Hindawi
Mobile Information Systems
Volume 2019, Article ID 5627497, 10 pages
https://doi.org/10.1155/2019/5627497

Research Article

Traceable and Authenticated Key Negotiations via Blockchain for Vehicular Communications

Yuling Chen,[1,2] **Xiaohan Hao**,[3] **Wei Ren**,[1,3,4] **and Yi Ren**[5]

[1] *Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, Guizhou 550025, China*
[2] *College of Computer Science & Technology, Guiyang, Guizhou 550025, China*
[3] *School of Computer Science, China University of Geoscience, Wuhan 430074, China*
[4] *Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China*
[5] *School of Computing Science, University of East Anglia, Norwich NR4 7TJ, UK*

Correspondence should be addressed to Wei Ren; weirencs@cug.edu.cn

Received 26 April 2019; Revised 9 September 2019; Accepted 26 September 2019; Published 21 December 2019

Academic Editor: Raul Montoliu

While key negotiation schemes, such as those based on Diffie–Hellman, have been the subject of ongoing research, designing an efficient and security scheme remains challenging. In this paper, we propose a novel key negotiation scheme based on blockchain, which can be deployed in blockchain-enabled contexts such as data sharing or facilitating electric transactions between vehicles (e.g., unmanned vehicles). We propose three candidates for flexible selection, namely, key exchanges via transaction currency values through value channels (such as the amount in transactions), automated key exchanges through static scripts,and dynamic scripts, which can not only guarantee key availability with timeliness but also defend against MITM (man-in-the-middle) attacks, packet-dropping attacks, and decryption failure attacks.

1. Introduction

Key negotiation schemes have been extensively studied [1–4], including those based on Diffie–Hellman, such as DH-RSA, DH-DSA, and DH-ECDSA. However, there appear limited attempts to integrate blockchain in the design of such schemes given its potential to realize certain properties. Specifically, blockchain is a technology that uses hash chains with digital signatures to implement accounting consensus and distributed storage, thus ensuring traceability and integrity. In recent years, there have been various applications of blockchain, including in vehicular communications [5–8].

Although key negotiation is the core supporting technology of blockchain data encryption, it also has great significance for the practical application of blockchain technology. However, targeted research is still in the initial stage. At present, some of the Diffie–Hellman protocols faced with a multitude of problems for vehicular communications: they cannot resist man-in-the-middle (MITM) attacks because there are no certifications of the other side of the negotiation; they cannot resist packet-dropping attacks because the negotiation packages are discarded by enemies, which caused the negotiation to be incomplete; they cannot resist decryption failure attacks because keys were not confirmed and the parties do not know whether the other party has received the negotiated package.

In this paper, traceable and authenticated key negotiations via blockchain are proposed by exchanging the information about master keys; that is, they are embedded in the transaction currency value and can be exchanged through using value channels, static scripts, and dynamic scripts. Besides, our proposal solves transaction credit and security issues. Application of blockchain technology in the field of electric vehicle trading can effectively solve the problems of data security, data tampering, history tracking, effective monitoring, and trading trust. The contributions of the paper are listed as follows:

(1) We propose traceabe and authenticated key negotiation schemes based on blockchain that can guarantee key availability with timeliness in vehicular communications, which can defend against MITM attacks, packet-dropping attacks, and decryption failure attacks

(2) We propose to use value channels, static scripts, or dynamic scripts for fast, automatic, and confirmable delivery of key negotiation materials in vehicular communications, especially, piggyback with normal payments

The organization of this paper is as follows. Related work is introduced in Section 2. The system model and adversary model are presented in Section 3. Section 4 describes our scheme, and its evaluation and analysis are presented in Section 5. We conclude the paper in Section 6.

2. Related Work

In recent years, a brief summary of the relevant concepts in key management was presented. Li et al. [1] proposed a novel privacy-preserving incentive announcement network based on blockchain via an efficient anonymous vehicular announcement aggregation protocol. Lei et al. [9] proposed a framework for providing a secure key management within heterogeneous network and demonstrated the efficiency of the framework by providing extensive simulation and analysis. Besides, Park et al. [2] proposed a RSU-based decentralized key management (RDKM), dedicated for the multicast services in the vehicle communication systems. In addition, they also proposed the enhanced CDKM (Cell-based Decentralized Key Management) which improved the rekeying performance through the reduction of the size of the subgroup [3]. According to the features of health blockchain, Zhao et al. [10] used a body sensor network to design a lightweight backup and efficient recovery scheme for keys of health blockchain, which could be used to protect private messages on health blockchain effectively. Alagheband and Aref [11] proposed a dynamic key management framework based on elliptical curve cryptography and signcryption method for heterogeneous WSNs. Anita et al. [12] proposed a novel polynomial-based Q composite random scheme for the establishment of triple key among communicating nodes in a network, which enabled a secure communication between wireless sensor nodes. One of the fundamental problems in cryptography is the generation of a common secret key between two legitimate parties to prevent eavesdropping. To solve the above challenge, Peng et al. [4] proposed an information-theoretic secret key generation (SKG) method for time-division duplexing which based orthogonal frequency-division multiplexing (OFDM) systems over multipath fading channels. Chen and Willems [13] proposed a novel construction method to eliminate the effect of noise and bias in SRAM-PUFs. In terms of defense against attacks, Bernardini et al. [14] presented a mechanism to protect differential privacies of the topology from an eavesdropper who has unauthorized access to the estimator. Internet of Things (IoT) marrying with vehicle communication is a new topic, and the kinds of literature are limited. Novo [15] proposed a new architecture for arbitrating roles and permissions in IoT. The results showed that the blockchain technology could be used as access management technology in specific scalable IoT scenarios. Polyzos and Fotiou [16] explored the potential of a blockchain-assisted information distribution system for the IoT. They identified key security requirements of such a system and discussed how they could be satisfied using blockchains and smart contracts. Zhou et al. [17] proposed a novel blockchain-based threshold IoT service system, in which servers could process user's data by performing homomorphic computations on the data without learning anything from them. In addition, Eze et al. [18] proposed a novel CR Assisted Vehicular NETwork (CRAVNET) framework which empowers CR-enabled vehicles to make opportunistic usage of licensed spectrum on the highways and developed a novel cooperative three-state spectrum sensing and allocation model. Zhang et al. [19] presented a novel perspective on vehicular communication architecture.

With the development of communications and social vehicles, many researchers have been studying and analyzing socially aware Internet of Vehicles with the assistance of an agent-based model intended to reveal hidden patterns behind superficial data. Malagund et al. [20] discussed the growth of Internet of Things in vehicular communication and presented surveys of the routing protocols. The application of blockchain technology in the field of vehicular communications can effectively solve the problems of data security, data tampering, history tracking, effective supervision and control, which has broad application fields and important application value. On the security of vehicular communications on blockchain, since transaction security and privacy protection issues present serious challenges, Kang et al. [5] explored a promising consortium blockchain technology to improve transaction security without reliance on a trusted third party. Pustisek et al. [6] introduced a concept of autonomous blockchain-based negotiation to select the most convenient electric vehicle charging stations. In addition, Yang et al. [7] proposed a decentralized trust management system in vehicular networks based on blockchain techniques which were effective and feasible in vehicular network. Cebe et al. [8] integrated Vehicular Public Key Management (VPKI) to the proposed blockchain to provide membership establishment and privacy. As blockchain technology has just emerged, the academic research on blockchain privacy protection and related key management is still in its infancy [21]. In 2015, Zyskind proposed a scheme to protect personal privacy by using blockchain, but the scheme focused only on how to construct blocks and implement authentication of block access and did not give a related key management scheme. The above analyses show that although the negotiation of data encryption key is the core supporting technology of blockchain data encryption, it has great significance for the practical application of blockchain technology in the field of key negotiations.

3. Problem Formulation

3.1. Master Key Negotiation via Blockchain. In the scenario of vehicular communication, the communication between vehicles and between vehicles and people need to require low latency, high reliability, and traceability. In order to meet these requirements, we take the master key information on blockchain and realize the master key negotiation through the blockchain. Our scheme is suitable for the scenarios where there is a need for payment. When the master key (mk) needs to be changed, exchange the key information in a certain payment to establish the key for the next secure communication. The following are three options for implementing master key negotiation, including value channel, static script, and dynamic script. In this scheme, there are several parameters for generating the master key, which are shown in Table 1.

Table 1: Explanation of parameters in our scheme.

p	A large prime number
g	A meta of g
T	The period of generating p and g
mk	Master key
sk	Session key
OP_Exponent	An exponential and modular operation we defined

3.2. Adversary Model and Security Requirement. In this section, we identify five potential vulnerabilities that could be exploited by an adversary to undermine our scheme: key leakage, packet-dropping attack, decryption failure attack, and man-in-the-middle (MITM) attack.

Definition 1 (Packet-Dropping Attack). Key negotiation packets are dropped by attackers.

Definition 2 (Decryption Failure Attack). One peer of key negotiation peers decrypts encrypted data packets from the other peers. As the encrypted key is not acknowledged, it may be encrypted by a wrong key.

Definition 3 (Man-in-the-Middle (MITM) Attack). MITM attacks can intercept normal network communication, as well as sniff or tamper with data, while the communication parties are unaware of it. That is, attacker acts as an intermediary, forming two pairs of keys, decrypting, encrypting, and forwarding in the middle.

In addition, since our key negotiation scheme is public, the attacker can gain full knowledge of the procedures and methods of our scheme.

It is worth noted that there are many deficiencies in the traditional key negotiations:

(1) It is vulnerable to packet-dropping attacks because the swap part can be lost or dropped. Since there is no confirmation mechanism, whether the other party has successfully negotiated the key is uncertain.

(2) The key was not confirmed. The parties do not know whether the other party has received the negotiated package. Sending encrypted data may cause decrypting failure with high energy consumption.

(3) It is vulnerable to MITM attacks. The third party C acts as the receiver B when communicating with the sender A or acts as A when communicating with B. Both A and B negotiate a key with C; thus, C can monitor and transmit trade. A MITM attack is described as follows:

   (a) B sends the public key in a message to A.

   (b) C intercepts and analyses the message. Next, C saves the public key from B and sends a message to A by using the public key Y_C of C and disguises as B. After receiving the message from C, A stores B's ID and Y_C together. Similarly, C uses Y_C to send a message to B and disguises as A.

   (c) B calculates secret key K_1 based on private key X_B and Y_C. A calculates the secret key K_2 based on the private key X_A and Y_C. C uses private key X_C and Y_B to compute K_1 and uses X_C and Y_A to compute K_2.

   (d) From now on, C can forward A's message to B or B's message to A and modify their ciphertext. Neither A nor B knows that they are sharing communication with C.

In this paper, we focus on the following security requirements:

(1) No password is stored in plaintext unless it is placed in a sufficiently secure cryptographic device. Besides, no operation on cryptographic devices can make keys appear outside the cryptographic devices as plaintext.

(2) To ensure the separation of keys, different communication entities use different keys, and they must be irrelevant.

(3) Keys need to have a certain backup mechanism. All transactions are recorded, and the above conditions can be met by adding a timestamp on blockchain.

(4) Keys must be valid. When the old key expires, it needs to be replaced in time. At the same time, the security of the new key and the old key should be separated. Even if the old key is leaked, the security of the new key should not be compromised.

4. Proposed Scheme

4.1. Value Channel. The proposed scheme utilizes a transaction data structure to generate key pairs similar to the Diffie–Hellman key-exchange process, that is, the mk data exchange can be completed by embedding the data exchange process of the mk exchange into the currency value of the transaction and using the value channel (such as the amount in the transaction data structure). The mk negotiation process is shown in Figure 1, and the specific implementation process is as follows:

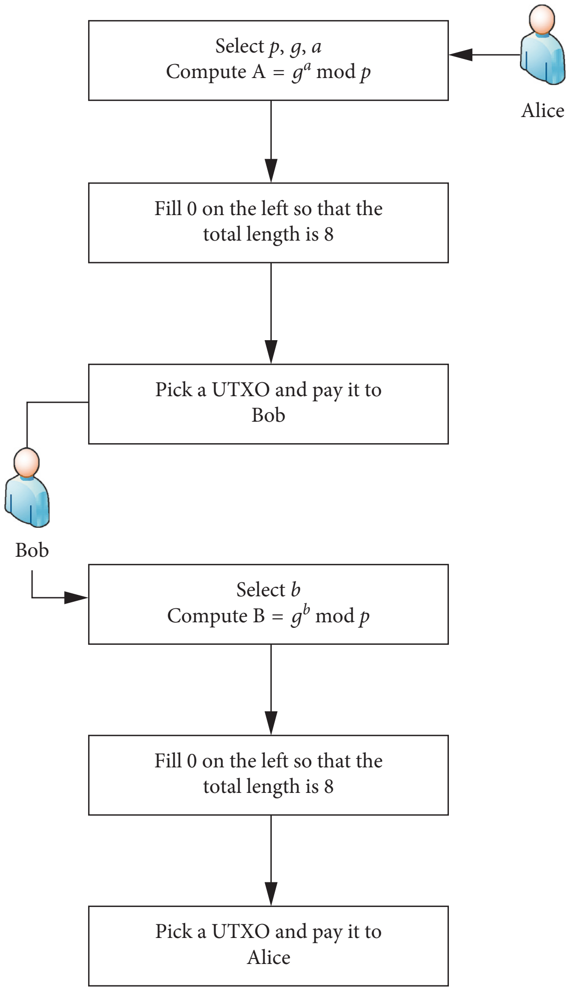

Select p, g, a
Compute $A = g^a \bmod p$
Alice
Fill 0 on the left so that the total length is 8
Pick a UTXO and pay it to Bob
Bob
Select b
Compute $B = g^b \bmod p$
Fill 0 on the left so that the total length is 8
Pick a UTXO and pay it to Alice

Figure 1: Master key-exchange process.

(1) The large prime number p and the meta g belong to system parameters which are placed on blockchain in advance for a special transaction.

(2) Alice reads blockhead and assigns to p, g; Alice randomly selects $L < a < p$, $A = g^a \bmod p$; Alice fills the left-hand side with 0 so that the total length is 8 such as 0.0...000A, 0.0 . . . 0000g, and 0.0...0000p; 1 btc $= 10^8$ cong.

(3) Alice publics broadcast ledger: picks an UTXO that someone else pays and pays for Bob: a cong, g cong, and p cong.

(4) After the ledger is recorded in the public ledger, Bob selects $L < b < p$ and calculates $B = g^b \bmod p$; Bob fills the left-hand side with 0 so that the total length is 8 such as 0.0...0000B.

(5) Bob publics broadcast ledger: picks an UTXO that someone else pays and pays for Alice: B cong.

(6) The mk shared by Alice and Bob is $A^b \bmod p$ or $B^a \bmod p$, which is used to encrypt subsequent communication between them.

```
Data: Null
while $(time() - starttime) > T$ do
    $p \longleftarrow random()$;
    while $(!IsPrime(p))$ do
        $p \longleftarrow random()$;
    end
    $g \longleftarrow random()\%p$;
    Transaction $\longleftarrow$ code($p$, $g$);
    starttime $\longleftarrow$ time();
end
```

Algorithm 1: System parameters are generated and uploaded into blockchain (type-I).

```
Data: Null
$p \longleftarrow random()$;
while $(!IsPrime(p))$ do
    $p \longleftarrow random()$;
end
$g \longleftarrow (random()\%p)$;
$a \longleftarrow (random()\%(p - 1))$;
$A \longleftarrow (g^a \; \% \; p)$;
Transaction $\longleftarrow$ code($p$, $g$, A);
```

Algorithm 2: System parameters are generated and uploaded into blockchain (type-II).

```
Data: $p$, $g$
Key initiator:
$p \longleftarrow readblock()$;
$g \longleftarrow (readblock()\%p)$;
$a \longleftarrow (random()\%(p - 1))$;
$A \longleftarrow (g^a \; \% \; p)$;
Transaction $\longleftarrow$ code($p$, $g$, A);
Submit;
Key receiver:
$p \longleftarrow readblock()$;
$g \longleftarrow (readblock()\%p)$;
$b \longleftarrow (random()\%(p - 1))$;
$B \longleftarrow (g^b \; \% \; p)$;
Transaction $\longleftarrow$ code(B);
Submit;
```

Algorithm 3: Master key parameters are generated to blockchain (type-I).

The specific implementation algorithm is shown in Algorithms 1–5. Algorithms 1 and 2 show that parameters are generated and uploaded into blockchain. Algorithm 3 and 4 show the process of generating data for mk negotiation and uploading to blockchain. Algorithm 5 is a mk generation algorithm on blockchain. According to above algorithms, it is worth to know that the time cost of Algorithm 1 is $T + \log p * t$, t is one time to test prime (p). Assuming the length of large prime number is n, the success rate of generating a large prime number in the algorithm is $1/n$. Therefore, the number of attempts to generate a large prime number in the algorithm depends on the length of the large

```
Data: $p$, $g$
Key initiator:
$p \longleftarrow readblock()$;
while ($!IsPrime(p)$) do
    $p \longleftarrow random()$;
end
$g \longleftarrow (random()\%p)$;
$a \longleftarrow (random()\%(p-1))$;
$A \longleftarrow (g^a \ \% \ p)$;
Transaction $\longleftarrow$ code($p$, $g$, A);
Submit;
Key receiver:
$p \longleftarrow readblock()$;
$g \longleftarrow (readblock()\%p)$;
$b \longleftarrow (random()\%(p-1))$;
$B \longleftarrow (g^b \ \% \ p)$;
Transaction $\longleftarrow$ code(B);
```

Algorithm 4: Master key parameters are generated to blockchain (type-II).

```
Data: $a$, $b$, A, B
$p \longleftarrow readblock()$;
$A \longleftarrow decodetransaction()$;
$B \longleftarrow decodetransaction()$;
$mk = (A^b \% p) = (B^a \% p)$;
```

Algorithm 5: Master key parameters are generated to a master key.

prime number. The time complexity of Algorithms 2 and 4 is $O\ (n)$. In addition, the time complexity of Algorithms 3 and 5 is $O\ (1)$.

In order to protect the security of the transaction, such as negotiating price and electricity sales, it is necessary to exchange mk-related information by using the value channel (for example, the amount in the transaction data structure) and embed the exchange process of mk-related information into the transaction currency value. In total, our proposal has the following advantages:

(1) The mk exchange process is publicly documented.

(2) There is a timestamp which can be traced.

(3) Since only A (sender) and B (receiver) can determine the negotiated mk, other users cannot interpret the message (confidentiality). A knows that only user B can use this mk to generate a message (authentication) which protects the security of the transaction and resists decryption failure attacks.

(4) It is characterized by the fact that the sk is generated only when needed, which is reducing the chance of attacks by storing sk for a long time.

In addition, the methods of electric trading in the existing technology are still faceing the defect of transaction security. In this paper, we provide a traceable and authenticated mk negotiation via blockchain for vehicular communications, which is ebbing the data exchange process of key exchange into transaction currency value, through the exchange of data which can be accomplished by using the value channel. It is worth to note that this method satisfies the following:

(1) The two parties which need secure communication can use this method to determine the negotiated mk.

(2) The key-exchange protocol can only be used for the exchange of mk, but cannot be used for the encryption and decryption of messages. After both parties determine the mk, the session key sk will be generated from the master key, sk = hash (mk, last block header hash).

4.2. Static Scripts. In this section, we discuss the specific process of sharing information with static scripts. The specific model is shown in Figure 2, and the specific implementation steps are as follows:

(1) Alice generates a transaction with two outputs: one is normal P2PKH script (Alice's public key) with the number of bitcoins as input and the other is messageA with zero bitcoins, messageA = {A_len, 4 byte; g_len, 4 byte; p_len, 4 byte; A, A_len byte; g, g_len byte; p, p_len byte; timestamp; type}

(2) Alice uses Bob's public key to encrypt messageA and then regards it as the entire output script

(3) Bob confirms receipting the money, finds the transaction on blockchain, reads messageA, and parses messageA to get A, g, p

(4) Bob generates a transaction with two outputs: one is normal P2PKH script (Bob's public key) with the number of bitcoins as input and the other is messageB with zero bitcoins, messageB = {B_len, 4 byte; B, B_len byte; timestamp; type}

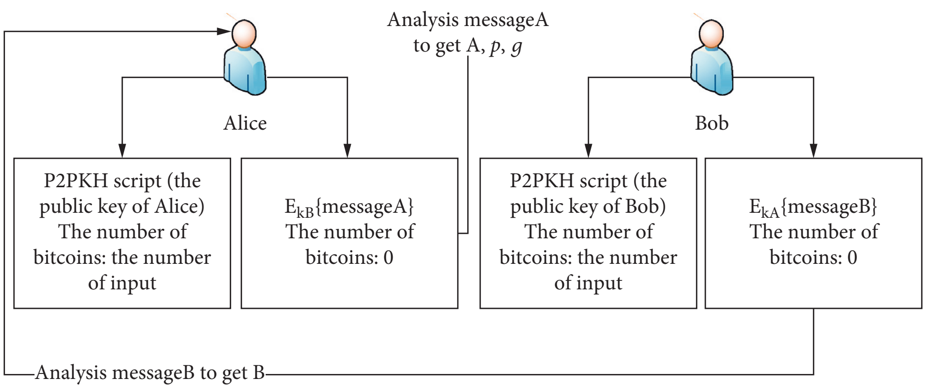
(5) Bob uses Alice's public key to encrypt messageB and then regards it as the entire output script

(6) Alice confirms receipting of the money, finds the transaction on blockchain, reads messageB, and parses messageB to get B

(7) Alice and Bob use their private keys to generate redemption scripts to redeem bitcoins

For the above process, we make the following explanations: since it would be costly to put all bitcoins in one script which contains the data to be exchanged and put them on blockchain, we design two output scripts. One is a valid output script that utilizes P2PKH transaction mode so that the traders can generate redemption scripts, and the other transaction output contains data which needs to be exchanged as an invalid output.

In this paper, the method we proposed has two main validations. The first is whether the public key can be converted to the correct address, and the second is whether signature is correct regardless of you are the owner of the public key. The main content of signature is calculating abstract (the hash of transaction information) by using the


Analysis messageA to get A, p, g
Alice
Bob
P2PKH script (the public key of Alice) The number of bitcoins: the number of input
E_{kB}{messageA} The number of bitcoins: 0
P2PKH script (the public key of Bob) The number of bitcoins: the number of input
E_{kA}{messageB} The number of bitcoins: 0
Analysis messageB to get B

MessageA = {A_len, 4 byte; g_len, 4 byte; p_len, 4 byte; A, A_len byte; g, g_len byte; p, p_len byte; time stamp; type;}
messageB = {B_len, 4 byte; B, B_len byte; time stamp; type;}

Figure 2: Static scripts.

private key. In the case of verification, the signature and public key are computed. If the transaction abstract is correctly obtained, the transaction will be successful.

4.3. Dynamic Script. For bitcoin scripts, we customize a script operator OP_Exponent to implement exponential and modular operations in the stack so that the existence of timestamp can satisfy the traceability of exchanging mk. Since transactions are executed through bitcoin scripts, packet-dropping attackers cannot discard the negotiation package, which makes it impossible to complete the negotiation. In addition, the protocol can also be authenticated to ensure that the other party receives the information and gets the negotiated mk after passing the authentication correctly. In other words, our solution can defend against MITM attacks, packet-dropping attacks, and decryption failure attacks. Besides, dynamic scripts are more secure and flexible than static scripts.

Next, we discuss the specific implementation. Alice uses a UTXO, which has a locked script to set up the transaction. Besides, Alice puts a, g, and p into the lock script. The lock script is as follows:

1–75 a g p OP_Exponent OP_DUP OP_HASH160 be10f0a . . . OP_EQUALVERIFY OP_CHECKSIG

The comments of some script statements are shown in the Table 2.

Bob gives an unlock script and combines unlock script with lock script to confirm that the transaction is valid; thus, Bob can get the required data (g^a mod p). Since Bob uses the private key b to generate the required mk, a secure mk negotiation can be achieved.

However, this solution also faces two challenges. On the one hand, BVM keeps only the most basic instructions and extends them unless required. On the other hand, BVM upgrades require consensus across the bitcoin community which may cause bifurcations. There are also some problems we need to solve in future research.

Table 2: Comments on some statements in scripts.

1–75	Put the next N bytes on the stack, and the value of N is between 1 and 75
OP_Exponent	Implementing exponential and modular operations on stack data
OP_DUP	Copy the top data and place it on the top of the stack
OP_HASH	Perform ripemd160 (sha256 (data)) on top of the stack
be10f0a...	Bob's bitcoin address
OP_EQUALVERIFY	Operation which terminates the script in failure unless the two entries below it on the stack are equivalent
OP_CHECKSIG	Verified signature

4.4. Session Key Negotiation via Blockchain. Since the key information cannot be stored in the blockchain all the time, we propose to divide the key into the master key and the session key. The session key is abolished after use. After the master key is generated, in the subsequent communication, the session key (sk) is generated from the master key. sk = Hash (mk, last block header hash). Since the hash header is equivalent to a random number generator, sk can automatically change according to the latest block header. In this way, the times of generating keys through blockchain can be reduced to C (n, 2). Due to the timestamp, mk and sk can be traced to provide a guarantee for the security of key negotiation. At the same time, our scheme can realize resistance packet-dropping attack, decryption failure attack, and MITM attack which will elaborate in the next section.

4.5. Examples. The general vehicle network has two communication modes: communication between vehicle nodes and roadside infrastructures and communication between two vehicle nodes. One of the roadside infrastructures is roadside unit (RSU). Through these RSUs, such as 802.11 wireless access points, vehicles can access data stored in the roadside unit or upload their data. When a vehicle enters the communication range of the roadside node, it can access the wired network through the roadside node. On the other hand, the vehicle can also communicate with RSUs via multihop relay through other vehicles. For example, there are two cars with a certain distance which need to pass multiple RSUs, and they have to communicate with each other; thus, the transmitted information needs to be encrypted. In this scenario, if the traditional key negotiations are adopted, this transformation may be subjected to several attacks such as MITM attacks, packet-dropping attack, and decryption failure attack. However, those attacks can be avoided by packing them into the blockchain.

By combining key negotiations with blockchain, the key exchange process is publicly recorded and timestamped, which can be traced back to a long time ago.

Since only users and intelligent charging devices can determine the key, other users cannot interpret the message, which ensures the confidentiality of the message. In addition, users know that only intelligent charging devices can use negotiated keys to generate the message; thus, our proposal protects the security of electricity trading and resists decryption failure attacks.

Example 1. Scenario 1: reservation maintenance.

Vehicles can negotiate keys to 4S stores, and 4S stores automatically distribute keys to vehicles. At the appointed time, 4S stores confirm the identity of the vehicle by searching the negotiated mk on the blockchain and generate sk based on the latest block header. After the key negotiation is completed, they can maintain vehicles. In this way, the workload of people can be reduced. Owing to the existence of timestamp, the records can be traced back, and the records can be prevented from being tampered with.

Example 2. Scenario 2: remote fault diagnosis.

In the case of vehicle failure, it is worth discussing how to overcome the geographical and time constraints and achieve a collaborative diagnosis of remote experts. Our key negotiation can be used to find out the mk from blockchain and generate sk based on the latest block header to diagnose the vehicle when it fails.

Example 3. Scene 3: rent a car (share car).

Recently, more and more shared cars are appearing in our life. Especially, sharing cars has certain risks because it needs to connect people, machines, systems, and so on, and the connection type and quantity are rich and varied. Attackers can destroy the in-vehicle network system and realize the possibility of remotely maneuvering cars to increase traffic accidents. To prevent this phenomenon, we can take advantage of the traceability of the blockchain. The user must negotiate the mk with the vehicle (which may be the transaction amount) and use the vehicle by the generated sk from mk at the agreed time. Since the usage record can be found in the ledger, the possibility of an attacker implementing the attack is reduced.

5. Security Analysis and Performance Analysis

5.1. Security Analysis

Proposition 1. *The scheme proposed in this paper can resist MITM attacks.*

Proof. Diffie–Hellman key-exchange protocol exists MITM attacks, supposing there is a third party who desires to steal A and B's messages. The third party can impersonate the receiver to get the sender's A, p, g, select $1 < b_1 < p$, construct $\mathrm{B}_1 = g^{b1}$, and get the sender's key pair $\mathrm{K} = g^{ab1}$. Next, the third party pretends the sender, obtains the receiver's B, randomly selects $l < a_1 < p$, constructs $\mathrm{A}_1 = g^{a1}$, and obtains a pair of key pairs $\mathrm{K} = g^{a1b}$ with the receiver. From now on, C can forward A's message to B or B's message to A and modify their ciphertext. Neither A nor B knows that they are sharing communication with C.

The MITM attack exists because initiators do not authenticate the other party, but they will authenticate the negotiating party due to the signature on blockchain. Therefore, our proposed scheme can resist MITM attacks. □

Proposition 2. *Our proposed key negotiation scheme can resist packet-dropping attacks.*

Proof. Diffie-Hellman key exchange protocol exists packet-dropping attacks, because the negotiated parts may be lost or dropped. Whether the other party has successfully received information about the negotiated key and whether two parties can successfully negotiate the key are uncertain because there is no confirmation mechanism. However, blockchain can guarantee the success of the negotiation, that is, each client can see all parts of the negotiation and both parties can certainly form a pair of key and confirm that the other party knows the negotiated key. In addition, due to the timestamp, our scheme can satisfy the traceability of the negotiated key. □

Proposition 3. *The scheme we proposed in this paper can resist decryption failure attacks.*

Proof. Diffie-Hellman key-exchange protocol exists decryption failure attacks because the key was not confirmed and the parties do not know whether the other party has received the negotiated package. If you send encrypted data, it may cause decrypting failure with high energy consumption. In this paper, since only the sender and receiver can determine the key, other demanders cannot interpret the message (confidentiality). The receiver knows that only the sender can use the key to generate the message to protect the transaction security and resist decryption failure attacks, which satisfies the authenticity of the negotiated key. □

Proposition 4. *$g^i \pmod p \neq g^j \pmod p$($p$ prime number), where $i \neq j$ and i, j between 1 and $(p-1)$, then 325g is the original root of p.*

Proof. Assuming that the number g is the original root of p, then the result of $g^i \pmod p$ is different, and there is

Table 3: Scheme comparison.

Proposed key negotiations	Lei et al. [9] A secure key management within the heterogeneous network	Our scheme Three key negotiations via blockchain	Park et al. [2] RSU-based decentralized key negotiation	Our scheme Three key negotiations via blockchain
Analyze the performance of proposed scheme	✓	✓	✓	✓
Propose optimization algorithms	×	Three steps to generate a key	Determine the design parameters	Three steps to generate a key
Defend against MITM attacks	✓	✓	×	✓
Defend against packet-dropping attacks	✓	✓	×	✓
Defend against decryption failure attacks	×	✓	×	✓
Satisfy traceability of key exchange	✓	✓	×	✓

$1 < g < p$ and $0 < i < p$, in the final analysis, $g^{p-1} = 1 \pmod{p}$ is established when and only when the exponent is $p-1$ (here p is a prime number). □

Proposition 5. *Even if you know p, g, A, and B, it is hard to guess mk.*

Proof. Based on the assumption that DHP is difficult, given the finite cyclic group G, the generator g, and the randomly selected elements $\alpha = g^a$ and $\beta = g^b$ on G, it is also difficult to calculate $\gamma = g^{ab}$, so even if we know p, g, A, and B, it is difficult to deduce K. □

Proposition 6. *Our proposed key negotiation scheme is correct; both sides of the transaction (Alice and Bob) get the same mk.*

Proof. Alice randomly chooses large prime number p and generators g. Alice randomly chooses $L < a < p$ and calculates $\mathrm{A} = g^a \pmod{p}$. Bob chooses $l < b < p$ and calculates $\mathrm{B} = g^b \pmod{p}$, $\mathrm{mk} = \mathrm{A}^b \pmod{p} = (g^a)^b \pmod{p} = (g^a)^b \pmod{p} = (g^b)^a \pmod{p} = \mathrm{B}^a \pmod{p}$, which shows that Alice and Bob get the same mk. In addition, because of the existence of signature, the authenticity of the negotiated key can be satisfied. □

Proposition 7. *sk can automatically change based on the latest block header and can be traced.*

Proof. Since the block header is equivalent to a pseudorandom number generator, the session key

$$\mathrm{sk} = \mathrm{Hash}\,(\mathrm{mk}, \text{last block header hash}) \tag{1}$$

can be guaranteed to be generated randomly. In addition, due to the timestamp, sks can be traced to enhance the security of our scheme. □

Proposition 8. *The cost and time of cracking key negotiations are not feasible in calculation, so the provable safety is satisfied.*

Proof. Due to the complexity of the algorithms involved, it is tricky to estimate the algorithm accurately, but our analysis gives some conservative estimates. For the most common strength of Diffie–Hellman (1024 bits), building a machine based on dedicated hardware which can crack one Diffie–Hellman prime every year costs hundreds of millions of dollars. Therefore, we conclude that the cost and time of cracking our key negotiations are not feasible in calculation. □

5.2. Performance Analysis

Proposition 9. *The overhead of time and space is low.*

Proof. Only mks are generated by the key information on the blockchain. In the case that the master key does not need to be changed, it only needs to be generated C $(n, 2)$ times. In addition, the time cost of Algorithm 1 is $T + \log p * t$, t is one time to test prime (p). sk can be generated quickly; thus, the time overhead of key negotiations can be reduced. □

Proposition 10. *Assuming the length of a large prime number is n, the time complexity of the algorithm is O (n).*

Proof. Assuming the length of large prime number is n, the success rate of generating a large prime number in the algorithm is $1/n$. Therefore, the number of attempts to generate a large prime number in the algorithm depends on the length of the large prime number. For example, assuming that the prime length is 100 bits, it usually takes 100 times to produce a prime number.

According to the comparison of three key negotiations in Table 3, the existing key negotiation schemes cannot satisfy traceability, authenticity, or resist some attacks synchronously. When the information recorded on blockchain involves personal privacy, trade secrets, and even national security, the data must be encrypted and authorized to access. Therefore, it brings the following challenge to the management of data encryption keys: since blockchain ciphertext is open, statistical analysis of ciphertext must be avoided. While blockchain is used in electronic transactions, it also faces greater challenges: key leakage, packet-dropping attack, modify and falsify messages, man-in-the-middle attack (MITM), and decryption failure attacks. □

6. Conclusions

In this paper, we propose a traceable and authenticated key negotiation scheme based on blockchain. Specifically, key exchanges rely on value channels, static scripts, or dynamic scripts. The key materials can be traced back publicly by timestamps upon request and can be confirmable to avoid decryption failure attacks. Negotiation peers can be authenticated to resist MITM attacks. The packet-dropping attacks are defended against as the timeliness of key negotiation can be guaranteed due to the availability of channels and scripts. Last but not least, the key negotiation process can be preconfigurable and automatically executed.

Data Availability

We mainly focus on theoretical analysis and do not refer to data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research was financially supported by the Major Scientific and Technological Special Project of Guizhou Province under grant no. 20183001, Open Funding of Guizhou Provincial Key Laboratory of Public Big Data under grant no. 2018BDKFJJ009, 2017BDKFJJ006, Open Funding of Hubei Provincial Key Laboratory of Intelligent Geo–Information Processing under grant no. KLIGIP2016A05, and National Natural Science Foundation of China, “Theory and Key Technologies of Distributed Autonomous Security Supervision for Data Opening and Sharing” (grant nos. 61962009 and 2020/01-2023/12).

References

[1] L. Li, J. Liu, L. Cheng et al., “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.

[2] M.-H. Park, G.-P. Gwon, S.-W. Seo, and H.-Y. Jeong, “RSU-based distributed key management (RDKM) for secure vehicular multicast communications,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 644–658, 2011.

[3] M. H. Park, G. P. Gwon, and S. W. Seo, “Enhancement of cell-based decentralized key management in vehicular communication network,” in *Proceedings of the 2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, Chennai, India, February 2011.

[4] Y. Peng, P. Wang, W. Xiang, and Y. Li, “Secret key generation based on estimated channel state information for TDD-OFDM systems over fading channels,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5176–5186, 2017.

[5] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, “Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[6] M. Pustisek, A. Kos, and U. Sedlar, “Blockchain based autonomous selection of electric vehicle charging station,” in *Proceedings of the 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, pp. 217–222, Beijing, China, October 2016.

[7] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, “Blockchain-based decentralized trust management in vehicular networks,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[8] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, “Block4Forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles,” *IEEE Communications Magazine*, vol. 56, no. 10, pp. 50–57, 2018.

[9] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, “Blockchain-based dynamic key management for heterogeneous intelligent transportation systems,” *IEEE Internet of Things Journal*, vol. 4, no. 6, 2017.

[10] H. Zhao, P. Bai, Y. Peng, and R. Xu, “Efficient key management scheme for health blockchainficient key management scheme for health blockchain,” *CAAI Transactions on Intelligence Technology*, vol. 3, no. 2, pp. 114–118, 2018.

[11] M. R. Alagheband and M. R. Aref, “Dynamic and secure key management model for hierarchical heterogeneous sensor networks,” *IET Information Security*, vol. 6, no. 4, pp. 271–280, 2012.

[12] E. A. M. Anita, R. Geetha, and E. Kannan, “A novel hybrid key management scheme for establishing secure communication in wireless sensor networks,” *Wireless Personal Communications*, vol. 82, no. 3, pp. 1419–1433, 2015.

[13] B. Chen and F. M. J. Willems, “Secret key generation over biased physical unclonable functions with polar codes,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 435–445, 2018.

[14] C. Bernardini, M. R. Asghar, and B. Crispo, “Security and privacy in vehicular communications: challenges and opportunities,” *Vehicular Communications*, vol. 10, pp. 13–28, 2017.

[15] O. Novo, “Blockchain meets IoT: an architecture for scalable access management in IoT,” *IEEE Internet of Things Journal*, vol. 5, no. 2, p. 1, 2018.

[16] G. C. Polyzos and N. Fotiou, “Blockchain-assisted information distribution for the internet of things,” *IEEE International Conference on Information Reuse and Integration (IRI)*, vol. 1, pp. 75–78, 2017.

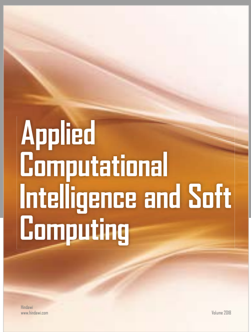
[17] L. Zhou, L. Wang, Y. Sun, and P. Lv, “A blockchain-based IoT system with secure storage and homomorphic computation,” *IEEE Access*, vol. 6, p. 99, 2018.

[18] J. Eze, S. Zhang, E. Liu, and E. Eze, “Cognitive radio-enabled internet of vehicles: a cooperative spectrum sensing and allocation for vehicular communication,” *IET Networks*, vol. 7, no. 4, pp. 190–199, 2018.

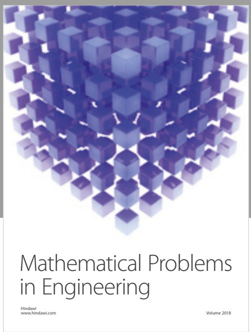
[19] Y. Zhang, F. Tian, B. Song, and X. Du, “Social vehicle swarms: a novel perspective on socially aware vehicular communication architecture,” *IEEE Wireless Communications*, vol. 23, no. 4, pp. 82–89, 2016.

[20] K. Malagund, S. Mahalank, and R. Banakar, "Evolution of IoT in smart vehicles: an overview," in *Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 804–809, Greater Noida, India, October 2015.

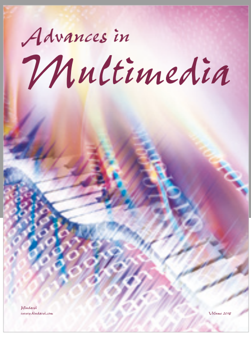
[21] Y. Yuan and F. Y. W. Blockchain, "The state of the art and future trends," *Acta Automatica Sinica*, vol. 42, pp. 481–494, 2016.

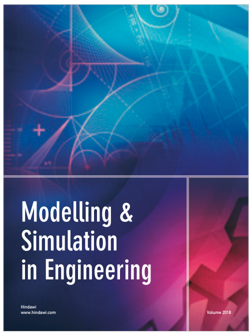

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

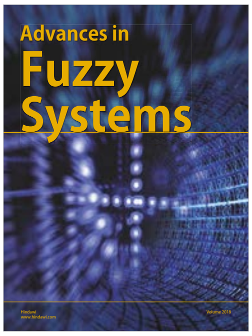
Journal of Engineering

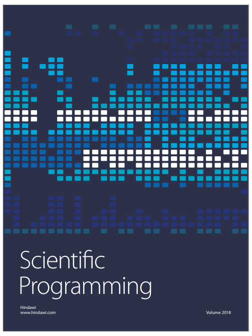

Advances in Multimedia

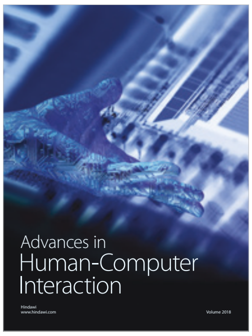

Modelling & Simulation in Engineering

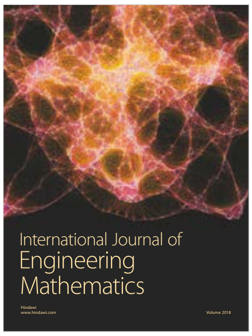
Advances in Artificial Intelligence

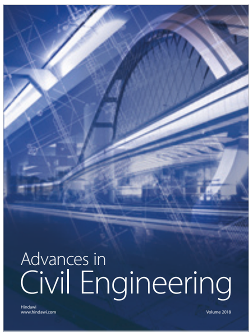
International Journal of Reconfigurable Computing

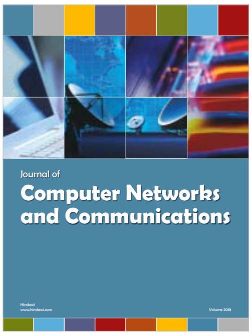

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

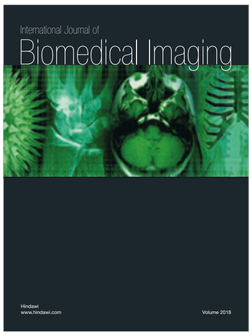

International Journal of Engineering Mathematics


Advances in Civil Engineering

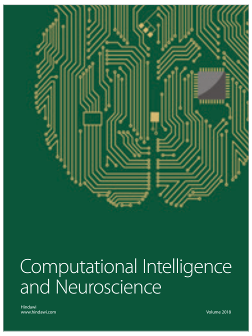

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience