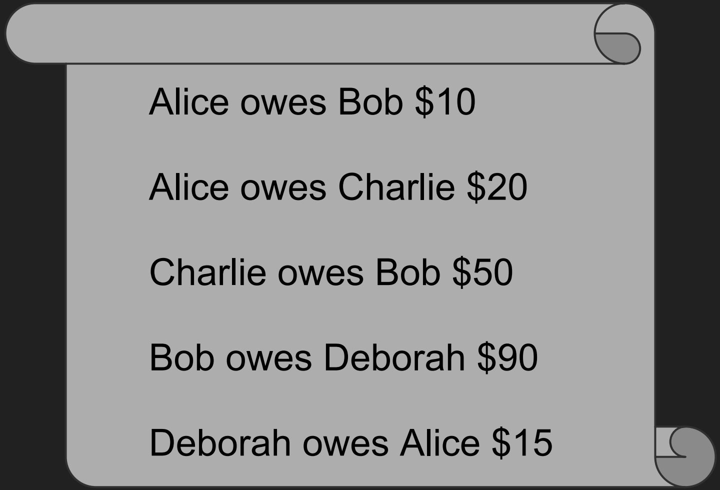
Cryptocurrency Intro

Cryptocurrency: what is it?

Fundamentally, it’s a **ledger**.



Cryptocurrency: what is it?

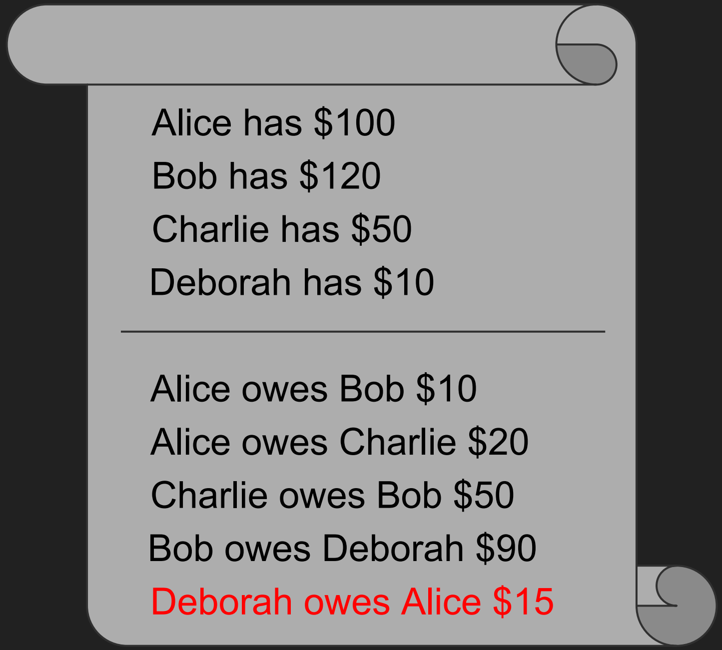
It’s **public**.

Issues that arise:

- Privacy?
- How do we know both parties agree to a transaction? And how do we know people aren’t impersonating others?
    - Digital Signatures -- more on this later!
- How do we know everyone will pay their debts?
    - Don’t allow people to go net negative.

Cryptocurrency: what is it?

It's **public**.

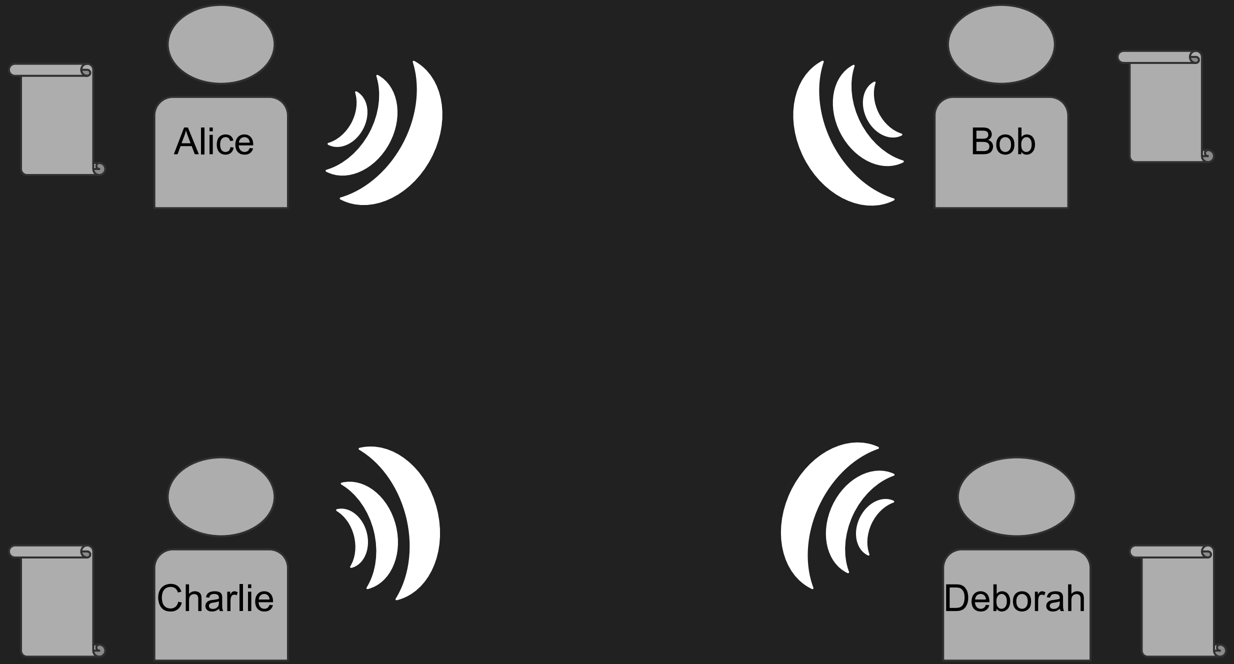


Cryptocurrency: what is it?

It’s not tied to any other currency.

Cryptocurrency: what is it?

It's **decentralized**.



Cryptocurrency: what is it?

It's **decentralized**.

Issues that arise:

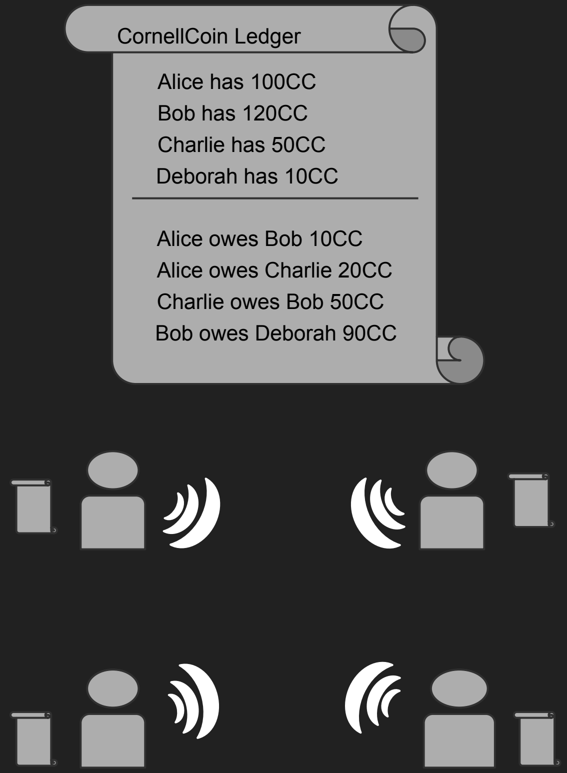
- What is the source of truth? How do we know that everyone has the same ledger?
  - Need to guarantee some kind of consensus.
- How is new money introduced to the system? And when? And to who?
  - "Work" -- more on this later!

Cryptocurrency: what is it?

It's a **ledger**.

It's **public**.

It's **decentralized**.



Digital Signatures

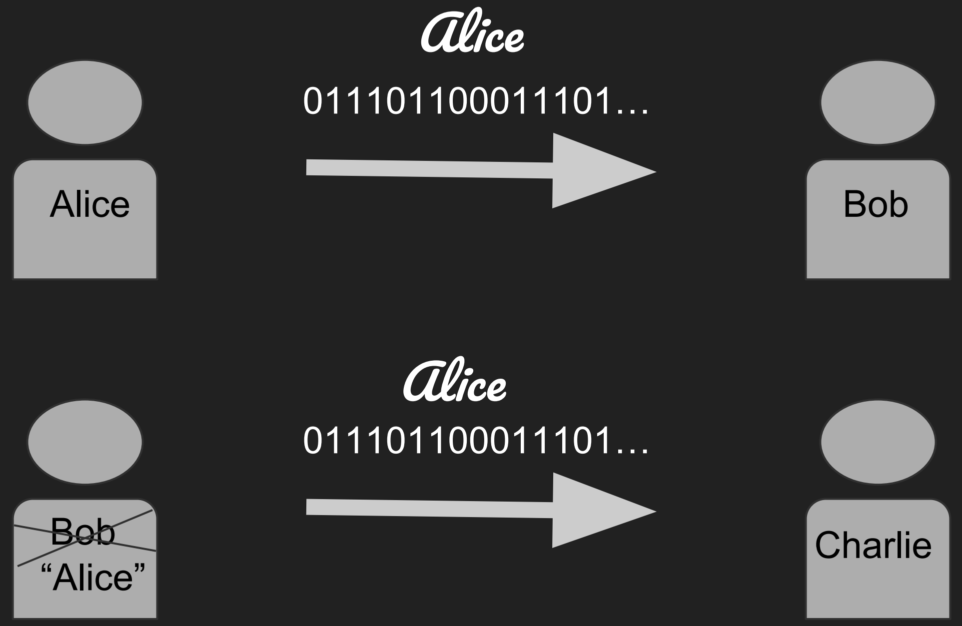
Digital Signatures

Goal:

- Identify a particular person digitally
- Verifiable by a third party
- Not forgeable

Digital Signatures

Why is this hard?



Digital Signatures

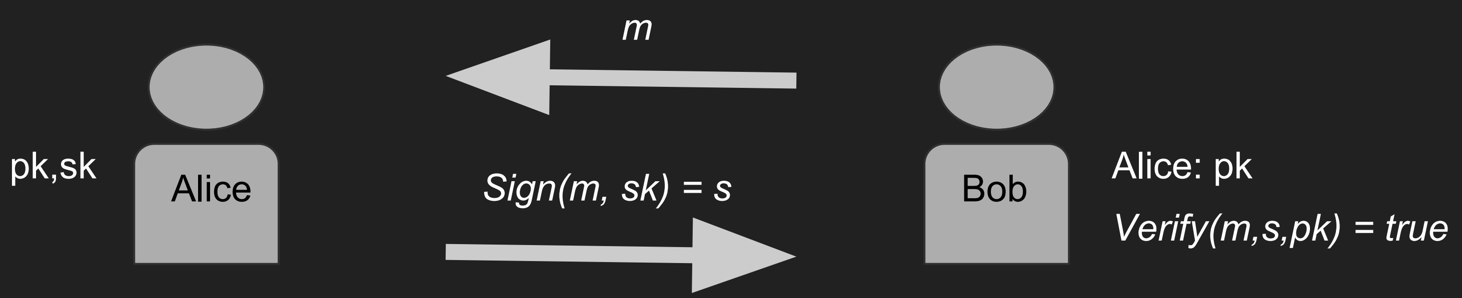
1. Key Generator
    a. Produces a Public Key and Private/Secret Key
    b. Example: RSA

2. Sign(message, private_key) -> signature
    a. Not reversible (without private_key)
    b. Output should appear uncorrelated with input

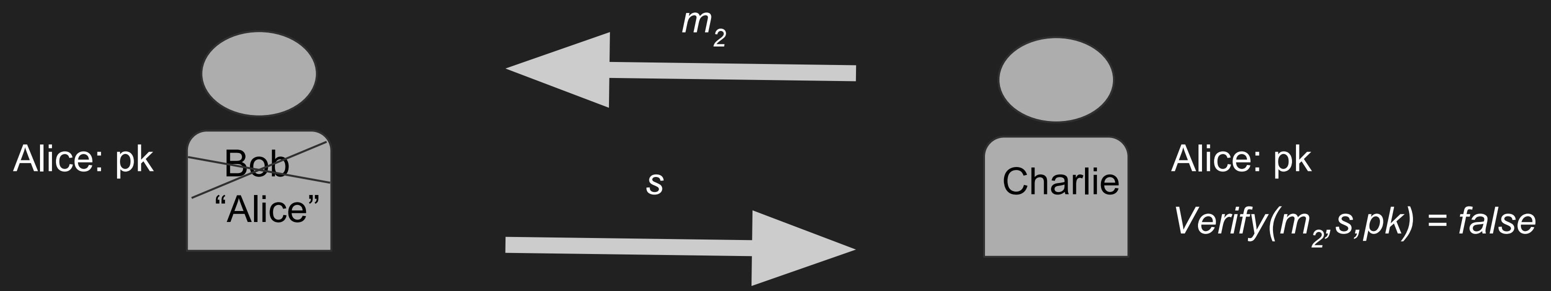
3. Verify(message, signature, public_key) -> boolean
    a. Note: doesn’t involve the private key!

Digital Signatures

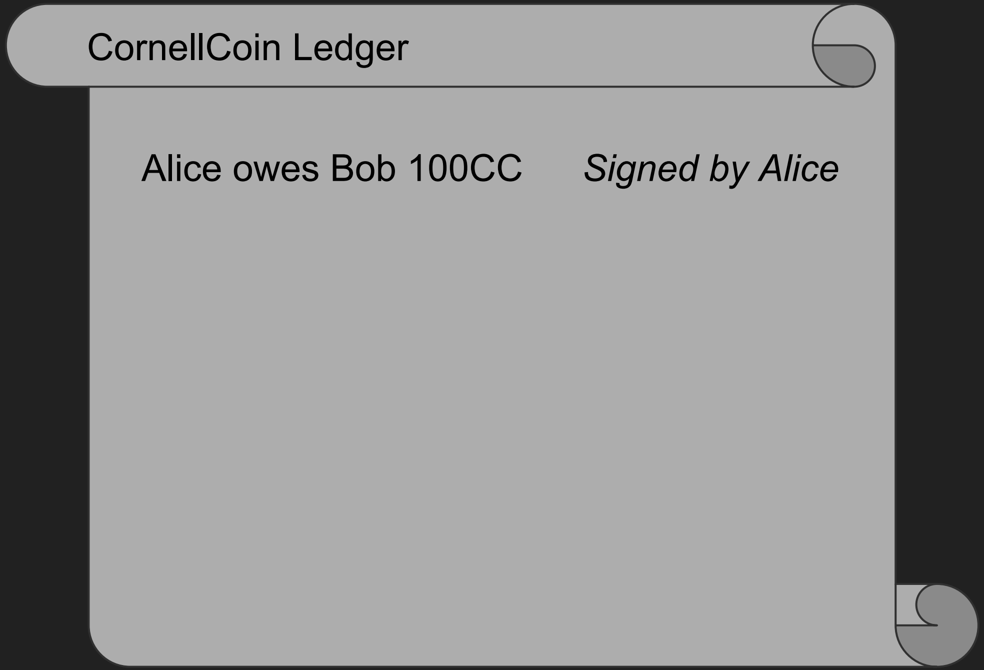
Digital Signatures



Digital Signatures



Digital Signatures



Digital Signatures

CornellCoin Ledger

Alice owes Bob 100CC	*Signed by Alice*
Alice owes Bob 100CC	*Signed by Alice*
Alice owes Bob 100CC	*Signed by Alice*
Alice owes Bob 100CC	*Signed by Alice*
Alice owes Bob 100CC	*Signed by Alice*

Digital Signatures

Digital Signatures: Implementation

- Implementation is for fun!
  - Presenting a (simplified) version of the RSA implementation
- Prime numbers
- Modulo arithmetic
  - Reminder: this is getting the remainder after division
  - 10 mod 4 = 2
  - 15 mod 5 = 0
- Goal: Find *e, d, n,* such that $(m^e)^d \equiv m \pmod{n}$
  - Should also be tricky to determine *d* given *e, n, m*
- Examples taken from Wikipedia

OPTIONAL MATERIAL

Digital Signatures: KeyGen Implementation

- Choose two prime numbers: *p, q*
    - Should be chosen randomly
- Compute $n = pq$
- Compute $\lambda(n) = LCM((p-1, q-1))$
- Choose $1 < e < \lambda(n)$, such that e and $\lambda(n)$ are coprime
- Solve for d: $d*e = 1 \bmod \lambda(n)$

- Public key: (n, e)
- Private key: (n, d)

OPTIONAL MATERIAL

Digital Signatures: Implementation

- Reminder: Find *e, d, n,* such that $(m^e)^d \equiv m \pmod{n}$
- Public key: (n, e)   Private key: (n, d)

- *Sign(m, sk):* $m^d \bmod n$
- *Verify(m, s, pk):* $s^e \bmod n == m$

OPTIONAL MATERIAL

Digital Signatures: KeyGen Implementation Example

- $p = 61, q = 53$
- $n = pq = 61*53 = 3233$
- $\lambda(3233) = LCM(60, 52) = 780$
- Choose $e = 17$
- Solve for d: $d*17 = 1 \text{ mod } 3233 \rightarrow d = 413$

- Public key: (3233, 17)
- Private key: (3233, 413)

OPTIONAL MATERIAL

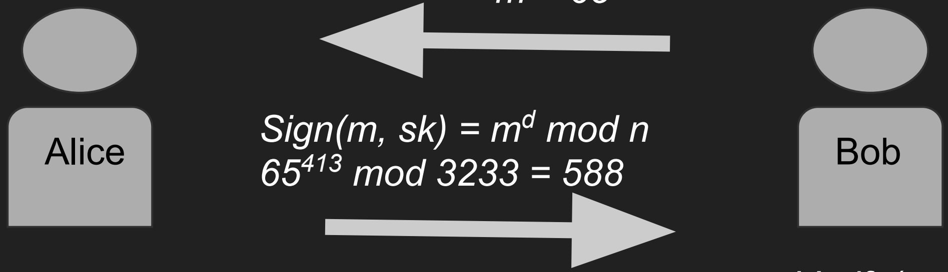
Digital Signatures: Example

OPTIONAL MATERIAL

Digital Signatures: Example

m = 65

pk = (3233, 17)
sk = (3233, 413)



Sign(m, sk) = m^d *mod n*
65^{413} *mod 3233 = 588*

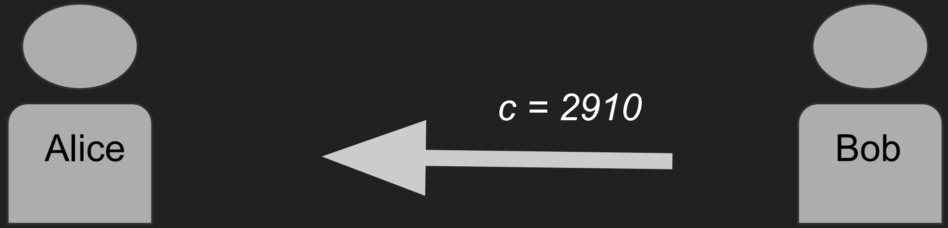
Alice: (3233, 17)

Verify(m,s,pk) = (s^e *mod n == m)*
588^{17} *mod 3233 == 65*

OPTIONAL MATERIAL

Public/Private Key Encryption

pk = (3233, 17)
sk = (3233, 413)



c = 2910

Alice: (3233, 17)

m = 92

$Decrypt(c,sk) = c^d \bmod n$
$2910^{413} \bmod 3233 = 92$

$Encrypt(m,pk) = m^e \bmod n$
$92^{17} \bmod 3233 = 2910$

Digital Signatures

Unanswered questions:

- Why on earth does this work?
  -
- How do Bob and Charlie know that the public key they received is actually from Alice?
  - Practically… not a technical solution. Certificate Authorities do the job.
  - Can the blockchain be used for this?