Web-based XCAT tool for easy ZEC/BTC atomic trading

Zcash Foundation Grant Proposal (2017 Q4)

Jason Davies <jason@jasondavies.com>

Motivation and overview

Building on the excellent work by @arielgabizon, @arcalinea, and others, I propose to make their work more accessible by providing a simple web-based tool that facilitates:

- **Easy setup of an XCAT trade**. This includes optional generation of the random secret in-browser. Once the trade parameters have been agreed between the parties (secret hash, 4 addresses, 2 amounts), a URL can be generated for easy (and optional) sharing/bookmarking. The page would provide instructions for creating and sending the relevant transactions. The page *may* also allow the transactions to be broadcast, in the case where users cannot easily do so themselves.
- **Optional monitoring of a trade**. The XCAT mechanism will be illustrated visually, and progress of the requisite transactions can be optionally monitored. The visual aspect helps educate users and developers about how XCAT works, and if monitoring is enabled, it allows users to see which parts of the trade have completed, and easily see what to do next to complete the trade (or revoke it if something goes wrong).

The purpose of the tool is twofold: facilitate easy creation of XCAT trades *and* educate users and wallet developers about how XCAT works.

Technical approach

- The majority of the work involves designing a nice user interface written in JavaScript, HTML and CSS, and intuitive visualisations for the trade and its progress along the two selected blockchains.
- Initially only ZEC and BTC will be supported.
- The optional monitoring of trade progress will require a server running both Zcash and Bitcoin nodes ideally with `txindex=1`.

Background and qualifications

MA (Cantab) CompSci, Univ. of Cambridge, contributor to various open-source projects; most relevant to this proposal is my work on D3.js, and you can see various visualisations on my personal website.

I've been following Zcash since its inception and have made some minor contributions so far:

- zcash-sprout-verifier - verifier for Zcash zk-SNARK proofs in Rust.
- zcash-vanity - vanity z-addr generator in Rust and OpenCL.

I also helped @arielgabizon test the first ever XCAT trade with ZEC/BTC on their respective testnets.

Evaluation plan

The following deliverables are required:

- A web page that allows the main XCAT parameters to be entered and agreed upon between two parties.
- The web page should provide instructions for generation and broadcast of the appropriate transactions.
- A visualisation of the progress of the trade, with optional monitoring which requires some information to be sent to the server.

Security considerations

The basic vision for this web-based tool does not involve any private keys being shared with the tool; it should only instruct the two parties to enter their {fund,redeem} addresses for the respective blockchains used in the trade, and later they may optionally monitor the trade's progress if they are happy to share some information with the server.

Given the XCAT parameters (4 addresses, hash of secret, 2 redeemblocknums) agreed upon by the counterparties, the tool should generate appropriate HTLC (escrow) scripts and accompanying P2SH addresses, and then instruct the users to send their funds to these addresses in the correct order while checking for sufficient confirmations. Users should be encouraged to double-check for sufficient confirmations on other blockchain explorers and/or their own clients.

The page should be served over HTTPS. The HTLC script and address generation should all be implemented in JavaScript to minimise information sent to the server.

Users can also bookmark or share their trade, but this does not leak any trade information to the server as the trade parameters can be stored in the URI fragment.

The parties to the trade should be able to select the level of privacy with respect to leaking information to the server, ranging from *no information* (other than their IP address, which can be mitigated using Tor or similar), or if they elect to monitor the progress of the trade, their escrow addresses need to be shared with the server so that it knows when funds have been moved to and from the HTLC scripts.

Furthermore, care should be taken when suggesting `redeemblocknum` parameters to the users, such that sufficient margins are in place for the cases where the counterparty does not follow through, and the trade needs to be aborted.

Schedule

There are three main milestones:

- Implement XCAT in JavaScript (HTLC generation, P2SH address generation) and store parameters in URI fragment. This is the bare min-

imum that allows users to make a trade. (Estimate: end of November 2017.)
- Implement an interactive visualisation that illustrates the XCAT process. (Estimate: end of November 2017.)
- Implement server-side monitoring on both BTC and ZEC blockchains, for which users can opt-in, with real-time integration with the XCAT visualisation. (Estimate: end of December 2017.)

Note that I have submitted a separate proposal for implementing Zcash JoinSplit & Payment Disclosure validation in Rust, but I do not anticipate this will significantly affect the above schedule if both proposals are accepted.

Budget and justification

I have spent some time thinking about how the UI and visualisations might look, so I'm already part-way along in the process. I have also helped Ariel test their XCAT code so I also already have an understanding of how it fits together.

I estimate that around $15k would be sufficient to complete the development of this tool, with a turnaround time of around 1-2 months. I'm happy to take on hosting costs etc. myself.