A Secure and Efficient Bitcoin Payment Channel Using Intel SGX

Yankai Xie*, Chi Zhang*, Lingbo Wei†*, Qingtao Wang* and Zhe Yang*
*School of Cyber Science and Technology
University of Science and Technology of China, Hefei, Anhui 230027, P. R. China
†China Nanhu Academy of Electronics and Information Techology, Jiaxing, Zhejiang 314001, P. R. China
E-mails: ykxie@mail.ustc.edu.cn, {chizhang, lingbowei}@ustc.edu.cn, {wqt, yzpag}@mail.ustc.edu.cn

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500569

Abstract—**Hardware trusted execution environment (TEE) provided by Intel SGX enclave has been introduced in existing payment channel schemes as a root-of-trust to enforce faithful protocol execution so that participants do not need to monitor Bitcoin blockchain anymore. However, the security of these schemes relies totally on enclaves. Since private keys of all channel funds are kept by both payment channel participants' enclaves, a malicious participant can steal funds from the counterparty by defeating her own enclave. To solve the above problem, we present a novel TEE-based payment channel scheme that transfers the responsibility of running enclaves from participants to a third party committee, while relieving both participants from monitoring the blockchain at the same time. Furthermore, since committee members can try to steal funds by defeating their own enclaves, we exploit the additive homomorphic property of signature keys in Elliptic Curve Cryptography to design a novel secret sharing scheme to tolerate a subset of committee members to be malicious. By using the above secret sharing scheme, private keys of the channel funds are never constructed in any committee member's enclave, so that a malicious committee member cannot steal funds by defeating his own enclave. Finally, experiment shows our scheme can ensure payment channel funds security without efficient compromises compared with existing TEE-based payment channel schemes.**

Keywords—**Bitcoin, Payment channel, Trusted execution environment, Intel SGX, Secret sharing**

I. INTRODUCTION

In terms of market capitalization and the total number of users, Bitcoin [1] has been the leading cryptocurrency since its inception. Unfortunately, due to its underlying consensus protocol, Bitcoin suffers from limited transaction throughput [2]. To improve transaction throughput without modifying the consensus protocol, the proposed Lightning Network [3] allows Bitcoin users to execute off-chain transactions while maintaining the security guarantees of the blockchain. Interconnected bilateral payment channels are the basic elements of the Lightning Network. Specifically, parties publish an on-chain transaction to lock their funds, which opens a payment channel. By generating off-chain commitment transactions privately, they adjust the funds distribution in the channel to transfer value. At any time, either party can settle the channel by uploading the latest commitment transaction to the Bitcoin blockchain, which pays each party the number of funds they own in the channel.

However, a malicious party can publish an outdated commitment transaction to steal counterparty's funds in the channel. Unless the counterparty disputes it within a time period which pre-defines during channel initialization, this channel will pay each party based on the outdated commitment transaction. Specifically, if a commitment transaction is included in the blockchain, it cannot be disputed anymore after t blocks mined. Within time of t block mined, all channel funds will be paid to the counterparty to punish malicious party if the counterparty disputes it. Unfortunately, this solution has some drawbacks [4]. Firstly, each party has to access the blockchain at least every t block mined to ensure the funds security. If not, malicious party could steal funds. Moreover, an honest party cannot obtain her funds immediately if she settles payment channel by uploading the latest commitment transaction. She would obtain her own funds after t blocks mined. Finally, to provide each other with means to cancel outdated commitment transaction, the above designs complicate the payment channel protocol and increases blockchain cost.

To overcome the above drawbacks, Lind et al. [5] explore a design for payment channel in which each party controls a trusted execution environment (TEE) as a root-of-trust to enforce faithful protocol execution. They implement Teechan using TEE provided by Intel Software Guard Extensions (SGX) [6] enclaves that isolate execution environment from untrusted part, such as operating system and other applications, in SGX host. By doing so, either party has administrative privileges over the untrusted part cannot affect confidentiality of data or the execution of code in the enclave. In Teechan, each party in a payment channel runs an enclave to conduct off-chain transactions. Parties construct payment channel via locking funds in an address whose corresponding private key is controlled by both enclaves. Parties conduct off-chain transactions by updating the record of channel funds allocated in both enclaves. Either party can instruct her enclave to sign and publish commitment transaction to settle the payment channel and withdraw her own funds. The code in the enclave rejects malicious instruction, such as settling the payment channel under an outdated record, so the scheme overcomes

This work was supported by the National Key Research and Development Program of China under grant 2018YFB0804201, and by the Natural Science Foundation of China (NSFC) under grants 61702474, 62072426, U19B2023, and 61871362.

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

the drawbacks of lightning network payment channel.

Although TEEchan addresses the issues of typical payment channel, potential security threat of SGX puts the channel funds at risk. Since the funds are locked in an address controlled by parties' enclaves, TEEchan cannot ensure withdrawal of all Bitcoin if either party defeats the hardware defence provided by SGX enclave. The work in [7] has shown the confidentiality of SGX suffers from a great number of attacks. Those attacks assume that attacker holds host and controls the untrusted part. By reinitializing enclave, replaying inputs, and so on, an attacker knowing the mapping of memory addresses to cache lines can obtain confidentiality data in the enclave. The security of SGX has become a wide concern in both academia and industry.

To mitigate against compromised enclaves, the work in [4] combines m of n multi-signature with different Bitcoin full nodes to enhance channel funds security. Parties select n enclaves running in different Bitcoin full nodes to constitute a committee that uses m of n multi-signature address to lock the payment channel funds. Either party can settle the payment channel via interacting with m enclaves. Even if colluding with $m-1$ enclave hosts which defeat the hardware defence, the attacker cannot steal the channel funds. Unfortunately, parties would lose full control of their funds. Specifically, parties cannot settle the payment channel without cooperating with the committee. Moreover, existing m of n multi-signature in Bitcoin are highly restricted due to Bitcoin redeem script size [8], [9]. Thus, it is difficult for parties to customize a secure payment channel settlement strategy without modifying the Bitcoin protocol. To make matters worse, either party cannot withdraw the channel funds if more than $n-m$ enclaves fail.

In this paper, we propose a novel TEE-based payment channel scheme that introduces a third party committee composed of enclaves as arbiter to participate in the payment channel. Without cooperating with the committee, parties can still settle the payment channel, which outperforms the above solutions such as Teechain [4]. To achieve that, we utilize 2 of 3 multi-signature to design our payment channel scheme. Each party controls one of the three private keys and the whole committee controls another (called arbitration private key). Thus parties can safely settle the payment channel even if all enclaves fail.

Since enclave hosts can try to steal the arbitration private key by defeating their own enclaves, enclaves use Shamir secret sharing to distribute the shares of the arbitration private key to all enclaves for tolerating $m-1$ enclave hosts to be malicious. However, the arbitration private key is still constructed in enclaves when the payment channel is initializing or settling. To address the issue, we utilize additive homomorphic property of keys in Elliptic Curve Cryptography (ECC) [10] to design a novel secret sharing protocol, and design the interaction procedures between payment channel participants. Even if $m-1$ enclave hosts collude with a counterparty and defeat the hardware defence in the whole cycle life of payment channel, they cannot steal the payment channel funds because the arbitration private key has never been constructed in any enclave. Our main contributions are summarized below:

- We use additive homomorphic property of keys in ECC to design a novel secret sharing scheme. Without constructing private key, enclaves can construct corresponding public key and distribute shares of the private key.
- We design a new TEE-based payment channel using the above secret sharing scheme, which allows parties to withdraw their own funds when no more than $m-1$ collusive enclave hosts defeat the hardware defence or all enclaves fail.
- We demonstrate that our scheme allows Bitcoin users to construct a secure yet efficient payment channel to transfer value.

The rest of this paper is organized as follows. Section II covers overview of additive homomorphic property of keys in ECC and secret sharing scheme. Section III describes the system model, some assumptions and design goals. Section IV presents the detail of our scheme. Section V gives security analysis of the scheme, and performance analysis is shown in Section VI. Ultimately, we conclude in Section VII.

II. PRELIMINARIES

A. Additive Homomorphic Property of Keys in ECC

Elliptic Curve Cryptography used in Bitcoin signature algorithm has the following additive homomorphic property of keys [10]: for any number of key pairs, the sum of private keys and the sum of corresponding public keys are still the valid key pair. For example, the sum of private keys is:

$$sk = (sk_1 + sk_2 + ... + sk_n)\ mod\ p \tag{1}$$

and the sum of corresponding public keys is:

$$\begin{aligned} pk &= pk_1 + pk_2 + ... + pk_n \\ &= (sk_1 + sk_2 + ... + sk_n) \cdot G \\ &= sk \cdot G \end{aligned} \tag{2}$$

Thus, the (sk, pk) is a new key pair.

B. Secret Sharing Scheme

Secret sharing [11] refers to the method which distributes a secret amongst a group of participants. The secret can be reconstructed only when a sufficient number of shares are combined together. The secret sharing scheme proposed by Shamir provides an elegant (t, n)-threshold construction using the Lagrange interpolation algorithm. Imaging that there is a group contains n participants, and a third party gives a share of the secret to each participant. A sub-group contains t or more participants (for threshold) can reconstruct the secret together, but a sub-group of fewer than t participants cannot. Such protocol is called (t, n)-threshold secret sharing scheme.

III. PROBLEM STATEMENT

A. System Model

As shown in Fig. 1, the TEE-based payment channel we proposed involves three entities: the committee composed of enclaves running in full nodes, transaction party A, and transaction party B. The enclave provides the trusted execution environment to separate the code and data from the untrust part in full node. Each enclave running the same code connects to the Bitcoin network to sync the blockchain and assist parties to construct payment channel. Before constructing payment channel, all participants execute remote attestation which validates the open source code in the enclaves and constructs the secure connections.

Figure 1 also shows the life cycle of payment channel. When parties reach an agreement, they create or update payment channel with the help of enclaves selected randomly as a committee. To create a payment channel, parties assist the committee to generate commitment transaction (CTx) to allocate channel funds, and then lock their funds in the channel by publishing funding transaction (FTx) to the blockchain. When transferring value, they update the commitment transaction in the committee to reallocate channel funds. Finally, either party can instruct the committee to use the latest commitment transaction to settle the payment channel without connecting to the counterparty (Figure 1 uses B as an example).

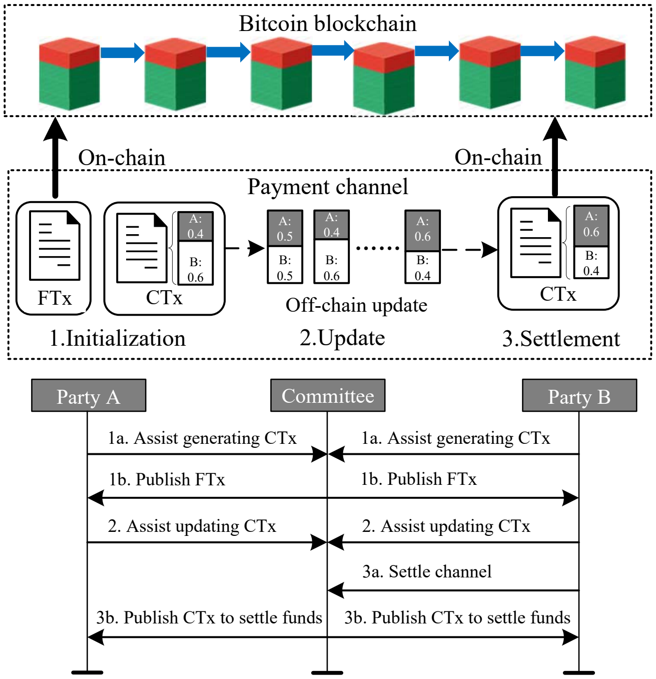

Fig. 1. **System model**

B. Threat Model

On one hand, parties and full nodes are rational, selfish and potentially malicious. Full nodes may attempt to steal funds in the payment channel or deviate from the payment protocol if they can obtain benefit. Moreover, a subset of malicious full nodes (less than m) can still collude with a counterparty to steal funds in the payment channel.

On the other hand, we assume that enclaves running on the full nodes are normally trustworthy, but the full nodes which hold the enclaves may obtain application confidential data in the enclaves via unknown security vulnerabilities. In addition, some full nodes may delay or prevent parties from accessing enclaves, or stop running enclaves. In the worst case, parties cannot establish connections with any enclave. If so, they cannot settle the payment channel via enclaves.

C. Design Goals

In view of above threat model, the following security requirements are necessary to ensure that enclaves construct TEE-based payment channel in security yet efficient way:

- Security: parties can safely transfer value via the payment channel. Even though a subset of malicious full nodes (less than m) colluding with a counterparty defeat the hardware defence, they cannot steal funds locked in the payment channel.
- Reliability: parties can withdraw all funds in the payment channel even if all enclaves fail.
- Privacy: unless defeating the hardware defence, any full node cannot know any off-chain transaction detail.
- Efficiency: we wish to maximize the channel throughput and minimize blockchain cost of our scheme.

IV. System Design

The life cycle of a payment channel requires three phases: initialization, update and settlement. In the initialization phase and the update phase, committee uses the novel secret sharing scheme we proposed to generate arbitration key pair. Thus, we first introduce the secret sharing scheme.

A. The Novel Secret Sharing Scheme

Assuming that parties have randomly selected n enclaves in full nodes as the arbitration committee to participate in the full life cycle of the payment channel. All enclaves in the committee should collaborate to generate an arbitration key pair and distribute the shares of arbitration private key to all enclaves. To improve the efficiency, parties designate an enclave in the committee as the leader.

The following steps are required to generate an arbitration key pair. First, each enclave, E_i $(1 \leq i \leq n)$, uses the ECC key generation algorithm to generate a key pair, private key sk_i and public key pk_i. Note that each private key sk_i is a seed of the arbitration private key. Then, in order to generate the arbitration public key, each enclave uploads the generated pk_i to the leader enclave. The leader enclave constructs the arbitration public key $APK = \sum_{i=1}^{n} pk_i$ according to the additive homomorphic property of keys. The private key corresponding to APK, $ASK = \sum_{i=1}^{n} sk_i$, would not appear in any enclave. The seeds of ASK are respectively stored in each enclave, and each enclave would not upload the seed to the leader enclave.

Next, each enclave uses Shamir secret sharing scheme to distribute the shares of each private key seed its holds to all enclaves. To distribute seed sk_i, the E_i chooses random elements $a_1, ..., a_{m-1}$ to construct a polynomial $P_i(x) = sk_i + \sum_{j=1}^{m-1} a_j x^j$. Then, the E_i computes $s_{ik} = P_i(\alpha_{ik})$, $(1 \leq k \leq n)$, and sends (α_{ik}, s_{ik}) to corresponding E_k

in committee. Finally, each enclave E_k holds the shares, $(\alpha_{ik}, s_{ik}), 1 \leq i \leq n$, of each seed sk_i.

Through this scheme, parties obtain the arbitration public key, and the arbitration private key does not appear in any enclave. Even if $m-1$ collusive hosts defeat the hardware defence, they cannot construct the arbitration private key.

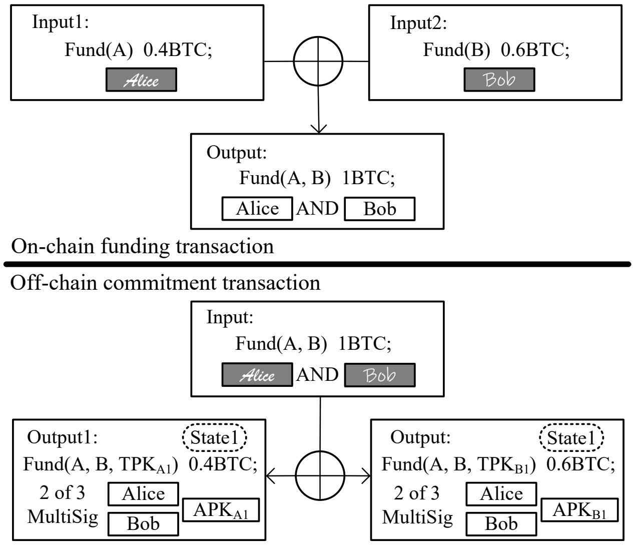

Fig. 2. **Payment channel initialization**. Input of the transaction has been signed (black box, grey background, and white text), and outputs of the transaction lock the funds in a 2 of 3 multi-signature address (black box, white background, and black text).

B. Initialization

In the initialization phase, parties randomly select n enclaves in Bitcoin network as a committee. They validate the code in the enclaves and construct secure connections by remote attestation. Moreover, they need to generate funding transaction and upload it to the blockchain to lock the funds. Furthermore, they also need to use the arbitration public keys to generate a commitment transaction and upload it to the committee. After that, either party can settle the payment channel with the committee at any time.

After constructing secure connections, all enclaves use the scheme in section IV-A to generate two arbitration key pairs and send two arbitration public keys, APK_{A1} and APK_{B1}, to parties. Then parties generate on-chain and off-chain transactions to complete payment channel initialization. For example, as shown in Fig. 2, they generate a funding transaction to lock their funds in a 2 of 2 multi-signature script controlled by A and B. In order to avoid locking the funds if either party fails during the initialization, parties use segregated witness [12] to obtain the output of the unsigned funding transaction to generate commitment transaction. Without segregated witness, parties cannot obtain the output unless signing the transaction.

Subsequently, parties cooperate with the committee to create an initial commitment transaction to allocate the channel funds of parties. In order to improve efficiency, parties interact with the committee through the leader enclave. The commitment transaction includes one input and two outputs. The input corresponds to the output of the funding transaction, and the two outputs correspond to the respective channel funds of parties. For example, as shown in Fig. 2, one output of the commitment transaction locks 0.4 BTC in the 2 of 3 multi-signature script controlled by A, B, APK_{A1}, and the other 0.6 BTC is locked in the 2 of 3 multi-signature script controlled by A, B, APK_{B1}. Parties sign the commitment transaction with their private keys separately, and upload the transactions which only have a signature to the leader enclave. The leader enclave extracts signatures from the uploaded transaction to construct a valid commitment transaction. To be resilient against enclave failures, the leader enclave broadcasts the commitment transaction to all enclaves in the committee. Finally, parties sign the funding transaction and upload it to the blockchain to complete the initialization. This is the initial state of the payment channel, also called State 1.

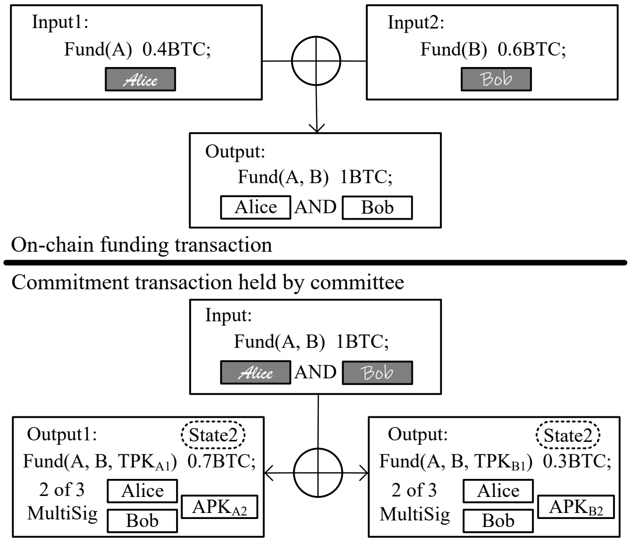

Fig. 3. **Payment channel update**

C. Update

Parties would cooperate with committee to update the commitment transaction when transferring value, which ensures either party can settle the payment channel by uploading latest commitment transaction with the help of committee.

By generating and uploading new commitment transaction to the committee through leader enclave, parties can update the payment channel. To ensure funds security if the enclave cannot guarantee confidentiality of data, parties update arbitration key pairs used in the payment channel. Thus, the leader enclave coordinates the other enclaves to generate new arbitration key pairs via the scheme in section IV-A. For instance, as shown in Fig. 3, when parties update the payment channel from State 1 to State 2, the committee provides new arbitration public keys APK_{A2} and APK_{B2} to parties, and distributes new arbitration private key seeds shares to all enclaves. Since modifying the output of the outdated commitment transaction would invalidate the signature, parties regenerate a new commitment transaction to replace the outdated commitment transaction.

To update to the State 2, the leader enclave constructs the new commitment transaction based on the signatures uploaded from parties. Parties construct a new unsigned commitment transaction. Similar to the outdated commitment transaction, the input of the new commitment transaction corresponds to the output of the funding transaction. The difference is that the new arbitration public keys are used in the locking script of new commitment transaction outputs. As shown in Fig. 3, if B transfers 0.3 BTC to A, one output of the new commitment transaction locks 0.7 BTC in the 2 of 3 multi-signature script controlled by A, B, APK_{A2}, and the other locks 0.3 BTC in the 2 of 3 multi-signature script controlled by A, B, APK_{B2}. Parties sign the commitment transaction with their private keys separately, and upload the transactions which only have a signature to the leader enclave.

Finally, leader enclave constructs the valid commitment transaction of State 2 by extracting the signatures, and broadcasts it to all enclaves. After that, each enclave deletes outdated data, including the outdated commitment transaction and shares. By doing so, we minimize the overhead of enclaves.

D. Settlement

Either party can settle the payment channel at any time. Our scheme provides two methods to settle payment channel. One requires both parties are online, and the other allows either party settles payment channel with the help of the committee.

If both parties are online, they can directly spend the output of the funding transaction to settle the payment channel. Parties generate a settlement transaction, which contains two outputs corresponding to their respective latest shares of channel funds. Then they sign and upload the transaction to blockchain to withdraw their funds. When the transaction is conformed by blockchain, all enclaves in the committee know that the payment channel has been closed. Thus, they delete all data about the payment channel to minimize storage overhead.

If one party is not online, counterparty can settle the payment channel with the help of the committee. In order to prevent the enclave which does not have the latest commitment transaction from uploading the outdated commitment transaction to blockchain, it needs to synchronize the latest commitment transaction with at least $m-1$ enclaves. Then the leader enclave uploads the latest commitment transaction to blockchain. By obtaining seeds shares of the arbitration private key, either party can independently withdraw owned funds. For example, as mentioned before, to withdraw 0.7 BTC in the payment channel, party A should retrieve the arbitration private key ASK_{A2}. Thus, she queries the seeds shares $(\alpha_{ik}, s_{ik}), 1 \leq i \leq n$ from enclave $E_k, 1 \leq k \leq n$. Each enclave responses when the latest commitment transaction is confirmed by blockchain. Once m shares are collected, she constructs each $sk_i, 1 \leq i \leq n$ through computing:

$$sk_i = \sum_{\gamma=1}^{m} s_{i\gamma} \prod_{1 \leq j \leq m, j \neq \gamma} \frac{\alpha_{ij}}{\alpha_{ij} - \alpha_{i\gamma}} \tag{3}$$

Afterwards, through the additive homomorphic property of keys, she can compute the private key $ASK_{A2} = \sum_{i=1}^{n} sk_i$. Party A controls the two private keys of the 2 of 3 multi-signature, so she can spend the output. When B goes online, he can use the same way to construct the ASK_{B2} to withdraw his funds.

V. Security Analysis

In this section, we show the security analysis of the proposed scheme to verify that it meets the security requirements mentioned in Section III-C.

A. Security

The proposed scheme could tolerate $m-1$ malicious full nodes who can defeat the hardware defence and collude with either transaction party at the same time. In the whole life cycle of the payment channel, no arbitration private key appears in any enclave. Each enclave only holds the shares of the arbitration private key seeds. Thus, a full node hosting an enclave can only obtain seeds shares even though defeating the hardware defence in any phase. Even if colluding with a party, they cannot generate the redeem script of the 2 of 3 multi-signature to steal funds from payment channel.

Moreover, our scheme still ensures the channel funds security if malicious full nodes obtain the outdated commitment transaction from enclave. Even if the arbitration private key cannot be constructed, malicious full nodes may upload the outdated commitment transaction to blockchain to destroy the payment channel. Since the arbitration private keys corresponding to the outdated commitment transaction have been destroyed, it prevents the payment channel from settling in the outdated state. To withdraw funds, selfish parties could use their private keys to generate the redeem script of the 2 of 3 multi-signature and upload the transaction to blockchain.

B. Reliability

First of all, the proposed scheme allows either party to settle the payment channel at any time. Even if a transaction party is not online, the other party can settle the payment channel via interacting with the enclaves in the committee. If both transaction parties are online, they can directly settle the payment channel without connecting to committee.

Then, we use the secret sharing scheme to tolerate a subset of enclaves to be failure. In our scheme, by interacting with m enclaves, either party can obtain the seeds shares. Then she can construct the arbitration private key to withdraw her funds via secret sharing scheme and additive homomorphic property of keys. Comparing with TEEchain [4], our scheme overcomes the number limit of committee members caused by the size limit of the redeem script, parties can customize suitable number of members according to need.

C. Privacy

The proposed scheme protects the off-chain transaction privacy of parties from full nodes. The communication between all enclaves and parties utilizes secure channels, which means

TABLE I
THROUGHPUT OF DIFFERENT STRATEGIES

Parameter	Throughput of a committee (tx/sec)	Throughput of Bitcoin network (tx/sec)
m=4, n=6	1,832	3,237,630
m=5, n=8	1,490	1,973,929
m=7, n=10	1,111	1,178,040
m=8, n=12	894	789,395

TABLE II
BLOCKCHAIN COST OF DIFFERENT PAYMENT CHANNELS

Payment channel	Transaction number (Both parties are online)	Transaction number (one party is online)
Lightning Network	2	3
TEEchain	2	2
our scheme	2	2

that malicious full nodes cannot decode the transaction detail through secure channels. Moreover, protocol is performed in the enclaves which ensure the confidentiality and integrity of data and code. Unless defeating the hardware defence, the full nodes know nothing about the off-chain transaction detail. Even if defeating the hardware defence, full nodes cannot obtain off-chain transaction history because enclaves have deleted the data to minimize storage overhead.

VI. PERFORMANCE EVALUATION

A. Performance of Payment Channels

By utilizing existing implementations of SGX SDK and Crypto ++ [13], we develop a prototype implementation to show throughput of our scheme. We implement and evaluate performance on n instances, which have four core Intel i7-8700 @ 3.20 GHz CPU, 8 GB RAM, and 512 GB HDD.

In our scheme, executing an off-chain transaction requires the enclaves in the committee to generate two arbitration public keys and distribute their private key seeds shares, which is the main overhead of the proposed scheme. To ensure funds security, parties customize their own arbitration key generation strategy which defines security-threshold m and the total enclaves number n. Those parameters are related to the performance of the scheme. To evaluate the performance of different strategies, we vary the m and the n.

Table 1 shows the observed throughput of our scheme. Based on the parameters, we show the throughput of a committee. With the pipelining payment, a committee can perform about 900 tx/sec under m=8 and n=12. According to the total number of active full nodes in the Bitcoin network, we speculate the maximum throughput of the scheme when each full node supports the payment channel. In brief, there are about 10,600 active full nodes in Bitcoin network [14]. They can build up about $10600/n$ committees to provide the payment channel services for Bitcoin users. Thus, Bitcoin network based on our scheme can perform approximately millions of transactions per second. As the number of full nodes increases, the throughput will be further improved.

B. Blockchain Cost

We evaluate the number of transactions placed on the blockchain. Optimizing on-chain transaction number can minimize transaction fee of using payment channel. Table 2 shows the number of on-chain transactions placed on the blockchain of classic payment channel schemes. Compared with other schemes, our scheme minimizes on-chain transactions number, which decreases transaction fee of using payment channel.

VII. CONCLUSION

Without modifying consensus, payment channel significantly improves Bitcoin blockchain throughput through off-chain transactions, but classic payment channel user should access blockchain within bounded time to ensure channel funds security. Existing TEE-based schemes address the issue and further decrease the blockchain cost, but those schemes cannot ensure funds security if either enclave host defeats the hardware defence. To address the issue, we design a new TEE-based payment channel using a novel secret sharing scheme we proposed without modifying Bitcoin protocol. We have shown that our scheme allows Bitcoin users to construct efficient yet secure TEE-based payment channel. Even if $m-1$ collusive enclave hosts defeat the hardware defence, the scheme still ensures channel funds security.

REFERENCES

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] “Blockchain charts.” [Online]. Available: https://www.blockchain.com/charts

[3] J. Poon and T. Dryja, “The bitcoin lightning network: Scalable off-chain instant payments,” 2016. [Online]. Available: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf

[4] J. Lind, O. Naor, I. Eyal, F. Kelbert, E. G. Sirer, and P. Pietzuch, “Teechain: a secure payment network with asynchronous blockchain access,” in *ACM Symposium on Operating Systems Principles*, Huntsville, Canada, October 2019.

[5] J. Lind, I. Eyal, P. Pietzuch, and E. G. Sirer, “Teechan: payment channels using trusted execution environments,” arXiv, 2016. [Online]. Available: https://arxiv.org/abs/1612.07766

[6] V. Costan and S. Devadas, “Intel SGX explained,” IACR Cryptology ePrint Archive, 2016. [Online]. Available: http://css.csail.mit.edu/6.858/2020/readings/costan-sgx.pdf

[7] A. Nilsson, P. N. Bideh, and J. Brorsson, “A survey of published attacks on intel SGX,” Lund University, 2020. [Online]. Available: https://portal.research.lu.se/portal/files/78016451/sgx_attacks.pdf

[8] “Length of redeemscript,” Bitcoin Forum. [Online]. Available: https://bitcointalk.org/index.php?topic=615250.0

[9] “What are the limits of m and n in m-of-n multisig addresses?” Stack Exchange. [Online]. Available: https://bitcoin.stackexchange.com/questions/23893/what-are-the-limits-of-m-and-n-in-m-of-n-multisig-addresses

[10] N. P. Smart, “Elliptic curve cryptosystems over small fields of odd characteristic,” *Journal of Cryptology*, vol. 12, no. 2, pp. 141–151, 1999.

[11] A. Beimel, “Secret-sharing schemes: a survey,” in *International Conference on Coding and Cryptology*, Qingdao, China, May 2011.

[12] “Segregated witness,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/SegWit

[13] “Crypto++.” [Online]. Available: https://www.cryptopp.com/

[14] “Bitcoin full nodes number.” [Online]. Available: https://bitnodes.io/