

Summary

The RabbitHole is an innovative feature that aims to solve the issue of sandwich attacks for those wallets which don't support the Flashbots option.

As of 17.10.22, the feature will be implemented only for MetaMask wallets on the Mainnet. While some wallets support Flashbots, MetaMask doesn't and, therefore, after creating a transaction and signing it with the private key, it is always sent to the memory pool rather than directly to the validators. In the memory pool, transactions are vulnerable to sandwich attacks. So, the 1inch RabbitHole acts as a proxy server and intermediary between MetaMask and Flashbots. This proxy server switches MetaMask from the Mainnet to the ad hoc network RabbitHole, which substitutes the Mainnet as it has the same ID - "1".

So, MetaMask signs the transaction and sends it to the blockchain, recognizing the network as the Mainnet. But the transaction is sent via the RabbitHole, which directs it to our node. The node checks the transaction destination and, if it is the 1inch Router, the transaction is sent to Flashbots. If there is another destination, it is sent to the Mainnet

In the current version of the RabbitHole, the user can't change the network to the RabbitHole in the dApp and has to do that in MetaMask. There are two reasons for that:

1. When calling MetaMask to change the network, the chain ID is used as the function argument. The Ethereum Mainnet and the RabbitHole have the same chain ID = 1. RPC URL is not specified as an argument, so the app cannot change the grid through this call.
2. If we create any other chain ID for the RabbitHole, it won't work. E.g., we will create chain ID 666. The chain ID, along with all callData, is hashed in a cryptographic signature. So, this transaction signed in a MetaMask wallet connected to the network with ID = 666 will be available for execution only in the network with ID = 666. If we try to send this transaction to a validator on the Mainnet, it won't work. We cannot change this parameter on the go, as it is built into the signature to protect transaction data against manipulations.
3. The network change call:

await

```
window . ethereum . request ( {
```

```
method :
```

```
'wallet_switchEthereumChain' ,
```

```
params :
```

```
[ {
```

```
chainId :
```

```
'0x61'
```

```
}] ,
```

```
// chainId must be in hexadecimal numbers });
```

We will be able to implement the connection of the RabbitHole network from the dApp if MetaMask enables programmatically setting a custom network with the same chain ID but a different RPC URL. Alternatively, if MetaMask can sign the transaction but not broadcast it, there will be no need for the RabbitHole, as it will be possible to implement the Flashbots option. [Edit this page](#) [Previous Oracle Deployment Guide](#) [Next Flow](#)