We would like to introduce XCLAIM - a protocol for trustless cross-chain communication via cryptocurrency-backed assets.

XCLAIM is described in the following academic research paper:

- “XCLAIM: Trustless, Interoperable Cryptocurrency-Backed Assets” (https://eprint.iacr.org/2018/643.pdf). To appear at IEEE S&P 2019.

We provide a short summary below (also available on website w. figures:https://xclaim.io).

XCLAIM is a framework for achieving trustless and efficient cross-chain exchanges using cryptocurrency-backed assets (CbAs). That is, it allows to create assets which are 1:1 backed by existing cryptocurrencies, without requiring trust in a central operator. While this approach is applicable to a wide range of cryptocurrencies, we currently focus on implementing Bitcoin-backed tokens on Ethereum, i.e. XCLAIM(BTC,ETH).

We use b

to refer to the backing coin (BTC) and B

to the backing blockchain (Bitcoin). Analogous for the issuing blockchain (Ethereum/ETH): i

and I

. Assets on I

backed by units of b

are denoted as i(b)

.

Please refer to the paper for detailed and more formal definitions and protocol descriptions.

The main actors in XCLAIM are as follows (not all listed for simplicity):

- Requester

: Locks b

on B

to request i(b)

on I

.

- Redeemer

: Destroys i(b)

on I

to request the corresponding amount of b

on B

- Backing Vault (vault)

: Non-trusted and collateralized intermediary liable for fulfilling redeem requests of i(b)

for b

on B

.

- Issuing Smart Contract (iSC)

: Smart contract on I

managing correct issuing and exchange of i(b)

. The iSC enforces correct behavior of the vault.

XCLAIM introduces four protocols. Issue

and Redeem

as specifically of interest, while Transfer

and Swap

are trivial. Note: upper / lower bounds for delays introduced in XCLAIM are provided in the paper, based on Kiayias et al. Backbone model[1]. Adversaries are assumed to be economically rational.

Precondition:

- The iSC is deployed on I

and defines an over-collateralization rate (> 0) for vaults and provides an exchange rate (>= 1.0) (for now, we assume an oracle. Improved oracles are WIP).

- Vault registers with the iSC by locking up collateral, as defined by the over-collateralization and exchange rates. The amount of locked up collateral defines how many i(b)

can be issued with this vault.

Issue

:

1. Requester commits to issuing by locking up a small amount of collateral i

with the iSC and specifies address (def. by pub.key) on I

where the i(b)

are to be issued to. The iSC blocks the corresponding amount of collateral of the vault for pre-defined delay(i.e., the vault cannot withdraw the blocked collateral until timeout expires).

1. Requester sends b

to vault on B

.

1. Requester submits a TX inclusion proof to the iSC (via a chain relay).
2. iSC issues i(b)

to the requester.

It is easy to see that Issue

is non-interactive. As long as a vault is registered with collateral in the iSC, any user can lock b

and issue i(b)

. No permission by the vault is required(!) Replay protection and counterfeit prevention is discussed in Sec. VII of the paper.

Redeem

:

1. Redeemer locks i(b)

with the iSC and specifies his address (def. by pub. key) ob B

1. iSC emits event signalling that the vault must send b

to the redeemer on B

such that |b

| == |i(b)

|, within some pre-defined delay.

1. Vault sends b

to redeemer on B

1. Vault submits TX inclusion proof to iSC, showing that the redeem was executed correctly
2. iSC unblocks the vault's collateral on I

and destroys the locked i(b)

If the vault fails to provide a proof

, the iSC reimburses the requester in i

(exchange rate + penalty from over-collateralization).

XCLAIM uses a multi-stage over-collateralization scheme and allows users to opt-in for automatic liquidation, should the collateralization rate of a vault drop below a certain threshold (e.g. 1.05). Note: coll. rate <1.0 results in the vault being incentivized to misbehave, as the revenue gained from stealing locked b

exceeds the penalty incurred in i

).

The operation of the automatic liquidation depends on the implementation of the oracle (can be oracle triggered or, more likely, users/watchtowers must submit a transaction to the iSC to trigger).

As such, XCLAIM ensures (Value) Redeemability

: a user who owns i(b)

is guaranteed to receive the corresponding amount of b

or be reimbursed with the equiv. economic valut in i

.

Finally, XCLAIM allows any

user to become a vault by locking up collateral with the iSC. This allows the system to scale and makes it resilient against DoS / censorship attacks.

XCLAIM is still a first PoC and there remain many challenges to be solved. First implementation evaluations are provided in the paper. We also provide some PoC code (incl. a WIP BTCRelay implementation in Solidity): https://github.com/crossclaim/.

We look forward to receive feedback and suggestions for future work from the Ethereum Community.

PS: We use the "sharding" tag, since cross-chain communication as designed in XCLAIM can also apply to sharding - the principles are very similar. Feel free to correct/suggest better tags

PPS: XCLAIM will be presented at EthCC and EDCON, in case you'd like to have a chat in person. There's also a video of a previous protocol version presented at ScalingBitcoin'18

[[1] Garay, Juan, Aggelos Kiayias, and Nikos Leonardos. "The bitcoin backbone protocol: Analysis and applications." Annual International Conference on the Theory and Applications of Cryptographic Techniques

. 2015.](https://eprint.iacr.org/2014/765.pdf)