CRYPTOCURRENCY: USER-FRIENDLY MICROPAYMENT CHANNELS

Medha Upadhyay
UCSD
mupadhya@ucsd.edu

Isa Vidanes
UCSD
ividanes@ucsd.edu

Minh Luc
UCSD
mluc@ucsd.edu

Priscilla Hui
UCSD
pyhui@ucsd.edu

Mentor: Sheffield Nolan
Franklin Templeton
Sheffield.Nolan@franklintempleton.com

INTRO

- Micropayment channels are used to manage frequent/repeated transactions efficiently
- Existing models are inconvenient andnot beginner friendly; every transaction requires a set gas fee
- We created a user-friendly micropayment channel: logs payments off-chain and sends one lump sum at the end

METHODS

- Connecting to MetaMask is implemented through Solidity smart contracts (converted to bytecode) and JavaScript
- New transactions are logged using key-value pairs in local storage
- A unique signature is generated using the web3.js package
- Lump sum payment is transferred when the channel is closed
- User must provide the unique signature in order to validate the transaction

RESULTS

- Integration with MetaMask ensures that interaction, including contract deployment and transactions, are authenticated and confirmed by the user
- Smart contract can be deployed through our interface and will be recorded on the Ethereum blockchain
- Users can log payments off-chain using our interface; the running total will be displayed
- Once the sender and receiver decide that it is time to close the channel, they can do so through our interface; this will send the amount owed from the sender to the receiver and close the contract

USING OUR CHANNEL

Enter the receiver address on the interface; the interface will use a piece of bytecode to deploy the micropayment channel

This bytecode references the Solidity smart contract code for a micropayment channel

The payment channel will be published on-chain by the interface (no work required for the user!)

Instead of sending every transaction on-chain and having to pay a fee every time, use the interface to log each payment; save the unique signature!

Close the channel through the interface by using the unique signature; one lump sum payment will be made on chain, meaning that transaction fees only have to be paid once

DISCUSSION

- Key points: transparency, immutability, user friendly-ness, and security
- Hosted on a testnet; with further testing, this framework can be expanded to be deployed on the Ethereum blockchain and be used for real transactions among trusted parties
- Potential areas of future development: adapting the contract so it can be suitable with blockchain networks other than Ethereum

CONCLUSION

- Interface guides users through creating and using a micropayment channel on the Ethereum blockchain
- MetaMask integration ensures secure and authenticated transactions
- Users can interact with the micropayment channel on the Ethereum testnet and use it to sent testnet Ethereum to trusted users
- What sets our approach apart: streamlined and accessible approach to promote user friendly-ness

REFERENCES

- Solidity. 2018. "Solidity By Example."
- VOYIX, NCR. 2022. "The Creation of Payment Channels & How They Work."