Overview of Cryptography, Public Key Infrastructures, and Related Technologies

Topics

- Basics
- Security with Cryptography
- Types of Cryptography
  - Stream vs Block
  - Symmetric vs. Asymmetric
- Digital Signatures
- Certificates – Public Key Infrastructure
  - Structure of Certificates
  - Usage of Certificates
- Modern Applications
  - Secret Splitting
  - Blockchain

Basics

- **Encryption**
  - Scrambling data to provide privacy
  - A **key** is used to scramble data to be protected
- **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

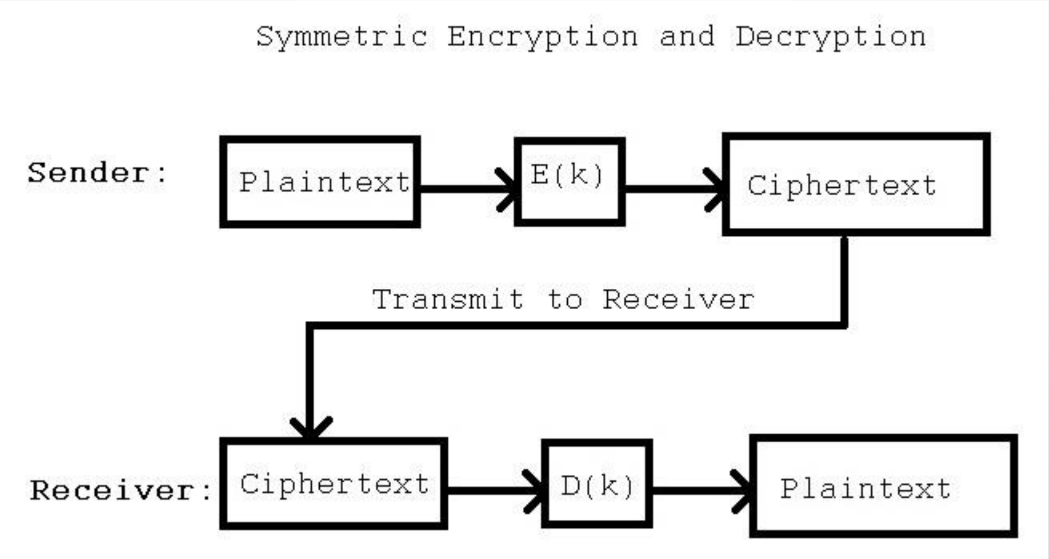
- Encrypting data provides **privacy** keeping data secure from being ‘snooped’
- Stream Ciphers
  - Encrypt 1 bit at a time
    - a bit is a ‘1’ or ‘0’
    - Current characters on a computer take 8 bits or 16 bits to encode
  - Algorithm produces a ‘stream’ of bits based on the **Key**
  - Uses Exclusive-Or 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) 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: DES, CAST, IDEA, AES, Blowfish, RC6, many others

Type of Cryptography
Symmetric vs. Asymmetric

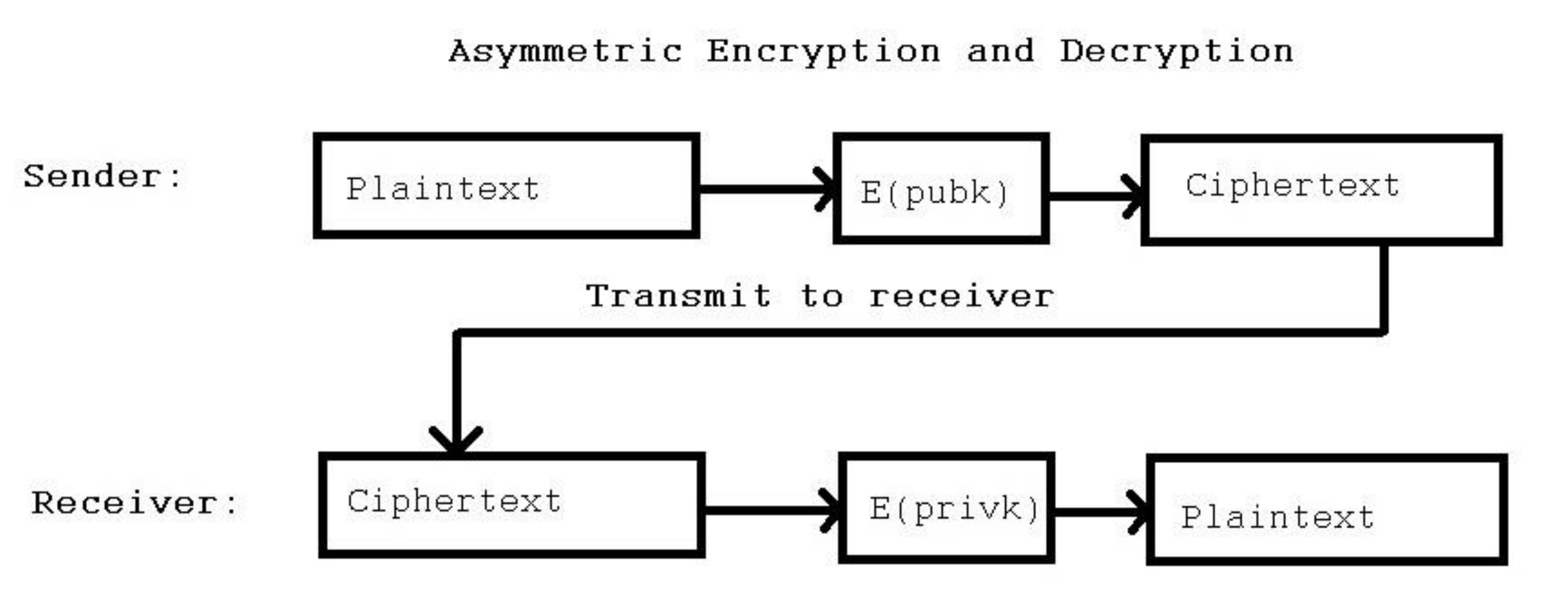
- Symmetric cryptograph requires sender and receiver to share the same key
  - Problem: How do we communicate the shared secret key without someone intercepting it?



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



Asymmetric Encryption and Decryption

Sender: Plaintext → E(pubk) → Ciphertext

Transmit to receiver

Receiver: Ciphertext → E(privk) → Plaintext

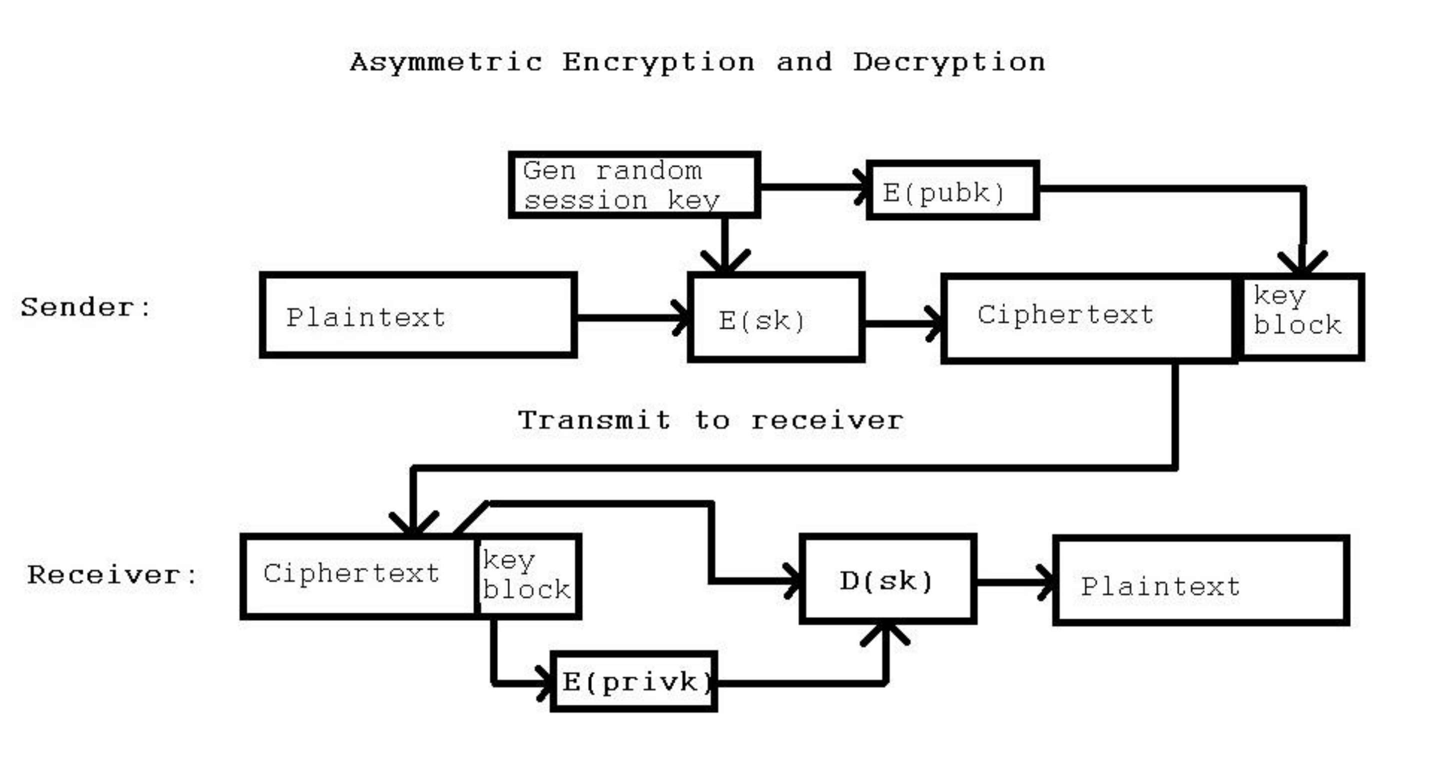
Type of Cryptography
Symmetric vs. Asymmetric

- 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

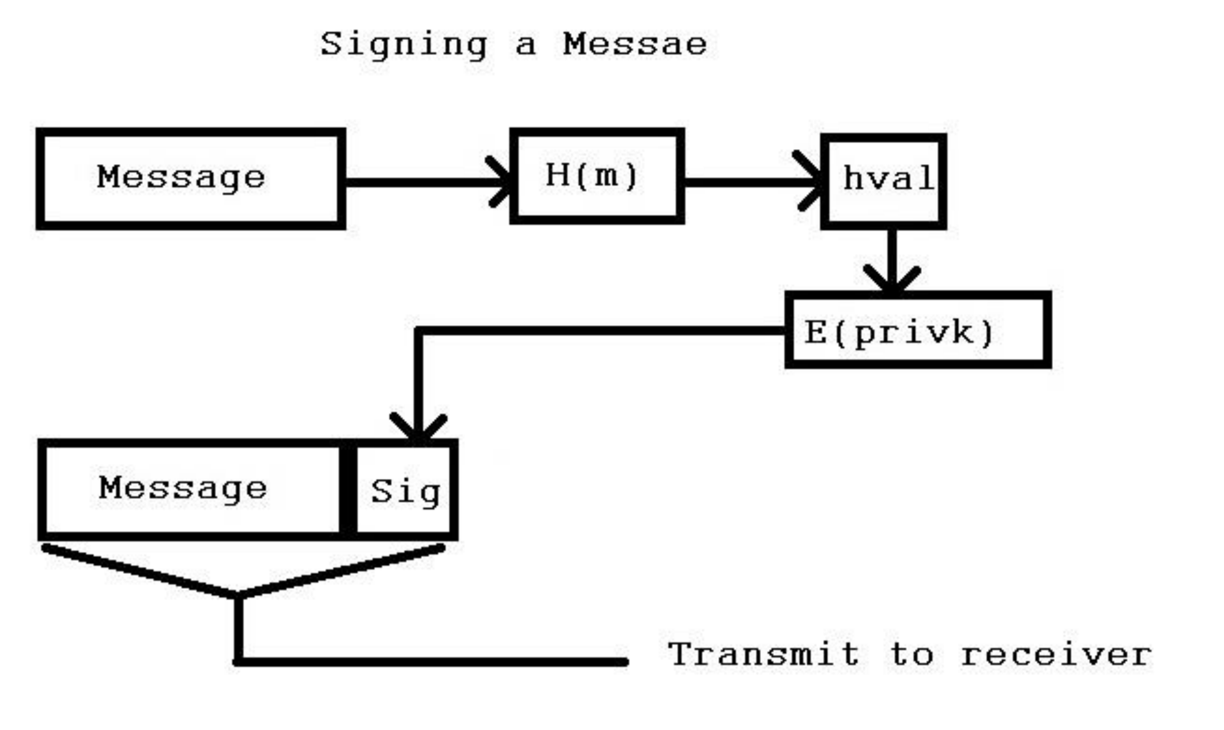
Digital Signatures

- 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

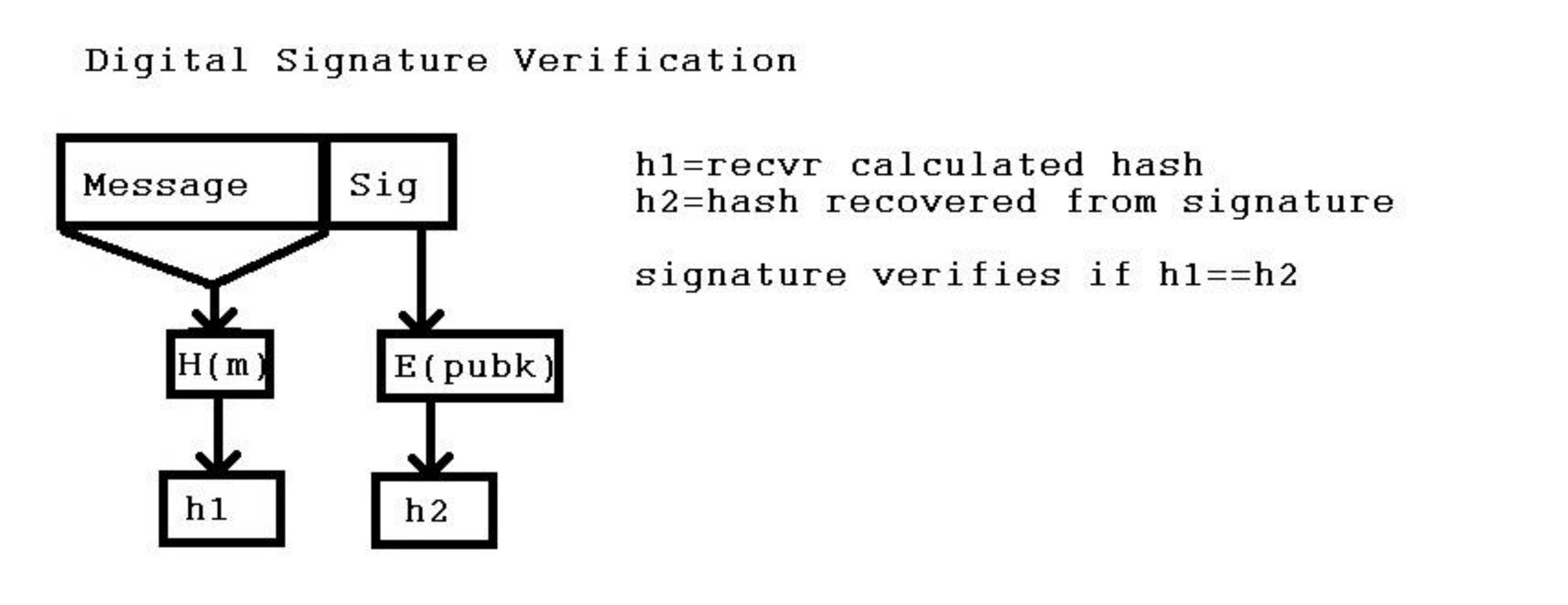
Digital Signatures

- Procedure:
  - Run a hash on the message
  - Take the hash value (128-256 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

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

- A validated certificate does NOT mean you trust the owner of the private key
- 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, certificate has not been revoked, 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 – Generation Process

- RA (Registration Authority):
  - Helps CA gather user information and public key
  - vettes information
- CA (Certificate Authority):
  - creates and signs certificate using its (very secure) private key
  - Publishes certificate in a repository
  - Periodically creates, signs and publishes Certificate Revocation Lists (CRLs)

Digital Certificates - Requirements

- Issuer Name + Serial Number must be Globally Unique and never reused
- Extension fields marked as *critical* must be followed or certificate must be rejected

Digital Certificates – Certificate Chain Validation

- Validating Certificates involve a number of steps: Given a collection of certificates (or a chain from a ‘Trust Anchor’ down to the cert to be validated:
  - Perform validation steps on the Trust Anchor [Certificate validation steps discussed soon]
    - The ‘Trust Anchor’ is a self-signed certificate of a known registered Certificate Authority
  - Perform validation steps on each certificate in the chain (in order) until you reach the certificate being validated for the current operation

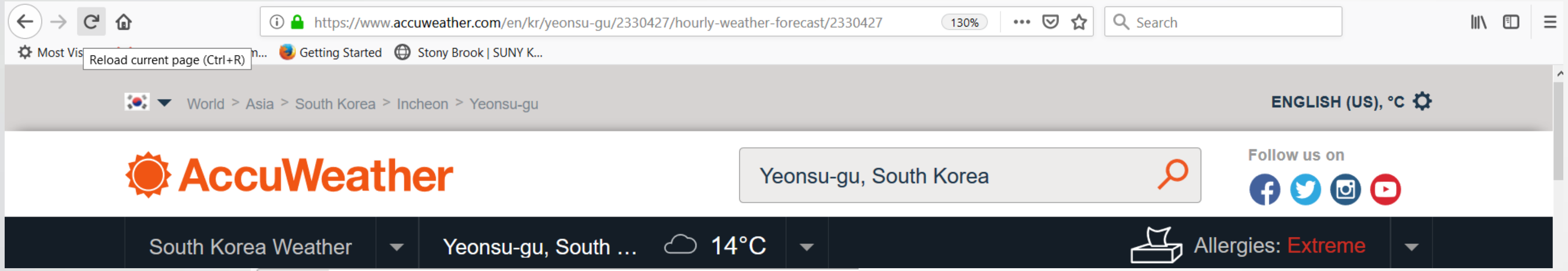
Digital Certificates – Certificate Validation

- Several checks are performed to validate a certificate:
    - Verify the signature using the public key of the certificate Issuer (must be a CA), the public key algorithm and public key parameters indicated in the certificate
    - Verify the current date is between the not-before and not-after dates encoded in the certificate
    - Verify the certificate is not revoked (not on a Certificate Revocation List published by the CA)
    - Verify the Subject DN is within 'domain subtrees' allowed by certificate policies of the CAs above this cert
    - Verify the Subject DN is not within excluded 'domain subtrees' given in certificate policies
    - Other policy checks
- Detailed in: https://www.rfc-editor.org/rfc/pdfrfc/rfc5280.txt.pdf [RFC5280] (good for Masochists or those needing a good sleep aid!)

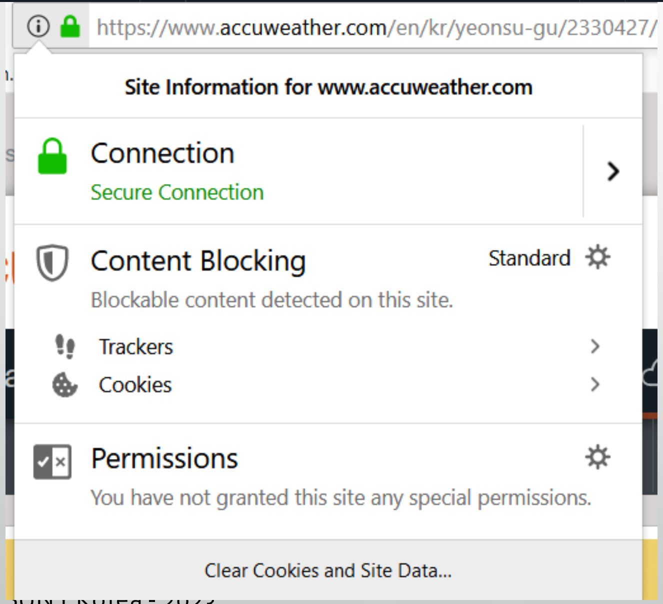
Digital Certificates
Example Use – SSL/TLS

- 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' to client
      - 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



https://www.accuweather.com/en/kr/yeonsu-gu/2330427/hourly-weather-forecast/2330427



https://www.accuweather.com/en/kr/yeonsu-gu/2330427/

Site Information for www.accuweather.com

Connection
Secure Connection

Content Blocking — Standard
Blockable content detected on this site.

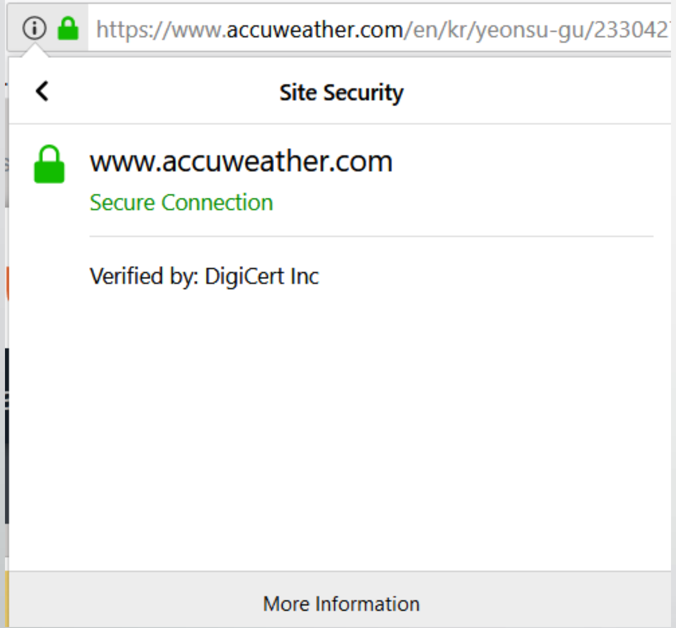
Trackers

Cookies

Permissions
You have not granted this site any special permissions.

Clear Cookies and Site Data...

Digital Certificates
Example Use: SSL/TLS

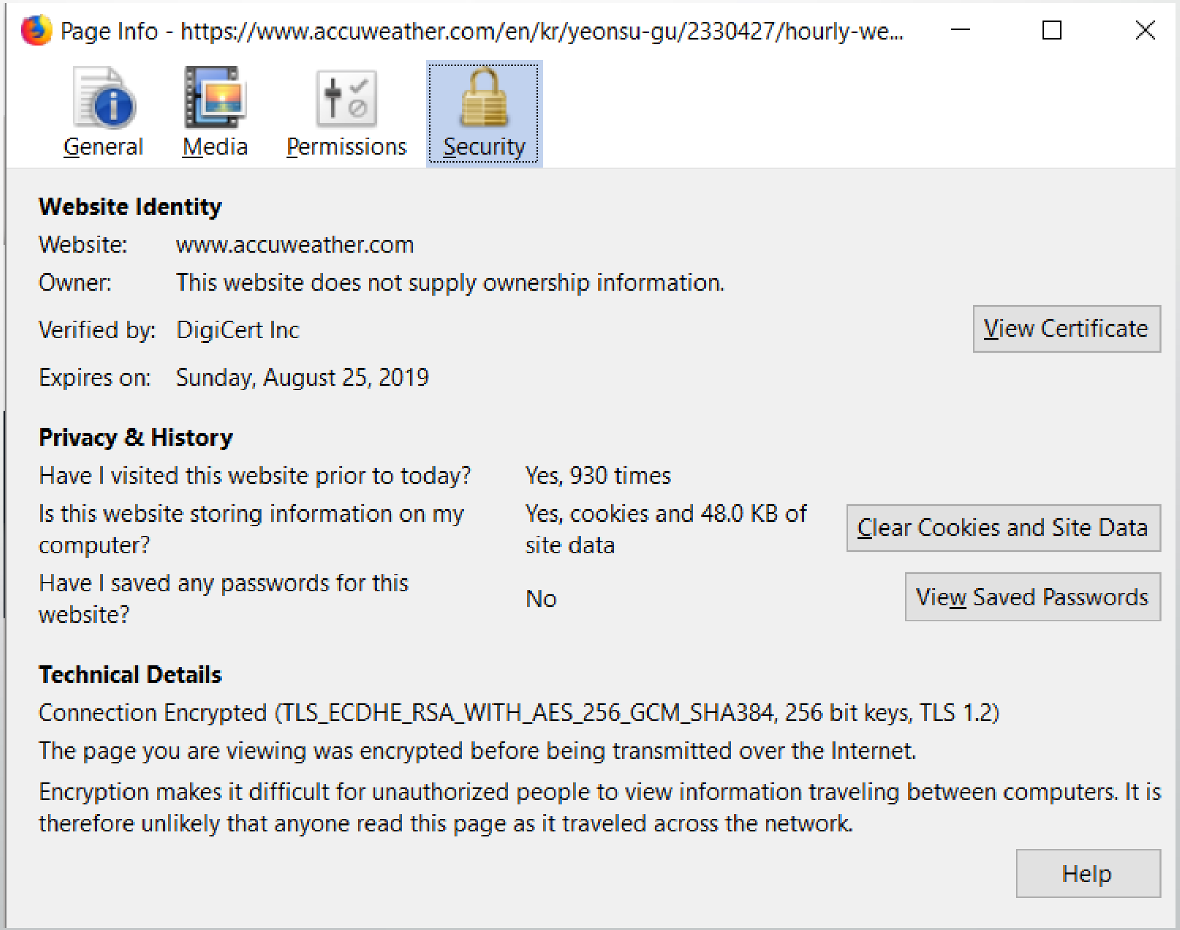


https://www.accuweather.com/en/kr/yeonsu-gu/233042

www.accuweather.com

Secure Connection

Verified by: DigiCert Inc



Page Info - https://www.accuweather.com/en/kr/yeonsu-gu/2330427/hourly-we...

Website Identity

Website: www.accuweather.com

Owner: This website does not supply ownership information.

Verified by: DigiCert Inc

Expires on: Sunday, August 25, 2019

Privacy & History

Have I visited this website prior to today? Yes, 930 times

Is this website storing information on my computer? Yes, cookies and 48.0 KB of site data

Have I saved any passwords for this website? No

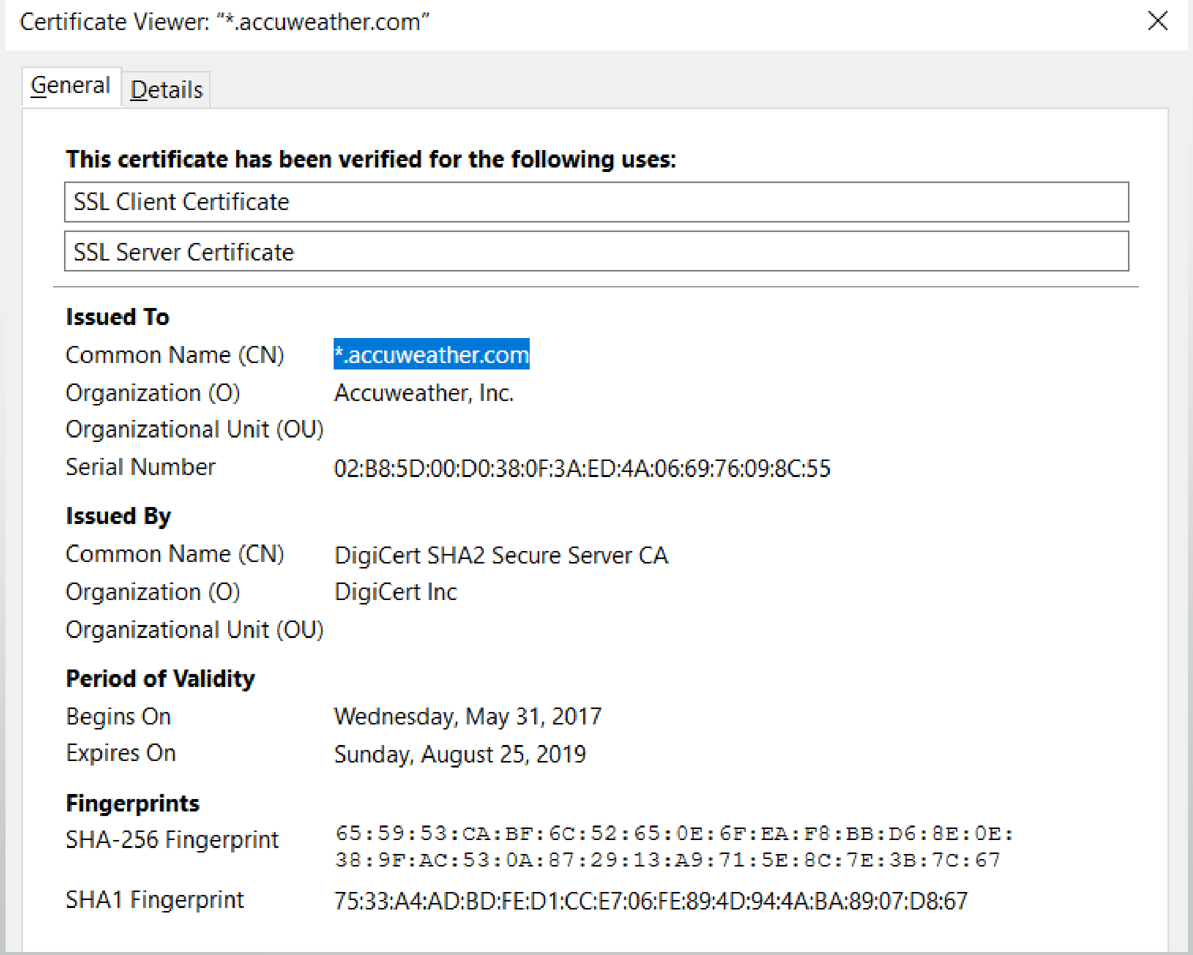
Technical Details

Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, 256 bit keys, TLS 1.2)

The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.

Digital Certificates
Example Use: SSL/TLS



Certificate Viewer: “*.accuweather.com”

This certificate has been verified for the following uses:

SSL Client Certificate

SSL Server Certificate

Issued To

Common Name (CN)	*.accuweather.com
Organization (O)	Accuweather, Inc.
Organizational Unit (OU)	
Serial Number	02:B8:5D:00:D0:38:0F:3A:ED:4A:06:69:76:09:8C:55

Issued By

Common Name (CN)	DigiCert SHA2 Secure Server CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	

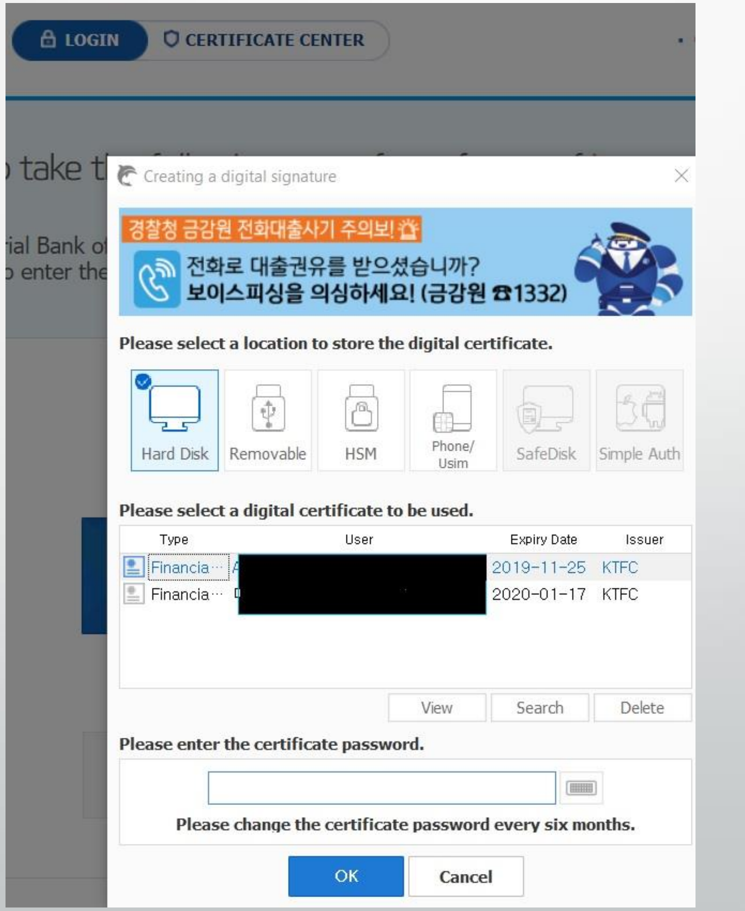
Period of Validity

Begins On	Wednesday, May 31, 2017
Expires On	Sunday, August 25, 2019

Fingerprints

SHA-256 Fingerprint	65:59:53:CA:BF:6C:52:65:0E:6F:EA:F8:BB:D6:8E:0E: 38:9F:AC:53:0A:87:29:13:A9:71:5E:8C:7E:3B:7C:67
SHA1 Fingerprint	75:33:A4:AD:BD:FE:D1:CC:E7:06:FE:89:4D:94:4A:BA:89:07:D8:67

Digital Certificates
Example Use - Authentication



Creating a digital signature

경찰청 금감원 전화대출사기 주의보!

전화로 대출권유를 받으셨습니까?
보이스피싱을 의심하세요! (금감원 ☎1332)

Please select a location to store the digital certificate.

☑ Hard Disk · Removable · HSM · Phone/Usim · SafeDisk · Simple Auth

Please select a digital certificate to be used.

Type	User	Expiry Date	Issuer
Financia…	[redacted]	2019-11-25	KTFC
Financia…	[redacted]	2020-01-17	KTFC

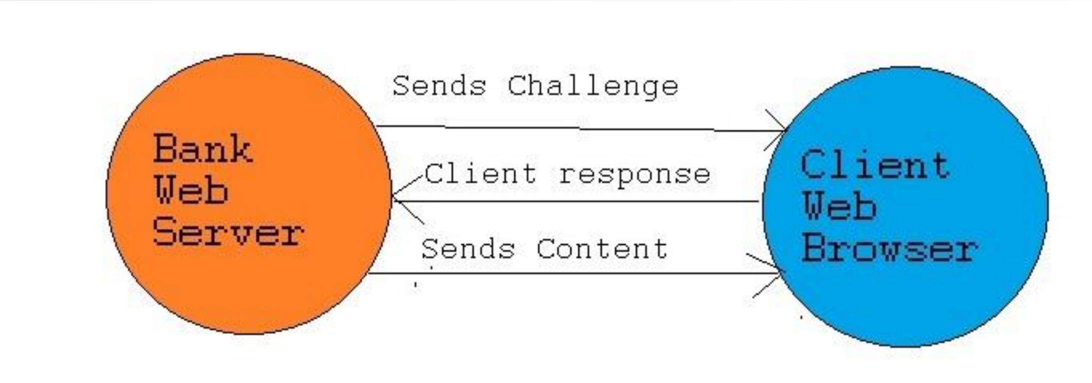
Please enter the certificate password.

Please change the certificate password every six months.

Digital Certificates
Example Use - Authentication

- Passphrase given on certificate login page
  - NOT a password to be sent to server!
  - Unlocks/Decrypts local copy of Private key

Digital Certificates
Example Use - Authentication



1. Bank server issues ‘challenge’ (random data)

2. Client Browser (with plugin code) generates Response based on some type of ‘signing’ operation using private key

a. usually signature is done on a ‘modified’ copy of challenge data

3. Bank server verifies signature which proves client has control of private key

4. Bank server sends content (allows access to client accounts)

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 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 should not have access
  - **Recovery** – Combining sufficient shares to recover a ‘secret’ (usually a key to decrypt valuable data).

Secret Sharing : Basic Approaches

- **M of M** – All distributed shares are required to reconstruct secret. Any less provides NO 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 information about the secret).
- **General Access Scheme** – Certain 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 fro mthe 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

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.

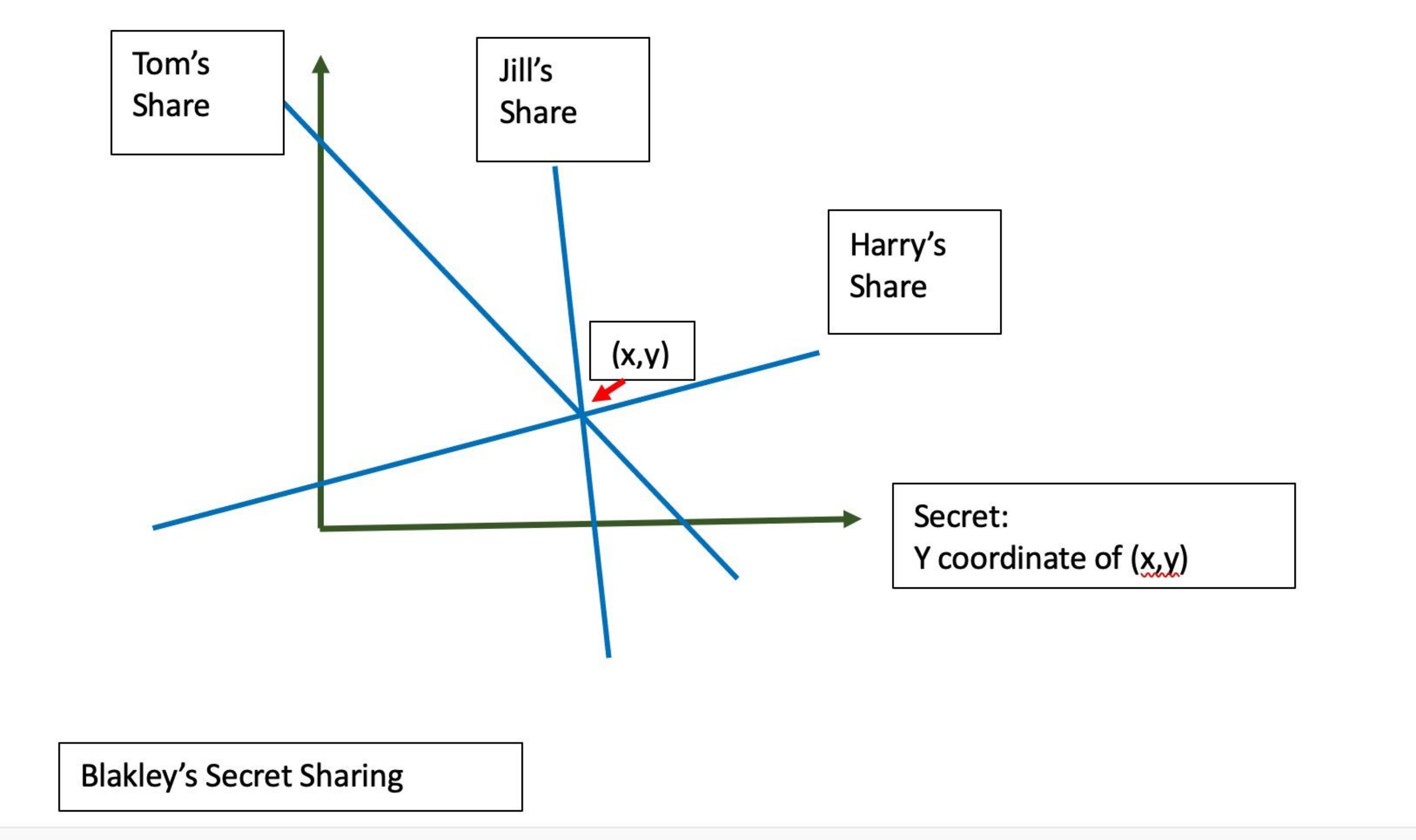
Geometric Secret Sharing

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

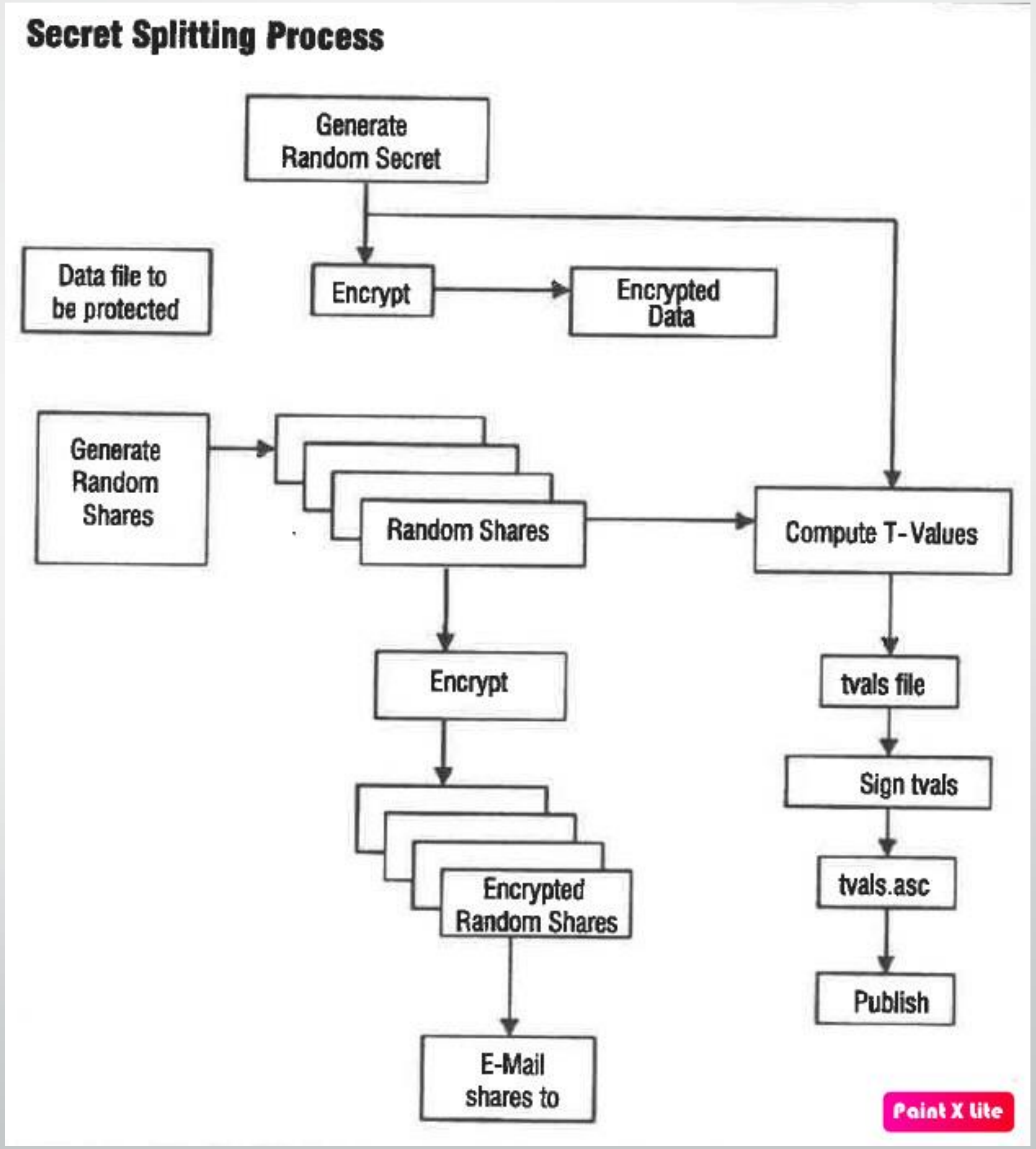
Online Secret Sharing : Cachin

- 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 cryptograph 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

Acces 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)

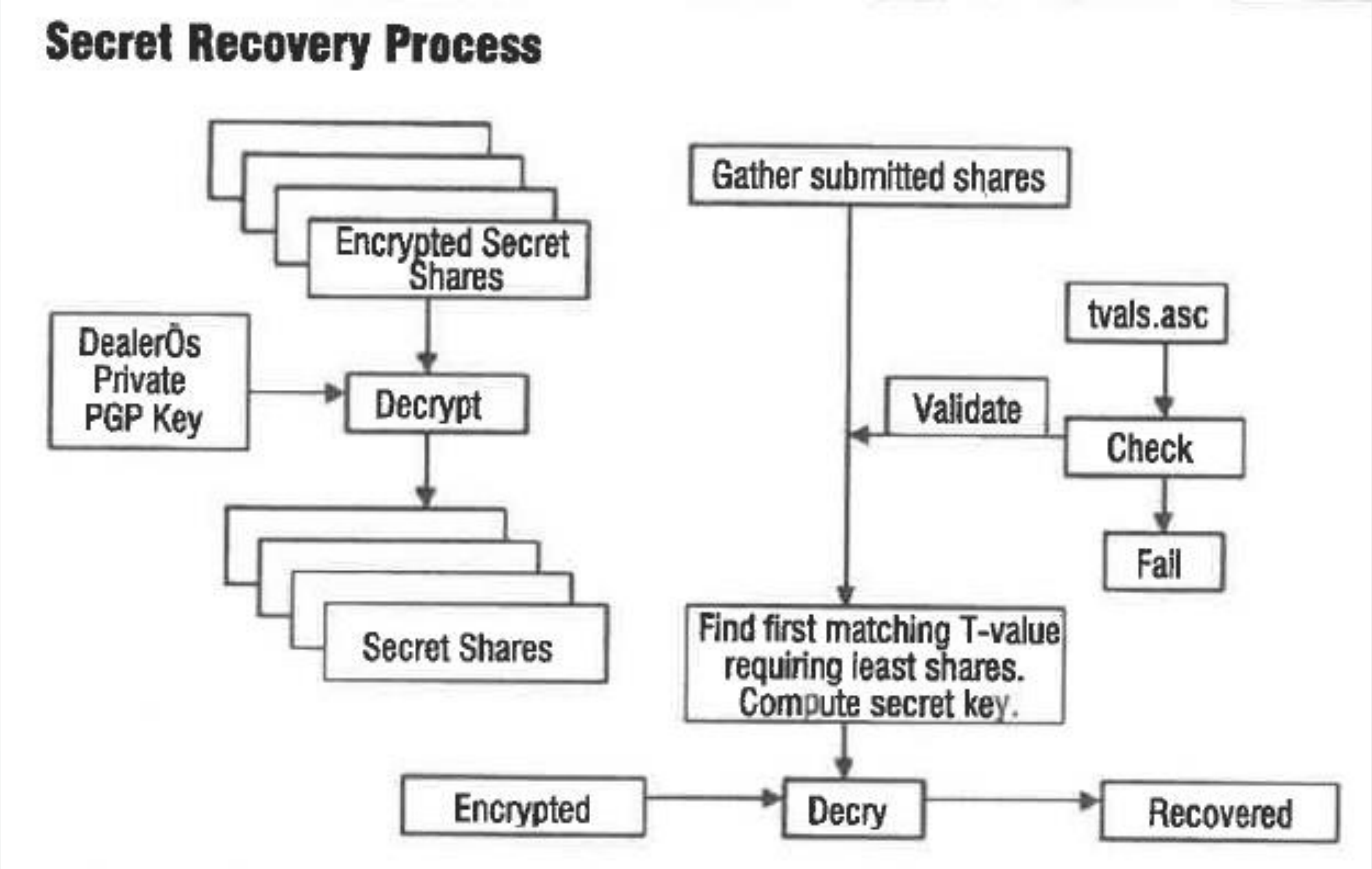
(Securely delete Key)

Publish
Tau[Joe,Phil]
Tau[Phil, Corrine, Bethany]
Tau[Joe, Alan, Melissa]
Tau[Alan, Corrine]

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
  - How to assure participants don’t cheat (have an easy means to discover other shares)
  - How to protect against share ‘leakage`
- A number of research efforts are on going to solve these and other problems/security issues.

Blockchain Technologies

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

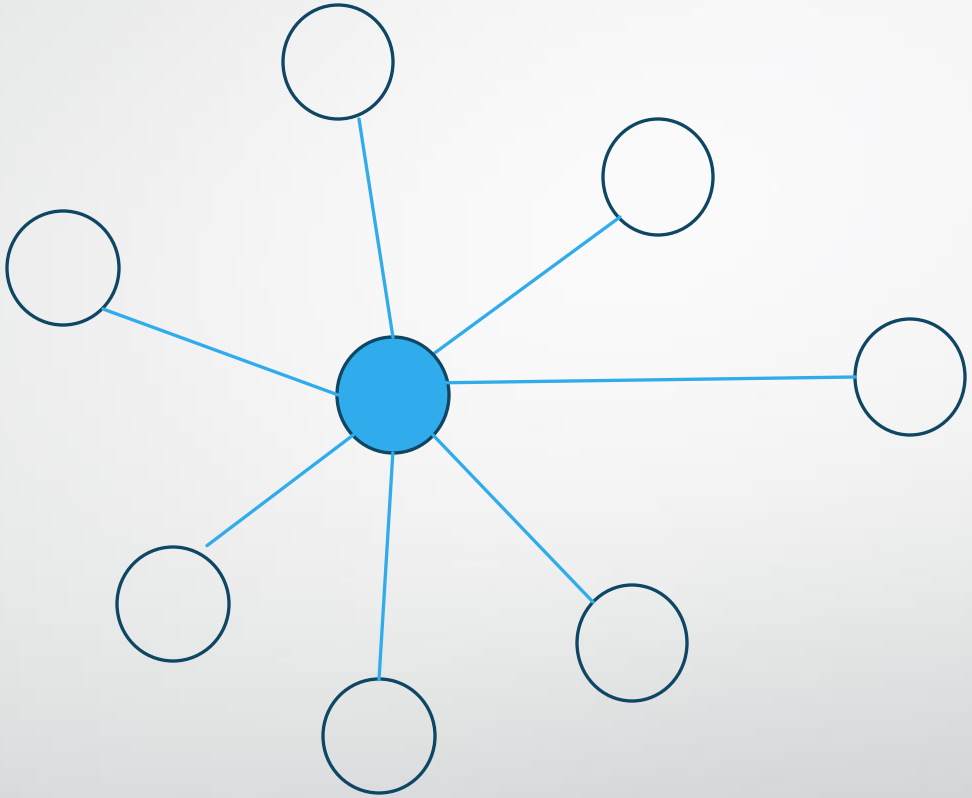
What Is Blockchain?

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

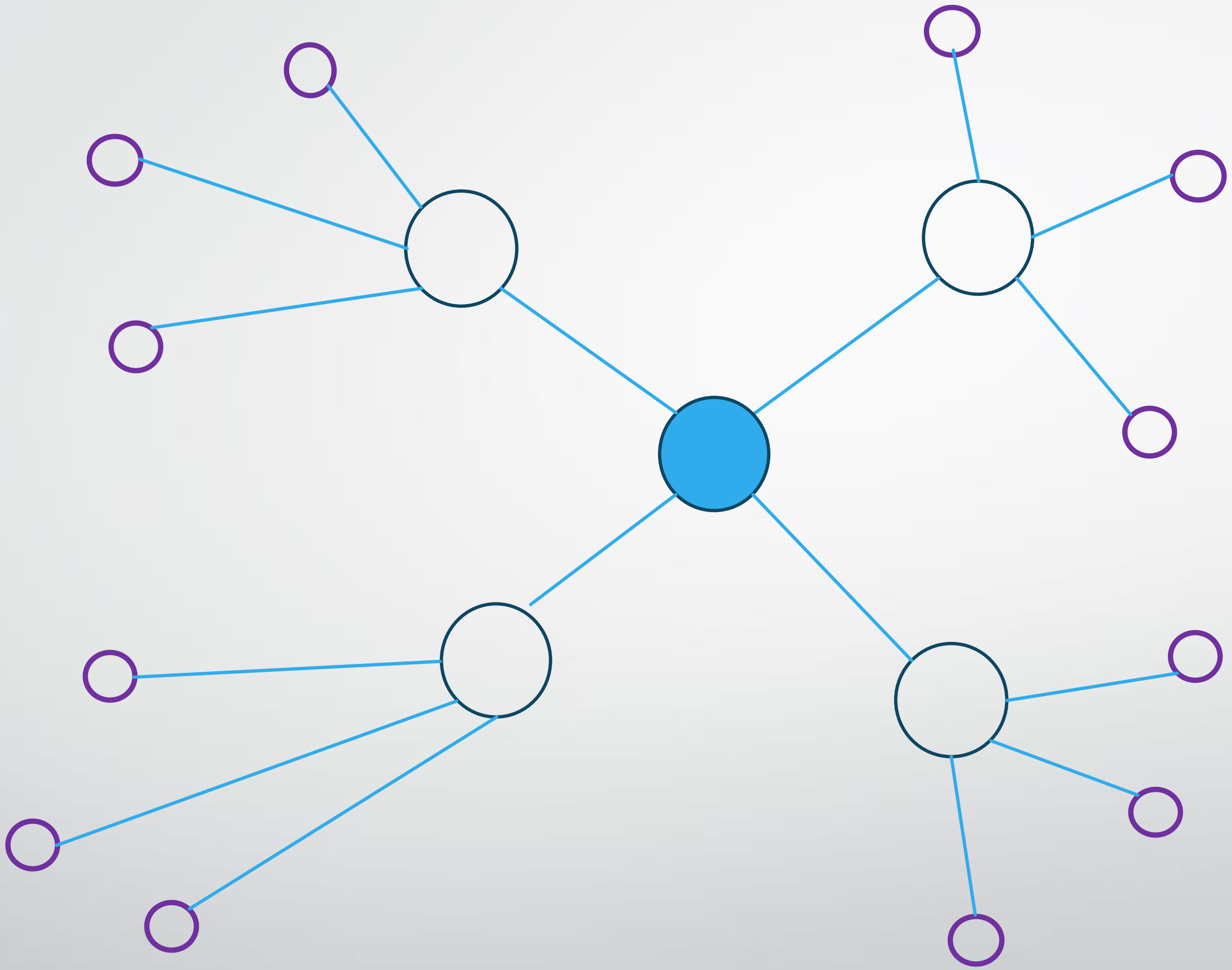
Blockchain Features

- **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.

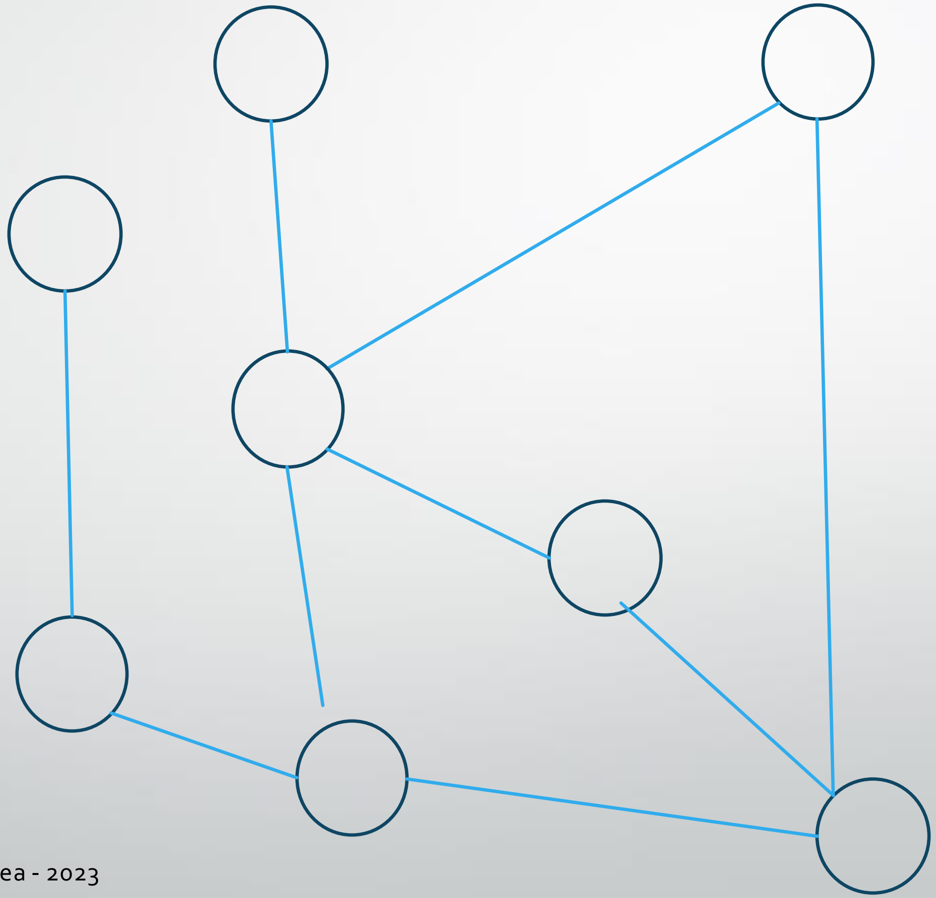
Architecture – [Traditional, Centralized]



Architecture - [Blockchain, Decentralized]



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’]

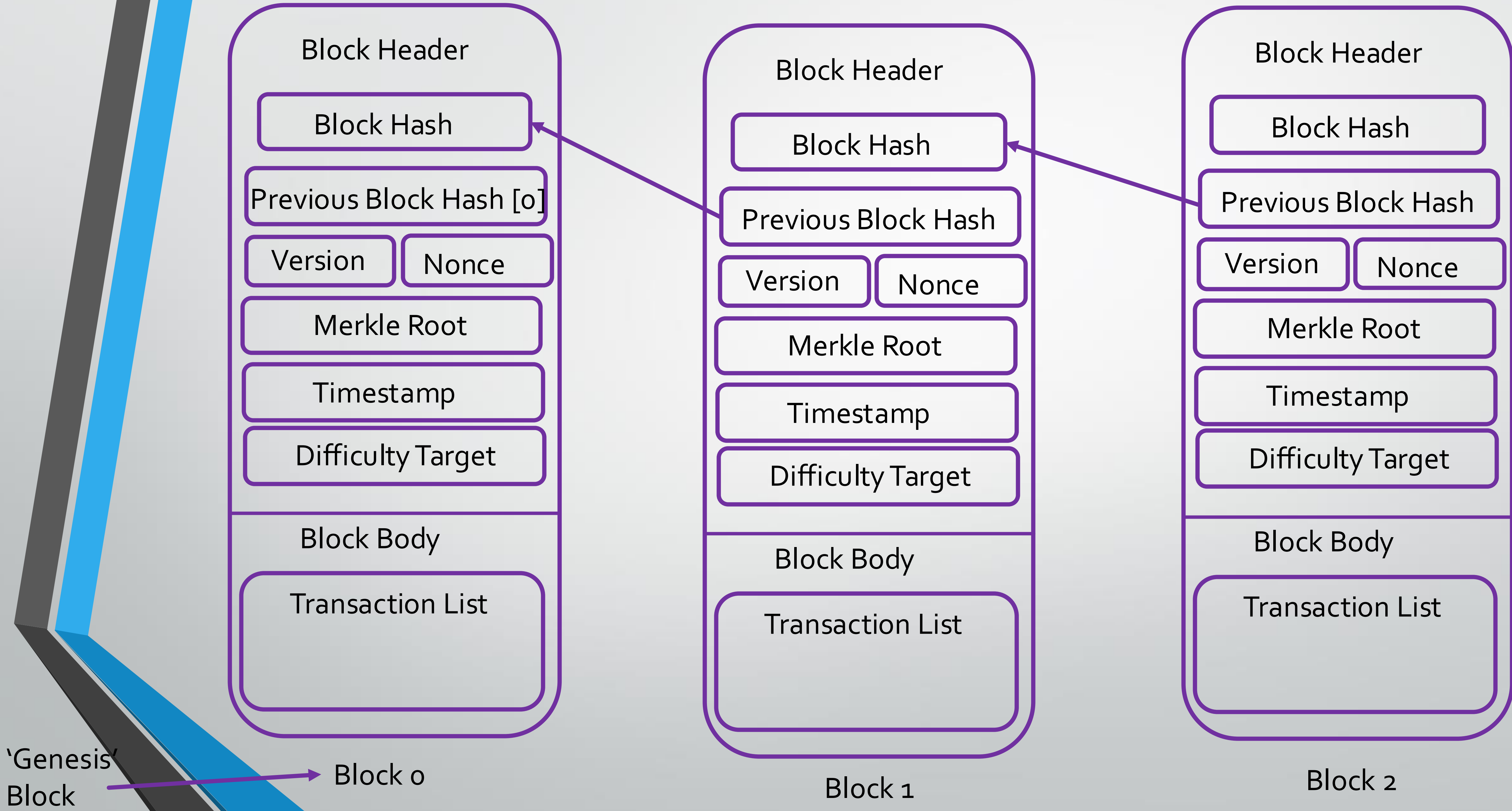
Blockchain Operations

- Collect transactions and build a block
- Submit block to one of the member nodes
- Nodes validate blocks
- Nodes vote on blocks (>50% => Add node to 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

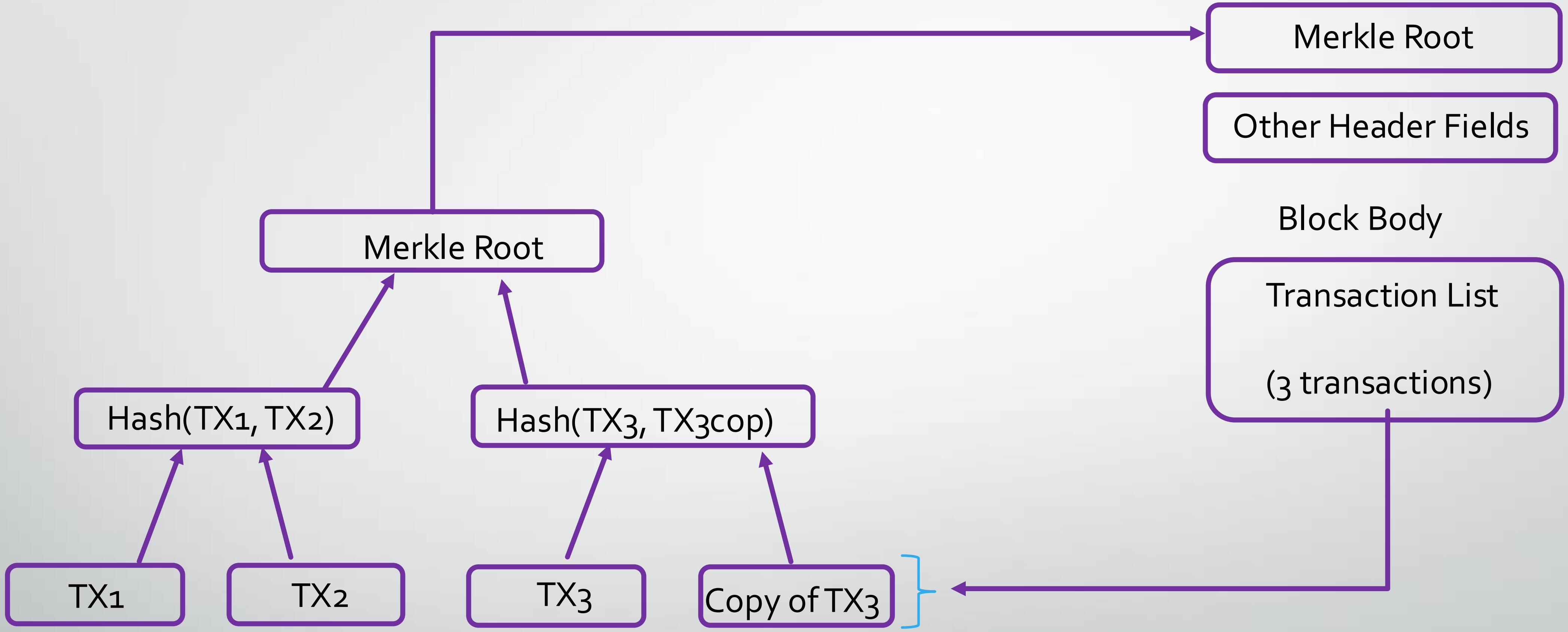
Blockchain Security

- 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



Merkle Root

Other Header Fields

Block Body

Transaction List

(3 transactions)

Merkle Root

Hash(TX1, TX2)

Hash(TX3, TX3cop)

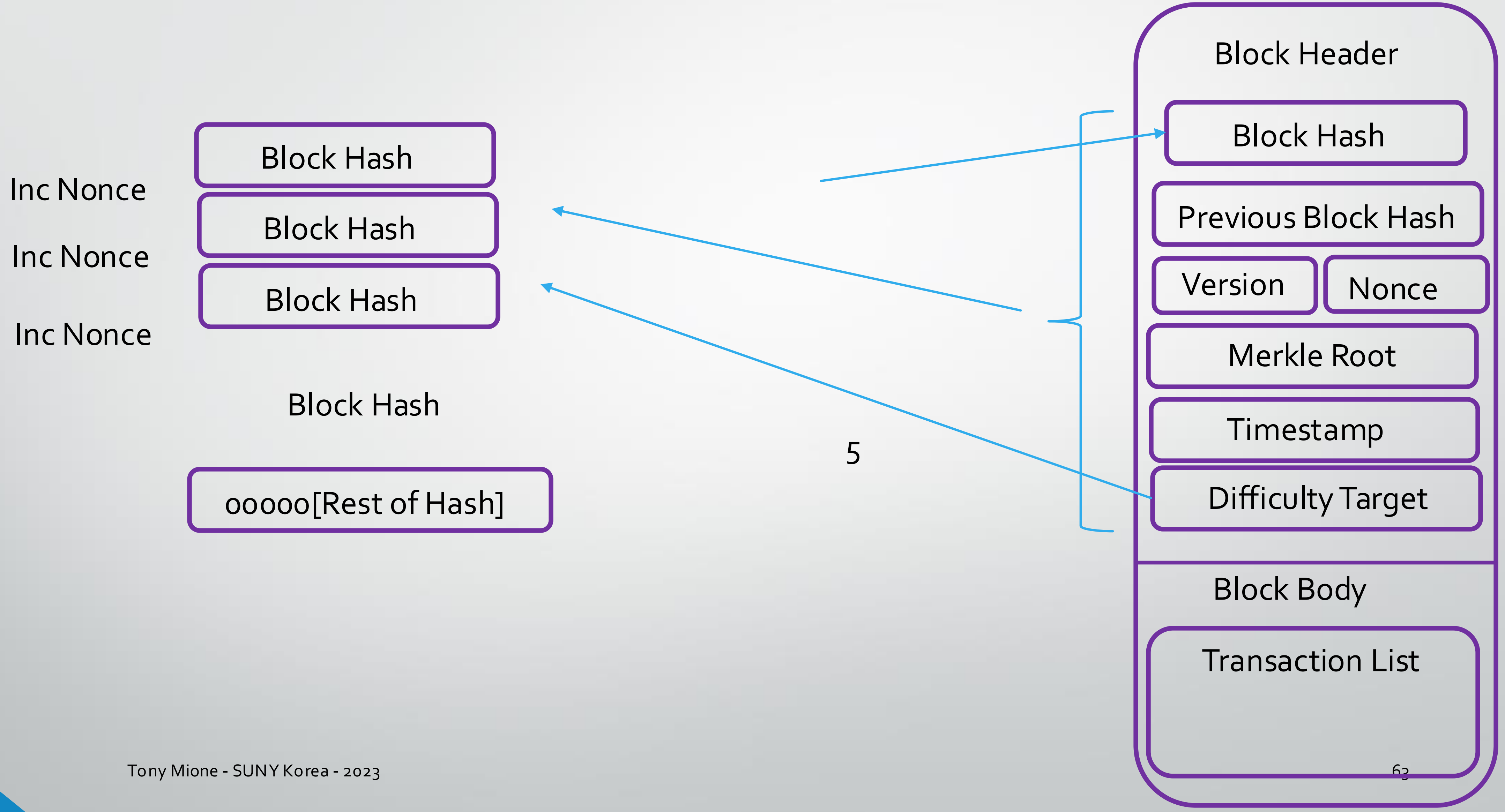
TX1

TX2

TX3

Copy of TX3

Blockchain – PoW (Proof of Work)



Block Hash

Inc Nonce

Block Hash

Inc Nonce

Block Hash

Inc Nonce

Block Hash

ooooo[Rest of Hash]

5

Block Header

Block Hash

Previous Block Hash

Version

Nonce

Merkle Root

Timestamp

Difficulty Target

Block Body

Transaction List

Questions

- Thanks for your attention!