

Tornado Cash

Using SNARKs for Privacy and Scalability

Saravanan Vijayakumaran

Department of Electrical Engineering
Indian Institute of Technology Bombay

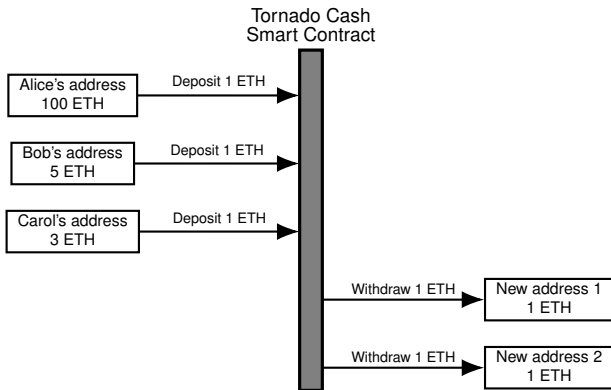
April 1, 2024

Motivation

- Consider the following scenario
 - You have 100 ETH stored in a self-custodial wallet
 - You take your family on a vacation to an exotic country
 - The hotel accepts ETH as a mode of payment
 - You pay the room rent of 1 ETH while checking in
 - The front desk clerk notices that you love your family and that your ETH address has 99 ETH
 - He has friends in the kidnapping industry
- How can you prevent leaking the total amount of ETH you hold?
 - **Option A:** You could store your ETH on an exchange and pay using their interface.
 - You risk losing funds due to exchange hacks
 - Hackers can steal customer data and sell it to their kidnapper friends
 - **Option B:** You could send 1 ETH to a fresh address from your 100 ETH address and use that to pay the room rent
 - Now suppose you decide to extend your stay
 - You make another 1 ETH transfer from your main ETH address
 - The clerk can now infer that you control a large amount of ETH
- Tornado Cash is a better **Option B**
 - It is a smart contract on Ethereum which implements a **mixer**

Tornado Cash Overview

Pre-Nova Version



- Desired functionality
 - **Soundness**
 - Only past depositors should be able to withdraw
 - No double withdrawal (only one withdrawal per deposit)
 - **Privacy**: A withdrawal should not be linkable to a particular past deposit

Deposit Workflow (1/2)

Choose amount and chain

https://app.tornado.cash

tornado Airdrop Mining Voting Compliance Docs Go GoEli Settings

Deposit Withdraw

Token: ETH

Amount: 1 ETH

0.1 ETH 1 ETH 10 ETH 100 ETH

Deposit

eth-1.tornado.cash.eth

Your IP 193.21.127.60, Mumbai, IN

Statistics 1 ETH

Anonymity set 1

4884 equal user deposits

Latest deposits

4884, 13 minutes ago	4879, a day ago
4883, 3 hours ago	4878, a day ago
4882, 10 hours ago	4877, 2 days ago
4881, 20 hours ago	4876, 2 days ago
4880, a day ago	4875, 2 days ago

https://app.tornado.cash

tornado Airdrop Mining Voting Compliance Docs Go GoEli Settings

Deposit Withdraw

Token: ETH

Amount: 1 ETH

0.1 ETH 1 ETH 10 ETH

Deposit

eth-1.tornado.cash.eth

Your IP 193.21.127.60, Mumbai, IN

Change network

- Ethereum Mainnet
- Binance Smart Chain
- xDAI Chain
- Polygon (Matic) Network
- Arbitrum One
- Avalanche Mainnet
- GoEli Ethereum GoEli (selected)

CS 1 ETH

4884 equal user deposits

4884, 13 minutes ago	4879, a day ago
4883, 3 hours ago	4878, a day ago
4882, 10 hours ago	4877, 2 days ago
4881, 20 hours ago	4876, 2 days ago
4880, a day ago	4875, 2 days ago

Deposit Workflow (2/2)

Connect wallet, save note, and deposit

The image displays two screenshots of the tornado.cash web application interface. The top screenshot shows the 'Deposit' screen. The 'Token' dropdown is set to 'ETH'. The 'Amount' slider is positioned at 1 ETH. A 'Deposit' button is visible. The 'Statistics' section on the right shows '4884 equal user deposits' and a list of 'Latest deposits' with their respective timestamps.

The bottom screenshot shows the same 'Deposit' screen with a modal window open for saving a private note. The modal contains the following text:

Your private note

Please backup your note. You will need it later to withdraw your deposit back. Treat your note as a private key - never share it with anyone, including tornado.cash developers.

```
tornado-eth-1-5-0xac607c7b40e4cf1419161316d70d2f8d5c321614feed23087b8df3ed6fa62cf89ce22c0fb9a941684b4854577f15e6c5a5c78dba6185abb61c743b04bb3
```

The browser will ask to save your note as a file: backup-tornado-eth-1-5-0xac607c7b.txt

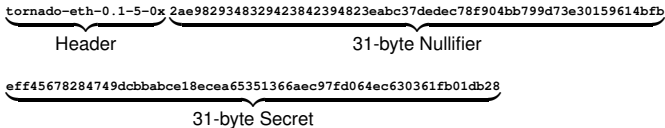
You can also save encrypted notes on-chain by setting up the Note Account, create one on the [account](#) page.

I backed up the note

Send deposit

Deposit Steps (1/2)

- Anatomy of a Tornado Cash private note



- The 62 bytes in the nullifier and secret are randomly generated on the user's computer
- A **commitment** (Pedersen hash of the 62 bytes) is calculated and submitted to the contract

$$\text{Pedersen hash of bitstring } b_1 b_2 \dots b_n = g_1^{b_1} g_2^{b_2} \dots g_n^{b_n}.$$

- Contract checks that `_commitment` has not been seen before

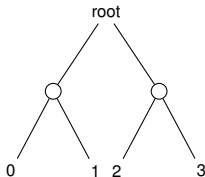
```
mapping(bytes32 => bool) public commitments;
// <snip>
require(!commitments[_commitment], "The commitment has been submitted");
```

Deposit Steps (2/2)

- Contract inserts `_commitment` into a Merkle tree

```
uint32 insertedIndex = _insert(_commitment);
```

- Tree has 20 levels
- `insertedIndex` is the index of new leaf



- No leaf deletions allowed \implies Maximum of 2^{20} deposits
- Stores the fact that `_commitment` has been seen

```
commitments[_commitment] = true;
```
- Checks that ETH being sent equals contract denomination

```
require(msg.value == denomination, "Please send 1 ETH with transaction");
```
- Emits an event

```
emit Deposit(_commitment, insertedIndex, block.timestamp);
```

Withdrawal Workflow (1/3)

Enter note string and recipient address

https://app.tornado.cash

tornado Airdrop Mining Voting Compliance Docs Go Goerli Settings

Deposit Withdraw

Note

Amount 1 ETH
Time passed 2 hours
Subsequent deposits 3 deposits

Recipient Address Donate

Total
Gas Price 4 Gwei
Network fee 0.0022 gETH
Relayer fee 0.000099 ETH
Total fee 0.002299 ETH
Tokens to receive 0.997701 ETH

Withdraw

Statistics 1 ETH

Anonymity set 4087 equal user deposits

Latest deposits

4087, an hour ago	4082, 11 hours ago
4086, an hour ago	4081, 21 hours ago
4085, an hour ago	4080, a day ago
4084, 2 hours ago	4079, a day ago
4083, 5 hours ago	4078, a day ago

Your IP 103.21.127.60, Mumbai, IN

eth-1.tornadocash.eth

Withdrawal Workflow (2/3)

Choose relay

The screenshot shows the 'Withdrawal settings' dialog box in the tornado.cash application. The dialog is titled 'Withdrawal settings' and has a close button (X) in the top right corner. It is divided into two sections: 'Relayer' and 'Wallet'. The 'Relayer' section is currently active, showing a dropdown menu with 'goerli-v2.tornadosolutions.eth' selected. Below the dropdown, it displays 'Relayer fee' as 0.1% and 'Relayer status: OK'. The 'Total' section shows a breakdown of costs: Gas Price (4 Gwei), Network fee (0.0022 gETH), and Relayer fee (0.001 ETH), resulting in a Total fee of 0.0032 ETH. The 'Tokens to receive' is listed as 0.9968 ETH. At the bottom of the dialog are two buttons: 'Reset' and 'Save'. The background shows the 'Deposit' form with fields for Note, Amount, Recipient Address, and Total Tokens to receive, along with a 'Withdraw' button.

https://app.tornado.cash

tornado Airdrop Mining Voting Compliance Docs GO Goerli Settings

Withdrawal settings

Relayer	Wallet
Relayer	
goerli-v2.tornadosolutions.eth	
Relayer fee	0.1%
Relayer status: OK	
Total	
Gas Price	4 Gwei
Network fee	0.0022 gETH
Relayer fee	0.001 ETH
Total fee	0.0032 ETH
Tokens to receive	0.9968 ETH

Reset Save

Withdraw

eth-1.tornado.cash

Withdrawal Workflow (2/3)

Choose relay

The screenshot shows the Tornado Cash web application interface. A modal window titled "Withdrawal settings" is open, displaying a list of relays. The interface includes a navigation bar with "Goerli" selected, a "Deposit" section with a note and recipient address, and a "Withdraw" button. The relay list includes various relays with their respective fees, and "goerli-v2.tornadosolutions.eth" is highlighted in green.

Withdrawal settings

Relayer Wallet

Relayer

- goerli-v2.tornadosolutions.eth
- goerli-v2.poanet.eth - 0.05%
- goerli.v2.odanrot.eth - 0.01%
- goerli-v2.releth.eth - 0.05%
- goerli-v2.relaymy.eth - 0.05%
- goerli-v2.gaasservices.eth - 0.05%
- v2.goerli.thewizardseye.eth - 0.01%
- goerli-v2.reasoned.eth - 0.1%
- goerli.t-relay.eth - 0.045%
- goerli-v2.there relayer.eth - 0.03%
- goerli.relayer-service.eth - 0.05%
- goerli-v2.tornadosolutions.eth - 0.1%
- goerli-v2.torn.eth - 0.0099%
- Custom

Withdrawal Workflow (2/3)

Choose wallet if you have an unlinkable address with ETH

The screenshot shows the Tornado.cash web application interface. A modal dialog box titled "Withdrawal settings" is open, with two tabs: "Relayer" and "Wallet". The "Wallet" tab is selected. A warning message is displayed in a yellow box: "Make sure that gETH used to pay for the gas fee is not linkable to ANY of your addresses. Otherwise, the anonymity of the withdrawal will be compromised. We recommend using a Relayer instead." Below the warning, the "Total" section shows "Tokens to receive" as "1 ETH". At the bottom of the dialog are two buttons: "Reset" and "Save".

https://app.tornado.cash

tornado Airdrop Mixed Voting Compliance Docs Goerli Settings

Deposit

Note

1f67d7af1e1ba91af14b77de3b

Amount

Time passed

Subsequent deposits

Recipient Address

0x3081b697847374ee427Db8b7

Total

Tokens to receive 1 ETH

Withdraw

Withdrawal settings

Relayer **Wallet**

Make sure that gETH used to pay for the gas fee is not linkable to ANY of your addresses. Otherwise, the anonymity of the withdrawal will be compromised. We recommend using a Relayer instead.

Total

Tokens to receive 1 ETH

Reset Save

4082 11 hours ago

4081 11 hours ago

4080 1 day ago

4079 1 day ago

4078 1 day ago

Your IP: 103.21.117.16, Protocol: JS

Withdrawal Workflow (3/3)

Generate proof and confirm withdrawal

The image displays two overlapping browser windows from the Tornado.cash application. The top window shows a dark interface with a central green circular progress indicator and the text "Generating proof...". The bottom window shows the main application interface with a "Withdraw" tab selected. A modal dialog titled "Withdrawal Confirmation" is open, displaying the message: "Your zk-Snark proof has been successfully generated! Please click Confirm to initiate the withdrawal." Below the message is a prominent red "Confirm" button. The background interface includes a navigation bar with "Deposit" and "Withdraw" tabs, a "Statistics" section, and a list of recent deposits.

Withdrawal Steps (1/2)

- Recall our requirements
 - **Soundness**
 - Only past depositors should be able to withdraw
 - No double withdrawal (only one withdrawal per deposit)
 - **Privacy:** A withdrawal should not be linkable to a particular past deposit
- The `withdraw` method is executed

```
function withdraw(  
    bytes calldata _proof,  
    bytes32 _root,  
    bytes32 _nullifierHash,  
    address payable _recipient,  
    address payable _relayer,  
    uint256 _fee  
    // <snip>  
)
```

- `_proof` is a SNARK proof for the following statement:
*I know the secret and nullifier for a commitment which is included in the Merkle tree with root `_root`.
Furthermore, `_nullifierHash` is the Pedersen hash of the commitment's nullifier.*

Withdrawal Steps (2/2)

- Contract checks that `_nullifierHash` has not been seen before.

```
mapping(bytes32 => bool) public nullifierHashes;  
// <snip>  
require(!nullifierHashes[_nullifierHash], "Note already spent");
```

This prevents double withdrawal

- Checks that `_root` is any of the last 100 Merkle roots
- It then verifies the SNARK proof on-chain

```
require(  
    verifier.verifyProof(_proof,  
        [uint256(_root), uint256(_nullifierHash), ...]  
    ),  
    "Invalid withdraw proof"  
);
```

- Stores the fact that `_nullifierHash` has been seen
`nullifierHashes[_nullifierHash] = true;`
- Sends relevant amounts to `_recipient` and `_relayer`
`_recipient.call.value(denomination - _fee)("");`
`_relayer.call.value(_fee)("");`
- The SNARK proof also “signs” the `_recipient`, `_relayer`, `_fee` fields to prevent tampering
- The verifier contract is generated using the `circom` compiler.

withdraw.circom

```
template Withdraw(levels) {
  signal input root;
  signal input nullifierHash;
  signal input recipient; // not taking part in any computations
  signal input relayer; // not taking part in any computations
  signal input fee; // not taking part in any computations
  signal private input nullifier;
  signal private input secret;
  signal private input pathElements[levels];
  signal private input pathIndices[levels];

  component hasher = CommitmentHasher();
  hasher.nullifier <== nullifier;
  hasher.secret <== secret;
  hasher.nullifierHash === nullifierHash;

  component tree = MerkleTreeChecker(levels);
  tree.leaf <== hasher.commitment;
  tree.root <== root;
  for (var i = 0; i < levels; i++) {
    tree.pathElements[i] <== pathElements[i];
    tree.pathIndices[i] <== pathIndices[i];
  }

  // Add hidden signals to make sure that tampering with recipient or fee will invalidate the snark proof
  // Most likely it is not required, but it's better to stay on the safe side and it only takes 2 constraints
  // Squares are used to prevent optimizer from removing those constraints
  signal recipientSquare;
  signal feeSquare;
  signal relayerSquare;
  recipientSquare <== recipient * recipient;
  feeSquare <== fee * fee;
  relayerSquare <== relayer * relayer;
}

component main = Withdraw(20);
```

OFAC Sanctions

- On Aug 8, 2022, the US Office of Foreign Assets Control placed Tornado Cash addresses on a sanction list
- US residents/businesses cannot interact with entities on the list
- Allegations include facilitating money laundering by ransomware operators and smart contract attackers
- Github removed source repos and three contributors had Github accounts suspended
- Due to the efforts of Prof. Matthew Green and EFF, OFAC allowed use of code for educational purposes
- Github repositories and accounts restored in 2023
- Developer Alexey Pertsev arrested in Netherlands in Aug 2022; released on bail in April 2023
- Developer Roman Storm arrested in US on Aug 23, 2023 and later released on bail
- Pertsev's trial began on March 26, 2024. Verdict expected on May 14

References

- **Tornado Cash App** <https://tornadoeth.cash/>
- **Tornado Cash Docs** <https://docs.tornadoeth.cash/>
- **Circom** <https://docs.circom.io/>
- <https://github.com/tornadocash/tornado-core>
- **EFF article on OFAC sanctions**
- **EFF update in April 2023**