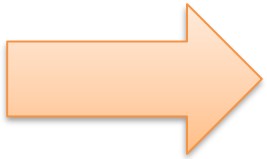
Lecture 7
A Brush-up on Number Theory and Algebra

Stefan Dziembowski
www.crypto.edu.pl/Dziembowski

University of Warsaw

14.11.18 **version 1.0**

Plan



1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Group Z_N^* and its subgroups
6. Elliptic curves

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"**).

<u>Additional advantage</u>

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

Number theory in cryptography - disadvantages

1. cryptography based on number theory is much **less efficient**!

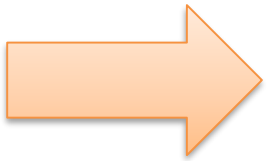
2. the number-theoretic "structure" may help the cryptoanalyst...

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. Group $\boldsymbol{Z_N^*}$ and its subgroups
6. Elliptic curves

Famous algorithmic problems in number theory

primality testing:

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

output:

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

this problem is **computationally easy**

factoring:

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

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 $|x| = \log_2 x$

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 **¼**.

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 et al.** (**2002**)
polynomial but **very inefficient in practice**

How to select a **random** prime of length ***n***?

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

Prime Number Theorem

Let

$$\pi(x) := \text{number of n's such that } n \in \{1,\ldots,x\} \text{ and } n \text{ is prime}$$

Then

$$\lim_{x\to\infty} \frac{\pi(x)}{x/\ln(x)} = 1$$

$$\pi(x) \approx \frac{x}{\ln(x)}$$

For example if $x = 2^{1000}$ then $\pi(x)/x \approx 0.0014$

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...

Notation

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

$a \mid b$:

- **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 divisor** of **a** and **b**"

lcm(a,b) = "the **least common multiple** of **a** and **b**"

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

How to compute **gcd(*a*,*b*)**?

Euclidean algorithm

Recursion:

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

gcd(*a*,*b*) = if *b* | *a*

then return ***b***

else return **gcd(*b*, *a* mod *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

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

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

Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Group Z_N^* and its subgroups
6. Elliptic curves

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**

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,
$g + \cdots + g$ (n times) is denoted with ng.

[**multiplicative notation**]
If the groups operation is denoted "$\times$" or "$\cdot$", then:
sometimes we write gh instead of $g \cdot h$,
the inverse of g is denoted g^{-1} or $1/g$.
the neutral element is denoted with 1,
$g \cdot \cdots \cdot g$ (n times) is denoted with g^n
$(g^{-1})^n$ is denoted with g^{-1}.

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**

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 $h \in H$.
- **$(g,h) \bullet (g',h') = (g \circ g', h \square h')$**.

It is easy to verify that it is a group.

Examples of groups

- **R** (reals) is not a group with multiplication.
- **R \ {0}** is a group with multiplication.
- **Z** (integers):
  - is a group under **addition** (identity element: **0**),
  - is **not** a group under **multiplication**.
- $Z_N = \{0,\ldots,N-1\}$ (integers modulo N) is a group under **addition modulo** N (identity element: **0**)
- If p is a prime then $Z_p^* = \{1, \ldots, p-1\}$ is a group under **multiplication modulo** p (identity element: **1**)
  (we will discuss it later)

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 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	**1**	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	**1**	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	**1**	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	**1**

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:

$$ab = nk + 1$$

$$\downarrow$$

$$ab - nk = 1$$

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

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

QED

Z_N^*

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

Then Z_N^* is an abelian group under multiplication modulo N.

Proof

First observe that Z_N^* is **closed under multiplication** modulo N.

This is because is a and 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 $a \in Z_N^*$ there exist $b \in Z_N^*$ that is an **inverse of a modulo N**.

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

$$a \cdot b = 1 \bmod N$$

Lemma
Suppose that $\gcd(a,N) = 1$. Then for every $a \in Z_N^*$ there always exist an element $X \in Z$ such that

$$X \cdot a \bmod N = 1.$$

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

$$Xa + YN = 1.$$

Therefore $Xa = 1 \pmod N$.

QED

Observation
Such an X can be efficiently computed (using the **extended Euclidian algorithm**).

What remains?

X (from the previous lemma) can be such that

$$X \notin Z_N^*$$

What to do?

define $b := X \bmod N$

we need to show that

$$a \cdot b = 1 \bmod N$$

This will imply that $b \in Z_N^*$ because if $a \cdot b = 1 \bmod N$ then $\gcd(b,N)=1$

If

$b := X \bmod N$

then $b = X + tN$

So

$$a\,b = a \cdot (X + tN)$$
$$= aX + atN$$
$$= 1 \pmod N$$

Remember that X is such that

$aX \bmod N = 1.$

Hence we are done!

An example

$\boldsymbol{p}$ – a prime

$$\boldsymbol{Z_p^* := \{1, \ldots, p-1\}}$$

$\boldsymbol{Z_p^*}$ is an abelian group under multiplication modulo $\boldsymbol{p}$.

A simple observation

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

$$ac = bc$$

then

$$a = b.$$

Corollary

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

$$f : G \to G$$

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

is a bijection.

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

Example: Z_{11}^*

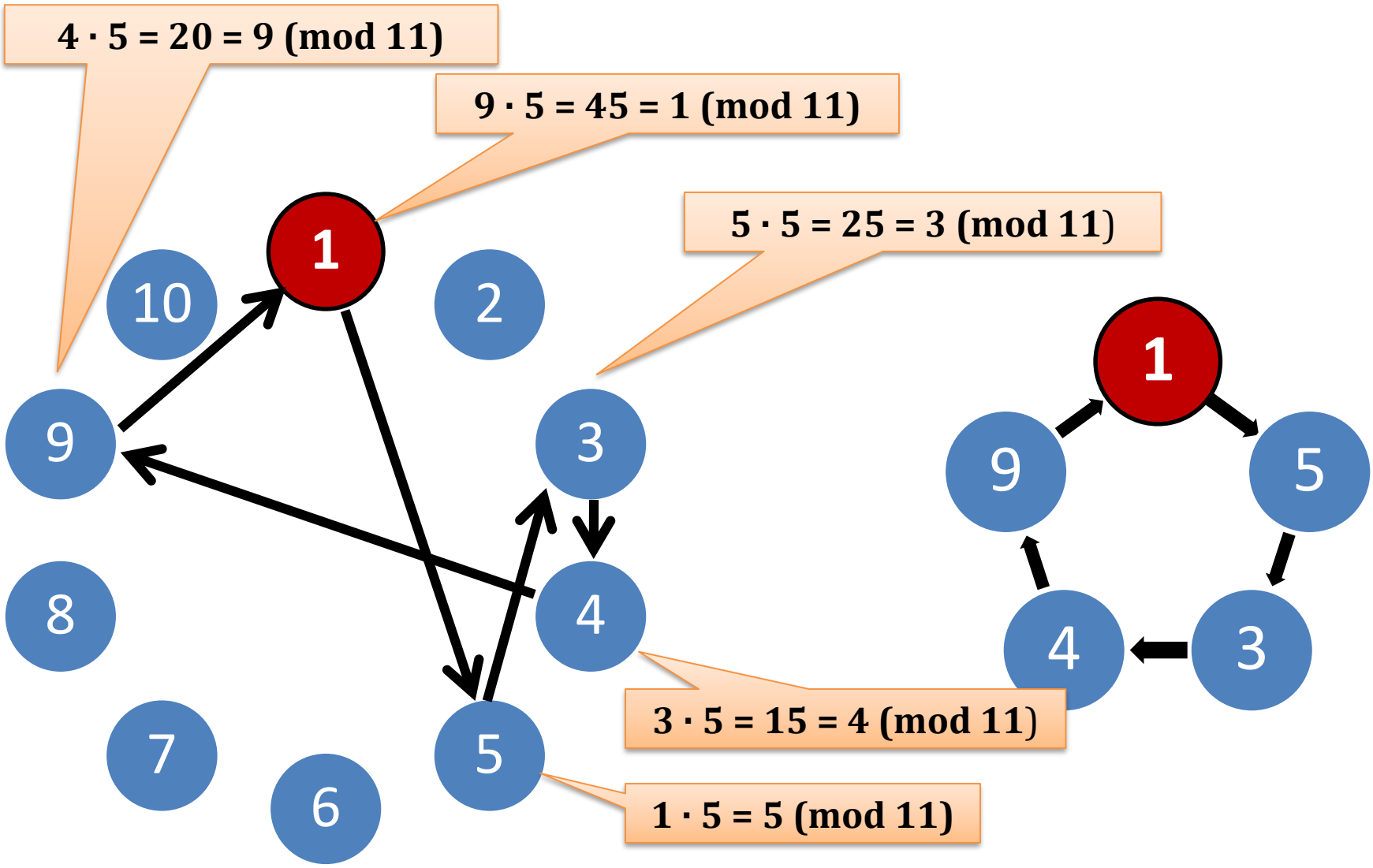
x	1	2	3	4	5	6	7	8	9	10
f(x)	5	10	4	9	3	8	2	7	1	6

$f(x) = 5 \cdot x \bmod 11$

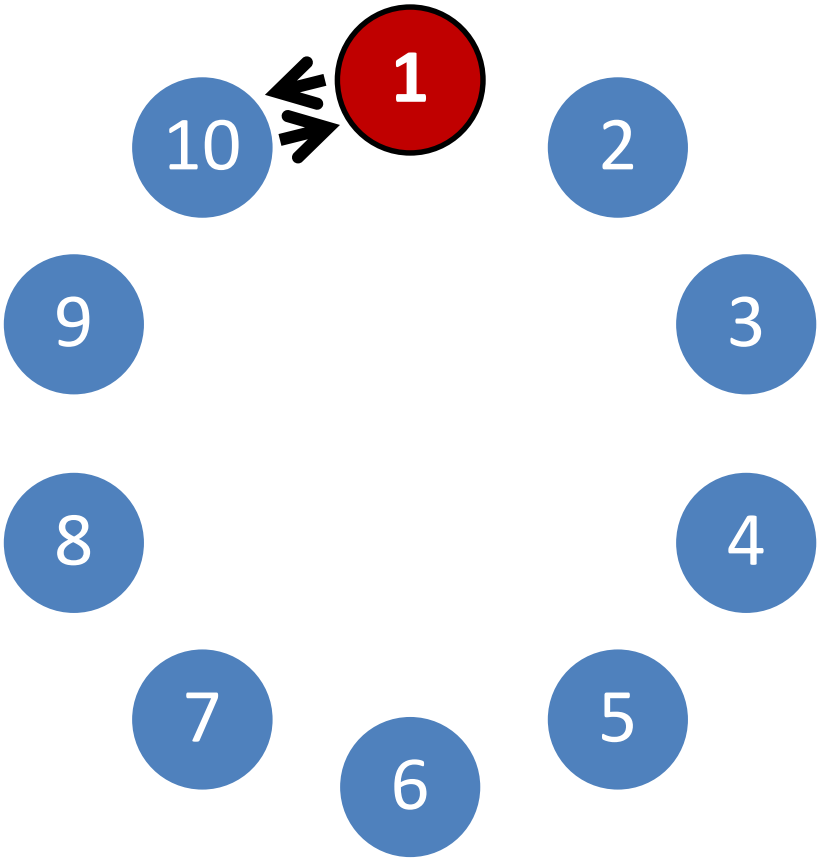
Permutations have cycles.

Let's look now at the cycles that contain **1**!

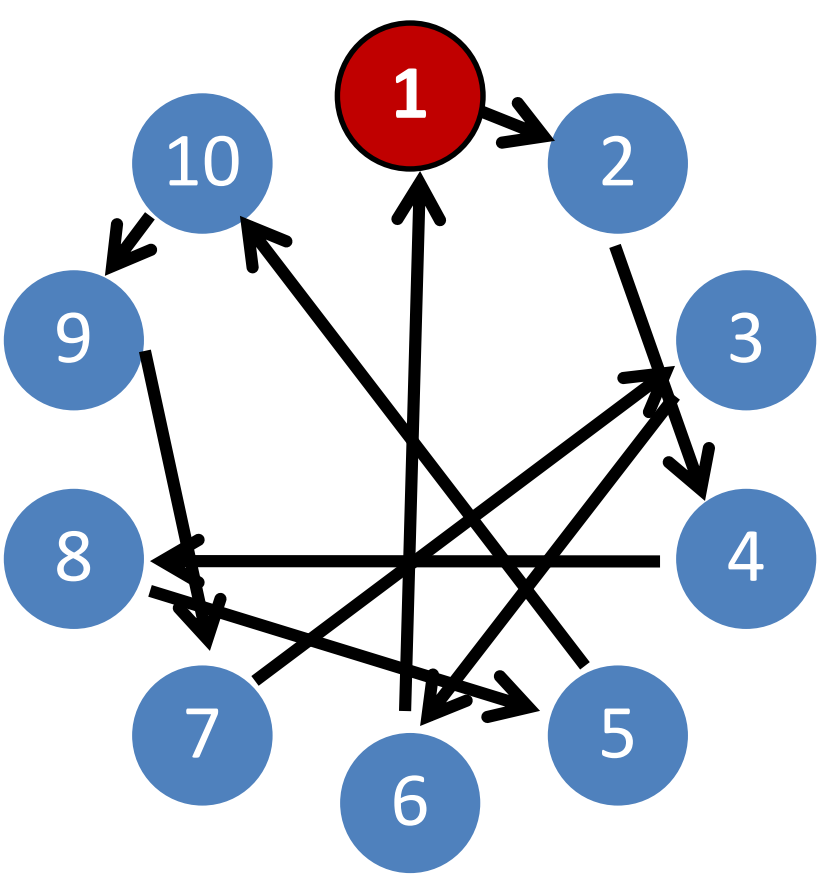
Example: $f(x) = 5 \cdot x \bmod 11$

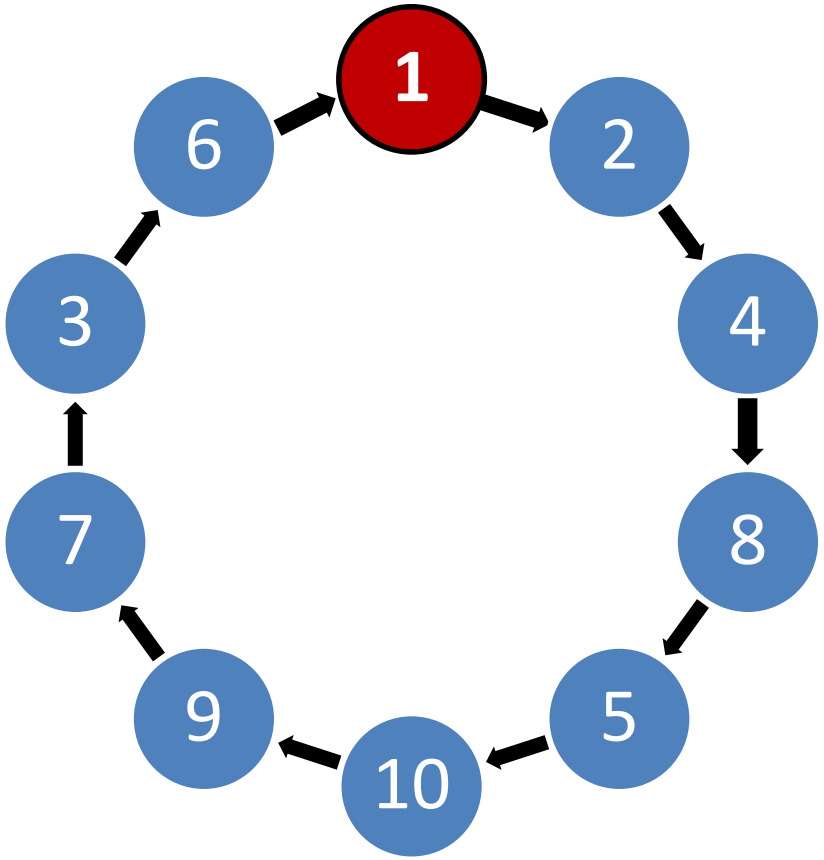


Example: $f(x) = 10 \cdot x \bmod 11$



Example: $f(x) = 2 \cdot x \bmod 11$





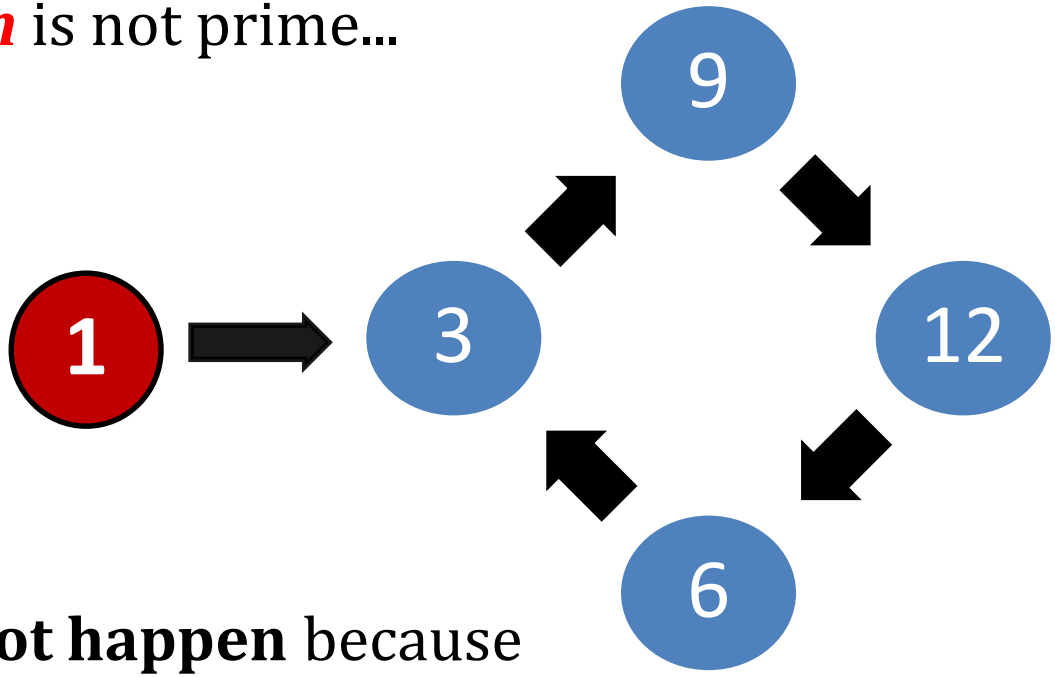
It has to be a cycle!

If we do it in Z_n^*, where n is not prime...

for example:

$n = 15$

$g = 3$



If n is a prime this **cannot happen** because

$f(x) = x \cdot g \bmod n$

is a **permutation**

so we cannot have

$f(x_1) = f(x_2)$

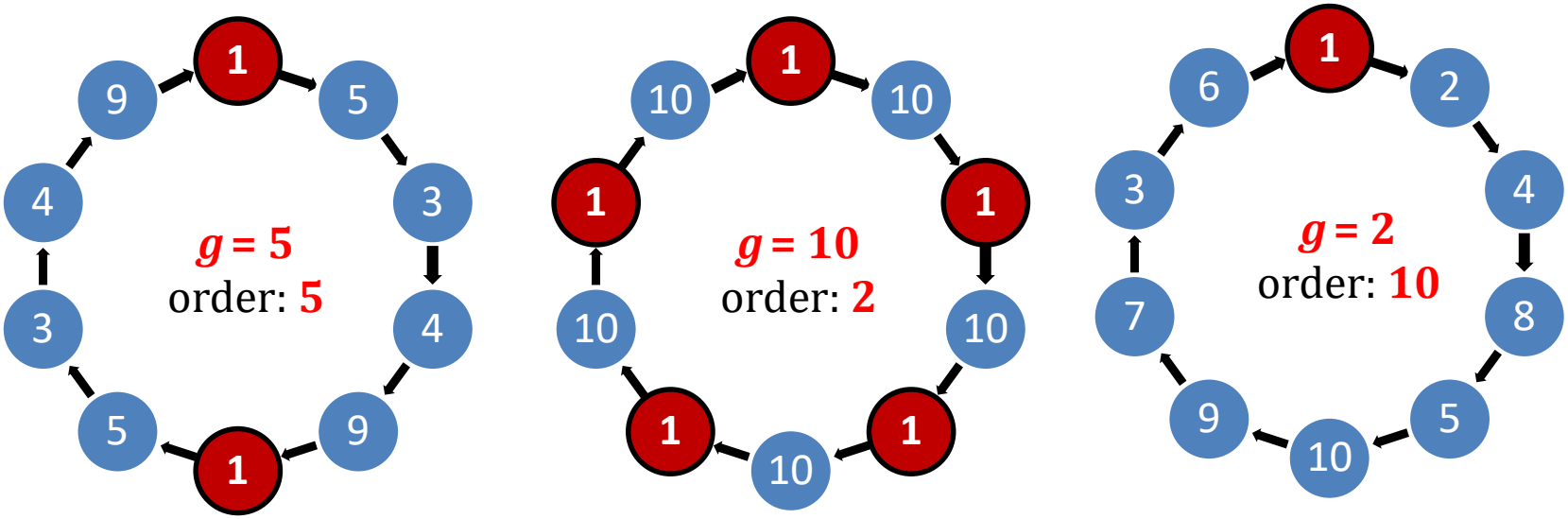
for $x_1 \neq x_2$

Order of an element

Definition

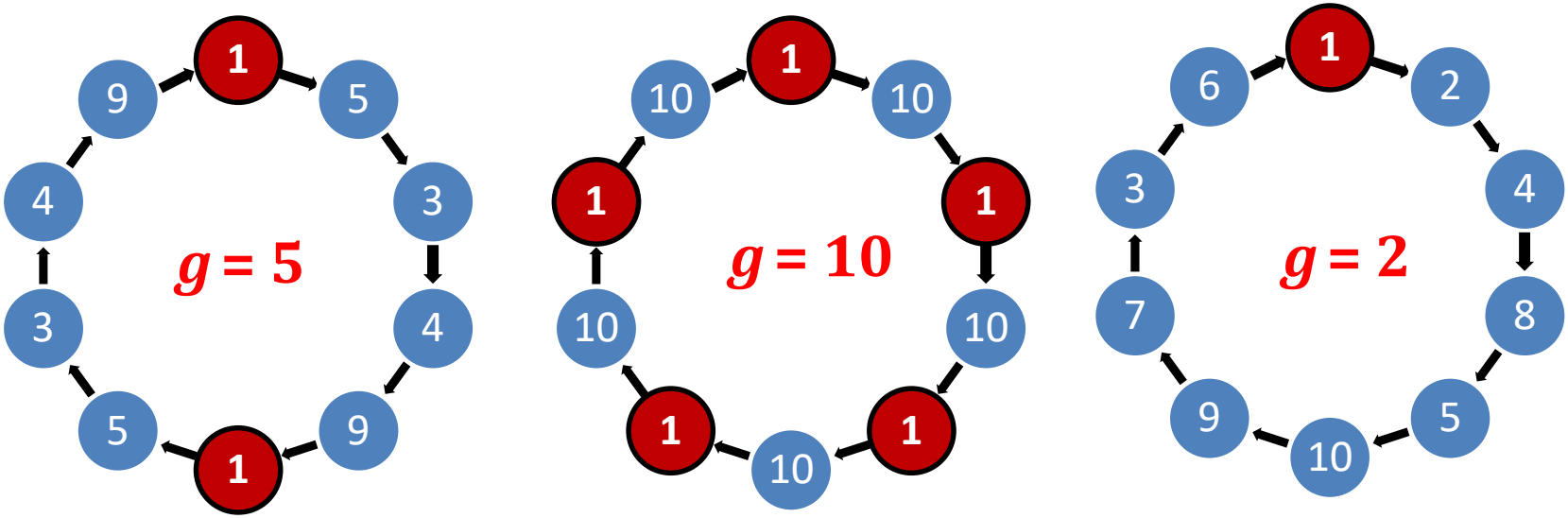
An **order of g** (denoted **$\mathrm{ord}(g)$**) is the smallest integer **$i > 0$** such that **$g^i = 1$**.

Of course **$i \leq |G|$**



Look...

Let $m := |Z_{11}^*| = 10$



Observe: in these examples

- $g^m = 1$
- the order of g divides the order of the group G.

we will now show that it's not a coincidence

Lemma

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

Then $g^m = 1$.

Proof

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

Observe that

$$g_1 \circ \cdots \circ g_m$$
$$= (g \circ g_1) \circ \cdots \circ (g \circ g_m)$$
$$= g^m \circ (g_1 \circ \cdots \circ g_m)$$

from associativity and commutativity

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, $m := |G|$, $g \in G$, $i \in \mathbf{N}$.
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

Which orders are possible?

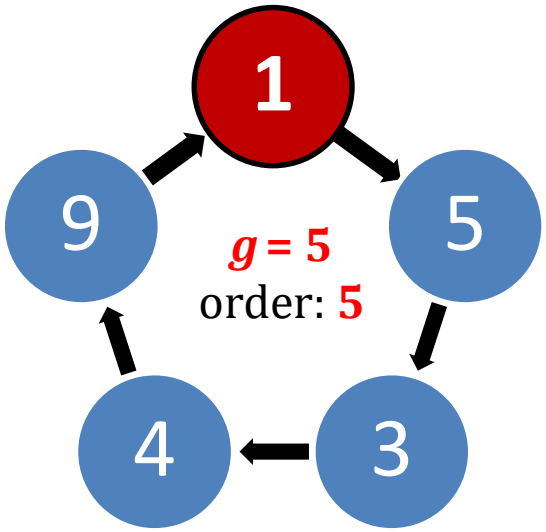
For Z_{11}^*:

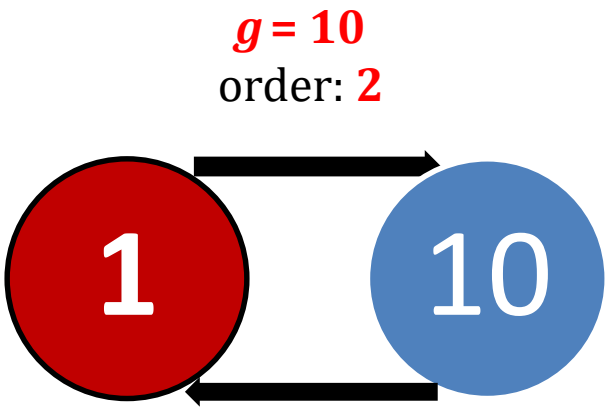
1,2,5,10

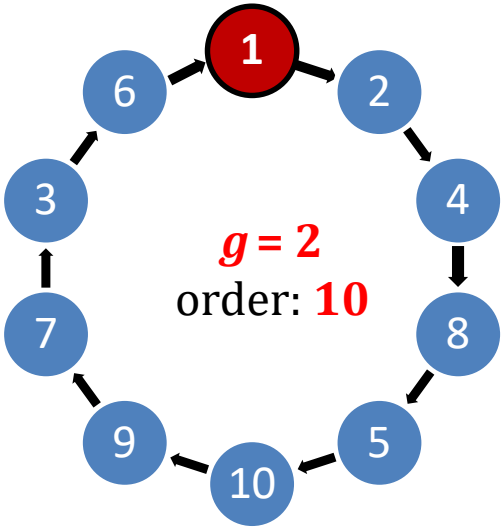
What do the have in common?

They are the divisors of

$10 = |Z_{11}^*|$







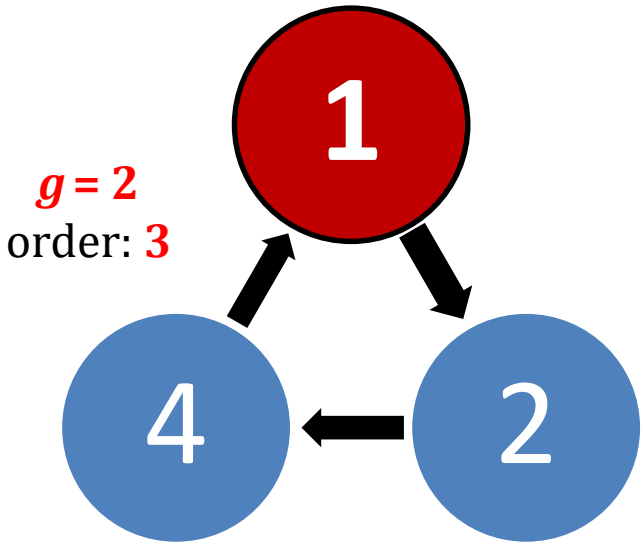
How does it look for Z_7^*?

For Z_7^*:

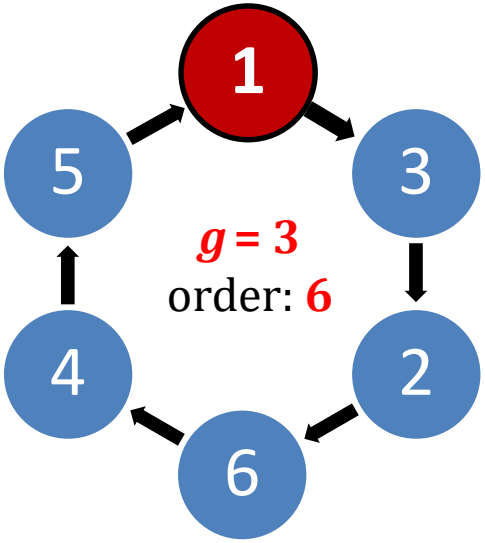
1,2,3,6

They are the divisors of

$6 = |Z_7^*|$



***g* = 2**
order: **3**



***g* = 3**
order: **6**

***g* = 6**
order: **2**

***g* = 1**
order: **1**

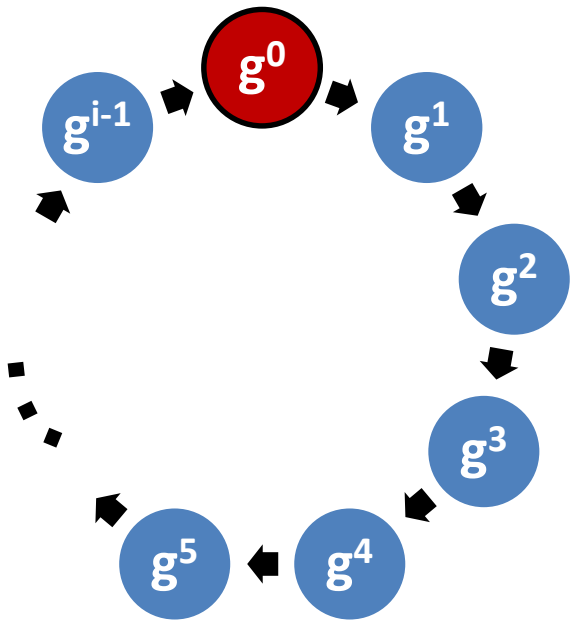
Generated subgroups

Definition

G – a group, $g \in G$, i – order of g

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

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



Why?

because:

1. it is closed under multpilication

$$g^a \cdot g^b = g^{a+b \bmod i}$$

2. the inverse of every g^a exists, and it is equal to

$$g^{i-a}$$

Because: $g^{i-a} \cdot g^a = g^i = 1$

Observe

order of an element g

=

order of the group $\langle g \rangle$

We can now use the Lagrange's Theorem

Lagrange's Theorem

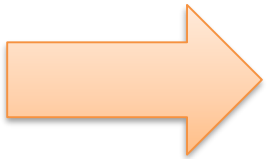
If **H** is a **subgroup** of **G** then

|H| divides **|G|**

So, that's why the order of g divided the order of the group G.

Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Group Z_N^* and its subgroups
6. Elliptic curves



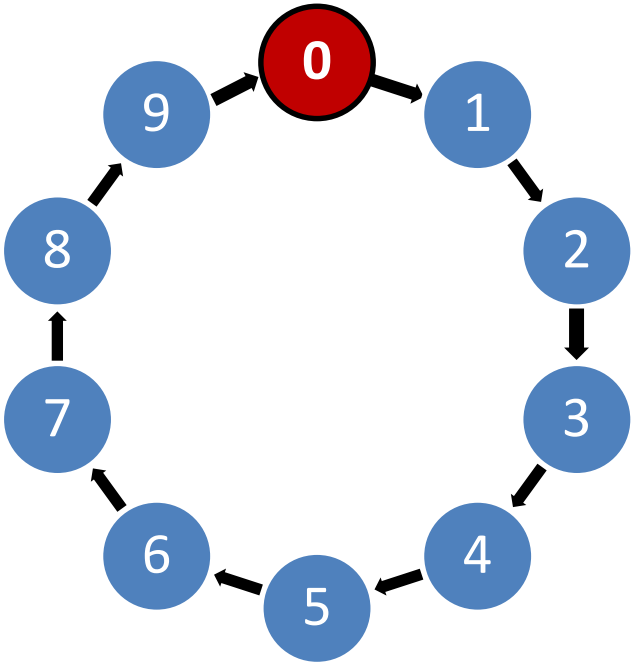
Cyclic groups

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

Such a $\boldsymbol{g}$ is called a **generator of $\boldsymbol{G}$**.

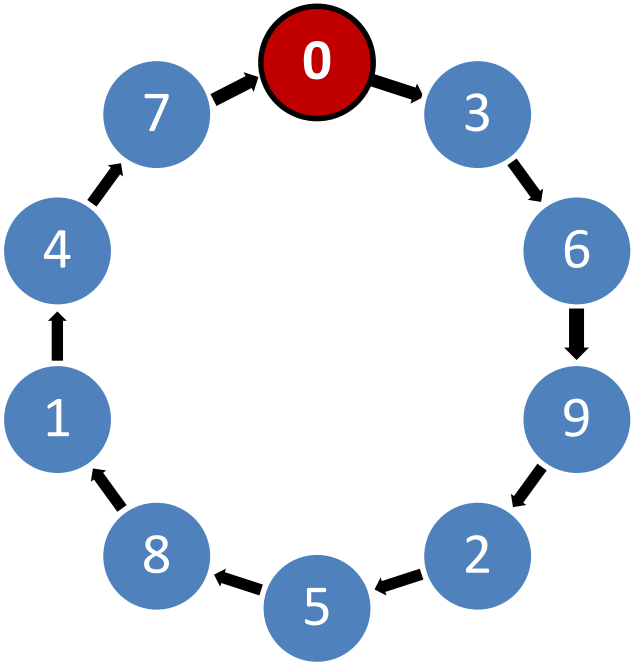
Example

1 is a generator of $\mathbf{Z_{10}}$



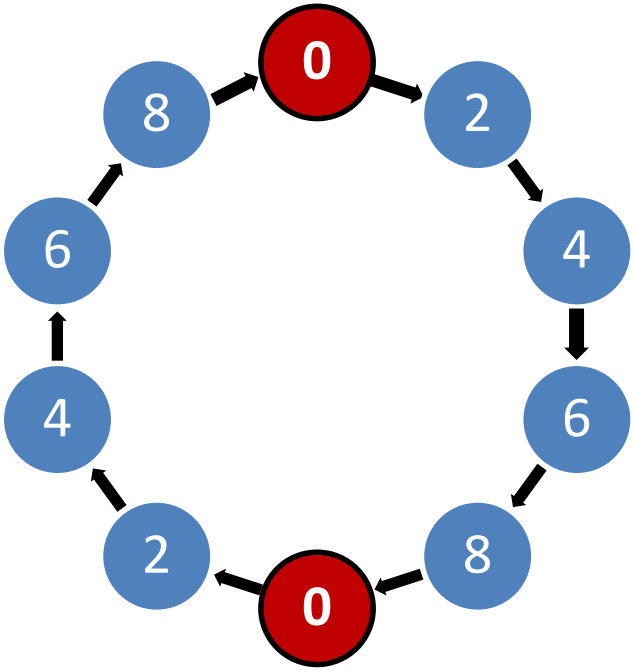
Example

3 is a generator of $\mathbf{Z_{10}}$



Example

2 is **not** a generator of $\mathbf{Z_{10}}$



Observation

Every group **G** of a prime order 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***.

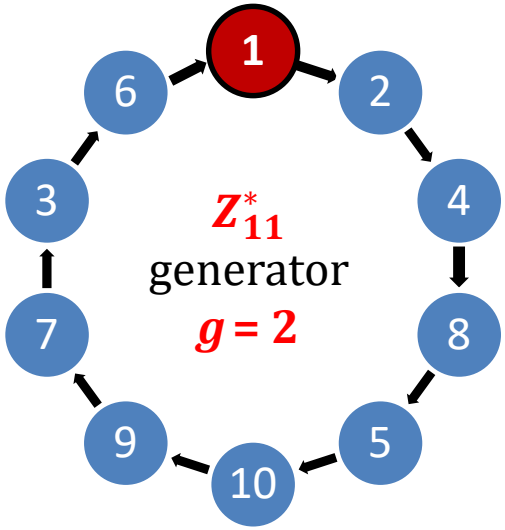
Another fact

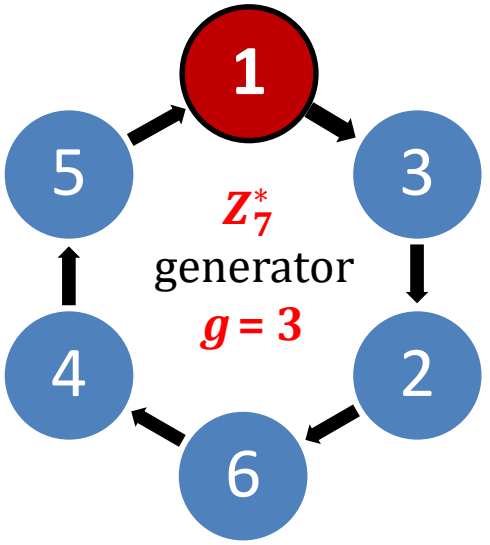
Theorem

If p is prime, then Z_p^* is cyclic.

We leave it without a proof.

We verified that it is true for $p=11$ and $p=7$.





Of course:

Not every element of

$$Z_p^*$$

is its generator.

For example:

$$p\text{-}1$$

has order 2 because

$$(p-1)^2 = p^2 - 2p + 1 = 1 \pmod{p}$$

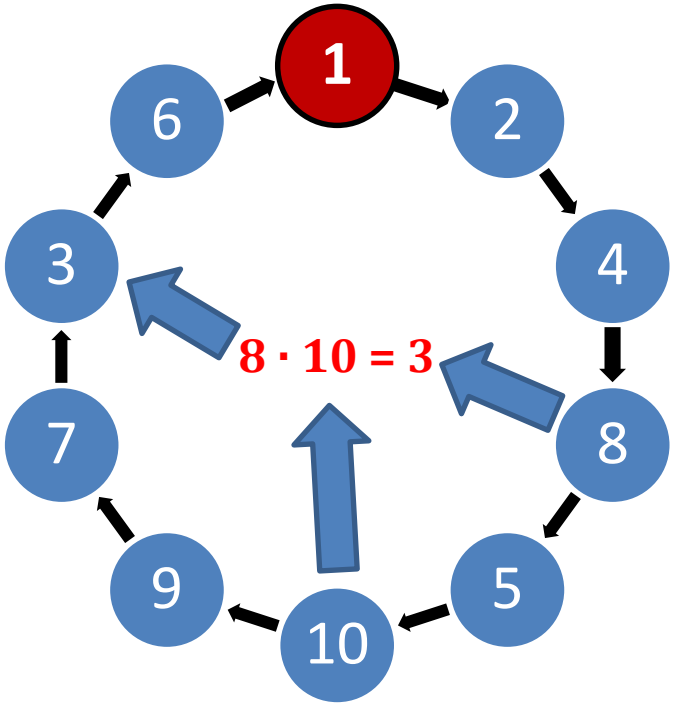
Example of a group that is not cyclic

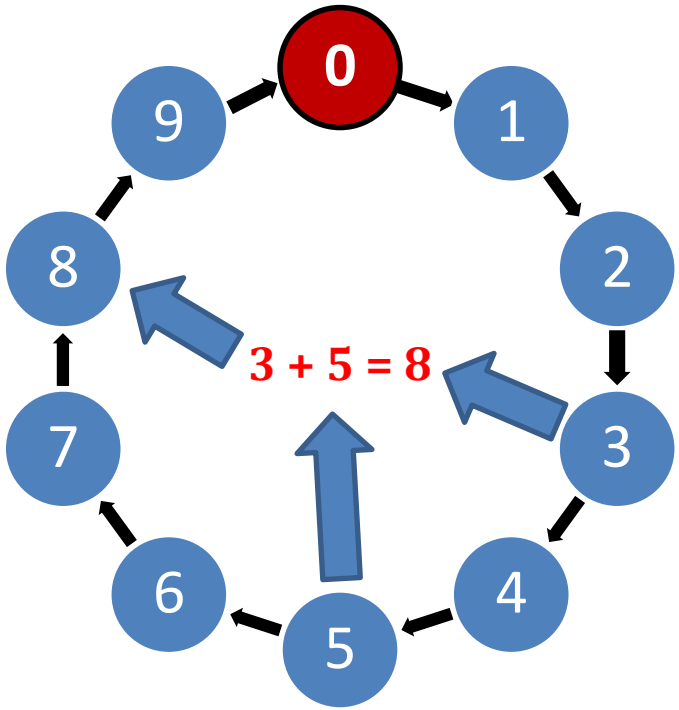
Z_{15}^{*}:

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

The maximal order is **4**...

Look...





Z_{11}^* and Z_{10} are essentially the same group:

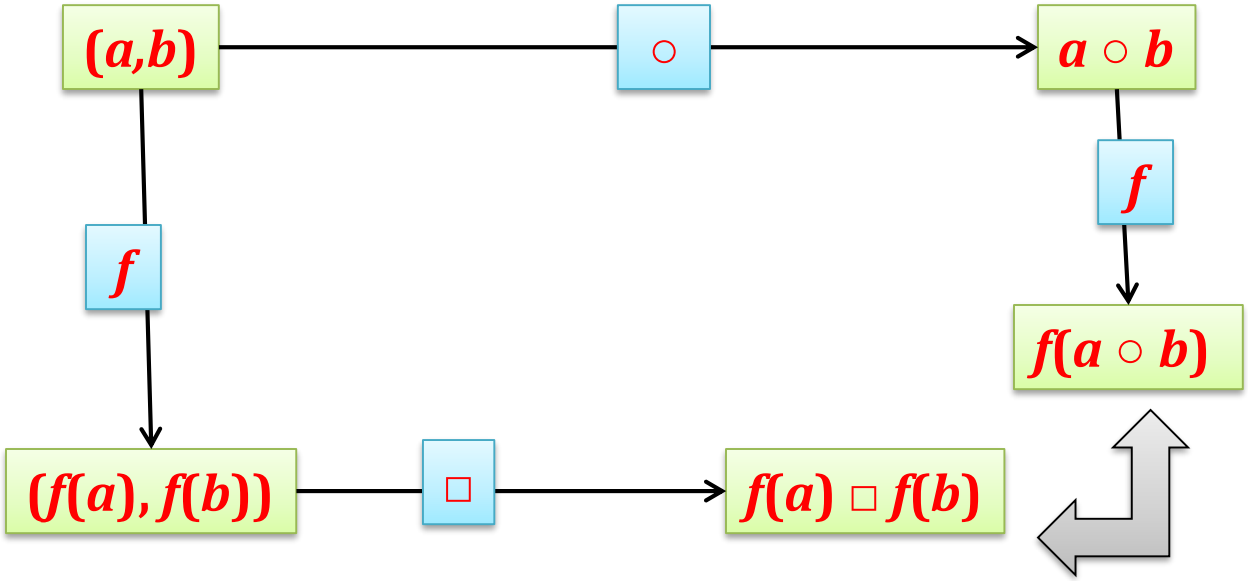
$$g^a \cdot g^b \bmod 11 = g^{a+b} \bmod 10$$

In other words: Z_{11}^* and Z_{10} are **isomorphic**.

Group isomorphism

G – a group with operation $\circ$
H – a group with operation $\square$

Definition A function $f: G \to H$ is a **group isomorphism** if
1. it is a **bijection**, and
2. it is a **homomorphism**, i.e.: for every $ab \in G$ we have
$$f(a \circ b) = f(a) \square f(b).$$



Isomorphic groups

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

Of course isomorphism is an equivalence relation.

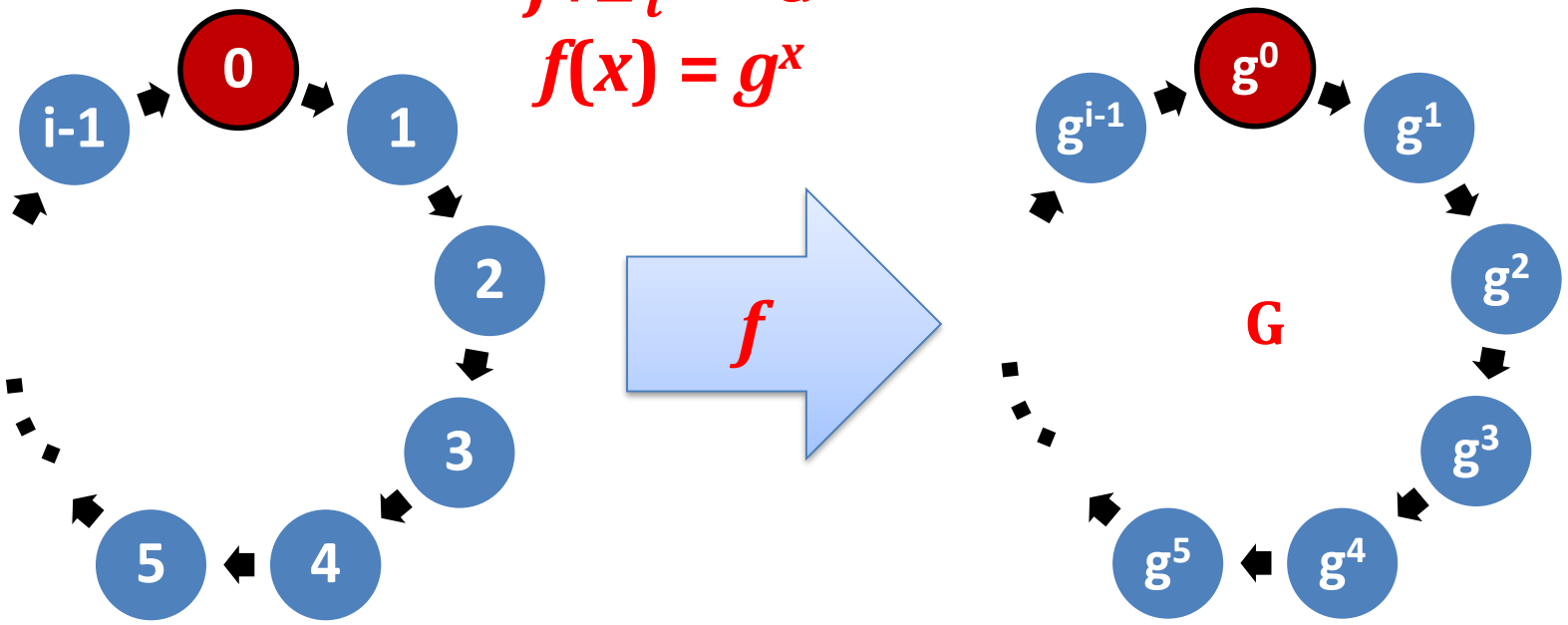
This is an isomorphism

G – a cyclic group of order ***i***

g – a generator of **G**

$$f: Z_i \rightarrow G$$

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

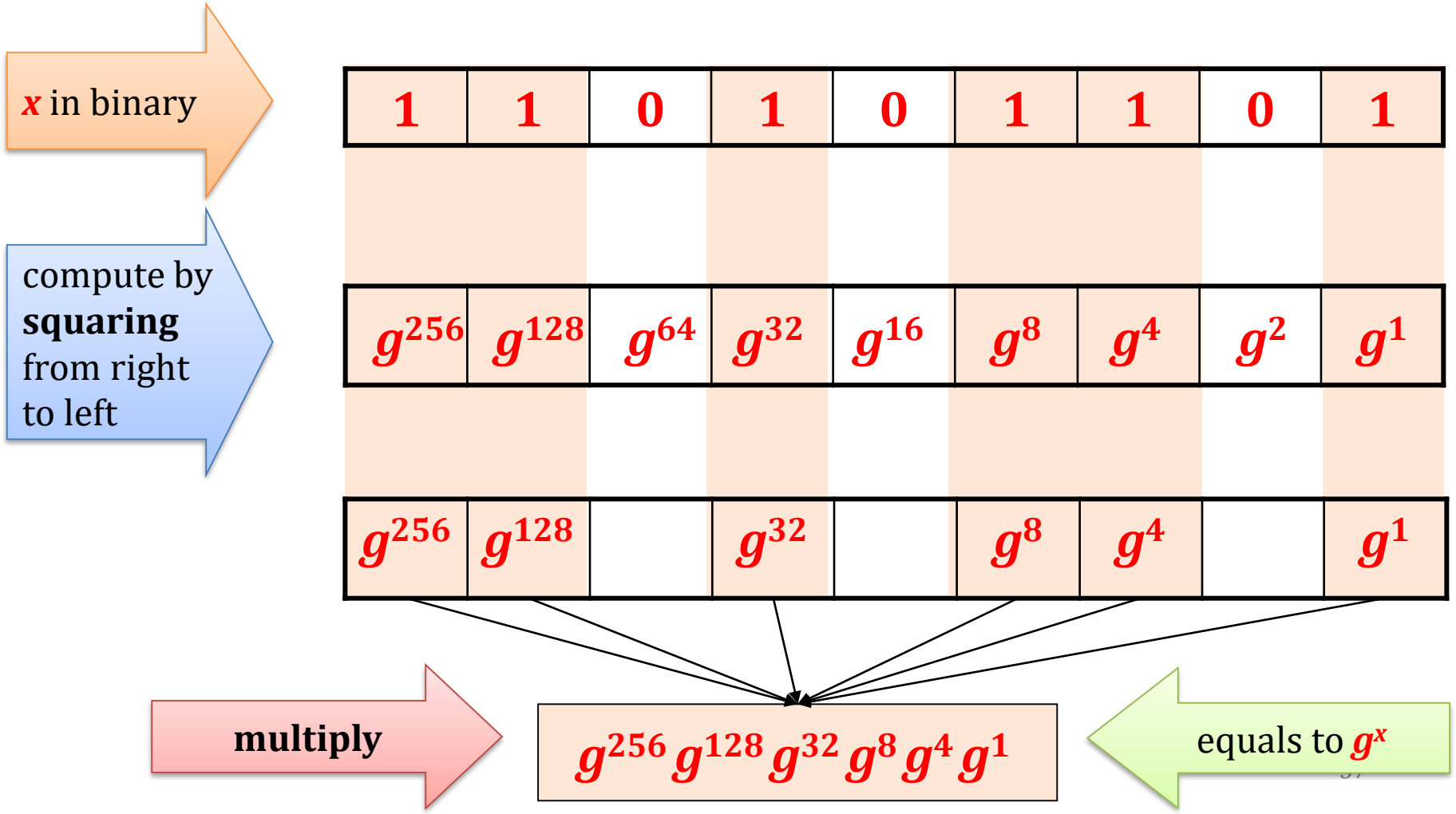


Why? Because $g^a \cdot g^b = g^{a+b \bmod i}$

How to compute g^x for large x?

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

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

is hard!

The discrete logarithm

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

$$y = g^x$$

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

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

Informally speaking:

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

Hardness of the discrete log

In some groups it is easy:

- in Z_n it is **easy** because $a^e = e \cdot a \bmod n$
- In 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 $\mathbf{1}^n$.

Therefore we need an algorithm **H** that

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

Example

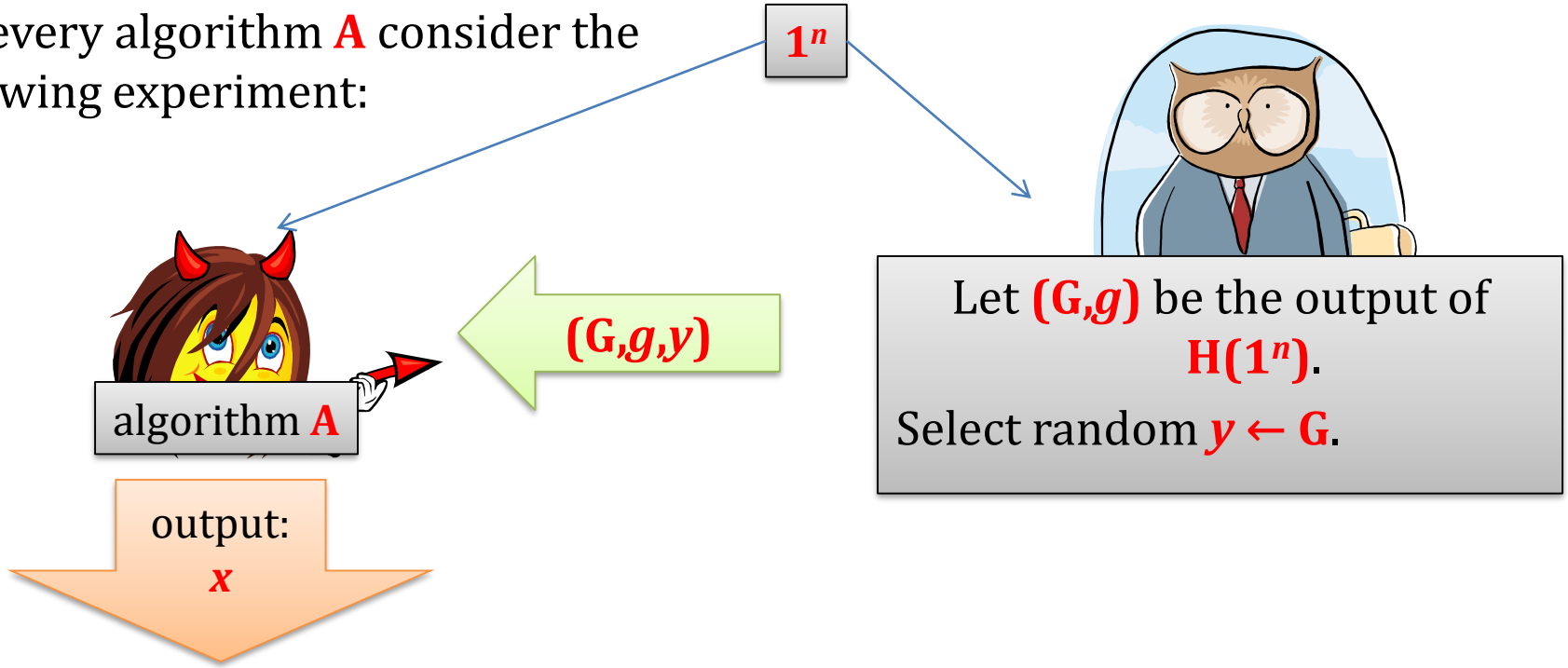
H on input $\mathbf{1}^{n}$:

outputs a

- random prime p of length n
- a generator of 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** $\quad$ **P(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 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 FFFFFFFF 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 FFFFFFFF FFFFFFFF.

the generator is: **2**.

A problem

$$f: \{0, \ldots, p-1\} \rightarrow Z_p^*$$

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

$$a = b^2 \bmod p$$

$\mathbf{QR}_p$ – a set of quadratic residues modulo p

$\mathbf{QR}_p$ is a subgroup of Z_p^*

$\mathbf{QNR}_p := Z_p^* \setminus \mathbf{QR}_p$

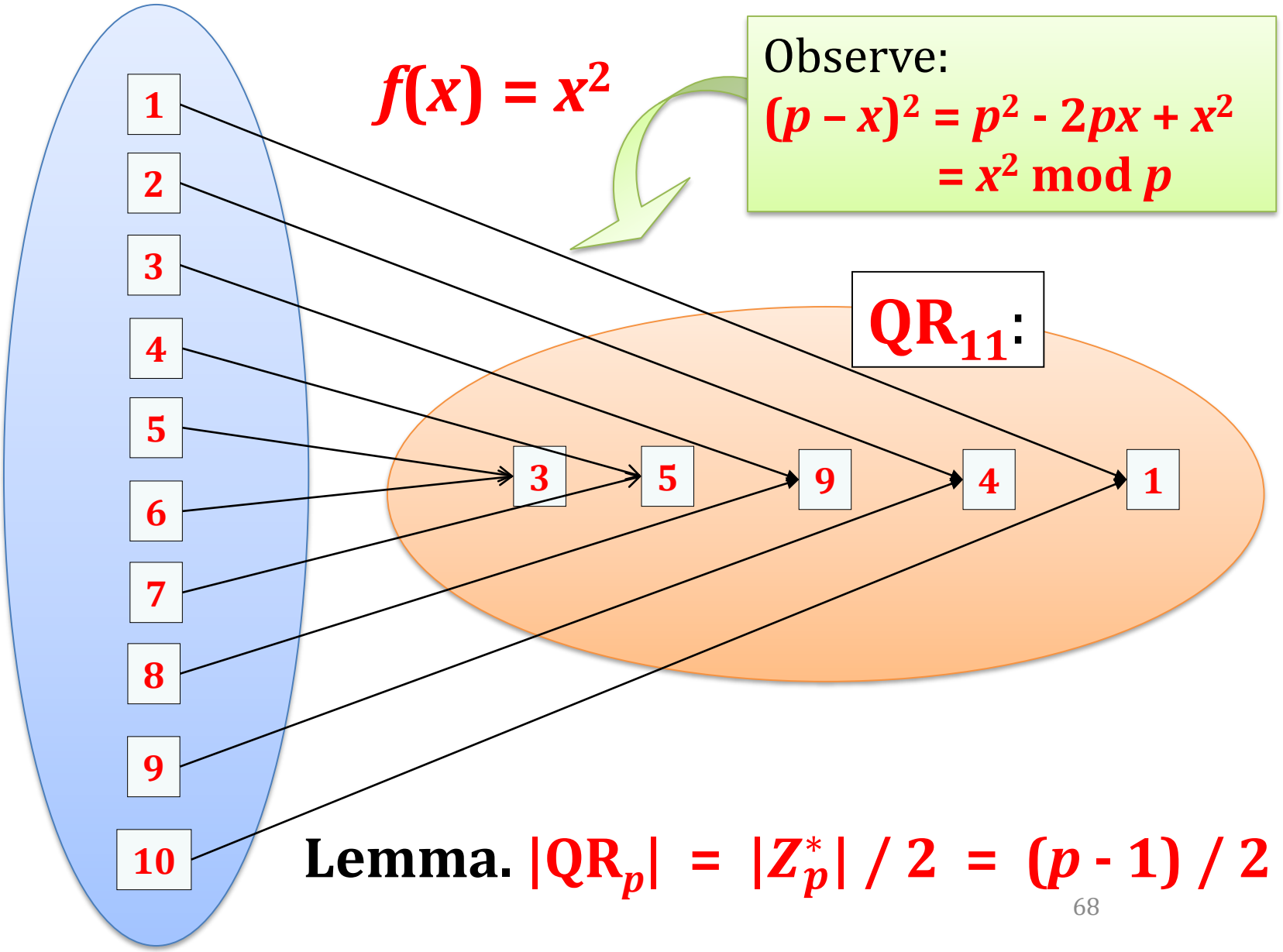
Why?

because:

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

What is the size of $\mathbf{QR}_p$?

Example: QR_{11}



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^*.

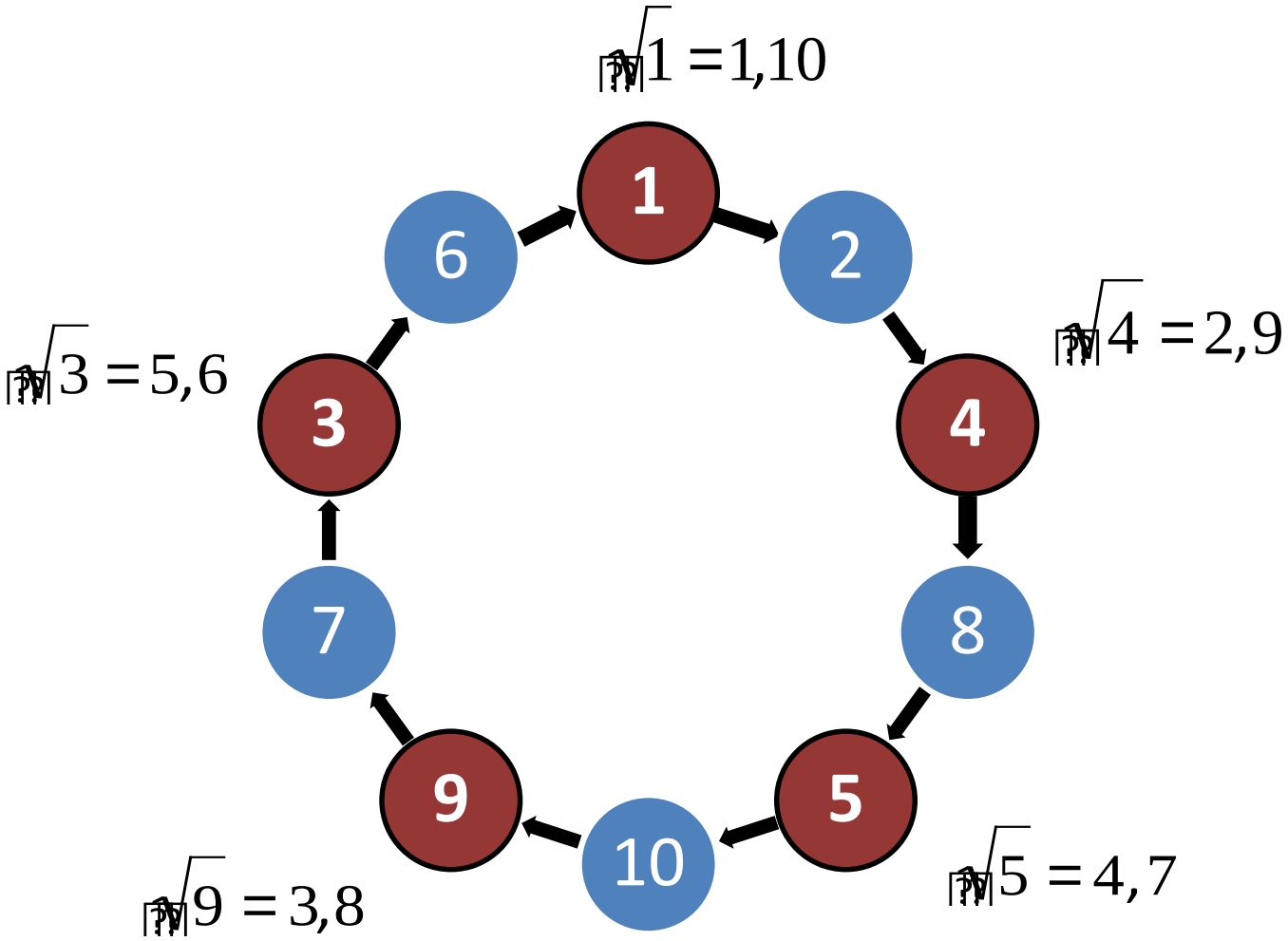
Then $QR_p = \{g^2, g^4, \ldots, g^{p-1}\}$.

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.

Example: $QR_{11} = \{1,4,5,9,3\}$



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

Yes!

Observation

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

Proof

(→)

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

Hence

$$a^{(p-1)/2}$$

$$=$$

$$(g^{2i})^{(p-1)/2}$$

$$=$$

$$g^{i(p-1)} = 1.$$

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

(←)

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

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

$$a^{(p-1)/2}$$

$$= \left(g^{2i+1}\right)^{(p-1)/2}$$

$$= g^{i(p-1)} \cdot g^{(p-1)/2}$$

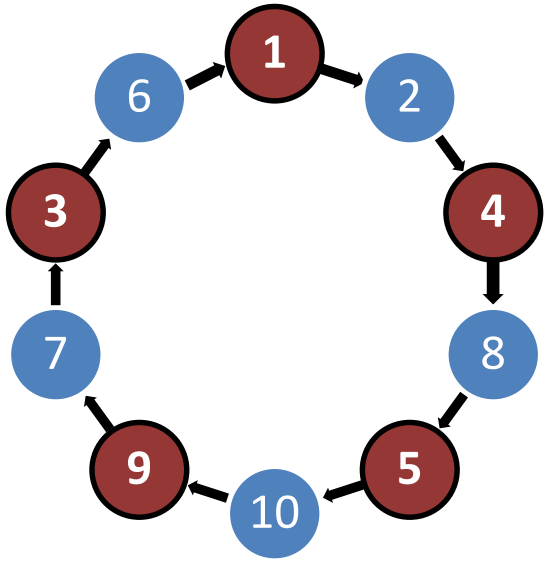
$$= g^{(p-1)/2}$$

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

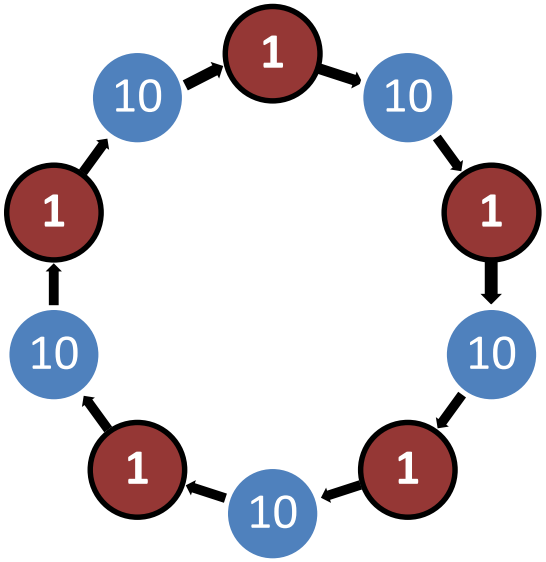
QED

Example: Z_{11}^*

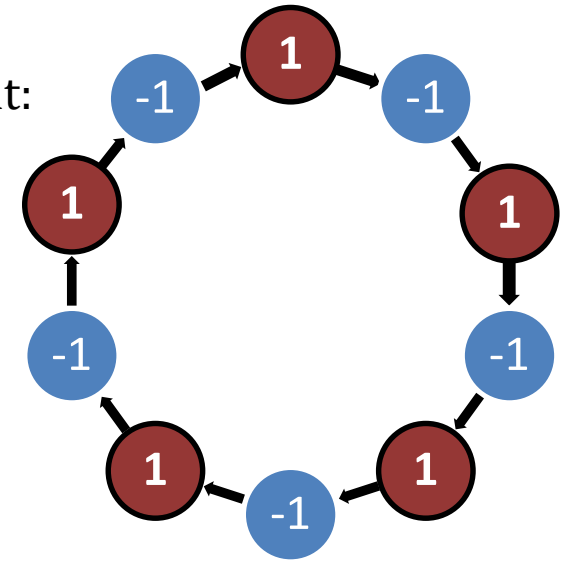
$(11 - 1)/2 = 5$



$f(x) = x^5$



another way to look at it:



Not a coincidence:

$(x^{(p-1)/2})^2 = 1$ implies that $x = \pm 1$

Hence we get a problem:

g – a generator of Z_p^*

f: $\{0,\ldots,p-1\} \to Z_p^*$ defined as $f(x) = g^x$ is a one-way function, but

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

(by checking if $f(x) \in \mathbf{QR}$)...

For some applications this is not good.

(but sometimes people don't care)

What to do?

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

How to find a generator of $\mathbf{QR}_p$?
Choose p that is a **strong prime**, that is:

$p = 2q + 1$, with q prime.

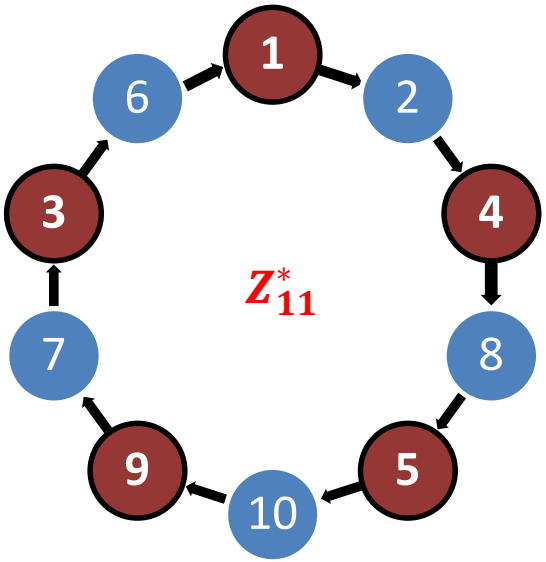
Hence $\mathbf{QR}_p$ has a prime order (q).

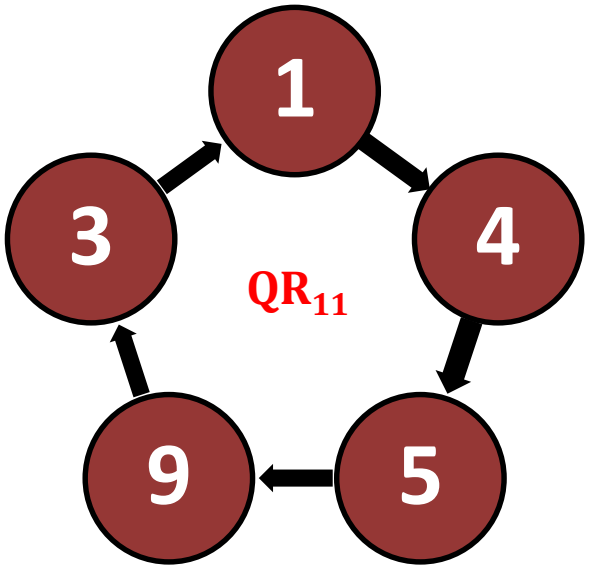
Every element (except of **1**) of a group of a prime order is its generator!

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

Example

11 is a strong prime (because **5** is a prime)





How to compute square roots modulo a prime p?

Yes!

We show it only for $p = 3 \pmod 4$ (for $p = 1 \pmod 4$ this fact also holds, but the algorithm and the proof are more complicated).

How to compute square root of x in reals?

One method: compute $x^{1/2}$

Problem "$1/2$" doesn't make sense in Z_n^*...

Write $p = 4m + 3$.

Fact $\sqrt{x} = x^{m+1}$

Proof:

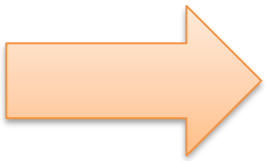
$$(x^{m+1})^2 = x^{2(m+1)}$$
$$= x^{2m+1+1}$$
$$= x^{2m+1} \cdot x^1$$
$$= x^1$$

Hence: order of QR_p is equal to $(p-1)/2 = (4m+2)/2 = 2m + 1$

x^{2m+1} is equal to 1 because of this

Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Group Z_N^* and its subgroups
6. Elliptic curves



Chinese Remainder Theorem (CRT)

Let **N = pq**, where p and q are two **distinct** primes.
Define: **f(x) := (x mod p, x mod q)**

Chinese Remainder Theorem (CRT):

f is an isomorphism between

1. Z_N and $Z_p \times Z_q$
2. Z_N^* and $Z_p^* \times Z_q^*$

To prove it we need to show that

- f is a **homorphism** .
  - between Z_N and $Z_p \times Z_q$, and
  - between Z_N^* and $Z_p^* \times Z_q^*$.
- f is a **bijection**:
  - between Z_N and $Z_p \times Z_q$, and
  - between Z_N^* and $Z_p^* \times Z_q^*$.

$f: Z_N \to Z_p \times Z_q$ is a homomorphism

Proof:

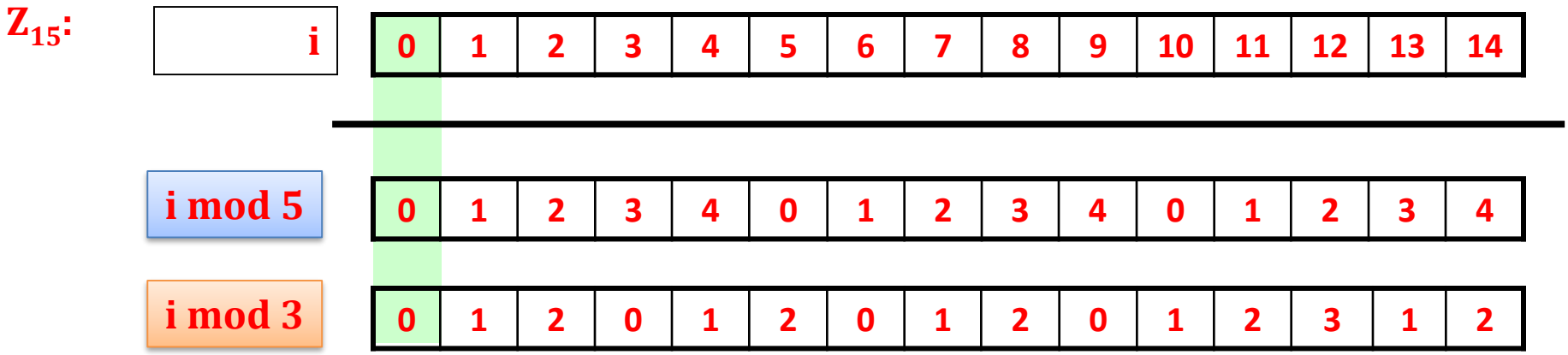
$$f(a+b)$$
$$=$$
$$(a+b \bmod p,\ a+b \bmod q)$$
$$=$$
$$(((a \bmod p)+(b \bmod p)) \bmod p,\ ((a \bmod q)+(b \bmod q)) \bmod q)$$
$$=$$
$$(a \bmod p, a \bmod q)+(b \bmod p, b \bmod q)$$
$$=$$
$$f(a)+f(b)$$

$f: Z_N^* \to Z_p^* \times Z_q^*$ is a homomorphism

Proof:

$$f(a \cdot b)$$
$$=$$
$$(a \cdot b \bmod p,\ a \cdot b \bmod q)$$
$$=$$
$$(((a \bmod p) \cdot (b \bmod p)) \bmod p,\ ((a \bmod q) \cdot (b \bmod q)) \bmod q)$$
$$=$$
$$(a \bmod p, a \bmod q) \cdot (b \bmod p, b \bmod q)$$
$$=$$
$$f(a) \cdot f(b)$$

An example



i mod 5

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

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

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

Z_{24}:

i mod 6

i mod 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 distinct primes then $f: Z_N \to Z_p \times Z_q$ is a bijection

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

because p and q are distinct primes

Proof:

We first show that it is injective.

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

$i \bmod p = j \bmod p \rightarrow p$ divides $i-j$

and $i \bmod q = j \bmod q \rightarrow q$ divides $i-j$

$\rightarrow N=pq$ divides $i-j$

$\rightarrow i = j \bmod N$

Since $|Z_N| = N = pq = |Z_p \times Z_q|$ we are done!

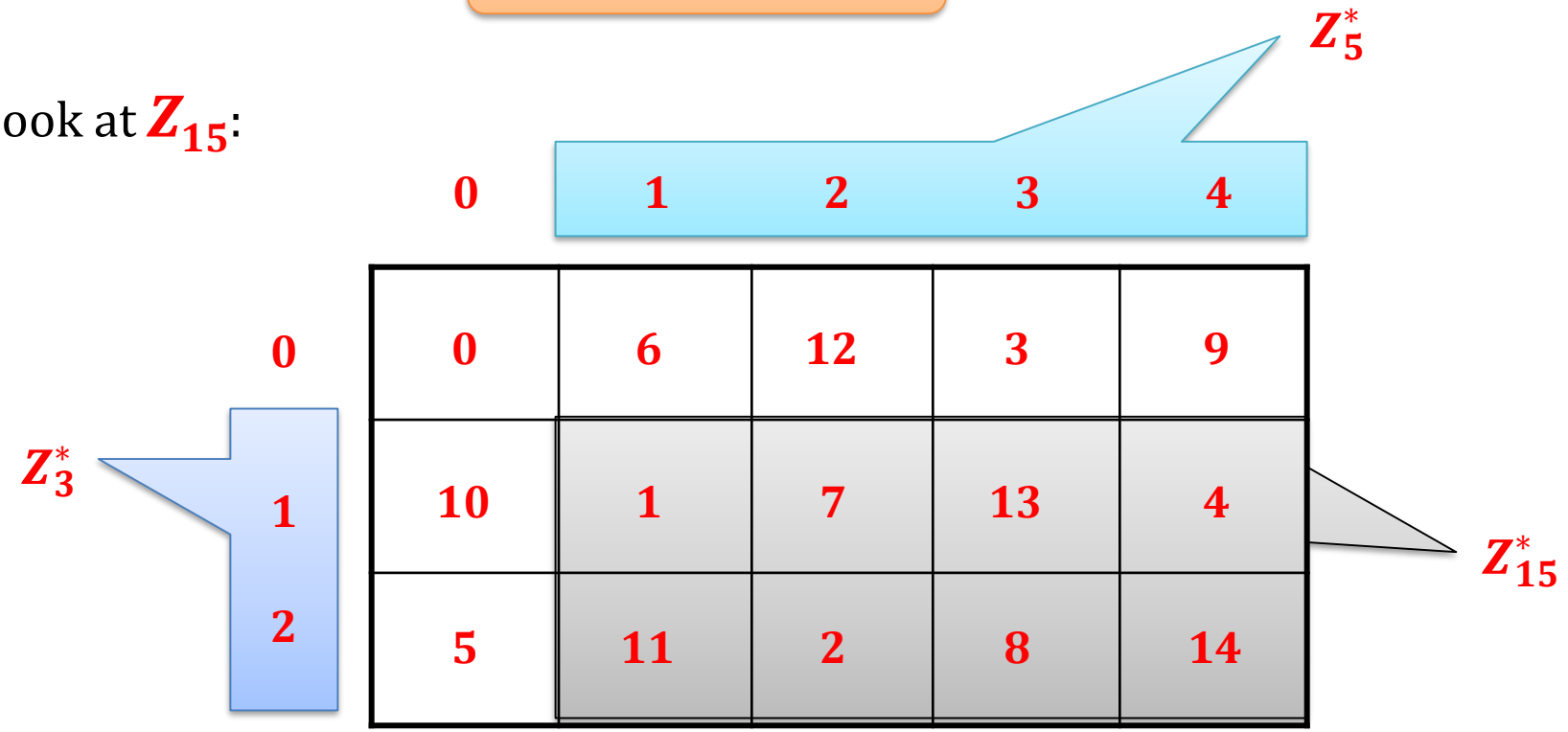
$f: Z_N^* \to Z_p^* \times Z_q^*$ is also a bijection

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

$$|Z_N^*| = |Z_p^*| \times |Z_q^*|$$

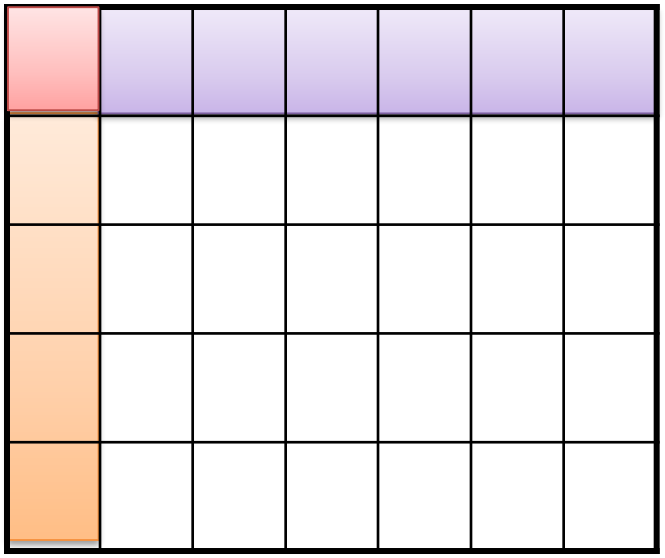
$= (p\text{-}1)(q\text{-}1)$

Look at Z_{15}:



$N = pq$

Which elements of Z_N are not in Z_N^*?



These sets are disjoint since p and q are distinct primes

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

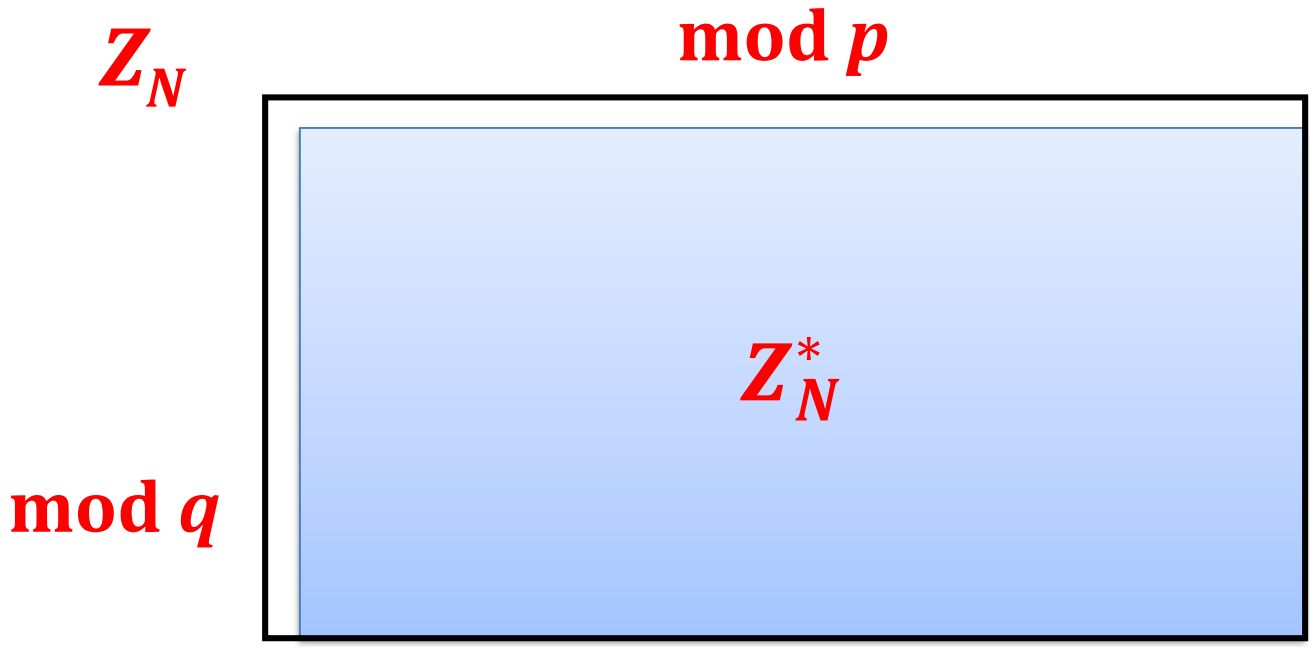
- Summing it up: $1 + (q - 1) + (p - 1) = q + p - 1$

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

So Z_N^* has $pq - (q + p - 1)$ elements.

QED

How does it look for large p and q?



pq is called **RSA modulus**

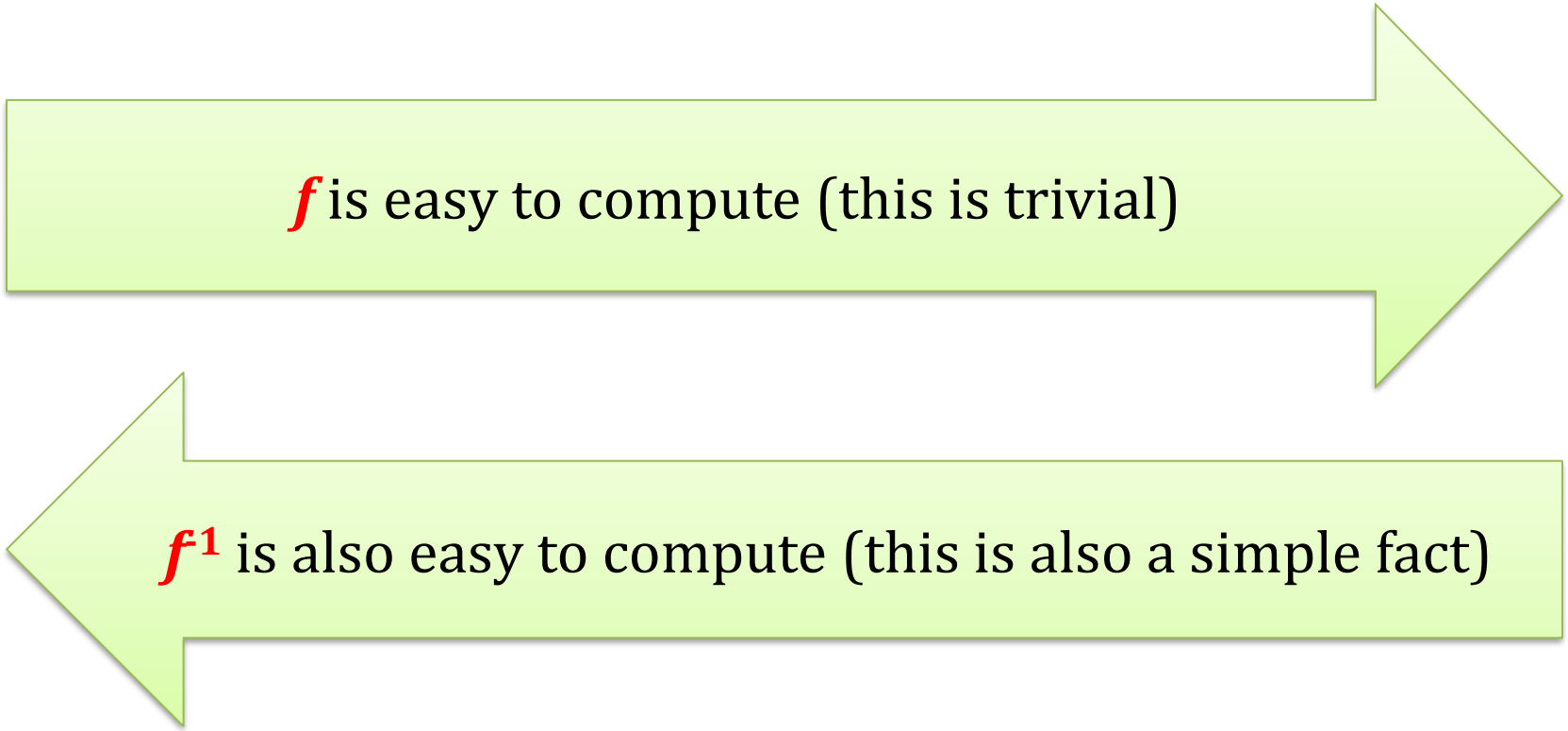
Z_N^* is called an **RSA group**

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)

Fact

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



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

Let

$c_1 := (q \bmod p)^{-1} \bmod p$

$c_2 := (p \bmod q)^{-1} \bmod q$

Then

$$g(y_1,y_2) := (q\, c_1\, y_1 + p\, c_2 y_2) \bmod pq$$

is the inverse of f.

(exercise)

By the way

Remember that we observed that Z_{15}^* is not cyclic?

Now we know why:

$$a^x \bmod pq = 1$$

iff

$$a^x \bmod p = 1 \text{ and } a^x \bmod q = 1$$

iff

$$x \mid p - 1 \text{ and } x \mid q - 1$$

iff

$$x \mid \text{lcm}(p-1, q-1)$$

for $p=3$ and $q=5$ it is equal to:

$$\text{lcm}(2,4) = 4$$

More general version of CRT

$p_1,\ldots, p_n$ – such that for every i and j we have

$$\gcd(p_i,p_j)$$

Define

$$f(x) := (x \bmod p_1,\ldots, x \bmod p_n)$$

Let $M = p_1 \cdot \cdots \cdot p_n$. Then foollowing f is an isomorphism

$$f: Z_M \to Z_{p_1} \times \cdots \times Z_{p_n}$$

and

$$f: Z_M^* \to Z_{p_1}^* \times \cdots \times Z_{p_n}^*$$

Moreover f and f^{-1} can be computed efficiently.

Euler's φ function

Define

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

Euler's theorem:

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

(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 $a \in \{1, \ldots, p-1\}$ we have

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

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

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

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

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

Fact
Computing $\varphi(N)$ is as hard as factoring N.

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

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

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

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

We can solve it:

(2) $p = N/q$

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

it is a quadratic equation
so we can solve it (in **R**)

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

Which problems are easy and which are hard in Z_N^* ($N = pq$)?

- **multiplying elements?**

  easy!

- **finding inverse?**

  easy! (Euclidean algorithm)

- **computing $\varphi(N)$** ?

  hard! - as hard as factoring N

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

  easy!

- **computing eth root** (for a large e)?

Computing eth roots modulo N

In other words, we want to invert a function:

$$f: Z_N^* \to Z_N^*$$

defined as

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

This is possible only if f is a permutation.

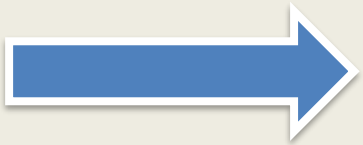
Lemma

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

In other words: $e \in Z_{\varphi(N)}^*$ (note: a "new" group!)

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

1.

$\gcd(e, \varphi(N)) = 1$  $f(x) = x^e \bmod N$ is a permutation

Let d be an inverse of e in $Z^*_{\varphi(N)}$. That is:

d is such that $d \cdot e = 1 \bmod \varphi(N)$.

Then:

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

2.

$\gcd(e, \varphi(N)) = 1$  $f(x) = x^e \bmod N$ is a permutation

[exercise]

Computing eth root – easy, or hard?

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

We have shown that the function

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

has an inverse

$f^{-1}(x) = x^d \bmod N$, where d is an inverse of e in $Z_{\varphi(N)}^*$

Moral:
If we know $\varphi(N)$ we can compute the roots efficiently.

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

Can we compute the eth root if we do not know $\varphi(N)$?

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:

$$\mathbf{P}(y^e = x \bmod N : \; y := A(x,N,e)) \;\; \textbf{is negligible}$$

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 $Z_{\varphi(N)}^*$

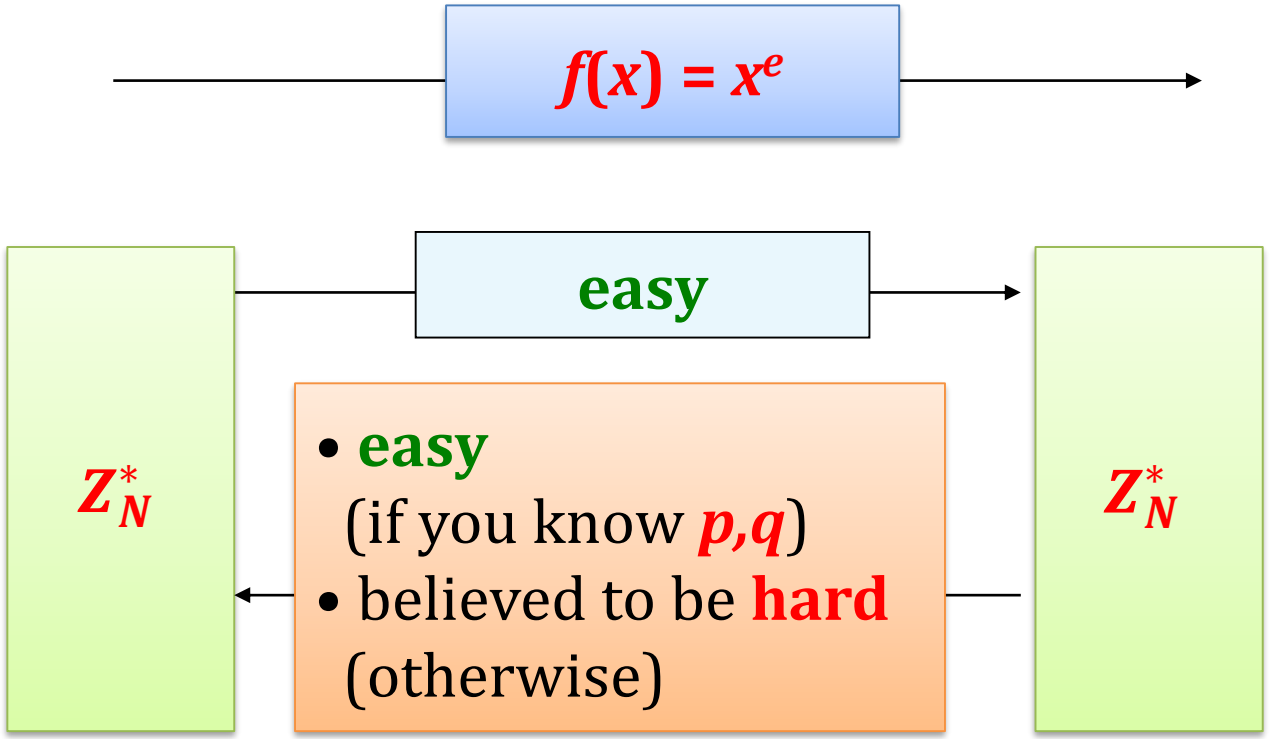
What can be shown?

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.

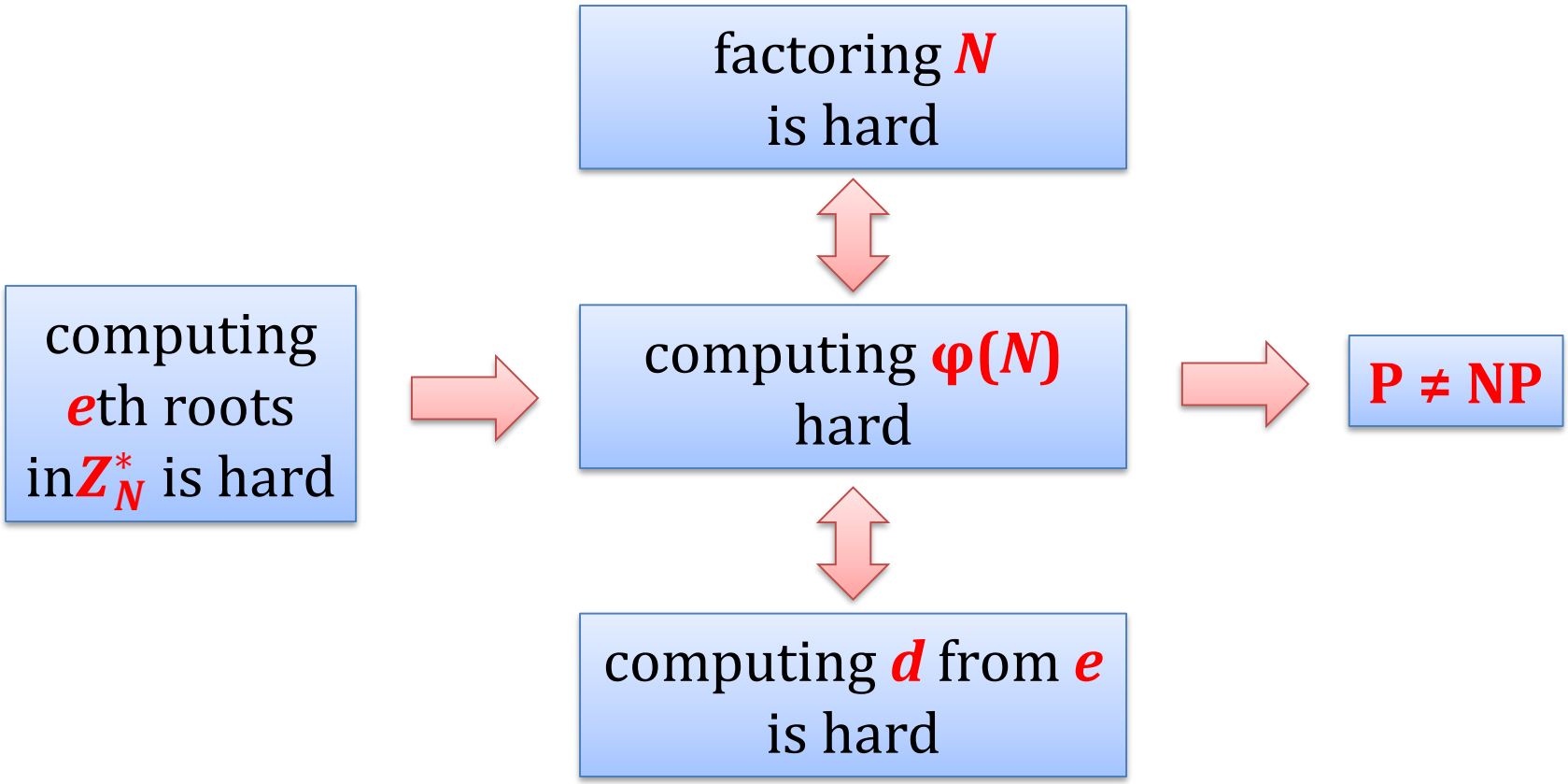


Functions like this are called **trap-door one-way permutations**.

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 eth root modulo N.

What about the case when $e = 2$?

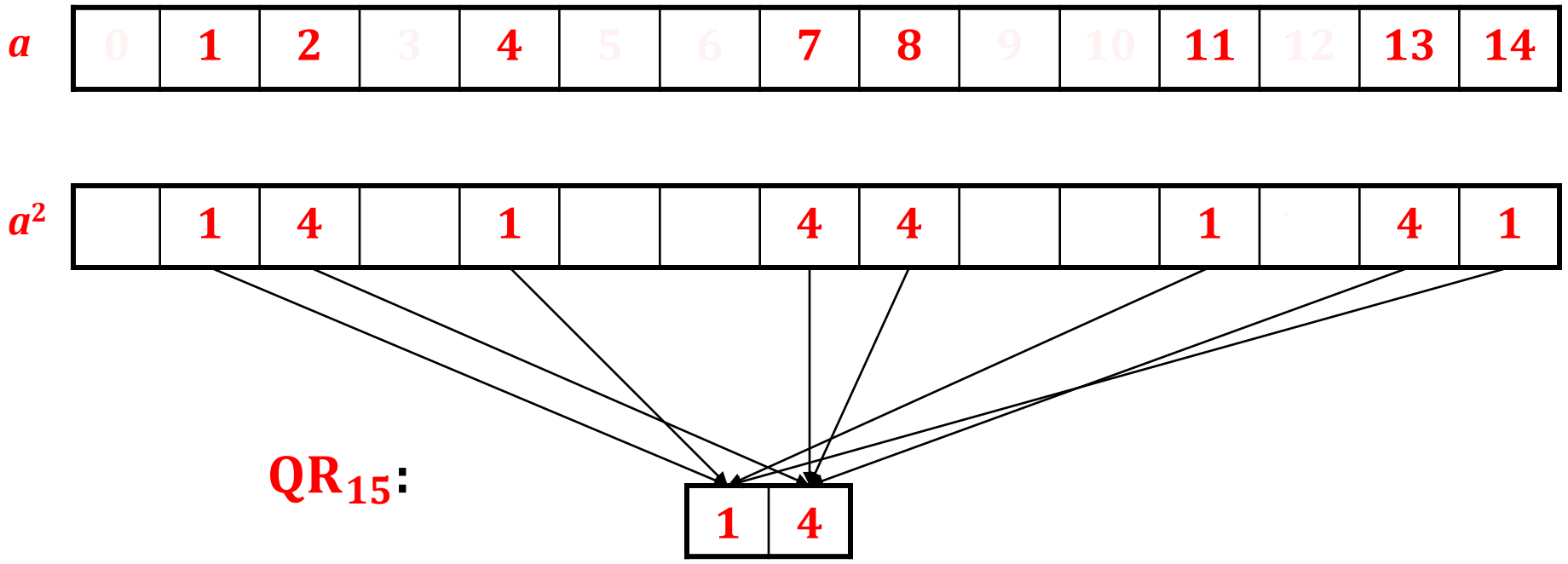
Clearly $\gcd(2,\varphi(N)) \neq 1$, so $f(x) = x^2$ is not a bijection.

Question

Which elements have a square root modulo N?

Quadratic Residues modulo pq

Z_{15}^*:



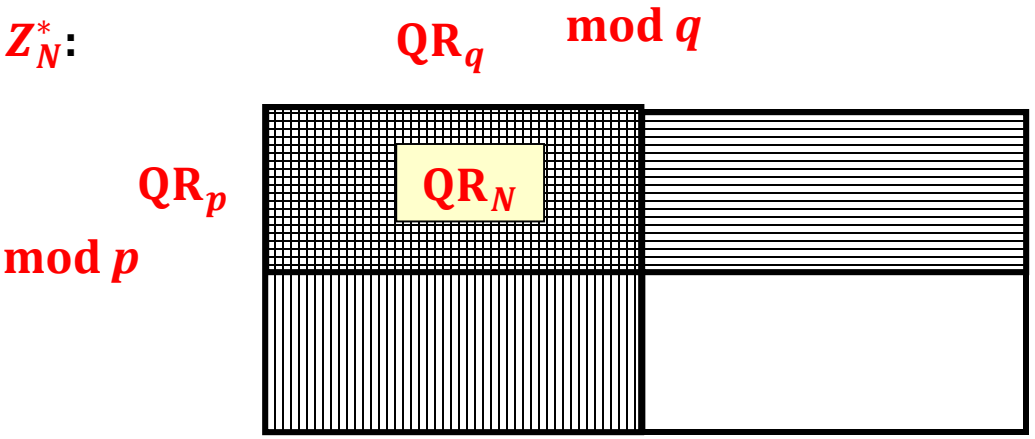
Observation: every quadratic residue modulo **15** has **exactly 4** square roots, and hence $|QR_{15}| = |Z_{15}^*| / 4$.

A lemma about QRs modulo pq

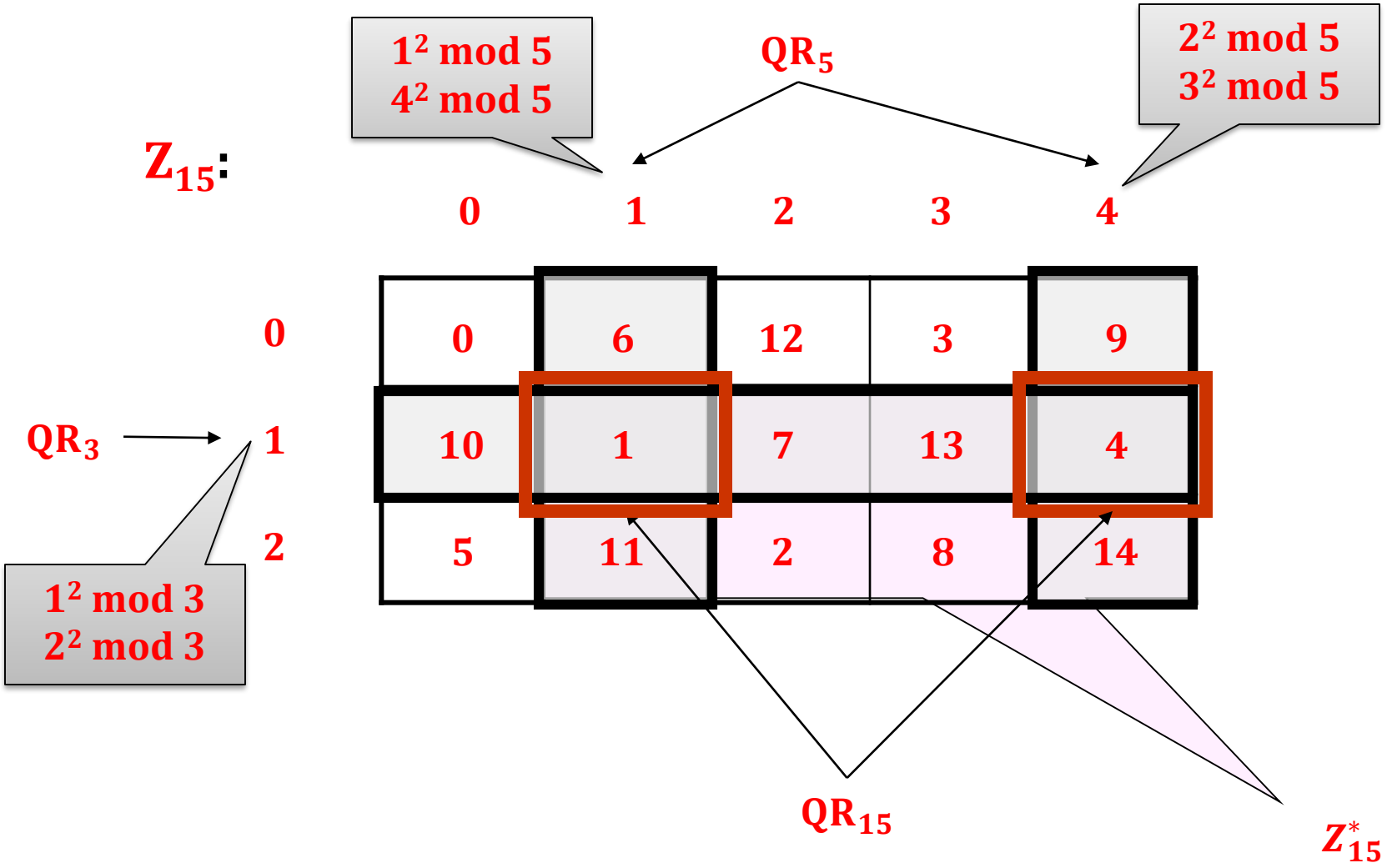
Fact: For $N=pq$ we have $|QR_N| = |Z_N^*| / 4$.

Proof:

$x \in QR_N$
iff
$x = a^2 \bmod N$, for some a
iff (by **CRT**)
$x = a^2 \bmod p$ and $x = a^2 \bmod q$
iff
$x \bmod p \in QR_p$ and $x \bmod q \in QR_q$



QRs modulo pq – an example



Every $x \in QR_N$ has exactly 4 square roots

More precisely, every $z = x^2$ has the square roots x_{++} and x_{+-}, x_{-+}, x_{--} such that:

- $x_{++} = x \pmod p$ and $x_{++} = x \pmod q$
- $x_{+-} = x \pmod p$ and $x_{+-} = -x \pmod q$
- $x_{-+} = -x \pmod p$ and $x_{-+} = x \pmod q$
- $x_{--} = -x \pmod p$ and $x_{--} = -x \pmod q$

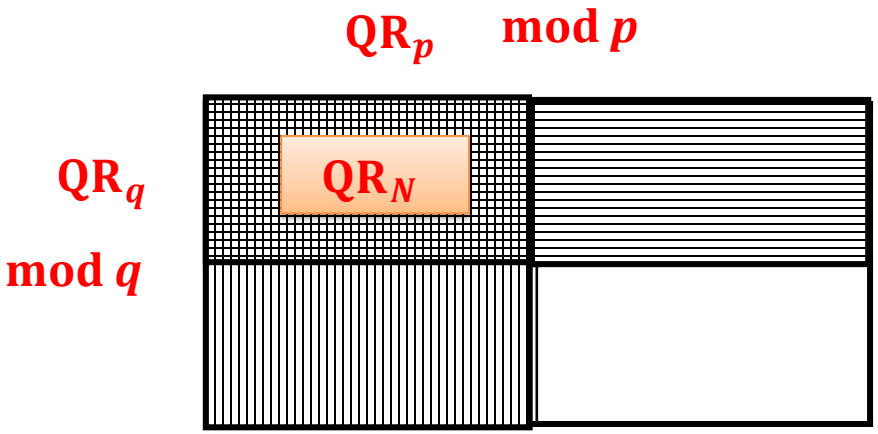
equals to x

equals to $-x$

Jacobi Symbol

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

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 Z_N^*

$Z_N^+ := \{x: J_n(x) = +1\}$

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 $\mathbf{QR}_N$?

Fact: if one knows p and q – yes!

Because:

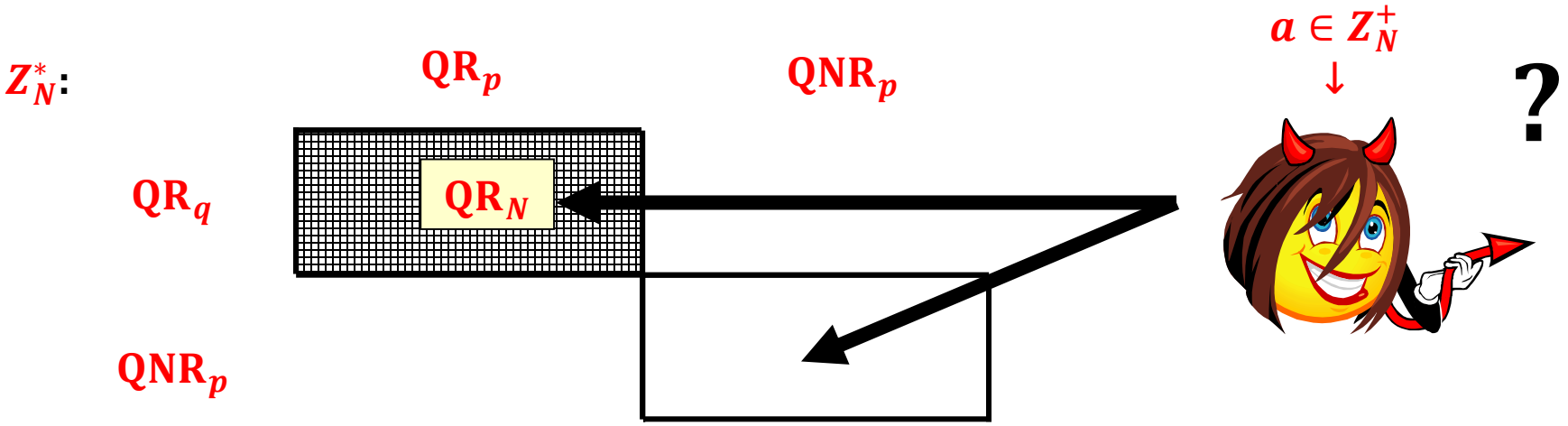
1. testing membership modulo a prime is easy
2. the "CRT function"

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

can be efficiently computed in both directions

What if one doesn't know p and q?

Quadratic Residuosity Assumption



$Q(N,a) = 1$ if $a \in QR_N$
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:

$$|P(D(N,a) = Q(N,a)) - ½|$$

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

So, how to compute a square root of $x \in \mathbf{QR}_N$?

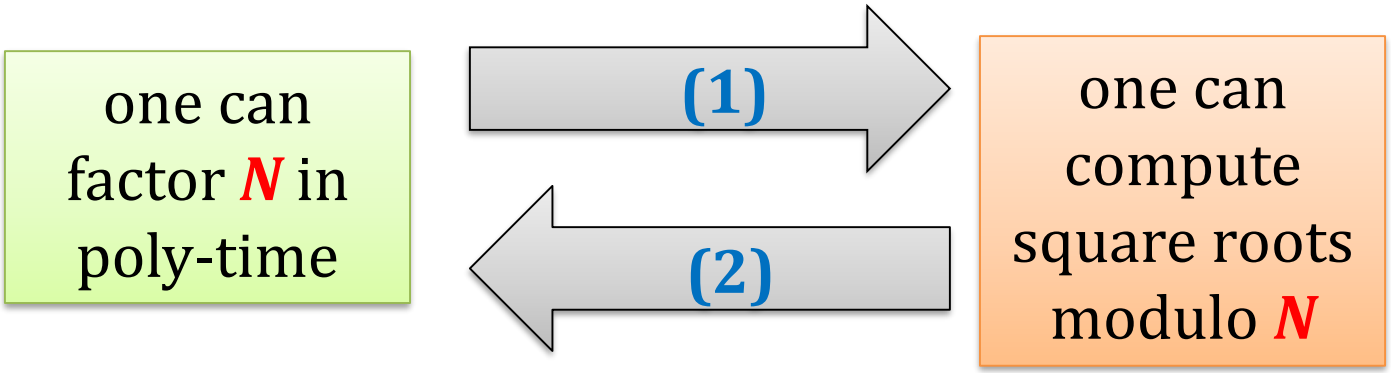
Fact

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

The problem of computing square roots (modulo N) of random elements in $\mathbf{QR}_N$ is poly-time equivalent to the problem of factoring N.

Proof

We need to show that:



one can factor N in poly-time

(1)

one can compute square roots modulo N

This follows from the fact that computing square roots modulo a prime p is easy.

$f(x) = (x \bmod p, x \bmod q)$ – the "CRT function"

x

1. Let
   $(a,b) = f(x)$

2. Compute α and β such that
   - $\alpha^2 = a \bmod p$
   - $\beta^2 = a \bmod q$

3. Output
   - $f^{-1}(\alpha,\beta)$
   - $f^{-1}(-\alpha,\beta)$
   - $f^{-1}(\alpha,-\beta)$
   - $f^{-1}(-\alpha,-\beta)$

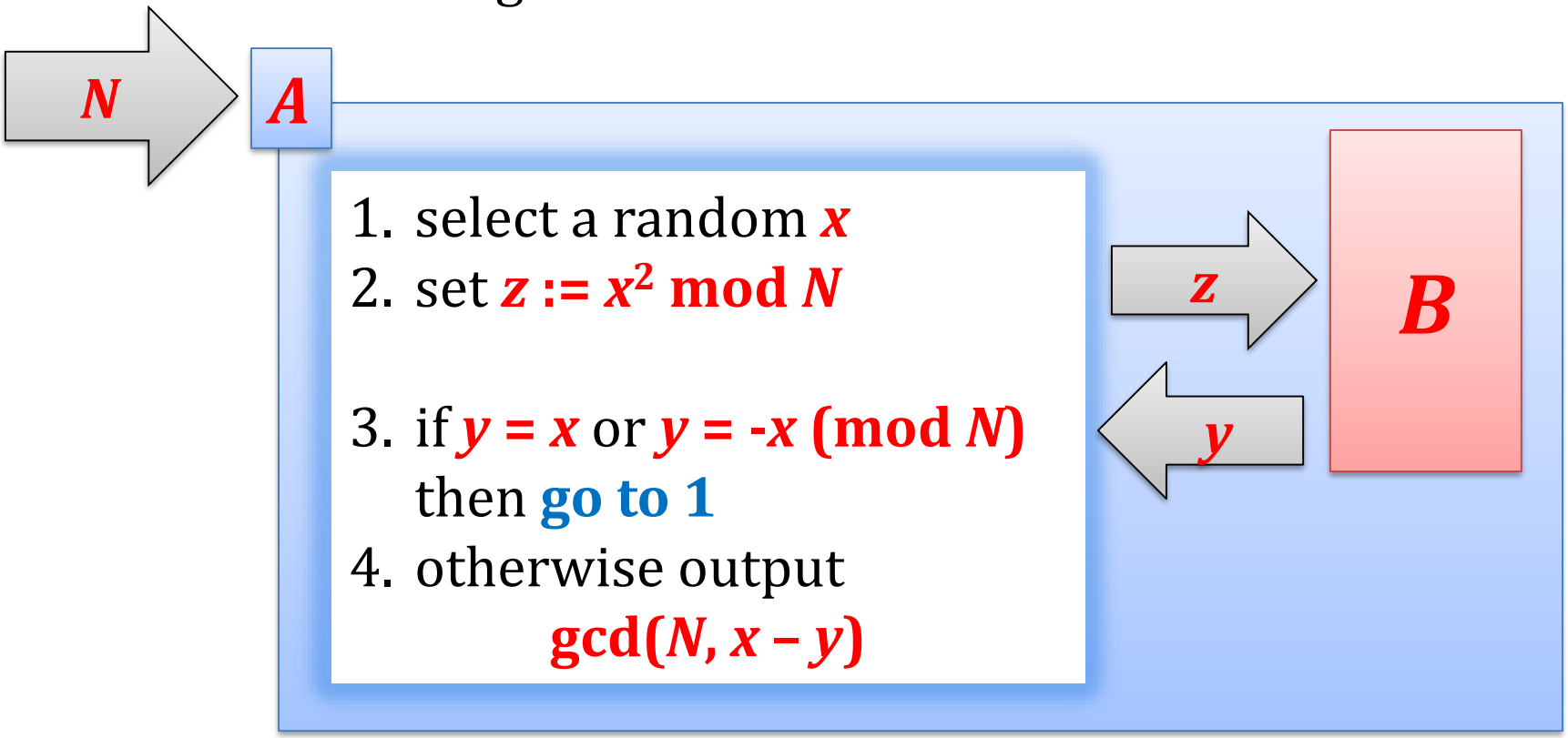
one can factor N in poly-time

(2)

one can compute square roots modulo N

Suppose we have an algorithm B that computes the square roots.

We construct an algorithm A that factors N.



To complete the proof we show that:

1. the probability that $y = x$ or $y = -x$ is equal to $1/2$,

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

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

"the probability π that $y = x$ or $y = -x$ is equal to $1/2$"

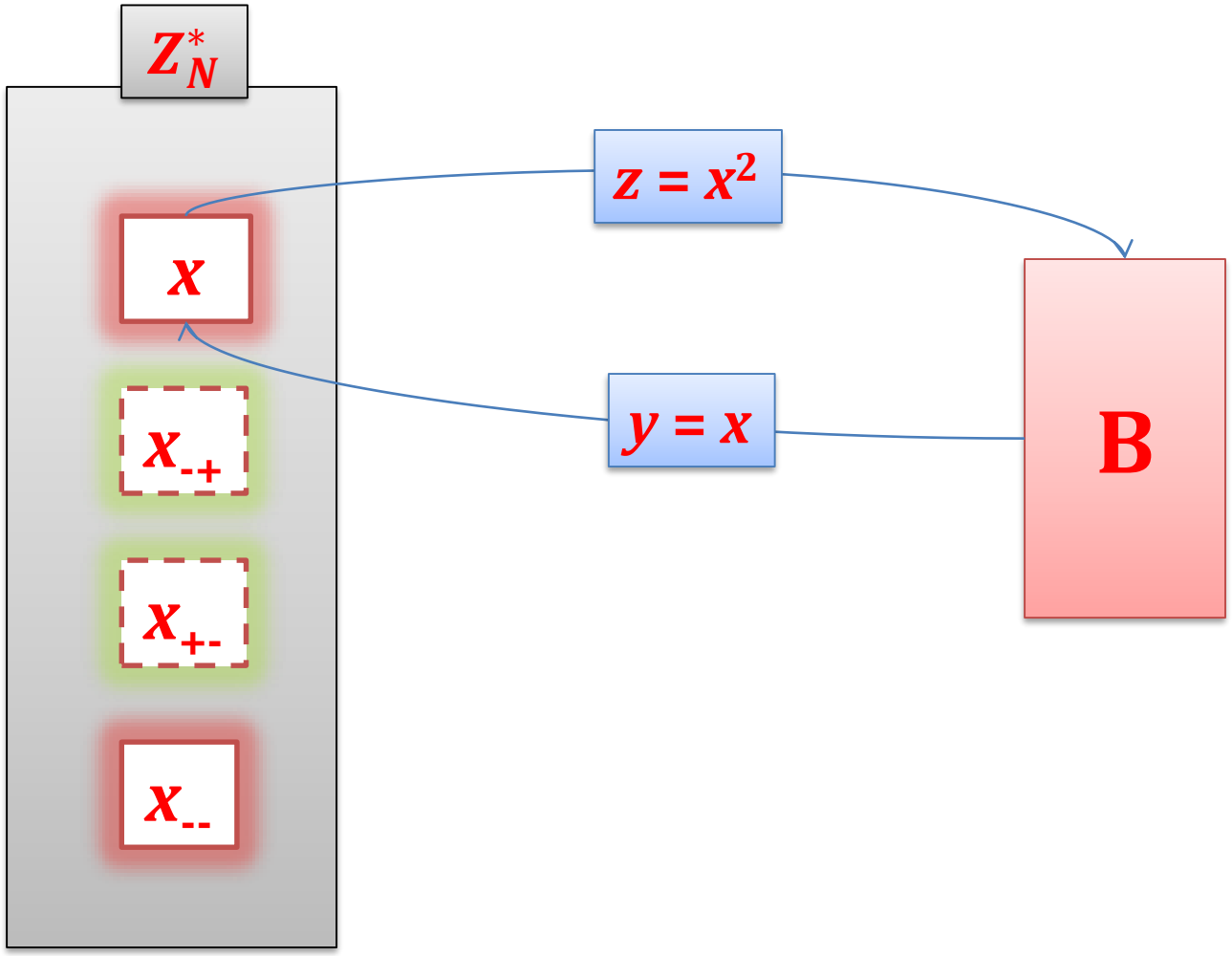
Recall that every $z = x^2$ has the square roots x_{++} and x_{+-}, x_{-+}, x_{--} such that:

- $x_{++} = x \pmod p$ and $x_{++} = x \pmod q$
- $x_{+-} = x \pmod p$ and $x_{+-} = -x \pmod q$
- $x_{-+} = -x \pmod p$ and $x_{-+} = x \pmod q$
- $x_{--} = -x \pmod p$ and $x_{--} = -x \pmod q$

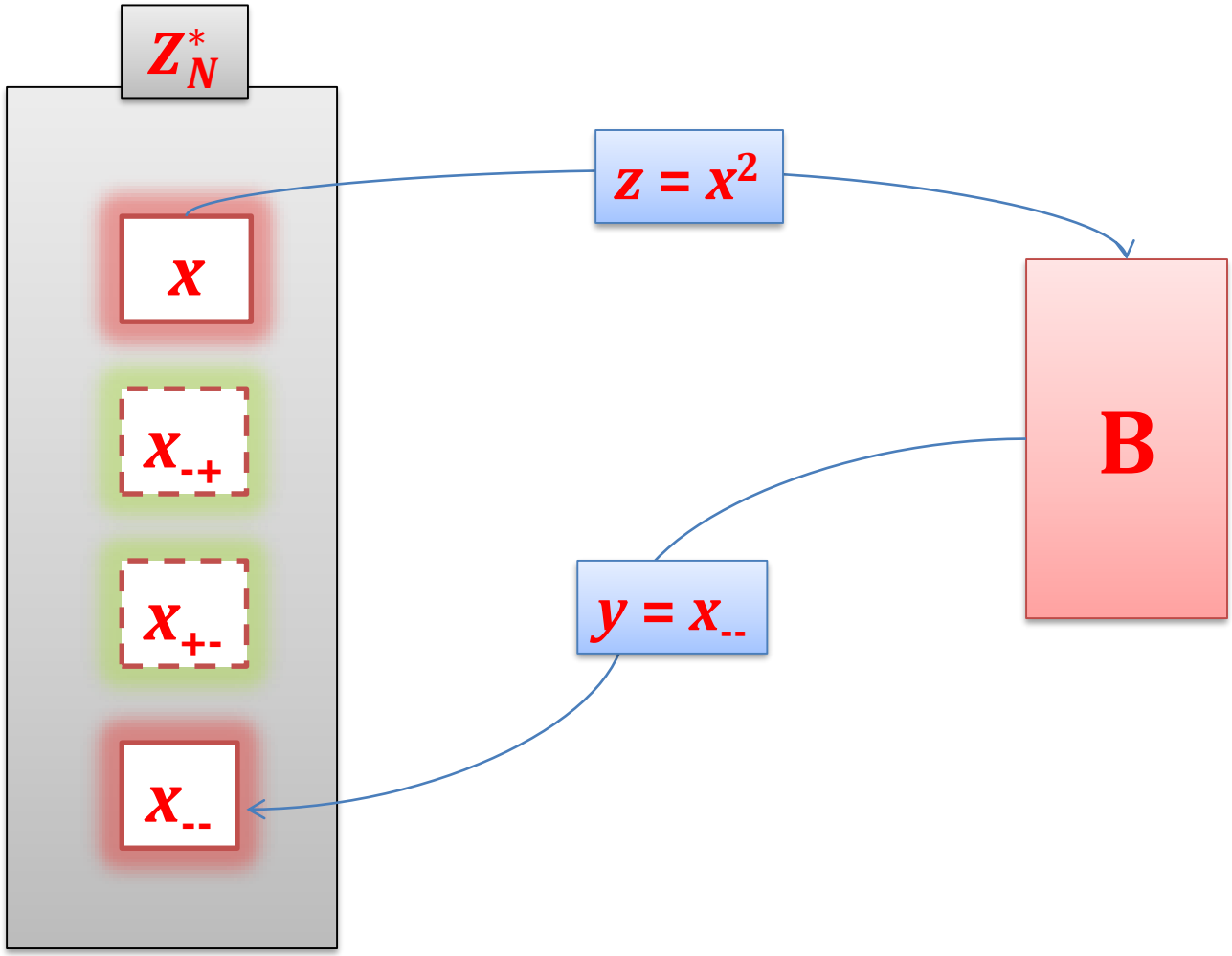
equals to x

equals to $-x$

If we are unlucky it always happens that:

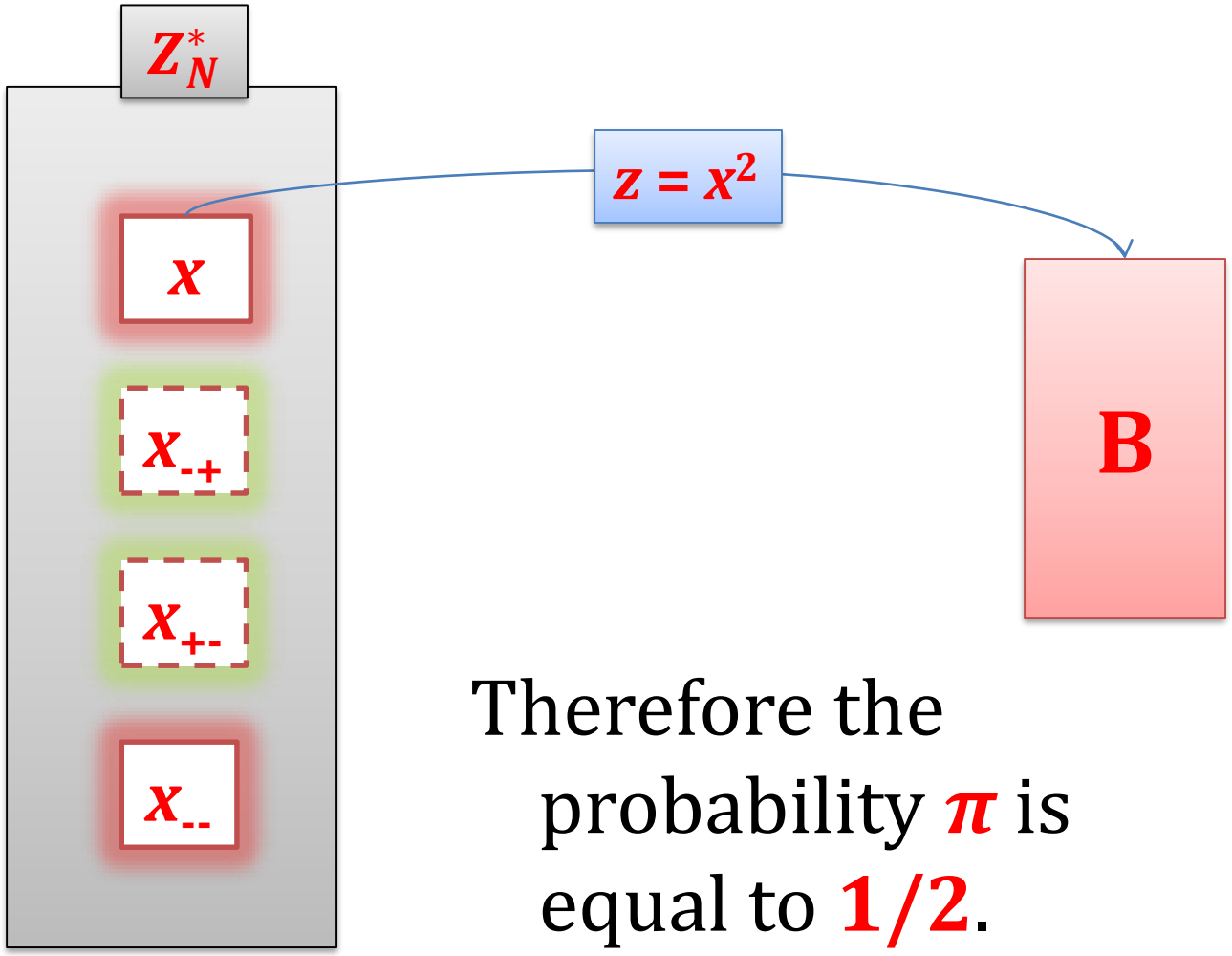


Or:



Observation

Since x is chosen randomly each x_{++}, x_{+-}, x_{-+}, x_{--} is chosen with the same probability



Therefore the probability π is equal to $1/2$.

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

We know that y is such that

$$y = x \pmod{p} \text{ and } y = -x \pmod{q}$$

(the other case is symmetric)

Hence $y \neq x \bmod N$, and therefore $y - x \neq 0 \bmod N$.

On the other hand:

$$y - x = 0 \bmod p$$

Therefore

$$\gcd(N, y - x) = p.$$

QED

Outlook

Groups that we have seen:

- Z_p^* — **hard problem: discrete log**
- Z_N^* for $N = pq$ — **hard problem: computing the eth root**
- subgroups: $\mathbf{QR}_p$ and $\mathbf{QR}_N$

Other interesting groups

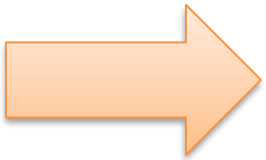
- multiplicative groups of a field **GF(2^p)**,
- groups based on the **elliptic curves**

advantage:
much smaller key
size in practive

we will now talk
about it now

Plan

1. Role of number theory in cryptography
2. Classical problems in computational number theory
3. Finite groups
4. Cyclic groups, discrete log
5. Group Z_N^* and its subgroups
6. Elliptic curves



Elliptic curves over the reals

Let $a,b \in \mathbf{R}$ be two numbers such that

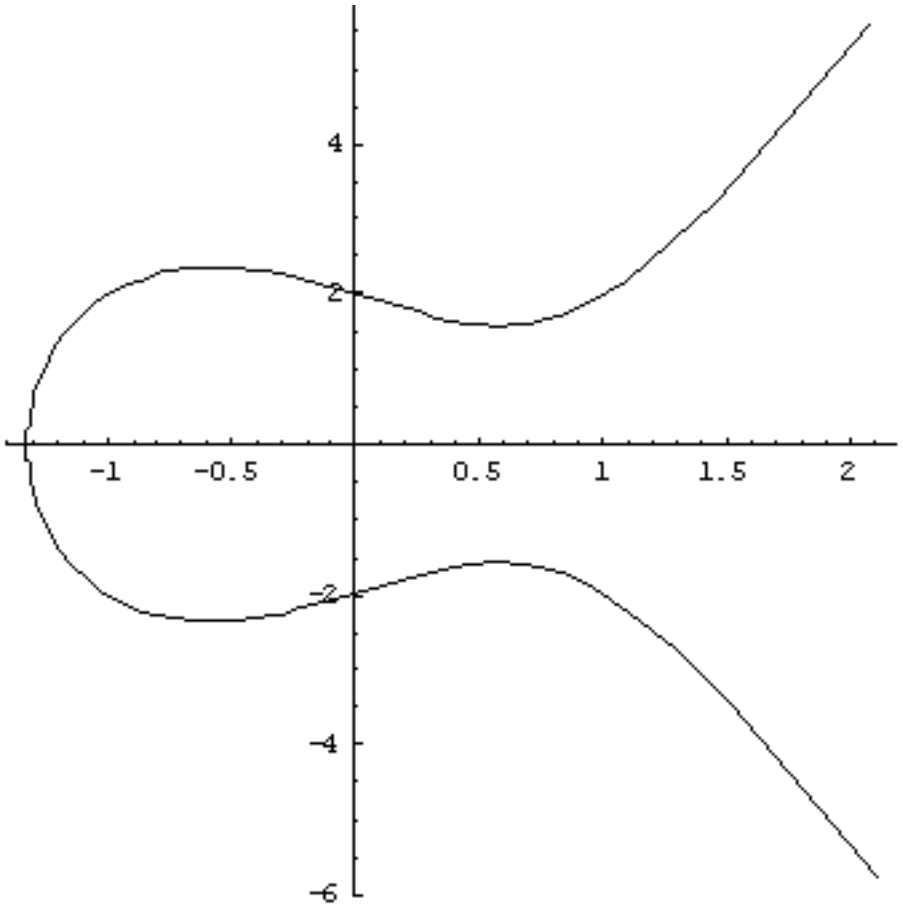
$$4a^3 + 27b^2 \neq 0$$

A **non-singular elliptic curve** is a set **E** of solutions $(x,y) \in \mathbf{R}^2$ to the equation

$$y^2 = x^3 + ax + b$$

together with a special point $\mathcal{O}$ called the **point in infinity**.

Example $y^2 = 4x^3 - 4x + 4$



An abelian group over an elliptic curve

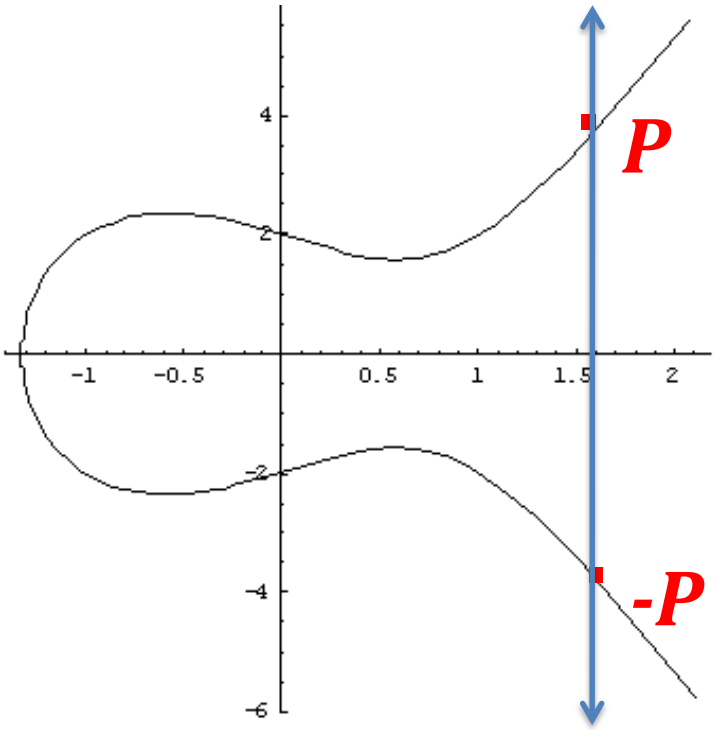
E – elliptic curve

(E,+) – a group

neutral element: $\mathcal{O}$

inverse of $P = (x,y)$:

$P = (x,-y)$



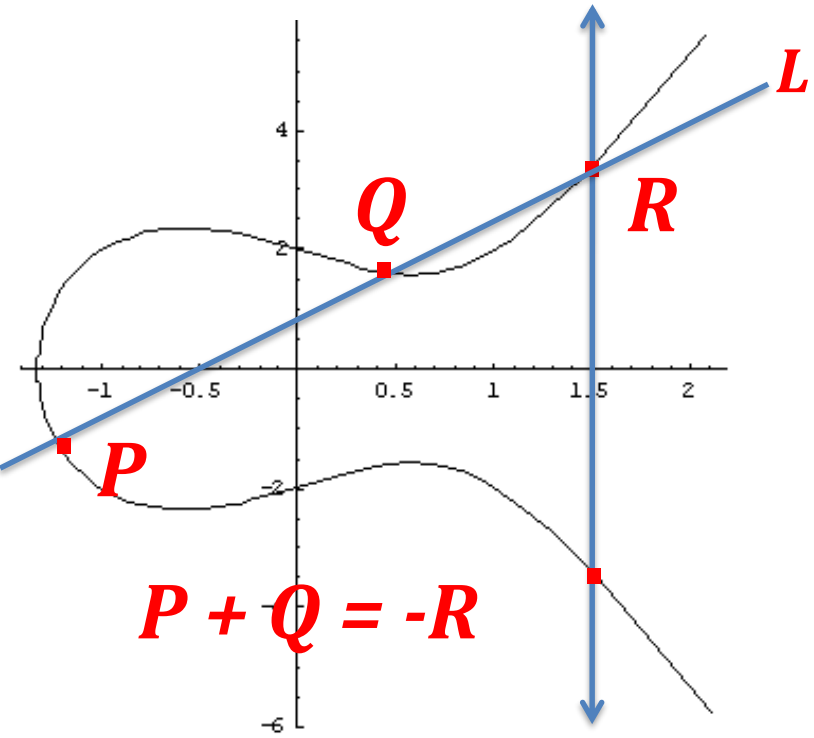
“Addition”

Suppose $P,Q \in E \setminus \{\mathcal{O}\}$ where $P=(x_1,y_1)$ and $Q=(x_2,y_2)$. Consider the following cases:

1. $x_1 \neq x_2$
2. $x_1 = x_2$ and $y_1 = -y_2$
3. $x_1 = x_2$ and $y_1 = y_2$

Case 1: $x_1 \neq x_2$

$P=(x_1,y_1)$ and $Q=(x_2,y_2)$



L – line through **P** and **Q**

Fact

L intersects **E** in exactly one point $R = (x_3,y_3)$.

where:

$x_3 = \lambda^2 - x_1 - x_2$

$y_3 = \lambda(x_1 - x_3) - y_1$

and

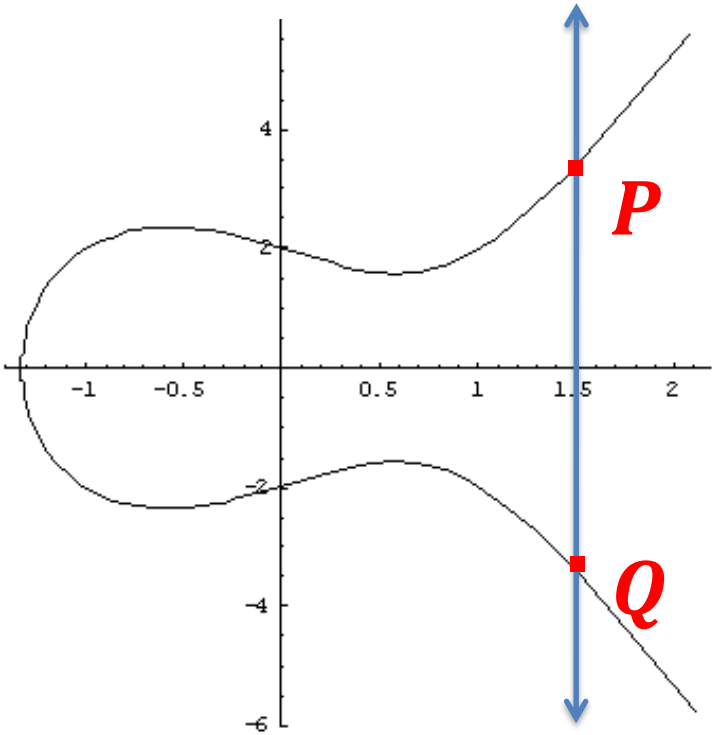
$\lambda = (y_2 - y_1)/(x_2 - x_1)$

Case 2:

$x_1 = x_2$ and $y_1 = -y_2$

$P=(x_1,y_1)$ and $Q=(x_2,y_2)$

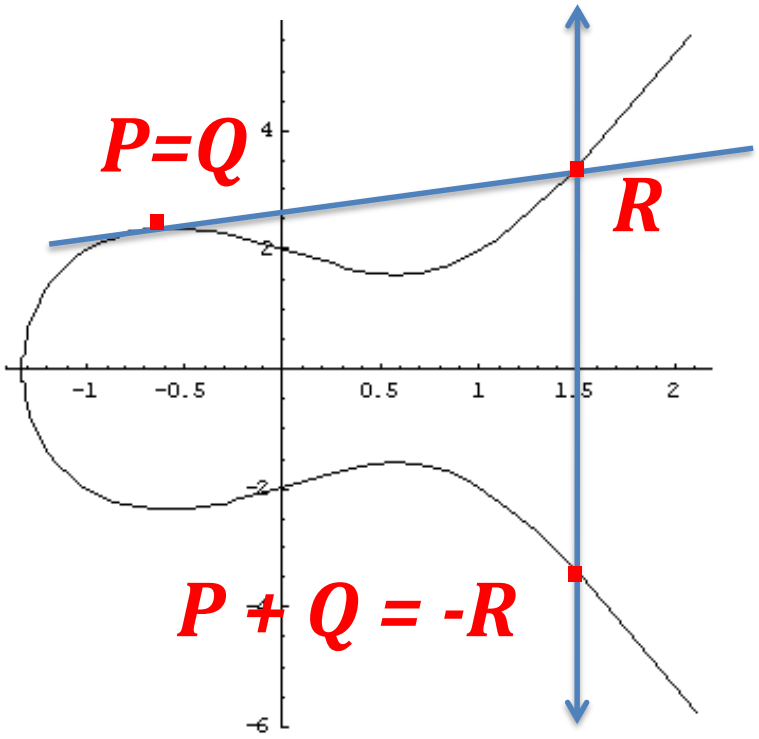
$P + Q = \mathcal{O}$



Case 3:

$x_1 = x_2$ and $y_1 = y_2$

$P=(x_1,y_1)$ and $Q=(x_2,y_2)$



L – line tangent to E at point R

Fact

L intersects E in exactly one point $R = (x_3,y_3)$.

where:

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

and

$$\lambda = (3x_1^2 y_2 + a)/(2y_1)$$

How to prove that this is a group?

Easy to see:

- addition is closed on the set **E**
- addition is commutative
- $\mathcal{O}$ is an identity
- every point has an inverse

What remains is **associativity** **(exercise).**

How to use these groups in cryptography?

Instead of the reals use some finite field.

For example Z_p, where p is prime.

All the formulas remain the same!

Example

x	$x^3 + x + 6 \bmod 11$	quadratic residue?	y
0	6	no	
1	8	no	
2	5	yes	4,7
3	3	yes	5,6
4	8	no	
5	4	yes	2,9
6	8	no	
7	4	yes	2,9
8	9	yes	3,8
9	7	no	
10	4	yes	2,9

Hasse's Theorem

Let **E** be an elliptic curve defined over $\boldsymbol{Z_p}$ where $\boldsymbol{p > 3}$ is prime.

$$\boldsymbol{p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p}}$$

How to use the elliptic curves in cryptography?

(E,+) - elliptic curve

Sometimes **(E,+)** is cyclic or it contains a large cyclic subgroup **(E',+)**.

There are examples of such **(E,+)** or **(E',+)** where the **discrete-log problem** is believed to be computationally hard!

©2018 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*.