Universität Heidelberg
Wintersemester 18/19

Group Parallel and Distributed Systems (PVS)
Artur Andrzejak, Diego Costa

Problem Set 12 for lecture Distributed Systems I (IVS1)

Due: 29.01.2019, 14:00 Uhr

Exercise 1 (2 Points)

Consider the steps involved in processing Bitcoin transactions[1]. Which of these steps are computationally expensive? If you're an entity validating many transactions (say, a miner) what data structure might you build to help speed-up verification?

Exercise 2 (1 Point)

What design features Bitcoin script language possesses that benefits the security of the bitcoin mining process? What problems would arise if miners could use a Turing-complete language? Provide some examples in your answer.

Exercise 3 (3 Points)

For the following questions, you're free to use non-standard transactions and op codes that are currently disabled. You can use <data> as a shorthand to represent data values pushed onto the stack. For a quick reference, check the Bitcoin wiki page[2].

a. Write the Bitcoin ScriptPubKey script for a transaction that can be redeemed by anybody who supplies a square root of 1764.

b. Write a corresponding ScriptSig script to redeem your transaction.

Exercise 4 (Bonus, 2 Points)

Alice is backpacking and is worried about her devices containing private keys getting stolen. So she would like to store her bitcoins in such a way that they can be redeemed via knowledge of only a password. Accordingly, she stores them in the following ScriptPubKey address:

OP_SHA1

<0x084a3501edef6845f2f1e4198ec3a2b81cf5c6bc>

OP_EQUALVERIFY

a. Write a ScriptSig script that will successfully redeem this transaction. **Hint:** it should only be one line long.

b. Explain why this is not a secure way to protect Bitcoins using a password.

c. Would implementing this using Pay-to-script-hash (P2SH) fix the security issue(s) you identified? Why or why not?

[1] `https://en.bitcoin.it/wiki/Protocol_rules#.22tx.22_messages`

[2] `https://en.bitcoin.it/wiki/Script`

Exercise 5 (3 Points)

Bitcoins scripts can be used to accomplish tasks far beyond the simple verification of public keys. Research and explain the concept behind the following applications, and with an example describe how the Bitcoin script language enables the process of such transactions.

a. Escrow Transactions,

b. Green addresses,

c. Efficient micro-payments.

Exercise 6 (2 Points)

What are some ways to burn bitcoins, i.e., to make a transaction unredeemable? Which of these allow a proof of burn, that is, convincing any observer that no one can redeem such a transaction? How could you use such unredeemable transactions to store text that is kept forever in the block-chain?

Exercise 7 (2 Points)

Read the article on Bitcoin Wallets[3] and briefly describe what are wallets and what are the different types of wallets available in the market. Which wallet would you recommend to Bitcoin newcomers, and which one fit best the requirements of a company fund?

Exercise 8 (3 Points)

The block-chain has many other applications that go beyond digital currencies, Block-chain concept of identity, transactions, record-keeping and consensus enables a variety of decentralized applications. Read the article „What is Ethereum?“[4] (and the watch the embedded videos), and answer the questions below:

a. What is the Etherium platform and what motivated its creation? What limitations of Bitcoin script and mining protocol posed a challenge to be applicable to general applications?

b. What are Smart Contracts? How can they be used to enforce the involved partners to comply with contract requirements?

c. Describe the reasons that lead Etherium to perform a hard fork, and what this entails for the platform's future.

[3] `http://cryptorials.io/bitcoin-wallets-explained-how-to-choose-the-best-wallet-for-you/`

[4] `https://blockgeeks.com/guides/ethereum/`