

专栏首页 区块链安全技术 UniSwap V3协议浅析(下)



# UniSwap V3协议浅析(下)

2021-07-21 阅读 364

## NoDelegateCall

NoDelegateCall合约的主要功能是提供一个修饰器来阻止对使用修饰器修饰过的函数进行delegatecall调用, 合约代码如下:

```
// SPDX-License-Identifier: BUSL-1.1
pragma solidity =0.7.6;

/// @title Prevents delegatecall to a contract
/// @notice Base contract that provides a modifier for prevent
abstract contract NoDelegateCall {
    /// @dev The original address of this contract
    address private immutable original;

    constructor() {
        // Immutables are computed in the init code of the con
        // In other words, this variable won't change when it'
        original = address(this);
    }

    /// @dev Private method is used instead of inlining into m
    /// and the use of immutable means the address bytes a
    function checkNotDelegateCall() private view {
        require(address(this) == original);
    }

    /// @notice Prevents delegatecall into the modified method
    modifier noDelegateCall() {
        checkNotDelegateCall();
        _;
    }
}
```

## UniswapV3Factory

UniswapV3Factory的主要功能是提供创建pool的接口并且跟踪所有的pool, 完整代码如下所示:

```
// SPDX-License-Identifier: BUSL-1.1
pragma solidity =0.7.6;

import './interfaces/IUniswapV3Factory.sol';

import './UniswapV3PoolDeployer.sol';
import './NoDelegateCall.sol';

import './UniswapV3Pool.sol';

/// @title Canonical Uniswap V3 factory
/// @notice Deploys Uniswap V3 pools and manages ownership and
contract UniswapV3Factory is IUniswapV3Factory, UniswapV3PoolD
    /// @inheritdoc IUniswapV3Factory
    address public override owner;
```

## 作者介绍



Al1ex

安全研究员

关注

专栏

文章	阅读量	获赞	作者排名
123	31K	315	1966

## 精选专题

腾讯云原生专题

云原生技术干货, 业务实践落地。

## 活动推荐

一键订阅《云荐大咖》...

获取官方推荐精品内容, 学技术不迷路!

立即查看

腾讯云自媒体分享计划

入驻云加社区, 共享百万资源包。

立即入驻

运营活动

腾讯云 广告

**云产品限时秒杀**

云服务器**1核2G** 首年**38元**

立即抢购

```

mapping(address => uint24) public override feeAmountTickSpacing;
/// @inheritdoc IUniswapV3Factory
mapping(address => mapping(address => mapping(uint24 => ad

constructor() {
    owner = msg.sender;
    emit OwnerChanged(address(0), msg.sender);

    feeAmountTickSpacing[500] = 10;
    emit FeeAmountEnabled(500, 10);
    feeAmountTickSpacing[3000] = 60;
    emit FeeAmountEnabled(3000, 60);
    feeAmountTickSpacing[10000] = 200;
    emit FeeAmountEnabled(10000, 200);
}

/// @inheritdoc IUniswapV3Factory
function createPool(
    address tokenA,
    address tokenB,
    uint24 fee
) external override noDelegateCall returns (address pool) {
    require(tokenA != tokenB);
    (address token0, address token1) = tokenA < tokenB ? (
    require(token0 != address(0));
    int24 tickSpacing = feeAmountTickSpacing[fee];
    require(tickSpacing != 0);
    require(getPool[token0][token1][fee] == address(0));
    pool = deploy(address(this), token0, token1, fee, tick
    getPool[token0][token1][fee] = pool;
    // populate mapping in the reverse direction, delibera
    getPool[token1][token0][fee] = pool;
    emit PoolCreated(token0, token1, fee, tickSpacing, poo
}

/// @inheritdoc IUniswapV3Factory
function setOwner(address _owner) external override {
    require(msg.sender == owner);
    emit OwnerChanged(owner, _owner);
    owner = _owner;
}

/// @inheritdoc IUniswapV3Factory
function enableFeeAmount(uint24 fee, int24 tickSpacing) pu
    require(msg.sender == owner);
    require(fee < 1000000);
    // tick spacing is capped at 16384 to prevent the situ
    // TickBitmap#nextInitializedTickWithinOneWord overflo
    // 16384 ticks represents a >5x price change with tick
    require(tickSpacing > 0 && tickSpacing < 16384);
    require(feeAmountTickSpacing[fee] == 0);

    feeAmountTickSpacing[fee] = tickSpacing;
    emit FeeAmountEnabled(fee, tickSpacing);
}
}

```

合约最初声明当前工程合约的owner、费用的TrickSpacing(刻度间距, 如果为0则表示未启用, 一经添加无法删除)、根据token与fee检索交易池的映射:

```

/// @inheritdoc IUniswapV3Factory
address public override owner;

/// @inheritdoc IUniswapV3Factory
mapping(uint24 => int24) public override feeAmountTickSpac

```

之后通过构造函数初始化的合约的owner以及三个TickSpace:

```

constructor() {
  owner = msg.sender;
  emit OwnerChanged(address(0), msg.sender);

  feeAmountTickSpacing[500] = 10;
  emit FeeAmountEnabled(500, 10);
  feeAmountTickSpacing[3000] = 60;
  emit FeeAmountEnabled(3000, 60);
  feeAmountTickSpacing[10000] = 200;
  emit FeeAmountEnabled(10000, 200);
}

```

上面定义的三个TickSpace与费率的关系如下所示:

之后通过createPool来创建交易池,此时需要提供以下三个参数:

- fee:期望的费率
- tokenA: 交易池中的两个Token之一
- tokenB: 交易池中的两个Token之一

在createPool函数中首先会检查tokenA与tokenB是否是同一Token,之后将TokenA与TokenB根据地址进行升序排列,之后检查token0地址是否为空地址,之后根据费率检索TickSpace并检查TickSpace是否为0(构造函数会进行初始化一次),之后检查当前新建的池子是否已经存在,之后通过deploy创建池子,然后新增池子记录,在新增记录时可以看到也提供了反向映射,这样做的好处是在减少后期检索时比较地址的成本,最后通过emit触发池子创建事件

```

/// @inheritdoc IUniswapV3Factory
function createPool(
  address tokenA,
  address tokenB,

```

[展开阅读全文](#)

举报

点赞 3

分享

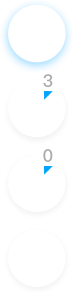
0 条评论

我来说两句

[登录](#) 后参与评论

### 相关文章

UniSwap V3协议浅析(上)



Altex

### Uniswap V3 介绍

Uniswap v1 于 2018 年 11 月推出，主要是验证自动做市商 (AMM) 的概念，是一种任何人都可以将资产汇集到资金池做市策略的去中心化交易所。

Tiny熊

### Instadapp 治理已经启动!

我们在三年前构建了 Instadapp，以使 DeFi 更易于访问和使用。从那时起，Instadapp 已成为 DeFi 基础设施的关键任...

公众号---人生代码

### 剖析DeFi交易产品之Uniswap: V2上篇

在 DeFi 赛道中，DEX 无疑是最核心的一块，而 Uniswap 又是整个 DEX 领域中的龙头，如 SushiSwap、PancakeSwap 等都是 F...

Keegan小钢

### Uniswap V3 路径编码的进一步优化

前几天群里有讨论 Uniswap V3[2] 中询价的处理，简单翻了下代码，发现与 Uniswap V2[3] 相比，V3 变化真的很大~

Tiny熊

### 使用TheGraph 获取Uniswap数据(c#)

Uniswap 是一个建立在以太坊上的去中心化协议，用户可以交换 ERC-20 代币，不需要买家和卖家创造需求。它是最受欢迎的去中心化交易所(DEX)，在撰写本...

Tiny熊

### 协议分析|HTTP协议浅析

HTTP和HTTPS Cookie、Session HTTP请求方法 (Get/Post) 一次HTTP请求的过程 一次请...

Ms08067安全实验室

### 协议分析|HTTP协议浅析

HTTP和HTTPS Cookie、Session HTTP请求方法 (Get/Post) 一次HTTP请求的过程 一次请...

谢公子

### Defi? Uniswap 项目漏洞教程新骗局

昨晚突然看到群里的一个消息，揭秘uniswap-defi项目漏洞-割韭菜新手法，心想还有这事？而且还是中英文介绍。

Seebug漏洞平台

UniSwap V3 开发包适用于为 Web 前端或 Node.js 应用快速增加对 UniSwap 协议的支持能力。即支持使用自有部署以太坊区块链节点的应用场景，也支持使用第三方节点的...

用户5687508

### JavaScript对接Uniswap开发教程

Uniswap.js 开发包适用于为 Web 前端或 Node.js 应用快速增加对 UniSwap 协议的支持能力。即支持使用自有部署以太坊区块链节点的应用场景，也支持使...

用户5687508

### 爬虫 | 浅析HTTP协议

爬虫就是模拟浏览器向服务器发送请求，然后获得相应的过程。而在这种通信过程中，HTTP协议起到至关重要的作用...

bugsume

### 浅析 HTTPS 和 SSL/TLS 协议

1.ssl协议：通过认证、数字签名确保完整性；使用加密确保私密性；确保客户端和服务端之间的通讯安全 2.tls协议：...

一个会写诗的程序员

### 浅谈TCP IP协议栈(四)IP协议解析

通过之前的网络层基础知识，IP地址以及路由器的简介，大家应该对于TCP/IP有一个大致的了解，在脑海里应该对于网络的几个基础概念有个大概的了解，简单点说整个...

233333

### [ffffff0x] 工控协议:S7COMM协议分析(下)

在上一篇文章中，我们介绍了S7Comm协议的S7Comm Header和Job 和 Ack\_Data机制。本篇文章，我们将继续...

r0fus0d

### 10 大趋势带你预见 DeFi 2020!

本文是加密货币研究平台 Messair 近日发布的《Crypto Theses for 2020》报告的节选部分，将探讨 DeFi...

区块链大本营

### 浅析 HLS 流媒体协议

在最近工作中接触到了视频点播和直播业务，也了解到了一些流媒体的后端技术，这段时间希望将了解到的一些知识总结下来，这篇文章主要介绍 HLS 流媒体协议 的基础...

高楼Zee

### 基于Flash Loan的套利软件开发教程【一】

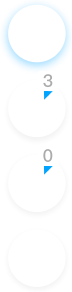
套利 (Arbitrage)、智能合约开发 (Contract Coding) 和乐观转账 (Optimistic Transfer) 共同创造了一个令人印象深刻的新工具...


用户5687508

现在是2022年。为应对covid-19经济衰退而采取的量化宽松政策已导致眼镜蛇效应,全世界已经陷入了更严重的衰退。...

 Tiny熊

[更多文章](#)



社区	活动	资源	关于	云+社区
专栏文章	原创分享计划	技术周刊	视频介绍	 扫码关注云+社区 领取腾讯云代金券
阅读清单	自媒体分享计划	社区标签	社区规范	
互动问答	邀请作者入驻	开发者实验室	免责声明	
技术沙龙	自荐上首页		联系我们	
技术快讯	在线直播		友情链接	
团队主页	生态合作计划			
开发者手册				
腾讯云TI平台				

热门产品	域名注册	云服务器	区块链服务	消息队列	网络加速	云数据库	域名解析
	云存储	视频直播					
热门推荐	人脸识别	腾讯会议	企业云	CDN 加速	视频通话	图像分析	MySQL 数据库
	SSL 证书	语音识别					
更多推荐	数据安全	负载均衡	短信	文字识别	云点播	商标注册	小程序开发
	网站监控	数据迁移					