27th March 2023/ part. 2994

Address epic cash wallets in epicbox using addresses that are easy to remember and save. This method is more anonymous and secure than using standard wallet public keys in combination with an epicbox domain name.

Sit down comfortable. Open your mind. Read it carefully. Understand. Read again. Ask questions.

enjoy short period when my cat is in the past and my mind can do something more then only react on MEOW. All described methods are implemented now in my test wallet and epicboxnodejs.

1. How the transaction between epic cash wallets is performed.

Epic cash wallets provide one of the most anonymous forms of value exchange. These wallets in their original version did not need addresses. Transactions between wallets always consist of three steps. Step 1. Preparation of data by the sending wallet and transferring them to the receiving wallet. Step 2. Preparation of data by the receiving wallet and transferring them to the sending wallet. Step 3. The sending wallet completes the transaction by recording the transaction on the blockchain. Data transfer between wallets is currently done using the following methods: files, keybase, http, tor and epicbox. If both wallets are connected to the Internet, the transaction is completed within a few seconds. However, when one of the wallets is offline then only the file and epicbox method can be used.

2. How does the epicbox network work?

Epicbox is a server that stores and transmits the data that the wallet exchanges with each other. If one wallet is absent then epicbox stores the data (Slate) from the other wallet until the wallet reconnects to epicbox. Epicboxes operate under any subdomains (e.g. https://epicbox.fastepic.eu) and can be managed by independent operators. Epicbox recognizes the wallet of the recipient and the sender using the public keys of the wallets, which are transferred to epicbox by the wallet in the PostSlate message in open form.

Steps (examples use epic-wallet cli):

- user insert send command

epic-wallet send -m epicbox -d esYHuXpBRJu9X14w8KKAdUiDaQAagHnmKXxmwvVMzY48PGmKPJSG@epicbox.fastepic.eu 100

where es...SG is public key of receiver wallet. It is hard to remember in the mind and rather doesn't change for wallet.

- wallet connects with epicbox to which is want connected (setup in toml file). It doesn't have to be the same epic box whose domain appears in the recipient's wallet address. The wallet receives a challenge string from epicbox that will be used later by the wallet.

- the wallet encrypts with the recipient's public key (part of the recipient's address) data for the recipient's wallet. So they can only be read by the recipient's wallet. The wallet prepares a message (PostSlate) for epicbox consisting of encrypted data for the recipient's wallet, the recipient's open address, the sender's open address and signs the whole of such a message with his private key. Add the signature to the message.

- epicbox receives the message from the sender's wallet and verifies that the sender's and recipient's open addresses are correct (public key and domain). It verifies the signature to make sure that the sender is indeed the wallet listed as the sender in the message. It then reads the recipient's domain. If the recipient's domain is the same as the epicbox domain, the message is saved in the epicbox database. If the recipient's domain is different from the epicbox domain, the entire message is sent to epicbox working under the domain from the recipient's address. If everything went well, the sender's wallet receives an Ok message.

- the sender's wallet may terminate the connection with epicbox at this point, or remain listening (Subscribe) waiting for a return message from the recipient's wallet.

- the recipient's wallet connects to his epicbox (domain same as recipient's address). Receives a Challenge string.

- the recipient's wallet signs the Challenge string with his private key and generates a Subscribe message containing his public key (address) and signature.

- epicbox receives a Subscribe message from the recipient's wallet. Verifies whether the challenge string has been signed with a private key corresponding to the public key that is provided by the recipient's wallet.

- if the Subscribe message verification was successful, epicbox searches for the oldest message for this wallet in the epicbox database and sends

it to the wallet (message Slate) adding its own identification number to it.

- After receiving a Slate message from epicbox, the wallet sends a Made message to epicbox with the id of the epicbox message, so that it can remove it from its database, and then decrypts the part encrypted with its public key using its private key. It reads the sender's address and prepares a data message for the sender's wallet, encrypting it with the sender's public key. He adds his open address and the address of the sender to the message and signs the whole thing with his public key. Include a signature signature in the message and send it to epicbox (PostSlate).

- sender's wallet receives data from his epic box using Subscribe. He confirms the receipt of the data by sending a Made message with his signature and message number from epicbox. message with signature for Challenge string using his public key.

- it decrypts the data from the receiving wallet using its private key and finalizes the transaction in the blockchain.

That’s all in current epicbox method.

What I don't like!

- I am unable to remember the recipient's address or mine.
- I have to save them on my computer for future use, even if I don't want them saved.
- I can't give the address orally or quickly write it down securely on a piece of paper for the sender. Because no one is able to rewrite it to your wallet
- The person to whom I gave my address may pass it on to another person or epicbox operator to check with whom I still conduct operations.
- My address, which uniquely identifies my wallet, is used many times by different users on different computers, where it can be taken over by unauthorized screenshots or copied from data.

That was the boring part.

Now let's try to shake our minds and take epic cash to the next level of usability while increasing the level of anonymity and security when transacting through epicbox.

I will present here the technology and theory to it, which I am currently testing on a test cli wallet and working epicbox under the domain – epicbox.fastepic.eu

What I want or not.

- So that addresses can be **remembered in the head** - especially the important ones.
- I **want** senders' computers to never display my public key that uniquely points to my wallet.
- I **don't want** my address to be saved on the sender's computer
- I **want** different addresses for different senders.
- I **want** an address that will disappear after use and will not identify me after disappearing from the base in epicbox.
- I **want** the address for receiving the value to be confirmed by the recipient's wallet, even if he is not connected to the Internet at the moment.
- I **don't want** to trust epicbox operators.

Look how I fulfilled all my wishes.

Wallet received additional comments and epixboxnods received additional messages.

Registration of a new address (up to 10 on epixbox).

epic-wallet niceaddress king@epixbox.fastepic.eu -e 60

argument -e 60 means that the address could be deleted from the epicbox database after 60 seconds.

- the wallet takes the string king@epicbox.fastepic.eu and combines it with the lifetime of the address - the string king@epicbox.fastepic.eu60 is created

- the wallet signs the created string with its private key and creates a SetAddress message appending niceaddress, its public key (address), niceaddress and lifetime and signature.

- epicbox checks if the public address is correct, I verify the signature and if everything is ok, it saves in the epicbox database niceaddress, public address, lifetime and signature. If the address is to be on a different domain, it sends to another epicbox.

Any user wants to send values to king@epicbox.fastepic.eu.

epic-wallet send -m epicbox -d king@epicbox.fastepic.eu 100

- the wallet signs the string king@epicbox.fastepic.eu with its private key, attaches its signature and its public address and sends GetAddress messages to its epicbox.

- epicbox checks if the address of the requesting wallet is correct and checks the signature for the signed niceaddress for which the wallet is asked. In this way, we protect accidental attempts to download nice addresses by applications that are not wallets. If everything is fine, it checks the niceaddress domain - if another, it sends a message to another epicbox. If appropriate for this, epicbox searches if there is such a niceaddress in the epicbox database.

- If there is such a nice address, it sends back to the wallet GetAddress a message containing the public address corresponding to the nice address, lifetime and signature stored in the database.

- The wallet, after receiving the response, combines the niceaddress with the lifetime (king@epicbox.fastepic.eu60) and checks the signature, making sure that the provided niceaddress was ordered by the recipient's public address provided with the epicbox. This guarantees that the provided public address combined with niceaddress was not generated by e.g. epicbox operator in order to extort funds.

- The wallet will send the funds using the received and verified public address using the traditional epicbox method described at the beginning of this document. The niceaddress does not appear anywhere in the transaction anymore.

Drop niceaddress

epic-wallet dropnice king@epicbox.fastepic.eu60

- wallet must sign king@epicbox.fastepic.eu by private key and send Message with own public address, nice address and signature

Receiver can still receive waiting slates from epicbox after drop address.

Thus all my wishes were fulfilled.

Copyrights and idea by fastepic 2023