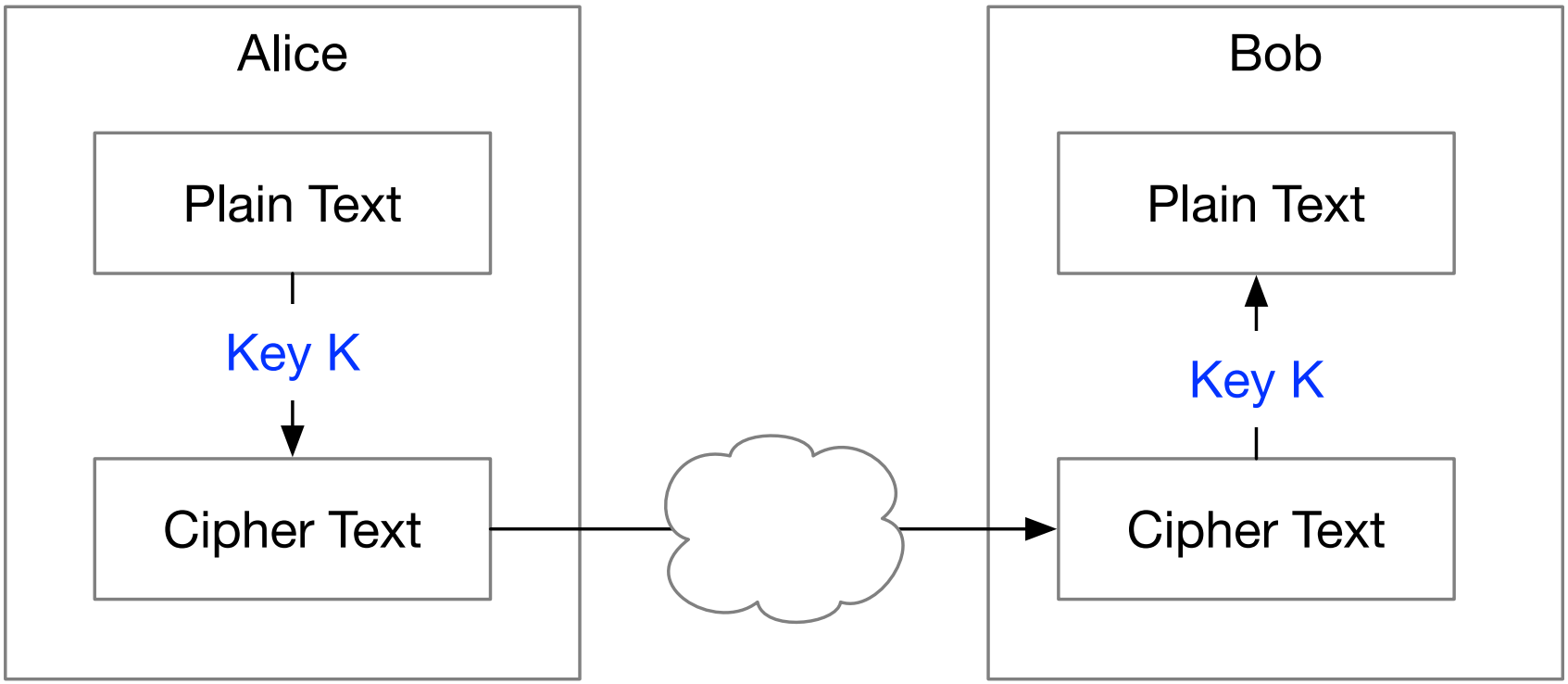
Intro to blockchain

Zhehao Wang

Oct 2018

Symmetric cryptography

Consider symmetric cryptoraphy, where



The problem: how do I securely transfer *key K* over the network?

Asymmetric cryptography, RSA algorithm

Find 3 very large positive integers e, d, n s.t.

$$(m^e)^d \equiv m \pmod{n}, \quad \forall m,\ 0 \leq m \leq n$$

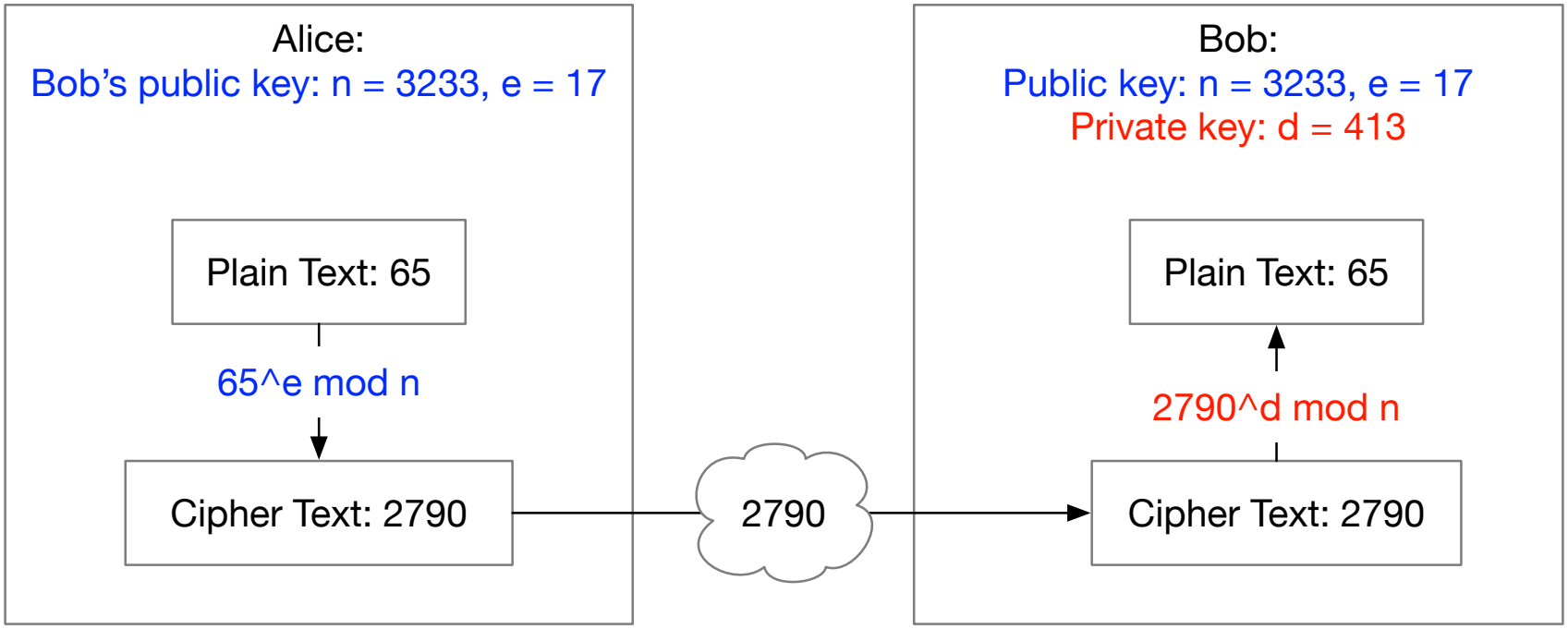
Knowing e, n or even m it's extremely hard to find d.

- Public key: n, e
- Private key: n, d
- Message: m

Operations:

- Encryption(m): $c = m^e \pmod{n}$
- Decryption(c): $m' = c^d \pmod{n} = (m^e)^d \bmod n$, $m' = m \pmod{n}$

RSA algorithm, example



Anyone who wants to talk to Bob can retrieve Bob's public key, use it to encrypt the message, and know that only holder of the corresponding private key can decrypt.

RSA algorithm in practice

In practice, key pairs are much longer.

```
$ ssh-keygen -t rsa -b 4096
```

Each key pair corresponds with an **identity**.

The two operations:

- Encryption/Decryption: Alice uses Bob's public key to encrypt, Bob uses his private key to decrypt
- Signing/Validation: Alice uses her own private key to sign, others use Alice's public key to verify

Vs symmetric encryption: more computation, but solves the problem of key transfer (and many others)

Related concepts: digital signature, certificate, public key infrastructure

To design cryptocurrency

Imagine designing a cryptocurrency

- An account is a public/private key pair!
- If I pay someone (a public key identity), I sign with my private key: others can verify *I* made the payment, and I cannot refute later on

But how does anyone know I have enough money left to make the payment? *Double spending*

- The **centralized** way: we all agree on (and preinstall) having one party to trust, who keeps track of everyone's account balances
- The **decentralized** way: is it possible to have *all* of us keep track of account balances together?

To design cryptocurrency - cont

Design considerations:

- Distributed
- Trustless, no a-prori trust relationship established
- Consensus, everyone agrees on account balances

Why is this problem unique:

- I don't trust anyone I talk to! (think Raft, dynamo, etc) But we can still agree on something.

Distributed trustless consensus - intuition

Distributed consensus: leadership election, and **state-machine replication** (think raft)

If we can agree on the entire history of transactions, we would know if someone is trying to double spend. (think log structured merge tree (e.g. Cassandra) / journaling file system / git)

To agree on the history,

- what if we assume: there are more peers in the network who are honest, than those who are not
- we can have everyone tell everyone else their view of the entire history, and hope they converge...

Blockchain in bitcoin whitepaper

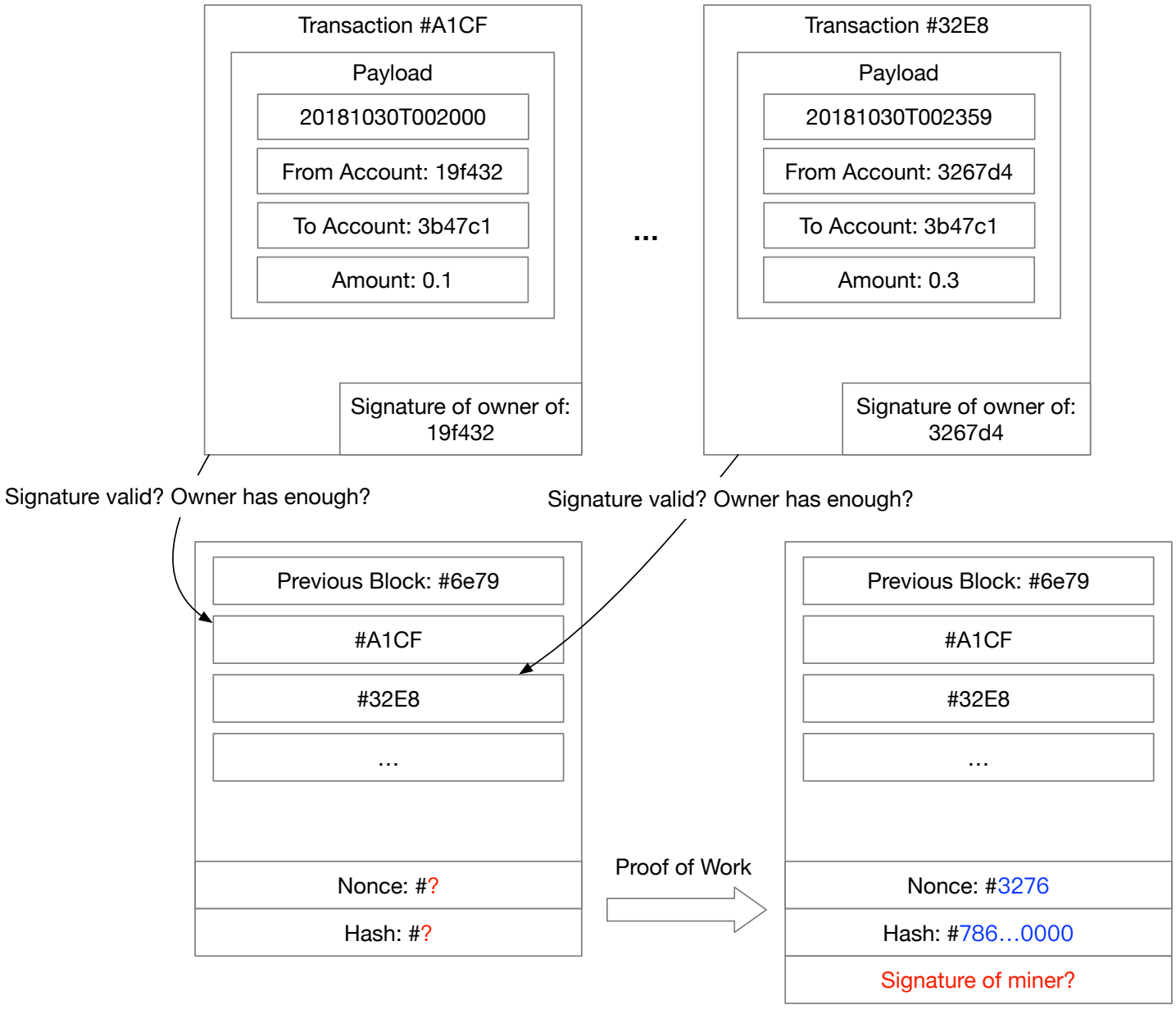
Blockchain makes a different assumption

- **honest peers own more computation power than those who are dishonest**

Consequently, if it takes computation power to grow the history, then honest peers can grow the history faster than dishonest ones

- Peers agree on **trusting the longest chain**, and grow based on that
- In order to grow the history, peers perform a **proof of work**, which is computationally non-trivial

Building a block



Growing a chain

Proof of work

Mining and incentive