

# Lecture 4

## A Brush-up on Number Theory

Stefan Dziembowski  
University of Rome  
*La Sapienza*

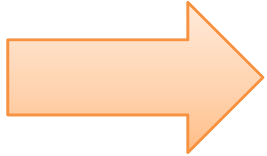


SAPIENZA  
UNIVERSITÀ DI ROMA

BiSS 2009  
Bertinoro International  
Spring School  
2-6 March 2009



# Plan



1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Euler's  $\phi$  function, group isomorphism, product of groups
6. Chinese Remainder Theorem, groups  $\mathbb{Z}_N^*$ , and  $\mathbb{QR}_N$ , where  $N=pq$

# Number theory in cryptography - advantages

1. security can (in principle) be based on **famous mathematical conjectures**,
2. the constructions have a “**mathematical structure**”, this allows us to create more advanced constructions (**public key encryption, digital signature schemes, and many others...**).
3. the constructions have a natural security parameter (hence they **can be “scaled”**).

## Additional advantage

a **practical application** of an area that was **never believed to be practical...** (wonderful argument for all theoreticians!)

# Number theory in cryptography - disadvantages

1. cryptography based on number theory is much **less efficient!**
2. the number-theoretic “structure” may help the cryptanalyst...

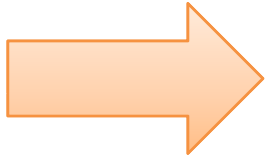
# Number theory as a source of hard problems

In this lecture we will look at some **basic number-theoretic** problems,

identifying those that **may be useful in cryptography**.

# Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Euler's  $\phi$  function, group isomorphism, product of groups
6. Chinese Remainder Theorem, groups  $\mathbb{Z}_N^*$ , and  $QR_N$ , where  $N=pq$



# Famous algorithmic problems in number theory

## primality testing:

input:  $a \in \mathbb{N}$

output:

- **yes** if  $a$  is a prime,
- **no** otherwise

this problem is

computationally easy

## factoring:

input:  $a \in \mathbb{N}$

output: factors of  $a$

this problem is believed to be computationally hard if  $a$  is a product of two long random primes  $p$  and  $q$ , of equal length.

# Primality testing

**x** – the number that we want to test

**Sieve of Eratosthenes** (ca. **240 BC**):

takes  $\sqrt{x}$  steps, which is exponential in  $|\log_2 x|$

**Miller-Rabin** test (late **1970s**) is **probabilistic**:

- if **x** is prime it always outputs **yes**
- if **x** is composite it outputs **yes** with probability at most  $\frac{1}{4}$ .

Probability is taken only over the internal randomness of the algorithm, so **we can iterate!**

The **error goes to zero exponentially fast**.  
This **algorithm is fast and practical!**

**Deterministic algorithm** of **Agrawal, Saxena and Kayal (2002)**

polynomial but **very inefficient in practice**

# How to select a **random** prime of length **m**?

Select a random number **x** and test if it is prime.

## Theorem

There exists a constant **c** such that for any **m** the number on **m**-bit primes is:

$$c \cdot 2^m / m.$$

Hence, the set of primes is “**dense**”.

# Factoring is believed to be hard!

## Factoring assumption.

Take random primes  $p$  and  $q$  of length  $n$ .

Set  $N = pq$ .

No polynomial-time algorithm that is given  $N$  can find  $p$  and  $q$  in with a **non-negligible probability**.

**Factoring is a subject of very intensive research.**

Currently  $|N|=2048$  is believed to be a safe choice.

# So we have a one-way function!

$f(p,q) = pq$  is **one-way**.  
(assuming the factoring assumption holds).

Using the theoretical results [HILL99] this is enough to construct secure encryption schemes.

It turns out that we can do much better:

based on the number theory we can construct  
**efficient schemes**,  
that have some **very nice additional properties**  
(**public key cryptography!**)

**But how to do it?**

We need to some more maths...<sup>11</sup>

# Notation

Suppose **a** and **b** are integers, such that **a**  $\neq$  0

**a** | **b**:

- **a** divides **b**, or
- **a** is a **divisor** of **b**, or
- **a** is a **factor** of **b**

(if **a**  $\neq$  1 then **a** is a **non-trivial factor** of **b**)

**gcd(a,b)** = “the greatest common factor of **a** and **b**”

If **gcd(a,b)** = 1 then we say that  
**a** and **b** are **relatively prime**.

# How to compute $\text{gcd}(a,b)$ ?

## Euclidean algorithm

**Recursion:**

(assume  $a \geq b \geq 0$ )

$\text{gcd}(a,b) =$  if  $b \mid a$   
then return  $b$   
else return  $\text{gcd}(b, a \bmod b)$

It can be shown that

- this algorithm is **correct** (induction),
- it terminates in **polynomial number of steps**.

# Example

computing **gcd(185,40)**:

a	b	a mod b
185	40	25
40	25	15
25	15	10
15	10	5
10	5	0



# Claim

Let **a** and **b** be positive integers.

There always exist integers **X** and **Y** such that

$$\mathbf{Xa + Yb = \gcd(a,b)}$$

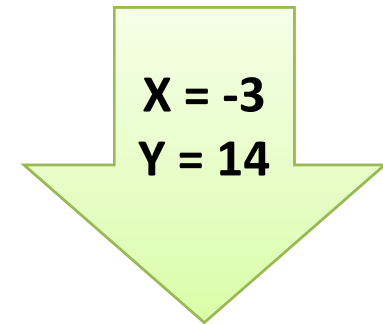
**X** and **Y** can be computed using the extended  
**Euclidian algorithm.**

# Example of an execution of the extended Euclidian algorithm

computing **X** and **Y** such that

$$\mathbf{X \cdot 185 + Y \cdot 40 = 5}$$

$$\mathbf{a = 185 \quad b = 40}$$



a	b	a mod b
---	---	---------

$$185 - 40 \cdot 4 = 25$$

$$40 - 25 \cdot 1 = 15$$

$$25 - 15 \cdot 1 = 10$$

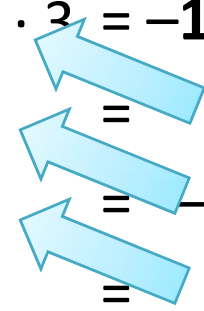
$$15 - 10 \cdot 1 = 5$$

$$5 = 40 \cdot 2 - (185 - 40 \cdot 4) \cdot 3 = -185 \cdot 3 + 40 \cdot 14$$

$$5 = -25 + (40 - 25 \cdot 1) \cdot 2 = 40 \cdot 2 - 25 \cdot 3$$

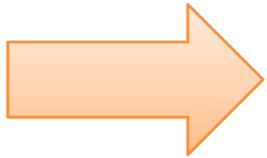
$$5 = 15 - (25 - 15 \cdot 1) \cdot 1 = -25 \cdot 1 + 15 \cdot 2$$

$$5 = 15 - 10 \cdot 1 = 15 \cdot 1 - 10 \cdot 1$$



# Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Euler's  $\phi$  function, group isomorphism, product of groups
6. Chinese Remainder Theorem, groups  $\mathbb{Z}_N^*$ , and  $\text{QR}_N$ , where  $N=pq$



# Groups

A **group** is a set **G** along with a binary operation  $\circ$  such that

- [**closure**] for all  $g, h \in G$  we have  $g \circ h \in G$ ,
- there exists an **identity**  $e \in G$  such that for all  $g \in G$  we have

$$e \circ g = g \circ e = g,$$

- for every  $g \in G$  there exists an **inverse of**, that is an element  $h$  such that

$$g \circ h = h \circ g = e,$$

- [**associativity**] for all  $g, h, k \in G$  we have

$$g \circ (h \circ k) = (g \circ h) \circ k$$

- [**commutativity**] for all  $g, h \in G$  we have

$$g \circ h = h \circ g$$

if this holds, the group is called **abelian**

**order** of **G** =  $|G|$ .

# Subgroups

A group  $G$  is a **subgroup** of  $H$  if

- $G$  is a subset of  $H$ ,
- the group operation  $\circ$  is the same as in  $H$

# Additive/multiplicative notation

## Convention:

- [additive notation]

If the groups operation is denoted with  $+$ , then:

- the inverse of  $g$  is denoted with  $-g$ ,
- the neutral element is denoted with  $0$ ,
- $g + \dots + g$  ( $n$  times) is denoted with  $ng$ .

- [multiplicative notation]

If the groups operation is denoted with  $\bullet$ , then:

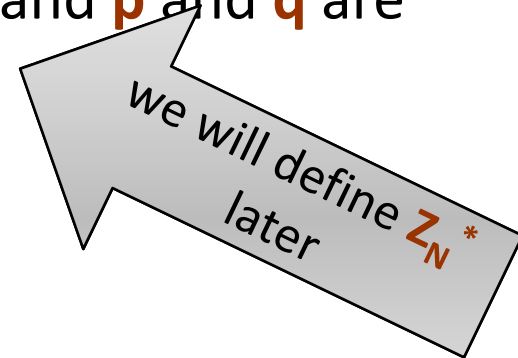
- sometimes we write  $gh$  instead of  $g \bullet h$ ,
- the inverse of  $g$  is denoted with  $g^{-1}$  or  $1/g$ .
- the neutral element is denoted with  $1$ ,
- $g \bullet \dots \bullet g$  ( $n$  times) is denoted with  $g^n$ .

# Examples of groups

- $\mathbf{R}$  (reals) is not a group under multiplication.
- $\mathbf{R} \setminus \{0\}$  is a group.
- $\mathbf{Z}$  (integers):
  - is a group under **addition** (identity element:  $\mathbf{0}$ ),
  - is **not** a group under **multiplication**.
- $\mathbf{Z}_N = \{0, \dots, N-1\}$  (integers modulo  $\mathbf{N}$ ) are a group under **addition** (identity element:  $\mathbf{0}$ ).
- $\mathbf{Z}_p^* = \{1, \dots, p-1\}$  are a group under **multiplication** (identity element:  $\mathbf{1}$ ).

# Which groups are useful in cryptography?

- $\mathbf{Z}_n$  – is not useful, because all natural problems are easy in this group.
- Useful groups:
  - a multiplicative group  $\mathbf{Z}_p^* = \{1, \dots, p-1\}$ , where  $\mathbf{p}$  is a prime,
  - a multiplicative group  $\mathbf{Z}_N^*$ , where  $\mathbf{N}=\mathbf{pq}$  and  $\mathbf{p}$  and  $\mathbf{q}$  are primes,
  - groups based on elliptic curves,
  - other...All of them “**have some hard problems**”.



we will define  $\mathbf{Z}_N^*$  later

$\mathbb{Z}_N$  is a group under addition. Is it also a group under multiplication?

**No:** **0** doesn't have an inverse.

What about other elements of  $\mathbb{Z}_N$ ?

Example **N = 12**.

Only: **1,5,7,11**  
have an inverse!

Why?

Because they  
are **relatively  
prime** to **12**.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	<b>1</b>	2	3	4	5	6	7	8	9	10	11
2	0	2	4	6	8	10	0	2	4	6	8	10
3	0	3	6	9	0	3	6	9	0	3	6	9
4	0	4	8	0	4	8	0	4	8	0	4	8
5	0	5	10	3	8	<b>1</b>	6	11	4	9	2	7
6	0	6	0	6	0	6	0	6	0	6	0	6
7	0	7	2	9	4	11	6	<b>1</b>	8	3	10	5
8	0	8	4	0	8	4	0	8	4	0	8	4
9	0	9	6	3	0	9	6	3	0	9	6	3
10	0	10	8	6	4	2	0	10	8	6	4	2
11	0	11	10	9	8	7	6	5	4	3	2	<b>1</b>

## Observation

If  $\gcd(a,n) > 1$  then for every integer  $b$  we have  
 $ab \bmod n \neq 1$ .

## Proof

**Suppose** for the sake of contradiction that  $ab \bmod n = 1$ .

Hence we have:

$$ab = nk + 1$$



$$ab - nk = 1$$

Since  $\gcd(a,n)$  divides both  $ab$  and  $nk$  it also divides  $ab - nk$ .

Thus  $\gcd(a,n)$  has to divide  $1$ . Contradiction.

**QED**

$$\mathbb{Z}_N^*$$

Define  $\mathbb{Z}_N^* = \{a \in \mathbb{Z}_N : \gcd(a, N) = 1\}$ .

Then  $\mathbb{Z}_N^*$  is an abelian group under multiplication modulo  $N$ .

### Proof

First observe that  $\mathbb{Z}_N^*$  is **closed under multiplication** modulo  $N$ .

This is because if  $a, b$  are relatively prime to  $N$ , then  $ab$  is also relatively prime to  $N$ .

**Associativity** and **commutativity** are trivial.

**1** is the identity element.

It remains to show that for every  $a \in \mathbb{Z}_N^*$  there always exist an  $b \in \mathbb{Z}_N^*$  that is an **inverse of  $a$  modulo  $N$** .

We say that  $b$  is an **inverse of  $a$  modulo  $N$**  if:

$$a \cdot b = 1 \pmod{N}$$

### Lemma

Suppose that  $\gcd(a, N) = 1$ . Then for every  $a \in \mathbb{Z}_N^*$  there always exist an element  $b \in \mathbb{Z}_N^*$  such that

$$a \cdot b \pmod N = 1.$$

Since  $\gcd(a, N) = 1$  there always exist integers  $X$  and  $Y$  such that

$$Xa + YN = 1.$$

Therefore clearly  $Xa = 1 \pmod N$ .

Of course  $X$  may not belong to  $\mathbb{Z}_N^*$ .

What to do?

Define  $b := X \pmod N$ .

Hence  $b = X + tN$ . (for some integer  $t$ )

We have

$$\begin{aligned} ab &= a \cdot (X + tN) \\ &= aX + atN \\ &= 1 \pmod N \end{aligned}$$

Hence  $b$  is an inverse of  $a$ . And it can be efficiently computed (using the extended Euclidian algorithm).

Can it be that  
 $\gcd(X + tN, N) = c > 1$ ?  
No, because then  $c \mid X$  and hence  
 $\gcd(X, N) = c > 1$

**QED**

# An example

**p** – a prime

$$\mathbb{Z}_p^* = \{1, \dots, p-1\}.$$

$\mathbb{Z}_p^*$  is an abelian group under multiplication modulo **p**.

# A simple observation

For every  $a, b, c \in G$ . If

$$ac = bc$$

then

$$a = b.$$

## Proof

$$ac = bc$$

↓

$$(ac) c^{-1} = (bc) c^{-1}$$

↓

$$a (cc^{-1}) = b (cc^{-1})$$

↓

$$a \cdot 1 = b \cdot 1$$

↓

$$a = b$$

# Corollary

In every group  $G$  and every element  $g \in G$  the function

$$f : G \rightarrow G$$

$$f(x) = x \circ g$$

is a bijection.

(or, in other words, a **permutation on  $G$** ).

# Look at the following

$$Z_{11}^* \quad (|Z_{11}^*| = 10)$$

Take some element  $g$  and start computing

$$g^1, g^2, g^3, g^4, \dots$$

- If  $g = 2$  then we get  
 $2, 4, 8, 5, 10, 9, 7, 3, 6, 1, 2, 4, 8, 5, 10, 9, 7, 3, 6, 1, 2, \dots$
- If  $g = 5$  then we get  
 $5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, \dots$

We start  
a  
repeating  
after  
**10** or **5**  
iterations

# Lemma

$G$  – an abelian group,  $m := |G|$ ,  $g \in G$ .

Then  $g^m = 1$ .

## Proof

Suppose  $G = \{g_1, \dots, g_m\}$ .

Observe that

from associativity  
and commutativity

$$\begin{aligned} & \cancel{g_1} \circ \dots \circ \cancel{g_m} \\ &= (g \circ g_1) \circ \dots \circ (g \circ g_m) \\ &= g^m \circ (\cancel{g_1} \circ \dots \circ \cancel{g_m}) \end{aligned}$$

these are  
the same  
elements  
(permuted),  
because the  
function  
 $f(x) = g \circ x$   
is a permutation

Hence  $g^m = 1$ .

# Observation

**G** – an abelian group, **m** := **|G|**, **g** ∈ **G**, **i** ∈ **N**.  
Then  **$g^i = g^{i \bmod m}$** .

## Proof

Write  **$i = qm + r$** , where  **$r = i \bmod m$** , and **q** is some integer.

We have

$$g^i = g^{qm+r} = (g^m)^q \cdot g^r = 1^q \cdot g^r = g^r$$

**QED**

# Order of an element

$G$  – a group,  $g \in G$ .

$$\langle g \rangle := \{g^0, g^1, \dots\}$$

$\langle g \rangle$  is a **subgroup** of  $G$  **generated by**  $g$ .

## Definition

An **order of**  $g$  (denoted  $\langle g \rangle$ ) is the smallest integer  $i > 0$  such that  $g^i = 1$ .

Clearly:  $\langle g \rangle := \{g^0, \dots, g^{i-1}\}$ .

Of course  $i \leq |G|$

# Example

$$Z_{11}^* \quad (|Z_{11}^*| = 10)$$

Take some element  $g$  and start computing

$$g^1, g^2, g^3, g^4, \dots$$

- If  $g = 2$  then we get  
 $2, 4, 8, 5, 10, 9, 7, 3, 6, 1, 2, 4, 8, 5, 10, 9, 7, 3, 6, 1, 2, \dots$   
the order of  $2$  is  $10$
- If  $g = 5$  then we get  
 $5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, \dots$   
the order of  $5$  is  $5$

## Lemma

$G$  – a group,  $g \in G$  – an element of order  $i$ .  
Then  $g^x = g^y$  if and only if  $x = y \pmod{i}$ .

## Proof

( $\leftarrow$ )

$$\begin{aligned} g^x &= g^{(x \bmod i) + ti} \\ &= g^{(x \bmod i)} \cdot (g^i)^t \\ &= g^{x \bmod i} \end{aligned}$$

for some integer  $t$

=1

equal!

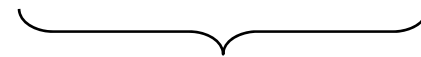
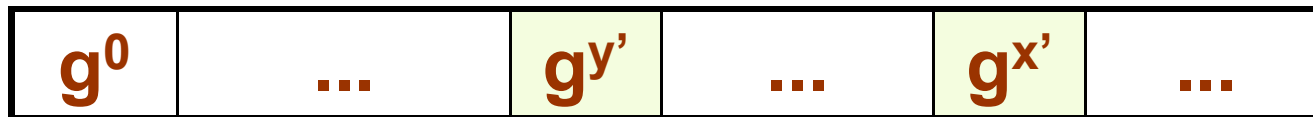
Using the same reasoning:  $g^y = g^{y \bmod i}$

( $\rightarrow$ ) if " $g^x = g^y$  then  $x = y \pmod i$ "

Set  $x' := x \pmod i$ , and  $y' = y \pmod i$ .

For the sake of contradiction suppose that  $x' \neq y'$ .

Suppose  $x' > y'$ .



$$g^{x'} = g^{y'}$$



$$g^{x'-y'} = 1$$

**Contradiction**, since  $x' - y' < i$  and  $g$  has order  $i$ .

**QED**

# Look again at the example:

$$Z_{11}^* \quad (|Z_{11}^*| = 10)$$

- If  $g = 2$  then we get  
 $2, 4, 8, 5, 10, 9, 7, 3, 6, 1, 2, 4, 8, 5, 10, 9, 7, 3, 6, 1, 2, \dots$   
the order of  $2$  is  $10$
- If  $g = 5$  then we get  
 $5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, 3, 4, 9, 1, 5, \dots$   
the order of  $5$  is  $5$
- If  $g = 10$  then we get  
 $10, 1, 10, 1, 10, 1, 10, 1, 10, 1, \dots$   
the order of  $10$  is  $2$
- If  $g = 1$  then we get  
 $1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots$   
the order of  $1$  is  $1$

Observation: All those numbers divide  $10$

## Lemma

**G** – a group of order **m**.

Suppose some **g**  $\in$  **G** has order **i**.

Then **i** | **m**.

## Proof

For the sake of contradiction assume that **i** does not divide **m**.

By our previous lemma:

Of course

$$g^m = 1$$

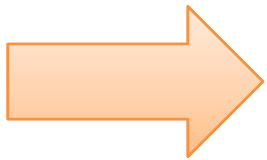
$$g^m = g^{m \bmod i}$$

Observe that  
 $0 < m \bmod i < i$

So we obtain a contradiction with the assumption that **g** has order **i**.

# Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Euler's  $\phi$  function, group isomorphism, product of groups
6. Chinese Remainder Theorem, groups  $\mathbb{Z}_N^*$ , and  $\text{QR}_N$ , where  $N=pq$



# Cyclic groups

If there exists  $g$  such that  $\langle g \rangle = G$  then we say that  $G$  is **cyclic**.

Such a  $g$  is called a **generator of  $G$** .

For example  $g = 2$  is a generator of  $\mathbb{Z}_p^*$

**2,4,8,5,10,9,7,3,6,1,2,4,8,5,10,9,7,3,6,1,2,...**

## Observation

Every group  $G$  of a prime order  $p$  is cyclic.

Every element  $g$  of  $G$ , except the identity is its generator.

## Proof

The order of  $g$  has to divide  $p$ .

So, the only possible orders of  $g$  are  $1$  or  $p$ .

Trivial:  $x$  has “order  $1$ ” if  $x^1 = 1$

Only identity has order  $1$ , so all the other elements have order  $p$ .

# Another fact

## Theorem

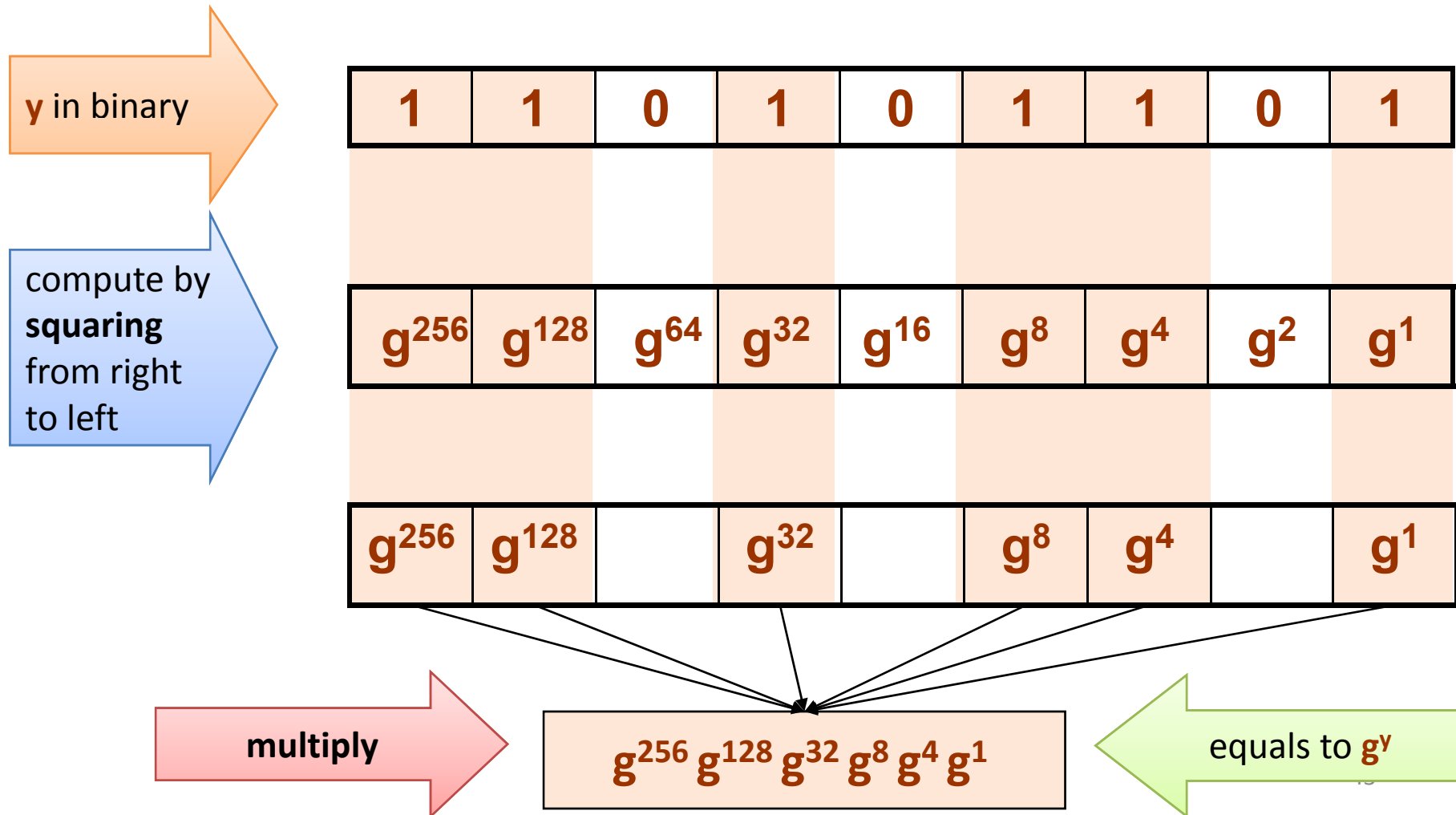
If  $p$  is prime, then  $\mathbb{Z}_p^*$  is cyclic.

We leave it without a proof.

# How to compute $g^y$ for large $y$ ?

If the multiplication is easy then we can use the “square-and-multiply” method

**Example**



# What about the other direction?

(**g** – a generator)

It turns out the in many groups inverting

$$f(y) = g^y$$

is hard!

# The discrete logarithm

Suppose  $G$  is cyclic and  $g$  is its generator.  
For every element  $x$  there exists  $y$  such that

$$x = g^y$$

Such a  $y$  will be called a **discrete logarithm** of  $x$ , and it is denoted as  $y := \log x$ .

In many groups computing a discrete log is **believed to be hard**.

Informally speaking:

$f: \{0, \dots, |G| - 1\} \rightarrow G$  defined as  $f(y) = g^y$  is believed to be a **one-way function** (in some groups).

# Hardness of the discrete log

In some groups it is easy:

- in  $\mathbb{Z}_n$  it is **easy** because  $a^e = e \cdot a \bmod n$
- In  $\mathbb{Z}_p^*$  (where  $p$  is prime) it is believed to be **hard**.
- There exist also **other groups** where it is believed to be **hard** (e.g. based on the **Elliptic curves**).
- Of course: if **P = NP** then computing the discrete log is easy.

(in the groups where the exponentiation is easy)

# How to define formally “the discrete log assumption”

It needs to be defined for *any* parameter  $1^n$ .

Therefore we need an algorithm **H** that

- on input  $1^n$
- outputs:
  - a description of a cyclic group **G** of order **q**, such that  $|q| = n$ ,
  - a generator **g** of **G**.

# Example

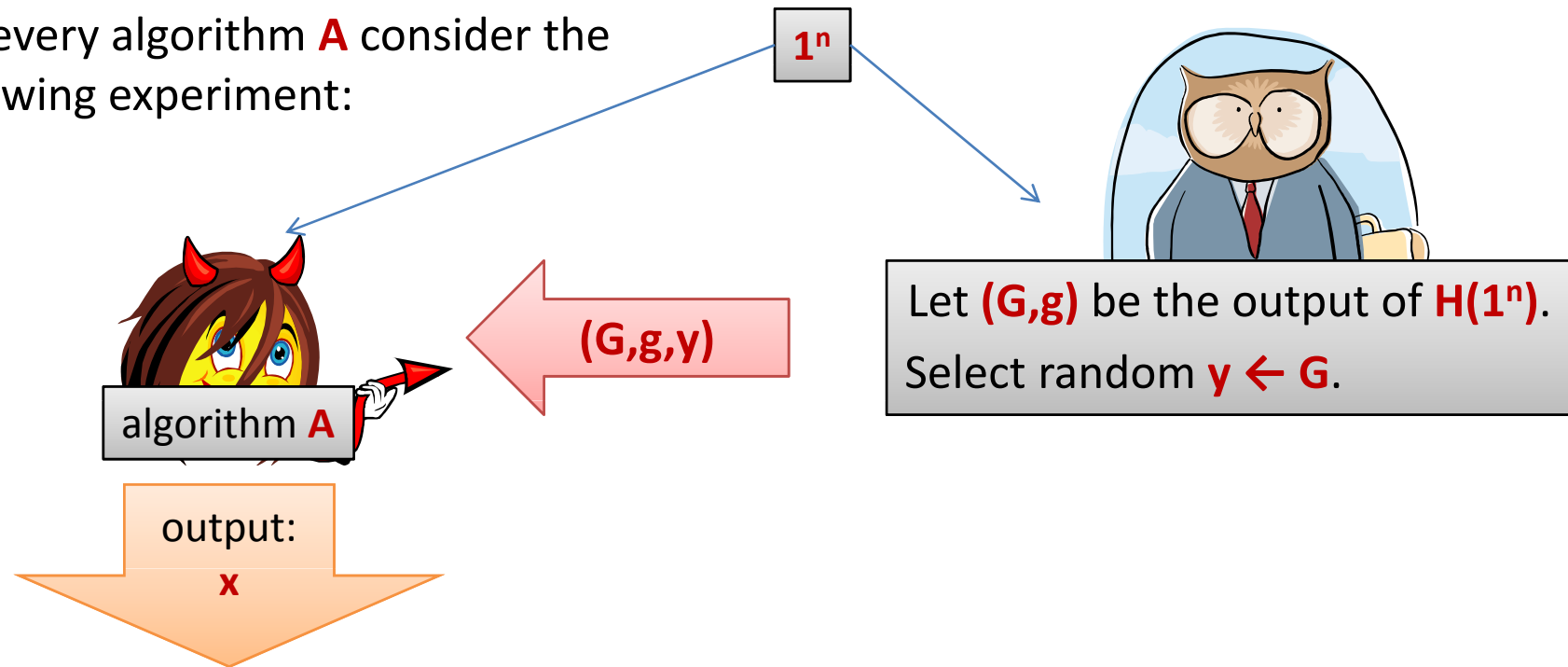
**H** on input  $1^n$ :

outputs a

- random prime **p** of length **n**
- a generator of  $\mathbb{Z}_p^*$

# The discrete log assumption

For every algorithm **A** consider the following experiment:



We say that a **discrete logarithm problem is hard with respect to  $H$**  if

$\forall$  poly-time algorithm **A**  $P(\mathbf{A}$  outputs  $x$  such that  $g^x = y)$  is negligible in  $n$

# One way function?

This looks almost the same as saying that

$$f(x) = g^x$$

is a one-way function.

The only difference is that the function **f** depends on the group **G** that was chosen randomly.

We could formalize it, by defining:

“one-way function families”

# Concrete functions

For the practical applications people often use concrete groups.

In particular it is common to chose some  $\mathbb{Z}_p^*$  for a fixed prime  $p$ .

For example the RFC3526 document specifies the primes of following lengths: **1536, 2048, 3072, 4096, 6144, 8192**.  
This is the **1536**-bit prime:

```
FFFFFFFF FFFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA237327 FFFFFFFFF FFFFFFFF.
```

the generator is: **2**.

# An problem

$f: \{0, \dots, p - 1\} \rightarrow \mathbb{Z}_p^*$  defined as  $f(y) = g^y$  is believed to be a **one-way function** (informally speaking),

but

from  $f(x)$  one can compute the parity of  $x$ .

We now show how to do it.

# Quadratic Residues

## Definition

**a** is a **quadratic residue modulo p** if there exists **b** such that

$$a = b^2 \pmod{p}$$

$QR_p$  – a set of quadratic residues modulo **p**

$QR_p$  is a subgroup of  $Z_p^*$

$$QNR_p := Z_p^* \setminus QR_p$$

Why?

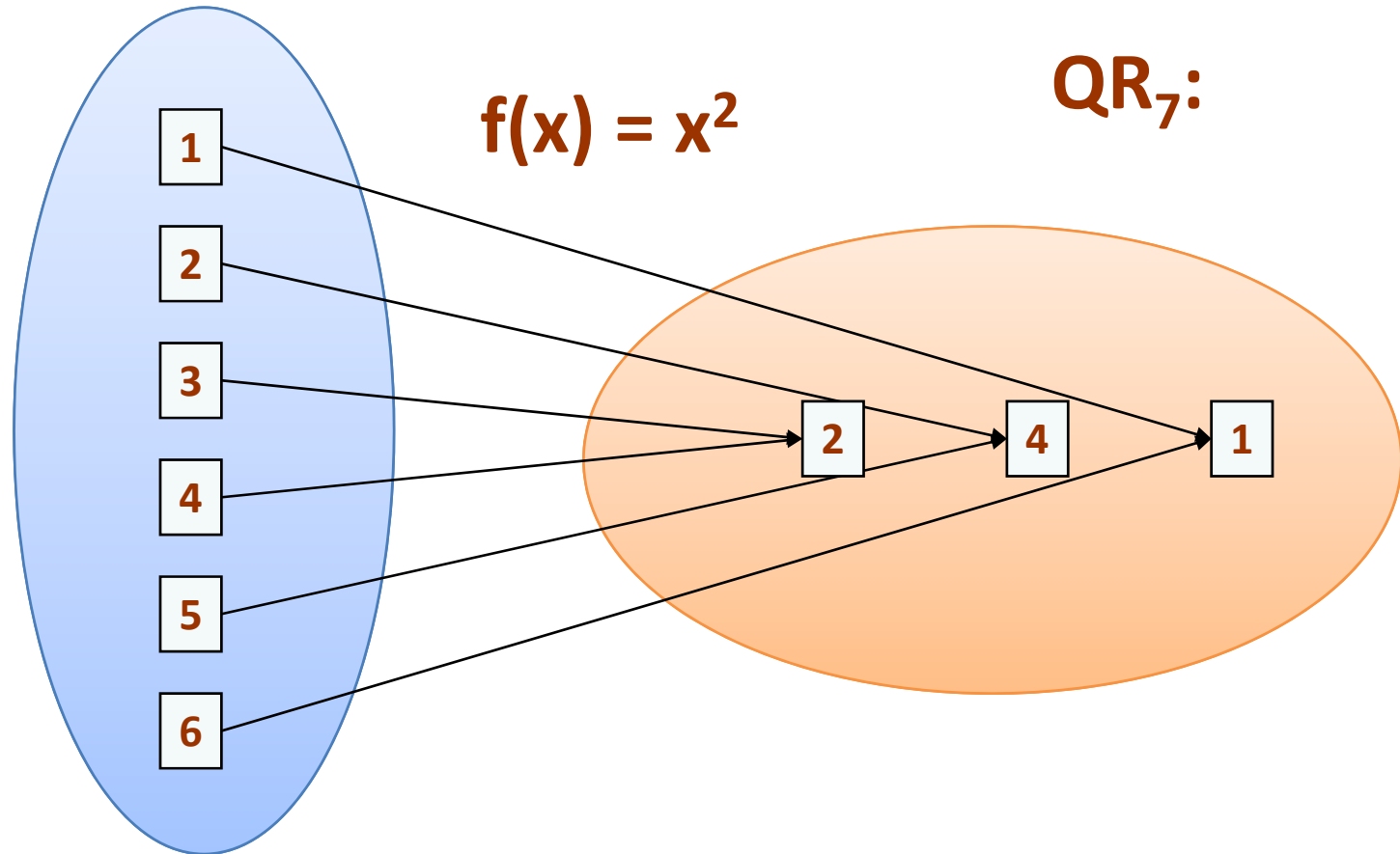
because:

- $1 \in QR$
- if  $a, a' \in QR$  then  $aa' \in QR$

What is the size of  $QR_p$ ?

# Example: $QR_7$

$Z_7^*$ :



**Lemma.**  $|QR_p| = |Z_p^*| / 2 = (p - 1) / 2$

A proof that

$$|QR_p| = (p - 1) / 2$$

Observation

Let  $g$  be a generator of  $Z_p^*$ .

Then  $QR_p = \{g^2, g^4, \dots, g^{p-1}\}$ .

**Proof**

Every element  $x \in Z_p^*$  is equal to  $g^i$  for some  $i$ .

Hence  $x^2 = g^{2i \bmod (p-1)} = g^j$ , where  $j$  is even.

Is it easy to test if  $a \in \text{QR}_p$ ?

Yes!

### Observation

$a \in \text{QR}_p$  iff  $a^{(p-1)/2} = 1 \pmod{p}$

### Proof

( $\rightarrow$ )

If  $a \in \text{QR}_p$  then  $a = g^{2i}$ .

Hence

$$\begin{aligned} a^{(p-1)/2} &= \\ &= (g^{2i})^{(p-1)/2} \\ &= g^{i(p-1)} = 1. \end{aligned}$$

$$a \in \text{QR}_p \text{ iff } a^{(p-1)/2} = 1 \pmod{p}$$

(←)

Suppose **a** is not a quadratic residue.

Then **a** =  $g^{2i+1}$ . Hence

$$\begin{aligned} & a^{(p-1)/2} \\ &= (g^{2i+1})^{(p-1)/2} \\ &= g^{i(p-1)} \cdot g^{(p-1)/2} \\ &= g^{(p-1)/2} \end{aligned}$$

which cannot be equal to **1** since **g** is a generator.

**QED**

# A problem

$f: \{0, \dots, p-1\} \rightarrow \mathbb{Z}_p^*$  defined as  $f(y) = g^y$  is a one-way function, but

from  $f(y)$  one can compute the parity of  $y$  (by checking if  $y \in QR$ )...

For some applications this is not good.

(but sometimes people don't care)

# What to do?

Instead of working in  $\mathbb{Z}_p^*$  work in its subgroup:  $\mathbb{QR}_p$

How to find a generator of  $\mathbb{QR}_p$ ?

Choose  $p$  that is a **strong prime**, that is:

$$p = 2q + 1, \text{ with } q \text{ prime.}$$

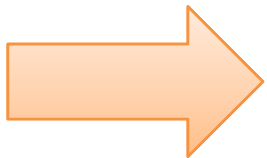
Hence  $\mathbb{QR}_p$  has a prime order ( $q$ ).

Every element (except of **1**) of a group of a prime order is its generator!

Therefore: every element of  $\mathbb{QR}_p$  is a generator. Nice...

# Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Euler's  $\phi$  function, group isomorphism, product of groups
6. Chinese Remainder Theorem, groups  $\mathbb{Z}_N^*$ , and  $\mathbb{QR}_N$ , where  $N=pq$



# Euler's $\phi$ function

Define

$$\phi(N) = |Z_N^*| = |\{a \in Z_N : \gcd(a, N) = 1\}|.$$

## Euler's theorem:

For every  $a \in Z_N^*$  we have  $a^{\phi(N)} = 1 \pmod N$ .

(trivially follows from the fact that for every  $g \in G$  we have  $g^{|G|} = 1$ ).

## Special case (“Fermat's little theorem”)

For every prime  $p$  and every  $a \in \{1, \dots, p-1\}$  we have

$$a^{p-1} = 1 \pmod N.$$

# Group isomorphism

**G** – a group with operation  $\circ$

**H** – a group with operation  $\square$

## Definition

A function

$$f: G \rightarrow H$$

is a **group isomorphism** if

1. it is a **bijection**, and
2. it is a **homomorphism**, i.e.: for every  $a, b \in G$  we have

$$f(g \circ h) = f(g) \square f(h).$$

If there exists an isomorphism between **G** and **H**, we say that they are **isomorphic**.

# A cross product of groups

$(G, \circ)$  and  $(H, \square)$  – groups

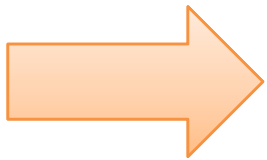
**Define** a group  $(G \times H, \bullet)$  as follows:

- the elements of  $G \times H$  are pairs  $(g, h)$ , where  $g \in G$ , and  $h \in H$ .
- $(g, h) \bullet (g', h') = (g \circ h, g' \square h')$ .

It is easy to verify that it is a group.

# Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Euler's  $\phi$  function, group isomorphism, product of groups
6. Chinese Remainder Theorem, groups  $\mathbf{Z_N^*}$ , and  $\mathbf{QR_N}$ , where  $\mathbf{N=pq}$



# Chinese Remainder Theorem (CRT)

Let  $N = pq$ , where  $p$  and  $q$  are two **distinct** primes.

Define:  $f(x) := (x \bmod p, x \bmod q)$

## Chinese Remainder Theorem (CRT):

$f$  is an isomorphism between

1.  $\mathbb{Z}_N$  and  $\mathbb{Z}_p \times \mathbb{Z}_q$
2.  $\mathbb{Z}_N^*$  and  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$

To prove it we need to show that

- $f$  is a **homomorphism**.
  - between  $\mathbb{Z}_N$  and  $\mathbb{Z}_p \times \mathbb{Z}_q$ , and
  - between  $\mathbb{Z}_N^*$  and  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ .
- $f$  is a **bijection**:
  - between  $\mathbb{Z}_N$  and  $\mathbb{Z}_p \times \mathbb{Z}_q$ , and
  - between  $\mathbb{Z}_N^*$  and  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ .

# **f** is a homomorphism

**f**:  $\mathbb{Z}_N \rightarrow \mathbb{Z}_p \times \mathbb{Z}_q$  is an **homomorphism**

Proof:

$$\begin{aligned} & \mathbf{f(a + b)} \\ & \quad \parallel \\ & \mathbf{(a + b \bmod p, a + b \bmod q)} \\ & \quad \parallel \\ & \mathbf{(((a \bmod p) + (b \bmod p)) \bmod p, ((a \bmod q) + (b \bmod q)) \bmod q)} \\ & \quad \parallel \\ & \mathbf{(a \bmod p, a \bmod q) + (b \bmod p, b \bmod q)} \\ & \quad \parallel \\ & \mathbf{f(a) + f(b)} \end{aligned}$$

# **f** is a homomorphism

**f**:  $Z_N^*$   $\rightarrow$   $Z_p^*$   $\times$   $Z_q^*$  is an **homomorphism**

Proof:

$$\begin{aligned} & \mathbf{f(a \cdot b)} \\ & \quad \parallel \\ & \mathbf{(a \cdot b \bmod p, a \cdot b \bmod q)} \\ & \quad \parallel \\ & \mathbf{(((a \bmod p) \cdot (b \bmod p)) \bmod p, ((a \bmod q) \cdot (b \bmod q)) \bmod q)} \\ & \quad \parallel \\ & \mathbf{(a \bmod p, a \bmod q) \cdot (b \bmod p, b \bmod q)} \\ & \quad \parallel \\ & \mathbf{f(a) \cdot f(b)} \end{aligned}$$

# An example

$Z_{15}$ :

<b>i</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
<b>i mod 5</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>i mod 3</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>2</b>

		<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>0</b>		<b>0</b>	<b>6</b>	<b>12</b>	<b>3</b>	<b>9</b>
<b>1</b>	<b>i mod 3</b>	<b>10</b>	<b>1</b>	<b>7</b>	<b>13</b>	<b>4</b>
<b>2</b>		<b>5</b>	<b>11</b>	<b>2</b>	<b>8</b>	<b>14</b>

# By the way: it's not always like this!

Consider  $p = 4$  and  $q = 6$ :

$Z_{24}$ :

	$i \bmod 6$					
	0	1	2	3	4	5
$i \bmod 4$	0	1	2	3	4	5
0	0,12		8,20		4,16	
1		1,13		9,21		5,17
2	6,18		2,14		10,22	
3		7,19		3,15		11,23

If  $p$  and  $q$  are prime then  
 $f : \mathbb{Z}_N \rightarrow \mathbb{Z}_p \times \mathbb{Z}_q$  is a bijection

$$f(x) := (x \bmod p, x \bmod q)$$

because  $p$  and  $q$  are prime

**Proof:**

We first show that it is injective.

If  $f(i) = f(j)$  then

$$\begin{array}{l} i \bmod p = j \bmod p \rightarrow p \text{ divides } i-j \\ \text{and } i \bmod q = j \bmod q \rightarrow q \text{ divides } i-j \end{array} \left. \vphantom{\begin{array}{l} i \bmod p = j \bmod p \\ \text{and } i \bmod q = j \bmod q \end{array}} \right\} \rightarrow n=pq \text{ divides } i-j$$
$$\downarrow$$
$$i = j \bmod n$$

Since  $|\mathbb{Z}_N| = N = pq = |\mathbb{Z}_p \times \mathbb{Z}_q|$  we are done!

QED

$f : \mathbb{Z}_N^* \rightarrow \mathbb{Z}_p^* \times \mathbb{Z}_q^*$  is also a bijection

Since we have shown that  $f$  is injective it is enough to show that

$$|\mathbb{Z}_N^*| = |\mathbb{Z}_p^*| \times |\mathbb{Z}_q^*|$$

$$= (p-1)(q-1)$$

Look at  $\mathbb{Z}_{15}^*$ :

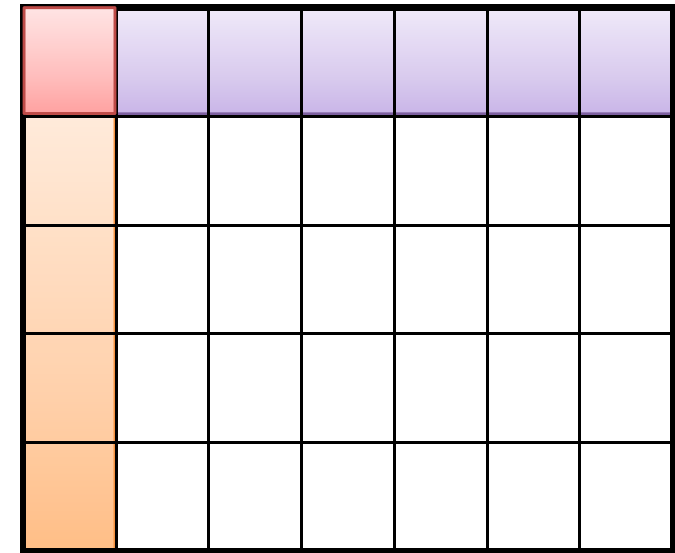
	0	1	2	3	4
0	0	6	12	3	9
1	10	1	7	13	4
2	5	11	2	8	14

$$N = pq$$

Which elements of  $\mathbb{Z}_N$  are not in  $\mathbb{Z}_N^*$ ?

- 0
- multiples of  $p$ :  
 $\{p, \dots, (q-1)p\}$   
(there are  $q-1$  of them)
- multiples of  $q$ :  
 $\{q, \dots, (p-1)q\}$   
(there are  $p-1$  of them).

These sets are disjoint since  $p$  and  $q$  are distinct prime



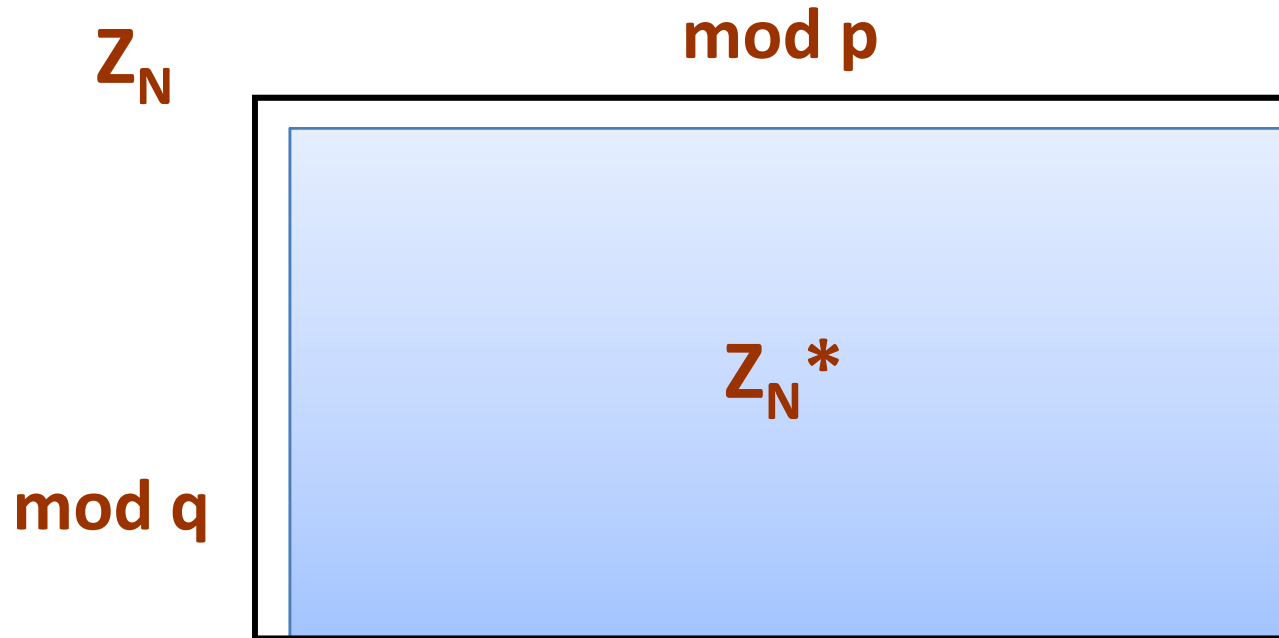
- Summing it up:  
 $1 + (q - 1) + (p - 1) = q + p - 1$

$$\begin{aligned} &= pq - p - q + 1 \\ &= (p - 1)(q - 1) \end{aligned}$$

So  $\mathbb{Z}_N^*$  has  $pq - (q + p - 1)$  elements.

QED

# How does it look for large $p$ and $q$ ?



$pq$  is called **RSA modulus**  
 $Z_N^*$  is called an **RSA group**

technical assumption:  $p \neq q$

we will often forget to mention it  
(since for large  $p$  and  $q$  the  
probability that this  $p = q$  is  
negligible)

How to compute  $\phi(N)$ , where  $N = pq$ ?

Of course if  $p$  and  $q$  are known then it is easy to compute  $\phi(N)$ , since

$$\phi(N) = (p-1)(q-1).$$

Hence, computing  $\phi(N)$  cannot be harder than factoring.

**Fact**

Computing  $\phi(N)$  is as hard as factoring  $N$ .

# Computing $\phi(N)$ is as hard as factoring $N$ .

Suppose we can compute  $\phi(N)$ . We know that

$$\begin{cases} (p-1)(q-1) = \phi(N) & \text{(1)} \\ pq = N & \text{(2)} \end{cases}$$

It is a system of **2** equations with **2** unknowns (**p** and **q**).

We can solve it:

(2)  $p = N/q$

(1)  $(N/q - 1)(q - 1) = \phi(N)$

it is a quadratic equation so we can solve it (in  $\mathbb{R}$ )

$$q^2 + (\phi(N) - N - 1)q + N = 0$$

Which problems are easy and which are hard in  $\mathbb{Z}_N^*$  ( $N = pq$ )?

- multiplying elements?

easy!

- finding inverse?

easy! (Euclidean algorithm)

- computing  $\phi(N)$  ?

hard! - as hard as factoring  $N$

- raising an element to power  $e$  (for a large  $e$ )?

easy!

- computing  $e$ th root (for a large  $e$ )?

# Computing $e$ th roots modulo $N$

In other words, we want to invert a function:

$$f : \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$$

defined as

$$f(x) = x^e \bmod N.$$

This is possible only if  $f$  is a permutation.

## Lemma

$f$  is a permutation if and only if  $\gcd(e, \phi(N)) = 1$ .

In other words:  $e \in \mathbb{Z}_{\phi(N)}^*$  (note: a “new” group!)

“ $f(x) = x^e \bmod N$  is a permutation if and only if  $\gcd(e, \phi(N)) = 1$ .”

1.

$$\gcd(e, \phi(n)) = 1$$



$f(x) = x^e \bmod N$  is a permutation

Let  $d$  be an inverse of  $e$  in  $\mathbb{Z}_{\phi(N)}^*$ . That is:  
 $d$  is such that  $d \cdot e = 1 \bmod \phi(N)$ .

Then:

$$f^d(x) = (x^e)^d = x^{ed} = x^{ed \bmod \phi(N)} = x^1$$

2.

$$\gcd(e, \phi(n)) = 1$$



$f(x) = x^e \bmod N$  is a permutation

[exercise]

# Computing $e$ th root – easy, or hard?

Suppose  $\gcd(e, \phi(N)) = 1$

We have shown that the function

$$f(x) = x^e \bmod N \text{ (defined over } \mathbb{Z}_N^*)$$

has an inverse

$$f^{-1}(x) = x^d \bmod N, \text{ where } d \text{ is an inverse of } e \text{ in } \mathbb{Z}_{\phi(N)}^*$$

**Moral:**

If we know  $\phi(N)$  we can compute the roots efficiently.

What if we don't know  $\phi(N)$ ?

Can we compute the **e**th root if we do not know  **$\phi(N)$** ?

It is conjectured to be hard.

This conjecture is called an **RSA assumption**. More precisely:

**RSA assumption**

For any randomized polynomial time algorithm **A** we have:

**$P((A(x,N,e))^e = x \bmod N)$  is negligible**

where  **$N = pq$**  where **p** and **q** are random primes such that  **$|p| = |q|$** , and **x** is a random element of  **$Z_N^*$** , and **e** is random element of  **$Z_{\phi(N)}^*$**

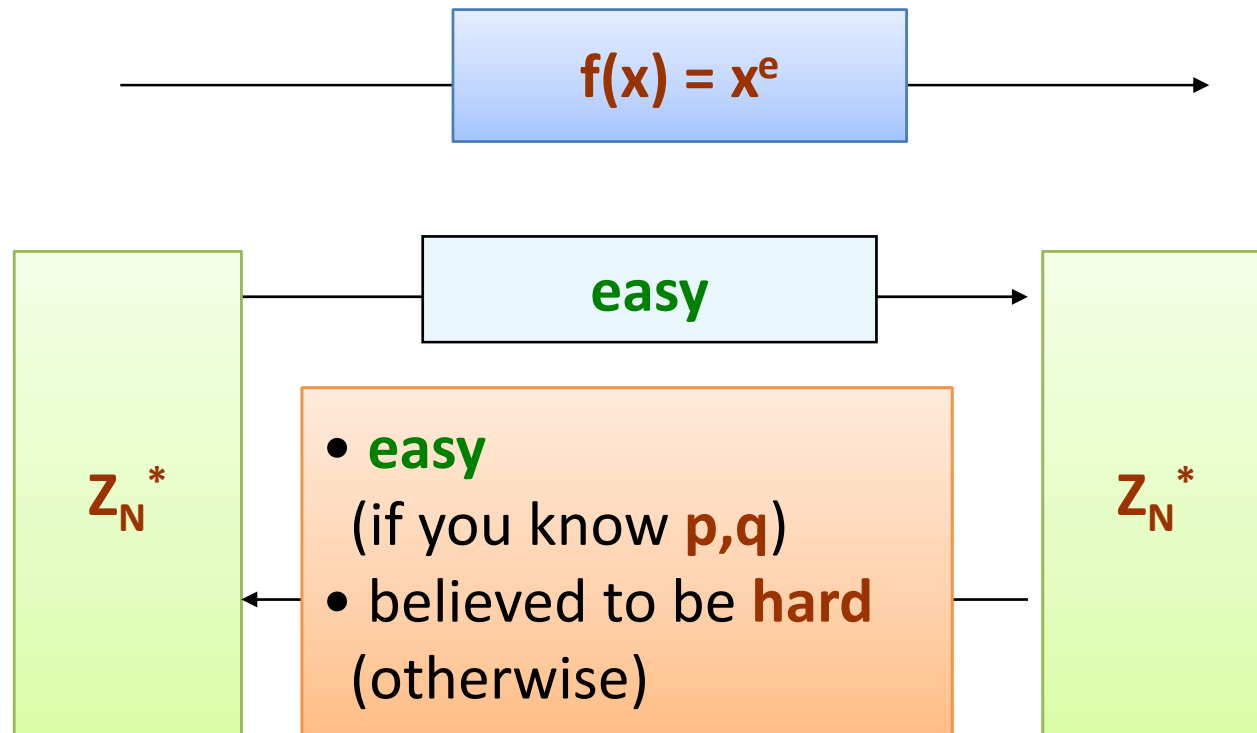
# What can be shown?

Does the **RSA assumption** follow from the assumption that factoring is hard?

We don't know...

What **can** be shown is that

**computing  $d$  from  $e$  is not easier than factoring  $N$ .**

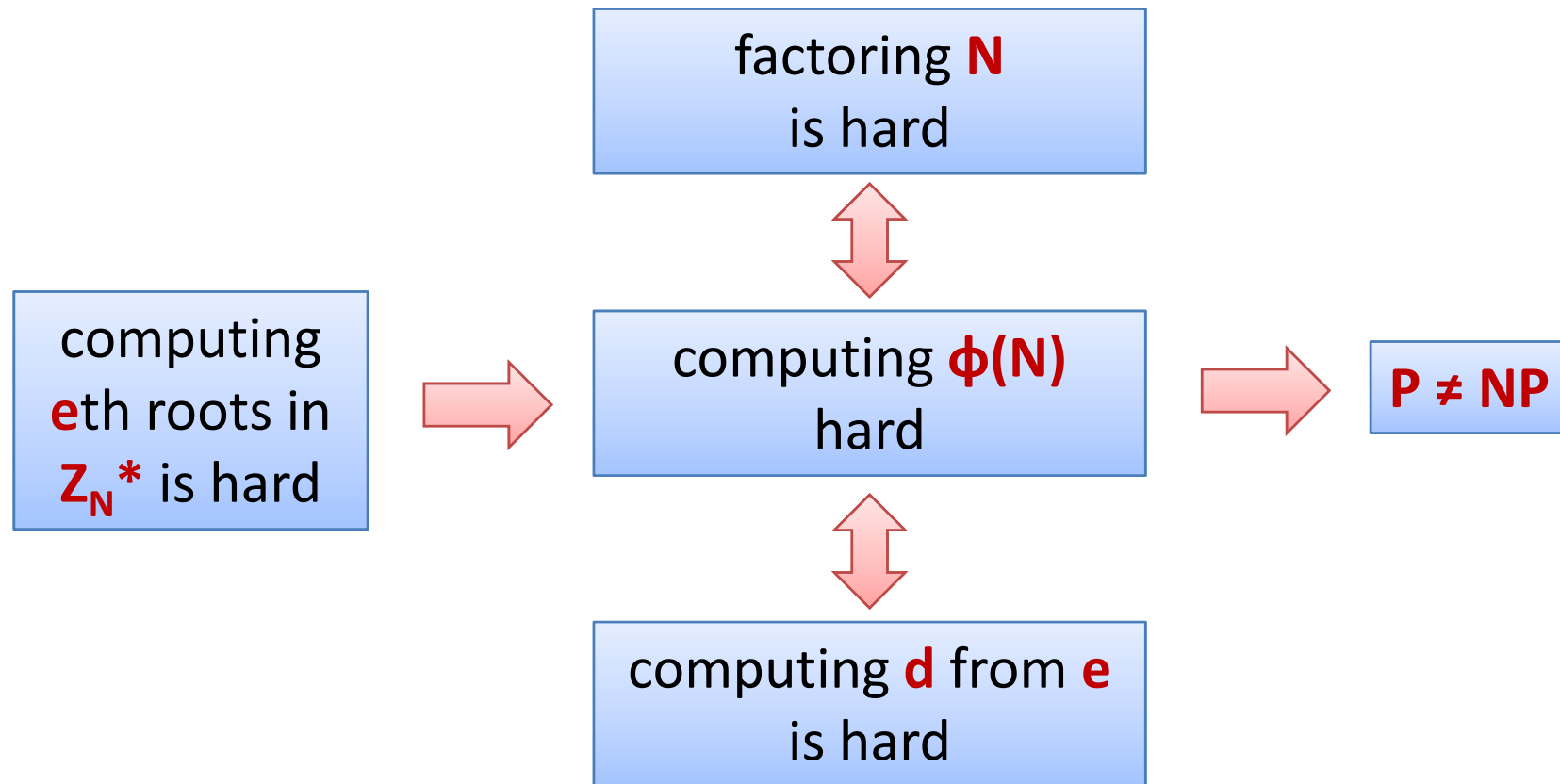


Functions like this are called **trap-door one-way permutations**.

**f** is called an **RSA function** and is extremely important.

# Outlook

**N** – a product of two large primes



# Square roots modulo $N=pq$

So, far we discussed a problem of computing the  $e$ th root modulo  $N$ .

What about the case when  $e = 2$ ?

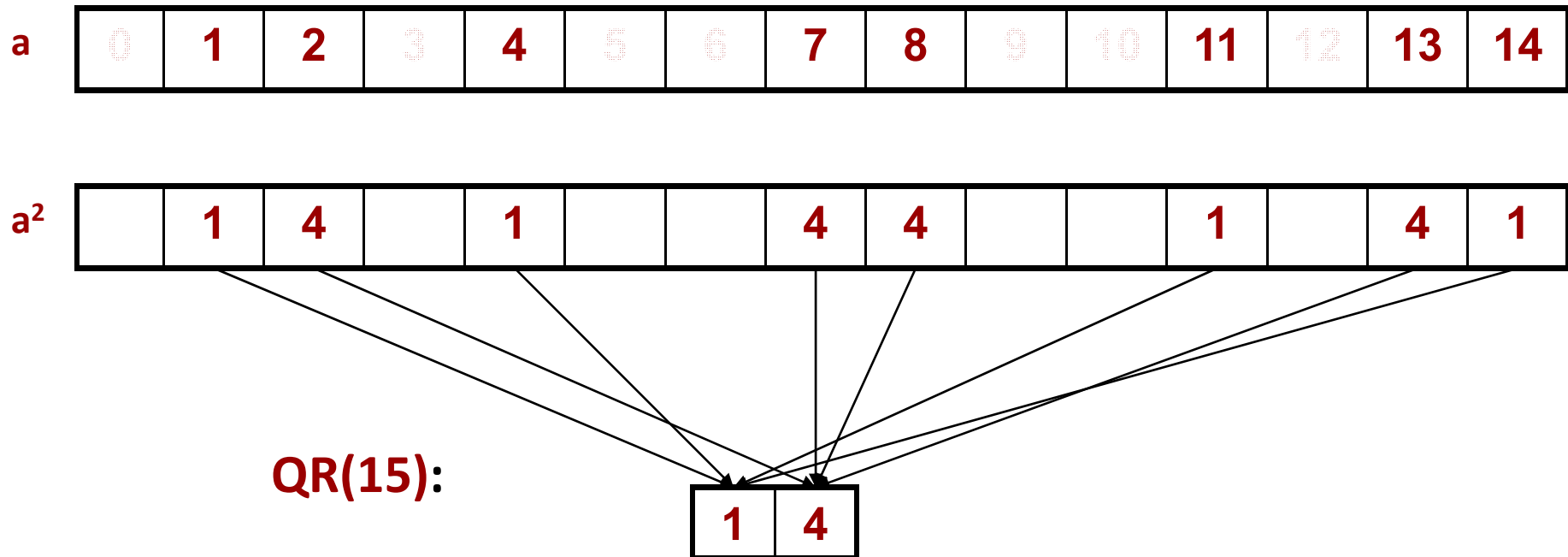
Clearly  $\gcd(2, \phi(N)) \neq 1$ , so  $f(x) = x^2$  is not a bijection.

## Question

Which elements have a square root modulo  $N$ ?

# Quadratic Residues modulo $pq$

$Z_{15}^*$ :



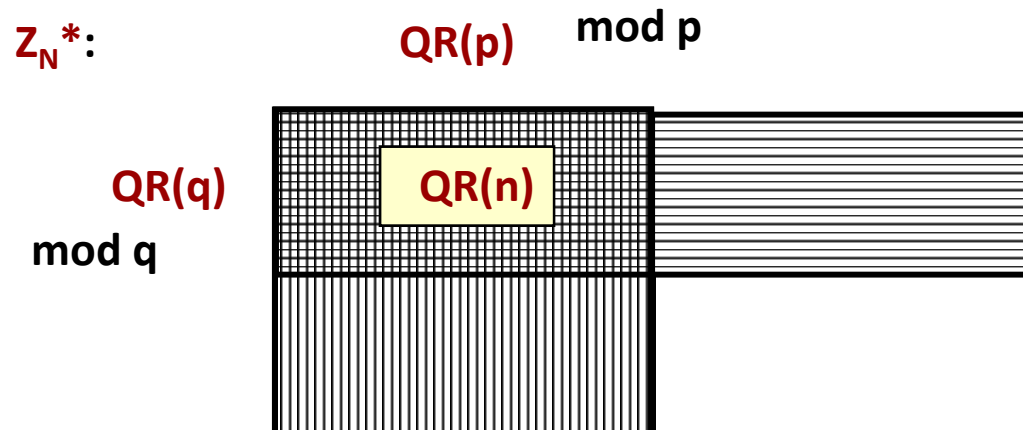
**Observation:** every quadratic residue modulo **15** has **exactly 4** square roots, and hence  $|QR(15)| = |Z_{15}^*| / 4$ .

# A lemma about QRs modulo $pq$

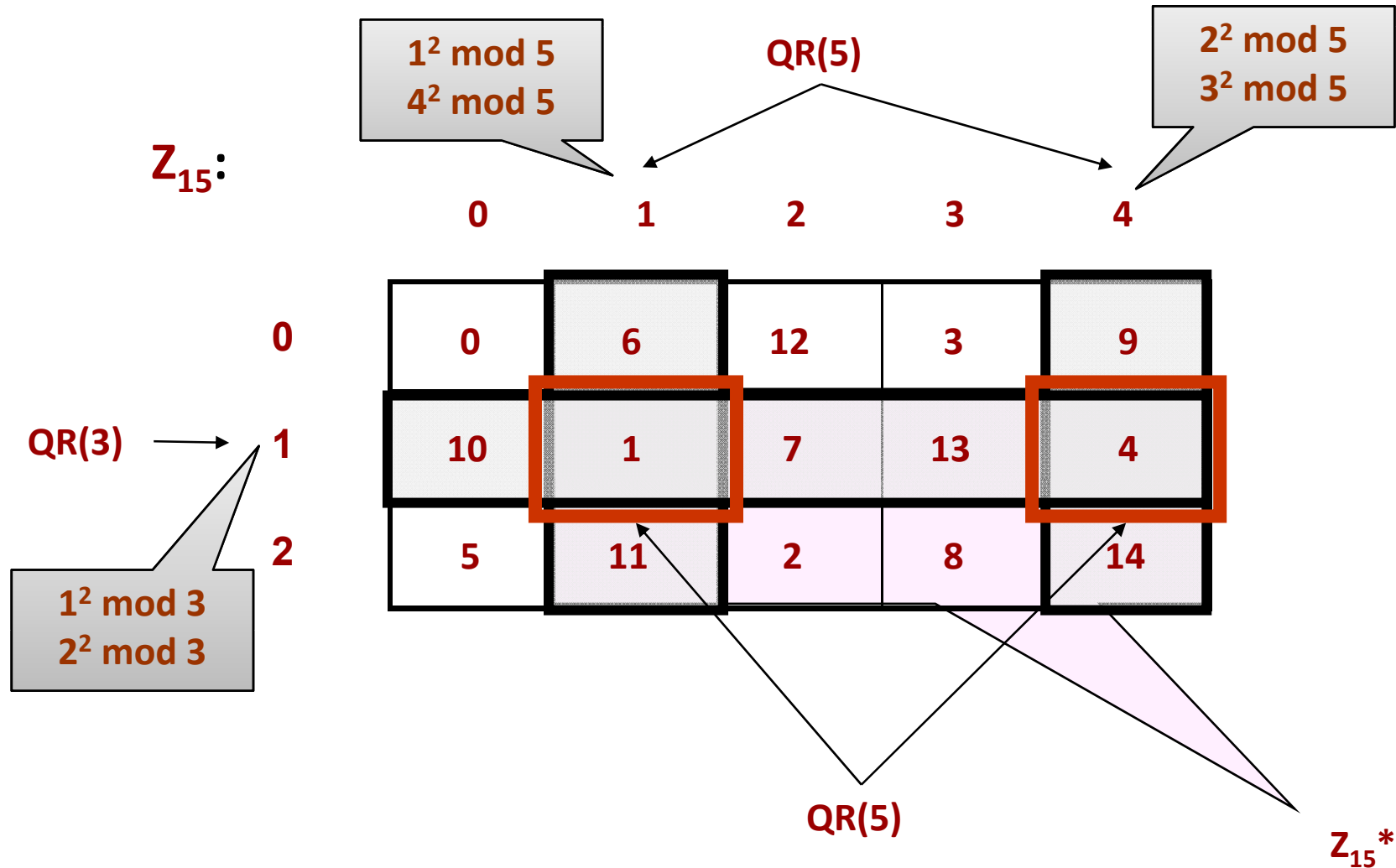
**Fact:** For  $N=pq$  we have  $|QR(N)| = |Z_N^*| / 4$ .

**Proof:**

$$\begin{aligned}
 & x \in QR(N) \\
 & \text{iff} \\
 & x = a^2 \bmod n, \text{ for some } a \\
 & \text{iff (by CRT)} \\
 & x = a^2 \bmod p \text{ and } x = a^2 \bmod q \\
 & \text{iff} \\
 & x \bmod p \in QR(p) \text{ and } x \bmod q \in QR(q)
 \end{aligned}$$



# QRs modulo $pq$ – an example



Every  $x \in \mathbb{QR}_N$  has exactly 4 square roots

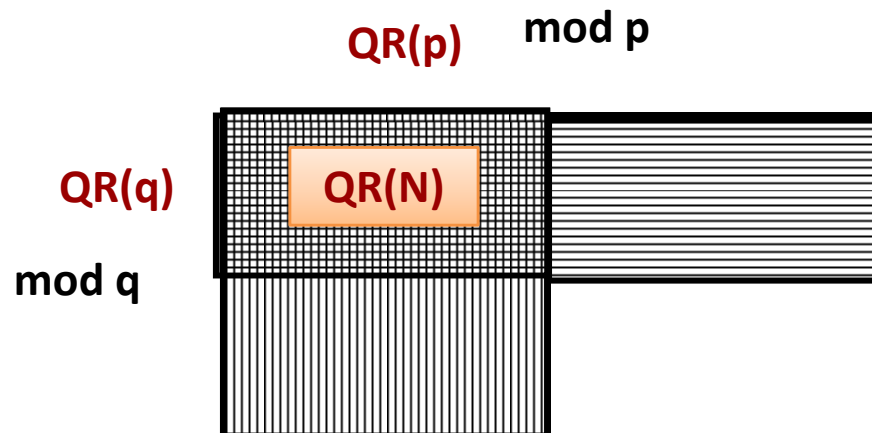
More precisely, every  $z = x_1^2$  has the square roots  $x_1$  and  $x_2, x_3, x_4$  such that:

- $x_2 = x_1 \pmod{p}$  and  $x_2 \neq x_1 \pmod{q}$
- $x_3 = -x_1 \pmod{p}$  and  $x_3 = x_1 \pmod{q}$
- $x_4 = -x_1 \pmod{p}$  and  $x_4 = -x_1 \pmod{q}$

# Jacobi Symbol

for any prime  $p$  define  $J_p(x) := \begin{cases} +1 & \text{if } x \in \text{QR}_p \\ -1 & \text{otherwise} \end{cases}$

for  $N=pq$  define  $J_N(x) := J_p(x) \cdot J_q(x)$



$J_N(x) :=$

+1	-1
-1	+1

It is a subgroup of  $\mathbb{Z}_N^*$

$$\mathbb{Z}_N^+ := \{x : J_n(x) = +1\}$$

**Jacobi symbol can be computed efficiently!** (even in  $p$  and  $q$  are unknown)

# Algorithmic questions about QR

Suppose  $N=pq$

Is it easy to test membership in  $QR(N)$ ?

**Fact**: if one knows  $p$  and  $q$  – yes!

**Because:**

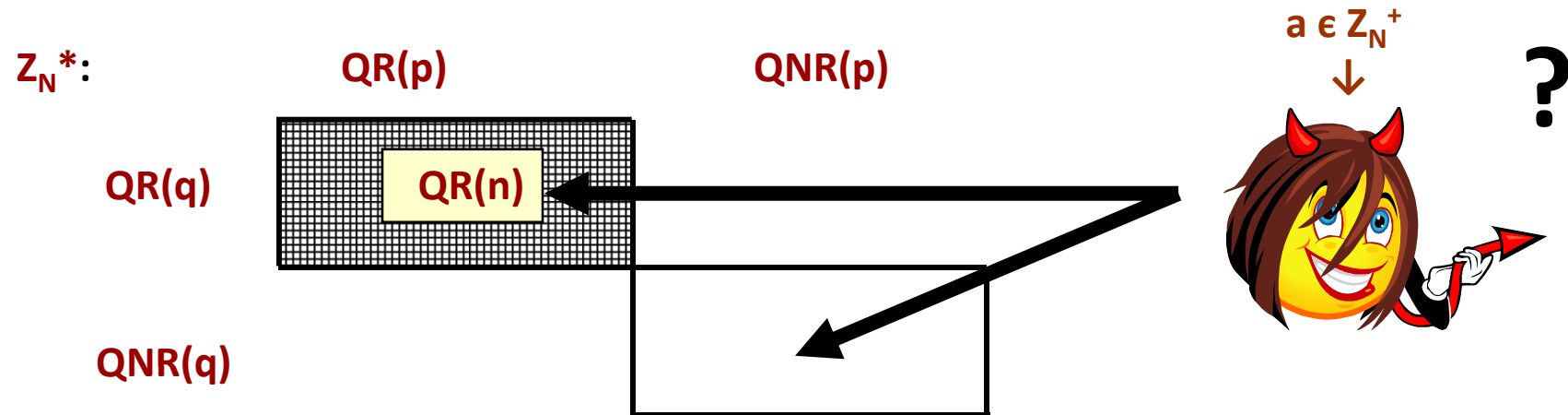
1. testing membership modulo a prime is easy
2. the “CRT function”

$$f(x) := (x \bmod p, x \bmod q)$$

can be efficiently computed in both directions

What if one doesn't know  $p$  and  $q$ ?

# Quadratic Residuosity Assumption



$Q(N,a) = 1$  if  $a \in QR(N)$   
 $Q(N,a) = 0$  otherwise

## Quadratic Residuosity Assumption (QRA):

For a random  $a \in Z_N^+$  it is computationally hard to determine if  $a \in QR(N)$ .

**Formally:** for every polynomial-time probabilistic algorithm  $D$  the value:

$$|P(D(N,a) = Q(N,a)) - 0.5|$$

(where  $a \leftarrow Z_N^+$ ) is negligible.

# So, how to compute a square root of

$$x \in QR_N ?$$

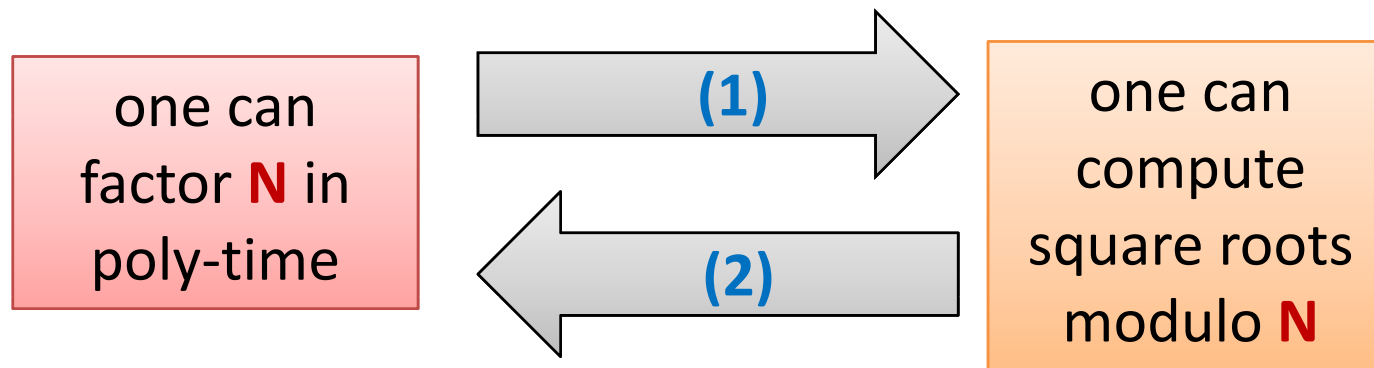
## Fact

Let  $N$  be a random **RSA** modulus.

The problem of computing square roots (modulo  $N$ ) of random elements in  $QR_N$  is poly-time equivalent to the problem of factoring  $N$ .

## Proof

We need to show that:



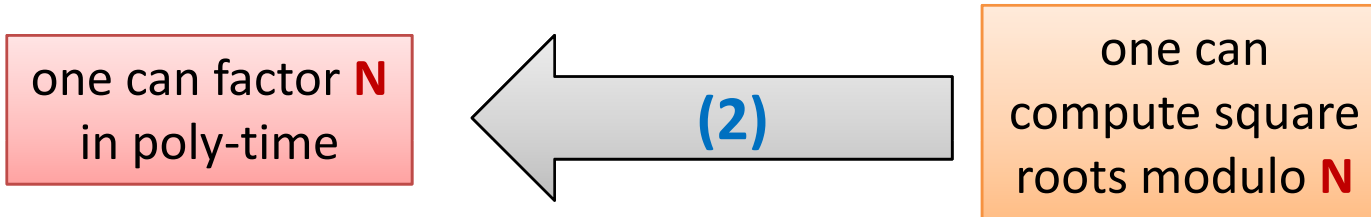


From CRT it is enough to show how to compute square roots modulo a prime  $p$ .

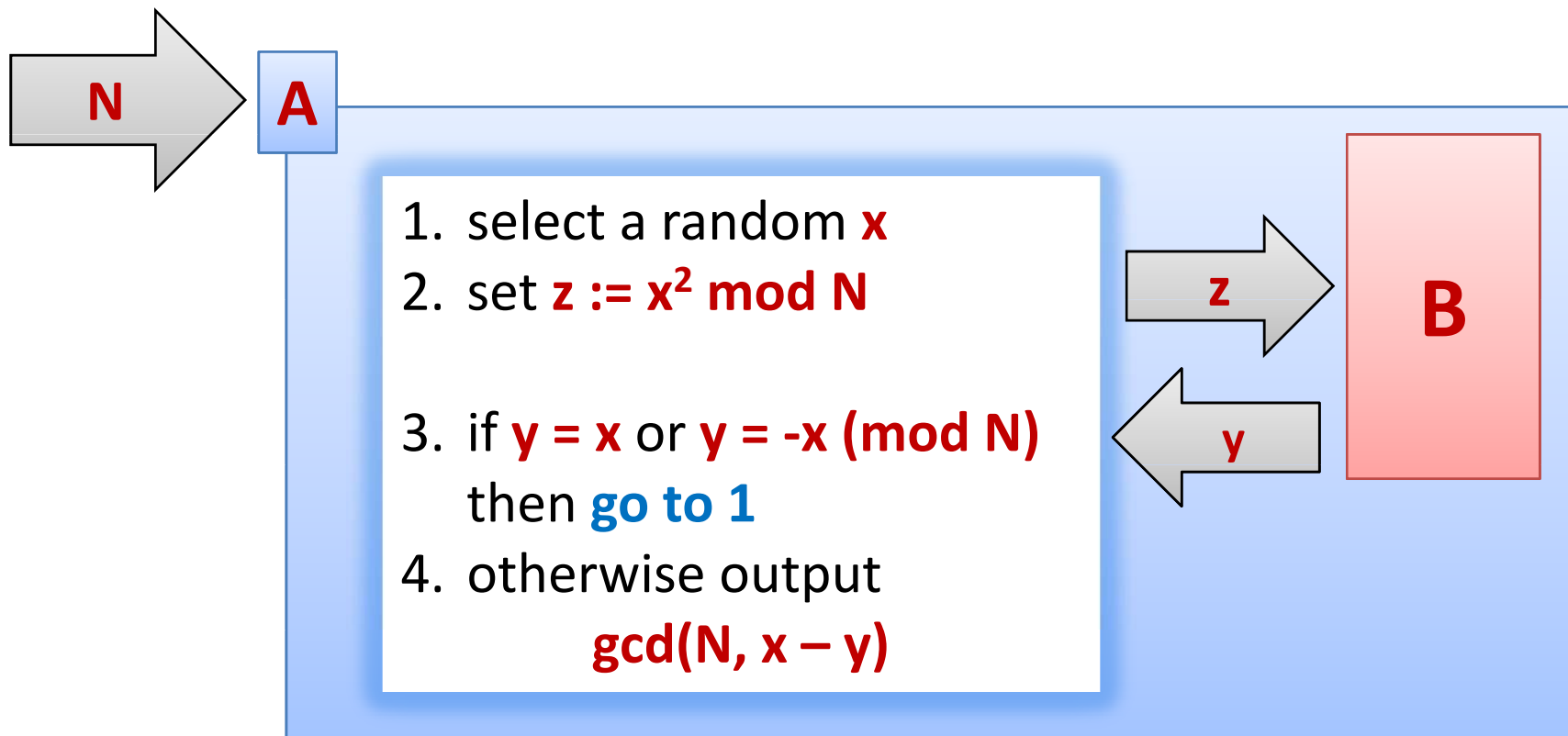
We show it only for  $p = 3 \pmod{4}$  (for  $p = 1 \pmod{3}$  this fact also holds, but the proof is more involved).

Write  $p = 4m + 3$ . We have (all computations are  $\pmod{p}$ ):

$$\begin{aligned}
 \mathbf{1} = \mathbf{x}^{\frac{p-1}{2}} &\longrightarrow \mathbf{x} = \mathbf{x}^{\frac{p-1}{2}+1} \\
 &= \mathbf{x}^{\frac{4m+2}{2}+1} \\
 &= \mathbf{x}^{2m+1+1} \\
 &= \mathbf{x}^{2(m+1)} \\
 &= (\mathbf{x}^{m+1})^2 \longrightarrow \mathbf{x}^{m+1} \text{ is a square root of } \mathbf{x}
 \end{aligned}$$



Suppose we have an algorithm **A** that computes the square roots. We construct an algorithm **B** that factors **N**.



To complete the proof we show that:

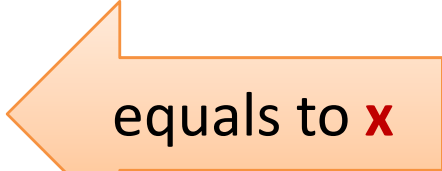
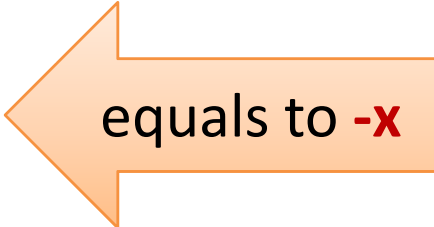
1. the probability that  $y = x$  or  $y = -x$  is equal to **0.5**,

2. If  $y \neq x$  and  $y \neq -x$  then

$$\gcd(N, x - y) > 1.$$

“the probability  $\pi$  that  $y = x$  or  $y = -x$  is equal to **0.5**”

Recall that the square roots  $x_1, x_2, x_3, x_4$  of every  $z = x^2$  are such that:

- $x_1 = x \pmod{p}$  and  $x_1 = x \pmod{q}$   equals to  $x$
- $x_1 = x \pmod{p}$  and  $x_1 = -x \pmod{q}$
- $x_1 = -x \pmod{p}$  and  $x_1 = x \pmod{q}$
- $x_1 = -x \pmod{p}$  and  $x_1 = -x \pmod{q}$   equals to  $-x$

Since  $x$  is chosen randomly the probability  $\pi$  is equal to **0.5**.

“Suppose that  $y \neq x$  and  $y \neq -x$ .  
Then  $\gcd(N, x - y) > 1$ ”

We know that  $y$  is such that

- $y = x \pmod{p}$  and  $y = -x \pmod{q}$ , or
- $y = -x \pmod{p}$  and  $y = x \pmod{q}$ .

Hence  $y \neq x \pmod{N}$ , and therefore  $y - x \neq 0 \pmod{N}$ .

On the other hand:

$$y - x = 0 \pmod{p} \text{ or } y - x = 0 \pmod{q}.$$

Therefore

$$\gcd(y-x, N) = p \text{ or } \gcd(y-x, N) = q.$$

QED

# Outlook

Groups that we have seen:

- $Z_p^*$

hard problem:  
discrete log

- $Z_N^*$  for  $N=pq$

hard problem:  
computing the  $e$ th root

- subgroups:  $QR_p$  and  $QR_N$

©2009 by Stefan Dziembowski. Permission to make digital or hard copies of part or all of this material is currently granted without fee *provided that copies are made only for personal or classroom use, are not distributed for profit or commercial advantage, and that new copies bear this notice and the full citation.*