

Introduction to Cryptography

Prepared by Dr. Mohammad Alkhatib

Associate Professor

Major: Information Security

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

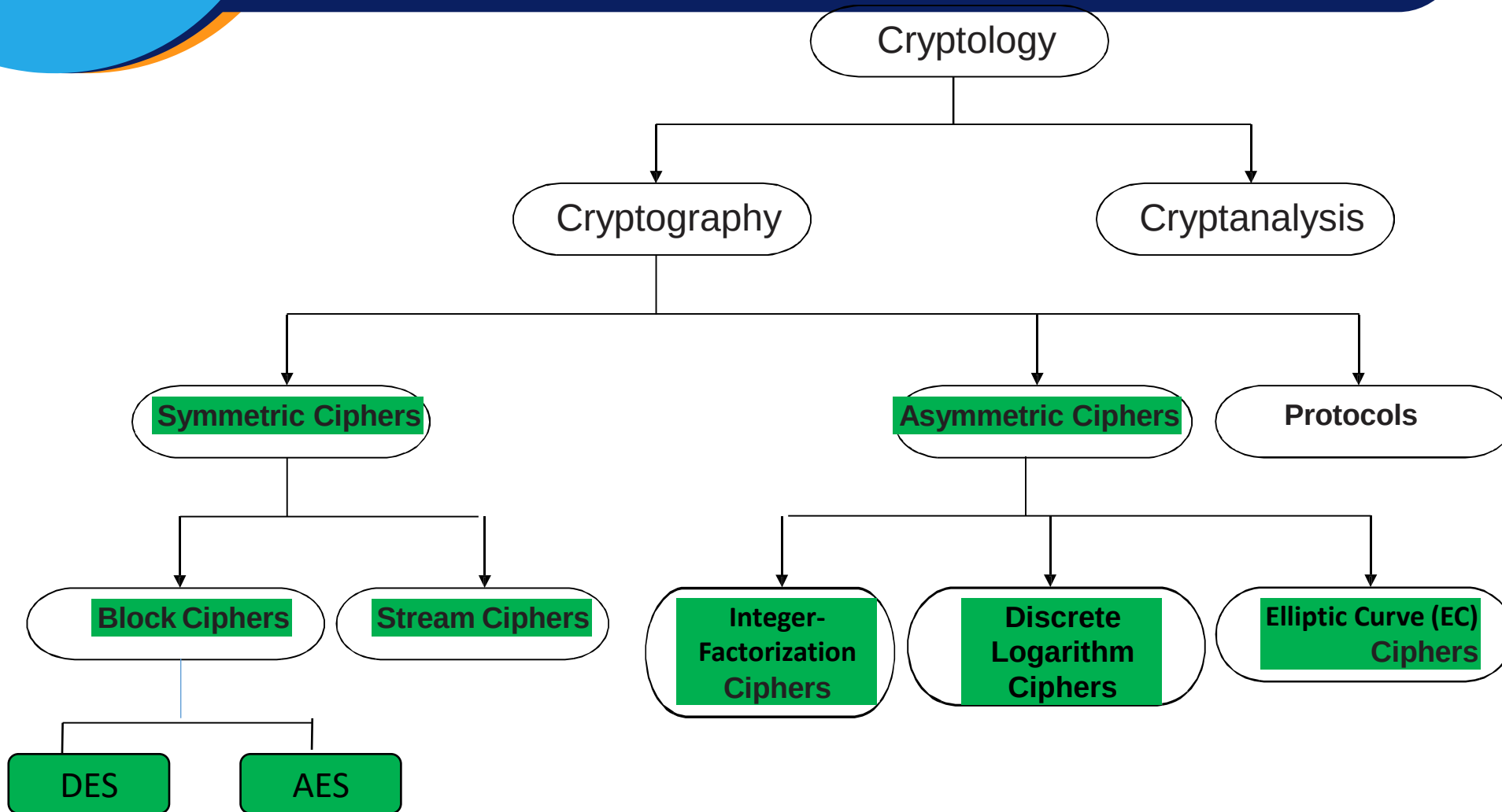
Lecture #9: Key Exchange Protocol



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

Topics to be covered

- Introduction
- Diffie–Hellman Key Exchange Protocol
- Diffie–Hellman Example
- Mathematical Concepts essentials for Discrete Logarithm Problem
- Security of Diffie–Hellman protocol



- ❑ The Diffie–Hellman key exchange
- ❑ Cyclic groups which are important for a deeper understanding of Diffie–Hellman key exchange
- ❑ The discrete logarithm problem, which is of fundamental importance for many practical public-key algorithms

- ❑ The security of many cryptographic schemes relies on the computational intractability of finding solutions to the *Discrete Logarithm Problem (DLP)*.
- ❑ Well-known examples of such schemes are the Diffie–Hellman key exchange and the Elgamal encryption scheme.

Diffie–Hellman key exchange

- ❑ The *Diffie–Hellman key exchange (DHKE)* is very simple and provides a practical solution to the key distribution problem,
- ❑ DHKE enables two parties to derive a common secret key by communicating over an insecure channel.

Diffie–Hellman key exchange

- ❑ The DHKE is a very impressive application of the discrete logarithm problem
- ❑ This fundamental key agreement technique is implemented in many open and commercial cryptographic protocols like Secure Shell (SSH), Transport Layer Security (TLS), and Internet Protocol Security (IPSec).

Diffie–Hellman key exchange

- ❑ The basic idea behind the DHKE is that exponentiation in Z_p^* , p prime, is a one-way function and that exponentiation is commutative, i.e., $k = (\alpha^x)^y \equiv (\alpha^y)^x \pmod{p}$
- ❑ The value $k \equiv (\alpha^x)^y \equiv (\alpha^y)^x \pmod{p}$ is the joint secret which can be used as the session key between the two parties.

Diffie–Hellman key exchange

- ❑ Let us now consider how the Diffie–Hellman key exchange protocol over Z_p^* works.
- ❑ In this protocol we have two parties, Alice and Bob, who would like to establish a shared secret key.
- ❑ There is possibly a trusted third party that properly chooses the public parameters which are needed for the key exchange.
- ❑ However, it is also possible that Alice or Bob generate the public parameters.
- ❑ DHKE consists of two protocols, the set-up protocol and the main protocol, which performs the actual key exchange

Diffie–Hellman key exchange

The set-up protocol consists of the following steps:

Diffie–Hellman Set-up

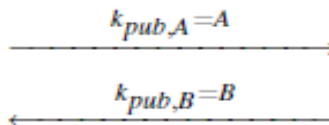
1. Choose a large prime p .
2. Choose an integer $\alpha \in \{2, 3, \dots, p-2\}$.
3. Publish p and α .

□ These two values are sometimes referred to as domain parameters. If Alice and Bob both know the public parameters p and α computed in the set-up phase, they can generate a joint secret key k with the following key-exchange protocol

Diffie–Hellman Key Exchange

Alice
 choose $a = k_{pr,A} \in \{2, \dots, p-2\}$
 compute $A = k_{pub,A} \equiv \alpha^a \pmod{p}$

Bob
 choose $b = k_{pr,B} \in \{2, \dots, p-2\}$
 compute $B = k_{pub,B} \equiv \alpha^b \pmod{p}$



$$k_{AB} = k_{pub,B}^{k_{pr,A}} \equiv B^a \pmod{p}$$

$$k_{AB} = k_{pub,A}^{k_{pr,B}} \equiv A^b \pmod{p}$$

Module 3

Diffie–Hellman key exchange Protocol

Here is the proof that this surprisingly simple protocol is correct, i.e., that Alice and Bob in fact compute the same session key k_{AB} .

Proof. Alice computes

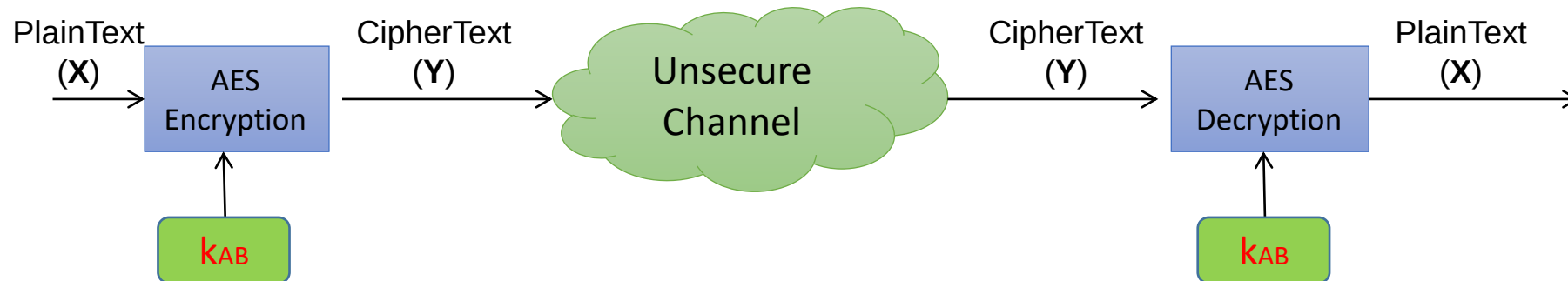
$$B^a \equiv (\alpha^b)^a \equiv \alpha^{ab} \pmod{p}$$

while Bob computes

$$A^b \equiv (\alpha^a)^b \equiv \alpha^{ab} \pmod{p}$$

and thus Alice and Bob both share the session key $k_{AB} \equiv \alpha^{ab} \pmod{p}$.

The key can now be used to establish a secure communication between Alice and Bob, e.g., by using k_{AB} as key for a symmetric algorithm like AES

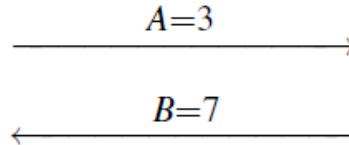


Diffie–Hellman key exchange Protocol: Example

Example: The Diffie–Hellman domain parameters are $p = 29$ and $\alpha = 2$. The protocol proceeds as follows:

Alice
choose $a = k_{pr,A} = 5$
 $A = k_{pub,A} = 2^5 \equiv 3 \pmod{29}$

Bob
choose $b = k_{pr,B} = 12$
 $B = k_{pub,B} = 2^{12} \equiv 7 \pmod{29}$



$k_{AB} = B^a \equiv 7^5 = 16 \pmod{29}$

$k_{AB} = A^b = 3^{12} \equiv 16 \pmod{29}$

As one can see, both parties compute the value **$k_{AB} = 16$** , which can be used as a joint secret, e.g., as a session key for symmetric encryption.

Diffie–Hellman key exchange Protocol

- ❑ The computational aspects of the DHKE are similar to those of RSA.
- ❑ During the set-up phase, we generate p using the probabilistic prime-finding algorithms (*such as: Fermat Primality Test, and Miller–Rabin Primality Test*)
- ❑ The prime number p should have a similar length as the RSA modulus n , i.e., 1024 or beyond, in order to provide strong security.

Diffie–Hellman key exchange Protocol

- ❑ The integer α needs to have a special property: It should be a primitive element, a topic which we discuss in the following slides.
- ❑ The session key k_{AB} that is being computed in the protocol has the same bit length as p .
- ❑ If we want to use k_{AB} as a symmetric key for algorithms such as AES, we can simply take the 128 most significant bits.
- ❑ Alternatively, a hash function is sometimes applied to k_{AB} and the output is then used as a symmetric key.

Diffie–Hellman key exchange Protocol

- ❑ During the actual protocol, we first have to choose the private keys a and b .
- ❑ They should stem from a true random generator (randomly) in order to prevent an attacker from guessing them.
- ❑ For computing the public keys A and B as well as for computing the session key, both parties can make use of the square-and-multiply algorithm.

Diffie–Hellman key exchange Protocol

- ❑ The public keys are typically pre-computed. The main computation that needs to be done for a key exchange is thus the exponentiation for the session key.
- ❑ In general, since the bit lengths and the computations of RSA and the DHKE are very similar, they require a similar effort.
- ❑ However, the trick of using short public exponents that was implemented in RSA cryptosystem is not applicable to the DHKE

Diffie–Hellman Key Exchange & Discrete logarithm problem

- ❑ What we showed so far is the classic Diffie–Hellman key exchange protocol in the group $\mathbb{Z}^*_p$, where p is a prime.
- ❑ The protocol can be generalized, in particular to groups of elliptic curves. This gives rise to elliptic curve cryptography, which has become a very popular asymmetric scheme in practice.
- ❑ In order to better understand elliptic curves and schemes such as Elgamal encryption, which are also closely related to the DHKE, we introduce the discrete logarithm problem (DLP) in the following slides. This problem is the mathematical basis for the DHKE.
- ❑ The DLP can be defined in the concept of cyclic group. Therefore, it is very important to introduce such mathematical principals.

Groups Definition

❖ Formal Definition of Groups

A group is a set of elements G together with an operation $\circ$ which combines two elements of G . A group has the following properties.

- 1. The group operation $\circ$ is closed. That is, for all $a, b, \in G$, it holds that $a \circ b = c \in G$.*
- 2. The group operation is associative. That is, $a \circ (b \circ c) = (a \circ b) \circ c$ for all $a, b, c \in G$.*
- 3. There is an element $1 \in G$, called the neutral element (or identity element), such that $a \circ 1 = 1 \circ a = a$ for all $a \in G$.*
- 4. For each $a \in G$ there exists an element $a^{-1} \in G$, called the inverse of a , such that $a \circ a^{-1} = a^{-1} \circ a = 1$.*
- 5. A group G is abelian (or commutative) if, furthermore, $a \circ b = b \circ a$ for all $a, b \in G$.*

□ Note that in cryptography we use both multiplicative groups, i.e., the operation “ $\circ$ ” denotes multiplication, and additive groups where “ $+$ ” denotes addition. The latter notation is used for elliptic curves

Groups Definition

❑ Example of a group:

❑ $(\mathbb{Z}, +)$ is a group, i.e., the set of integers $\mathbb{Z} = \{ \dots, -2, -1, 0, 1, 2, \dots \}$ together with the usual addition forms an abelian group, where $e = 0$ is the identity element and $-a$ is the inverse of an element $a \in \mathbb{Z}$.

❑ However, all of these groups do not play a significant role in cryptography because we need groups with a finite number of elements.

❑ Let us now consider the group $\mathbb{Z}_{*n}$ which is very important for many cryptographic schemes such as DHKE, Elgamal encryption, digital signature algorithm and many others

❑ Theorem:

The set $\mathbb{Z}_{*n}$ which consists of all integers $i=0, 1, \dots, n-1$ for which $\gcd(i, n) = 1$ forms an abelian group under multiplication modulo n . The identity element is $e = 1$.

Module 3

Finite Groups Definition

- ❑ In cryptography, a finite group is needed.
- ❑ Example: Is Z_{*9} a group ??
- ❑ $Z_{*9} = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$
- ❑ $\circ$ = multiplication operation mod 9
- ❑ We will check now whether the previous properties can be satisfied with this group:
- ❑ We will exclude all elements that do not have a multiplicative inverse.
- ❑ Element zero does not have inverse, as well as elements 3 and 6.
- ❑ Therefore, $Z_{*9} = \{1, 2, 4, 5, 7, 8\}$, which is a multiplicative group mod 9
- ❑ Closeness property is fulfilled since all results will be in the group, such that:

Multiplication table for Z_{*9}

$\times \text{ mod } 9$	1	2	4	5	7	8
1	1	2	4	5	7	8
2	2	4	8	1	5	7
4	4	8	7	2	1	5
5	5	1	2	7	8	4
7	7	5	1	8	4	2
8	8	7	5	4	2	1

Finite Groups Definition

- ❑ In DLP cryptography, we usually use Z_{*p} , where p means a (very large) prime number.
- ❑ Such group forms a multiplicative group mod p , as follows:
- ❑ For every element in the group: $\gcd(a, p) = 1$.
- ❑ This condition applies for all elements, because p is prime number, except the element zero.
- ❑ $Z_{*p} = \{1, 2, 3, \dots, p-1\}$
- ❑ This means that for every element there is a multiplicative inverse.
- ❑ The inverse can be computed using the Extended Euclidean Algorithm

Finite Groups Definition

□ Formal Definition of Finite Groups:

A group $(G, \circ)$ is finite if it has a finite number of elements. We denote the cardinality or order of the group G by $|G|$.

Note: The cardinality means the number of elements in the group

Example: $|Z_{*9}| = 6 \rightarrow \{1, 2, 4, 5, 7, 8\}$

Cyclic Groups

□ We will use an example first to introduce essential terminologies:

□ **Example:** $\mathbb{Z}_{*11} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

□ What happens if we compute all powers of an element $a = 3$ (for example)

□ $a^1 = 3 \pmod{11}$

□ $a^2 = 9 \pmod{11}$

□ $a^3 = 27 \equiv 5 \pmod{11}$

□ $a^4 = 5 \cdot 3 \equiv 4 \pmod{11}$

□ $a^5 = 4 \cdot 3 \equiv 1 \pmod{11}$

□ $a^6 = 1 \cdot 3 = 3 \pmod{11}$

□ $a^7 = 3 \cdot 3 = 9 \pmod{11}$

□ **Question:** What is the result of $3^{324298} \pmod{11}$??

□ **Answer:** There are only five possible results.

□ The exponentiation operation is the major operation in DHKE and other cryptosystems such as RSA.

□ This property of an element 3 is called the $\text{order}(3) \equiv 5$ (cycle length)

❑ **Example:** $\mathbb{Z}_{*11} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

❑ What happens if we compute all powers of an element $a = 2$ (for example)

❑ $a^1 = 2 \pmod{11}$

❑ $a^2 = 4 \pmod{11}$

❑ $a^3 = 8 \pmod{11}$

❑ $a^4 = 16 \equiv 5 \pmod{11}$

❑ $a^5 = 5 \cdot 2 \equiv 10 \pmod{11}$

❑ $a^6 = 10 \cdot 2 = 9 \pmod{11}$

❑ $a^7 = 9 \cdot 2 = 7 \pmod{11}$

❑ $a^8 = 7 \cdot 2 = 3 \pmod{11}$

❑ $a^9 = 3 \cdot 2 = 6 \pmod{11}$

❑ $a^{10} = 6 \cdot 2 = 1 \pmod{11}$

❑ $a^{11} = 1 \cdot 2 = 2 \pmod{11}$

Cyclic Groups

□ The formal definition of the order of an element **a** within a group :

The order $\text{ord}(a)$ of an element a of a group $(G, \circ)$ is the smallest positive integer k such that

$$a^k = \underbrace{a \circ a \circ \dots \circ a}_{k \text{ times}} = 1,$$

where 1 is the identity element of G .

K: The smallest positive integer needed for exponentiation to obtain the result = 1, which is the identity element of the group.

For example:

$$\square a^{10} \equiv 1 \pmod{11}$$

$$\square a^{20} \equiv 1 \pmod{11}$$

$$\square a^{30} \equiv 1 \pmod{11}$$

$$\square a^{\dots} \equiv 1 \pmod{11}$$

We need the smallest positive integer

❑ The formal definition of the cyclic group :

❑ A group G which contains an element α with maximum order $\text{ord}(\alpha) = |G|$ is said to be cyclic. Elements with maximum order are called **primitive elements** or **generators**.

❑ An element α of a group G with maximum order is called a generator since every element a of G can be written as a power $\alpha^i = a$ of this element for some i

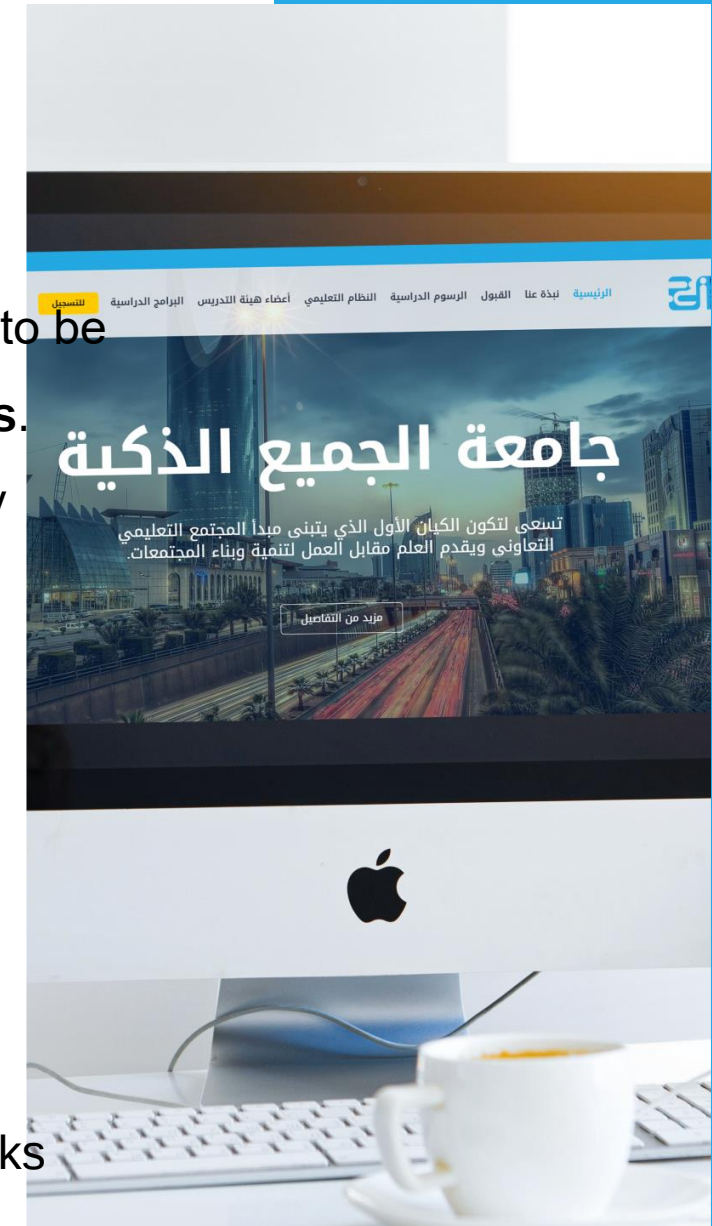
❑ In other words: α generates the entire group.

❑ Therefore: $\text{ord}(a=2) = 10 = |Z_{*11}|$

❑ Again, notice all the elements that are generated by powers of two:

i	1	2	3	4	5	6	7	8	9	10
a^i	2	4	8	5	10	9	7	3	6	1

❑ By looking at the bottom row, we see that that the powers 2^i in fact generates all elements of the group Z_{*11} . We note that the order in which they are generated looks quite arbitrary. This seemingly random relationship between the exponent i and the group elements is the basis for the Diffie–Hellman key exchange protocol



Module 03

Cyclic Groups

❑ From previous example we see that the group $\mathbb{Z}_{*11}$ has the element 2 as a generator.

❑ It is important to stress that the number 2 is not necessarily a generator in other cyclic groups $\mathbb{Z}_{*n}$.

❑ For instance, in $\mathbb{Z}_{*7}$, $\text{ord}(2) = 3$, and the element 2 is thus not a generator in that group.

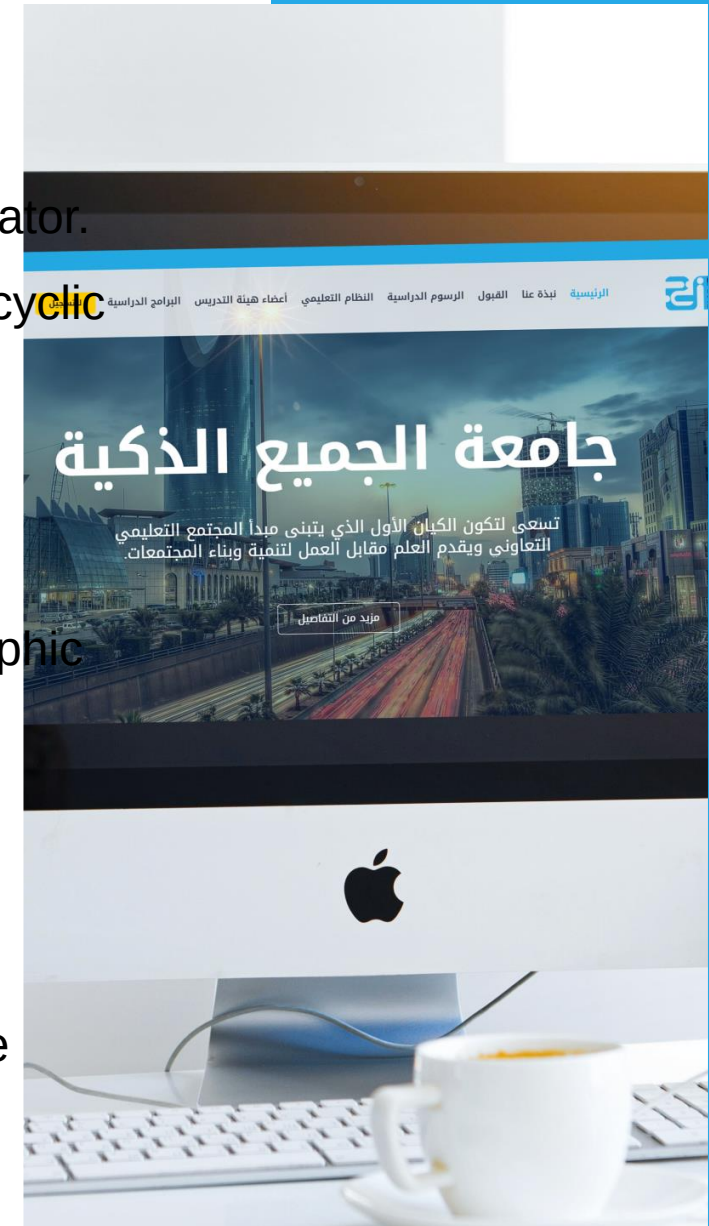
❑ Cyclic groups have interesting properties. The most important ones for cryptographic applications are given in the following theorems:

❑ **Theorem:** For every prime p , $(\mathbb{Z}_{*p})$ is an abelian finite cyclic group.

❑ This theorem states that the multiplicative group of every prime field is cyclic.

❑ This has far reaching consequences in cryptography, where these groups are the most popular ones for building discrete logarithm cryptosystems.

❑ In order to underline the practical relevance of these somewhat esoteric looking theorem, consider that almost every Web browser has a cryptosystem over $\mathbb{Z}_{*p}$ built in



Module 03

Cyclic Groups

Example: We consider again the group $\mathbb{Z}_{11}^*$, which has a cardinality of $|\mathbb{Z}_{11}^*| = 10$.

The only element orders in this group are 1, 2, 5, and 10, since these are the only integers that divide 10.

We verify this property by looking at the order of all elements in the group:

$$\text{ord}(1) = 1$$

$$\text{ord}(2) = 10$$

$$\text{ord}(3) = 5$$

$$\text{ord}(4) = 5$$

$$\text{ord}(5) = 5$$

$$\text{ord}(6) = 10$$

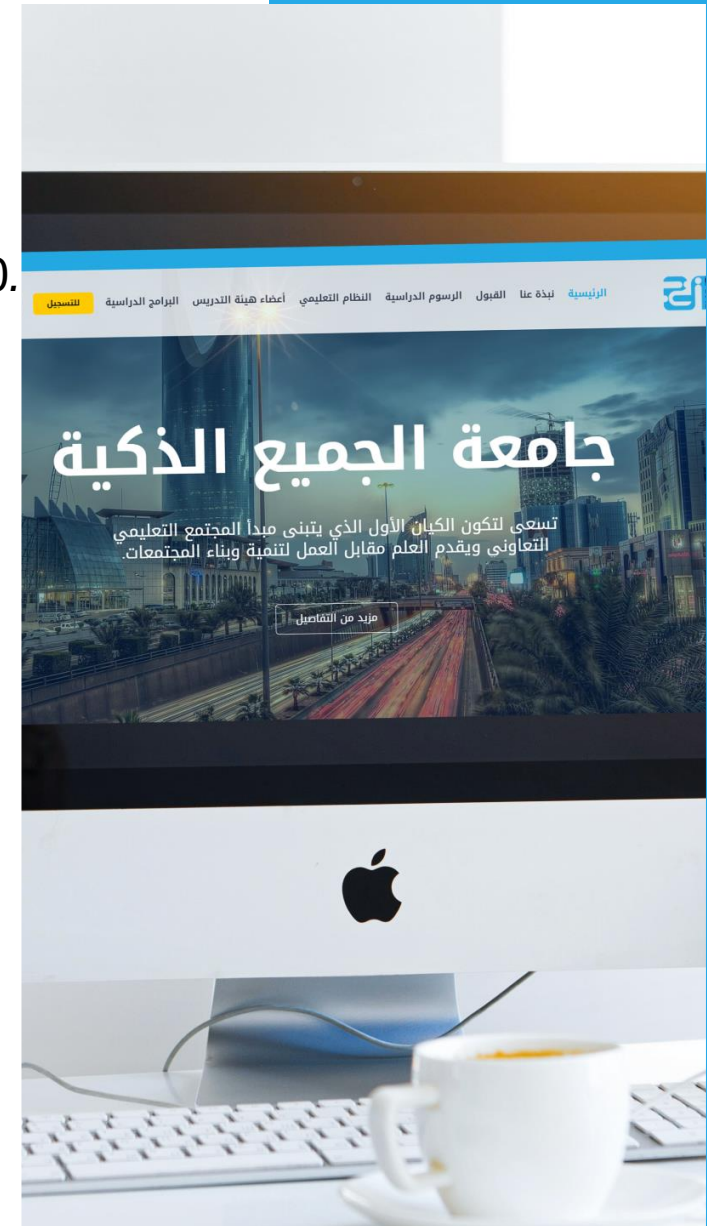
$$\text{ord}(7) = 10$$

$$\text{ord}(8) = 10$$

$$\text{ord}(9) = 5$$

$$\text{ord}(10) = 2$$

Indeed, only orders that divide 10 occur.



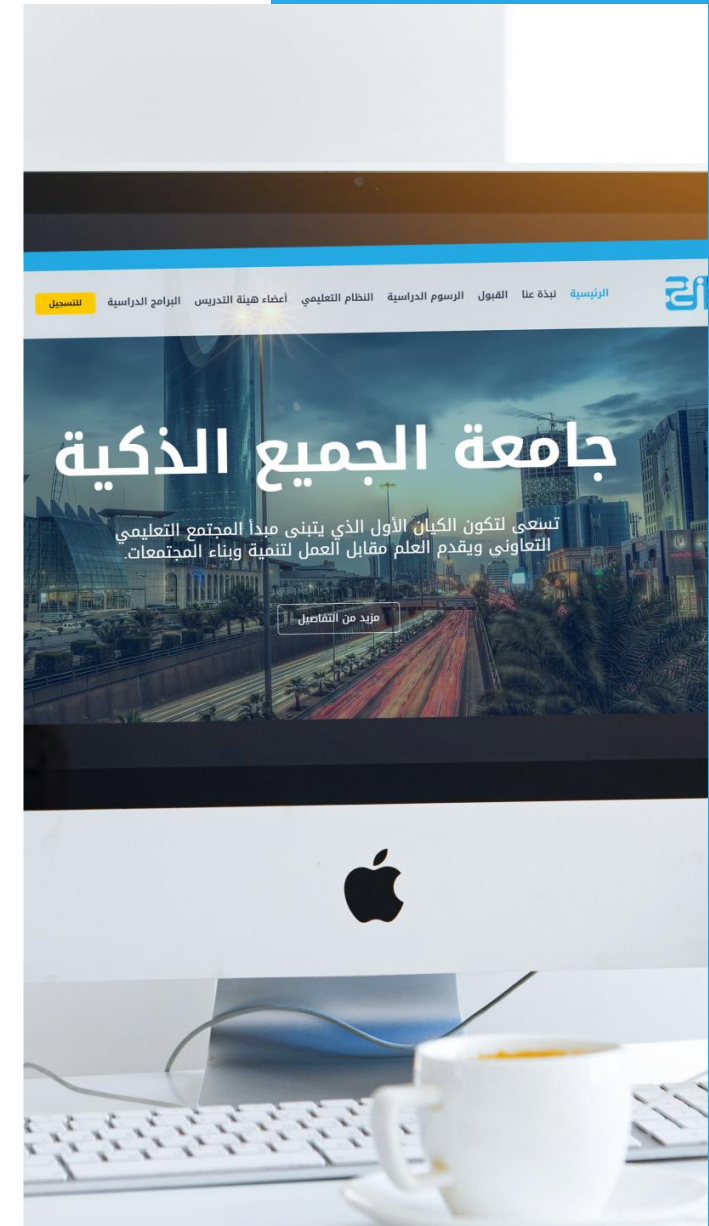
Theorem: Let G be a finite cyclic group. Then it holds that:

1. The number of primitive elements of G is $\Phi(|G|)$.
2. If $|G|$ is prime, then all elements $a \neq 1 \in G$ are primitive.

❑ The first property can be verified by the example above. Since $\Phi(10) = (5-1)(2-1) = 4$, the number of primitive elements is four, which are the elements 2, 6, 7 and 8.

❑ The second property follows from the previous theorem. If the group cardinality is prime, the only possible element orders are 1 and the cardinality itself.

❑ Since only the element 1 can have an order of one, all other elements have order p .



Module 03

Discrete Logarithm Problem

❑ After the somewhat lengthy introduction to cyclic groups one might wonder how they are related to the DHKE protocol.

❑ It turns out that the underlying one-way function of the DHKE, the discrete logarithm problem (DLP), can directly be realized using cyclic groups.

❑ The Discrete Logarithm Problem in Prime Fields

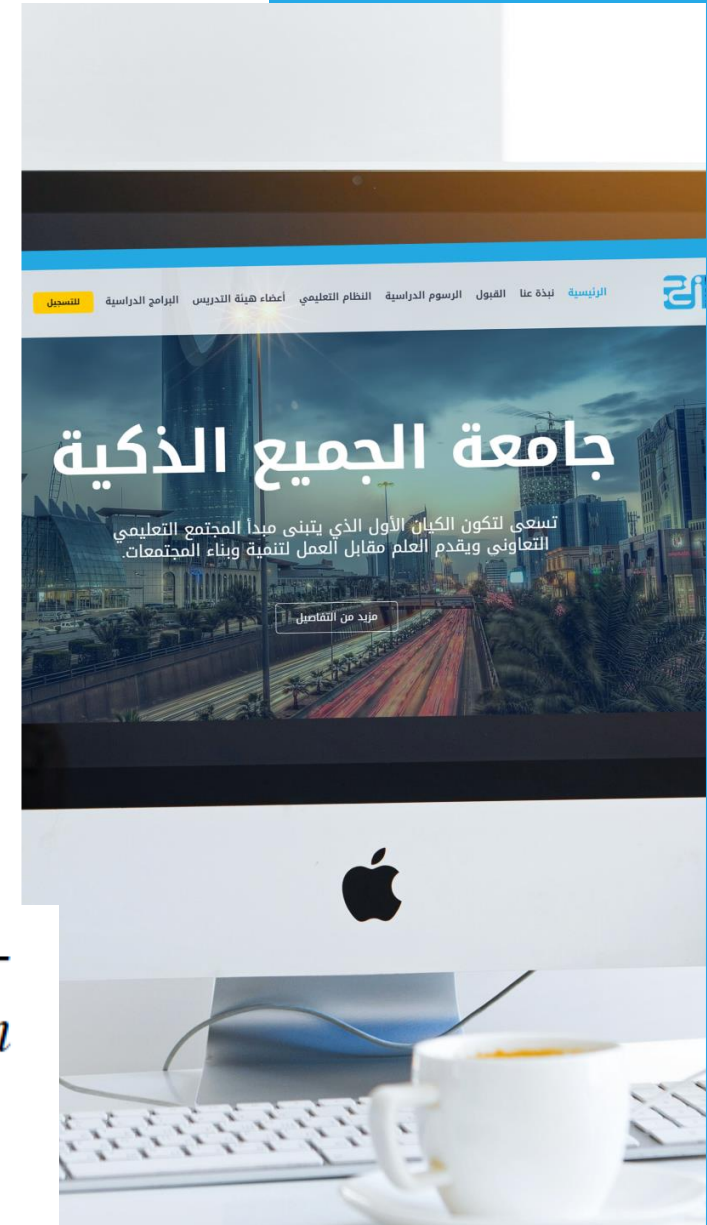
We start with the DLP over $\mathbb{Z}_p^*$, where p is a prime

Theorem:

Discrete Logarithm Problem (DLP) in $\mathbb{Z}_p^*$:

Given is the finite cyclic group $\mathbb{Z}_p^$ of order $p - 1$ and a primitive element $\alpha \in \mathbb{Z}_p^*$ and another element $\beta \in \mathbb{Z}_p^*$. The DLP is the problem of determining the integer $1 \leq x \leq p - 1$ such that:*

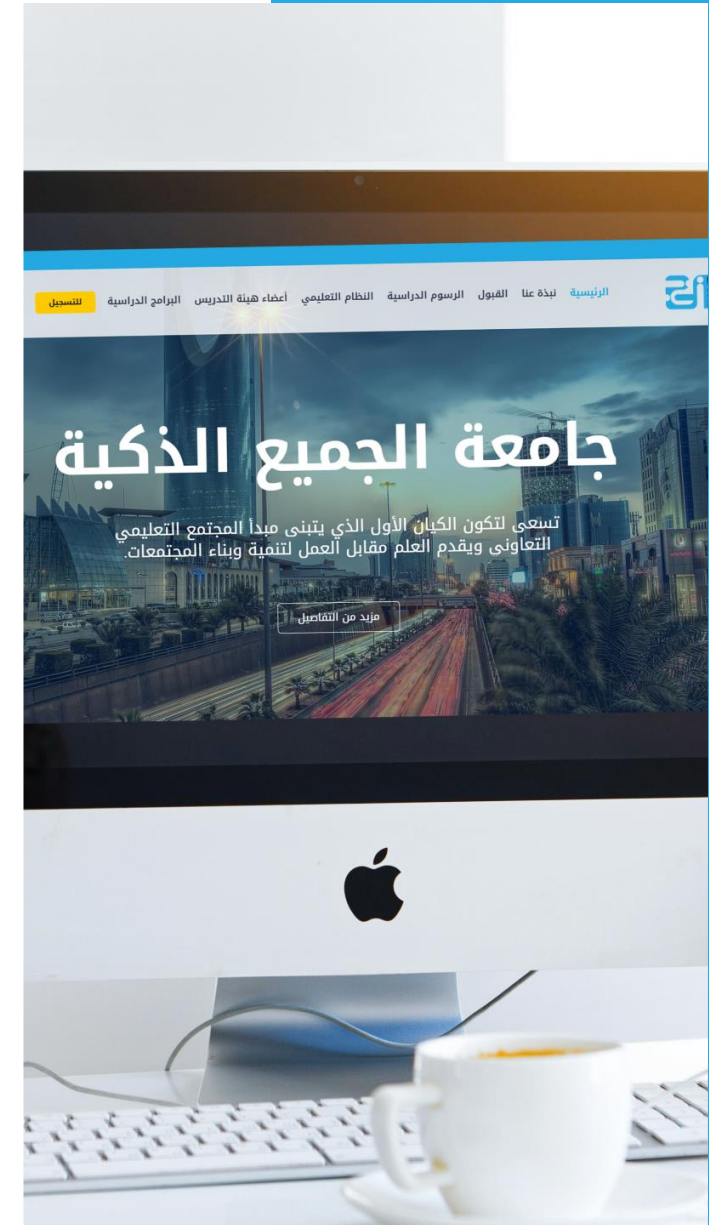
$$\alpha^x \equiv \beta \pmod{p}$$



Module 03

Discrete Logarithm Problem

- ❑ Recall that such an integer x must exist since α is a primitive element and each group element can be expressed as a power of any primitive element.
- ❑ This integer x is called the discrete logarithm of β to the base α , and we can formally write: $x = \log_{\alpha} \beta \bmod p$.
- ❑ Computing discrete logarithms modulo a prime is a very hard problem if the parameters are sufficiently large.
- ❑ Since exponentiation $\alpha^x \equiv \beta \bmod p$ is computationally easy, this forms a one-way function

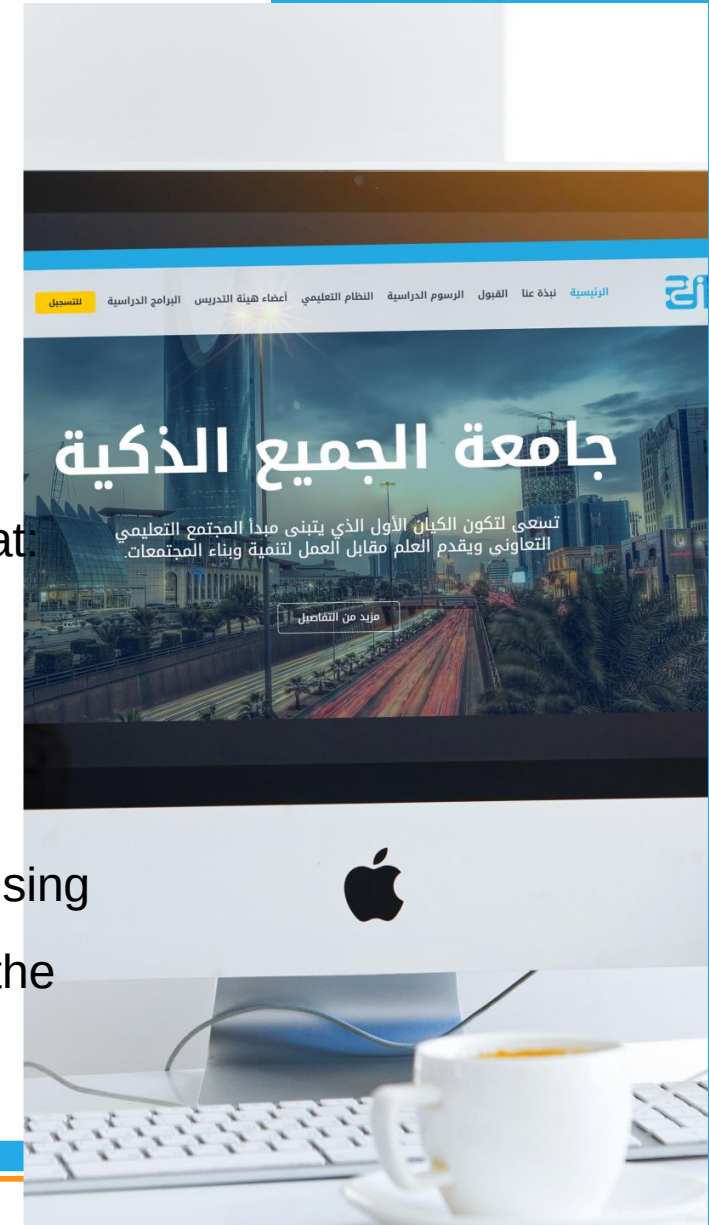


Module 03

Discrete Logarithm Problem: Example

Example:

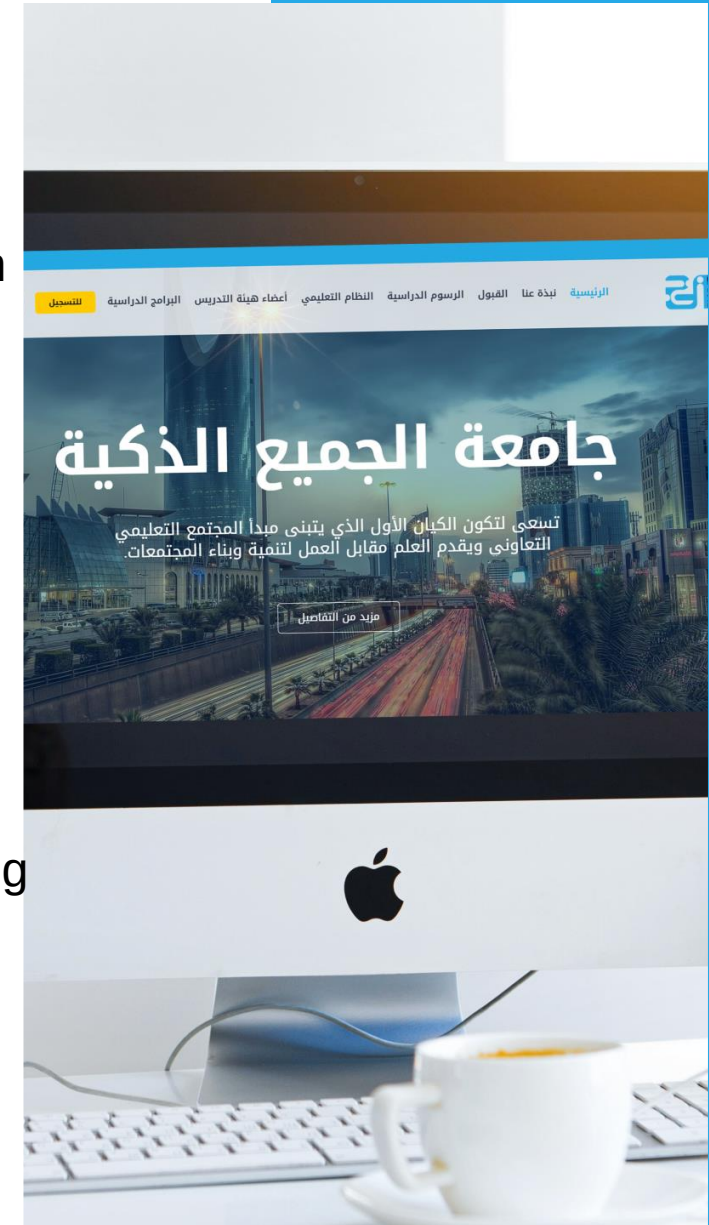
- ❑ We consider a discrete logarithm in the group Z_{47}^* , in which $\alpha = 5$ is a primitive element.
- ❑ For $\beta = 41$ the discrete logarithm problem is: Find the positive integer x such that:
 $5^x \equiv 41 \pmod{47}$
- ❑ Therefore, to compute x , we need to compute the logarithm of β , such that:
 $x = \log_5 41 \pmod{47}$
- ❑ Even for such small numbers, determining x is not entirely straightforward. By using a brute-force attack, i.e., systematically trying all possible values for x , we obtain the solution $x = 15$.



Module 03

Security of Discrete Logarithm

- ❑ We consider the security of DHKE against passive attacks, where the attacker can only listen (passive attacker), and he can not manipulate the message.
- ❑ The goal of such attacker is to compute the session key k_{AB} shared by Alice and Bob
- ❑ Which information does Oscar get from observing the DHKE protocol?
- ❑ Certainly, Oscar knows α and p because these are public parameters chosen during the set-up protocol.
- ❑ In addition, Oscar can obtain the values $A = k_{pub,A}$ and $B = k_{pub,B}$ by eavesdropping on the channel during an execution of the key-exchange protocol.
- ❑ **Question:** Can Oscar compute $k = \alpha^{ab}$ from α , p , $A \equiv \alpha^a \pmod{p}$ and $B \equiv \alpha^b \pmod{p}$.
- ❑ This problem is called the Diffie–Hellman problem (DHP). Like the discrete logarithm problem it can be generalized to arbitrary finite cyclic groups.



Module 03

Security of Discrete Logarithm

❑ The Definition of the Generalized Diffie–Hellman Problem (DHP)

Given is a finite cyclic group G of order n , a primitive element $\alpha \in G$ and two elements $A = \alpha^a$ and $B = \alpha^b$ in G . The Diffie–Hellman problem is to find the group element α^{ab}

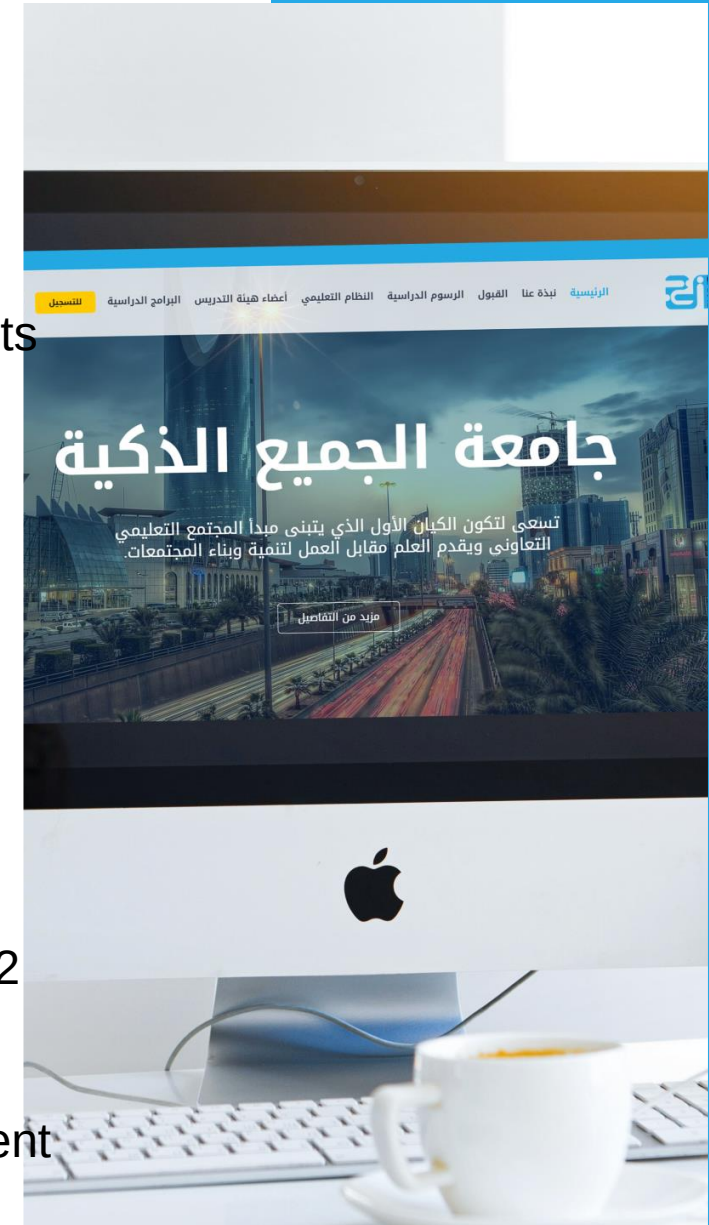
One way of solving the DHP is to do the following:

Step1: Compute a (or b), which requires computing the logarithm: $a = \log_{\alpha} A \bmod p$

Step2: Compute $B^a = K_{AB} \bmod p$

❑ But the first step is computationally very hard if p is large enough (1024 bits – 3072 bits)

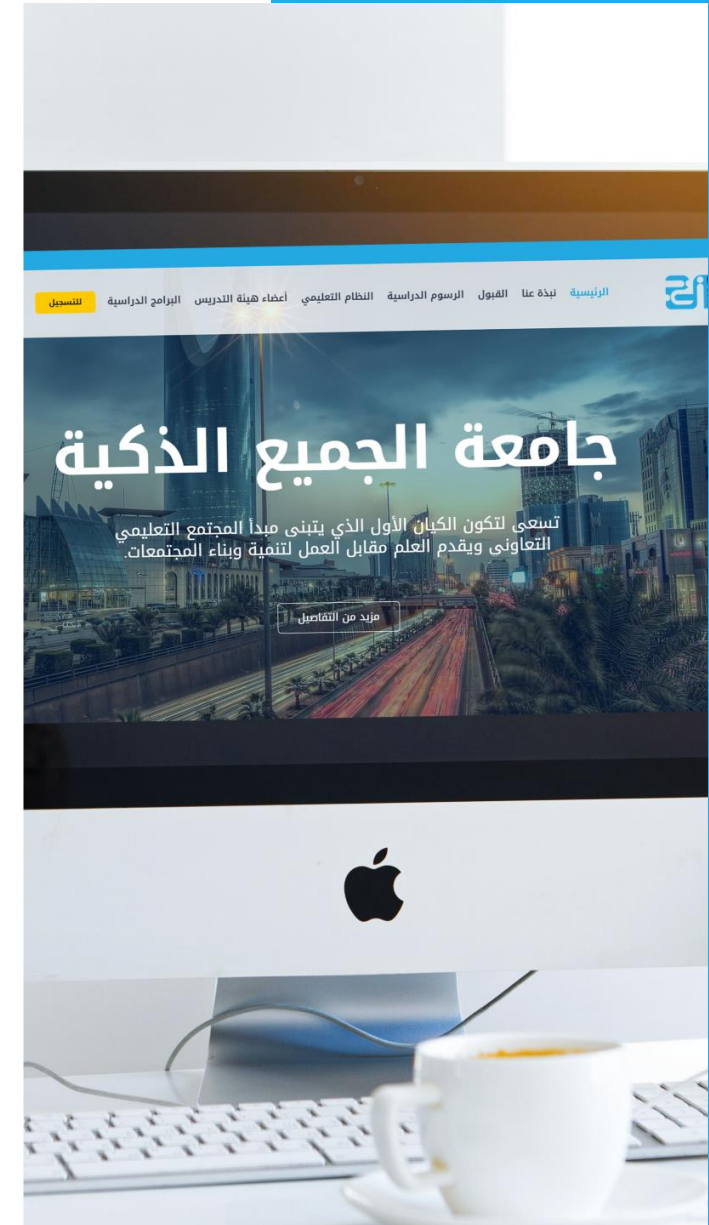
❑ If the only way for solving DHP requires solving the DLP, then, the DHP is equivalent to the DLP (they are the same problem), similar situation can be found in RSA



Module 03

Man-in-the-Middle Attack

- Mallory prepares by creating two private / public keys
- Alice transmits her public key to Bob
- Mallory intercepts this and transmits his first public key to Bob. Mallory also calculates a shared key with Alice
- Bob receives the public key and calculates the shared key (with Mallory instead of Alice)
- Bob transmits his public key to Alice
- Mallory intercepts this and transmits his second public key to Alice. Mallory calculates a shared key with Bob
- Alice receives the key and calculates the shared key (with Mallory instead of Bob)
- Mallory can then intercept, decrypt, re-encrypt, forward all messages between Alice & Bob



Module 03

Man-in-the-Middle Attack

