Bitcoin Smart Contracts

Saravanan Vijayakumaran
sarva@ee.iitb.ac.in

Department of Electrical Engineering
Indian Institute of Technology Bombay

August 26, 2019

Smart Contracts

- Computer protocols which help execution/enforcement of regular contracts
- Minimize trust between interacting parties
- Hypothetical example: Automatic fine for noise pollution
  - IITB hillside community hall parties use loudspeakers
  - Party organizers pay bitcoin security deposit
  - If noise rules violated, deposit distributed to nearby residents
- Two actual examples
  - Escrow
  - Micropayments

Escrow Contract

Problem Setup

- Alice wants to buy a rare book from Bob
- Alice and Bob live in different cities
- Bob promises to ship the book upon receiving Bitcoin payment
- Alice does not trust Bob
- Alice proposes an escrow contract involving a third party Carol

Escrow Contract

- Alice requests public keys from Bob and Carol
- Alice pays *x* bitcoins to a 2-of-3 multisig output

OP_2 <PubKeyA> <PubKeyB> <PubKeyC> OP_3 OP_CHECKMULTISIG

- Bob ships book once Alice's transaction is confirmed
- Bitcoins can be spent if **any two of the three** provide signatures
- Any of the following scenarios can occur
  - Alice receives book.
    Alice and Bob sign.
  - Alice receives the book but refuses to sign.
    Bob provides proof of shipment to Carol.
    Bob and Carol sign.
  - Bob does not ship the book to Alice.
    Bob refuses to sign refund transaction.
    Alice and Carol sign.
- Escrow contract fails if Carol colludes with Alice or Bob
- Also proof of shipment is not proof of contents

Lock Times

Transaction Lock Time

Regular Transaction Format

nVersion
Number of Inputs N
Input 0
⋮
Input $N-1$
Number of Outputs M
Output 0
⋮
Output $M-1$
nLockTime

- `nLockTime` is a 4-byte field which specifies the earliest time the transaction can be included in a block

`nLockTime` Values

- If `nLockTime` $< 5 \times 10^8$, then it is interpreted as a block height
  - Transaction with `nLockTime` = 600,000 will not be included in any block with height < 600,000
- If `nLockTime` $\geq 5 \times 10^8$, then it is interpreted as a Unix time
  - Unix time = Number of seconds since Jan 1, 1970 12:00AM UTC
  - Unix time of 1,514,797,200 = 9:00 AM on January 1, 2018
  - Transaction with Unix time lock time will not be included unless the median-time-past of the latest block exceeds the `nLockTime` value
  - The median-time-past of a block at height h is the median of the `nTime` values in the 11 blocks at heights $h, h-1, \ldots, h-10$.
  - The `nTime` field of a candidate block at height N must exceed the median-time-past of the block at $N-1$.
- What if we need block height $\geq 5 \times 10^8$ or Unix time < 5×10^8?
  - It would take 9,500 years to reach block height 5×10^8
  - Unix time of 5×10^8 is 12:53AM on Nov 5, 1985

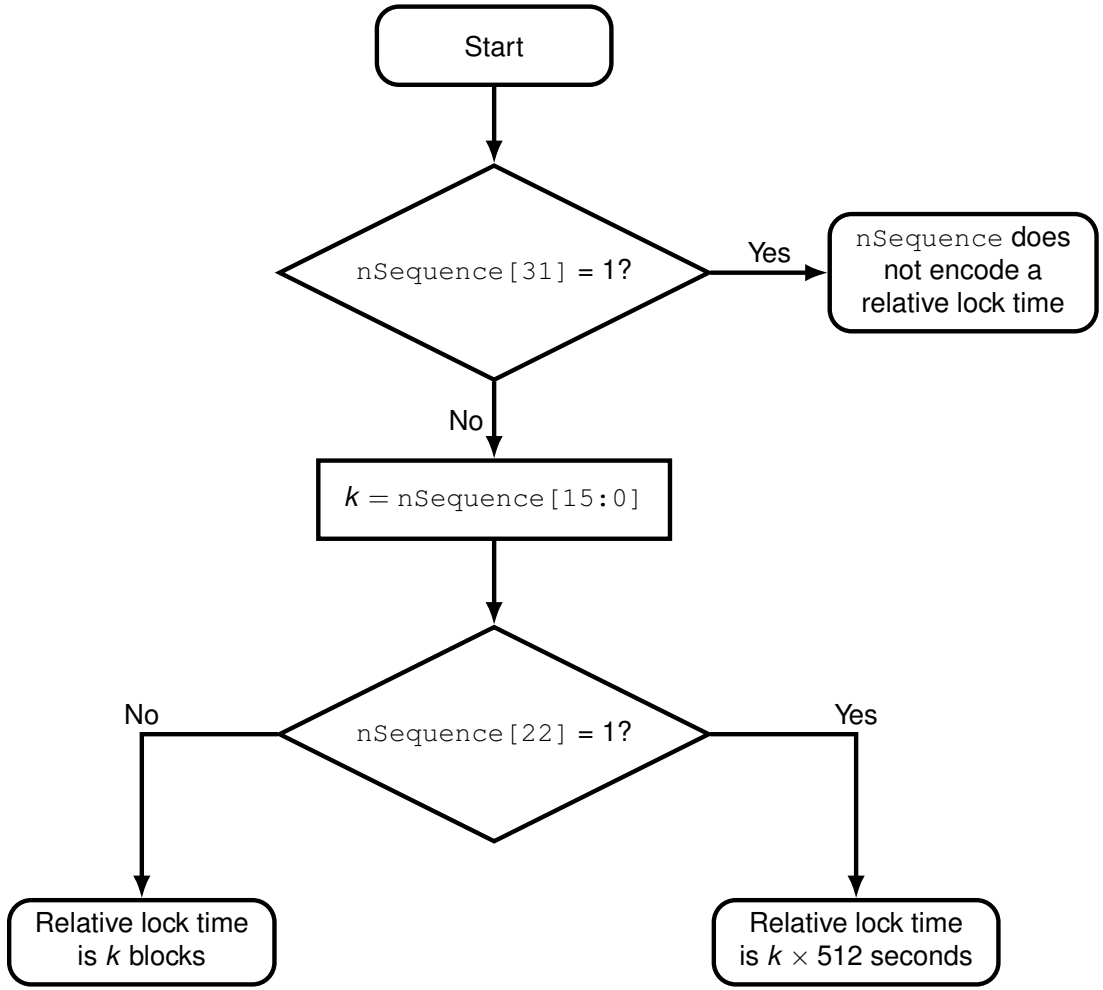
Relative Lock Times

Input Format

`hash`
`n`
`scriptSigLen`
`scriptSig`
`nSequence`

- The 4-byte `nSequence` field is used to specify a *relative lock time* of an input
- Can have units which of either blocks or seconds
- Suppose the relative lock time of an input is k blocks
- If the output which is being unlocked by this input is in block K, then a transaction containing this input cannot be included in a block whose height is less than $K + k$
- A similar condition holds for relative lock time in seconds

Relative Lock Time from `nSequence` Value



- Maximum relative lock time in blocks is $2^{16} - 1 = 65,535$ blocks ≈ 1.25 years
- Maximum relative lock time in seconds is $(2^{16} - 1) \times 512 = 33,553,920$ seconds ≈ 1.06 years

Micropayments

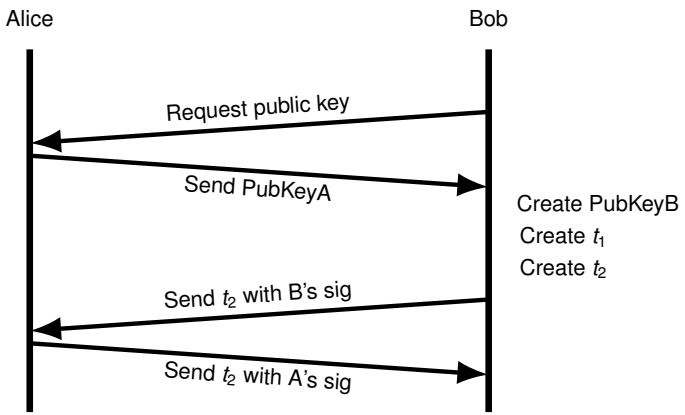
Problem Setup

- Bitcoin transaction fees make small payments expensive
- Micropayments contract can aggregate small payments
- Alice offers proofreading and editing services online
- She accepts bitcoins as payments
- Clients email documents to Alice
- Alice replies with typos and grammatical errors
- Alice charges a fixed amount of bitcoins per edited page
- To avoid clients refusing payment, Alice uses micropayments contract
- Suppose Bob wants a 100 page document edited
- Alice charges 0.0001 BTC per page
- Bob expects to pay a maximum of 0.01 BTC to Alice

Micropayments Contract (1/3)

Creating Refund Transaction

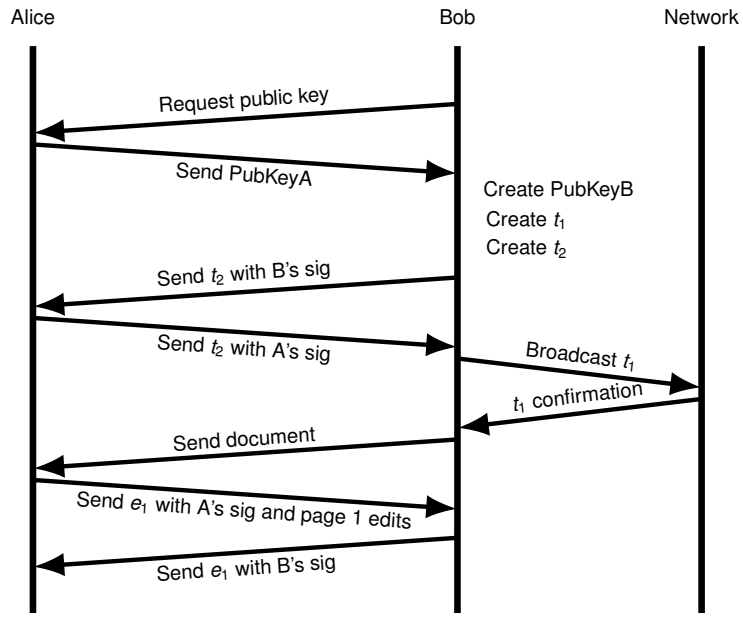
- Bob requests a public key from Alice
- Bob creates a transaction t_1 which transfers 0.01 bitcoins to a 2-of-2 multisig output
- Bob does not broadcast t_1 on the network
- Bob creates a refund transaction t_2 which refunds the 0.01 BTC
- A relative lock time of n days is set on t_2
- Bob includes his signature in t_2 and sends it to Alice
- If Alice refuses to sign, Bob terminates the contract
- If Alice signs t_2 and gives it Bob, he has the refund transaction



Micropayments Contract (2/3)

Getting Paid for First Page Edits

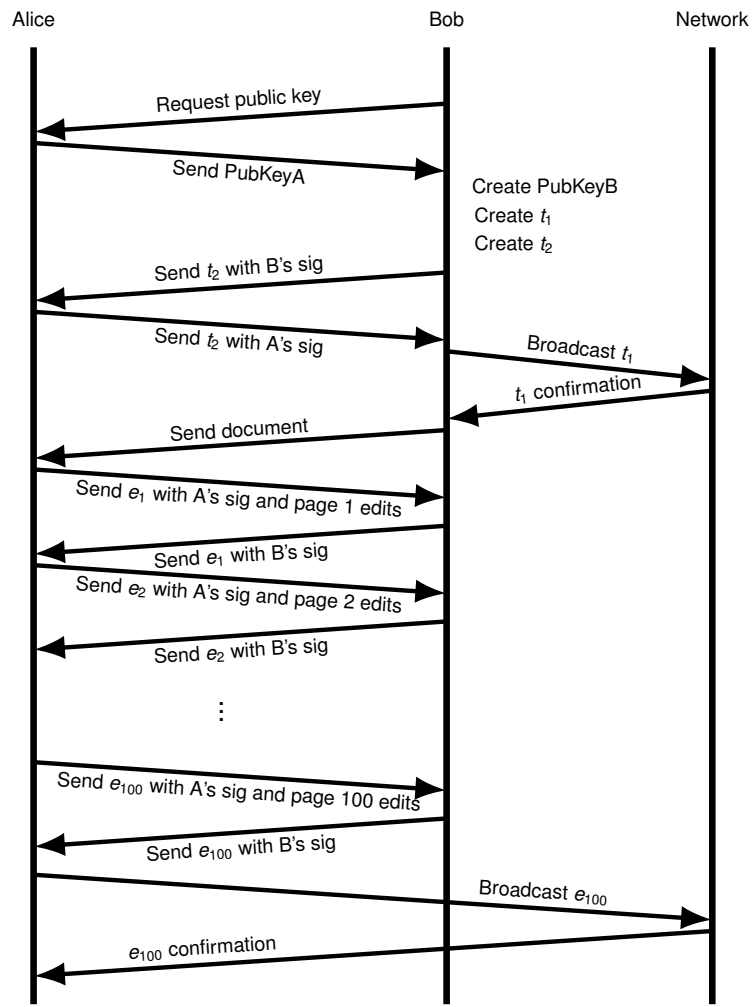
- Bob broadcasts t_1 on the network
- Once t_1 is confirmed, he sends Alice his document
- Alice edits only the first page of the document
- She creates a transaction e_1 which unlocks t_1 and pays her 0.0001 BTC and 0.0099 BTC to Bob
- Alice signs e_1 and sends it to Bob along with the first page edits
  - If Bob refuses to sign e_1, then
    - Alice terminates the contract.
    - Bob broadcasts t_2 after lock time expires
  - If Bob signs e_1 and returns it to Alice, then Alice is guaranteed 0.0001 bitcoins if she broadcasts e_1 before lock time on t_2 expires.



Micropayments Contract (3/3)

Getting Paid for Second Page, Third Page . . .

- Alice edits the second page of the document
- She creates a transaction e_2 which unlocks t_1 and pays her 0.0002 BTC and 0.0098 BTC to Bob
- Alice signs e_2 and sends it to Bob along with the second page edits
  - If Bob refuses to sign e_2, then Alice terminates the contract.
    Alice broadcasts e_1 and receives 0.0001 BTC.
  - If Bob signs e_2 and returns it to Alice, then Alice is guaranteed 0.0002 bitcoins if she broadcasts e_2 before lock time on t_2 expires.
- Alice continues sending edited pages along with transactions requesting cumulative payments
- She has to finish before the refund transaction lock time expires



Key Takeaways

- Smart contracts reduce the need for trust
- Bitcoin's scripting language enables some smart contracts
- Not powerful enough to express complex contracts

References

- Chapters 5, 6 of *An Introduction to Bitcoin*, S. Vijayakumaran, `www.ee.iitb.ac.in/~sarva/bitcoin.html`