School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : **ECDSA Workshop _ Digital Signatures Demo**

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM:

1. Open the website _ learnmeabitcoin.com/technical/cryptography/elliptic-curve/ecdsa
2. Read the concept of ECDSA _ Understand how elliptic curves and digital signatures work.
3. Understand Private & Public Key generation _ How a public key is derived from a private key.
4. Learn the Digital Signature process _ How signing and verification of transactions works.
5. Note down key points _ Full form of ECDSA, its role in Bitcoin, and the verification steps.
6. Write a summary _ Create a short explanation in your own words.

* Softwares used

1. Brave Web Browser
2. Text Editor

*As applicable according to the experiment.

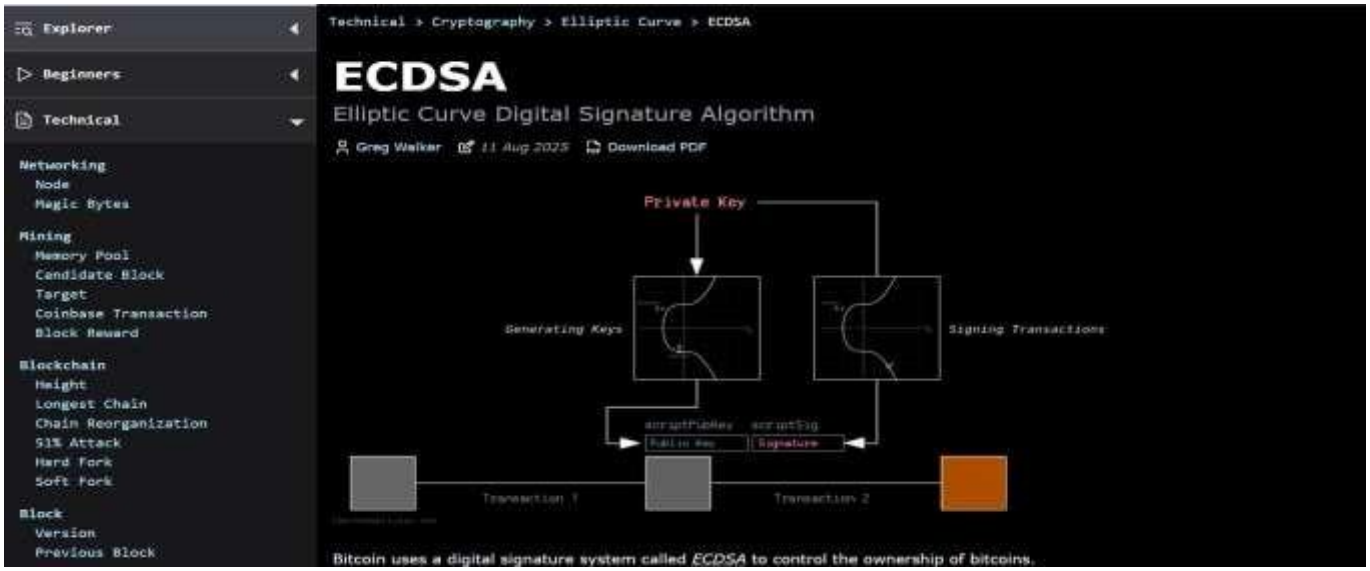
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

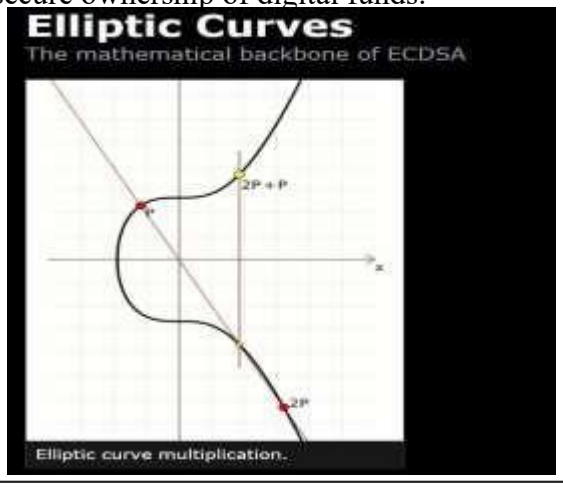
What is ECDSA:

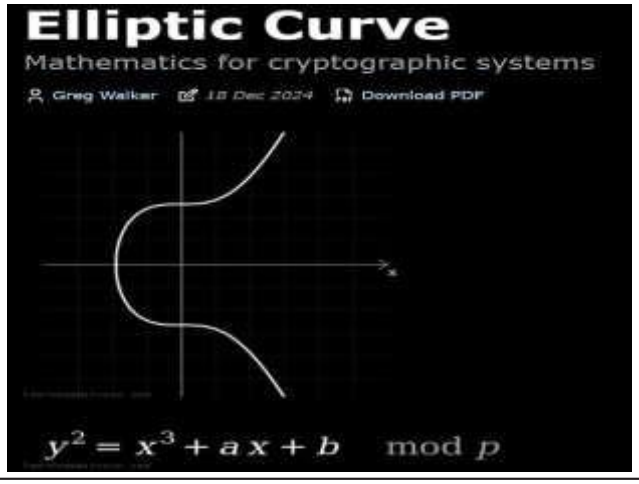
ECDSA (Elliptic Curve Digital Signature Algorithm) is a cryptographic algorithm used for creating and verifying digital signatures based on elliptic curve mathematics. It provides a secure way to prove ownership without revealing the private key. In Bitcoin and other cryptocurrencies, ECDSA is essential for verifying transactions: a user generates a private key (a secret number) and derives a public key from it using elliptic curve multiplication. When a transaction is signed with the private key, the network uses the corresponding public key to verify the signature's validity, ensuring that only the rightful owner can authorize spending of funds.

Visit this website to explore how can we generate public and private key and our digital signature:-
https://learnmeabitcoin.com/technical/cryptography/elliptic-curve/ecdsa/



Bitcoin uses a digital signature system called ECDSA (Elliptic Curve Digital Signature Algorithm) to manage the ownership of bitcoins. This system works by allowing users to create a pair of keys: a private key, which must be kept secret, and a public key, which can be shared openly. The private key is used to generate digital signatures that prove ownership of the corresponding public key without revealing the private key itself. This mechanism ensures that users can securely receive and send bitcoins through transactions. Anyone can generate their own key pair, and others can send bitcoins to them by associating the funds with their public key. However, only the person who holds the matching private key can create valid signatures to unlock and spend those bitcoins. This prevents unauthorized access and guarantees secure ownership of digital funds.

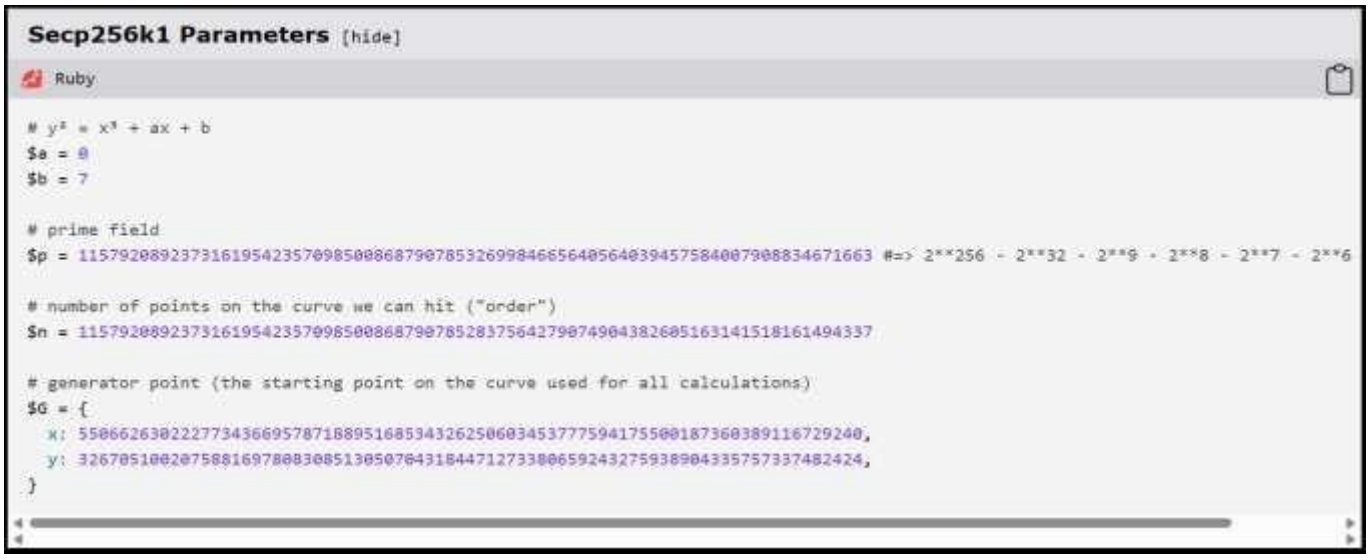




*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

ECDSA uses elliptic curve cryptography as the foundation for its digital signature system. In simple terms, public keys and signatures are points on an elliptic curve. If both the public key and the signature are generated from the same private key (a large secret number), there exists a geometric relationship between these points that proves ownership. The main mathematical operation involved is elliptic curve multiplication, which means repeatedly adding a point on the curve to itself a certain number of times to reach a new point. This operation is easy to perform in one direction but practically impossible to reverse, making it highly secure. Because of this property, elliptic curves are widely used in cryptography for generating keys and digital signatures.



Key Generation:

*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

In this step, I used the Random button in the EC Multiply tool to generate a random multiplier, which represents a private key. After clicking Generator Point and then Random, the tool displayed a random number as the multiplier and calculated the corresponding point (x, y) on the elliptic curve. This new point acts as the public key. The private key (multiplier) is a large random number, and the public key is derived from it using elliptic curve multiplication. This process is secure because it is practically impossible to reverse the operation to find the private key from the public key. **Sign:**

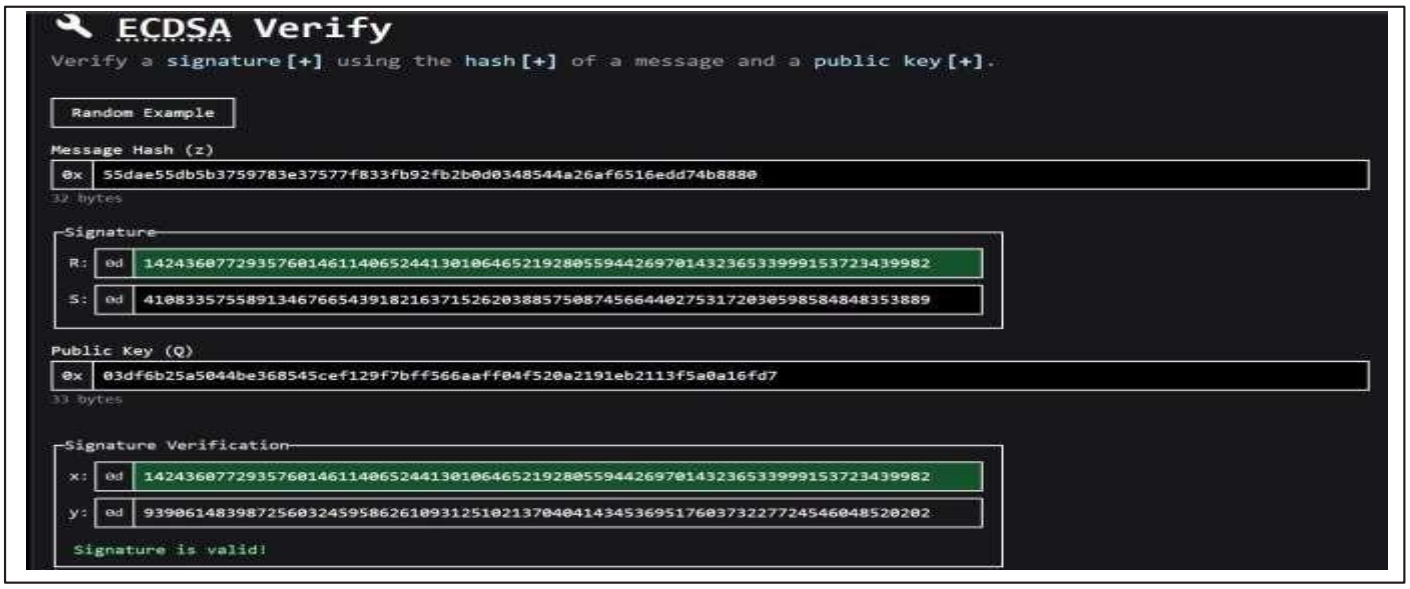
In this step, I performed the ECDSA signing process using the online tool. The signing process requires three main inputs: a message hash (z), a private key (d), and a random nonce (k). After entering these values and clicking the Sign button, the tool generated two numbers, r and s, which together form the digital signature. The value of r is calculated from an elliptic curve point derived using the nonce, while s is computed using the message hash, private key, and r. This signature proves ownership of the private key without exposing it and can later be verified with the corresponding public key.

Verify:

In the verification step, I explored how the network or a verifier checks whether a digital signature is valid without needing access to the private key. The process uses three main elements: the original message hash (z), the public key, and the signature values (r and s). When these are provided to the verification tool, it performs several elliptic curve calculations to confirm if the signature corresponds to the given public key. The core idea is that the verifier calculates a specific point on the curve using r, s, and the generator point and then checks if it matches the relationship established during signing. If the calculated point matches the expected value, the signature is considered valid, proving that the signer owns the private key linked to the public key. This step ensures authenticity and integrity in Bitcoin transactions because only the correct private key holder can produce a valid signature, but anyone can verify it using the public key.

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)



* Observations

1.ECDSA uses elliptic curve cryptography to generate a public key from a private key using a one-way multiplication function, making it secure.
2.The signing process creates a digital signature (r, s) that proves ownership without revealing the private key.
3.Verification only requires the public key, message hash, and signature, ensuring the private key remains confidential.
4.Reusing the same nonce during signing can compromise security by exposing the private key, so randomness is critical.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Signature of the Faculty:

Name :

Regn. No. :

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.