Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9911915, 7 pages
https://doi.org/10.1155/2022/9911915

*Research Article*

# A Multipath Payment Scheme Supporting Proof of Payment

**Hangguan Qian and Lin You**

*College of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, 310018 Zhejiang, China*

Correspondence should be addressed to Hangguan Qian; 171270010@hdu.edu.cn

Received 26 October 2021; Revised 26 November 2021; Accepted 29 November 2021; Published 4 January 2022

Academic Editor: Alireza Souri

Blockchain technology has always been plagued by performance problems. Given this problem, many scaling schemes have been put forward. A layer 2 network is a technology that solves the performance problem of blockchain. Connected parties in this network can set up channels to send digital currency to each other. Since the interaction with the blockchain is only required when the channel is established and closed, a large number of transactions do not need to be recorded on the blockchain, thus reducing the blockchain capacity. Due to the special structure of the payment channel, the distribution of funds in the channel is often unbalanced, which limits the route payment to a certain extent. This paper improves the original payment method in the second layer network by introducing new scripts. The new payment scheme supports proof of payment which is integral to the nature of the lightning network and divides the payment into several subpayments, so the large payment can be divided into relatively small payments. Due to the capacity limitation of the payment channel, theoretically, the success rate of the micropayment route is higher. This paper tests the new payment scheme on the simulated network and validates the nature of this solution to have a high routing success rate while supporting proof of payment.

## 1. Introduction

Blockchain is a new type of distributed system. Due to its characteristics of immutability and decentralization, it has a wide range of applications in the fields of digital currency, certificate storage, and anticounterfeiting. The concept of blockchain originated in a 2008 paper titled "A P2P Network Electronic Currency System" written by Satoshi Nakamoto [1]. The blockchain is a decentralized payment system that does not rely on a central authority and allows users to make payments by sending digital currencies. However, due to the distributed nature of blockchain, its performance is far less than that of traditional centralized systems, and as the number of users of the first blockchain increases, its performance issues become prominent.

Layer 2 payment network technology is a solution to the performance problem of blockchain. Different from the layer 1 scheme, the second layer payment network which focuses on the construction of the off-chain payment network does not need to change the main chain protocol or only needs to change a few protocols. Due to the limitations of block structure, network delay, and other factors [2], the on-chain scheme is difficult to truly solve the performance problem of blockchain, while the off-chain scheme provides a possible way.

The lightning network [3] which uses asymmetric revocable commitments and hash time lock contracts (HTLC) to build an off-chain payment network is a layer 2 network payment scheme; a large number of transactions can be done in the second payment network, with the main chain only responsible for the records to create channels and close the deal; as a result, the entire blockchain system performance is improved greatly.

However, the channels in the lightning network have problems such as capacity limitations and uneven distribution of funds, which make it difficult to route large payments. An atomic multipath payment scheme [4] improves the success rate of large-amount payment routing by dividing large-amount payments into several small-amount payments. In this paper, we propose a new atomic multipath payment scheme, which ensures the atomicity of payment and supports proof of payment. In this paper, by introducing new scripts, we propose a new atomic multipath payment scheme that ensures atomicity of payments and supports proof of payment. Proof of payment is an important nature of the lightning network, and without it, the lightning network will not operate smoothly. Therefore, this paper implements a simulation network with the same topology structure as the lightning network. By simulating payment on the simulation network, the new scheme can be verified to have a higher routing success rate.

In the next sections, we present background knowledge related to this paper, including blockchain, lightning network, and atomic multipath payments. In Section 3, we describe in detail how the new scheme is constructed and the new features it has. In Section 4, we design a simulation network to test and verify the success rate of the new scheme for routing payments by running the new scheme on the simulation network.

## 2. Backgrounds

*2.1. Blockchain.* Blockchain technology is derived from the underlying technology of the digital currency. Nakamoto combined several previous inventions, such as B-Money and Hashcash, to create a completely decentralized online payment system that does not rely on central authorities. Its key innovation was the creation of a proof of work algorithm that conducted an election every 10 minutes on average, enabling a distributed network to reach a consensus on the status of transactions.

In recent years, blockchain technology has developed rapidly. After the first blockchain, Ethereum with smart contracts [5] and Hyperledger with an access mechanism have been proposed one after another [6]. At the same time, the corresponding intrusion detection technologies [7] and cryptocurrency regulations are rapidly improving [8]. Due to the continuous progress of blockchain-related technology, more and more blockchain-related applications are appearing in our daily life.

With the increase in the application of blockchain, a large number of users join the blockchain, and the performance problems of the blockchain gradually appear. Various solutions have been proposed to improve the performance of blockchain. Blockchain performance solutions can be divided into two categories: one is a capacity expansion on the first layer chain, which is the improvement of the original blockchain, and the second is the improvement of off-chain technology, which is the use of the second layer network to reduce the burden on the main chain. Theoretically, a large number of transactions travel through the second network, with the main chain only responsible for registering the results of the transactions. In this way, the performance of the blockchain could be greatly improved.

The first layer solution includes blockchain cash with increased block size [9] and segregated witness which compressed block size [10] and sharding technology [11]. Elastico [12] is the first sharding protocol for the permission-less blockchain. OmniLedger [13], a more recent distributed ledger based on the sharding technique, builds closely on Elastico and tries to solve the problems of Elastico. It uses a bias-resistant public randomness protocol for shard assignment. Various new consensus algorithms are used to accelerate blockchain transaction processing efficiency [14–16], which increase transaction throughput and reduce latency, respectively. As for the off-chain solutions, the second-layer solutions such as the first blockchain's lightning network, Ethereum's Raiden network [17], and Plasma [18] are in full swing. Along with the development of scaling technology, some security issues are also drawing attention and many related studies [19–22] have started to be proposed. As the technology tends to mature, applications of blockchain [23, 24] in various scenarios continue to emerge.

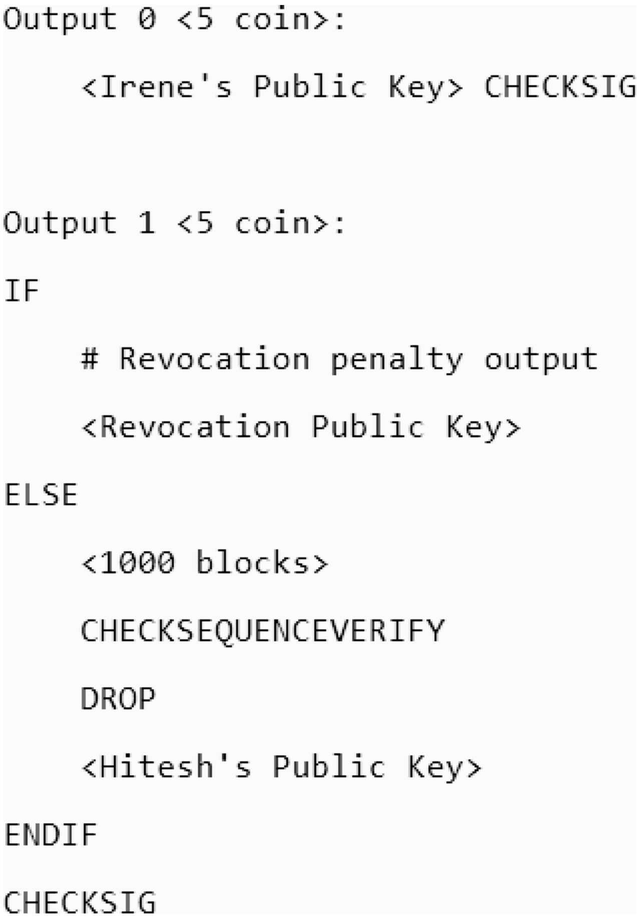
*2.2. Lightning Network.* The lightning network is a layer 2 network protocol based on the blockchain and has been proposed as a solution to the first blockchain's scalability performance problem. It is a peer-to-peer system that requires no escrow and allows users to make payments using the lightning network, a network of bidirectional payment channels. To date, tens of thousands of simultaneous micropayments can be accommodated in the lightning network, in contrast to the main chain, which can only process a few transactions per second.

The lightning network opens the payment channel by submitting a specific format of the transaction to the main chain, the layer 1 network, and then makes any number of lightning network transactions, updating the tentative allocation of funds from that channel without broadcasting those funds to the blockchain. Finally, the payment channel is closed, and funds for the channel are allocated by broadcasting the final version of the settlement transaction. By using asymmetric revocable commitment and hashing time lock contract (HTLC), the lightning network can punish cheaters and route transactions.

The lightning network uses contracts that can revoke previously promised transactions. When both parties sign a new commitment transaction, the revoking key needs to be exchanged. This is designed so that when one party tries to cheat, the revoking key can be used as a punishment. Specifically, the party trying to cheat will broadcast the old promised transaction to the main network in his favor. However, due to the existence of the time lock, he has to wait for some time before he can get the funds. The other party who has the revoking key can show the revoking key during this period and immediately get the corresponding funds. By such a design, the old contract is rendered invalid, and fraud cannot be carried out, as shown in Figure 1.

To create a HTLC, the payee will first create a secret $R$. They then calculate the hash $H$ of this $R$: $H = \mathrm{Hash}(R)$. The resulting hash is contained in the lock script for the contract output. Anyone who knows the secret can use it to exchange for output. The secret $R$ is also known as the preimage of the hash function, which is the data used as input to the hash function.

The second part of the HTLC is the time lock component. If the secret is not revealed, the HTLC payer can get a refund after a while. This is done by using an absolute time lock.



```
Output 0 <5 coin>:

    <Irene's Public Key> CHECKSIG


Output 1 <5 coin>:

IF

    # Revocation penalty output

    <Revocation Public Key>

ELSE

    <1000 blocks>

    CHECKSEQUENCEVERIFY

    DROP

    <Hitesh's Public Key>

ENDIF

CHECKSIG
```

Figure 1: An example of asymmetric revocable commitment. The difference from the regular script is the addition of a revocation key and a time lock of 1000 blocks long in output 1.

Anyone who knows the corresponding secret $R$ that can make a hash equal to $H$ can redeem the output by exercising the first clause of the IF statement. If the secret is not revealed, the HTLC states that after a certain number of blocks, the payee can claim a refund using the second clause of the IF statement. HTLC can take different forms by fine-tuning the script. For example, add a CHECKSIG operator and a public key to the first clause to restrict the conversion of the hash value to a specified recipient, who must know the secret $R$, as shown in Figure 2.

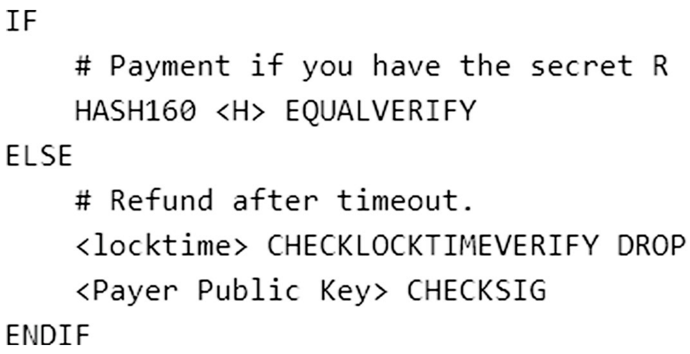
Lightning networks can allow any participant to route payments from one channel to another without trusting any intermediary. Suppose there are a payment channel between Alice and Bob and a payment channel between Bob and Carol but no payment channel between Alice and Carol. The lightning network allows Alice's funds to be routed to Carol. The specific operation is as follows: Claire generates a secret, does a hash operation on the secret, and then sends the result of the hash operation to Alice. Alice can use the result of this hash operation to create the HTLC contract and send it to Bob. As long as Bob shows the secret within a certain period, he can get the funds agreed in the contract. For now, Bob cannot reveal the secret, because only Carol knows the secret. Therefore, Bob needs to create a contract with Carol, which contains the same result of the hash operation. As long as Carol shows the secret within a certain time, he can get the money agreed in the contract. Carol is the secret generator and knows the secret. Carol shows the secret to Bob and gets the money in his contract with Bob. Bob gets the secret and uses it to get the money he agreed to in his contract with Alice. In this way, the transfer of funds between different channels can be realized; that is, funds can be routed in the lightning network.

*2.3. Atomic Multipath Payments.* Suppose a node has to pay another node 8000 Satoshis for something, and that node has only three channels with a 3000-Satoshi limit. Under traditional payment methods, a transaction cannot be completed with a single payment, while if multiple payments are used, the atomicity of the transaction cannot be guaranteed, and if one payment fails, the payment sender may need to request the recipient to return the other payment that has been completed. Another problem is that there is currently a ceiling on the number of channels that can be paid. If the payment exceeds this limit, it must be split into multiple payments, which also leads to the problem of payment failure.

Conner Fromknecht and Olaoluwa Osantokun proposed the atomic multipath payments (AMP) to solve the above two problems. The AMP scheme splits a large payment into several smaller payments, each of which can be routed to the recipient via a different path. Because the lightning network's channels can only pass up to an amount equal to their own capacity and there are a large number of small capacity channels in the lightning network to date, micropayments have a much higher routing success rate. By splitting the parent secret into multiple child secrets, the child payments carry the child secrets and send them to the receiver. Only when all the child secrets are collected can the recipient know the parent secret, thus obtaining the funds in all the payments, which ensures the atomicity of the payments. However, the scheme requires the sender to know the parent secret in advance, which is incompatible with proof of payment in the lightning network. Proof of payment means that the sender can show the parent secret to prove that the recipient has received the payment.

Basic atomic multipath payments (BAMP) are another multipath payment scheme. BAMP uses the same payment condition (payment hash) in all paths, and the recipient will release the parent secret preimage only after receiving all child payments. This is guaranteed by a financial incentive, since the recipient, by releasing the paternal secret, indicates that he has received all the funds paid.

In addition, Lin et al. proposed a multipath payment mechanism, called Rapido [25], which implements a multipath payment protocol by designing a D-HTLC smart contract. A study [26] similar to that of multipath payment uses new routing methods between the channels of the lightning network by designing more complex routing protocols to speed up the flow of funds between channels.



```
IF
    # Payment if you have the secret R
    HASH160 <H> EQUALVERIFY
ELSE
    # Refund after timeout.
    <locktime> CHECKLOCKTIMEVERIFY DROP
    <Payer Public Key> CHECKSIG
ENDIF
```

Figure 2: An example of HTLC. Only those who know the secret $R$ can redeem the output. If no one redeems the output within a certain period of time, the amount in the contract will be returned the way it was.

## 3. Payment Scheme

In this paper, a new atomic multipath payment scheme is designed. Compared with other schemes, the new scheme supports proof of payment and conforms to the definition of atomicity of payment. The new scheme includes the sender scheme and the receiver scheme. The sending case consists of four parts: parameter determination, secret determination, payment condition construction, and secret sending. The receiver scheme includes the generation of public and private key pairs, secret splicing, the creation of subsecret keys, and payment acceptance. The flow chart of the whole scheme is shown in Figure 3.

### 3.1. The Sender

(1) Parameter determination: determine the amount of payment fund which is $F$, and the number of fund shares is $N$; i.e., the funds are divided into $f_1, f_2, \cdots f_n$ by the sender and $F = f_1 + f_1 \cdots + f_1$

(2) Secret determination: the sender randomly generates a parent secret (ps) and then generates $n$ child secrets for secret sampling. The sampling scheme adopted in this paper is as follows: generate $n-1$ random numbers $x_1, x_2, \cdots x_{n-1}$. The value of the first subsecret $s_1$ is $x_1$, the value of the second subsecret $s_2$ is $x_2$, and so on; the value of the $n-1$st subsecret is $x_{n-1}$, and the value of the last subsecret is $\mathrm{ps} \oplus x_1 \oplus x_2 \cdots \oplus x_{n-1}$. In this sampling method, the parent secret can be restored by combining all the child secrets with XOR operation

(3) Construction of payment terms: the sender uses serial number ID which tags every child pay and the serial number from 1 to $N$, parent secret ps, and receiver's public key $K_{\mathrm{par}}$ structure subpayment terms $K_i = H(i \| \mathrm{ps} \| K_{\mathrm{par}}) * G + K_{\mathrm{par}}$. The receiver needs to know the serial number I, the parent secret ps, and the receiver's private key $k_{\mathrm{par}}$ to redeem the payment

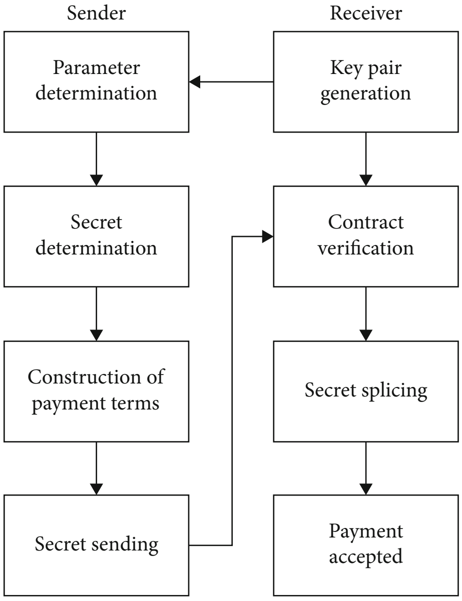


Figure 3: The flow chart of the new scheme. The sender scheme includes parameter determination, secret determination, payment condition construction, and secret sending. The receiver scheme includes the generation of public and private key pairs, secret splicing, the creation of subsecret keys, and payment accepted.

(4) Secret sending: the payer sends the triple (ID, $V_i$, and $S_i$) and the contract containing the payment terms in the previous step

### 3.2. The Receiver

(1) Public and private key pair generation: the receiver generates the public and private key pair and sends the public key to the sender

(2) Contract verification: the receiver waits for the arrival of subpayment, obtains parameter ID of each subpayment, fund amount $f_i$, and the corresponding subsecret $s_i$, and then verifies the format and content of the contract at the same time

(3) Secret splicing: after the arrival of all subpayments, the receiver will perform XOR operation on each $s_i$ obtained, and the parent secret ps is obtained by splicing

(4) Payment accepted: the receiver using the parent secret ps and the serial number ID, as well as the receiver private key $k_{\text{par}}$ and public key $K_{\text{par}}$, calculates all terms of unlocking payment. We use $k_i$ to denote the unlocking condition and $k_i = H(i\|\text{ps}\|k_{\text{par}}) + k_{\text{par}}$; using $k_i$ can obtain each child pay of the payer for the money. When the payment is accepted, the recipient needs to show the private key in the script. Once the private key is shown, it indicates the acceptance of funds, which can be used as proof of payment

In this scheme, the receiver must wait for all transactions to arrive before the parent secret can be concatenated, which ensures the atomicity of the payment. In other words, the recipient cannot say that only part of the payment has been received, because only after receiving all of the child payments can the recipient get the parent secret, and without the parent secret, the recipient cannot get any of the child payments. In addition, after the payment is successful, the sender can present the invoice with $K_{\text{PAR}}$ to prove that the sender has received all the payments, that is, the payment proof supported by the scheme, as shown in Table 1. The new payment protocol mainly acts on the layer 2 network, and the interaction with the main chain does not change; therefore, the protocol changes in this scheme do not affect the performance of the main chain in other ways.

## 4. Simulation

Since it is difficult to observe the channel fund distribution, routing, and other information in the real lightning network, we implement a simulation network to simulate the transactions in the lightning network. To more accurately restore the characteristics of the real network, the simulated network adopts the same channel topology, channel capacity, and routing algorithm as the original network. The initial channel is evenly distributed, with half of the money at each end.

The whole simulated network has a total of 4968 nodes and 59,335 channels. Figure 4 shows the topology structure of some nodes in the simulated network (sorting the number of node channels from high to low, taking the first 100 nodes).

The simulated transactions are divided into 4 groups, each group carries out 9 rounds of transactions, and 100 pairs of 200 nodes are randomly selected for each round. One pair of nodes includes the payer node and the receiver node. Each payer node sends a transaction to the corresponding receiver node using the nonsplit transaction scheme and this scheme, respectively. In theory, a successful routing path can be found as long as the number of split copies is sufficient. In this simulation, we set the maximum number of splits to 10.

In the first group, the amount of money sent for each payment in the first round is 1000 Satoshis. The amount of money sent for each subsequent round of payment is increased by 1000 Satoshis. The amount of money sent for each payment in the second round is 2000 Satoshis, and the amount of money sent for each payment in the last round is 9000 Satoshis. The second group sends 10,000 Satoshis for each payment in the first round, followed by an increase of 10,000 Satoshis for each subsequent round. The third group sent 100,000 Satoshis for each payment in the first round and increased the amount by 100,000 Satoshis for each subsequent round. The fourth group sends 1,000,000 Satoshis for each payment in the first round and then increases the amount sent by 1,000,000 Satoshis for each subsequent round. Considering the wide range of real amounts in the lightning network, we conducted group experiments. In this section, by grouping the payment amounts, the payment amounts differ by a factor of 10 between groups, while the payment amounts within groups are increasing in equal increments. The experimental results show that such a choice can more clearly demonstrate the difference in the routing success rate of the two payment schemes at different amounts.

The success rate of routing in the network decreases as the amount of payment gets larger because whether a payment can go through a channel depends on the amount of money the channel has in the direction of payment. The multipath payment scheme can split a large payment into several small payments. Since the small payments have more channels to choose from in the network, the success rate of routing will also increase. The nonsplit payment scheme and this scheme are used to simulate the payment on the simulation network, respectively, and the routing success

Table 1: Comparison of payment schemes.

| Payment scheme | Proof of payment | One million satoshi routing success rate |
|---|---|---|
| Non-split payment | Supportive | 3.1% |
| AMP | Unsupported | 42.9% |
| Our scheme | Supportive | 42.6% |

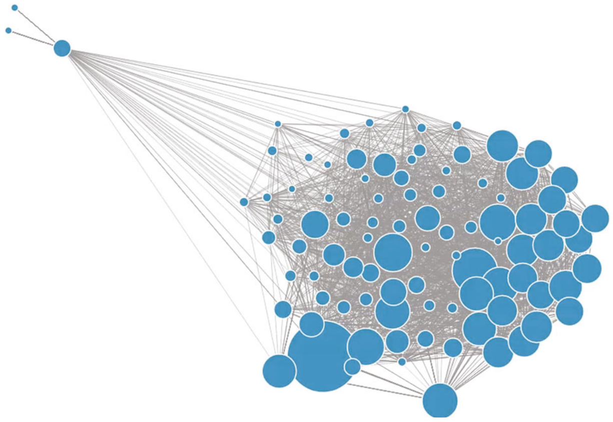


Figure 4: Topology structure of the main nodes in the simulated network. The node size represents the sum of the capacity of all channels connected to the node.

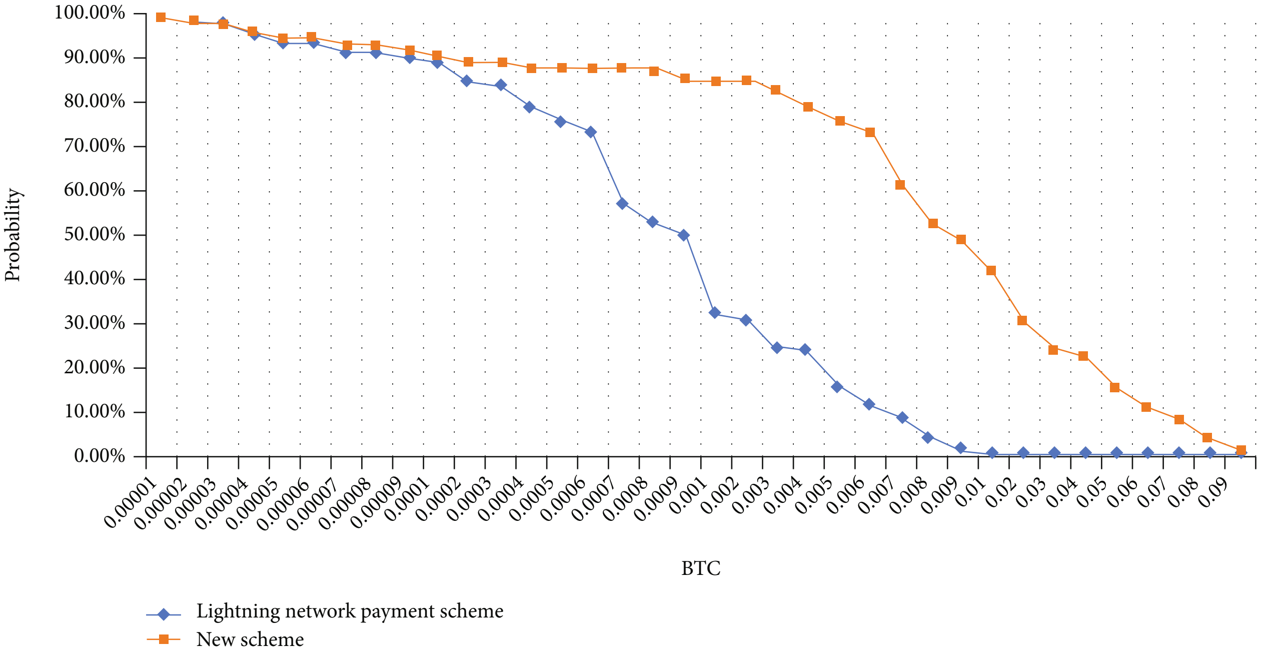


Figure 5: In the simulation network, the routing success rate of this scheme for large payments is significantly higher than that of the original lightning network payment scheme.

rate of the payment is counted. It is verified that the routing success rate of this scheme for large payments is significantly higher than that of the nonmultipath payment scheme, as shown in Figure 5.

## 5. Conclusion

The off-chain payment network is an important solution to the blockchain scalability problem. We propose a new multipath payment scheme that supports proof of payment while retaining the relatively small channel capacity requirements of the original multipath payment scheme for the lightning network. By creating the simulated network and making simulated payment, the advantages of the scheme in this paper are verified in terms of the routing success rate. In the future, we plan to design a fund partitioning algorithm to improve the fund balance in the channel. We also consider taking the routing fees into consideration in the partitioning algorithm to further reduce the routing cost. In addition, whether the new payment protocols will have an impact on the master chain is subject to further study. In addition, we intend to increase the size of the simulated transactions, so that the simulated results are closer to the results of the real scenario.

## Data Availability

All the data used are given in the paper.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] S. Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, 2009, https://bitcoin.org/en/bitcoin-paper.

[2] J. Wang and H. Wang, “Monoxide: scale out blockchains with asynchronous consensus zones,” *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, vol. 2019, pp. 95–112, 2019.

[3] J. Poon and T. Dryja, *The lightning network: scalable off-chain instant payments [EB/OL]*https://www.blockchainlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf.

[4] O. Osuntokun, *Amp: atomic multi-path payments over lightning[EB/OL]*https://lists.linuxfoundation.org/pipermail/lightning-dev/2018-February/000993.html.

[5] G. Wood, “Ethereum: a secure decentralised generalised transaction ledger,” *Ethereum Project Yellow Paper*, vol. 5, no. 1, pp. 1–32, 2014.

[6] E. Androulaki, A. Barger, V. Bortnikov et al., “Hyperledger fabric,” in *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, New York, 2018.

[7] B. S. Bhati and C. S. Rai, “Intrusion detection technique using coarse Gaussian SVM,” *International Journal of Grid and Utility Computing*, vol. 12, no. 1, pp. 27–32, 2021.

[8] S. P. Yadav, K. K. Agrawal, B. S. Bhati, F. Al-Turjman, and L. Mostarda, “Blockchain-based cryptocurrency regulation: an overview,” *Computational Economics*, vol. 16, pp. 1–17, 2020.

[9] N. Webb, “A fork in the blockchain: income tax and blockchain cash hard fork,” *North Carolina Journal of Law & Technology*, vol. 19, no. 4, p. 283, 2018.

[10] M. Corallo, *BIP 152: compact block relay[EB/OL]*https://github.com/blockchain/bips/blob/master/bip-0152.html.

[11] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, “A secure sharding protocol for open blockchains,”

in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 17–30, New York, 2016.

[12] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 931–948, New York, 2018.

[13] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: a secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 583–598, IEEE, San Francisco, CA, USA, 2018.

[14] I. Bentov, R. Pass, and E. Shi, "Snow white: provably secure proofs of stake," *IACR Cryptology ePrint Archive*, vol. 11, no. 3, pp. 36–49, 2016.

[15] I. Eyal, A. E. Gencer, E. G. Sirer, R. van Renesse et al., "Blockchain-ng: a scalable blockchain protocol NSDI," vol. 21, pp. 45–59, 2016.

[16] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 51–68, New York, 2017.

[17] H. Hees, "Raiden network: off-chain state network for fast DApps," in *Devcon two*, Ethereum Foundation, 2016.

[18] J. Poon and V. Buterin, "Plasma: scalable autonomous smart contracts," https://www.plasma.io/plasma-deprecated.pdf.

[19] A. M. Rahmani, M. Mohammadi, S. Rashidi et al., "Questioning the security of three recent authentication and key agreement protocols," *IEEE Access*, vol. 9, pp. 98204–98217, 2021.

[20] M. Hosseinzadeh, A. M. Rahmani, and B. Vo, "Improving security using SVM-based anomaly detection: issues and challenges," *Soft Computing*, vol. 25, no. 4, pp. 3195–3223, 2021.

[21] N. Maleki, A. M. Rahmani, and M. Conti, "SPO: a secure and performance-aware optimization for MapReduce scheduling," *Journal of Network and Computer Applications*, vol. 176, article 102944, 2021.

[22] S. Akhbarifar, H. H. S. Javadi, A. M. Rahmani, and M. Hosseinzadeh, "A secure remote health monitoring model for early disease diagnosis in cloud-based IoT environment," in *Personal and Ubiquitous Computing*, pp. 1–17, Springer, 2020.

[23] *Blockchain for Cybersecurity and Privacy: Architectures, Challenges, and Applications*, CRC Press, 2020.

[24] Z. Sisi and A. Souri, "Blockchain technology for energy-aware mobile crowd sensing approaches in Internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 11, no. article e4217, 2021.

[25] C. Lin, N. Ma, X. Wang, and J. Chen, "Rapido: scaling blockchain with multi-path payment channels," *Neurocomputing*, vol. 406, pp. 322–332, 2020.

[26] V. Sivaraman, S. B. Venkatakrishnan, K. Ruan et al., "High throughput cryptocurrency routing in payment channel networks," *Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, vol. 7, pp. 777–796, 2020.