



CSE 127: Computer Security

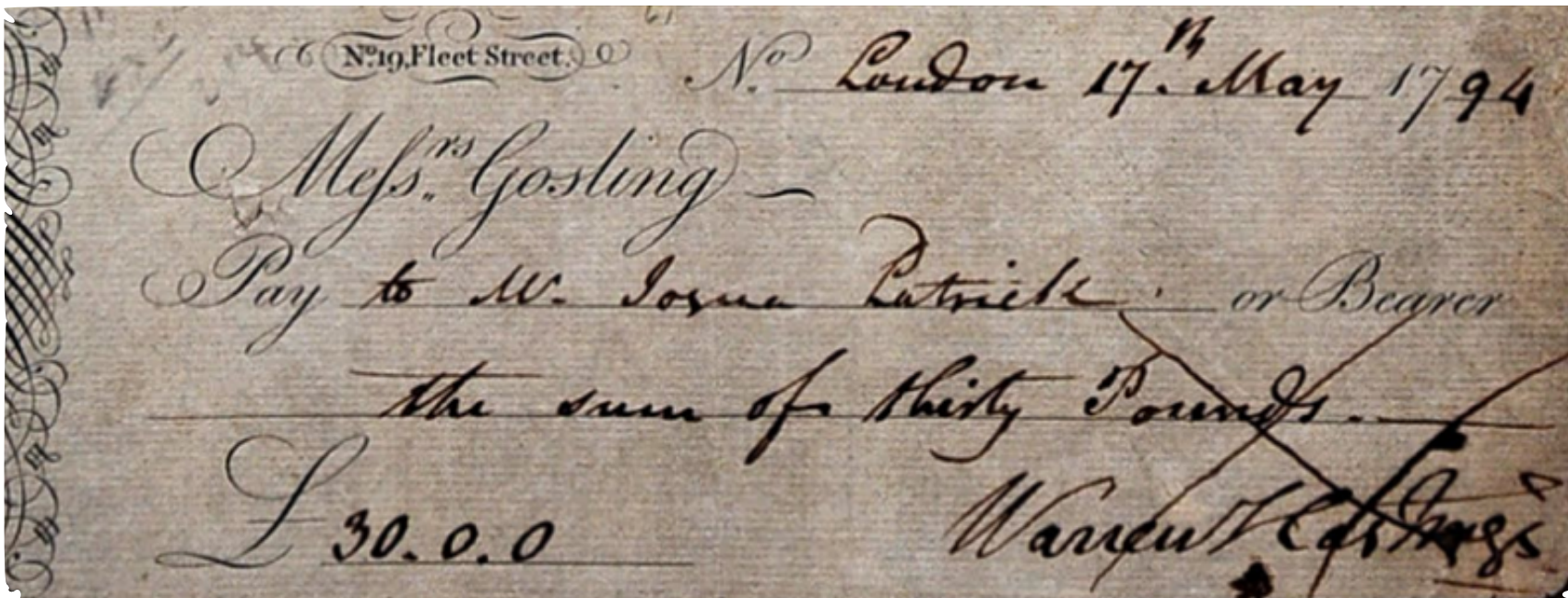
Bitcoin

Kirill Levchenko

December 7, 2017

Electronic Payment System

- ❖ *Payment system* allows people to pay each other
- ❖ **Example:** paper checks allow you to draw on money stored in your account to pay someone else
- ❖ Can we create an electronic analog?



Electronic Analog of Checks

❖ Checks guarantee

- To payer: her bank will only honor a properly signed check
- To payee: payer's bank will honor check and issue payment
 - Subject to availability of funds

❖ How do we provide this electronically?

❖ Main concern is integrity/authenticity

Example of a Check

JAMES C. MORRISON

1765 SHERIDAN DRIVE
YOUR CITY, STATE 12345

00-6789/0000

1 0 1

DATE Dec. 7, 2017

PAY TO THE
ORDER OF

UC Regents

\$

127.15

— One hundred twenty-seven and 15/100 — DOLLARS



Security Features
Included.
Details on Back.



*Commerce Bank*SM

MEMO _____

James Morrison (signed)

MP

+ 1 0 1 1 3 0 0 1 4 2 1 1 2 3 4 5 6 7 8 1 1 0 1 0 1

Elements of a Check

JAMES C. MORRISON

1765 SHERIDAN DRIVE
YOUR CITY, STATE 12345

00-6789/0000

1 0 1

DATE date to pay

PAY TO THE
ORDER OF

Payee name

\$

Amount

Amount (spelled out to make tampering obvious)

DOLLARS



Security Features
Included.
Details on Back.



Commerce BankSM

MEMO _____

Signature (authentication)

MP

+ 1 0 1 1 3 0 0 1 4 2 1 1 2 3 4 5 6 7 8 1 1 0 1 1

Electronic Analog of Checks

- ❖ Create digital document giving:
 - Payer name and account number
 - Payee
 - Amount
- ❖ Sign with payer's digital signature
 - Tamper-proof
 - Authenticated

Bitcoin Transactions

Transaction View information about a bitcoin transaction

2022c081e42c4ccc5efec883a3c0b7731d850b5f2560bc880d575a13869c9795

1ChLzzpgHBXq7jxPEbSU3Y57pfbVubwwcu
1KDQ1ssN25DNYHK1mb2DaJUu1YxaDoJkRs



122p8qwV6PNXBFE2YaoVx1ZZZ9h1ckR2b9 0.7 BTC
1LjSF3WB3kcm8zm5k9mKCBWQYniquFDcoE 0.02639745 BTC

7 Confirmations

0.72639745 BTC

Summary

Size	438 (bytes)
Received Time	2013-06-04 21:34:28
Included In Blocks	239771 (2013-06-04 21:34:50 +0 minutes)
Confirmations	7 Confirmations
Relayed by IP	5.9.24.81 (whois)
Visualize	View Tree Chart

Inputs and Outputs

Total Input	0.72689745 BTC
Total Output	0.72639745 BTC
Fees	0.0005 BTC
Estimated BTC Transacted	0.7 BTC
Scripts	Show scripts & coinbase

1ChLzzpgHBXq7jxPEbSU3Y57pfbVubwwcu

1KDQ1ssN25DNYHK1mb2DaJUu1YxaDoJkRs

122p8qwV6PNXBFE2YaoVx1ZZZ9h1ckR2b9

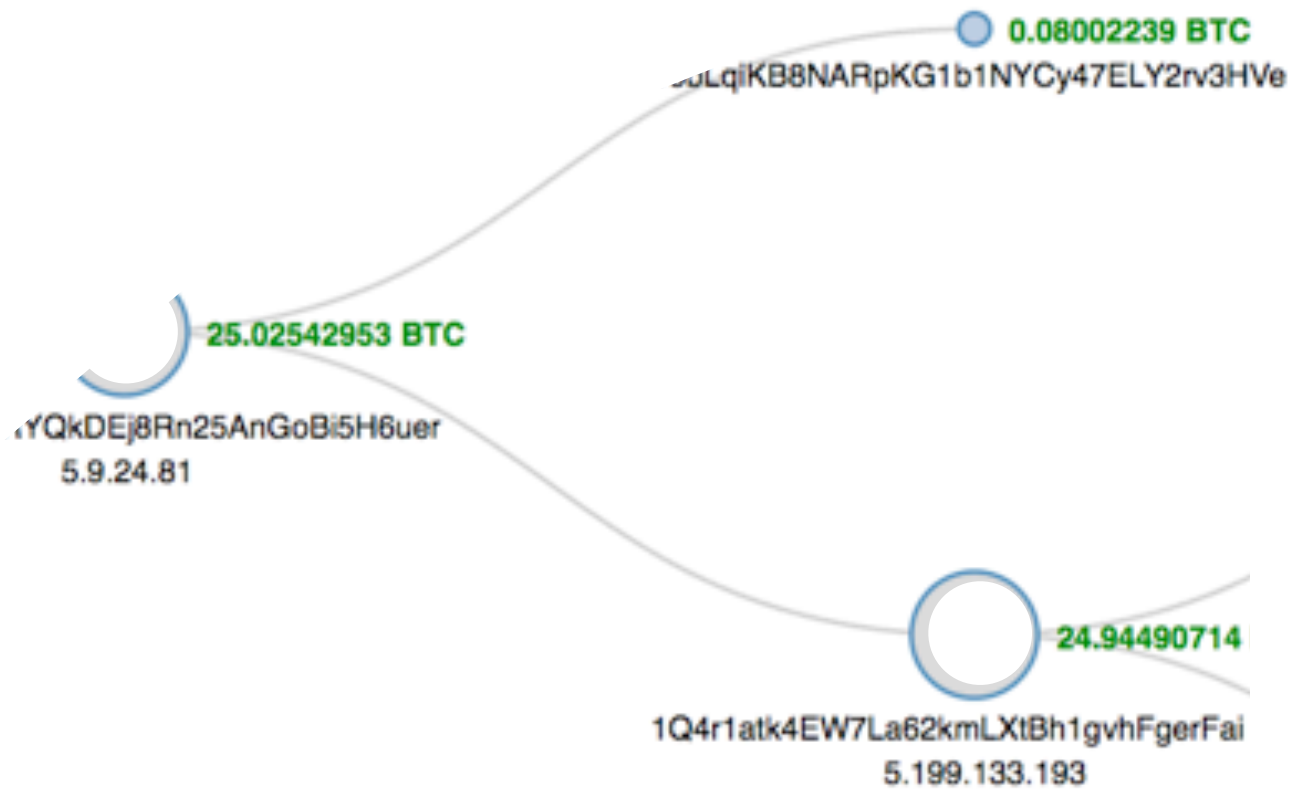
0.7 BTC

1LjSF3WB3kcm8zm5k9mKCBWQYniquFDcoE

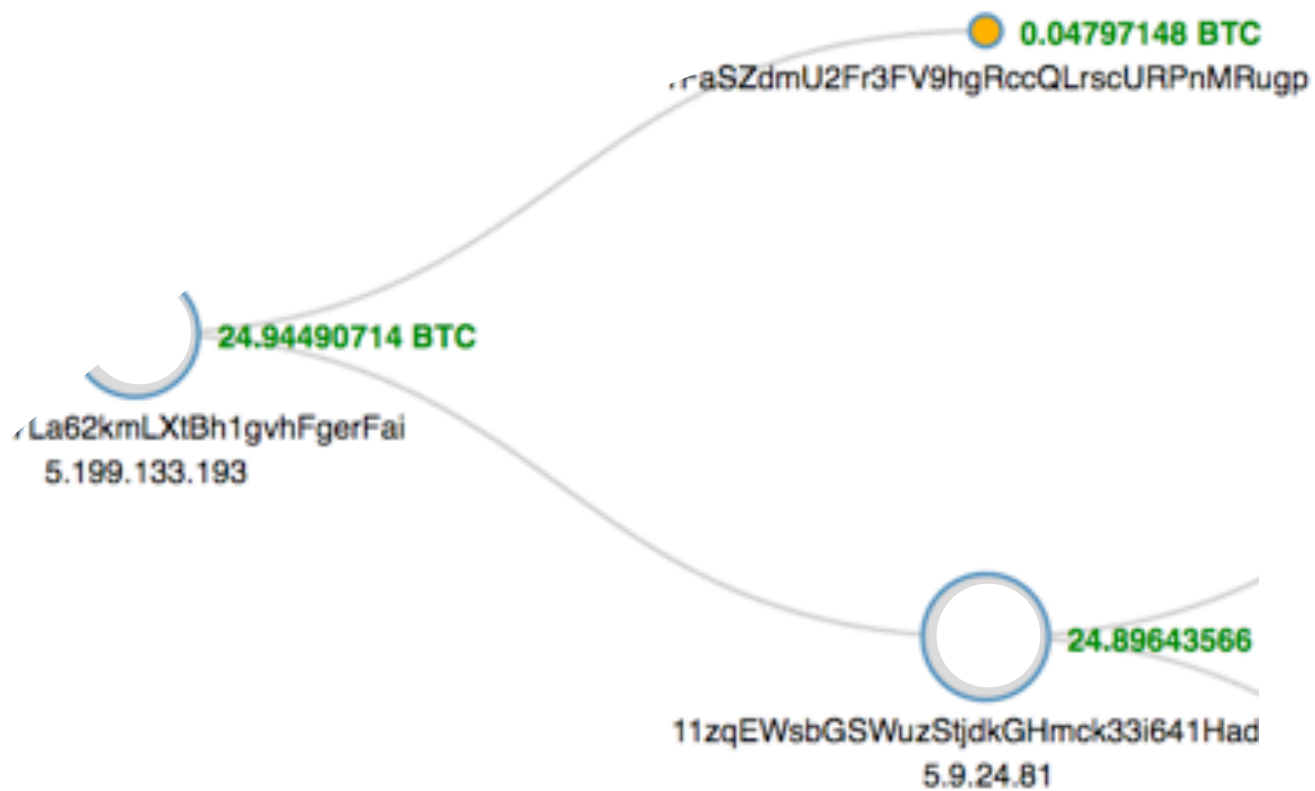
0.02639745 BTC

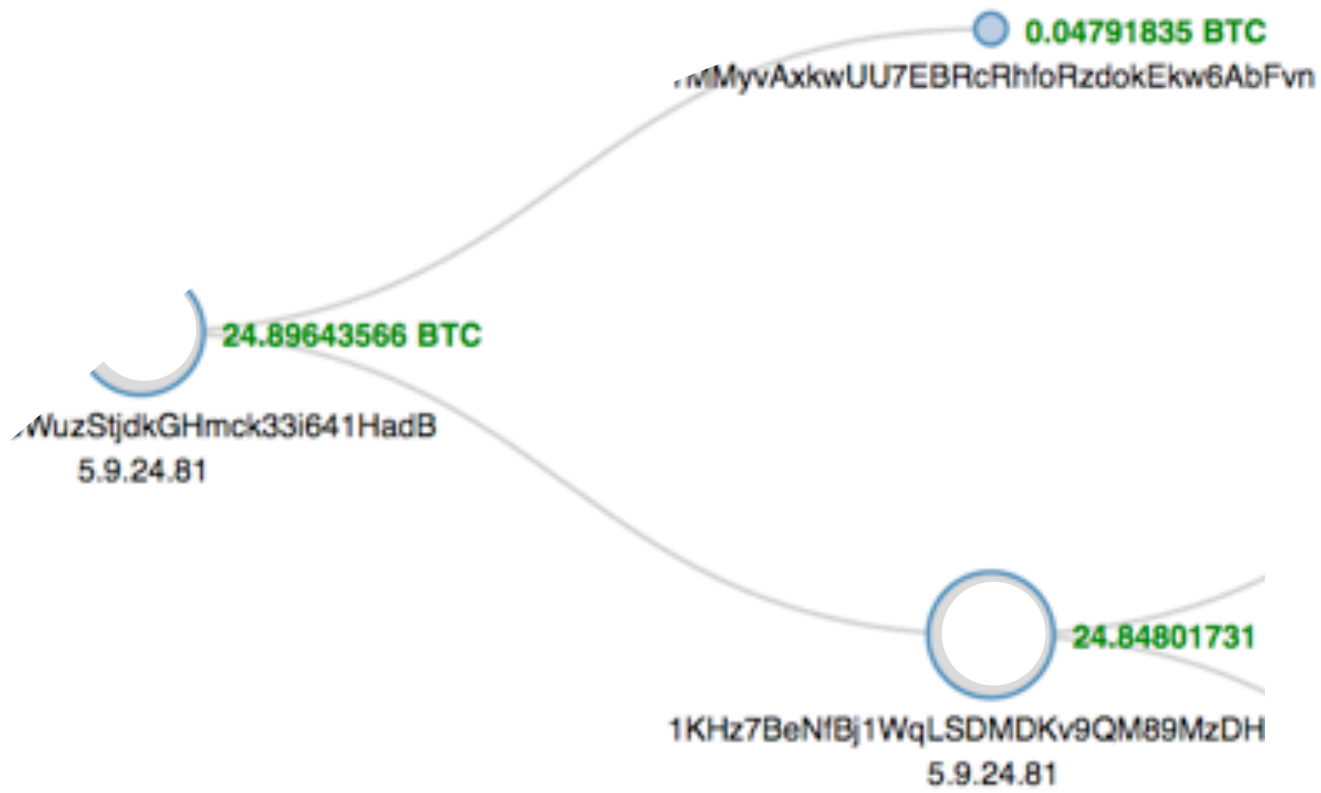
7 Confirmations

0.72639745 BTC



(change)





Problem: Double Spending

- ❖ Transactions draw on output of prior transactions
- ❖ How do we ensure there is no double-spending?
 - Double spending: spending output of same transaction twice
- ❖ Make entire ledger public!
 - Anyone can verify if transaction has been spent
 - Need to commit transaction to ledger for it to be valid
- ❖ All participants in Bitcoin system have copy of ledger
 - Can browse the ledger at blockchain.info

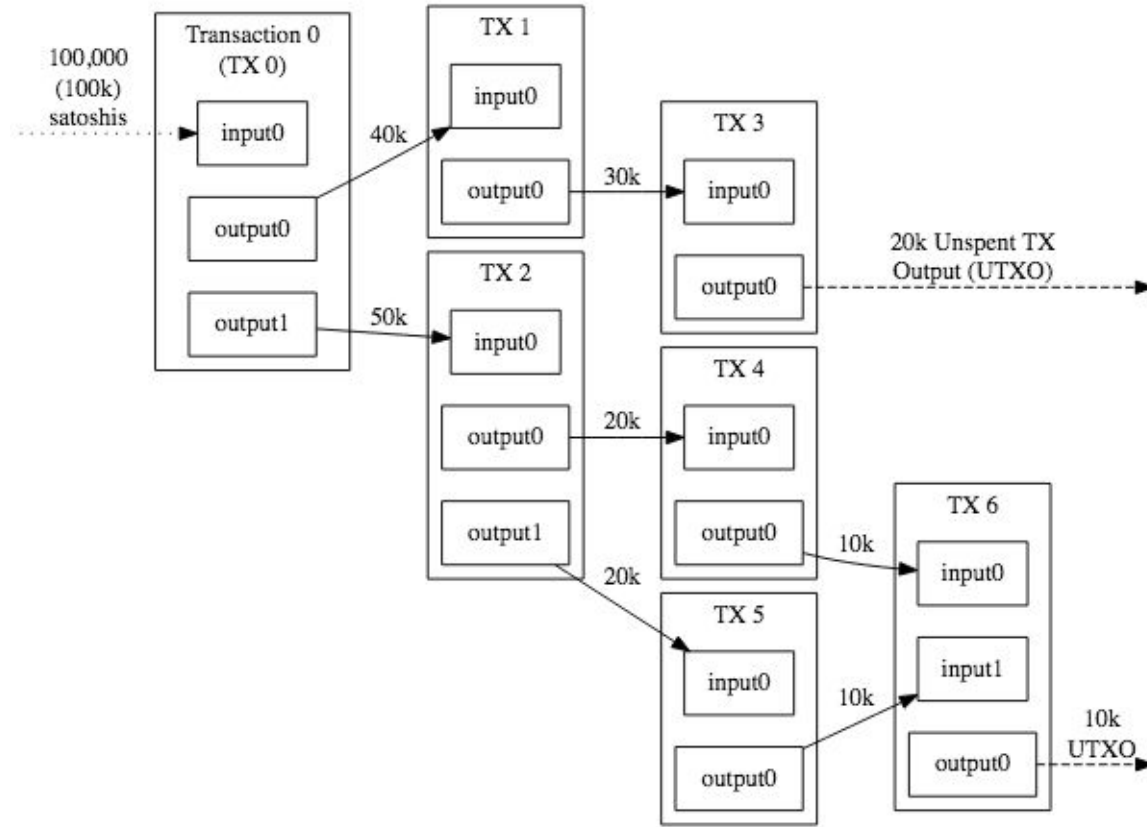


BASIC CONCEPTS - TRANSACTIONS

Decal Lecture 2

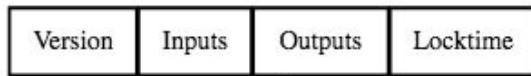
Source: [Bitcoin Developer Guide](#)

- Maps input addresses to output addresses
- Typical tx: one input, two outputs
- Contains signature of owner of funds

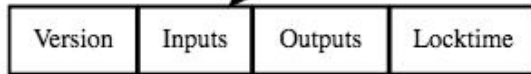


Each input spends a previous output

The Main Parts Of Transaction 0

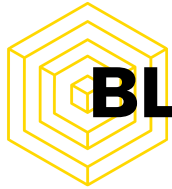


The Main Parts Of Transaction 1



Each output waits as an Unspent TX Output (UTXO) until a later input spends it

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



BLOCKS AND BLOCKCHAIN

Decal Lecture 2

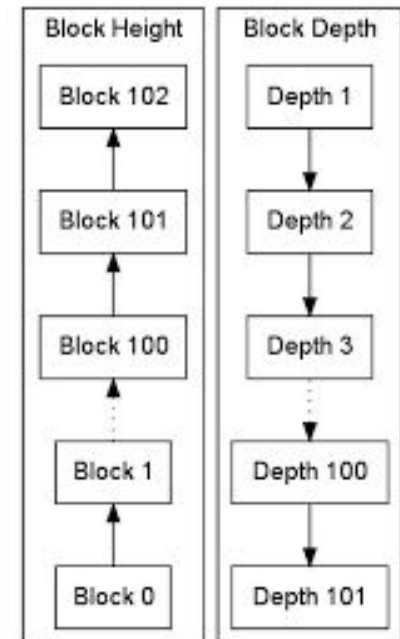
Blocks

- Contains an ordered bunch of transactions
 - Timestamps the transactions, are **immutable**
- Each block References a previous block

Blockchain

- The entire series of blocks 'chained' together

32



Block Height Compared To Block Depth

Source: [Bitcoin Developer Guide](#)





MINING SKETCH - FINDING BLOCKS

Decal Lecture 2

Components hashed together:

- Merkle Root
 - 'summary' of the transactions in the block
- Hash of previous block
- Nonce
 - Randomness of SHA-256 is useful here!

Formally:

- Output = $\text{SHA-256}(\text{Merkle Root} + \text{SHA-256}(\text{PreviousBlock}) + \text{Nonce})$
- Solution (Proof-of-work): an output that contains a requisite number of leading 0 bits
 - The number of 0 bits is the **difficulty**
 - Difficulty adjusts every every 2016 blocks* to regulate block creation
 - *technically every 2015 blocks

