EE817/IS893 Blockchain and Cryptocurrency Cryptographic Primitives

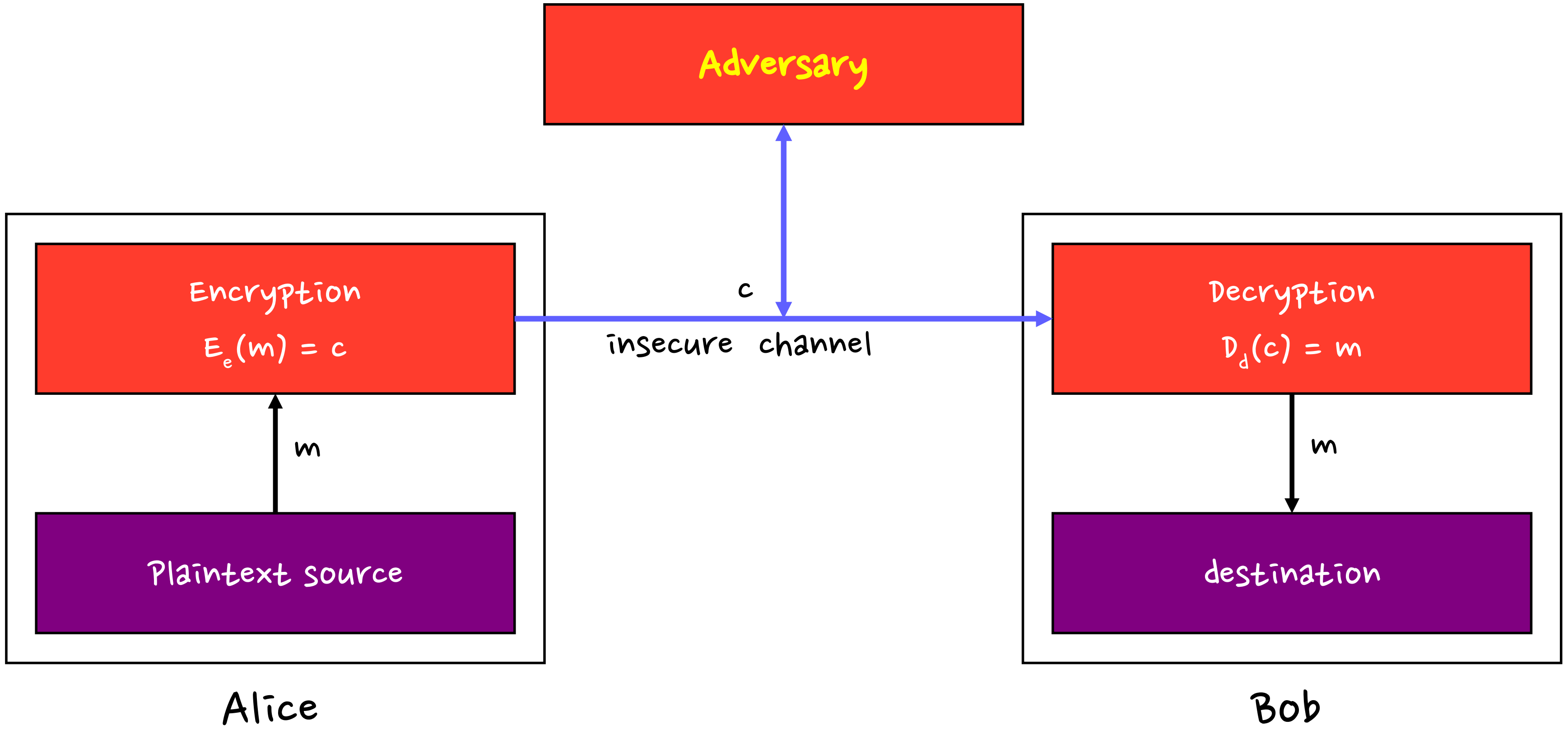
Yongdae Kim

KAIST

Admin

- Student Information Survey
  - https://goo.gl/forms/VnjAyN5N1bmswLNP2
- Paper Presentation Survey
- Paper Presentation vs. Reading Report Scoring
- Project

Encryption

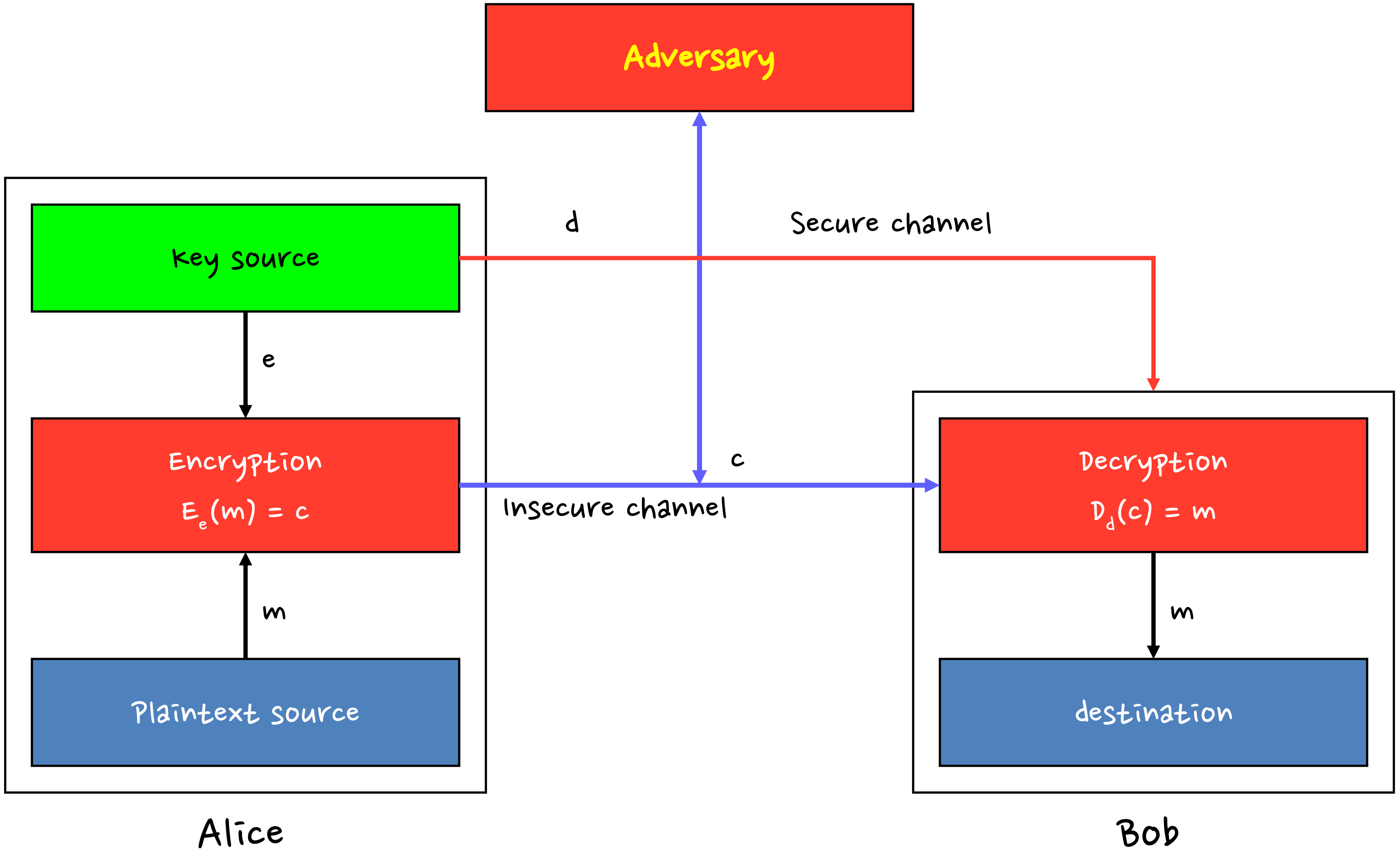


- Why do we use key?
  - Or why not use just a shared encryption function?

Symmetric-key encryption

- Encryption scheme is symmetric-key
  - if for each (e,d) it is easy computationally easy to compute e knowing d and d knowing e
  - Usually e = d
- Block Cipher
  - Breaks plaintext into block of fixed length
  - Encrypts one block at a time
- Stream Cipher
  - Takes a plaintext string and produces a ciphertext string using keystream
  - Block cipher with block length 1

SKE with Secure channel

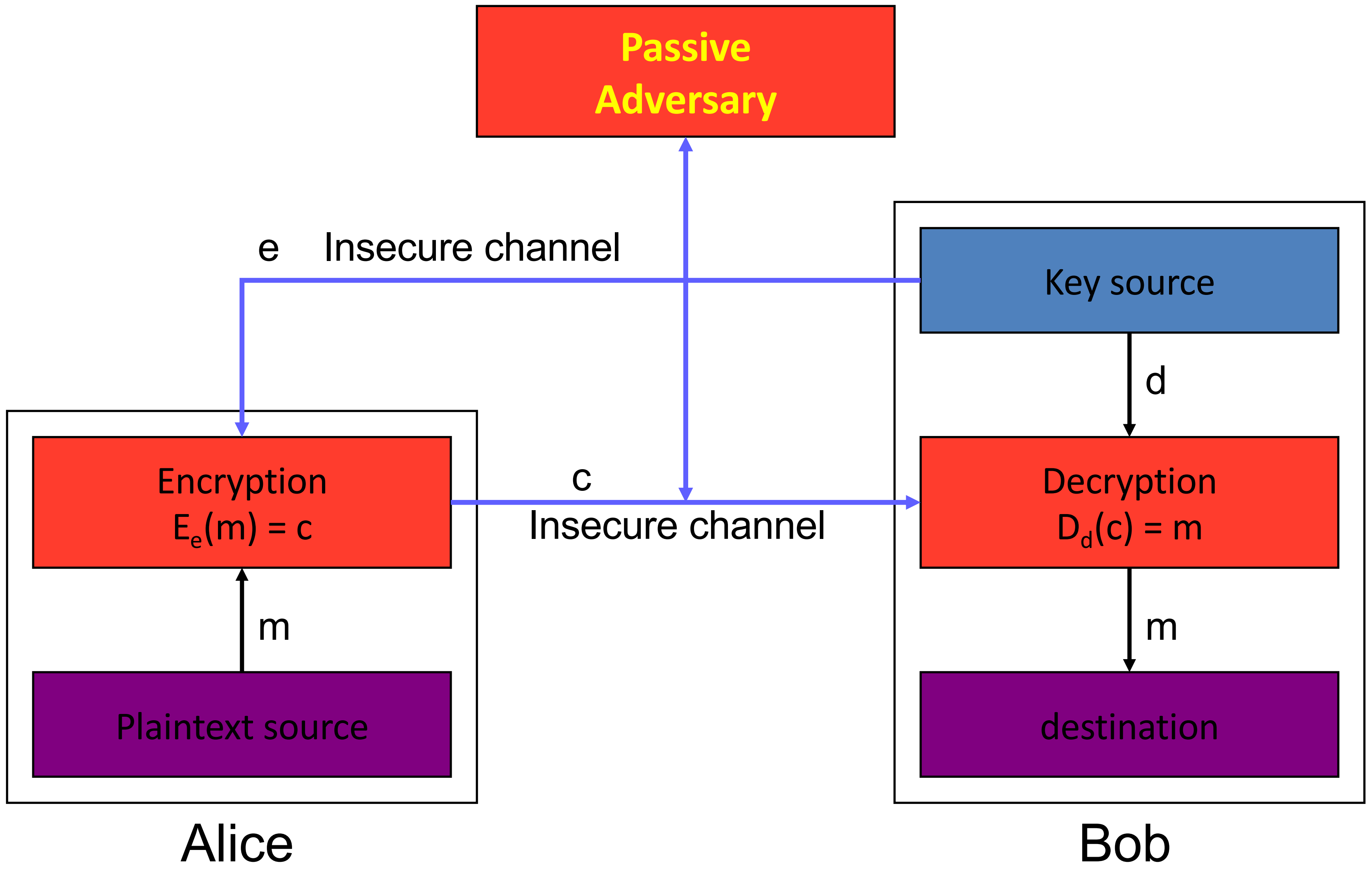


Public-key Encryption (Crypto)

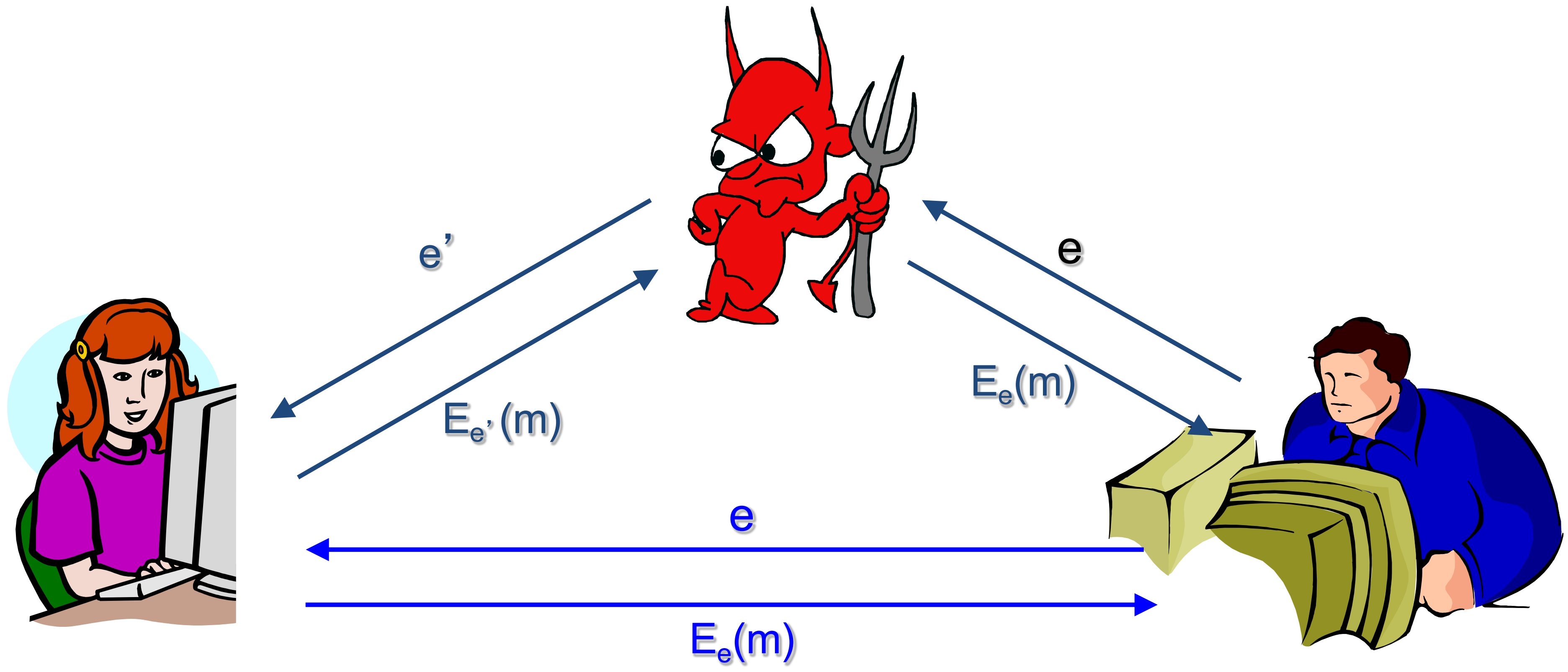
- Every entity has a private key SK and a public key PK
  - Public key is known to all
  - It is computationally infeasible to find SK from PK
  - Only SK can decrypt a message encrypted by PK

- If A wishes to send a private message M to B
  - A encrypts M by B's public key, $C = E_{B_{PK}}(M)$
  - B decrypts C by his private key, $M = D_{B_{SK}}(C)$

PKE with Insecure Channel



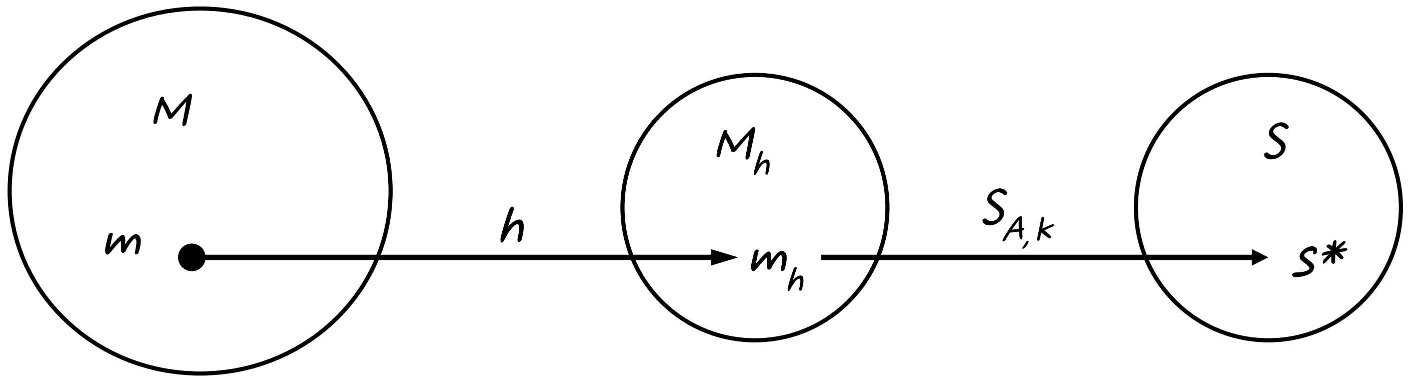
Public Key should be authentic!

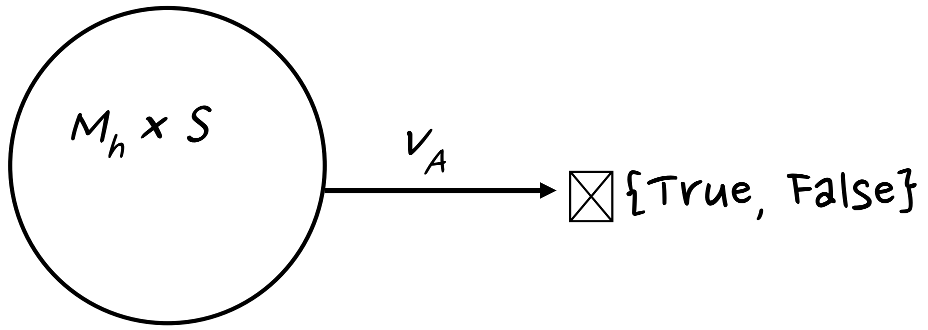


Digital Signatures

- Primitive in authentication and non-repudiation
- Signature
  - Process of transforming the message and some secret information into a tag
- Nomenclature
  - M is set of messages
  - S is set of signatures
  - S_A is signature transformation from M to S for A, kept private
  - V_A is verification transformation from M to S for A, publicly known

Digital Signature with Appendix





$$s^* = S_{A,k}(m_h)$$

$$u = v_A(m_h, s^*)$$

Hash function and MAC

- A hash function is a function h
  - compression — h maps an input x of arbitrary finite bitlength, to an output h(x) of fixed bitlength n.
  - ease of computation — h(x) is easy to compute for given x and h
  - Properties
    - » one-way: for a given y, find x' such that h(x') = y
    - » collision resistance: find x and x' such that h(x) = h(x')

- MAC (message authentication codes)
  - both authentication and integrity
  - MAC is a family of functions h_k
    - » ease of computation (if k is known !!)
    - » compression, x is of arbitrary length, $h_k(x)$ has fixed length
    - » computation resistance: given $(x', h_k(x'))$ it is infeasible to compute a new pair $(x, h_k(x))$ for any new $x \neq x'$

Message Authentication Code MAC

- MAC is a family of functions h_k
  - ease of computation (if k is known !!)
  - compression, x is of arbitrary length, $h_k(x)$ has fixed length
  - computation resistance: given $(x', h_k(x'))$ it is infeasible to compute a new pair $(x, h_k(x))$ for any new x□ x'

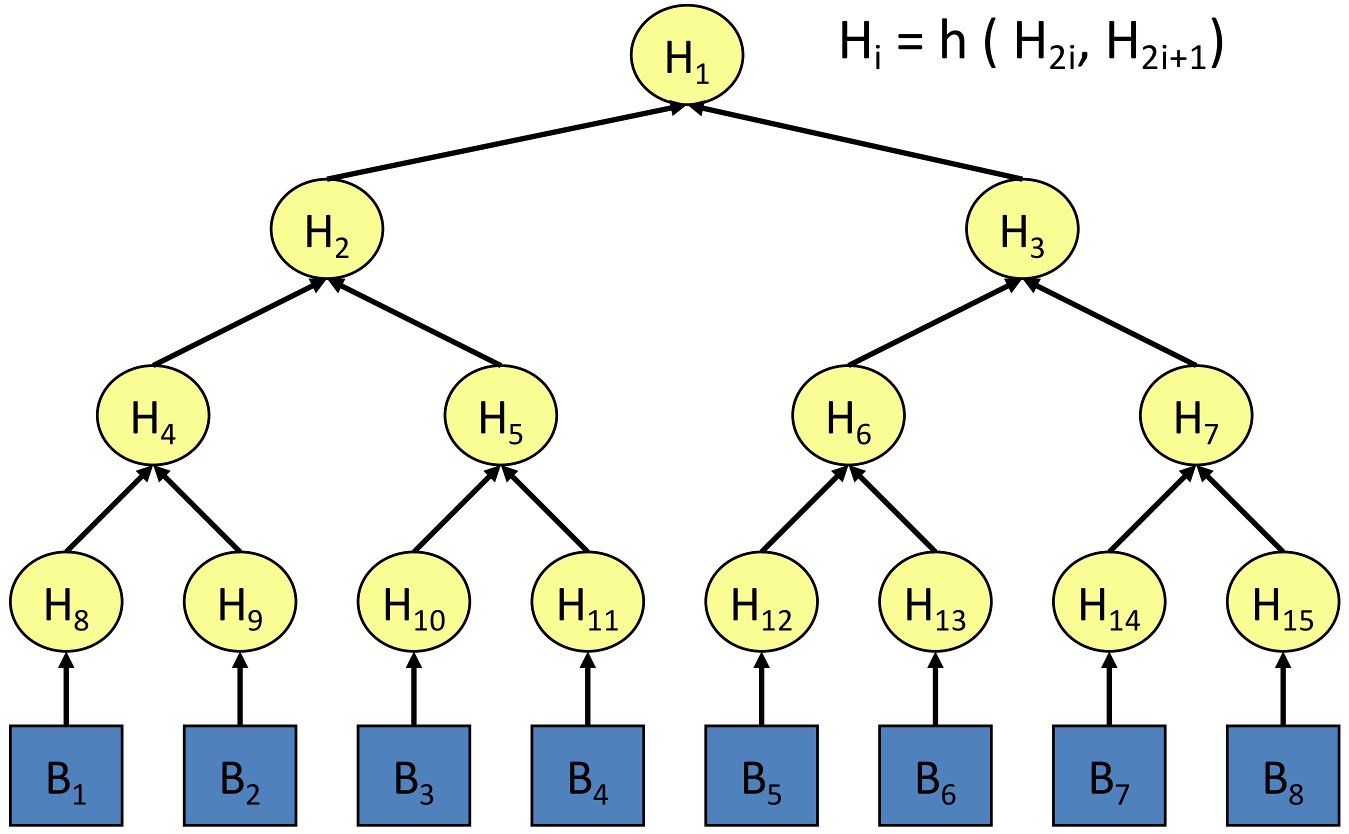
- Typical use
  - $A \rightarrow B: (x, H = h_k(x))$
  - B: verifies if $H = h_k(x)$

- Properties
  - Without k, no one can generate valid MAC.
  - Without k, no one can verify MAC.
  - both authentication and integrity

Authentication

- How to prove your identity?
  - Prove that you know a secret information
- When key K is shared between A and Server
  - A $\rightarrow$ S: $HMAC_K(M)$ where M can provide freshness
  - Why freshness?
- Digital signature?
  - A $\rightarrow$ S: $Sig_{SK}(M)$ where M can provide freshness
- Comparison?

Merkle Hash Tree



$H_i = h\,(H_{2i}, H_{2i+1})$

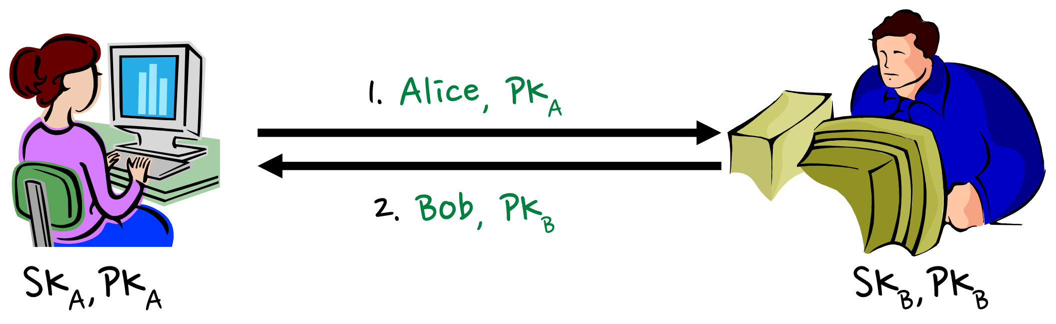
Key Management

- Key establishment
  - Process to whereby a shared secret key becomes available to two or more parties
  - Subdivided into key agreement and key transport.

- Key management
  - The set of processes and mechanisms which support key establishment
  - The maintenance of ongoing keying relationships between parties

Key Management Through PKE

0xDAD12345	Alice
0xBADD00D1	Bob



- Advantages
  - TTP not required
  - Only *n* public keys need to be stored
  - The central repository could be a local file
- Problem
  - Public key authentication problem
- Solution
  - Need of TTP to certify the public key of each entity

Public Key Certificates

- ❑ Entities trust a third party, who issues a certificate
- ❑ Certificate = (data part, signature part)
  - ▹ Data part = (name, public-key, other information)
  - ▹ Signature = (signature of TTP on data part)
- ❑ If B wants to verify authenticity of A's public key
  - ▹ Acquire public key certificate of A over a secured channel
  - ▹ Verify TTP's signature
  - ▹ If signature verified A's public key in the certificate is authentic

Symmetric vs. Public key

	Pros	Cons
SKE	▪ High data throughput ▪ Relatively short key size	▪ The key must remain secret at both ends ▪ $O(n^2)$ keys to be managed ▪ Relatively short lifetime of the key
PKE	▪ $O(n)$ keys ▪ Only the private key must be kept secret ▪ longer key life time ▪ digital signature	▪ Low data throughput ▪ Much larger key sizes

Questions?

- Yongdae Kim
  - email: yongdaek@kaist.ac.kr
  - Home: http://syssec.kaist.ac.kr/~yongdaek
  - Facebook: https://www.facebook.com/y0ngdaek
  - Twitter: https://twitter.com/yongdaek
  - Google "Yongdae Kim"