

Onionchain: Towards Balancing Privacy and Traceability of Blockchain-Based Applications

Yue Zhang, Jian Weng, *Member, IEEE* Jiasi Weng, Ming Li, and Weiqi Luo,

Abstract—With the popularity of Blockchain comes grave security related concerns. Achieving privacy and traceability simultaneously remains an open question. Efforts have been made to address the issues, while they may subject to specific scenarios. This paper studies how to provide a more general solution for this open question. Concretely, we propose Onionchain, featuring a suite of protocols, offering both traceability and privacy. As the term implies, our Onionchain is inspired by Onion routing. We investigate the principles of Onion routing carefully and integrate its mechanism together with Blockchain technology. We advocate the Blockchain community to adopt Onionchain with the regards of privacy and traceability. To this end, a case-study of Onionchain, which runs in the context of Vehicular Ad Hoc Networks (VANETs), is proposed, providing the community a guideline to follow. Systematic security analysis and extensive experiments are also conducted to validate our secure and cost-effective Onionchain.

Index Terms—Blockchain, Traceability, Privacy, Vehicles Communication System, Onion Routing.

1 INTRODUCTION

OVER the past few years, Blockchain has drawn significant attention from both academy and industry [1]. Blockchain is a novel paradigm where distrustful parties make transactions and manage data without involving a trustworthy third-party. Here transactions refer to interactions occurred between these parties. Blockchain achieves tamper-resistance and traceability for the transactions, offering anonymity and decentralization for the parties. Due to these advanced features, Blockchain can be applied into a wide spectrum of applications, ranging from cryptocurrency, financial services, crowd-sourcing systems [2], [3], and Vehicular Ad Hoc Networks (VANETs) [4], [5]. According to a report from Meticulous Research, the global Blockchain market will hit \$28 billion by 2025 [6].

However, along with its popularity, Blockchain has come an increasing number of attacks, severely undermining the victim's security and privacy. In regards to privacy, although Blockchain can provide anonymity innately, it subjects to various cyber-attacks. For example, Fergal *et al.* [7] show that an attacker may disclosure the real identity of a given victim by analyzing his public transaction history. Efforts have been made to counter these attacks. An example that addresses the privacy concern is HAWK proposed by Kosba *et al.*, storing the encrypted transactions instead of plain-text ones so that the transactional privacy is guaranteed [8]. The term privacy in our paper refers to data privacy unless explicitly stated otherwise. In data privacy schemes, the identities of parties are public. However, given a message,

no efficient adversary can determine if the message is from a specific party.

While achieving privacy, these solutions may fail to provide traceability, hindering them to adapt to some scenarios. For example, in crowd-sourcing systems (or crowd-sensing system in VANETs [9]), employers may release tasks for employees, while employees choose the tasks of interest and get paid when they offer proper solutions. During the life-cycle of a task, Blockchain endorses the behaviors of both employers and employees, so that employers/employees can be held accountable when malicious behaviors occur. Offering the privacy blindly in such a scenario will lead a malicious employers/employees to evade responsibilities.

The fact that Blockchain fails to provide traceability and privacy simultaneously impedes the progress of its deployment. To address the issue, there are a few existing works. For example, the primitive, linkable group/ring signatures [10], [11], are possible solutions for this issue. In these manners, a signature can be generated anonymously to present the willing of the parties in this group, and the group manager can reveal the identity/identities that perform the signature generation. However, the drawback is also obvious, since the group manager can be malicious, leading the abuse of the identity disclosure. Lu *et al.* propose ZebraLancer [12], where a new primitive common-prefix-linkable is presented to realize a trade-off between traceability and anonymity. In the context of crowdsourcing, a malicious employee can submit his effort twice, to gain a doubled reward. ZebraLancer is capable of identifying the double-submission problem and tracing the dishonest party. However, ZebraLancer is not omnipotent, since it fails to discuss how to deploy their protocol in other scenarios where there is no such malicious behavior like double-submission. Liu *et al.* [13] achieve the traceability and privacy simultaneously by introducing an identity management entity (IDM) and let IDM recover the identity of the misbehaving party. While the problem solved, their design partly violates the discretization of Blockchain. Other

- Yue Zhang, Jian Weng, Jiasi Weng, Ming Li and Weiqi Luo are with the College of Informatin Science and Technology / College of Cyber security, National Joint Engineering Research Center of Network Security Detection and Protection Technology, and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, 510632, China;
- Jian Weng is the corresponding author : cryptjweng@gmail.com ;

Manuscript received XX,XXXX; revised XX,XXXX

solutions, such as [14], may also subject to specific scenarios. Therefore, balancing the traceability and privacy simultaneously remains an open question.

This paper tries to provide a general solution for this issue. To this end, we propose Onionchain, offering traceability and privacy at the same time. The intuition behind our research is that we observed that Onion routing can achieve traceability and privacy in different contexts. Onion routing [15] is an infrastructure designed for anonymous communication. It uses a set of onion routers instead of using regular routers. The onion routers encrypt and relay packets between a source node and a destination node. In terms of privacy, onion routers resistant to both eavesdropping and traffic analysis innately, since the encryption process is present. In terms of traceability, Onion routing is a communication mechanism featuring routing and packets transmitting. In the view of the onion routers, they can identify an intended source and an intended destination by decrypting the encrypted packets accordingly. But, for a single onion router, it has limited routing information and provides packets forwarding partly. Therefore, for most of the time, as a destination or an onion router, they cannot trace the packets back to the source. This is because tracing a packet needs the efforts of all relaying onion routers, but the onion routers will not work cooperatively to make this happen since they are different nodes in the network. [15] Inspired by these two features, we systematically investigate the underlying principles of the Onion routing and design Onionchain.

The high-level idea is that we introduce Blockchain as a trustworthy party, and enable the onion router-like nodes to decrypt the packets to Blockchain conditionally. That is, Blockchain will perform the identity disclosure according to the willing of majority. In such a way, Onionchain achieves traceability and privacy simultaneously: Normally, Onionchain will offer privacy for each party. In the special case where disclosure is needed, parties will work closely to make the disclosure happen. To notice, our Onionchain integrate the mechanism of Onion routing together with Blockchain technology instead of using Onion routing as a building block directly. Using Onion routing with no change is fairly trivial. More importantly, while achieving privacy, Using onion routing barely fails to provide a proper solution for nodes to work jointly to disclose a specific identity [16].

We advocate the community to adopt Onionchain when Blockchain applications require achieving both of traceability and privacy. As a case study, we deploy our Onionchain in the case of VANETs. In such a context, vehicles by sharing the information of the road environment work jointly to prevent traffic jams and accidents. Therefore, it is crucial for vehicles to obtain information without any errors in a timely manner. Blockchain provides a decentralized data sharing environment for VANETs, as shown in prior efforts [17], [18], [19], offering privacy-preserving for vehicles. However, a malicious vehicle may poke the pitfalls of VANETs in various ways, spreading false information intentionally, misleading other vehicles. One example is Blockhole attack [20], where a malicious vehicle pretends to be another benign vehicle, and discard any packets that flow toward it, causing severe packet loss. In such a context, providing privacy without traceability, such

as what the traditional Blockchain-based solutions have done, is not enough. As a countermeasure, we introduce our Onionchain to guide behaviors of vehicles. Our Onionchain can provide the privacy-preserving for a benign vehicle, and identify the malicious vehicle when an attack occurs.

Major contributions can be summarized as follows:

- 1) We design Onionchain, featuring a suite of customized protocols and algorithms, achieving traceability and privacy simultaneously.
- 2) We demonstrate our Onionchain in the context of vehicles communication systems, avoiding vehicles to spread false information or discard valuable information intentionally. For the sake of generality, the demonstration in our paper can be trivially extended to other similar scenarios.
- 3) We validate our Onionchain and the proposed countermeasure for VANETs. We prototype the countermeasure on Ethereum. Excessive experiments are performed to evaluate our implementation, showing the overload is fairly acceptable.

The rest of our paper is organized as follows. In section 2, we introduce primitives involved in our paper, including Blockchain, smart contracts, Onion routing, and VANETs. In Section 3, we present the security model of Onionchain. In Section 4, we elaborate on the design criteria of Onionchain and deploy it in the context of the vehicles communication system. We systematically analyze the security of Onionchain in Section 5 and evaluate Onionchain experimentally in Section 6. Related works are reviewed in Section 7. We conclude the paper in Section 8.

2 BACKGROUND

In this section, we first provide a brief introduction of Blockchain and smart contracts technology. Afterward, we elaborate an overview of onion routing to demonstrate the principles behind. We also show the architecture of Vehicular Ad Hoc Networks (VANETs).

2.1 Blockchain

Blockchain was first proposed by Nakamoto in [21], as the fundamental primitive of decentralized digital currency, Bitcoin. In the context of Bitcoin, Blockchain is a distributed public ledger, which is initially designed for mutually distrustful parties to make transactions without involving a trustworthy center. To achieve this, each party needs to maintain the entire copy of Blockchain and works cooperatively to record the transactions. The transactions are organized as a data block, which is composed of a block header and a block body. Basically, the transactions are stored in the block body, while the block header contains a root hash and a reference. The root hash is the hash of a Merkle Tree [22] of all transactions in the block body. The reference, which is also a hash value, is computed from the block header of the previous block.

Each party has an identity, termed address, which uniquely refers to a specific party. To obtain an identity, each party needs to generate a pair of public/private key. The private key is used for signature verification, and the public key is used for signature generation and builds the

address. To notice, not all parties can record transactions on Blockchain. The one who has the priority to append data, termed miner, needs to provide a solution to a certain puzzle, known as proof-of-work ((PoW) [23] or proof-of-stake (PoS) [24] etc. For example, the PoW is a hash computing puzzle that can only be solved when an appropriate answer is fed into the hash function. Due to the property of the hash function, the puzzle requires much efforts to find out such an answer, while the validity of the answer can be trivially checked.

Blockchain records all transactions by default, and it achieves tamper-resistance and traceability innately. The main reasons are: (i) For a single block, each block maintains the hash of a Merkle Tree that computed from transactions in the block body, so that no more modifications can be made once the block is generated, since changes will make the hash root of Merkle Tree fail to match the original one; (ii) For the entire Blockchain, Blockchain is maintained by each party, so that it is impossible for a single party to modify it, since modifying the Blockchain of himself will not make any influences to other parties. Moreover, due to the reference to the previous block, the integrity of the entire Blockchain is also guaranteed, since each party can verify it chronologically without any changes; (iii) The only way to make a modification happen, i.e., adding a new transaction to Blockchain, may require a lot of efforts since the one who wants to do so must solve the puzzle first.

2.2 Smart Contracts

Smart contracts are digital agreements that are made between different parties. The term was first introduced in 1994 by Nick Szabo [25], and gained many attentions with the booming of Blockchain. In smart contracts, instead of using the printed contracts, and enforcing to execute via a suite of regulations or laws, smart contracts are grouped into lines of codes, and will be executed when specific conditions are fulfilled. To ensure the proper execution of the codes, a secure environment must be guaranteed. Otherwise, an attacker may compromise the codes to maximize his benefits. This was a giant barrier that hinders the development of smart contracts. However, as mentioned earlier, Blockchain contributes a lot to the popularity of smart contracts, since decentralized Blockchain networks and other advanced features of Blockchain can offer such a secure environment for smart contracts. Examples of Blockchain that run smart contracts include Ethereum [26] and Hyperledger [27].

2.3 Onion Routing

Onion Routing [15] is an infrastructure for anonymous communication. Instead of creating socket connections between the two communicating machines directly, onion routing achieves the communication via a sequence of machines termed onion routers. Onion routers relay the two communicating machines, offering routing and addressing, allowing the two communicating machines to remain anonymous in one way or both ways. One application of onion routing is anonymous web browsing. In this case, a user may want to browse a website without disclosing his identity to the webserver. That is, anything related to him,

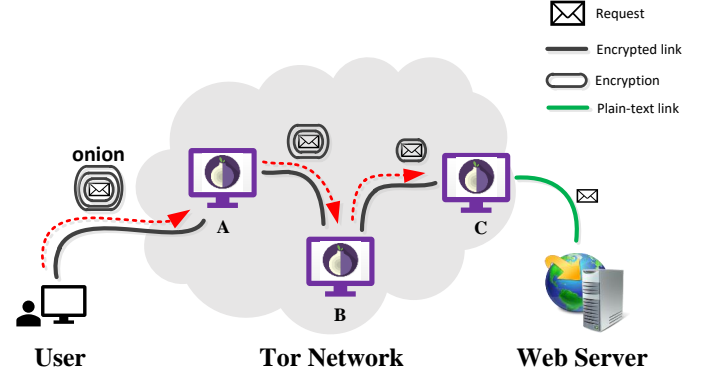


Fig. 1. The principle of onion routing

such as IP address or MAC address, shall be removed from the browsing request.

To demonstrate the principle of onion routing, Fig. 1 illustrates the process of anonymous web browsing. Specifically, when the user wants to browse a website anonymously, he must follow the steps elaborated below:

- 1) Initially, the user uses an onion proxy to randomly choose three onion routers (A, B, C in the figure) in the network and negotiate three different keys with the three onion routers. We denote the encryption keys as Key_{User-A} , Key_{A-B} , Key_{B-C} , where the first letter in subscript refers to the sender of the packet, while the second letter refers to the responder of the packet. To notice, we are aware that the public cryptography and other technologies/roles are also involved in this process, and we will not discuss the key negotiation in detail for the sake of brevity.
- 2) To achieve anonymous communication, the user encrypts the browsing request with the encryption keys successively. The user also hard-code the next-hop information inside the encrypted browsing request, to make sure that each onion router is aware of its next hop. The encrypted browsing request is called the onion.
- 3) The user sends the onion to the first onion router A . The first onion router removes the first layer encryption by decrypting the onion with the key Key_{User-A} , which is negotiated previously. Once decrypted, the onion router then knows which onion router is the one it should send the onion to. In our context, the first onion router will send the onion to B .
- 4) B receives the onion and removes the second layer encryption by decrypting the onion with the key Key_{A-B} , so that it obtains its next-hop information. The second onion router then sends the onion to C .
- 5) C removes the last layer of the onion by decrypting the onion with the key Key_{B-C} , and sends the browsing request, which is in plain-text, to the intended web server. The three onion routers then communicate with each other and work jointly to relay the packets between the user and the webserver.

Since the links between each other are encrypted, each router may only obtain limited information about the connection. For example, for the onion router A , it only knows

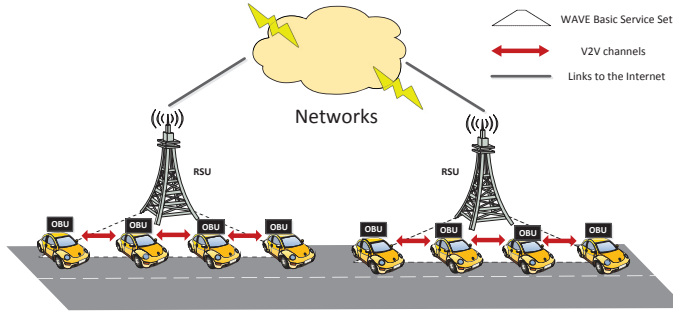


Fig. 2. Overview of vehicles communication system

who is the sender of the packets and who is its next hop. For the onion router *C*, it only knows the information about the webserver and its previous hop. For the onion router *B*, it neither knows the information about the user nor the information about the webserver. The only information he obtained is the hop information. In such a way, onion routing achieves anonymous communication.

2.4 Vehicular Ad Hoc Networks (VANETs)

Vehicular Ad Hoc Networks (VANETs) [28], [29], [30], as the term itself implies, is designed for vehicles to achieve communication. It has aroused researchers interest since it offers a paradigm for facilitating smart city. In vehicles communication systems, vehicles are equipped with a set of sensors, Global Positioning System (GPS) [31], and Radio Frequency Identification (RFID) tags [32], which allows them to sense the environment around. Examples of the applications of vehicles communication systems include navigation, path planning. For instance, in the case of path planning, vehicles may communicate with each other to share the traffic information, so that a suitable route can be selected, avoiding the traffic jams [33].

Fig. 2 shows an overview of the vehicles communication system. Instead of using traditional Wireless Local Area Network (WLAN) and the wired network, vehicles communication system introduce Road-Side Units (RSUs) to connect the internet. These RSUs are installed along the road-sides, providing Internet access for vehicles. The coverage of each RSU is termed Wireless Advanced Vehicle Electrification (WAVE) Basic Service Set. Basically, vehicles connect to the RSUs via their On-Board Units (OBU), which is a component mounted inside each vehicle. When there is no RSU available, the vehicles can also achieve local area communication.

3 SECURITY MODEL

In this section, we first present the security goals of Onionchain. We will also make the basic assumptions for our design afterwards.

3.1 Security Goals

We claim the ultimate goals of Onionchain. It can be observed from the introduction and the principle of Blockchain that the current design of Blockchain applications fails to

achieve privacy and traceability simultaneously. Therefore, Onionchain has the following essential design requirements:

Privacy: First of all, the privacy issue is our main concern. The term privacy refers to data privacy, where identities of parties are public, while there is no efficient adversary can determine if a given message is sent from a specific party. That is, as long as a party behave honestly, no body can link the message he sent to his public identity. The identity may include IP address, Mac address, or other personal information. Onionchain will provide a counter-measure for these parties, against their privacy being leaked.

Traceability: Second, with regard to traceability, Onionchain will disclose a specific identity of a party following the willing of the majority. That is, when a dishonest/malicious behavior occurs, the majority of parties may require the Onionchain to reveal the identity of a party, and Onionchain will fulfill the requirement.

3.2 Assumptions

In our system, an attacker's goal is to reveal the identities of parties and pokes the design flaws of our Onionchain. Without loss of generality, we make the following assumptions for attackers and our system.

- 1) In our design, cryptographic mechanisms will be involved, and we assume these cryptographic mechanisms are not a source of vulnerability. For example, an attacker cannot break the AES encryption or forge a digital signature.
- 2) We assume that the fundamental assumption of Onion routers still holds. That is, the nodes in the network are randomly distrusted. Therefore, when a party chooses nodes from network randomly, the chosen nodes are not all manipulated by an attacker.
- 3) We assume that an attacker can access all the transaction history without any change. This is reasonable since Blockchain is a public ledger that is free to download for anyone. This assumption offers the worst scenario for our design. Our design can hinder the privacy leakage in this scenario, and it also hinders other scenarios.
- 4) We assume that a large number of parties are involved in our system. This is reasonable since systems, such as vehicles communication systems, may involve hundreds of participants.
- 5) A malicious party may spread false message intentionally, misleading other parties in various ways. We assume that after a message being spread, other parties have ways to identify whether the information is a false one. For example, false information can be fake traffic information on a specific road, which can be trivially identified when a vehicle arrivals there. However, more ways of identifying false information are out of our focus, and we will not discuss it.

4 ONIONCHAIN

We first explain our insights behind our design. From the introduction of onion routing in Section 2.3, we learn that onion routing resists to both eavesdropping and traffic analysis innately. We also observe that the onion routers

will relay the packets between an intended source and an intended destination. For most of the time, as a destination, it cannot trace the packets back to the source. This is because that the destination and its previous hop, which is an onion router, are different nodes in the network. They shall not share the hop or routing information with each other. But what if the destination node and the three onion routers are all manipulated by a third-party behind? In this case, the third-party can obtain all the routing information, and trace from a destination back to a source. That is, he is capable of disclosing the identity of the source. Here, such a third party can be a good or evil, depending on if and how he will perform the disclosure. If he is ultimately fair and only behaves according to the willing of majority, he is a good one. However, does such a flawless third party exist? Actually, Blockchain can be such a third party. Motivated by this, we have designed a suite of protocols, termed Onionchain which make Blockchain to be such a third party. In the following of this section, we will present the criteria of our design in detail.

4.1 Overview

Onionchain offers three core protocols for parties to achieve privacy-preserving and tracing transactions, including the registration, message transmitting, and identity disclosure. Registration is used for parties to join Onionchain. To this end, each party must provide their real identity to Onionchain distributedly, and Onionchain will store these information onto Blockchain. All these information are free to access for the public. In regard of privacy-preserving, although the information is publicly accessible, an attacker cannot link a specific message to its sender according to our design. The privacy of each party is still preserved. Message transmitting is a protocol that defines how two parties transmit a message via the network. Different from the traditional network communication, the message sender and the message responder may need to write data onto Blockchain. The data, termed evidence in our case, is encrypted by negotiated keys, and also plays an important role in the identity disclosure protocol. The disclosure protocol will perform when the parties have the requirement to disclose a specific sender. Say, when false information is identified, the parties want to know who is the sender of this false information. In this case, our identity disclosure protocol can link the false information to a specific sender by decrypting the evidence.

4.2 Design Criteria

In this section, we will elaborate on the design criteria of registration protocol, message transmitting protocol, and identity disclosure protocol. For ease of description, we summarize the basic operations in Table 1.

Registration protocol: This protocol is used for parties to sign up in our system. It works as follows: (i) When a party A wants to join the system, he first creates a public/private key pair, denoted as SK_A , $PubK_A$, which can be used to perform signature generation and verification. (ii) He uses his private key SK_A to sign his real identity ID , and put his public key $PubK_A$ and the generated signature S together to generate a registration request, i.e. $regReq =$

TABLE 1
Summary of notations

Notation	Description
$a \rightarrow b$	a routing message, where a is the source and b is the destination
$PubK_i$	the public key of a party i
SK_i	the private key of a party i
$s_1 s_2$	a combination of a string s_1 and a string s_2
$ENC(key, msg)$	The AES encryption process, where key refers to a encryption key, and msg refers to a message
$DEC(key, ctext)$	The decryption process, where key refers to a encryption key, and $ctext$ refers to a cipher-text
$SIGN(SK, msg)$	The signature generation process, where SK refers to a private key, and msg refers to a message
$VERIFY(PubK, sig, m)$	The signature generation process, where $PubK$ refers to a private key, and sig refers to a signature, while m is the message

$(PubK_A || S)$. The ID here uniquely refers to a specific party. He submits the registration request to the Blockchain P2P network and preserves his private key carefully. To notice, before the request having been approved and written onto Blockchain, A is not permitted to send any other request. (iii) Other parties will verify the registration request before they write it onto Blockchain. The verification process is to make sure the signature is generated from the attached public key $PubK_A$, and the one who sends the request has the same identity as attached. Otherwise, other parties will reject the request. The other parties also need to check if the public key attached has been used by other parties else. Since the request can be viewed by all the parties, it is trivial for them to identify a duplicate public key. That is, each party compares the attached public key with his own public key and confirms that they are not the same. If they are the same, the party with the same public key will broadcast a confliction. The confliction is also a type of request that has a higher priority for other parties to process. Also, other parties may need to check if the identity used is a valid one or fabricated by the party itself. Therefore, the Blockchain used in our paper is a permissioned blockchain. The permissioned blockchain is a type of Blockchain that requires permission to join, and limits the parties who can provide the solution for the puzzle, i.e., being the miners. Recall that all parties other than miners can submit their transactions into Blockchain network, but only miners have the permission to record the transactions. This is reasonable because it reduces the risk of being attacked by some attacks, e.g., 51% attack and selfish mining attack. The process of how to group the requests as transitions, and how to write the transactions are as same as Bitcoin, and we will not go to details due to the page limitation.

Message transmitting: This protocol is the core engine of our design. As shown in Fig. 3, we use three hops as relays between a transmitter T and a receiver R to demonstrate our principle. Using three hops as relays are the minimum

requirement of our system. However, parties can choose more than three hops to achieve better privacy. Specifically, we elaborate on the steps below:

- 1) Initially, the sender who wants to send the message first randomly chooses three nodes in the Blockchain P2P network. The three nodes are denoted as A, B, C respectively. The sender also negotiates the three different keys with the three nodes. The key negotiation process is as same as that in onion routing, so we will not repeat it due to the page limitation. We denote the encryption keys as $K_{T-A}, K_{A-B}, K_{B-C}$, where the first letter in subscript refers to the sender of the packet, while the second letter refers to the packets responder. For example, K_{T-A} is used to encrypt a message that sent from T to A .
- 2) T encrypts the message with the three encryption key successively. The user also hard-code the next-hop information, e.g., $A \rightarrow B$, inside the encrypted message, to make sure that each node is aware of its next-hop respectively. As mentioned earlier, the encrypted message is called evidence in our paper. To notice, during the message transmitting, there is more than one evidence are generated. We denote this evidence EV_0 . We also denote the message as m , which contains a timestamp to grantee the freshness of messages. Formally, EV_0 can be represented as equation (1) :

$$EV_0 = \text{ENC}(K_{T-A}, (A \rightarrow B \parallel \text{ENC}(K_{A-B}, (B \rightarrow C \parallel \text{ENC}(K_{B-C}, (C \rightarrow R \parallel m)))))) \quad (1)$$

- 3) A first removes the first layer encryption by decrypting the EV_0 with the key K_{T-A} , which is negotiated early. We denote the decrypted onion-like packets as V_0 . Formally, V_0 can be represented as equation (2) :

$$V_0 = \text{ENC}(K_{A-B}, (B \rightarrow C \parallel \text{ENC}(K_{B-C}, (C \rightarrow R \parallel m)))) \quad (2)$$

Once decrypted, A then knows which node is the next hop. In our context, A will send V_0 packet to B , and the key used between A and B is K_{A-B} . Before this process, A also needs to generate a new evidence and submit it to Blockchain first. Here, A and T are required to work closely, and generate a the new evidence EV_1 . To this end, T signs EV_0 with his private key, and sends $EV_0 \parallel \text{SIGN}(SK_T, EV_0)$ to the first node A . The first node verifies the signature using the public key of T . Recall the public key was written onto Blockchain in the registration process and free to index. This step is used to ensure the message is sent from T , not other parties. Once the verification processed, A signs $\text{SIGN}(SK_A, \text{SIGN}(SK_T, EV_0))$ with his own private key SK_A . Thereafter, the two parties negotiate a new key PK_{T-A} , termed proof key, then encrypts $\text{SIGN}(SK_A, \text{SIGN}(SK_T, EV_0))$ with PK_{T-A} to generate the new evidence EV_1 . This step ensures that A receives the message successfully. The EV_1 are suppose to be written on Blockchain. Here, T and A have the same proof key PK_{T-A} , so that they can check the signatures and confirms that they all follow the protocol properly, while other parties will have no information about EV_1 . To notice, the K_{T-A} and

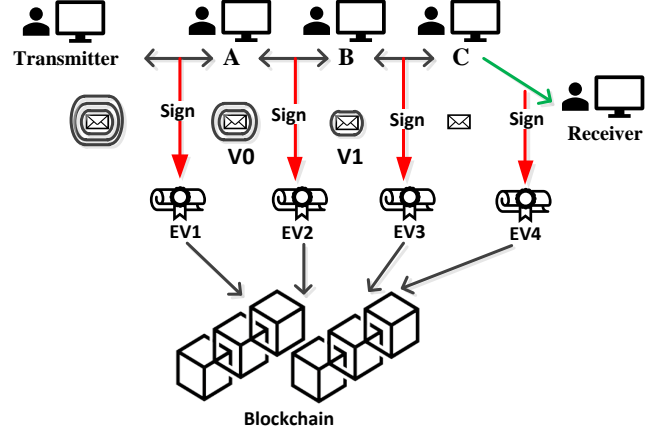


Fig. 3. Overview of message transmitting protocol

PK_{T-A} can not be the same, and each party is required to keep the PK carefully for the further usage. Also, every time when a new message is sent, each party needs to negotiate a new proof key. Formally, EV_1 can be represented as equation (3):

$$EV_1 = \text{ENC}(PK_{T-A}, \text{SIGN}(SK_A, (\text{SIGN}(SK_T, EV_0)))) \quad (3)$$

- 4) A will wait until EV_1 is written onto Blockchain. Thereafter, A will send V_0 packet to B . B then removes the second layer encryption with the key K_{A-B} and generate the onion-like packets as V_1 . Formally, V_1 can be represented as equation (4):

$$V_1 = \text{ENC}(K_{B-C}, (C \rightarrow R \parallel m)) \quad (4)$$

Afterwards, A and B work closely to generate a new evidence EV_2 . The process is similar to what we have elaborated in previous step, and we will not repeat it. Specifically, EV_2 can be represented as equation (5):

$$EV_2 = \text{ENC}(PK_{A-B}, \text{SIGN}(SK_B, \text{SIGN}(SK_A, V_0 \parallel EV_1))) \quad (5)$$

- 5) B will wait until EV_2 is written onto Blockchain. Thereafter, B will send V_1 to C . C then removes the finally layer encryption with the key K_{B-C} and obtains the message m in plain-text. At this time, B and C work closely to generate a new evidence EV_3 . Specifically, EV_3 can be represented as shown in equation (6):

$$EV_3 = \text{ENC}(PK_{B-C}, \text{SIGN}(SK_C, \text{SIGN}(SK_B, V_1 \parallel EV_2))) \quad (6)$$

- 6) C will wait until EV_3 is written onto Blockchain. Afterwards, C will send message to R . R and C then work closely to generate a new evidence EV_4 . Specifically, EV_4 can be represented as equation (7):

$$EV_4 = \text{ENC}(PK_{C-R}, \text{SIGN}(SK_R, \text{SIGN}(SK_C, m \parallel EV_3))) \quad (7)$$

Identity disclosure:

Identity disclosure is available when a dishonest behavior has been detected. As shown in algorithm. 1, we elaborate the entire process of Identity disclosure. Basically, it is

the reverse process of message transmitting. To perform an identity disclosure, the parties that involved in the message transmitting may need to decrypt the evidence using the proof key. We termed the decryption process of each party as “plea of innocence” since each party uses this process to prove that he is innocent. Suppose that a receiver R identifies a message m being a false message and wants to find the transmitter T . Once the identity disclosure request has been approved by the majority, the identity disclosure process will be launched. Recall that we assume that parties have the abilities to identify false information. A request can be approved only when most parties has identified the information is a false one. As shown in Fig. 4, we use the former example to elaborate on the identity disclose process below:

Algorithm 1: Identity disclosure protocol

Data: m_{fake}, EV_i
Output: R_x
 $j = i - 1;$
 $V_j = m_{fake};$
for $j > 0$ **do**
 Relaying node R_{j-1} **release his proof key**
 $PK_{R_{j-1}}:$
 if $PK_{R_{j-1}} == \emptyset$ **then**
 $R_x = R_{j-1};$
 RETURN $R_x;$
 end
 else
 $DEC(PK_{j-1}, EV_j)$
 $= SIGN(SK_{R_{j-2}}, SIGN(SK_{R_{j-1}} EV_{j-1} || V_j));$
 $= S_j$
 if $VERIFY(PubK_{j-1}, S_j, (V_j || EV_{j-1}))$ **then**
 if $VERIFY(PubK_{j-2}, S_j, (V_j || EV_{j-1}))$ **then**
 $j=j-1;$
 end
 else
 $R_x = R_{j-2};$
 RETURN $R_x;$
 end
 end
 else
 $R_x = R_{j-1};$
 RETURN $R_x;$
 end
 end
 end
end
 $R_x = R_j;$
RETURN $R_x;$

- 1) The receiver R requires a party C to perform the plea of innocence since R receives the false message from C . To prove so, R locates evidence EV_4 on Blockchain, and makes the location of EV_4 and the PK_{C-R} publicly accessible. In this way, all parties in the Blockchain P2P network can perform the decryption process, as show in equation (8).

$$\begin{aligned} DEC(PK_{C-R}, EV_4) &= SIGN(SK_R, \\ SIGN(SK_C, m || EV_3)) &= S_4 \end{aligned} \quad (8)$$

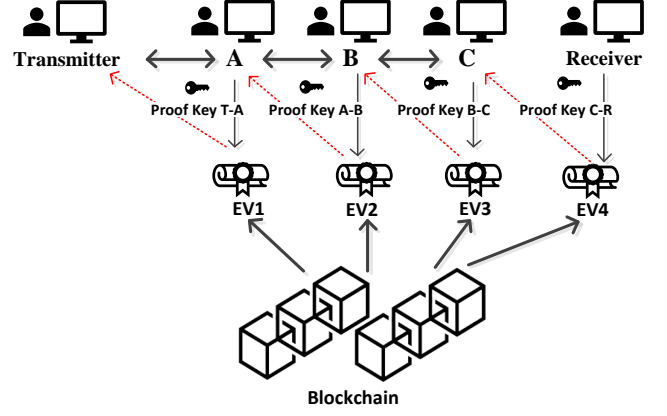


Fig. 4. Overview of identity disclosure protocol

After the decryption process, all parties know that the previous hop is C , which was confirmed by C and R , since their signature are present. As shown in equation 9, all parties can perform their signature verification without any changes¹.

$$VERIFY(PubK_C, S_4, (m || EV_3)) \quad (9)$$

- 2) In this case, C is required to perform the plea of innocence. Different from the first step, the evidence EV_3 is contained in the plain-text of EV_4 . So C is only required to make the PK_{B-C} publicly accessible. In such a way, parties know that previous hop is B , which was confirmed by B and C , since their signature are present. Meanwhile, new evidence EV_2 shows up.
- 3) Thereafter, B is required to perform the plea of innocence. Similar to the previous process, B finally reveals a piece of new evidence EV_1 .
- 4) Afterwards, A is required to perform the plea of innocence. The process is also similar to the previous one. Finally, a piece of new evidence EV_0 shows up.
- 5) Finally, T is required to perform the plea of innocence. A is required to make a proof key publicly accessible. However, T does not have a proof key for EV_0 , since T is the message transmitter. Therefore, T is accused of false message spreading. In this case, T can also release the three keys, which is considered as a “confession”.

4.3 Onionchain Based Vehicular Ad Hoc Networks

We then introduce the work-flow of our Onionchain-based Vehicular Ad Hoc Networks (OVANETs), avoiding vehicles to spread false information intentionally. The motivation of this section is to demonstrate the generality of our Onionchain.

From a high level, OVANETs is built on a Blockchain-based reputation system [34], [35]. In such a reputation system, vehicles share useful information, such as traffic information or information about road conditions, to gain their reputations. They are self-motivated by a suite of incentive mechanism. For example, vehicles with higher reputation may have more priority to access resources, or

1. The verification process is similar to that of R , and we will not repeat it.

they can use reputations to earn their gas, while a vehicle with a lower reputation may fail to access some services. Therefore, each vehicle values its reputation and tries to have a higher reputation by behaving honestly and offering good service for other vehicles. Basically, a Blockchain-based database provides endorsements for our system. The endorsements are tamper-proof due to the advanced features of Blockchain.

In such a scenario, privacy-preserving is an indispensable requirement [36]. Vehicles may require to remain anonymous for various reasons. For example, vehicles may be easy to be convinced by a message from a vehicle with higher reputation, while they may fail to believe information from a vehicle with a low reputation. In this case, the vehicles with a low reputation, like a vehicle which has just joined the system, may never have a chance to earn its reputation. Therefore, Onionchain offers opportunities for vehicles with lower reputations due to our privacy-preserving. On the other hand, Onionchain can also work inversely and identifies these vehicles spreading false information. Once identified, punishments are enforced for these dishonest vehicles. One punishment can be decreasing their reputations. Finally, considering the limited storage resource of vehicles, the designer can choose RSUs to deploy the Onionchain, while vehicles can communicate with RSUs via VANET.

It can be observed that our Onionchain can be extended to other similar scenarios with a little hindrance. For example, in the context of crowd-sourcing systems, employers/employees can also set up such an Onionchain based system to achieve privacy and traceability simultaneously. They may also use the idea of a reputation system to evaluate an employer/employee, and penalize the malicious parties when un-honest behaviors are detected. Moreover, by using the smart contracts introduced in section 2.2, the entire process may be executed automatically without the human's involvement, reducing the burdens of management.

5 SECURITY ANALYSIS

In this section, we will give the security analysis of our Onionchain. Specifically, an attacker may intentionally create a craft-packets and try to deploy attacks on our Onionchain in various ways. Notably, we examine the transmitter, relaying nodes, receiver throughout the life-cycle of a transaction and conduct five attack vectors in the regards of security analysis, as shown in Table 2. It can be observed that our Onionchain can hinder all the attack vectors without any change.

Malicious-Transmitter Attack: In this type of attack, a malicious transmitter may create a false message intentionally, and uses another message, which is considered benign, to generate shreds of evidence. We assume that the false message is m_{fake} , and the benign message is m . His motivation is to evade responsibility when the false message is detected. However, it is not possible for attackers to achieve so. In this case, the party A will not allow the transmitter to do so since the evidence is not the one A received from the transmitter. Even if A is compromised by T , this type of attack still fails, since A may require to publish all the keys eventually, and $m_{fake} \neq m$.

TABLE 2
Attack Surface(✓ refers to our Onionchain can defend the attack without any changes)

Attack Name	Initiator	Number of Attackers	Onionchain
Malicious-Transmitter	Transmitter	Single	✓
Malicious-Messenger	Relaying nodes	Single	✓
Replay	All participants	Single	✓
Calumniating	Receiver	Single	✓
Collusion	All participants	Multiple	✓

Malicious-Messenger Attack: In this case, the malicious party is one of the relaying nodes, e.g. B in our case. B creates a false message m_{fake} intentionally, and instead of using the original evidence, which is EV_2 in our case, B crafts new evidence EV'_2 based on a fake message m_{fake} . He also crafts a fake V'_1 based on the fake message m_{fake} . To notice, B does not know who will be the receiver, so he chooses a receiver R' randomly. His goal is to conceive others to believe the false message is from the transmitter. This type of attack will fail quickly. Since when the identity disclose protocol occurs, the attacker can not link the fake evidence EV'_2 to its previous evidence EV_1 . However, if he only crafts a fake V'_1 based on the fake message m_{fake} , and uses the original EV_2 , the attack will still fail. The reason is similar to the first case, and we will not repeat.

Replay Attack: Replay attack occurs when malicious relaying nodes resend a previous message and use the same evidence that used before. However, our timestamp can provide freshness for each message. The message receiver will discard the messages when they are stale.

Calumniating Attack: In this type of attack, an malicious receiver may create a false message/evidence intentionally, and tries to conceive others to believe the false message is from the transmitter. We assume that the false message is m_{fake} , and the original message is m . Therefore, in this case, EV_4 can be represented as follows:

$$EV_4 = \text{ENC}(PK_{C-R}, \text{SIGN}(Sk_R, \text{SIGN}(Sk_C, m_{fake} || EV_3))) \quad (10)$$

However, the attacker can not modify or replace the evidence EV_3 . This is because EV_4 is generated by C and R corporately, and C will not put his signature on it, if he detects $EV_3 \neq EV_4$. Therefore, as described in the identity disclosure protocol, EV_3 can be traced back to EV_0 . At this moment, T will make the three keys, i.e. K_{T-A} , K_{A-B} , K_{B-C} , publicly accessible, so that all parties can recover the original message m . It can be observed that $m_{fake} \neq m$, which turns out that R tires to perform an Calumniating Attack.

Collusion Attack: In this case, two parties work jointly, and tries to craft a fake message. However, this type of attack is subject to the analysis in the previous examples. That is, the goal of an attacker is to create fake evidence or a fake message, and these fake ones fail to equal the previous evidence when parties perform the identity disclose. Therefore, our Onionchain can defend this type of attacks with no changes. Moreover, in our paper, we demonstrate the case

where there are only three relaying nodes are involved. It can be much more complicated for the attacker to deploy a Collusion Attack when more relaying nodes are presented, which hinders the Collusion Attack effectively.

6 EVALUATION

In this section, we evaluate our presented Onionchain. We first introduce the simulation environment of our prototype. We then introduce a few metrics to validate the feasibility and effectiveness of Onionchain.

6.1 Simulation Environment

We simulate Onionchain as a Java Program (version 1.8). The Java Program communicate a local Ethereum Environment via web3j Library and the smart contracts are programmed by Solidity language. Basically, the Oniiochain client generates a transaction locally, and submits the transaction through the Web3j Interface, since Solidity language has limited support of cryptographic functionality. Also, we exclude the propagation delay, since it is negligible when compared with cryptographic operations, and it would be tricky to run our phototype on multiple machines. In terms of cryptographic operations, we use AES-128 as our symmetric encryption algorithm and ECDSA as our signature generation/verification algorithm. We also assume that one party will perform our protocol instantly once he receives a specific packet. That is, we do not take human intervention as a consideration. This is reasonable since it is not possible to measure time cost when human intervention is involved. Besides, The testing experiments are set up on a desktop computer with 2.20GHz Intel(R) Core(TM) Processor and 32GB memory.

6.2 Numbers of Relay Nodes

Recall that the security of Onionchain partly depends on the numbers of relaying nodes. A transmitter may choose to use multiple relaying nodes to achieve better privacy. Fig 5 shows the relationship between numbers of relaying nodes and the time cost. We choose different numbers of relay nodes and run our message transmitting protocol and identity disclosure protocol, respectively. Here, the message used to transmit via the network is 128 bit. It can be observed that all the two protocols cost less than 200,000 *us* (i.e. 200 *ms*), even if there are 30 relaying nodes are involved. Also, performing message transmitting protocol costs more time when compared with the other. This is because that message transmitting protocol uses more cryptographic operations. For example, message transmitting protocol involves signature generation while the identity disclosure is not. During our experiments, we also observe that when we execute one function twice during a short period of time, the second execution will be much faster than the first one. This may be because (i) if two calls are close to each other, the important information, such as the address of a function, might still be in the cache, meaning that the CPU does not need to fetch it again from memory. (ii) The hardware for executing the function is still active and ready after the first execution. This discovery will make more sense when we process a large number of messages in a real environment.

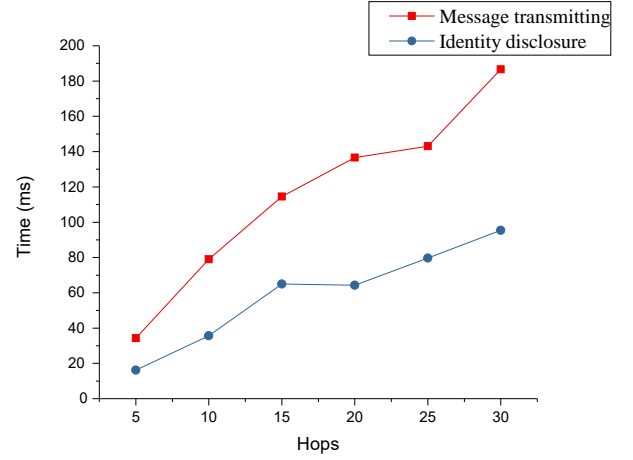


Fig. 5. Numbers of relay nodes V.S. Time cost

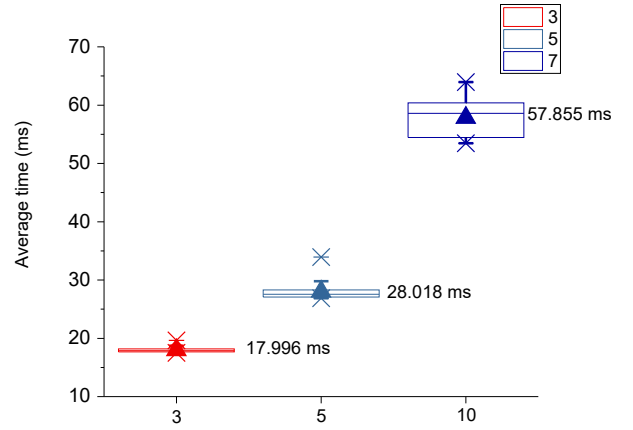


Fig. 6. Average time cost of message transmitting

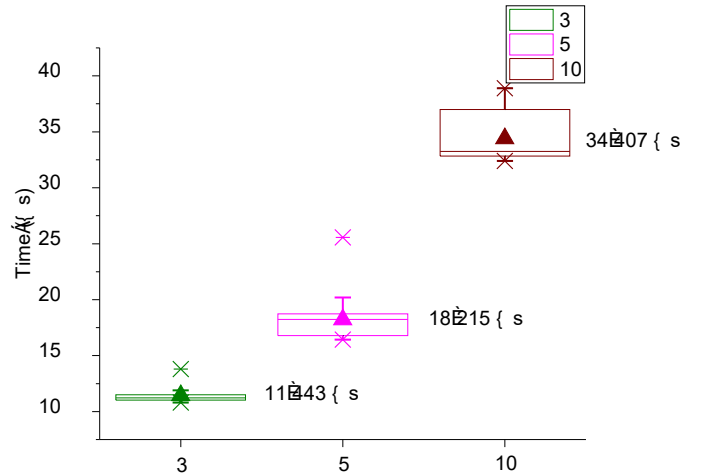


Fig. 7. Average time cost of identity disclosure

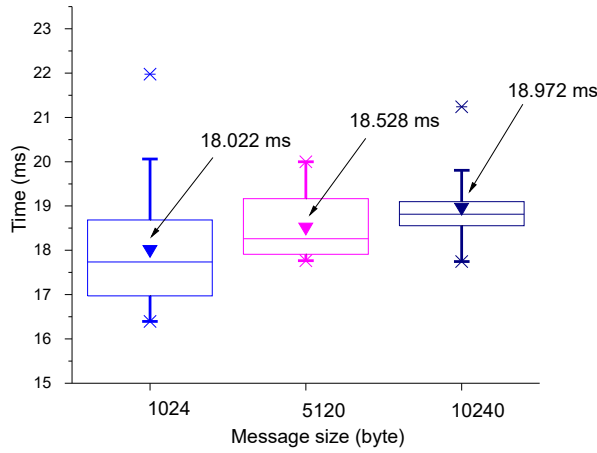


Fig. 8. Time cost for different size of packets in message transmitting

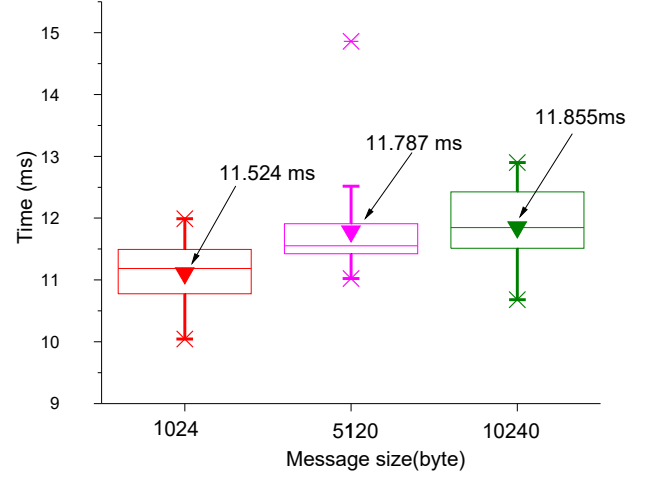


Fig. 9. Time cost for different size of packets in identity disclosure

6.3 Average Time Cost

We conduct experiments to evaluate Onionchain performance in regard of average time cost. To this end, we run the message transmitting protocol and identity disclosure protocol for 30 times and measure the time cost on average. Each time we use 3, 5, 10 relaying nodes respectively, to explore the performance in different privacy protection level. Here, the message used to transmit via the network is 128 bit. It can be observed in Fig. 6 that the mean value of time cost of message transmitting is only 17996 *us* (i.e. 17.996 *ms*), when there are 3 relaying nodes present. Even there are 10 relaying nodes involved, the mean value of time cost is still below 60 *us*, which is more than acceptable. As shown in Fig. 7, the measurement from the identity disclosure protocol shows a better performance. It only costs around 10 *ms* to perform, when there are 3 relaying nodes. When there are 10 nodes present, it costs around 35 *ms* on average.

6.4 Throughput

We now evaluate the time cost for different size of packets. Recall that in our previous experiment, the message used to transmit via the network is 128 bit. We choose this because a large packet may bring burden to the storage of Blockchain. Data expansion is still an open question that has not been well addressed for Blockchain applications. Even so, 128 bit may not be enough in exceptional circumstances, especially when files are involved. Therefore, we will explore the potential of Onionchain in terms of throughput. Fig. 8 and Fig. 9 show the results. It can be observed that even though the packets have grown to 10 Mbit, the overheads are still fairly small. The time costs have only increased less than 1 *ms*, in the cases of message transmitting and identity disclosure. We infer that this is because of the optimization of the run-time environment. For example, the function for encryption may remain active and ready for a long while, as long as it has been loaded into the memory.

7 RELATED WORK

In this section, we will review the research efforts that are related to ours. In addition to what we have reviewed in the introduction, We will review more research efforts on Blockchain in the term of privacy and traceability. Afterwards, we then review Blockchain-based vehicles communication systems.

We first review the privacy-preserving solutions related to Blockchain. In this regards, a lot of efforts have been made, while they fail to address remain the traceability. For example, Kosba *et al.* present HAWK [8], which stores the encrypted transactions instead of plain-text ones to protect transactional privacy. In addition, instead of presenting new solutions for privacy, there are a bulk of works that use Block as building blocks to address the privacy-preserving issues existing in a specific scenario. Zyskind *et al.* [37] use the Blockchain to manage personal data. In terms of privacy, the traditional third party is replaced by decentralized Blockchain system in their system, so that users will not have any concerns about the leakage of their privacy. Dorri *et al.* [38] discuss how to employ a Blockchain-based system to offer access control while preserving privacy in the context of Internet of Things. Ouaddah *et al.* [39] address a similar problem in IoT context using Blockchain. An extension has also been proposed by Dorri [40], which discusses a few open questions for Internet-connected automotive in the regards of security and privacy. Yue *et al.* [41] propose Healthcare Data Gateway, which addresses the privacy issues in the intelligence of healthcare systems. They use the Blockchain to manage and share data securely without violating privacy. A similar issue has also been addressed in Esposito's work [42]. Aitzhan *et al.* [43] provide a Blockchain-based solution in the area of smart grid. Multi-signatures and anonymous encrypted messaging streams are involved in addressing transaction security in decentralized smart grid systems. A proof-of-concept is also implemented to validate their solution. Liang [44] propose a decentralized provenance architecture using blockchain technology, providing tamper-proof records, and achieving privacy of the provenance data in the cloud environment. However, as discussed, these Blockchain-based solutions

are not omnipotent, since they may subject to powerful transaction analysis attacks [7].

We now review the Blockchain-based solutions in the term of traceability. Tian [45] propose a Blockchain-based system to offer traceability for food supply chain, guarantee the food safety effectively. Another paper from him [46] demonstrate how the same idea can work in the food supply chain with HACCP (Hazard Analysis and Critical Control Points). Similarly, Lu *et al.* [47], Kamath [48] and Galvez *et al.* [49] discuss how to trace the origin of products/food using Blockchain. In addition to the food supply chain, other applications are proposed based on the traceability of Blockchain. For example, Di [50] *et al.* use the Traceability of Blockchain to observe inter-organizational business processes. The work that is most closely related to ours is the CreditCoin [51] by Li *et al.* In their work, their goal is to provide a privacy-preserving Blockchain-Based incentive mechanism. They also discuss the traceability in their work. Our work is different from theirs. The reasons are (i) Their main concern is the incentive mechanism rather than traceability; (ii) They involve the Trace manager to trace malicious nodes. A trace manager is a group of decentralized parties other than regular users. In our paper, we do not make such an assumption. That is, every party can be the one who traces the transactions back. Therefore, our design is more realistic and decentralized; (iii) The generality of their design is not well discussed, while our work can adapt to many applications with little hindrance.

We now review the vehicle communication systems that use Blockchain as their core engine. Sean Rowan *et al.* [52] present a light inter-vehicle session key establishment protocol using Blockchain technology, which establishes a trustworthy relationship between vehicles and manufacturers. Madhusudan Singh *et al.* [53] propose an Intelligent Vehicle-Trust Point (IV-TP) mechanism using Blockchain technology, which provides security and reliability for vehicles behavior. Blockchain provides verifiable technical support for their solution. Similarly, Sharma *et al.* [4] build a transport system based on Blockchain, allowing vehicles to achieve secure Resources access. Jamin *et al.* [54] deploy Blockchain technology in the context of a vehicular ad-hoc network (VANET), providing decentralized VANET services, including vehicle insurance, updates on traffic jams and weather forecasts. It can be observed that most of the efforts introduce Blockchain to solve a specific issue in the current vehicles communication systems. Different from their works, our onionchain uses the vehicles communication systems to demonstrate our generality.

8 CONCLUSION

The fact that Blockchain technology fails to provide traceability and privacy-preserving simultaneously hinders Blockchain from developing and expanding its applications. To counter the issue, we propose Onionchain, offering applications a balanced option. Specifically, Onionchain integrates the mechanism of Onion routing into our protocol, enabling parties to perform transactions in an anonymous way. Besides, Onionchain records intermediate variables generated during the transactions, termed evidence, so that decentralized parties can trace a malicious party back when

dishonest behaviors occur. An Onionchain-based Vehicular Ad Hoc Networks is demonstrated to provide community guidelines to follow, showing the generality of our design. Extensive experiments are also performed to validate our Onionchain.

We admit the current design also has limitations. For example, we do not take the scalability of blockchain into consideration [55]. That is, our design also suffers from the transaction data explosion. To such a concern, we argue that (i) scalability of Blockchain is an open problem, which is out of our focus. (ii) The solutions to this issue have been widely discussed in previous works, such as [56], [57]. Our approach does not retrofit the ecosystem of Blockchain. Therefore, these approaches can be integrated into our design with little efforts. (iii) For many Blockchain applications such as VANET, the Blockchain is deployed in RSUs or other Infrastructures, which have enough disk spaces. An infrastructure like RSU will have no concerns on the scalability of Blockchain. Finally, the next stage of our work falls into three aspects : (i) we may extend our work by exploring more scenarios. That is, more Onionchain based system will appear. (ii) We may integrate solutions that address the scalability to refine our system. (iii) we may make our prototype public access, so that community can benefit from it in a timely manner.

ACKNOWLEDGEMENTS

Jian Weng was partially supported by National Key R&D Plan of China (Grant Nos. 2017YFB0802203, 2018YFB1003701), National Natural Science Foundation of China (Grant Nos. 61825203, U1736203, 61732021), Guangdong Provincial Special Funds for Applied Technology Research and Development and Transformation of Important Scientific and Technological Achieve (Grant Nos. 2016B010124009 and 2017B010124002). Yue Zhang was partially supported by National Natural Science Foundation of China (Grant Nos. 61877029). Jiasi Weng was partially supported by National Natural Science Foundation of China (Grant Nos. 61802145, 61872153). Ming Li was partially supported by National Natural Science Foundation of China (Grant Nos. 11871248, U1636209).

REFERENCES

- [1] J. Pearce, "Blockchain in academia: A literature review," 2019.
- [2] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xi-ang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2018.
- [3] C. H. Liu, J. Fan, P. Hui, J. Wu, and K. K. Leung, "Toward qoi and energy efficiency in participatory crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4684–4700, 2014.
- [4] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-vn: A distributed blockchain based vehicular network architecture in smart city," *JIPS*, vol. 13, no. 1, pp. 184–195, 2017.
- [5] S.-F. Tzeng, S.-J. Horng, T. Li, X. Wang, P.-H. Huang, and M. K. Khan, "Enhancing security and privacy for identity-based batch verification scheme in vanets," *IEEE transactions on vehicular technology*, vol. 66, no. 4, pp. 3235–3248, 2015.
- [6] Meticulous Research, "Report: Blockchain market to be worth over \$28 billion by 2025," <https://www.newsbtc.com/2018/11/27/report-blockchain-market-to-be-worth-over-28-billion-by-2025/>.

- [7] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.
- [8] 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.
- [9] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203–4214, 2014.
- [10] D. Chaum and E. Van Heyst, "Group signatures," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1991, pp. 257–265.
- [11] E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in *Annual International Cryptology Conference*. Springer, 2002, pp. 465–480.
- [12] Y. Lu, Q. Tang, and G. Wang, "Zebalancer: Private and anonymous crowdsourcing system atop open blockchain," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 853–865.
- [13] D. Liu, A. Alahmadi, J. Ni, X. Lin, and X. Shen, "Anonymous reputation system for iiot-enabled retail marketing atop pos blockchain," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3527–3537, 2019.
- [14] A. Miller, M. Möser, K. Lee, and A. Narayanan, "An empirical analysis of linkability in the monero blockchain.(2017)," 2017.
- [15] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," NAVAL RESEARCH LAB WASHINGTON DC CENTER FOR HIGH ASSURANCE COMPUTING SYSTEMS, Tech. Rep., 1999.
- [16] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected areas in Communications*, vol. 16, no. 4, pp. 482–494, 1998.
- [17] M. Singh and S. Kim, "Blockchain based intelligent vehicle data sharing framework," *arXiv preprint arXiv:1708.09721*, 2017.
- [18] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, 2018.
- [19] F. Knirsch, A. Unterwiesing, and D. Engel, "Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions," *Computer Science-Research and Development*, vol. 33, no. 1-2, pp. 71–79, 2018.
- [20] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "Detecting blackhole attack on aodv-based mobile ad hoc networks by dynamic learning method," *IJ Network Security*, vol. 5, no. 3, pp. 338–346, 2007.
- [21] S. Nakamoto et al., "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [22] M. Szydlo, "Merkle tree traversal in log space and time," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2004, pp. 541–554.
- [23] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 2016, pp. 3–16.
- [24] S. King and S. Nadal, "Pcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper*, August, vol. 19, 2012.
- [25] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.
- [26] G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
- [27] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, 2016, p. 4.
- [28] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study," *IEEE Transactions on vehicular technology*, vol. 42, no. 4, pp. 546–554, 1993.
- [29] J.-S. Weng, J. Weng, Y. Zhang, W. Luo, and W. Lan, "Benbi: Scalable and dynamic access control on the northbound interface of sdn-based vanet," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 822–831, 2018.
- [30] J. Zander, "Distributed cochannel interference control in cellular radio systems," *IEEE transactions on vehicular Technology*, vol. 41, no. 3, pp. 305–311, 1992.
- [31] C. M. Willson, Z. H. Doany, and T. A. Parkinson, "Gps interface for locating device," Jan. 15 2008, uS Patent 7,319,387.
- [32] K. Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010.
- [33] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time-the scoot method," *IEEE Transactions on vehicular technology*, vol. 40, no. 1, pp. 11–15, 1991.
- [34] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, "A reputation-based announcement scheme for vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4095–4108, 2012.
- [35] N. Haddadou, A. Rachedi, and Y. Ghamri-Doudane, "A job market signaling scheme for incentive and trust management in vehicular ad hoc networks," *IEEE transactions on vehicular technology*, vol. 64, no. 8, pp. 3657–3674, 2014.
- [36] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE transactions on vehicular technology*, vol. 61, no. 1, pp. 86–96, 2011.
- [37] G. Zyskind, O. Nathan et al., "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*. IEEE, 2015, pp. 180–184.
- [38] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 2017, pp. 618–623.
- [39] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in iot," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*. Springer, 2017, pp. 523–533.
- [40] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.
- [41] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of medical systems*, vol. 40, no. 10, p. 218, 2016.
- [42] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018.
- [43] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 840–852, 2016.
- [44] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "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, 2017, pp. 468–477.
- [45] F. Tian, "An agri-food supply chain traceability system for china based on rfid & blockchain technology," in *2016 13th international conference on service systems and service management (ICSSSM)*. IEEE, 2016, pp. 1–6.
- [46] —, "A supply chain traceability system for food safety based on haccp, blockchain & internet of things," in *2017 International Conference on Service Systems and Service Management*. IEEE, 2017, pp. 1–6.
- [47] Q. Lu and X. Xu, "Adaptable blockchain-based systems: A case study for product traceability," *IEEE Software*, vol. 34, no. 6, pp. 21–27, 2017.
- [48] R. Kamath, "Food traceability on blockchain: Walmarts pork and mango pilots with ibm," *The Journal of the British Blockchain Association*, vol. 1, no. 1, p. 3712, 2018.
- [49] J. F. Galvez, J. Mejuto, and J. Simal-Gandara, "Future challenges on the use of blockchain for food traceability analysis," *TrAC Trends in Analytical Chemistry*, 2018.
- [50] C. Di Ciccio, A. Cecconi, J. Mendling, D. Felix, D. Haas, D. Lilek, F. Riel, A. Rumpl, and P. Uhlig, "Blockchain-based traceability of inter-organisational business processes," in *International Symposium on Business Modeling and Software Design*. Springer, 2018, pp. 56–68.
- [51] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE*

Transactions on Intelligent Transportation Systems, vol. 19, no. 7, pp. 2204–2220, 2018.

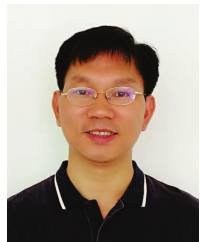
- [52] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. M. Goldrick, "Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels," *arXiv preprint arXiv:1704.02553*, 2017.
- [53] M. Singh and S. Kim, "Intelligent vehicle-trust point: Reward based intelligent vehicle communication using blockchain," *arXiv preprint arXiv:1707.07442*, 2017.
- [54] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 137–140.
- [55] G. Karame, "On the security and scalability of bitcoin's blockchain," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 2016, pp. 1861–1862.
- [56] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.
- [57] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016, pp. 45–59.



Ming Li received his B.S. in electronic information engineering from University of South China in 2009, and M.S. in information processing from Northwestern Polytechnical University in 2012. From 2016, he started his Ph. D. at Jinan University. His research interests include crowdsourcing, blockchain and its privacy and security.



Yue Zhang received his B.S. in information security from Xi'an University of Posts & Telecommunications in 2013, and M.S. in information security from Xi'an University of Posts & Telecommunications in 2016. From 2016, he started his Ph. D. at Jinan University. His research interests include Bluetooth, system security and Android security. He has published papers in international conferences and journals such as IEEE TDSC, IEEE TPDS, RAID etc.



Jian Weng is a professor and the Executive Dean with College of Information Science and Technology in Jinan University. He received B.S. degree and M.S. degree at South China University of Technology in 2001 and 2004 respectively, and Ph.D. degree at Shanghai Jiao Tong University in 2008. His research areas include public key cryptography, cloud security, blockchain, etc. He has published 80 papers in international conferences and journals such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, CT-RSA, IEEE

TDSC, etc. He also serves as associate editor of IEEE Transactions on Vehicular Technology.



Weiqi Luo received his B.S. degree and M.S. degree from Jinan University in 1982 and 1985 respectively, and Ph.D. degree from South China University of Technology in 1999. Currently, he is a professor with School of Information Science and Technology in Jinan University. His research interests include network security, big data, artificial intelligence, etc. He has published more than 100 high-quality papers in international journals and conferences.



Jiasi Weng obtained the B.S degree in Software engineering from South China Agriculture University in June 2016. She became a graduate student in Technology of Computer Application from Jinan University in September 2016. Her research interests include cryptography and information security, Blockchain and security in Software Defined Network..