Overview of Cryptography and It's Applications

Topics

- Basics
- Security with Cryptography
- Types of Cryptography
  - Stream vs Block
  - Symmetric vs. Asymmetric
- Digital Signatures
- Intermission: What about Quantum Computing?
- Certificates – Public Key Infrastructure
  - Structure of Certificates
  - Usage of Certificates
- Modern Applications
  - Passkeys
  - Secret Splitting
  - Blockchain

Basics

- **Encryption**
  - Scrambling data to provide privacy
  - A **key** is used to scramble data to be protected (We say the key 'configures' the crypto algorithm)
- **Key**
  - Special value needed to encrypt or decrypt data
- **Decryption**
  - Recovering original data using the key
- **Plaintext** – Original data before encryption
- **Ciphertext** – Encrypted data
- **Cryptanalysis** – Analyzing encrypted data to attempt breaking a cipher

Security with Cryptography

- Main Data Security Concerns
  - **Privacy** – other parties cannot read private data
  - **Integrity** – Data has not been maliciously or accidentally altered
  - **Authentication** – Parties can prove they are who they claim to be

Types of Cryptography
Stream vs. Block

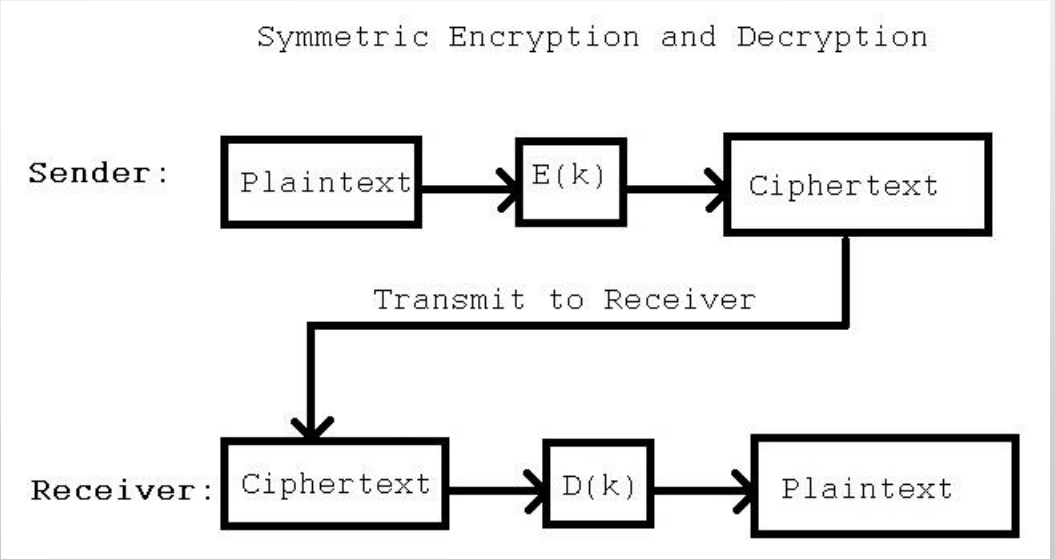
- Encrypting data provides **privacy** keeping data secure from being 'snooped'
- Stream Ciphers
  - Encrypt 1 bit at a time
    - a bit is a '1' or '0'
    - Current characters on a computer take 8 bits or 16 bits to encode
  - Algorithm produces a 'stream' of bits based on the **Key**
  - Uses Exclusive-Or to combine this with data to encrypt
  - Example:
    - RC4 – ssl/tls, wep/wpa
    - A5/1 – cell phone encryption

Types of Cryptography
Stream vs. Block

- Block Ciphers
  - Encrypt 1 block of data at a time
  - Algorithm scrambles a block (32, 64, 128 bits, etc) at a time based on the **Key**
  - **Modes of Operation** are applied to the encryption process. These perform operations combining ciphertext and plaintext to make cryptanalysis more difficult
  - Examples:
    - US or international: DES, CAST, IDEA, AES, Blowfish, RC6, many others
    - Korean: SEED, ARIA, and LEA (also HIGHT which is lightweight but older)

Type of Cryptography
Symmetric vs. Asymmetric

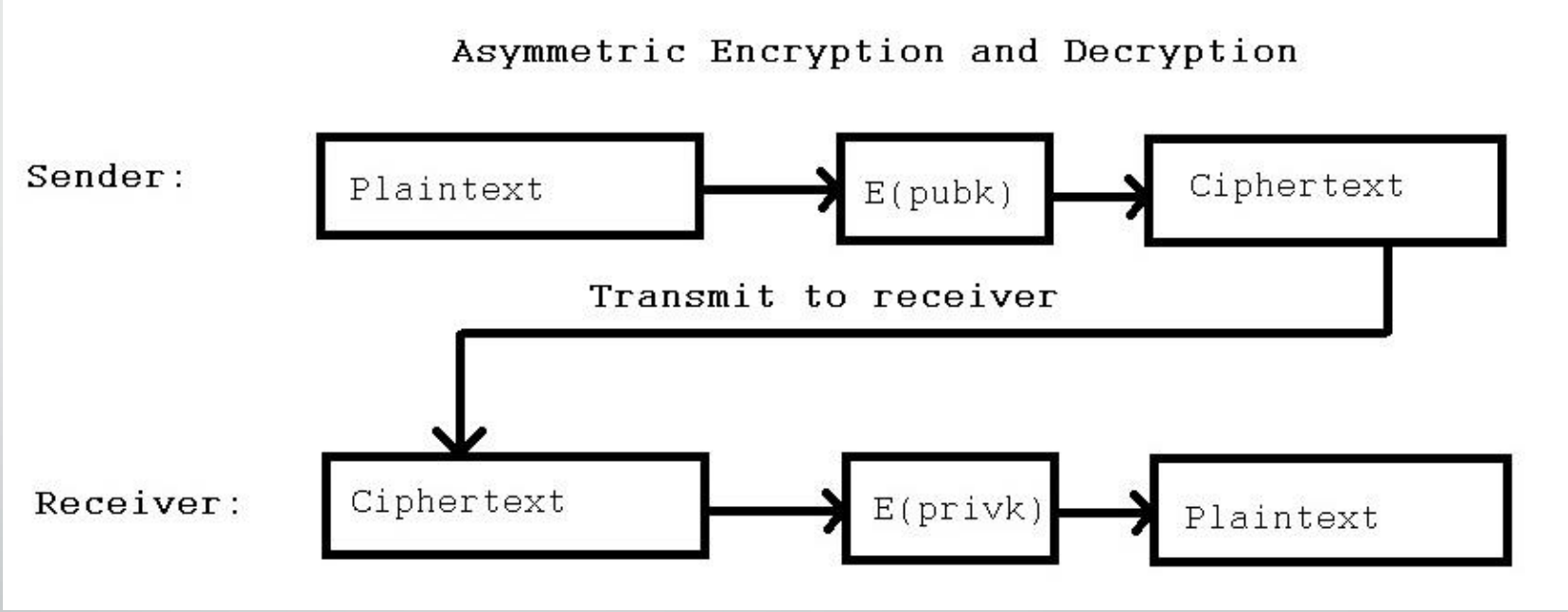
- Symmetric cryptograph requires sender and receiver to share the same key
  - Problem: How do we communicate the shared secret key without someone intercepting it?



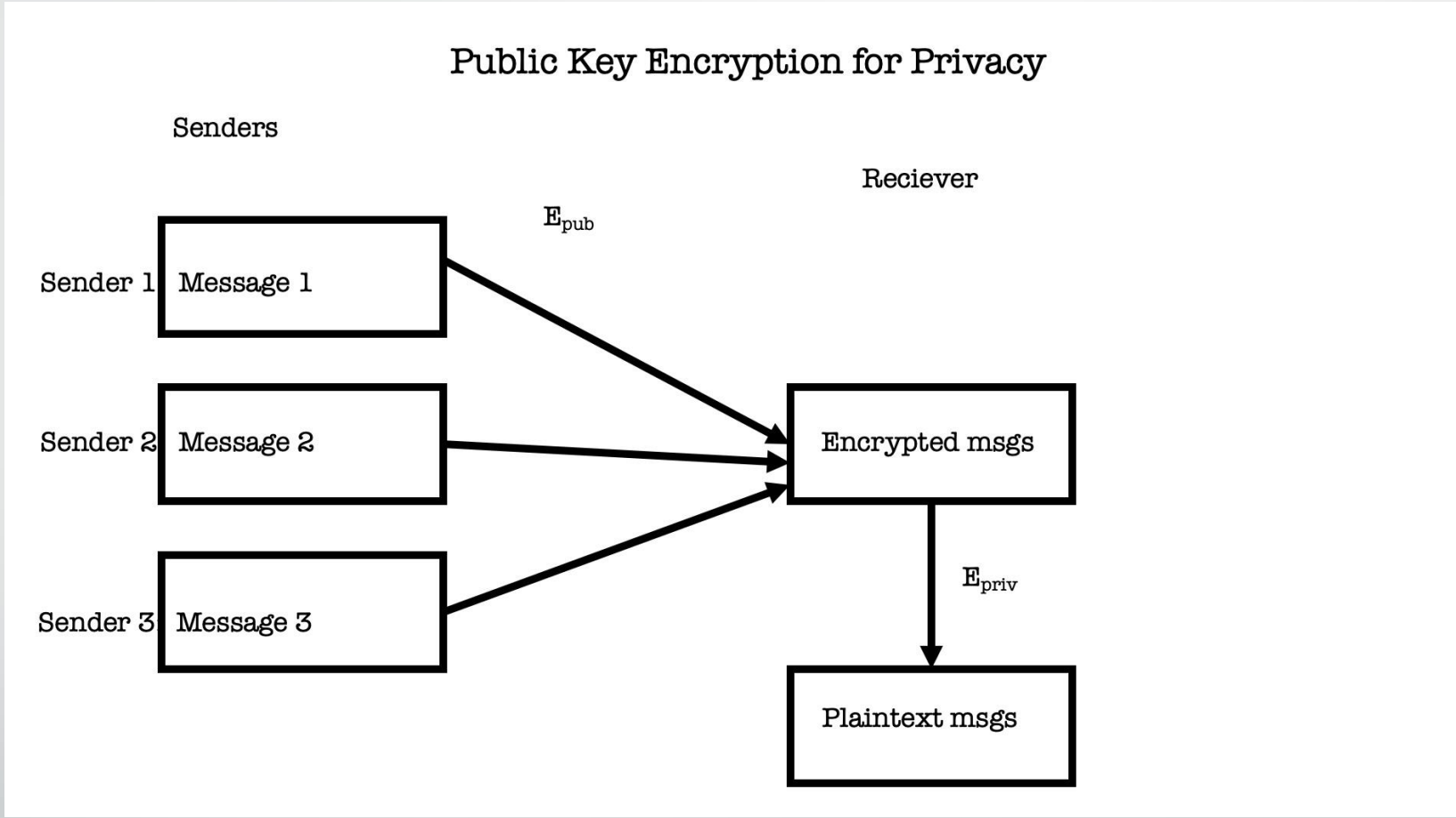
Type of Cryptography
Symmetric vs. Asymmetric

- Great Idea #1: Create a cryptosystem where there are 2 keys: 1 Public, 1 Private
  - Any data encrypted with public key can ONLY be decrypted with private key
- **Asymmetric cryptography** involves decrypting data with a different key than the key with which it was encrypted
  - Generate a pair of ***related keys.*** Keys are mathematically related and can allow one key to decrypt what is encrypted by the other
  - Designate 1 key as **Public** to be given out to anyone who needs to encrypt for the receiver
  - Designate other key as **Private.** This must be kept secure by receiver
  - Examples: RSA, Elliptic Curve
  - Solves problem of how to transmit secret key!

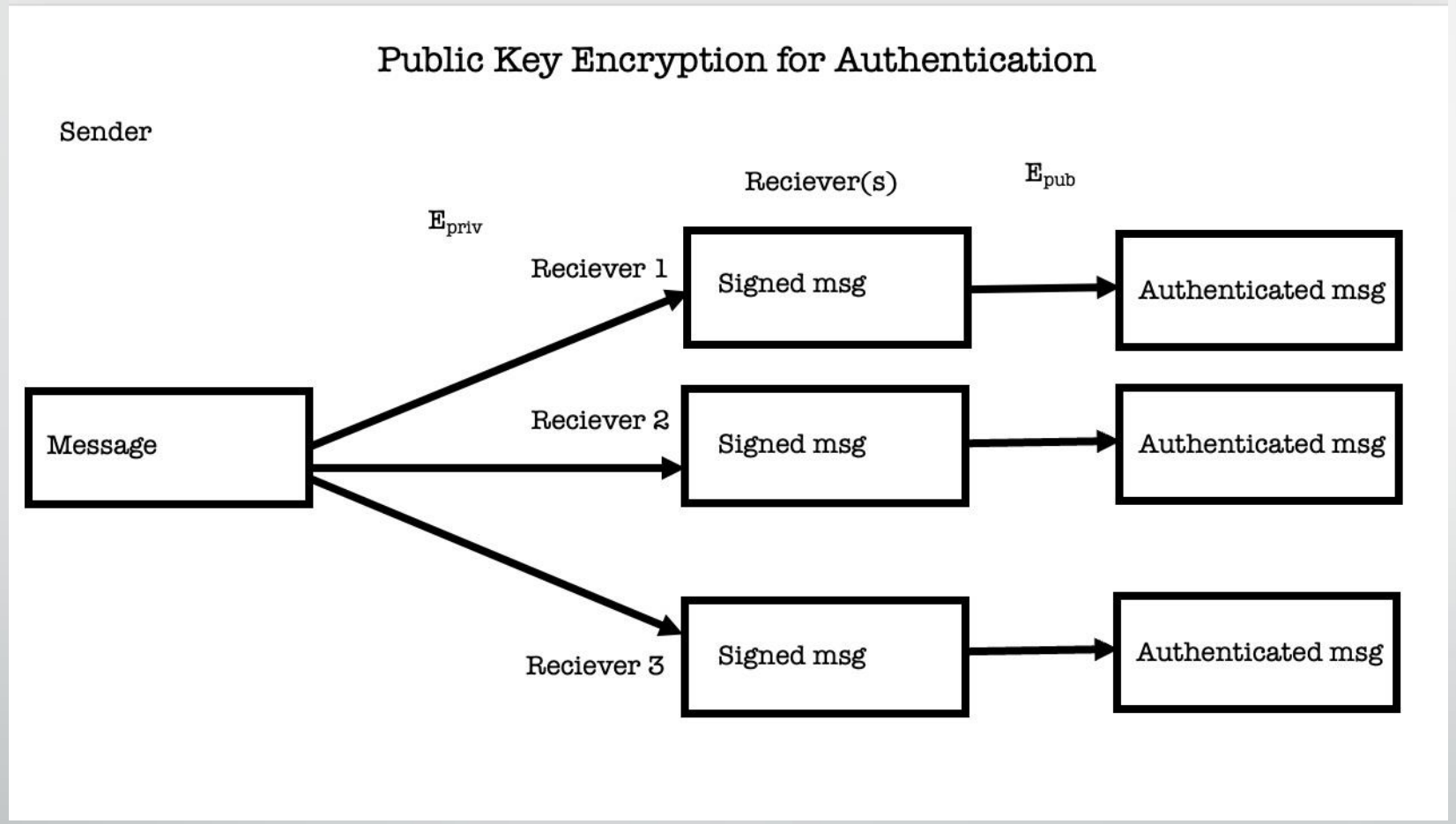
Type of Cryptography
Symmetric vs. Asymmetric



Asymetric Cryptograph (Graphical Summary)



Asymetric Cryptograph (Graphical Summary)

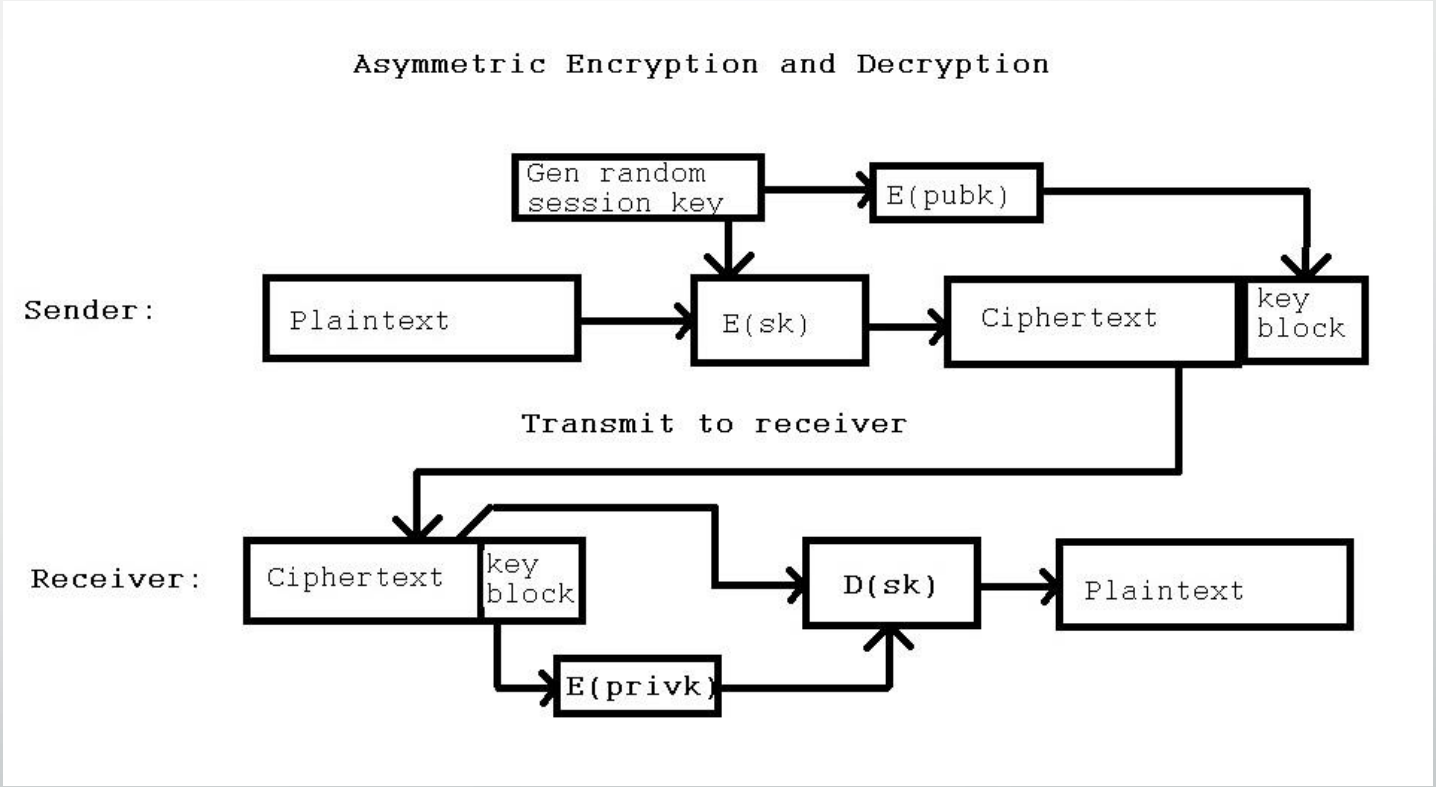


Type of Cryptography
Symmetric vs. Asymmetric

- New Problems:
  1. Asymmetric algorithms are computationally expensive (take lots of CPU)
  2. How do we know the public key sent to us belongs to the person we are trying to communicate with? [We'll fix this later]

Type of Cryptography
Symmetric vs. Asymmetric

Fixes problem of 'sharing' a secret over open communication line:
- Session key is generated
- Sent with the ciphertext after encrypting with public key



Type of Cryptography
Symmetric vs. Asymmetric

- Problem 2: How do we know the public key belongs to the named person?
  1. Digital Signature
  2. Certificates

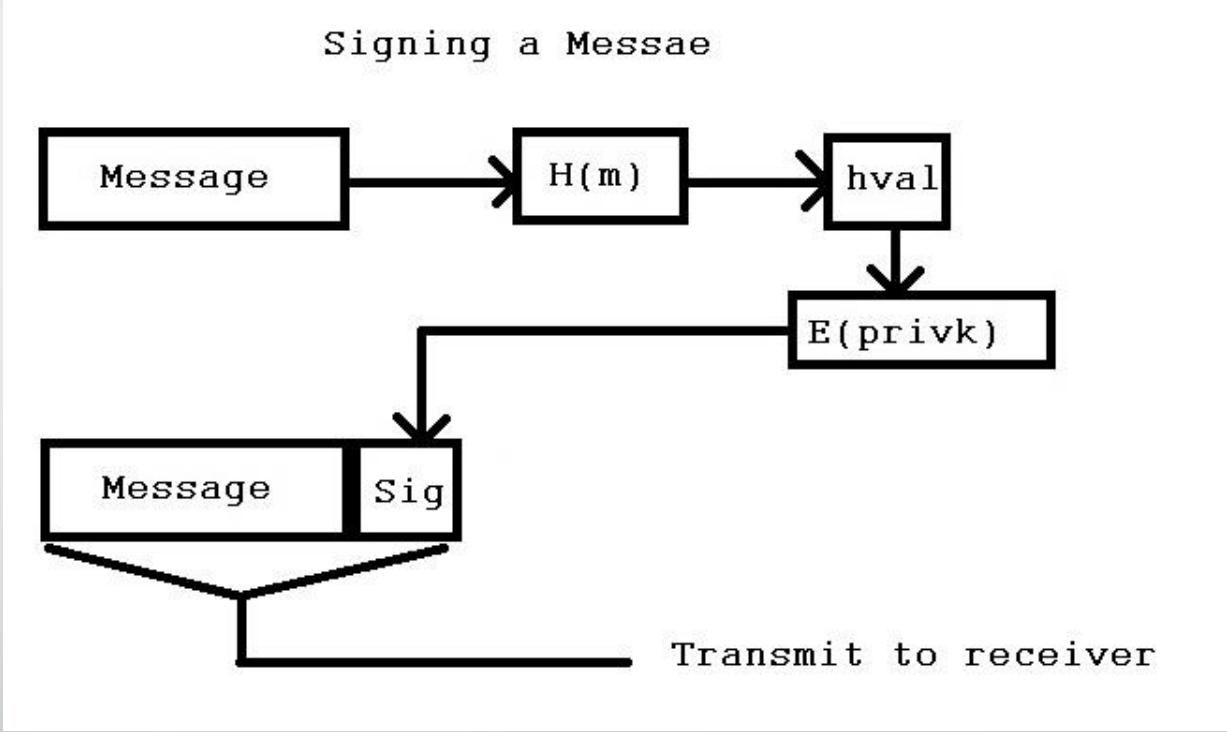
Digital Signatures

- Digital Signatures provide both **integrity** and **authentication**
- Signatures are based on cryptographic hashes
- **Cryptographic hashes**
  - Support **integrity** by indicating if the attached data has been altered or corrupted
  - They are mathematical 'summaries' of data in a file or message
  - It is [very] hard to alter text in a way that will produce the same hash value as the original

Digital Signatures

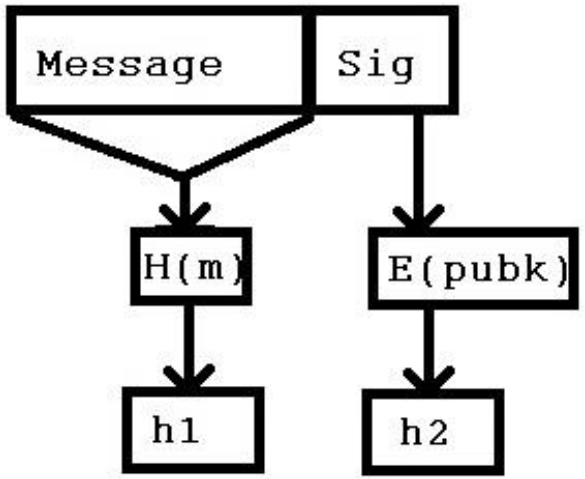
- Example Procedure:
  - Run a hash on the message (i.e. SHA256)
  - Take the hash value (128, 192, or 256 bits) and encrypt with **Private key of the signer**
    - This means only the owner of the private key could have produced the signature
    - This also means ANYONE can decrypt the encrypted hash with the public key of the signer
    - This is how digital signatures provide **authentication**

Digital Signatures



Digital Signatures

Digital Signature Verification



h1=recvr calculated hash
h2=hash recovered from signature

signature verifies if h1==h2

Intermission : What about Quantum Computing?

Caveat: I know little about quantum computing. These slides are based on [light] reading.

- Development of practical quantum computers threatens security of nearly all existing cryptographic algorithms
  - **Shor's algorithm** – Speeds factorization of large composite numbers and solving discrete logarithm problem.
  - **Grover's algorithm** – Greatly reduces cost of brute forcing symmetric encryption and hash algorithms

Intermission : What about Quantum Computing?

- Solutions:
  - Developing entirely new classes of difficult math problems
  - Currently two major classes of algorithms
    - Lattice based cryptography
    - Hash based signatures
  - NIST is currently publishing approved quantum resistant crypto and signature algorithms [FIPS 203, FIPS 204, FIPS 205]

Digital Certificates

- Digital signatures provide **integrity** and **authentication** for messages
- Still have not solved the problem of public key ownership assurance
- Digital Certificates solve this

Digital Certificate

- When digital certificates first came into use, you heard the word 'trust' from a lot of marketers. Very misleading messages were sent to potential customers.
  - A validated certificate does NOT mean you **trust the owner** of the private key
  - Rather, it means you trust the BINDING between the entity's name and the contained public key…as long as:
    - The Certificate Authority was competent in securing their signing key
    - The CA followed reasonable security procedures in creating the certificate
    - The user's private key has been kept totally secure (passphrase not shared, no malware or key logger on user's computer, the certificate has not been revoked by the CA, etc)

Digital Certificates

- Digital Certificates
  - Collections of data about an **entity** (person, company, device, website, etc)
  - Data includes a name or label for the entity and a public key
  - Signed with a private key of a different party (Certificate Authority or CA)
  - Most widely used certificates PKIX (Pubic Key Infrastructure X-509)

Digital Certificates - Structure

- Certificates contain data to be signed and a digital signature
- Data to be signed (tbs) includes:
  - Certificate version number (usually 3)
  - Issuer Name – The Certificate Authority issuing the certificate
  - Serial Number – The Serial Number of the Cert
  - Signature – The algorithm used to sign the cert (i.e. Sha256-with-RSA)
  - Subject Name – The entity to whom the cert will belong
  - Validity – Validity dates (Not Before, Not After)
  - Extensions – Extra Information including key usage

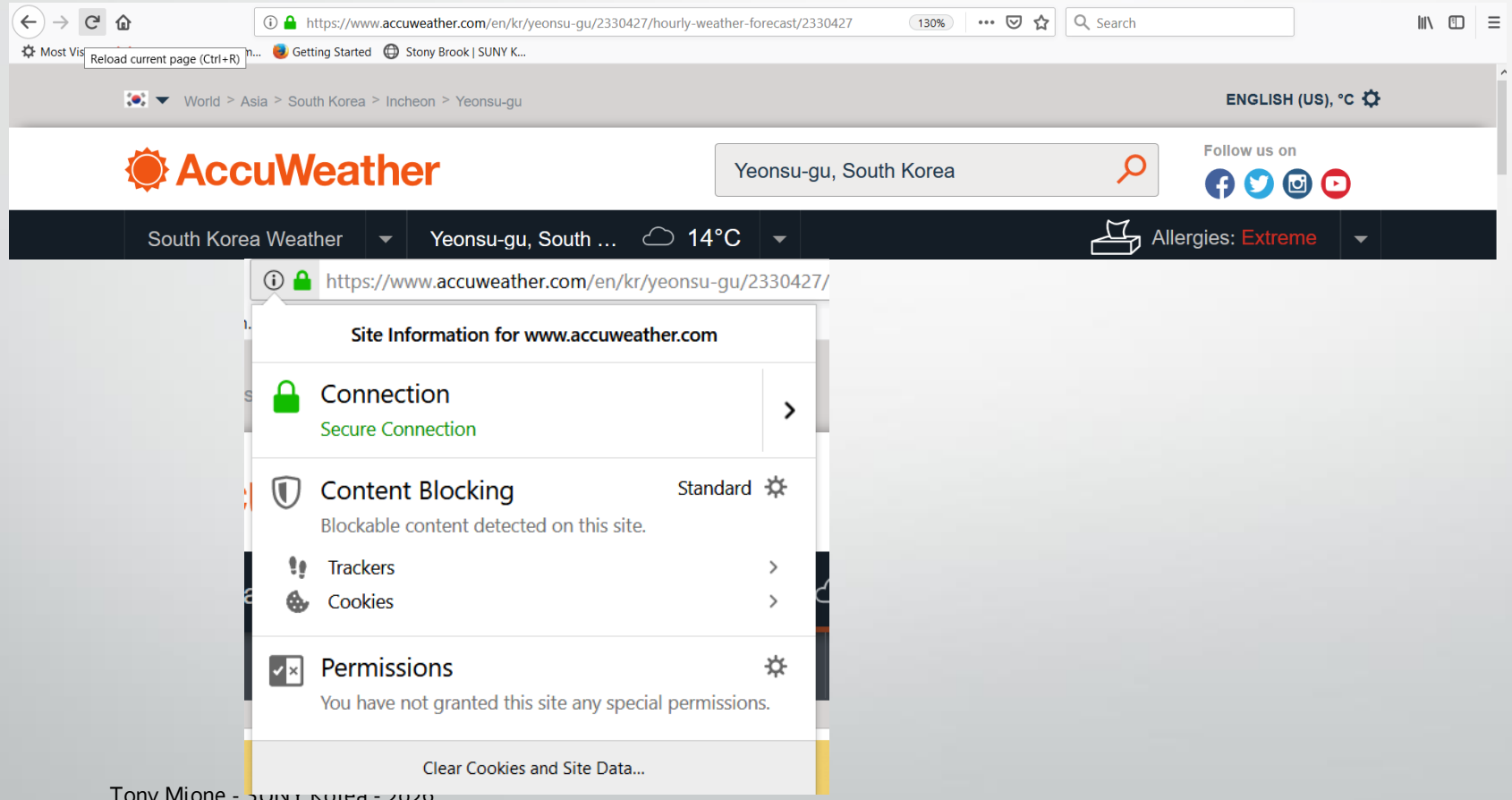
Digital Certificates - Structure

- Issuer and Subject Names are in the form of a 'Distinguished Name'
  - Contains components describing the entity, organization, and country of the entity
  - The name uniquely identifies (worldwide) an entity (person, company, server, computer, etc)
  - Ex: CN=Antonino N. Mione, OU=Dept of Computer Science, O=SUNY Korea, C=South Korea
- Signature block includes:
  - Algorithm Used
  - Signature over tbs section

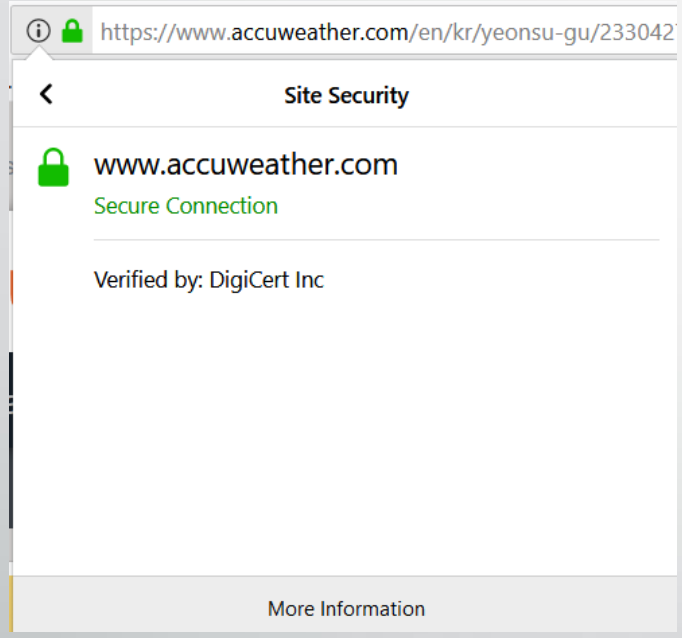
Digital Certificates
Example Use – SSL/TLS

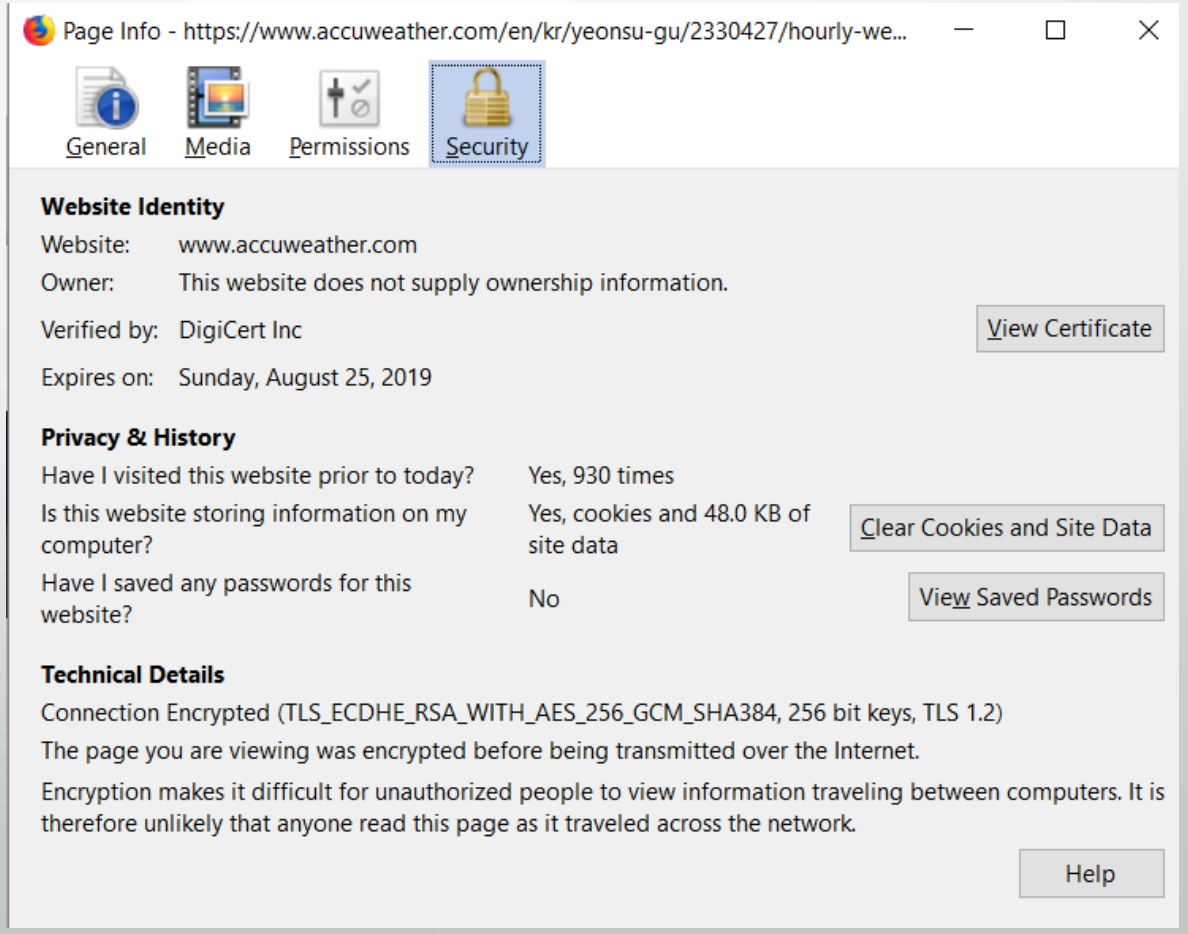
- Secure Socket Layer /Transport Layer Security
  - Sets up encrypted communication links, usually for web traffic (https:)
    - Uses public keys in server certificate to:
      - Transfer a random 'session key' sent from client to server
      - Or uses its private key and client public key to generate a session key with a 'key agreement' protocol [i.e. Diffie-Hellman]
    - Server's public key is also used to authenticate the server to a client
      - If certificate chain validation fails for the server, then user gets a warning dialog

Digital Certificates
Example Use: SSL/TLS

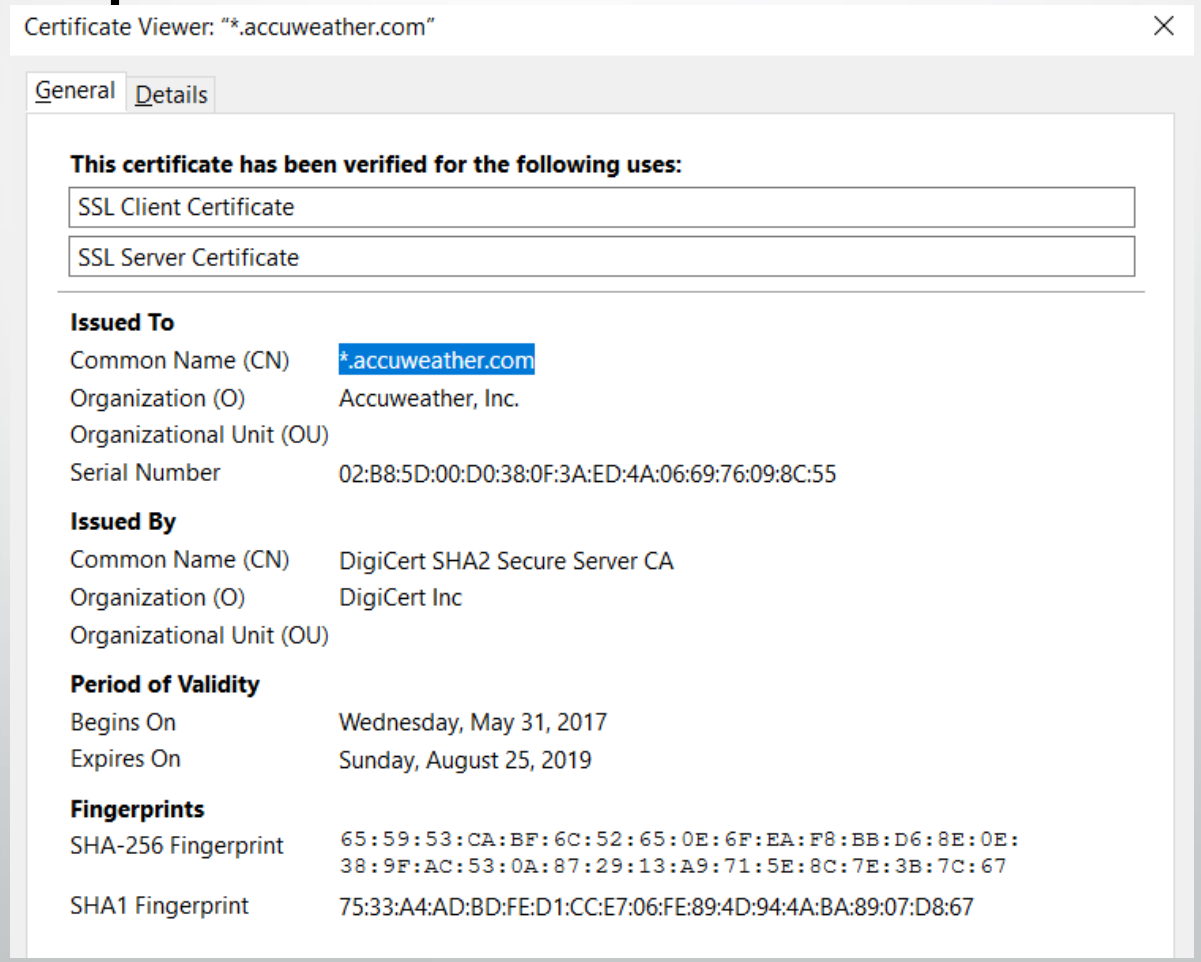


Digital Certificates
Example Use: SSL/TLS





Digital Certificates
Example Use: SSL/TLS



Applications

- Passkeys
- Secret Splitting/Sharing
- Blockchain

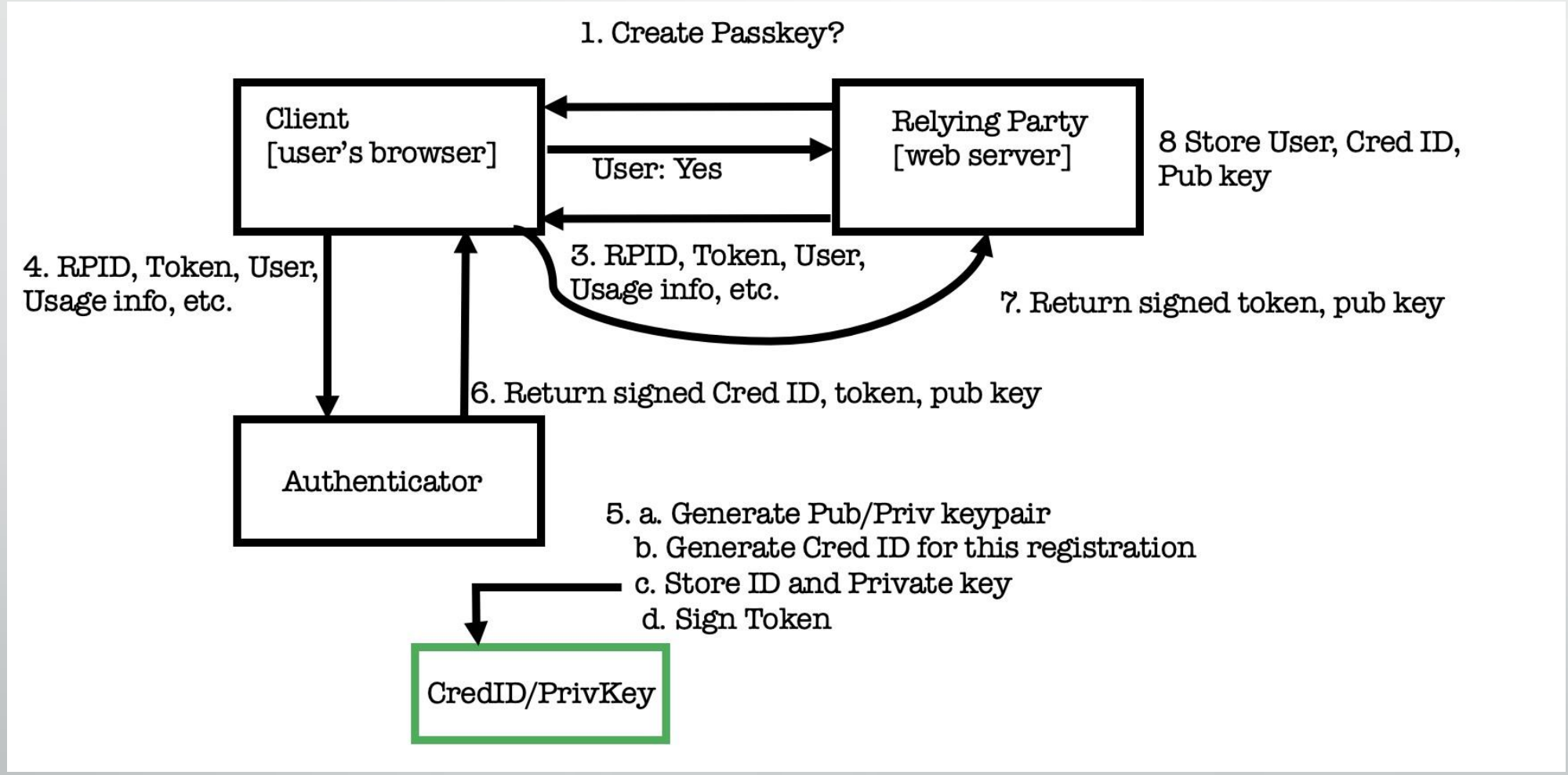
Passkeys

- Passkeys
  - Eliminate [almost] the need for passwords (which have many security issues)
  - Use Public Key cryptography and Digital signatures
  - Include the concept of a 'token' which holds information used as part of the authentication
  - Are difficult to 'Phish'
  - Minor drawbacks:
    - Usually are not 'mobile'. Tied to one device/browser/client, etc.
    - If lost, need separate recorvery mechanism (password, biometric, etc)

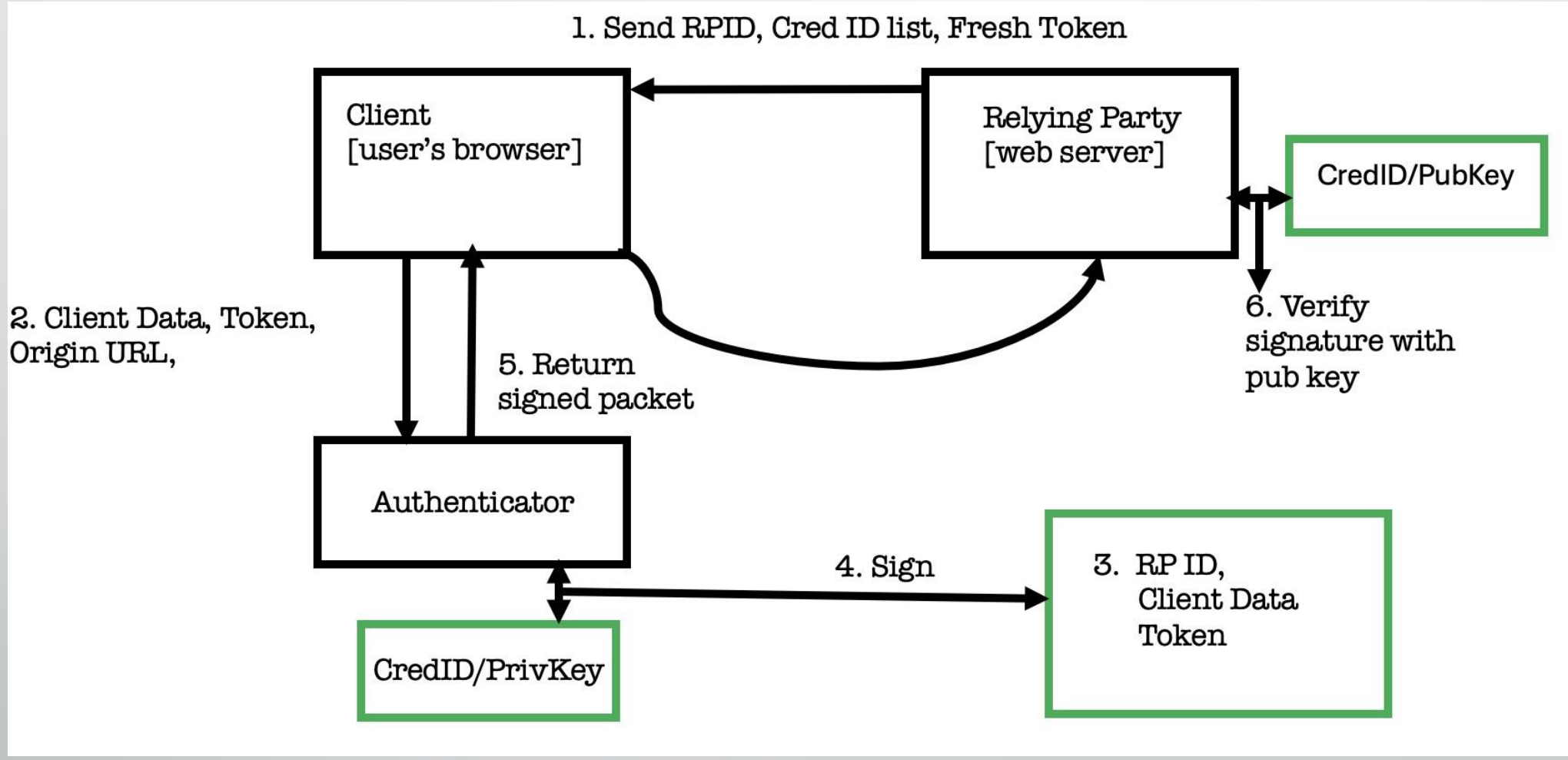
Passkeys

- Two operations
  - Generation or registration of a Passkey
  - Authentication using a Passkey

Passkey Creation/Registration



Passkey Usage for Session Authentication



Advanced Uses of Crypto
Secret Sharing

- Problem: You want to secure an information resource but the info is too valuable to entrust to an individual
  - A competitor may offer large amounts of cash for someone with access to extract specific information and hand it over.
  - An enemy of the country may offer INCREDIBLY large amounts of money for someone to extract classified (TS/SCI) information and hand it over [This does happen on occasion, anyway]
- Solution: [Short answer] – Assure that more than 1 person must collude to access the data (Secret Sharing)

Secret Sharing : Terms

- Terminology
  - **Participants** – An entity that recieves and controls a 'share' of access information
  - **Share** – Part of a secret
  - **Dealer** – A [trusted] system that distributes shares to participants
  - **Cheater** – A participant (or outsider) who tries to gain knowledge of one or more shares to which they are not entitled.
  - **Recovery** – Combining sufficient shares to recover a 'secret' (usually a key to decrypt valuable data).

Secret Sharing : Basic Approachs

- **M of M** – All distributed shares are required to reconstruct secret. Any less provides NO additional information about the value of the secret
- **M of N Threshold** – Of some number of shares distributed (N), at least M shares are required to recover the secret (M-1 shares provides NO additional information about the secret).
- **General Access Scheme** – Certain specific, predefined arbitrary combinations of shares are required. No other combinations can retrieve the secret.

Secret Sharing : Concerns/Issues

- **Security**: Can we know that insufficient shares cannot access the secret and that no information about the secret is given?
- **Verifyability**: How does a participant receiving a share from the dealer know that the share is valid?
- **Security/Leaking of shares** – What happens if shares are leaked?
- **Cheating**: Can we detect if a valid participant is cheating by acquiring unauthorized shares?
- **Dealer Trust**: Can we trust the dealer? Can the dealer cheat?
- **Size of Shares**: Depending on the amount of security needed, the share size may grow very large.

Secret Sharing : Approaches to Solve Issues

- **Cheat Detection** : Schemes may include a mechanism to detect cheaters
- **Proactive Secret Sharing** : Schemes like this 'renew' shares or provide fresh shares and invalidate older shares. Workaround for share leakage.
- **Verifiable Secret Sharing** : The scheme includes operations that prove to a participant their share is valid.

Some Published Schemes

- **Simple XOR (n of n threshold)**
- **Simple Geometric scheme based on lines (2 of n threshold) [Blakley]**
- Shamir's Secret Sharing [Based on Legrange Interpolation of polynomials]
  - m of n threshold scheme
- **Online Secret Sharing – General Access Sctructures**

XOR

- Give out shares (binary values of a particular size) to n participants
- Secret is the XOR of all shares
  - Advantages
    - Simple
    - Regardless of the number of participants, the size of each share is no larger than the size of the secret.

XOR Example

Participant	Shares [Calculating secret w n shares]	Calculating secret with n-1 shares
Tom	10110101	10110101
Jill	11010111	11010111
Harry	00110011	
Secret (decryption key)	01010001	01100010 (??)

Participants: Tom, Jill, Harry – Each get a random binary value (final value is calculated from first n-1 participants and the secret)

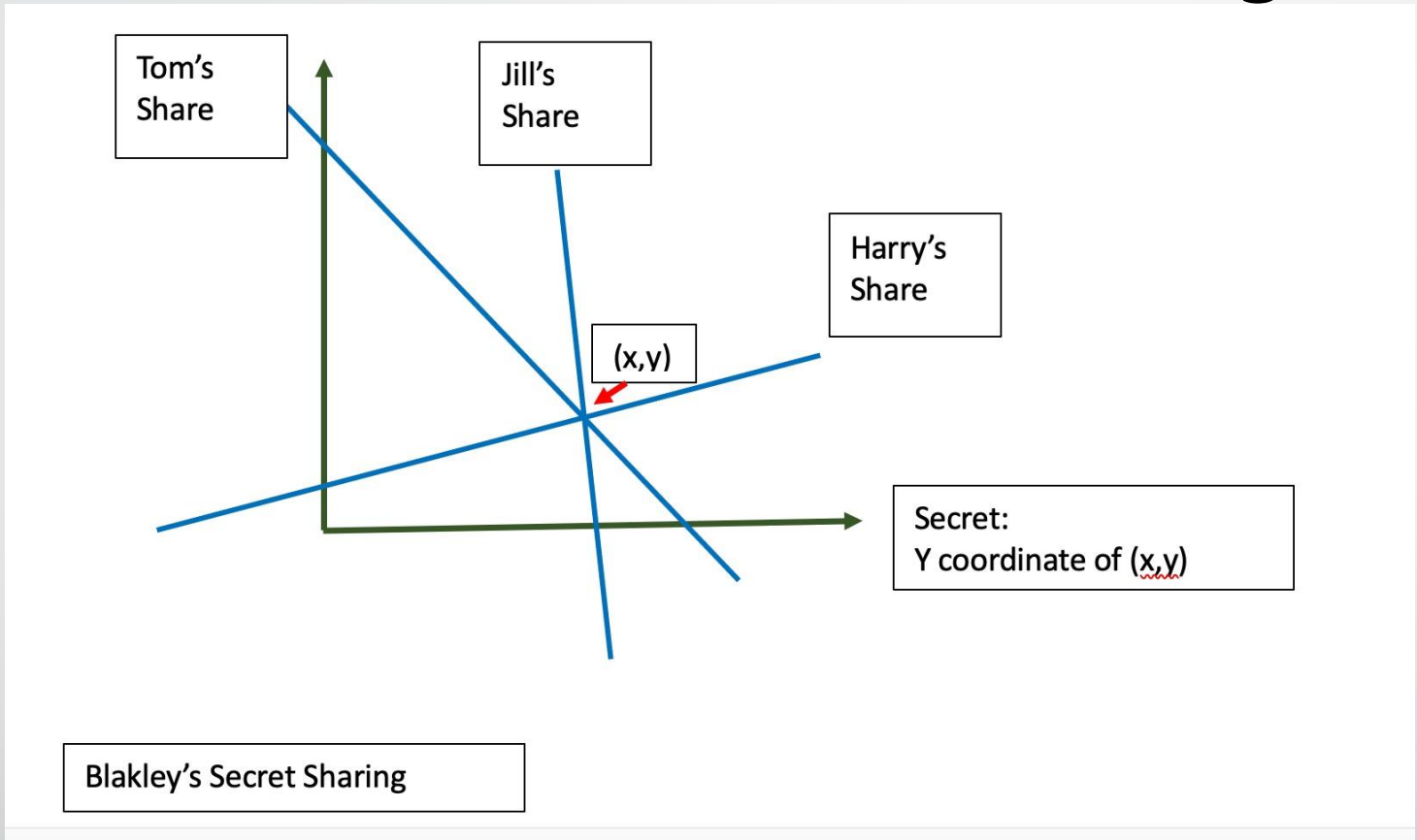
Need all n to reconstruct secret

N-1 shares gives NO information about the final value of the secret.

Geometric Secret Sharing [Blakly]

- Give out equations for lines
- All lines intersect at 1 x,y coordinate
- Need two of the n shares to reconstruct secret (secret is either coordinate or both)

Geometric Secret Sharing



Note: Can make this a 3 of n threshold scheme using planes in a 3D coordinate system (4 of n in a 4D geometry system, etc).

Online Secret Sharing : Cachin

- Published in 1995
- Implements a 'General Access Structure'
  - More flexible than m of n threshold scheme

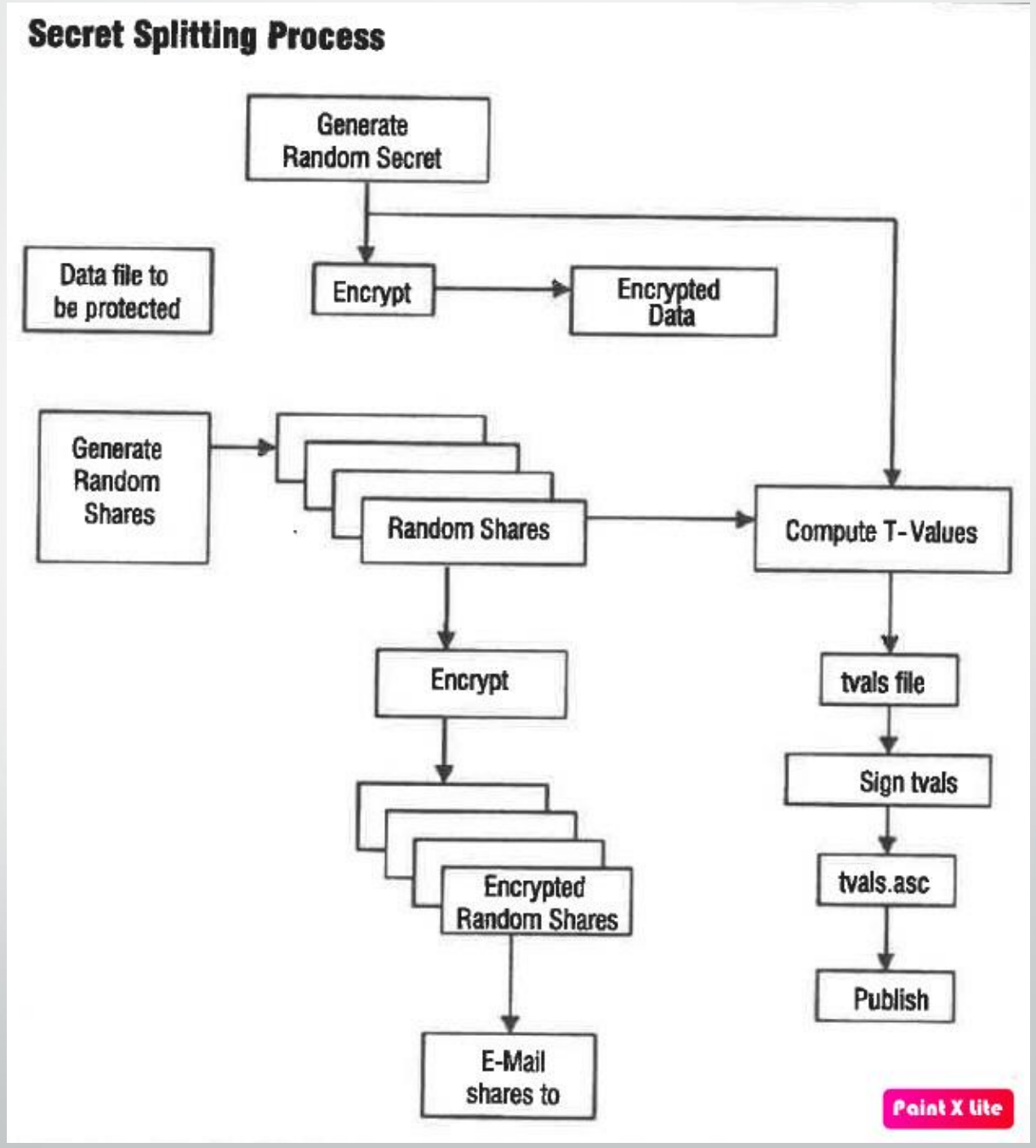
Online Secret Sharing : Cachin

- Procedure:
  - Register participants
  - Generate 1 share (randomly) for each participant and distribute [securely]
  - Decide on Access Structure (exaclty which participant groups can reconstruct secret)
  - Key is generated randomly and data to be secured is encrypted
  - Tau values are:
    - For each group of 'authorized' participants:
      - Hash the sum (or concatenation) of their shares
      - Subtract that hash value from the key => This is the Tau value for that group
  - Generate Tau values and publish list on some bulletin board
  - Discard the decryption key securely

Online Secret Sharing : Cachin

- List of values include the names of the required participants
- Since an attacker needs to know all the shares of at least 1 group, it is difficult to recover the secret.
- The Tau values (since the result from a calculation on a cryptographic hash) are not useful unless all of a specific set of shares are avaialble.

Splitting Process



Example : Access Structure and Tau Values

Participants:

- Joe
- Phil
- Corrine
- Bethany
- Alan
- Melissa

Access Structure
Joe, Phil
Phil, Corrine, Bethany
Joe, Alan, Melissa
Alan, Corrine

Compute
Tau[Joe,Phil] = Key - Hash(JoeShare + PhilShare)
Tau[Phil, Corrine, Bethany] = Key - Hash(Phil, Corrine, Bethany)
Tau[Joe, Alan, Melissa] = Key - Hash(Joe, Alan, Melissa)
Tau[Alan, Corrine] = Key - Hash(Alan, Corrine)

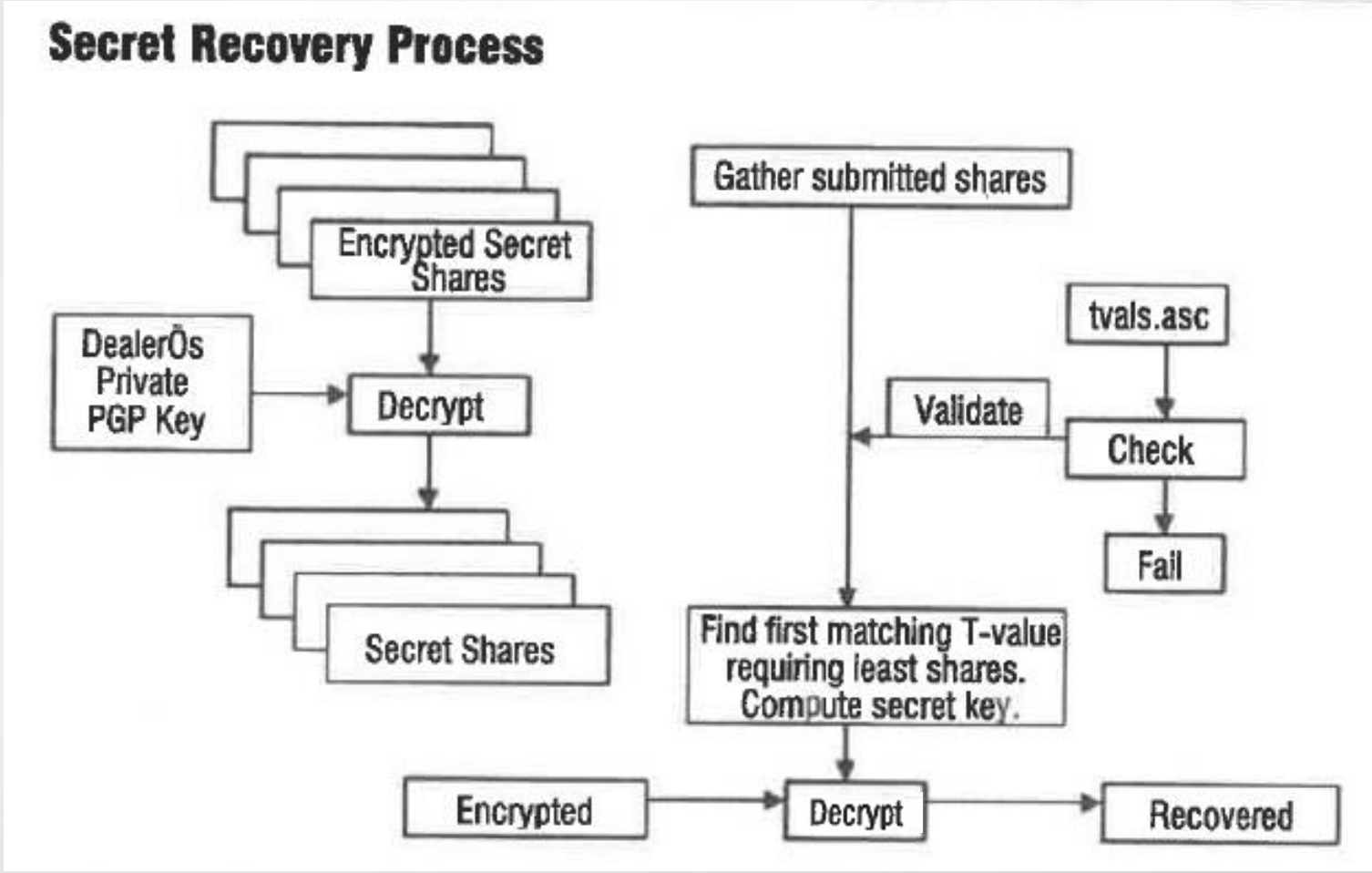
(Securely delete Key)

Bulletin Board - Secret <SomeID>		
Group		Tau Value
Tau[Joe,Phil]		0xA9052433C804…..
Tau[Phil, Corrine, Bethany]		0x217CF815BB95…..
Tau[Joe, Alan, Melissa]		0x567AB93FE912……
Tau[Alan, Corrine]		0x99A7CCD345A2…..
Dealer Signature	SHA256-With_RSA	0xFF90125DBEE29AB71…..

Online Secret Sharing : Cachin

- Recovery:
  - All members of a particular group authorized to recover the secret, combine shares and generate the hash.
  - The hash is added to that group's Tau value to recover the key
  - The key is used to decrypt the protected data.

Recovery Process



Secret Sharing

- Solution: [Longer answer] – Numerous problems to solve:
  - How to assure participants can verify their shares are valid [Verifiable SS]
  - How to assure participants don't cheat (have an easy means to discover other shares)
  - How to protect against share 'leakage' [Proactive SS]
- A number of research efforts have been on going to solve these and other problems/security issues.

Blockchain Technologies

- What is Blockchain?
- Features
- [Potential] Uses
- Incorporated Technologies/Concepts
- Basic Architecture

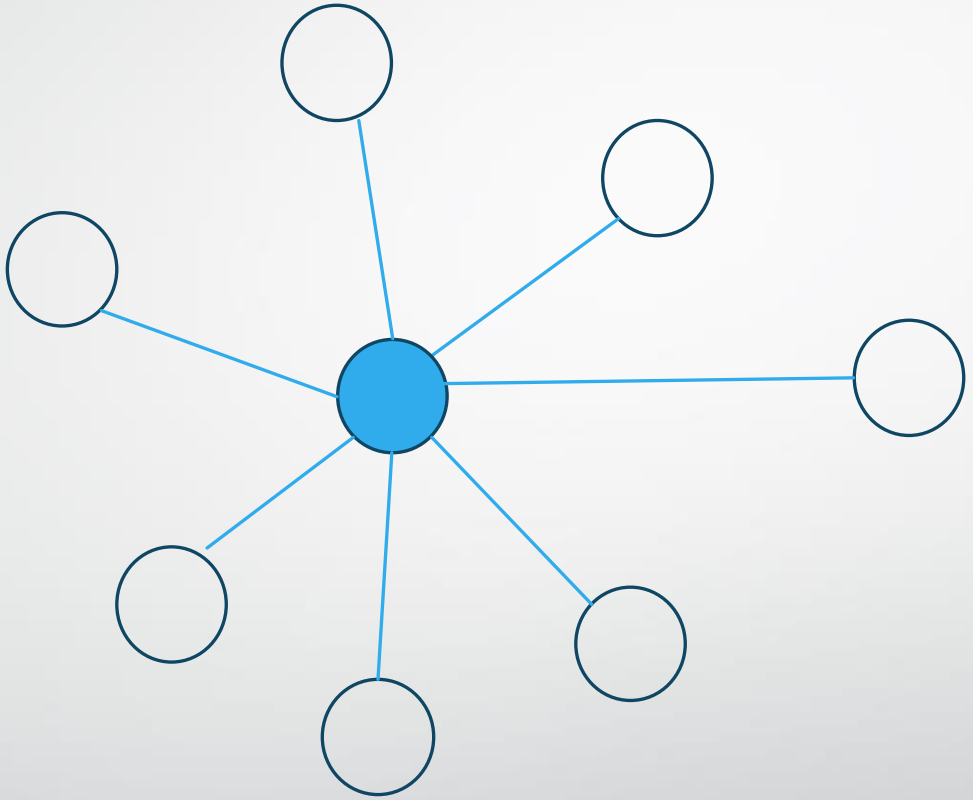
What Is Blockchain?

- A large database containing a linked list or chain of blocks with immutable data
  - => This is a form of 'Decentralized Ledger'

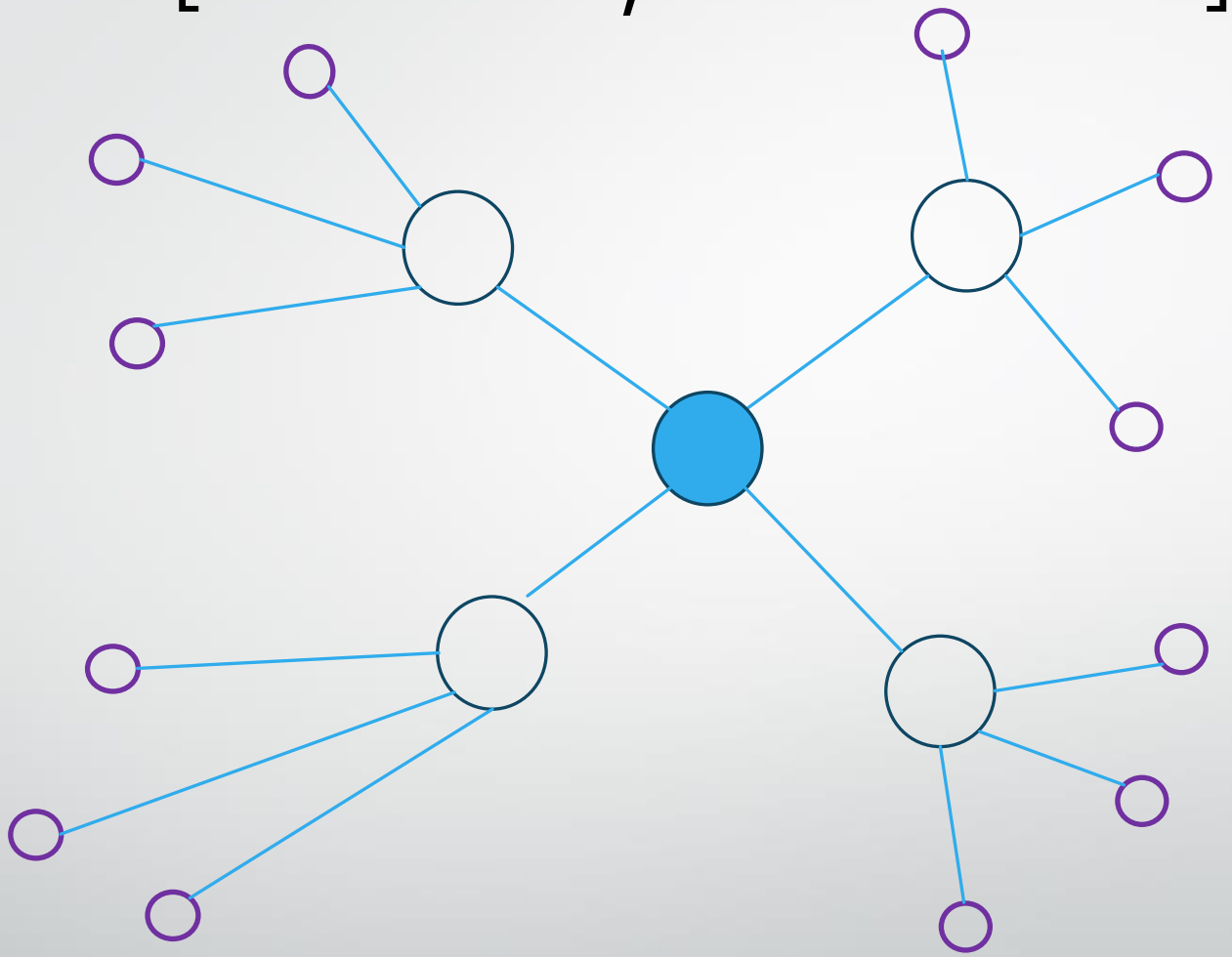
Blockchain Features

- **Decentralized** – Database maintenance and control of adding data is not tied to a central organization.
- **Transparency** – The process is clearly defined
- **Security** – Cryptographic techniques are used to secure blocks and other components of the chain.

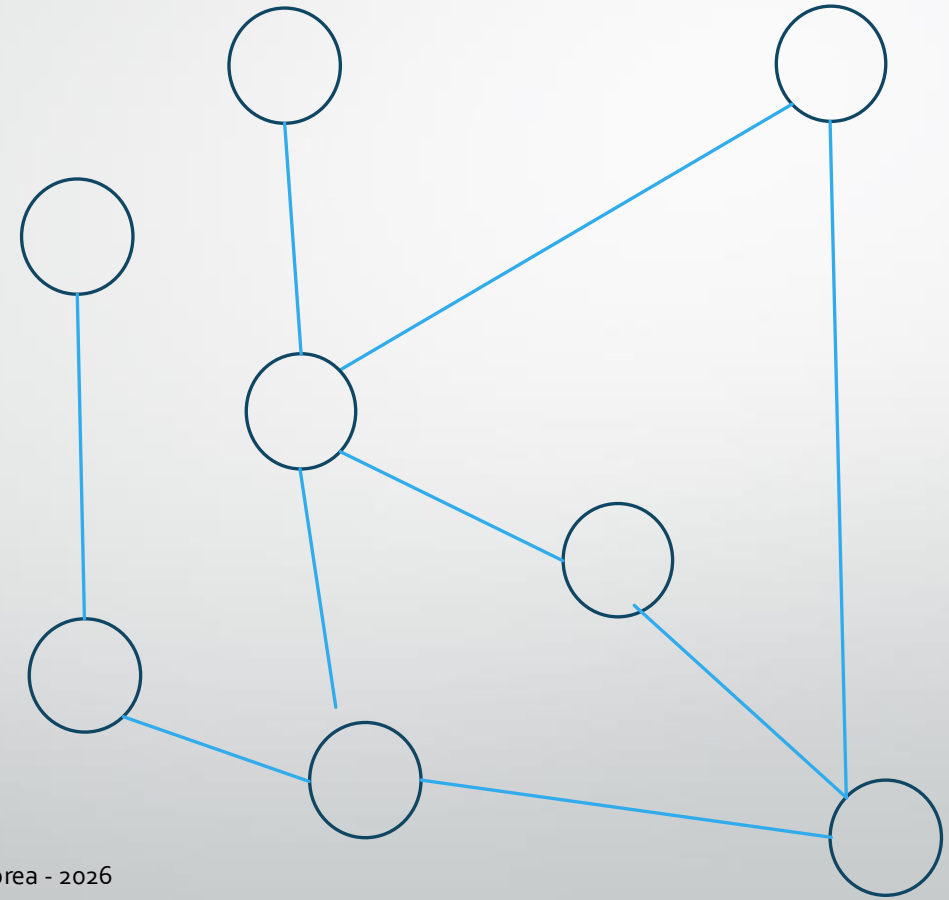
Financial and DB Systems Architecture – [Traditional, Centralized]



Financial Systems and DB Architecture - [Blockchain, Decentralized]



Financial and DB Architecture – [Blockchain, Distributed]



Typical Blockchain Structure

- Blockchain networks are Decentralized and/or Distributed (usually a hybrid these)
  - Distribution focuses on structure [physical and logical organization of nodes]
  - Decentralization focuses on control [How decisions are made and who has authority to do what actions]
- A 'consensus' mechanism is used to determine if blocks are added.
- A 'Difficulty factor' is added to slow down block creation for miners [Proof of Work]

Blockchain Participants

- Public at large generates transactions for the ledger
- Miners –
  - Can be anyone or almost anyone
  - Task, mine transactions and build valid blocks to be added to the chain (ledger)
- Validators – Usually a selected group of entities
- Nodes –
  - A node holds a complete copy of the blockchain
  - Several nodes in the system. All validate and vote on adding new blocks ['concensus mechanism']

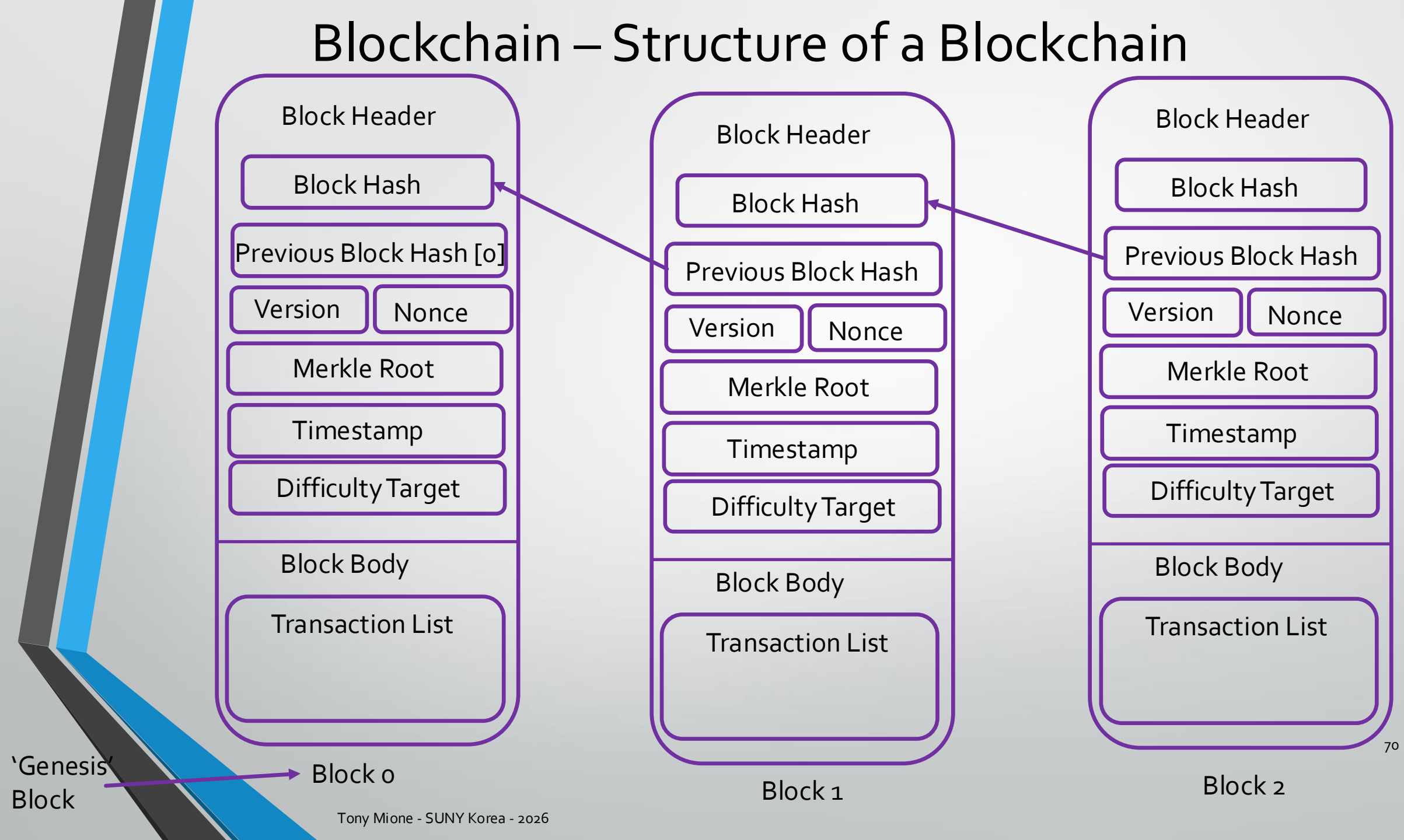
Blockchain Operations

- Collect transactions and build a block
- Submit block to one of the member nodes
- Nodes validate blocks
- Nodes vote on blocks (>50% => Add block to the chain)
- Some Blockchain networks have financial incentives to 'mine' blocks
- Chains may develop a fork with 2 succeeding chains
  - Longer chain is considred valid and other chains are 'pruned'.
  - Transactions return to a pool for re-mining

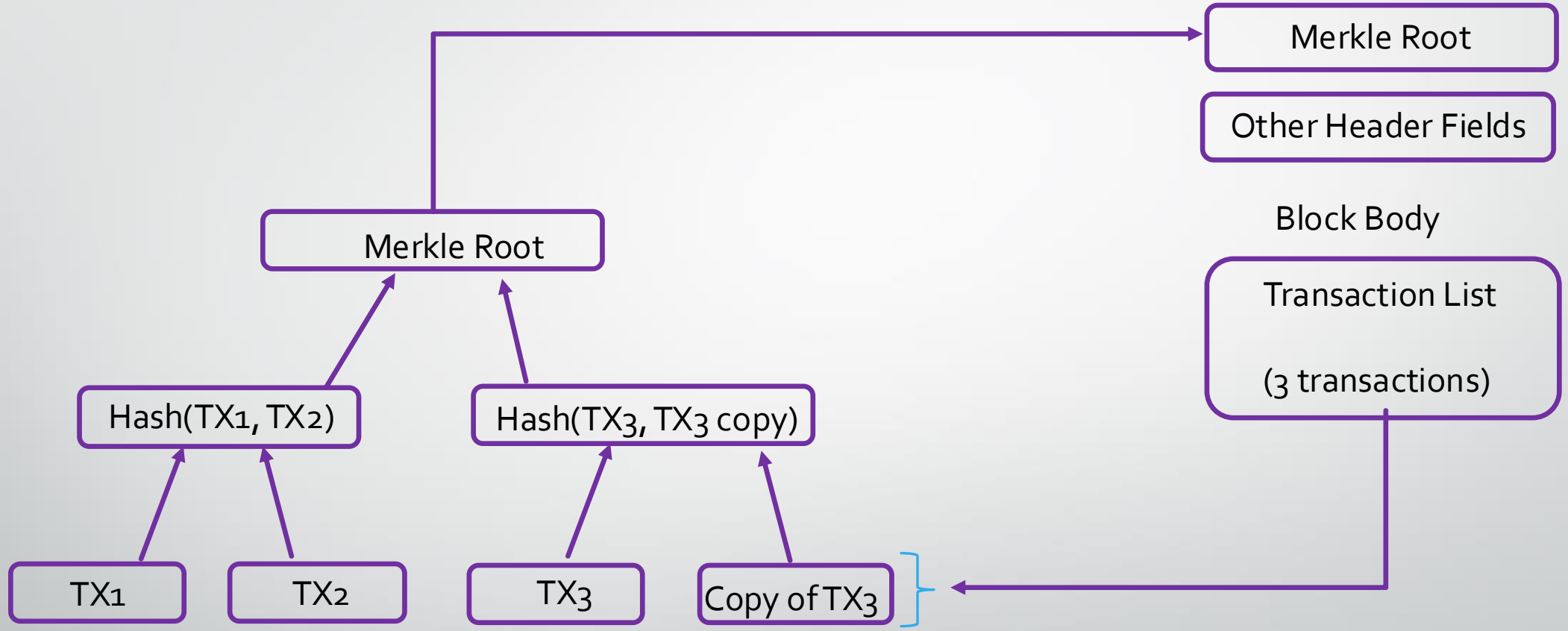
Blockchain Security

- Blocks are constructed and validated with hashes. Chains are formed by including hash of previous block in chain
- Any alterations to a block are not possible due to invalidating the hash. If an entity could rewrite a block including the hash, it would have to rework every succeeding block in the chain
- Generating blocks are time consuming due to the 'Proof of Work' requirement (could take almost 10 minutes to generate a block)

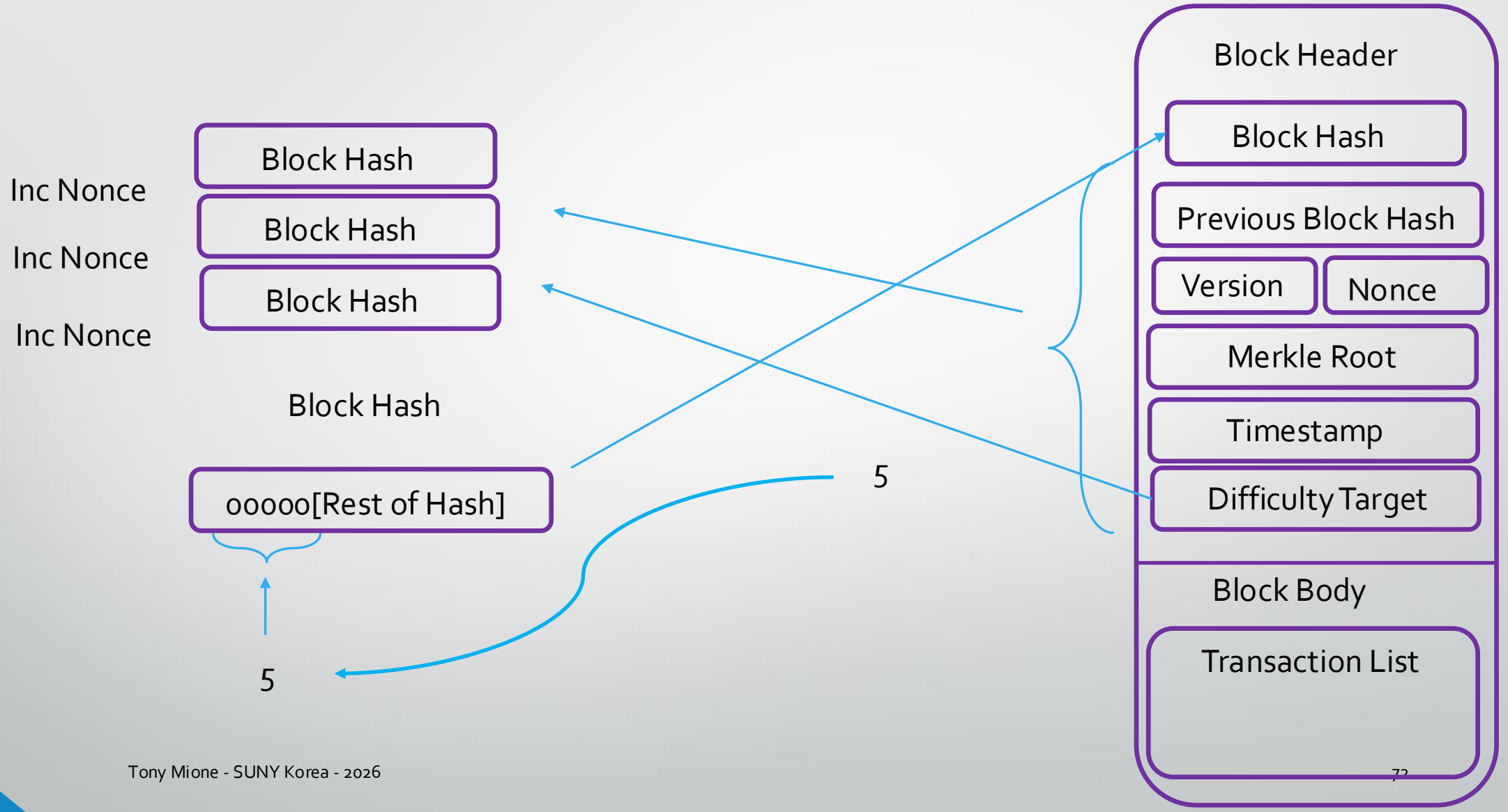
Blockchain – Structure of a Blockchain



Merkle Root Calculation



Blockchain – PoW (Proof of Work)



Questions

- Thanks for your attention!