计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(2): 443−457, 2025

一种 DeFi 价格操纵攻击在线防御机制

林炼升[1] 郑焕钦[1] 苏 申[1,2] 雷 凯[3] 陈晓丰[4,5] 田志宏[1]

[1](广州大学网络空间安全学院 广州 510006)

[2](数字化学习技术集成与应用教育部工程研究中心(国家开放大学) 北京 100039)

[3](深圳市内容中心网络与区块链重点实验室(北京大学) 广东深圳 518055)

[4](区块链与数据安全全国重点实验室(浙江大学) 杭州 310027)

[5](杭州趣链科技有限公司标准化与课题管理部 杭州 310051)

(843674016@qq.com)

An On-Chain Mechanism Against DeFi Price Manipulation Attacks

Lin Liansheng[1], Zheng Huanqin[1], Su Shen[1,2], Lei Kai[3], Chen Xiaofeng[4,5], and Tian Zhihong[1]

[1](*Cybersecurity Institute of Advanced Technology*, *Guangzhou University*, *Guangzhou* 510006)

[2](*Engineering Research Center of Integration and Application of Digital Learning Technology* (*The Open University of China*), *Ministry of Education*, *Beijing* 100039)

[3](*Shenzhen Key Laboratory of Information-centric Networking & Blockchain* (*Peking University*), *Shenzhen*, *Guangdong* 518055)

[4](*State Key Laboratory of Blockchain and Data Security* (*Zhejiang University*), *Hangzhou* 310027)

[5](*Standardization and Project Management Department*, *Hangzhou Qulian Technology Co., Ltd.*, *Hangzhou* 310051)

Abstract Price manipulation attacks manipulate the on-chain prices of decentralized financial (DeFi) projects by altering the digital asset stock, thereby attacking their liquidation mechanisms to achieve improper profits. Nowadays, price manipulation attacks have emerged as the most significant security threats to the current decentralized financial ecosystem. To defend from the price manipulation attacks, the oracle obtains the exchange prices from the real world, which are difficult to manipulate. However, the maintenance expense of the oracle is very high due to frequent on-chain data update, making it challenging to meet industrial demand. To address these issues, we propose a defense mechanism against price manipulation attacks. This mechanism utilizes off-chain prices to guide the identification of on-chain price manipulation behaviors and intercepts price manipulative transactions through a contract proxy. The mechanism reduces the frequency of price submissions and the cost of updating off-chain data on-chain through low-frequency price feeding. This compromise aims to balance the cost of defense against price manipulation attacks with the precision of identification. Based on the experimental findings, we have conclusively demonstrated that our innovative method substantially diminishes the overall maintenance cost by over 30%, concurrently achieving an outstanding success rate of 97.5% in effectively safeguarding against price manipulation attacks.

收稿日期: 2023-04-03; **修回日期**: 2024-01-22

基金项目: 广东省重点研发计划项目(2020B0101090003); 国家重点研发计划项目(2022YFB2702305); 国家自然科学基金项目(61902083, 62172115, 61976064); 数字化学习技术集成与应用教育部工程研究中心创新基金项目(1221045); 广东省高校创新团队项目(2020KCXTD007); 广州市高校创新团队项目(202032854)

This work was supported by the Key Research and Development Program of Guangdong Province (2020B0101090003), the National Key Research and Development Program of China (2022YFB2702305), the National Natural Science Foundation of China (61902083, 62172115, 61976064), the Innovation Foundation of Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education (1221045), the Higher Education Innovation Group of Guangdong Province (2020KCXTD007), and the Higher Education Innovation Group of Guangzhou City (202032854).

通信作者: 苏申(sushen@gzhu.edu.cn)

Key words blockchain; smart contract; decentralized financial; price manipulation attacks; oracle price feeding

摘 要 价格操控攻击通过改变去中心化金融项目的数字资产存量操控资产链上价格,从而攻击其清算机制以实现不当牟利,是目前去中心化金融生态最主要的安全威胁之一.目前主流的安全防御手段是通过预言机获取不易被操控的链下价格,但频繁将链下数据更新上链会导致预言机的维护成本高昂,因而无法满足工业界需求.为解决上述问题,提出一种针对价格操控攻击的防御机制,通过链下价格来指导链上价格操控行为的识别,以合约代理的形式实现对价格操纵交易的拦截,并通过低频的价格获取降低交易提交的频率和链下数据更新上链的成本,进而实现价格操控攻击的防御成本和识别精度之间的折中.实验表明,该方法在降低预言机30%以上运维成本的前提下,对价格操纵攻击的防御率达到97.5%.

关键词 区块链;智能合约;去中心化金融;价格操纵攻击;预言机喂价

中图法分类号 TP309.2

DOI: 10.7544/issn1000-1239.202330291 **CSTR:** 32373.14.issn1000-1239.202330291

区块链技术是一种去中心化、不可篡改的分布式账本技术,具有高度的安全性和透明性,并且区块链公开的账本和共识机制保证了交易的可信度和完整性[1],因此区块链技术天然带有金融属性,而智能合约[2]的出现则使得金融业务在区块链上的实现成为可能.智能合约是区块链最前沿的发展方向,它通过将代码部署在区块链中,以去中心化且自动执行的方式实现多方之间信任和价值交换,进一步拓展了区块链技术的应用场景[3-4].

去中心化金融[5](decentralized finance, DeFi)正是基于智能合约实现的自动化的去中心化金融业务体系,是当前区块链技术最重要的应用场景之一.DeFi中的金融业务,包括资产兑换、借贷等业务的正常开展都依赖于资产价格的正常获取,而DeFi中资产价格数据主要来源于去中心化交易所[6](decentralized exchanges, DEX).与现实金融体系不同,DeFi不强制要求使用某种通用的法定货币来衡量所有资产的价格.因此在DeFi中资产价格更应该被称为资产对的价格,即2种资产之间的兑换比例.例如,若当前资产对A和B的价格比为1∶2,则意味着1个资产A可以兑换2个资产B,而DeFi的价格体系正是由大量这样的资产对价格构成.资产价格的确定依赖于特定的自动做市商算法,例如恒定乘积做市商[7](constant product market maker, CPMM),在CPMM中2种资产的数量关系如图1所示(其中num_A和num_B分别代表资产A和资产B的数量).

资产兑换发生在DEX的资产兑换池中,而CPMM规定兑换池中2种资产数量的乘积为一个定值(图1中的K),这意味着当兑换池中资产A的数量减少时,用户需要更多的资产B才能兑换出同等数量的资产A.CPMM通过兑换池中2种资产的数量比例关系体现资产的价格关系.然而这也为攻击者提供了攻击的可能,他们可以通过操纵资产兑换行为来控制兑换池内资产的数量,从而操纵资产价格.这种价格操纵行为可能影响基于该资产对进行结算的金融业务,被称为价格操纵攻击(price manipulation attacks).

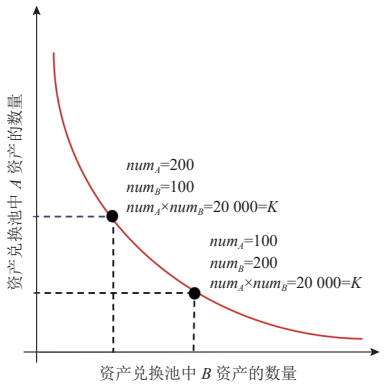


Fig. 1 The relationship between two asset quantities in a DEX

图1 去中心化交易所中2种资产数量的关系

价格操纵攻击给DeFi生态带来了巨大的经济损失,2020年部署在以太坊(Ethereum)[8]的借贷项目bZx在短短4天之内就遭受了2次价格操纵攻击,合计损失超过100万美元[9-11];2021年的ANCH事件[12]中攻击者通过在金融结算前利用闪电贷[13](flash loan)瞬间抬高结算资产价格的方式超额领取代币奖励,造成项目方约4万美元的损失;2023年Optimism事件[14]中攻击者通过操纵USDC代币[15]的价格获利约5万美元.慢雾科技安全事件资料库显示,仅2022年发生的价格操纵攻击事件就超过30件,合计造成损失超1 000万美元[16].

面对价格操纵攻击的威胁,DeFi资产需要更加可信(不易被操纵)的价格来源或定价策略,当前DeFi生态基于如何获取可信价格的研究方向主要有

2 个：1）采用更加复杂的自动做市商算法或者资产价格确定机制，如 Bancor 协议[17]、Curve Finance[18] 提出的 Stableswap 协议[19] 和 Uniswap V3 协议[20] 等；2）通过引入链下价格数据源用于金融结算. DeFi 的闪电贷（一种无抵押贷款）降低了攻击者操纵价格的门槛，但链下不存在无抵押贷款且链下价格不由自动做市商算法决定，因此链下价格无法被简单操纵，可以作为链上金融结算的可信数据源. 从链下拿取数据的项目被称为预言机，当前主流的预言机项目有 Chainlink[21] 和 Band Protocol[22] 等.

上述 2 种获取可信价格的研究方向都存在各自的问题. 研究新的自动做市商算法或者资产定价机制的问题在于无法从根源上解决价格操纵. 区块链为了保证程序运行的一致性不允许虚拟机直接从外部获取数据，因此所有运行在区块链上的程序结果都是可预测的，再加上所有部署在链上的合约都对外开源，这也就意味着无论资产价格确定机制设计得多复杂，只要它需要体现资产在市场上的供需关系，那么攻击者就可以根据开源的代码提前模拟并寻找可能存在的价格操控的机会，最直接的证明就是当前仍有各种基于不同的资产定价策略的价格操纵事件发生[16]. 使用预言机引入链下价格数据虽然可以使结算价格不被操纵，但是由于链下数据需要由矿工节点[23] 打包并经过节点共识（consensus），因此预言机需要向矿工节点支付手续费且价格数据也需要等待共识结束后才能上链，这也就意味着持续使用预言机喂价（price feeding）的维护成本过高，并且数据上链有时延也会引发新的套利攻击风险[10].

我们认为通过预言机引入链下数据源是一个可行的方向，因为如果忽略维护成本和延时的问题，那么链下价格就是由市场倾向决定的真实价格，不存在被算法操纵的空间. 因此我们迫切需要一种价格操纵在线防御机制，该机制能够在降低预言机维护成本的同时解决当前预言机喂价存在时延的问题，并且还抵御价格操纵攻击.

针对上述需求，本文提出了一种基于现有预言机改进的喂价框架，其中主要包括链下的喂价机制和链上的真实价格（real price）计算方法. 该框架结合了已有的 2 种研究方向的优点，在降低链下价格更新成本的同时，还能够以链下价格为依据计算一个链上实时的真实资产价格.

本文的主要贡献包括 3 个方面：

1）提出一种针对价格操控攻击的防御机制，通过预言机链下获取价格指导链上异常价格识别，实现了链上数字资产的实时防护.

2）面向上述防御机制，提出一种喂价策略，主被动结合地实现了预言机价格的动态维护，确保了价格操控攻击的有效识别.

3）基于真实采集的数字资产真实价格和攻击事件，本文证明了上述方法的可行性. 实验表明，本文方法可以保证 97.5% 的价格操控攻击识别精度，同时相比主流的预言机技术减少至少 30% 的维护成本.

1 背景知识

1.1 DeFi 价格操纵攻击

DeFi 是以智能合约代码形式编写，基于去中心化应用（decentralized application, DAPP）层面实现的金融系统，所有金融业务都依托于智能合约自动化执行，因此 DeFi 中资产的定价也依赖基于代码实现的自动化算法，这就为价格操纵攻击提供了实现基础.

而闪电贷则是扩大价格操纵攻击对 DeFi 危害的催化剂，闪电贷允许任何用户无抵押地借贷到大量资金，只需要在一笔交易最后归还本金和利息即可[11]，这意味着只要有获利空间，任何用户都可以近乎零成本地通过闪电贷对 DeFi 中的资产进行价格操纵（通过借来的资产操纵兑换池内资产的存量进而影响资产价），这也正是 DeFi 中价格操纵与现实中价格操纵最大的不同.

价格操纵攻击的流程为：

① 攻击者从闪电贷资金池中无抵押贷款获得 6 000 个 ETH[24]；

② 在一家抵押平台抵押 5 000 个 ETH 以获得 45 万个 USDC；

③ 将剩下的 1 000 个 ETH 投入目标 DEX 以操纵代币对 ETH/USDC 的价格，由于目标 DEX 的 2 种代币资产的储备量低，1 000 个 ETH 就可以使 ETH 价格大幅度降低（相对的 USDC 价格大幅度升高）；

④攻击者向被攻击抵押平台抵押身上所有的 USDC，由于被攻击抵押平台在评估 USDC 价格时参考的是步骤③中攻击者进行价格操纵的 DEX，该抵押平台错误地高估了抵押物 USDC 的价值并因此发放了超额的 ETH；

⑤攻击者归还在步骤①中借贷的 6 000 个 ETH 和对应的利息，最终收益约 2 000 个 ETH.

整个攻击流程的关键就在于抵押平台在进行金融结算时参考了错误的价格来源，最终进行了错误的结算. 上述流程如图 2 所示[10].

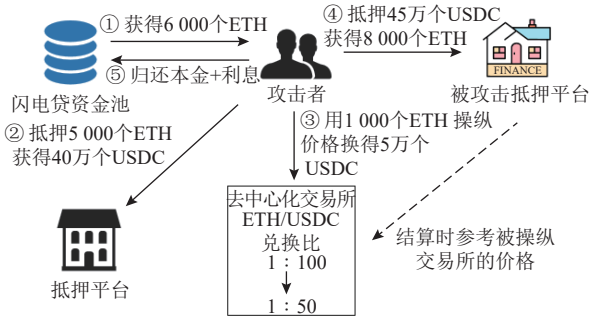


Fig. 2 DeFi price manipulation attack simplified process

图 2 DeFi 价格操纵攻击简易流程

1.2 链下数据上链

为了保证账本的一致性和确定性，智能合约需要运行在封闭的虚拟机环境中，无法直接从外界获取数据[2,25-26]，因此链下数据都需要被打包进区块，并经过矿工节点的共识后才能上链.

我们可以通过调用智能合约内的函数发起一笔交易(transaction)将链下数据以参数的形式传入函数，矿工在执行完交易内的函数后会将交易执行结果打包进区块并上链，价格数据就会随着区块的确定最终存储在智能合约中[27]，链下价格数据正是以这种方式存储在资产价格合约内并被其他 DeFi 项目所使用.

以以太坊为例，为了激励矿工打包交易消耗计算资源以维持公链运作，交易发起者需要为发起的每笔交易和所调用智能合约执行的每条操作指令支付一定数量的 gas(以太坊手续费计量单位)，而在这些指令中又以更新存储变量(storage)的指令所消耗的 gas 最高(需要将数据存储在所有矿工节点中)，约为发起 1 笔转账交易的 1/4[23].

因此预言机维护资产价格数据时需要频繁发起交易并频繁更新存储变量，运维成本高，同时因为区块同步需要时间，价格更新存在延时.

1.3 资产价格和偏差阈值

本文将会涉及多个与资产价格相关的名词，具体含义为：

1)链下价格(off-chain price). 预言机项目从链下可信数据源获取并聚合得到的资产价格. 链下价格需要通过交易存储到链上才可被智能合约获取.

2)链上价格(on-chain price). 链上可被智能合约直接获取的价格，由于价格由 DEX 中的自动做市商算法确定，因此存在被操纵的风险.

3)真实价格. 真实反映资产价值的价格，由于链下价格不依赖于自动化算法，因此通常情况下同一时刻真实价格等同于链下价格. 在不存在链上价格操纵行为的前提下，由于用户自发的套利行为，链下价格、链上价格以及真实价格应当相同.

4)异常价格(abnormal price). 被攻击者恶意操纵或因其他原因导致偏离真实价格的链上价格，异常价格的使用可能造成用户或项目方的损失.

5)喂价交易(price-feeds transaction). 链下价格数据需要通过交易存储到链上合约中才能被 DeFi 项目使用，而将链下价格同步上链的交易就是喂价交易.

6)偏差阈值(deviation threshold). 预言机将资产的链下价格存储在链上合约后就固定不变，而随着市场需求变化，该资产的真实价格发生变动之后这两者之间就会产生偏差，当预言机检测到该偏差大于某个设定好的阈值后会发起喂价交易，将最新的链下价格存储在合约中供链上使用. 上述喂价过程中预言机设定的阈值就是偏差阈值.

1.4 开发语言

本文提出的喂价框架基于以太坊实现，所使用的智能合约开发语言为 Solidity[28]，该语言的编译器 solc 会将其编译成字节码(bytecode)并部署在以太坊虚拟机(EVM)[8]中，框架的链下部分由 Golang[29]语言实现.

2 系统设计

本文提出了一种基于现有 DeFi 价格预言机改进的喂价技术，在基于该技术实现的框架中，链下价格不像传统链上价格一样直接被用于金融结算，而是作为计算链上真实价格时的清洗异常价格的参考，而计算出来的真实价格可以直接用于金融结算，也可以作为判断链上价格是否被操纵的依据. 由于链下价格不再直接用于结算，因此预言机可以以更粗略的精度进行链下价格更新，从而降低更新频率.

本节首先对该喂价技术的系统模型(改进喂价框架)进行介绍，然后对系统内的各个模块的实现方法与工作流程作进一步的阐述.

2.1 系统模型

该系统模型主要由 3 个部分组成，具体细节如图 3 所示.

1)链下聚合(aggregation)喂价模块. 该模块负责从链下的可信数据源聚合链下价格数据，并通过改进的喂价策略将链下价格存储到链上的智能合约中. 可信数据源可以是链下的中心化交易所(centralized exchange, CEX)或提供付费数据聚合服务的数据聚

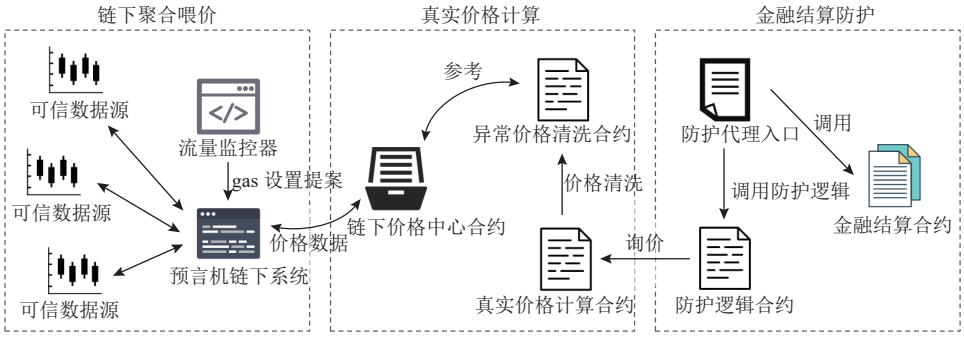


Fig. 3 Improved price-feeds system model

图 3 改进喂价系统模型

合商，如 Amberdata[30]. 为了最大化降低喂价的成本，本文设计了一种相较于传统预言机喂价成本更低且效率更高的喂价策略，由于在改进喂养系统模型中链下价格不被直接用于金融结算，因此触发喂价的偏差阈值可以比传统预言机项目更大，同时为了让喂价交易能够在预期时间内上链，该模块需要一个流量监控器对当前的交易流量进行评估，并为这笔喂价交易设置合理的交易手续费.

2）真实价格计算模块. 该模块的功能是通过考量当前链上 DeFi 生态中各个交易所的链上价格计算出某个资产在当前生态中的真实价格. 因为在计算的过程中某个交易所可能存在被价格操纵的风险，因此需要在考量每一个链上价格的时候用链下价格与之对比进行异常价格清洗. 计算出来的真实价格可以直接用于金融结算，也可以作为拦截价格操纵交易的依据.

3）金融结算防护模块. 新上链的金融项目可以直接从真实价格计算模块拿取资产的真实价格用于结算以避免价格操纵，但是由于智能合约上链之后不能更改，因此针对已上链的金融项目，金融结算防护模块提供额外的防护代理抵御并拦截价格操纵攻击.

2.2 链下聚合喂价策略

链下聚合喂价模块的核心是经过改进后能有效降低喂价成本的喂价策略. 该策略的核心点有 2 个：一是降低喂价交易的频率；二是高效率（费用低且也能在预计的时间内上链）地进行交易手续费设置.

链下的喂价形式有 2 种，分别是心跳（heartbeat）喂价和被动喂价，其中心跳喂价指的是固定每隔一段时间就主动进行一次链下价格更新，被动喂价则是指当链下价格和资产的真实价格偏差超过一定的阈值之后为了避免链下价格失去参考意义而被动进行的价格更新. 在这 2 种喂价形式中，降低喂价交易频率针对的是被动喂价，而高效率地进行交易手续费设置则对 2 种喂价形式都有效.

能够降低喂价频率主要是因为在整个改进喂价框架中链下价格不再直接用于金融结算，这相当于降低了链下价格的准确性敏感度，从而可以增大预言机的更新阈值. 而交易手续费的设置策略则依赖于流量监控器，其运行过程如图 4 所示.

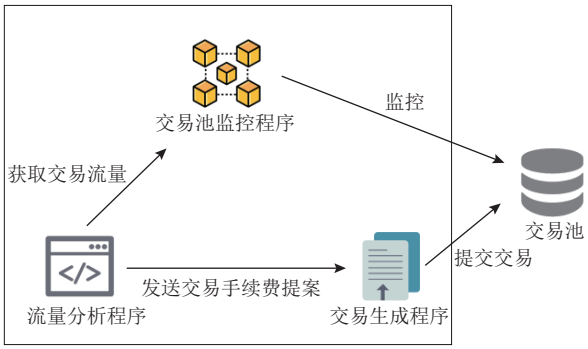


Fig. 4 Operating process of the traffic monitor

图 4 流量监控器的运行过程

流量监控器通过对区块链的交易池进行监控获得“交易手续费-交易上链时间”的流量数据，再根据对流量数据进行统计分析结果设置喂价交易的交易手续费.

2.3 真实价格计算策略

真实价格计算模块主要负责计算某个资产的真实价格，并在计算的过程中清洗掉可能存在被操纵风险的异常价格数据，其流程如图 5 所示.

因为主流的 DEX 资产锁仓量大不易被操纵，真实价格计算模块会从当前 DeFi 中主流（考量资产兑换交易的交易量占比）的 DEX 中获取所计算资产的链上价格数据.

由于链上价格可以被操纵，因此为了避免被操纵的价格最后影响到真实价格的计算，真实价格计算模块在将某个来自于 DEX 的链上价格代入计算公

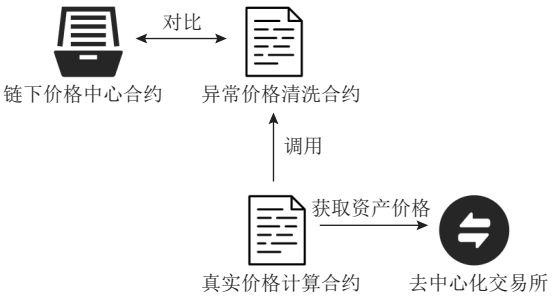


Fig. 5 Real price calculation process

图 5 真实价格计算流程

式前, 需要将该价格与来自链下的可信价格数据进行对比, 当发现两者相差过大时则将该 DEX 的链上价格抛弃, 不纳入此次计算. 由于真实价格是由链上价格数据计算而来, 因此真实价格数据具有高准确度的同时也不存在由直接使用预言机带来的数据延时问题. 最终计算出来的真实价格会综合考量各个交易所的交易量占比, 详见本文第 3 节.

2.4 金融结算防护

新的 DeFi 项目使用真实价格计算模块计算的资产真实价格进行结算就可以避免价格操纵攻击, 但由于区块链不可更改的特性, DeFi 项目上链后不可更改, 因此改进喂价框架的金融结算防护模块为已上链的 DeFi 项目提供价格操纵解决方案, 具体流程如图 6 所示.

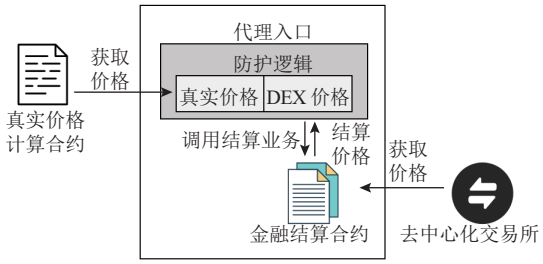


Fig. 6 Price manipulation protection process

图 6 价格操纵防护流程

防护代理入口先调用防护逻辑合约, 防护逻辑合约会先从真实价格计算模块获取真实价格作为判断依据, 然后从结算合约参考的 DEX 获取用于结算的资产链上价格进行价格异常判断, 当发现结算的价格正常时才调用金融结算业务, 若防护逻辑合约发现 DEX 的价格和真实价格偏差较大时则会抛出价格操纵风险并拦截交易.

在上述过程中, 防护逻辑合约对金融结算合约的接口进行了封装, 因此不需要改动结算合约的业务逻辑, 部分项目因实现的方式不同可能需要在其路由(router)合约添加调用入口.

3 系统实现

本节对改进喂价框架各个部分的实现细节进行详细的阐述.

3.1 链下聚合喂价模块

第 2 节提到链下聚合模块主要负责聚合链下价格数据并且在链下价格与真实价格偏差超过偏差阈值时发起喂价交易, 在发起交易前该模块还会为交易设置一个高效率的交易手续费. 因此该模块的核心在于如何确定偏差阈值, 以及如何设置喂价交易的交易手续费.

由于偏差阈值被用于触发预言机的被动喂价, 因此偏差阈值的大小会直接影响喂价交易的更新频率, 进而影响喂价维护成本. 在本文提出的改进喂价框架中, 预言机维护的链下价格不被直接用于金融结算, 因此预言机不再需要为了维护一个精准的链下价格选取较小的偏差阈值, 而更大范围的偏差阈值则意味着更低的喂价维护成本. 预言机喂价的价格数据在 DeFi 金融结算中起到关键作用, 因此偏差阈值的设定直接关系到当前 DeFi 生态中结算业务的可靠性, 其确定具有现实意义且需要综合考虑实际情况. 为了确定合适的偏差阈值, 本文采用实证研究分析法, 即通过收集、统计和分析现实 DeFi 生态中的喂价数据来进行研究并确定偏差阈值. 本文对代币对(token pair)不同阈值下的喂价数据进行研究, 如表 1 所示.

Table 1 Statistics of the Number of Updates for Asset Off-Chain Prices at Different Deviation Thresholds Within 7 Days

表 1 7 天内不同偏差阈值下各资产链下价格更新次数统计

代币对类型	代币对	0.25%	0.5%	0.75%	1%	1.25%	1.5%
稳定币/美元[31]	USDT[32]/USD	0	0	0	0	0	0
	DAI[33]/USD	4	0	0	0	0	0
	BUSD[34]/USD	0	0	0	0	0	0
其他币/美元	BTC[35]/USD		48	25	14	8	6
	ETH/USD		65	43	25	13	9
	DOGE[36]/USD		82	55	36	23	18

注: USD 是美元简称.

我们从 CoinMarketCap[37] 收集了不同资产代币对 7 天的喂价数据, 并且计算统计不同偏差阈值下预言机 7 天内需要喂价的次数.

需要说明的是, 当前主流预言机项目, 如 Chainlink

对币对“稳定币/美元”设置的偏差阈值一般为 0.25%，“其他币/美元”则为 0.5%. 我们通过“偏差阈值增大效益比”来评估每增大 0.25% 偏差阈值所引起的更新次数减少所带来的效益，具体为：

$$benefit = \frac{\Delta count_{0.25\%}}{rawUpdateCount}, \quad (1)$$

其中 $\Delta count_{0.25\%}$ 的含义为每增大 0.25% 阈值所减少的喂价次数，*rawUpdateCount* 则表示在主流预言机项目选取的偏差阈值下喂价的次数，因此 *benefit* 表示每增大 0.25% 偏差阈值所减少的喂价次数与原始喂价次数的占比，*benefit* 越大则表示该扩大的 0.25% 阈值带来的效益越大.

利用式(1)对喂价数据进行分析后发现，由于稳定币大部分会对美元进行强锚定，因此针对大部分“稳定币/美元”类型的代币对，偏差阈值从 0.25% 增大至 0.5% 时的 $benefit = 0$. 而针对“其他币/美元”类型的代币对，当偏差阈值从 1.25% 增大至 1.5% 时，$benefit < 10\%$，由增大偏差阈值所带来的减少更新次数导致效益大幅度降低.

综上所述，在链下聚合喂价模块中，针对“其他币/美元”类型的代币对，其偏差阈值可以增大至 1.25%. 在现有 DeFi 生态中还存在“其他币/其他币”类型的代币对，这些代币对同样可以通过上述方法确定增大后的偏差阈值.

此外，当前主流预言机项目所选取的偏差阈值即使仅有 0.5% 也存在被套利的空间，而本文提出的改进喂价框架由于不将链下价格作为结算价格，而是作为判断某个链上是否有被操纵的依据，并且实时计算真实价格用于结算，因此即使偏差阈值相对更大也不存在被套利的风险，最终结算的真实价格数据也更精准(真实价格不存在 0.5% 的误差).

当链下聚合喂价模块需要进行喂价时，为了更快地让链下价格数据上链，模块需要为这笔交易设置一个较高的交易手续费，但是越高的交易手续费意味着越高的喂价成本，因此链下聚合喂价模块还需要解决“时间-运费平衡”问题，即事务在规定时间内完成，且费用尽可能低.

文献 [38] 提出了一种评估“交易手续费-交易上链时间”对应关系的方法，该方法先统计并分析大量区块链交易中交易手续费和对应上链时间 t 的数据，然后通过最小二乘法[39] 计算从交易手续费映射到 t 的一元线性回归方程模型“$t=kG+d$”，其中 k 和 d 为通过最小二乘法对交易“交易手续费-t”数据进行线性拟合后得到的系数，G 为交易中交易手续费的值.

一元线性回归方程模型“$t=kG+d$”可以直接应用于改进喂价框架中的交易手续费设置，但是我们希望在改进喂价框架中 gas 能被更高效率地使用(即原本能用于让 10 个交易在 1 个区块内上链的 gas 开销现在可以让 15 个交易在可接受的时间范围内上链)，因此我们对上述一元线性回归方程做出调整. 在详细描述链下聚合喂价模块的交易手续费设置策略前，我们先给出一些符号的定义和解释.

定义 1. “*t*-交易手续费”方程. 通过对一元线性回归方程模型“$t=kG+d$”取逆函数得到从 t 映射到 G 的一元线性回归方程“$G=(t-d)/k$”，其中 t 表示交易上链的时间，G 表示交易设置的交易手续费值. “*t*-交易手续费”方程的含义在于随期望上链时间的变化，交易所需设置的交易手续费期望值的变化趋势，其功能主要是调整交易手续费设置来优化交易上链效率. 本文简化“*t*-交易手续费”方程为 $G=f(t)$，f 为根据输入 t 计算 G 的一元线性计算规则.

定义 2. 实际价格偏差 *actDeviation*. 指喂价时合约中存储的链下价格与该资产当前的真实价格之间的偏差百分比，例如当前合约中存储的链下价格为 90，而该资产当前的真实价格为 100，则喂价时 *actDeviation* 为 10%.

定义 3. 异常价格阈值 *abThreshold*. 是指用于判定某一资产的链上价格存在被操纵风险时，链上价格与资产当前真实价格之间的最小偏差值. 即当我们从某个 DEX 中获取的链上价格与该资产的真实价格(在具体工程实现时使用的是链下价格)偏差超过 *abThreshold* 时，该 DEX 存在被价格操纵的风险，*abThreshold* 主要用于判断某个链上价格是否存在被操纵风险，被应用于真实价格计算模块中的异常价格清洗过程. *abThreshold* 的确定方案会在 3.2 节中详细描述.

流量监控器可以直接用 $G=f(t)$ 算出某笔喂价交易所需设置的交易手续费，但是我们希望能够更加高效地进行交易手续费的设置，当 *actDeviation* 较大时，意味着存储在链下价格中心合约中的链下价格与真实价格误差较大，此时该资产被价格操纵的可能性变高，因此我们必须设置较高的交易手续费以保证这笔喂价交易能够尽快上链(侧重考虑上链速度). 当 *actDeviation* 较小时，则此时链下价格中心合约中的链下价格与真实价格误差较小，喂价交易并不需要快速上链(侧重考虑交易手续费设置性价比)，基于这个需求，我们对 $G=f(t)$ 公式做了一定的优化：

$$G=\left(a+b\frac{actDeviation}{abThreshold}\right)f(t), \tag{2}$$

其中 G 表示喂价时需要设置的交易手续费, a 是基础系数, 即在不考虑喂价时链下价格中心合约中的链下价格与真实价格存在偏差的情况下 G 与 $f(t)$ 的倍数关系. b 为偏差系数, 即控制喂价时价格偏差对交易手续费设置的影响, b 越大则价格偏差对交易手续费设置的影响越大, a 和 b 可以根据每种资产的具体情况单独调整. 若 $b\neq0$, 喂价时链下价格中心合约中的链下价格与真实价格相差越小则计算出的 G 就越小, 而当价格偏差较大, 甚至超过异常价格阈值 *abThreshold* 时, 最终得到的 G 甚至可能为 $f(t)$ 的数倍. 式(2)中的自变量为 t 和 *actDeviation*, 也就是说最终的交易手续费大小受预期的时间和价格偏差影响.

链下聚合喂价模块进行心跳喂价时, *actDeviation* 一般较小, 因此交易手续费计算公式近似为 $G=af(t)$. 进行被动喂价时, 此时 *actDeviation* 为被动更新的偏差阈值, 因此 G 的结果会比 $f(t)$ 稍大, 而倘若被动喂价时因交易池拥塞导致链下价格不能上链, 则最终可能出现 *actDeviation* 远大于被动更新阈值设置超过 *abThreshold* 的情况, 此时式(2)就会计算出一个较高的 G, 保证即使交易池拥塞的情况下所设置的交易手续费依旧能高于大部分交易, 并在预期时间内上链.

流量监控器通过在虚拟机插桩的方式监控交易池, 获取“交易手续费-t”形式的流量数据以调整模型 f, 其工作流程如图 7 所示.

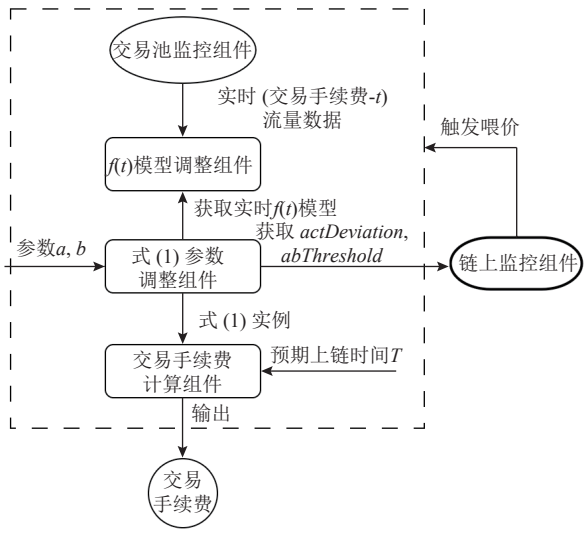


Fig. 7 Workflow of the traffic monitor

图 7 流量监控器工作流程

流量监控器的工作流程分为 3 个步骤:

1)链上监控组件发现链下价格中心合约中的链下价格与真实价格偏差超过偏差阈值或更新时间超过心跳间隔时, 向预言机发送喂价请求, 预言机触发流量监控器的交易手续费计算工作流.

2)式(2)参数调整组件向链上监控组件获取参数 *actDeviation* 和 *abThreshold*, 然后向 $f(t)$ 模型调整组件获取最新的 $f(t)$ 模型, 获得式(2)实例.

3)交易手续费计算组件根据式(2)实例和预先设置好的预期上链时间 T 计算交易需要设置的交易手续费.

在上述过程中, 交易池监控组件会实时监控区块链交易池并向 $f(t)$ 模型调整组件发送交易流量, 调整组件就能根据网络状况动态调整模型, 以适应实时变化的交易流量环境.

3.2 真实价格计算模块

真实价格计算模块主要负责计算某个资产在当前生态中的真实价格, 而为了保证计算的真实价格的“真实性”(指价格没有被操纵), 在计算的过程中需要对用于计算链上价格进行价格清洗. 由于真实价格已经排除了被操纵的风险, 该价格可以直接被用于金融结算, 或者是作为价格标准判断某笔交易中是否存在价格操纵的风险.

真实价格计算模块会根据各个 DEX 的交易量占比进行价格加权计算. 以以太坊为例, 我们可以在以太坊区块链浏览器 Etherscan[40] 中直接获取 7 天时间内各个 DEX 中代币兑换交易数量占所有代币兑换交易数量的比例, 如图 8 所示.

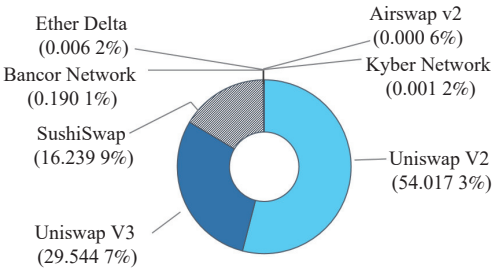


Fig. 8 Proportion of DEX trading volume within 7 days

图 8 7 天内去中心化交易所交易量占比

根据交易量占比计算真实价格的公式为:

$$R=A\alpha+B\beta+C\gamma+\cdots, \tag{3}$$

其中 R 表示所计算的真实价格, A, B, C 为所计算资产在交易量占比最高的 3 个 DEX 中的链上价格(参考的 DEX 数量根据实际占比情况变动, 不限定为 3 个), α, β, γ 为对应 DEX 的交易量占比, 由于所选取 DEX 交易量占比之和不一定为 1, 因此需要对权重进行归一化处理(即使得 $\alpha+\beta+\gamma=1$). 选择交易量占比高的 DEX 的原因在于这些 DEX 中资产的锁仓量

较大，这会增加攻击者的操纵难度，而参考多家 DEX 也使得真实价格被操纵的可能性被最大程度降低（攻击收益无法覆盖通过闪电贷操纵交易量占比最高的几个 DEX 所需要的利息）.

由于交易量占比高的 DEX 也存在被操纵的情况，且一旦计算真实价格过程中某个被参考的 DEX 的链上价格被操纵会大幅度影响真实价格的准确性，因此在计算真实价格之前还需要判断链上价格是否存在被操纵风险，若存在被操纵风险则在此次计算时排除该价格数据.

真实价格的计算流程共有 3 个步骤，如图 9 所示.

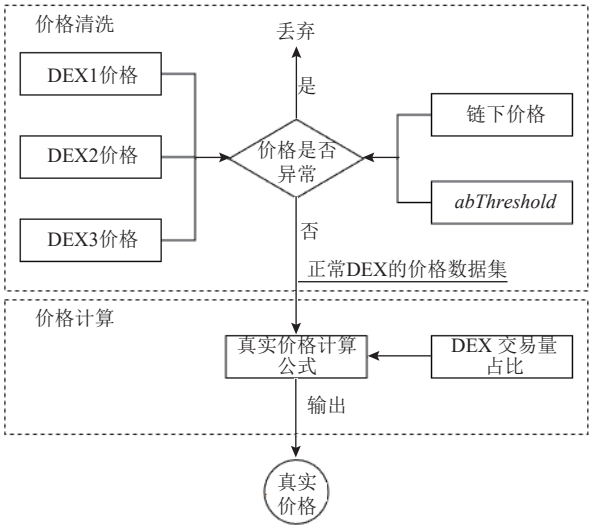


Fig. 9 Abnormal price cleaning and real price calculation

图 9 异常价格清洗和真实价格计算

1）异常价格清洗合约获得需要进行异常价格判定的 DEX 中的链上价格.

2）针对每一个获得的链上价格，根据链下价格和 *abThreshold* 进行价格清洗，即当链上价格和链下价格的偏差百分比超过 *abThreshold* 时，舍弃该价格.

3）异常价格清洗合约将清洗后的价格数据集传给真实价格计算合约，真实价格计算模块根据式（3）和 DEX 交易量占比计算真实价格.

在具体的实现中，DEX 中的链上价格由真实价格计算模块获取，并在真实价格计算模块调用异常价格清洗模块时被作为参数传入.

abThreshold 在整个异常价格清洗的过程中起着至关重要的作用，该值的合理性直接决定了整个改进喂价框架对价格操纵的防护效果. 由于 *abThreshold* 的选定和偏差阈值一样具有现实意义且需要综合考虑实际情况，因此同样采用实证研究分析法选定该值.

前面提到基于闪电贷的价格操纵攻击会在 1 笔交易内完成对链上价格的操纵，将某个资产的链上价格操纵至一个可以获利的价格区间（为了覆盖攻击成本并获利，每次操纵都会大幅度改变资产链上价格），因此 1 笔价格操纵攻击交易就可以引起链上价格的剧烈变动.

基于上述价格操纵攻击交易存在的特点，我们可以对资产的兑换交易进行监控，分析资产链上价格波动范围和对应交易数量的分布关系.

本文从区块链开源数据平台 Dune[41] 获取代币兑换交易的数据进行分析，以 BTC/ETH 代币交易池为例，我们采集并分析了 70 万笔 BTC/ETH 兑换交易，其中导致不同程度链上价格波动的各种交易的数量分布如表 2 所示.

Table 2 Distribution Table of the Number of Trades Causing Different On-Chain Price Fluctuations

表 2 导致不同链上价格波动的交易数量分布表

价格波动范围/%	交易数量
≤1	655 436
1~90	44 512
≥90	52

从表 2 看出，在全部 70 万笔交易中，有约 65 万笔兑换交易对资产链上价格造成的影响小于 1%，而仅有 52 笔交易造成超过 90% 的价格波动，剩下的 44 512 笔交易对链上价格的影响幅度为 1%~90%. 价格影响小于 1% 的交易是正常的代币兑换交易，而影响大于 90% 的交易则会严重影响资产链上价格（相当于 1 笔交易就让价格涨幅约 1 倍），因此我们需要在 1%~90% 的价格波动范围之间寻找一个阈值，该阈值在不影响绝大部分用户兑换行为的同时还能筛选出价格操纵交易. 我们对剩下的 44 512 笔交易进行分析，其“价格波动范围-总交易数量”分布图如图 10 所示.

图 10 的横坐标代表资产链上价格波动的范围（用百分比表示），纵坐标代表对价格波动小于某个横坐标范围的总交易数量，例如一个点的坐标为（17.9%，42 912），这代表对价格波动影响小于 17.9% 的交易共有 42 912 个.

从图 10 可以看出，随着价格波动范围增大，总交易数量的增加量逐渐减少（斜率逐渐减小），因此我们可以在图中找到一个“拐点”（如图 10 中虚线标识出来的点），价格波动范围小于该点的交易占大部分（如果加上小于 1% 的交易则数量更多），而大于该点的交易虽然存在但是数量较少，且这些交易每一笔

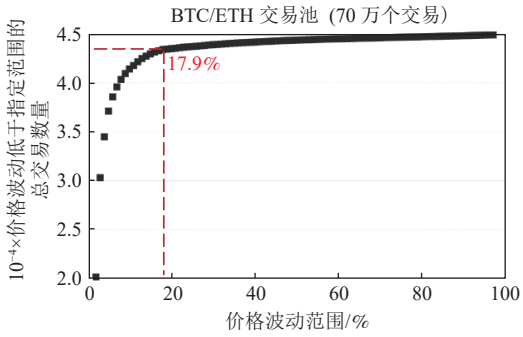
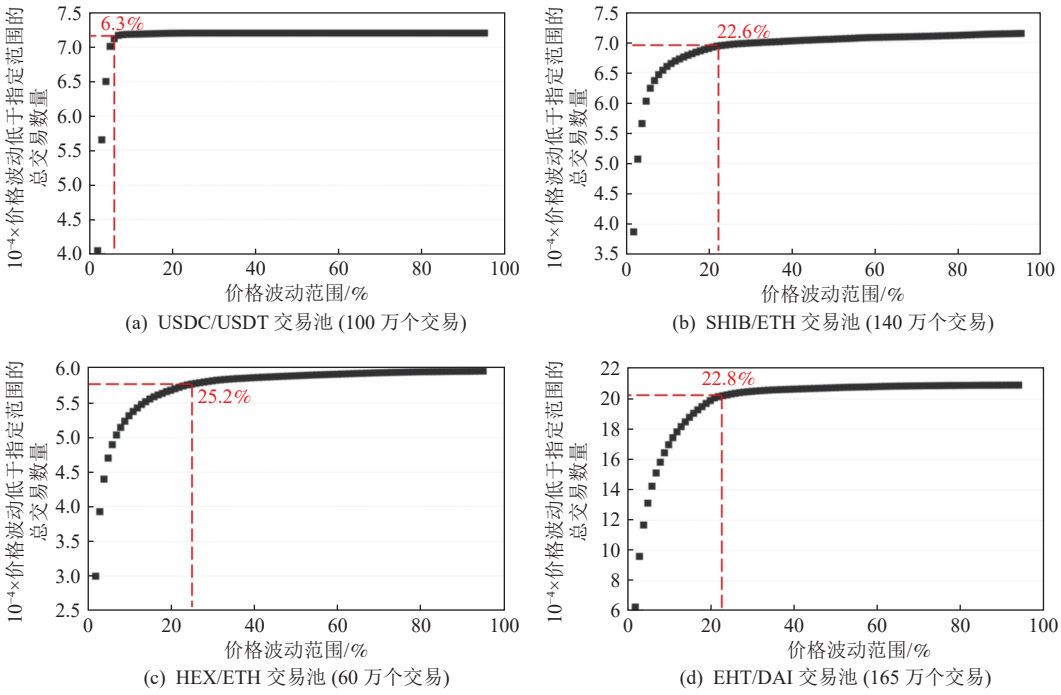


Fig. 10 Distribution chart of price fluctuation range and total trade volume for the BTC/ETH token pair

图 10 BTC/ETH 代币对“价格波动范围-总交易数量”分布图

都会对资产链上价格产生较大的影响, 则我们以该点对应的价格波动范围作为 *abThreshold*, *abThreshold* 可以作为每笔交易的价格波动安全上限, 当一笔交易造成的链上价格波动超过 *abThreshold* 则会被认定存在价格操纵风险(即使是正常交易也会被认为存在风险). 在具体实现中, 由于“价格波动范围-总交易数量”的分布满足价格波动范围越大交易数量增加越少的规律, 则我们可以通过检测交易数量的增长率来确定 *abThreshold*, 当增长率低于当波动范围提高 0.1%, 交易数量对应增加 10 时的增长率时, 则认为该点的百分比为 *abThreshold*, 因此用该方法可以得出 BTC/ETC 代币对的 *abThreshold* 值为 17.9%.

通过上述方法得到的 *abThreshold* 可以确保将绝大部分(超过 99.5%)交易认定为正常交易, 仅将小部分会剧烈影响资产链上价格的交易标识为异常交易, 所以基于 *abThreshold* 开展的任何防护行为都不会影响该交易池的正常兑换业务. 我们还采集分析了多个不同资产的兑换交易数据并计算了各自的 *abThreshold*, 结果如图 11 所示. 从图 11 可以发现不同的资产交易池, 由于其资产的属性和作用各不相同, 其数据分布也具有不同的特征, 因此最终计算出的 *abThreshold* 也不相同.



(a) USDC/USDT 交易池 (100 万个交易)

(b) SHIB/ETH 交易池 (140 万个交易)

(c) HEX/ETH 交易池 (60 万个交易)

(d) EHT/DAI 交易池 (165 万个交易)

Fig. 11 *abThreshold* of different assert pairs

图 11 不同资产对的 *abThreshold*

3.3 金融结算防护模块

由于真实价格本身就已经排除了异常价格的干扰, 所以它可以直接被用于金融结算而不必担心被价格操控. 但是由于智能合约上链后不可更改, 目前已经运行在链上的且存在价格操纵风险的金融项目并不能简单地就完成对业务合约的升级以接入链上价格计算模块接口.

使用 EIP-1822[42] 通用可升级合约代理标准或 OpenZeppelin[43] 合约代理升级框架的项目由于业务逻辑与业务数据分离, 可以以较低的成本完成对业务合约的更新, 但是没有使用合约可升级代理框架的项目则需要进行数据迁移, 耗时耗力. 为了应对这种情况, 改进喂价框架提供了另一种代理方案, 该方案不对价格操纵的判断逻辑嵌套在业务内, 而是封

装在了整个金融结算业务的最外层. 防护合约代理在调用结算业务前先将结算合约参考的 DEX 中的目标资产链上价格和真实价格进行对比(目标 DEX 合约地址会作为参数传入), 当发现价格异常则直接拦截交易, 防护逻辑的简易伪代码如图 12 所示.

```
①function defense(address dex)
②   public returns(bool){
③   realPrice=getRealPrice();
④   state=isValid(dex.getPrice(),realPrice);
⑤   if(!state){
⑥      revert();
⑦   }
⑧   settlement();
⑨}
```

Fig. 12 Simple pseudocode for the protective agent contract

图 12 防护代理合约简易伪代码

图 12 的行④中防护代理会从真实价格计算模块获取目标资产的真实价格, 然后在行⑤中判断目标交易所的链上价格是否正常, 若价格正常则触发行⑧的结算业务, 否则就在行⑥中提前中止该笔交易. 由于不同项目的结算逻辑, 业务复杂程度各不同, 项目方可以通过公开的接口获取资产真实价格, 并根据不同的项目需求编写不同形式的防护代理逻辑.

4 实验结果与分析

针对本文提出的改进喂价框架, 我们从 2 个角度切入进行实验设计:

1)改进喂价框架的维护成本是否有效降低;

2)是否能够成功防御价格操纵攻击.

4.1 实验准备

出于设备和实验成本的考虑, 本文的实验主要在自行搭建的以太坊私链中进行, 以太坊客户端版本为 geth[44] v1.10.16, 所有合约基于 Solidity v0.8.0 开发. 链下系统使用 Golang v1.18 和 web3.js[45] 开发, 预言机网络则使用的是 Chainlink 框架(已简化, 主要保留喂价部分), 均部署在 Centos7.5 系统环境中.

我们通过 CoinMarketCap 的 Quotes Historical v2[46] 接口获取 2022 年 10 月 1 日至 2022 年 12 月 31 日的部分资产历史链下价格数据进行历史喂价交易重放以评估改进喂价框架的维护成本, 并且从主网中采集相同或近似时间点下 Chainlink 喂价所设置的交易手续费参数, BTC/USD 和 ETH/USD 资产的心跳喂价频率设置为每小时 1 次, DOGE/USD 和 ETH/DAI 为每 24 小时 1 次(与 Chainlink 的心跳喂价策略相同). 针对偏差阈值的设置, Chainlink 对照组与它当前在生产环境中的设置一致, 而改进喂价框架为 2%.

在进行历史喂价交易重放实验的同时我们会记录 Chainlink 框架和改进喂价框架每一笔喂价交易的交易确认时间(从交易进入交易池到最终上链的时间)、交易手续费参数设置以及 *actDeviation*, 并对所得数据进行分析. 我们将式(2)中的 a 和 b 设置为 0.8, 通过调整出块难度将出块时间设置约为 12 s, 并且通过采集主网历史交易手续费数据调整交易手续费设置模型以设置每一笔喂价交易的交易手续费, 同时为了模拟对应历史时间的网络拥塞情况, 我们编写了交易自动发送脚本自动向私链交易池发送交易, 脚本会根据对应时间点主网区块打包的交易数量和交易手续费评估网络状况, 动态调整交易发送速度和设置交易手续费以模拟当时的网络情况.

在重放实验的过程中我们总共进行了 120 次针对不同资产的攻击模拟实验和 200 次正常结算模拟实验, 并且重放了部分真实发生过的价格操纵攻击, 以评估改进喂价框架对价格操纵攻击的防御表现.

4.2 结果与分析

实验结果表明改进喂价框架能明显降低预言机的喂价维护成本, 并且对于价格操纵也有很显著的防护效果.

表 3 为将 2022 年 10 月 1 日至 2022 年 12 月 31 日的部分资产历史喂价交易重放后 Chainlink 和改进喂价框架进行喂价的次数和成本比较表. 从表 3 中可以

Table 3 Comparison of Price-Feeds Cost for Partially Assets Based on Replayed Historical Prices-Feeds Transaction

表 3 部分资产基于历史喂价交易重放后的喂价成本比较

代币对	总喂价次数		被动喂价次数		花费的总 gas 费用/wei[8]		总 gas 费用节约比率/%
	Chainlink	改进喂价框架	Chainlink	改进喂价框架	Chainlink	改进喂价框架	
BTC/USD	3 272	2 228	1 088	44	528 483 444	349 187 756	31.91
ETH/USD	4 729	2 237	2 545	53	750 180 186	354 864 258	52.70
DOGE/USD	4 511	709	4 420	618	677 809 327	106 532 213	84.28
ETH/DAI	531	144	44	53	79 305 381	21 506 544	72.88

看出改进喂价框架喂价时所花费的总 gas 费用与 Chainlink 相比明显减少, 各代币对的喂价消耗减少比例均超过 30%, 并且心跳喂价频率为每小时 1 次的资产(BTC/USD, ETH/USD)节约的 gas 费用比例相对于心跳喂价频率为每 24 小时 1 次的资产(DOGE/USD, ETH/DAI)节约的 gas 费用比例要低, 这是因为改进喂价框架主要是通过增加偏差阈值以降低预言机的被动喂价频率, 进而降低预言机整体的喂价运维成本, 因此对于改进喂价框架而言, 心跳喂价频率越低, 则被动喂价的影响越明显, 整体喂价成本越低.

表 4 为历史喂价交易重放实验过程中 Chinlink 和改进喂价框架所有正常上链的喂价交易手续费均值.

Table 4 Comparison of Average Price-Feeds Transaction Fees in Replay Experiments

表 4 重放实验中平均喂价交易手续费比较

代币对	平均交易手续费/Gwei		节约比率/%
	Chainlink	改进喂价框架	
BTC/USD	25.327	20.226	20.14
ETH/USD	27.658	21.134	23.59
DOGE/USD	31.125	27.776	10.76
ETH/DAI	26.435	20.348	23.03

注: Gwei 表示 10^9×wei.

由表 4 可以发现改进喂价框架的交易手续费设置普遍低于 Chainlink, 这是因为改进喂价框架在为每一笔交易设置交易手续费时都会根据 *actDeviation* 评判当前这笔喂价交易的紧迫程度(由于改进喂价框架的链下价格不被用于结算, 所以即使是被动喂价交易的紧迫程度也较低), 对于不紧迫的喂价交易会设置一个较低的交易手续费.

预言机喂价运维成本主要取决于 3 个因素:

1)心跳喂价频率.

2)被动喂价频率, 取决于偏差阈值.

3)喂价花费的 gas 数量和交易手续费.

在实验中, Chainlink 和改进喂价框架在同一资产的心跳喂价频率一致, 并且由于喂价时仅改变链下价格的值而不改变数据格式, 因此 Chainlink 和改进喂价框架在喂价时花费的 gas 数量基本一致. 而从表 3 和表 4 可以看出改进喂价框架的被动喂价频率和交易手续费都相对更低, 所以改进喂价框架的喂价维护成本低于 Chainlink. 在实际的生成环境中, 改进喂价框架的心跳喂价频率可以更低.

表 5 为历史喂价交易重放实验过程中所有 Chainlink 和改进喂价框架的喂价交易的确认时间平均值.

Table 5 Average Confirmation Time of Price-Feeds Replay Transactions

表 5 喂价重放交易平均确认时间

代币对	平均确认时间/s		提升比率/%
	Chainlink	改进喂价框架	
BTC/USD	7.82	7.54	3.58
ETH/USD	6.19	6.23	–0.65
DOGE/USD	8.83	8.43	4.55
ETH/DAI	5.22	5.47	–4.79

可以发现改进喂价框架的喂价交易平均确认时间与 Chainlink 框架相比没有明显优势. 原因在于改进喂价框架的交易手续费设置策略并不是一味追求喂价交易上链的速度, 而是追求交易手续费的使用效率和价格数据的安全性, 即当某资产的 *actDeviation* 较大, 该资产被价格操纵的可能性变高时, 改进喂价框架必须设置较高的交易手续费以保证这笔喂价交易能够尽快上链, 此时侧重考虑上链速度(优先保障价格安全); 而当 *actDeviation* 较小时, 此时链下价格中心合约中的链下价格与真实价格误差较小, 喂价交易并不需要快速上链, 则侧重考虑交易手续费设置性价比(注重节约 gas 开销), 将交易手续费设置在当前网络均值以下. 因此在上述策略下, 当 *actDeviation* 较大时, 改进喂价框架的喂价交易确认时间比 Chainlink 短, 反之则较长.

表 6 和表 7 为改进喂价框架在价格操纵防护方面的误报率和防护率数据.

Table 6 Price Manipulation False Positive Rate of the Improved Price-Feeds Framework

表 6 改进喂价框架的价格操纵误报率

代币对	结算次数	攻击识别次数	误报率/%
BTC/USD	50	1	2
ETH/USD	50	0	0
DOGE/USD	50	2	4
ETH/DAI	50	0	0
总计	200	3	1.5

我们在重放实验进行的过程中分别在各个代币对池中进行了 30 次攻击和 50 次正常结算模拟, 发现改进喂价框架对于价格操纵攻击的误报率为 1.5%, 防护率为 97.5%.

产生误报是因为我们模拟了在结算行为发生前某用户向被参考交易所提交了一笔较大幅度改变币对链上价格的兑换交易, 并且该链上价格改动幅度超过异常价格阈值 *abThreshold*. 此时由于被参考交

Table 7 Price Manipulation Defense Rate of the Improved Price-Feeds System

表 7 改进喂价系统的价格操纵防护率

代币对	攻击次数	攻击识别次数	防护率/%
BTC/USD	30	30	100
ETH/USD	30	29	96.66
DOGE/USD	30	28	93.33
ETH/DAI	30	30	100
总计	120	117	97.5

易所的目标资产链上价格与真实价格偏差超过异常阈值，此次结算交易被认为存在价格操纵风险. 当然，由于造成超过异常阈值范围的正常兑换交易同样存在潜在的价格操纵风险，因此改进喂价框架也会将此类交易归为价格操纵交易.

出现对价格操纵攻击防护失败的情况与误报的情况类似，即攻击者操纵价格的范围小于异常阈值，因此不被判断为价格操纵攻击，然而在实际的攻击事件中，价格操纵攻击导致的资产波动范围要远大于异常阈值，否则攻击者的获利无法覆盖攻击成本（包括交易手续费、闪电贷手续费等）.

我们还重放了历史上真实发生过的价格操纵攻击以检验改进喂价框架的事件，重放的攻击包括第 1 次 bZx 价格操纵[10]、第 2 次 bZx 价格操纵[11]、ANCH 事件[12] 以及 Optimism 事件[14]，每个攻击事件重放 5 次，结果如表 8 所示. 由于真实攻击案例中价格操纵导致的资产变化幅度远大于 *abThreshold*（往往会导致链上价格产生数倍的变化），因此所有的重放攻击均被改进喂价框架识别并拦截，防护率为 100%.

Table 8 Defense Effect of Improved Price-Feeds System on Real Price Manipulation Attacks

表 8 改进喂价系统对真实价格操纵攻击的防护效果

攻击事件	重放次数	防御次数	防护率/%
bZx 价格操纵[7]	10	10	100
ANCH 事件[10]	5	5	100
Optimism 事件[12]	5	5	100
总计	20	20	100

5 相关工作

当前面对价格操纵攻击防护的研究方向主要有 2 个：一个是研究更加复杂的资产链上价格算法或确定机制，如 Bancor 协议[17]、Stableswap 协议[19]、Uniswap V3 协议[20] 等；另一个则是通过预言机引入链下数据源，如 Chainlink[21]，Band Protocol[22]，Teller[47] 等.

针对资产链上价格算法，无论是 Bancor，Stableswap 又或者是 Uniswap V3，它们都需要依赖于资产的锁仓量或者是市场的供需关系自动计算资产链上价格，这就为价格操纵预留了操纵接口，由于所有运行在区块链上的程序结果都是可预测的，且所有部署在链上的合约都对外开源，这也就意味着无论资产链上价格确定机制设计得多复杂，攻击者都可以根据开源的代码寻找资产链上价格确定方法的“锚定物”，并发现可能存在的价格操控的机会.

而预言机项目虽然可以较为有效地防御价格操纵攻击，但是普遍的缺点就是喂价维护成本高且链下价格数据不够精确. Chainlink 为了降低喂价频率，增大了部分代币对的被动更新阈值，这在一定程度上牺牲了价格数据的精准性；Band Protocol 虽然将链下价格数据维护在了交易成本较低的高性能区块链 BandChain[48] 上. 但是在向其他区块链生态发送链下价格数据时却需要支付高昂的跨链费用；Tellor 则是利用加密货币经济激励机制激励矿工提供正确数据，因此链下数据来源可能存在风险.

上述工作没有很好地解决价格操纵攻击带来的问题，有些工作还引入了新的问题，因此仍有待改进和完善.

6 讨 论

本文提出的改进喂价框架虽然比传统预言机项目节约喂价成本，并且能够防御价格操纵攻击，但是仍然存在 2 个方面的不足：

1）提高了结算业务的成本. 由于区块链中更改存储状态操作需要的 gas 数量较高，为了降低维护成本，改进喂价框架不将真实价格存储在合约中，而是在每次需要它时才实时计算，这也就要求用户每次触发资产结算时都需要计算一遍真实价格，而跨合约调用获取 DEX 中目标资产的链上价格和计算真实价格的操作都需要消耗额外的 gas（比更改存储状态的操作要小得多），因此若是涉及需要频繁大量触发金融结算的业务，则需要权衡喂价降低的 gas 和计算真实价格增加的 gas 之间的关系（大部分情况下改进喂价框架所消耗的 gas 费用都比传统预言机方案要少）.

2）可以直接调用真实价格计算接口进行业务代码编写. 金融结算防护模块的代理防护方案只能作为临时解决方案，真正能让改进喂价框架为金融项目提供高效率防护的方式还是直接将真实价格的计

算逻辑嵌入清算逻辑中,这就需要已部署上链的金融项目对合约进行升级,而新的金融项目直接调用真实价格计算接口进行业务代码编写即可.

7 总 结

本文提出一种面向 DeFi 价格操控攻击的防御机制,通过链外获取的链下价格识别被操控的链上价格,进而保护 DeFi 去中心化应用项目的清算机制. 本文实验表明,本文方法可有效对抗价格操控攻击,相比于目前的主流业界实现方案,本文方法在具有 97.5% 防护率的基础上可降低 30% 的防御成本,具有良好的推广价值.

作者贡献声明:林炼升提出论文思路,并负责设计、开展实验以及进行论文的撰写与修改;郑焕钦辅助完成实验,并负责实验数据的整理与分析;苏申负责把握论文的技术方向,并对论文内容进行指导和提出修改意见;雷凯、陈晓丰和田志宏对论文写作进行指导.

参 考 文 献

[1] Nofer M, Gomber P, Hinz O, et al. Blockchain[J]. Business & Information Systems Engineering, 2017, 59: 183−187

[2] Wang Shuai, Yuan Yong, Wang Xiao, et al. An overview of smart contract: Architecture, applications, and future trends[C]//Proc of the 2018 IEEE Intelligent Vehicles Symp (IV). Piscataway, NJ: IEEE, 2018: 108−113

[3] Liu Yuan, Zhang Chuang, Yan Yu, et al. A semi-centralized trust management model based on blockchain for data exchange in IoT system[J]. IEEE Transactions on Services Computing, 2022, 16(2): 858−871

[4] Su Shen, Tian Zhihong, Li Shuang, et al. IoT root union: A decentralized name resolving system for IoT based on blockchain[J]. Information Processing & Management, 2021, 58(3): 102553

[5] Jensen J R, von Wachter V, Ross O. An introduction to decentralized finance (DeFi)[J]. Complex Systems Informatics and Modeling Quarterly, 2021(26): 46−54

[6] Wood A, Cohen A, Allen S, et al. What are decentralized exchanges, and how do DEXs work[EB/OL]. [2023-01-11]. https://cointelegraph.com/defi-101/what-are-decentralized-exchanges-and-how-do-dexs-work

[7] Berenzon D. Constant function market makers: DeFi's "zero to one" innovation[EB/OL]. [2023-01-11]. https://medium.com/bollinger-investment-group/constant-function-market-makers-defis-zero-to-one-innovation-968f77022159

[8] Buterin V. A next-generation smart contract and decentralized application platform[EB/OL]. [2023-01-11]. https://ethereum.org/en/whitepaper/

[9] Pesse S. First attacks[EB/OL]. [2023-01-11]. https://xiangganzi.gitbook.io/flashloan/shan-dian-dai-gong-ji-ce-lve/bzx-di-yi-ci-gong-ji-ce-lve

[10] Hellobtc. DeFi world's Oracle attack[EB/OL]. [2023-01-11]. https://www.blocktempo.com/why-defi-arbitrageurs-earn-more-than-hackers

[11] Williams C, Craig T, Oliver J, et al. 2388 ETH estimated lost in bZx's second exploit[EB/OL]. [2023-01-11]. https://cryptobriefing.com/2388-eth-estimated-lost-bzxs- second-exploit

[12] Pan Yubo, Gong Quanyu. Beosin: Anch price manipulation[EB/OL]. [2023-01-11]. https://www.chaincatcher.com/article/2077562

[13] Wang Dabao, Wu Siwei, Lin Ziling, et al. Towards a first step to understand flash loan and its applications in DeFi ecosystem[C]//Prof of the 9th Int Workshop on Security in Blockchain and Cloud Computing. New York: ACM, 2021: 23−28

[14] Odaily. Optimism price manipulation[EB/OL]. [2023-01-13]. https://www.odaily.news/newsflash/310619

[15] Picardo E. USD coin (USDC): Definition, how it works in currency, and value[EB/OL]. [2023-01-13]. https://www.investopedia.com/usd-coin-5210435

[16] SlowMist. Slowmist hacked[EB/OL]. [2023-01-13]. https://hacked.slowmist.io

[17] Hertzog E, Benartzi G. Bancor Protocol[EB/OL]. [2023-01-13]. https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf

[18] Curve. Curve Finance[EB/OL]. [2023-01-13]. https://www.curve.com

[19] Wombat Exchange. What is Stableswaps[EB/OL]. [2023-01-13]. https://medium.com/wombat-exchange/what-is-a-stableswap-58786d32e3d4

[20] Adams H, Zinsmeister N, Salem M, et al. Uniswap V3 core[EB/OL]. [2023-01-13]. https://berkeley-defi.github.io/assets/material/Uniswap%20v3%20Core.pdf

[21] Breidenbach L, Cachin C, Chan B, et al. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks[EB/OL]. [2023-01-13].https://research.chain.link/whitepaper-v2.pdf?_ga=2.22594877.704683947.1680333446-783667085.1668944739

[22] Srinawakoon S, Suriyalarn S, Nattapatsiri P. Band Protocol documentation[EB/OL]. [2023-01-13]. https://docs.bandchain.org

[23] Wang Wenbo, Hoang D T, Hu Peizhao, et al. A survey on consensus mechanisms and mining strategy management in blockchain networks[J]. IEEE Access, 2019, 7: 22328−22370

[24] Curry B, Rodeck D. What is Ethereum[EB/OL]. [2023-01-14]. https://www.forbes.com/advisor/investing/cryptocurrency/what-is-ethereum-ether

[25] Liu Yuan, Yu Wangyuan, Ai Zhengpeng, et al. A blockchain-empowered federated learning in healthcare-based cyber physical systems[J]. IEEE Transactions on Network Science and Engineering. 2023, 10(5): 2685−2696

[26] Xiang Jie, Yang Zhemin, Zhou Shunfan, et al. A runtime information based defense technique for Ethereum smart contract[J]. Journal of Computer Research and Development, 2021, 58(4): 834−848 (in Chinese)
(向杰, 杨哲慜, 周顺帆, 等. 一种基于运行时信息的以太坊智能合约防御技术[J]. 计算机研究与发展, 2021, 58(4): 834−848)

[27] Liu Yuan, Xiong Zehui, Hu Qin, et al. VRepChain: A decentralized and privacy-preserving reputation system for social internet of vehicles based on blockchain[J]. IEEE Transactions on Vehicular Technology. 2022, 71(12): 13242−13253
[28] Chriseth, Beregszaszi A, Sliwak K, et al. Solidity, thecontract-oriented programming language[EB/OL]. [2023-01-14]. https://github.com/ethereum/solidity
[29] Griesemer R, Pike R, Thompson K, et al. Build simple, secure, scalable systems with Go[CP/OL]. [2023-01-15]. https://go.dev
[30] Amberdata. Your lens into the entire cryptoeconomy[EB/OL]. [2023-01-15]. https://www.amberdata.io
[31] Mita M, Ito K, Ohsawa S, et al. What is stablecoin? A survey on price stabilization mechanisms for decentralized payment systems[C]//Proc of the 8th Int Congress on Advanced Applied Informatics (IIAI-AAI). Piscataway, NJ: IEEE, 2019: 60−66
[32] Tamplin T. What is tether(USDT)[EB/OL]. [2023-01-15]. https://www.financestrategists.com/wealth-management/cryptocurrency/tether-usdt/?gclid=EAIaIQobChMInoPMwcn2_QIV1DMrCh1PZQ6QEAAYASAAEgK0m_D_BwE
[33] Wikipedia. DAI(cryptocurrency)[EB/OL]. [2023-01-15]. https://en.wikipedia.org/wiki/Dai_(cryptocurrency)
[34] Binance. What is BUSD[EB/OL]. [2023-01-15]. https://academy.binance.com/en/articles/what-is-busd
[35] Vranken H. Sustainability of bitcoin and blockchains[J]. Current opinion in environmental sustainability, 2017, 28: 1−9
[36] Markus B, Palmer J. What is Dogecoin[EB/OL]. [2023-01-15]. https://dogecoin.com/
[37] CoinMarketCap. Cryptos[EB/OL]. [2023-01-15]. https://coinmarketcap.com
[38] Pacheco M, Oliva G A, Rajbahadur G K, et al. Is my transaction done yet? An empirical study of transaction processing times in the Ethereum blockchain platform[J]. ACM Transactions on Software Engineering and Methodology, 2023, 32(3): 1−46
[39] Abdi H. The method of least squares[J]. Encyclopedia of Measurement and Statistics, 2007, 1: 530−532
[40] Tan M, Chuan W, Pauline, et al. DEX activity[EB/OL]. [2023-01-16]. https://etherscan.io/stat/ dextracker
[41] Dune. Dune dashboards[EB/OL]. [2023-01-16]. https://dune.com/browse/dashboards
[42] Barros G, Gallagher P. EIP−1822: Universal upgradeable proxy standard (UUPS) [EB/OL]. [2023-01-16]. https://eips.ethereum.org/EIPS/eip-1822
[43] OpenZeppelin. Securely code, deploy and operate your smart contract[EB/OL]. [2023-01-16]. https://www.openzeppelin.com/
[44] Ethereum Foundation. Go-ethereum[CP/OL]. [2023-01-16]. https://geth.ethereum.org
[45] Lee W M. Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript[M]. New York: Apress, 2019: 169−198
[46] CoinMarketCap. Quotes Historical v2[EB/OL]. [2023-01-16]. https://coinmarketcap.com/api/documentation/v1/#operation/getV2CryptocurrencyQuotesHistorical
[47] Tellor. A decentralized Oracle protocol[EB/OL]. [2023-01-16]. https://tellor.io/whitepaper
[48] Srinawakoon S, Suriyalarn S, Nattapatsiri P. A high-performance blockchain[EB/OL]. [2023-01-16]. https://bandprotocol.com/bandchain



Lin Liansheng, born in 1998. Master. His main research interests include blockchain security and DeFi security.
林炼升, 1998 年生. 硕士. 主要研究方向为区块链安全、DeFi 安全.



Zheng Huanqin, born in 1999. Master, His main research interest includes smart contract security.
郑焕钦, 1999 年生. 硕士. 主要研究方向为智能合约安全.



Su Shen, born in 1985. PhD, professor. Senior member of CCF. His main research interests include blockchain security, DNS security, and routing security.
苏　申, 1985 年生. 博士, 教授, CCF 高级会员. 主要研究方向为区块链安全、DNS 安全、路由安全.



Lei Kai, born in 1976. PhD, professor. Distinguished member of CCF. His main research interests include blockchain network and new network architecture.
雷　凯, 1976 年生. 博士, 教授. CCF 杰出会员. 主要研究方向为区块链网络、新型网络架构.



Chen Xiaofeng, born in 1985. Master, director of standardization and project management department of Qulian Technology Co., Ltd. His main research interests include automated testing, and blockchain technology research and testing.
陈晓丰, 1985 生. 硕士, 趣链科技标准化与课题管理部总监. 主要研究方向为自动化测试、区块链技术研究与测试.



Tian Zhihong, born in 1978. PhD, professor, PhD supervisor. Distinguished member of CCF. Changjiang distinguished professor. His main research interests include network attack and defense competition, network security, and intrusion detection.
田志宏, 1978 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 教育部长江学者特聘教授. 主要研究方向为网络攻防对抗、网络安全、入侵检测.