Navcoin-js Hackathon Day 2

https://github.com/aguycalled/navcoin-js-hackathon

alex@nav.community

Thanks for coming back!

Today…

- First round of questions (if any), you can show your projects if you want.
- A bit of theory, I will talk about the on-chain trading protocol that will activate Q1 2022.
- What has been added to navcoin-js in the last week.
- What is dotNav?
- Adding dotNav to the private chat app.

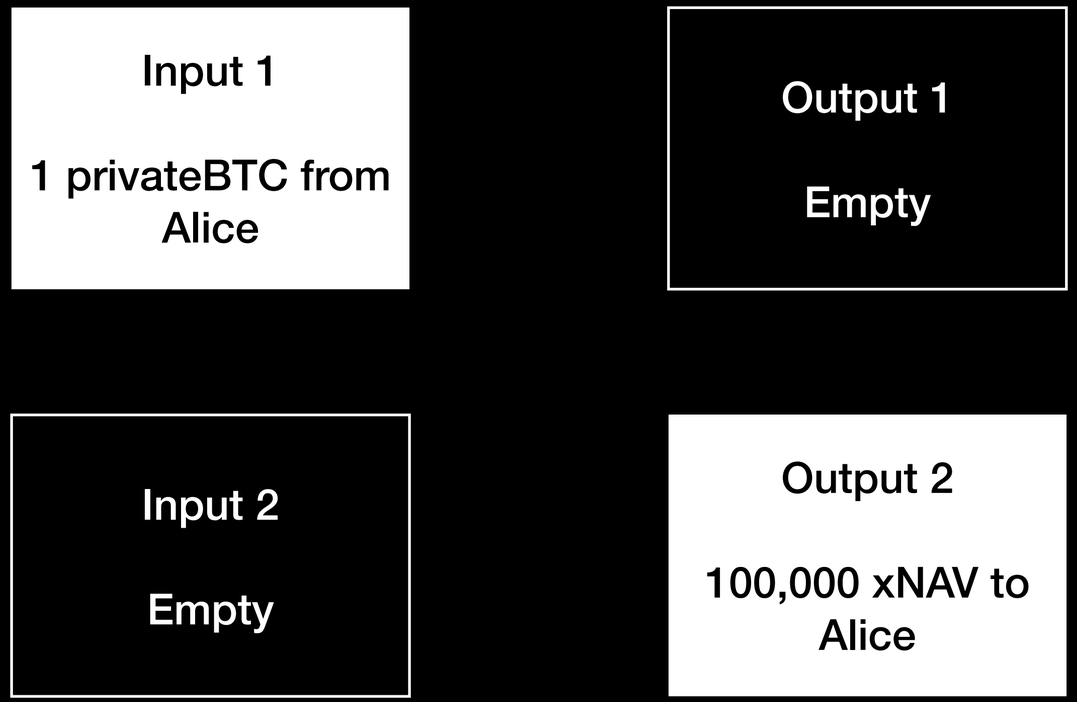
Questions. Personal projects.

- Any question?
- Remember, deadline to submit is December,17th. Up to 25,000NAV in prizes.
- Next week (December, 10th). Private Tokens, NFTs.
- Do you want to show your personal projects? Coordination with other participants so no-one creates a duplicated app.

On-Chain Trading Protocol

- Possible thanks to BLS signatures.
- ECDSA signatures (bitcoin) are very rigid. One signature per transaction. You can’t do much more.
- BLS signatures are very flexible. One signature per input/output. But, you can aggregate or reorder them. Once aggregated you can’t recover the single signatures (great for privacy!)
- This allows to construct partial transactions which later an user can complete to finalise an action.

On-Chain Trading Protocol



Alice signs Input 1 and Output 2. The sum of those two signatures is the signature of this partial transaction.

This transaction is invalid, because:

- It's trying to send 100,000 xNAV but those coins come from nowhere. Alice is only sending 1 privateBTC.

But Alice can broadcast this transaction and wait for others to complete it.

On-Chain Trading Protocol

Input 1

1 privateBTC from Alice

Output 1

1 privateBTC to Bob

Input 2

100,000 xNAV from Bob

Output 2

100,000 xNAV to Alice

Bob signs Input 2 and Output 1. The aggregation of those two signatures with the previous signature complete the transaction's signature.

This transaction is valid, because:

The same amount of each asset is present as inputs and outputs.

The transaction is final, because BLS signatures can be aggregated but not segregated (*).

On-Chain Trading Protocol: NFT minting

Input 1

Empty

Output 1

Empty

Input 2

Alice (NFTs owner) authorises minting 1 NFT

Output 2

1,000 xNAV to Alice

Alice signs Input 2 and Output 2. The sum of those two signatures is the signature of this partial transaction.

This transaction is invalid, because:

- It's trying to send 1,000 xNAV but those coins come from nowhere.

But Alice can make this partial transaction public, so others can do the rest of the work…

On-Chain Trading Protocol: NFT minting

Input 1

1,000 xNAV from Bob

Output 1

1 NFT to Bob

Bob signs Input 1 and Output 1. The aggregation of those two signatures with the previous signature complete the transaction's signature.

Input 2

Alice (NFTs owner) authorises minting of 1 NFT

Output 2

1,000 xNAV to Alice

This transaction is valid and once broadcasted to the network, Bob will be buying 1 NFT for 1,000 xNAV, trustlessly and privately.

On-Chain Trading Protocol

- The same concept can be used for dotNav names, Private Tokens (ICOs, initial distribution), etc etc.
- On going development. Fully implemented Q1 2022.

What’s new on navcoin-js

- Current version v.1.0.40
- How to update to the last version: npm update navcoin-js
- Support for dotNav names.
- New methods:
  - ResolveName, RegisterName, UpdateName, IsMyName, GetMyNames

What is dotNav?

- On-chain version of DNS. Secured by Navcoin’s blockchain.
- Names ending in .nav
- Each name can have multiple pairs of key/values (similar to DNS system with A, CNAME, MX, etc records)
- Each name can have a subdomain.

What is dotNav? Example

- aguycalledalex.nav:
  - Key: _key Value: public key which owns the name
  - Key: _expiry Value: Block height when the name will expire
  - Key: nav Value: N12djs41MN…. Nav address
  - Key: email Value: alex@nav.community

What is dotNav? Example 2

- google.nav:
  - Key: _key Value: public key which owns the name
  - Key: _expiry Value: Block height when the name will expire
  - Key: ip Value: 8.8.8.8

What is dotNav? Example 3

- nftsale.nav:
  - Key: _key Value: public key which owns the name
  - Key: _expiry Value: Block height when the name will expire
  - Key: nftsale Value: ipfs resource with the partial transactions to buy the NFTs.

What is dotNav?

- Very interesting possible application for dotNav:
  - A DNS proxy which resolves .nav names.
  - E.g: User tries to access name.nav in the browser, the proxy transparently resolves name.nav, gets the ip entry of the name and forwards the request.

Adding dotNav to the private chat app

- Let’s code…

That’s all for today!

- Updated repository with todays’ material: https://github.com/aguycalled/navcoin-js-hackathon
- Questions?

- Happy hacking!