

IEEE Standard for Blockchain Interoperability Data Authentication and Communication Protocol

IEEE Computer Society

Developed by the
Blockchain and Distributed Ledgers Committee

IEEE Std 3205™-2023

IEEE Standard for Blockchain Interoperability Data Authentication and Communication Protocol

Developed by the

Blockchain and Distributed Ledgers Committee
of the
IEEE Computer Society

Approved 30 March 2023

IEEE SA Standards Board

Abstract: Abstract: Blockchain interoperability is the ability of two or more blockchain systems or applications to exchange information and to mutually use the information that has been exchanged. The interfaces and protocols play a very important role in realizing interoperability. Therefore, the standard of cross-chain interoperability interfaces and protocols, especially those for data authentication and communication among homogeneous and heterogeneous blockchains systems, is needed. Such protocols coordinate blockchains while supporting multiple cross-chain models and levels to meet business demands without the need to customize gateways or exchanges for specific use cases. Provided in this standard are an infrastructure of cross-chain interoperability, as well as interfaces and protocols of data authentication and communication for homogeneous and heterogeneous blockchain interoperability. The protocols include the distributed identity protocol, metadata protocol, on-chain proof transformation protocol, and cross-chain communication protocol.

Keywords: Keywords: heterogeneous blockchain, homogeneous blockchain, IEEE 3205, interoperability

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 28 April 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9620-9 STD26095
Print: ISBN 978-1-5044-9621-6 STDPD26095

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

¹Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

²Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are

³Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴Available at: <https://standards.ieee.org/standard/index.html>.

⁵Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this standard was completed, the Blockchain Interoperability—Data Authentication and Communication Protocol Working Group had the following entity membership:

Ying Yan, Chair
Xiaohe Liu, Vice Chair
Lu Zhang, Secretary

| <i>Organization Represented</i> | <i>Name of Representative</i> |
|---|-------------------------------|
| Alipay (China) Technology Co., Ltd. | Honglin Qiu |
| Baidu Online Network Technology (Beijing) Co., Ltd. | Jin Bo |
| Beijing Academy of Blockchain and Edge Computing..... | Jin Dong |
| Chinese Academy of Sciences | Yi Sun |
| China Electronics Standardization Institute | Ming Li |
| China Zheshang Bank Co., Ltd..... | Cheng Zang |
| Hangzhou Qulian Technology Co., Ltd. | Lu Zhang |
| Hangzhou Yunphant Network Technology Co., Ltd..... | Butian Huang |
| Neusoft Corporation | Sihan Liu |
| Shanghai Bianjie AI Technology Co., Ltd. | Yin Tao |
| Shandong Computer Science Center..... | Zhen Zhang |
| Shanghai Development Center of Computer Software Technology | Bingrong Dai |
| Shanghai Plian Info Tech Co., Ltd. | Feng Cao |
| Shanghai Pudong Development Bank..... | Yang Gao |
| Sichuan Changhong Electric Co., Ltd..... | Bo Tang |
| South China University of Technology..... | Haobo Lai |
| State Grid Corporation of China | Dong Wang |
| Wuxi SensingNet Industrialization Research Institute..... | Mingjuan Wu |
| Zhejiang Lab..... | Haitao Wang |
| Zhejiang University | Bingsheng Zhang |

The Working Group gratefully acknowledges the contributions of the following participants. Without their assistance and dedication, this standard would not have been completed.

Liang Cai
Wenting Chang
Haoyong Chen
Jiajun Chen
Jianhai Chen
Shenglong Chen
Xi Chen
Dongsheng Guo
Dejun Huang
Chao Li

Fan Li
Nuqie Li
Wei Li
Guoxin Liu
Jian Liu
Zhenguang Liu
Jin Peng
Weiwei Qiu
Kui Ren
Yanduo Ren

Lin Sun
Zhiguo Wan
Lei Wu
Shicheng Xu
Shu Yin
Xu Yin
Liwei Yuan
Xiaomeng Zhang
Lihua Zhao
Xueyan Zou

The following members of the entity Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

| | | |
|--|---|---|
| OxSenses Corporation | Inspur Electronic Information Industry Co., Ltd. | Shanghai Development Center of Computer Software Technology |
| 1 st Cycle Corporation | Institute of Biomedical Engineering, Chinese Academy of Medical Sciences & Peking Union Medical College | Shenzhen University |
| Alipay (China) Technology Co., Ltd. | NXP Semiconductors | South China University of Technology |
| Beijing Academy of Blockchain and Edge Computing | Shandong Computer Science Center | State Grid Corporation of China (SGCC) |
| CRRC Zhuzhou Institute Co., Ltd. | Shanghai Bianjie AI Co., Ltd. | Wuxi SensingNet Industrialization Research Institute |
| Fuzhou Institute for Data Technology | | Yokosuka Telecom Research Park, Inc. |
| Haier Group Corporation | | Zhejiang Lab |
| Hangzhou Qulian Technology Co., Ltd. | | Zhejiang University |
| Hangzhou Yunphant Network Technology | | |

When the IEEE SA Standards Board approved this standard on 30 March 2023, it had the following membership.

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

| | | |
|-----------------------|----------------------|-------------------|
| Sara R. Biyabani | Joseph S. Levy | Paul Nikolich |
| Doug Edwards | Howard Li | Annette D. Reilly |
| Ramy Ahmed Fathy | Gui Lin | Robby Robson |
| Guido R. Hiertz | Johnny Daozhuang Lin | Lei Wang |
| Yousef Kimiagar | Kevin W. Lu | F. Keith Waters |
| Joseph L. Koepfinger* | Daleep C. Mohla | Karl Weber |
| Thomas Koshy | Andrew Myles | Philip B. Winston |
| John D. Kulick | | Don Wright |

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 3205-2023, IEEE Standard for Blockchain Interoperability Data Authentication and Communication Protocol.

Blockchain interoperability is the ability of two or more blockchain systems or applications to exchange information and to mutually use the information that has been exchanged.

The interfaces and protocols play a very important role in realizing interoperability. Therefore, a global standard of cross-chain interoperability interfaces and protocols, especially those for data authentication and communication for homogeneous and heterogeneous blockchains systems, is needed. Such protocols can coordinate blockchains while supporting multiple cross-chain models and levels to meet business demands without the need to customize gateways or exchanges for specific use cases.

Contents

| | |
|---|----|
| 1. Overview | 10 |
| 1.1 Scope | 10 |
| 1.2 Word usage | 10 |
| 2. Normative references | 10 |
| 3. Definitions, acronyms, and abbreviations | 11 |
| 3.1 Definitions | 11 |
| 3.2 Acronyms and abbreviations | 11 |
| 4. Challenge of blockchain interoperability | 12 |
| 5. General description of blockchain interoperability | 12 |
| 5.1 Overview | 12 |
| 5.2 System framework | 13 |
| 5.3 Functional reference architecture | 14 |
| 6. Identity protocol | 15 |
| 6.1 Introduction | 15 |
| 6.2 Blockchain domain name system (BCDNS) | 15 |
| 6.3 Blockchain identity | 16 |
| 6.4 Identity setup procedures | 17 |
| 7. Proof transformation protocol | 21 |
| 7.1 Introduction | 21 |
| 7.2 Third-party trust anchor and verifiable claims | 21 |
| 7.3 Proof transformation procedure | 22 |
| 7.4 Implementation of proof transformation component | 23 |
| 8. Data protocol | 24 |
| 8.1 Introduction | 24 |
| 8.2 Global dictionary table | 24 |
| 8.3 Unified cross-chain packet structure | 24 |
| 8.4 The procedure of receiving UCP | 25 |
| 9. Communication protocol | 25 |
| 9.1 Introduction | 25 |
| 9.2 Authentic message protocol | 26 |
| 9.3 Smart contract datagram protocol | 28 |
| 10. Addressing protocol | 28 |
| 10.1 Introduction | 28 |
| 10.2 Off-chain relay component information verifiable claim | 28 |
| 10.3 The procedure of addressing protocol | 29 |
| 11. Cross-chain data authentication and communication procedure | 30 |
| 12. Technical and security requirements | 31 |
| 12.1 Technical requirements | 31 |
| 12.2 Security requirements | 32 |
| Annex A (informative) Examples of blockchain interoperability | 33 |

IEEE Standard for Blockchain Interoperability Data Authentication and Communication Protocol

1. Overview

1.1 Scope

This standard provides an infrastructure of cross-chain interoperability, as well as interfaces and protocols of data authentication and communication for homogeneous and heterogeneous blockchain interoperability. The protocols include the distributed identity protocol, metadata protocol, on-chain proof transformation protocol, and cross-chain communication protocol.

1.2 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{6,7}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

There are no normative references in this standard.

⁶The use of the word *must* is deprecated and cannot be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.

⁷The use of *will* is deprecated and cannot be used when stating mandatory requirements, *will* is only used in statements of fact.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The IEEE Standards Dictionary Online should be consulted for terms not defined in this clause.⁸

blockchain domain naming system: A system that provides a globally unique identifier for a blockchain which can facilitate interoperability between different blockchain networks, allowing for seamless cross-chain communication.

blockchain trust anchor: A trust anchor of the blockchain verifies the data provided by another blockchain.

NOTE—Blockchain trust anchor may be different for different types of blockchain.

formatted ledger data: Ledger data that is encoded with a standardized data structure and format that can be used by cross-chain applications.

ledger proof: Proof that verifies the existence, authenticity, and integrity of the ledger data on the blockchain.

off-chain relay component: A functional program that facilitates communication between different blockchains by receiving, forwarding, and delivering cross-chain messages.

on-chain program: Refer to the code that is executed on a blockchain network with the agreement of the consensus algorithm.

NOTE 1—Business smart contracts and system contracts running on the blockchain are types of on-chain programs.

NOTE 2—On-chain programs can participate in cross-chain interoperability process and can be invoked not only by off-chain applications but also by on-chain applications.

original ledger data: Refer to ledger data that is encoded using the data structure and format of the original blockchain.

proof transformation component: A functional program that provides authenticity verification services for cross-chain data and generates the third-party proof.

3.2 Acronyms and abbreviations

| | |
|-------|-------------------------------|
| AMP | authentic message protocol |
| BCDNS | blockchain domain name system |
| BTA | blockchain trust anchor |
| DID | decentralized identifier |
| DNS | domain name system |
| IP | internet protocol |
| PKI | public key infrastructure |
| SDP | smart contract data protocol |

⁸IEEE Standards Dictionary Online is available at: <http://dictionary.ieee.org>. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

| | |
|--------|-------------------------------------|
| TP-BTA | third party blockchain trust anchor |
| UCP | unified cross-chain packet |
| UDP | user diagram protocol |
| VC | verifiable claim |

4. Challenge of blockchain interoperability

At present, there has been an influx of blockchain platforms, but the majority of the mainstream blockchain platforms remain siloed. In the realistic scenario with increasingly complex business requirements, there is a lack of unified interconnection mechanisms among blockchains, which greatly limits the development of blockchain technology and application ecosystem. In order to address this challenge, there is a growing demand for cross-chain interoperability to realize the circulation of on-chain assets.

The exchange of transaction information between different blockchains is completed through cross-chain technology. However, the underlying architecture and data structure of different blockchains are different. This results in that the asset exchange and data exchange between them mainly rely on centralized mechanisms. The centralized exchanges are often considered unsafe and opaque, so interoperability barriers between various blockchain applications are very high making it unable to effectively share information on the blockchain.

Security and mutual trust between different blockchains can be achieved through cross-chain technology. When designing and utilizing blockchain, it is important to establish various security mechanisms, including smart contract security, password security, ledger data security, network security, consensus mechanism security while different chains have different security boundaries when facing different businesses. To fully achieve the security and mutual trust between multiple chains, cross-chain technology shall be utilized to establish mutual trust conditions between different chains.

At present, the application of blockchain has been successful in many fields such as finance, government affairs, and energy. The contract logic of different business scenarios varies significantly, resulting in more complex business logic when dealing with different business scenarios compared to the traditional cross-chain of the same business scenario. Therefore, ensuring the integrity and consistency between events is the key for cross-chain technology to establish reliable business ecosystems.

Under the requirements of both business and technology, more and more attention has been paid to the interoperability among multiple blockchains to enable asset and service exchange and boost the blockchain industry.

5. General description of blockchain interoperability

5.1 Overview

The cross-chain technology is essentially a technology that passes the information from Chain A to Chain B securely and credibly and produces the desired result on Chain B. [Figure 1](#) illustrates an abstract interoperability model between blockchains where smart contracts on the blockchain (or other on-chain programs, such as system modules) generate cross-chain information that undergoes consensus confirmation. The cross-chain information and its proof are transmitted to the destination chain through an off-chain relay component. The destination chain verifies the cross-chain information and its proof through consensus, and the verified cross-chain information is used by smart contracts on the chain (or other on-chain programs, such as system modules).

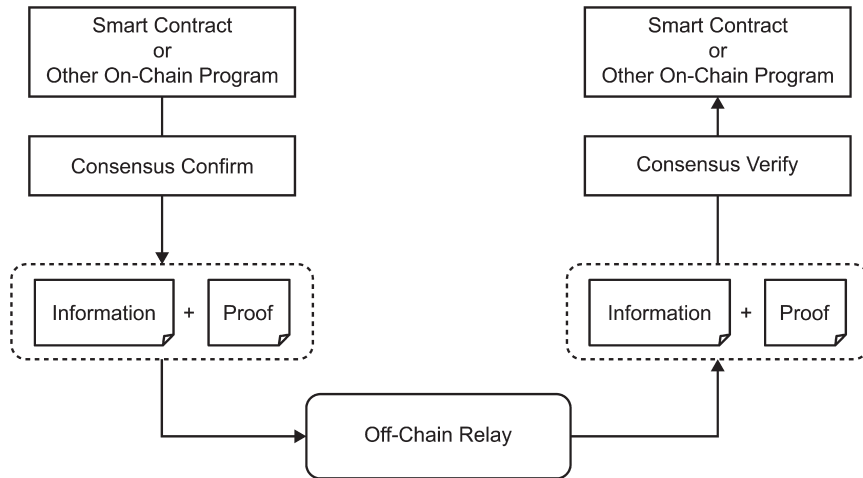


Figure 1—Illustration of cross-chain message interoperability

5.2 System framework

The interoperability framework is implemented according to [Figure 2](#). Under this framework, arbitrary interoperability among blockchains is supported.

The interoperability framework consists of several components, including blockchains, blockchain domain name system, off-chain relay components, and proof transformation components:

- Blockchains: Either homogeneous or heterogeneous;
- Off-chain relay components: Responsible for connecting the data transmission among multiple blockchains. The off-chain relay components are connected with each other and form a network so that blockchains connected with off-chain relay components realize intercommunication;
- Proof transformation component: Provides a trusted verification service for cross-chain data during the cross-chain process so that the received cross-chain data's validity is verified by the signature of the data;
- Blockchain domain name service (BCDNS): Assigns a globally unified identity and domain name upon a request from a valid blockchain. The domain name will be used for identity authentication during cross-chain communications.

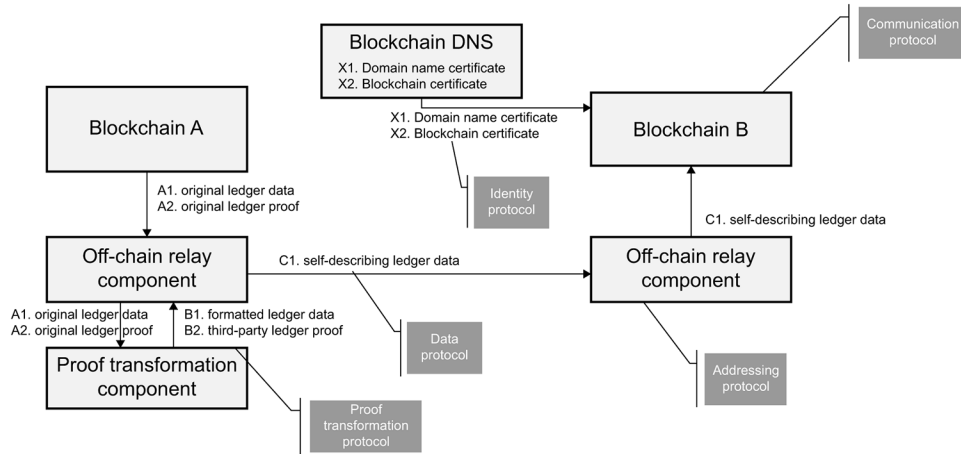


Figure 2—Cross-chain system framework

5.3 Functional reference architecture

The reference architecture of cross blockchain interoperability includes five layers:

- Application layer
- Cross-chain transaction layer
- Transport layer
- Secure authentication layer
- Data link layer

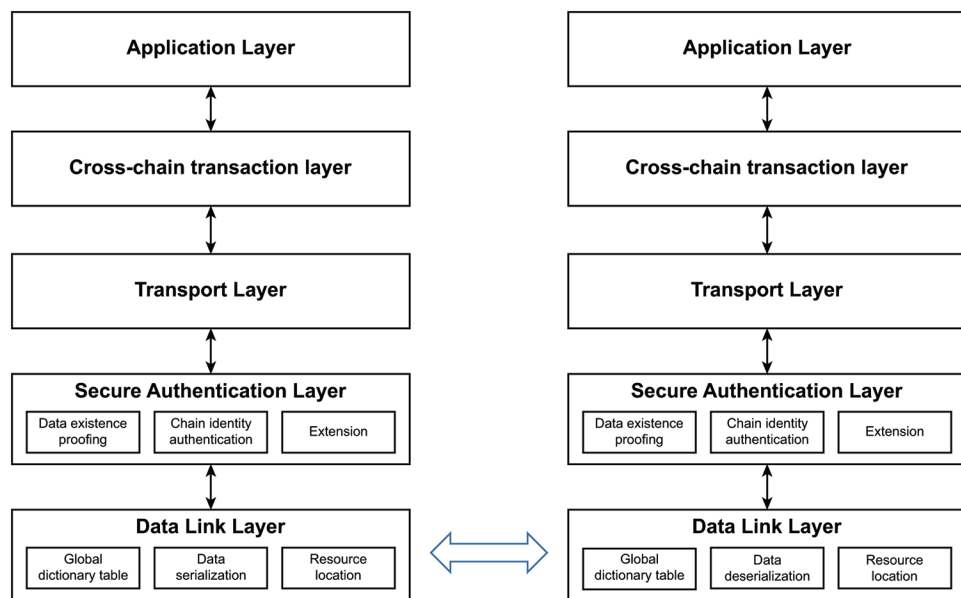


Figure 3—Functional architecture of cross-chain interoperability

5.3.1 Application layer

The application layer protocol enables information exchange between two blockchain systems allowing for cross-chain services such as cross-chain query, remote cross-chain call, cross-chain asset transfer, and so on.

5.3.2 Cross-chain transaction layer

The cross-chain transaction layer protocol is designed to achieve atomicity, consistency, isolation, and durability (ACID) properties of distributed transactions. This involves the exchange of necessary messages and related processes.

5.3.3 Transport layer

The transport layer facilitates communication service between two blockchain systems, enabling them to send and receive both ordered and non-ordered cross-chain messages.

5.3.4 Secure authentication layer

The secure authentication layer parses messages from the data link layer and performs several authentication checks, such as data existence proof, chain identity authentication, and other reserved extension(s):

- Data existence proof validates the existence of data in the cross-chain messages, as well as authenticity and integrity.
- Chain identity authentication validates the identity of the peer block chain system.
- Extension(s) Are reserved for future enhanced secure function(s), e.g., access restriction.

5.3.5 Data link layer

The data link layer provides the cross-chain message format, data serialization, and resource location protocols:

- Global dictionary table: Used to register different blockchain systems and their specific properties, such as type, data format, algorithm, and other information for mutual recognition.
- Data serialization and deserialization protocol: The transmitter utilizes serialization protocol to encapsulate the useful information with header bit fields and generates the entire data transmission packet. The receiver utilizes deserialization protocol to parse the received message and extract useful information.
- Resource location protocol: Provides data location and addressing.

6. Identity protocol

6.1 Introduction

This protocol introduces a blockchain domain name mechanism to assign a unique identity to every blockchain.

6.2 Blockchain domain name system (BCDNS)

The BCDNS provides a globally unique name to identify a blockchain. It consists of BCDNS certificate authority, BCDNS trust anchor, and BCDNS verifiable claim.

6.2.1 BCDNS certificate authority (CA)

The BCDNS certificate authority (CA) is an entity that verifies the domain name applying requests and assigns a blockchain domain name to the blockchain by issuing digital certificates. It can be realized by either a PKI-based system or a decentralized identifiers (DID) system. Since the BCDNS trust anchor is publicly accessible, the authenticity of the blockchain domain name certificate is confirmed.

6.2.2 BCDNS trust anchor and verifiable claims

Blockchain DNS trust anchor is the root certificate of blockchain DNS, which is used to either directly or indirectly issue a blockchain domain name certificate.

Blockchain DNS verifiable claim (BCDNS VC) allows the domain name owner to publish the verifiable blockchain information binding to the domain name. The identity protocol defines various types of VC that serve different cross-chain verification purposes, as listed in [Table 1](#).

Table 1—Types of BCDNS verifiable claims

| Field | Description |
|-----------|---|
| BTA VC | Used to claim the trust anchor of the blockchain. |
| TP-BTA VC | Used to claim the third-party trust anchor of the blockchain. It is defined in proof transformation protocol. |
| AMP VC | Used to claim the trust anchor of the AMP. It is defined in communication protocol. |

6.3 Blockchain identity

6.3.1 Blockchain domain name certificate

Blockchain domain name certificate is a globally unique identity of a blockchain and is designed to claim the ownership. The essential bit fields are listed in [Table 2](#).

Table 2—The structure of blockchain domain name certificate

| Field | Type | Description |
|--------------------|------------------|---|
| Version | Unsigned integer | The version of certificate. |
| Domain name | String | The domain name to be claimed in this certificate. |
| Subject public key | String | The public key of certificate owner, that is k_{pub} . There is a pair key k_{priv} , that is held and protected by domain name owner. |
| Issuer info | String | The issuer information. If the certificate is issued by a CA-based BCDNS, it will include the public key of CA, signature algorithm, and the signature. |

6.3.2 Blockchain trust anchor and verifiable claims

Blockchain trust anchor (BTA): The blockchain needs to publish its trust anchor so that the receiver verifies its identity and data proof. Different blockchains have different trust anchor structures.

BTA VC (blockchain trust anchor VC) is defined to bind the blockchain domain name to the blockchain trust anchor, which uniquely identifies the blockchain such as genesis block hash that verifies a client can load it as a blockchain trust anchor. The essential bit fields of BTA VC are listed in [Table 3](#).

Table 3—The structure of BTA VC

| Field | Type | Description |
|----------------------|------------------|---|
| Version | Unsigned integer | The version of VC. |
| Domain name | String | The blockchain’s domain name. |
| Subject product type | String | The type of blockchain. |
| Subject product SVN | String | The security version number of blockchain certificate. A blockchain certificate with a lower SVN is outdated. Blockchain domain name owner could publish a blockchain certificate with higher SVN based on security considerations. |
| Subject trust anchor | String | The unique trust anchor of blockchain. Different subject product type has different trust anchor structure. For example, some blockchains use the public key set of consensus nodes. |
| Signature algorithm | String | The signature algorithm. |
| Signature | String | The signature, which is signed by private key of blockchain domain name certificate. |

6.4 Identity setup procedures

The blockchain configurator submits a domain name request to BCDNS CA, and the BCDNS CA issues the domain name by signing a blockchain domain certificate. The blockchain configurator then binds the blockchain trust anchor to the domain name by signing the verifiable claims.

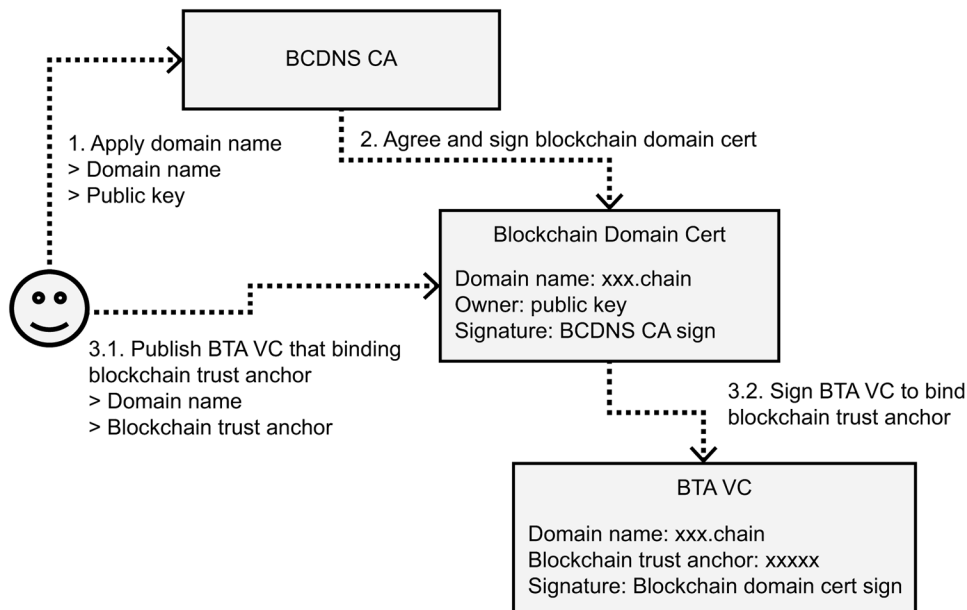


Figure 4—Illustration of blockchain domain name application

6.4.1 Blockchain DNS trust anchor setup procedure

For each blockchain, the trust anchor of the blockchain DNS must be set up. Usually, it is implemented by calling smart contract, and the procedure is illustrated in [Figure 5](#):

- 1) Blockchain DNS publishes its trust anchor in a public place so the blockchain configurator gets the accurate and correct BCDNS trust anchor;
- 2) Blockchain configurator constructs the setup transaction and sends it to the on-chain program. Blockchain executes a transaction to install the blockchain DNS trust anchor;
- 3) The blockchain verifies the integrity of the trust anchor and saves the trust anchor of BCDNS if verification has been passed.

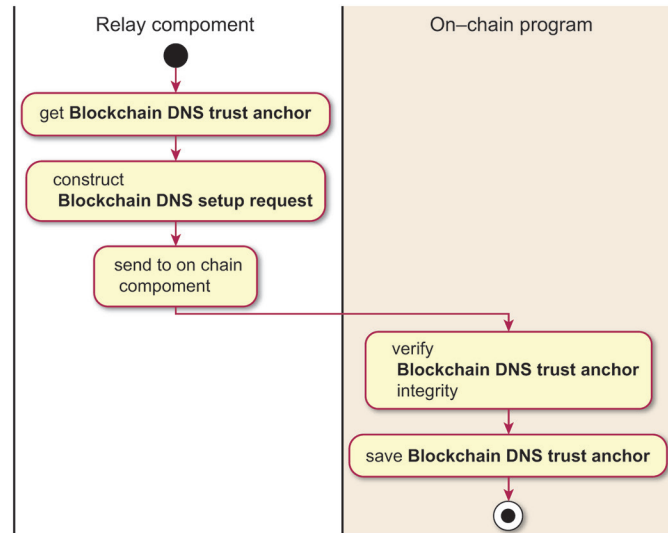


Figure 5—Blockchain DNS trust anchor setup procedure

6.4.2 Blockchain domain name certificate application procedure

Application shall be submitted for a blockchain domain name certificate before any message is transmitted in a blockchain system. The application procedure for a blockchain domain name certificate is illustrated in Figure 6.

- 1) Blockchain configurator sends a request to the blockchain DNS that includes the domain name and the public key k_{pub} ;
- 2) Blockchain DNS issues the unique domain name to the applicant by signing the domain name certificate;
- 3) Blockchain configurator issues the BTA VC including blockchain trust anchor, and signs it with the private key k_{priv} , which is associated with k_{pub} in step 1);
- 4) Blockchain DNS records the BTA VC and enables it to be publicly downloaded and certified.

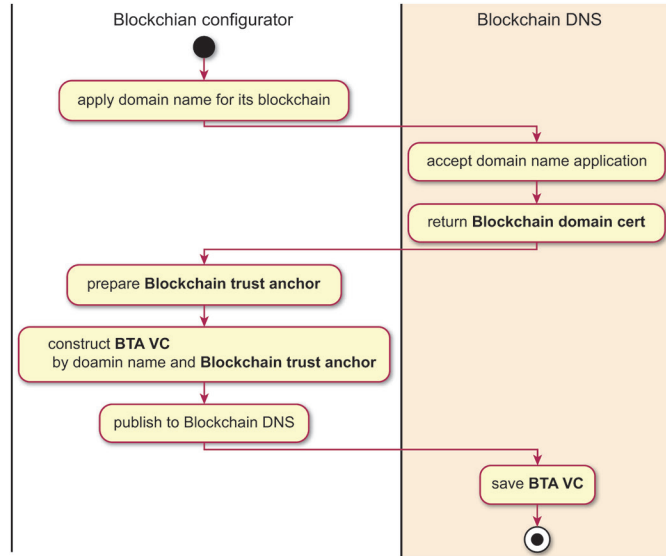


Figure 6—Procedure for application of blockchain domain name certificate

6.4.3 Blockchain domain name certificate installation procedure

Before the target blockchain verifies the received cross-chain packet, it needs to install the blockchain domain name certification of the source blockchain, as illustrated in Figure 7.

- 1) Before the data link layer submits the received cross-chain data packet to the secure authentication layer, it shall check whether the secure authentication layer has installed the blockchain domain name certificate of the source blockchain;
- 2) If not, the data link layer shall query the blockchain domain name certificate from the blockchain DNS and construct the blockchain transaction and send it to the secure authentication layer;
- 3) The secure authentication layer shall load the blockchain DNS trust anchor and use it to verify the correctness and authenticity of the blockchain domain name certificate. It shall save the valid domain name certificate.

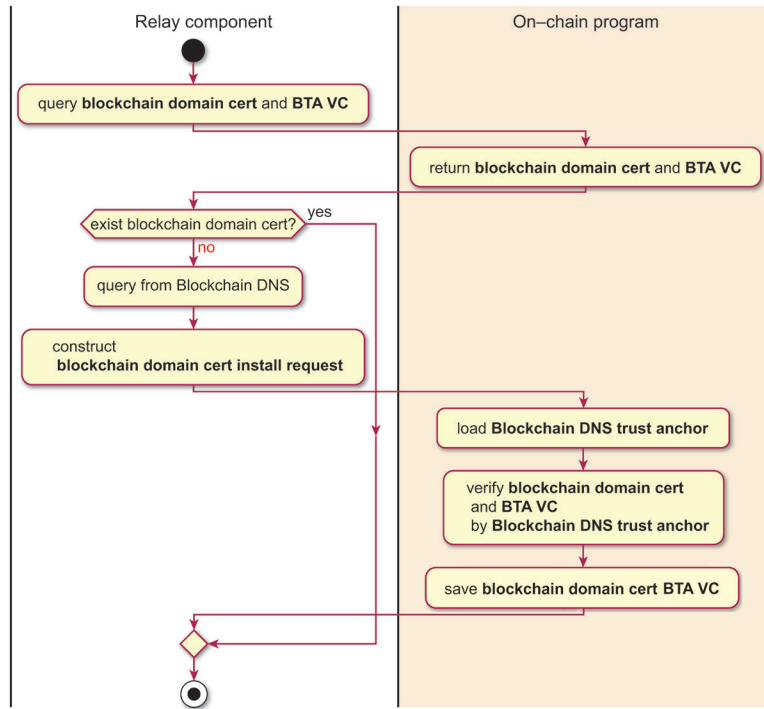


Figure 7—Illustration of blockchain domain name certification installation

6.4.4 Blockchain domain name certificate verification procedure

When the target blockchain receives the cross-chain packet, it will verify the received packet, as illustrated in Figure 8.

- 1) The target relay component receives the cross-chain packets and constructs a cross-chain packet request accordingly;
- 2) The on-chain program verifies the integrity of the cross-chain data packet and reads the blockchain domain name of the source of the cross-chain data;
- 3) The BCDNS VC of the domain name and the trust anchor in the verifiable claims is loaded and then uses them to verify the ledger proof;
- 4) If verification is passed, it means that the cross-chain packet came from the claimed blockchain.

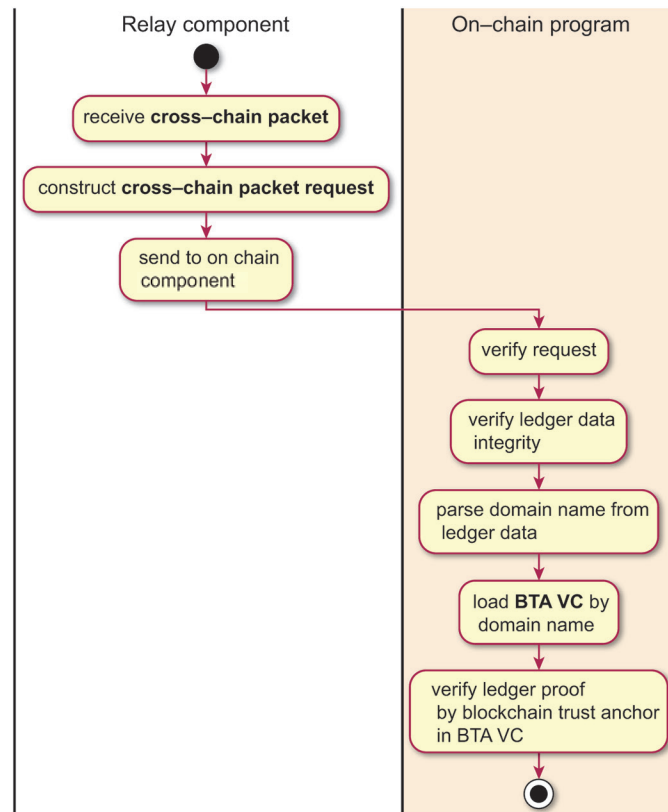


Figure 8—Illustration of cross-chain request verification

7. Proof transformation protocol

7.1 Introduction

The target blockchain is expected to verify the existence of data from other blockchains, so it needs to set up multiple verification clients. For example, if one source blockchain connects to multiple target blockchains, then all target blockchains need to install the verification client. This causes higher complexity and low performance. Additionally, certain smart contracts may not be able to implement the verification client due to their design limitations.

The proof conversion protocol is introduced. It offers a trusted third party to verify the ledger proof of cross-chain data at the source blockchain and generates the third-party ledger proof. The trusted third party may be a trusted execution environment, a notary group, or a relay blockchain.

7.2 Third-party trust anchor and verifiable claims

Third-party trust anchor refers to a set of public keys, which is used to verify the authenticity and integrity of the third-party's ledger proof.

TP-BTA VC is signed by the source blockchain. The source blockchain gets the TP-BTA from the proof transformation component. With TP-TBA VC, the domain owner publishes the third-party trust anchor.

The proof transformation component is responsible for using the blockchain verification client which loads BTA defined in BTA VC to verify blockchain data and generate a new signature as a third-party proof.

For a verifier, it shall trust the TP-BTA VC and use the public key published in TP-BTA VC to verify the blockchain data.

Table 4—The structure of TP-BTA VC

| Field | Type | Description |
|---|------------------|---|
| Version | Unsigned integer | The version of VC. |
| Domain name | String | The blockchain domain name. |
| Blockchain domain name certificate hash | String | The hash of the blockchain domain certificate. |
| Verification keys | String | A set of public keys, which is utilized to verify the authenticity and integrity of third-party’s ledger proof. |
| Signature algorithm | String | The signature algorithm of this VC. |
| Signature | String | The signature that signs by private key k_{priv} of the blockchain domain name certificate. |

7.3 Proof transformation procedure

The proof transformation procedure shall include two sub-processes. In the first, the blockchain configurator configures the proof transformation component. In the second, the verifier requests the proof transformation component to obtain a third-party ledger certificate.

- 1) First, the blockchain configurator selects a proof transformation component;
- 2) The blockchain configurator sends the BTA VC to the proof transformation component;
- 3) The proof transformation component uses BTA VC to initialize the blockchain verification client, signs a TP-BTA VC, and finally returns the VC to the blockchain configurator;
- 4) After the source relay component retrieves the original blockchain data and proof, the component forwards them to the proof transformation component for proof transformation;
- 5) The proof transformation component uses the blockchain verification client to verify the original ledger data and ledger proof. If the verification is passed, it transforms the original ledger format into a new format, such as protobuf-encoded transactions formatted into Json-encoded formats. Finally, the formatted ledger data and the third-party ledger proof are generated;
- 6) After the relay component obtains the third-party blockchain data and proof, the component sends it to the destination blockchain verifier. The verifier then uses TP-BTA to verify the formatted ledger data and third-party ledger proof for authenticity and integrity.

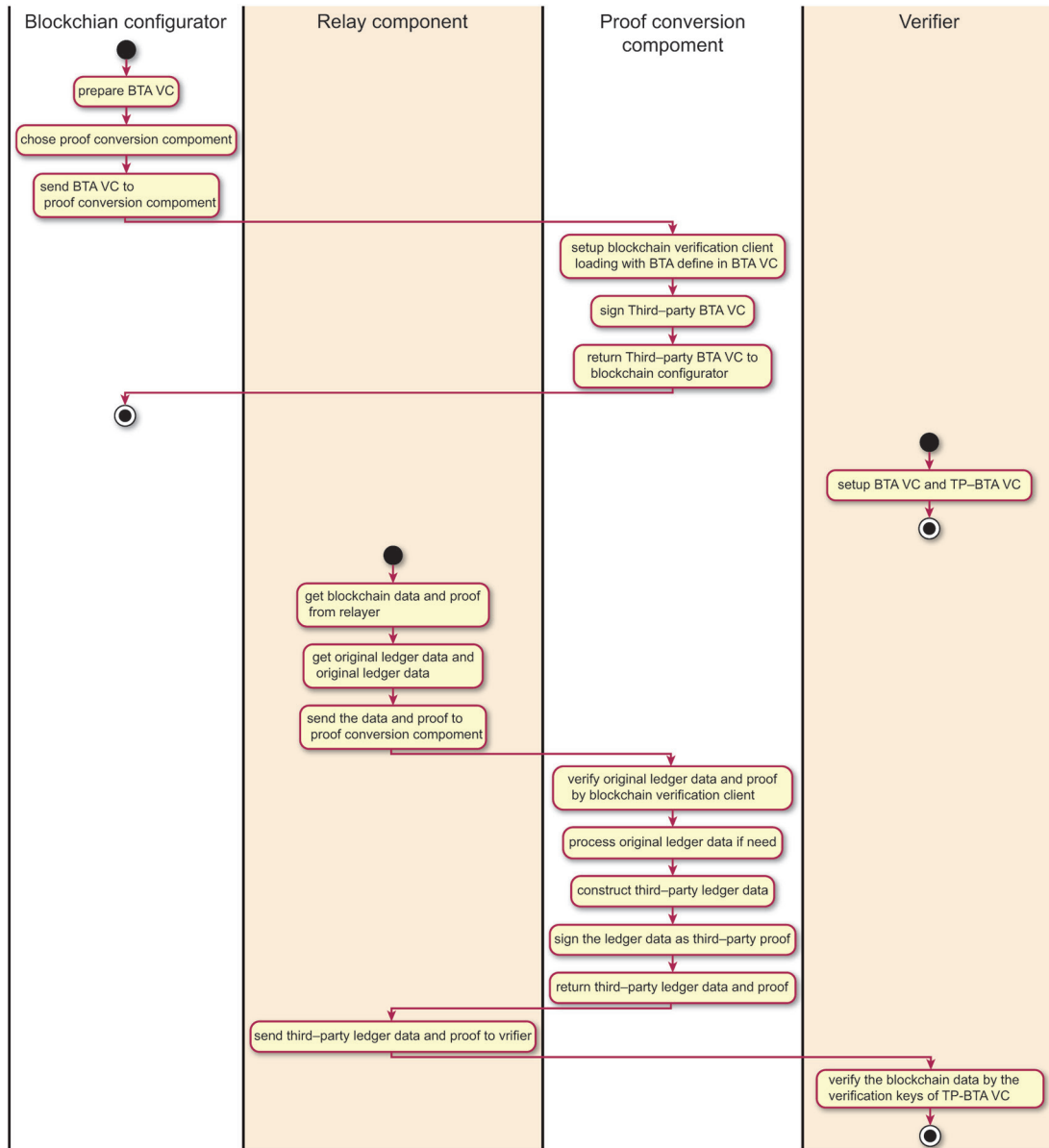


Figure 9—Illustration of proof transformation

7.4 Implementation of proof transformation component

The proof transformation component shall publish an API of its service. At least, it shall provide the Setup APIs for essential configuration and the transformation API to perform requests.

- 1) The *input Setup API* and *output Setup API*
 - a) The input API enables ingress of the blockchain domain name certificate and other necessary information, such as the IP of the blockchain node to which it connects.
 - b) The output API enables egress of the TP-BTA certificate.
- 2) The *input Transformation API* and *output Transformation API*

- a) The input API enables ingress of the original ledger data and ledger proof.
- b) The output API enables egress of formatted ledger data and third-party ledger proof.

8. Data protocol

8.1 Introduction

The data protocol is used to solve the issue of parsing cross-chain data packets in the cross-chain interoperability by incorporating a global dictionary table and unified cross-chain packet.

8.2 Global dictionary table

The global dictionary table is used to define the basic metadata dictionary required for cross-chain data packet parsing, including the definition of various blockchain types and the blockchain data structures such as blocks, transactions, receipts, and others. It also includes encoding algorithms such as SHA256, SHA3, and more.

The global dictionary table consists of name, code, alias, and description. [Table 5](#) exemplifies the definition of a dictionary table.

Table 5—The structure of global dictionary table

| Name | Code | Alias | Description |
|----------------------|------|-----------------|---------------------------------|
| Mychain block | 0x01 | mc-block | Code of mychain’s block. |
| Mychain block header | 0x02 | mc-block-header | Code of mychain’s block header. |
| Mychain transaction | 0x03 | mc-tx | Code of mychain’s transaction. |

8.3 Unified cross-chain packet structure

Unified cross-chain packet (UCP) is a type of self-describing cross-chain data packet, which is transmitted among off-chain relay components. The UCP includes the necessary information, which is used by the receiver to verify and parse the payload, as illustrated in [Table 6](#).

Table 6—The structure of UCP

| Name | Type | Description |
|---------------|--------|--|
| Domain name | String | The domain name of the source blockchain. |
| Protocol type | String | The protocol type of this packet. For example, AMP indicates that this is an authentic message protocol packet. |
| Data type | String | The type of data field, which is defined in global dictionary table. The receiver reads the data type to identify the structure of the data field and loads the corresponding tools to access it. |
| Data | String | The original ledger data. |
| Proof | String | The original ledger proofs. |
| TP-data type | String | The type of TP-data field, which is defined in global dictionary table. The receiver reads the TP-data type to identify the structure of the data field and load the corresponding tools to access it. |
| TP-data | String | The third-party ledger data. |
| TP-proof | String | The third-party ledger proofs. |

8.4 The procedure of receiving UCP

The UCP will be transmitted between the off-chain relay component and smart contract or other on-chain program. They need to parse the UCP then execute the corresponding protocol procedure.

- a) Verify the UCP
 - 1) Extract the domain name from UCP
 - 2) Load the corresponding blockchain domain name certificate or TP-BTA certificate
 - 3) Load the corresponding blockchain verification client
 - 4) Verify the ledger proof or third-party ledger proof using blockchain verification client
 - 5) If it passes, it is a valid packet from the source blockchain
- b) Decode data
 - 1) Extract the data type from UCP
 - 2) Decode the data based on the data type
 - 3) Get the final format data with the data schema
- c) Execute protocol procedure
 - 1) Read the protocol of this packet
 - 2) Load the protocol executor and execute it with decode data

For example, when an off-chain relay component receives a UCP, it first verifies the UCP. The UCP will be discarded if verification fails. The relay component then extracts the protocol type. If it is AM protocol, the relay component extracts the target domain name. If the blockchain identified by the target domain name is connected with this relay component, the cross-chain data will be sent to the target blockchain.

9. Communication protocol

9.1 Introduction

The communication protocol is a protocol stack. The basic protocol is authentic message protocol (AMP), which enables an on-chain program, e.g., a smart contract to initiate a cross-chain message. Within the AMP, there exists a bit field of *Type*, which is utilized to derive subprotocols. The smart contract datagram protocol (SDP) is a subprotocol within AMP. It is designed to enable a smart contract on one blockchain to send a datagram to another smart contract on a different blockchain.

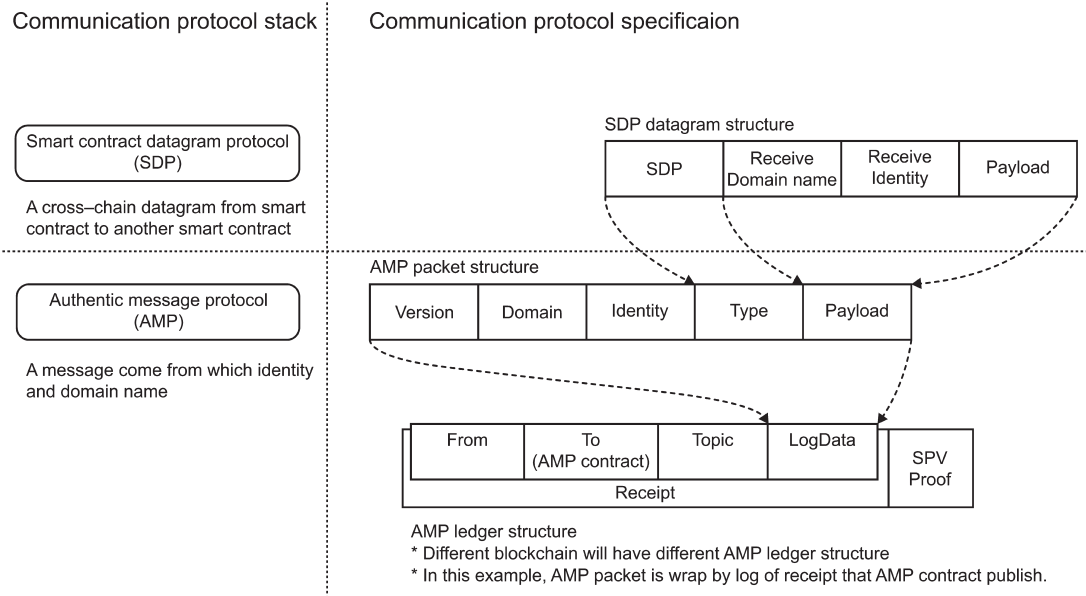


Figure 10—The communication protocol stack

9.2 Authentic message protocol

9.2.1 AMP packet structure

The authentic message protocol is designed to solve the key problem of determining the sender’s identity on which blockchain the smart contract or on-chain program initiates the cross-chain communication.

The structure of AMP contains data identifying the initiator of the message, the type of message, and the content, as illustrated in Table 7. The authentic message protocol provides a mechanism to authenticate the sender’s identity and help prevent any impersonation by unauthorized parties.

Table 7—The structure of AMP

| Field | Type | Description |
|----------|------------------|--|
| Identity | String | The identity of who initiates this message. Only a dedicated on-chain component, e.g., system smart contract, has the permission to fulfill it, so that the field is reliable because it cannot be impersonated. |
| Type | Unsigned integer | The type of this message. It is defined by the AMP. For example, it can be a protocol like UDP in TCP/IP. |
| Payload | String | The content of this message. The data structure of content is defined according to <i>Type</i> . |

9.2.2 AMP ledger structure

AMP is executed by a dedicated on-chain program called the AMP system smart contract. The AMP system contract shall publish the AMP packet in the ledger in order to be obtainable and verifiable.

Obtainable

The off-chain entity obtains the AMP packet by fetching and parsing the ledger according to the AMP ledger structure.

For example, an AMP system smart contract publishes the AMP packet by emitting a log which will be written into the field “To” in receipt of the transaction. The off-chain entity can filter the receipt with the *To* field equal to the address of the AMP system smart contract.

Verifiable

The AMP ledger shall be verifiable. The off-chain entity can verify that the AMP ledger has actually come from the blockchain. For example, if an AMP packet is published in log of receipt, it can be verified through the SPV proof of the receipt’s root.

Second, the AMP ledger itself shall be verifiable. The AMP receiver shall anchor to the corresponding AMP system contract address and verify the AMP ledger to help ensure it is produced by the corresponding AMP system smart contract.

9.2.3 AMP verifiable claim

The blockchain configurator issues the AMP verifiable claim (VC) and contains the AMP trust anchor, as illustrated in [Table 8](#). It is signed by the blockchain and uploaded to the BCDNS server.

Table 8—The structure of AMP verifiable claim

| Field | Type | Description |
|---------------------|--------|---|
| Domain name | String | The domain name of this VC. |
| AMP trust anchor | String | The trust anchor of the AMP ledger. Different blockchain type has different AMP trust anchor object. For example, it may be an AMP system smart contract address. |
| Signature algorithm | String | The signature algorithm of this VC. |
| Signature | String | The signature is signed using the private key of the blockchain domain name certificate. |

9.2.4 On-chain implementation of AMP

The component that implements authentic message protocol must be an on-chain component, e.g., system smart contract.

The on-chain AMP component offers an API to receive the authentic message. This API ensures the integrity of the message by performing key operations, including:

- a) Real identity verifies the caller’s identity is the same as the identity in the authentic message. The message shall be rejected if the verification fails.
- b) Valid content checks the validity of the content by verifying the restrictions of the *type* field and *payload* field.

The on-chain AMP component shall publish a receipt for a valid authentic message. Then the off-chain entity collects the authentic messages by fetching the receipts of the on-chain AMP component. The off-chain entity is responsible for helping ensure that the authentic message is initiated by the right on-chain component by checking the ledger proof in the receipt. Different blockchains implement the authentic messages in different ways, but they shall provide instructions to inform the off-chain entity how to fetch and verify the authentic message.

9.3 Smart contract datagram protocol

9.3.1 SDP packet structure

SDP (Smart contract Datagram Protocol) is a subprotocol of an Authentic Message Protocol. Just like the UDP (User Datagram Protocol) extends the IP (Internet Protocol), SDP extends the AMP.

SDP allows smart contracts to send a datagram to another smart contract on the other blockchain encoded into authentic message. As illustrated in Table 9, the SDP contains the target blockchain domain name, the target identity of the smart contract, and the payload.

Table 9—The structure of SDP

| Field | Type | Description |
|--------------------|--------|--|
| Target domain name | String | The domain name of the target blockchain which receives the datagram. |
| Target identity | String | The identity of the target smart contract in the target blockchain that receives the datagram. |
| Payload | String | The content of this datagram. |

9.3.2 On-chain implementation of SDP

The component that implements SDP shall be an on-chain component, such as a smart contract. The application-level smart contract invokes the SDP component to generate the payload of AMP message and helps ensure that the *identity* field of the AMP message corresponds to the application-level smart contract's identity.

The *type* field of the AMP message is 0x00, which denotes the payload as an SDP. In order to be verifiable, it is suggested that different types of AMP messages are written into the same on-chain AMP component. This way, the off-chain entity configures the AMP information only once and verifies all types of AMP sub-protocols.

10. Addressing protocol

10.1 Introduction

The addressing protocol defines a protocol is designed to allow off-chain relay components to exchange information regarding their associated blockchains and establishes secure connections for relaying UCP among them.

This protocol is built on the top of BCDNS, which allows the blockchain configurator to publish its blockchain domain name network information, including the network information of the connected off-chain relay component. When one blockchain wants to establish the connection with another blockchain, it or the connected off-chain component sends a request with the domain name to the BCDSN and gets the network information.

10.2 Off-chain relay component information verifiable claim

The off-chain relay component information verifiable claim contains the network information and is signed by the connecting blockchain, as illustrated in Table 10. The off-chain relay component information VC can not only help the off-chain relay component connected to the source blockchain to locate the off-chain relay component connected to the target blockchain, but also serves to verify the authenticity and establish a secure channel.

Table 10—The structure of off-chain relay component information VC

| Field | Type | Description |
|---|------------------|--|
| Version | Unsigned integer | The version of VC. |
| Domain name | String | The blockchain domain name. |
| Off-chain relay component public key | String | The relayer’s public key, which represents its identity. |
| Off-chain relay component network address | String | The network address of the off-chain relay component which contains the host and port. |
| Signature algorithm | String | The signature algorithm of this VC. |
| Signature | String | The signature that is signed by private key of blockchain domain name certificate. |

10.3 The procedure of addressing protocol

The procedure of addressing protocol consists of the following steps:

- 1) The blockchain configurator chooses one suitable proof transformation component to use;
- 2) The blockchain configurator constructs the off-chain relay component information VC and signs it with the blockchain domain name certificate. Then the blockchain configurator registers the off-chain relay component information VC in BCDNS;
- 3) BCDNS uses the public key in the blockchain domain name certificate to verify the VC to help ensure that it is not illegally generated;

The off-chain relay component connected with the source blockchain sends a request to BCDNS to obtain the information of the off-chain relay component connected with the target blockchain;

- 4) The secure channel is set up between two off-chain relay components, and the UCP is forwarded.

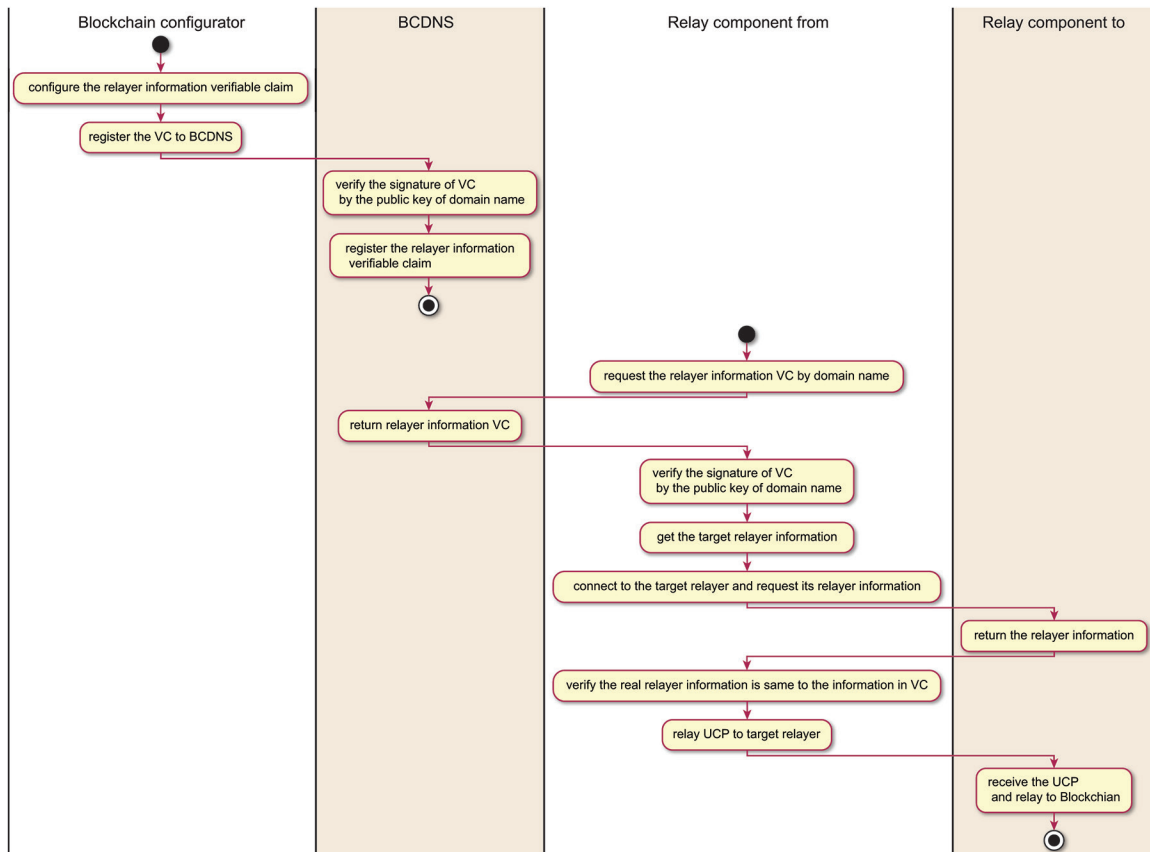


Figure 11—Illustration of addressing procedure

11. Cross-chain data authentication and communication procedure

Figure 12 illustrates the one-way cross-chain message data authentication and communication procedure with the message transmitted from source blockchain to target blockchain.

- 1) The application layer of the source blockchain initiates a cross-chain message, constructs the SDP, and sends SDP to the transport layer;
- 2) The transport layer of the source blockchain generates an AMP message and writes it into the blockchain ledger, e.g., writing receipt events;
- 3) The data link layer of the source blockchain reads the newly generated AMP ledger. Then it uses the original ledger and the original ledger proof to construct the UCP and invokes a proof transformation procedure inside the secure authentication layer;
- 4) The secure authentication layer verifies the original ledger proof in the UCP. If the verification is passed, the third-party ledger proof is issued and returned;
- 5) The data link layer of the source blockchain sends the UCP to the data link layer of the target blockchain;
- 6) The target off-chain relay component verifies the UCP and commits valid UCP to secure the authentication layer;
- 7) The secure authentication layer of the target blockchain verifies the UCP based on third-party ledger proof and commits valid UCP to the transport layer;

- 8) The target transport layer parses the SDP from the UCP and sends it to the target application layer;
- 9) Finally, the target application layer receives the SDP from the source application layer.

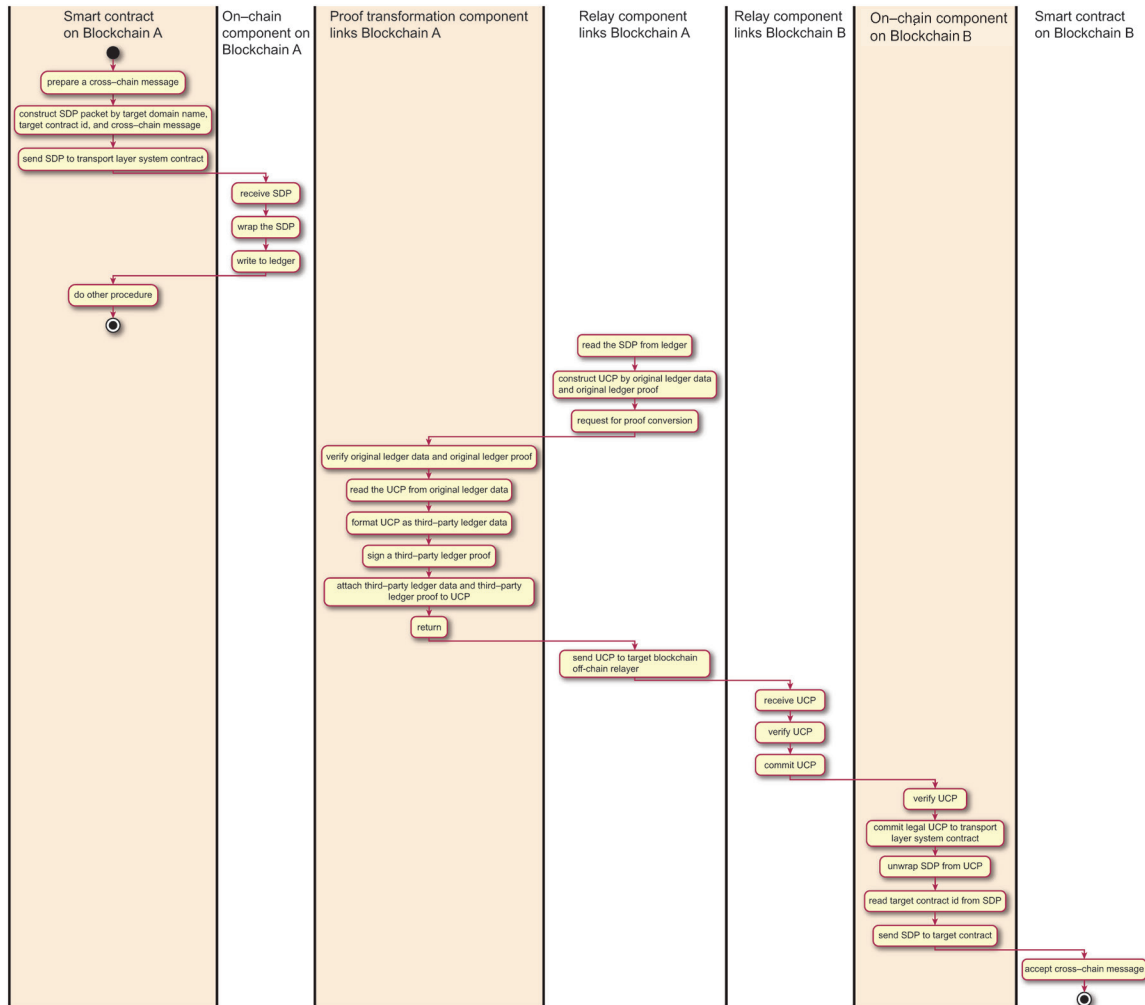


Figure 12—Illustration of one-way cross-chain communication procedure

12. Technical and security requirements

12.1 Technical requirements

The following technical requirements shall be satisfied:

- 1) Ability to implement blockchain assets freezing operation, and set the locking and unlocking conditions;
- 2) Support for inter-chain ledger access
- 3) Support for inter-chain data query
- 4) Support for inter-chain smart contract message communication;

- 5) High parallel processing capacity;
- 6) Support for the interoperability of both homogeneous and heterogeneous blockchain systems;
- 7) Ability to achieve the final atomicity, meaning the transaction is either executed successfully or failed in both blockchain systems;
- 8) Support globally verifiable blockchain identity and non-interactive cross-chain identity authentication;
- 9) Support the verification of cross-chain data through a third-party trusted verifier. The third-party trusted verifier mechanism includes single signature, multiple signature, and trusted execution environment (TEE).

12.2 Security requirements

The cross-chain technology shall satisfy the following security requirements:

- 1) Designing the cross-chain principle and implementation mechanism with security in mind to avoid problems such as malicious transactions;
- 2) Confirming the integrity of cross-chain packet data and confirm the authenticity of the source;
- 3) Being able to resist eclipse attack, double spend attack, and other blockchain threats;
- 4) Establishing a secure communication channel and confirm the confidentiality, integrity, and availability of the cross-chain data information;
- 5) Confirming that the source blockchain and target blockchain are able to verify the domain name certification of each other.

Annex A

(informative)

Examples of blockchain interoperability

A.1 Cross-chain smart contract communication

A smart contract on the blockchain #X sends a message to another smart contract on a different blockchain #Y to achieve communication between the two.

A.2 Asset transfer

After the assets on the blockchain #X have been destroyed, the destruction information is transmitted to the blockchain #Y. The blockchain #Y issues the equivalent assets according to the cross-chain asset transfer agreement. As shown in the following figure, Alice wants to transfer 100 assets from blockchain #X to blockchain #Y. After a successful transaction, the 100 assets on the blockchain #X are reduced (the reduced 100 assets will be frozen at the specific address of chain #X), and the corresponding equivalent assets will be generated on the blockchain #Y.

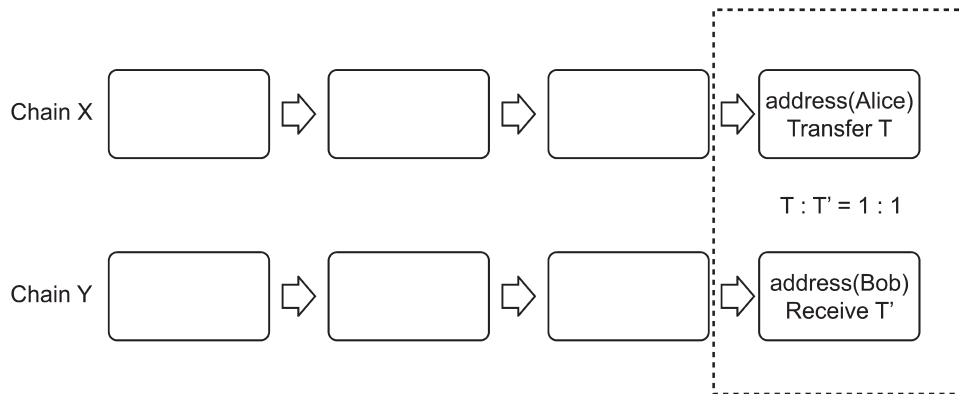


Figure A.1—An example of asset transfer

A.3 Asset exchange

In a cross-chain asset exchange scenario, users can use the assets on the blockchain #X to exchange for assets on blockchain #Y. The total amount of assets on each chain remains constant, but the asset ownership changes, and the ownership change is synchronized between the two blockchain systems. For example, as shown in this figure, Alice wants to exchange 10 assets on the blockchain #X for Bob's 100 assets on the blockchain #Y. After the successful transaction, Bob's 100 assets on the blockchain #Y are transferred to Alice's address on the blockchain #Y, while Alice's 10 assets on the blockchain #X are transferred to Bob's address on the blockchain #X. The total assets of blockchain #X and blockchain #Y do not increase or decrease.

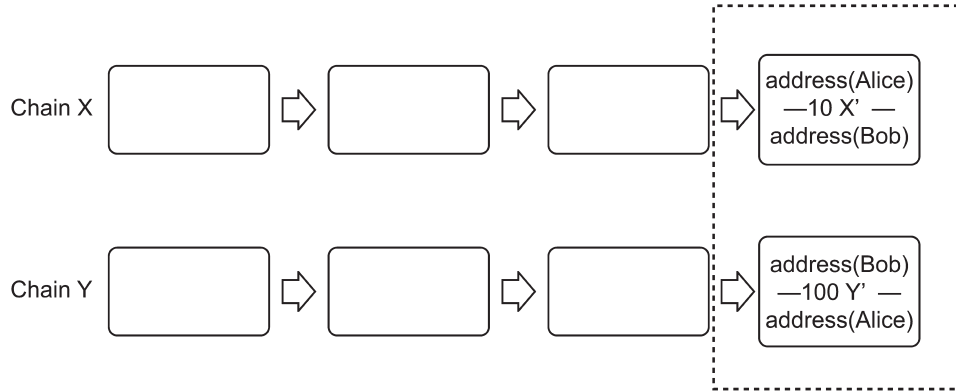


Figure A.2—An example of asset exchange

A.4 Blockchain platforms

Blockchain platforms have been built to cover various types of applications in major fields (e.g., the State Grid blockchain covers the fields of energy, government affairs, and finance). Through the blockchain, platforms can support the penetrating trust between industry subjects and external industry subjects, realizes data collection and aggregation, audit tracking analysis, point-to-point data transmission, and multi-party collaboration between isomorphic chains/heterogeneous chains, and realizes transaction, dispatching, production, finance, and materials. The whole process innovation of marketing and other businesses can be structured to support the intelligent decision-making of production, the innovation of the business model, the optimal allocation of resources, and the cultivation of industrial ecology.

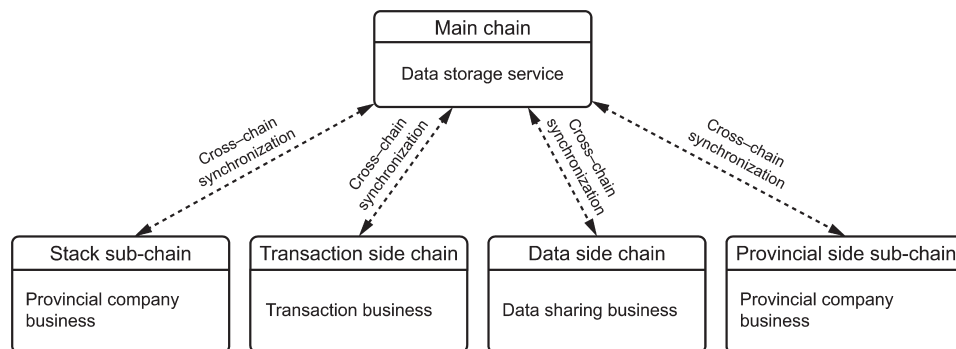







Figure A.3—Overall structure of a blockchain

Blockchain platforms can fully consider the actual situation and application needs of an electric utility and various units within the utility, and form an overall architecture including main-chain, stack sub-chain, transaction side chain, data side chain, and provincial side sub-chain based on the alliance chain architecture. Through a cross-chain platform, the data interconnection between main-chain, sub-chain, and cross-chain services can be realized to provide expandable boundaries. The blockchain can have high cohesion and low coupling and the ability to connect with a national blockchain platform. This can help solve the problems of low industrial coordination efficiency and difficult mutual trust in the construction of an energy Internet by building application scenarios such as transaction, materials, online industrial chain finance, and judicial certificate deposit based on blockchain for business fields such as energy, finance, and government affairs.



RAISING THE WORLD'S STANDARDS

Connect with us on:

-  **Twitter:** twitter.com/ieeesa
-  **Facebook:** facebook.com/ieeesa
-  **LinkedIn:** linkedin.com/groups/1791118
-  **Beyond Standards blog:** beyondstandards.ieee.org
-  **YouTube:** youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060