PPM: A Provenance-Provided Data Sharing Model for Open Banking via Blockchain

Conference Paper · February 2020

DOI: 10.1145/3373017.3373022

Qin Wang
CSIRO Data61
25 PUBLICATIONS **179** CITATIONS

SEE PROFILE

Sheng Wen
Deakin University
104 PUBLICATIONS **2,615** CITATIONS

SEE PROFILE

PPM: A Provenance-Provided Data Sharing Model for Open Banking via Blockchain

Zhiyu Xu
Swinburne University of Technology
zhiyuxu@swin.edu.au

Qin Wang
Swinburne University of Technology
qinwang@swin.edu.au

Ziyuan Wang
Swinburne University of Technology
ziyuanwang@swin.edu.au

Donghai Liu
Swinburne University of Technology
donghailiu@swin.edu.au

Sheng Wen
Swinburne University of Technology
swen@swin.edu.au

Rob Hanson
Csiro, Data61
Rob.Hanson@data61.csiro.au

ABSTRACT

Open banking becomes more and more prevailing in Australia in recent years. It aims to make the users' personal financial data mutually transfer and exchange across different banks in a secure way. The sensitive data in a financial area requires higher authentication and provenance for participants. In this paper, we propose a provenance-provided data sharing model (PPM) via blockchain to meet the requirements of open banking. The model employs the programmable smart contracts as the middle witness between users and third-party services, and provides the modifications on data layer (data content, transaction structure), smart contract layer (ACL, logic), and application layer (customized APIs). Based on that, our PPM model possesses the properties of transparent authentication, privacy-provided control, and auditable provenance. The analyses and discussion show that our model is a secure and achievable system in the face of open banking.

KEYWORDS

Open Banking, Blockchain, Access Control, Provenance, Data Sharing

ACM Reference Format:
Zhiyu Xu, Qin Wang, Ziyuan Wang, Donghai Liu, Sheng Wen, and Rob Hanson. 2020. PPM: A Provenance-Provided Data Sharing Model for Open Banking via Blockchain. In *Proceedings of the Australasian Computer Science Week Multiconference (ACSW 2020), February 4–6, 2020, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373017.3373022

1 INTRODUCTION

Open banking, as an innovative idea from the UK government, has become a hot topic in Australia in recent years. Traditional banks in the worlds independently run their services, with a loose connection to other companies. One bank unusually holds some specific customers and users, which makes it hard to get the data from other customers at alt-banks. The obstacle of information sharing limits many services, such as money transferring across different banks. Therefore, to make large banks, huge companies and financial services providers easier to offer new products, open banking becomes more and more prevailing in Australia.

Open banking is a system that enables users, including small businesses securely share the data across different banks. It aims at providing a platform that can make the personal financial data mutually communicate and exchange securely. More specifically, for third-party servers providers, employing open banking rules and standards, the companies will be able to develop new online and mobile applications. These applications would, potentially, quickly to get the grant of the data from different sources, which benefits the design and selling of their products. Moreover, the transparency of their behaviours will significantly improve the credit of the bank. For customers and users, open banking provides super-fast payment methods and innovative products for customers to help them make a better financial plan.

There are several difficulties in designing an open banking system in current times. Firstly, mutual authentication is hard to be transparently managed. As the essential and critical feature in open banking, the access right of users' data should be carefully considered, and the invocation of the sensitive data should be recorded at each time. Secondly, The privacy provided service enables users to control and share personal data by customizing the Access Control List (ACL). The list, on the one hand, contains the registration of entities, including both users and Third-Party Service (TPS), and on the other hand contains the actions of various activities such as debit, credit, mortgage and so on. The control lists lay the foundation of authentication and management. Thirdly, provenance and regulation are necessary for accountability. Data provenance makes users able to audit and trace the resource in case of any misbehaviour or attacks. Lack of provenance, large universal banks will have a considerable advantage by imitating most innovations at a low cost. Therefore, it is crucial to record the history of data request for security and accountability.

Blockchain, with the properties of unforgeability, tampering resistance, transparency and verifiability, perfectly meet the requirements of an open banking system. Originate from Bitcoin [11] and after a long-time development, blockchain has evolved into a smart-contract supported system [18]. The smart-contract platform enables the predefined laws and rules programmable, which brings significant influence in the finance area. The design of blockchain ensures that no single entity can modify, delete, or even append any record to the ledger without consensus from other network participants, which provides the unforgeability and tampering resistance

ACSW 2020, February 4–6, 2020, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-7697-6/20/02…$15.00
https://doi.org/10.1145/3373017.3373022

of data. The distributed system also guarantees transparency and verifiability due to the public readable ledgers.

Based on the requirements of open banking and properties of blockchain [6] [2], we design a Provenance-Provided Data Sharing Model (PPM) for open banking system via blockchain technology. By factoring the processes of data sharing, decoupling the architecture of blockchain and redesigning the modules and functions, we construct the PPM model compatible with completeness and robustness. Our PPM model achieves the following properties:

- *Transparent authentication:* The authentication of our model is transparent and public on a distributed ledger. The transparency means users and third-party services can freely obtain the records on who authenticates, whom the actions authenticate to, what kind of authenticated actions, etc.
- *Privacy-Provided Control:* The customized access control enables users to control and share personal data with other parties with their willingness. The public key encryption system protects the authentication of actions' control with the users' identities.
- *Auditable Provenance:* The authentication is publicly recorded on the chain, where both user and the third party can get the whole history of their actions. Whenever the powerful authority does evil, the activity will be traced with accountability.

Our PPM model is presented in detail at the following sections, and the comprehensive analyses and discussions are also provided. PPM model, designed for the open banking system, will also give insight for other applications in different scenarios.

2 RELATED WORK

In the open banking system, authentication for actions and privacy of users attracts lots of concerns since the leakage of sensitive data leads to dramatic financial loss and property damage. To address the critical problems, several types of research about data sharing focus on transparent authentication, privacy provided, and provenance has been proposed in recent years. These frameworks utilize the immutability, autonomy, and anonymity properties of blockchain to solve the issues in the traditional areas.

Transparent Authentication: Authentication service is an essential feature in the distributed system. Like the front door, authentication to access the websites provides a secure and quick way to connect the users with service providers. Authentication, usually in the form of Certificate Authority (CA), is managed by a powerful third party, which causes single-point failure and inevitable distrust. Therefore, blockchain-based PKI system emerges to overcome the previous issues. Bo [12] proposed a distributed certificate scheme to distribute the centralized CA. The public data is recorded in the nodes, and decentralized Certificate libraries are realize through the modified Merkle trees. Matsumoto proposed the Instant Karma PKI (IKP) model [10] to record the traditional CA's behaviour on the blockchain. The recording method can monitor CA and detect malicious actions made by the powerful authority. Gan's approach [5] builds a private blockchain to store and manage IoT devices' latest public keys. Users can update or modify public keys through sending transactions. Other proposed models, such as [1][4][8][14], also solve authentication issues to some extent.

Privacy-Provided Control: Several issues still exist in traditional data privacy services, such as hiding users' identity [3], sealing transaction amounts [16], data control [21] [20], and access control list maintenance [15]. Our PPM focuses on the customizing consumers' Access Control List (ACL) to realize fine-grained management. The customizing the ACL enables users to control and share personal data with other parties with their own willingness. Zyskind [21] proposed a new system which combines and off-chain storage though defining T_{access} and T_{data} in the system. T_{access} focuses on users' permission allocation and data management. T_{data} is designed for data storage and sharing. This model achieves access control by sending different transactions. Xia proposed BBDS model [19] to enhance the privacy aspect for medical records in the remote servers. It modifies the transaction and block header to meet requirements for medical records. Moreover, BBDS implements access control by using identity-based authentication.

Auditable Provenance: Data provenance provides users or institutions with the ability to audit and trace the resource. Provenance record the history of the data request, which is necessary for users and institutions. ProvChain [9] is a provenance recording system based on blockchain. It treated blockchain as an online distributed database to store data. Those data is validated by provenance auditor offline. Due to the immutability of blockchain, data cannot be erased, which guarantees the integrity of data. DataProv [13] combine provenance data and smart contract together to provide secure and reliable service for sensitive cloud data. When users send a data request to nodes, the smart contract will automatically send changes to blockchain network. Finally, modifications to data will be approved or rejected by nodes. Many applications employ the provenance-provided blockchain to address the real-life problem [7][17].

3 PROBLEM STATEMENT

The Facebook-Cambridge Analytica scandal in 2018 arouses the data privacy awareness of the public. People have paid plenty of attention to their data. They have worried about if the personal data has been collected without their authentication. Currently, two advanced approaches are solving data-sharing issues in different scenarios.

In Gan's paper [5], they proposed a structure about using PKI specified in the Internet of Things(IoT). This system is an authentication system based on key management. Traditionally, some third-party institutions maintain CA, which may exist trust issues. Gan's approach builds a private blockchain to store and manage all nodes' latest public keys. Device Manufacturer Validator(DVMs) is the entity that connects CCA and IoT devices. The IoT Manufacturers maintain DVMs, and all IoT devices are connected. In the beginning, a new DVM requests the CCA to authenticate and joins the blockchain network. When passing the authentication, DVM's public key, hash address, and CCA's signature will be packaged and broadcasted to the blockchain network. Similarly, new IoT devices can register into the network by creating transactions in the same way. Moreover, DVMs can update or invoke IoT devices' public keys by sending transactions.

Zyskind [21] proposed a new system which combines and off-chain storage. This data management service has a good performance on privacy-preserving. There are three roles in this system, which are mobile users, service providers and blockchain nodes. Users are keen on using the service or obtaining applications. Service providers act as middlemen to request personal data and processing data according to users requirements. Blockchain nodes are the institutions which maintain the blockchain network and store private key-value data locally. There are two kinds of transactions in this model, which are T_{access} and T_{data}. T_{access} focuses on users' permission allocation and data management. T_{data} is designed for data storage and sharing. Users and service providers can overwrite a new set of permissions with the correct signature in T_{access}. The T_{data} provide encrypted data for those authorized users and service providers. Hence, unauthorized users have no chance to decrypt the data package. Zyskind's methods provide new ideas on data privacy. Through applying blockchain on the traditional data-sharing model, the privacy issues can be solved to some extent.

Both works mentioned above are focusing on different aspects. Gan's approach discusses how to enhance IoT devices' authentication and verification in the blockchain. Zyskind's work introduces a privacy-oriented model that implements two different transactions to achieve permission allocation and data storage. As a result, we propose the PPM model with the provenance, data verification, and user privacy in open banking service, respectively. Initially, there are three points we need to discuss:

- **Business Model:** For the commercial banks, open banking service will break their traditional profit model. Data transparency enables bank data to public access. Although they can earn profit from the new model, it is hard to define the incentive regulation that meets all banks requirements.
- **Data Security:** Traditional bank system is a closed system, and users sensitive data circulate among banks internally. With the popularization of an open bank, third-party institutions can visit some users private data for commercial purpose. Only if the open banking service can promise the privacy and security of users data, it can be widely accepted.
- **Technical Risk:** Due to the data is sharing between various institutions, more entities will own the duplicate. In this case, sensitive information is more likely to be hacked or visited without authorization. Also, frequent data access will cause data interception.

3.1 System Model

Our model have been proposed to solve the sensitive data sharing in the open banking system. There are three entities in our framework: user, third-party service, and smart contract, as presented in Figure 1. The illustrations and the behavior of three components are stated as follow:

- **User:** A user is the entity who owns bank account and interest in sharing personal data with authenticated institutions. Users public keys are submitted to the blockchain ledger. Any update of the public key is sending through transactions.
  User end behavior:

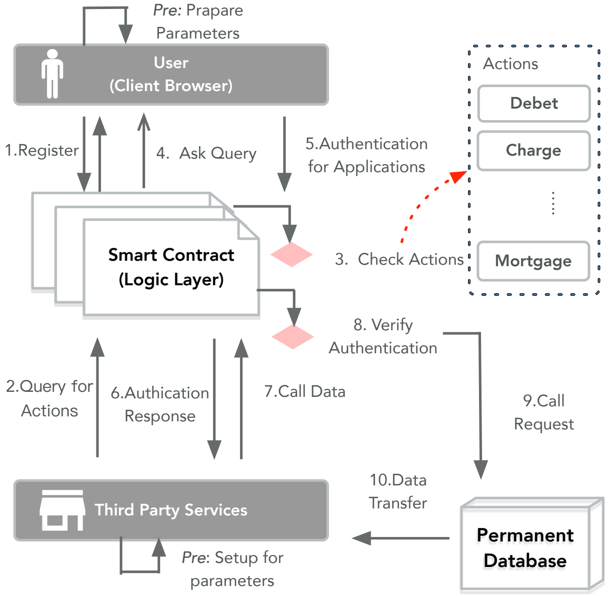

Figure 1: Framework of PPM

  Generate the public key and identity
  Register identity into the contract for records
  Authenticate the access of different actions

- **TPS end behavior:** Third-party service provider are the institutions who provide service and applications for bank users. They process and analyze users' bank data to achieve business purpose.
  TPS End:

  $TPSEnd$: Setup for the parameters
  Register for the Actions
  Call the data

- **Smart Contract** Smart Contract is the entity responsible for logic control. Users registration, data request, and provenance recording are all managed by the smart contract.
  Smart Contract logic:

  Check the registration of users
  Check the existed actors
  Check the authentications of TPS

We combine blockchain ledger and offline data storage together to achieve bank data sharing.

3.2 Design Goals

- **Membership Authentication and Registration:** Authenticated nodes are responsible for verifying all registration requests. And we implements asymmetric cryptography in the system to validate users' identity.
- **Secure Data Sharing:** Bank data is sensitive because it contains all users personal data. In PPM system, we will build the application programming interface (API) to normalize the data request. Different scenarios call the targeted APIs.

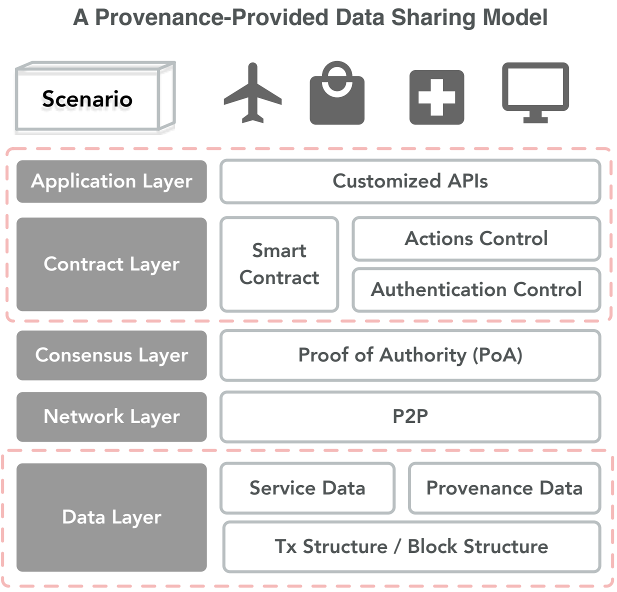

Figure 2: Layer-based Architecture

In addition, our model restricts the specific nodes to call APIs, which implements the access control of bank data.

- **Data Request Provenance:** All request about personal data will be recorded on the blockchain through the smart contract. The provenance data can help users or banks to audit and track the history and behaviors.

4 PPM: A PROVENANCE-PROVIDED DATA SHARING MODEL VIA BLOCKCHAIN

4.1 Overview of PPM

In this paper, we design an open banking data-sharing model based on blockchain architecture, as shown in Figure 2. There are four layers in the PPM system, which are the contract layer, consensus layer, network layer, and data layer. We customize the structure in the data layer to store provenance data. Therefore, manipulations on users' bank data can be recorded on the blockchain.

In the application layer, we presents the application programming interface(API) to regulate external data access. Only authorized third-party companies can visit bank data and obtain the duplicate. In the consensus layer, we adopt proof of authority(PoA) as the consensus algorithm. Traditional banks will act as supernodes, which are responsible for authenticating service providers and verifying data requests.

4.2 Data Structure

Data management is the core layer of our system. In the data-sharing banking system, data security has the highest level of attention. Therefore, considering what categories of data to update and maintained in an open banking system is significant. We will discuss the kind of sensitive data we put in our service and the structure of provenance data in the following paragraphs.

Service Data: As a open banking data sharing system, consumer data regulation is significant in our model. We choose the data which can identify an individual and prove an individual has account in the bank system as the public-access information. The data includes:

- *Financial Product:* Financial product data includes bank products, bank rates, bonds and other services which reflects what the banks provided.
- *Client Information:* Client data contains users' personal information, which is address, phone number, ID number. etc. These data confirms the unique of user's identity.
- *Account Balance:* Account data includes the account balance and account number.
- *Transactions:* Transaction data includes all history about user's account and how much users spent monthly.

All categories we mentioned above will be store into framework's off-line database. When users send a request to grant authorised service providers, the duplicate of data will response after the verification.

Provenance Data: Data provenance is necessary for a distributed system. It can report and track requests about user's data. In the PPM system, users can track manipulations on sensitive data. For instance, if users grant authorized institutions to access their data, they can audit the provenance data to see if there have malicious actions on their data. In our model, when a new request is verified, a form will created to store critical data.

Algorithm 1: Registration

```
Input: Registration Query Message(requestForm, pk,
       userString)
Output: Results
% check the signature of request;
while validSignature(pk, requestForm, id) do
    read current;
    % check the duplication of userString;
    if checkUserString(userString) then
        return "Private String Already Exist";
    else
        addDatabase(requestForm, pk);
        addProvenance(requestForm, currentTime);
        return "Successfully Registered";
    end
end
```

We build the provenance data as the six-tuple, which includes *CREATED TIME*, *FROM*, *TO*, *DURATION*, *VERIFIER* and *TYPE*. The field *CREATED TIME* identifies the time a user creates the request. *FROM* and *TO* presents the hash identity of request initiator and recipient. *DURATION* and *VERIFIER* illustrate the valid time about a data duplicate and node who grant this request, respectively. There are four different values in field *TYPE*, which are "TRANSACTION", "PRODUCT", "ACCOUNT", and "PERSONAL". These values are correspondingly related to the data types as discussed before.

4.3 User Layer

The proposed framework adopts identity-based protocol as the basis on authentication and encryption service. Our system enables users to generate the asymmetric key based on bank account and user-identified string. In this section, we will illustrates the process about how to register an account in our model.

For a user who intends to attend the model, they need to set a private string by themselves at first. The string will be used to generate the asymmetric key pairs. The blockchain will record the public key. Then, when a user sends the registration request, the consensus nodes receive the package and obtain the corresponding public key from the blockchain. In Algorithm 1, we illustrate the process of user registrations. Firstly, nodes will verify the signature with the public key. Then the model will examine the private string preset by users to check if the string has existed in our system. If the signature is valid and the private string is unique, then a package that includes the user's essential data will be included in the transaction and broadcast to the blockchain network.

The offline database will include all user's personal information. Besides, the registration and timestamp information will form the user's initial provenance, which stores into the blockchain after the process completes.

4.4 Third-Party Service

Third-party institutions in our model perform as service providers. They provide various products or suggestions for users on property management, monthly budget, and vehicle registration. Users can obtain proper suggestions from service providers when they send their financial data to institutions. Due to the sensitive and valuable banking data, we implement access control in our model.

As third-party companies, they need to submit the request to supernodes so that they can obtain authorization. When supernodes validate the identities and qualifications of institutions, they will package related data and broadcast them into the blockchain network. Besides, if institutions intend to obtain the duplicate of, they need to request data through remote process call(RPC). We build a series of essential APIs in our model, which will discuss in the security analysis part. The establishment of APIs offers service providers a uniform standard to request financial data. They can legally extend their products to maximize their profit.

4.5 Logic Layer

The predefined laws and rules are written into logic and then developed/deployed into the smart contract. The contract holds the critical access control lists, and one is *Registration* list and the other is *Action* list. The lists record the permitted users and corresponding actions. Whenever a TPS intends to invoke/call data, the authentication is processed under the ACLs. Here is a logic to show how the smart contract proceeds.

As demonstrated in Figure 3, the interactions between both users and third-party service provider (TPS) are presented. There are three main stages in the workflow: *Prepare*, *connected with smart contract* and *invoke data*. Prepare stage represents the initial preparation of both users and TPS. The second stage provides the interactive steps to the smart contract. The smart contract embodies the logic of predefined rules, where the authentication and ACL are depended on. The third stage is to invoke data from the offline database.

Algorithm 2: Logic in Smart Contract

```
Input: Ask Query Messages
Output: Results
% check the Register List;
while RegisterList(pk, RegsterQuery) do
    read current registerList;
    % check the duplication of RegsterString;
    if checkRegister(registerString) then
        return "Register Already Exist";
    else
        addList(requestForm, pk,id);
        return "Successfully Registered";
    end
end
% check the Action list;
while actionList(pk,TID,id,ActionQuer) do
    read current actions;
    func(mortgage);
    func(debitt);
    func(charge);
    % check the duplication of Actions;
    if chkAct(actionString) then
        return "Action Already Exist";
    else
        Action(actionString,pk,TID);
        return "Successfully Added";
    end
end
% check the Authentication list;
while validSignature(pk, requestForm) do
    % verify the authentication of TPS;
    if checkTPSString(actionString,TID,id) then
        callData(actionString,TID,id);
        return "loadingData";
    else
        return "Not Authenticated";
    end
end
```

- **Prepare:**
  - *Pre-User:* The preparation of user is to generate keys pair (sk, pk), identities id , and keep it at local.
  - *Pre-TPS:* The preparation of TPS is to generate an unique identity, and the required actions.
- **Connected with smart contract:**
  - *Step1:* The user sends a registered query to the smart contract. The contract is run inside blockchain oracle as a state transition. When received the query, it will first check whether the user existed in the list. If not, add it into the list. The function is denoted as `Registration`.

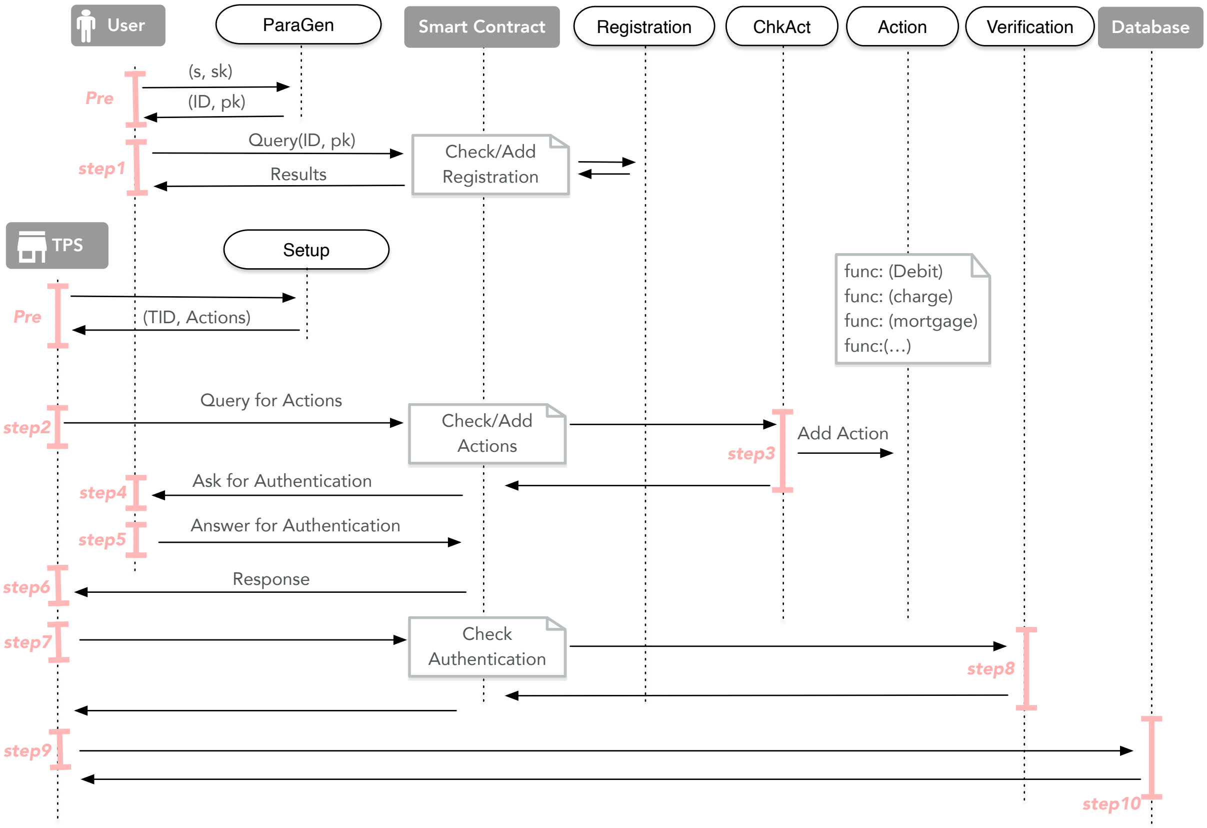

Figure 3: Concrete Workflow of PPM

- *Step2:* The TPS sends a query to the smart contract, which contains the TPS unique identity, and targeted actions for further authentication. When received the query, the contract will first check whether the actions are inside the action list, if not, add it into the list. The function is denoted as `ChkAct`.
- *Step3:* The the function containing specific activity logic is denoted as `Action`. The function of action includes financial-related activities such as debit, mortgage, charge, payment, and so on.
- *Step4:* The smart contract transfer the query to the user for the authentication on specific action. Each query can only represent one action.
- *Step5:* The user reply to the query with Yes/No. The decision depends on the users personal wiliness. The authentication can be set as with three state: one-time authentication, certain time-period authentication, and forever authentication. The authentication reply is sent back to the smart contract, and being recorded into the list. *Step6:* The TPS get the response from the smart contract. The authentication of actions decides whether the TPS can proceed.
- *Step7:* If the response is yes, the function is going to invoke and call the data. This step needs to firstly send a query to the smart contract for verification. When received the query, the contract will check whether the authentication inside the list. If yes, call the data, if not, deny the query. The function is denoted as `Verification`.

- **Invoke data:**
  - *Step9:* After the verification from the access control list on the smart contract, the TPS can call/invoke the data from an independently offline database. The offline database ensures the security of data and decreases the burden of a block. Also, the offline database can be flexibly mounted and layered.
  - *Step10:* Received the query, and the database sends the sensitive data to the TPS. Since then, the whole process of the data sharing from user to TPS is achieved.

5 ANALYSIS OF PPM

Due to the particularity of financial data, we choose proof of authority(PoA) as the consensus algorithm. PoA is the improved algorithm on proof of stake(PoS) to some extent. In this section, we analyze the proposed model in business feasibility and attacks defending.

5.1 Business Feasibility

Our provenance-provided model aims at providing a data-sharing platform for traditional banks. Financial data is sensitive and of great value, which enables banks to choose not to open their service

Table 1: Core Interfaces of PPM

FUNCTION	PARAMETER	RETURN	DESCRIPTION
_addAccount	account_address	1: address, 0: 0x0	Create a new account for users and institutions
_setProvenanceData	object{request details} user_address institution_address	Promise<address> 1: asset address, 0: ”0x0”	Record the request on the blockchain, including time and some detailed data
_verifyTransaction	object{request details} user_address	Promise<object>	Check and validate if the request is legal
_approveData	institution_address approve time, user_address	Promise<object>	Send user’s bank information to service providers
_lockAccount	account_address	Promise<boolean> 1: True, 0: False	Lock the user/institution account
_unlockAccount	account_address	Promise<boolean> 1: True, 0: False	Unlock the user/institution account

data. For the bank industry, keeping financial data private is a kind of temporary protection. The closed database makes banks lose the opportunity to take advantage of sharing data and advanced technology. Nowadays, customers have more critical demands on banking services and more technology giants participate in finance. The closed banking system will lose the chance to build financial ecosphere.

Our model is an open platform that provides a secure protocol for the bank industry to share partial service data. It provides small innovative financial firms opportunities to compete fairly with traditional banks. The benefits for customers are 1) they can manage all bank accounts in a simple interface. 2) they can choose bank products according to personal requirements. 3) they can manage their property and deposit in an efficient way.

5.2 Security Analysis

Public Key Management: In the registration process, the key pairs are generated according to users’ bank account and the private string. Users are required to set a private string for generating their private keys. Furthermore, the blockchain ledger records the corresponding public key. This methodology guarantees user identity anonymity. If their bank accounts are stolen and in inappropriate use, the private key can prevent financial damage.

In addition, if users intend to update their public key, they can update the public key by sending a request. The consensus nodes will update the key through broadcasting transactions.

Audit Provenance Data: When a user sends the request about sharing data, a provenance form is created. It includes the timestamp, user’s address, and the institution’s address. Then after the verifier confirms the signature, the identity of the verifier will be added into the provenance form. Finally, the data type, which values can be ’full’ or ’part’, will be added into the form after the duplicate shares to service providers.

Provenance data, which recorded all manipulations on users’ financial data, are stored in the blockchain. If consumers have an issue about their data access, they can audit provenance records to identify whether their sensitive data has malicious calls. As a result, provenance information can help consumers manage their sensitive data properly.

Access Control: Access control is a significant feature in a distributed model. We build a series of APIs to control external data accesses. As presented in Table 1, we design six critical functions in a smart contract and illustrate in the following paragraph.

- *addAccount:* When users or service providers successfully registered in our system, this function will be triggered, and their essential information will be recorded on the blockchain
- *setProvenanceData:* When the data request is granted, and institutions receive the duplicate, this process will generate a provenance form. The function will be called.
- *verifyTransaction:* This function will be invoked when a new request needs to check the signature.
- *lockAccount:* This function can lock users and institutions account with their address.
- *unlockAccount:* This function can unlock users and institutions account with their address.

Through designing these interfaces, all data accesses will be managed under the regulation. If unauthorized service providers try to obtain personal data, the model will reject the request. In this way, we can secure users’ financial data.

6 DISCUSSION

Table 2 presents the comparison between current related models. We set 7 metrics to evaluate our PPM model, and provide the comparison with 6 related works. *Blockchain-promised* means the basic prototype is based on the blockchain oracle which inherits the properties of irreversibility, transparency, verifibility and traceability. *Identity Management* means the identity can be bound with its corresponding rights. *Programmable Laws* means the predefined laws and rules can be adaptively written into logic and then developed

Table 2: Comparison between Related Models

Features	Gan[5]	Zys[21]	BBDS[19]	Xia[20]	DataProv[13]	ProvChain[9]	**PPM**
Blockchain-promised	Y	Y	Y	Y	Y	Y	Y
Identity Management	Y	Y	Y	Y	N	N	Y
Programmable Laws	Y	Y	Y	Y	N	N	Y
Access Control List(ACL)	Y	Y	N	Y	Y	N	Y
User Privacy	N	N	Y	N	Y	N	Y
Scalable Storage	N	N	N	N	N	N	Y
Data Accountability	Y	Y	Y	Y	Y	Y	Y

and deployed on smart contract. *Access Control List* represents the permissions of users and third party services. *User Privacy* is the customized access control functions, which can be implemented as APIs. *Scalable Storage* represents the permanent storage. Usually the blockchain stores its data on chain, which is limited by the block size. Scalable storage means the storage can flexibly add and drop according to the requirements. *Data Accountability* means the data history can be traced, so that the malicious behaviors can be caught by the public.

According to the table 2, we find that the PPM possess the features where other projects may not. PPM achieves user privacy and ACL by customizing the authentication lists. The internal functions also be hidden into APIs. The scalable storage is realized by the independent offline database for large scale data. The independent offline storage can also be flexibly mounted. The identities is hold by user itself, and the key pair (sk, pk) is generated by themselves. The programmable laws and data accountability inherent from the fundamental properties of blockchain.

7 CONCLUSION

In this paper, we propose a provenance-provided data sharing model (PPM) for the open banking via blockchain. Our PPM model is designed with the properties of transparent authentication, privacy-provided control, and auditable provenance to perfectly meets the requirements of open banking area. Our model employs the programmable smart contracts as the middle witness between uesrs and third party services to guarantee the reliability and trust communication. The predefined laws and rules are written in form of logics inside contract, and are automatically executed in blockchain. Based on that, we decouple the blockchain into hierarchical layers, and present our modifications separately at data layer (data content, transaction structure), smart contract layer (ACL, logic), and application layer (customized APIs). Furthermore, the detailed model module designs and workflow are provided. The analyses and discussion shows our model is a secure and achievable system in face of open banking.

REFERENCES

[1] Karl Aberer, Anwitaman Datta, and Manfred Hauswirth. 2005. A decentralized public key infrastructure for customer-to-customer e-commerce. *International Journal of Business Process Integration and Management* 1, ARTICLE (2005), 26–33.

[2] Tomaso Aste, Paolo Tasca, and Tiziana Di Matteo. 2017. Blockchain technologies: The foreseeable impact on society and industry. *Computer* 50, 9 (2017), 18–28.

[3] Noe Elisa, Longzhi Yang, Fei Chao, and Yi Cao. [n. d.]. A framework of blockchain-based secure and privacy-preserving E-government system. *Wireless Networks* ([n. d.]), 1–11.

[4] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. 2014. A Decentralized Public Key Infrastructure with Identity Retention. *IACR Cryptology ePrint Archive* 2014 (2014), 803.

[5] Saptarshi Gan. 2017. An IoT simulator in NS3 and a key-based authentication architecture for IoT devices using blockchain. *Indian Institute of Technology Kanpur* (2017).

[6] Ye Guo and Chen Liang. 2016. Blockchain application and outlook in the banking industry. *Financial Innovation* 2, 1 (2016), 24.

[7] Henry M Kim and Marek Laskowski. 2018. Toward an ontology-driven blockchain design for supply-chain provenance. *Intelligent Systems in Accounting, Finance and Management* 25, 1 (2018), 18–27.

[8] Neeraj Kumar, Rahat Iqbal, Sudip Misra, and Joel JPC Rodrigues. 2015. An intelligent approach for building a secure decentralized public key infrastructure in VANET. *J. Comput. System Sci.* 81, 6 (2015), 1042–1058.

[9] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. 2017. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing.* IEEE Press, 468–477.

[10] Stephanos Matsumoto and Raphael M Reischuk. 2017. IKP: Turning a PKI around with decentralized automated incentives. In *2017 IEEE Symposium on Security and Privacy (SP).* IEEE, 410–426.

[11] Satoshi Nakamoto et al. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[12] Bo Qin, Jikun Huang, Qin Wang, Xizhao Luo, Bin Liang, and Wenchang Shi. 2017. Cecoin: A decentralized PKI mitigating MitM attacks. *Future Generation Computer Systems* (2017). https://doi.org/10.1016/j.future.2017.08.025

[13] Aravind Ramachandran, Dr Kantarcioglu, et al. 2017. Using blockchain and smart contracts for secure data provenance management. *arXiv preprint arXiv:1709.10000* (2017).

[14] Konstantinos Rantos, George Drosatos, Konstantinos Demertzis, Christos Ilioudis, and Alexandros Papanikolaou. 2018. Blockchain-based Consents Management for Personal Data Processing in the IoT Ecosystem.. In *ICETE (2).* 738–743.

[15] Hossein Shafagh, Lukas Burkhalter, Anwar Hithnawi, and Simon Duquennoy. 2017. Towards blockchain-based auditable storage and sharing of IoT data. In *Proceedings of the 2017 on Cloud Computing Security Workshop.* ACM, 45–50.

[16] Qin Wang, Bo Qin, Jiankun Hu, and Fu Xiao. 2017. Preserving transaction privacy in bitcoin. *Future Generation Computer Systems* (2017). https://doi.org/10.1016/j.future.2017.08.026

[17] Martin Westerkamp, Friedhelm Victor, and Axel Kupper. 2019. Tracing manufacturing processes using blockchain-based token compositions. *Digital Communications and Networks* (2019).

[18] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–32.

[19] Qi Xia, Emmanuel Sifah, Abla Smahi, Sandro Amofa, and Xiaosong Zhang. 2017. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information* 8, 2 (2017), 44.

[20] QI Xia, Emmanuel Boateng Sifah, Kwame Omono Asamoah, Jianbin Gao, Xiaojiang Du, and Mohsen Guizani. 2017. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access* 5 (2017), 14757–14767.

[21] Guy Zyskind, Oz Nathan, et al. 2015. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops.* IEEE, 180–184.