PA193 Security technologies

Team projects: parser for cryptocurrency blockchain entry

Petr Švenda
Faculty of Informatics, Masaryk University, Brno, CZ

CROCS
Centre for Research on Cryptography and Security

Project idea

1. Write custom parser for selected cryptocurrency
2. Code review of other team parser
3. Create patch for a selected flow and pull request

Project

1. Identify, design and implement cryptocurrency blockchain parser (now)
   - Parser will take two consecutive blocks, assume validity of a first one and verify the validity of the second one
   - Must be written completely from scratch, your own code
   - For cryptographic functions (sign, hash) use some existing library
   - Use git (GitHub), document functions, create tests
   - Not Bitcoin or its direct copycats (with same or very similar block structure)
   - Parse original binary format of the block (not some postprocessed JSON etc.)
   - See https://www.worldcoinindex.com/ for list of coins
2. Review of code of other group's parser
   - Find bugs by code review, using automated tools etc. (static analyzers, fuzzers…)
   - Prepare presentation with findings
3. Implement bugfix for one selected issue and create pull request

Teams

- 3 people per team
  - Assigned today (within group)
- Teams must use GitHub for cooperation
  - Distribute work load between all members
  - Contribution from all team members must be visible in commits (git commits from the member)
  - Your evaluation will be partially based on your participation
- Start working early, especially with implementation

Immediate next steps

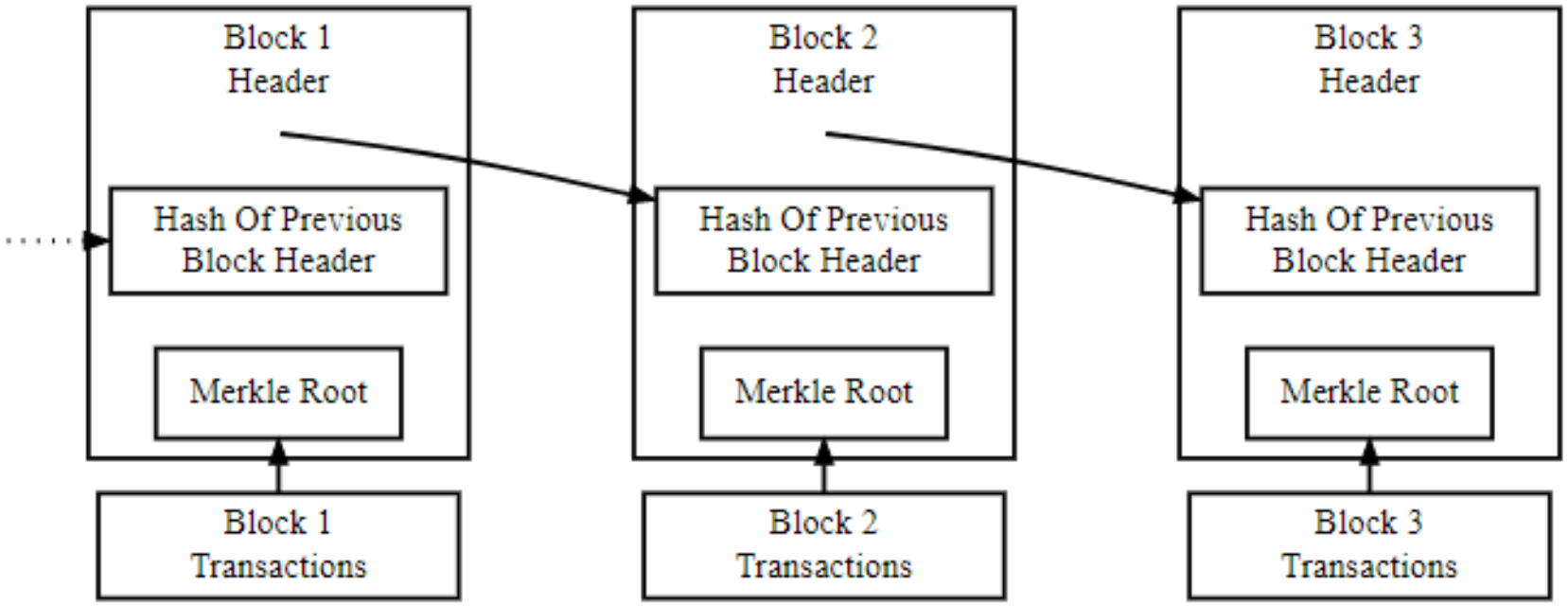
- Form group, exchange emails
- Get together and select your favorite currency
- Find example of valid binary blockchain blocks
- Send me email with your selection together with an example block (**25.09.2017**)
  - Wait for confirmation
- Setup repo at GitHub '*PA193_test_parser_XXX*' where XXX is name of your currency
- Write parser implementation together with Tests!

Projects - timeline

1. Write code (GitHub): **10 points** (**5.10.2017 & 9.11.2017**)
   - Demonstrate basic setup, Github repo, Travis CI, example inputs blocks (valid & invalid) [short presentation 5.10.2017 at your seminar]
   - Completed implementation + presentation [9.11.2017, your seminar group, random team member]
2. Review and attack implementations: **7 points** (**14.12.2017**)
   - Review and attack implementations of other teams
   - Initial kickoff together with implementation team [9.11.2017]
   - Report + presentations [14.12.2017]
3. Write patch for a selected bug: **3 points** (**19.1.2018**)
   - Notify me once patch is committed and accepted by target team

- At least 10 points (total) from project are required

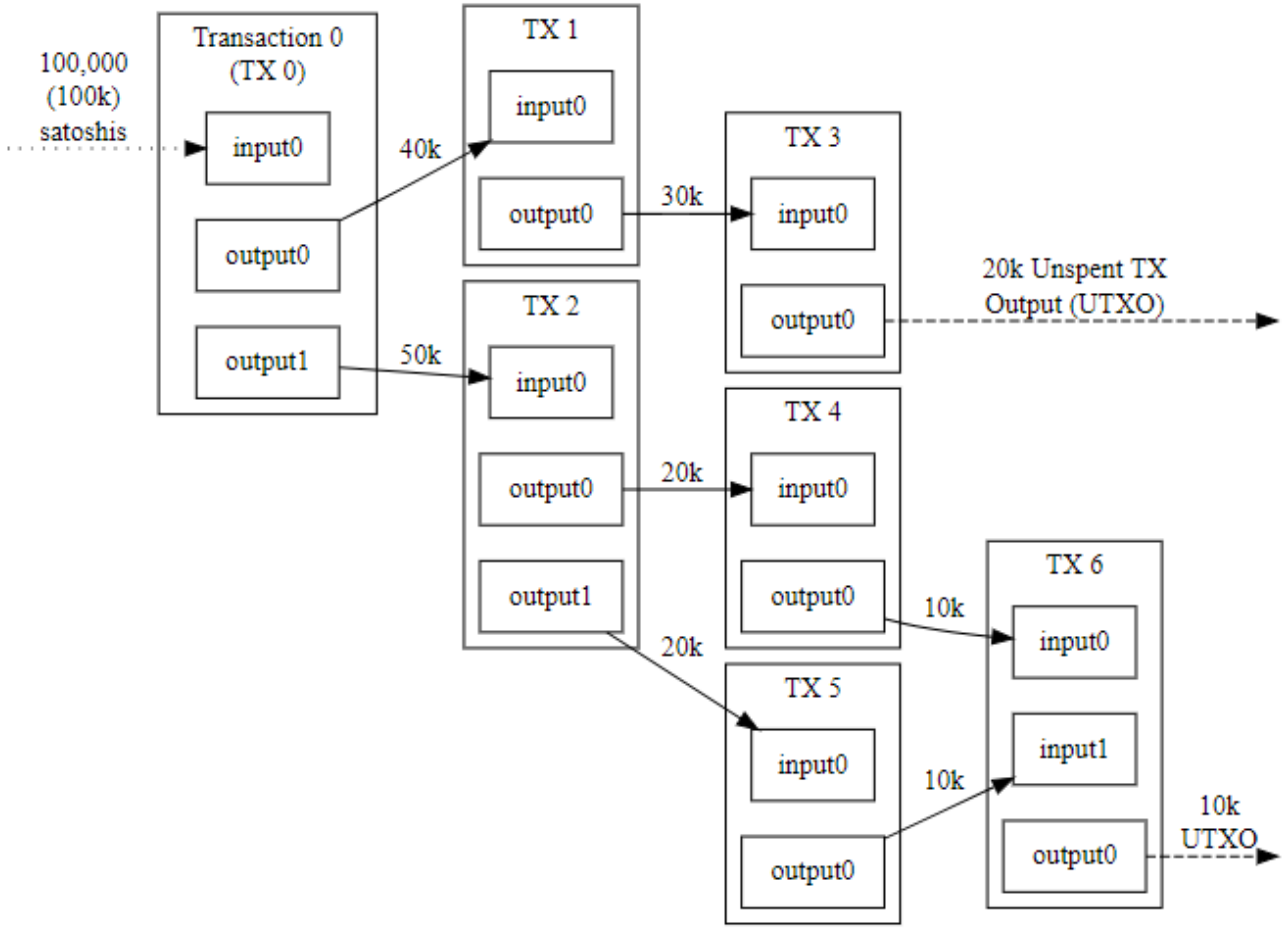
Basics of block chain



Simplified Bitcoin Block Chain

https://bitcoin.org/en/developer-guide#block-chain

Basics of block chain II.



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

https://bitcoin.org/en/developer-guide#block-chain

Bitcoin block (example invalid)

- Bitcoin block 74638 (15

CBlock(hash=000000000790ab3, ver=1, hashPrevBlock=0000000000068d8
nTime=1281891957, nBits=1c0080
CTransaction(hash=012cd8, ver=
CTxIn(COutPoint(000000, -1), c
CTxOut(nValue= 50.51000000, scriptPubKey=0x4F4BA55D1580F8C5A8A
CTransaction(hash=1d5e51, ver=1, vin.size=1, vout.size=2, nLockTime=0
CTxIn(COutPoint(237fe8, 0), scriptSig=0xA87C02384E1F184B79C6AC)
CTxOut(nValue=**92233720368.54275808**, scriptPubKey=OP_DUP OP_H
CTxOut(nValue=**92233720368.54275808**, scriptPubKey=OP_DUP OP_H
vMerkleTree: 012cd8 1d5e51 618eba

Block hash: 000000
Transaction hash: 1d5e512a97

Mining block reward
(was 50BTC at 2010, is 12.50BTC now)

Input transaction (with 0.5BTC)
https://blockexplorer.com/tx/237fe8348fc77ace11049931058abb034c99698c7fe99b1cc022b1365a705d39

2 output transactions (each with $9*10^{10}$ BTC) !!!

Should have been rejected by miners as value(output) >> value(input), but was not!

NOTE: Teams with 4 members are expected to implement full verification of the whole blockchain instead of single block

TEAMS

- Ethereum, C++: https://github.com/kasparjarek/PA193_test_parser_Ethereum
  - Jaroslav Kašpar
  - Šimon Struk
  - Vojta Polasek
- Litecoin, C++: https://github.com/Urvek/PA193_test_parser_litecoin
  - Urvekkumar C Shah
  - Hitesh Lilhare
  - Akhilesh kumar Soni
  - Vikas Lamba

- Novacoin, C++:
- https://github.com/arvindrao7589/PA193_test_parser_Novacoin
  - Arvind Rao <476357@mail.muni.cz>
  - Kuldeep Goyal <goyalkuldeep@gmail.com>
  - Surendra Kumar Yadav <476364@mail.muni.cz>
- Dash, C++: https://github.com/JakubMar/PA193_test_parser_dash
  - Jakub Martinka,
  - Marek Vančík
  - Peter Benčík

- Peercoin, Java: REPO
  - Yevhenii Babichenko , UCO 468589;
  - Ram Singh, UCO 476358;
  - Sushma Verma
- Monero, C++: https://github.com/adamjanovsky/PA193_test_parser_monero/
  - Adam Janovský | 410390,
  - Kristián Kozák | 422361,
  - Jegruš Lysý | 374217.

- Litecoin, C/C++: https://github.com/coolsujit/PA193_test_parser_Litecoin
  - Sujeet Deshmukh - 476359@mail.muni.cz
  - Bhupendra Singh - 476370@mail.muni.cz
  - Nidhi Pokhriyal - 476361@mail.muni.cz

- NEM, C++: REPO
  - Jakub Kremláček <410131@mail.muni.cz>
  - Richard Kalinec

- Qtum, C++: https://github.com/mitko501/PA193_test_parser_Qtum
  - Michal Hajas(422190)
  - Lenka Svetlovská(433637)
  - Andrea Turiaková(422387)
- SixEleven, LANG: https://github.com/dogukanucak/PA193_test_parser_SixEleven
  - Deniz Ağaoğlu (459192@mail.muni.cz)
  - Doğukan Uçak (476348@mail.muni.cz)

- DogdeCoin, Java: https://github.com/securecodingproject/PA193_test_parser_dogecoin
  - Eduardo Roldan
  - Clement Thiallet
  - Barna Bruder
- ZCash, C++: https://github.com/nirajkalra/PA193_test_parser_ZEC
  - Niraj Kalra <476365@mail.muni.cz>
  - Chintan Khanna <476362@mail.muni.cz>
  - Gajraj Kuldeep <476369@mail.muni.cz>