Research on Data Verification and Exchange of Heterogeneous Blockchains for Electricity Application

Hongkai Wang¹, Dong He¹, Yan Gao², Xiaoyi Wang³, Caichao Xu³, Weiwei Qiu³, Yiyang Yao¹ and Qiang Wang²*¹

¹State Grid Zhejiang Electric Power Corporation Information & Telecommunication Branch. Room 641, No. 8 Huanglong Road, Hangzhou 310000, China
²College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
³Hangzhou Qulian Technology Co., Ltd. 20F, Venture Building 2, No. 399 Danfeng Road, Binjiang District, Hangzhou 310000, China
Email: wangqiang@zju.edu.cn

Abstract. This paper proposes a verification and exchange method for cross-chain transactions of heterogeneous blockchains for electricity application scenarios and designs a chain-level cross-chain event verification mechanism. For a blockchain platform that requires cross-chain transactions, a Merkel-structured cross-chain event tree is systematically constructed with the collected cross-chain transactions in the generation stage of the block. The target blockchain platform verifies the credibility of cross-chain events based on the Merkel-structured cross-chain event tree and related credible proofs. Cross-chain events are captured through smart contract events, and cross-chain event Merkel trees and Merkel paths are constructed to achieve credible and rapid verification of cross-chain events. The cross-chain experiments are carried out in the main-side chain mode of electricity business scenario, and the integrity of cross-chain transaction is ensured by a cross-chain transaction consistency guarantee mechanism, so as to achieve the purpose of trusted circulation and secure transactions of electricity business data between different application chains.

1. Introduction

With the development of blockchain technology and the emergence of more application scenarios, it is an urgent need to communicate and share data between different blockchains. The current cross-chain technologies can be roughly divided into three categories: notary mechanism, side chain/relay chain technology and hash lock. In these cross-chain technologies, the methods of data verification and exchange are different.

The notary mechanism is the simplest in cross-chain technology. One or more trusted groups are responsible for determining the correctness of cross-chain events. These groups can automatically monitor and respond to events. The representative of the notary mechanism is the Ripple Interledger protocol. In 2012, Ripple Lab proposed the Interledger protocol [1-2], which aims to connect different ledgers and achieve collaboration between them. The Interledger protocol is applicable to all accounting systems and can tolerate the differences of all accounting systems. The goal of the protocol is to create a unified global payment standard and create a unified network financial transmission protocol.

Sidechain [3-5] is a new type of blockchain based on anchoring tokens on a certain original chain. It is a fusion way to achieve the goal of cryptocurrency financial ecology, rather than excluding existing
systems like other cryptocurrencies. Using the sidechain, we can easily establish various intelligent financial contracts, stocks, futures, derivatives, etc. You can have tens of thousands of sidechains anchored to Bitcoin, with different characteristics and purposes. All these sidechains depend on the flexibility and scarcity guaranteed by the Bitcoin main blockchain. On this basis, the side chain technology further expands the application scope and innovation space of blockchain technology, so that the traditional blockchain can support a variety of asset types, as well as small and micro payment, smart contract, security processing mechanism, real world property registration, etc., and can enhance the privacy protection of blockchain. The more well-known Bitcoin sidechains are the BTC relay of ConsenSys, Rootstock and the Elements of Blockstream, and non-Bitcoin sidechains include Lisk and domestic Asch. BTC Relay [3-4] allows users to verify Bitcoin transactions on the Ethereum blockchain by using Ethereum’s smart contract function, connecting the Ethereum network and the Bitcoin network in a secure and decentralized manner.

Lightning network provides a scalable bitcoin micropayment channel network, which greatly improves the transaction processing capacity of the Bitcoin network chain. If both parties of the transaction have a payment channel in advance on the blockchain, they can realize micro-payments with rapid confirmation multiple times, high-frequency, and two-way. If there is no direct peer-to-peer payment channel between the two parties, as long as there is a payment path in the network that connects the two parties and is composed of multiple payment channels, the lightning network can also use this payment path to achieve the reliable transfer of funds between the two parties. The key technology of Lightning Network is HTLC (Hashed Timelock Contract) [6], the basic principle of which is as follows: Alice and Bob can reach an agreement: the agreement will lock Alice’s 0.1 BTC before T arrives (T is expressed in the height of a certain blockchain in the future), if Bob can show Alice an appropriate R, which is called a secret, so that the hash value of R is equal to the pre-agreed value H(R), Bob can get this 0.1 BTC; if Bob still fails to provide a correct R until time T, this 0.1 BTC will be automatically unfrozen and returned to Alice.

In June 2014, the concept of Relay was proposed in the Chain fibers scheme. In this scheme, a single relay-chain and multiple homogeneous chains are proposed for cross-chain data exchange, which is realized by using the characteristics of Decoherence and Transaction Lantecy to coordinate the transactions of multiple parts in the system [7]. Various cross-chain projects including Polkadot [8], Cosmos [9] have been improved on this basis, enriching the original cross-chain architecture so that it can be used for data and asset exchange in a variety of heterogeneous chains. Polkadot introduces a special state verification method to facilitate the verification of the validity of messages across the chain, including four special roles to participate in verification.

- **Validator**: The Validator is responsible for Polkadot's network block generation. The Validator will run a client of the relay chain and verify the block that is nominated by the parallel chain in each round of block generation. After the parallel chain spans are determined by their set of sub-validators, the Validators will assemble all the parallel chain block headers into the relay chain blocks and make a consensus.

- **Collator**: The Collator will help the Validators to collect, verify and submit alternative parallel chain blocks, and maintain a full chain of parallel nodes.

- **Fisherman**: The Fisherman mainly obtains income by reporting illegal transactions or blocks.

- **Nominator**: The Nominator owns stake and maintains and is responsible for the security of the Validator.

The security of the notary mechanism is fully guaranteed by the notary, and the parties involved in the cross-chain need to give greater trust to the intermediary. The idea of hash lock is widely used in the payment field, such as Lightning Network, Raiden Network and Interledger, a cross-chain asset transfer protocol, etc. However, hash lock is currently only suitable for the exchange of partial assets or key data, and it does not even support transfer, so its trial scenarios are limited. BTC Relay requires additional trust and maintenance costs, and the data storage inside the smart contract will have a problem of volume expansion. However, the sidechain mechanism can provide more cross-chain interaction scenarios than hash locking. The idea of sidechain and SPV-like verification is suitable for all cross-chain scenarios.
The security guarantee of cross-chain exchange between parallel chains of Polkadot mainly comes from the feature of shared security. Shared security makes cross-chain transactions and common transactions occur simultaneously, so there is no cross-chain data inconsistency such as double spend attack in other cross-chain scenarios.

Ethereum uses MPT (Merkle Patricia Trie) [10] to build a transaction tree and a receipt tree, and the relevant Merkel tree roots are saved in the Ethereum block header.

The cross-chain events thrown by Ethereum will be saved in the receipt. We can prove that the cross-chain events exist in the receipt tree by constructing the SPV path of the cross-chain events from the receipt tree. However, because the block header lacks the signature information of the relevant validator, it is difficult for other chains to directly prove that the receipt tree exists in Ethereum, and this problem can only be solved by the method of multi-node data comparison.

Although there is no receipt tree similar to Ethereum in Fabric, its transactions and receipts carry the signature information of the Endorser node [11]. Other chains can verify the signature information of the Endorser node to confirm the authenticity of cross-chain events, but this method is too inefficient.

This paper proposes a trusted cross-chain data verification and exchange method of heterogeneous blockchains for electricity application scenarios, and designs a chain-level cross-chain event verification mechanism.

2. Construction of Cross-Chain Event Merkel Tree

In the cross-chain transaction scenario of electricity applications, for the blockchain that needs cross-chain transactions, the system collects the cross-chain transactions in the generation stage of the block to construct the cross-chain event Merkel tree. The roots of the traditional Merkel tree and the cross-chain event tree are merged upward to generate a new Merkel tree root, which is stored in the block header, as shown in figure 1.

![Figure 1. Merkel tree for cross-chain transactions.](image_url)

The leaf nodes of the cross-chain event Merkel tree are the hash values of various cross-chain events in the block, and a Merkel tree is generated from these leaf nodes. The verifier will sign the block hash. The steps to build a trustworthiness attestation of cross-chain events based on the Merkel path are as follows:

(1) For a blockchain platform that requires cross-chain transactions, as shown in figure 1, the system builds a Merkel structured cross-chain event tree by collecting relevant cross-chain transactions in the generation stage of this block;
(2) Under the call of the application service platform, the cross-chain blockchain platform throws cross-chain events by triggering cross-chain smart contracts;

(3) After the cross-chain events are encapsulated and organized inside the platform, they are inserted into the cross-chain event Merkel tree. The blockchain platform will sign the root of the entire cross-chain event tree after each round of cross-chain consensus;

(4) The blockchain platform builds a credible Merkel proof of the cross-chain event based on the Merkel cross-chain event tree;

(5) The target blockchain platform verifies the credibility of the cross-chain event according to the cross-chain event Merkel tree on the source blockchain and the related credible proofs. If the verification fails, the illegal event rollback process is executed. If it passes, the subsequent cross-link routing process is executed.

The cross-chain event Merkel tree constructed by the system is parallel to the Merkel tree of the ledger state. The cross-chain event tree is publicly accessible to the outside world. The throwing or generation of cross-chain events must be achieved through the built-in cross-chain smart contract. Cross-chain events, like ordinary blockchain transactions, are confirmed by consensus algorithms throughout the network.

3. Cross-Chain Verification Process
The cross-chain transaction verification method of heterogeneous blockchain for power business scenarios includes the following steps:

Receive the first cross-chain transaction request information sent by the first application chain, and generate transaction information after analyzing the cross-chain transaction request information to obtain attribute information of the first application chain.

Determine the first verification rule corresponding to the first application chain blockchain according to the attribute information, where the first verification rule is recorded on the verification engine of the relay chain;

The first verification rule is executed according to the transaction information to verify whether the first cross-chain transaction complies with the first verification rule.

After the first verification rule is successfully executed, the first cross-chain transaction is passed to the execution engine to execute the first cross-chain transaction.

Before receiving the first cross-chain transaction request information sent by the first application chain, register the request information to record the attribute information of the first application chain in the verification engine of the relay chain.

The transaction information includes at least one of the following: source chain contract ID1, destination chain contract ID2, cross-chain transaction index, source chain block number, cross-chain transaction certificate, and custom fields.

Transaction information is encapsulated by IBTP (Inter Blockchain Transfer Protocol), including:

- From: Source chain contract ID1
- To: Destination chain contract ID2
- Index: Cross-chain transaction index
- Timestamp: Source chain block number
- Payload: Cross-chain call content encoding
- Proof: cross-chain transaction certificate
- Extra: Custom fields
- CID: Called contract ID3
- Func: Called function
- Args: Called function parameters
- Callback: Callback function

The Proof field stores the legitimacy and existence proof information of cross-chain transactions and provides specific verification information for the relay chain cross-chain verification engine. The
The content of the Proof field may vary according to the characteristics of the specific application chain. Specific verification rules and methods can be dynamically loaded into the cross-chain verification engine through flexible verification rules. There may be different structures according to different application chains, and the main structure is determined by the script processing of the verification rules. For example, cross-chain transaction certification is mainly composed of transaction content hash and transaction hash signature. The content hash is a hash of the transaction content of the cross-chain transaction. The transaction hash signature may generally be the signature of the endorser node in the application chain to hash the transaction content. Performing the first verification rule according to the transaction information to verify whether the first cross-chain transaction complies with the first verification rule includes: throwing the cross-chain transaction certificate into the virtual machine of the relay chain, where the virtual machine records a number of corresponding second application chains.

The verification engine also includes a receiving and parsing unit, a verification rule matching unit, a verification execution unit, a registration information receiving unit, and a transaction content verification unit.

- The receiving and parsing unit is configured to receive the first cross-chain transaction request information sent by the first application chain. It generates transaction information after parsing the cross-chain transaction request information to obtain attribute information of the first application chain.
- The verification rule matching unit is configured to determine the first verification rule corresponding to the first application chain according to the attribute information, wherein the first verification rule is recorded on the verification engine of the relay chain.
- The verification execution unit executes the first verification rule according to the transaction information to verify whether the first cross-chain transaction complies with the first verification rule.
- The registration information receiving unit is configured to receive registration request information sent by the first application chain to record the attribute information of the first application chain in the verification engine of the relay chain.
- The transaction content verification unit is configured to determine whether the transaction content of the first cross-chain transaction is correct according to the attribute information of the first application chain obtained after the receiving and parsing unit parses the first cross-chain transaction request information.

4. Experiments

We conduct cross-chain experiments in the main-side chain mode of the electricity business scenario. In this cross-chain scenario of the main-side chain, it is also necessary to design a cross-chain transaction consistency guarantee mechanism to ensure the integrity of cross-chain transactions. The cross-chain transaction management mechanism of the main-side chain system architecture borrows from the traditional two-phase commit mechanism. It divides cross-chain transaction operations into Prepare stage and Commit/Rollback stage. As the coordinator, the main chain assists the cross-chain piers of both parties to synchronize the state in a passive way to realize the management of cross-chain transactions.

The cross-chain transaction information is encapsulated by IBTP. The Payload field records the specific cross-chain content. This field does not limit the content format. According to the requirements of review and verification, the electricity monitoring data can be customized with data types and formats.

As shown in figure 2, the block with height 108 is used for cross-chain transactions. For the contract with the address “0x12345” in blockchain B, the contract method called by this transaction is “put” and the parameter is “bitxhub, 1000”. The Proof field is the path information of the Merkle tree, where the Hash field is the hash of cross-chain transaction related content, the Path field is the hash of the SPV path, the Merkle Root field is the hash of the final root, and the Signature field is the signature of the root’s hash.
Figure 2. A cross-chain transmission transaction structure of an application chain.

5. Conclusion

Based on the cross-chain transactions for electricity application scenarios, this paper designs a cross-chain event Merkel tree, as well as a verification engine and verification process for cross-chain events. The experiments include the cross-chain transaction management mechanism of the main-side chain architecture, which shows that the verifying method proposed in this paper can achieve the purpose of trusted circulation and secure transactions of electricity business data between different application chains. In the future, we can continue to improve the efficiency of cross-chain, and join the privacy data protection mechanism of cross-chain transactions.

References

[1] Adrian H B and Stefan T 2016 Interledger: Creating a standard for payments Proc. of the 25th International World Wide Web Conferences Steering Committee (Switzerland: International World Wide Web Conferences Steering Committee) pp 281-2.
[2] Koenst T and Poll E 2019 Assessing interoperability solutions for distributed ledgers Pervasive and Mobile Computing 59 101079.
[3] BTC Relay 2017 BTC Relay https://github.com/ethereum/btcrelay.
[4] Chow J 2020 A Bridge between the Bitcoin Blockchain & Ethereum Smart Contracts http://btcrelay.org/.
[5] Adam B, Matt C, Luke D, Mark F, Gregory M, Andrew M, Andrew P, Jorge T and Pieter W 2014 Enabling Blockchain Innovations with Pegged Sidechains https://www.blockstream.com/sidechains.pdf.
[6] Poon J and Dryja T 2016 H ashed Timelock Contract (HTLC) https://lightning.network/lightning-network-paper.pdf#page=30.
[7] BTC Relay 2016 Chain Fibers Redux https://github.com/ethereum/wiki/wiki/Chain-Fibers-Redux#blockchain-scalability-chain-fibers-redux.
[8] Polkadot 2020 Polkadot: Decentralized Web 3.0 Blockchain Interoperability Platform https://polkadot.network/.
[9] Commos 2020 Cosmos Network - Internet of Blockchains https://cosmos.network/.
[10] Thereum 2018 Merkle Patricia Trie Specificationn https://github.com/ethereum/wiki/wiki/Patricia-Trie.
[11] Hyperledger Fabirc 2020 Transaction Flow https://hyperledger-fabric.readthedocs.io/en/release-2.0/txflow.html.