Group Generators

Let G be a finite group of order m and let $g \in G$.

Define the set:

$$\langle g \rangle = \{g^0, g^1, g^2, \ldots, g^m\}$$

How many elements are in $\langle g \rangle$?

Group Generators

Let G be a finite group of order m and let $g \in G$.

Define the set:

$$\langle g \rangle = \{g^0, g^1, g^2, \ldots, g^m\}$$

How many elements are in $\langle g \rangle$?

- If g is the identity element, then $g^0 = 1$, $g^1 = 1$, $g^2 = 1$, $\ldots$

Group Generators

Let G be a finite group of order m and let $g \in G$.

Define the set:

$$\langle g \rangle = \{g^0, g^1, g^2, \ldots, g^m\}$$

How many elements are in $\langle g \rangle$?

- If g is the identity element, then $g^0 = 1$, $g^1 = 1$, $g^2 = 1$, ... $\implies$ it can happen that $|\langle g \rangle| = 1$

Group Generators

Let G be a finite group of order m and let $g \in G$.

Define the set:

$$\langle g \rangle = \{g^0, g^1, g^2, \ldots, g^m\}$$

How many elements are in $\langle g \rangle$?

- If g is the identity element, then $g^0 = 1$, $g^1 = 1$, $g^2 = 1$, $\ldots$ $\implies$ it can happen that $|\langle g \rangle| = 1$
- On the other hand, we know that $g^m = 1$, hence...

  $g^m = g^0$, $g^{m+1} = g$, $g^{m+2} = g^2$

Group Generators

Let G be a finite group of order m and let $g \in G$.

Define the set:

$$\langle g \rangle = \{g^0, g^1, g^2, \ldots, g^m\}$$

How many elements are in $\langle g \rangle$?

- If g is the identity element, then $g^0 = 1$, $g^1 = 1$, $g^2 = 1$, $\ldots$ $\implies$ it can happen that $|\langle g \rangle| = 1$
- On the other hand, we know that $g^m = 1$, hence...

  $g^m = g^0$, $g^{m+1} = g$, $g^{m+2} = g^2$ $\implies$ $|\langle g \rangle| \leq m$

Group Generators

Let G be a finite group of order m and let $g \in G$.

Define the set:

$$\langle g \rangle = \{g^0, g^1, g^2, \ldots, g^m\}$$

How many elements are in $\langle g \rangle$?

- If g is the identity element, then $g^0 = 1$, $g^1 = 1$, $g^2 = 1$, $\ldots$ $\Longrightarrow$ it can happen that $|\langle g \rangle| = 1$
- On the other hand, we know that $g^m = 1$, hence...

  $g^m = g^0$, $g^{m+1} = g$, $g^{m+2} = g^2$ $\Longrightarrow$ $|\langle g \rangle| \leq m$

If $\langle g \rangle$ contains all m elements, then g is a **generator** of G.

- We can obtain all elements in G (in **some** order) by exponentiating g.

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)?
    - Is 1 a generator?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)?
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No
- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)?
  - Is 4 a generator?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)?
  - Is 4 a generator? $\langle 4 \rangle = \{1, 4, 7\} \neq \mathbb{Z}_N$ No

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)?
  - Is 4 a generator? $\langle 4 \rangle = \{1, 4, 7\} \neq \mathbb{Z}_N$ No
  - Is 2 a generator?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)? Yes!
  - Is 4 a generator? $\langle 4 \rangle = \{1, 4, 7\} \neq \mathbb{Z}_N$ No
  - Is 2 a generator? $\langle 2 \rangle = \{1, 2, 4, 8, 7, 5\} = \mathbb{Z}_N$ Yes

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)? Yes!
  - Is 4 a generator? $\langle 4 \rangle = \{1, 4, 7\} \neq \mathbb{Z}_N$ No
  - Is 2 a generator? $\langle 2 \rangle = \{1, 2, 4, 8, 7, 5\} = \mathbb{Z}_N$ Yes
  - Is 5 a generator?

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)? Yes!
  - Is 4 a generator? $\langle 4 \rangle = \{1, 4, 7\} \neq \mathbb{Z}_N$ No
  - Is 2 a generator? $\langle 2 \rangle = \{1, 2, 4, 8, 7, 5\} = \mathbb{Z}_N$ Yes
  - Is 5 a generator? $\langle 5 \rangle = \{1, 5, 7, 8, 4, 2\} = \mathbb{Z}_N$ Yes

Cyclic Groups

If G has a generator, then G is called a **cyclic group**

- A cyclic group can have multiple generators
- Not every element of a cyclic group is a generator

Examples:

- Is $\mathbb{Z}_8$ cyclic (under addition modulo 8)? Yes!
  - Is 1 a generator? $\langle 1 \rangle = \{0, 1, 2, 3, \ldots, 7\} = \mathbb{Z}_8$ Yes
  - Is 2 a generator? $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$ No

- Is $\mathbb{Z}_9^*$ cyclic (under multiplication modulo 9)? Yes!
  - Is 4 a generator? $\langle 4 \rangle = \{1, 4, 7\} \neq \mathbb{Z}_N$ No
  - Is 2 a generator? $\langle 2 \rangle = \{1, 2, 4, 8, 7, 5\} = \mathbb{Z}_N$ Yes
  - Is 5 a generator? $\langle 5 \rangle = \{1, 5, 7, 8, 4, 2\} = \mathbb{Z}_N$ Yes

Notice that the elements are generated in a different order

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic?

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic?

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

Theorem: Any group of prime order p is cyclic, and every non-identity element is a generator

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

Theorem: Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

> **Theorem:** Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1$

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

> **Theorem:** Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1 \quad \implies g^{p \bmod i} = 1$

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

Theorem: Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1 \quad \Longrightarrow \; g^{p \bmod i} = 1$

Since $p \bmod i < i$, the choice of i ensures that $p \bmod i = 0$

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

Theorem: Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1 \quad \Longrightarrow \; g^{p \bmod i} = 1$

Since $p \bmod i < i$, the choice of i ensures that $p \bmod i = 0 \quad \Longrightarrow \; i$ is a divisor of p

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

> **Theorem:** Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1 \quad \Longrightarrow \; g^{p \bmod i} = 1$

Since $p \bmod i < i$, the choice of i ensures that $p \bmod i = 0 \qquad \Longrightarrow \; i$ is a divisor of p

The only divisor of p greater than 1 is p

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

> **Theorem:** Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1 \quad \Longrightarrow \quad g^{p \bmod i} = 1$

Since $p \bmod i < i$, the choice of i ensures that $p \bmod i = 0 \quad \Longrightarrow \quad i$ is a divisor of p

The only divisor of p greater than 1 is $p \quad \Longrightarrow \quad i = p$

Cyclic Groups

- Is $\mathbb{Z}_{12}^*$ cyclic? No. Recall that $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$
  - $\langle 5 \rangle = \{1, 5\}$
  - $\langle 7 \rangle = \{1, 7\}$
  - $\langle 11 \rangle = \{1, 11\}$

When is a a group cyclic? A sufficient condition:

> **Theorem:** Any group of prime order p is cyclic, and every non-identity element is a generator

Proof: Fix a $g \in G$ with $g \neq 1$ and let i be the smallest positive integer such that $g^i = 1$.

This integer exists by Fermat's little theorem and we have $i > 1$ since $g^1 = g \neq 1$.

$g^p = 1 \quad \Longrightarrow \; g^{p \bmod i} = 1$

Since $p \bmod i < i$, the choice of i ensures that $p \bmod i = 0 \qquad \Longrightarrow \; i$ is a divisor of p

The only divisor of p greater than 1 is $p \quad \Longrightarrow \; i = p \; \Longrightarrow \; g$ is a generator.

$\square$

Cyclic Groups: Sufficient Conditions

Theorem: Any group of prime order p is cyclic, and every non-identity element is a generator

Theorem: If p is prime then $\mathbb{Z}_p^*$ is cyclic

Cyclic Groups: Sufficient Conditions

Theorem: Any group of prime order p is cyclic, and every non-identity element is a generator

$$\not\Downarrow$$

Theorem: If p is prime then $\mathbb{Z}_p^*$ is cyclic

Notice that the order of $\mathbb{Z}_p^*$ is $\phi(p) = p - 1$, which is not prime (for $p > 3$)

Cyclic Groups: Sampling and Discrete Logarithms

Let G be a cyclic group of order m, and let g be a generator

- We can easily sample (u.a.r.) an element h from G

Cyclic Groups: Sampling and Discrete Logarithms

Let G be a cyclic group of order m, and let g be a generator

- We can easily sample (u.a.r.) an element h from G

> - Choose $x \in \{0, 1, 2, \ldots, m-1\}$ u.a.r.
> - Compute $h = g^x$
> - Return h

Cyclic Groups: Sampling and Discrete Logarithms

Let G be a cyclic group of order m, and let g be a generator

- We can easily sample (u.a.r.) an element h from G

> - Choose $x \in \{0, 1, 2, \ldots, m-1\}$ u.a.r.
> - Compute $h = g^x$
> - Return h

- Given an element $h \in G$, there is a unique value $x \in \{0, 1, \ldots, m-1\}$ such that $g^x = h$

Cyclic Groups: Sampling and Discrete Logarithms

Let G be a cyclic group of order m, and let g be a generator

- We can easily sample (u.a.r.) an element h from G

> - Choose $x \in \{0, 1, 2, \ldots, m-1\}$ u.a.r.
> - Compute $h = g^x$
> - Return h

- Given an element $h \in G$, there is a unique value $x \in \{0, 1, \ldots, m-1\}$ such that $g^x = h$

Definition: the **discrete logarithm** of h with respect to g (in the group G of order m) is denoted by $\log_g h$ and is the unique value $x \in \{0, 1, \ldots, m-1\}$ such that $g^x = h$.

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$?

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$?

$$2^0 = 1 \qquad 2^1 = 2 \qquad 2^2 = 4 \qquad 2^3 = 8 \qquad 2^4 = 5 \qquad 2^5 = 10 \qquad 2^6 = 9$$

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$? $\log_2 9 = 6$

$2^0 = 1$ $2^1 = 2$ $2^2 = 4$ $2^3 = 8$ $2^4 = 5$ $2^5 = 10$ $2^6 = 9$

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$? $\qquad \log_2 9 = 6$

$$2^0 = 1 \qquad 2^1 = 2 \qquad 2^2 = 4 \qquad 2^3 = 8 \qquad 2^4 = 5 \qquad 2^5 = 10 \qquad 2^6 = 9$$

What is $\log_8 6$ in $\mathbb{Z}_{11}^*$?

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$? $\log_2 9 = 6$

$2^0 = 1$ $2^1 = 2$ $2^2 = 4$ $2^3 = 8$ $2^4 = 5$ $2^5 = 10$ $2^6 = 9$

What is $\log_8 6$ in $\mathbb{Z}_{11}^*$?

$8^0 = 1$ $8^1 = 8$ $8^2 = 9$ $8^3 = 6$

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$? $\quad \log_2 9 = 6$

$2^0 = 1 \quad 2^1 = 2 \quad 2^2 = 4 \quad 2^3 = 8 \quad 2^4 = 5 \quad 2^5 = 10 \quad 2^6 = 9$

What is $\log_8 6$ in $\mathbb{Z}_{11}^*$? $\quad \log_8 6 = 3$

$8^0 = 1 \quad 8^1 = 8 \quad 8^2 = 9 \quad 8^3 = 6$

Discrete Logarithms

What is $\log_2 9$ in $\mathbb{Z}_{11}^*$? $\qquad \log_2 9 = 6$

$2^0 = 1 \qquad 2^1 = 2 \qquad 2^2 = 4 \qquad 2^3 = 8 \qquad 2^4 = 5 \qquad 2^5 = 10 \qquad \boxed{2^6 = 9}$

What is $\log_8 6$ in $\mathbb{Z}_{11}^*$? $\qquad \log_8 6 = 3$

$8^0 = 1 \qquad 8^1 = 8 \qquad 8^2 = 9 \qquad \boxed{8^3 = 6}$

What is $\log_2 1656755742$ in $\mathbb{Z}_{3092091139}^*$?

Discrete Logarithms

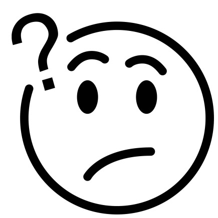
What is $\log_2 9$ in $\mathbb{Z}_{11}^*$? $\log_2 9 = 6$

$2^0 = 1$ $2^1 = 2$ $2^2 = 4$ $2^3 = 8$ $2^4 = 5$ $2^5 = 10$ $2^6 = 9$

What is $\log_8 6$ in $\mathbb{Z}_{11}^*$? $\log_8 6 = 3$

$8^0 = 1$ $8^1 = 8$ $8^2 = 9$ $8^3 = 6$

What is $\log_2 1656755742$ in $\mathbb{Z}_{3092091139}^*$?



The Discrete Logarithm Problem

The **discrete logarithm problem in G:** given a generator g and an element h, compute $\log_g h$

The Discrete Logarithm Problem

The **discrete logarithm problem in G:** given a generator g and an element h, compute $\log_g h$

Discrete logarithm assumption in G (informal):
Solving the discrete logarithm problem in G is hard when h is chosen u.a.r.

The Discrete Logarithm Problem

The **discrete logarithm problem in G:** given a generator g and an element h, compute $\log_g h$

Discrete logarithm assumption in G (informal):
Solving the discrete logarithm problem in G is hard when h is chosen u.a.r.

How do we formalize this?

The Discrete Logarithm Problem

The **discrete logarithm problem in G:** given a generator g and an element h, compute $\log_g h$

Discrete logarithm assumption in G (informal):
Solving the discrete logarithm problem in G is hard when h is chosen u.a.r.

How do we formalize this?

Let $\mathcal{G}$ be a polynomial-time group-generation algorithm that takes 1^n as input, and outputs:

- (a description of) a cyclic group G;
- the order q of G with $\log q \geq n$;
- a generator g of G.

The Discrete Logarithm Assumption

For a group-generation algorithm $\mathcal{G}$ and an algorithm $\mathcal{A}$, define the experiment $\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n)$ as:

- Run $\mathcal{G}(1^n)$ to obtain (G, q, g), where G is a cyclic group of order q (and q is an n-bit integer), and g is a generator of G.
- Choose a uniform $h \in G$.
- G, q, g and h are given to $\mathcal{A}$
- $\mathcal{A}$ outputs $x \in \{0, \ldots q-1\}$
- The outcome of the experiment is 1 if $g^x = h$. Otherwise the outcome is 0.

The Discrete Logarithm Assumption

For a group-generation algorithm $\mathcal{G}$ and an algorithm $\mathcal{A}$, define the experiment $\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n)$ as:

- Run $\mathcal{G}(1^n)$ to obtain (G, q, g), where G is a cyclic group of order q (and q is an n-bit integer), and g is a generator of G.
- Choose a uniform $h \in G$.
- G, q, g and h are given to $\mathcal{A}$
- $\mathcal{A}$ outputs $x \in \{0, \ldots q - 1\}$
- The outcome of the experiment is 1 if $g^x = h$. Otherwise the outcome is 0.

Definition The discrete-logarithm problem is hard relative to $\mathcal{G}$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function ε such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] \leq \varepsilon(n).$$

The Discrete Logarithm Assumption

For a group-generation algorithm $\mathcal{G}$ and an algorithm $\mathcal{A}$, define the experiment $\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n)$ as:

- Run $\mathcal{G}(1^n)$ to obtain (G, q, g), where G is a cyclic group of order q (and q is an n-bit integer), and g is a generator of G.
- Choose a uniform $h \in G$.
- G, q, g and h are given to $\mathcal{A}$
- $\mathcal{A}$ outputs $x \in \{0, \ldots q-1\}$
- The outcome of the experiment is 1 if $g^x = h$. Otherwise the outcome is 0.

Definition The discrete-logarithm problem is hard relative to $\mathcal{G}$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function ε such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] \leq \varepsilon(n).$$

The discrete logarithm assumption: there exists a group-generation algorithm $\mathcal{G}$ for which the discrete-logarithm problem is hard.

The Diffie-Hellman Problem(s)

We need two more related (but not equivalent) assumptions:

Given $g, h_1, h_2 \in G$, define: $\mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

The Diffie-Hellman Problem(s)

We need two more related (but not equivalent) assumptions:

Given $g, h_1, h_2 \in G$, define: $\mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

In other words, if $g^{x_1} = h_1$ and $g^{x_2} = h_2$ then: $\mathsf{DH}_g(h_1, h_2) = g^{x_1 \cdot x_2}$

The Diffie-Hellman Problem(s)

We need two more related (but not equivalent) assumptions:

Given $g, h_1, h_2 \in G$, define: $\mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

In other words, if $g^{x_1} = h_1$ and $g^{x_2} = h_2$ then: $\mathsf{DH}_g(h_1, h_2) = g^{x_1 \cdot x_2} = h_1^{x_2} = h_2^{x_1}$

The Diffie-Hellman Problem(s)

We need two more related (but not equivalent) assumptions:

Given $g, h_1, h_2 \in G$, define: $\mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

In other words, if $g^{x_1} = h_1$ and $g^{x_2} = h_2$ then: $\mathsf{DH}_g(h_1, h_2) = g^{x_1 \cdot x_2} = h_1^{x_2} = h_2^{x_1}$

The **Computational Diffie-Hellman (CDH) problem** is that of computing $\mathsf{DH}_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

The Diffie-Hellman Problem(s)

We need two more related (but not equivalent) assumptions:

Given $g, h_1, h_2 \in G$, define: $\mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

In other words, if $g^{x_1} = h_1$ and $g^{x_2} = h_2$ then: $\mathsf{DH}_g(h_1, h_2) = g^{x_1 \cdot x_2} = h_1^{x_2} = h_2^{x_1}$

The **Computational Diffie-Hellman (CDH) problem** is that of computing $\mathsf{DH}_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

Definition The CDH problem is hard relative to $\mathcal{G}$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function ε such that

$$\Pr[\mathcal{A}(G, q, g, h_1, h_2) = \mathsf{DH}_g(h_1, h_2)] = \varepsilon(n),$$

where the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs (G, q, g), and uniform $h_1, h_2 \in G$ are chosen.

The Diffie-Hellman Problem(s)

We need two more related (but not equivalent) assumptions:

Given $g, h_1, h_2 \in G$, define: $\mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

In other words, if $g^{x_1} = h_1$ and $g^{x_2} = h_2$ then: $\mathsf{DH}_g(h_1, h_2) = g^{x_1 \cdot x_2} = h_1^{x_2} = h_2^{x_1}$

The **Computational Diffie-Hellman (CDH) problem** is that of computing $\mathsf{DH}_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

Definition The CDH problem is hard relative to $\mathcal{G}$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function ε such that

$$\Pr[\mathcal{A}(G, q, g, h_1, h_2) = \mathsf{DH}_g(h_1, h_2)] = \varepsilon(n),$$

where the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs (G, q, g), and uniform $h_1, h_2 \in G$ are chosen.

The CDH assumption: there exists a group-generation algorithm $\mathcal{G}$ for which the CDH problem is hard

The Diffie-Hellman Problem(s)

Given $g, h_1, h_2 \in G$, define: $\quad \mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

The **Decisional Diffie-Hellman (DDH) problem** is that of distinguishing $\mathsf{DH}_g(h_1, h_2)$ (computed as above) from an element chosen u.a.r. from G

The Diffie-Hellman Problem(s)

Given $g, h_1, h_2 \in G$, define: $\quad \mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

The **Decisional Diffie-Hellman (DDH) problem** is that of distinguishing $\mathsf{DH}_g(h_1, h_2)$ (computed as above) from an element chosen u.a.r. from G

Definition The DDH problem is hard relative to $\mathcal{G}$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function ε such that

$$\Big|\ \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy}) = 1]\ \Big| \leq \varepsilon(n),$$

where the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs (G, q, g), and then uniform $x, y, z \in \{0, 1, \ldots, q-1\}$ are chosen (therefore g^x and g^y are uniformly distributed in G).

The Diffie-Hellman Problem(s)

Given $g, h_1, h_2 \in G$, define: $\quad \mathsf{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

The **Decisional Diffie-Hellman (DDH) problem** is that of distinguishing $\mathsf{DH}_g(h_1, h_2)$ (computed as above) from an element chosen u.a.r. from G

Definition The DDH problem is hard relative to $\mathcal{G}$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function ε such that

$$\Big| \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy}) = 1] \Big| \leq \varepsilon(n),$$

where the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs (G, q, g), and then uniform $x, y, z \in \{0, 1, \ldots, q-1\}$ are chosen (therefore g^x and g^y are uniformly distributed in G).

The DDH assumption: there exists a group-generation algorithm $\mathcal{G}$ for which the DDH problem is hard

Examples

The **Computational Diffie-Hellman (CDH) problem** is that of computing $DH_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

- What is $\mathsf{DH}_2(7, 5)$ in Z^*_{11}?

Examples

The **Computational Diffie-Hellman (CDH) problem** is that of computing $DH_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

- What is $\mathsf{DH}_2(7, 5)$ in Z_{11}^*?
- $\langle 2 \rangle = \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\}$

Examples

The **Computational Diffie-Hellman (CDH) problem** is that of computing $DH_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

- What is $\mathrm{DH}_2(7, 5)$ in Z_{11}^*?
- $\langle 2 \rangle = \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\}$
- $\log_2 7 = 7$ and $\log_2 5 = 4$

Examples

The **Computational Diffie-Hellman (CDH) problem** is that of computing $DH_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

- What is $\mathsf{DH}_2(7,5)$ in Z_{11}^*?
- $\langle 2 \rangle = \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\}$
- $\log_2 7 = 7$ and $\log_2 5 = 4$
- $2^{7 \cdot 4} = 2^{28} = 2^{28 \bmod \phi(\mathbb{Z}_{11}^*)} = 2^{28 \bmod 10} = 2^8 = 3$

Examples

The **Computational Diffie-Hellman (CDH) problem** is that of computing $DH_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

- What is $\mathrm{DH}_2(7,5)$ in Z_{11}^*?
- $\langle 2\rangle = \{1,2,4,8,5,10,9,7,3,6\}$
- $\log_2 7 = 7$ and $\log_2 5 = 4$
- $2^{7\cdot 4} = 2^{28} = 2^{28 \bmod \phi(\mathbb{Z}_{11}^*)} = 2^{28 \bmod 10} = 2^8 = 3$

You have polynomial-time to figure that out with non-negligible probability (in a suitable group)

Examples

The **Computational Diffie-Hellman (CDH) problem** is that of computing $DH_g(h_1, h_2)$ given a group G, a generator g, and two elements h_1, and h_2 chosen u.a.r. from G

- What is $\mathrm{DH}_2(7,5)$ in Z_{11}^*?
- $\langle 2 \rangle = \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\}$
- $\log_2 7 = 7$ and $\log_2 5 = 4$
- $2^{7 \cdot 4} = 2^{28} = 2^{28 \bmod \phi(\mathbb{Z}_{11}^*)} = 2^{28 \bmod 10} = 2^8 = 3$

You have polynomial-time to figure that out with non-negligible probability (in a suitable group)

CDH assumption: no algorithm can do that (in a suitable group)

Examples

The **Decisional Diffie-Hellman (DDH) problem** is that of distinguishing $DH_g(h_1, h_2)$ (computed as above) from an element chosen u.a.r. from G

- I'm considering the group $\mathbb{Z}^*_{3092091139}$ and I'm interested in the value $\mathrm{DH}_2(1656755742, 938640663)$
- Is 1994993011 the correct answer, or did I just give you a random element from $\mathbb{Z}^*_{3092091139}$?

Examples

The **Decisional Diffie-Hellman (DDH) problem** is that of distinguishing $DH_g(h_1, h_2)$ (computed as above) from an element chosen u.a.r. from G

- I'm considering the group $\mathbb{Z}^*_{3092091139}$ and I'm interested in the value $\mathrm{DH}_2(1656755742, 938640663)$
- Is 1994993011 the correct answer, or did I just give you a random element from $\mathbb{Z}^*_{3092091139}$?

You have polynomial-time to figure that out (with a non-negligible advantage over random guessing)

Examples

The **Decisional Diffie-Hellman (DDH) problem** is that of distinguishing $DH_g(h_1, h_2)$ (computed as above) from an element chosen u.a.r. from G

- I'm considering the group $\mathbb{Z}^*_{3092091139}$ and I'm interested in the value $\mathrm{DH}_2(1656755742, 938640663)$
- Is 1994993011 the correct answer, or did I just give you a random element from $\mathbb{Z}^*_{3092091139}$?

You have polynomial-time to figure that out (with a non-negligible advantage over random guessing)

DDH assumption: no algorithm can do that (in a suitable group)

Relating the Discrete Logarithm and the DH Problems

The discrete-logarithm problem is hard relative to $\mathcal{G}$

$$\Uparrow$$

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$

$$\Uparrow$$

The Decisional Diffie-Hellman (DDH) problem is hard relative to $\mathcal{G}$

Relating the Discrete Logarithm and the DH Problems

The discrete-logarithm problem is hard relative to $\mathcal{G}$

$\Uparrow$

We don't know whether the converse implication holds.

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$

$\Uparrow$

The Decisional Diffie-Hellman (DDH) problem is hard relative to $\mathcal{G}$

Relating the Discrete Logarithm and the DH Problems

The discrete-logarithm problem is hard relative to $\mathcal{G}$

$$\Uparrow$$

We don't know whether the converse implication holds.

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$

$$\Uparrow \quad \not\Downarrow$$

We know that there are groups for which the the CDH problem is hard but the DDH problem is not hard

The Decisional Diffie-Hellman (DDH) problem is hard relative to $\mathcal{G}$

Hardness of CDH $\Longrightarrow$ Hardness of DL

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$	$\Rightarrow$	The discrete-logarithm problem is hard relative to $\mathcal{G}$

Hardness of CDH $\implies$ Hardness of DL

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$	$\Rightarrow$	The discrete-logarithm problem is hard relative to $\mathcal{G}$

Proof: We show that a polynomial-time algorithm $\mathcal{A}$ that solves the discrete-logarithm problem (i.e., wins the DLog experiment with non-negligible probability) can be used to solve the CDH problem

Suppose that discrete-logarithm problem is not hard w.r.t. G and consider an algorithm $\mathcal{A}$ such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] = \varepsilon(n) \quad \text{where } \varepsilon(n) \text{ is not negligible.}$$

Hardness of CDH $\Longrightarrow$ Hardness of DL

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$ $\Rightarrow$ The discrete-logarithm problem is hard relative to $\mathcal{G}$

Proof: We show that a polynomial-time algorithm $\mathcal{A}$ that solves the discrete-logarithm problem (i.e., wins the DLog experiment with non-negligible probability) can be used to solve the CDH problem

Suppose that discrete-logarithm problem is not hard w.r.t. G and consider an algorithm $\mathcal{A}$ such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] = \varepsilon(n) \quad \text{where } \varepsilon(n) \text{ is not negligible.}$$

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, h_1, h_2
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, h_1 to compute a candidate $x_1 = \log_g h_1$
- $\mathcal{A}'$ outputs $h_2^{x_1}$ (recall that $h_2^{x_1} = (g^{\log_g h_2})^{x_1} = g^{\log_g h_2 \cdot \log_g h_1} = \mathsf{DH}_g(h_1, h_2)$)

Hardness of CDH $\Longrightarrow$ Hardness of DL

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$	$\Rightarrow$	The discrete-logarithm problem is hard relative to $\mathcal{G}$

Proof: We show that a polynomial-time algorithm $\mathcal{A}$ that solves the discrete-logarithm problem (i.e., wins the DLog experiment with non-negligible probability) can be used to solve the CDH problem

Suppose that discrete-logarithm problem is not hard w.r.t. G and consider an algorithm $\mathcal{A}$ such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] = \varepsilon(n) \quad \text{where } \varepsilon(n) \text{ is not negligible.}$$

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, h_1, h_2
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, h_1 to compute a candidate $x_1 = \log_g h_1$
- $\mathcal{A}'$ outputs $h_2^{x_1}$ (recall that $h_2^{x_1} = (g^{\log_g h_2})^{x_1} = g^{\log_g h_2 \cdot \log_g h_1} = \mathsf{DH}_g(h_1, h_2)$)

$$\Pr[\mathcal{A}'(G, q, g, h_1, h_2) = \mathsf{DH}_g(h_1, h_2)] \geq \Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1]$$

(If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds)

Hardness of CDH $\implies$ Hardness of DL

The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$ $\Rightarrow$ The discrete-logarithm problem is hard relative to $\mathcal{G}$

Proof: We show that a polynomial-time algorithm $\mathcal{A}$ that solves the discrete-logarithm problem (i.e., wins the DLog experiment with non-negligible probability) can be used to solve the CDH problem

Suppose that discrete-logarithm problem is not hard w.r.t. G and consider an algorithm $\mathcal{A}$ such that

$$\Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] = \varepsilon(n) \quad \text{where } \varepsilon(n) \text{ is not negligible.}$$

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, h_1, h_2
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, h_1 to compute a candidate $x_1 = \log_g h_1$
- $\mathcal{A}'$ outputs $h_2^{x_1}$ (recall that $h_2^{x_1} = (g^{\log_g h_2})^{x_1} = g^{\log_g h_2 \cdot \log_g h_1} = \mathsf{DH}_g(h_1, h_2)$)

$$\Pr[\mathcal{A}'(G, q, g, h_1, h_2) = \mathsf{DH}_g(h_1, h_2)] \geq \Pr[\mathsf{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] = \varepsilon(n) \qquad \text{non-negligible!}$$

(If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds) $\square$

Hardness of DDH $\Longrightarrow$ Hardness of CDH

The Decisional Diffie-Hellman (DDH) problem is hard relative to $\mathcal{G}$	$\Rightarrow$	The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$

Proof: We show that a polynomial-time algorithm $\mathcal{A}$ that solves the CDH problem (with non-negligible probability) can be used to solve the DDH problem

Suppose that CDH problem is not hard w.r.t. $\mathcal{G}$ and consider an algorithm $\mathcal{A}$ such that

$$\Pr[\mathcal{A}(G, q, g, h_1, h_2) = \mathsf{DH}_g(h_1, h_2)] = \varepsilon(n), \quad \text{where } \varepsilon(n) \text{ is not negligible.}$$

Hardness of DDH $\Longrightarrow$ Hardness of CDH

The Decisional Diffie-Hellman (DDH) problem is hard relative to $\mathcal{G}$	$\Rightarrow$	The Computational Diffie-Hellman (CDH) problem is hard relative to $\mathcal{G}$

Proof: We show that a polynomial-time algorithm $\mathcal{A}$ that solves the CDH problem (with non-negligible probability) can be used to solve the DDH problem

Suppose that CDH problem is not hard w.r.t. $\mathcal{G}$ and consider an algorithm $\mathcal{A}$ such that

$$\Pr[\mathcal{A}(G, q, g, h_1, h_2) = \mathsf{DH}_g(h_1, h_2)] = \varepsilon(n), \quad \text{where } \varepsilon(n) \text{ is not negligible.}$$

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

Hardness of DDH $\implies$ Hardness of CDH (cont.)

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

When $h = g^{xy}$:

- If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] \geq \Pr[\mathcal{A}(G, q, g, g^x, g^y) = g^{xy}] = \varepsilon(n)$$

Hardness of DDH $\Longrightarrow$ Hardness of CDH (cont.)

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

When $h = g^{xy}$:

- If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] \geq \Pr[\mathcal{A}(G, q, g, g^x, g^y) = g^{xy}] = \varepsilon(n)$$

When h is an element chosen u.a.r. from G:

- The value of h does not depend on h'

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] = \Pr[h = h']$$

Hardness of DDH $\implies$ Hardness of CDH (cont.)

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

When $h = g^{xy}$:

- If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] \geq \Pr[\mathcal{A}(G, q, g, g^x, g^y) = g^{xy}] = \varepsilon(n)$$

When h is an element chosen u.a.r. from G:

- The value of h does not depend on h'

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] = \Pr[h = h'] \;\; = \tfrac{1}{q}$$

Hardness of DDH $\implies$ Hardness of CDH (cont.)

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

When $h = g^{xy}$:

- If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] \geq \Pr[\mathcal{A}(G, q, g, g^x, g^y) = g^{xy}] = \varepsilon(n)$$

When h is an element chosen u.a.r. from G:

- The value of h does not depend on h' (recall that q is a n-bit number)

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] = \Pr[h = h'] = \frac{1}{q} \leq \frac{1}{2^{n-1}}$$

negligible

Hardness of DDH $\implies$ Hardness of CDH (cont.)

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

When $h = g^{xy}$:

- If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] \geq \Pr[\mathcal{A}(G, q, g, g^x, g^y) = g^{xy}] = \varepsilon(n)$$

When h is an element chosen u.a.r. from G:

- The value of h does not depend on h' (recall that q is a n-bit number)

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] = \Pr[h = h'] = \frac{1}{q} \leq \frac{1}{2^{n-1}} \quad \text{negligible}$$

$$\Big|\ \Pr[\mathcal{A}'(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}'(G, q, g, g^x, g^y, g^{xy}) = 1]\ \Big| = \left|\ \varepsilon(n) - \frac{1}{q}\ \right|$$

Hardness of DDH $\Longrightarrow$ Hardness of CDH (cont.)

Build $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ takes as input G, q, g, g^x, g^y, h
- $\mathcal{A}'$ simulates $\mathcal{A}$ with inputs G, q, g, g^x, g^y to compute a candidate $h' = g^{xy}$
- $\mathcal{A}'$ outputs 1 if $h' = h$. Otherwise $\mathcal{A}$ outputs 0

When $h = g^{xy}$:

- If $\mathcal{A}$ succeeds then $\mathcal{A}'$ succeeds

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] \geq \Pr[\mathcal{A}(G, q, g, g^x, g^y) = g^{xy}] = \varepsilon(n)$$

When h is an element chosen u.a.r. from G:

- The value of h does not depend on h' (recall that q is a n-bit number)

$$\Pr[\mathcal{A}'(G, q, g, g^x, g^y, h) = 1] = \Pr[h = h'] = \frac{1}{q} \leq \frac{1}{2^{n-1}}$$ negligible

$$\Big| \Pr[\mathcal{A}'(G, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}'(G, q, g, g^x, g^y, g^{xy}) = 1] \Big| = \left| \varepsilon(n) - \frac{1}{q} \right|$$ non-negligible

$\square$

Choice of Groups

The cryptographic schemes can be described in terms of a generic group

- We can focus on the key idea of the construction, ignoring the details of the specific group
- To build the scheme in practice, we can instantiate the theoretical construction with any *suitable* group

Choice of Groups

The cryptographic schemes can be described in terms of a generic group

- We can focus on the key idea of the construction, ignoring the details of the specific group
- To build the scheme in practice, we can instantiate the theoretical construction with any *suitable* group

Before describing the actual constructions, we briefly argue on some possible choices for these groups

Choice of Groups

The cryptographic schemes can be described in terms of a generic group

- We can focus on the key idea of the construction, ignoring the details of the specific group
- To build the scheme in practice, we can instantiate the theoretical construction with any *suitable* group

Before describing the actual constructions, we briefly argue on some possible choices for these groups

We would like the group order to be prime

Choice of Groups

The cryptographic schemes can be described in terms of a generic group

- We can focus on the key idea of the construction, ignoring the details of the specific group
- To build the scheme in practice, we can instantiate the theoretical construction with any *suitable* group

Before describing the actual constructions, we briefly argue on some possible choices for these groups

We would like the group order to be prime

- The discrete-logarithm problem in a group of order q becomes *easier* (not necessarily easy!) if q has (small) prime factors

Choice of Groups

The cryptographic schemes can be described in terms of a generic group

- We can focus on the key idea of the construction, ignoring the details of the specific group
- To build the scheme in practice, we can instantiate the theoretical construction with any *suitable* group

Before describing the actual constructions, we briefly argue on some possible choices for these groups

We would like the group order to be prime

- The discrete-logarithm problem in a group of order q becomes *easier* (not necessarily easy!) if q has (small) prime factors
- The DDH problem is easy if the group order has small prime factors

Choice of Groups

The cryptographic schemes can be described in terms of a generic group

- We can focus on the key idea of the construction, ignoring the details of the specific group
- To build the scheme in practice, we can instantiate the theoretical construction with any *suitable* group

Before describing the actual constructions, we briefly argue on some possible choices for these groups

We would like the group order to be prime

- The discrete-logarithm problem in a group of order q becomes *easier* (not necessarily easy!) if q has (small) prime factors
- The DDH problem is easy if the group order has small prime factors
- Finding a generator in such groups is trivial (pick any element except for the identity)

Choice of Groups: the group $\mathbb{Z}_p^*$

The group $\mathbb{Z}_p^*$, for prime p has several nice properties:

- Easy to represent:
  - — To identify the group, it suffices to know p.
  - — Elements are integers in $\{1, \ldots, p-1\}$.

Choice of Groups: the group $\mathbb{Z}_p^*$

The group $\mathbb{Z}_p^*$, for prime p has several nice properties:

- Easy to represent:
  - — To identify the group, it suffices to know p.
  - — Elements are integers in $\{1, \ldots, p-1\}$.
- Trivial to sample one element and to check whether an element is in $\mathbb{Z}_p^*$.

Choice of Groups: the group $\mathbb{Z}_p^*$

The group $\mathbb{Z}_p^*$, for prime p has several nice properties:

- Easy to represent:
  - — To identify the group, it suffices to know p.
  - — Elements are integers in $\{1, \ldots, p-1\}$.
- Trivial to sample one element and to check whether an element is in $\mathbb{Z}_p^*$.
- Easy to build a group generation algorithm:
  - — Pick a n-bit prime p uniformly at random
  - — Output p (trivial), the order $q = p - 1$ (trivial), an a group generator (can be found in poly-time)

Choice of Groups: the group $\mathbb{Z}_p^*$

The group $\mathbb{Z}_p^*$, for prime p has several nice properties:

- Easy to represent:
  - — To identify the group, it suffices to know p.
  - — Elements are integers in $\{1, \ldots, p-1\}$.
- Trivial to sample one element and to check whether an element is in $\mathbb{Z}_p^*$.
- Easy to build a group generation algorithm:
  - — Pick a n-bit prime p uniformly at random
  - — Output p (trivial), the order $q = p - 1$ (trivial), an a group generator (can be found in poly-time)
- The discrete-logarithm problem is conjectured to be hard on $\mathbb{Z}_p^*$

Choice of Groups: the group $\mathbb{Z}_p^*$

The group $\mathbb{Z}_p^*$, for prime p has several nice properties:

- Easy to represent:
  — To identify the group, it suffices to know p.
  — Elements are integers in $\{1, \ldots, p-1\}$.
- Trivial to sample one element and to check whether an element is in $\mathbb{Z}_p^*$.
- Easy to build a group generation algorithm:
  — Pick a n-bit prime p uniformly at random
  — Output p (trivial), the order $q = p - 1$ (trivial), an a group generator (can be found in poly-time)
- The discrete-logarithm problem is conjectured to be hard on $\mathbb{Z}_p^*$

However

- The order of the group $q = p - 1$ is not a prime number
- The DDH problem is known **not to be hard** in such groups (in general)

Choice of Groups: the group of r-th residues modulo p

Solution:

- Pick two prime numbers p, q such that $p = qr + 1$ for some r
- Consider the set of r*-th residues modulo* p, defined as:

$$G = \{h^r \pmod p \mid h \in \mathbb{Z}_p^*\}$$

Choice of Groups: the group of r-th residues modulo p

Solution:

- Pick two prime numbers p, q such that $p = qr + 1$ for some r
- Consider the set of r-*th residues modulo* p, defined as:

$$G = \{h^r \pmod{p} \mid h \in \mathbb{Z}_p^*\}$$

- The set G is a group (under multiplication modulo p)

Choice of Groups: the group of r-th residues modulo p

Solution:

- Pick two prime numbers p, q such that $p = qr + 1$ for some r
- Consider the set of r*-th residues modulo* p, defined as:

$$G = \{h^r \pmod{p} \mid h \in \mathbb{Z}_p^*\}$$

- The set G is a group (under multiplication modulo p)
- The order of G is q

Choice of Groups: the group of r-th residues modulo p

Solution:

- Pick two prime numbers p, q such that $p = qr + 1$ for some r
- Consider the set of *r-th residues modulo p*, defined as:

$$G = \{h^r \pmod{p} \mid h \in \mathbb{Z}_p^*\}$$

- The set G is a group (under multiplication modulo p)
- The order of G is q
- We can quickly pick a uniform element in G: pick $h \in \mathbb{Z}_p^*$ and return h^r

Choice of Groups: the group of r-th residues modulo p

Solution:

- Pick two prime numbers p, q such that $p = qr + 1$ for some r
- Consider the set of *r-th residues modulo* p, defined as:

$$G = \{h^r \pmod{p} \mid h \in \mathbb{Z}_p^*\}$$

- The set G is a group (under multiplication modulo p)
- The order of G is q
- We can quickly pick a uniform element in G: pick $h \in \mathbb{Z}_p^*$ and return h^r
- There is a polynomial-time algorithm to test whether an element h is in G

Choice of Groups: the group of r-th residues modulo p

Solution:

- Pick two prime numbers p, q such that $p = qr + 1$ for some r
- Consider the set of r*-th residues modulo* p, defined as:

$$G = \{h^r \pmod{p} \mid h \in \mathbb{Z}_p^*\}$$

- The set G is a group (under multiplication modulo p)
- The order of G is q
- We can quickly pick a uniform element in G: pick $h \in \mathbb{Z}_p^*$ and return h^r
- There is a polynomial-time algorithm to test whether an element h is in G
- There is a polynomial-time algorithm to find a generator of G

Choice of Groups: other options

- Subgroups of finite fields when using the polynomial representation for elements

- Elliptic curves

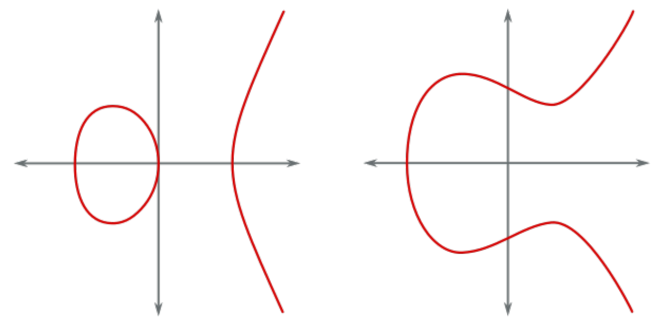
  - Consider cubic equations modulo p with two variables x, y of the form

$$y^2 = x^3 + Ax + B \pmod p$$

  - Let $E(\mathbb{Z}_p)$ be the set of points $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ that satisfy the equation, plus a special *point at infinity* $\mathcal{O}$

  - It is possible to define a suitable *addition* operation over $E(\mathbb{Z}_p)$

  - The set $E(\mathbb{Z}_p)$ is a group under the addition operation, and the identity element is $\mathcal{O}$



$y^2 = x^3 - x$ $\qquad$ $y^2 = x^3 - x + 1$