PETchain: A Blockchain based Privacy Enhancing Technology

Gummalla Ganapathi Rajesh
PG scholar, Department of MCA, DNR College, Bhimavaram, Andhra Pradesh.
B.S.Murthy
(Assistant Professor), Master of Computer Applications, DNR college, Bhimavaram, Andhra Pradesh.

Abstract: With the increasing use of smart devices and sensors, enormous amounts of data are being generated continuously. The data is commonly stored in centralized cloud platforms and consumed by different services. The data is indeed a valuable resource for many service providers who provide advanced features and utilities to their subscribers. However, user data include personal and sensitive information which can be misused in many ways. There is no way for a subscriber to confirm that their service provider is compliant with data privacy regulations. The existing privacy enhancing techniques such as anonymization and differential privacy substantially reduce data usability while ensuring privacy. Therefore, it remains essential to provide a feasible solution that allows service providers to take advantage of user data while guaranteeing their privacy. In this paper, we present PETchain: a novel privacy enhancing technology using blockchain and smartcontract. In PETchain, data is stored securely in a distributed manner and processed in a user-selected trusted execution environment. Users deploy the smartcontract that allows them to decide whether and how their data can be exploited by service providers. The feasibility and performance of PETchain are presented by implementing PETchain over a consortium Ethereum blockchain.

INDEX TERMS Blockchain, Ethereum, Privacy Enhancing Technology, Privacy Preservation, Smartcontract.

I. INTRODUCTION

The rapid development in the fields of IoT, social networks, and cloud computing has led to the increased generation, storage, and processing of personalized data. In this new era of big data, both public and private service providers are collecting large amounts of data from their users to provide them with enhanced services and features. However, most of the collected contains private and confidential information that can be easily abused. Service providers focus on providing strong authentication, integrity, and confidentiality solutions but generally under-look user's privacy while collecting, storing, and processing their personal data [1].

Once a service provider acquires the data it can easily misuse or distribute it without user consent. Thus, users must completely trust their service providers and have little or no control over their data. Furthermore, the users are unable to define and implement access control for their data, to decide who and when can access their data, and how can they process it. The General Data Protection Regulation (GDPR)1 came into act on 25 May 2018 in the European Union. The GDPR contains provisions and requirements to protect users' rights related to their data. This includes the user's right to remain informed about their data being gathered, processed, and distributed by their service provider.

It also provides the user with the right to have their data updated or even deleted forever [2]. However, users, for now, do not have any ability to check whether their service providers are GDPR compliant or not. Users have to completely trust their service providers with their data if they wish to use their services. Furthermore, if any investigation on a privacy breach is conducted, the supervisory authority does not have any reliable or auditable logs to inspect. Authorities must rely on the logs maintained by the service providers themselves.

The major contributions of this paper are as follows: 1) A novel privacy enhancing technology, "PETchain", that allows service providers to utilize user data while guaranteeing their privacy. PETchain allows users to securely store and maintain their data in IPFS. Data owners are given complete control over their data by deciding who can access, process, and utilize their data. The data is processed in a user-selected isolated secure environment, assuring the confidentiality and integrity of the data. Logs of each data access are maintained in an immutable and auditable manner in the blockchain. 2) A smartcontract that allows users to define their access control policy. The PETchain smartcontract consists of several functions. With the set_identifier function the user stores data identifiers of the data files uploaded over IPFS, whereas with the set_authorization function users authorize service providers to utilize their data. Get_identifier is the only function that can be called by the service provider to request data access. Moreover, with the destroy_smartcontract and pause_smartcontract functions the users can pause and remove their smartcontract at any time. 3) An implementation of PETchain over a consortium blockchain. We choose to use a consortium blockchain as it maintains user control and privacy while having reduced cost and high throughput when compared to the public blockchain.

Furthermore, unlike a private blockchain network, it does not consolidate power to a single party and distributes it across different organizations. We analyze the performance of PETchain on a consortium blockchain by using four performance metrics: transaction GAS cost, transactions per second, number of lost blocks, and propagation delay. Different parameters such as sealers, block-time, and block-gaslimit were analyzed to achieve the best possible results for our consortium blockchain. We choose to implement our solution using the Ethereum platform as it facilitates developers to deploy their smart contracts in a secure and simple manner. However, PETchain can be implemented over any consortium blockchain platform that supports the deployment of smart contacts.

II. LITEARTURE SURVEY

The rapid growth of the Internet of Things (IoT) has introduced significant challenges regarding the privacy and security of data collected from diverse sources. In this context, multiple technologies and regulatory frameworks have been developed to safeguard user information.

Cha et al. [1] provide an in-depth analysis of Privacy-Enhancing Technologies (PETs) tailored for IoT environments. Their study highlights the complexities of securing heterogeneous IoT systems while ensuring data privacy, especially in constrained devices. The authors examine PETs such as differential privacy, homomorphic encryption, and trusted execution environments (TEEs), evaluating their applicability and limitations. They emphasize the need for balancing privacy with usability, latency, and resource constraints within IoT networks.

Voigt and Von dem Bussche [2] present a comprehensive legal overview of the General Data Protection Regulation (GDPR) introduced by the European Union. Their practical guide outlines core principles of data protection, including consent, purpose limitation, data minimization, and user rights. GDPR compliance is particularly critical for IoT applications that handle sensitive personal data, underscoring the importance of privacy-by-design and default principles in system architecture.

Zhou and Wornell [3] introduce an efficient homomorphic encryption scheme suitable for integer vectors, enabling computation on encrypted data without compromising privacy. Their approach improves performance in practical scenarios and can be adapted to privacy-preserving analytics in IoT and cloud environments. The application of homomorphic encryption ensures that sensitive data remains confidential even during processing, addressing a major privacy concern in distributed systems.

The concept of k-anonymity, introduced by Sweeney [4], serves as a foundational privacy model by ensuring that each individual’s data cannot be distinguished from at least k-1 other individuals. This model has become a cornerstone for anonymization techniques in datasets and is particularly useful in healthcare and smart city applications, where large volumes of personal data are shared.

The DIN ISO 4997:2020-04 standard [7] introduces a novel model for implementing privacy by design using blockchain technology. This framework defines a standardized approach for processing personal data securely and transparently through blockchain, ensuring data immutability, decentralization, and user control. The standard emphasizes compliance with GDPR and proposes that blockchain systems can inherently support auditability and trust in privacy-sensitive applications.

Together, these works illustrate a multi-disciplinary approach to privacy—from technical implementations like encryption and anonymization to regulatory and standardization efforts. As IoT systems evolve, the integration of legal frameworks, advanced encryption methods, and design-oriented privacy principles will be essential in fostering secure and trustworthy environments.

III. PROPOSED SYSTEM

Now-a-days all online service providers are asking users to signup to access their services and after signup users can login and then store their personal data on service providers online servers as this service providers provide data storage or access services at any time with cheap cost. Often user data consists of sensitive information and this data will be away from their control and can be misused by service provider's internal employees or attackers.

To provide security to user's sensitive data there is plenty of algorithms are available such as Homomorpic or Anonimity but this algorithms will not satisfy all user requirements like Data Sharing control, Accountability etc. Another problem is centralized data storage at single server and if this server down then users cannot access services till server up.

To overcome from above issue author of this paper introducing PETCHAIN (privacy enhancing tool) using Blockchain which has inbuilt support for data immutability (data cannot be alter), block verification and Distributed data storage (which means Blockchain store data in all network nodes and if one down then services can be access from other working nodes).

PETCHAIN provide following services for user data privacy

1) User: An individual or organization that owns data and has the right to allow who can access and process it. This users consider as subscribers
2) Service Provider: An online public or private organization that is established to deliver various applications to their consumers. They provide services and features to their subscribers by acquiring and processing their data. This user consider as service providers
3) Distributed Storage: A peer-to-peer distributed open-source file-sharing system that allows data to be stored, maintained, and distributed in a fast and secure manner. Blockchain services which store data in multiple nodes as distributed storage
4) Trusted Executor: A trusted individual or organization that provides a secure isolated environment that allows code to be executed over some data while guaranteeing the confidentiality and integrity of the data as well as the code there loaded. Here code will be executed using Smart Contract and data will be secured using AES encryption
5) Blockchain: A peer-to-peer distributed platform that supports decentralized applications. It should supports smart contract code execution and storage over the distributed network such as Ethereum.

So above PETCHAIN modules allow users to get security and full control over their data while storing in service provider services. PET CHAIN using Blockchain to store user data and using IPFS (inter planetary file system) to store user data files). PET CHAIN Application will first encrypt user data using AES encryption and then store this file in IPFS and IPFS will returned ADDRESS of file storage and this address will be stored in Blockchain to retrieve file from IPFS.

IV. RESULT

Now-a-days all online service providers are asking users to signup to access their services and after signup users can login and then store their personal data on service providers online servers as this service providers provide data storage or access services at any time with cheap cost. Often user data consists of sensitive information and this data will be away from their control and can be misused by service provider's internal employees or attackers.

To provide security to user's sensitive data there is plenty of algorithms are available such as Homomorpic or Anonimity but this algorithms will not satisfy all user requirements like Data Sharing control, Accountability etc. Another problem is centralized data storage at single server and if this server down then users cannot access services till server up.

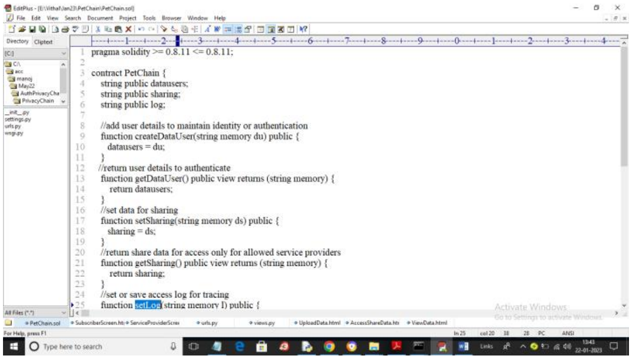
To overcome from above issue author of this paper introducing PETCHAIN (privacy enhancing tool) using Blockchain which has inbuilt support for data immutability (data cannot be alter), block verification and Distributed data storage (which means Blockchain store data in all network nodes and if one down then services can be access from other working nodes).

PETCHAIN provide following services for user data privacy

6) User: An individual or organization that owns data and has the right to allow who can access and process it. This users consider as subscribers
7) Service Provider: An online public or private organization that is established to deliver various applications to their consumers. They provide services and features to their subscribers by acquiring and processing their data. This user consider as service providers
8) Distributed Storage: A peer-to-peer distributed open-source file-sharing system that allows data to be stored, maintained, and distributed in a fast and secure manner. Blockchain services which store data in multiple nodes as distributed storage
9) Trusted Executor: A trusted individual or organization that provides a secure isolated environment that allows code to be executed over some data while guaranteeing the confidentiality and integrity of the data as well as the code there loaded. Here code will be executed using Smart Contract and data will be secured using AES encryption
10) Blockchain: A peer-to-peer distributed platform that supports decentralized applications. It should supports smart contract code execution and storage over the distributed network such as Ethereum.

So above PETCHAIN modules allow users to get security and full control over their data while storing in service provider services. PET CHAIN using Blockchain to store user data and using IPFS (inter planetary file system) to store user data files). PET CHAIN Application will first encrypt user data using AES encryption and then store this file in IPFS and IPFS will returned ADDRESS of file storage and this address will be stored in Blockchain to retrieve file from IPFS.

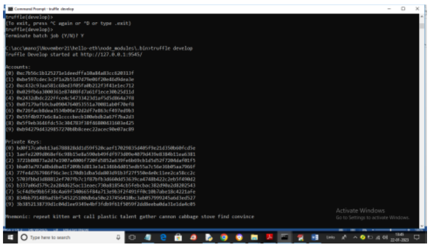
To store data in Blockchain we need to designed Smart Contract which contains functions to store and retrieve data from Blockchain. Below screen showing Smart Contract code



```
pragma solidity >= 0.8.11 <= 0.8.11;

contract PetChain {
   string public datausers;
   string public sharing;
   string public log;

   //add user details to maintain identity or authentication
   function createDataUser(string memory du) public {
     datausers = du;
   }
  //return user details to authenticate
   function getDataUser() public view returns (string memory) {
      return datausers;
   }
   //set data for sharing
   function setSharing(string memory ds) public {
     sharing = ds;
   }
   //return share data for access only for allowed service providers
   function getSharing() public view returns (string memory) {
      return sharing;
   }
   //set or save access log for tracing
   function setLog(string memory l) public {
```

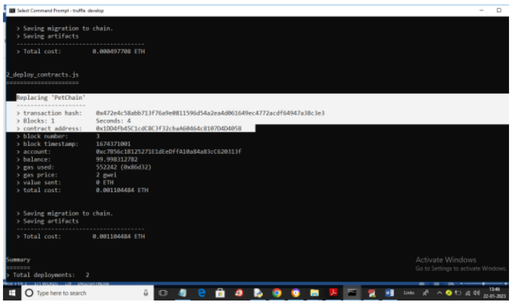
Above smart contract is designed using SOLIDITY code and it contains various functions to store and get data. Now we need to deploy above contract in Blockchain Ethereum using Truffle tool and by following below steps

1) First go inside hello-eth/node_modules/.bin' folder and then double click on 'runBlockchain.bat' file to get below screen



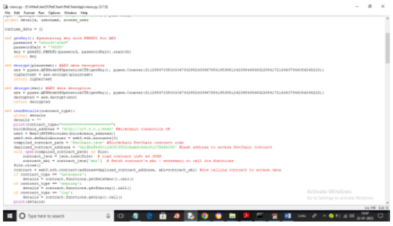
```
truffle(develop)>
(To exit, press ^C again or ^D or type .exit)
truffle(develop)>
Terminate batch job (Y/N)? Y

C:\acc\manoj\November21\hello-eth\node_modules\.bin>truffle develop
Truffle Develop started at http://127.0.0.1:9545/

Accounts:
(0) 0xc7b56c1b125271e1deedffa10a84a83cc620313f
(1) 0xdbe597cdec3c2f1a2b51d7d79e06f20e46d9dea3e
(2) 0xc432c93aa581c68ed3f05fa0b212f3f41e1ec712
(3) 0x029fb6a3000361e87408fd7a61f1ece30b25d11d
(4) 0x2432dbdc222ffce4c54733423d1af5d5d864a7f8
(5) 0x07179afb9cba0904764053551a70081ab0f70ef8
(6) 0x726facb8dea3534b06e72d2df7e863cf497ed9b3
(7) 0x55f4b977e6c8a1ccccbecb100ebdb2a67f7ba2d3
(8) 0x5f9eb3646fdc53c304783f38fd48800431603e425
(9) 0xd94279d432985727 0b8b8ceec22acec90e07ac89

Private Keys:
(0) bd0f17ca0eb13a6788828dd1d59f520caef17029835d405f9e21d350b60fcd5e
(1) 1aefe2209d068ef6c98b15e8a590eb49fdf973d09e4079d439e8384b11ea6381
(2) 3721b80873a2d7e1907a4000f720fd5852a639fe6b09cb1d5d52f7204daf01f5
(3) bba03a797a8bddba41f209b3d813e3a1346b8d015edb55a7c56e38b05aa7966f
(4) 77fe4d767986f96c3ec170db1dba5da803d91b3f27f550ede0c11ee2ca58cc2c
(5) 5703fbbd3d88812ef707fb7c1f87bfb3d660dd53639ca4748b422c2eb5f490d2
(6) b337a06d579c2a284d625ac11eaec738a81854cb5febcbac382d90a2d8202543
(7) 5cf4d9e9bb5f38c4a69f340665f84a713e9b3f2f491ff0c10b7abe18c4221afe
(8) 834bb791489ad5bf545225100db6a50e237456410bc3ab057999245a6d3ed527
(9) 3b3852138739d1c04d1ee9349e4bf3fdb9f61f5059f2dd8eeba0da31e1da4c05

Mnemonic: repeat kitten art call plastic talent gather cannon cabbage stove find convince
```

In above screen Blockchain started and generate some default account addresses and private keys and now type command as 'migrate' and press enter key to deploy contract and get below output

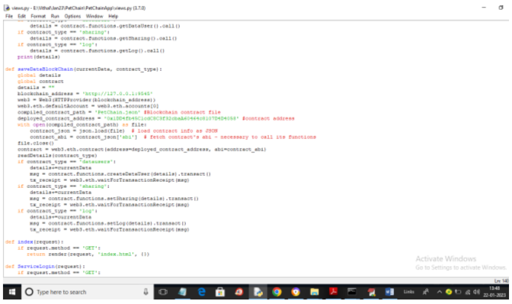


```
  > Saving migration to chain.
  > Saving artifacts
  -------------------------------------
  > Total cost:          0.000497708 ETH


2_deploy_contracts.js
=====================

  Replacing 'PetChain'
  --------------------
  > transaction hash:    0x472e4c58abb713f76a9e0811596d54a2ea4d061649ec4772acdf64947a38c3e3
  > Blocks: 1            Seconds: 4
  > contract address:    0x1DD4fb45C1cdC8C3f32cbaA60464c8107D4D4058
  > block number:        3
  > block timestamp:     1674371001
  > account:             0xc7B56c1B125271E1dEeDffA10a84a83cC620313f
  > balance:             99.998312782
  > gas used:            552242 (0x86d32)
  > gas price:           2 gwei
  > value sent:          0 ETH
  > total cost:          0.001104484 ETH


  > Saving migration to chain.
  > Saving artifacts
  -------------------------------------
  > Total cost:          0.001104484 ETH


Summary
=======
> Total deployments:   2
```

In above screen we can see PETCHAIN contract deployed and we got contract address also and this address we need to specify in Python program to store and retrieve or access data. In below screen showing Python program to access Blockchain contract


views.py - E:\Vithal\Jan23\PetChain\PetChainApp\views.py (3.7.0)
File
Edit
Format
Run
Options
Window
Help
Activate Windows
Go to Settings to activate Windows.
Type here to search

```
global details, username, access_user

runtime_data = []

def getKey(): #generating key with PBKDF2 for AES
    password = "[illegible]"
    passwordSalt = '[illegible]'
    key = pbkdf2.PBKDF2(password, passwordSalt).read(32)
    return key

def encrypt(plaintext): #AES data encryption
    aes = pyaes.AESModeOfOperationCTR(getKey(), pyaes.Counter([illegible]))
    ciphertext = aes.encrypt(plaintext)
    return ciphertext

def decrypt(enc): #AES data decryption
    aes = pyaes.AESModeOfOperationCTR(getKey(), pyaes.Counter([illegible]))
    decrypted = aes.decrypt(enc)
    return decrypted

def readDetails(contract_type):
    global details
    details = ""
    print(contract_type+"======================")
    blockchain_address = 'http://127.0.0.1:9545' #Blockchain connection IP
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'PetChain.json' #Blockchain PetChain contract code
    deployed_contract_address = '[illegible]' #hash address to access PetChain contract
    with open(compiled_contract_path) as file:
        contract_json = json.load(file)  # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi) #now calling contract to access data
    if contract_type == 'datausers':
        details = contract.functions.getDataUser().call()
    if contract_type == 'sharing':
        details = contract.functions.getSharing().call()
    if contract_type == 'log':
        details = contract.functions.getLog().call()
    print(details)
```


views.py - E:\Vithal\Jan23\PetChain\PetChainApp\views.py (3.7.0)
File
Edit
Format
Run
Options
Window
Help
Activate Windows
Go to Settings to activate Windows.
Type here to search
13:48
22-01-2023

```
        details = contract.functions.getDataUser().call()
    if contract_type == 'sharing':
        details = contract.functions.getSharing().call()
    if contract_type == 'log':
        details = contract.functions.getLog().call()
    print(details)

def saveDataBlockChain(currentData, contract_type):
    global details
    global contract
    details = ""
    blockchain_address = 'http://127.0.0.1:9545'
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'PetChain.json' #Blockchain contract file
    deployed_contract_address = '0x1DD4fb45C1cdC8C3f32cbaA60464c8107D4D4058' #contract address
    with open(compiled_contract_path) as file:
        contract_json = json.load(file)  # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
    readDetails(contract_type)
    if contract_type == 'datausers':
        details+=currentData
        msg = contract.functions.createDataUser(details).transact()
        tx_receipt = web3.eth.waitForTransactionReceipt(msg)
    if contract_type == 'sharing':
        details+=currentData
        msg = contract.functions.setSharing(details).transact()
        tx_receipt = web3.eth.waitForTransactionReceipt(msg)
    if contract_type == 'log':
        details+=currentData
        msg = contract.functions.setLog(details).transact()
        tx_receipt = web3.eth.waitForTransactionReceipt(msg)

def index(request):
    if request.method == 'GET':
       return render(request, 'index.html', {})

def ServiceLogin(request):
    if request.method == 'GET':
```

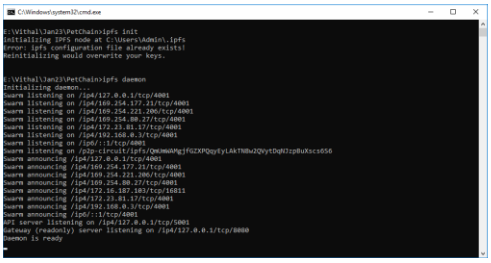
In above two screens read red colour comments to know how to call Blockchain Smart Contract. Now contract deployed and let it run.

To implement this project we have designed two modules

1) Service Provider: this user can signup and login and then can view and download all shared file details uploaded by subscribers. This service providers will provide space to store user or subscriber data
2) Subscriber: subscriber can signup and login and then can upload data by encrypting and then give permission to allow or not allow for service providers. User can view and download all his past saved data and can view log of entire access done by Service providers.
3) In this project we allow both users to login to application using 2FA technique where user has to enter username, password and upload graphical image for authentication

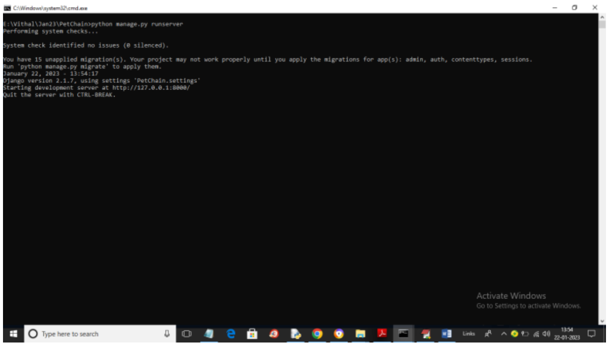
SCREEN SHOTS

First double click on 'Start_IPFS.bat' file to start IPFS server and get below screen



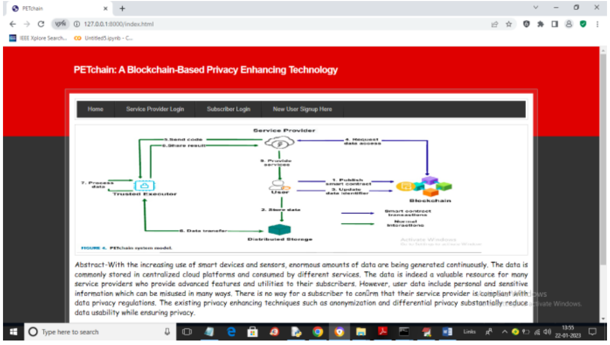
```
E:\Vithal\Jan23\PetChain>ipfs init
initializing IPFS node at C:\Users\Admin\.ipfs
Error: ipfs configuration file already exists!
Reinitializing would overwrite your keys.


E:\Vithal\Jan23\PetChain>ipfs daemon
Initializing daemon...
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/169.254.177.21/tcp/4001
Swarm listening on /ip4/169.254.221.206/tcp/4001
Swarm listening on /ip4/169.254.80.27/tcp/4001
Swarm listening on /ip4/172.23.81.17/tcp/4001
Swarm listening on /ip4/192.168.0.3/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit/ipfs/QmUmWAMgjfGZXPQqyEyLAkTN8w2QVytDqNJzpBuXscs6S6
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/169.254.177.21/tcp/4001
Swarm announcing /ip4/169.254.221.206/tcp/4001
Swarm announcing /ip4/169.254.80.27/tcp/4001
Swarm announcing /ip4/172.16.187.103/tcp/16811
Swarm announcing /ip4/172.23.81.17/tcp/4001
Swarm announcing /ip4/192.168.0.3/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

In above screen IPFS server started and now let it run and now double click on 'runServer.bat' file to start python web server and get below screen


C:\Windows\system32\cmd.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
13:54
22-01-2023

```
E:\Vithal\Jan23\PetChain>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 22, 2023 - 13:54:17
Django version 2.1.7, using settings 'PetChain.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

In above screen python web server started and now open browser and enter URL as 'http://127.0.0.1:8000/index.html' and press enter key to get below page

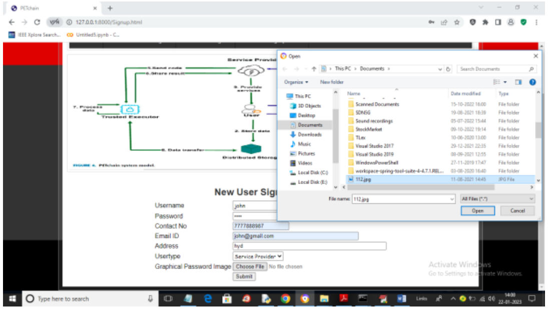


PETchain: A Blockchain-Based Privacy Enhancing Technology

FIGURE 4. PETchain system model.

Abstract-With the increasing use of smart devices and sensors, enormous amounts of data are being generated continuously. The data is commonly stored in centralized cloud platforms and consumed by different services. The data is indeed a valuable resource for many service providers who provide advanced features and utilities to their subscribers. However, user data include personal and sensitive information which can be misused in many ways. There is no way for a subscriber to con rm that their service provider is compliant with data privacy regulations. The existing privacy enhancing techniques such as anonymization and differential privacy substantially reduce data usability while ensuring privacy.

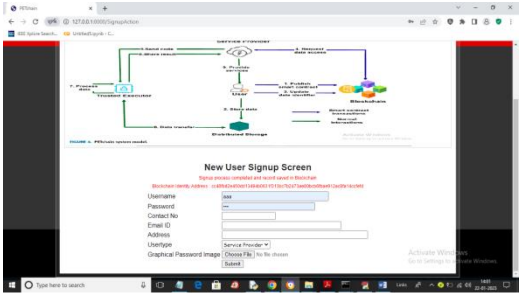
In above screen user can click on ‘New User Signup Here’ link to get below signup page


New User Signup Screen

Username: john
Password: ••••
Contact No: 7777889967
Email ID: john@gmail.com
Address: hyd
Usertype: Service Provider
Graphical Password Image: Choose File No file chosen
Submit

File name: 112.jpg
Open Cancel

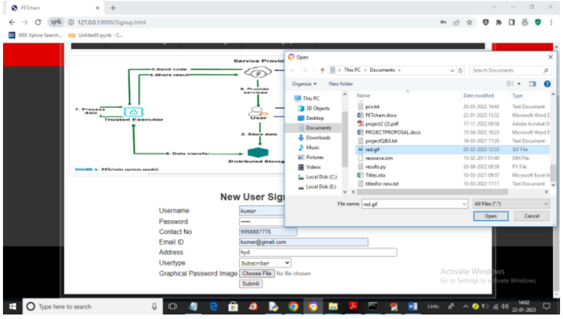
In above screen I am signing up service provider as ‘John’ and then uploading 2FA image graphical password and then press ‘Open’ and ‘Submit’ button to store data in Blockchain and get below output


New User Signup Screen

Signup process completed and record saved in Blockchain

Username: aaa
Password: •••
Contact No
Email ID
Address
Usertype: Service Provider
Graphical Password Image: Choose File No file chosen
Submit

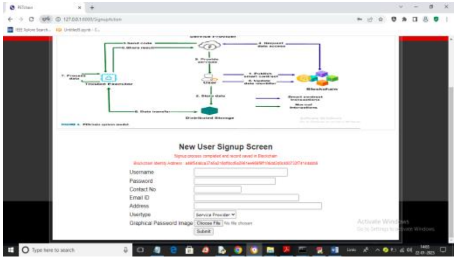
In above screen service provider signup done and in red colour text we can see the storage hash code details. Similarly you can signup subscriber like below screen


New User Signup Screen

Username: kumar
Password: •••
Contact No: 9999887776
Email ID: kumar@gmail.com
Address: hyd
Usertype: Subscriber
Graphical Password Image: Choose File No file chosen
Submit

File name: red.gif
Open Cancel

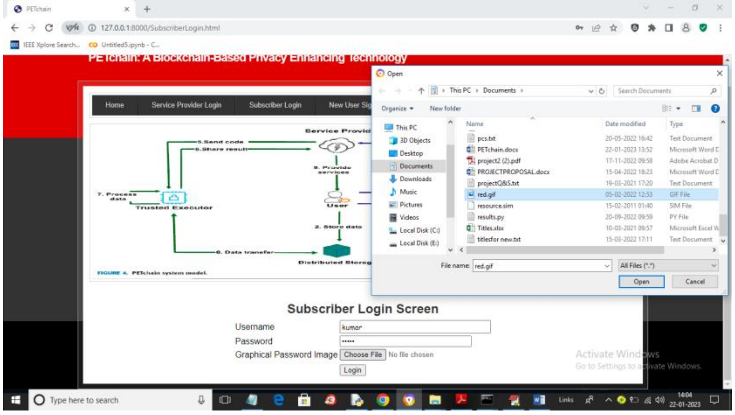
In above screen I am adding subscriber by selecting ‘User Type’ as subscriber from drop down box and then uploading Graphical password image and now click ‘Open and Submit’ button to complete subscriber signup and get below output


New User Signup Screen

Signup process completed and record saved in Blockchain

Username
Password
Contact No
Email ID
Address
Usertype: Service Provider
Graphical Password Image: Choose File No file chosen
Submit

In above screen subscriber added and now click on ‘Subscriber Login’ to get below page

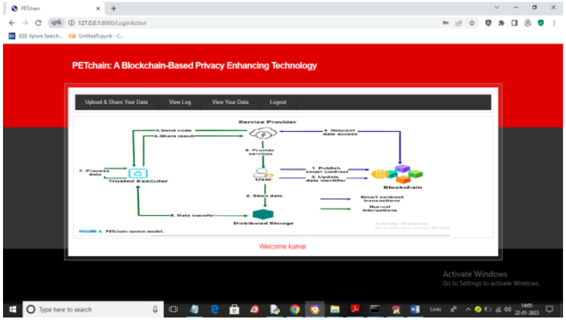

PETchain: A Blockchain-Based Privacy Enhancing Technology

Subscriber Login Screen

Username: kumar
Password: •••••
Graphical Password Image: Choose File No file chosen
Login

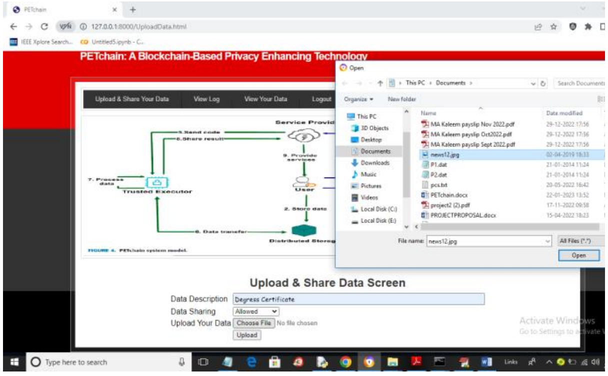
File name: red.gif
Open Cancel

In above screen subscriber is login with Password and image password and then click ‘Open and Login’ button to get below page


PETchain: A Blockchain-Based Privacy Enhancing Technology

Welcome kumar

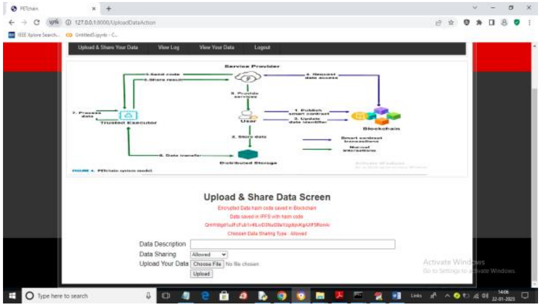
In above screen subscriber login successful and he can upload and view data and now click on ‘Upload & Share Data’ link to get below page



Upload & Share Data Screen

Data Description: Degress Certificate
Data Sharing: Allowed
Upload Your Data: Choose File No file chosen
Upload

In above screen user entering data description and then select share option as 'Allowed or Not Allowed' and then upload data and then click on 'Open and Upload' button to save data in Blockchain

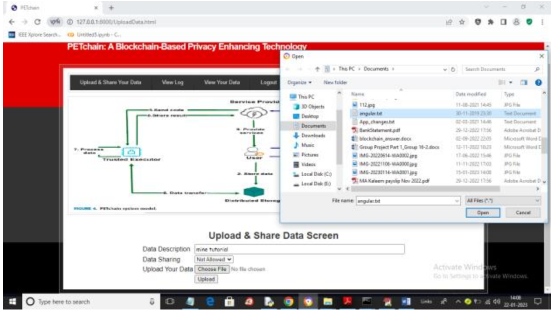


Upload & Share Data Screen

Encrypted Data hash code saved in Blockchain
Data saved in IPFS with hash code
QmWdgd1uJFcFub1x4tLwD3NuG8aYzgdqivKgAXF5RonAi
Chooosen Data Sharing Type : Allowed

Data Description:
Data Sharing: Allowed
Upload Your Data: Choose File No file chosen
Upload

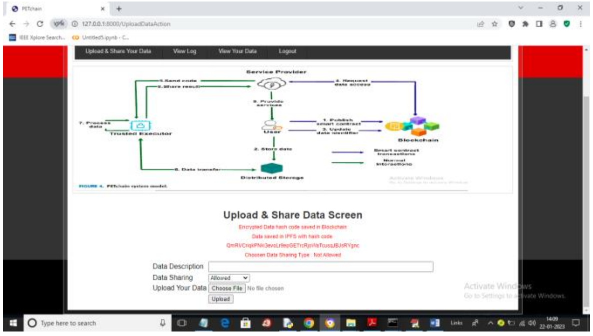
In above screen in red colour text we can see data storage details and similarly you can upload and test other files



Upload & Share Data Screen

Data Description: mine tutorial
Data Sharing: Not Allowed
Upload Your Data: Choose File No file chosen
Upload

In above screen I am uploading another data with share option as 'Not Allowed' so Blockchain will not allow Service provider to view or access this data and click button to get below output

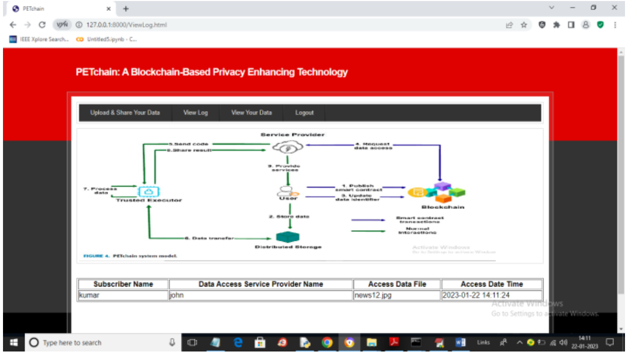


Upload & Share Data Screen

Encrypted Data hash code saved in Blockchain
Data saved in IPFS with hash code
QmRVCriqkPNki3evoLr9epGETrcRjoWaTcusqJBJsRYgnc
Chooosen Data Sharing Type : Not Allowed

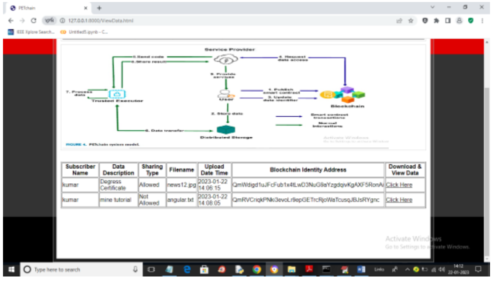
Data Description:
Data Sharing: Allowed
Upload Your Data: Choose File No file chosen
Upload

In above screen data is saved in Blockchain and now click on 'View Log' link to view details of all users who access this files



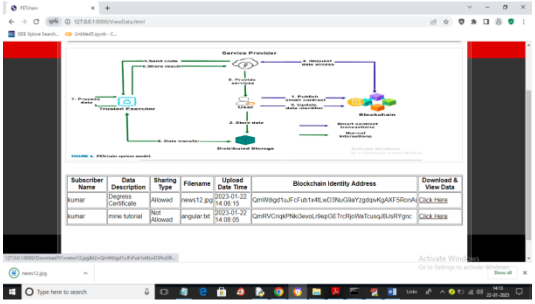
Subscriber Name	Data Access Service Provider Name	Access Data File	Access Date Time
kumar	john	news12.jpg	2023-01-22 14:11:24

In above log screen subscriber can view names of all service providers who access which file at which point of time. Now click on 'View Your Data' link to view all files uploaded by subscriber



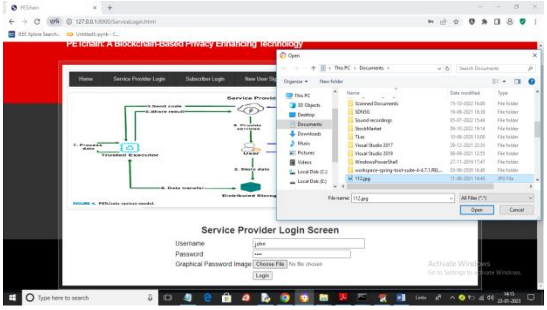
Subscriber Name	Data Description	Sharing Type	Filename	Upload Date Time	Blockchain Identity Address	Download & View Data
kumar	Degress Certificate	Allowed	news12.jpg	2023-01-22 14:06:15	QmWdgd1uJFcFub1x4tLwD3NuG8aYzgdqivKgAXF5RonAi	Click Here
kumar	mine tutorial	Not Allowed	angular.txt	2023-01-22 14:08:05	QmRVCriqkPNki3evoLr9epGETrcRjoWaTcusqJBJsRYgnc	Click Here

In above screen subscriber can view all files with sharing status and now click on 'Click Here' link to download that file in decrypted format



Subscriber Name	Data Description	Sharing Type	Filename	Upload Date Time	Blockchain Identity Address	Download & View Data
kumar	Degress Certificate	Allowed	news12.jpg	2023-01-22 14:06:15	QmWdgd1uJFcFub1x4tLwD3NuG8aYzgdqivKgAXF5RonAi	Click Here
kumar	mine tutorial	Not Allowed	angular.txt	2023-01-22 14:08:05	QmRVCriqkPNki3evoLr9epGETrcRjoWaTcusqJBJsRYgnc	Click Here

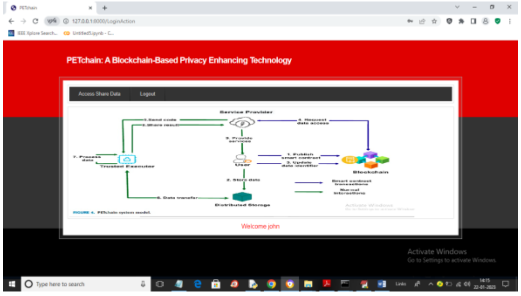
In above screen in browser status bar we can see file is downloading and similarly you view and download past uploaded file. Now logout and login as 'Service Provider' to access share files



Service Provider Login Screen

Username: john
Password: ••••
Graphical Password Image: Choose File No file chosen
Login

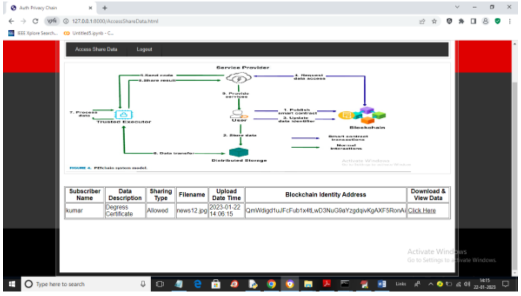
In above screen service provider is login and after login will get below page



PETchain: A Blockchain-Based Privacy Enhancing Technology

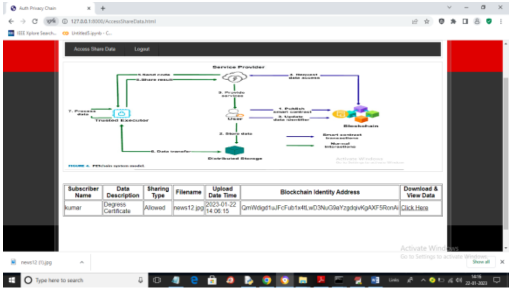
Welcome john

In above screen service provider can click on 'Access Share Data' link to get list of shared files



Subscriber Name	Data Description	Sharing Type	Filename	Upload Date Time	Blockchain Identity Address	Download & View Data
kumar	Degress Certificate	Allowed	news12.jpg	2023-01-22 14:06:15	QmWdigd1uJFcFub1x4tLwD3NuG9aYzgdqivKgAXF5RonA	Click Here

In above screen we can see subscriber gave permission to access only one file so Blockchain showing that file only for service provider to access and at any time service provider can click on 'Click Here' link to download that file



Subscriber Name	Data Description	Sharing Type	Filename	Upload Date Time	Blockchain Identity Address	Download & View Data
kumar	Degress Certificate	Allowed	news12.jpg	2023-01-22 14:06:15	QmWdigd1uJFcFub1x4tLwD3NuG9aYzgdqivKgAXF5RonA	Click Here

In above screen in browser status bar we can see file is downloading for service provider

CONCLUSION

In this paper, we proposed a novel privacy enhancing technology based on blockchain to manage and exploit user data. The proposed technique aims to address user privacy holistically. In our approach, users can define their access control policy by deploying their smartcontract. Users upload encrypted data to IPFS and store its hash to their smartcontract. The authorized service providers can access the hash but can only decrypt and process the data in an isolated execution environment. This allows service providers to process user data without acquiring nor misusing the data. To the best of our knowledge, this is the first-ever implementation of PET using blockchain technology that holistically addresses privacy. In our approach, the users select a trusted execution environment where their data can be processed in a privacy enabled manner. Moreover, they can store their data independently in a distributed manner using IPFS. The proposed solution has been implemented on a consortium blockchain due to its privacy, cost, and performance advantages over the public blockchain. The performance of PETchain is analyzed using the Ethereum platform. It was observed that a high TPS can be achieved by having low block-time and a high gas-limit. However, lower block-time was observed to have a high amount of lost blocks, decreasing the performance of the blockchain. Therefore, a bock-time of 10 is proposed that allows a TPS of around 60. Moreover, a lower number of sealers is recommended, to reduce propagation delay in the network. For our future work, we aim to analyze and improve PETchain by checking its compatibility with GDPR.

REFERENCES

1. [1] S. Cha, T. Hsu, Y. Xiang, and K. Yeh, “Privacy enhancing technologies in the internet of things: Perspectives and challenges,” IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2159–2187, 2019.

2. [2] P. Voigt and A. Von dem Bussche, “The eu general data protection regulation (gdpr),” A Practical Guide, 1st Ed., Cham: Springer International Publishing, 2017.

3. [3] H. Zhou and G. Wornell, “Efficient homomorphic encryption on integer vectors and its applications,” in 2014 Information Theory and Applications Workshop (ITA), 2014, pp. 1–9.

4. [4] L. Sweeney, “K-anonymity: A model for protecting privacy,” Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol. 10, no. 5, p. 557–570, Oct. 2002. [Online]. Available: https://doi.org/10.1142/S0218488502001648

5. [5] H. Zhou and G. Wornell, “Efficient homomorphic encryption on integer vectors and its applications,” in 2014 Information Theory and Applications Workshop (ITA), 2014, pp. 1–9.

6. [6] S. Cha, T. Hsu, Y. Xiang, and K. Yeh, “Privacy enhancing technologies in the internet of things: Perspectives and challenges,” IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2159–2187, 2019.

7. [7] “DIN ISO 4997:2020-04, Privacy by Blockchain Design: A standardised model for processing personal data using blockchain technology.” [Online]. Available: https://www.beuth.de/de/technische-regel/din-spec4997/321277504