How do cryptocurrencies relate to web security?

1. JavaScript Bitcoin clients exist; you can use site visitors to make money for you
2. Malware exists to steal computing resources expressly for mining bitcoins

What is currency?

- A means of exchange, so that we don't have to carry around goats.

- Something we all agree is valuable…
- …maybe because people with guns tell us it is valuable?

History of currency

- 2000 BC – Receipts represented grain stored in Sumerian temple granaries *(representative money)*
- 600-700 BC – Coins developed in Anatolia, Greece, India, and China *(commodity money)*
  - Value of these coins tied to metal content
- 900 AD – Jiaozi banknote developed in China
  - A *fiat* money– valuable because government says so
- 1971 – U.S. breaks away from the gold standard

So what is a *digital* currency?

Properties of an ideal digital currency

(Okamoto and Ohta)

1. **Independence**
   - not dependent on any physical location
   - can be transferred through computer networks
2. **Security**
   - cannot be copied and reused
3. **Privacy**
4. **Off-line Payment**
   - No need to be linked to a host to process payment
5. **Transferability**
6. **Divisibility**

DigiCash

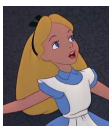
- 1983 – David Chaum invents *blinding formula* allowing anonymous, verifiable transactions
- Late 1980s – Chaum starts DigiCash
- 1998 – DigiCash goes bankrupt, in part because of difficulty working with banks
- Today, opencoin builds on some of Chaum's ideas. http://opencoin.com/

Bitcoin

- Protocol designed by Satoshi Nakamoto in 2008 https://bitcoin.org/bitcoin.pdf
- First Bitcoin client launched in 2009
- Peer-to-peer– no centralized control
  - Every client keeps track of the history of all bitcoins

Building a Cryptocurrency

Digital Currency – Take 1



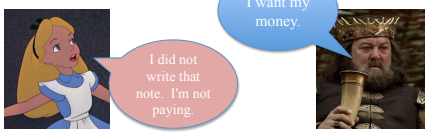
"I Alice agree to pay the bearer of this note $100"

Alice

Bob

Any issues with this approach?

Digital Currency 1 – Repudiation



Alice

Bob

"I Alice agree to pay the bearer of this note $100"

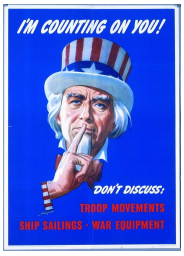
Public Key Cryptography &

Digital Signatures

Symmetric Key Cryptography

- One key, shared between sender and receiver
  - `E(P,K) = C` Encrypt plaintext P with key K
  - `D(C,K) = P` Decrypt ciphertext C with key K
- In existence for millennia
- Fast and efficient
- Useless for digital signatures
  - Everyone with the key could have encrypted the message

Until about the 1960s, information security was largely the domain of the government and the military.



And then…

Private companies and individuals began to need information security.

Public Key Encryption

- Relies on 'trap-door' functions
- Uses two separate keys
  - Public key is known by everyone
  - Private key known only to the owner
- Analogy: locked mailbox
  - Anyone can put a letter in the mailbox
  - Only the mail carrier can get them

Digital Signatures

- Reverse process is used for digital signatures:
  - Private key can encrypt a message
  - Public key can decrypt the message
- Analogy: Enclosed bulletin board
  - Anyone can read the messages
  - Only the owner could have put the messages there

Public Key Notation

- **Sign** message M with Alice's **private key:** $[M]_{Alice}$
- **Encrypt** message M with Alice's **public key:** $\{M\}_{Alice}$
- Then

  $\{[M]_{Alice}\}_{Alice} = M$

  $[\{M\}_{Alice}]_{Alice} = M$

RSA



RSA

- By Clifford Cocks (GCHQ), independently, **R**ivest, **S**hamir, and **A**dleman (MIT)
  - RSA is the ***gold standard*** in public key crypto
- Let p and q be two large prime numbers
- Let N = pq be the **modulus**
- Choose e relatively prime to (p-1)(q-1)
- Find d such that
  ed mod (p-1)(q-1) = 1
- **Public key** is (N,e)
- **Private key** is d

RSA

- Message M is treated as a number
- To encrypt M we compute
  $C = M^e \bmod N$
- To decrypt ciphertext C compute
  $M = C^d \bmod N$
- Recall that e and N are public
- If Trudy can factor N=pq, she can use e to find d since ed mod (p-1)(q-1) = 1
- **Factoring the modulus breaks RSA**

Simple RSA Example

- Example of RSA
  - Select "large" primes p = 11, q = 3
  - Then N = pq = 33 and (p – 1)(q – 1) = 20
  - Choose e = 3 (relatively prime to 20)
  - Find d such that ed mod 20 = 1
    - We find that d = 7 works
- **Public key:** (N, e) = (33, 3)
- **Private key:** d = 7

Simple RSA Example

- **Public key:** (N, e) = (33, 3)
- **Private key:** d = 7
- Suppose message M = 8
- Ciphertext C is computed as
  $C = M^e \bmod N = 8^3 = 512 = 17 \bmod 33$
- Decrypt C to recover the message M by
  $M = C^d \bmod N = 17^7 = 410{,}338{,}673$
  $= 12{,}434{,}505 * 33 + 8 = 8 \bmod 33$

Problem 3.2 – RSA

See details on
http://cs31.cs.sjsu.edu/labs/exercise3.pdf

Node.js review

- A JavaScript runtime and library designed for use outside the browser, based off of Google's V8 engine
- npm: package manager for Node.js
- http://nodejs.org/

myFile.txt

```
This is my file.
There are many like it,
but this one is mine.
```

File I/O in Node.js

```
var fs = require('fs');
fs.readFile('myFile.txt',
  function(err,data) {
    if (err) throw err;
    console.log(""+data);
  });
console.log('all done');
```

Callback function

Resulting Output

```
all done
This is my file.
There are many like it,
but this one is mine.
```

Synchronous File IO in Node

```
var data = fs.readFileSync(
      './myFile.txt');
console.log(data.toString());
console.log('all done');
```

Problem 3.3 – Using RSA with Node.js

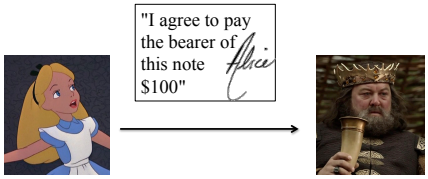
Install the `keypair` module for node:

```
npm install keypair
```

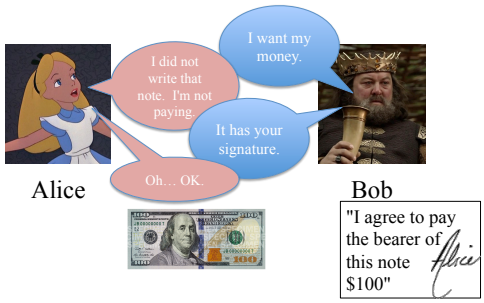
a) Download genKeys.js and testSig.js from the course website. Generate a public and private key. Sign a message and verify that the signature is valid. Turn in your modified testSig.js code.

b) Download message.txt, sig.txt, alicePub.txt, bobPub.txt, and charliePub.txt.
Who signed the message?

Digital Currency – With Signatures



Digital Currency with Signatures Nonrepudiation



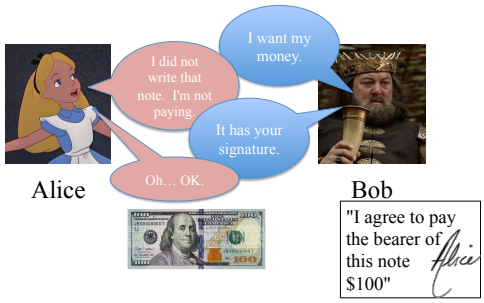
Problem 3.4 – Create a cryptocurrency

Create a first version of a cryptocurrency. Download cryptoCurrSimple.js, alicePriv.txt, alicePub.txt, bobPriv.txt, bobPub.txt.

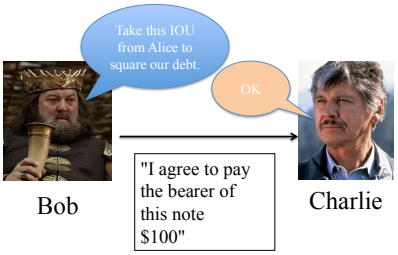
Update the `sendIOU` and `receiveIO` methods so that:

- The sender signs the message and includes the signature in the `transaction` object (in a field named `sig`).
- The receiver verifies the signature

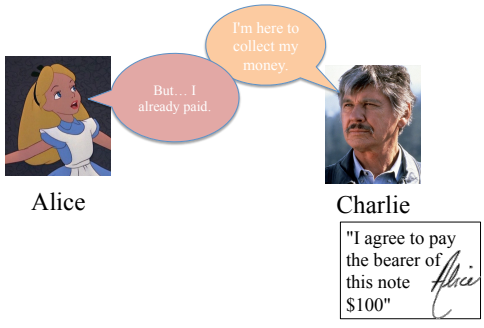
Digital Currency with Signatures



Double Spending



Double Spending



A centralized authority could monitor all transactions…

Because everyone trusts banks

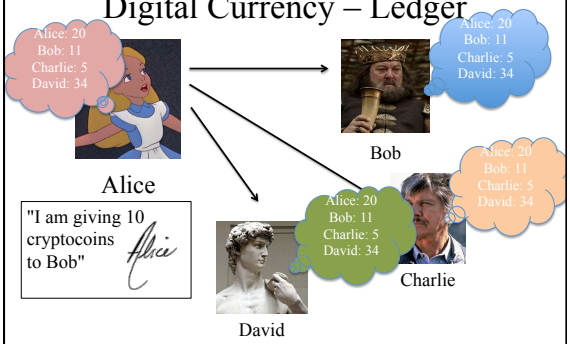
Or everyone could keep track of the transactions, and vote whenever a discrepancy arises.

What is a cryptocoin worth?

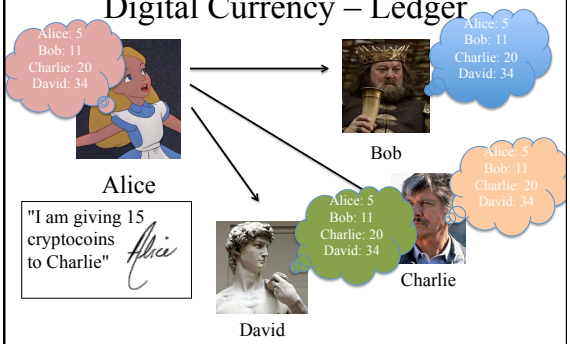
In our earlier examples, the value of our IOUs is tied to US dollars.

Other cryptocurrencies (such as bitcoins) are not tied to any other currency. We'll follow this model from now on.

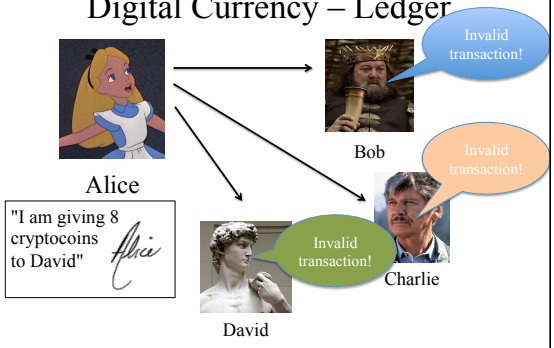
Digital Currency – Ledger



Digital Currency – Ledger



Digital Currency – Ledger



Problem 3.5
Cryptocurrency with Ledger

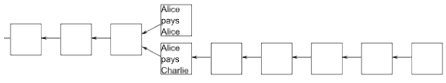
Download cryptoCurrLedger.js.

Update the `validateTransfer` method. Verify that the user has enough coins, and that the signatures are valid. Update the local copy of the ledger if everything seems valid.

Of course, people will always be tempted to cheat. How can we catch them in a peer-to-peer system?

Bitcoin Proof of Work

- The Bitcoin protocol uses proof of work to verify *block chains*, which determine transaction history.
- This strategy prevents *double spending*, where Alice tries to spend the same coin with both Bob and Charlie.



Review: Cryptographic hash functions

- Hash functions compress input to a small, fixed-length output.
  - hash("Some long input") = 87d9417d8cd12635
- A good hash function should be:
  - **one-way**: given a value y it is infeasible to find an x such that h(x) = y
  - **collision-resistant**: infeasible to find ***any*** x and y, with x ≠ y such that h(x) = h(y)
  - **efficient**
  - **compression**

Mining

- Miners hash transaction details plus a "proof" of spending computational resources
  - Reward: some bitcoins are generated plus there can be transaction fees
- Cost to discover proof – 2^N hashes
- Cost to verify proof – One hash
- The Bitcoin protocol is designed to make mining more profitable than cheating
  - https://bitcoin.org/bitcoin.pdf

Mining Pools

- Making money with bitcoins is like buying a bunch of lottery tickets – you might win sometimes, but it is very irregular
- Mining pools: collaborations of bitcoin miners
  - More regular income for the miners
  - But a large mining pool could seize control of the money supply: **"Majority is not Enough: Bitcoin Mining is Vulnerable", Eyal & Sirer 2013**

Hashcash – Spam Reduction

- Let M = email message
  R = value to be determined
  T = current time
- Sender must find **R** so that
  h(M,**R**,T) = (00…0,X), where
  N initial bits of hash value are **all zero**
- Sender then sends (M,**R**,T)
- Recipient accepts email, provided that…
  h(M,**R**,T) begins with N zeros

Spam Reduction

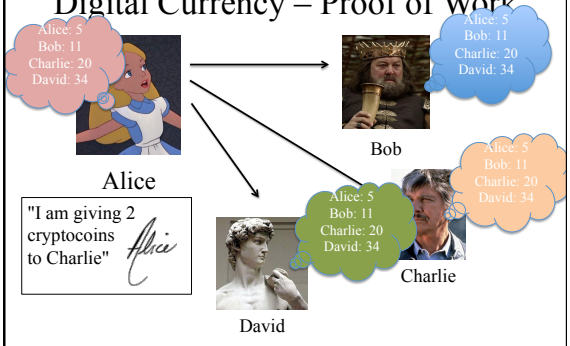
- Sender: h(M,R,T) begins with N zeros
- Recipient: verify that h(M,R,T) begins with N zeros
- **Work for sender:** about **2^N hashes**
- **Work for recipient:** always **1 hash**
- Sender's work increases exponentially in N
- Small work for recipient regardless of N
- Choose N so that…
  - Work acceptable for normal email users
  - Work is too high for spammers

Problem 3.6
Implement Hashcash

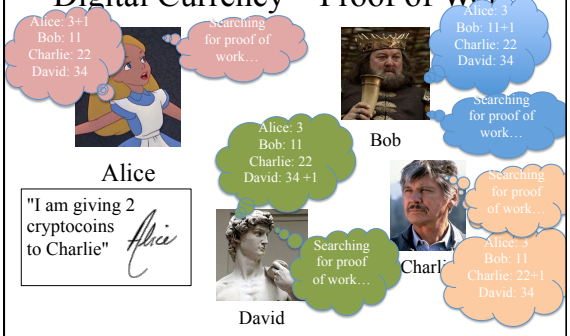
Download proofOfWork.js from course website.
Note that `hash` uses SHA256 to return the hash of a String as a binary String.

a) Implement the `findProof` function. This function should return a value proof such that `hash(value + proof)` has the specified number of leading zeroes.
b) How long does it take to find a proof for 2 leading zeroes? How about 10? 20?
c) Implement a `verifyProof` function.

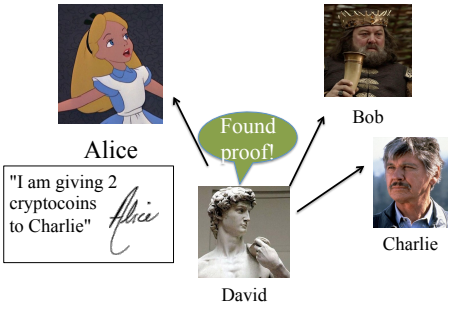
Digital Currency – Proof of Work



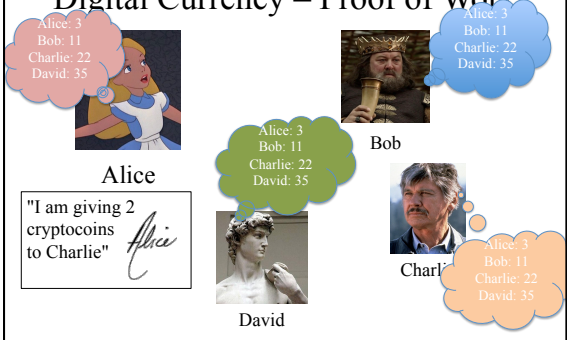
Digital Currency – Proof of Work



Digital Currency – Proof of Work



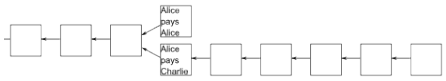
Digital Currency – Proof of Work



Handling Discrepancies

Discrepancies between ledgers are resolved in favor of the chain with the greatest amount of work used in its creation.

Generally, this means that we take the longest blockchain. (In our code, the blockchain is the series of `ledger.prev`).



Problem 3.7
Cryptocurrency Proof of Work

Combine the ledger-based cryptocurrency with your proof of work code.

See details on
http://cs31.cs.sjsu.edu/labs/exercise3.pdf

Next Session: Advanced Programming Language Defenses

Install Ruby before the next class