The RSA Adventure — How to Hide Secrets with Math

A friendly example for curious minds

—

The Story Begins

Imagine two friends: **Sophie** and **Tim**. They want to share secret numbers with each other, but anyone watching their messages could intercept them. They need a way to talk safely — even if everyone sees what they send. That's the magic of the **RSA cryptosystem.**

—

How It Works (Intuitively)

RSA is like a special kind of lock. Sophie builds a lock (the *public key*) and gives it to Tim. Only she knows the key to open it (the *private key*).

Tim can put any secret message in the box and lock it with her public key, but once locked, only Sophie can open it again.

Under the hood, this "lock" and "key" are made with a neat number trick: it's easy to multiply two primes, but very hard to figure out what they were if you only see the result.

—

Example 1 — The One-Number Secret

Let's start small.

Step 1. Pick two secret primes

Sophie chooses two small prime numbers:

$$p = 3, \quad q = 11.$$

She multiplies them to make:

$$n = p \times q = 3 \times 11 = 33.$$

She also computes:

$$\phi(n) = (p-1)(q-1) = 2 \times 10 = 20.$$

—

Step 2. Choose a public exponent e

We want e to have no common factors with 20. Let's pick $e = 3$. (She could also have chosen 7, 9, 11, etc., but 3 keeps the math friendly.)

—

Step 3. Compute the private key d

Now we find d such that:

$$e \times d \equiv 1 \pmod{20}.$$

Try some small numbers:

$$3 \times 7 = 21 \equiv 1 \pmod{20}.$$

So $d = 7$.

$$\textbf{Public key: } (n, e) = (33, 3) \quad \text{and} \quad \textbf{Private key: } (n, d) = (33, 7)$$

—

Step 4. Encrypting the message

Tim wants to send Sophie the number $M = 4$. He computes:

$$C = M^e \bmod n = 4^3 \bmod 33 = 64 \bmod 33 = 31.$$

He sends Sophie the number **31**.

—

Step 5. Decrypting the message

Sophie uses her private key to recover it:

$$M = C^d \bmod n = 31^7 \bmod 33.$$

After some modular arithmetic (or a calculator):

$$31^7 \bmod 33 = 4.$$

She gets the original message back!

$$\boxed{\textit{Encrypted}: 31 \quad \longrightarrow \quad \textit{Decrypted}: 4.}$$

—

Example 2 — Sending a Word ("HI")

Now let's level up! We'll send the word **HI**.

We'll keep Sophie's keypair the same:

$$(n, e) = (33, 3), \quad (n, d) = (33, 7).$$

Step 1. Convert letters to numbers

Let's use A=1, B=2, ..., Z=26.

$$H = 8, \quad I = 9.$$

—

Step 2. Encrypt each letter

For each letter M, we compute $C = M^3 \bmod 33$.

$$\begin{aligned} C_H &= 8^3 \bmod 33 = 512 \bmod 33 = 17, \\ C_I &= 9^3 \bmod 33 = 729 \bmod 33 = 3. \end{aligned}$$

So the ciphertext for "HI" is **17, 3**.
—

Step 3. Decrypt each letter

Sophie uses $M = C^7 \bmod 33$:

$$\begin{aligned} M_H &= 17^7 \bmod 33 = 8, \\ M_I &= 3^7 \bmod 33 = 9. \end{aligned}$$

$$\boxed{\text{Decrypted back to ``HI''!}}$$

—

Example 3 — Why RSA Works

Here's the beautiful math behind the curtain: When you encrypt $C = M^e \bmod n$ and then decrypt $M = C^d \bmod n$, you're really computing:

$$M^{ed} \bmod n.$$

And because of how we chose d, we know:

$$ed \equiv 1 \pmod{\phi(n)}.$$

That means:

$$M^{ed} \equiv M^{1+k\phi(n)} \equiv M \times (M^{\phi(n)})^k \bmod n.$$

Euler's Theorem tells us $M^{\phi(n)} \equiv 1 \pmod n$ when M and n share no factors — so everything collapses neatly back to M.

It's like a secret handshake between math and logic.
—

The Takeaway

- RSA is built on the fact that multiplying is easy, but factoring is hard.
- Even small numbers follow the same rules as huge ones.
- Every encryption is a power and every decryption an inverse power.
- Behind the scenes, it's just modular arithmetic with a heroic name.

"Mathematics is the only place where you can multiply two secrets and get a public truth."