Marathwada Mitra Mandal's

COLLEGE OF ENGINEERING, PUNE

Accredited with 'A++' Grade by NAAC

Department of Computer Engineering

Lab Manual

410246 : LP-III: Blockchain Technology

Prepared by,

Prof. Mane Aishwarya

Prof. Reshma Kapadi

BE COMP (2019 Pattern)

Academic Year 2024-25 Sem I

Preface

We may be at the dawn of a new revolution. This revolution started with a new fringe economy on the Internet, an alternative currency called Bitcoin that was issued and backed not by a central authority, but by automated consensus among networked users. Its true uniqueness, however, lay in the fact that it did not require the users to trust each other. Through algorithmic self-policing, any malicious attempt to defraud the system would be rejected. In a precise and technical definition, Bitcoin is digital cash that is transacted via the Internet in a decentralized trustless system using a public ledger called the blockchain. It is a new form of money that combines BitTorrent peer-to-peer file sharing with public key cryptography.

Since its launch in 2009, Bitcoin has spawned a group of imitators alternative currencies using the same general approach but with different optimizations and tweaks. More importantly, blockchain technology could become the seamless embedded economic layer the Web has never had, serving as the technological underlay for payments, decentralized exchange, token earning and spending, digital asset invocation and transfer, and smart contract issuance and execution.

Bitcoin and blockchain technology, as a mode of decentralization, could be the next major disruptive technology and worldwide computing paradigm (following the mainframe, PC, Internet, and social networking/mobile phones), with the potential for reconfiguring all human activity as pervasively as did the Web.

There is a worldwide awareness of realizing the importance of software testing. software testing in the context of testing concepts and methods that can be implemented in practice. The main features of this subject is

- It focuses on the importance, significance and limitations of blockchain technology.
- It presents techniques that may help to design digital wallets with the help of a program.
- It is an immutable public digital ledger, which means when a transaction is recorded, it cannot be modified.
- Due to the encryption feature, Blockchain is always secure
- The transactions are done instantly and transparently, as the ledger is updated automatically.
- As it is a decentralized system, no intermediary fee is required.
- The authenticity of a transaction is verified and confirmed by participants.


'येथे बहुतांचे हित ।'
Marathwada Mitra Mandal's
COLLEGE OF ENGINEERING
Karvenagar, Pune
Permanently affiliated to SPPU | Accredited with 'A' Grade by NAAC
Recipient of 'Best College' award in 2018-19 by SPPU
Vision
To aspire for the Welfare of Society
through excellence in Science and Technology.
Mission
Our Mission is to
Mould young talent for higher endeavours.
Meet the challenges of globalization.
Commit for social progress with values and ethics.
Orient faculty and students for research and development.
Emphasize excellence in all disciplines.

'येथे बहुतांचे हित ।'

Marathwada Mitramandal's
COLLEGE OF ENGINEERING

Karvenagar, Pune - 411052

Department of Computer Engineering

Vision

To contribute to welfare of society by empowering students with latest skills, tools and technologies in the field of Computer Engineering through excellence in education and research

Mission

- To provide excellent academic environment for continuous improvement in the domain knowledge of Computer Engineering to solve real world problems
- To impart value-based education to students, with innovative and research skills to make them responsible engineering professionals for societal upliftment
- To strengthen links with industries through partnerships and collaborative developmental works

Program Educational Objectives (PEOs)

- To develop globally competent graduates with strong fundamental knowledge and analytical capability in latest technological trends
- To prepare the graduates as ethical and committed professionals with a sense of societal and environmental responsibilities
- To inculcate research attitude in multidisciplinary domains with experiential learning and developing entrepreneurship skills
- To groom graduates by incorporating investigative approach among them to effectively deal with global challenges

Program Specific Outcomes (PSOs)

A graduate of the Computer Engineering Program will be able to

- Analyze the problems and design solutions in the areas of Artificial Intelligence & High Performance Computing
- Develop advanced digital solutions using standard software engineering practices

Program Outcomes (POs)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. Problem Analysis: Identify, formulates, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design / development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research – based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Rubrics for Lab Assessment (CAS)

Dimension	Scale				
	1	2	3	4	5
Regularity and punctuality	Did not Perform, submitted in time	Performed and submitted later than scheduled date	Performed on schedule; submitted two weeks late	Performed on schedule; submitted one week late	Performed and submitted as per schedule
Understanding and preparation for Objective	Neither shows any understanding of the objective nor can relate it to theory.	States the objective very vaguely	Can only state the objective but shows poor understanding	Understands objective but cannot place it in context of a theory	Understands objective and can relate it to an appropriate theory topic
Participation in performance and conduction of experiment	Does not participate in experiment	Performs the experiment only with the help from supervisor/others and is confused and untidy.	Performs the experiment with some supervisory help; but forgets some crucial reading and is confused and untidy.	Performs experiment on own without supervisory help; records all readings properly but untidy.	Performs experiment on his/her own without supervisory help; records all readings properly. Keeps the setup clean and tidy.
Post experiment skills	Cannot follow the procedure and do any work	Follows procedure half-heartedly	Follows right procedure; but cannot analyze data and interpret it	Follows right procedure and can analyze data and interpret it	Follows right procedure; can analyze data and interpret it with justification

Syllabus

Savitribai Phule Pune University

Fourth Year of Computer Engineering (2019 Course)

410246: Laboratory Practice III

Teaching Scheme:	Credit	Examination Scheme:
Practical: 04 Hours/Week	02	Term work: 50 Marks Practical: 50 Marks

Companion Course: Design and Analysis of Algorithms (410241), Machine Learning(410242), Blockchain Technology(410243)

Course Objectives:

- Learn effect of data preprocessing on the performance of machine learning algorithms
- Develop in depth understanding for implementation of the regression models.
- Implement and evaluate supervised and unsupervised machine learning algorithms.
- Analyze performance of an algorithm.
- Learn how to implement algorithms that follow algorithm design strategies namely divide and conquer, greedy, dynamic programming, backtracking, branch and bound.
- Understand and explore the working of Blockchain technology and its applications.

Course Outcomes:

After completion of the course, students will be able to

CO1: Apply preprocessing techniques on datasets.

CO2: Implement and evaluate linear regression and random forest regression models.

CO3: Apply and evaluate classification and clustering techniques.

CO4: Analyze performance of an algorithm.

CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch and bound.

CO6: Interpret the basic concepts in Blockchain technology and its applications

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as a softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to a journal must be avoided. Use of DVD containing student programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Assessment of each Laboratory assignment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, punctuality, documentation and neatness.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Instructors may also set one assignment or mini-project that is suitable to each branch beyond the scope of the syllabus.
Operating System recommended :- 64-bit Open source Linux or its derivative
Programming tools recommended: - C++, Java, Python, Solidity, etc.

Virtual Laboratory:

- http://cse01-iiith.vlabs.ac.in/
- http://vlabs.iitb.ac.in/vlabs-dev/labs/blockchain/labs/index.php
- http://vlabs.iitb.ac.in/vlabs-dev/labs/machine_learning/labs/index.php

Group C: Blockchain Technology	
Any 4 assignments and a Mini project are mandatory.	
1.	Installation of Metamask and study spending Ether per transaction.
2.	Create your own wallet using Metamask for crypto transactions.
3.	Write a smart contract on a test network, for Bank account of a customer for following operations: • Deposit money • Withdraw Money • Show balance
4.	Write a program in solidity to create Student data. Use the following constructs: • Structures • Arrays • Fallback Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.
5.	Write a survey report on types of Blockchains and its real time use cases.
6.	**Mini Project:** Create a dApp (de-centralized app) for e-voting system.

CO PO and PSO Mapping

A. Course Outcome

Course Outcome	Statement	BL
	At the end of the course, student will be able to	
410246.6	Interpret the basic concepts in Blockchain technology and its applications	BL5

Course Outcome	Program Outcomes												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
410246.6	2	1	1	1	-	-	-	1	1	-	-	1	-	2

INDEX

Sr. No.	Ass No.	Title of Assignment	COs	POs
1	1	Understand the concept of Metamask with the procedure of installation for spending Ether per transaction.	CO6	PO1,PO2,PO3, PO4,PPO5
2	2	Create your own wallet using Metamask for crypto transactions and check how it differs from other crypto wallets.	CO6	PO1,PO2,PO3, PO4,PPO5
3	3	Using a Bank application, write a smart contract and perform following operations on Remix IDE. ● Deposit money ● Withdraw Money ● Show balance	CO6	PO1,PO2,PO3, PO4,PPO5
4	4	Write a program in solidity to create Student data. Use the following constructs: ● Structures ● Arrays ● Fallback Deploy this as a smart contract on Ethereum and Observe the transaction fee and Gas values.	CO6	PO1,PO2,PO3, PO4,PPO5
5	5	Write a survey report on types of Blockchains and its real time use cases.	CO6	PO1,PO2,PO3, PO4,PPO5
6	6	Mini Project: Create a dApp (decentralized app) for ● e-voting system. ● Hospital record management system ● Decentralized system for any real time application.	CO6	PO1,PO2,PO3 ,PO4, PO5,PO8,PO9, PO12
7	7	Content beyond syllabus : ● Supply chain monitoring	CO6	PO4,PO6,PO9, PO10,PO11
8	8	Virtual Lab : Study Cryptography and perform Digital Signature Algorithms.	CO6	PO9, PO10

Software Required:

1. 64 bit open source operating system
2. Remix IDE

Write-ups must include:

- **Assignment No.**
- **Title**
- **Problem Statement**
- **Prerequisites**
- **Course Objectives**
- **Course Outcomes**
- **Theory(in brief)**
- **Conclusion**
- **FAQs:**
- **Output: Printout of program with output.**

ASSIGNMENT NO: 1

TITLE: Installation of Metamask.

PROBLEM STATEMENT: Understand the concept of Metamask with the procedure of installation for spending Ether per transaction.

PREREQUISITES:

Understanding the basics concept of Metamask.

COURSE OBJECTIVE:

Learn installation and use of Metamask.

COURSE OUTCOME:

Able to install Metamask

Able to use Metamask

THEORY:

MetaMask is a type of cryptocurrency wallet that is used to interact with user interfaces of different web3 applications(For example, Mist browsers, DApps) and the regular web (For example, Google Chrome, Mozilla Firefox, websites).

Its function is to inject a JavaScript library called web3.js into the namespace of each page your browser loads. Web3.js is written by the Ethereum core team. MetaMask is mainly used as a plugin in the web browser. Let's walk through the steps to install it on Google Chrome.

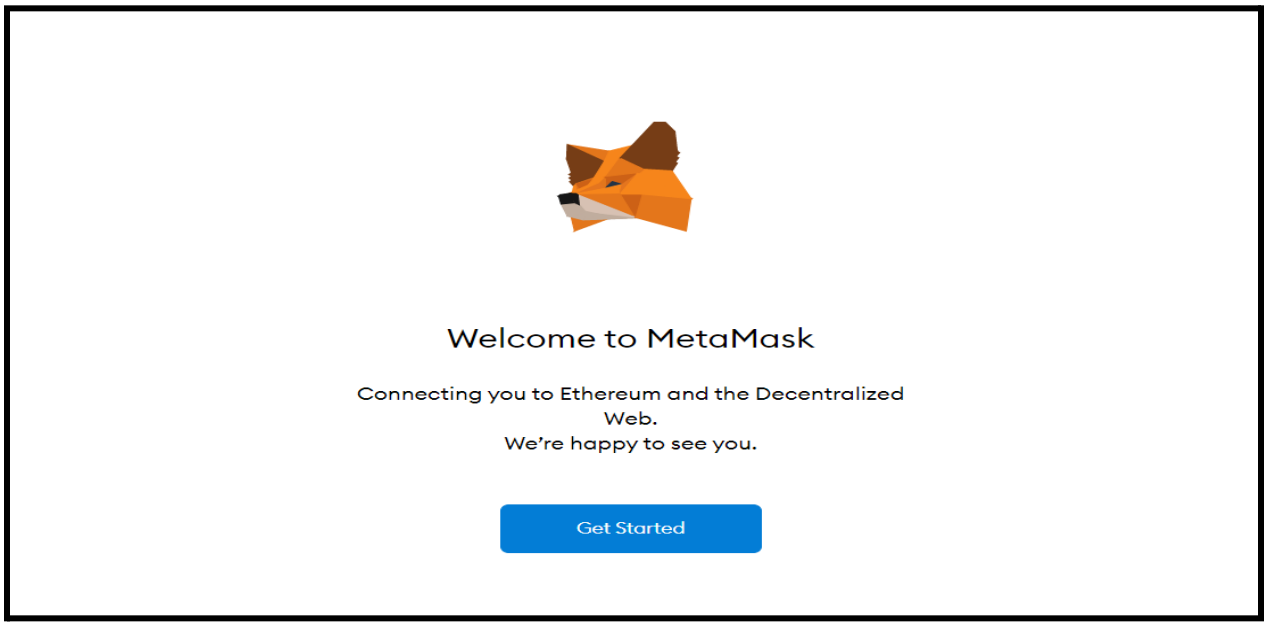
Step 1: Go to Chrome Web Store Extensions Section.

Step 2: Search MetaMask.

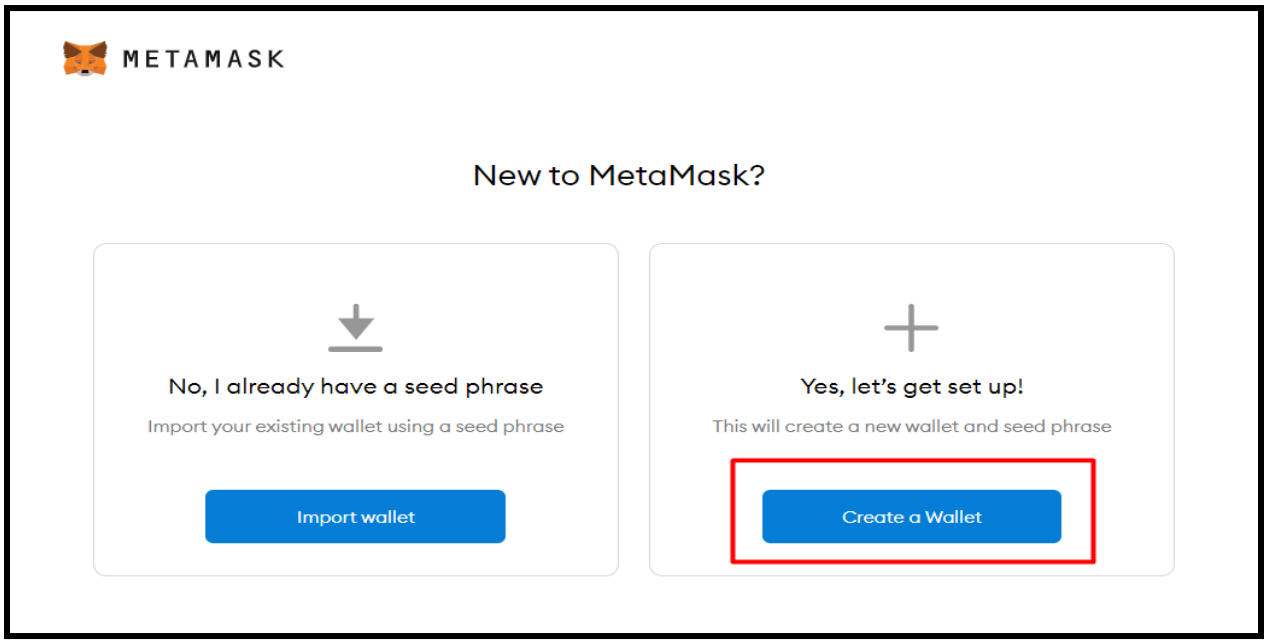
Step 3: Check the number of downloads to make sure that the legitimate MetaMask is being installed, as hackers might try to make clones of it.

Step 4: Click the Add to Chrome button.

Step 5: Once installation is complete this page will be displayed. Click on the Get Started button.

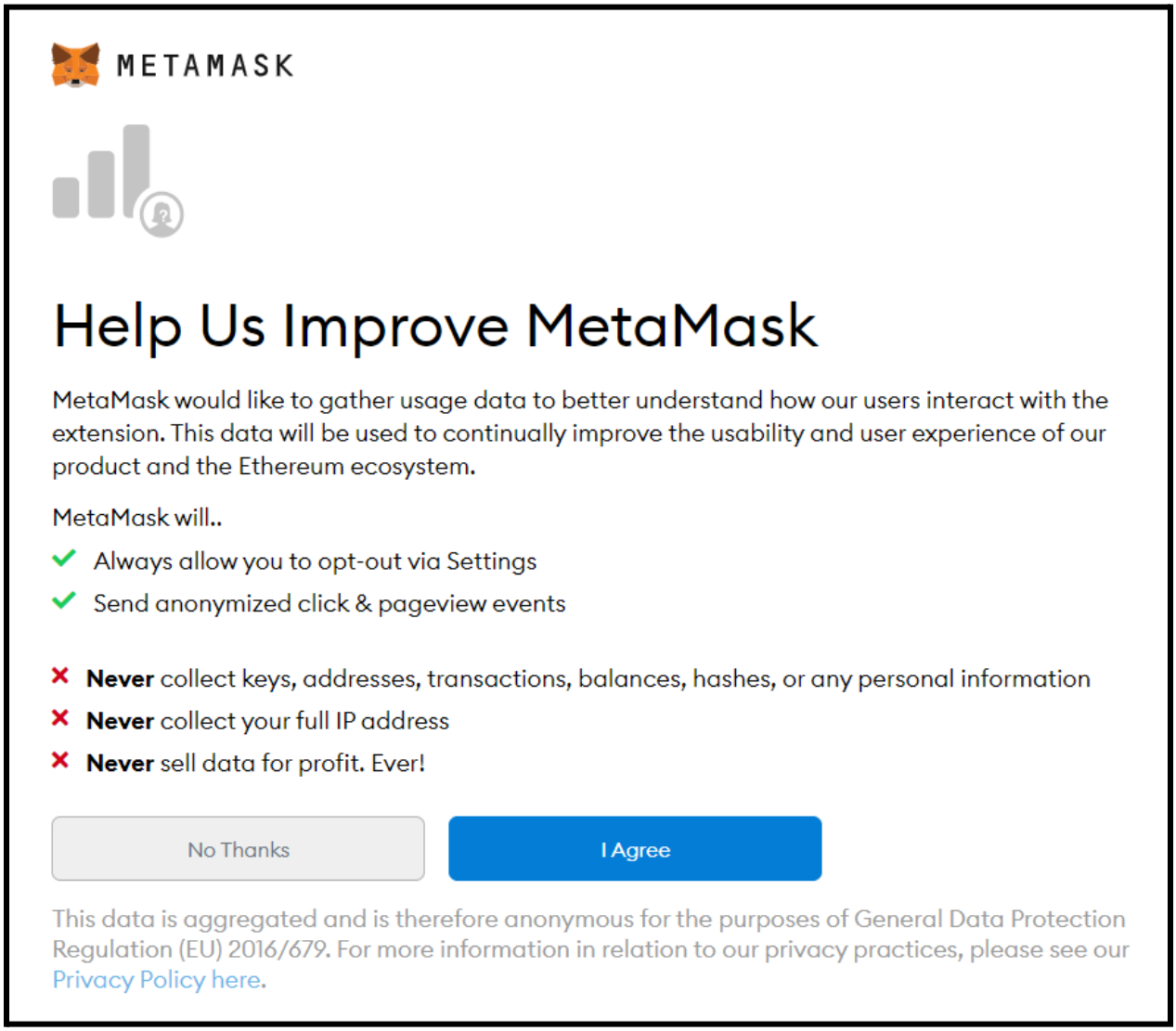


Step 6: This is the first time creating a wallet, so click the Create a Wallet button. If there is already a wallet then import the already created using the Import Wallet button.



Step 7: Click the I Agree button to allow data to be collected to help improve MetaMask or else click the No Thanks button. The wallet can still be created even if the user will click on the No Thanks button.

Step 8: Create a password for your wallet. This password is to be entered every time the browser is launched and wants to use MetaMask. A new password needs to be created if chrome is uninstalled or if there is a switching of browsers. In that case, go through the Import Wallet button. This is because MetaMask stores the keys in the browser. Agree to Terms of Use.



CONCLUSION: Hence we have successfully installed Metamask.

FAQs:

Why do we need metamask?

MetaMask is a browser plugin that serves as an Ethereum wallet, and is installed like any other browser plugin. Once it's installed, it allows users to store Ether and other ERC-20 tokens, enabling them to transact with any Ethereum address.

Why is Metamask safer than Coinbase?

MetaMask and Coinbase wallet are both non-custodial wallets and can both be secure with a Ledger hardware wallet so they are essentially even when it comes to security.

Is Metamask the best wallet?

Yes, MetaMask is the perfect beginner wallet for someone who is only interested in Ethereum. If you're also interested in Bitcoin or another cryptocurrency, then you may want to opt for a different wallet.

OUTPUT :

ASSIGNMENT NO: 2

TITLE: Create your own wallet using Metamask for crypto transactions and check how it differs from other crypto wallets.

PROBLEM STATEMENT: Understand the concept of Metamask with the procedure of installation for spending Ether per transaction.

PREREQUISITES:

Understanding the basics concept of Metamask.

COURSE OBJECTIVE:

Learn installation and use of Metamask.

COURSE OUTCOME:

Able to perform Transactions using Metamask

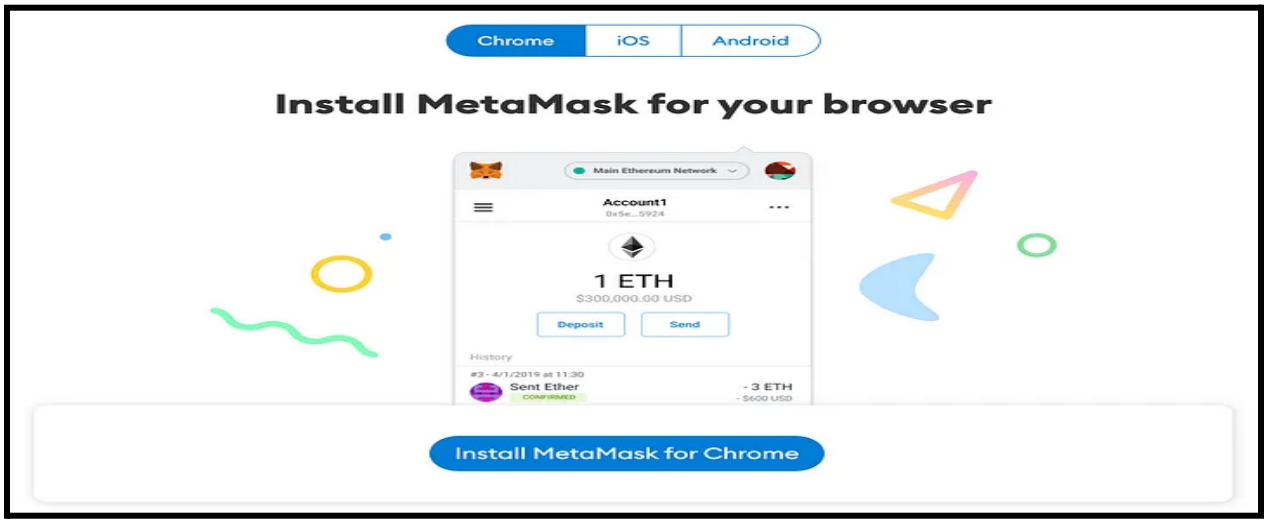
THEORY:

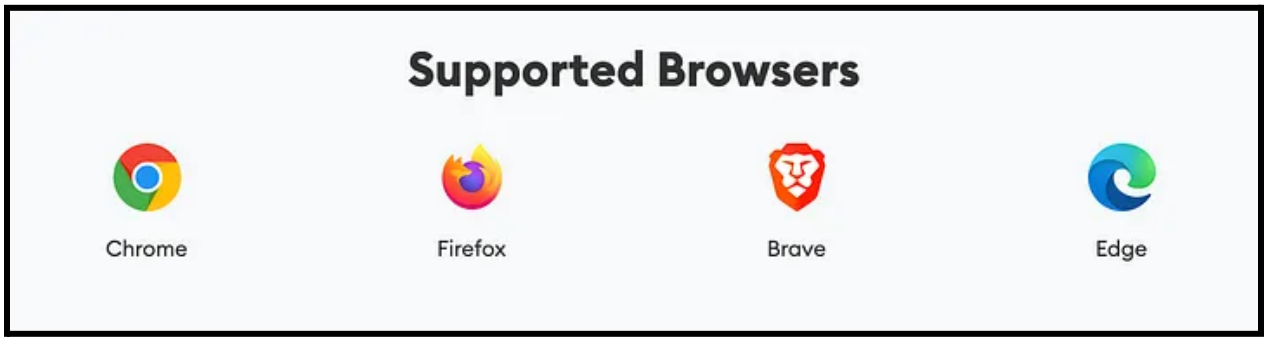
MetaMask is a cryptocurrency wallet used to interact with the Ethereum blockchain. It can be accessed through an app or through a browser extension.

Step 1: Download MetaMask wallet

Go to https://metamask.io/ and click on "Download". Choose your preferred browser or mobile application and install the MetaMask extension.

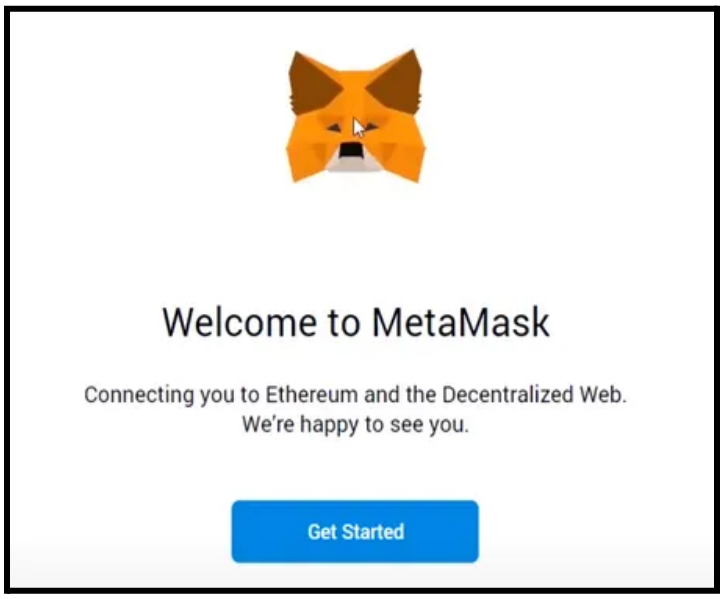
MetaMask supports iOS, Android native apps along with Chrome, Firefox, Brave and Edge browser extensions.



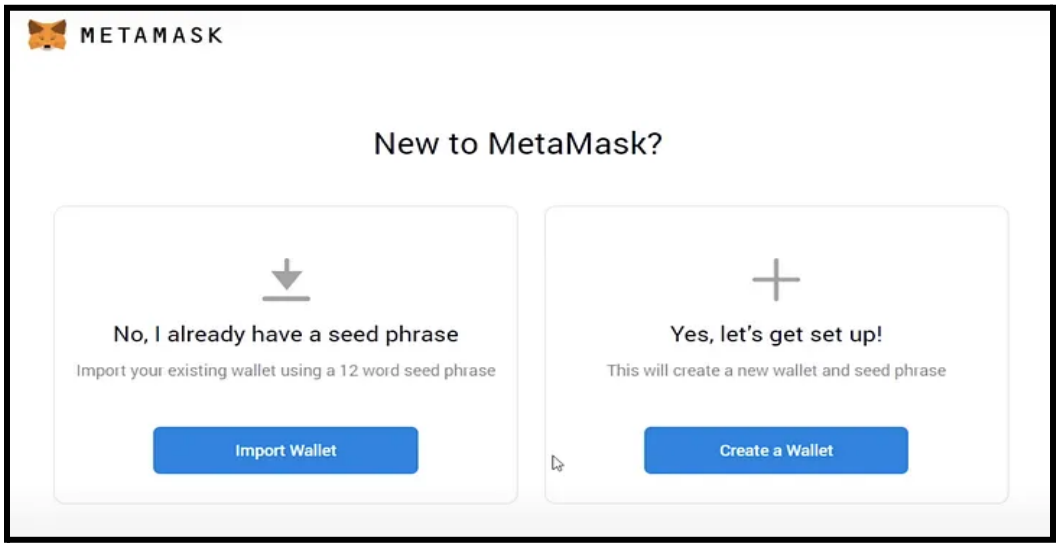


Step 2: MetaMask wallet installation

Click on the MetaMask extension and click on "Get Started".

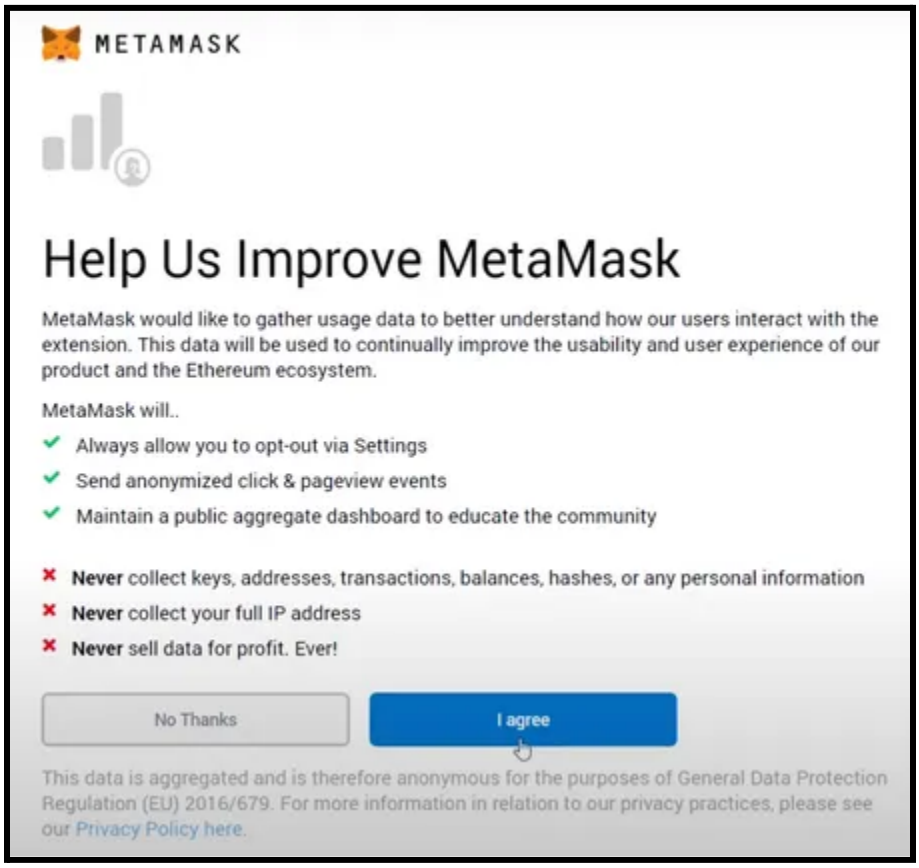


You can either import an existent wallet using the seed phrase or create a new one.

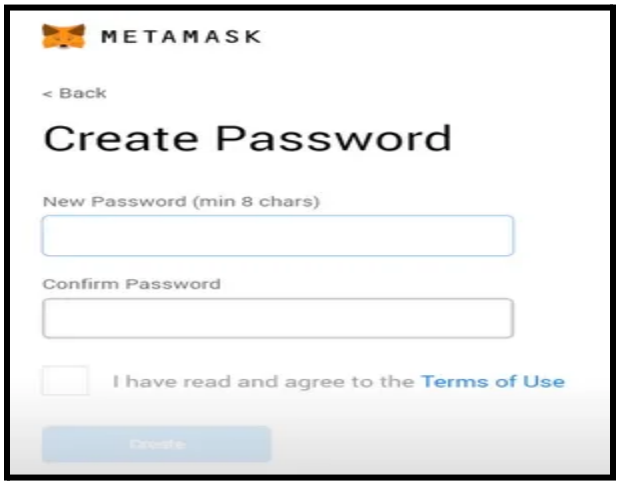


Step 3: How to create a new MetaMask wallet

Click on “Create a Wallet” and in the next window click on “I agree” if you would like to help improve MetaMask or click on “No Thanks” to proceed.



Step 4: Create a strong password for your wallet.



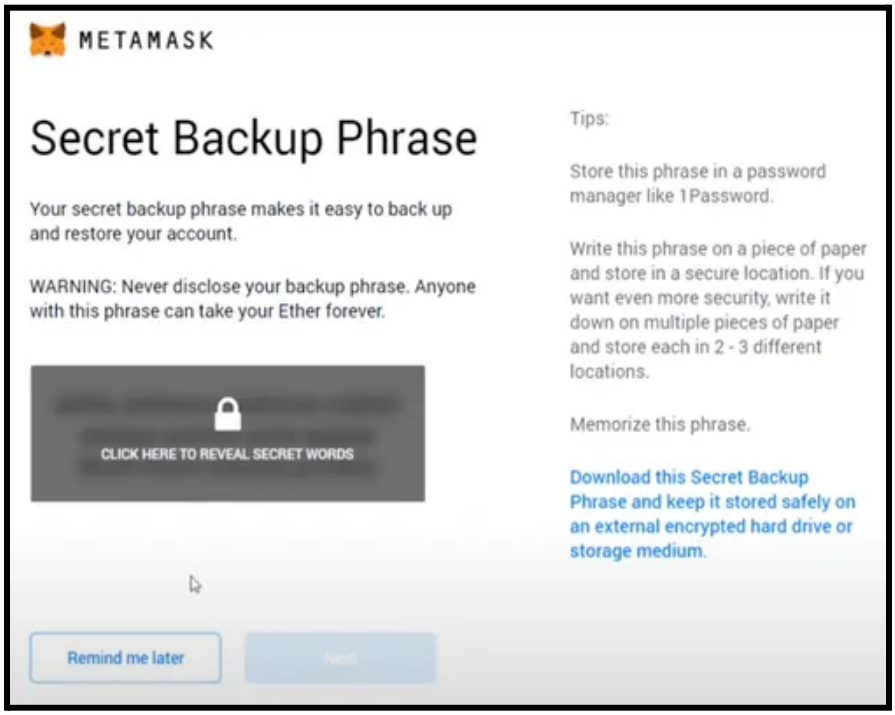
Step 5: Securely store the seed phrase for your wallet Click on "Click here to reveal secret words" to show the seed phrase.

MetaMask requires that you store your seed phrase in a safe place. It is the only way to recover your funds should your device crash or your browser reset.

We recommend you write it down. The most common method is to write your 12-word phrase on a piece of paper and store it safely in a place where only you have access.

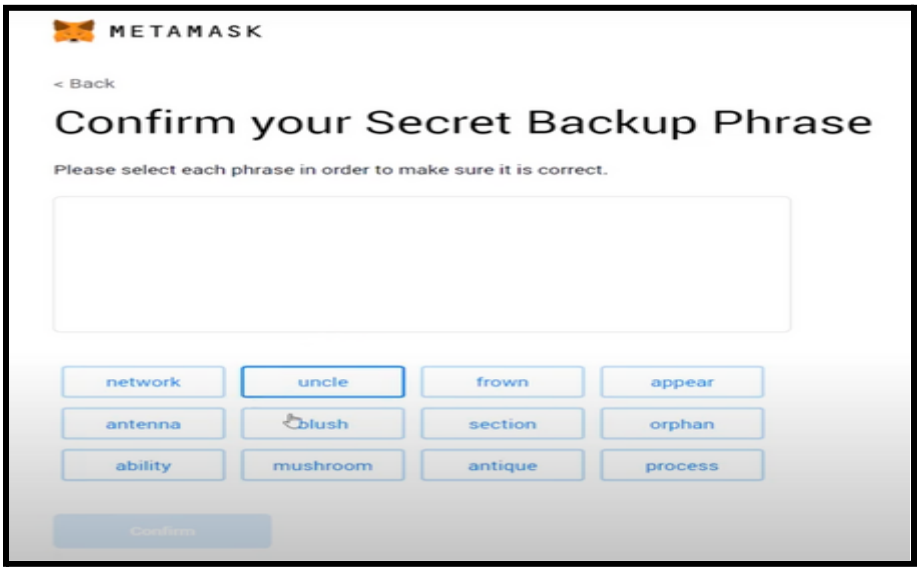
Note: if you lose your seed phrase, MetaMask can't help you recover your wallet and your funds will be lost forever.

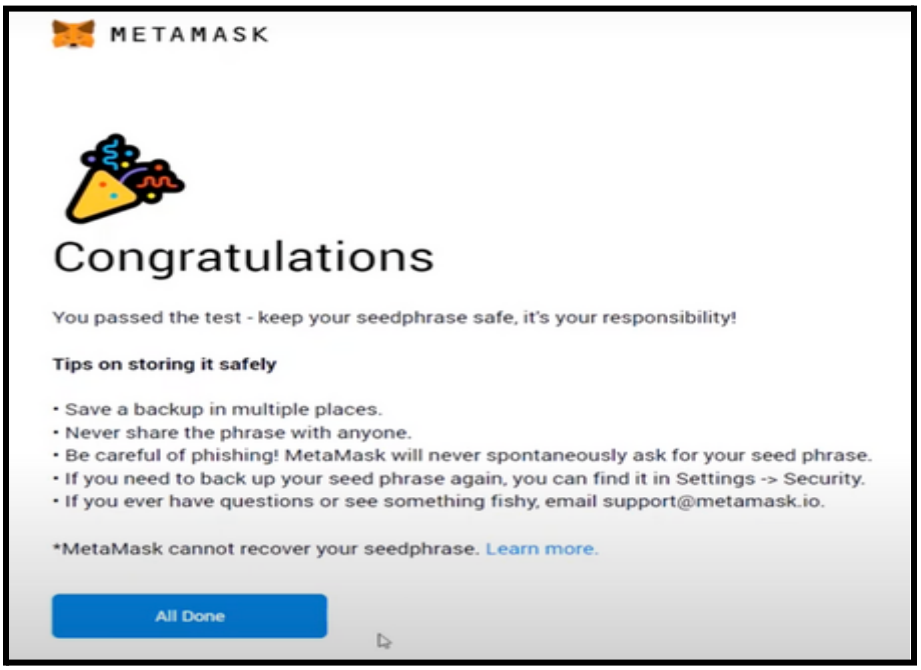
- Never share your seed phrase or your private key to anyone or any site, unless you want them to have full control over your funds.



Click on "Next".
Step 6: Seed phrase confirmation

Confirm your secret backup phrase by clicking on each word in the order in which the words were presented on the previous screen. Click on "Confirm" to proceed.





Congratulations! Your MetaMask wallet has been set up successfully.
You can now access your wallet by clicking on the MetaMask icon at the top-right-end corner of your preferred browser.

CONCLUSION: Hence , We have Successfully Created a wallet to perform crypto translation.

FAQS:

What advantages does MetaMask have?

● Popular - It is commonly used, so users only need one plugin to access a wide range of dapps.
● Simple - Instead of managing private keys, users just need to remember a list of words, and transactions are signed on their behalf.
● Saves space - Users don't have to download the Ethereum blockchain, as MetaMask sends requests to nodes outside of the user's computer.
● Integrated - Dapps are designed to work with MetaMask, so it becomes much easier to send Ether in and out.

What disadvantages does it have?

Third-party - MetaMask holds private keys within the user's browser. This is less secure

than a hardware or paper wallet, but is a reasonable compromise for the ease-of-use.

What are the alternatives to MetaMask?

● Coinbase Wallet - A self-custody wallet developed by crypto exchange Coinbase.
● Brave Wallet - a wallet developed by crypto-powered browser Brave and integrated directly into the browser.
● MyEtherWallet - A popular online wallet which enables you to use Ethereum coins and ERC-20 tokens.

OUTPUT :

ASSIGNMENT NO: 3

TITLE : Build banking applications using operations on Remix IDE

PROBLEM STATEMENT: Using a Bank application, write a smart contract and perform following operations on Remix IDE

- Deposit money
- Withdraw Money
- Show balance

PREREQUISITES:

Understanding the basics about operations on Remix IDE and bank application scenario

COURSE OBJECTIVE:

Learn installation and use of Remix IDE

COURSE OUTCOME:

Able to learn use of Remix IDE to develop bank application

THEORY:

A Brief introduction to MetaMask

MetaMask is an open-source, straightforward, and easy-to-use cryptocurrency wallet. It functions as a web browser extension available for Chrome, Firefox, Brave, or a mobile application for iOS or Android. Initially, this wallet supported only Ether and ERC-20 tokens, and now it is compatible with ERC-721 and ERC-1155 token standards. Furthermore, MetaMask benefits include interaction with websites; hence, it can function as a connection node for various DApps on Ethereum.

Adrian Devis and Dan Finlay are the MetaMask developers. Their idea was revolutionary and straightforward; they intended to create a web browser extension that would allow managing cryptocurrency and using the browser for fast and secure access with DApps. ConsenSys Software Inc. — a development company, focusing on applications that use Ethereum ‘s blockchain, implemented the idea in 2016.

The solution used Ethereum ‘s interface and a web API called web3.js. This Ethereum library is the fundament of MetaMask since it allows the browser to interact with the local or remote blockchain nodes via HTTP, IPC, and WebSocket; also, it gained the ability to record and read data from smart contracts, transfer tokens, etc. In another way, web3.js allowed the blockchain developers to create proxy and communication bridges between MetaMask, DApps, and the user.

You need an environment to code, compile, test and deploy your smart contracts. Truffle is the most popular development framework for EVM based blockchains. But to keep this tutorial

simple let's use an online-based IDE called Remix.

Remix IDE is a browser-based development environment for Smart Contracts. The IDE provides several plugins and compilers for different solidity versions.

To launch the Remix IDE, visit http://remix.ethereum.org

Next, let's configure the compiler version by clicking the 'Solidity Compiler' button and selecting the compiler configuration shown in the image below.

Banking Smart Contract

This smart contract will have all basic functionalities like ,

- Account Creation
- Deposit Amount
- Withdraw amount
- Transfer Amount
- Send Amount to wallet

First need to add solidity compiler version

```
// SPDX-License-Identifier: MIT
pragma solidity  >=0.4.22 <0.7.0;
```

Then creating Banking contract ,

```
contract banking{
...
}
```

now let's create variables or objects

```
 mapping(address=>uint) public userAccount;
  mapping(address=>bool) public userExists;
```

Here userAccount will contain amount for each registered acccount and userExists for account restrictions .

Banking Functions

Now lets Create functions for each mentioned operations ,

1. createAcc() functions :

```
function createAcc() public payable returns(string memory){
    require(userExists[msg.sender]==false,'Account Already Created');
    if(msg.value==0){
        userAccount[msg.sender]=0;
        userExists[msg.sender]=true;
        return 'account created';
    }
    require(userExists[msg.sender]==false,'account already created');
    userAccount[msg.sender] = msg.value;
    userExists[msg.sender] = true;
    return 'account created';

}
```

Here we create user account using boolean method by making userExists mapping true after using createAcc() function

2. deposit() function :

```
function deposit() public payable returns(string memory){
    require(userExists[msg.sender]==true, 'Account is not created');
    require(msg.value>0, 'Value for deposit is Zero');
    userAccount[msg.sender]=userAccount[msg.sender]+msg.value;
    return 'Deposited Succesfully';

}
```

With the help of userExists mapping we are only allowing registered users to deposit into our Smart Contract Bank .

3. withdraw(uint amount) function :

```
function withdraw(uint amount) public payable returns(string memory){
    require(userAccount[msg.sender]>amount, 'insufficeint balance in Bank account');
    require(userExists[msg.sender]==true, 'Account is not created');
```

```
    require(amount>0, 'Enter non-zero value for withdrawal');
    userAccount[msg.sender]=userAccount[msg.sender]-amount;
    msg.sender.transfer(amount);
    return 'withdrawal Successful';

  }
```

4. TransferAmount() function :

```
 function TransferAmount(address payable userAddress, uint amount) public returns(string
memory){
    require(userAccount[msg.sender]>amount, 'insufficeint balance in Bank account');
    require(userExists[msg.sender]==true, 'Account is not created');
    require(userExists[userAddress]==true, 'to Transfer account does not exists in bank accounts
');
    require(amount>0, 'Enter non-zero value for sending');
    userAccount[msg.sender]=userAccount[msg.sender]-amount;
    userAccount[userAddress]=userAccount[userAddress]+amount;
    return 'transfer successfully';

  }
```

CONCLUSION: We have to develop bank application scenarios using Remix IDE

FAQs:

Q. 1 How long does it take to transfer ether?

Ethereum confirmation times vary widely depending upon the amount of gas one is willing to spend, along with other market factors. You can view the median wait times at ethgasstation.info.

Q. 2 What is Ethereum coded in?

Ethereum smart contracts can be coded in Solidity, Serpent, LLL, and Mutan. These are contract-oriented, high-level languages. Their purpose is to target the Ethereum Virtual Machine (EVM) to provide a means for smart contracts. The Ethereum protocol has been developed using a variety of languages, from C++ to Python, Ruby, Go, Java, Rust, and more.

Q. 3 What is an Ethereum block?

A “block” in the Ethereum blockchain refers to a block of transactions that has been broadcast to the network. The Ethereum mainnet currently uses the Proof of Work consensus algorithm to verify blocks of transactions.

Output :

ASSIGNMENT NO: 4

TITLE: Create Student Data in Solidity Program with help of construct.

PROBLEM STATEMENT : Write a program in solidity to create Student data. Use the following constructs:Structures,Arrays,Fallback Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values

PREREQUISITES: Understanding the use of constructs to develop student data with the help of solidity.

COURSE OBJECTIVE: Understand and explore the working of Blockchain Technology and its applications.

COURSE OUTCOME: Understand the use of constructs to create student data.

Theory:

Introduction

Blockchain is a decentralized, distributed public ledger that lets us collaborate and coordinate the members that do not trust each other to make a secure transaction. Many of you understand blockchain as a bitcoin, but bitcoin is a cryptocurrency that takes the help of blockchain technology to operate.

Blockchain Technology

First, when Windows was launched to create reports and work on any project, we used MS word, where only one person could edit at a time and then send to another, and the process goes on, one after the other method. After some technological evolution, we have seen Google docs, google sheets in the market where online multiple people can work on a single document simultaneously. But the problem here is that it works on centralized architecture where a single server maintains google docs and various nodes are connected. Still, if the significant server crashes or gets corrupt, all the nodes get disconnected, and all the work gets destroyed. So the solution to this is decentralized blockchain technology. Decentralized means that no single server or single node is managing the network. Data is replicated to multiple nodes so that if any node goes down, other nodes operate as it is, and original data can quickly be recovered.

What is Ethereum?

Ethereum is a decentralized blockchain designed to be highly secure, fault-tolerant, and programmable. Ethereum blockchain is a choice for many developers and businesses. As said programmable, the main task of Ethereum is to securely execute and verify the application code known as smart contracts. Ethereum helps to build native scripting language(solidity) and EVM. Ethereum consensus mechanism is proof of work to operate to verify the new transaction. Now we will learn about smart contracts and how it runs on the Ethereum platform.

Overview of Smart Contracts

A smart contract is a small program that runs on an Ethereum blockchain. Once the smart contract is deployed on the Ethereum blockchain, it cannot be changed. To deploy the smart contract to Ethereum, you must pay the ether (ETH) cost. Understand it as a digital agreement that builds trust and allows both parties to agree on a particular set of conditions that cannot be tampered with.

Learn | Write | Earn

To understand the need for a smart contract, suppose there was one grocery shop, and ram went to buy some groceries. He purchased the groceries for 500 rupees and kept on debt that would pay the money next month when he returned, so the shopkeeper jotted down his purchase in his ledger. In between the period somehow shopkeeper changed 500 to 600 and when next month ram went to pay the money, the shopkeeper has demanded 600 INR and ram has no proof to show that he has only bought 500 INR so in this case, smart contracts play an essential role which prevents both the parties to tamper the agreement and only gets terminate when all the conditions satisfy after the deal. There are a couple of languages to write smart contracts, but the most popular is solidity.

Introduction to Solidity Programming

Solidity is object-oriented, high-level statically-typed programing language used to create smart contracts. Solidity programming looks similar to Javascript, but there are a lot of differences between both languages. In solidity, you need to compile the program first, while in Javascript, you can run the program directly in your browser or by using Node JS. With solidity, you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets. It is also a case-sensitive programming language. Visit the official solidity documentation to read more and be updated about any new functionality release.

What is Remix IDE?

It is an online IDE for creating solid, smart contracts, so you do not need to install or download anything to do any setup. You can develop, deploy, and Administer your solidity smart contract using Remix IDE. Visit this link to access the Remix IDE where you will find multiple options and a window shown below. The window is a little bit similar to VS code where on the left-hand side, you will find some icons to terminate to other options like a compiler, file explorer, search files, deploy, etc.

What is Remix IDE?

It is an online IDE for creating solid, smart contracts, so you do not need to install or download anything to do any setup. You can develop, deploy, and Administer your solidity smart contract using Remix IDE. Visit this link to access the Remix IDE where you will find multiple options and a window shown below. The window is a little bit similar to VS code where on the left-hand side, you will find some icons to terminate to other options like a compiler, file explorer, search files, deploy, etc.

Using the file explorer, you can create and open any file. We can set the compiler version, run our smart contract, and observe the output using the compiler. Each compiler type provides a different amount of fake ethers used for practicing purposes. **Solidity Compilation Process** Smart contract compilation is a critical process to understand how a smart contract runs when created using solidity. We will understand the process using the below flow chart. We can see that the smart contract written in solidity with sol extension first gets the compiler version. After it goes under the compiler, It gets split into two parts where one is Byte code, and the other is ABI (Abstract Binary Interface) key. Byte code is only executed and deployed on the Ethereum blockchain, not the complete smart contract. Whenever any smart contract wants to communicate with this smart contract, they need the ABI key to call functions and variables.

To observe how ABI and Byte code is generated on Remix IDE, visit IDE, open any contract in the contracts folder, and compile and run it. Scroll down, and you will find two options: ABI and Byte code, where you can copy and paste them into any notepad and observe how your code gets converted to Byte code.

To create a smart contract, the first thing is to define the compiler version to use using the Pragma keyword (you can also determine whether the program supports multiple versions or the version in a particular range); after this, you define the contract using the contract keyword which is same as creating a class in object-oriented programming.

Important points related to smart contract Compilation

1. Contract Bytecode is public in readable form – It means It does not get encrypted because It will run on different nodes of Ethereum. For then, It needs to decrypt again and again not to increase computation time. It is kept in a readable form.
2. The contract doesn‘t have to be public – It does not need to keep contracts public, but most organizations keep them public to maintain the trust.
3. Bytecode is immutable
4. ABI act as a bridge between application and smart contract
5. ABI and bytecode cannot be generated without source code

State and Local Variables in Solidity

Any variable declared on the contract level is known as a state variable. The critical property of the state variable is that it is permanently stored in the blockchain, so you have to pay some amount of gas and use the state variable with care. Solidity does not have a concept of Null or None; indeed, each data type has one default value which on declaration is assigned to that variable. To define Public before any variable or function, automatically, one get function is set with that variable, and you can access its value. Storage to state variable is not dynamically allocated (To initialize state variable with the value, you need to assign a value at declaration time, use constructor, use getter and setter functions). An instance of a contract variable cannot have another state variable besides those already declared.Local variables are those variables that are declared in the function body and are stored in a stack, not in contract storage. Local variables don‘t cost gas; some types reference the storage by default. Memory keywords cannot be used at the contract level.

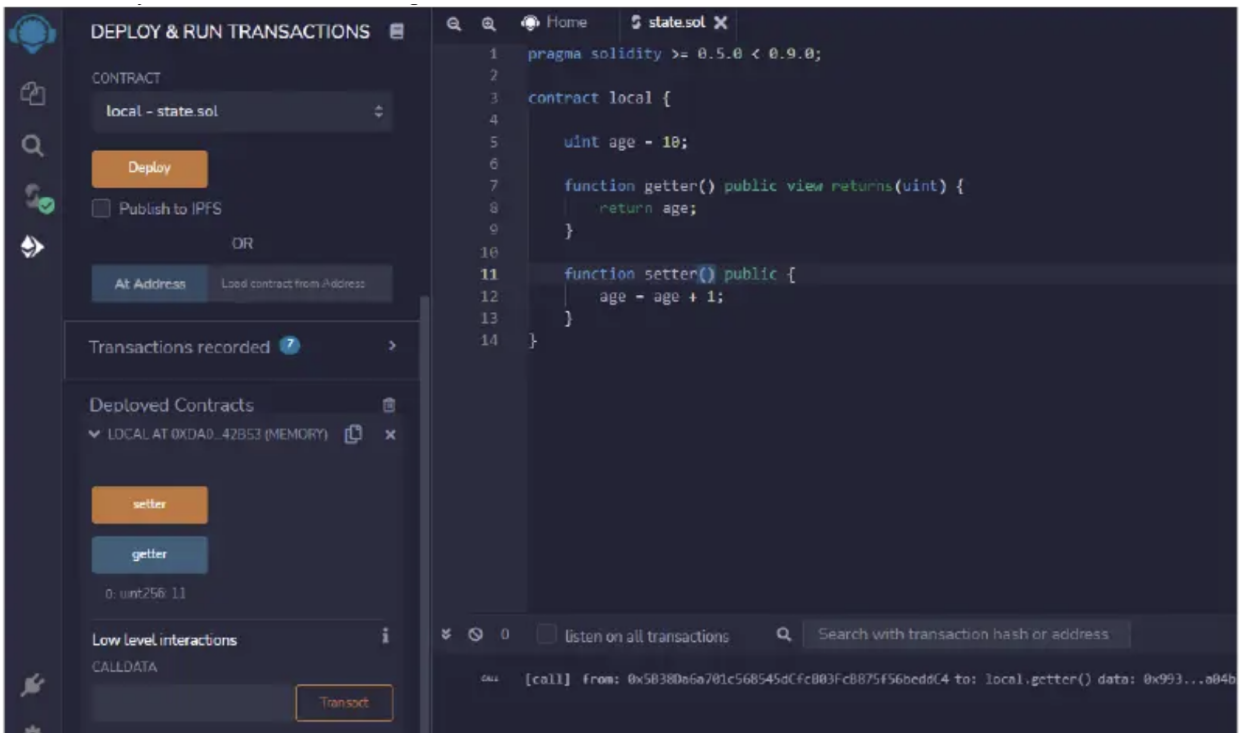
Functions in Solidity

Functions are an essential part of any programming language for the reusability of a particular code. We will see the getter and setter function in solidity to learn how to create a function in solidity. The getter function is a function from which we can access the value of our variables. It is a view-only function, so we can define it as a view or Pure, which states that the value of the state variable cannot be changed it returns the

variable ‘s value, so we define the return type of value. On the other side, the setter function changes the value, so it is a simple public function.

```
pragma solidity >= 0.5.0 < 0.9.0;
contract local {
uint age = 10;
function getter() public view returns(uint) {
return age;
}
function setter() public {
age = age + 1;
}
}
```

After writing the above code on Remix IDE in a new file with sol extension, you can compile the code, visit the deploy section, and deploy the code to observe the deploy section output as shown below. The value will increase as you click the setter and getter function buttons.



Suppose you want to implement the setter function to set the new value of age so we can pass the parameter in the setter function and set the value of age. Thus, in the setter function, we change the matter, so we need to pay a certain amount of gas in the setter function, but the getter function is view-only and does not require any amount of gas to be paid. By default, the visibility of a function is private, so to make it public, we define a function as Public.

Pure and View in Solidity

We have seen that we use to view and pure where we are not updating the state variable. But pure, you cannot use where it is also reading the state variable. Pure is used where both reading and writing are not performed. Indeed, in View, reading is allowed, but writing is not permitted. When we do not define any one of the following to a function, it simply warns that we can provide one restriction of pure or View to function.

Constructor in Solidity

A constructor is a particular type of function which executes only once when you create your contract. Constructor is used to work with state variables and define smart contract's owners. You can create only one constructor, and it is optional to create. The compiler creates a default constructor if there is no explicitly defined constructor.

```
pragma solidity >= 0.5.0 < 0.9.0;
contract local {
uint public count;
constructor(uint new_count) {
count = new_count;
} }
```

the above code, we have created a constructor, and before clicking on deploy, you need to define the value of the constructor before it is called only once, so enter the value and click on deploy, and on scrolling, you can observe the value of count.

Control Statements in Solidity

All programming languages have control statements that help us check multiple conditions using loops and an if-else ladder, and solidity also supports loops and if-else statements.

Loops in Solidity

Solidity also supports three loops: a while loop, a Do while loop, and a loop. If you are familiar with any other programming language, you must know about control statements and using loops to run particular code multiple times with different values. In solidity, you cannot write the loops directly in contract storage; instead, you must declare them in any function.

While the loop runs a code snippet multiple times until the condition is proper, the loop terminates when the condition is false. In contrast, the loop runs 0 or multiple times.

Do while loop is a loop that runs even one time when the condition in the while loop is false. So it is used when you need to run a particular code at least once and if certain conditions meet, then run it multiple times.

For loop is a loop that is used when you know the start and end time of the loop and how many intervals you need to take. For loop, the initialization and iterator updating are part of loop syntax.

So let us look after the syntax of each type of loop using a sample program.

```
pragma solidity >= 0.5.0 < 0.9.0;
contract Loops {
uint [3] public arr;
uint public count;
function Whileloop() public {
while(count < arr.length) {
arr[count] = count;
count++;
} }
function Forloop() public {
for(uint i=count; i<arr.length; i++) {
arr[count] = count;
count++;
} }
function doWhileLoop() public {
do {
arr[count] = count;
count++;
}while(count < arr.length);
} }
```

If-else Statements in Solidity

If-else statements are an essential part of any programming language that helps compare two or more two types of values to make a particular decision. Below is the sample code snippet denoting the use of if-else in the solidity that you should try and deploy the contract. After deploying, check by entering the different values.

```
pragma solidity >= 0.5.0 < 0.9.0;
contract Array {
function check(int a) public pure returns(string memory) {
string memory value;
if(a > 0) {
value = "Greater Than zero";
}
else if(a == 0) {
value = "Equal to zero";
}
else {
value = "Less than zero";
}
return value;
} }
```

Arrays in Solidity

The array is a special data structure used to create a list of similar type values. The array can be of fixed size and dynamic-sized. With the help of index elements can be accessed easily. below is a sample code to create, and access a fixed-sized array in solidity.

```
pragma solidity >= 0.5.0 < 0.9.0;
contract Array {
uint [4] public arr = [10, 20, 30, 40];
function setter(uint index, uint value) public {
arr[index] = value;
}
function length() public view returns(uint) {
return arr.length;
} }
```

Creating Dynamic Array

A dynamic array is an array where we can insert any number of elements and delete the details easily using an index. So solidity has functions like push and pops like python, making it easy to create a dynamic array. Below is a code using which you can create a dynamic array. After writing code, compiles and deploy the code by visiting the deploy section in the left-side navigation bar. After that, try inserting and deleting some elements from an array.

```
pragma solidity >= 0.5.0 < 0.9.0;
contract Array {
uint [] public arr;
function PushElement(uint item) public {
arr.push(item);
}
function Length() public view returns(uint) {
return arr.length;
}
function PopElement() public {
arr.pop();
} }
```

Structure in Solidity

The structure is a user-defined data type that stores more than one data member of different data types. As in array, we can only store elements of the same data type, but in structure, you can keep elements of different data types used to create multiple collections. The structure can be made outside and inside the contract storage, and the Structure keyword can be used to declare the form. The structure is storage type, meaning we use it in-store only, and if we want to use it in function, then we need to use the memory keyword as we do in the case of a string.

```
pragma solidity >= 0.5.0 < 0.9.0;
struct Student {
uint rollNo;
string name;
}
contract Demo {
Student public s1;
constructor(uint _rollNo, string memory _name) {
s1.rollNo = _rollNo;
s1.name = _name;
}
```

Conclusion: We have studied program writing in solidity to create Student data by using constructs like structures, arrays, fallback and also deployed this as smart contract on Ethereum and Observe the transaction fee and Gas values.

FAQs:

1) What are features of Remix IDE ?
   Remix IDE is a no-setup tool with a GUI for developing smart contracts. Used by experts and beginners alike, Remix will get you going in double time. Remix plays well with other tools, and allows for a simple deployment process to the chain of your choice. Remix is famous for its visual debugger.

2) What is meant by Smart Contract ?
Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss.

Output :

ASSIGNMENT NO: 5

TITLE: Create Survey report on real time use cases.

PROBLEM STATEMENT : Write a survey report on types of Blockchains and its real time use cases.

PREREQUISITES: Understanding the applications of blockchain

COURSE OBJECTIVE: Understand and explore the working of Blockchain Technology in various applications.

COURSE OUTCOME: Understand the applications of blockchain.

Theory:

Blockchain & Types

Blockchain technology is being used to carry and transfer the transactions or exchange of information through a secure network. Blockchain technology and distributed ledger technology is used parallel to the digital cryptocurrency to the people. Blockchain is being used for the purpose of private networking and uses too where only the restricted network users can get the authorization and access. Here network administrators are authorized to administrate the activities and any new nodes or users who wish to get permission, need to contact with the system or network administrators. Primarily there are two types of Blockchain technology viz. private Blockchain and public Blockchain. Though based on some other criteria and analysis Blockchain technology can also be noted and called as consortium blockchain technology, and hybrid blockchain technology. It is important to note that every kind of Blockchain basically consists of a cluster of nodes, and this is working on the peer-to-peer (P2P) network system.

Every node in the network has a copy of the shared ledger and further that is timely updated and also being verified the transactions, initiate and receive transactions. Keeping in mind the broad nature, experts classified Blockchain Technology into following three

Public Blockchain,

Private Blockchain, and

Hybrid Blockchain.

1. **Public Blockchain** is a major type of Blockchain, and that is not only open but also decentralized in nature. And in this type of Blockchain technology computer networks are basically accessible to anyone interested in transactions. Here based on validation the validated person basically receives the transaction rewards and furthermore, two kinds of Proof-of-work and Proof-of-stake models are being used. The Public Blockchain is furthermore a non-restrictive and distributed ledger system which is doesn‘t seek any kind of permission, and anyone having access can be authorized one to get the data or part pf the Blockchain. This kind of Blockchain also gives authorization regarding the current and past records verification. Additionally, this is being used for mining and exchanging cryptocurrencies

In this segment most common is Bitcoin and Litecoin blockchains. It is mostly secure upon following strict security rules as well as methods. However, upon non-following the security protocols it may be risky. Some of the examples of this type of Blockchain are - Bitcoin, Ethereum, and Litecoin.

Here given figure depicted some of the features and advantages regarding Public Blockchain Systems and Technology.

Fig. Salient features and functions of the public blockchains

There are two common examples of public blockchains and these are Bitcoin and Ethereum as per the experts This type of Blockchain is concerned with the following type of features viz.

High Security and Privacy,

- Open and Flexible Environment,
- Anonymous Nature,
- No regulations and strict Policies,
- Full Transparency and Systems.
- Distributed, etc.

However, according to the experts, the following are being considered as important advantages and benefits of the Public Blockchain.

Trustable and Faith

Public Blockchainis trusted and here unlike private blockchains, the participants are don‘t need to think of authenticity. In this type of Public Blockchain, they no need of knowing other nodes, and therefore there is no fraud in the transactions. In this category nodes can contact blindly without feeling the needing to trust individual nodes

Secure and Safe

In the Public Blockchain,there are opportunities in connecting with the other participants and nodes in the same public network, and this results in secure, largest, and greater communication and participation. Owing to this feature, it is difficult for the attackers to enter the systems and here every node will do the mverifications and transactions as per norms. Here thoughtful cryptogenic encrypting methods are being used and therefore it is much safer than the private blockchain according to some experts.

Open and Transparent

Public Blockchain is also having the features of openness and here data is basically transparent to all the nodes and in this mechanism, one blockchain record is normally available to all the authorized nodes.Therefore, here all the nodes become open and transparent and there is an absence of fake transactions or hiding any information. Though there are plenty of advantages and benefits but it is also having a different kind of disadvantages and weaknesses, and some of them are as under.

Lower Transaction per Second

In Public Blockchain System the rate of transaction per second is also very low, and this is due to having a large number of nodes and huge network. Here each node has to verify the transaction and also do proof-of-work is time consuming. Here in public systems seven (07) transactions happen per second and additionally, here Ethereum network has about a 15 TPS rate.

Scalability Matters

Similar to the previously mentioned issue on a lower transaction per second in public blockchain another issue is scalability according to the experts. The huge size basically creates the scalability in this regard and here bitcoins lightning networks are considered as important to overcome the problem according to the experts.

High Energy Consumption

The public blockchain also suffers from higher energy consumption due to the proof-of-work energy consumption. As it needs special algorithms therefore high energy consumption is considered as important in energy, environment, and financial context.

2. Private blockchains

Private Block Chain are restricted and not open, such kind of blockchain also has features of access. This blockchain allows permission for the transaction from the support of the system administrator
Private blockchain solutions develop these platforms having the features of the following—
^^ Full of privacy,
^^ High efficiency,
^^ Faster transaction,
^^ Better scalability,
^^ Faster and speediness.
This type of blockchain is works on closed systems and networks only and these are usually useful in the organizations, enterprises from which only selected members can be joined. This type of blockchain contains proper security, authorizations, permissions as well as accessibility. According to the experts, private blockchains are deployed forvoting, regarding supply chain management, for finding and managing digital identity, regarding asset ownership,and so on. There are certain popular private blockchains like Multichain, Hyperledger projects, Corda, etc. Participants are

Fig.: Salient features and functions of the private blockchains

Private blockchains as running with the authorized nodes; therefore no one from the outside of the private network is able in accessing information and transaction related data exchanged between two nodes.

3. **Hybrid Blockchains** is a merger of public blockchain as well as private blockchain and it is required in better control for achieving higher goals. Hybrid Blockchain deals with centralized and decentralized systems and it is not open; however, it has the features of integrity, transparency, as well as security. It has several advantages over traditional blockchains as depicted in Fig. 5. In Hybrid Blockchains maximum customization is being considered as main benefits with private permission-based system as well as a public permission-less system. In this type of blockchain systems users are able in getting access and selected sections and rest can be recorded or keeps safe due to the benefits of the records from the ledger. Hybrid Blockchains is flexible enough so that users can join easily as private blockchain. This type of blockchain is able in enhancing the security and transparency of the blockchain network.

4. **Consortium Blockchain** is another type of semi-decentralized type of blockchain, and this type of blockchain is able in the organization of managing the blockchain network. This type of blockchain is able in doing activities even from a single organization. Here blockchain is able in exchange information or do the mining and are being used in the areas such as banks, government organizations, etc. Some of the examples of this type of consortium are Energy Web Foundation, R3, etc.

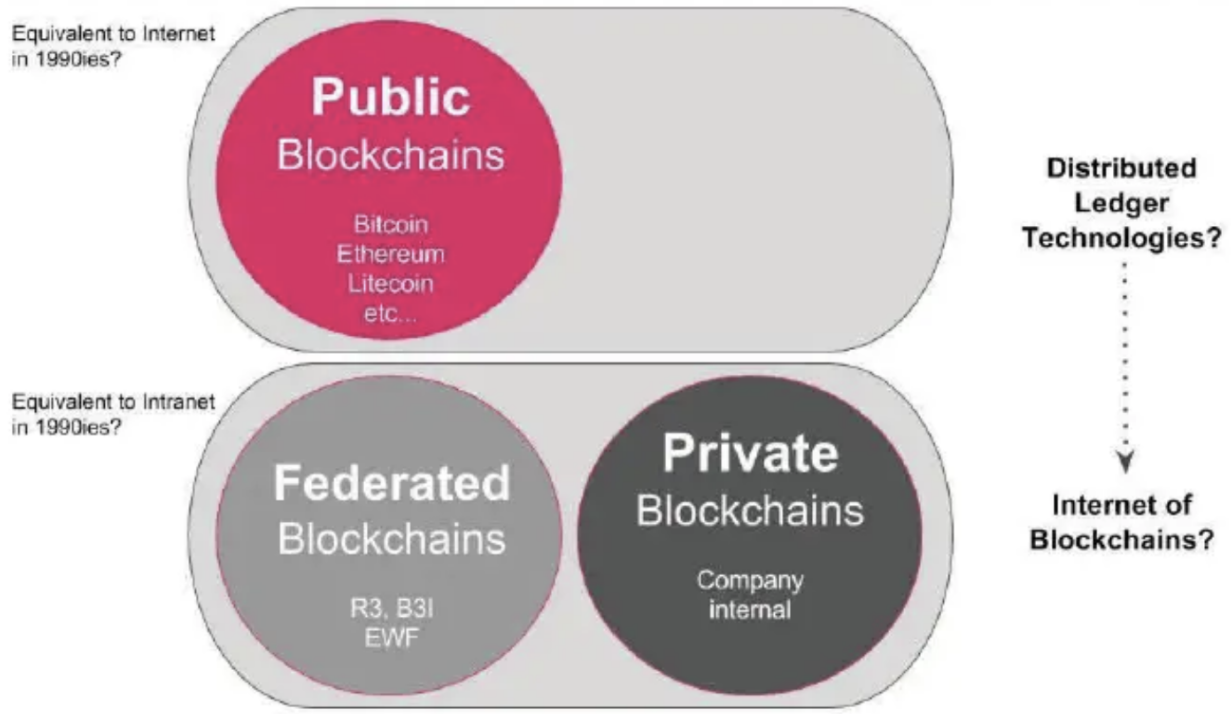


Fig. Types of Blockchains and roadmaps

Therefore, in a nutshell, all the Blockchains are having their own benefits and advantages and as a whole public and private is considered as major or worthy in terms of operations (as depicted in Fig. According to the expert's security, scalability, and transparency are considered as worthy and main points in the Blockchains of public and private types. It is important to note that private blockchains are not trustworthy; while the public network is important in proof-of-work based.

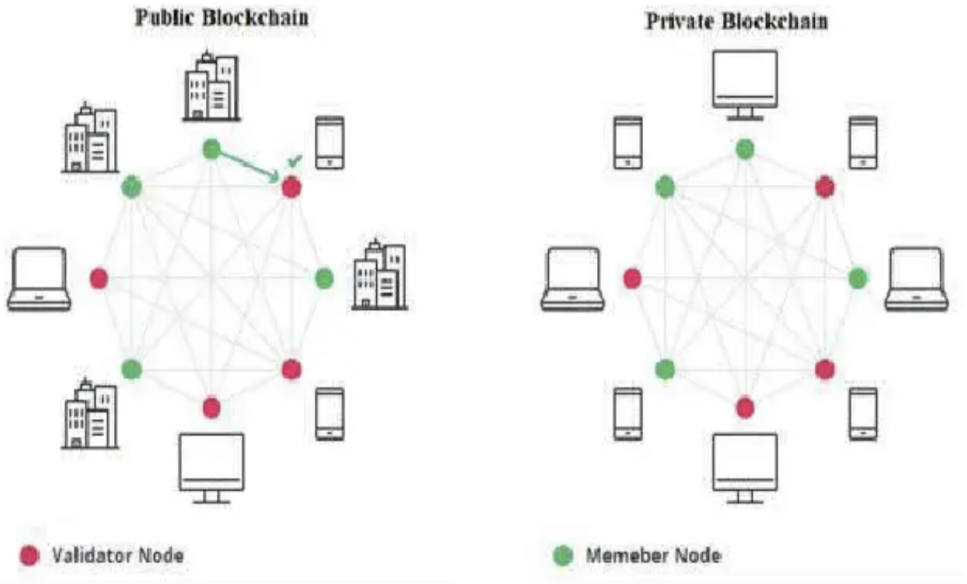


Fig. Architecture wise differences in public and private blockchains

Top blockchain use cases

Blockchain is "a general-purpose technology, which means it is applicable across sectors," said Christos Makridis, a research professor at Arizona State University, senior adviser at Gallup, digital fellow at Stanford University's Digital Economy Lab and CTO at arts and education technology start-up Living Opera. "For example, financial services can use it to write smart contracts between consumers and their banking institution. Similarly, healthcare can use it to write smart contracts between insurers and hospitals, as well as between patients and hospitals. The possibilities are endless." Blockchain use cases continue to expand. Here are some common commercial applications:

1. **Smart contracts.** The primary function of computer programs called "smart contracts" is to automate the execution of contract terms when conditions warrant them. The computer code follows a relatively simple command of "when/if _then___" to ensure that all parties receive the benefits or penalties as the contract stipulates and actions require. Smart contracts are useful to, and used by, most industries today for a variety of uses traditionally governed by paper contracts. The blockchain also makes a permanent record of every action and reaction in the transaction.

2. **Cybersecurity.** Blockchains are highly secure because of their permanency, transparency and distributed nature. With blockchain storage, there's no central entity to attack and no centralized database to breach. Because blockchains are decentralized, including those privately owned, and the data stored in each block is unchangeable, criminals can't access the information. "Essentially, the intruder needs keys to many different locations versus just one," Makridis noted. "The computing requirements for the intruder grow exponentially."

3. **IoT.** Two primary IoT uses of blockchains are in the supply chain sector and for asset tracking and inventory management. A third use is in recording measurements made by machines whether those sensors are in the Artic, the Amazon jungle, a manufacturing plant or on a NASA drone surveying Mars. "Whether it be reports of chemical data regarding oil grades or tracking shipments of electronics across the world through various ports of entry, the blockchain can be utilized anywhere there is data interacting with the real world," explained Aaron Rafferty, CEO of cryptocurrency investment firm R.F. Capital.

4. **Cryptocurrencies.** The blockchain concept was originally developed to manage digital currencies such as bitcoin. While the two technologies still compete against each other in alternative transactions, they've also been separated so blockchains could serve other purposes. Given the anonymity of crypto coins, blockchain is the only way to document transactions with accuracy and privacy for the parties involved.

5. NFTs. Nonfungible tokens are units of data certified to be unique and not interchangeable. In short, they are digital assets. According to Rafferty, NFTs are revolutionizing the digital art and collectibles world. "We are using decentralization and the ethereum blockchain to create a music live stream network where artists and streamers can connect with fans directly, sell their NFTs, receive contributions from fans and trade in their rewards and contributions for crypto tokens," said Shantal Anderson, founder and CEO of music and pop culture streaming network Reel Mood.

Conclusion: We have studied the survey report on types of Blockchains and its real time use cases.

FAQS:

1) What are applications of blockchain ?
   - Healthcare.
   - Finance and banking.
   - Real estate.
   - Retail.
   - Supply chain and logistics.
   - Insurance.
   - Voting and governance.

2) What happens when you try to deploy a file with multiple contracts?
   In Blockchain, deploying a file with multiple contracts is not possible. The compiler only deploys the last contract from the uploaded file and the remaining contracts are neglected.

Output :

ASSIGNMENT NO: 7

TITLE: Content Beyond Syllabus

PROBLEM STATEMENT : Supply chain monitoring

PREREQUISITES: Understanding the use of blockchain in Supply chain monitoring

COURSE OBJECTIVE: Understand and explore the working of Blockchain Technology in Supply chain monitoring.

COURSE OUTCOME: Able to understand supply chain monitoring with help of blockchain.

THEORY :

Supply chain management (SCM) refers to controlling the entire production flow, from acquiring raw materials to delivering the final product/service at the destination. In addition, it handles the movement of materials, information and finances associated with a good or service.

Even though the supply chain and logistics are sometimes confused, logistics is actually only one part of the supply chain. Traditional supply chain management systems involve steps like planning, sourcing, manufacturing, delivering and after-sales service to control the supply chain centrally.

That said, the process begins with deciding how to meet customers' needs and selecting suppliers to source the raw material to manufacture the product. The next step is to determine if the manufacturer will outsource or take care of delivery. And after a product is delivered, it is a network that will offer after-sales services, such as handling product returns and repairs, among others, which is crucial for customer satisfaction.

How does blockchain technology improve supply chain management?

Unlike traditional supply chains, blockchain-based supply chains will automatically update the data transaction records when a change is made, enhancing traceability along the overall supply chain network.

Blockchain-based supply chain networks might need a closed, private and permissioned blockchain with limited actors, in contrast to Bitcoin and other financial blockchain applications, which may be public. However, the possibility of a more open set of partnerships may still exist.

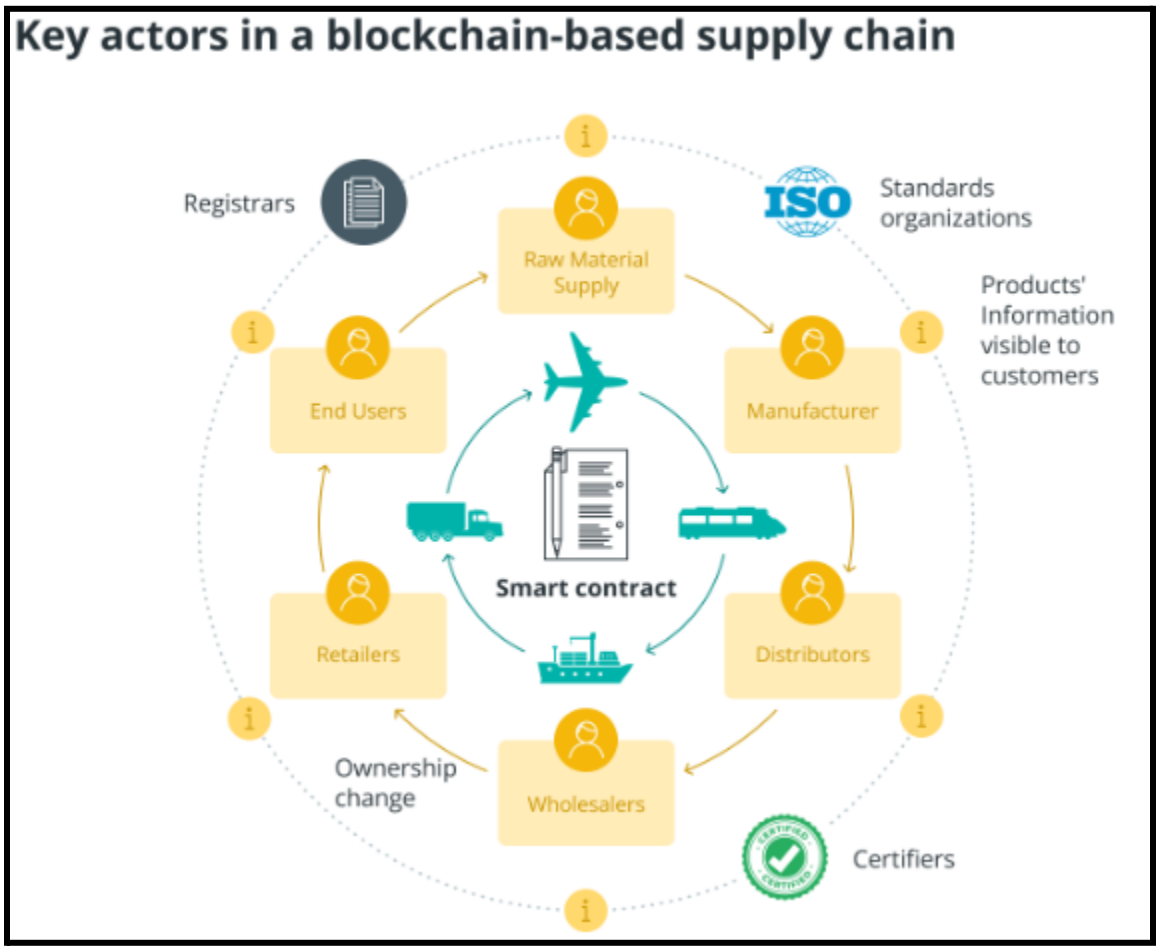
In blockchain-based supply networks, four key actors play roles, including registrars, standard organizations, certifiers, and actors:

Registrars: They provide network actors with distinct identities.

Standard organizations: These organizations develop blockchain rules and technical specifications or standards schemes, such as Fairtrade, for environmentally friendly supply chains.

Certifiers: They certify individuals for involvement in supply chain networks.

Actors: A registered auditor or certifier must certify participants or actors, such as producers, sellers and buyers, to retain the system's credibility.



How a product is "owned" or transferred by a specific actor is an intriguing feature of structure and flow management and among the benefits of blockchain in supply chain management. But does blockchain make supply chain management more transparent?

As the concerned parties are required to fulfill a smart contract condition before a product is transferred (or sold) to another actor to validate the exchange of goods or services, and the blockchain ledger is updated with transaction information after all participants have complied with their duties and processes, overall transparency across the value chain is improved.

Additionally, the nature, quantity, quality, location and ownership product dimensions are transparently specified by blockchain technology. As a result, customers can view the continuous chain of custody and transactions from the raw materials to the final sale, eliminating the requirement for a reliable central organization to administer and maintain digital supply chains.

How blockchain enhances traceability in the supply chain?

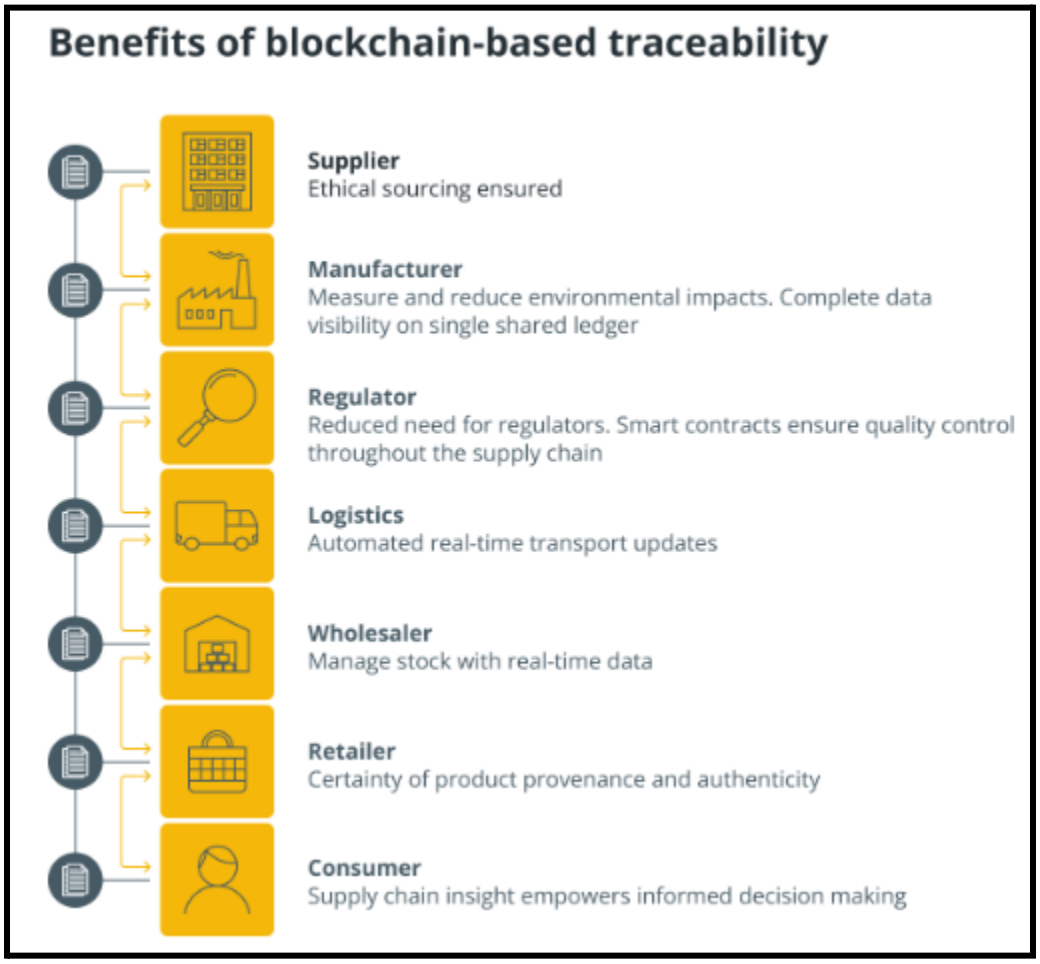
To trace the activities along the supply chain more efficiently, concerned parties can access price, date, origin, quality, certification, destination and other pertinent information using blockchain.

Traceability, as used in the supply chain sector, is the capacity to pinpoint the previous and current locations of inventory and a record of product custody. It involves tracking products as they move through a convoluted process, from raw materials to merchants and customers, after passing through many geographic zones.

Traceability is one of the significant benefits of blockchain-driven supply chain innovations. As blockchain consists of decentralized open-source ledgers recording data, which is replicable among users, transactions happen in real-time.

As a result, the blockchain can build a supply chain that is smarter and more secure since it allows for the tracking of products through a robust audit trail with almost concurrent visibility.

By connecting supply chain networks through a decentralized system, blockchain has the potential to enable frictionless movement between suppliers and manufacturers.



The future of blockchain-based supply chain:

The demand for the blockchain-based supply chain is driven by customers' need to know the specific source of their items and whether they were made according to ethical standards.

Blockchain technology use cases in supply chain management have the potential to address concerns in traditional supply chains, like removing the need to prepare burdensome paperwork. Moreover, a decentralized, immutable record of all transactions and organizations' digitization of physical assets can make it possible to track products from the manufacturing unit to the delivery destination, enabling a more transparent and visible supply chain.

However, the implementation of blockchain in the supply chain is yet to achieve mainstream adoption as high-level expertise is required to reap the benefits. Additionally, because blockchain technology is still in its infancy, it is governed by various laws in many nations, which would affect supply networks. Despite this, blockchain-based solutions will likely gradually replace conventional supply chain processes and networks; this transition won't occur all at once.

Conclusion : Hence, we have studied blockchain based supply chain case study.

FAQS:

1) How is blockchain related to the supply chain?

   Shippers and logistics providers that implement blockchain technology could be introduced to opportunities across the supply chain for greater security, efficiency, and transparency – as well as potential cost-savings.

   Every time a product changes hands, the transaction can be documented to create a permanent history of a product – from manufacture to sale. Aside from creating visibility and trust between shippers and customers (and anyone involved within the transaction), this could dramatically reduce time delays, additional costs, and human error.5 Although its potential can be applied to endless possibilities, blockchain could help improve areas including the following:

   Recording the quantity and transfer of assets, such as trailers, containers, and pallets as they move throughout the chain

   Tracking any trade-related documents, such as purchase orders, receipts, change orders, and shipment notifications

   Assigning certifications to products, for example, verifying whether a food product is organic or fair trade

   Sharing information with suppliers and vendors such as manufacturing process, assembly, and delivery for full transparency.

2) How costly is Blockchain for Supply Chain?

It depends on how you use it. Initially, we started using public blockchains back in 2016 to record hashes -essentially fingerprints of the data- that were sent from ERP systems. We did a few tests where it would not only put hashes but store more data, and it was very costly.

When we applied a common-sense approach towards using blockchains, so using it for what it's only intended to be as a layer of trust, we could significantly lower the cost.

For exchanging and storing data, blockchain is way more expensive, but you will not use it for that: it is like comparing apples and peaches. In terms of value, using blockchain -the right way, that is only as a consensus layer- is significantly surpassing the value of using centralized data.

3) How can I become a Blockchain Supply Chain expert?

Becoming an expert is way easier right now than it was a couple of years ago. There are so many resources online, you can just type "blockchain for supply chains" to have more granular.
My personal advice would be to stay curious, and not take yourself into one specific solution.
So stay open-minded because this is a fast-growing technology and something that is cool right now might be obsolete tomorrow.
It's all based on the open-source component, so if you keep your mind flexible, you can add certain components which are more applicable.

Output :

ASSIGNMENT NO: 8

TITLE: Virtual Lab - Study Cryptography and perform Digital Signature Algorithms.

PROBLEM STATEMENT : Study Cryptography and perform Digital Signature Algorithms.

PREREQUISITES: Understanding the concept of cryptography and digital signature algorithm

COURSE OBJECTIVE: Understand and explore the applications of cryptography and digital signature algorithm

COURSE OUTCOME: Understand the use of Cryptography and perform Digital Signature Algorithms.

THEORY :

Cryptography :
Cryptography is the technique of securing information and communications through use of codes so that only those persons for whom the information is intended can understand it and process it. Thus preventing unauthorized access to information. The prefix "crypt" means "hidden" and suffix "graphy" means "writing".

In Cryptography the techniques which are use to protect information are obtained from mathematical concepts and a set of rule based calculations known as algorithms to convert messages in ways that make it hard to decode it.

These algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on the internet and to protect confidential transactions such as credit card and debit card transactions.

A digital signature is a mathematical technique used to validate the authenticity and integrity of a digital document, message or software. It's the digital equivalent of a handwritten signature or stamped seal, but it offers far more inherent security. A digital signature is intended to solve the problem of tampering and impersonation in digital communications.

Digital signatures can provide evidence of origin, identity and status of electronic documents, transactions or digital messages. Signers can also use them to acknowledge informed consent. In many countries, including the U.S., digital signatures are considered legally binding in the same way as traditional handwritten document signatures.

Techniques used For Cryptography: In today's age of computers cryptography is often associated with the process where an ordinary plain text is converted to cipher text which is the text made such that intended receiver of the text can only decode it and hence this process is known as encryption. The process of conversion of cipher text to plain text is known as decryption.

\

Features Of Cryptography are as follows:

Confidentiality: Information can only be accessed by the person for whom it is intended and no other person except him can access it.

Integrity: Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.

Non-repudiation: The creator/sender of information cannot deny his intention to send information at a later stage.

Authentication: The identities of sender and receiver are confirmed. As well as the destination/origin of information is confirmed.

How do digital signatures work?

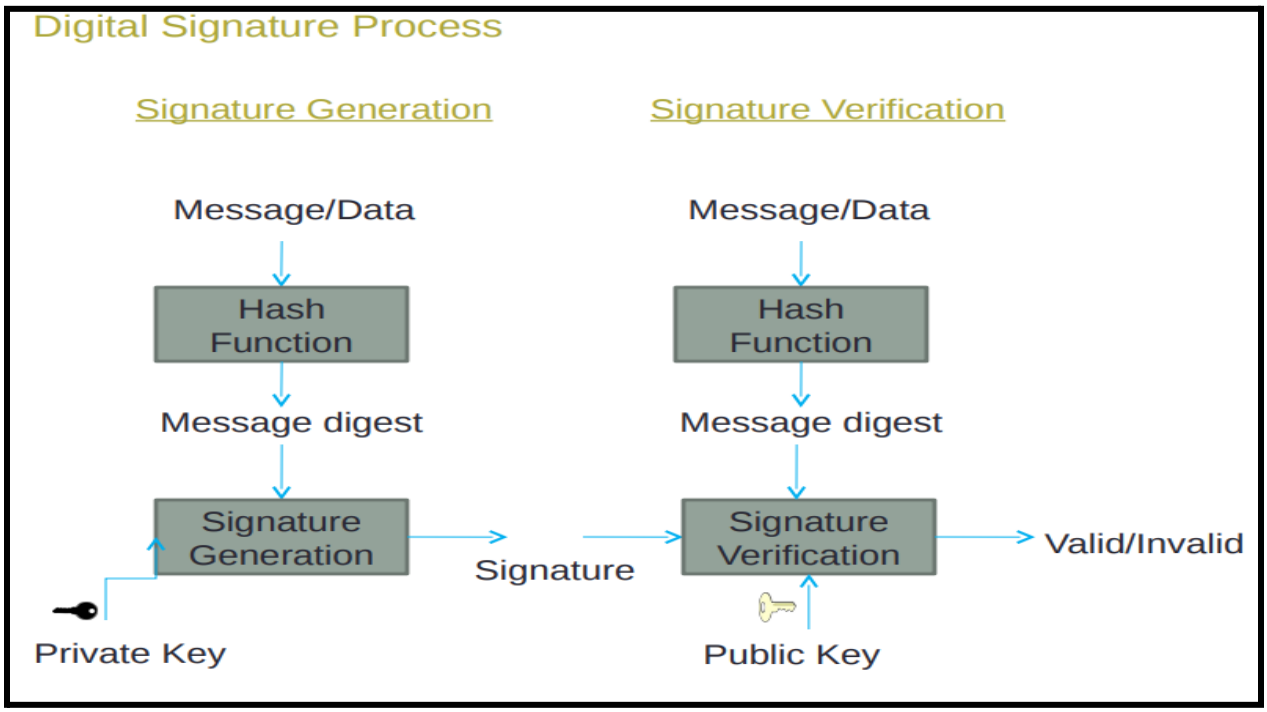
Digital signatures are based on public key cryptography, also known as *asymmetric cryptography*. Using a public key algorithm -- such as Rivest-Shamir-Adleman, or RSA -- two keys are generated, creating a mathematically linked pair of keys: one private and one public.

Digital signatures work through public key cryptography's two mutually authenticating cryptographic keys. For encryption and decryption, the person who creates the digital signature uses a private key to encrypt signature-related data. The only way to decrypt that data is with the signer's public key.

If the recipient can't open the document with the signer's public key, that indicates there's a problem with the document or the signature. This is how digital signatures are authenticated.

Digital certificates, also called public key certificates, are used to verify that the public key belongs to the issuer. Digital certificates contain the public key, information about its owner, expiration dates and the digital signature of the certificate's issuer. Digital certificates are issued by trusted third-party certificate authorities (CAs), such as DocuSign or GlobalSign, for example. The party sending the document and the person signing it must agree to use a given CA.

Digital signature technology requires all parties to trust that the person who creates the signature image has kept the private key secret. If someone else has access to the private signing key, that party could create fraudulent digital signatures in the name of the private key holder.



Digital signatures offer the following benefits:

- Security. Security capabilities are embedded in digital signatures to ensure a legal document isn't altered and signatures are legitimate. Security features include asymmetric cryptography, personal identification numbers (PINs), checksums and cyclic redundancy checks (CRCs), as well as CA and trust service provider (TSP) validation.
- Timestamping. This provides the date and time of a digital signature and is useful when timing is critical, such as for stock trades, lottery ticket issuance and legal proceedings.
- Globally accepted and legally compliant. The public key infrastructure (PKI) standard ensures vendor-generated keys are made and stored securely. With digital signatures becoming an international standard, more countries are accepting them as legally binding.
- Time savings. Digital signatures simplify the time-consuming processes of physical document signing, storage and exchange, enabling businesses to quickly access and sign documents.
- Cost savings. Organizations can go paperless and save money previously spent on the physical resources, time, personnel and office space used to manage and transport documents.

Digital Signature Algorithm (DSA)

- The DSA Algorithm or system guarantees three crucial advantages. The sender can be verified with the key. The content cannot be interfered with because the message cannot be decrypted.

- Further, the sender cannot reject the message since the signature confirms the sender. Realizing the great importance, the National Institute of Standards and Technology (NIST) globally standardized it in 1994 after the proposal in 1991. DSA became the FIPS or Federal Information Processing Standard or hallmark of digital signatures.

DSA Steps

- Using the key generation algorithm, the keys are used to sign the message.
- The digital signature algorithm provides the signature.
- The hash is used for making the message digest.
- Combining DSA and the message digest results in the digital signature.
- The digital signature accompanies the transmitted message.
- Verification algorithms help to confirm the validity, and the same hash function is used.

DSA Disadvantages

US National Standard follows the DSA that applies in private and non-private messages, but some weaknesses exist along with major advantages.

SHA stands for secure hash algorithms that have six types. They are responsible for trimming the variable length of messages to fixed parameters.

In DSA, data is not encrypted but can only be confirmed as valid. Verification follows complex procedures and requires ample time.

Conclusion : We have studied, Cryptography and perform Digital Signature Algorithms.

FAQS:

1) What are types of blockchain
   - Symmetric key cryptography.
   - Asymmetric key cryptography.
   - Hash Function

\

2) What are the significant dangers to any information or data that needs cryptography?

There are a ton of dangers indeed, and you may have no clue about that; as for headway in innovation, the converse impact of the equivalent has additionally improved everywhere.

Programmers can take information, and any delicate data after spilling can make issues for the business, an administration, monetary organization just as for an individual exclusively.

The break of private data can put the whole association in danger. In this manner, information should be secured.

3) Can We trust a document that has a digital signature?

In case there is any alteration in the document, the signature is no longer valid since the document has been modified and the signed hash does not match the current hash. On the other hand, if there is any indication that the user's private key has been compromised (among other reasons), the digital certificate can be revoked and can no longer be used for signing.

Output :