A semiprime is a natural number that is the product of exactly two prime numbers.
[6] A totient refers to the number of relatively prime positive integers less than a specified integer
[7] Wikipedia. 2024. "Euler's Totient Function." Wikimedia Foundation. Last modified May 12, 2024. https://en.wikipedia.org/wiki/Euler%27s_totient_function.
[8] Two numbers are considered coprime if they no share no common factors (besides 1).

III. Example - Part 1: Encryption using the public key

To encrypt a message M, we first turn M into a string of numbers. For the purpose of this article, we will define M as the letter “D” – equivalent to the number 4 (given by D’s position in the English alphabet).

To encrypt a message, we first raise M to the power of e and then apply a modulus of n (to prevent our encrypted message from becoming too large).
For instance, if we let $e = 7$, our encrypted message, M', will be: $4^7 \equiv 82 \pmod{209}$.

So for the rest of our demo, we will use the encrypted message: 82; being careful to remember our original message as: 4.

IV. Example - Part 2: Decryption using the private key

To decrypt our message, we have to find one more number, d.

We let d be equal to the *modular multiplicative inverse*[9] of $e \pmod{phi}$.
The modular multiplicative inverse of e under phi is defined as the number, d, such that $d * e \equiv 1 \pmod{phi}$.
To calculate this inverse, we invoke *Euler’s Theorem*[10], which states $a^{\phi(n)} \equiv 1 \pmod{n}$ where $\phi(n)$ is the totient of n.

Thus, $d = e^{\phi(phi)-1}$ to ensure $d * e \equiv 1 \pmod{phi}$. (This calculation takes phi to be n in the equation introduced as Euler’s Theorem).

In our example, $d \equiv e^{\phi(phi)-1} \equiv 7^{\phi(180)-1} \equiv 7^{48-1} \equiv 103 \pmod{phi}$

Testing our d value: $(7 * 103) \equiv 1 \pmod{180}$ shows that $d * e \equiv 1 \pmod{phi}$ holds. Thus, d is the modular multiplicative inverse of $e \pmod{phi}$.

Now, to decipher our encrypted message, M', we raise M' to the power of d and apply a modulus of n. So: $82^{103} \equiv 4 \pmod{209}$.
And as expected, we get 4, our original message.

---

[9] A modular multiplicative inverse of an integer, a, is the integer x such that the product a*x is congruent to 1 with respect to the modulus m.
Wikipedia. 2024. "Modular Multiplicative Inverse." Wikimedia Foundation. Last modified January 26, 2024. https://en.wikipedia.org/wiki/Modular_multiplicative_inverse.
[10] Wikipedia. 2024. "Euler's Theorem." Wikimedia Foundation. Last modified April 9, 2024. https://en.wikipedia.org/wiki/Euler%27s_theorem.

V. Proof: Why does this work?

Proving the mathematical rigor of RSA encryption will take several steps.

Given that e is the encrypting exponent, and d is the decrypting exponent, we are trying to prove that $(M^e \pmod n)^d \pmod n \equiv M$.

To express this more succinctly, we shall define the equation we are trying to prove as S, where S is explicitly $M^{e^d} \equiv M \pmod n$.

By the definition of the modular multiplicative inverse used to calculate d, we know that:
$e * d \equiv 1 \pmod{phi}$.

We also know:
$phi = (p-1) * (q-1)$, meaning:
$e * d = 1 + k * (p-1) * (q-1)$, where $k \in \mathbb{Z}$.

Therefore we can rewrite the lefthand side of S as $M^{(p-1)*(q-1)*(k)+1}$, which for the sake of repetition, we will denote also as the variable X.

Because n is the product of two relatively prime numbers, p and q (primes themselves), we will use the *Chinese Remainder Theorem*[11] (CRT) to prove that:
$X \equiv M \pmod n$ (or $X \equiv M \pmod{p*q}$), by proving:
$X \equiv M \pmod p$ and $X \equiv M \pmod q$.

As given by *Fermat's Little Theorem*[12] (FLT), for any prime r:
$z^{r-1} \equiv 1 \pmod r$.

Now, we shall rewrite the lefthand side of S as:
$(M^{(p-1)})^{(q-1)*k} * M$, which by applying FLT we can reduce to:
$X = (1 \pmod p)^{(q-1)*k} * M$.

To prove $X \equiv M \pmod p$, we start with the equation:
$(1 \pmod p)^{(q-1)*k} * M \equiv M \pmod p$.

---

[11] Wikipedia. 2024. "Chinese Remainder Theorem." Wikimedia Foundation. Last modified April 12, 2024. https://en.wikipedia.org/wiki/Chinese_remainder_theorem.
[12] Wikipedia. 2024. "Fermat's Little Theorem." Wikimedia Foundation. Last modified May 17, 2024. https://en.wikipedia.org/wiki/Fermat%27s_little_theorem.

Because the entire equation has a modulus of p (and only p) applied to it, we can rewrite the equation (as given by the *Exponential Law of Modular Arithmetic*) as:
$(1^{(q-1)*k}) * M \pmod{p} \equiv M \pmod{p}$, which simplifies to
$1 * M \pmod{p} \equiv M \pmod{p}$, which further reduces to
$M \pmod{p} = M \pmod{p}$.

To prove $X \equiv M \pmod{q}$, we start with the equation:
$(1 \pmod{p})^{(q-1)*k} * M \equiv M \pmod{q}$.

First, we manipulate the equation to look like:
$(1 \pmod{p})^{k})^{(q-1)} * M \equiv M \pmod{q}$.

Treating $(1 \pmod{p})^{k}$ as a single term, we can rewrite the equation (using FLT) as:
$(1 \pmod{q}) * M \equiv M \pmod{q}$.

Because the entire equation has a modulus of q (and only q) applied to it, we can reduce it similarly:
$(1 \pmod{q}) * M \equiv M \pmod{q}$ simplifies to
$1 * M \pmod{q} \equiv M \pmod{q}$, which further reduces to
$M \pmod{q} = M \pmod{q}$.

Therefore (as given by the CRT), because $X \equiv M \pmod{p}$, $X \equiv M \pmod{q}$, and p and q are relatively prime:
$X \equiv M \pmod{n}$.

Because X is merely a simplification of M^{e^d}, we can conclude:
$(M^e \pmod{n})^d \pmod{n} \equiv M \pmod{n}$ – thus proving asymmetric keys can be used (and are mathematically correct) to successfully encrypt and decrypt our message M.

VI. Next Steps: A quantum approach

Like every method of encryption, RSA has weaknesses that can be exploited in order to *crack*[13] the two primes involved in the private key. What sets the encryption methods that we use apart from the ones that have become obsolete is the severity with which the exploits present themselves. For example, the weakness of a *substitution cypher*[14] is that the encrypted text can

[13] To “crack,” cryptographically speaking, means to figure out the keys used to encode a specific message.
[14] A substitution cipher is a method of encryption where each letter in the plaintext is replaced by another letter or symbol according to a specific rule or key (normally a shift of some type).

be run through a *frequency analysis*[15] in order to semi-reliably determine which letters represent the most common English letters. This is why we do not use substitution cyphers to encode sensitive data. RSA's weakness comes in the form of factorizing the public key n, to determine the two original primes p and q. The fastest way to factorize this 30-plus digit long prime is not to guess (as this would take even the most modern supercomputer tens of millions of years to complete) but to instead follow a four-step process:

1. Choose a random number g such that $g < n$ and g is coprime to n (if this is not the case then by some miracle, you have guessed the first, and subsequently the second factor, of n).
2. As given by *Euler's Theorem*[16], there will always exist some r such that (given g and n are coprime) $g^r = n * k + 1$, where $k \in \mathbb{Z}$.
3. If r is even, rewrite the equation above as $g^r - 1 = n * k$, and factor the left side as a difference of two squares, reducing it to: $(g^{r/2} + 1) * (g^{r/2} - 1)$.
4. Use *Euclid's Algorithm*[17] to find the common factors in n and the two terms on the left, which will finally yield p and q.

The issue with this approach, as explained in the first section, is that it will still take a few million years. By far the longest part of this process is step two, taking over 95% of the time for any large n. However, this is where quantum computers come in handy. Without diving into too much detail (as I plan to write a full article on this topic in the next publication), it can simply be said that by harnessing the quantum property of entanglement, step two can be exponentially sped up, rendering RSA encryption (even with 30 digit keys) vulnerable to cracks in around 30 days.

VIII. Concluding Notes: Why is this unique?

RSA encryption was the first asymmetrical system of its time. Created by three (and contributed to by tens of) brilliant minds, it served as a new type of encryption – one that could be reliably established without ever needing to physically meet and exchange keys. This allowed the internet to expand, as strangers online could now securely talk from opposite sides of the globe. As cyber threats continue to evolve, you can now take peace in knowing that strong encryption mechanisms like RSA are safeguarding sensitive information and preserving your privacy…

At least, until we see the rise of quantum computers!

---

[15] A frequency analysis, in cryptography, is the study of the frequency of certain letters or groups of letters in a ciphertext to determine what letters may have been substituted for others.
[16] Wikipedia. 2024. "Euler's Theorem." Wikimedia Foundation. Last modified April 9, 2024. https://en.wikipedia.org/wiki/Euler%27s_theorem.
[17] Wikipedia. 2024. "Euclidean Algorithm." Wikimedia Foundation. Last modified April 9, 2024. https://en.wikipedia.org/wiki/Euclidean_algorithm.