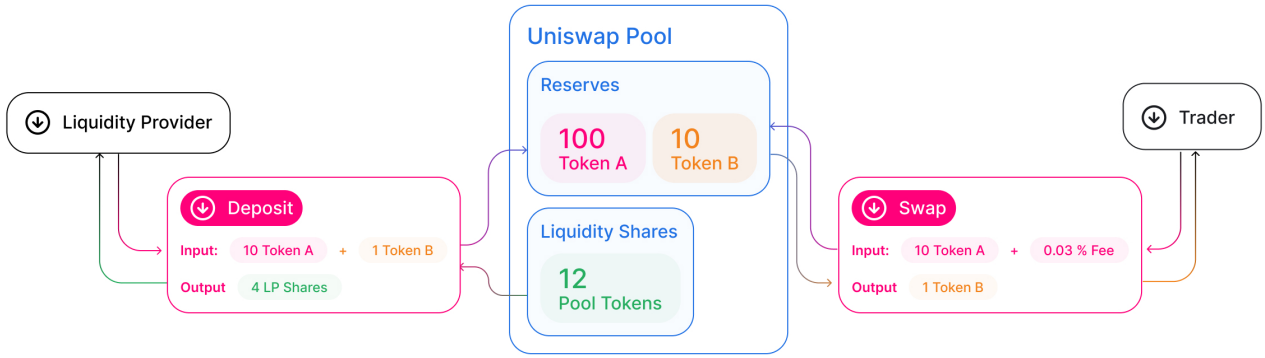
uniswap学习

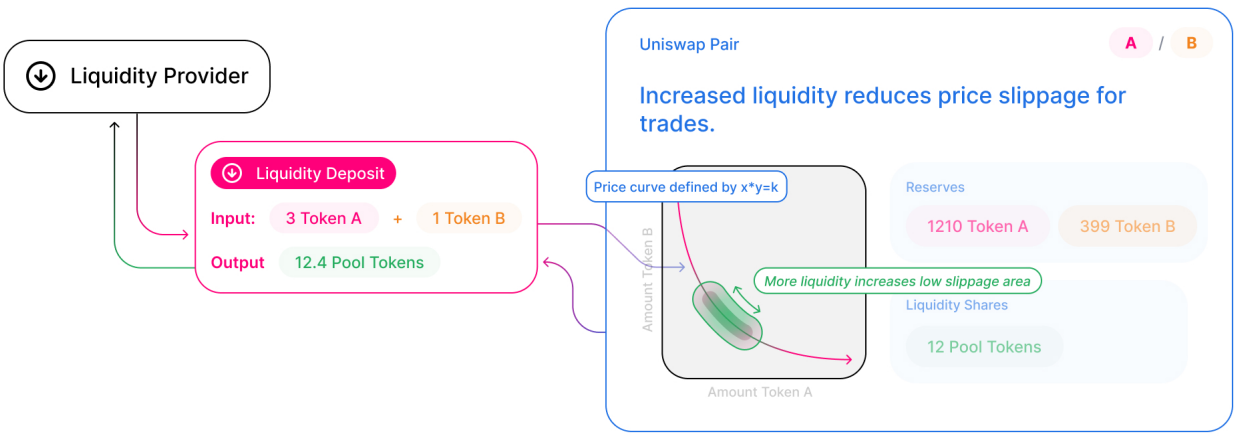
一、Uniswap 的工作原理



Uniswap 是一种自动流动性协议，由恒定产品公式提供支持，并在以太坊区块链 上的不可升级智能合约系统中实施。它消除了对可信中介的需求，优先考虑去中心化、抗审查和安全。Uniswap是根据GPL许可 的开源软件。

每个 Uniswap 智能合约或配对都管理一个由两个ERC-20代币储备组成的流动资金池。

任何人都可以通过存入每个基础代币的等值来换取池代币，从而成为池的流动性提供者 (LP)。这些代币跟踪总储备中按比例的 LP 份额，并且可以随时赎回基础资产。

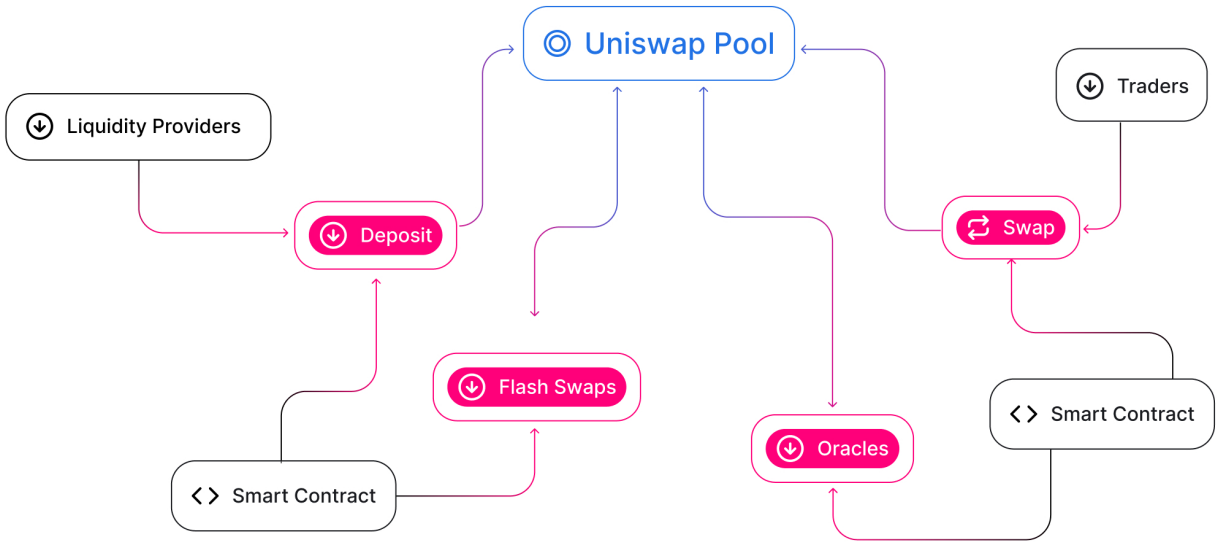


货币对充当自动做市商，只要保留“恒定乘积”公式，就准备好接受一个代币换另一个代币。该公式最简单地表示为x * y = k，表明交易不得改变k—对储备余额 (x和) 的乘积 (y)。因为k与交易的参考框架保持不变，它通常被称为不变量。该公式具有理想的特性，即较大的交易（相对于储备）以比较小的交易更差的指数执行。

在实践中，Uniswap 对交易收取 0.30% 的费用，该费用被添加到准备金中。结果，每笔交易实际上都在增加k。这起到了向 LP 支付的作用，这是在他们烧掉他们的池代币以提取他们在总储备中的一部分时实现的。将来，此费用可能会降低到 0.25%，剩余的 0.05% 将作为协议范围内的费用扣留。

由于两对资产的相对价格只能通过交易来改变，Uniswap 价格与外部价格的背离创造了套利机会。这种机制确保 Uniswap 价格始终趋向于市场出清价格。

生态系统参与者



Uniswap 生态系统主要由三类用户组成：流动性提供者、交易者和开发者。流动性提供者被激励向公共流动性池贡献ERC-20代币。交易者可以以固定的0.30% 的费用将这些代币相互交换（这将支付给流动性提供者）。开发人员可以直接与 Uniswap 智能合约集成，以支持与代币、交易界面、零售体验等的新的和令人兴奋的交互。

总的来说，这些类别之间的互动创造了一个积极的反馈循环，通过定义一种可以汇集、交易和使用代币的通用语言来推动数字经济。

1. 流动性提供者

流动性提供者或 LP 不是同质群体：

- 被动 LP 是希望被动投资资产以积累交易费用的代币持有者。
- 专业的有限合伙人专注于做市作为他们的主要策略。他们通常开发定制工具和方法来跟踪他们在不同 DeFi 项目中的流动性头寸。

- 代币项目有时会选择成为 LP 为其代币创建一个流动的市场。这使得代币买卖更容易，并通过 Uniswap 解锁与其他 DeFi 项目的互操作性。
- 最后，一些 DeFi 先驱正在探索复杂的流动性提供互动，例如激励流动性、流动性作为抵押品以及其他实验性策略。Uniswap 是项目尝试这些想法的完美协议。

2. 交易员

协议生态系统中有几类交易者：

- 投机者使用各种社区构建的工具和产品，利用从 Uniswap 协议中提取的流动性来交换代币。
- 套利机器人通过比较不同平台的价格以寻找优势来寻求利润。（虽然看起来很榨取，但这些机器人实际上有助于平衡更广泛的以太坊市场的价格并保持公平。）
- DAPP 用户在 Uniswap 上购买代币，用于以太坊上的其他应用程序。
- 通过实现交换功能（从 DEX 聚合器等产品到自定义 Solidity 脚本）在协议上执行交易的智能合约。

在所有情况下，根据协议进行的交易均需支付相同的固定费用。每一项对于提高价格的准确性和激励流动性都很重要。

3. 开发商/项目

在更广泛的以太坊生态系统中使用 Uniswap 的方式太多了，但一些例子包括：

- Uniswap 的开源、可访问性意味着有无数的 UX 实验和前端旨在提供对 Uniswap 功能的访问。您可以在大多数主要的 DeFi 仪表板项目中找到 Uniswap 功能。社区还构建了许多特定于 Uniswap 的工具。
- 钱包通常将交换和流动性提供功能作为其产品的核心产品。
- DEX（去中心化交易所）聚合器从许多流动性协议中提取流动性，通过拆分交易为交易者提供最优惠的价格。Uniswap 是这些项目最大的单一去中心化流动性来源。

智能合约开发人员使用可用的功能套件来发明新的 DeFi 工具和其他各种实验性想法。查看Unisocks或Zora等项目，其中包括许多其他项目。

4. Uniswap 团队和社区

Uniswap 团队与更广泛的 Uniswap 社区一起推动协议和生态系统的发展。

二、智能合约

Uniswap V2 是一个二进制智能合约系统。核心合约为与 Uniswap 交互的各方提供基本的安全保障。外围合约与一个或多个核心合约交互，但它们本身不是核心的一部分。

2.1 Core合约

合约代码github仓库地址

factory参考

pair参考

pair-erc-20参考

Core由一个单例factory和许多pair组成，factory负责创建和索引。这些合同非常少，甚至是野蛮的。这样做的简单理由是，具有较小表面积的合同更容易推理，更不容易出错，并且在功能上更优雅。也许这种设计的最大优点是系统的许多所需属性可以直接在代码中声明，几乎没有出错的余地。然而，一个缺点是核心合约在某种程度上对用户不友好。事实上，对于大多数用例，不建议直接与这些合约交互。相反，应该使用外围合约。

货币pair有两个主要目的：充当自动做市商和跟踪池代币余额。它们还公开可用于构建去中心化价格预言机的数据。

2.2 外围合约

合约代码github仓库地址

库参考

router参考

外围是一组智能合约，旨在支持与Core的特定领域交互。由于 Uniswap 的无许可性质，下面描述的合约没有特权，实际上只是可能的外围类合约宇宙的一小部分。但是，它们是如何安全有效地与 Uniswap V2 交互的有用示例。

关于router：使用该库的路由器完全支持前端提供交易和流动性管理功能的所有基本要求。值得注意的是，它原生支持多对交易（例如 x 到 y 到 z），将 ETH 视为一等公民，并提供用于消除流动性的元交易。

2.3 设计决策

1. sending-tokens

通常，需要代币来执行某些功能的智能合约要求潜在的交互者首先对代币合约进行批准，然后调用一个函数，该函数又调用代币合约上的 transferFrom。这不是V2 对接受令牌的方式。相反，配对在每次交互结束时检查他们的代币余额。然后，在下一次交互开始时，当前余额与存储的值不

同，以确定当前交互者发送的代币数量。请参阅白皮书以了解为什么会出现这种情况的理由，但要点是在调用任何需要令牌的方法之前必须将令牌转移到该货币对（此规则的一个例外是Flash Swaps。

V2白皮书

2. WETH

与 Uniswap V1 矿池不同，V2 对不直接支持 ETH，因此必须用 WETH 模拟 ETH⇄ERC-20 对。这种选择背后的动机是删除核心中特定于 ETH 的代码，从而产生更精简的代码库。然而，最终用户可以完全不了解这个实现细节，只需在外围包装/解包 ETH。

路由器完全支持通过 ETH 与任何 WETH 对进行交互。

3. Minimum Liquidity

为了改善舍入误差并增加流动性提供的理论最小刻度大小，货币对会烧掉第一个MINIMUM_LIQUIDITY池代币。对于绝大多数对，这将代表一个微不足道的价值。在第一次提供流动性期间，销毁会自动发生，之后总供应量将永远受到限制。

三、主题

3.1 费用

流动性提供者费用：

交换代币需要0.3%的费用。该费用由流动性提供者按其对流动性储备的贡献比例分配。 掉期费用立即存入流动性储备。这增加了流动性代币的价值，作为对所有流动性提供者的支付，与其在池中的份额成正比。费用通过燃烧流动性代币来收取一定比例的基础储备金。

由于费用被添加到流动性池中，因此在每笔交易结束时不变量都会增加。在单个事务中，不变量表示token0_pool / token1_pool上一个事务的结束。

有许多社区开发的工具来确定回报。您还可以在文档中阅读有关如何考虑LP 回报的更多信息。

协议费用：

目前没有协议费用。但是，将来可能会开启 0.05% 的费用。 未来，每笔交易 0.05% 的协议范围收费可能会生效。这代表 0.30% 费用的 ⅙ (16.6̄%)。如果feeTo不是address(0)(0x0000000000000000000000000000000000000000)，则费用有效，表明 feeTo 是费用的接收方。该金额不会影响交易者支付的费用，但会影响流动性提供者收到的金额。

不是计算掉期费用，这会显着增加所有用户的 gas 成本，而是在增加或删除流动性时计算费用。有关更多详细信息，请参阅白皮书。

3.2 价格是如何确定的?

正如我们在协议概述中了解到的，Uniswap 上的每一对实际上都由一个流动资金池支撑。流动资金池是智能合约，持有两种独特代币的余额，并执行有关存取它们的规则。主要规则是常数乘积公式。当提取（购买）代币时，必须存入（出售）一定比例的金额以保持不变。池中代币的比率，结合恒定乘积公式，最终决定了交换执行的价格。

Uniswap 如何处理价格：

在 Uniswap V1 中，交易始终以“可能的最佳”价格执行，在执行时计算。有点令人困惑的是，这个计算实际上是用两个不同的公式之一完成的，这取决于交易是否指定了准确的输入或输出量。在功能上，这两个功能之间的差异是微乎其微的，但差异的存在增加了概念的复杂性。在 V2 中支持这两个功能的最初尝试被证明是不优雅的，因此决定不在核心中提供任何定价功能。相反，对在每次交易后直接检查不变量是否得到满足（考虑费用）。这意味着，与其依赖定价功能强制执行不变量，V2 对简单而透明地确保它们自己的安全，很好地分离关注点。一个下游好处是 V2 对将更自然地支持可能出现的其他类型的交易（例如在执行时以特定价格交易）。

在高层次上，在 Uniswap V2 中，交易必须在外围定价。好消息是，库 提供了各种旨在简化此操作的功能，并且路由器中的所有交换功能都考虑到了这一点。

定价交易：

在 Uniswap 上交换代币时，通常希望接收尽可能多的输出代币以获得准确的输入金额，或者支付尽可能少的输入代币以获得准确的输出金额。为了计算这些金额，合约必须查看货币对的当前储备，以了解当前价格是多少。但是，执行此查找并依赖结果而不访问外部价格是不安全的。

假设一个智能合约天真地想向 DAI/WETH 对发送 10 个 DAI，并在给定当前准备金率的情况下接收尽可能多的 WETH。如果在调用时，简单的智能合约只是查看当前价格并执行交易，它很容易受到抢先交易的影响，并且可能会遭受经济损失。要了解原因，请考虑在确认交易之前看到该交易的恶意行为者。他们可以在天真的掉期通过之前立即执行一个大幅改变 DAI/WETH 价格的掉期，等待天真的掉期以低利率执行，然后进行交换以将价格改回天真的掉期之前的价格。这种攻击相当便宜且风险低，通常可以为盈利而执行。

为了防止这些类型的攻击，提交能够了解其交换应执行的“公平”价格的交换至关重要。换句话说，交换需要访问预言机，以确保他们可以从 Uniswap 获得的最佳执行与预言机认为的“真实”价格足够接近。虽然这听起来很复杂，但预言机可以像对当前市场价格的链下观察一样简单. 由于套利，通常情况下，一对区块内储备的比率接近“真实”市场价格。因此，如果用户在提交交易时考虑到这一点，他们可以确保由于抢先交易而造成的损失是有限的。例如，这就是 Uniswap 前端确保交易安全的方式。它计算给定观察到的块内价格的最佳输入/输出量，并使用路由器来执行交换，这保证交换将以不低于x观察到的块内速率差 % 的速率执行，其中x用户- 指定的滑点容差（默认为 0.5%）。

当然，预言机还有其他选项，包括原生 V2 预言机。

3.3 理解Returns

Uniswap 通过奖励提供者与其他用户与这些池进行交易时产生的费用来激励用户为交易池增加流动性。一般来说，做市是一项复杂的活动。与单纯持有资产相比，在标的资产价格大幅持续波动期间存在亏损风险。

风险 要了解与提供流动性相关的风险，您可以阅读https://medium.com/@pintail/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2以深入了解如何概念化流动性位置。

文章中的例子 考虑流动性提供者将 10,000 DAI 和 100 WETH 添加到池中（总价值为 20,000 美元）的情况，流动性池现在总共是 100,000 DAI 和 1,000 ETH。由于提供的数量等于总流动性的 10%，合约铸造并向做市商发送“流动性代币”，使他们有权获得池中可用流动性的 10%。这些不是要交易的投机代币。它们只是一种会计或簿记工具，用于跟踪流动性提供者的欠款。如果其他人随后添加/取出硬币，则铸造/销毁新的流动性代币，以使每个人在流动性池中的相对百分比份额保持不变。

现在让我们假设 Coinbase 上的交易价格从 100 美元到 150 美元。Uniswap 合约在一些套利之后也应该反映这种变化。交易者将添加 DAI 并移除 ETH，直到新的比率现在为 150:1。

流动性提供者会怎样？该合约反映了接近 122,400 DAI 和 817 ETH 的东西（为了检查这些数字是否准确，122,400 * 817 = 100,000,000（我们的常数产品）和 122,400 / 817 = 150，我们的新价格）。提取我们有权获得的 10% 现在将产生 12,240 DAI 和 81.7 ETH。这里的总市值为 24,500 美元。由于做市，错过了大约 500 美元的利润。

显然，没有人愿意通过慈善方式提供流动性，而且收入并不依赖于从良好交易中获得回报的能力（没有翻转）。相反，所有交易量的 0.3% 按比例分配给所有流动性提供者。默认情况下，这些费用会被放回流动资金池，但可以随时收取。如果不知道中间交易的数量，就很难知道费用收入和定向运动损失之间的权衡是什么。来回砍的越多越好。

为什么我的流动性比我投入的少 要了解为什么流动性提供者的股份价值会下降，尽管有手续费收入，我们需要更仔细地研究 Uniswap 用于管理交易的公式。公式真的很简单。如果我们忽略交易费用，我们有以下几点：

eth_liquidity_pool * token_liquidity_pool = constant_product 换句话说，交易者收到的 ETH 代币数量（反之亦然）是这样计算的，即交易后，两个流动性池的乘积与交易前相同。这个公式的结果是，对于与我们拥有的流动资金池规模相比价值非常小的交易：

eth_price = token_liquidity_pool / eth_liquidity_pool 结合这两个等式，假设总流动性恒定，我们可以计算出在任何给定价格下每个流动性池的规模：

eth_liquidity_pool = sqrt(constant_product / eth_price) token_liquidity_pool = sqrt(constant_product * eth_price) 因此，让我们看看价格变化对流动性提供者的影响。为简单起见，假设我们的流动性提供者向 Uniswap DAI 交易所提供 1 ETH 和 100 DAI，给他们 1% 的流

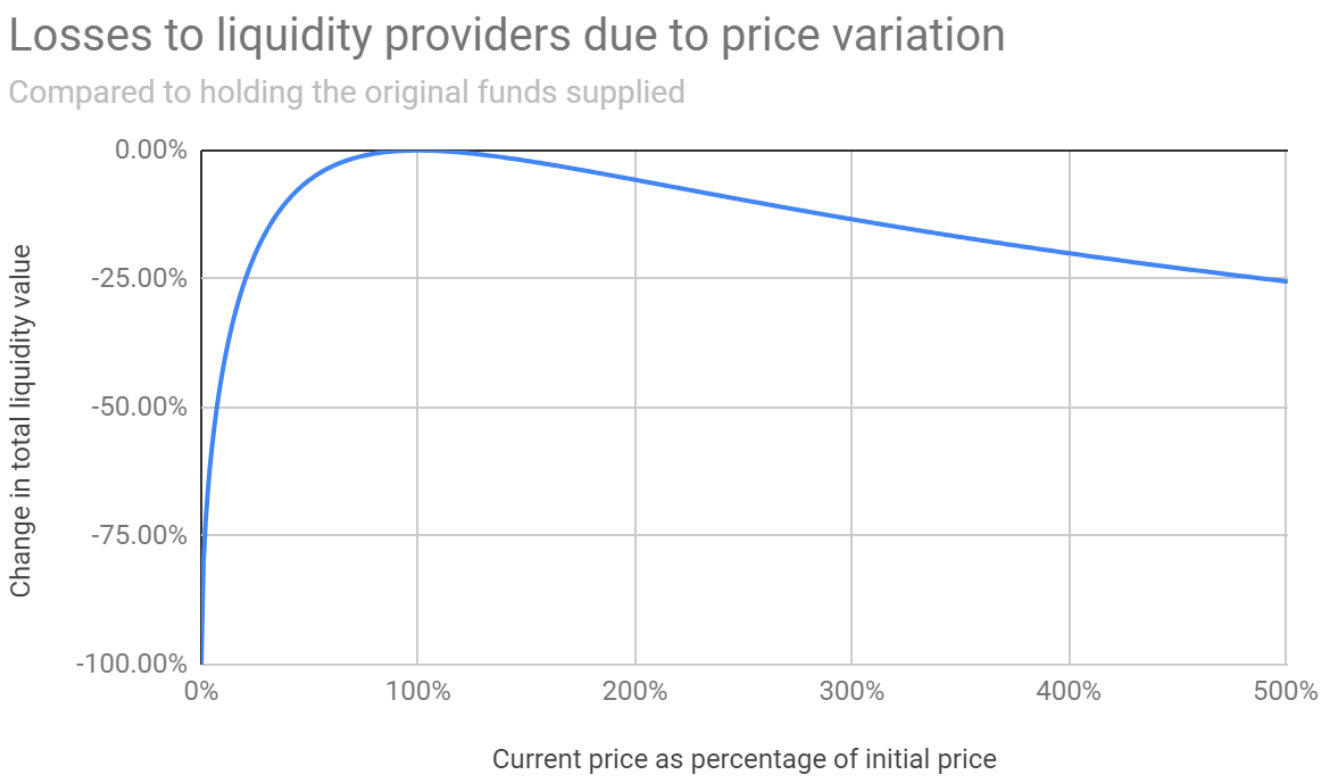
动性池，其中包含 100 ETH 和 10,000 DAI。这意味着 1 ETH = 100 DAI 的价格。还是忽略手续费，我们想象一下，经过一些交易，价格发生了变化；1 ETH 现在价值 120 DAI。流动性提供者股权的新价值是多少？将数字代入上面的公式，我们有：

eth_liquidity_pool = 91.2871 dai_liquidity_pool = 10954.4511 “由于我们的流动性提供者拥有 1% 的流动性代币，这意味着他们现在可以从流动性池中索取 0.9129 ETH 和 109.54 DAI。但由于 DAI 大约相当于美元，我们可能更愿意将全部金额转换为 DAI 来理解价格变化的整体影响。在当前价格下，我们的流动性总共价值 219.09 DAI。如果流动性提供者刚刚持有他们原来的 1 ETH 和 100 DAI 怎么办？好吧，现在我们可以很容易地看到，以新价格计算，总价值为 220 DAI。因此，我们的流动性提供者通过向 Uniswap 提供流动性而不是仅仅持有其初始 ETH 和 DAI，损失了 0.91 DAI。

“当然，如果价格回到流动性提供者添加流动性时的相同值，这种损失就会消失。因此，我们可以称之为 无常损失。使用上面的等式，我们可以推导出一个公式以提供流动性时与现在之间的价格比率来衡量无常损失的大小。我们得到以下结果：

" impermanent_loss = 2 * sqrt(price_ratio) / (1+price_ratio) — 1"

“我们可以绘制出来，以大致了解不同价格比率下无常损失的规模：”



“或者换一种说法：”

“相对于 HODL，1.25 倍的价格变化导致 0.6% 的损失” “相对于 HODL，1.50 倍的价格变化导致 2.0% 的损失” “相对于 HODL，1.75 倍的价格变化导致 3.8% 的损失” “相对于 HODL，2 倍的价格

变化导致 5.7% 的损失” “相对于 HODL，3 倍的价格变化导致 13.4% 的损失” “相对于 HODL，4 倍的价格变化导致 20.0% 的损失” “5 倍的价格变化导致相对于 HODL 损失 25.5%” “注意，无论价格变化发生在哪个方向，损失都是相同的（即价格翻倍导致与减半相同的损失）。” -->

3.4 安全

审计和形式验证

在 1 月 8 日至 4 月 30 日期间，一个由六名工程师组成的团队审查并正式验证了 Uniswap V2 智能合约的关键组件。

他们过去的工作包括多抵押 DAI 的智能合约开发和形式验证。

工作范围包括：

- 核心智能合约的形式化验证
- 核心智能合约的代码审查
- 数值误差分析
- 外围智能合约的代码审查（在持续开发中） 该报告还有一个“设计评论”部分，我们强烈建议您深入了解 Uniswap V2 中的某些选择。

漏洞赏金

Uniswap 有一个开放且持续的漏洞赏金计划。

在 Uniswap 上构建时的注意事项

将 Uniswap V2 集成到另一个链上系统时，必须特别注意避免安全漏洞、操纵途径和潜在的资金损失。

作为初步说明：智能合约集成可以发生在两个级别：直接使用Pair合约，或通过Router。直接交互提供了最大的灵活性，但需要做最多的工作才能做到正确。中介交互提供了更有限的能力，但更强大的安全保障。

Uniswap V2 有两种主要的风险类别。第一个涉及所谓的“静态”错误。这些可能包括在交换期间向一对发送过多的代币（或请求返回的代币太少）或允许交易在内存池中停留足够长的时间以使发送者对价格的预期不再准确。

可以通过相当简单的逻辑检查来解决这些错误。执行这些逻辑检查是路由器的主要目的。那些直接与配对交互的人必须自己执行这些检查（在图书馆的帮助下。

第二类“动态”风险涉及运行时定价。由于以太坊交易发生在对抗性环境中，天真编写的智能合约可以而且将会被利用来获取利润。例如，假设智能合约在运行时检查 Uniswap 池中的资产比率并进行交易，假设该比率代表这些资产的“公平”或“市场”价格。在这种情况下，它很容易受到操纵。

例如，恶意行为者可以在幼稚交易之前和之后简单地插入交易（所谓的“三明治”攻击），导致智能合约以更差的价格进行交易，以牺牲交易者为代价从中获利，然后返回以低廉的成本将合同恢复到原始状态。

防止这些攻击的最佳方法是引入价格预言机。预言机是返回所需信息的任何设备，在这种情况下，是一对现货价格。最好的“预言机”只是交易者对当前价格的链下观察，可以作为安全检查传递到交易中。这种策略最适合用户代表自己发起交易的零售交易场所。然而，通常情况下，可信的价格观察不可用（例如，在涉及 Uniswap 的多步骤、程序化交互中）。如果没有价格预言机，这些交互将被迫以 Uniswap 上的（可能被操纵的）汇率进行交易。有关预言机的 Uniswap V2 方法的详细信息，请参阅预言机。

附录资源

1. uniswap官方有关资料：

- 官方介绍

https://uniswap.org/blog/uniswap-v2/

https://uniswap.org/blog/uniswap-v3/

- 技术白皮书

https://docs.uniswap.org/whitepaper.pdf

https://uniswap.org/whitepaper-v3.pdf

- 智能合约代码

https://github.com/Uniswap/uniswap-v3-core

https://github.com/Uniswap/uniswap-v3-periphery

https://github.com/Uniswap/uniswap-v2-core

https://github.com/Uniswap/uniswap-v2-periphery

- Protocol

https://docs.uniswap.org/protocol/V2/introduction

https://docs.uniswap.org/protocol/introduction

- 其他

https://research.paradigm.xyz/uniswaps-alchemy

2. 相关Uniswap术语

- Uniswap Labs：负责开发Uniswap协议、网络接口的公司。
- The Uniswap Protocol：一个实现自动化做市商的智能合约全家桶，促进点对点做市和以太坊上ERC-20 token的交易的协议（即Uniswap核心技术，后续工作原理介绍也都是针对协议内容的解释）。
- The Uniswap Interface：为了方便使用Uniswap protocol而开发的网络接口，是与Uniswap protocol交互的众多方式之一（也可以直接与智能合约交互）。
- Uniswap Governance：一个Uniswap Protocol的民主治理系统（社区式治理方式，论坛模式）。

3.词汇表

- **Automated market maker** An automated market maker is a smart contract on Ethereum that holds on-chain liquidity reserves. Users can trade against these reserves at prices set by an automated market making formula.
- **Constant product formula** The automated market making algorithm used by Uniswap. See x*y=k.
- **ERC20** ERC20 tokens are fungible tokens on Ethereum. Uniswap supports all standard ERC20 implementations.
- **Factory** A smart contract that deploys a unique smart contract for any ERC20/ERC20 trading pair.
- **Pair** A smart contract deployed from the Uniswap V2 Factory that enables trading between two ERC20 tokens.
- **Pool** Liquidity within a pair is pooled across all liquidity providers.
- **Liquidity provider / LP** A liquidity provider is someone who deposits an equivalent value of two ERC20 tokens into the liquidity pool within a pair. Liquidity providers take on price risk and are compensated with fees.
- **Mid price** The price between what users can buy and sell tokens at a given moment. In Uniswap this is the ratio of the two ERC20 token reserves.
- **Price impact** The difference between the mid-price and the execution price of a trade.
- **Slippage** The amount the price moves in a trading pair between when a transaction is submitted and when it is executed.

- **Core** Smart contracts that are essential for Uniswap to exist. Upgrading to a new version of core would require a liquidity migration.
- **Periphery** External smart contracts that are useful, but not required for Uniswap to exist. New periphery contracts can always be deployed without migrating liquidity.
- **Flash swap** A trade that uses the tokens being purchased before paying for them.

x * y = k The constant product formula.

- **Invariant** The "k" value in the constant product formula