Manipulating Uniswap v3 TWAP Oracles

Michael Bentley

November 5, 2021

1 Introduction

To manipulate a geometric mean time-weighted average price (TWAP) on Uniswap v3, an attacker needs to manipulate the spot price for at least 1 block.

The scale of the manipulation depends on the deviation they can achieve between the natural spot price and the manipulated spot price, and how many blocks they can hold the price manipulation for.

The cost of the attack depends on how much existing liquidity sits in between the current spot price and the manipulated spot price, the extent to which arbitrageurs can revert the attacker's price manipulation, the ability of the attacker to back-run their own manipulation, and the cost of persuading miners to include their attack in the blockchain without arbitraging it themselves.

The cost of a manipulation can be estimated with a high degree of accuracy if we assume that there is at least some liquidity in a Uniswap v3 pool distributed across the entire price range. This allows us to use the constant-product formula from Uniswap v2 to derive exact solutions for the cost of a manipulated price on each block.

Here, we use this approach to provide estimates for the cost of a modest 5x price manipulation of the DAI-USDC TWAP over a window of 30 minutes. We use beneficial assumptions from the perspective of the attacker, but show that even a little liquidity in the range can push the cost of an attack to extraordinary levels.

We assume that the attacker can hold the price of the manipulation for 10 blocks, and that there is only $\$50,000$ worth of liquidity in the pool to begin with. We find that a naive attacker needs to perform a spot price manipulation of DAI from \$1 to around \$11.6b over 10 blocks, at a slippage/arbitrage cost of around \$2.7b per manipulated block. A sophisticated attacker can hope to reduce these costs somewhat, but must still expose themselves to a significant level of risk to manipulate an illiquid DAI-USDC pair by a modest amount.

1.1 How to Manipulate a Uni v3 TWAP Oracle

To manipulate a geometric mean TWAP requires an attacker to manipulate the spot price on at least one block within the TWAP window. The fewer blocks they are able to manipulate, the more aggressively they must manoeuvre the spot price in a single block in order to impact the average price given by the TWAP.

Consider the DAI-USDC price. The spot price at block i is denoted p_i. The geometric mean time-weighted average price (TWAP) can be calculated over n blocks as the nth root of the product of the spot price on each block:

$$\text{TWAP} = \left(\prod_{i=1}^{n} p_i \right)^{\frac{1}{n}}. \tag{1}$$

Now consider the effect of an attacker on the TWAP. Suppose the spot price on each block is manipulated to some target q, for a period of m blocks, and is otherwise some constant spot price p for the remaining $n - m$ blocks. Then the TWAP is calculated as

$$\text{TWAP} = \left(p^{(n-m)} \cdot q^m\right)^{\frac{1}{n}}. \tag{2}$$

An attacker wanting to manipulate the TWAP to some particular oracle price over m blocks will need to know what spot price q they need to move the normal spot price p to in each of those blocks. We can calculate this by simply rearranging equation (2) to get

$$q = \sqrt[m]{\frac{\text{TWAP}^n}{p^{(n-m)}}}. \tag{3}$$

This equation shows that it is surprisingly difficult to move the geometric mean TWAP from the natural spot price when manipulated blocks are few in number relative to unmanipulated blocks. That is, the spot price must be moved a significant distance from its natural price in order to have even a modest impact on the TWAP.

1.2 Examples

To help make this all clear, let us give a couple of examples. Let us assume that the Uniswap TWAP is calculated over $n = 144$ blocks (roughly a 30 minute window), and that the attacker feels that they can manipulate the spot price for $m = 10$ of those blocks (quite optimistic). We will take the starting spot price before the attack to be $p = 1$ USDC/DAI.

First, suppose that attacker wants to move the oracle price to TWAP = 5 USDC/DAI (i.e. a fairly significant increase in price of 5x). Then the spot price they need to manipulate the DAI price to in each block they control is

$$q_{\text{up}} = \sqrt[10]{\frac{(5 \text{ USDC/DAI})^{144}}{(1 \text{ USDC/DAI})^{(144-10)}}} \approx 11,618,981,559 \text{ USDC/DAI}. \tag{4}$$

Such a price could only really be feasibly achieved if the there was zero liquidity in the range (p, q). Even small amounts of liquidity in this range would almost certainly make the cost of the attack too expensive to perform.

Second, suppose that attacker wants to move the oracle price to TWAP = 0.8 USDC/DAI (i.e. a reasonably small decrease in price of 20%). Then the spot price they need to manipulate the DAI price to in each block they control is

$$q_{\text{down}} = \sqrt[10]{\frac{(0.8 \text{ USDC/DAI})^{144}}{(1 \text{ USDC/DAI})^{(144-10)}}} \approx 0.040224978 \text{ USDC/DAI}. \tag{5}$$

This price could perhaps be feasibly achieved even if there were some liquidity in the range (p, q), but it is still quite a large manipulation to have to make over 10 blocks to achieve only a 20% decrease in price.

We can calculate exactly how expensive these attacks might be using the mathematics of the constant-product formula from Uniswap v2.

1.3 Cost of Manipulating a Uni v3 TWAP Oracle

We continue by assuming we are working with a DAI-USDC pool. The spot DAI-USDC price is defined as $p = x_{\text{USDC}}/x_{\text{DAI}}$, where x_{USDC} is the total amount of USDC in the pool, and x_{DAI} is the total amount of DAI in the pool. Liquidity in the pool is defined by the constant-product formula

$$x_{\text{USDC}} \cdot x_{\text{DAI}} = k. \tag{6}$$

Suppose an attacker wants to a target manipulated spot price q. The swap they need to make depends on whether or not they want to manipulate the price up or down.

1.4 Manipulating price up

Let us begin by assuming they want to manipulate the spot price up, so that $q > p$. This requires them to swap USDC into the pool to withdraw DAI. How much USDC do they need to swap into the pool to achieve a price of q given the current price p and the liquidity already in the pool, k?

The manipulation swap requires them to trade an amount Δx_{USDC} into the pool to receive an amount $-\Delta x_{\text{DAI}}$ out of the pool. Swaps maintain liquidity in the pool. This means that after the swap, the new liquidity is

$$k = (x_{\text{USDC}} + \Delta x_{\text{USDC}})(x_{\text{DAI}} - \Delta x_{\text{DAI}}) = x_{\text{USDC}} \cdot x_{\text{DAI}}. \tag{7}$$

Rearranging, we find that the change $-\Delta x_{\text{DAI}}$ induced by a swap of Δx_{USDC} is

$$\begin{aligned} -\Delta x_{\text{DAI}} &= \frac{x_{\text{USDC}} \cdot x_{\text{DAI}}}{x_{\text{USDC}} + \Delta x_{\text{USDC}}} - x_{\text{DAI}} \\ &= x_{\text{DAI}} \left(\frac{x_{\text{USDC}}}{x_{\text{USDC}} + \Delta x_{\text{USDC}}} - 1 \right) \\ &= -x_{\text{DAI}} \cdot \frac{\Delta x_{\text{USDC}}}{x_{\text{USDC}} + \Delta x_{\text{USDC}}}. \end{aligned} \tag{8}$$

Now, recalling that the target price for the attacker after the swap is $q > p$, we have

$$q = \frac{x_{\text{USDC}} + \Delta x_{\text{USDC}}}{x_{\text{DAI}} - \Delta x_{\text{DAI}}}. \tag{9}$$

Substituting in to this equation for $-\Delta x_{\text{DAI}}$ from equation (14), we have

$$\begin{aligned} q &= \frac{x_{\text{USDC}} + \Delta x_{\text{USDC}}}{x_{\text{DAI}} - x_{\text{DAI}} \cdot \frac{\Delta x_{\text{USDC}}}{x_{\text{USDC}} + \Delta x_{\text{USDC}}}} \\ &= \frac{x_{\text{USDC}} + \Delta x_{\text{USDC}}}{x_{\text{DAI}} \left(1 - \frac{\Delta x_{\text{USDC}}}{x_{\text{USDC}} + \Delta x_{\text{USDC}}} \right)} \\ &= \frac{x_{\text{USDC}} + \Delta x_{\text{USDC}}}{x_{\text{DAI}} \cdot \frac{x_{\text{USDC}}}{x_{\text{USDC}} + \Delta x_{\text{USDC}}}} \\ &= \frac{(x_{\text{USDC}} + \Delta x_{\text{USDC}})^2}{k}. \end{aligned} \tag{10}$$

Solving for Δx_{USDC}, the amount of USDC needed to induce a change in spot price from p to q, we have

$$\Delta x_{\text{USDC}} = \sqrt{x_{\text{DAI}} \cdot x_{\text{USDC}} \cdot q} - x_{\text{USDC}}. \tag{11}$$

This is the amount of USDC a user needs to swap in to move the spot price of DAI from p to q on a single block. The amount of DAI they receive out is given by equation (14).

1.5 Manipulating price down

Now let us consider the scenario in which an attacker wants to manipulate the spot price down, so that $q < p$. This requires them to swap DAI into the pool to withdraw USDC. Again, we can ask how much DAI do they need to swap into the pool to achieve a price of q given the current price p and the liquidity already in the pool, k?

The manipulation swap requires them to trade an amount Δx_{DAI} into the pool to receive an amount $-\Delta x_{\text{USDC}}$ out of the pool. After the swap, the new liquidity is

$$k = (x_{\text{USDC}} - \Delta x_{\text{USDC}})(x_{\text{DAI}} + \Delta x_{\text{DAI}}) = x_{\text{USDC}} \cdot x_{\text{DAI}}. \tag{12}$$

Rearranging, we find that the change $-\Delta x_{\text{USDC}}$ induced by a swap of Δx_{DAI} is

$$-\Delta x_{\text{USDC}} = -x_{\text{USDC}} \cdot \frac{\Delta x_{\text{DAI}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}}. \tag{13}$$

Now, recalling that the target price for the attacker after the swap is $q < p$, we have

$$q = \frac{x_{\text{USDC}} - \Delta x_{\text{USDC}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}}. \tag{14}$$

Substituting in to this equation for $-\Delta x_{\text{USDC}}$ from equation (14), we have

$$\begin{aligned} q &= \frac{x_{\text{USDC}} - x_{\text{USDC}} \cdot \frac{\Delta x_{\text{DAI}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}} \\ &= \frac{x_{\text{USDC}} \left(1 - \frac{\Delta x_{\text{DAI}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}}\right)}{x_{\text{DAI}} + \Delta x_{\text{DAI}}} \\ &= \frac{x_{\text{USDC}} \cdot \frac{x_{\text{DAI}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}}}{x_{\text{DAI}} + \Delta x_{\text{DAI}}} \\ &= \frac{k}{(x_{\text{DAI}} + \Delta x_{\text{DAI}})^2} \end{aligned} \tag{15}$$

Solving for Δx_{DAI}, the amount of DAI needed to induce a change in spot price from p to q, we have

$$\Delta x_{\text{DAI}} = \sqrt{\frac{x_{\text{DAI}} \cdot x_{\text{USDC}}}{q}} - x_{\text{DAI}}. \tag{16}$$

This is the amount of DAI a user needs to swap in to move the spot price of DAI from p to q on a single block. The amount of USDC they receive out is given by equation (14).

1.6 Final manipulation cost

The difference between these two values gives the upper limit for the cost of the attack, c, which is measured as a kind of slippage:

$$c = \Delta x_{\text{USDC}} - \Delta x_{\text{DAI}} \cdot p, \tag{17}$$

where we have multiplied Δx_{DAI} by the unmanipulated price p in order to put the cost of the attack on the same unmanipulated price scale (in terms of USDC).

1.7 Examples

We can now use this equation to answer the question from the previous section about how costly it might be to move the spot price from $p = \$1$ to $q = q = 11,618,981,559$USDC in a single block.

Let us assume that before the attack the reserves of DAI and USDC are equal, so that $x_{\text{USDC}} = 25,000$ USDC, and $x_{\text{DAI}} = 25,000$ DAI. From equation (17) we find the attacker must swap in

$$\begin{aligned} \Delta x_{\text{USDC}} &= \sqrt{25,000 \text{ USDC} \cdot 25,000 \text{ DAI} \cdot 11,618,981,559 \text{ USDC / DAI}} - 25,000 \text{ USDC} \\ &\approx 2,694,759,495 \text{ USDC}. \end{aligned} \tag{18}$$

From equation (14), we find the attacker receives

$$\Delta x_{\text{DAI}} = 25,000 \text{ DAI} \cdot \frac{2,694,759,495 \text{ USDC}}{25,000 \text{ USDC} + 2,694,759,495 \text{ USDC}} \approx 25,000 \text{ DAI}. \tag{19}$$

The cost per block of this particular price manipulation, is therefore

$$c = 2,694,759,495 \text{ USDC} - 25,000 \text{ DAI} * 1 \text{ USDC / DAI} = 2,694,734,495 \text{ USDC}. \tag{20}$$

As a reminder, this is the amount of USDC an attacker must pay to move from a DAI-USDC spot price of 1 USDC / DAI to a TWAP (over a 30 minutes) of 5 USDC / DAI, given initial liquidity of 25,000 DAI and 25,000 USDC in the pool, assuming that they can control the price for 10 blocks.

In a naive implementation of the attack, this cost needs to be paid every block, but a more sophisticated attacker may be able to reduce the cost by back-running the transaction and arbitraging their own price manipulation in future blocks. To achieve this they would either need to control a significant portion of the hash power on Ethereum, to be able to reorganise transactions themselves within blocks to ensure that only they can exploit the arbitrage opportunity, or else use a system like flash bots to pay for this privilege. Both options expose the attacker to significant levels of risk, which means that the reward from moving the price of DAI 5x would need to be substantial.

2 Conclusion

Manipulation of the Uniswap v3 geometric mean TWAP is technically possible, but only under restrictive conditions. An attacker must have access to significant initial funds. The pool must have very little liquidity distributed over the entire range. The attacker must find a way to limit the costs of arbitrage from their manipulated price movement.

3 Acknowledgements

Thanks to banteg, Mudit Gupta, Dan Robinson, Luke Youngblood, and Chris Michel for reading this article and providing helpful feedback.