ICS
L

CIIA

团　体　标　准

T/ CIIA xxx---xxxx

区块链跨链交互协议

Blockchain Interoperability Protocol

（征求意见稿）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

中国信息协会　发布

目　录

前　言

本文件按照GB/T 1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件由中国信息协会提出并归口。

本文件起草单位：

本文件主要起草人：

区块链跨链交互协议

1 范围

本协议描述了跨链协议的参考框架、跨链协议各组成部件的功能特征、跨链消息交互流程和步骤，以及跨链协议参与方之间的数据处理过程、消息的格式与内容等。

本协议用于指导跨链基础设施的研发和设计，也适应于指导区块链系统接入跨链基础设施时的开发、测试和评估。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25069-2010 信息安全技术 术语
GB/T 22239-2019 信息安全技术 网络安全等级保护基本要求
GB/T 28448-2019 信息安全技术 网络安全等级保护测评要求
C 0116-2018 国家政务服务平台网络安全保障要求
GM/T 0054-2018 信息系统密码应用基本要求
GB/T 35276 信息安全技术 SM2密码算法使用规范
GB/T 35275 信息安全技术 SM2密码算法加密签名消息语法规范
GB/T 32905 信息安全技术 SM3密码杂凑算法
GB/T 32907 信息安全技术 SM4分组密码算法
GB/T 37932-2019 信息安全技术 数据交易服务安全要求
20173824-T-469 信息技术 区块链和分布式账本技术 参考架构（征求意见稿）
20210991-T-469 信息安全技术 区块链信息服务安全规范（征求意见稿）
20210998-T-469 信息安全技术 区块链技术安全框架（征求意见稿）

3 术语和定义

下列术语和定义适用于本文件。

3.1 区块链 blockchain

一种按照时间顺序将数据区块以顺序相连的方式组合成的链式数据结构，并以密码学方式保证不可篡改和不可伪造的分布式账本。

3.2 分布式账本 distributed ledger

可以在多个站点、不同地理位置或者多个机构组成的网络里实现共同治理及分享的资产数据库。

3.3 智能合约 smart contract

以数字形式定义的能够自动执行条款的合约。在区块链技术领域，智能合约是指基于预定事件触发、难以篡改、自动执行的计算机程序。

3.4 令牌 token

授权服务发放的凭证，用于证明某实体具有访问特定范围内受保护资源的权限。

3.5 第三方应用程序 third party application

请求访问受保护资源的应用程序。

3.6 源链 source chain

在跨链协议中，主动发起跨链请求的一方。

3.7 目标链 destination chain

在跨链协议中，被动接收跨链请求的一方。

3.8 原子性 atomic

事务中的操作要么都发生，要么都不发生。

3.9 跨链事务 cross-chain transaction

源链发起的以变更源链和目标链双方状态为目的的请求，此请求对双方状态的变更满足原子性的行为信息。

3.10 群组/通道 group/channel

群组，在某些区块链系统中又称通道、子链，将区块链网络中的全部或部分节点连接在一起，形成一个个逻辑独立的网络空间，实现了业务隔离的要求。群组内部节点成员共享同一份区块链账本，不同群组之间的区块链账本物理隔离。

3.11 节点安全监测系统 Chain Node Supervision System

安全监测系统为基础链和一般业务链提供精准的区块链安全监测服务，对节点的身份、行为、应用、链上链下数据进行全方位监测，对节点的接入、审批、下线、离链等进行监测与管理。

4 缩略语

下列缩略语适用于本文件。

API：应用程序接口（Application Programming Interface）

FAQs：经常被提出的问题（Frequently Asked Questions）

HTTPS：超文本传输安全协议（Hyper Text Transfer Protocol over Secure Socket Layer）

CA：证书认证机构（Certification Authority）

ECC：椭圆曲线密码（Elliptic Curve Cryptography）

SSL：安全套接层（Secure Sockets Layer）

VPN：虚拟专用网（Virtual Private Network）

5 跨链交互技术框架

5.1 跨链参考模型

跨链涉及到不同的区块链平台及架构的数据交互，需要实现不同区块链系统之间、不同区块链平台之间的互联互通，构造一个架构可扩展、安全性高、实用性强的模型架构是实现安全可靠的跨链技术的关键。

为了实现能够兼容异构区块链的跨链机制，需要对数据格式、消息类型、通信方式等进行统一，以支撑数据能在不同区块链之间畅通的流转；同时要保证各个不同的链具备高度的独立性，避免过度依赖，防止在某些链出现异常情况时影响其他链的正常运行。

基于以上考虑，本标准采用中继链技术为骨架来搭建跨链参考模型，采用区块链的技术来实现以链跨链，并结合政务等多领域的社会实践，提出两种不同的跨链模型，包括同级跨链模型和多级跨链模型。

5.1.1 同级跨链模型

当各个子链地位平等，需要横向跨链时，可采用同级跨链模型。其参考模型如图 1 所示。在同级跨链模型中，存在中继主链和业务子链两大主要部分。中继主链本身也是一个独立的区块链网络，用于桥接其他业务子链，中继主链中有跨链节点，专职负责业务子链的接入；各业务子链通过跨链节点，借助中继主链进行中转访问其他业务子链。各业务子链可采用同构或者异构的区块链底层平台，构建相互独立的区块链网络。

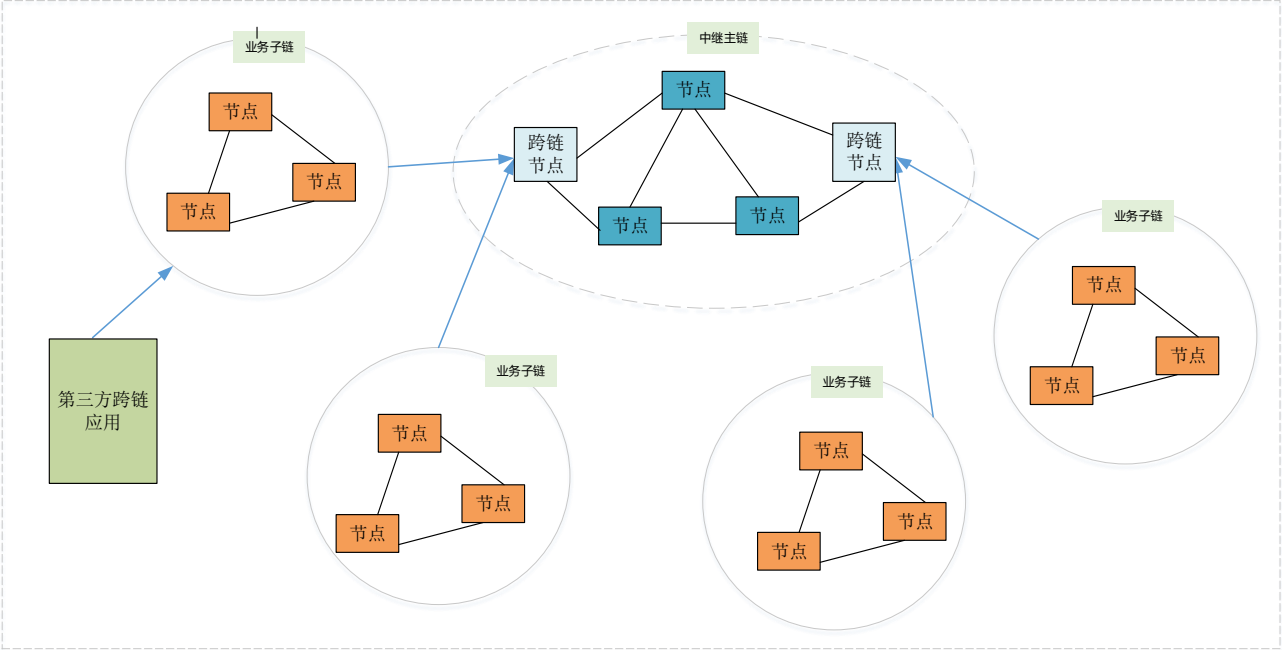

图 1 同级跨链模型

当第三方跨链应用在处理跨链业务时，将请求首先送达为其提供区块链服务的业务子链，业务子链通过跨链节点接入中继主链，中继主链将请求路由转发给需要处理的目标业务子链。

5.1.2 多级跨链模型

当各个子链存在层级关系时，需要纵向跨链时，可采用多级跨链模型。其参考模型如图 2 所示。在多级跨链模型中，每一层级都是一个完整的同级跨链模型，都存在中继主链和业务子链两大部分。同时各层级之间的主链组成树形结构。当业务子链需要与业务子链进行跨链时，首先通过主链树进行路由寻

址，找到对应的中继主链，并最终调度到目标业务子链之中。

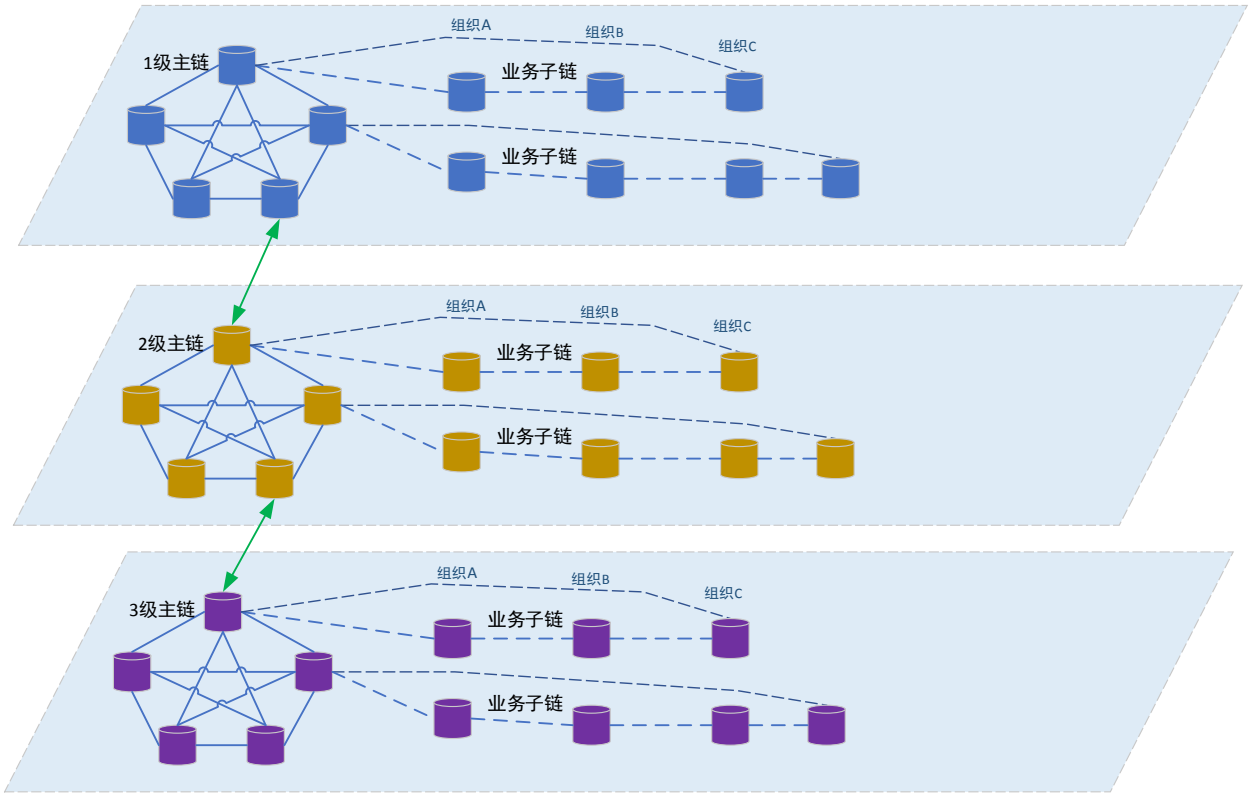

图 2 多级跨链模型

当某业务子链收到跨链业务处理请求时，将请求首先送达到为其提供中继服务的主链上，该主链将在主链树进行路由寻址，将请求转发至目标业务子链所接入的中继主链上，最后数据将路由到该业务子链上，完成整个跨链处理。

5.2 中继主链

中继主链为各业务子链提供了以中继链技术实现的数据桥梁。中继主链本身也是一个区块链系统，以区块链支撑包括主链管理、跨链中继、跨链记录、跨链统计等业务。

中继主链相对独立，并向各业务子链提供统一的接入接口。中继主链的节点包括普通节点与跨链节点，普通节点只负责维持中继主链网络，记录中继数据等功能，跨链节点面向业务子链，提供跨链接入。

5.3 跨链节点

跨链节点负责业务子链的统一接入，又可称为跨链网关。跨链节点是业务子链和中继主链交互的第一道关卡，其负责对请求进行安全性校验，阻止非法请求进入中继主链。

跨链节点提供统一接口供业务子链接入，当业务子链按统一规范的接口封装数据和调用接口，即可完成跨链互操作。

5.4 业务子链

业务子链以区块链技术提供具体的业务功能，如存证、监管、溯源、金融等行业应用业务。

业务子链通过跨链节点按统一的接口实现接入中继主链，由中继主链将跨链消息调度给需要跨入的目标业务子链。

5.5 第三方跨链应用

第三方跨链应用是指需要跨链技术来提供业务场景的应用程序。第三方跨链应用通常由一个业务子链为其提供区块链服务，并通知业务子链进行跨链操作。

6 区块链跨链交互流程

6.1 协议总流程

区块链跨链交互流程主要分为三个阶段，分别为子链线上审批、子链入网及日常跨链交互三个部分，如图 3 所示。

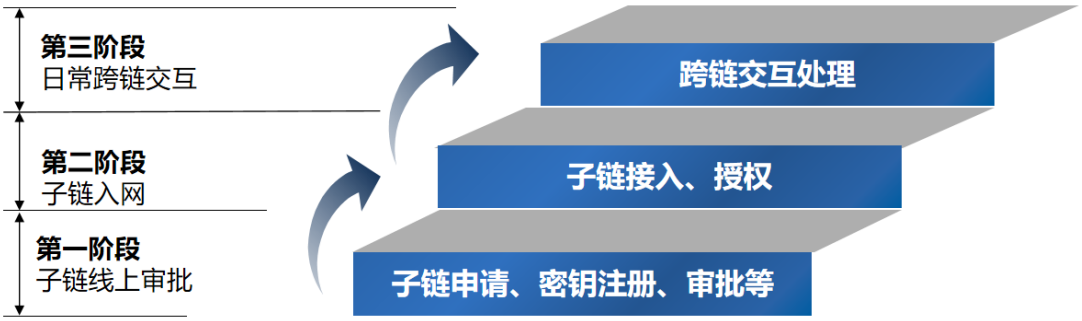

图 3 区块链跨链交互总流程

第一阶段主要完成子链线上审批，由主链管理部门对子链的新增、修改等进行审核，包括子链申请、密钥注册、子链审批等操作。

第二阶段主要完成子链的初始入网，包括子链平台或应用的接入、子链或群组的授权等。

第三阶段完成日常的跨链交互，包括源区块链与目标区块链之间的具体消息交互及数据交换等处理。

6.2 子链线上审批

该步骤主要完成子链线上审批，规划由主链管理部门统一审批。审批类型含新建子链审批和子链节点扩容审批。

6.2.1 审批信息录入

当需要新建子链和子链节点扩容时，需要主链管理部门进行审批。

当审批类型为新建子链审批时，录入内容包括：子链标识（系统审批后产生）、子链名称、子链描述、子链架构（平台选型）、责任单位（运营主体）、联系人、联系电话、子链节点情况（含节点数量、节点位置等信息）等。

当审批类型为子链节点扩容审批时，录入内容包括：子链标识、扩充节点情况（含扩充节点数量、扩充节点位置等信息）等。

6.2.2 审批处理

主链管理部门统一负责审批事务。审批内容包括：审批结论（同意或驳回）。

当审批成功通过后，将进行分发管理密钥、生成子链唯一标识等系统操作。

6.3 子链集成与入网

子链基于子链标识与管理密钥来向主链发起入网申请，主链对子链入网做协议审核，入网成功后，更新子链的入网状态。

6.3.1 子链鉴权接口

子链必须提供鉴权接口。鉴权接口用于子链完成对中继主链的身份鉴定和权限范围判断。

6.3.2 子链接入接口

6.3.2.1 上链存证接口

子链必须提供上链存证接口。上链存证接口用于处理跨链交易事务，完成业务子链状态的变更。

6.3.2.2 链上查询接口

子链必须提供链上查询接口。链上查询接口用于处理跨链查询事务，通常不会引起子链状态的变化。

6.3.2.3 跨链回滚接口

子链必须提供跨链回滚接口。跨链回滚接口将回滚对应的上链存证交易。

6.4 子链及群组授权

完成针对子链的访问控制，可配置子链及其群组（某些链称通道或 Channel）允许哪些子链或子链的群组进行访问。

6.4.1 授权子链访问

当前子链及其群组配置某一个或多个子链进行访问，配置成功后，当前子链的所有群组都允许访问该子链及其群组。

6.4.2 授权群组访问

当前子链及其群组配置某一个或多个群组进行访问，配置成功后，其他未授权的群组无法访问该子链及其群组。

6.5 跨链交互

6.5.1 网络模型

在同级跨链模型中，通常为“1中继主链+N业务子链”的网络模型。首先打造一个主链中继网络，负责子链与子链间的跨链交互，各业务子链通过调用协议接口的形式接入，期间各业务子链指定一个跨链节点，专属用于跨链交互，通过跨链节点实现了子链与主链的流通，继而通过主链完成与其他子链的跨链流程。其模型如图4所示。

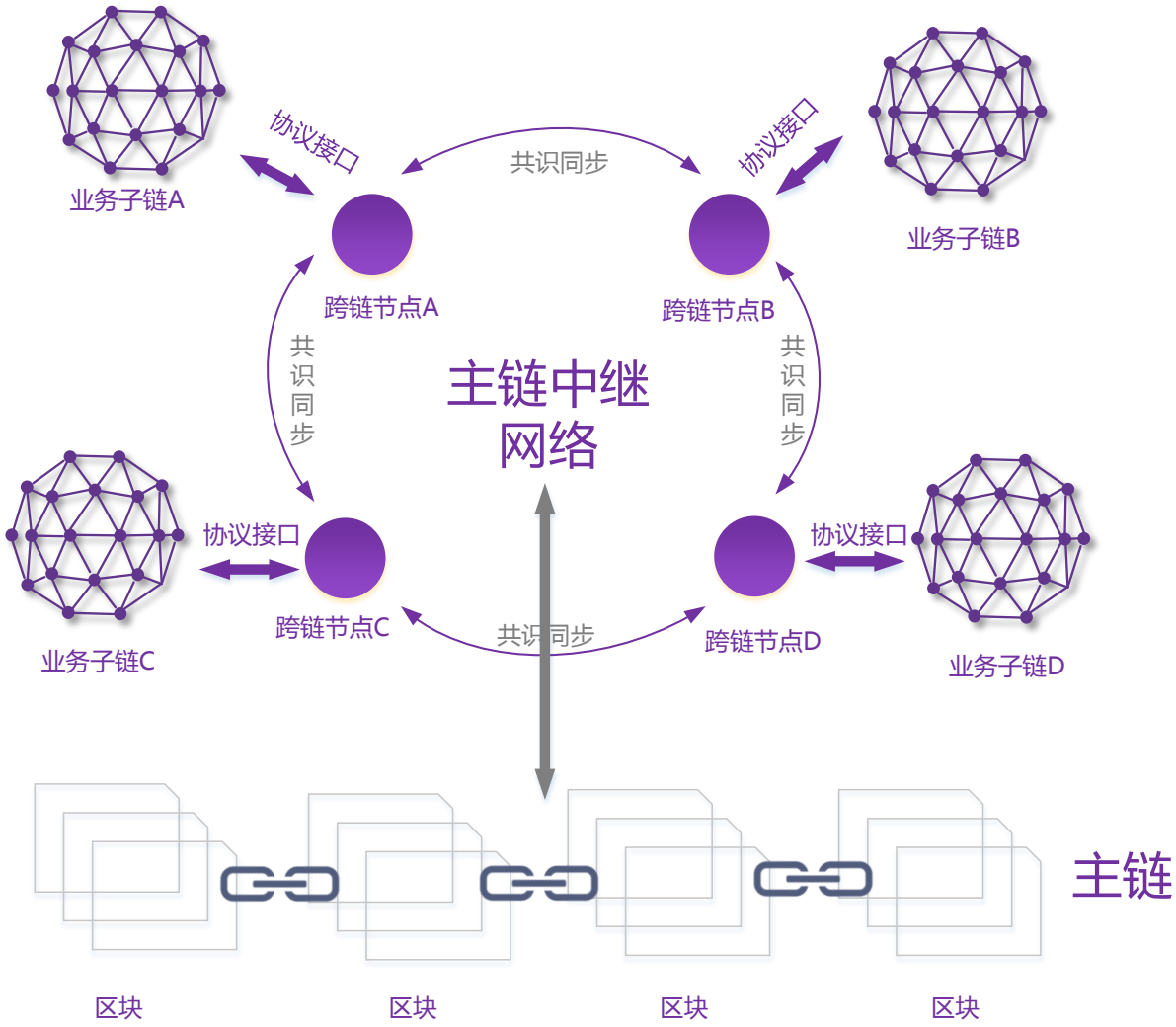

图 4 区块链跨链交互模型

详细步骤如下：

1、业务子链接入：业务子链将指定一个跨链节点，专职用于跨链事务，通过协议接口调用，业务子链将联通跨链节点，从而与主链产生数据交换。

2、跨链节点共识：各业务子链的跨链节点将组建成主链中继网络，完成共识机制，保持跨链事务在网络中的寻址与路由至目标区块链。

3、主链中继：主链记录跨链信息后，将依次执行各区块链的交易，完成跨链事务的处理。

6.5.2 交互流程

交互流程主要处理区块链跨链业务，该请求由第三方应用程序发起，由子链或跨链节点完成业务响应。如图5所示。

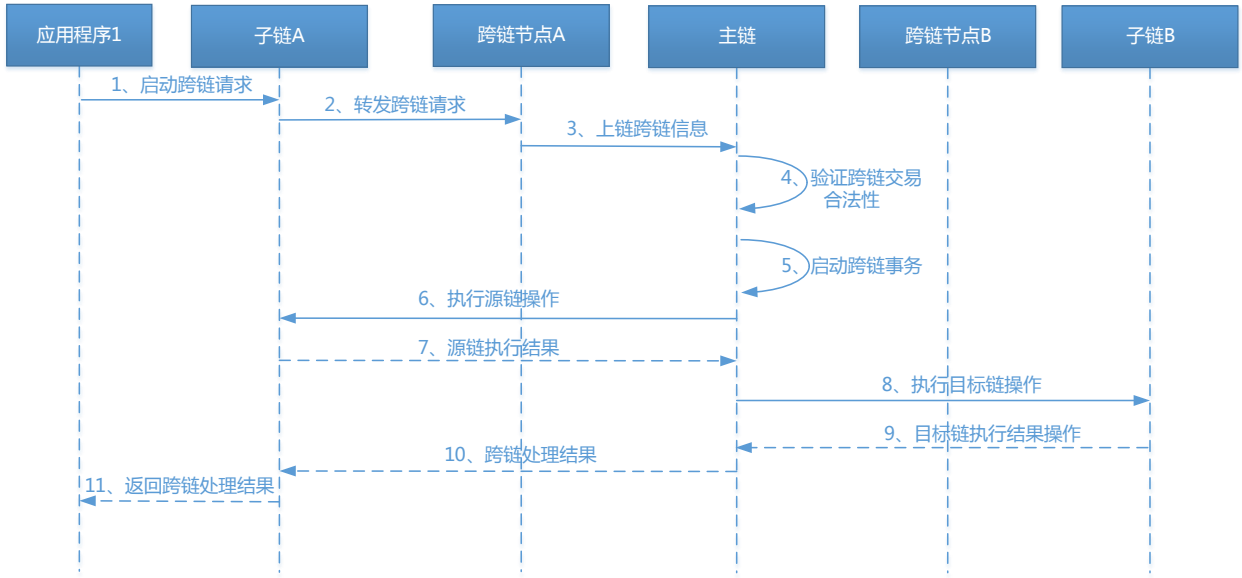

图 5 跨链交互消息流程

详细步骤如下：

1、应用程序1启动跨链请求：第三方应用程序开启跨链请求，该消息将由某子链A进行接收。

2、子链A转发跨链请求：子链A接收到第三方应用程序的跨链请求，并不会马上处理，而是转发给跨链节点A，由跨链节点A来完成跨链事务处理。

3、跨链节点A上链跨链请求：跨链节点A在收到跨链请求时，首先将通过主链进行上链跨链交易。

4、主链验证跨链交易合法性：主链A将验证子链A、子链B是否具备跨链权限，验证跨链请求的消息完整性等。

5、主链启动跨链事务：主链将启动跨链事务处理，跨链事务将具备原子性，即所有子链都执行成功，或所有子链都执行失败。

6、主链执行源链操作：主链调用源链执行交易，得到源链执行结果。

7、主链执行目标链操作：主链调用目标链执行交易，得到目标链执行结果。

8、主链向子链A返回跨链处理结果：主链判断是否完成跨链事务，如果都成功，即确认跨链成功；如果有子链执行失败，则对对应的链执行回滚操作。同时，将事务处理结果返回给子链A。

9、子链A返回跨链处理结果：子链A将跨链处理结果通过响应的方式返回给第三方应用程序。

6.5.3 跨链请求

跨链请求由子链发起，并由跨链节点响应，完成子链跨另一条子链的完整流程。跨链请求分为同步跨链请求及异步跨链请求。同步跨链指系统完成分布式跨链事务后，完成响应。异步跨链指系统接收到请求后，首先进行响应，提示已受理请求，同时启动一个异步任务，处理该请求，待处理成功后，更新请求状态。

6.5.3.1 同步跨链请求

请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/bciplatform/bci.do
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

请求消息格式

请求消息的参数说明如下：

参数	说明	参数类型	是否必填	备注
bci_tx_id	跨链交易id	字符串		跨链交易的唯一标识。如果填写则必须确保唯一性； 跨链交易id生成规则：Hash(appid_userid_20220102_6位随机数) 如果为空则系统自动生成，并在响应中返回。
src_chain_id	源链标识	字符串	必填	源链在系统中的唯一标识。
src_tx_id	源交易号	字符串		源链上的交易顺序号。用于溯源当前跨链交易。
target_chain_id	目标链标识	字符串	必填	目标链在系统中的唯一标识。
content	上链或查链内容	字符串	必填	
tx_type	交易类型	字符串	必填	值为“invoke”或“query”。 invoke:跨链执行类交易。表示成功执行该交易后，目标链的状态将会发生某种变化。 query：跨链查询类交易。表示仅仅查询目标链中的某个状态或值，通常不会引起状态的变化。
bci_time	跨链时间	字符串	必填	格式为yyyy-MM-dd HH:mm:ss。yyyy表年份数值，MM表月份数值，dd表日期数值，HH表小时数值，采用24小时制，

				mm表分钟数值，ss表秒钟数值。
bci_timestamp	跨链时间戳	字符串		
src_user_id	源链用户标识	字符串	必填	系统为源链访问跨链平台分发的用户标识。
target_user_id	目标链用户标识	字符串		系统为目标链访问跨链平台分发的用户标识。
bci_tx_hash	跨链交易哈希	字符串		对该交易做完整性哈希。
signature	签名值	字符串		计算交易哈希的签名值。
ca_info	证书信息	字符串		证书格式。
cert_type	证书类型	字符串		
encryption	是否加密	布尔型		跨链交易的报文是否经过加密。值为true或false，默认为false。
crypto_algorithm	密码算法	字符串		格式为“哈希算法类型-签名算法类型-加密算法类型”。

示例：

http://127.0.0.1:7055/bciplatform/bci.do


```
{
 "bci_tx_id":"
 0X3F900C90CDF98164FAC99D47E9C1FA35F48CB1ED26F1FC4A2FABB9FCA37D3B0B",
 "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
 "src_tx_id":"
 0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8FFFTQ",
 "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
 "content":"{\"action\":\"save\",\"key\":\"key1\",\"value\":\"value2\"}",
 "tx_type":"normal",
 "bci_time":"2021-07-25 00:00:00",
 "bci_timestamp":"
 0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF452FAFC4A",
 "src_user_id":"src_user_0x001",
 "target_user_id":"tar_user_0a009",
 "bci_tx_hash":"
 0X9C1FA39D47EA37D4ABB9FC4FAC1ED26F133BFF9098CBF9BF452FA0C90CD0816C",
 "signature":"
 0ee20a92c6b4dc1451226f474b0b9b6922e9a432b26a47d25b062e91664c1cec2ef33b6958b29
 8ea67740b5d9d24b65f3272adcb4f4587520cc177ed0e8df7e1",
 "ca_info":"
 0XB09C1FA39D4BC64FAD26AEA34BACDF90F40C7DF1C1EB9798CBFF952F3381FC90",
 "cert_type":"pem",
```

```
    "encryption":true,
    "crypto_algorithm":"SM3-SM2-SM4"
  }
```

响应消息格式

响应消息的参数说明如下：

参数	说明	备注
bci_tx_id	跨链交易id	跨链交易的唯一标识。
src_chain_id	源链标识	
src_tx_id	源交易号	申请方的交易顺序号。
target_chain_id	目标链标识	
bci_time	跨链返回时间	
bci_timestamp	跨链返回时间戳	
src_user_id	源链用户标识	
target_user_id	目标链用户标识	
success	是否成功	
result	交易结果	
error_code	错误码	
src_chain_result	源链处理结果	
target_chain_result	目标链处理结果	

示例：

成功：

```
{
  "bci_tx_id":"0X3F900C90CDF98164FAC99D47E9C1FA35F48CB1ED26F1FC4A2FABB9FCA37D
  3B0B ",
  "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
  "src_tx_id":"0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8F
  FFTQ ",
  "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
  "bci_time":"2021-07-25 00:00:00",
  "bci_timestamp":"0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF4
  52FAFC4A ",
  "src_user_id":"src_user_0x001",
  "target_user_id":"tar_user_0a009",
  "success": true,
  "result":"invoke success."
  "error_code":200,
  "src_chain_result":"invoke success",
  "target_chain_result":"OK"
}
```

```
失败：
{
  "bci_tx_id":"0X3F900C90CDF98164FAC99D47E9C1FA35F48CB1ED26F1FC4A2FABB9FCA37D
  3B0B ",
  "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
  "src_tx_id":"0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8F
  FFTQ",
  "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
  "bci_time":"2021-07-25 00:00:00",
  "bci_timestamp":"0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF4
  52FAFC4A",
  "src_user_id":"src_user_0x001",
  "target_user_id":"tar_user_0a009",
  "success": false,
  "result":"unknown exceptions."
  "error_code":500,
  "src_chain_result":"invoke success",
  "target_chain_result":"Fail"
}
```

6.5.3.2 异步跨链请求

异步跨链请求分为异步跨链受理请求和查询跨链受理结果两个请求消息。

异步跨链受理请求

请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/bciplatform/bci_async.do
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

请求消息格式

请求消息的参数说明如下：

参数	说明	备注
bci_tx_id	跨链交易id	跨链交易的唯一标识。
src_chain_id	源链标识	
src_tx_id	源交易号	申请方的交易顺序号。

target_chain_id	目标链标识	
content	上链/查链内容	
tx_type	交易类型	
bci_time	跨链时间	
bci_timestamp	跨链时间戳	
src_user_id	源链用户标识	
target_user_id	目标链用户标识	
bci_tx_hash	跨链交易哈希	对该交易做完整性哈希。
signature	签名值	计算交易哈希的签名值。
ca_info	证书信息	证书格式。
cert_type	证书类型	
encryption	是否加密	跨链交易的报文是否经过加密。
crypto_algorithm	密码算法	格式为“哈希算法类型-签名算法类型-加密算法类型”。

示例：

http://127.0.0.1:7055/bciplatform/bci_async.do


```
{
  "bci_tx_id":"
  0X3F900C90CDF98164FAC99D47E9C1FA35F48CB1ED26F1FC4A2FABB9FCA37D3B0B",
  "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
  "src_tx_id":"
  0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8FFFTQ",
  "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
  "content":"{\"action\":\"save\",\"key\":\"key1\",\"value\":\"value2\"}",
  "tx_type":"normal",
  "bci_time":"2021-07-25 00:00:00",
  "bci_timestamp":"
  0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF452FAFC4A",
  "src_user_id":"src_user_0x001",
  "target_user_id":"tar_user_0a009",
  "bci_tx_hash":"
  0X9C1FA39D47EA37D4ABB9FC4FAC1ED26F133BFF9098CBF9BF452FA0C90CD0816C",
  "signature":"
  0ee20a92c6b4dc1451226f474b0b9b6922e9a432b26a47d25b062e91664c1cec2ef33b6958b
  298ea67740b5d9d24b65f3272adcb4f4587520cc177ed0e8df7e1",
  "ca_info":"
  0XB09C1FA39D4BC64FAD26AEA34BACDF90F40C7DF1C1EB9798CBFF952F3381FC90",
  "cert_type":"pem",
  "encryption":true,
  "crypto_algorithm":"SM3-SM2-SM4"
}
```

响应消息格式

响应消息的参数说明如下：

参数	说明	备注
bci_tx_id	跨链交易id	跨链交易的唯一标识。
src_chain_id	源链标识	
src_tx_id	源交易号	申请方的交易顺序号。
target_chain_id	目标链标识	
bci_time	跨链返回时间	
bci_timestamp	跨链返回时间戳	
src_user_id	源链用户标识	
target_user_id	目标链用户标识	
success	是否受理成功	
result	交易结果	
error_code	错误码	
src_chain_result	源链处理结果	
target_chain_result	目标链处理结果	

示例：

成功：


```
{
  "bci_tx_id":"0X3F900C90CDF98164FAC99D47E9C1FA35F48CB1ED26F1FC4A2FABB9FCA37D
  3B0B ",
  "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
  "src_tx_id":"0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8F
  FFTQ ",
  "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
  "bci_time":"2021-07-25 00:00:00",
  "bci_timestamp":"0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF4
  52FAFC4A ",
  "src_user_id":"src_user_0x001",
  "target_user_id":"tar_user_0a009",
  "success": true,
  "result":"invoke success."
  "error_code":200,
  "src_chain_result":"invoke success",
  "target_chain_result":"OK"
}
```


失败：


```
{
  "bci_tx_id":"0X3F900C90CDF98164FAC99D47E9C1FA35F48CB1ED26F1FC4A2FABB9FCA37D
  3B0B ",
  "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
```

```
    "src_tx_id":"0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8F
    FFTQ",
    "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
    "bci_time":"2021-07-25 00:00:00",
    "bci_timestamp":"0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF4
    52FAFC4A",
    "src_user_id":"src_user_0x001",
    "target_user_id":"tar_user_0a009",
    "success": false,
    "result":"unknown exceptions."
    "error_code":500,
    "src_chain_result":"invoke success",
    "target_chain_result":"Fail"
  }
```


查询跨链受理结果

请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/bciplatform/get_bci_result.do
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

请求消息格式

请求消息的参数说明如下：

参数	说明	备注
bci_tx_id	跨链交易id	跨链交易的唯一标识。
src_chain_id	源链标识	
src_tx_id	源交易号	申请方的交易顺序号。
src_user_id	源链用户标识	

示例：

http://127.0.0.1:7055/bciplatform/get_bci_result.do


```
 {
   "bci_tx_id":"0XED00C7DCA3F426990CD33B09C1A399F1FAFC4ABFD47EBB8CBFF952F98164
   FAC1",
   "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
   "src_tx_id":"
   0XF981641FABB00C1ED26F133B0A90C9EA37DCBFDFA9C47452FAFD3C4998CBFF9C",
   "src_user_id":"src_user_0x001"
```

```
    }
```

响应消息格式

响应消息的参数说明如下：

参数	说明	备注
bci_tx_id	跨链交易id	跨链交易的唯一标识
src_chain_id	源链标识	
src_tx_id	源交易号	申请方的交易顺序号
target_chain_id	目标链标识	
bci_time	跨链返回时间	
bci_timestamp	跨链返回时间戳	
src_user_id	源链用户标识	
target_user_id	目标链用户标识	
success	是否成功	
result	交易结果	
error_code	错误码	
src_chain_result	源链处理结果	
target_chain_result	目标链处理结果	

示例：

成功：

```
{
  "bci_tx_id":"0XED00C7DCA3F426990CD33B09C1A399F1FAFC4ABFD47EBB8CBFF952F98164
  FAC1",
  "src_chain_id":" CN3601007a7a28c4a3a68778a164c2da603bd933",
  "src_tx_id":"0XF981641FABB00C1ED26F133B0A90C9EA37DCBFDFA9C47452FAFD3C4998CB
  FF9C",
  "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
  "bci_time":"2021-07-25 00:00:00",
  "bci_timestamp":"0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF4
  52FAFC4A ",
  "src_user_id":"src_user_0x001",
  "target_user_id":"tar_user_0a009",
  "success": true,
  "result":"query success."
  "error_code":200,
  "src_chain_result":"query result",
  "target_chain_result":"OK"
}
```

失败：

```
{
```

```
    "bci_tx_id":"0XED00C7DCA3F426990CD33B09C1A399F1FAFC4ABFD47EBB8CBFF952F98164
    FAC1",
    "src_chain_id":"CN3601007a7a28c4a3a68778a164c2da603bd933",
    "src_tx_id":"0XF981641FABB00C1ED26F133B0A90C9EA37DCBFDFA9C47452FAFD3C4998CB
    FF9C ",
    "target_chain_id":"CN360400ca5968684cbdff0aaa03c63fef074235",
    "bci_time":"2021-07-25 00:00:00",
    "bci_timestamp":"0XBB998CBFF900C90CDF98164FAC1ED26F133B09C1FA39D47EA37DCBF4
    52FAFC4A",
    "src_user_id":"src_user_0x001",
    "target_user_id":"tar_user_0a009",
    "success": false,
    "result":"unknown exceptions."
    "error_code":500,
    "src_chain_result":"query fail",
    "target_chain_result":"Fail"
  }
```


6.5.3.3 跨链交易 ID 生成规则

跨链交易ID由请求放生成，如未生成则由服务端统一生成，为防止ID碰撞，现ID生成规则如下：

（1） 拼接ID原值：由appid，userid，日期以及6位随机数组成，中间使用下滑线进行拼接生成S1，如：appId_userId_20220102_765432；

（2） 对S1进行哈希计算生成D1,D1则为跨链交易ID。

7 统一应用跨链协议

当第三方应用程序需要通过区块链跨链交互流程进行跨链时,可调用遵循统一应用跨链协议的SDK，与区块链网络其中一个业务子链建立通讯，进行区块链跨链事务处理。

7.1 协议流程

7.1.1 基本流程

统一应用跨链协议主要帮助第三方应用程序接入区块链跨链交互框架，其定义的总体流程如图6所示。

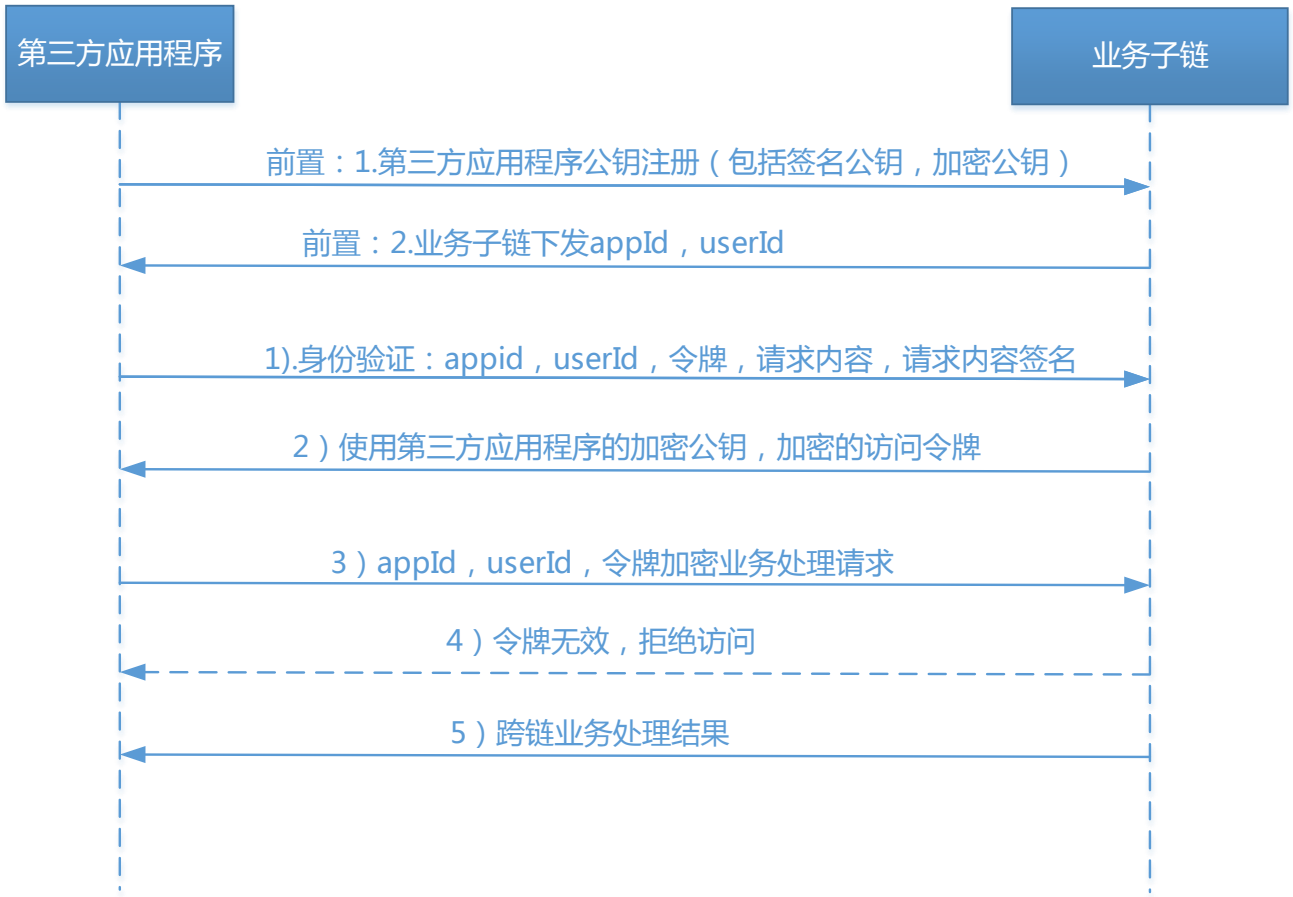

图 6 区块链统一应用跨链协议总体流程

如图所示，协议总体流程包含如下步骤：

前置步骤1：第三方应用程序公钥注册。

第三方应用程序公钥注册应通过以下方式进行，业务子链可选择其中一种进行实现。注册后该公钥对应的第三方应用程序将成为合法应用。

线下方式：第三方应用程序生成两对公私钥，同时将公钥发给业务子链的管理员，让其手动添加至业务子链中进行注册，其操作全部线下完成。

线上方式：第三方应用程序生成两对公私钥，同时将公钥线上注册至业务子链。

前置步骤2：业务子链下发appId、userId、推送业务子链公钥。

业务子链的管理员或管理应用为第三方应用程序指定一个唯一的应用标识，即appId。并推送当前业务子链签名公钥给第三方应用程序。以上操作可选择线上或线下进行，业务子链签名公钥将成为第三方应用程序验证消息来源是否为正确业务子链的依据。

步骤1：申请令牌。

第三方应用程序向业务子链申请令牌，如果申请成功，使用该令牌即可获得业务请求访问权限。申请令牌时消息体必须包含appId、userId、随机数及针对该随机数的签名，后者用于业务子链验证该应用程序的身份。

步骤2：发放令牌。

业务子链在验证完第三方应用程序的身份，并成功确认该应用程序合法后，将发放令牌，令牌有失效时间，确保仅在一定时间内有效。令牌发放时将使用第三方应用程序公钥进行加密，以确保在网络传输中不被盗取。同时使用节点私钥进行签名，以保障在网络传输中的令牌完整性、来源的真实性。

步骤3：业务请求。

第三方应用程序向业务子链发送业务请求，在发送请求的过程中对数据进行签名，保护消息的完整性、抗依赖性以及来源的真实性，同时使用令牌进行加密，确保业务请求数据的机密性。

步骤4：业务响应。

业务子链在验证应用程序业务请求的签名及令牌的有效性后，如果验证成功则转发给区块链业务服务进行处理，响应业务处理结果；如果验证失败，则拒绝该应用程序访问。

7.1.2 第三方应用程序安全要求

除遵循流程外，第三方应用程序在安全性上必须注意：

1）第三方应用程序应保障第三方应用程序私钥存储的安全性，将其加密存储在智能密码钥匙等密码模块上。

2）第三方应用程序应定期申请令牌，以避免令牌超时失效。

3）第三方应用程序在获得业务子链发放的令牌时，应使用业务子链签名公钥验证消息是否来源于该业务子链。

7.1.3 业务子链安全要求

除遵循流程外，业务子链在安全性上必须注意：

1）业务子链应保障业务子链私钥存储的安全性。将其加密存储在符合国家密码管理要求规定的密码模块上。

2）业务子链的令牌具备超时失效机制，以避免令牌被非法使用。

7.2 应用申请授权

7.2.1 前置处理

前置处理可在线下进行，为协议运行提供环境准备工作。包括第三方应用程序公钥注册、下发appId及推送业务子链签名公钥等步骤。

1）第三方应用程序公钥注册

第三方应用程序需生成一对公私钥，同时将公钥发给业务子链的管理员，让其手动添加至业务子链中进行注册。以上操作均在线下进行，注册后该公钥对应的第三方应用程序将成为合法应用。

详细步骤如下：

1、生成第三方应用程序密钥对。第三方应用程序生成一对公私钥，用于区块链系统识别其身份。该操作必须保障安全性，通常可借助硬件密码模块进行生成。

2、申请注册公钥。第三方应用程序在线下申请注册第三方应用程序公钥。

3、注册公钥。业务子链将第三方应用程序公钥注册到区块链系统中。

4、分配appId。业务子链为该应用程序生成一个全局唯一的标识符，来唯一指定该appId。

5、分配链上权限。业务子链为该应用程序分配访问区块链的权限，供业务请求时判断该应用程序是否具备相应权限。

2）下发appId及推送业务子链签名公钥

业务子链的管理员为第三方应用程序指定一个唯一的应用标识，即appId。并推送当前业务子链签名公钥给第三方应用程序。以上操作均在线下进行，业务子链签名公钥将成为第三方应用程序验证消息来源是否为正确业务子链的依据。

详细步骤如下：

1、分发appId、推送业务子链签名公钥：管理员在线下将appId和业务子链签名公钥发送给已注册的第三方应用程序。appId用来标识该应用程序，业务子链签名公钥用于应用程序验证业务子链传输消息的完整性、消息来源的合法性。

2、保存appId：第三方应用程序保存该appId，作为自身的唯一性标识。

3、保存业务子链签名公钥：第三方应用程序保存该业务子链签名公钥，用于识别业务子链传输消息的完整性及来源的真实性。

7.2.2 申请令牌

在接入区块链系统进行业务请求之前，第三方应用程序必须向业务子链申请令牌以获得授权。申请令牌时消息体必须包含appId、随机数及针对该随机数的签名。业务子链在收到申请令牌请求后，须要验证该应用程序的身份。

详细步骤如下：

1、生成随机数：第三方应用程序采用随机数生成算法生成一个16字节（128bit）的随机数，并将其转化为长度为32的十六进制字符串。

2、使用随机数进行签名：第三方应用程序利用第三方应用程序私钥，采用SM2算法对该随机数进行数字签名。

3、申请令牌：第三方应用程序通过网络向业务子链发送申请令牌请求。

4、验证签名：业务子链得到带签名的消息原文后，利用第三方应用程序公钥，采用SM2算法验证签名。如果验证签名成功，将进入发放令牌流程。

7.2.2.1 请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/restfulapi/token/get_token
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

7.2.2.2 请求消息格式

请求消息的参数说明如下：

参数	参数类型	说明
app_id	字符串	第三方应用程序应用标识。
user_id	字符串	用户标识。
data	字符串	一段随机数，表现形式为长度为32的十六进制字符串。
signature	字符串	签名值。 计算步骤： (1)先对随机数data进行哈希sm3计算H1 (2) 第三方应用程序用私钥对H1进行签名S1 (3)将签名值S1转成十六进制字符串

示例：
http://127.0.0.1:7055/restfulapi/token/get_token
{
"app_id":"1qaz2wsx3edc4rfv5tgb6yhn7ujm8ik9ol0p ",
"user_id":"zhangsan",
"data":"304402207d4c88cc6e021a0f6897bdf6",
"signature":"304402207d4c88cc6e021a0f6897bdf624c31c43dcb183110075fb1feacecfc74210ef0b02200575081042b67c1e16f0d20beace0029ed7f109eaa6e419948081ea719dd2996"
}		

7.2.3 发放令牌

业务子链在确认第三方应用程序的身份合法后，将发放令牌。令牌发放时须要使用第三方应用程序公钥进行加密，以确保在网络传输中不被盗取。同时使用节点私钥进行签名，以保障在网络传输中的令牌完整性、来源的真实性。

详细步骤如下：

1、发放令牌：业务子链在确认第三方应用程序的身份合法后，将计算出一个令牌，该令牌在一定时间周期内有效。然后利用第三方应用程序公钥，采用SM2算法对该令牌进行加密，得到令牌密文。同

时利用节点私钥，采用SM2算法对该令牌密文进行签名。最后，将令牌密文和签名一并发给申请令牌的第三方应用程序。

2、验签：第三方应用程序在获得响应消息后，利用业务子链签名公钥，采用SM2算法验证签名。如果签名验证失败，则认为申请令牌失败。

3、解密令牌：第三方应用程序验证签名成功后，利用持有的第三方应用程序私钥，采用SM2算法解密出令牌。解密出的令牌可用于之后的业务请求密码计算。

7.2.3.1 响应消息格式

响应消息的参数说明如下：

参数	参数类型	说明
access_token	字符串	访问令牌。 计算步骤： (1)生成 sm4 对称密钥； (2) 业务子链用第三方应用程序公钥对 sm4 密钥进行加密 e1； (3)加密结果 e1 转成十六进制字符串。
signature	字符串	签名值。 计算步骤： (1)先对access_token做哈希sm3计算H1； (2) 业务子链用私钥对H1进行签名S1； (3)对签名值S1转成十六进制字符串。
success	字符串	是否成功。
message	字符串	提示消息。

示例：

成功：


```
{
  "access_token":
  "04d4e1b88257293e84d19587e8cb6158570acc39ec85c2820f05ac35dadede50401b353a4759
  6cc704c68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef5121ae47f107
  7efb56347e40b77b91875eddb481eb76b5ed779da55fbd163d91acdca3863f4634aab1f39b36b
  c22f2df35ca51bb4675edba1525b",
  "signature":
  "304402207d4c88cc6e021a0f6897bdf624c31c43dcb183110075fb1feacecfc74210ef0b0220
  0575081042b67c1e16f0d20beace0029ed7f109eaa6e419948081ea719dd2996",
  "success": true,
  "message": ""
}
```


失败：


```
{
  "access_token": "",
  "signature": "",
```

```
    "success": false,
    "message":"Execute smart contract fail: unknown reason"
}
```

7.3 业务请求

7.3.1 通用业务请求

通用业务请求泛指需要区块链处理的业务，该请求由第三方应用程序发起，由业务子链完成业务响应。在发送请求的过程中第三方应用程序使用系统发放的令牌进行加密，确保业务请求数据的机密性。在响应的过程中业务子链须要验证令牌的有效性，以确保消息来源的真实性。

详细步骤如下：

1、使用令牌进行SM4加密：第三方应用程序利用令牌作为密钥，采用SM4算法对带数字签名的区块链业务请求内容进行对称加密。

2、上链请求：第三方应用程序通过网络向业务子链发送区块链业务请求。

3、SM4解密消息：业务子链收到消息后，利用令牌，采用SM4算法对消息进行解密。如果解密失败，则向第三方应用程序发出业务响应，提示令牌可能非法。

4、执行区块链业务：业务子链执行区块链业务，得到业务执行结果。

5、业务响应：业务子链将业务执行结果返回给第三方应用程序。

7.3.2 上链请求

上链请求主要处理区块链上链存储类业务，该请求由第三方应用程序发起，由业务子链完成业务响应。其流程遵循通用业务请求处理流程。

详细步骤如下：

1、使用令牌进行SM4加密：第三方应用程序利用令牌作为密钥，采用SM4算法对上链请求内容进行对称加密。

2、上链请求：第三方应用程序通过网络向业务子链发送上链请求。

3、SM4解密消息：业务子链收到消息后，利用令牌，采用SM4算法对消息进行解密。如果解密失败，则向第三方应用程序发出业务响应，提示令牌可能非法。

4、执行上链业务：业务子链执行上链业务，得到上链结果。

5、业务响应：业务子链将上链结果返回给第三方应用程序。

7.3.2.1 请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/restfulapi/smart_contract/invoke
访问方法	POST

字符集编码	UTF-8
参数格式	application/json

7.3.2.2 请求消息格

请求消息的参数说明如下：

参数	参数类型	说明
app_id	字符串	**第三方应用程序**应用标识。
user_id	字符串	用户标识。
encrypt_data	字符串	加密数据。 计算步骤： (1) **第三方应用程序**使用对称密钥，对明文数据data进行加密e1； (2)对加密值e1转成十六进制字符串。
encrypt_data对应的明文数据（Json格式）		
group_id	字符串	群组名
sc_name	字符串	智能合约名称
sc_args	字符串	智能合约入参

示例：

```
http://127.0.0.1:7055/restfulapi/smart_contract/invoke
  {
 "app_id":"1qaz2wsx3edc4rfv5tgb6yhn7ujm8ik9ol0p",
 "user_id":"zhangsan",
 "encrypt_data":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef5
  121ae47f1077efb56347e40b77b91875"
}
```

其中 encrypt_data 对应的明文数据为

```
  {
      "group_id":"test",
      "sc_name":"testsc",
      "sc_args":{
      "args":["move", "a", "b", "10"]
      }
}
```

7.3.2.3 响应消息格式

响应消息的参数说明如下：

参数	参数类型	说明
success	布尔型	是否成功。
message	字符串	提示消息。
encrypt_result	字符串	加密返回消息。 计算步骤： (1) **业务子链**使用对称密钥，对返回明文消息data进行加密，得到加密值e1； (2)将加密值 e1 转成十六进制字符串。
encrypt_result 对应的明文数据（Json 格式）		
data	字符串	上链处理结果

示例：

成功：


```
{
  "success": true,
  "message":"invoke success.",
"encrypt_result":"68e1f079c0417597884429348 0eafd9df2c7890724a0a9b1235676e3ef512
  1ae47f1077efb56347e40b77b91875"
}
```


其中 data 对应的明文数据为

0XB09C1FA39D478CBFF900F45F98164FAC1ED26F133BFC4AE2FAC90CDA37DCBB99

失败：


```
{
  "success": false,
  "message":"org.bcia.julongchain.commons.exception.SmartContractException:[Sma
  rtContract]Execute smart contract fail: 0012_unknown function: move"
}
```


7.3.3 查链请求

查链请求主要处理区块链查询类业务，该请求由第三方应用程序发起，由业务子链完成业务响应。其流程遵循通用业务请求处理流程。

详细步骤如下：

1、使用令牌进行SM4加密：第三方应用程序利用令牌作为密钥，采用SM4算法对查链请求内容进行对称加密。

2、查链请求：第三方应用程序通过网络向业务子链发送查链请求。

3、SM4解密消息：业务子链收到消息后，利用令牌，采用SM4算法对消息进行解密。如果解密失败，则向第三方应用程序发出业务响应，提示令牌可能非法。

4、执行查链业务：业务子链执行查链业务，得到查链结果。

5、业务响应：业务子链将查链结果返回给第三方应用程序。

7.3.3.1 请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/restfulapi/smart_contract/query
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

7.3.3.2 请求消息格式

请求消息的参数说明如下：

参数	参数类型	说明
app_id	字符串	**第三方应用程序**应用标识
user_id	字符串	用户标识
encrypt_data	字符串	加密数据。 计算步骤： (1) **第三方应用程序**使用对称密钥，对明文数据data进行加密e1，得到加密值e1； (2)对加密值e1转成十六进制字符串。
encrypt_data对应的明文数据（Json格式）		
group_id	字符串	群组名
sc_name	字符串	智能合约名称
sc_args	字符串	智能合约入参
示例： http://127.0.0.1:7055/restfulapi/smart_contract/query { "app_id":"1qaz2wsx3edc4rfv5tgb6yhn7ujm8ik9ol0p", "user_id":"zhangsan", "encrypt_data":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef5		

```
        121ae47f1077efb56347e40b77b91875"
}
其中 encrypt_data 对应的明文数据为
      {
          "group_id":"test",
          "sc_name":"testsc",
          "sc_args":{
          "args":["query", "a"]
          }
}
```

7.3.3.3 响应消息格式

响应消息的参数说明如下：

参数	参数类型	说明
success	布尔型	是否成功
message	字符串	提示消息
encrypt_result	字符串	加密返回消息 计算步骤： (1) 业务子链使用对称密钥，对返回明文消息data进行加密，得到加密值e1； (2)将加密值 e1 转成十六进制字符串。
encrypt_result 对应的明文数据（Json 格式）		
data	字符串	查询结果

示例：
成功：
{
"success": true,
"message":"query success.",
"encrypt_data":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef5121a e47f1077efb56347e40b77b91875"
}
 	
其中 data 对应的明文数据为
{key:luis,value:28}
 	
失败：
{		

```
    "success": false,
    "message":"org.bcia.julongchain.commons.exception.SmartContractException:[Sma
    rtContract]Execute smart contract fail: 0012_unknown function: query"
}
```


7.3.4 应用跨链上链请求

应用跨链上链请求主要处理区块链跨链上链存储类业务，该请求由第三方应用程序发起，由业务子链完成业务响应。业务子链收到应用跨链上链请求后，首先完成自身的上链操作，然后将通过中继主链发起跨链上链操作，其流程遵循通用业务请求处理流程。

详细步骤如下：

1、使用令牌进行SM4加密：第三方应用程序利用令牌作为密钥，采用SM4算法对上链请求内容进行对称加密。

2、上链请求：第三方应用程序通过网络向业务子链发送上链请求。

3、SM4解密消息：业务子链收到消息后，利用令牌，采用SM4算法对消息进行解密。如果解密失败，则向第三方应用程序发出业务响应，提示令牌可能非法。

4、执行跨链上链业务：业务子链执行自身的上链业务，得到源链上链结果；并调用中继主链完成跨链上链业务，得到目标链结果。

5、业务响应：业务子链将跨链上链结果返回给第三方应用程序。

7.3.4.1 请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/restfulapi/cross_chain/invoke
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

7.3.4.2 请求消息格式

请求消息的参数说明如下：

参数	参数类型	说明
app_id	字符串	第三方应用程序应用标识。
user_id	字符串	用户标识。
encrypt_data	字符串	加密数据。 计算步骤： (1) 第三方应用程序使

		用对称密钥，对明文数据data进行加密e1； (2)对加密值e1转成十六进制字符串。
encrypt_data对应的明文数据（Json格式）		
src_chain_id	字符串	源链ID。
src_group_id	字符串	源链组群ID。
target_chain_id	字符串	目标链ID。
target_group_id	字符串	目标链组群ID。
src_sc_name	字符串	源链智能合约名称。
target_sc_name	字符串	目标链智能合约名称。
src_sc_args	字符串	源链智能合约入参。
target_sc_args	字符串	目标链智能合约入参。

示例：

http://127.0.0.1:7055/restfulapi/cross_chain/invoke


```
{
 "app_id":"1qaz2wsx3edc4rfv5tgb6yhn7ujm8ik9ol0p",
 "user_id":"zhangsan",
 "encrypt_data":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef5
  121ae47f1077efb56347e40b77b91875"
}
```


其中 encrypt_data 对应的明文数据为


```
{
  "src_chain_id":"65b71efd4f01fcf01fc0e8baeed44919",
  "src_group_id":"mygroup",
  "target_chain_id":"93073bf541f4c9a35edbeda98d3b2f06",
  "target_group_id":"mychannel",
  "src_sc_name":"mycc1",
  "target_sc_name":"mycc2",
  "src_sc_args":{"args":["move", "a", "b", "10"]},
  "target_sc_args":{"args":["move", "b", "a", "10"]}
}
```


7.3.4.3 响应消息格式

响应消息的参数说明如下：

参数	参数类型	说明
success	布尔型	是否成功。
message	字符串	提示消息。
encrypt_result	字符串	加密返回消息。 计算步骤：

		(1) 业务子链使用对称密钥，对返回明文消息data进行加密，得到加密值e1； (2)将加密值 e1 转成十六进制字符串。
encrypt_result 对应的明文数据（Json 格式）		
src_chain	字符串	上链处理结果。
target_chain	字符串	目标链上链结果。

示例：

成功：


```
{
  "success": true,
  "message":"invoke success.",
"encrypt_result":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef512
  1ae47f1077efb56347e40b77b91875"
}
```


其中 encrypt_result 对应的明文数据示例：


```
      {
      "src_chain":{
        "success":true,
        "message":"invoke success",
        "txid":"
 0x0B30D0525D0F9AFF8195DFC2CFD501559A9A9ED2A9258C9005F5074AFE8F5F13"
      },
      "target_chain":{
        "success":true,
        "message":"invoke success",
        "txid":"
 0XF981641FABB00C1ED26F133B0A90C9EA37DCBFDFA9C47452FAFD3C4998CBFF9C"
      }
}
```


失败：


```
{
  "success": false,
  "message":"org.bcia.julongchain.commons.exception.SmartContractException:[Sma
  rtContract]Execute smart contract fail: 0012_unknown function: move"
}
```


7.3.5 应用跨链查询请求

应用跨链查询请求主要处理区块链跨链查询类业务，该请求由第三方应用程序发起，由业务子链完成业务响应。业务子链收到应用跨链上链请求后，通过中继主链发起跨链查询操作，其流程遵循通用业务请求处理流程。

详细步骤如下：

1、使用令牌进行SM4加密：第三方应用程序利用令牌作为密钥，采用SM4算法对上链请求内容进行对称加密。

2、上链请求：第三方应用程序通过网络向业务子链发送上链请求。

3、SM4解密消息：业务子链收到消息后，利用令牌，采用SM4算法对消息进行解密。如果解密失败，则向第三方应用程序发出业务响应，提示令牌可能非法。

4、执行跨链上链业务：业务子链调用中继主链完成跨链查询业务，得到目标链结果。

5、业务响应：业务子链将跨链上链结果返回给第三方应用程序。

7.3.5.1 请求方式

接口请求方式如下：

属性	值
访问URI	http://IP:port/restfulapi/cross_chain/query
访问方法	POST
字符集编码	UTF-8
参数格式	application/json

7.3.5.2 请求消息格式

请求消息的参数说明如下：

参数	参数类型	说明
app_id	字符串	第三方应用程序应用标识。
user_id	字符串	用户标识。
encrypt_data	字符串	加密数据。 计算步骤： (1) 第三方应用程序使用对称密钥,对明文数据data进行加密e1； (2)对加密值e1转成十六进制字符串。
encrypt_data对应的明文数据（Json格式）		
src_chain_id	字符串	源链ID。

src_group_id	字符串	源链组群ID。
target_chain_id	字符串	目标链ID。
target_group_id	字符串	目标链组群ID。
target_sc_name	字符串	目标链智能合约名称。
target_sc_args	字符串	目标链智能合约入参。

示例：

```
http://127.0.0.1:7055/restfulapi/cross_chain/query
  {
 "app_id":"1qaz2wsx3edc4rfv5tgb6yhn7ujm8ik9ol0p",
 "user_id":"zhangsan",
 "encrypt_data":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef5
   121ae47f1077efb56347e40b77b91875"
}
```

其中 encrypt_data 对应的明文数据为

```
  {
    "src_chain_id":"65b71efd4f01fcf01fc0e8baeed44919",
    "src_group_id":"mygroup",
    "target_chain_id":"93073bf541f4c9a35edbeda98d3b2f06",
    "target_group_id":"mychannel",
    "target_sc_name":"mycc2",
    "target_sc_args":{"args":["query", "a"]}
}
```

7.3.5.3 响应消息格式

响应消息的参数说明如下：

参数	参数类型	说明
success	布尔型	是否成功。
message	字符串	提示消息。
encrypt_result	字符串	加密返回消息。 计算步骤： (1) 业务子链使用对称密钥，对返回明文消息data进行加密，得到加密值e1； (2)将加密值 e1 转成十六进制字符串。
encrypt_result 对应的明文数据（Json 格式）		
data	字符串	目标链查询结果。
示例：		

成功：


```
{
  "success": true,
  "message":"invoke success.",
"encrypt_result":"68e1f079c04175978844293480eafd9df2c7890724a0a9b1235676e3ef512
  1ae47f1077efb56347e40b77b91875"
}
```


其中 data 对应的明文数据为

{key:luis,value:28}

失败：


```
{
  "success": false,
  "message":"org.bcia.julongchain.commons.exception.SmartContractException:[Sma
  rtContract]Execute smart contract fail: 0012_unknown function: move"
}
```

附 录 A
（基于 PCI-E 密码卡实现的统一应用跨链协议）

当第三方应用程序需要通过区块链跨链交互流程进行跨链时，可实现统一应用跨链协议，以下是第三方应用程序和业务子链统一采用PCI-E密码卡来完成密钥管理和密码计算形式的典型实例。

统一应用跨链协议主要帮助第三方应用程序接入区块链跨链交互框架，其定义的总体流程如图A-1示。

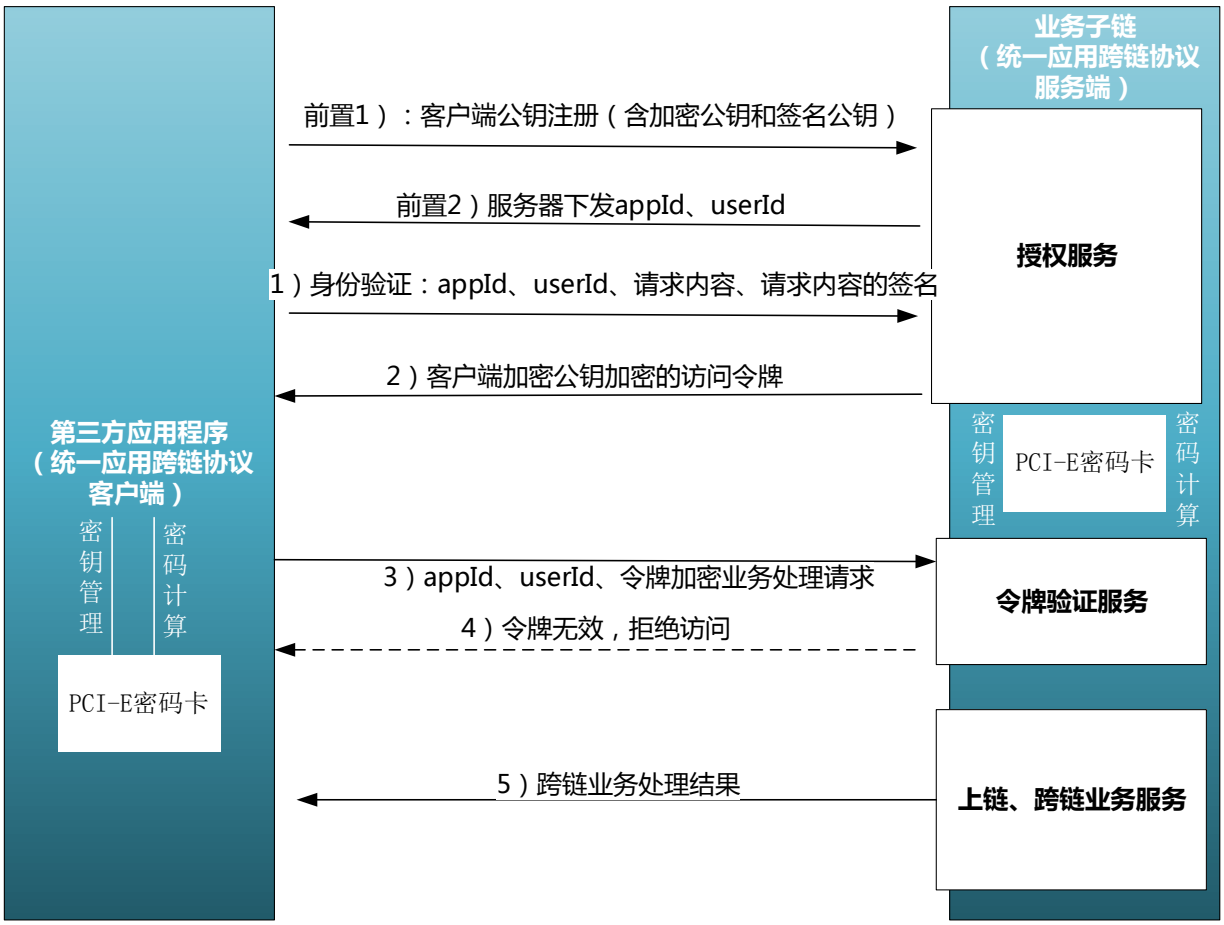

图 A- 1 使用密码卡计算的区块链统一应用跨链协议总体流程

如图所示，协议总体流程包含如下步骤：

前置步骤1：第三方应用程序公钥注册。

第三方应用程序公钥注册可采用线下或线上两种方式进行，业务子链可选择其中一种进行实现。

线下方式：第三方应用程序使用密码卡生成两对公私钥，同时导出公钥，并将公钥发给业务子链的管理员，让其手动添加至业务子链中进行注册。以上操作均在线下进行，注册后该公钥对应的第三方应用程序将成为合法应用。

线上方式：第三方应用程序使用密码卡生成两对公私钥，并导出公钥，同时将公钥线上注册至业务子链，注册后该公钥对应的第三方应用程序将成为合法应用。

前置步骤2：服务器下发appId、userId、推送业务子链公钥。

业务子链的管理员或管理应用为第三方应用程序指定一个唯一的应用标识，即appId。并导出业务子链公钥，推送当前业务子链签名公钥给第三方应用程序。以上操作可选择线上或线下进行，业务子链签名公钥将成为第三方应用程序验证消息来源是否为正确业务子链的依据。

步骤1：申请令牌。

第三方应用程序向业务子链申请令牌，如果申请成功，使用该令牌即可获得业务请求访问权限。申请令牌时消息体必须包含appId、userId、随机数及针对该随机数的签名，后者用于业务子链验证该应用程序的身份。

步骤2：发放令牌。

业务子链在验证完第三方应用程序的身份，并成功确认该应用程序合法后，将发放令牌，令牌有失效时间，确保仅在一定时间内有效。令牌发放时将使用第三方应用程序公钥进行加密，以确保在网络传输中不被盗取。同时使用节点私钥进行签名，以保障在网络传输中的令牌完整性、来源的真实性。

步骤3：业务请求。

第三方应用程序向业务子链发送业务请求，在发送请求的过程中对数据进行签名，保护消息的完整性、抗依赖性以及来源的真实性，同时使用令牌进行加密，确保业务请求数据的机密性。

步骤4：业务响应。

业务子链在验证应用程序业务请求的签名及令牌的有效性后，如果验证成功则转发给区块链业务服务进行处理，响应业务处理结果；如果验证失败，则拒绝该应用程序访问。

附 录 B
（跨链审计监督）

通过跨链监管系统（CAS）、节点安全监测系统（CNS）对接入的业务链进行安全监管，实现不同链之间的跨链管理和业务监管。

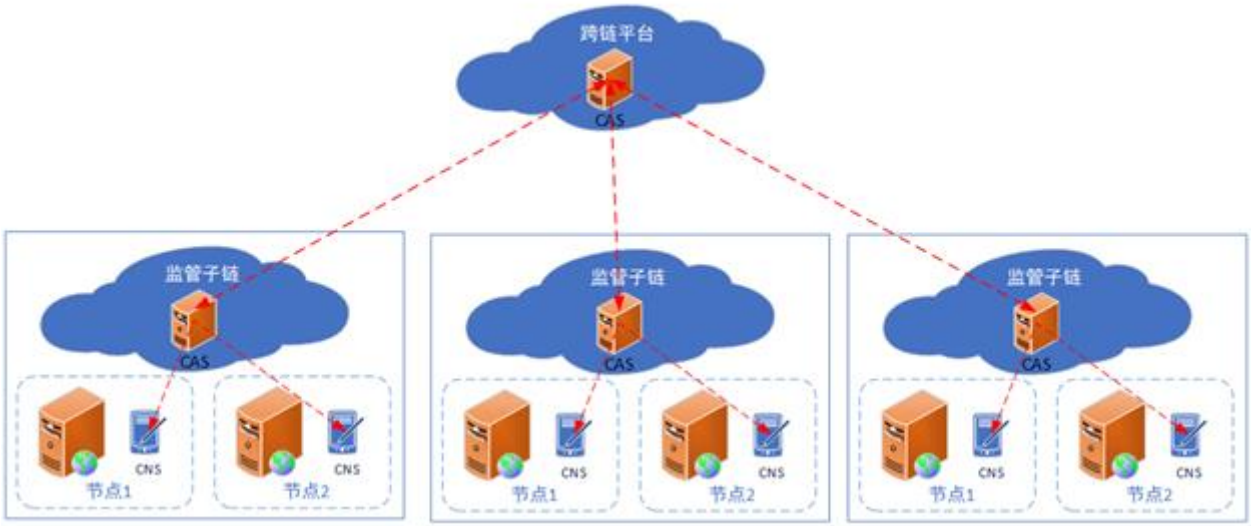

图 B-1 审计监管部署模式

本接口协议适用于一般业务链与安全监测系统数据对接规范，指导一般业务链开发对应规范接口，提供规范数据格式。

B.1 安全监测

一般业务链应通过CNS对自身进行安全监测，执行政务基础链下发的安全管理策略。

一般业务链应按照如下要求进行日志记录，日志路径可参考以下设置：

a)节点日志路径：/opt/hbCosService/log/node_*.log

b)共识日志路径：/opt/hbCosService/log/consensuses_*.log

c)区块链管理平台（BaaS）日志路径：/opt/hbCosService/log/admin_*.log

d)系统管理操作日志路径：/opt/hbCosService/log/system_*.log

e)其他日志路径：/opt/hbCosService/log/other_*.log

注：*表示日志时间（每天产生日志都有对应的日期。如：2022-07-29，2022-07-30等）：格式为yyyy-MM-dd(年-月-日)。

一般业务链的日志记录应包含以下参数。

表 B-1 安全监测日志记录关键信息列表

参数名	类型	是否必须	描述
id	String	是	日志编号
content	String	是	日志内容
type	String	是	日志类型
createDate	String	是	日志创建时间
timeStamp	Long	是	时间戳

B.1.1 节点日志

一般业务链应遵循下表格式进行节点日志记录。

表 B-2 一版业务链节点日志示例

示例：

{"timeStamp":1660141196342,"id":"446959bc5e594bd396d04370c3a37293","type":"nodeUpChain","content":"接收

交叉链信息content:2022-8-10

22:19,sign:MEUCIQDqxyvH50BNj3HpirzS6DKS5calE2O48rSjNfkLE3+WmQIgPqqS+H6F8P609gAzFjs3u4rvf8gGSSSb6ml2UGBP

9R0=,targetChainId:ffab7f107351439aa7ad01aa86c8c121,type:invoke,sourceAppId:11111111,targetAppId:222222

222,encryption:false","createDate":"2022-08-10 22:19:56"}

{"timeStamp":1660141200000,"id":"2f8de7ee9c2e4fc3aeac7f8fcfef1587","type":"nodeUpChain","content":"共识

数据","createDate":"2022-08-10 22:20:00"}

{"timeStamp":1660141200349,"id":"c0f275f45f7a48b6b98bb078cf07c045","type":"nodeUpChainError","content":

"上链成功","createDate":"2022-08-10 22:20:00"}

B.1.2 共识日志

一般业务链应遵循下表格式进行共识日志记录。

表 B-3 一般业务链共识日志示例

示例：

{"timeStamp":1660194011766,"id":"6a7987fe2ed442ffab282324c6764412","type":"consensusesHash","content":"

共识 hash 成功","createDate":"2022-08-11 13:00:11"}

{"timeStamp":1660194011820,"id":"e0401a379dab467ebd8cbc7157d9ca1f","type":"consensusesData","content":"

接收共识数据",",createDate":"2022-08-11 13:00:11"}

{"timeStamp":1660194012297,"id":"71d380b049b148baba95db89b253296a","type":"consensusesData","content":"开始共识数据到ip:10.10.1.85,名称：85节点,结果：成功","createDate":"2022-08-11 13:00:12"}

B.1.3 BaaS 日志

一般业务链应遵循下表格式进行BaaS日志记录。

表 B-4 一般业务链 BaaS 日志示例

示例：

{"timeStamp":1661224862599,"id":"104842280a03464c85d884621dde9583","type":"addressRegionTree","content":"获取地址树形结构","createDate":"2022-08-23 11:21:02"}

{"timeStamp":1661224862609,"id":"b9e079a2cd0949b09e4b81077c946da9","type":"queryBlockChainUnit","content":"查询区块链单位信息","createDate":"2022-08-23 11:21:02"}

{"timeStamp":1661224862749,"id":"91a39e18136347bf94c64cc2202e5da6","type":"addressRegion","content":"获取地址信息","createDate":"2022-08-23 11:21:02"}

B.1.4 系统管理操作日志

一般业务链应遵循下表格式进行系统管理操作日志记录。

表 B-5 一般业务链系统管理操作日志示例

示例：

{"timeStamp":1660806505476,"id":"5ec0cd02a5b8408fb5f4cc295fd350d0","type":"login","content":"用户 :admin 在登录地址：10.10.1.170 登录地点：内网 IP 操作系统：Windows 10 使用浏览器：Chrome 10 消息为：登录成功，状态成功，访问时间：null","createDate":"2022-08-18 15:08:25"}

{"timeStamp":1660806536366,"id":"446215b13327495f934b78fbd92c4820","type":"logout","content":"用户：admin在登录地址：10.10.1.170登录地点：内网IP操作系统：Windows 10使用浏览器：Chrome 10消息为：退出成功，状态：成功，访问时间：null","createDate":"2022-08-18 15:08:56"}

B.1.5 其他日志

预留（用于扩展其他类型日志）。