INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(6), 886-902, December 2017.

Using Blockchain to Protect Personal Privacy in the Scenario of Online Taxi-hailing

N. Zhang, S. Zhong, L. Tian

Ning Zhang, Shan Zhong, Li Tian*
School of Information,
Central University of Finance and Economics,
Beijing, China
zhangning@cufe.edu.cn, 2015211624@email.cufe.edu.cn
*Corresponding author: tianli@email.cufe.edu.cn

Abstract: Personal privacy protection issues has gradually caused widespread concern in society which will lead to economic and reputation losses, hinder network and E-commerce innovation or some other consequences if not handled properly. In this paper, we make use of the de-centralization, permanent and audibility of the blockchain to propose a blockchain-based personal privacy protection mechanism, which uses Online taxi-hailing as the application scenario. We not only provide the details of the blockchain custom transaction domain used by the scene, but also expound the information exchanging and blockchain auditing between passengers, Online taxi-hailing platform and drivers in Online taxi-hailing scene, providing a case model for the blockchain solution to personal privacy protection and a technical mechanism solution for further study of personal privacy protection issues.
Keywords: Online taxi-hailing, blockchain, personal privacy protection,

1 Introduction

With the popularization of computer and network, the issue of personal privacy protection has gradually become a concern and concern. If we ignore the personal privacy protection issue or handle it improperly, it may lead to reputation and economic losses or even hinder the network and E-commerce innovation. The problem of personal privacy protection can be solved by both legal and technical means, and it can be also effectively solved by blockchain technology currently. The blockchain is a distributed book that records the history of each transaction sent and verified, as well as the additional information contained in the transaction [3]. All the blocks in the blockchain are arranged in chronological order [11], created by the miners [5], and each node has a copy of the whole blockchain [12]. Because of the features of de-centralization [14], permanent recording and convenient auditing [10], blockchain technology can be used to meet the data security requirements such as integrity and audibility of privacy data, so it can be used as an effective solution of personal privacy protection issues.

There are few literatures on blockchain technology and the literature on the blockchain technology's application in personal privacy protection is more scarce. In the application aspect of blockchain's protecting personal privacy, Melanie Swan [13], referring to the seriousness of health privacy issues, argued that blockchain could provide a mechanism for protecting the privacy of personal health data against infringement, but she didn't give the scenario implementation details of blockchain's protecting personal privacy. In the aspect of using blockchain technology to solve practical problems and providing specific implementation details, Guy Zyskind [15] proposed a decentralized personal information management system in which bit-coins can be used to pass storage, query, and data analysis instructions through some agreement, ensuring that users own and control their own information, but the paper doesn't propose a specific case of application, neither does it provide any personal privacy theory on the issue. Ahmed Kosba [7]

argue that the current blockchain trading environment lacks the privacy protection of transactions, and that money flows between virtual addresses are completely exposed to the blockchain environment, so they propose a decentralized intelligent contract system called Hawk to protect the privacy of transactions by avoiding the clear text storage form of financial transactions on the blockchain. But the method can only protect the privacy of transaction information, which lacks a broader field of real life application. Amir Lazarovich [8] gave a detailed discussion on how to use blockchain technology protect personal privacy, and proposed a distributed-storage based third-party database named escrow, and a blockchain-audit based invisible ink system, taking medical information privacy protection as an example to illustrate the blockchain technology's application in personal privacy issues. But it lacks a discussion on individual privacy issues and the presentation of blockchain application details. In summary, domestic and foreign scholars' research on blockchain application for personal privacy protection is very limited. Therefore, by putting the solution method into the specific application scenarios, we will explain the details of blockchain's privacy protection application one by one, which can provide a mechanism for scholars to further study on the issue.

Some people concern about the information security risks of blockchain technology application like the haker attacks, as the attack sources of information security are characterized as spectral and concealed[9]. The practice shows that compared with other techniques, the possibility of blockchain being attacked by a hacker successfully is very low. Furthermore, it is the time we have come to start thinking about a new paradigm of law, which could balance the power of blockchain technology and emerging autonomous systems in ways that promote economic growth, free speech, democratic institutions, and the protection of individual liberties [14]. As businesses and government strive to accommodate this new way of doing business, countries have been started for blockchain application legal provisions. Such as Arizona Governor Doug Ducey signed HB 2417 into law in March 2017, Delaware's historic blockchain law became effective in August 2017 and Chinese first standard under the guidance of the government released in May 2017.

In addition, it is pointed out that the technical solution of personal privacy protection needs to consider the problems and needs of anonymity, data access control, auditing, online social network privacy protection, mobile location privacy protection and database privacy protection. Online taxi-hailing service just fit those above problems and has more universality and representation than others, so we use it as the application scenario of blockchain's protecting personal privacy. The Didi, one of the hottest online taxi-hailing platform, happened to information leakage because several staff use their permissions to check user travel records and make profits illegally. Leaked information, include the driver's identity, vehicle information and passenger's identity and common address and so on, are likely to threaten users' property and life security.

2 The online taxi-hailing scene and its exposure to personal privacy

2.1 Online taxi-hailing and its process

Online taxi-hailing is a "car rental & designated driving" model for passengers, providing a new way for urban travel.

Consider a scenario where passenger U1 wants to go to work by using Online taxi-hailing software (such as the current popular *Dididache* in China). First of all, the passenger input departure and destination on the Online taxi-hailing software. Then the Online taxi-hailing software automatically obtain departure of passengers, and recommend drivers to the passenger and send passenger's information and travel routes to those drivers. Finally, drivers receive

passenger's routes and decide whether to grab the order. The first one who completes it has the right to complete the transaction.

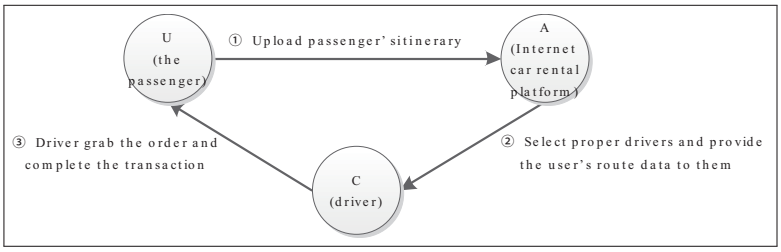

Figure 1: The original Online taxi-hailing scene

2.2 Exposed problems of processes

The passenger uses the software every day and his route is hardly changed. In the process, the passenger encounters some personal privacy issues:

- Problem 1: The passenger can't guarantee that the Online taxi-hailing platform will be confidential to his personal information, and his itinerary and other information may be stolen by the taxi platform to promote advertising, as well as acquisition of other commercial value (such as selling it to third-party data company).
- Problem 2: Platform and drivers can freely view passenger's driving route information, leading to passenger's anxiety that he can't control his personal privacy information.
- Problem 3: When the passenger decides to stop using the Online taxi-hailing software, his information stored on it can't be emptied.
- Problem 4: Imagine that the passenger has asked the Online taxi-hailing software to delete his personal information, and soon after the Online taxi-hailing platform was attacked by hackers. Unfortunately, the passenger found out that his name was on the list while he didn't have any evidence to file a lawsuit against the Online taxi-hailing software.

2.3 Solution

In order to solve the above four personal privacy issues, we use the third-party database and the data interactive audit platform to protect the personal privacy. The third-party certification, reputation, and return policy are interacted to produce a different overall effect on the level of trust [2].

Third-party database means that the user can choose his own trusted database to achieve personal privacy data hosting. Third-party database is an open source database that users can use it through its corresponding link. Different users may choose different third-party database, and through such a distributed data storage we can increase the security of data [7] and reduce the amount of data stored in an Online taxi-hailing software service. Moreover, the third-party database stores user data in the form of encryption, as a result of which the third-party database can't see the specific content of user's privacy and can only serve a loyal guardian on user's data.

Data interactive audit platform is a system built on the blockchain that audits all data operation behaviors. The platform uses blockchain to record all operations on the user data, including data-reading, data-writing, data-updating, and license management for the Online taxi-hailing platform, drivers, and other groups. Forced by pressures of personal privacy protection needs,

Online taxi-hailing software platform and other profit institutions can rent the data interactive audit platform to announce to the public their improvement on personal privacy protection. The platform can ensure that all relevant data operations are recorded in the blockchain which will further strengthen the operation compliance of Online taxi-hailing platform and other parties, allowing users to truly control and master their own data. In addition, due to the Online taxi-hailing software platform renting data interactive audit platform for some privacy purposes, so one data interactive platform may correspond to a number of Online taxi-hailing software platforms.

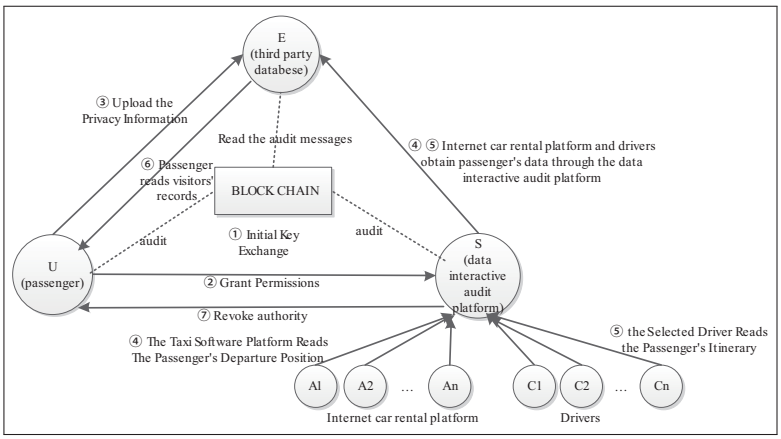

Figure 2: Solve Online taxi-hailing privacy problems

We can use third party database and data interactive audit platform to solve the problem of privacy protection in the Online taxi-hailing scene and its process is as follows.

Firstly, the passenger, the third party database, and data interactive audit platform have their initial key exchange between each other, and the passenger grants permissions to the data interactive audit platform. In order to rent a car, the passenger will upload his itinerary and other privacy information to his third party database. Through third party database and blockchain audit method, he can avoid the Online taxi-hailing software platform getting his entire itinerary (only get the departure, but not the destination). As a result of it, the problem 1 has been solved.

Then, the Online taxi-hailing software platform reads the passenger's departure location and recommends the corresponding matching driver to him. The selected driver reads the entire itinerary of the passenger and decides whether to receive the order. The third party database can't see the passenger's data cause the uploaded data is encrypted, so we can protect passenger's data security by the third party database's decentralized storage of user data. Online taxi-hailing software platform and the driver can access to passenger data through data interactive audit platform, and all operations are recorded in the blockchain. As a result of it, the passenger gets the ability to control his own personal privacy information. The problem 2 has been solved.

Passengers can improve their control of the data by reading the visitor records on the blockchain. At the same time, if the platform leak passenger's private information, the passenger can also have evidence against platform's illegal operation. Therefore, the problem 4 will be solved.

Finally, when the user doesn't want to continue to use the Online taxi-hailing software platform, he can withdraw the operating authority of the data interactive audit platform on the Online taxi-hailing platform to protect his privacy from infringement. So the problem 3 will be solved.

3 Model design and definition

3.1 Actors profiles

- Passenger mobile terminal (U1). The passenger mobile terminal can be a user handset that downloads the Online taxi-hailing platform APP, representing the user identity. U1's action is to upload passenger's itinerary, read other groups' information access records, and grant / revoke the data interactive audit platform S1's access to its data and so on.

- Data interactive audit platform (S1). S1 has been described above in detail, so we will not repeat them here again. S1's action is to help Online taxi-hailing software platform and drivers to read user data, and audit each operation on the blockchain.

- The server of the third party database (E1). E1 has been described above in detail, so we will not repeat them here again. E1's action is to find the blockchain audit records and to determine whether to accept the user's write-data request, and whether to accept software and drivers' read-data request.

- Online taxi-hailing platform (A1). A1 is the Online taxi-hailing company's platform. The action that A1 needs to complete is to obtain the passenger's departure location in order to provide optimal compatible drivers (for example, drivers less than three km away from U1).

- Online taxi-hailing software driver (C1). Online taxi-hailing software drivers can be those drivers' phones that download the Online taxi-hailing platform. The position of the driver is not required to be protected as personal information, so it is visible to the passenger (U1). When A1 finds out a plurality of optimum drivers (C1) that can be matched, then it will send U1's departure point to C1. C1 automatically applies to view the itinerary of U1, and then decides whether to compete for the transaction with U1.

3.2 Blockchain databases

For the design of blockchain database, it is necessary to refer to the specification of blockchain's transaction. Herbert and Litchfield [6] mentioned two approaches to solve the problem of authorization, which are the Master Bitcoin Model and the Bespoke Model. The problem they want to solve is how to use the blockchain to complete the authorization of software using, and how to protect software integrity, prevent software piracy, and complete software updating. In the Master Bitcoin Model, the software vendors transfer 1 unit of Master bitcoin [6] to the user's wallet address via blockchain to granting user the right of using the software. User's software can automatically read transactions. By verification, if the software find that the Master bitcoin in the user address does come from the specific vendor address, it will automatically starts the installation, otherwise it will not be allowed to install. The Bespoke model uses blockchain with a special specification [6], which contains a number of additional domain components in the specification. These additional domains are tailored for flexible requirements of software licensing. For example, we can add token, license, hash software, signature and other domains in the specification, to further implement the software ownership transfer, integrity checks and other advanced features. In addition, the blockchain specifications for different encryption currency is different, and sometimes we can build such a specification based on our own needs to meet the specific needs of the application.

The method also has some inspiration for our Online taxi-hailing case. Through different individuals' sending transactions on the blockchain, we can easily audit the content of their

transactions. Of course, we can also construct a more unique blockchain specification in accordance with our specific purpose. Our specific specification is shown below.

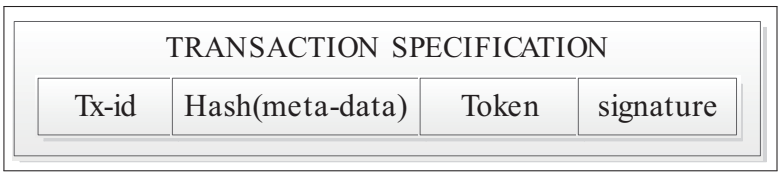

Figure 3: A special definition of specification fields

Hash(meta-data) is the hash value of the message meta-data that will be conveyed by the transaction, the purpose of which is to store the longer message meta-data in a shorter way. Token is used to record who has seen or operated the passenger U1's information. Signature is used to record the sender's signature, mainly to verify the identity of the sender, so as to determine the authenticity of the transaction. Tx-id is a specific transaction number, which plays a role in the blockchain that can uniquely identify the transaction. In fact, tx-id is automatically generated after the completion of the transaction. We put it in the transaction specification to effectively distinguish those different transactions.

In the custom transaction, the sender encrypts the message to a hash value using the public key of the receiver, and the receiver uses its own private key to decrypt it. The sender uses his own private key to generate the signature, and the receiver uses the sender's public key for signature verification.

3.3 Third party database

In addition, the third party database needs to have the ability to establish and maintain *the chain of title* for the user to record all blockchain transactions' number tx-id it has validated and its corresponding transaction *Hash(meta-data1) /Token /Signature /encrypt(meta-data2)* of the tx-id. The third party database needs to maintain a table for each user and data interactive audit platform's combination. The table is constructed as follows. The definitions of Tx-id, Token and Signature are the same as the transaction specification mentioned above, while the content of Hash(meta-data1) is other actors' manipulation on user's private data, and Encrypt(mata-data2) is the encryption of user's private data.

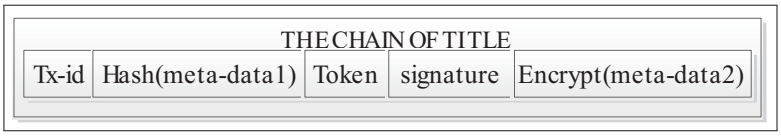

Figure 4: The chain of title

In addition, we should also note that *Hash(meta-data1)* is the hash value of meta-data1, which is invisible to E1. E1 can read token's content and E1 can take appropriate measures based on the token's specific content. Signature is the part that E1 need to verify by using sender's public key which can guarantee the authenticity of the transaction, as long as the verification is successful. *Encrypt(meta-data2)* is the content that E1 can write in the table in accordance with other actor's submitted information.

3.4 Data interactive audit platform database

As mentioned above, the data interactive audit platform is a system which builds on the blockchain and audits all data operation behaviors. The data interactive audit platform's database stores various actors (such as U/E/A/S)'s necessary information when they make some data exchange, including blockchain address (such as UAddress), ID number (such as UID), etc. In this section, the symmetric key (such as keyUS) and the private key of asymmetric key (such as UPrivateKey) are also put into the parameter information in order to express the logical structure of the system more clearly. But in the real world, for the sake of information security, the key can't be stored in the data interactive audit platform database, but should be stored and maintained by a special mechanism (such as the PKI system). This database's UML, attributes and methods are shown below.

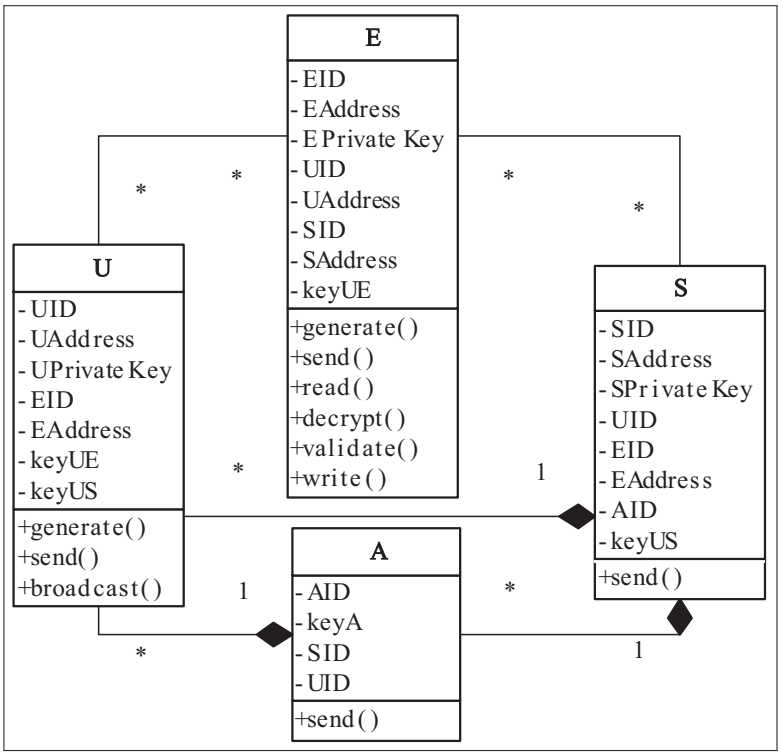

Figure 5: UML of data interactive audit platform database

Actors' attribute definitions are shown as follow.

- XID: X's non-blockchain address. It is the only identification of X. For example, X1's email address.
- XAddress: X's blockchain address, but also X's public key. It is not unique because X1 can have multiple blockchain addresses.
- XPrivateKey: X's private key. For X1, a blockchain address (public key) corresponds to a private key.
- KeyXY: Symmetric key between X and Y. As for U1, he can select multiple E at the same time, but he has a specific symmetric key with each E.

Actors' method definitions are shown as follow.

- Generate(X):

  There are different kinds of keys' generating methods.

  1. Symmetric key generation: The system will automatically generate the symmetric key by the user's mouse or touch screen sliding route. We can use route data as input, and use DES TripleDES algorithm to generate the key which can served as the shared key.
  2. Asymmetric key generation: The system will automatically generate the asymmetric key by the user's mouse or touch screen sliding route. We can use route data as input, and use DES TripleDES algorithm to generate the public/private keys.

- Send(X;Y):

  Send content Y to X. The receiving address includes the blockchain address and the non-blockchain address.

  1. Sending to blockchain address means that we can send transaction to actor's blockchain wallet address. Because the custom domain in the blockchain contains customized content such as meta-data, we can audit its loading information by this transaction. The general form of transmission is *U. send(XAddress; hash(meta-data, token, signature))*.
  2. Sending to non-blockchain address means that we can send information via email or some other means. Like E1 and S1, each actor has its own address and its receiving new message will trigger the corresponding action. The general form of transmission is *U.send(X;Y)*. The receiver has the ability to identify the contents of Y and thus triggers the corresponding action. There are several types as follows.
     (1) Send general information, such as *U.send (EID; UAddress, keyUE)*. It triggers the receiver to record the contents such as "*UAddress, keyUE*" into the database.
     (2) Send authorization instructions, such as *U.send (EID; permission)*. It triggers the receiver to check whether the sender's authorization action has been audited in the blockchain.
     (3) Send request instruction, such as *U.send (AID; request)*. It triggers the receiver to query the sender what information he needs.
     (4) Send authorization instruction, such as *U. send(SID; authorization)*. It triggers the recipient to continue to perform the desired action (as has been authorized).

- Broadcast(X):

  Broadcast content X on the blockchain. The so-called broadcast means that the user send the transaction on the blockchain which will be verified by each node, so as to be able to learn the specific content of X.

- Read(X):

  Read the transaction X on the blockchain. E1's reading transaction means that finding corresponding transaction from blockchain via tx-id. For example, we can find out a corresponding transaction according to tx-id "002", and then extract it for preparing for the next operation step.

- Decrypt(X):

  Decryption means that, after E1 extract corresponding transaction, he will use his own private key to get the specific content of the transaction, which includes tx-id, hash (meta-data), token, as well as signature.

- Validate(X):

  Verification means that E1 uses sender's public key to verify token and signature that he decrypts. After verification, E1 will save his extracted information to his own maintenance table the chain of title. E.validate (token) as an example, the pseudo code of its general process is as follows:

  E.read(token); / E reads the current token

  E.check(U.the_chain_of_title); / In the table the chain of title, E finds the table belongs to U.

  test=ture / Set test variables to true

  For(tx_id=1;tx_id<=tx_id.max; tx_id++) / Loop, in turn to traverse tx-id in U's table the chain of title

  {if(U.token =" U revoke S's authority on operating U's data")

  {test=false;break;}} / If the tx-id's corresponding token's content is revocation, the test variable change to false; if there is no revocation, the test variable is still true

  If(test=true) / After loop ending, check the value of the test variable

  {Insert token into U1.the_chain_of_title;

  Execute(token)} / If test is true, insert the token at the end of the table the chain of title (Of course, in details, it should be inserted together with tx-id, hash(meta-data1), token,signature, encrypt(meta-data2).

  Then execute the contents of the token.

  Else refuse. / If test is false, the execution is rejected.

- write(X):

  Writing content X in the database. When E1 receives tx-id and its corresponding content X from other actors to E1's non-blockchain address, he will write this information to the tx-id's corresponding place. For example, when E1 receives the information sent by U1 whose tx-id is "002" and corresponding content information is encryptkeyUS(meta-data2), he will write this message to the "002" entry.

3.5 Online taxi-hailing software platform database

Online taxi-hailing software platform database is maintained by the Online taxi-hailing software platform itself. In the real world, Online taxi-hailing software platform database are generally more complex, and it is designed according to the specific function and structure of the Online taxi-hailing software. However, this paper simplifies the database, leaving only the key parts of the blockchain interaction to display the database. This database's UML is shown as follows while its role's attributes and methods are defined in the front tables so we will not repeat them here.

4 Model framework and implementation

Overall, the model includes processes such as the initial key exchange, granting permissions, uploading privacy information, reading the passenger departure location, reading the passenger travel, reading the visitor log and revoking permissions.

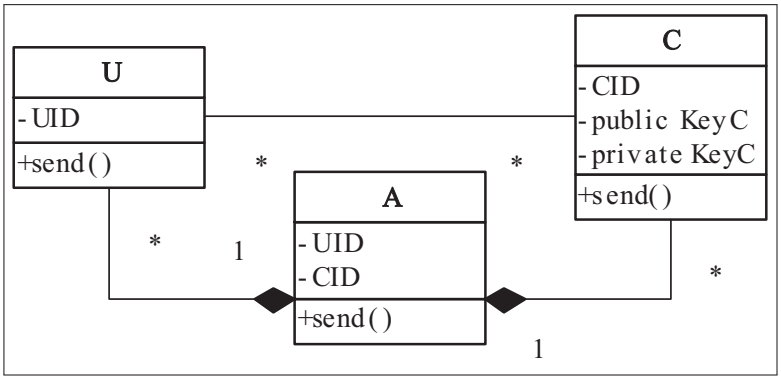

Figure 6: UML of Online taxi-hailing software platform database

4.1 Initial key exchange

U, S, E will exchange their keys firstly. Its process is as shown below.

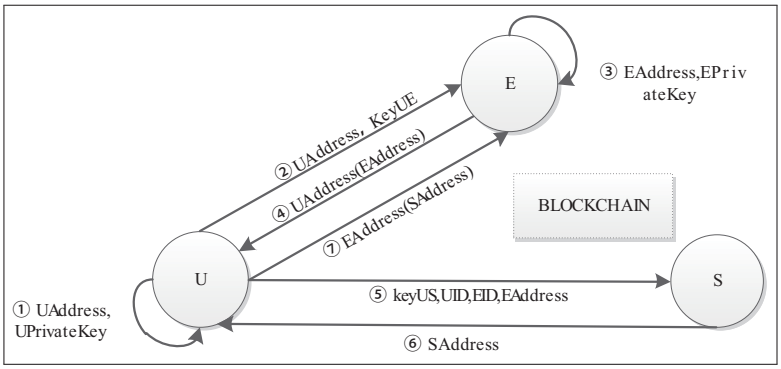

Figure 7: Initial key exchange

① U.generate(UAddress,UPrivateKey)
② U.send(EID; UAddress,keyUE)
③ E.generate(EAddress,EPrivateKey)
④ E.send(UID; UAddress(EAddress))
⑤ U.send(SID; keyUS,UID,EID,EAddress)
⑥ S.send(UID; SAddress)
⑦ U.send(EID; EAddress(SAddress))

4.2 Grant permissions

Passenger U1 requests to use the data interactive audit platform S1 on the Online taxi-hailing software platform and grants the privilege to S1. After authorization, S1 will have the right to read or manipulate U1's data.

① U.send(EAddress; hash(),token,signature)

Note that the content of hash () is *empty*, and the content of token is "*U grant data operation rights to S*".

② B.return(tx-id)
③ U.send(EID; permission,tx-id)
④ E.read(EAddress.transaction)
E.decrypt(EAddress.transaction)

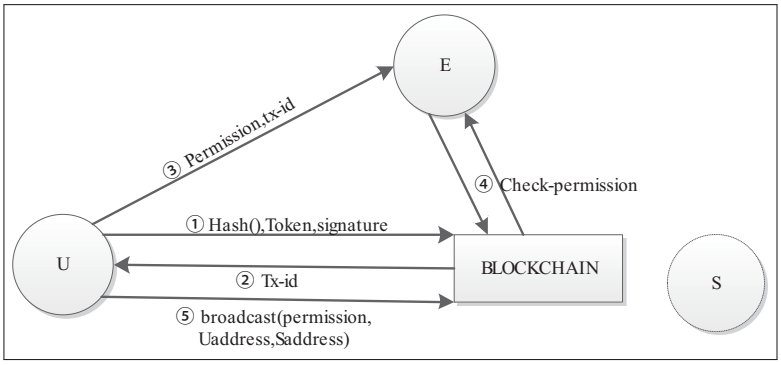

Figure 8: Grant permissions

E.validate(signature)

E.validate(token)

Note that E view the table *the chain of title* which belongs to some user U and find the according transaction through tx-id, finding that the transaction token's all following tokens do not write "*U revoke S's authority to operate its data*" at all. So we can conclude that it is still valid for S to operate U's data.

⑤ U.broadcast(permission,UAddress,SAddress)

4.3 Upload the privacy information

To carry out an Online taxi-hailing order, U1 should first provide personal privacy information such as his own departure and destination, and upload the above itinerary route information to E1 and audit them in the blockchain. As the block size is limited, the audited information uploaded to the blockchain is a shorter hash value. After the passenger upload the departure and destination to the third party database, it will automatically trigger a series of actions (such as the Online taxi-hailing platform recommends several drivers to the passenger). The automatic triggering mechanism is achieved through passenger's sending request to the Online taxi-hailing platform.

Thus, we solved the problem1 that the travel route of the passenger is leaked to A1, so that A1 can't obtain the entire trip route privacy of the passenger or sell passenger's itinerary to other groups.

① U.send(EAddress; hash(meta-data),token,signature)

meta-data's content is "*U1's itinerary and other private information*", while token's content is empty.

② B.return(tx-id)

③ U.send(AID; request)

④ U.send(EID; encryptkeyUS(meta-data),tx-id)

⑤ E.read(EAddress.transaction)

E.decrypt(EAddress.transaction)

E.validate(signature)

E.validate(token)

Note that, although the token's content is empty, E still has to perform the validation process in a flow.

⑥ E1.write(encryptkeyUS(meta-data),tx-id)

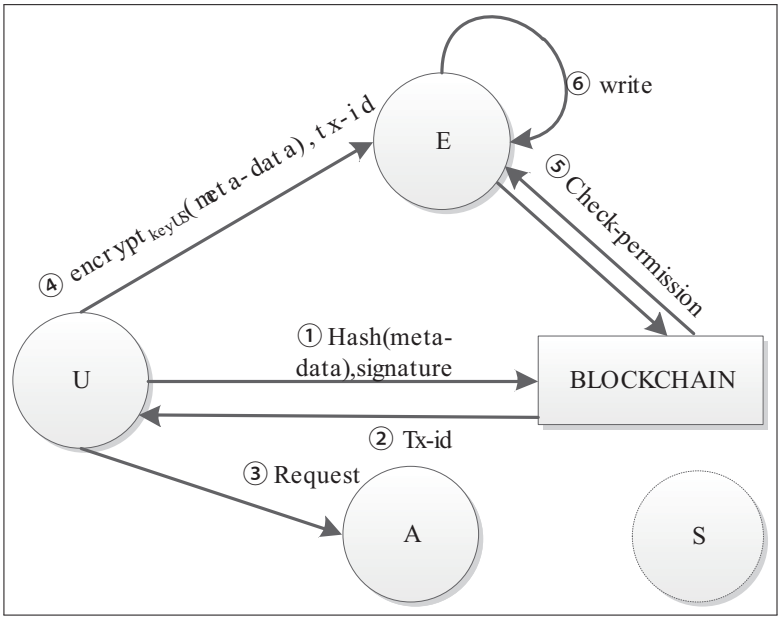

Figure 9: Upload privacy information

4.4 The Online taxi-hailing Software Platform Reads the Passenger's Departure Position

The taxi software platform A1 reads the departure position of the passenger U1 and provides recommended drivers to the passenger on the basis of it. S1 is the privacy protection system selected by the taxi software platform A1. Therefore, if A1 wants to acquire the departure of U1, it still needs to apply to the third party database E1 through S1.

① A.send(SID; keyA, meta-data1)
the content of Meta-data1 is "*A want to get the initial position of U*".
② S.send(UID; meta-data1)
③ U.send(SID; authorization)
④ S.send(EAddress; hash(meta-data1), token,signature)
the content of token is "*U1 authorized A1 to read its initial position*".
⑤ B.return(tx-id)
⑥ S.send(EID; meta-data1,tx-id)
⑦ E.read(EAddress.transaction)
E.decrypt(EAddress.transaction)
E.validate(signature)
E.validate(token)
⑧ E.send(SID,encryptkeyUS(meta-data2))
the content of Meta-data2 is "*the initial position of U1*".
⑨ S.send(AID,encryptkeyA(meta-data2))

4.5 The selected driver reads the passenger's itinerary and compete for the order

The taxi software A1 has read the passenger departure position, matched the optimum position driver C1 through the system, and sent the departure of U1 to C1. C1 will automatically apply to view the itinerary (if user think it's not convenient, he can set the system to automatically authorize, because the entire reading process will be recorded in the blockchain), and

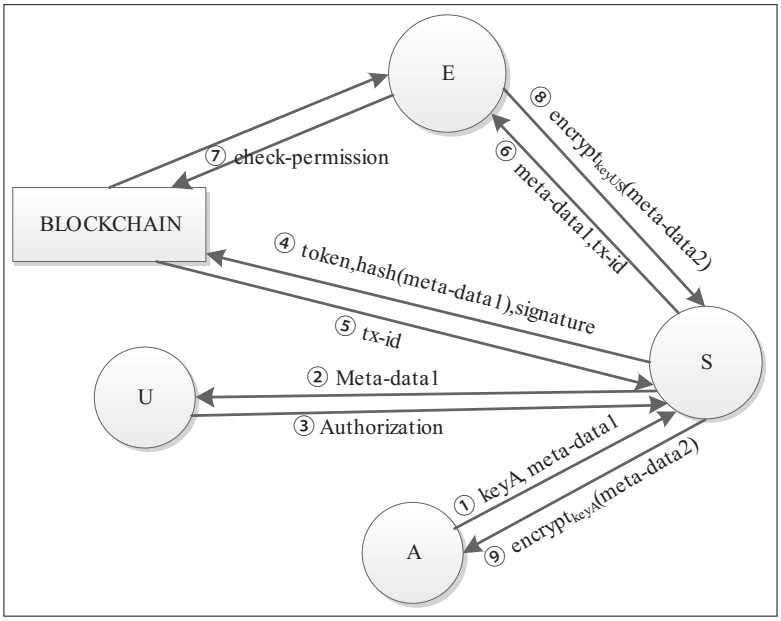

Figure 10: The Online taxi-hailing platform reads passenger departure location

decide whether to compete for it. The entire process avoids A1's obtaining U1's privacy of the itinerary information. After successful competition, some lucky drivers will have the opportunity to provide taxi service to the user.

Thus, we solve Problem 2 and then all operations of A1 and C1 are recorded on the blockchain, which can provide U1 the ability to control his own personal privacy information.

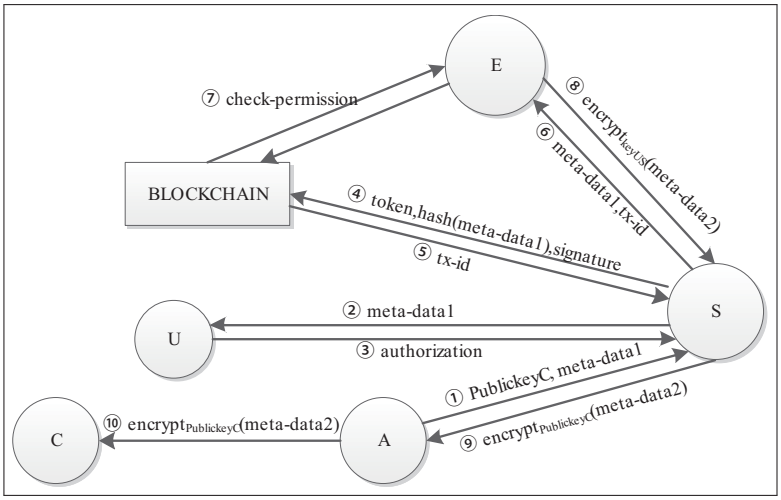

Figure 11: The selected driver reads the passenger's itinerary

① A.send(SID; publicKeyC, meta-data1)
The content of Meta-data1 is "*C1 wants to get U1's itinerary*".
② S.send(UID; meta-data1)
③ U.send(SID; authorization)
④ S.send(EAddress; hash(meta-data1), token,signature)
The content of token is: "*U1 authorizes C1 to get the itinerary of U1*".
⑤ B.return(tx-id)

⑥ S.send(EID; meta-data1,tx-id)
⑦ E.read(EAddress.transaction)
E.decrypt(EAddress.transaction)
E.validate(signature)
E.validate(token)
⑧ E.send(SID; encryptkeyUS(meta-data2))
The content of Meta-data2 is "*U's itinerary*".
⑨ S.send(AID; encryptpublicKeyC(meta-data2))
⑩ A.send(CID; encryptpublicKeyC(meta-data2))

4.6 The Passenger reads visitors' records

Passenger U1 can figure out which actor has attempted to read his data. Here we will use the information stored in the token before, because the token contains U1's authorization records about which visitor visited which information of U1.

Thus, we solved problem 2 and problem 4. U1 can not only see all the visitor records to strengthen his control over the data, but also have evidence to inform the privacy breach platform's illegal operation (failed to clear U1's personal information in time).

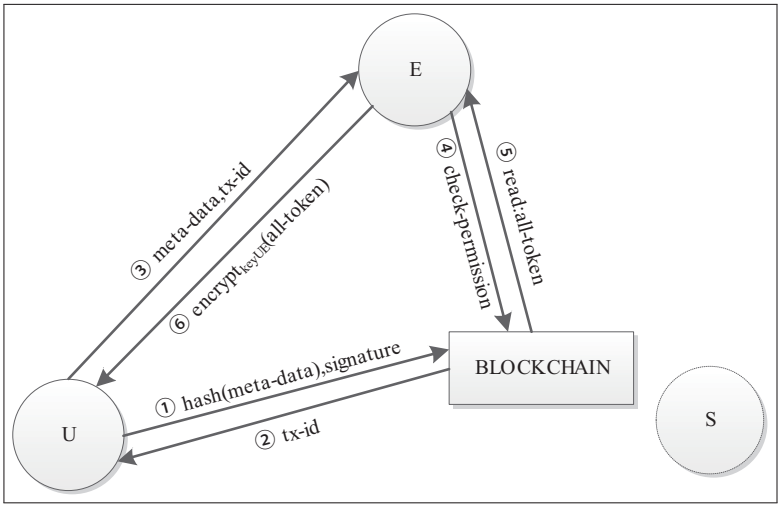

Figure 12: The passenger reads the visitor record

① U.send(EAddress; hash(meta-data),token,signature)
The content of Metadata is "*U1 wants to read which group has tried to access his privacy information*", or "*U1 want to read all his related tokens that have been audited in the blockchain*". The content of token is *null*.
② B.return(tx-id)
③ U.send(EID; meta-data,tx-id)
④ E.read(EAddress.transaction)
E.decrypt(EAddress.transaction)
E.validate(signature)
E.validate(token)
⑤ E.read(all-EAddress.all-token)
E.decrypt(all-EAddress.all-token)
E1 reads all tokens related to U1 from his addresses and then extracts them (E1 can use his own private key to decrypt them) .

⑥ E.send(UID; encryptkeyUE(all-token))

4.7 Revoke permissions

Passenger U1 may, for some reason, wish to terminate the use of the current Online taxi-hailing software platform A1 and wish to terminate A1's manipulation on his own data by revoking the permissions of its corresponding data interactive audit platform S1. On the other hand, U1 can give up the current third-party database E1 and select another one, but he doesn't need to change the current E1 because E1 can't see his privacy information at all.

Thus, we solved problem 3, so that once U1 decide to stop using the Online taxi-hailing software, he can immediately revoke S1's access to his data to protect his privacy from infringement.

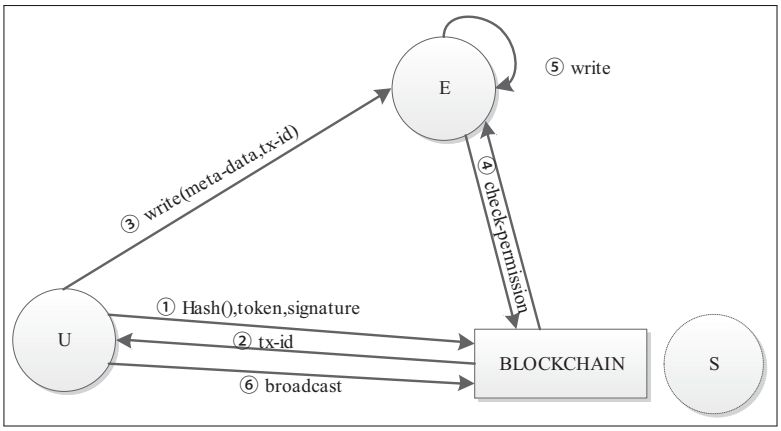

Figure 13: Revoke permissions

① U.send(EAddress; hash(),token,signature)

The content of hash() is *null*. The content of token is "*U want to revoke S's data operation authority on U*".

② B.return(tx-id)

③ U.send(EID; write(meta-data,tx-id))

The content of meta-data is: "*U request to revoke S's data operation authority on U* ". Write means that E is expected to be able to write meta-data in its corresponding table *the chain of title*.

④ E.read(EAddress.transaction)

E.decrypt(EAddress.transaction)

E.validate(signature)

E.validate(token)

⑤ E.write(meta-data,tx-id)

E writes meta-data and tx-id. From now on, as long as E reads "*U request to revoke the operating authority of S on U's data*" in meta-data, he will refuse to continue to provide S services related to U's data. E can also read tokens related to U in *the chain of title*. Once E finds that the last token record of U on S is "*U want to revoke S's data operating authority on U*", E will refuse to provide U's data related services for S.

⑥ U.broadcast(token)

5 Conclusions and future works

Personal privacy problem of Online taxi-hailing has not been effectively solved. In addition to lack of improvement motivation due to users' privacy protection consciousness, the Online taxi-hailing platform also lacks the improvement ability due to the current deficiency of privacy protection technology. Many developing countries, like China, have yet to enact special laws on personal information protection. However, blockchain technology provides an effective mechanism solution to personal privacy protection currently as a strong complement to the legal system. Through the application of our blockchain model, firstly, the taxi software platform can't obtain the entire itinerary privacy of the user, so that the privacy information won't be sold to other for-profit organization anymore. Secondly, the user can see all the visitor records, thereby enhancing his control over his privacy data. Moreover, once the user decides to stop using the Online taxi-hailing software, he can revoke its data interactive audit platform's access to his data. Finally, if the Online taxi-hailing software platform is attacked by hacker which leads to user's information leakage, the user can sue the platform for that it doesn't clear his information promptly, with the evidence of visitor records. In fact, in our opinion that blockchain technology can solve the problem of identity hacking, because if your identity is controlled by a private key, and your own holds that the private key, then there is no way to hack your identity, or at least compared with the traditional database system, the possibility of attack is very low. In addition, maybe separate blockchain may not be the solution to data hacking and identity theft from a technical point of view, but combining with other blockchains rather than fighting alone can solve it.

We boldly apply the blockchain technology to the issue of personal privacy protection, and take the lead in the current popular Online taxi-hailing scenario, where we comprehensively describe the details of blockchain's application in individual privacy protection through processes such as rights-granting, data-writing, data-reading and permissions-revoking, providing a technical mechanism solutions for scholars to further study personal privacy protection issues.

However, this paper still has some limitations. Firstly, from the theoretical point of view, our analysis to personal privacy protection is relatively a little less due to space limitations. Secondly, from the application point of view, our domestic Online taxi-hailing passengers' awareness is relatively weak about personal privacy protection issue. Finally, from the implementation point of view, we provides the data exchange between entities, but doesn't provide the specific implementation and simulation of the system, meaning that further study will be carried out to complete the specific implementation and simulation experiments of the system refer to papers of Barlas et al. [1] and Cotet et al. [4].

Acknowledgment

This research work was supported by National Social Science Foundation of China under Grant No. 13AXW010.

Bibliography

[1] Barlas P., Heavey C., Dagkakis G. (2015); An Open Source Tool for Automated Input Data in Simulation, *International Journal of Simulation Modelling*, 14(4), 596-608, 2015.

[2] Chang M.K., Cheung W., Tang M. (2013); Building trust online: Interactions among trust building mechanisms, *Information & Management*, 50, 439-445, 2013.

[3] Claassen R. A. (2016); *An introduction to bitcoin and blockchain technology*, Kaye Scholer LLP, 2016.

[4] Cotet C.E., Popa C.L., Enciu G., Popescu A., Dobrescu T. (2016); Using CAD and Flow Simulation for Educational Platform Design and Optimation, *International Journal of Simulation Modelling*, 15(1), 5-15, 2016.

[5] Forte P., Romano D., Schmid G. (2015); *Beyond bitcoin part 1: acritical look at blockchain-based systems*, IACR Cryptology ePrint Archive:1164, 2015.

[6] Herbert J., Litchfield A. (2015); A Novel Method for Decentralised Peer-to-Peer Software License Validation Using Cryptocurrency Blockchain Technology, *Australasian Computer Science Conference*, 27-35, 2015.

[7] Kosba A. E., Miller A. J., Shi E., Wen Z., Papamanthou C. (2016); Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts, *IEEE symposium on security and privacy*, 839-858, 2016.

[8] Lazarovich A. (2015); *Invisible Ink : blockchain for data privacy*, Thesis: S.M., Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2015.

[9] Li M., Tang M. (2013); Information Security Engineering: a Framework for Research and Practices, *International Journal of Computers Communications & Control*, 8(4), 578-587, 2013.

[10] Pilkington M.(2015); *Blockchain technology: principles and applications*, Social Science Electronic Publishing, 2015.

[11] Shultz B.L. (2015); *Certification of witness: mitigating blockchain fork attacks*, Undergraduate Thesis in Mathematics, Columbia University in the City of New York, 2015.

[12] Swan M. (2015); *Blockchain: Blueprint for a New Economy*, O'Reilly Media, Inc. 2015.

[13] Swan M. (2015); Blockchain thinking : the brain as a decentralized autonomous corporation, *IEEE Technology & Society Magazine*, 34(4), 41-52, 2015.

[14] Wright A., De Filippi P.(2015); *Decentralized Blockchain Technology and the Rise of Lex Cryptographia*, Social Science Electronic Publishing, 2015.

[15] Zyskind G., Nathan O., Pentland A.(2015); Decentralizing Privacy: Using Blockchain to Protect Personal Data, *Security and Privacy Workshops IEEE*, 180-184, 2015.