The idea behind public key cryptography (RSA)

The following is a description of the "idea" behind RSA security.

Goal: You want to provide a method for anyone in the world to send you an encoded message. Only you have the ability to decode the message.

You:

1. Choose two (huge) primes $p \neq q$.
2. Form the composite number $n = pq$.
3. Form the number $(p-1)(q-1)$ and choose an integer e such that $\gcd(e, (p-1)(q-1)) = 1$.
4. Run the Euclidean Algorithm with back substitution to obtain an integer d such that

$$de \equiv 1 \mod (p-1)(q-1).$$

5. You **publish** the numbers e and n for all to see, and you keep the numbers d, p, q private.

Second Person:

1. Wants to send you an encoded message.
2. Uses some industry standard method of encoding the extended alphabet (keyboard characters) as numbers, so the message becomes a string of numbers

$$m_1, m_2, \ldots$$

where each number is relatively prime to n (guaranteed by the fact that your original primes p and q are much bigger than the industry standard numbers for encoding keyboard characters).

3. **Encodes** each m_i by converting it to the integer

$$x_i \equiv m_i^e \mod n$$

4. Sends you the string of numbers

$$x_1, x_2, \ldots$$

You:

1. Receive the string of numbers

$$x_1, x_2, \ldots$$

2. **Decode** each integer by computing

$$x_i^d \equiv (m_i^e)^d \equiv m_i^{ed} \equiv m_i \mod n$$

since $ed \equiv 1 \mod (p-1)(q-1)$ and $n = pq$.

3. Use your newly obtained sequence of m_i and the industry standard translation to recover the original message.

Third person (with evil intent):

1. Intercepts the communication string
$$x_1, x_2, \ldots$$
2. Looks up your public information n and e.
3. Is stuck, since they don't know the decoder number d.
4. If the third person knew the exact factorization pq of n, then they could find d by using the Euclidean Algorithm with e and $(p-1)(q-1)$. It turns out that the problem of finding the decoder d is equivalent to the problem of finding the factorization pq of n.
5. This last problem (finding prime factorizations of extremely large composite numbers) is notoriously difficult. Note that although the third person knows that n is a composite number and in fact is a product of two huge prime numbers (because this is how RSA is designed) they still can't determine the p and q efficiently (fastest computers working with latest factorization methods may take decades).