JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2016)
DOI: XX.XXX.XXXX.XX

A Decentralized Blockchain-based Ride-hailing Mode with Attribute Encryption

YI-FAN ZHANG[1], YU-PING ZHOU[1,2]*, YU HU[1], XIN-YU YI[1], BEN XIE[1]
[1]*School of Computer Science*
Minnan Normal University
Zhangzhou, 363000, P.R.China
[2]*Key Laboratory of Data Science and Intelligence Application, Fujian Province University*
Zhangzhou, 363000, P.R.China
E-mail: 1021784165@qq.com; yp_zhou@mnnu.edu.cn; yuhucn@126.com;
2872668611@qq.com; 2677307826@qq.com

With the development of smart transportation, ride-hailing applications have become an essential part of people's lives. These ride-hailing apps provide convenience of contacting taxi for passengers. However, most present ride-hailing or ride-sharing systems rely on a trusted third party. It makes them be attacked vulnerably. A decentralized blockchain-based ride-hailing mode with attribute encryption is proposed in this paper. Attribute-based encryption is applied to ensure the drivers who meet the passenger's requirements can obtain the passenger's order in this mode. After the transaction has completed, the transaction information is saved on the blockchain. This mode supports the investigation of historical records via the blockchain technology. Besides, a new payment protocol and a reputation algorithm are used in this mode. The new payment protocol is based on trip distance. It applies smart contract and zero-knowledge set membership proof. The reputation of drivers based on drivers' past behavior is designed. The driver's reputation will be updated after the transaction is completed. Passengers can choose a driver with high reputation. Each phase of this mode is simulated in our test net of Ethereum. The results prove that this ride-hailing mode is efficient.

Keywords: Blockchain, Ciphertext policy attribute-based encryption (CP-ABE), Zero-knowledge proof, Smart contract, Ride-hailing service.

1. INTRODUCTION

In the past decade, online ride-hailing has gradually replaced the traditional taxi mode [1]. The advantages of online ride-hailing are reflected in the following points. First, people can choose the location where they are picked up by the driver. Second, people can confirm whether the driver's trip is correct in real-time. Third, all transactions will be saved in the database. People can check their historical records of riding hailing at any time.

Most of the current ride-hailing modes are based on third-party platform. Relying on centralized services makes the system more vulnerable. The centralized modes suffer Denial of Service (DoS) attacks easily [2]. If one single point is attacked successful, the system may crash [3]. In addition, the centralized modes lack transparency. Some

Received March 11, 2014; revised June 20, 2014; accepted September 28, 2014.
Communicated by the editor.

malicious third-party platforms sell the private data of users. Thus, it is not safe to store information on the third-party platform. Furthermore, the third-party platform requires high service costs [4, 5].

Compare with the current centralized ride-hailing mode, the Blockchain technology provides a distributed storage solution. Data is saved in the distributed store environment. It can avoid the situation where the data is completely controlled by someone. Meanwhile, the Blockchain can avoid the problem of single-point failure and DoS attacks by the decentralized structure. However, there are few works using Blockchain to build a decentralized ride-hailing mode [6, 7]. They are working on transparency and privacy.

In this paper, we propose a decentralized ride-hailing mode with attribute encryption. Compare with other taxi-hailing modes, our mode does not rely on the third-party platform. Moreover, we use attribute encryption to satisfy passenger's choice. Passengers can choose the special driver according to their attributes. The attributes include the sex of drivers, the years of driving, the type of the car etc. The passenger encrypts the requirement by the attributes tree and send it to the blockchain. Each vehicle will receive the requirement of the passenger encrypted by ciphertext-policy attribute-based encryption (CP-ABE). The driver with specific attributes has the right to decrypt taxi information. Then the passenger choose the driver based on the unit price and reputation of the driver.

In general, these are three contributions are followed in our mode.
1. The decentralized blockchain-based ride-hailing mode with attribute encryption is proposed. Attribute-based encryption is used in this mode. It guarantees that the user can choose the drivers with specific attributes.
2. Attribute-based encryption is different from identity-based encryption. A fine-grained access control policy is built for the driver. The two parties can complete more secure transactions without exposing their identities.
3. A deposit protocol ride-hailing services based on the zero-knowledge set membership is proposed to ensure the trust between a rider and a selected driver.

2. RELATED WORKS

2.1 Ride-hailing System

In the early days of the study, the centralized taxi mode was proposed. Aïvodji et al. [8] proposed SRide in 2018. It's a ride-sharing system based on homomorphic encryption. Because of the complex process and homomorphic encryption, SRide does not work well. He et al. [9] proposed a matching system based on the real distance between the drivers and the passengers. Sherif et al. [10] used group signature in their system in 2017.

Different from these previous research, many researchers were focusing on the blockchain. Yuan et al. [11] researched the effects of blockchain in ride-hailing. Li et al. [12] used private blockchain to build the system. The system consists of a large number of Road Side Units(RSUs). So it is costly owing to the large number of RSUs required in the system. Baza et al. [13, 14] proposed a ride-sharing system named B-Ride in 2019. B-Ride was deployed in Ethereum. The cloaking technology is used in B-ride to ensure the privacy of users' locations. Passengers and drivers communicate via blockchain.

2.2 Blockchain Technology and Smart Contract

Satoshi [15] created Bitcoin in 2008. As the core technology of Bitcoin, blockchain technology builds the framework of Bitcoin. The blockchain is composed of data blocks in chronological order. The data block consists of a data area and a pointer. The pointer shows the location of the previous block. There are many researches about the blockchain network. Myers et al. [16] and Ben-Sasson et al. [17] proposed two decentralized anonymous Bitcoin payment system. The zero-knowledge proof technology was used in these systems to protect users' privacy. A new protocol consensus algorithm was proposed by Ripple [18]. The algorithm is based on Unique Node Lists (UNLs). The data of the blockchain is open to everyone or open to the data owner. It doesn't have an efficient fine-grained control of the data of the Blockchain. The concept of smart contract was proposed by Vitalik Buterin [19]. After smart contract is deployed on the Blockchain, it can be executed autonomously. Furthermore, it avoids the interference of the third-party.

2.3 Attribute-Based Encryption

The fuzzy identity-based encryption (IBE) was proposed by Sahai and Waters [20] in 2005. Goyal et al. defined attribute-based encryption (ABE). The fine-grained access control was allowed by attribute-based encryption. If the attributes of the user conforms to the access control, the user can decrypt ciphertext. The user's identity is no longer important in the decryption process, instead of their attributes. So, ABE implements fine-grained access control to data.

Both ciphertext policy attribute-based encryption (CP-ABE) and key policy attribute-based encryption (KP-ABE) belong to ABE[21, 22]. In CP-ABE, the ciphertext is related to the access control policies, and attribute is related to the key. The situation is just on the contrary in KP-ABE. Thus, Bethencourt et al. [23] and Ibraimi et al. [24] used tree structure to build access policy. However, the complexity of the tree structure [25] is proportional to the time spent on encryption and decryption.

2.4 Zero Knowledge Set Membership (ZKSM) Proof

Zero Knowledge Proofs (ZKPs) is an essential cryptographic technique. Many researchers have tried to implement ZKPs by computer. ING[26] proposed Zero Knowledge Range Proofs (ZKRPs) in 2017. In ZKRPs, the secret value is permited in a certain interval. But ZKRPs does not support generic collections. ZKSMs solves this problem perfectly. Camenisch et al. [27] used the generic set as the secret value to build ZKSMs. In other word, ZKSMs has a wide range of applications as opposed to ZKRPs.

3. SYSTEM MODE

A decentralized blockchain-based ride-hailing mode is illustrated in Fig 1. There are three roles in our system mode:

1. Passenger: The passenger means someone who needs ride-hailing service. The passenger generates a ride-hailing message which contains an attribute access tree and ciphertext. The tree contains the attributes of the driver that the passenger is looking for. The ciphertext contains the passenger's location information and contact information.

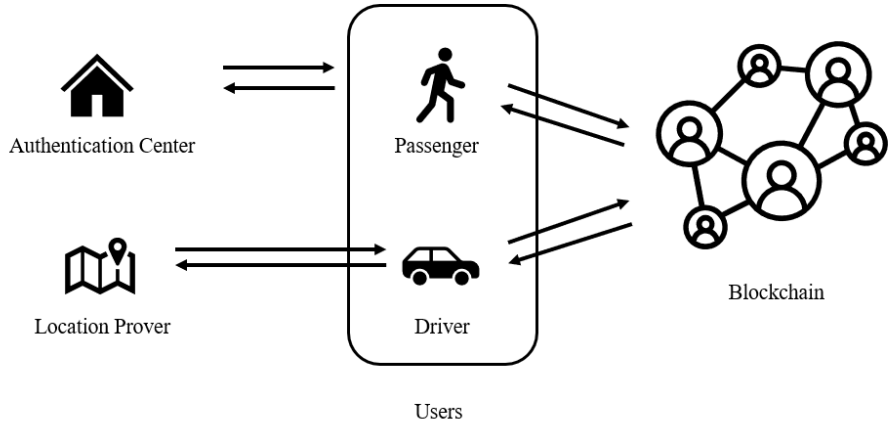

Fig. 1. Decentralized Blockchain-based Ride-hailing Mode.

2. Driver: The drivers means someone who provides ride-hailing service.The driver is responsible for receiving the order of passenger from the blockchain. After obtaining the passenger's information, if the driver's attributes meet the passenger's requirement, the driver will decrypt the requirement to decide whether take the order. If the driver takes the order, the driver will add a random number to the end of the decrypted message.
3. Location Prover(LP): The LP is deployed in different locations of the road as an equipment. In general, the Road Side Unit (RSU) is always used as the LP. The responsibility of the LP is to verify the location of the driver. After the driver arrives at the location, the driver requests confirmation of the location from the LP. If the address is correct, then the LP will sign the location of the driver.
4. Authentication Center(AC): AC is a trusted third party. Each user in the system can initiate a request to the AC. AC distributes attribute sets and secret keys to the users of this mode. It can also authenticate the user's identity.

4. METHOD

This mode consists of the following four phases: matching phase, deposit payment phase, fair payment phase, and reputation calculation phase.

4.1 Matching phase

This section proposes the way how passengers match drivers on public blockchain. Algorithm 1 and Algorithm 2 summarizes the process of the encryption and decryption. Otherwise, there are three necessary steps for bidding and selection in Fig 2 .

Algorithm 1 Encryption of information

Input security parameter λ, κ; a set of attribute sets of drivers S_{system}; a new message m; the identity of driver id; the pseudonym of driver pse.

Output The ciphertext C.

1: $Setup_{att}(1^{\lambda}) \rightarrow (PK,MK)$
2: $Setup_{sig}(1^{\kappa}) \rightarrow pp$ // pp means a public parameter which is used to generate sercet key and public key
3: $Keygen_{att}(MK,S) \rightarrow SK$ // S means a set of attributes. SK is the attribute sercet key related to S.
4: If the id is verified and pse is unique then $Keygen(pp,pse) \rightarrow (pk,sk)$
5: Selecting $S_{passenger} \in S_{system}$, $S_{passenger} \rightarrow T$ // $S_{passenger}$ means a set of the driver's attributes which the passenger want. T means the tree structure. Using attributes to construct tree structure.
6: $Enc_{att}(PK,m,T) \rightarrow C$

Algorithm 2 Decryption of information

Input the master key MK, a set of driver's attributes S_{driver}. The ciphertext C

Output The message m.

1: $Keygen_{att}(MK,S_{driver}) \rightarrow SK_{driver}$
2: $Dec_{att}(C,SK_{driver}) \rightarrow m$ //If S_{driver} meet the requirement of the passenger , then the driver can decrypt the information

4.1.1 Request from the passenger

The passenger generates their ride information. The information includes the starting point of the trip, end point of the trip, and the passenger's location. Then the passenger chooses the set of attributes of the driver who they want. They utilize it to build an access policy with a tree structure. This structure is used to encrypt the ride information. After getting the ciphertext, the passenger submits ride information to the blockchain.

4.1.2 Reply from the driver

If the driver's attributes match the access policy, the driver can decrypt the ciphertext and get the ride information from the passenger. The driver has the right to decide whether to accept the order or not. If the driver decides to accept the order, the driver will encrypt the unit price and the ride information with the public key of the passenger. Then the driver submits the ciphertext to the blockchain.

4.1.3 Matching completed

After the passenger obtains the driver's unit price from the blockchain, the passenger compares all drivers' unit price and past reputation. Finally, the passengers select a qualified driver to communicate via blockchain.

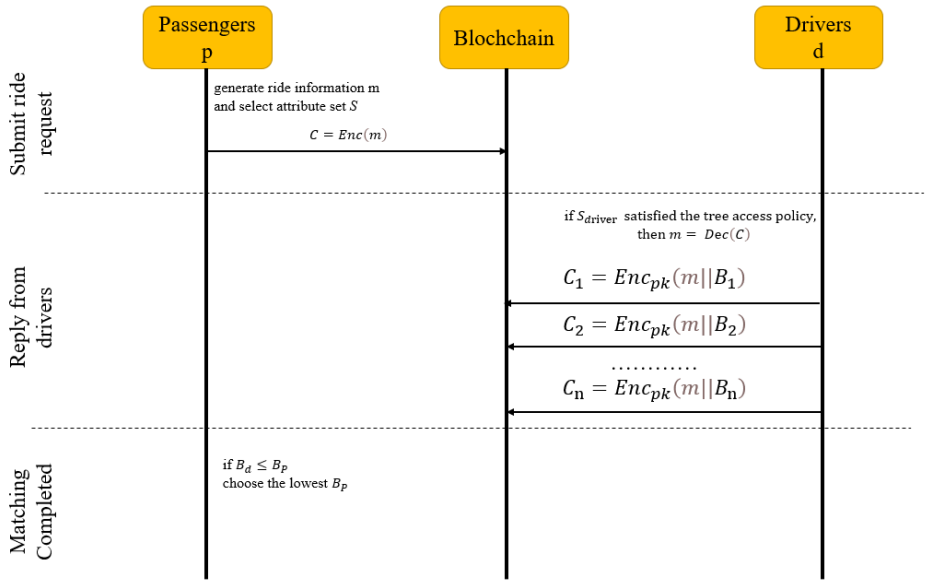

Fig. 2. Diagram of the Matching Phase.

4.2 Deposit Payment Phase

This section introduces a deposit payment protocol that prohibits malicious drivers and passengers from giving their respective ride offers or requests. The cost of malicious behavior increases due to the deposit. In most schemes, this part is supervised by a trusted third party. However, the third party is not always credible. In other words, the entire system cannot remain secure when the third party is attacked. Based on this situation, a ride-hailing deposit payment scheme was proposed. The Passenger publishes deposit payment contract on the blockchain. The driver sends D deposit to the smart contract when the driver arrives at the designated location. After paying the D deposit, the driver sends the request of the location authentication to the LP. If the driver doesn't arrive the designated location, the passenger will receive the deposit directly from the contract. This workflow of the deposit phase is shown in Fig 3.

4.2.1 Initialization

The passenger plans to generate their deposit payments by these following steps:
1) The passenger defines a location set ϕ , let $\phi \in \{ l_1, \cdots, l_k \}$. The set ϕ should include all pick-up locations.
2) The passenger chooses a number $x \in Z_p$ randomly, then compute $y \in g^x$, g is the generator of Z_p.
3) Computes $A_i = g^{\frac{1}{(x+i)}}, i \in \phi$.
4) Generates a contract. The contract is shown in Algorithm 3.

Algorithm 3 Deposit Payment

uint public balance
address payable passenger
address payable driver
uint public PassengerDeposit
uint public DriverDeposit
address Set
address A_i
function Depositpayment(_driver, _Set, _A_i, _PassengerDeposit)
    {
    _driver → driver;
    _PassengerDeposit → Balance;
    _Set → Set;
    _A_i → _A_i;
    }
function DriverDeposit(uint256 _DriverDeposit)
    {
    if block.timestamp ≥ expiration return;//Expiration means the valid time of the contract
    if msg.sender ≠ DriverAddress return;
    if msg.value ≠ DriverDeposit return;
    if now ≥ $T_{deadline}^{Accept}$ return;// $T_{deadline}^{Accept}$ means the deadline which the passenger can accept
    _DriverDeposit → Balance;//Confirming the feasibility of the transaction
    }
function ProofOfArrival((π , $\sigma_{LP}(C)$))
    {
    if msg.sender ≠ DriverAddress return;
    if now ≠ $T_o^{(p)}$ return;//$T_o^{(p)}$ means the time set by the passenger
    if (ZKSM.Verifier(π))
    score_1[msg.sender]+1 → score_1[msg.sender]; //the first reputation increases when the location is verified
    transfer(balance,driver);
    end
    }
function Transferbalance()
    {
    if block.timestamp ≥ expiration return;
    if msg.sender ≠ passenger return;
    transfer(balance,passenger);
    }

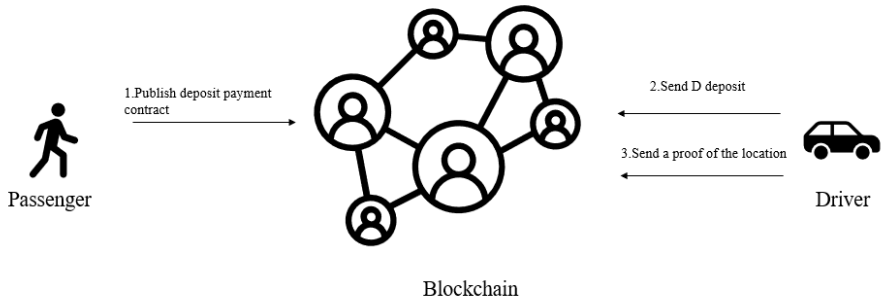

Fig. 3. Workflow of the deposit payment phase.

4.2.2 Driver Deposit and Claim

Upon arrival at the pick-up location, the driver uses Zero Knowledge member Set Membership Proof to prove that he or she has arrived:
1) Driver picks $v \in Z_p$ and computes $V = (A_{l_0^{(p)}})^v$, $A_{l_0^{(p)}}$ is the signature on $l_0^{(p)}$. Then, the driver computes $C = g^{l_0^{(p)}} h^l$.
2) The location is confirmed by the LP. If the location is correct, then LP will compute $C' = Cg^{-l_0^{(r)}} = g^{l_0^{(r)}} h^\iota g^{-l_0^{(r)}} = h^\iota$.
3) Driver chooses three numbers $s,t,m \in Z_p$ randomly, then computes $a = e(V,g)^{-s} \cdot e(g,g)^t, Q = g^s h^m$.
4) When the proof is received, the proof will be verified. If the proof belongs to the driver selected by the passenger and the time is validated. the commitment C is proved by the following statement : $PK\left\{(l_0^{(r)}, \iota, v) : C = g^{l_0^{(r)}} h^\iota \wedge V = g^{\frac{v}{x+l_0^{(r)}}}\right\}$.
Blockchain will check the following equation: $Q = C^c h^{z_\iota} g^{z_l}$ and $a = e(V,y)^c \cdot e(V,g)^{-z_l} \cdot e(g,g)^{z_v}$. If the equation is correct, then this proof is correct

4.2.3 Users authentication

The public key signature is used to prevent impersonation. The driver picks an integer $S \in Z_p$ randomly as a secret number then sends it to the passenger. The passenger uses the private key to sign S then generates $(\sigma_{sk(p)}(S))$. The passenger sends $(S||\sigma_{sk(p)}(S))$ to the driver. The driver verifies whether the signature is valid.

4.3 Fair Payment Phase

A new way of payment is proposed showed in Algorithm 4. The journey is divided into many small parts. For each part traveled, the passenger must pay the driver the corresponding amount of money. The deposit stored previously by the passenger is paid as the price of the first part. After this, the driver asks the passenger for payment for every part traveled. The passenger confirms whether the driving distance is correct. If

Algorithm 4 Fair Payment Depends On Distance

address payable passenger
address payable driver
uint public Tripdist
function Ridepayment(_driver, _Tripdist, $\_t_d^{(R)}$)
    {
    _driver $\rightarrow$ *driver*;
    _Tripdist $\rightarrow$ *Tripdist*;
    }
function ProofOfDistance(ElapsedDist)
    {
    if msg.sender $\neq$ *PassengerAddress return*;
    while *Dividedpart* $\leq$ Tripdist **do**//Dividedpart means the distance of the divided part
    transfer((balance × Dividedpart), driver))
    _TripDist − Dividedpart $\rightarrow$ *TripDist*;
    if (msg.sender == 0)
    score_2[DriverAddress]+1 $\rightarrow$ score_2[DriverAddress];//the second reputation increase when the order is finished
    end
    end
    }
function withdrawFunds()
    {
    if block.timestamp <expiration return;
    if msg.sender $\neq$ owner return;
    transfer(balance,owner);
    }

the distance is correct, the driver and the passenger sign this message and upload it to the smart contract. At the same time, the fare is transferred to the driver's account. If the passenger does not pay the money, the driver can terminate the trip at any time. This payment method ensures fairness for both users.

4.4 Reputation Calculation Phase

In this mode, the smart contract is also used to calculate the user's reputation value. We define the reputation score as β_d its value consists by β_d^{AP} and β_d^{AD}. When the driver arrives at the agreed location, β_d^{AP} increases. the second one β_d^{AD} increases after the trip is finished. Then the system calculates $\beta_d = \frac{\beta_d^{AP}}{\beta_d^{AD}}$. If $\beta_d = 1$ holds , then $\beta_d^{AP} = \beta_d^{AD}$. It indicates that the driver complete all orders. If $\beta_d < 1$, then some orders of the driver are not completed.

5. SECURITY ANALYSIS

The security in the mode is discussed in this section. This mode have the following characteristics.

1. Anonymity: The order which is encrypted by the attributes are stored on the blockchain. Therefore, the adversary who has obtained the ciphertext cannot get the user's identity. Even if the adversary is able to decipher the cipher in polynomial time, the adversary can only get the attributes of the user. In CP-ABE, there is no obvious relationship between the user's identity and attributes. Only the AC knows the real identity of the user. So users do not need to consider the anonymity of their identity.

2. Untraceability: Since the user's pseudonym needs to be authenticated by the AC. The pseudonym is unique in this mode. The adversary can easily find out the orders corresponding to the same pseudonym. Once the user's pseudonym is changed, it will be difficult for the adversary to obtain the connection between different pseudonyms. So the user's order is difficult to trace.

3. Fine-grained Access Control: While the information on the public blockchain can be accessed by all users. The order is encrypted with attributes. The selection of user attributes is already done after the message is encrypted. Only users whose attributes meet the requirements can access the contents of the order.

6. PERFORMANCE EVALUATIONS

In this section, the performance of attribute encryption and smart contracts is evaluated. For attribute encryption, the time spent on encryption is a very important metric. We tested the encryption time for different depths of the attribute access tree and the different attribute complexities. However, the most important metric for smart contracts in Ethereum is the consumption of gas. The consumption gas represents the fee which is required in the contract. The gas value in the contract should be as low as possible.

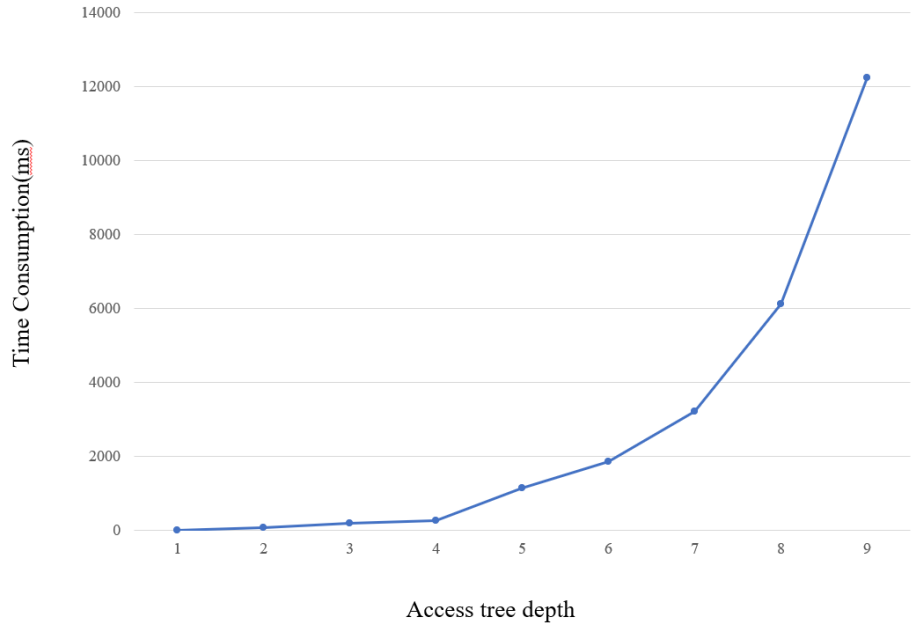

Fig. 4. Time of CP-ABE with a Different Access Tree Depth.

6.1 The Time Cost of The Attribute Encryption

We simulated the encryption phase on a personal computer with a 2.80 GHz Core i7-7700 CPU Intel, 24.0 GB RAM and 64-bit version of the Windows 10 operating system. The code of CP-ABE is coded in Java. We develop CP-ABE in Java Runtime Environment 1.8. The bilinear pairs in this mode are constructed with elliptic curves. We define all parent nodes in the access tree to have two child nodes. Tree access structure control is used for the encryption of our taxi solution. The relationship between the time consumption of CP-ABE and the depth of access tree is shown in Fig 4. For instance, the depth of the access policy [[Man and 7 years driving experience] or [Woman and 7 years driving experience]] is three. The depth of [Man and 7 years driving experience] is two. It is easy to see that the time consumption of CP-ABE increases with the increasing depth of the attribute tree. For passengers, if they choose too many attributes of driver. The time consumption of the system will increase significantly. Therefore, passengers should choose the number of attributes appropriately according to their time cost. The relationship between the time consumption of CP-ABE and the complexity of the attributes is shown in Fig 5. The complexity of the attributes represents the number of nodes other than the root and leaf nodes. It is simple to see the exponential growth between the encryption time and the complexity of attributes.

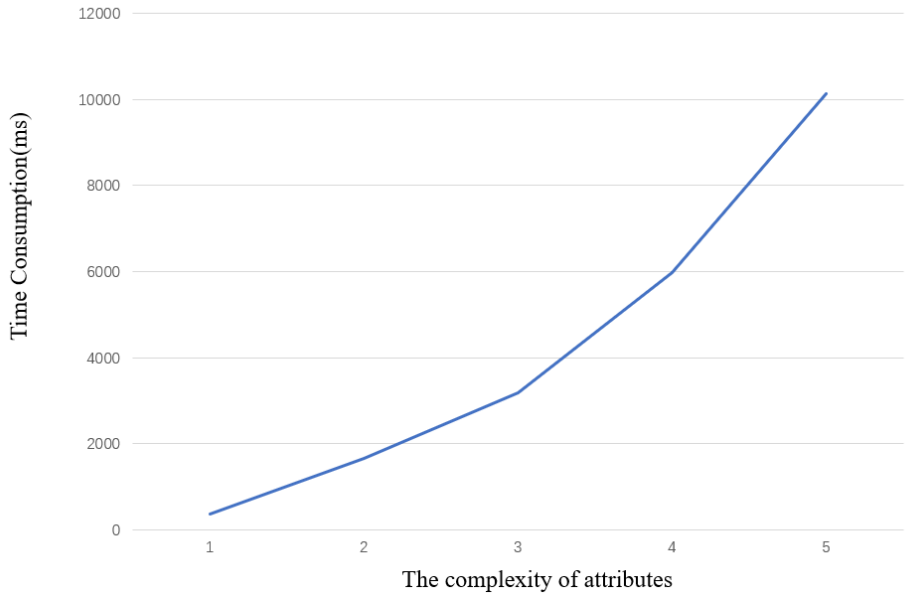

Fig. 5. Time of CP-ABE with Different Complexity of Attributes.

Table 1. Precompiled Contract Name and address

Precompiled contract name	Features	Address	Gas cost
bn256Add()	Addition on elliptic curve	0x6	500
bn256ScalarMul()	Scalar multiplication on elliptic curve	0x7	40000
bn256Pairing()	Checking pairing equation on curve	0x8	100000 + 80000 * k

6.2 The Gas Cost of This Mode

Gas is used as the cost of the transaction in Ethereum. All smart contracts in this mode are coded in Solidity. However, if ZKSM is coded into the contract, it will consume a lot of gas. We can save a lot of gas by using the precompiled contracts of Ethereum Virtual Machine. The precompiled contracts are shown in Table 1. Gas is spent respectively on the transaction cost and execution cost. Execution cost includes the overhead of storing global variables and the cost of method calls. But, transaction cost is related to the length of the compiled contract code. The value of transaction cost and execution cost should be constant for each execution of the same contract. The gas consumption of drivers and passengers is shown in Fig 6 and Fig 7. The cost of the matching phase is about 560K gas. The cost of the deposit phase is about 150K. The cost of the payment phase is about 330K. The cost of the distance proof is about 30K. The mode is used to compare with other modes. The result is shown in Table 2. It is not difficult to see that this mode ensures privacy when it can select the attributes of the driver.

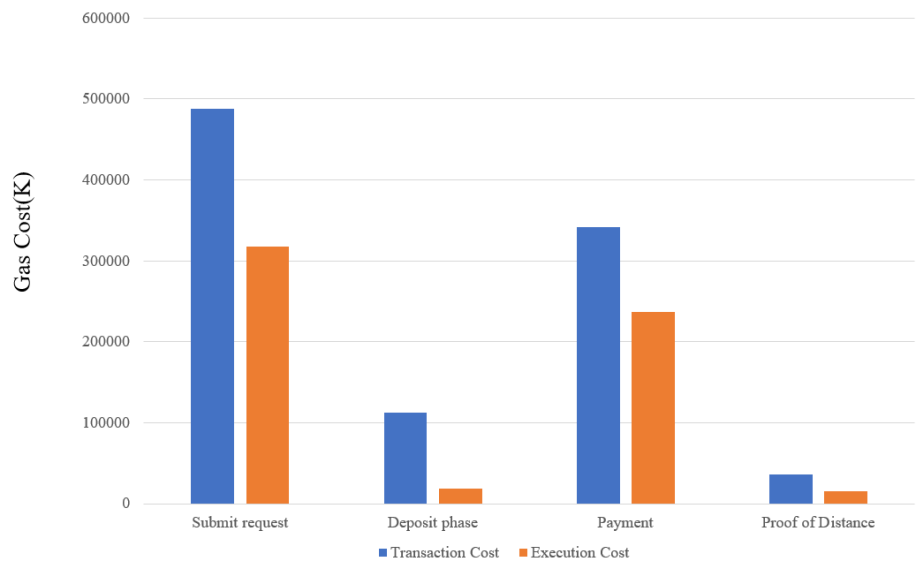

Fig. 6. Passenger's Gas Cost.

Table 2. Comparison of Various Taxi Models

	Architecture	Rider's privacy	Trust	Fair payment	Transparency	Select the driver's attributes
Current RSS	Centralized	×	✓	×	×	×
SRide	Centralized	✓	×	×	×	×
Co-utile	Decentralized	✓	×	×	×	×
DACSEE	Blockchain	×	×	×	✓	×
Arcade City	Blockchain	×	×	×	✓	×
B-Ride	Blockchain	✓	✓	✓	✓	×
Our mode	Blockchain	✓	✓	✓	✓	✓

7. CONCLUSIONS

In this paper, we propose a decentralized blockchain-based ride-hailing mode with attribute encryption. The blockchain and smart contract are applied to improve the transparency of system. The attribute encryption technology is used to make passengers can select the driver who they want. Passengers can choose the driver's attributes according to their needs. The Experiments and analysises are conducted to evaluate this mode. The results indicate that mode is practical and efficient.

ACKNOWLEDGMENT

This work was supported by The National Social Science Fund of China (No. 21XTQ015), and the Natural Science Foundation of Fujian Province of China (No.2020J01814). The authors also gratefully acknowledge the helpful comments and

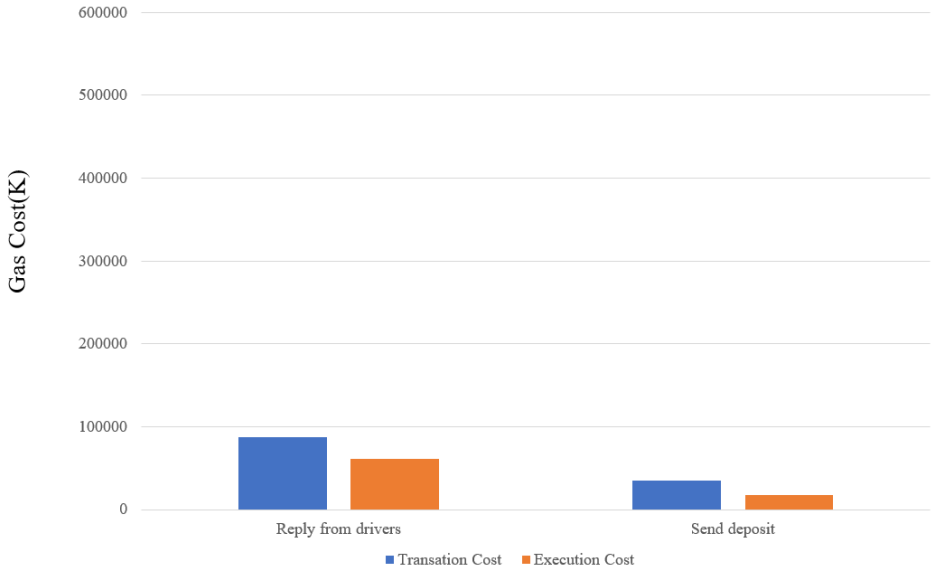

Fig. 7. Driver's Gas Cost.

suggestions of the reviewers, which have improved the presentation.

REFERENCES

1. Research and Markets, "Ride sharing market cap." 2019, accessed 2019. https://www.globenewswire.com/news-release/2019/01/17/1701096/0/en/218-Billion-Ride-Sharing-Market-Global-Forecast-to-2025.html.
2. M. Baza, M. Nabil, M. Ismail, M. Mahmoud, E. Serpedin, and M. A. Rahman, "Blockchain-based charging coordination mechanism for smart grid energy storage units," in *2019 IEEE international conference on blockchain (blockchain)*. IEEE, 2019, pp. 504–509.
3. J. Fullerton, "Uber's china problem," 2016, accessed May 25, 2016. https://www.vice.com/en/article/3daa55/ubers-china-problem.
4. M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 30, no. 6, 2018, pp. 1251–1266.
5. M. Baza, M. Nabil, N. Lasla, K. Fidan, M. Mahmoud, and M. Abdallah, "Blockchain-based firmware update scheme tailored for autonomous vehicles," in *2019 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2019, pp. 1–7.
6. A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 839–858.

7. W. Al Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmary, and K. Akkaya, "Privacy-preserving smart parking system using blockchain and private information retrieval," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2019, pp. 1–6.
8. U. M. Aïvodji, K. Huguenin, M.-J. Huguet, and M.-O. Killijian, "Sride: A privacy-preserving ridesharing system," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2018, pp. 40–50.
9. Y. He, J. Ni, X. Wang, B. Niu, F. Li, and X. Shen, "Privacy-preserving partner selection for ride-sharing services," *IEEE Transactions on Vehicular Technology*, Vol. 67, no. 7, 2018, pp. 5994–6005.
10. A. B. Sherif, K. Rabieh, M. M. Mahmoud, and X. Liang, "Privacy-preserving ride sharing scheme for autonomous vehicles in big data era," *IEEE Internet of Things Journal*, Vol. 4, no. 2, 2016, pp. 611–618.
11. Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE, 2016, pp. 2663–2668.
12. M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet of Things Journal*, Vol. 6, no. 3, 2018, pp. 4573–4584.
13. M. Baza, N. Lasla, M. M. Mahmoud, G. Srivastava, and M. Abdallah, "B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain," *IEEE Transactions on Network Science and Engineering*, Vol. 8, no. 2, 2019, pp. 1214–1229.
14. M. Baza, M. Mahmoud, G. Srivastava, W. Alasmary, and M. Younis, "A light blockchain-powered privacy-preserving organization scheme for ride sharing services," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–6.
15. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, 2008, p. 21260.
16. I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 397–411.
17. E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE symposium on security and privacy*. IEEE, 2014, pp. 459–474.
18. D. Schwartz, N. Youngs, A. Britto *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, Vol. 5, no. 8, 2014, p. 151.
19. V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, Vol. 3, no. 37, 2014.
20. A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2005, pp. 457–473.
21. Y. Zhou, L. Jiang, H. Huang, J. Chen, and Z. Huang, "Telemedicine system with privacy protection based on cp-abe," *INVESTIGACION CLINICA*, Vol. 60, no. 6, 2019, pp. 1615–1625.

22. H. H. Yu Hu, Yi-Fan Zhang and Y.-P. Zhou, "Attribute-based message recovery designated verifier proxy signature scheme in telemedicine system," *Journal of Network Intelligence*, Vol. 7, no. 1, 2022, pp. 101–113.
23. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007, pp. 321–334.
24. L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Ciphertext-policy attribute-based threshold decryption with flexible delegation and revocation of user attributes," *IEEE Trans. Image process*, 2009.
25. L. Cheng, J. Liu, G. Xu, Z. Zhang, H. Wang, H.-N. Dai, Y. Wu, and W. Wang, "Sctsc: A semicentralized traffic signal control mode with attribute-based blockchain in iovs," *IEEE Transactions on Computational Social Systems*, Vol. 6, no. 6, 2019, pp. 1373–1385.
26. E. Morais, T. Koens, C. Van Wijk, and A. Koren, "A survey on zero knowledge range proofs and applications," *SN Applied Sciences*, Vol. 1, no. 8, 2019, pp. 1–17.
27. J. Camenisch, R. Chaabouni *et al.*, "Efficient protocols for set membership and range proofs," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2008, pp. 234–252.

Yifan Zhang received his B.S. degrees in Northwest University, Shannxi , Xi'an, China, in 2017. He is currently a M.S. candidate at Minnan Normal University. His research interests include Blockchain and Internet of vehicles.

Yuping Zhou is a professor at the school of computer science, Minnan Normal University. She is received the Ph.D. degree in control theory and control engineering from Donghua University in China. Her research interest fields include information security and data mining. She publish more than 30 papers published in various journals. She has teaching experience of 23 years, has completed more than 10 scientific research projects.

Yu Hu received his B.S. degrees in Anqing Normal University, Anhui , Anqing, China, in 2018. He is currently a M.S. candidate at Minnan Normal University. His research interests include Blockchain and attribute signature.

Xinyu Yi received his B.S. degrees in Changsha University, Hunan , Changsha, China, in 2020. He is currently a M.S. candidate at Minnan Normal University. His research interests in Blockchain.

Ben Xie received his B.S. degrees in Hefei University of Economics, Anhui , Hefei, China, in 2021. He is currently a M.S. candidate at Minnan Normal University. His research interests in Blockchain.