(CS5543) The Blockchain : Theory and Practice

Assignment_Report

- Multisig 2 of 3 security layer for bitcoin wallet -

Algorithm proceeds as follows:

Step_1 :

Create 3 accounts.

'generateMultisig()' function creates 3 addresses along with private and public keys using ECKey class which is imported from bitcoin org.bitcoinj.core library.

Firstly, A key is generated using a text string as a secret code by createKeyFromSha256Passphrase() function.This key is used to generate address using toAddress() method associated with the key object, returning an Address object), private key(using getPrivateKeyEncoded() method associated with the key object, returning DumpedPrivateKey object) and public key(using getPubKey() method associated with the 'Address' object).

Similarly two more keys can be generated to get the corresponding addresses, private keys and public keys.

Step_2 :

Create a multisig account that returns address and redeem script which are essential for transactions.

Create an immutable list using three keys.Now the redeem script is generated using createRedeemScript() method of ‘ScriptBuilder’ static class.

This method takes two arguments that is required no. of keys for successful transactions and the list of keys, returning a ‘Script’ object.This script is then converted into hexcode.

Also Multisig address is created using the ‘fromP2SHScript’ method of ‘Address’ class, which uses the above script.

Step_3 :

First time sign in for the transaction.

Create a new transaction object that takes input script and output address to make the transaction complete.

Firstly, create a ‘TransactionInput’ object that holds the script of the address that we want to add bitcoins from.

Then specify the receiver’s address and convert it into a script object. Also, create a ‘Coin’ object that holds a minimum transaction fee for one transaction of bitcoins.These two objects are provided as arguments to ‘addOutput’ method of transaction object that we initialise first while configuring the ‘addInput’ method using ‘TransactionInput’ object.

Now a ‘TransactionSignature’ object is created by providing ‘ECDSASignature’ object, which is created using one of the three keys, and ‘Sha256Hash’ object, which is created using ‘hashForSignature’ method of the ‘Transaction’ object that we created before.

Now, ‘bitcoinSerialize’ method associated with previously created ‘Transaction’ object is used to generate a hexcode that is used for further transaction verification, using the ‘Script’ object that is generated using ‘createP2SHMultiSigInputScript’ method with ‘TransactionSignature’ object we generated before and the redeem script of the multisig account.

Step_4 :

Second time sign in for the transaction.

Firstly, convert the hexcode we got in step 3 into a ‘Transaction’ object. Then this object is used to generate a list of ‘ScriptChunk’ object. Now, we have to extract the last signature from it and this is the redeem script, which is stored as a ‘TransactionSignature’ object and added to signatures list.

Now, step 3 is repeated for the second key with the second signature added to the list of signatures as an argument to the ‘createP2SHMultiSigInputScript’ method, to get the ‘Script’ object.

Now, ‘bitcoinSerialize’ method associated with previously created ‘Transaction’ object is used to generate a hexcode. And this hexcode is used to verify the transaction.

https://coinb.in/#home

G.SAI GOUTHAM : CS15BTECH11017
D.SRINIVAS : CS15BTECH11015