Example 13 — A Tiny Toy Model of Public–Private Keys

Goal

To see, step by step, how a public key and private key can work together to lock and unlock information. We'll use numbers so small you can compute them in your head. This is not secure — it's a sandbox for understanding the flow.

Scene 1 — Choosing a "World" (the modulus)

Let's build a tiny arithmetic world where everything wraps around at 33. We'll say all operations are done "mod 33."

That means:

$$a \equiv b \pmod{33} \quad \text{if they differ by a multiple of 33.}$$

So, for instance, $40 \equiv 7 \pmod{33}$ because $40 - 7 = 33$.

Scene 2 — The Secret and the Public Keys

We pick two numbers that work as opposites in this modular world. We want one number e for "encrypt" and one number d for "decrypt," so that doing both in a row brings us back to the original message m.

We need:

$$(m^e)^d \equiv m \pmod{33}.$$

For this to happen, e and d must be inverses with respect to the totient of 33. (We'll skip the deep theory — we'll just choose a pair that works.)

Let's pick:

$$e = 3, \qquad d = 7.$$

We can check that these behave nicely because:

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

and 20 is the totient of 33 (that's a fancy way of saying there are 20 numbers less than 33 that don't share a factor with 33).

Perfect! They are modular inverses.

Private key: $d = 7$ **Public key:** $(e = 3,\ n = 33)$

Scene 3 — Sending a Secret Message

Suppose Bob wants to send Alice the number $m = 4$ (representing a small message).

He uses Alice's public key $(e, n) = (3, 33)$ to encrypt it:

$$c \equiv m^e \pmod{33} = 4^3 \bmod 33.$$

Compute: $4^3 = 64$, and $64 \bmod 33 = 64 - 33 = 31$.

So the ciphertext is:

$$c = 31.$$

Bob sends 31 to Alice.

Scene 4 — Unlocking the Secret

Alice uses her private key $d = 7$ to decrypt:

$$m' \equiv c^d \pmod{33} = 31^7 \bmod 33.$$

That looks nasty, but in modular arithmetic patterns repeat fast. Let's compute powers of

31 mod 33:

$$31^1 \equiv 31, \quad 31^2 \equiv 31 \cdot 31 = 961 \equiv 4, \quad 31^3 \equiv 4 \cdot 31 = 124 \equiv 25, 31^4 \equiv 25 \cdot 31 = 775 \equiv 13, \quad 31^5 \equiv 13 \cdot 31 = 403$$

So:

$$m' = 4.$$

It worked! Alice got back the original message. She never revealed d, and Bob never knew it — he only knew e and n.

Scene 5 — Reversing the Flow (Digital Signature)

Now Alice wants to prove that a message came from her. She uses her **private key first**, and anyone can check it with her **public key**.

She signs $m = 5$ by computing:

$$s \equiv m^d \pmod{33} = 5^7 \bmod 33.$$

Compute $5^7 = 78{,}125$. Divide by 33: $33 \times 2367 = 78{,}111$, remainder 14. So $s = 14$.

Alice sends $(m = 5, s = 14)$.

Verification: Anyone with her public key $(e = 3, n = 33)$ checks:

$$s^e \equiv 14^3 \bmod 33 = 2744 \bmod 33.$$

Compute: $33 \times 83 = 2739$, remainder 5.

So $s^e \equiv 5 \pmod{33}$ — it matches m. The signature checks out!

Scene 6 — What We Learned

- The **public key** lets anyone lock a box that only the private key can open.

- The **private key** can sign something that anyone can verify with the public key.
- The keys are linked by a one-way relationship: d and e are modular inverses mod $\varphi(n)$.
- Real systems like RSA or ED25519 do this same dance — just with gigantic primes and far more sophisticated math.

Takeaway: Encryption and signatures are two directions of the same beautiful math. Public locks, private keys, and modular worlds — all making trust possible in the land of numbers.