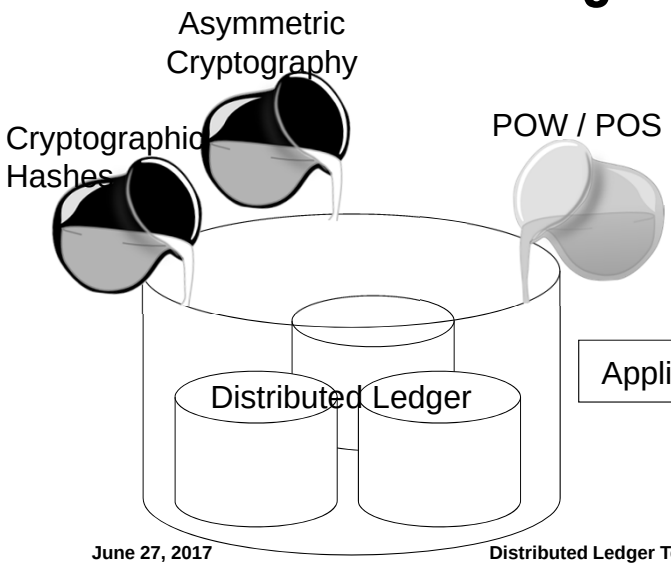
Distributed Ledgers

Blockchain Technology

Michael B. Spring
School of Information Sciences
University of Pittsburgh
spring@pitt.edu

The Big Picture



A Perfect World

Overview

- Introduction
  - Transaction Costs
  - Distributed Ledgers
  - The Bitcoin proposal
- The Blockchain Approach
  - Asymmetric cryptography and cryptographic hashes
  - Transactions and Ledgers
  - Blockchain Ledger Reliability
  - Proof of work
  - 51% vulnerability
- Evolving Distributed Ledgers
  - Private Permissioned
  - Public Permissioned
- Applications
  - CU Ledger
  - Sovrin
  - Other Possible Applications
- Additional Information
  - More on Bitcoin Transactions
  - More on Identity & Certificates

Blockchains and Transaction Costs

- In 1937, Ronald Coase published an article titled “The Nature of the Firm” (*Economica (Blackwell Publishing)* ***4*** *(16): 386–405.)*
  - Transaction costs in an open market include search, negotiation, execution, and policing or enforcement
  - The basic premise of the paper is that as the costs associated with transactions decreased, firms would shrink in size.
- Digital systems reduce the costs of many types of transactions by orders of magnitude.
- Distributed digital ledgers can be used to further reduce the costs of the ledger that provides for secure trusted transactions.

What is a Distributed Ledger

- At the most basic level, a distributed ledger is simply a replicated data structure – e.g. one or more database tables.
- The concept of a distributed public permissionless ledger, as first introduced by Bitcoin, was given the name “blockchain”.
- The things that make a blockchain special are:
  - It is distributed – there are multiple copies that are synchronized
  - It is “public” and “permisionless” – anyone can work on it
  - It is tamper resistant – additions to the data structure are made in a way that provides reasonable assurance they are not fraudulent
  - There are mechanisms that provide incentives for organizations to contribute to the maintenance of the data structure
- There are other approaches such as hash trees or Merkle trees and hashgraphs

The BitCoin Proposal (2008)

- There has been a long search for digital cash – a way to exchange funds the same way we do with cash.
- In 2008, Satoshi Nakamoto published a paper entitled “Bitcoin: A Peer-to-Peer Electronic Cash System”*

  It should be noted that there is no definitive identity for the author. Despite significant efforts to determine the identity of the individual or group responsible, no one has as yet been identified.
- The bitcoin proposal provided a mechanism for:**
  - “Private” transactions of the kind we normally use cash for.
  - “Trusted” transactions of the kind normally managed by a bank.

* To read the original paper on bitcoin, see https://bitcoin.org/bitcoin.pdf

** Michael Nielsen provides a wonderful introduction to the concepts in: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

Banking Refrain

- In a traditional banking scenario, a single institution maintains a “private ledger” with accounts.
- Most commonly, the bank validates transactions that are not anonymous between two individuals/organizations.
- Within this system, an individual can withdraw cash from an account and conduct anonymous transactions.
- Cryptocurrencies are the most recent effort directed at anonymous digital transactions – Bitcoin and Ethereum are among the best known.

General Notions

- To allow for cash like transactions, Bitcoin needed to provide for anonymous transactions
- To guard against fraud, Bitcoin needed to provide a distributed ledger system that was not privately managed
- To accomplish these goals, various cryptographic technologies are employed
  - Asymmetric cryptography (public-private keys) provide a mechanism for transactions
  - Cryptographic hashes provide a base of managing ledgers

Asymmetric Cryptography Primer

- Various algorithms – in the asymmetric cryptography family – produce pairs of keys such that one key can be used to encrypt data and the other key can be used to decrypt the data.
- One of the keys is made public and the other key is kept private.
- When a public key is used to encrypt a message, it may only be decrypted by the corresponding private key – thus a message can be sent that can only be read by the intended recipient (confidentiality).
- When a private key is used to encrypt a message, it can only be decrypted by the corresponding public key and thus its "origin" is known (validity).
- When the encrypted message that is sent has a corresponding secured hash of the message, the hash provides a means of insuring the message has not been altered (Integrity).

Anonymizing Transactions (Quickly)

- Owners transfer information (e.g. money) by digitally signing a hash of a previous transaction (one that gave them something) and the public key of the next owner, adding these to the end.
- Individuals commit transactions which are broadcast
  - The owners private key validates the origin
  - The recipients public key identifies that the transaction or transfer is for them
- If the public/private keys are signed by a known authority, it is also possible:
  - To assert the validity of the message
  - To hide the owner of the keys
- Further, by using one time key pairs, traffic analysis of key usage can be frustrated.

Cryptographic Hash Primer

- Hash functions have three basic characteristics:
  - Given any input, the function produces a fixed size output
  - The output may not be used to reproduce the input
  - Any change to the input will result in a different output
- Hash functions serve a variety of cryptographic uses
  - Hashes ("fingerprints") of programs allow users to make sure code has not been modified. If a hacker were to modify the code, checking the hash of your program would show it had been changed.
  - Hash functions can be used to assure the integrity of ledgers of transactions.

For an online hash calculator, see:
http://www.xorbin.com/tools/sha256-hash-calculator

Transactions, Blocks and Blockchains

- Transactions are broadcast to the network where interested parties collect and bundle the transactions to make a block.
- Under Bitcoin the individual who creates the block that is used to update the ledger is compensated in two ways
  - To prime the system, new Bitcoins are provided by the system
  - In the future, transactions would include a "transaction fee" paid to the parties that build the blocks that are added to the block chain
- As explained below, these parties are called miners

Essence of the Bitcoin Blockchain

- Transactions distributed to “the network” are combined into a block that is linked to the existing blocks – a blockchain
- The blockchain serves as the distributed ledger and prevents faking ownership of, or double spending of, funds
- Anyone can add a block to the Bitcoin blockchain.
- The process is based on an assumption about human behavior:
  - The more effort humans need to expend to “cheat” with only a small probability of succeeding, the less attractive cheating becomes.
  - The integrity of the ledger is assured by the fact that multiple individuals invest time – a process that involves “proof of work” – in building the blockchain in exchange for rewards

Ledger Reliability

- Keep in mind, if a trusted party kept the ledger, that party would be responsible for insuring the ledger could be relied on – such cannot be assumed in a public permisionless distributed ledger.
  - Contributions to the distributed ledger go through a process that requires “proof of work”.
    - The “work” is such that the contributor, called a miner, cannot be guaranteed their contribution will be accepted – discouraging fraud.
  - Blocks are linked in the block chain such that changes to historical blocks are virtually impossible
    - Each block contains a hash of the previous block such that a change to an earlier block would cause a hash mismatch in every succeeding block

"Proof of Work" Conceptually

- "Proof of work" in block chain management is conceptually simple:
  - A miner who seeks to add a block to the chain is presented with a task for which the answer is difficult to find, but easy to validate.
  - If checked and found correct, the user's solution is accepted
- Consider a simple example of "proof of work":
  - You are asked to determine the number between 1 and 100 that I am thinking of
  - You must guess, on average, 50 numbers to get the right answer. If 10 people are guessing, someone might get it after 2 guesses and someone else after 99.
  - Whoever guesses the right number is allowed to add the next block.
  - If you are cheating, there is no guarantee you will be the one to add the block – you may do a lot of work for nothing
- The next slides explain the actual task set out for those who would add blocks

Blockchain 'Proof of Work': Nonces and Hashes

- The Shaw Hash Algorithm (SHA-256) generates a 256 bit value. The examples below show the beginning of some hashes in hexadecimal.

Text	Hash
BlockChainHash	c799bf416f8303fcb0370f231ab97d…
BlockChainHash1	beda7b6bf8ca5b9c98ca1c5ecf46a…
BlockChainHash3	05ea628ed8d7fa552636eb9e1998…

- Note that the hash of the string changes completely if we add anything to it
- In these examples, look at the beginning of the hash
  - In the third example the first hex digit is 0, which equates to four binary digits being "0"
  - By adding different suffixes (or nonces), we will eventually find one that will produce a number of "0"s at the beginning.
  - Finding a suffix that puts 40 binary 0's at the beginning could take 2^{40} or a trillion tries

"Proof of Work" in the BitCoin Blockchain.

- Individuals who propose blocks to be added to the blockchain are called "miners"
- A miner composes a hash of the transactions they have collected.
- In addition, a miner must add a nonce to a hash provided from the previous block in the chain such that the hash plus the nonce result in a hash that begins with some number of "0"'s
- Depending on the goal, finding a nonce that achieves this result requires many tries (as many as 4 trillion hashing operations if we ask for 40 zeros.)
- This is the "proof of work" test that miners must do to be allowed to add blocks to the Bitcoin blockchain.
- Also note that the other nodes in the network do virtually no work to prove that the hash of the previous block plus the nonce works

Basic Assumptions about Mining

- The goal is to allow miners to provide Proof of Work within 10 minutes
  - For Bitcoin, the proof of work must be a result hash with some minimum number of zeros at the beginning.)
  - The target is adjusted to keep the effort required to around 10 minutes
- Miners are rewarded by block contribution fees
- If two contributions are made simultaneously, ties are broken by the longest proof of work chain
- Once a new block is proposed, other miners have the choice of:
  - continuing to search for an alternative more favorable to themselves or
  - accepting the new proposal and searching for the next one.
- The community understands that they are following a natural consensus and that for each block different miners will likely generate the blocks. (All participants have equal probabilities in each round.)

The Result of "Proof of Work"

- Keep in mind, block chains could be constructed without the concept of proof of work if everyone behaved honorably
- The reward that comes from generating a block of transactions is the incentive for people to contribute to generating the ledger
- The fact that others are competing to extend the ledger and that doing so requires work means any one individual has a low probability of actually contributing any given block.
- This discourages people from trying to corrupt the ledger by modifying, adding, or omitting transactions
- One exception to this occurs when one or more colluding contributor(s) control the majority of the computing power – this is the 51% vulnerability (see next slide.)

51% Attack Vulnerability

- An important assumption in the design of the Bitcoin blockchain is the notion that "miners" in the network are widely distributed and each has a roughly equivalent amount of computing power.
- If 1000's of miners are at work, each block is likely to be more or less randomly selected from one of them and the efforts of fraudulent participants are discouraged.
- If one of the miners were to have significantly more power (more than 50%), they might be able to dominate the blockchain manipulations so as to corrupt the chain.

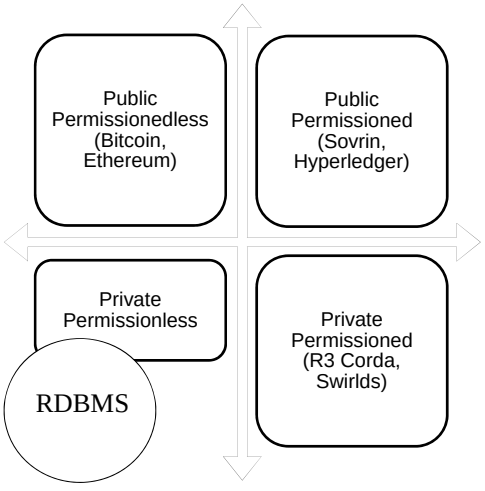
Other Distributed Ledger Architectures

One of the issues with the Bitcoin blockchain is the throughput.

Allowing multiple contributors requires proof of work to insure integrity and that requires time and throttles throughput

Non-Bitcoin Distributed Ledgers

- Researchers have begun to explore ways of building distributed ledgers beyond the Bitcoin blockchain
- The table to the right shows one way of classifying distributed ledgers based on whether they are:
  - Public or private – open to all or restricted to a small group
  - Permissioned or permissionless – managed by privileged nodes or managed by anyone who wants to manage



Private and Permissioned Distributed Ledgers

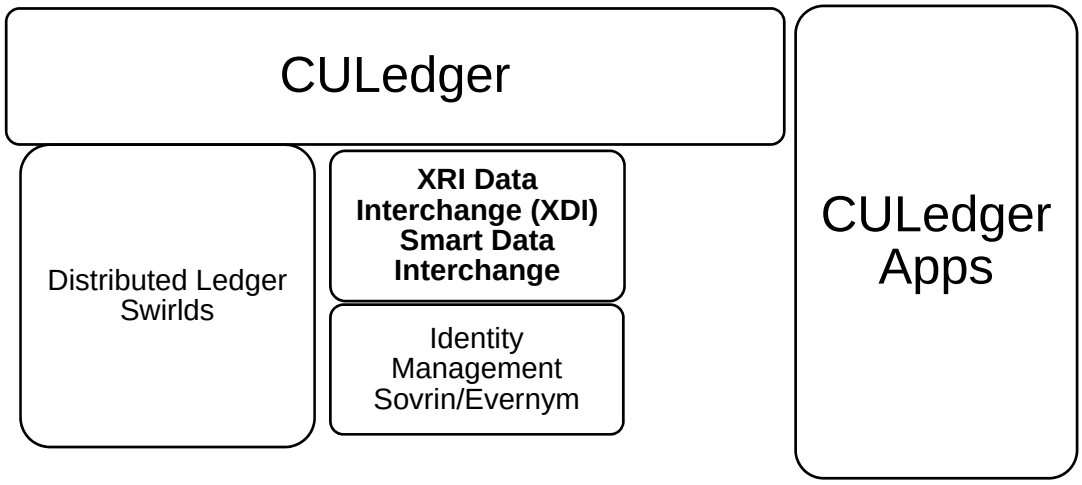
- Private systems with restricted users – who are assumed to be honest – do not require proof of work and have greater throughput – Swirlds is an example
  - Swirlds uses a hashgraph to insure integrity of the distributed system
- Public systems that are permissioned – have a mechanism other than proof of work that provides an assurance of honest contribution -- Sovrin is an example
  - The Sovrin ledger is globally distributed ledger of identity records maintained by "trusted institutions" around the world who would be governed by the Sovrin Foundation*

*https://sovrin.org/wp-content/uploads/2017/04/The-Technical-Foundations-of-Sovrin.pdf

CULedger Project

- CULedger is a demonstration use of Distributed Ledger Technology being moved forward by the Mountain West CU Association and the Credit Union National Association
- It is meant to demonstrate the use of permissioned DLT -- Swirlds -- and distributed identity technology – Sovrin/Evernym
- For more information, see http://culedger.com/
- The next Page shows how the component pieces relate

CULedger Components



Swirlds

- Swirlds* is a distributed ledger based on a “hashgraph”
- While it can be run in permissionless mode, it is most frequently permissioned and avoids the delays and throughput limitations of a blockchain solution based on proof of work
  - When run in permssionless mode it makes use of proof of stake
- It uses a distributed consensus algorithm and a gossip protocol to allow it to order transactions in the system

- *http://www.swirlds.com/

Evernym/Sovrin

- Evernym developed Plenum, a dervivative of the Practical Byzantine Fault Tolerant (PBFT) protocol, a protocol to assure reliable distributed systems
  - PBFT was improved through protocols such as aardvark and RBFT (Redundant..)
  - Jason Law & Lovesh Harchandani proposed Plenum as yet the next step in a protocol specifically designed to support a DLT for identity management
- Sovrin* uses Plenum and a qualification or permissioning process to operate the nodes of a DLT

- *https://sovrin.org/

The Sovrin Identity Magement Layer

- The goal of the Sovrin effort is to provide an identity management layer for the internet
  - It provides a means of moving beyond individual password systems, beyond SSO efforts, and beyond federated systems
- Steward organizations provide the trust layer for writing to the distributed ledger
- Individuals completely own their identity – called self sovereign

How Sovrin Identity Management Works

- The ledger contains a set of transactions for an identity
- Identity transactions are inserted by Stewards
- The owner of an identity will have both public and private attributes
  - Public attributes will be stored unencrypted in the ledger
  - Private attributes will be encrypted on a personal device and linked to the public ledger by a hash of the private ledger
- The owner will use the ledger to manage their identity using several types of operations (see next slide)

Basic Operations in Sovrin

- Identifiers: cryptonyms using Edwards-curve Digital Signature Algorithm Variant (EdDSA25519) :
  - Signing key (private)
  - Verification key (public)
- Claims: assertions or attestations about an identity
  - There are standard claim definitions – schemas and ontologies
  - They can be self-asserted (name) or asserted by others(diploma)
  - Each claim is digitally signed
- Disclosures and Proofs
  - Disclosures are built by the owner and consist of a bundle of claims about the owner
  - Provides a Proof of the attributes of the owner to the reciepient
  - Does not allow the recipient to access other information from attesters to the claims included
  - Disclosures can require Consent Receipts which detail how data can be used

Some Other DLT

- Permissioned Ledgers
  - Ripple Labs -- https://ripple.com/
  - Clearmatics -- http://www.clearmatics.com/
  - Hyperledger -- https://www.hyperledger.org/
  - Corda -- https://www.corda.net/
- Permissionless Ledgers
  - Ethereum -- https://www.ethereum.org/
  - NXT -- https://nxt.org/
  - Tendermint -- https://tendermint.com/

Possible Uses of Distributed Ledgers

- Micro transactions:
  - Music, game, document use
  - Personal information use: some have suggested users should be compensated for Google and Amazon recommendations based on their behavior.
- Registries
  - Identity registries (such as Sovrin for identity management)
  - Ownership registries such as land, homes, cars
  - Agreement registries such as marriages, loans, births

Possible Uses of Distributed Ledgers

- Activity management
  - They could be used by collectives such as The Credit Union (CU) project described above
  - Uber drivers or Airbnb providers are collectives with their own record keeping rather than a centralized one.
- Process management
  - Execution and verification of service transactions (supply chain)
  - Recording of business processes (document management)
  - Provenance for scientific data
  - Transfers of guns, explosives, medical drugs, etc.

More on BitCoin

See https://blockchain.info/ for more info on Bitcoin and its current state

See https://blockexplorer.com/ to explore the blocks and transactions in the blockchain

Bitcoin Transactions

- The number of Bitcoin transactions has increased over time.*
  - Through 2010, the daily transactions numbered in the hundreds
  - In 2011-2012 they increased to thousands
  - Mid 2012 through 2014, with a couple exceptions, the daily number stayed under 100,000.
  - 2015- 2016, they ran between 100,000 and 250,000/day

*https://blockchain.info/charts/n-transactions?timespan=all

Bitcoins

- Like any currency, the value of a bitcoin has varied*
  - Through 2011, bitcoins had minimal value
  - On February 1 2011, the value of a bitcoin was $1US
  - There have been several spikes in value (266 on 4/11/2013; 1124.76 on 11/29/2013)
  - Valued between $250-700 in 2014-2016
- Bitcoins may be divided to a limit.
  - The smallest bitcoin value allowed is 10^{-8}.
  - This value is called a satoshi

*http://bitcoincharts.com/charts/bitstampUSD#igWeeklyztgTzm1g10zm2g10zl

Mining Reward System in BitCoin

- Miners are rewarded for the effort they expend in establishing blocks by an award of Bitcoins
  - In 2009 the award was 50 Bitcoins for every block accepted.
  - For every 210,000 blocks validated, the value of the reward is halved
  - The current block reward, established on November 28, 2012, is 25 Bitcoins
  - The next block reward reduction – to 12.5 – is expected in July 2016
  - Assuming bitcoins current trajectory, rewards for block validation will reach a theoretical minimum sometime around 2140.
- Miners also collect transaction fees which may be associated with each transaction
  - Initially most fees were 0, but they are expected to increase

Transactions

- Again, at a conceptual level, transactions stored in a block chain follow some simple rules
- Each transaction consists of an input and an output which must be equal (or at very least, the input must be greater than the output)
- The inputs represent sources from previous transactions that have been accumulated by the originator
- The outputs are values transferred to recipients
- To pay a small part from one existing value, the originator creates an output back to themselves for the difference

Transaction Anonymity

- By its nature, blockchains are public and therefore transactions are public.
- Better anonymity can be achieved via several mechanisms
  - Make sure your wallet and the key pairs it contains remain private
  - Buy your bitcoins anonymously using a service – either a company or a peer to peer exchange.
  - Use a new key pair for each transaction to keep them from being linked to a common owner
- The process of anonymizing your transactions can never be guaranteed to work, so read up on the array of techniques that can be used.

Transaction Verification (Simple)

- The transaction is formed as follows
  - The input script of the new transaction references the output script of previous transaction(s). The creator proves ownership by using its Public Key and verifying the Signature.
  - The output script for the new transaction is formed by storing the hash of the address(es) of the recipient.
- Verification is then simple. Assuming the ledger (blockchain) is valid:
  - The Blockchain is searched for the referenced outputs of a previous transaction(s) to be sure they exist and have not already been used.
  - The sum of the output is checked to make sure it is less than or equal to the input

Transaction Verification (More complex)

- Miners only include transactions that don't break the rules in blocks.
- The rules include semantic and syntactic checks. It is difficult to simply explain all the checks.
  - For a "simple list" see ttps://en.bitcoin.it/wiki/Protocol_rules#.22tx.22_messages
  - For full details see https://bitcoin.org/en/developer-guide
- Semantic checks include:
  - The input and outputs have values.
  - A matching transaction must exist for each input, i.e. the output must exist and not have been spent.
  - The input value is greater than the output value.
  - The values must be more than 0 and less than 21 million.
- See the next slide for some of the syntactic checks.

Syntactic Miner Checks on a Transaction

- Additional Checks that a transaction must pass
  - The transaction syntax and data structure are correct.
  - The transaction is less than the block size of 1 MB.
  - None of the inputs have a hash that is equal to 0.
  - The locktime is less than the maximum allowed number.
  - The number of signatures is less than the signature limit.
  - If the transaction is a coinbase transaction then it must have 100 confirmations.
  - The unlocking scripts for each input must be verified against the output locking scripts.
  - The transaction size is greater than or equal to 100 bytes.
  - The unlocking script can only push numbers onto the stack.
  - The locking script must match is standard format.
  - Check that each input value is in the required range.

More on Encryption and Certificates

Public-Private (Asymetric) Key Encryption

- Rivest, Shamir, and Adelman developed a public-private key encryption system known as RSA
- Very simply, RSA keys are developed as follows:
  - Two large(256 bit) primes, p and q are generated where n= p*q
  - Two large numbers, d and e are also generated such that e*d-1 is divisible by (p-1)*(q-1) (e shares no factors with p and q)
  - The public key is a function of (e, n) and the private key is a function of (d, n).
  - There is no known method of calculating the values of d, p, or q knowing only n and e.

Cryptographic Hashes

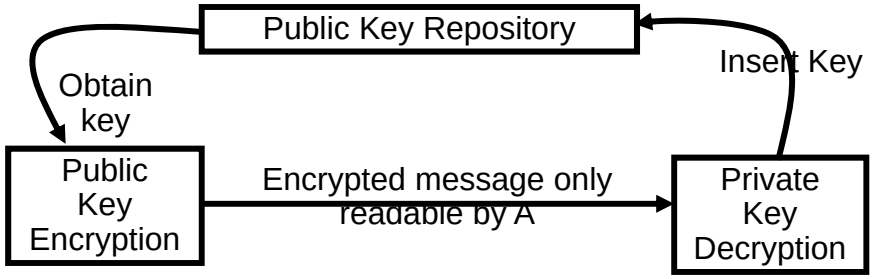
- Cryptographic hashes are one way functions – the original message cannot be derived from the hash.
  - A cryptographic hashes function reduces a message of any length to a single large number – normally between 128-256 bits.
  - A hash spreads influence from throughout the message across the hash -- for any changed input bit, there is a 50% chance of each output bit changing.
- Existing message digest functions are very computationally efficient and free of royalties
  - RSA developed a number of hashing algorithms – MD2, MD4, MD5, which were found to be inefficient or flawed in some way
  - NIST developed several versions of the Secure Hash Algorithm (SHA, SHA-1, SHA-256, etc.). SHA-1 is thought to be secure. SHA-256 was designed for use with AES
- They are used for a variety of purposes:
  - To generate text for encryption as a digital signature
  - To assert the integrity of a software package
  - To assert the integrity of a ledger

Use of Public/Private (Asymmetric) Keys

- Asymmetric keys make it possible to:
  - Transmit a message which can only be read by the intended recipient
  - Authenticate users based on digital signature
- Hybrid systems are the basis of
  - Secure sockets (SSL) and certificates (backed by certificate authorities)
- Asymmetric keys are subject to the same kinds of attacks as symmetric keys:
  - Brute force key search
  - Analytic attacks

Private Messages

- Consider two users, A and B.
- B wishes to send a private message to A
- B uses A's public key to encrypt the message
- Only A can decrypt the message using A's private key



Digital Signatures

- User A takes their private key and encrypts a message they wish to send.
- The resulting cipher is attached to the original message.
- User B uses A's public key to decrypt the cipher attached to the message
- If the clear text message and the decrypted message match, the message has integrity and it must have been sent by A.
- The cipher is called a digital signature.

Signed Encrypted Messages

- A “total” solution -- message from A to B
- Frequently, rather than encrypt a long message to create a signature, a hash of the message is created using a known method (e.g. MD5, SHA-256)
- Encrypt the message digest to create a signature using A’s Private Key
- Encrypt the message and signature using B’s Public Key – which assures privacy
- Decrypt the message using B’s Private Key and authenticate the origin by using A’s Public Key to verify the digest

Certificates

- The missing ingredient here is some assurance that I am who I say I am
- A certificate is a parcel of digitally signed information – basically a signed message
- The signature comes from a “Certificate Authority” who is recognized as trustable
- What the certificate says is considered to be true if the decrypted signature (using the Certificate Authorities Public Key) is valid.

An Example of What Might be Done

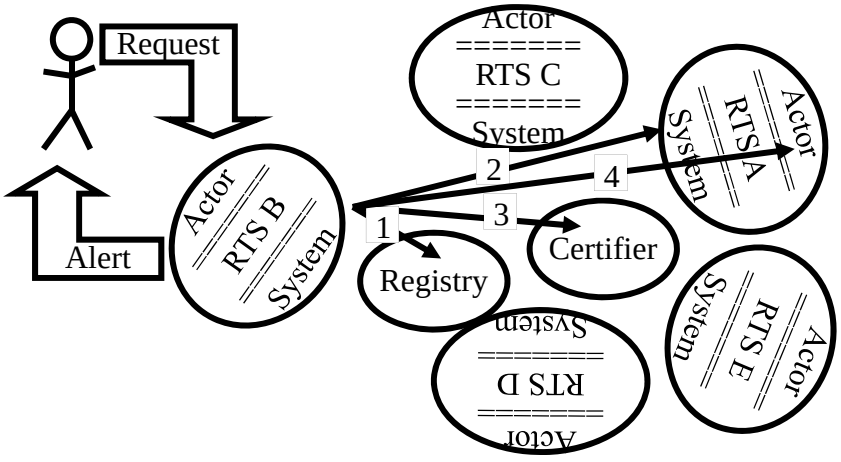
Referral Trust as an Example

- When a physician makes or serves as a referral, the advice of other physicians is sought.
- A referral trust surrogate might keep records of who has been a referrer or referral to who and what the satisfaction and/or assessment was.
- Referral trust agents could search for individuals in a given area and report back a web of trust.
- The same surrogate could serve as the physician's documented record for others to use.

Certificates and Directories for RTS

- If one considers the Referral Trust surrogate, it is clear that the architecture of the marketplace requires at least two supportive structures:
  - Registries that allow appropriate Referral Trust surrogates to be found.
  - Certificates and certificate authorities that can be used to establish trust.

Referral Trust Service Process View



The Process

- Participating physicians post a RTS. The RTS describes the physicians capabilities and knows the physicians trusted partners
- After a referral, the referring physicians RTS provides a certificate of engagement and potentially a certificate of assessment.
- When a referral is sought, the RTS searches categories and checks certificates.
- The initiator gets a report and activates the referral