

# Cryptography - Part 5

Mar. 25, 2025

# Recap question:

March 25, 2025

Which of the following numbers are prime?

2,3,4,5,6,7,8,9,10,200

For any number that is not prime, write down its prime factorization by finding one prime number that divides it. Then find a prime number that divides the quotient and keep going until the quotient is 1. The prime numbers you find in this way is the factorization.

# Recap question:

March 25, 2025

2,3,5,7 are prime. The prime factorizations of the other numbers are:

$$4 = 2 \times 2$$

$$6 = 2 \times 3$$

$$8 = 2 \times 2 \times 2$$

$$9 = 3 \times 3$$

$$10 = 2 \times 5$$

$$200 = 2 \times 2 \times 2 \times 5 \times 5$$

# Cryptography - Part 5

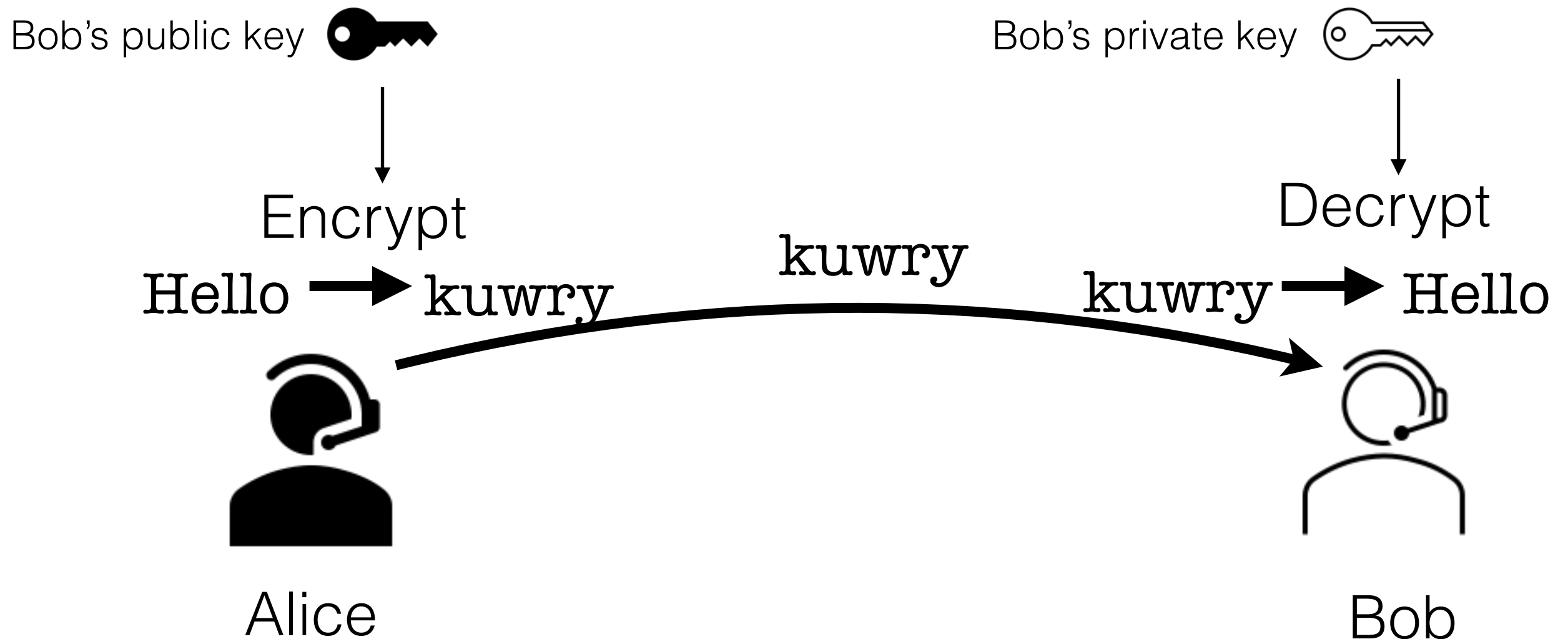
March 25, 2025

By the end of this lecture, you will be able to:

1. Use [the Euclidean algorithm](#) to compute the private key in RSA
2. State and use [Fermat's little theorem](#)

# Asymmetric-key encryption

1. Bob displays his public key on his website.
2. Anyone wishing to send him a message encrypts the message with the public key.
3. Bob keeps his private key secret. Using the private key is the only way to decrypt a message that is encrypted with the public key.



# RSA encryption

Encryption:  $y \equiv x^r \pmod{n}$

Decryption:  $x \equiv y^s \pmod{n}$

How to construct the keys  $r, n$  and  $s$ ?

In contrast to what we have done before, the keys should not be chosen randomly!  $r, n$  and  $s$  are chosen by the following procedure:

1. Choose two **prime numbers**  $p$  and  $q$
2. Choose  $n = p \times q$
3. Choose  $r$  as any number that has no prime factors in common with  $(p - 1) \times (q - 1)$
4. Compute  $s$  from  $p$  and  $q$  by the **Euclidean algorithm**

# The Euclidean algorithm

If  $a$  and  $m$  are any numbers (prime or not) and

$$ab \equiv 1 \pmod{m}$$

we say that  $b$  is a **multiplicative inverse of  $a$  modulo  $m$**

# The Euclidean algorithm

If  $a$  and  $m$  are any numbers (prime or not) and

$$ab \equiv 1 \pmod{m}$$

we say that  $b$  is a **multiplicative inverse of  $a$  modulo  $m$**

**Example:** 2 is the multiplicative inverse of 4 modulo 7, since

$$4 \times 2 = 8 \equiv 1 \pmod{7}.$$



# The Euclidean algorithm

If  $a$  and  $m$  are any numbers (prime or not) and

$$ab \equiv 1 \pmod{m}$$

we say that  $b$  is a **multiplicative inverse of  $a$  modulo  $m$**

Whenever  $a$  and  $m$  have no common factors (i.e., any prime appearing in the prime factorization of  $a$  does not appear in the prime factorization of  $m$ ), then we can always find a multiplicative inverse of  $a$  modulo  $m$  using the Euclidean algorithm.

# The Euclidean algorithm

**Example:** Find the multiplicative inverse of  $a = 20$  modulo  $m$ , for  $m = 63$ , that is, find  $b$  so that

$$20 \times b \equiv 1 \pmod{63}$$

# The Euclidean algorithm

**Example:** Find the multiplicative inverse of  $a = 20$  modulo  $m$ , for  $m = 63$ , that is, find  $b$  so that

$$20 \times b \equiv 1 \pmod{63}$$

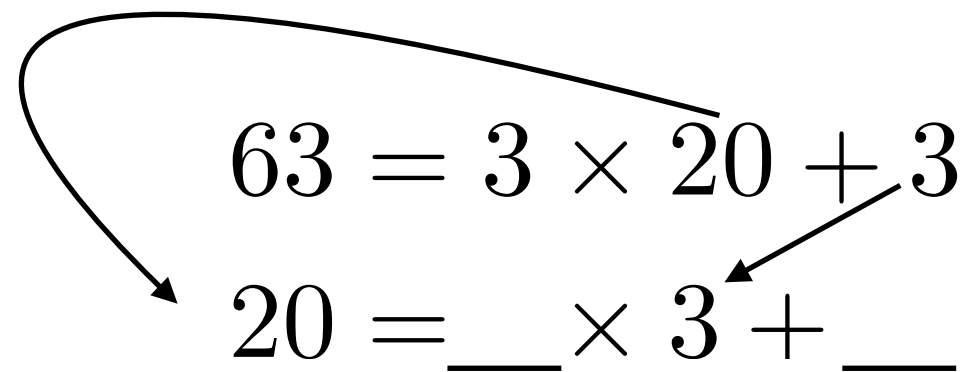
Note that the prime factorizations are  $20 = 2 \times 2 \times 5$  and  $63 = 3 \times 3 \times 7$ , so they have no common factor, and we can always find a multiplicative inverse of 20 modulo 63

**Step 1:** Sort the two elements 20 and 63, with the larger one first. Find the quotient and remainder when dividing the larger number by the smaller number. Then keep doing the same with the resulting divisors and remainders, until the remainder becomes 1.

**Step 1:** Sort the two elements 20 and 63, with the larger one first. Find the quotient and remainder when dividing the larger number by the smaller number. Then keep doing the same with the resulting divisors and remainders, until the remainder becomes 1.

$$63 = 3 \times 20 + 3$$

**Step 1:** Sort the two elements 20 and 63, with the larger one first. Find the quotient and remainder when dividing the larger number by the smaller number. Then keep doing the same with the resulting divisors and remainders, until the remainder becomes 1.


$$\begin{array}{l} 63 = 3 \times 20 + 3 \\ 20 = \_\_ \times 3 + \_\_ \end{array}$$

**Step 1:** Sort the two elements 20 and 63, with the larger one first. Find the quotient and remainder when dividing the larger number by the smaller number. Then keep doing the same with the resulting divisors and remainders, until the remainder becomes 1.

$$63 = 3 \times 20 + 3$$

$$20 = 6 \times 3 + 2$$

**Step 1:** Sort the two elements 20 and 63, with the larger one first. Find the quotient and remainder when dividing the larger number by the smaller number. Then keep doing the same with the resulting divisors and remainders, until the remainder becomes 1.

$$\begin{array}{l} 63 = 3 \times 20 + 3 \\ 20 = 6 \times 3 + 2 \\ 3 = \underline{\quad} \times 2 + \underline{\quad} \end{array}$$

The diagram illustrates the steps of the Euclidean algorithm for finding the GCD of 63 and 20. It consists of three equations arranged vertically. The first equation is  $63 = 3 \times 20 + 3$ . A curved arrow points from the remainder 3 in this equation to the first number 3 in the third equation  $3 = \underline{\quad} \times 2 + \underline{\quad}$ . Another curved arrow points from the remainder 2 in the second equation  $20 = 6 \times 3 + 2$  to the second number 2 in the third equation.



**Step 1:** Sort the two elements 20 and 63, with the larger one first. Find the quotient and remainder when dividing the larger number by the smaller number. Then keep doing the same with the resulting divisors and remainders, until the remainder becomes 1.

$$63 = 3 \times 20 + 3$$

$$20 = 6 \times 3 + 2$$

$$3 = 1 \times 2 + 1$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

so

$$1 = 3 - 1 \times 2 =$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

so

$$1 = 3 - 1 \times 2 = 3 - 1 \times (20 - 6 \times 3)$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

so

$$1 = 3 - 1 \times 2 = 3 - 1 \times (20 - 6 \times 3) = 7 \times 3 - 20$$



**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

so

$$\begin{aligned} 1 &= 3 - 1 \times 2 = 3 - 1 \times (20 - 6 \times 3) = 7 \times 3 - 20 \\ &= 7 \times (63 - 3 \times 20) - 20 \end{aligned}$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

so

$$\begin{aligned} 1 &= 3 - 1 \times 2 = 3 - 1 \times (20 - 6 \times 3) = 7 \times 3 - 20 \\ &= 7 \times (63 - 3 \times 20) - 20 = 7 \times 63 - 22 \times 20 \end{aligned}$$

**Step 2:** Solve the equations for the remainders, starting from the last equation. Then, plug the expressions for the remainders into the subsequent equations until you reach the original one.

$$1 = 3 - 1 \times 2$$

$$2 = 20 - 6 \times 3$$

$$3 = 63 - 3 \times 20$$

so

$$\begin{aligned} 1 &= 3 - 1 \times 2 = 3 - 1 \times (20 - 6 \times 3) = 7 \times 3 - 20 \\ &= 7 \times (63 - 3 \times 20) - 20 = 7 \times 63 - 22 \times 20 \end{aligned}$$

Recall that we were looking for a multiplicative inverse of 20 modulo 63, i.e., we wanted some number  $b$  such that

$$20 \times b \equiv 1 \pmod{63}$$

Recall that we were looking for a multiplicative inverse of 20 modulo 63, i.e., we wanted some number  $b$  such that

$$20 \times b \equiv 1 \pmod{63}$$

From our list, we have  $1 \equiv -22 \times 20 \pmod{63}$

so  $-22$  is a multiplicative inverse of 20 modulo 63

Recall that we were looking for a multiplicative inverse of 20 modulo 63, i.e., we wanted some number  $b$  such that

$$20 \times b \equiv 1 \pmod{63}$$

From our list, we have  $1 \equiv -22 \times 20 \pmod{63}$

so  $-22$  is a multiplicative inverse of 20 modulo 63

If we prefer positive numbers, we can replace  $-22$  by

$$-22 \pmod{63}$$

i.e., 41, so 41 is also a multiplicative inverse of 20 modulo 63.

**Verifying the result:**

$$20 \times 41 = 820 = 13 \times 63 + 1$$

so

$$20 \times 41 \equiv 1 \pmod{63}.$$

# RSA encryption

Encryption:  $y \equiv x^r \pmod{n}$

Decryption:  $x \equiv y^s \pmod{n}$

How to construct the keys  $r, n$  and  $s$ ?

In contrast to what we have done before, the keys should not be chosen randomly!  $r, n$  and  $s$  are chosen by the following procedure:

1. Choose two prime numbers  $p$  and  $q$
2. Choose  $n = p \times q$
3. Choose  $r$  as any number that has no prime factors in common with  $(p - 1) \times (q - 1)$
4. Compute  $s$  from  $p$  and  $q$  by the Euclidean algorithm



## How do we compute the private key?

We find  $s$  by computing the multiplicative inverse of  $r$  modulo  $(p-1)(q-1)$  using the Euclidean algorithm. In other words, we find a number  $s$  such that

$$rs \equiv 1 \pmod{(p-1)(q-1)}$$

This means that  $rs = 1 + L \times (p-1)(q-1)$  for an integer  $L$ .

## Why does RSA work?

Suppose that our message is a number  $x$  smaller than  $n$  (any text message can be turned into a sequence of small numbers).

We compute  $y \equiv x^r \pmod{n}$

We say that  $y$  is the encrypted version of  $x$ .

From  $y$ , we compute

$$z \equiv y^s \pmod{n},$$

We now show that  $z$  is our original  $x$  again.

## Fermat's little theorem:

If  $p$  is prime, then  $a^p \equiv a \pmod{p}$  for all integers  $a$ .

## Fermat's little theorem:

If  $p$  is prime, then  $a^p \equiv a \pmod{p}$  for all integers  $a$ .

**Example:**  $p = 3$ ,  $a^3 \equiv a \pmod{3}$

For  $a = 0$ ,  $a^3 \equiv 0 \pmod{3} \equiv a \pmod{3}$

For  $a = 1$ ,  $a^3 \equiv 1 \pmod{3} \equiv a \pmod{3}$

For  $a = 2$ ,  $a^3 \equiv 8 \pmod{3} \equiv 2 \pmod{3} \equiv a \pmod{3}$

## Fermat's little theorem:

If  $p$  is prime, then  $a^p \equiv a \pmod{p}$  for all integers  $a$ .

Multiplying both sides by  $a^{p-1}$ , we obtain

$$a^{p-1} \times a^p \equiv a^p \equiv a \pmod{p}$$

Repeating the procedure  $N - 1$  times, we obtain

$$a^{(N-1)(p-1)} \times a^p \equiv a \pmod{p}$$

or  $a^{N(p-1)+1} \equiv a \pmod{p}$ .

This expression is true for any  $N$  and  $a$ , which is the foundation of the RSA algorithm.

Encryption:  $y \equiv x^r \pmod{n}$

Decryption:  $z \equiv y^s \pmod{n},$

To show that RSA works, we first show that

$$z \equiv x \pmod{n}.$$

Encryption:  $y \equiv x^r \pmod{n}$

Decryption:  $z \equiv y^s \pmod{n},$

To show that RSA works, we first show that

$$z \equiv x \pmod{n}.$$

Since  $s$  is the multiplicative inverse of  $r$  modulo  $(p-1)(q-1)$ , we have

$$rs = 1 + L \times (p-1)(q-1)$$

Encryption:  $y \equiv x^r \pmod{n}$

Decryption:  $z \equiv y^s \pmod{n},$

To show that RSA works, we first show that

$$z \equiv x \pmod{n}.$$

Since  $s$  is the multiplicative inverse of  $r$  modulo  $(p-1)(q-1)$ , we have

$$rs = 1 + L \times (p-1)(q-1)$$

and

$$x^{rs} = x^{L(p-1)(q-1)+1} \equiv x \pmod{p}.$$

Here we use the relation

$$a^{N(p-1)+1} \equiv a \pmod{p}$$

for any  $N$  and  $a$ .



Therefore, we have  $x^{rs} - x = \text{multiple of } p$ .

Therefore, we have  $x^{rs} - x = \text{multiple of } p$ .

Similarly,

$$x^{rs} = x^{L(p-1)(q-1)+1} \equiv x \pmod{q},$$

and we have  $x^{rs} - x = \text{multiple of } q$ .

Therefore, we have  $x^{rs} - x = \text{multiple of } p$ .

Similarly,

$$x^{rs} = x^{L(p-1)(q-1)+1} \equiv x \pmod{q},$$

and we have  $x^{rs} - x = \text{multiple of } q$ .

Since  $p$  and  $q$  are two different primes,  $x^{rs} - x$  can be a multiple of both  $p$  and  $q$  only if it is a multiple of their product  $n = p \times q$ , which implies that  $x^{rs} \equiv x \pmod{n}$ .

Since  $y \equiv x^r \pmod{n}$  (encryption of RSA)  
and  $z \equiv y^s \pmod{n}$  (decryption of RSA),  
we can write

$$y^s \equiv (x^r)^s \pmod{n} \equiv x^{rs} \pmod{n}$$

Since  $y \equiv x^r \pmod{n}$  (encryption of RSA)  
and  $z \equiv y^s \pmod{n}$  (decryption of RSA),  
we can write

$$y^s \equiv (x^r)^s \pmod{n} \equiv x^{rs} \pmod{n}$$

Therefore,  $z \equiv x^{rs} \pmod{n} \equiv x \pmod{n}$ .

This equivalence precisely means that  $z$  and  $x$   
have the same remainders when divided by  $n$ .

Since  $y \equiv x^r \pmod{n}$  (encryption of RSA)  
and  $z \equiv y^s \pmod{n}$  (decryption of RSA),  
we can write

$$y^s \equiv (x^r)^s \pmod{n} \equiv x^{rs} \pmod{n}$$

Therefore,  $z \equiv x^{rs} \pmod{n} \equiv x \pmod{n}$ .

This equivalence precisely means that  $z$  and  $x$   
have the same remainders when divided by  $n$ .

Since both  $z$  and  $x$  are integers smaller than  $n$ ,  
this precisely means that  $z = x$ . RSA therefore  
works!

## Recap of RSA

Let  $x$  be a block of “plain text” in the form of numbers. Given the pair of values  $n$  and  $r$  (the public key), the encrypted message  $y$  is given by

$$y \equiv x^r \pmod{n} \text{ (encryption of RSA).}$$

Given the value of  $s$  (the private key), we can retrieve  $x$  by

$$x \equiv y^s \pmod{n} \text{ (decryption of RSA).}$$

# RSA encryption

Encryption:  $y \equiv x^r \pmod{n}$

Decryption:  $x \equiv y^s \pmod{n}$

How to construct the keys  $r,n$  and  $s$ ?

In contrast to what we have done before, the keys should not be chosen randomly!  $r,n$  and  $s$  are chosen by the following procedure:

1. Choose two prime numbers  $p$  and  $q$
2. Choose  $n = p \times q$
3. Choose  $r$  as any number that has no prime factors in common with  $(p - 1) \times (q - 1)$
4. Compute  $s$  from  $p$  and  $q$  by the Euclidean algorithm



# Why is RSA secure?

We need to know  $s$  to decrypt. Now  $s$  is the multiplicative inverse of  $r$  modulo  $(p - 1)(q - 1)$ . Any outsider knows the value of  $r$ , and if they knew  $(p - 1)(q - 1)$ , then it would be easy (with the Euclidean Algorithm) to compute  $s$ . However, they does not know  $(p - 1)(q - 1)$ . They only know  $n$ , which is equal to  $pq$ .

# Why is RSA secure?

We need to know  $s$  to decrypt. Now  $s$  is the multiplicative inverse of  $r$  modulo  $(p - 1)(q - 1)$ . Any outsider knows the value of  $r$ , and if they knew  $(p - 1)(q - 1)$ , then it would be easy (with the Euclidean Algorithm) to compute  $s$ . However, they does not know  $(p - 1)(q - 1)$ . They only know  $n$ , which is equal to  $pq$ .

To find  $(p - 1)(q - 1)$ , they would need to know the prime factors  $p$  and  $q$  of  $n$ , and factoring large numbers is computationally infeasible.

## **Example (encryption and decryption in RSA):**

1. Choose two primes  $p = 5$ ,  $q = 7$ .
2. Compute  $n = pq = 35$ .
3. Choose a positive integer  $r$  that has no common prime factor with  $(p-1)(q-1) = 24$ . This has the prime factorization  $24 = 2 \times 2 \times 2 \times 3$ . We pick  $r = 5$ .
4. Compute the multiplicative inverse of  $r$  modulo  $(p-1)(q-1)$ . That is, find  $s$  such that

$$rs \equiv 1 \pmod{(p-1)(q-1)}$$

using the Euclidean algorithm.

## Example (encryption and decryption in RSA):

The Euclidean algorithm gives:

$$24 = 4 \times 5 + 4$$

$$5 = 1 \times 4 + 1$$

so

$$1 = 5 - 1 \times 4$$

$$4 = 24 - 4 \times 5$$

It follows that

$$\begin{aligned} 1 &= 5 - 1 \times 4 = 5 - 1 \times (24 - 4 \times 5) \\ &= 5 \times 5 - 24 \end{aligned}$$

so  $5 \times 5 \equiv 1 \pmod{24}$  which means that  $s = 5$  is a multiplicative inverse of  $r = 5$ .

5. The public key is  $(n, r) = (35, 5)$  and the private key is  $s = 5$ .

6. In order to encrypt  $x = 11$ , we compute

$$y \equiv x^r \pmod{n},$$

and we have

$$y \equiv 11^5 \pmod{35} \equiv 16 \pmod{35}.$$

7. In order to decrypt  $y = 16$ , we compute

$$\begin{aligned} x &\equiv y^s \pmod{n} \\ &\equiv 16^5 \pmod{35} \equiv 11 \pmod{35}. \end{aligned}$$