Cybersecurity for Future Presidents

Lecture 12:
Applications of Cryptography and Trust Management:
Anonymity and Digital Currency

Any Questions?

My office hours:
Wed. afternoon, 12-3pm, 442 RH. Signup sheet circulating

- About previous lecture?
- About homework? (misc. exercises on storage sizes, scams, computational hardness, secure hash)
- About reading? (D is for Digital - Algorithms)

Reading for next week: For bitcoin debate (both available on Canvas):

1. Bitcoin: Under the Hood. Communications of the ACM, Sept. 2015.
2. How A Credit Card is Processed. CreditCards.com.

Watch: Khan Academy overview of Bitcoin (11 minutes)

- https://www.khanacademy.org/economics-finance-domain/core-finance/money-and-banking/bitcoin/v/bitcoin-overview
- This is the first of a series of tutorial videos; watch others (e.g., secure hashing) if you wish

Exercises: Questions for debates

Cybersecurity events from the past week of interest to future (or current) Presidents:

- E-mail:
  - Microsoft sues Justice Dept over secrecy orders to read emails on Fourth Amendment grounds
  - House judiciary committee votes unanimously for legislation to update ECPA to require warrants for access to email older than 180 days
- More appeals courts side with government on warrentless cellphone tracking
- Congressman Lieu complains that old SS7 vulnerability allowing eavesdropping of phone conversations remains unfixed
- Administration appoints members to commission on enhancing national cybersecurity
- US meets with Russia to prevent accidental cyberwar

Coming up: … ?

Anonymity on the Internet (1993)

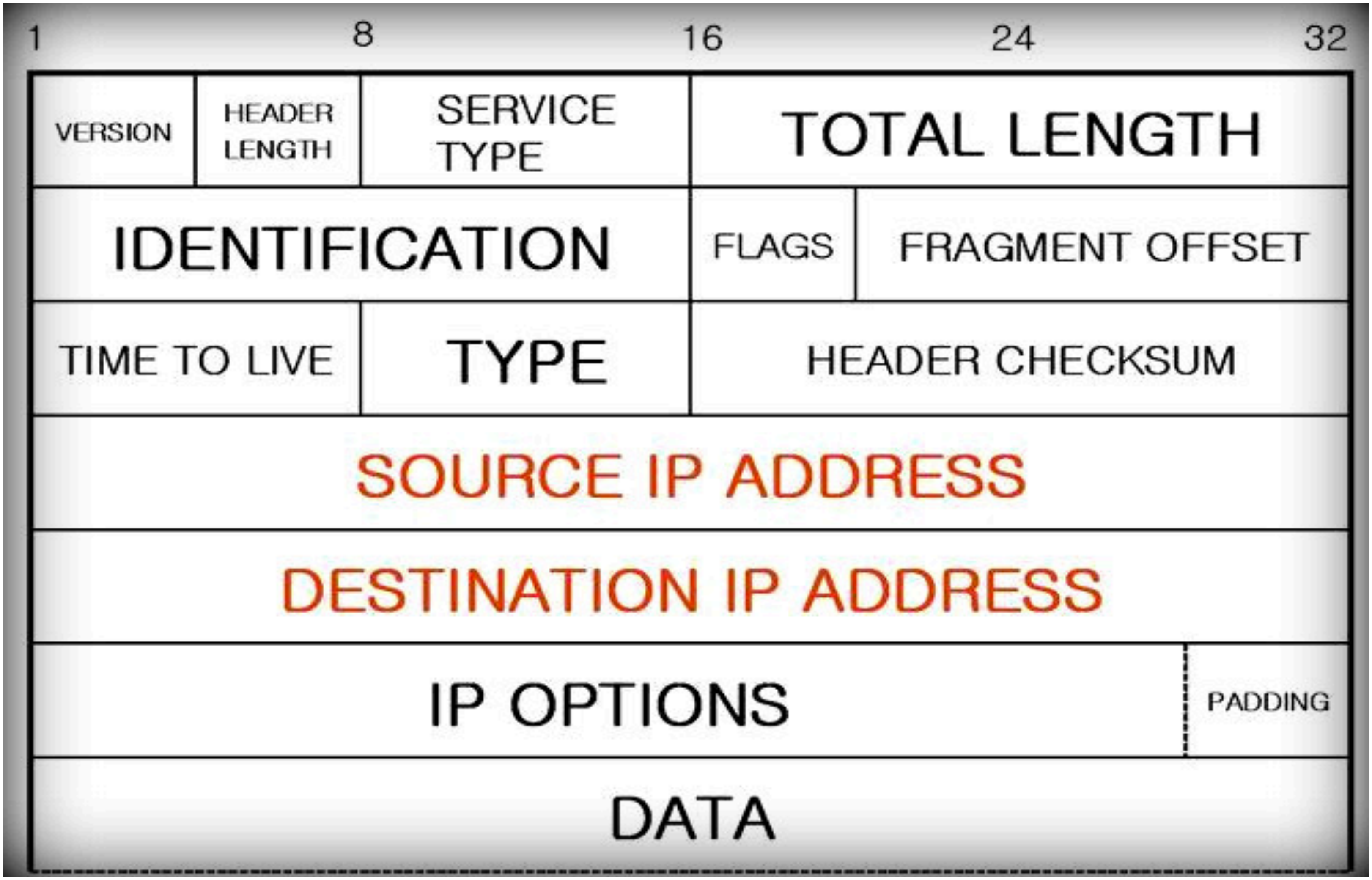
"On the Internet, nobody knows you're a dog."

"On the Internet, nobody knows you're a dog"

Peter Steiner (7/5/1993)

But how anonymous are you, really?

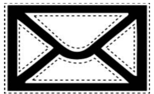
- Internet packet header format:



In a <u>packet-switched</u> network, the routers must be able to read the packet headers

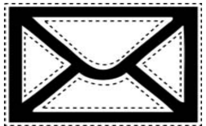
- So the source and destination addresses (and other packet meta data) are normally in the clear
- Even if the data in the packet is encrypted, much intelligence can be gained from the meta-data, for example:
  - Who is talking to whom?
  - How often?
  - What lengths of messages, in which direction?
- So, the routers may not know you are a dog, but they can know the other IP addresses you are communicating with, how often, etc.

What is super-encryption?



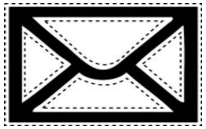
It's just encrypting a message that's already encrypted:

- If M = message, $E_{k1}[M]$ is the message encrypted under Key k1
- $E_{k2}[E_{k1}[M]\ ]$ is M encrypted first under key k1 and then re-encrypted under key k2 (doubly encrypted)



To decrypt $E_{k2}[E_{k1}[M]\ ]$, first remove the outer layer of encryption:

- $D_{k2}[E_{k2}[E_{k1}[M]\ ] \rightarrow E_{k1}[M]$



Then remove inner layer of encryption:

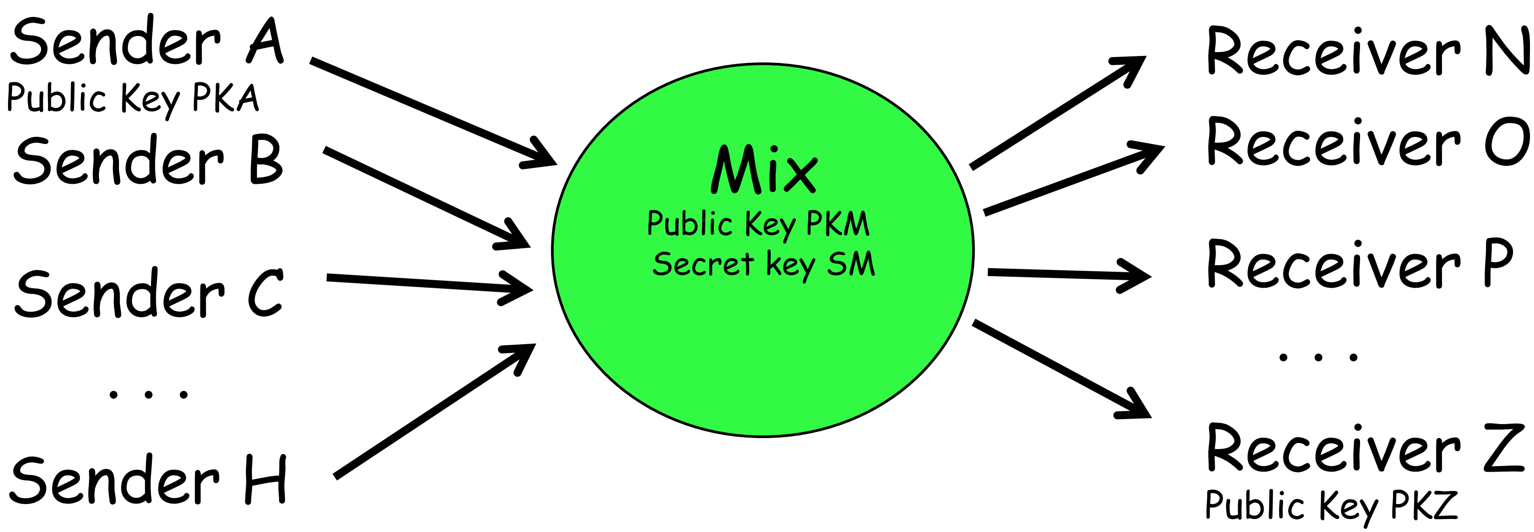
- $D_{k1}[E_{k1}[M]] \rightarrow M$

Think of this as simply a set of nested envelopes.

- Each time you add a layer of encryption, you've created a new (outer) envelope.

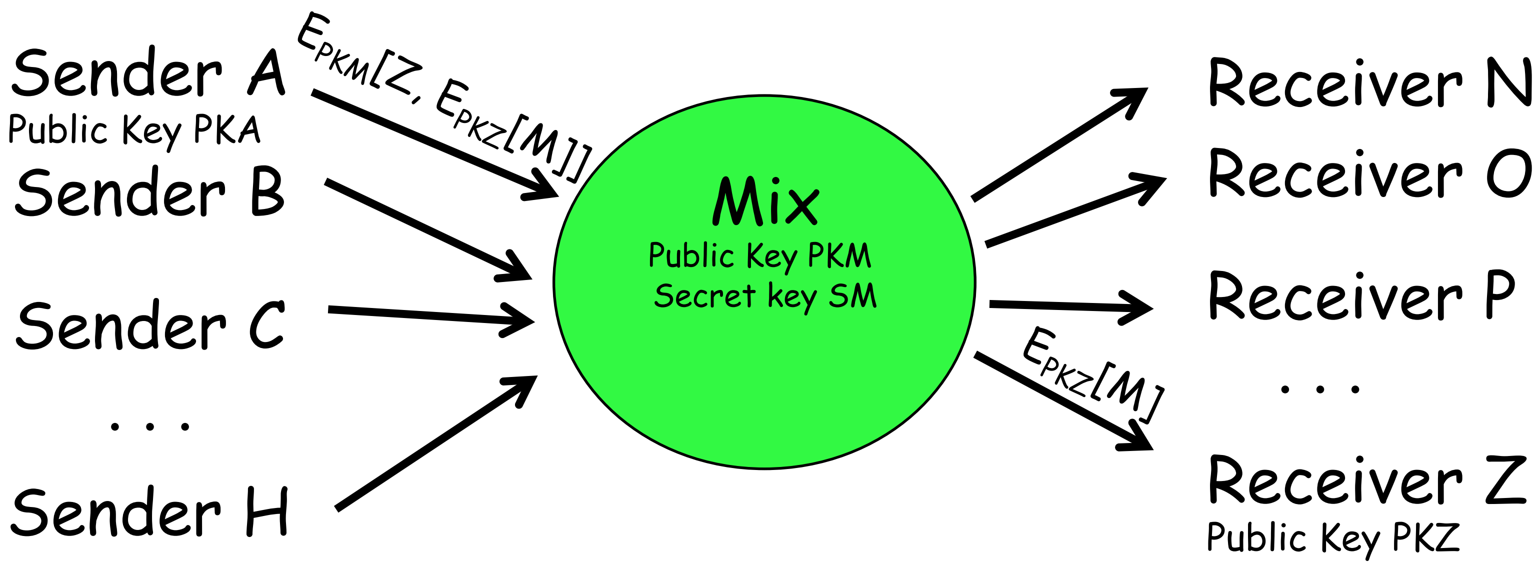
What is a (Chaum) Mix?
(Or, how to use superencryption for anonymity

- In 1981 David Chaum published a scheme that would support anonymous email over the Internet, using what he called a Mix
- Think of the Mix as a single node that has a public key PKM and secret key S_M
- Other nodes also have public/secret key pairs. The setup:



How to send a message anonymously from A to Z

- A encrypts the message under Z's public key, making $E_{PKZ}[M]$
- A adds Z's address and encrypts the whole thing under the Mix public key: $E_{PKM}[Z, E_{PKZ}[M]]$ and sends to the Mix
- The Mix decrypts with its secret key, extracts the destination address and the (still encrypted) message for Z and forwards to Z:
  - $D_{SM}[E_{PKM}[Z, E_{PKZ}[M]] = Z, E_{PKZ}[M]$
- Z receives and uses its secret key to decrypt:
  - $D_{SZ}[E_{PKZ}[M]]$
- Z gets the message but only sees IP address of the Mix

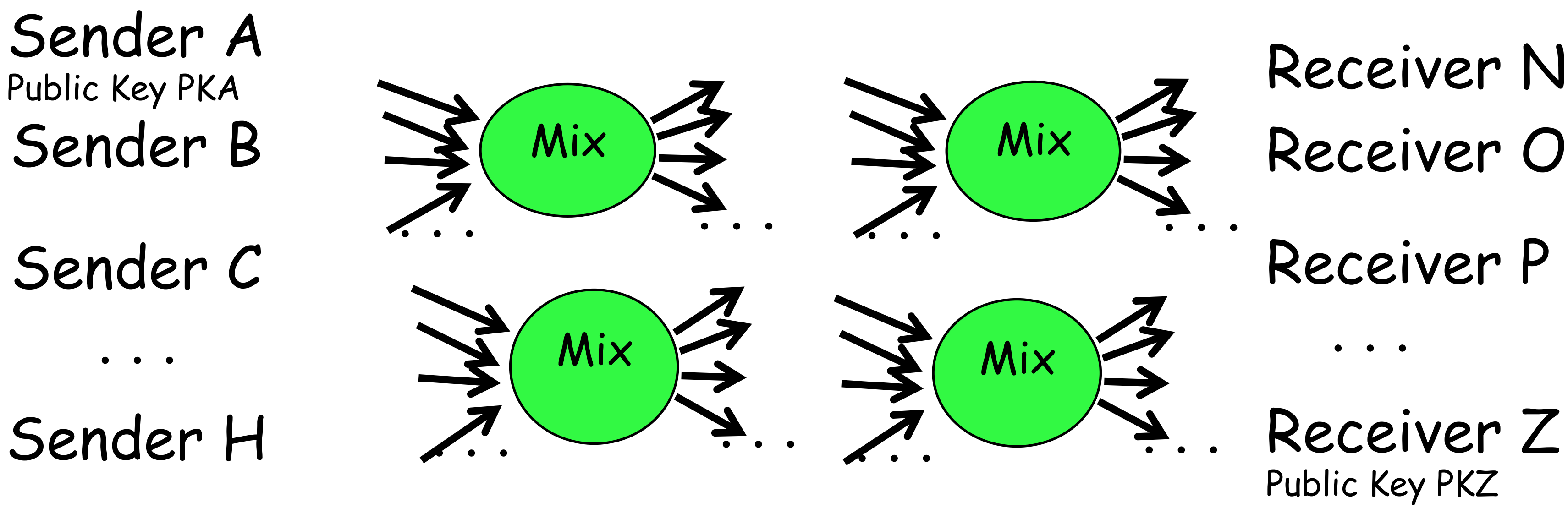


Some issues with this scheme

- Timing: Observer might be able to watch traffic enter and exit the Mix node and figure out from the timing and message lengths who is communicating with whom
  - Solution:
    - batch the traffic: Mix waits for sufficient number of messages to accumulate and then sends them all back out in a burst.
    - Also, chop/pad traffic so it's all fixed length blocks
- Guessing messages: Observer could perhaps encrypt messages likely to be sent under the public keys of possible recipients and might recognize the traffic
  - Solution: add "salt" (a random number) to the message before encrypting it; recipient removes the salt after decryption
- Single Mix node is a central point of failure
  - Make it a MixNet (see next slide) – this also makes it harder to execute timing attacks

A Mix-net

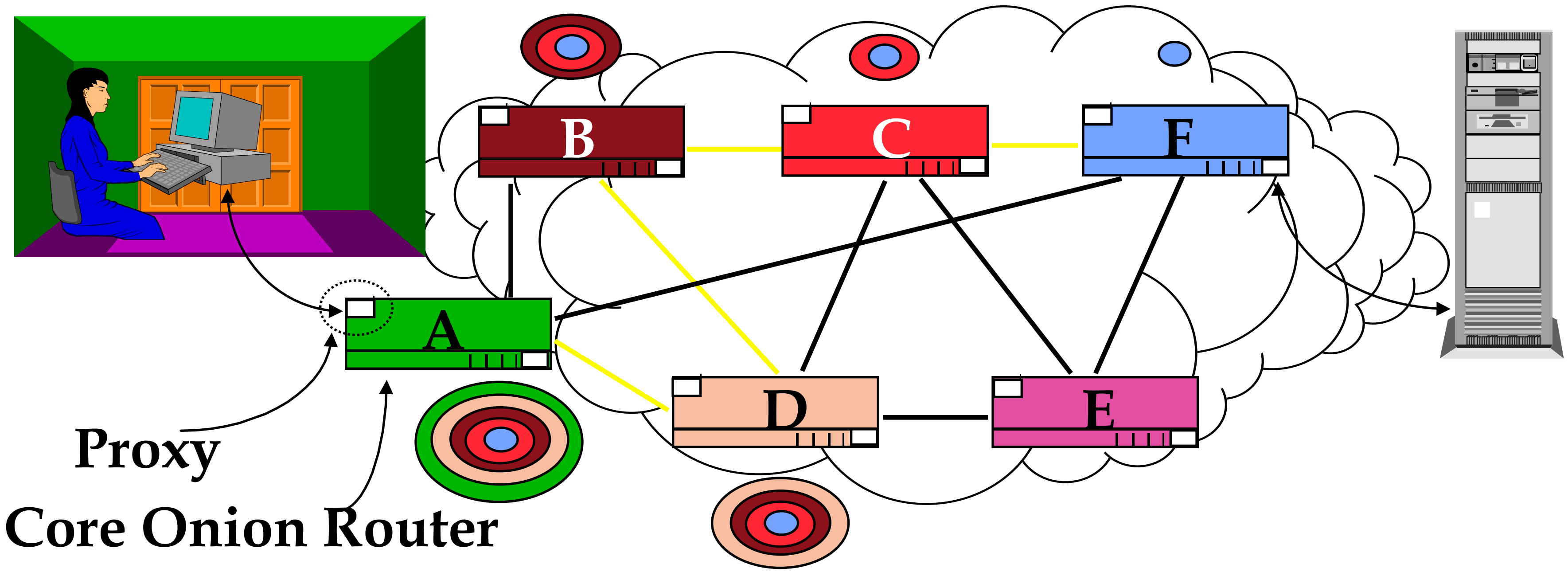
Idea: add more layers of encryption and let traffic bounce among Mix nodes before being delivered



How does Onion Routing (Tor) work?

- Original goal for Onion Routing was to enable traffic flow security for military communications, including web-browsing over the Internet – not for anonymity between end points
- Tor is essentially a large mix-net, but with some developments
  - "Onions" are constructed by sender using public keys to initiate a connection within the Tor network that uses symmetric-key crypto (faster) for the subsequent traffic
  - Mixes were designed to work with email (non real-time)
  - Tor wants to support web-browsing, so can't delay traffic too much
    - Consequently may be possible to detect correlations in traffic flow. The hope is that if there is enough traffic, it will be hard for opponents.

Connection Setup



◆The initial proxy knows the Onion Routing network topology, selects a route, and generates the onion

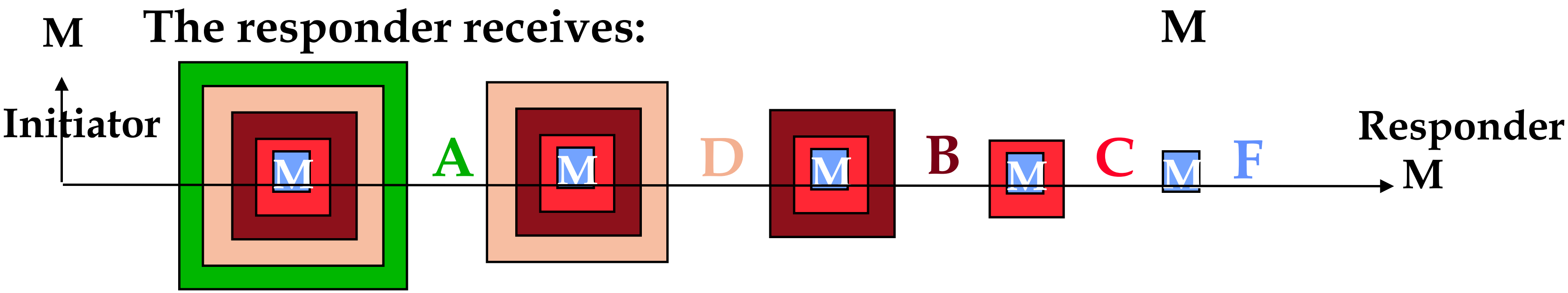
◆Each layer of the onion identifies the next hop in the route and contains the cryptographic keys to be used at that node.

Data Movement

As data moves through the anonymous connection, it looks different at each onion router.

A message M sent from an initiator to a responder via a 5-hop onion route will change as follows:

The initiator pre-crypts M giving:	E(E(E(E(E(M)))))
Entering A, the message will look like:	E(E(E(E(E(M)))))
Entering D, the message will look like:	E(E(E(E(M))))
Entering B, the message will look like:	E(E(E(M)))
Entering C, the message will look like:	E(E(M))
Entering F, the message will look like:	E(M)
The responder receives:	M



What happened

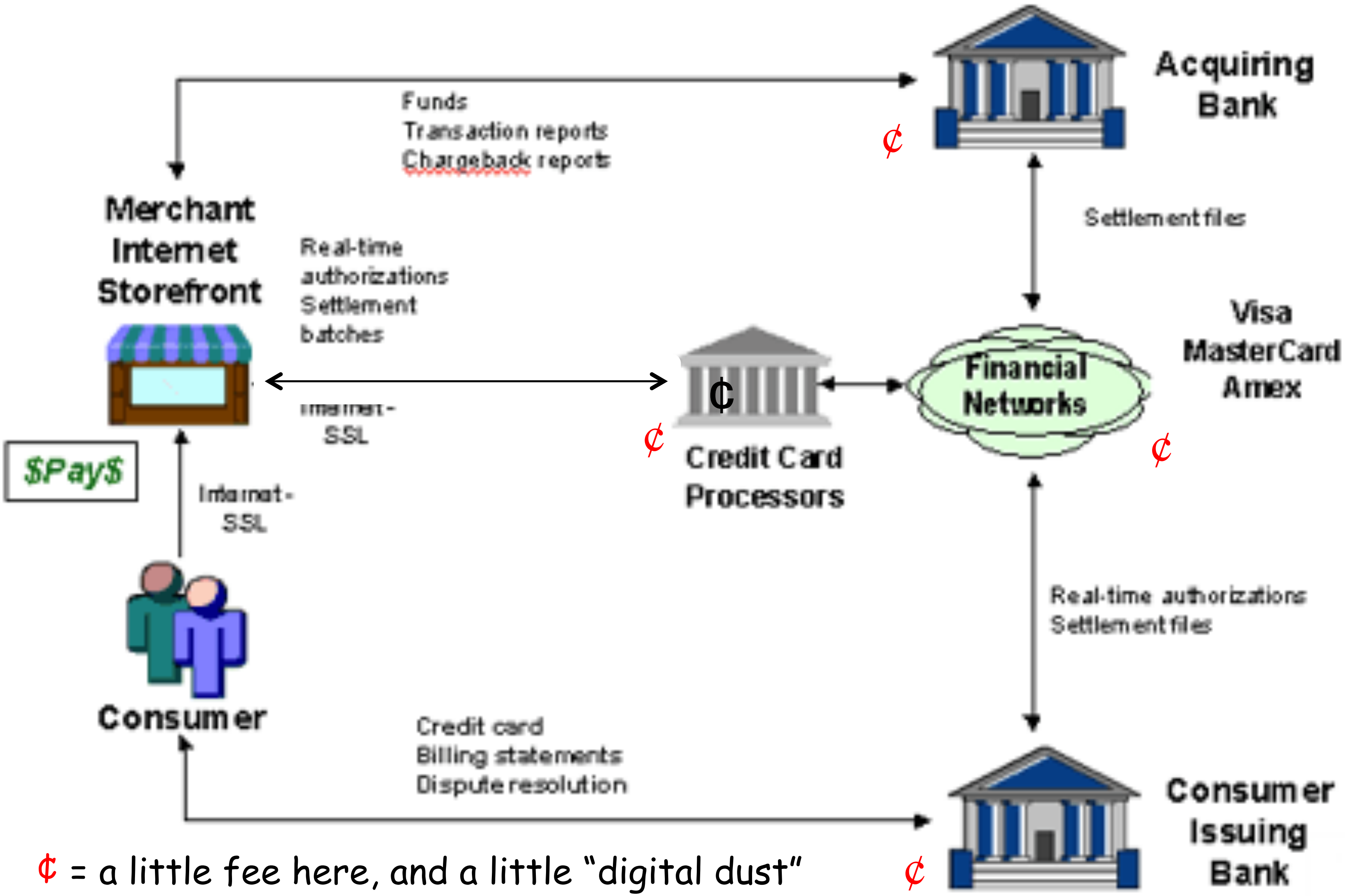
- Prototypes developed and demonstrated (late 1990s)
- One of the inventors (Syverson) pushes to open source the technology (late 1990's to early 2000's)
  - Continues to push for full scale development
  - Additional funding and personnel located to continue the development
  - Deployment of full scale technology (2005+, I think)
  - Picked up by State Department as enabling dissent, free speech
  - Picked up by criminals as enabling cloaking of illegal markets (Silk Road)

Conventional payment processing (credit cards)

Players:

- credit reporting agencies (Experian, Equifax, TransUnion)
  - Collects reports of payment history, % of credit limit used, total balances/debt, inquiries
- credit scoring agencies (FICO, VantageScore) draw on credit reports, predict how likely creditor is to meet obligations
- card issuer (issuing bank) decides to issue card to custorer
- card holder, buys things with the card and makes payments
- merchant, accepts the card for payment
- acquiring bank/processor, reimburses the merchant

How transactions work – credit cards



¢ = a little fee here, and a little "digital dust"

Trust relationships – credit card transaction

- Cardholder: trusts merchant and payment processing infrastructure with credit card number (not to lose or abuse it)
- Merchant: effectively gets a loan from bank until transaction settles; trusts that the transaction will settle
- Banks trust the customer to pay and the merchant to deliver (to avoid chargebacks)
- The payment processing system, which effectively maintains the Ledger, represents a **Trusted Third Party**: something that both cardholders and merchants must trust to not allow unauthorized transactions or drain accounts
- Cardholder and bank also trust merchants not to reveal credit card info it may have stored

Ledger (simplified, one of several)

Account	Amount	Payee	Product	Account	Amount	Payer
Carl	-$28	Amazon	Book	Amazon	+$28	Carl

Underlying technologies and information flows – credit cards

- Encryption:
  - Secure information in transit, (and sometimes information at rest (in files, databases))
  - Digital signatures for data integrity
- Databases: to store transaction information (ledgers), customer data, credit cards, etc.
- Card mag stripes or "chip and PIN" (which can reduce trust in merchant's point-of-sale operation
- Trusted Third Parties

Digital currency

- General characteristics –
  - Anonymous or pseudonymous, private/untraceable
  - Irreversible, accountable transactions
  - Integrity: no forgery/duplication
- General problems:
  - double spending
  - Theft/loss of keys involved
  - Lack of incentive for existing institutions to adopt them
- History
  - 1983 David Chaum (the same one) published an scheme for electronic cash based on blind signatures
  - 1989 Chaum started a company, Digicash, to commercialize this, but it went bankrupt in 1998. Credit cards won out for ecommerce.
  - 2008 "Satoshi Nakamoto" publishes "Bitcoin: A Peer-to-Peer Electronic Cash Systems" and in 2009 provides a reference implentation of software for it

Bitcoin Philosophy and Technologies

- Underlying Philosophy
  - No central authority, everything decentralized
  - Growing pool of currency, but with a finite limit

- Underlying technologies
  - Public key cryptography (Asymmetric crypto)
  - Encryption: Secure hashing (one-way functions)
  - Block Chain: single ledger for all transactions, widely replicated

Bitcoin Elements and Status

1. P2P communication network
2. Transactions and Blockchain
3. Mining and Consensus

- Blockchain provides a single ledger that records all bitcoin transactions worldwide
- The number of bitcoins in circulation increases slowly over time until the maximum number of coins(about 21,000,000) is reached (by about 2040). About 15M currently in circulation
- Current price of a bitcoin in dollars; about $420-$430
- See https://blockchain.info/ for current transactions

Properties of the ledger

- Holds all the transactions
- Transactions can't be altered after the fact
- Transactions can't be inserted after the fact
- Only agreed-upon transactions are added

How to assure these properties with digital data structure?

Bitcoin Basics: Transactions

Transactions:

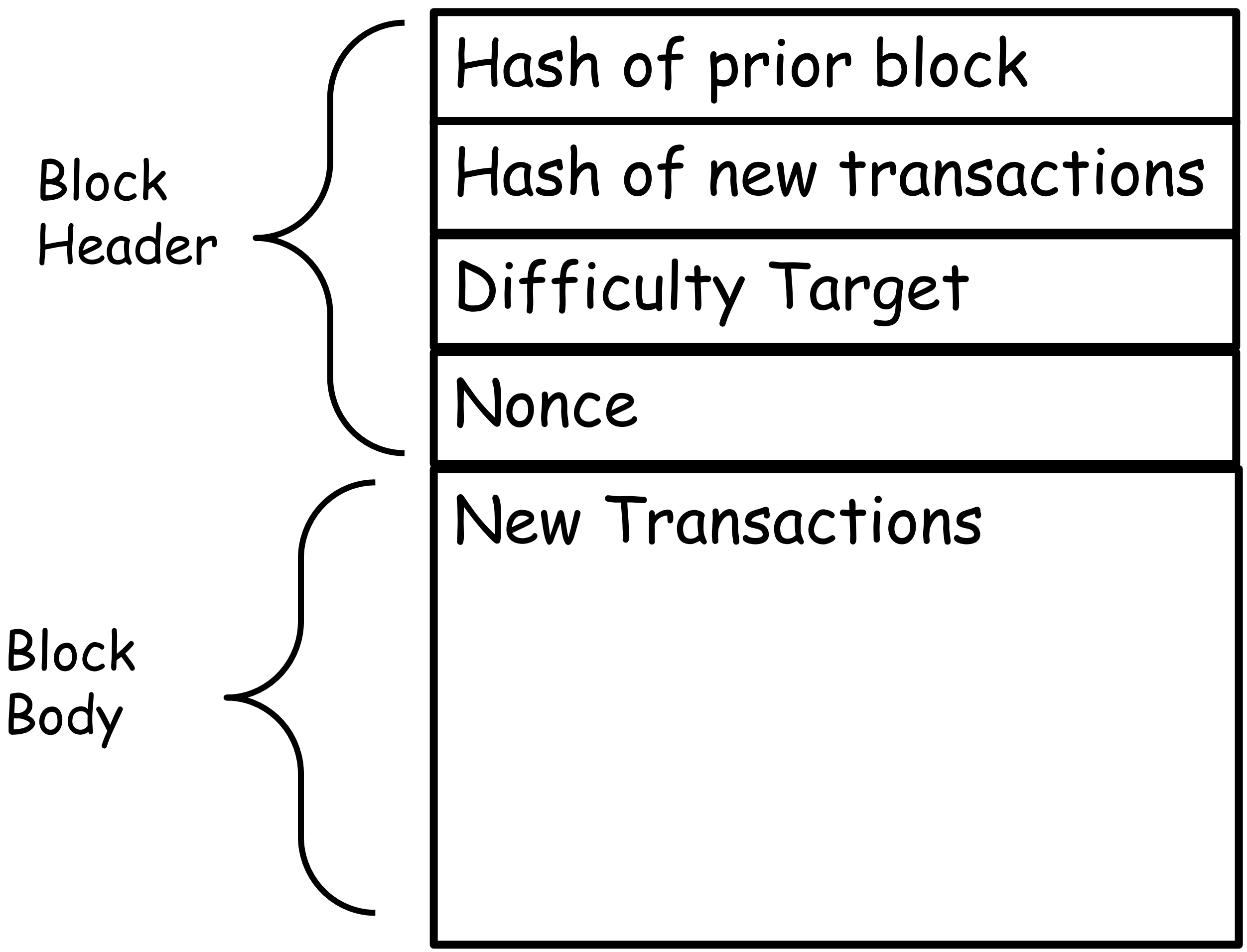
- Alice wants to send Bob some bitcoin
- Alice creates transaction that includes
  - Inputs: bitcoin transactions that sent bitcoin to Alice
  - Outputs: number of bitcoins to transfer to Bob (and others)
- "Alice" and "Bob" are really just account numbers (i.e., hashes of public keys).
  - Being able to sign with private key means you "own" that account.
  - Accounts can be created anytime by any participant, by generating a new public/secret key pair
- A broadcasts the proposed transaction to the entire bitcoin P2P network
- A new transaction becomes real when it is incorporated into the Blockchain – a chain of all bitcoin transactions, ever (see next slide)

Bitcoin basics: “Mining” and consensus

- How does a new transaction get added to the blockchain?
  - There is no central authority to perform this operation (having one would mean it would be able to “censor” transactions, charge high fees, etc. – it would become powerful and trusted
  - Instead, there are many bitcoin nodes (i.e. computer systems running bitcoin mining software)
    - Each listens for new transactions, collects them over a short period, and makes them into a new “Block”
    - Miner checks that proposed transactions are valid (signatures are right, no double spending in relation to existing blockchain, etc.)
    - Miner must find a block: new transactions, hash of prior blockchain, and an arbitrary “nonce” value that has a secure hash value smaller than some specified number

What the miner is racing to do:

Possible New Block to be added to the chain



Mining Task:

Compute Hash of the Block Header for different values of "Nonce" until you find a hash that is numerically less than the difficulty target

(i.e., it has a specified number of leading zeroes)

When you find it, broadcast the block with the lucky "Nonce" you found and you win@!

Why would anybody play this game?

The winning miner gets paid in two ways:

- It gets to keep some new bitcoins
- It gets to keep the transaction fees from this block

(transactions for both of these payments were included in the block of new transactions)

What's the benefit?

- The validity of the new block can be checked by anyone
- We have a way to achieve consensus on the contents of the global blockchain without resorting to a trusted third party

Some issues with Bitcoin

- Key management:
  - Where do I store my private keys?
    - If I lose my private key I've lost the ability to do transactions on that account
    - If somebody steals my private key they can do transactions for me
- Consensus:
  - It's possible for two miners to propose new valid blocks at about the same time. This can lead to "forks" in the blockchain
  - There is a way to resolve these in practice (pick chain with the higher "proof of work" but there is no strong proof of convergence

Comparison of some aspects of bitcoin and credit card transactions

- Credit card transactions
  - Trusted third party (several)
  - Non-private
  - Reversible
  - Settlement time in days
- Bitcoin transactions
  - No trusted third party
  - Pseudonymous / semi-anonymous
  - Non-reversible
  - Settlement time in minutes/hours

Backup slides follow