How to send a bitcoin

In Ruby...

Sjors Provoost
sjors@purpledunes.com
Twitter: @provoost

Utrecht.rb
June 3, 2013

Wallets

Paper wallet:

Transfer →

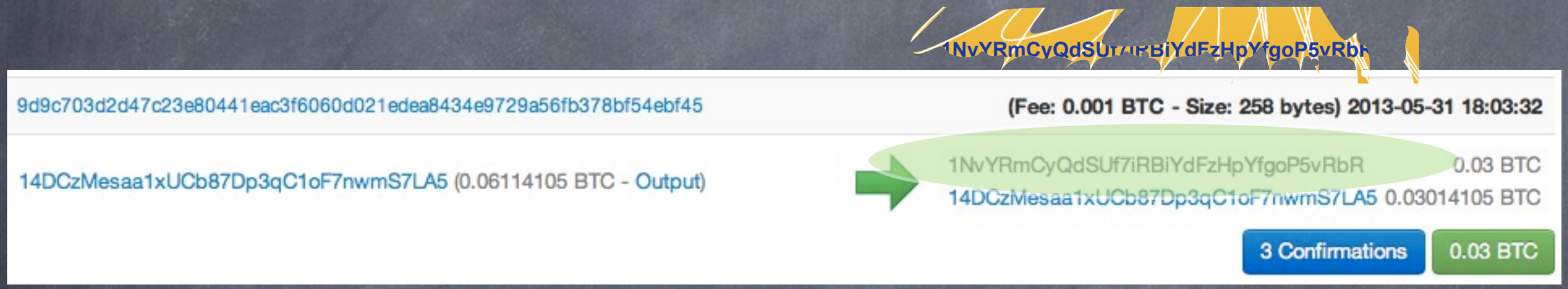
Recipient:

1KHXSZFPDM337XTBEYFBVBS9LZC1BFDU8K

Paper wallet in a safe place...

Inputs

Transactions	
No. Transactions	1
Total Received	0.03 BTC
Final Balance	0.03 BTC



- It's not about balance, it's about inputs
- We received 0.03 btc in a single transaction
- So we have one input worth 0.03 btc

Outputs

1NvYRmCyQdSUf7iRBiYdFzHpYfgoP5vRbR

0.019 ~~0.02~~ btc

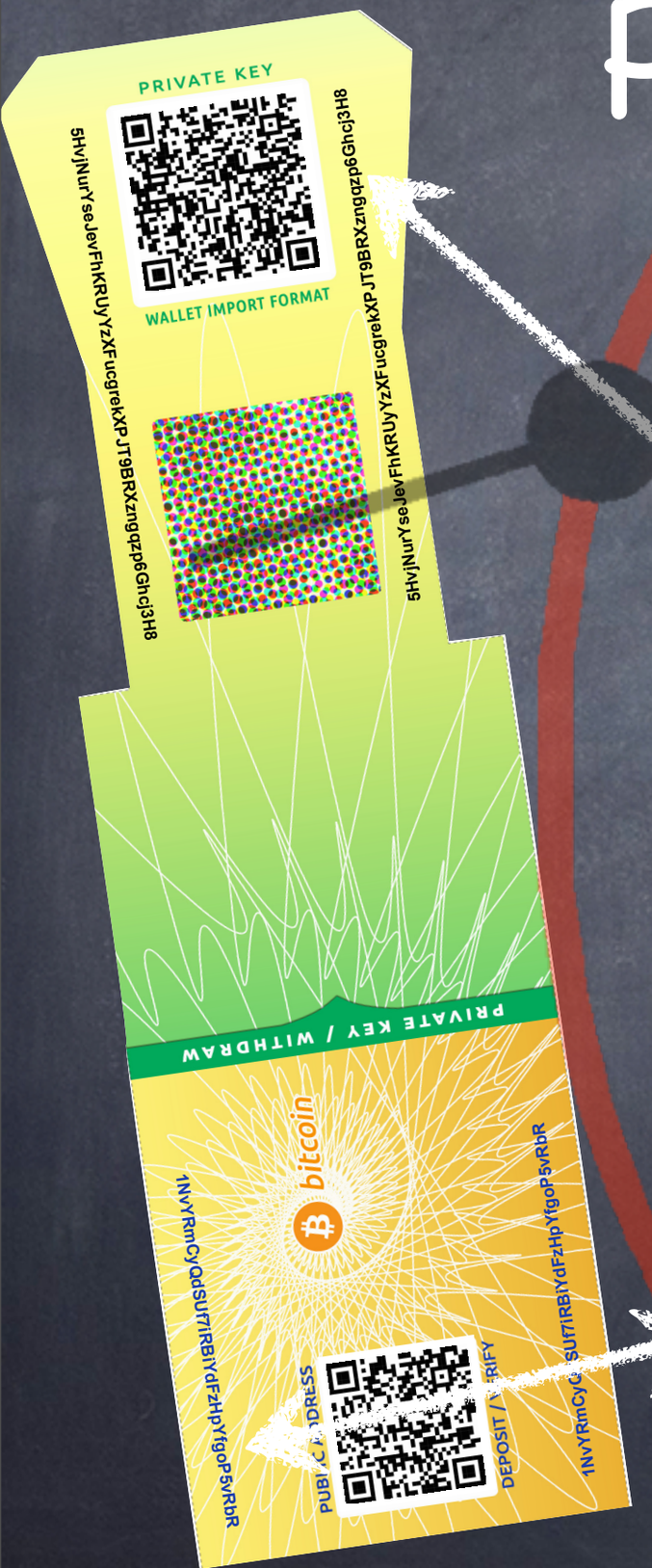
0.01 btc

1NvYRmCyQdSUf7iRBiYdFzHpYfgoP5vRbR

1KHXSZFPDM337XTBEYFBVBS9LZC1BFDU8K

- You must spend the entire input
- Return remainder as change
- Leave something for the miners

Public & private keys



- 256 bit Elliptic Curve Private Key
- Base58 encode ... 890 ... KkLlMmNnOoPp...
- Public key
- SHA-256 & RIPEMD-160
- Add checksum & Base58 encode
- Address

Simplified Transaction

```
{
   "time":1370023412,
   "hash":"9d9c703d2d4....",
   "inputs":[
      {"prev_out":{
         "transaction" : "...."
         "n":1,
         "value":6114105,
         "addr":"14DCzMesaa...",
      }}],
   "out":[
      {"n":0,
      "value":3000000,
      "addr":"1NvYRmC...",
      {"n":1, 1NvYRmCyQdSUrdR
      "value":3014105,
      "addr":"14DCzMe...",
      }],
}
```

- Transaction identifier
- Input is earlier output
- Those outputs had the sender as their destination
- The first output is us, the second is change

Code!

- Get current balance from Blockchain.info

```
@sender = "1NvYRmCyQdSUf7iRBiYdFzHpYfgoP5vRbR"
url = "https://blockchain.info/address/#{ @sender }?format=json"
res = JSON.parse(open(url).read)
@balance = BigDecimal.new(res["final_balance"]) / SATOSHI_PER_BITCOIN
```

- Gather inputs (remember, it's not about balance)

```
url = "https://blockchain.info/unspent?active=#{@sender}&format=json"
res["unspent_outputs"].each do |output|
    @inputs <<  {
      previousTx: [output["tx_hash"]].pack("H*").reverse.unpack("H*")
[0],        index: output["tx_output_n"],
       scriptSig: nil # We'll sign it later
    }
    amount = BigDecimal.new(output["value"]) / SATOSHI_PER_BITCOIN
    input_total += amount
    break if input_total >= @amount + @transaction_fee
end
```

Outputs

- Receipient and change
- Script says: "he who can sign another transaction with this public key, may spend"
- Low level stuff like the length of the pub. key

```
@outputs = [
  {
   value: @amount,
   scriptPubKey: "OP_DUP OP_HASH160 " +
                 (recipient_pubkey_hex.size / 2).to_s(16) +
                 " " + recipient_pubkey_hex +
                 " OP_EQUALVERIFY OP_CHECKSIG "
  },{
   value: @change,
   scriptPubKey: "OP_DUP OP_HASH160 " ....
```

Serialize

```
{:version=>1,
 :in_counter=>1,
 :inputs=>
  [{:previousTx=>
     "9d9c703d2d47c23e80441eac3f6060d021edea8434e9729a56fb378bf54ebf45",
    :index=>0,
    :scriptLength=>25,
    :scriptSig=>
     "OP_DUP OP_HASH160 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c OP_EQUALVERIFY OP_CHECKSIG ",
    :sequence_no=>"ffffffff"}],
 :out_counter=>2,
 :outputs=>
  [{:value=>#<BigDecimal:7fd51c32bf60,'0.1E-1',9(18)>,
    :scriptPubKey=>
     "OP_DUP OP_HASH160 14 c8a73488183dd49f63a11dea0a3b242ae70942d2 OP_EQUALVERIFY OP_CHECKSIG "},
   {:value=>#<BigDecimal:7fd51c26de98,'0.2E-1',9(27)>,
    :scriptPubKey=>
     "OP_DUP OP_HASH160 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c OP_EQUALVERIFY OP_CHECKSIG "}],
 :lock_time=>0,
 :hash_code_type=>"01000000"}
```

Our address (hex)

OP_HASH160 → Ripemd160(Sha256(...)) → 14

Verify signature of whoever spends this output

```
01000000
01
45bf4ef58b97fb569a72e93484eaed21d060603fac1e44803ec2472d3d709c9d 00000000
19
76 a9 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c 88 ac  ffffffff
02
40420f0000000000
19
76 a9 14 c8a73488183dd49f63a11dea0a3b242ae70942d2 88 ac
80841e0000000000
19
76 a9 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c 88 ac
00000000
01000000
```

Sign

- Chicken-egg problem

```
01000000
01
45bf4ef58b37fb569a72e93484eaed21d060603fac1e44803ec2472d3d709c9d 00000000
19
76 a9 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c 88 ac ffffffff
02
....
```

Our signature needed to spend this

- Sha256 hash

```
6a9a0ce373eaa03b1da3cd476abfed1d69a142db0a912fe0443fc8cfb4189abe
```

- Sign and insert into transaction

```
01000000
01
45bf4ef58b37fb569a72e93484eaed21d060603fac1e44803ec2472d3d709c9d 00000000
8b
48

3045022002338998a410c7da841e83a33bdf78c5322896b605d5d9718638658d22120389022100ea077a2ecbb4b0d862d38426ae6
d71dc947eaa649e93dfb63e5023182c59222f
01 41

047aeeaaffd99f46af7ba63e5fcd42acd2e421eb54c1af3e042311dd2da8a7be97cae06ff4d12a6ad71d4698be307dbd0ed275253
146601328b0d8645bab16cbe8
ffffffff
02
....
```

Signature

Demo

- Download and run with parameters
  - your address
  - your private key
- Optionally change the addressee, me :-)

```
git clone http://gist.github.com/5574485.git
ruby bitcoin-pay.rb 1NvYRmCyQdSUf7iRBiYdFzHpYfgoP5vRbR
                    5HvjNurYseJevFhKRUyYzXFucgrekXPJT9BRXzngqzp6Ghcj3H8
```

- Copy result and paste into https://blockchain.info/pushtx

If demo doesn't work… (1/2)

```
About to send 0.01 bitcoins from 1NvYRm... to 1KHxSz...
Fetching the current balance for NvYRm from blockchain.info...
Current balance of sender: 0.03 BTC
Using 0.03 from output 0 of transaction 45bf4e...
Spend 0.01 and return 0.02 as change.
Public key matches private key, so we can sign the transaction...
Readable version of the transaction (numbers in strings are hex, otherwise decimal)

{:version=>1,
 :in_counter=>1,
 :inputs=>
  [{:previousTx=>
     "9d9c703d2d47c23e80441eac3f6060d021edea8434e9729a56fb378bf54ebf45",
    :index=>0,
    :scriptLength=>25,
    :scriptSig=>
     "OP_DUP OP_HASH160 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c OP_EQUALVERIFY OP_CHECKSIG ",
    :sequence_no=>"ffffffff"}],
 :out_counter=>2,
 :outputs=>
  [{:value=>#<BigDecimal:7fd51c32bf60,'0.1E-1',9(18)>,
    :scriptPubKey=>
     "OP_DUP OP_HASH160 14 c8a73488183dd49f63a11dea0a3b242ae70942d2 OP_EQUALVERIFY OP_CHECKSIG "},
   {:value=>#<BigDecimal:7fd51c26de98,'0.2E-1',9(27)>,
    :scriptPubKey=>
     "OP_DUP OP_HASH160 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c OP_EQUALVERIFY OP_CHECKSIG "}],
 :lock_time=>0,
 :hash_code_type=>"01000000"}
```

If demo doesn't work… (2/2)

```
Hex unsigned transaction:
01000000
01
45bf4ef58b37fb569a72e93484eaed21d060603fac1e44803ec2472d3d709c9d 00000000
19
76 a9 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c 88 ac  ffffffff
02
40420f0000000000
19
76 a9 14 c8a73488183dd49f63a11dea0a3b242ae70942d2 88 ac
80841e0000000000
19
76 a9 14 f07aebdb79282ef81eaaba45a35bb82e340fb97c 88 ac
00000000
01000000

Hash that we're going to sign: 6a9a0ce373eaa03b1da3cd476abfed1d69a142db0a912fe0443fc8cfb4189abe

Hex signed transaction: (258 bytes)

0100000001 45bf4ef58b37fb569a72e93484eaed21d060603fac1e44803ec2472d3d709c9d 000000008b48304502201e9f2
c93ed4a3f57db4b4f79c1b57c616a468f85e1b89fa633de1b3a32570b67022100a5884cdbd43ca1db025f680a30d30a17cc
b264a293706ad0d5736cad173c30bb0141047aeeaaffd99f46af7ba63e5fcd42acd2e421eb54c1af3e042311dd2da8a7be9
7cae06ff4d12a6ad71d4698be307dbd0ed2752531466013 28b0d8645bab16cbe8ffffffff 0240420f000000000 1976a914
c8a73488183dd49f63a11dea0a3b242ae70942d288ac 80841e000000000 1976a914f07aebdb79282ef81eaaba45a35bb82
e340fb97c88ac 00000000

Copy paste the transaction and transmit it at https://blockchain.info/pushtx
```

Questions?

- Shameless plug
- Get notified about this transaction in your iPhone's Passbook:
- bitcoin-passbook.com

Sjors Provoost
sjors@purpledunes.com
Twitter: @provoost

Sources

- Ruby script to transfer Bitcoin: https://gist.github.com/Sjors/5574485
- Paper wallet: https://bitcoinpaperwallet.com
- General Bitcoin info: http://bitcoin.it
- Elliptic curve image: http://www.hpl.hp.com/research/info_theory/images/curveplot.gif