CS2105 Lecture 8
Network Security

18 March, 2013

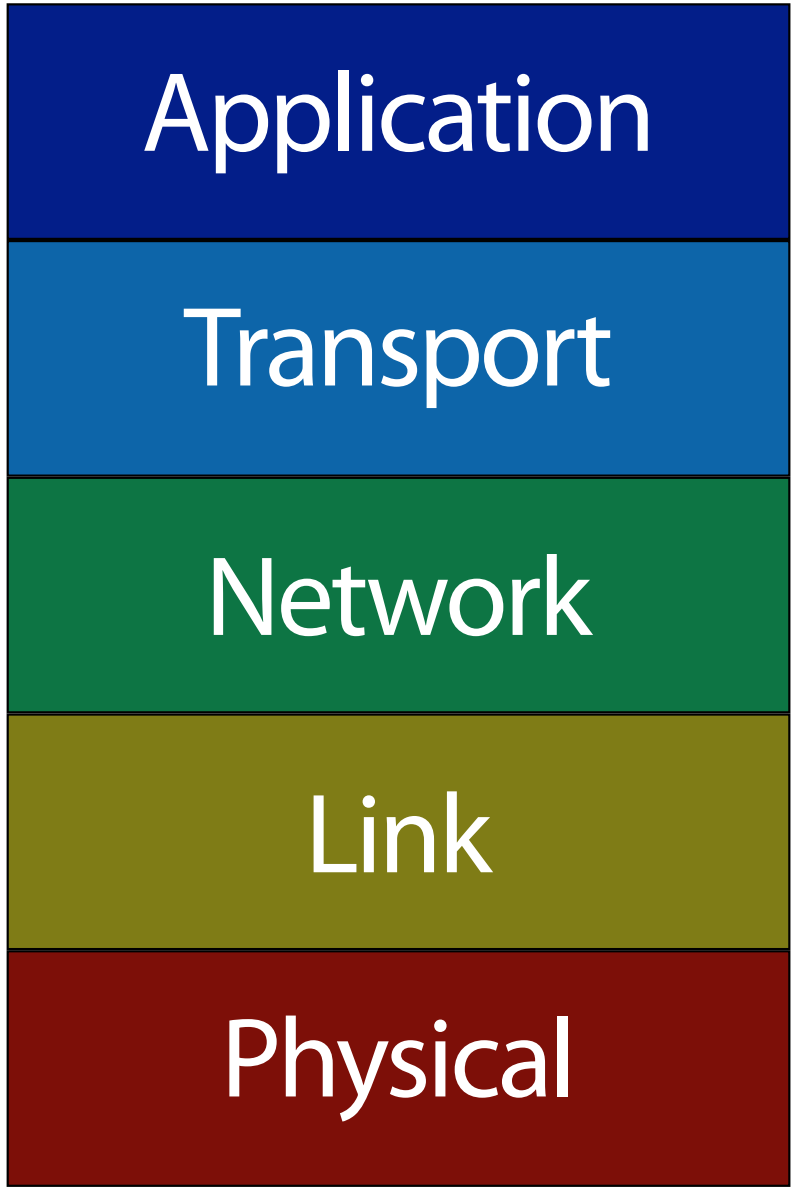
CS2105 Lecture 8

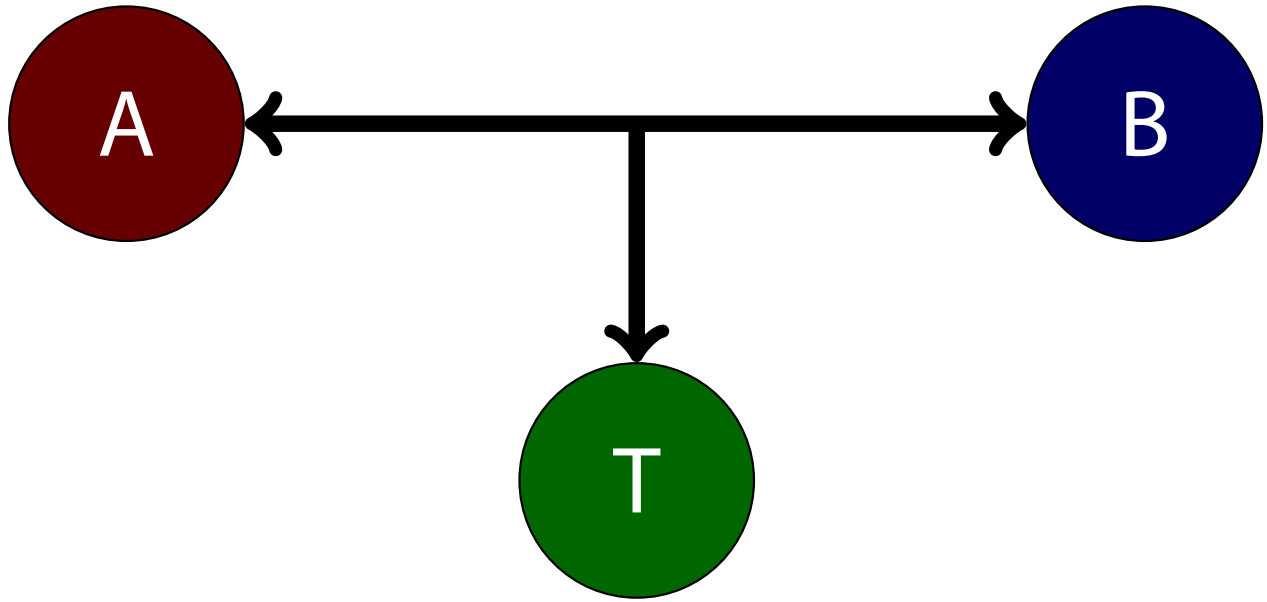
Network Security

18 March, 2013

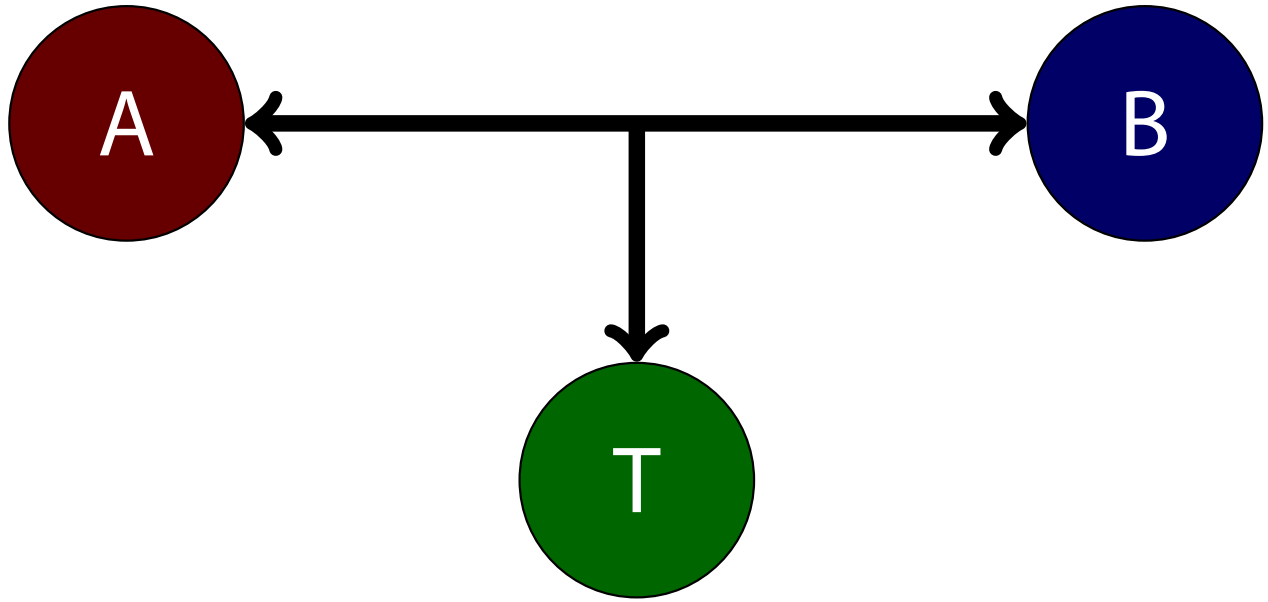
After this class, you are expected to be able to:

- describe how network security can be compromised.
- analyze simple security protocols and identify flaws.
- understand (i) how symmetric key cryptography and public key cryptograph can be used to ensure confidentiality; (ii) how MAC (with authentication key and cryptographic hash function) ensure message integrity; (iii) how digital signature (with nonce) ensure source authenticity.
- understand the role of certificate authority and digital certificates.
- understand how the different techniques are applied in secure email, VPN, IPSec, SSL.

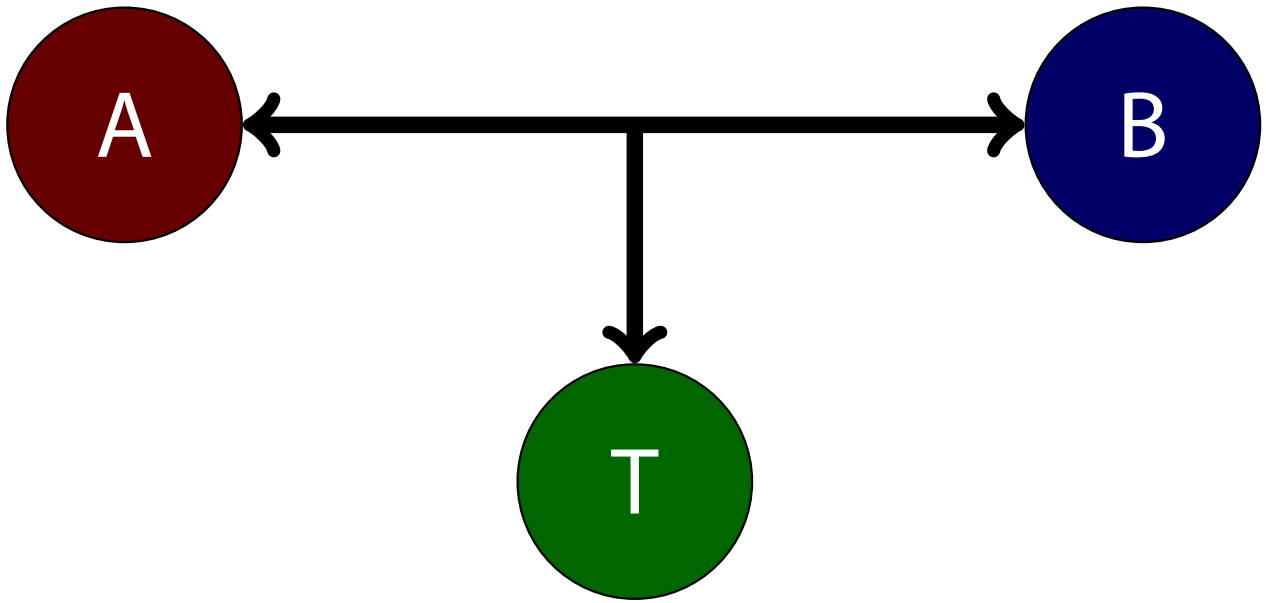

Application
Transport
Network
Link
Physical


A
B
T

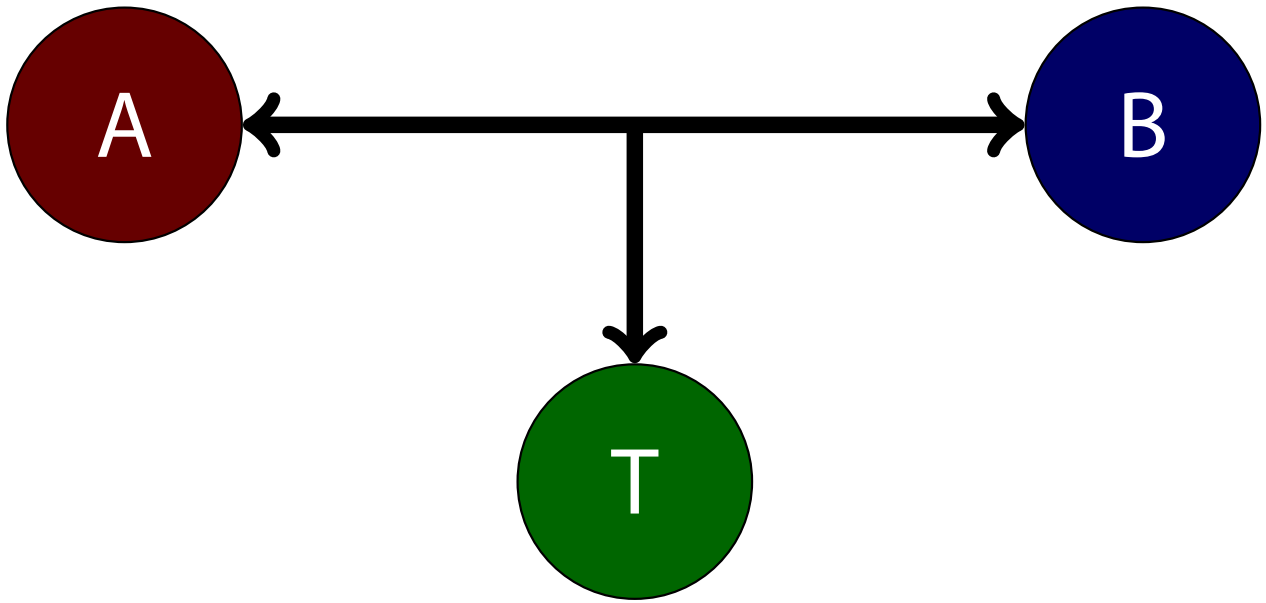
T can **eavesdrop** on *A* and *B*.
Need to ensure **confidentiality**.



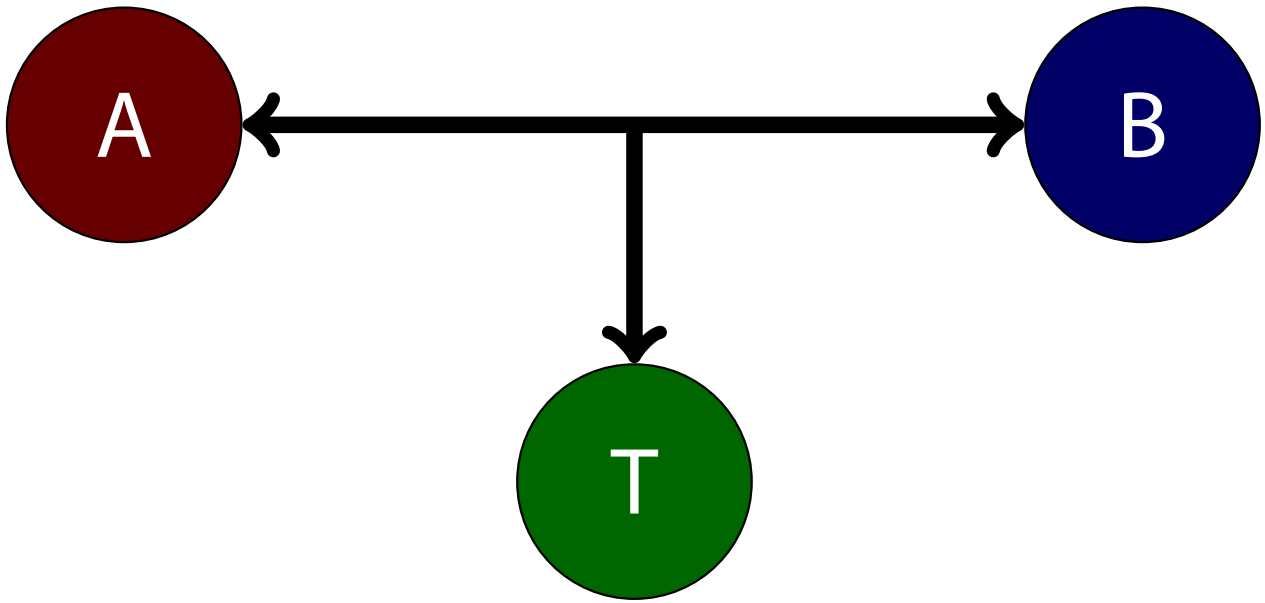
T can **modify, insert, delete messages** between *A* and *B*.
Need to ensure **message integrity**.



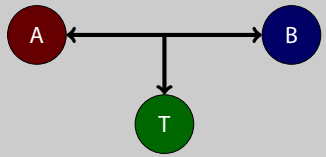
T can **pretend** to be *A* or *B*. Need to ensure **end-point authenticity**.



T can **attack the communication channel** between *A* or *B*.
Need to ensure **operational security**.



T can **attack the communication channel** between *A* or *B*.
Need to ensure **operational security**.



We skip two important practical topics in CS2105, firewalls and intrusion detection system, since they are rather straightforward. You can read all about them yourself in Section 8.9.

Ensuring **confidentiality** with **cryptography**.

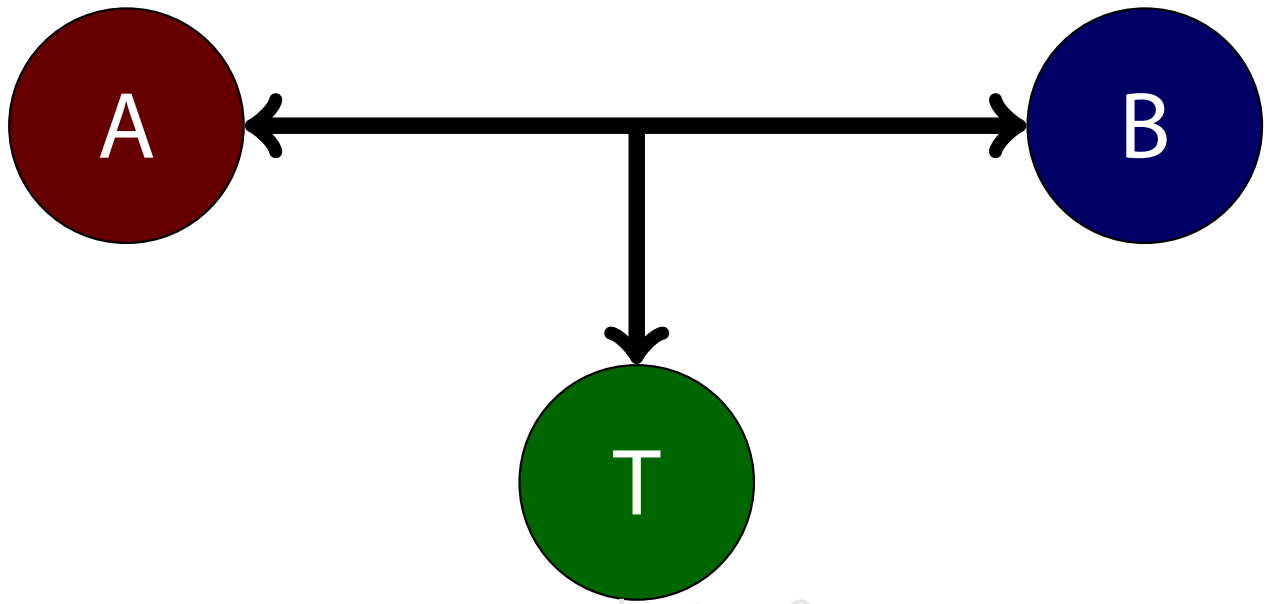
Abstraction

ciphertext = encrypt(plaintext, key)

plaintext = decrypt(ciphertext, key)

Abstraction

$$m' = K_A(m)$$
$$m = K_B(K_A(m))$$



Property of K_A and K_B

Given $K_A(m)$, it should be computationally hard to compute m without K_B.

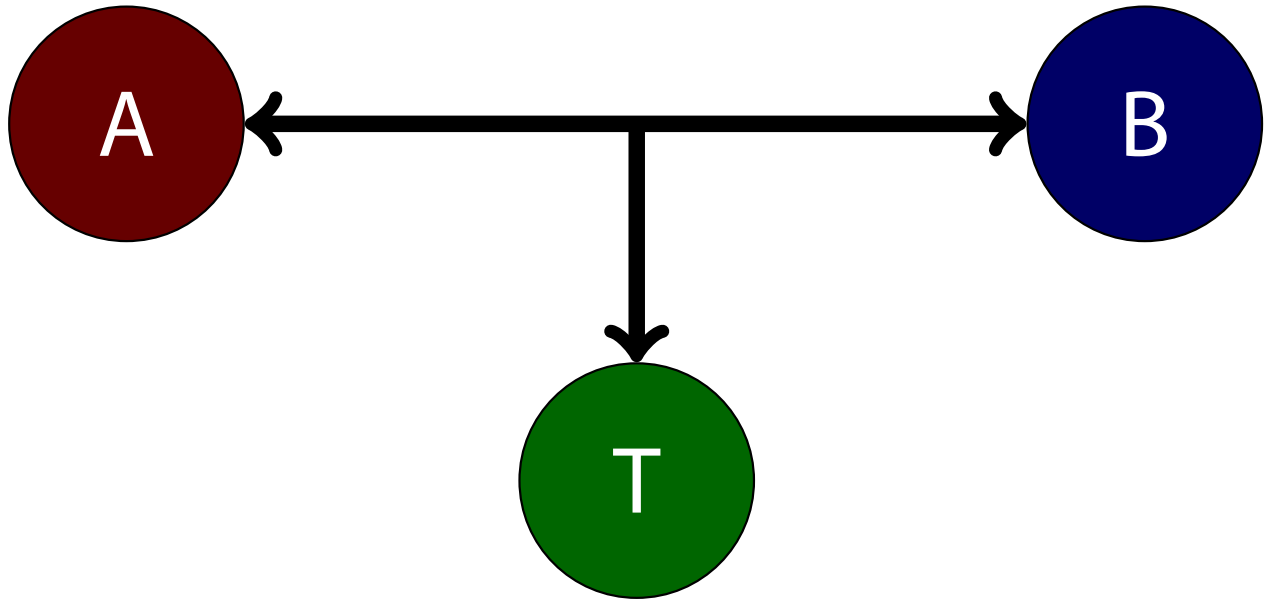
Property of K_A and K_B

Given $K_A(m)$, it should be computationally hard to compute m without K_B.

We skip the mathematical details on how K_A and K_B are implemented. Interested students can read Section 8.2 in details and take CS4236.

Symmetric Key Cyptography

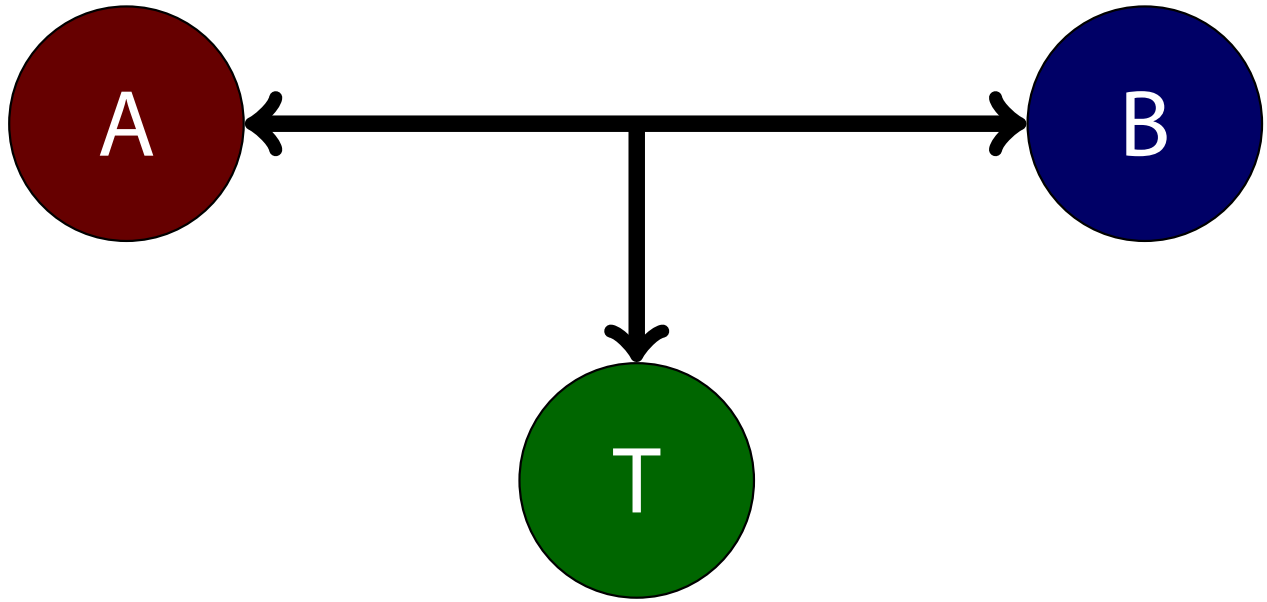
$$K_A = K_B$$



Public Key Cyptography

$$K_A \neq K_B$$

$$m' = K_B^+(m)$$
$$m = K_B^-(K_B^+(m))$$


A
B
T

Ensuring **message integrity** with **message authentication code (MAC)**.

Cryptographic Hash Function

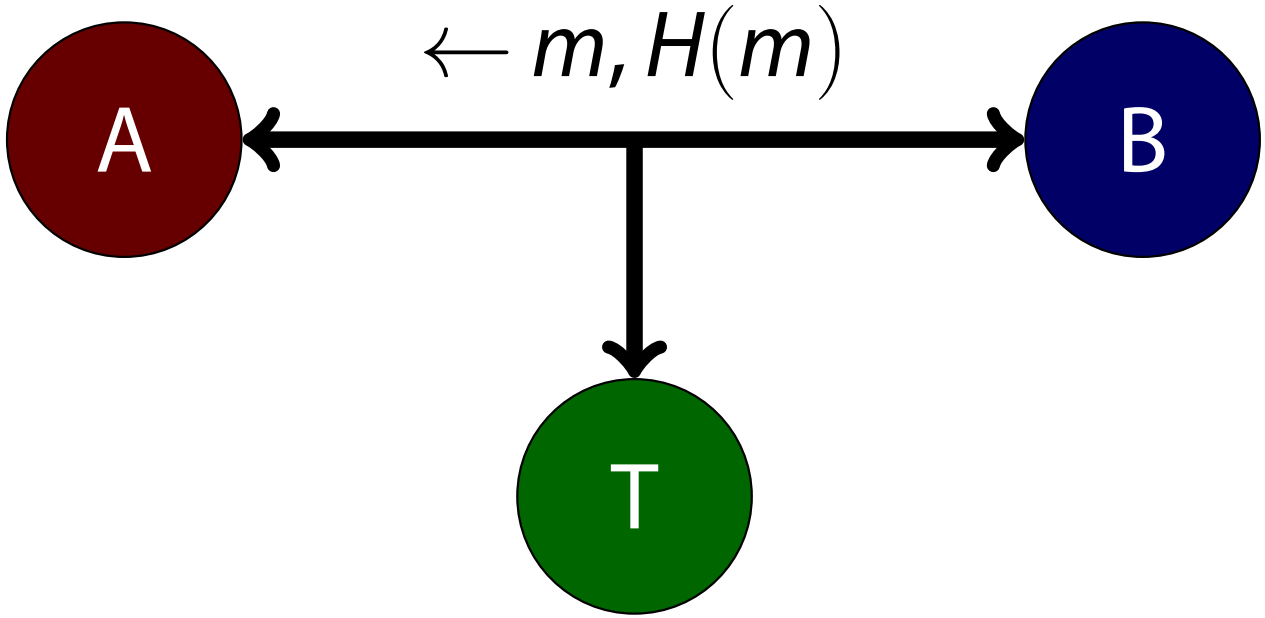
$$\text{hash} = H(\text{message})$$

Property of *H*

Given *H*(m), it should be computationally hard to construct m' such that $H(m') = H(m)$.

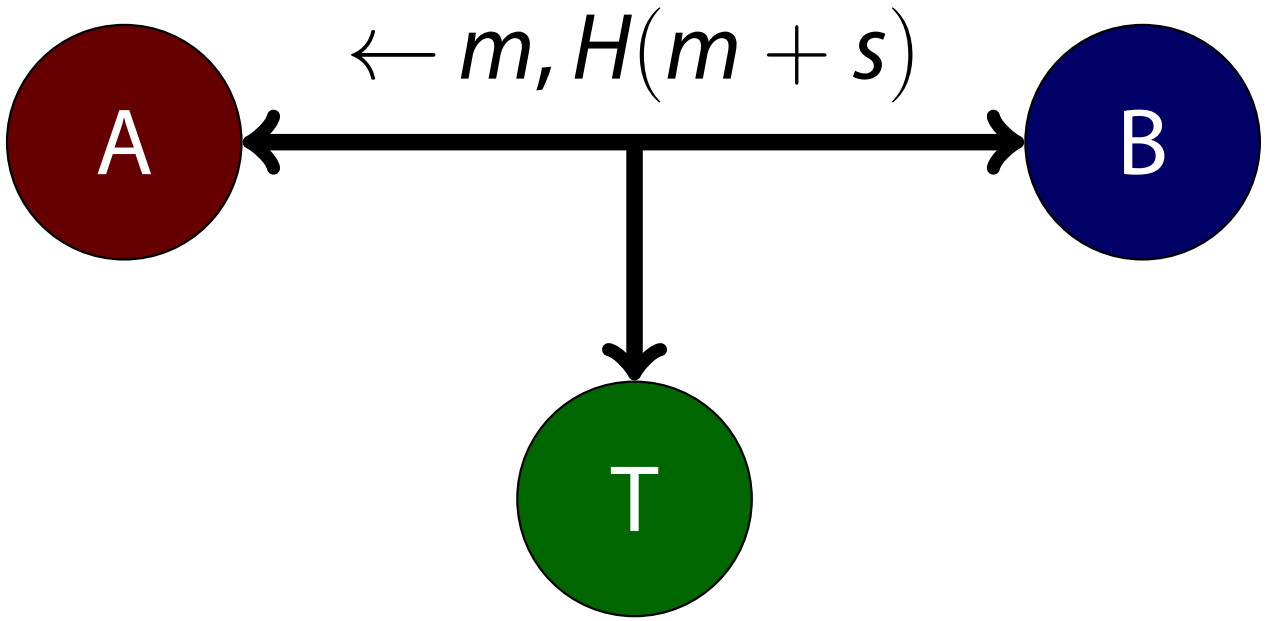
Message Authentication Code

(flawed)



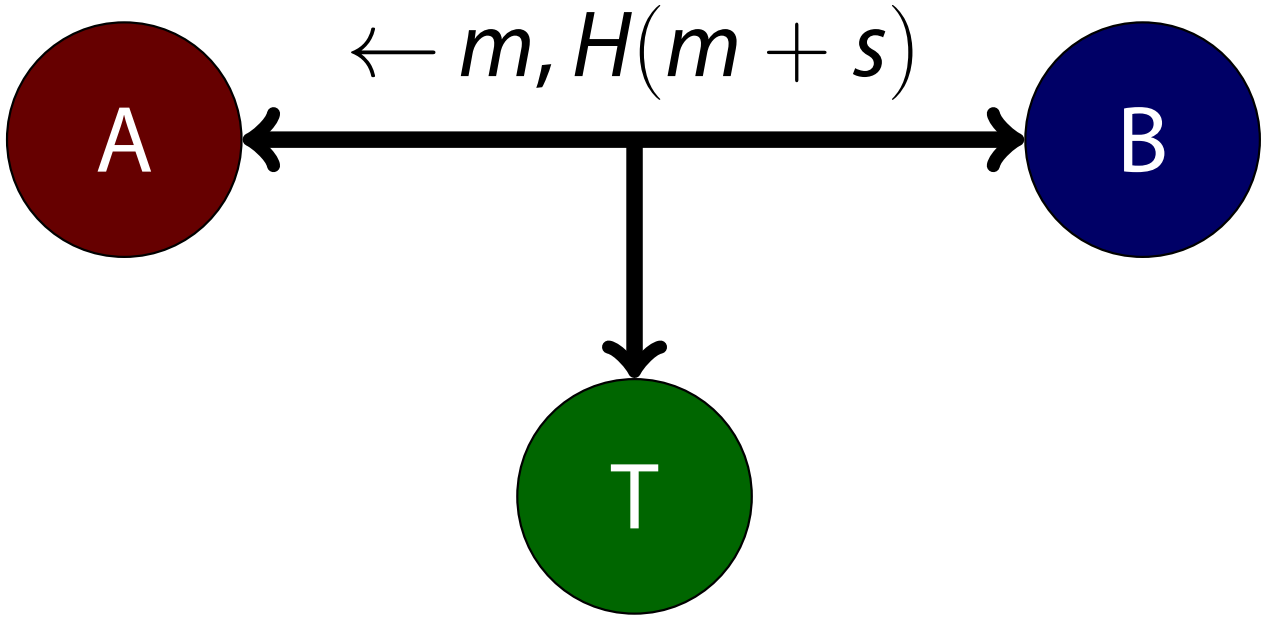
Message Authentication Code

(fixed with authentication key *s*, a shared secret between *A* and *B*)

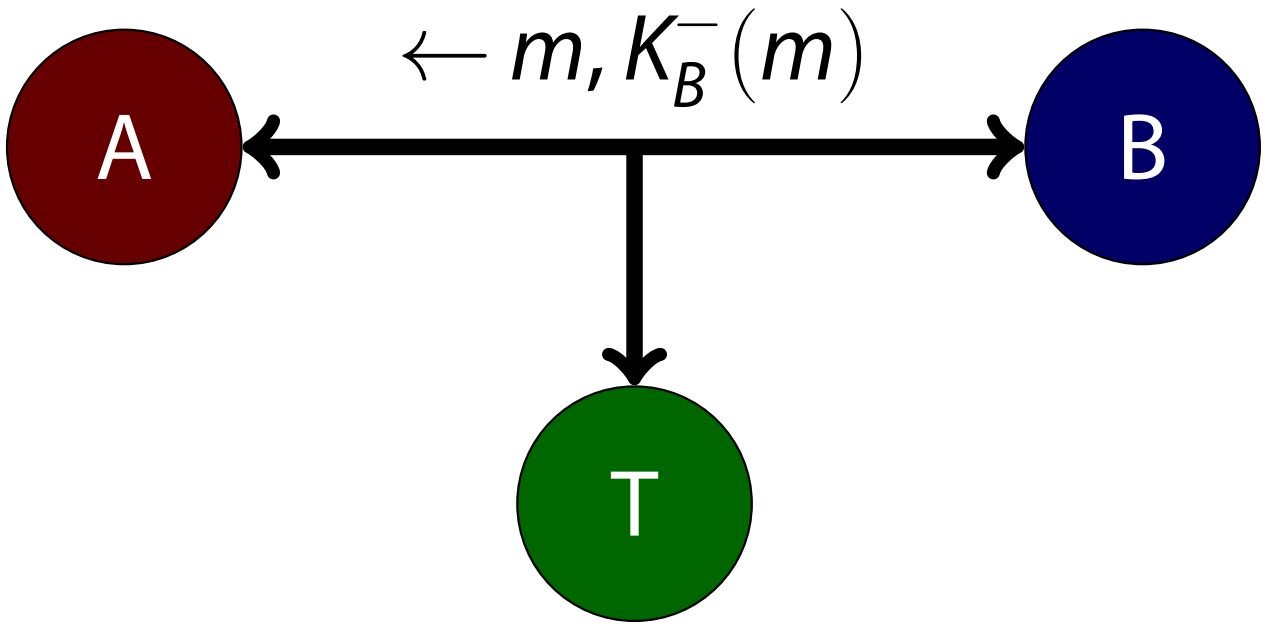


Verifying **message source** with **digital signature**.

Digital Signature
(flawed)

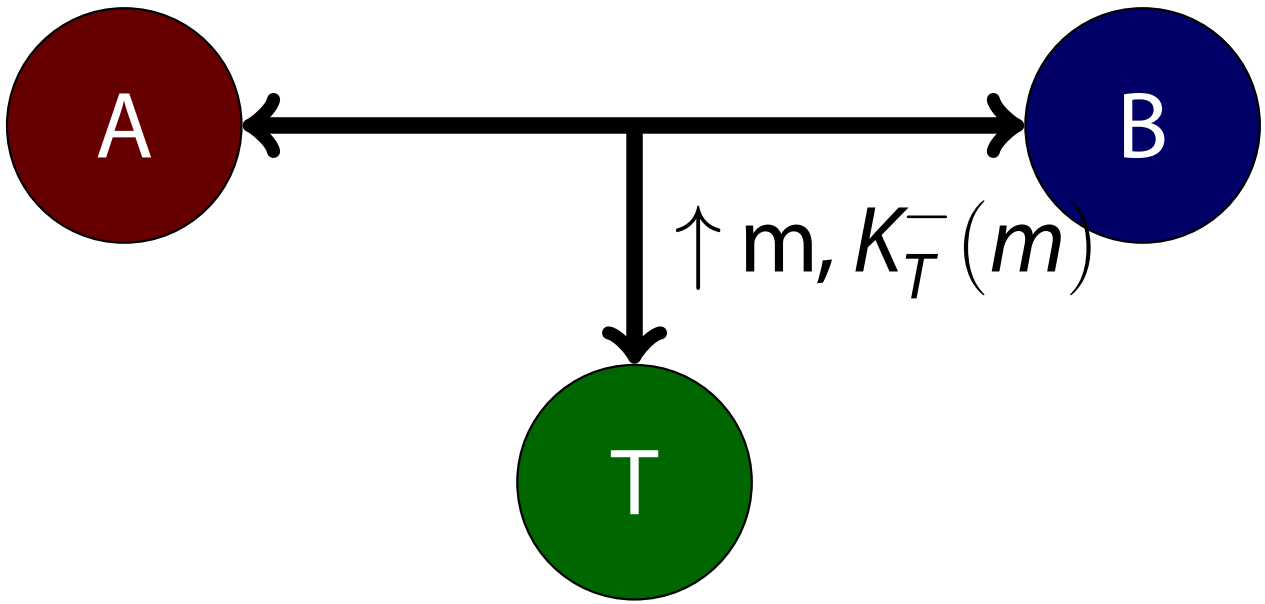


Digital Signature
(fixed with public/private key)



Digital Signature

(flawed with untrusted public key)



Digital Certificate

(public key, name)

Certificate Authority (CA)

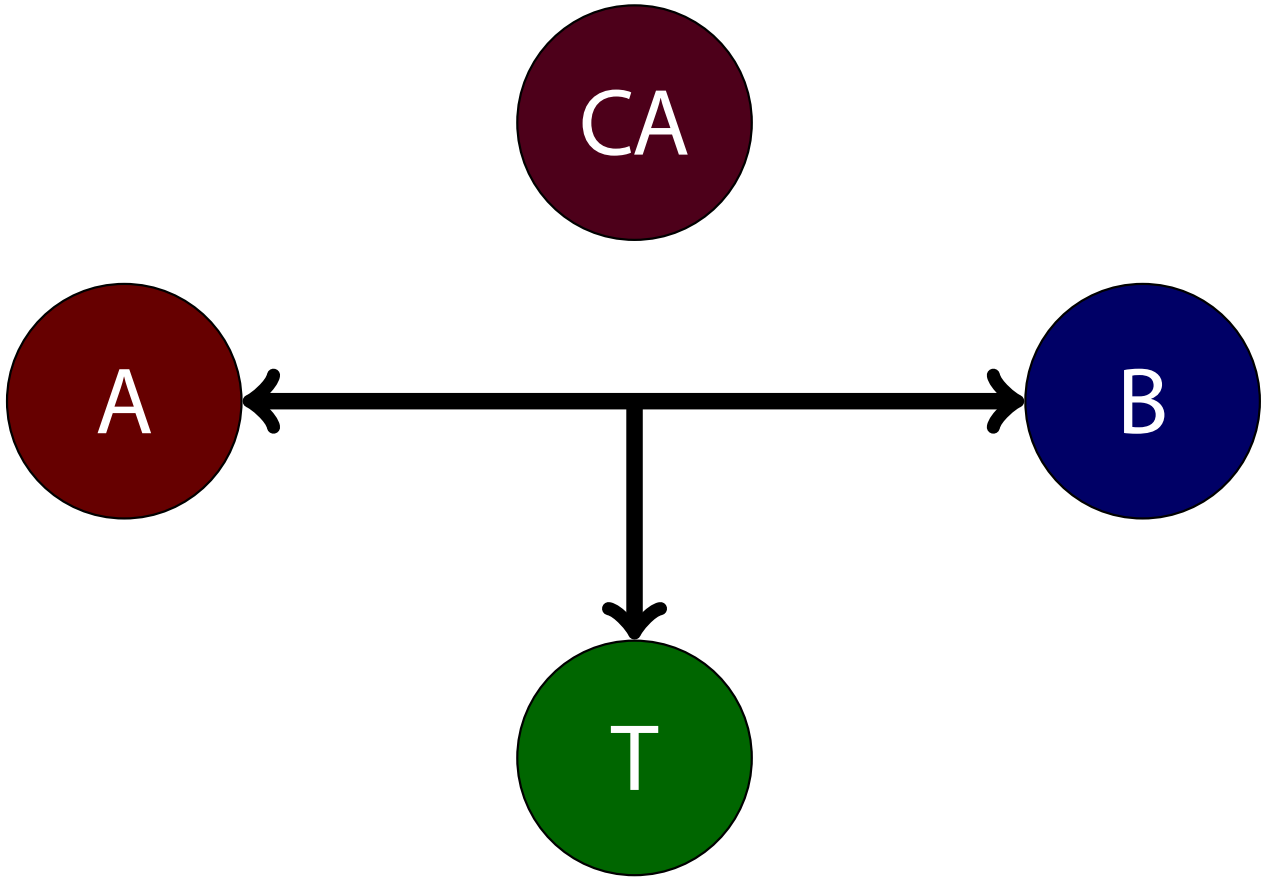
issue digitally signed digital certificates

Certificate Authority (CA)

issue digitally signed digital certificates

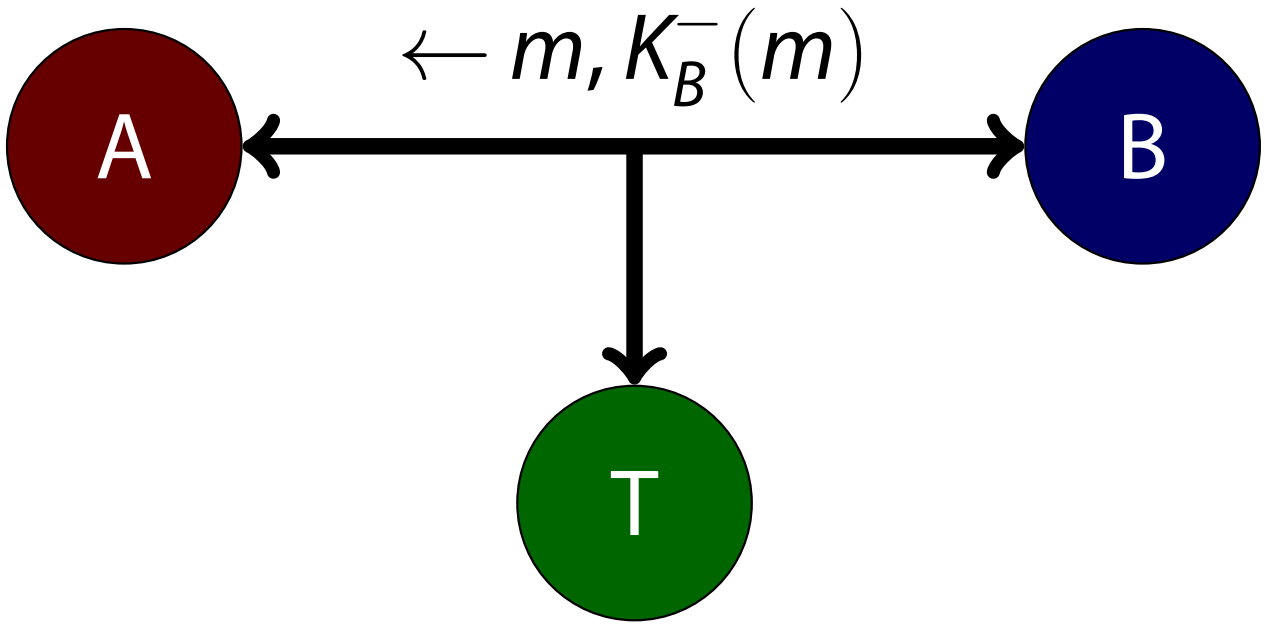
Can you trust the digital signature (and therefore the public key) of a CA? We reserve the discussion on this for Problem Set 6.

Digital Signature
(with trusted public key)

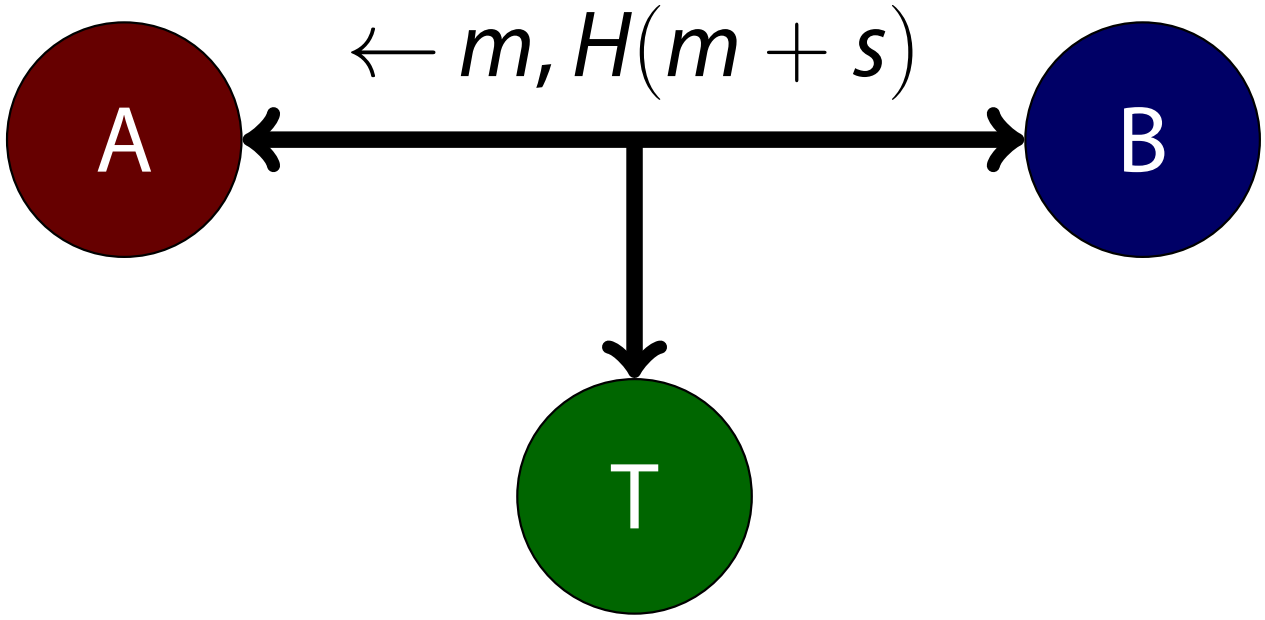


End Point Authentication

Replay Attack
(with digital signature)



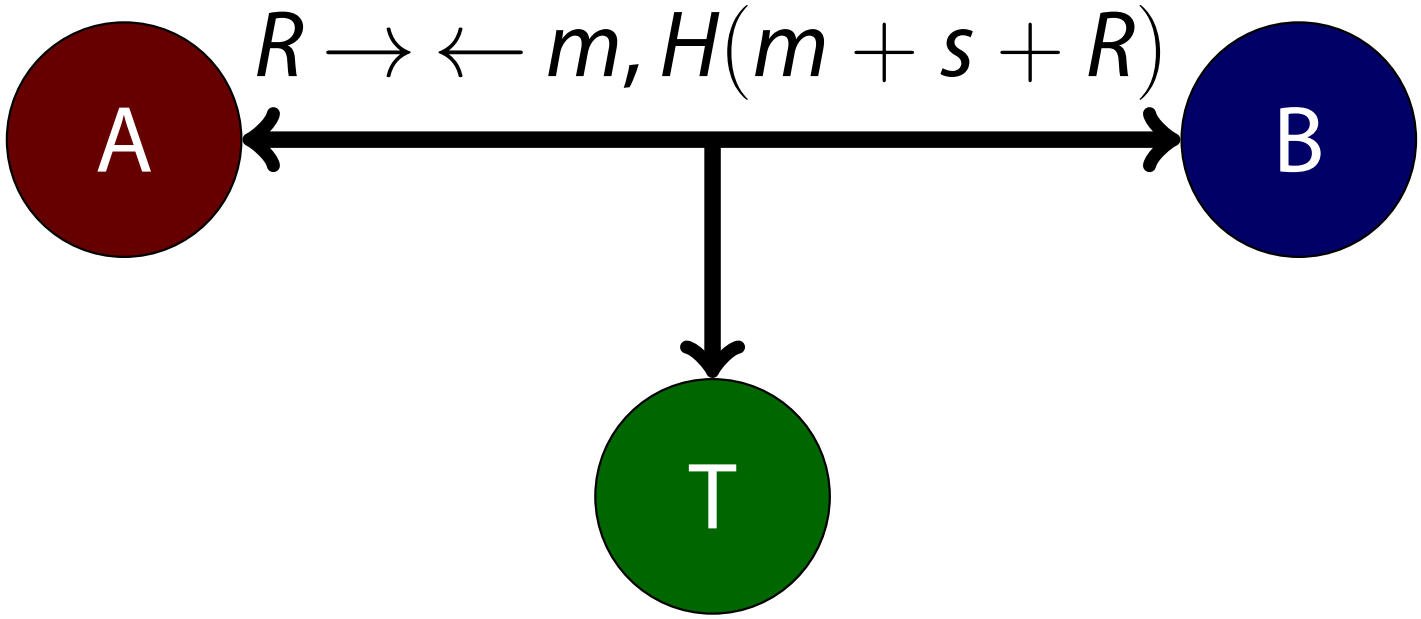
Replay Attack
(with MAC)



nonce: a number used only once

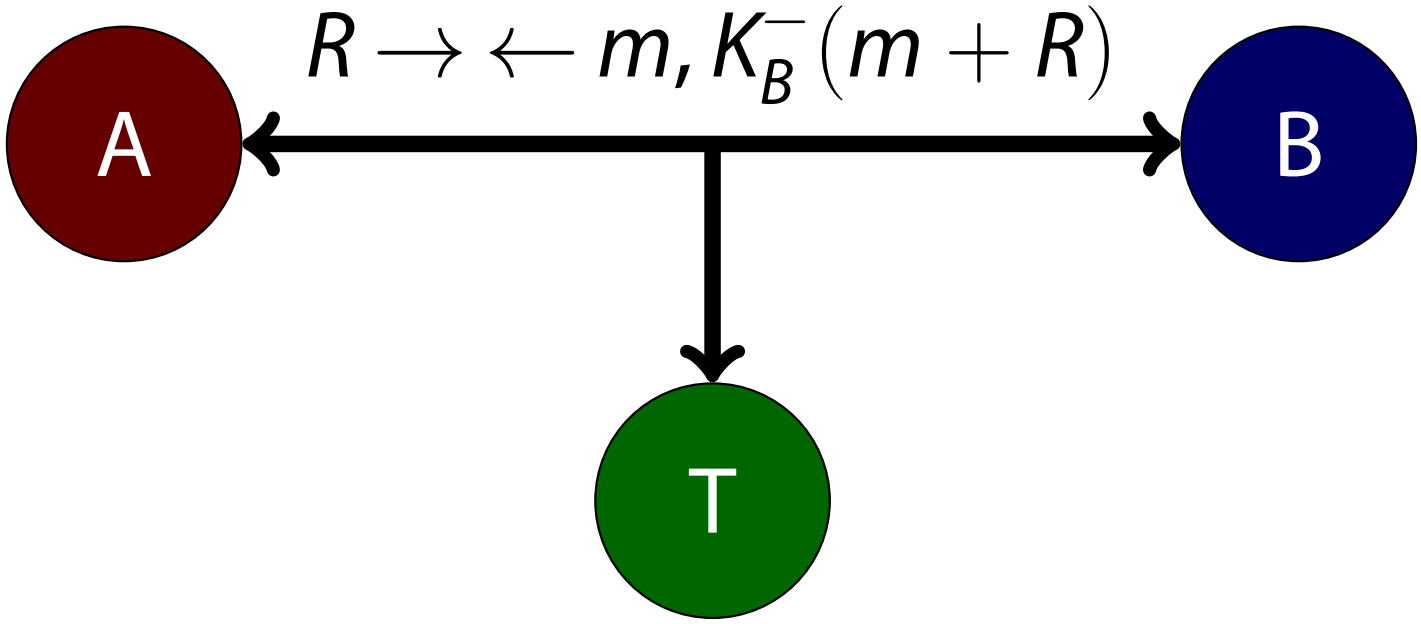
End Point Authentication

(with MAC and nonce)



End Point Authentication

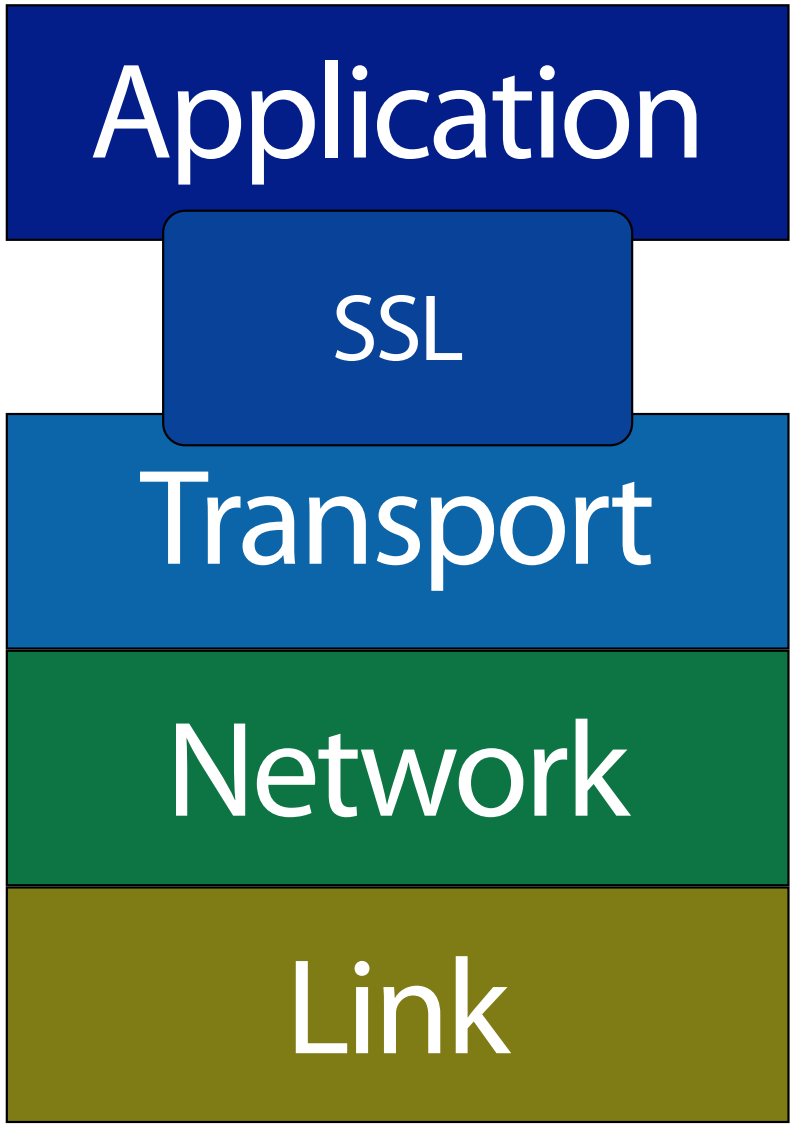
(with public key and nonce)



Putting everything together: **Securing E-mail**

Putting everything together:
Securing E-mail

Read Section 8.5 in the textbook for how these techniques are used together to send secure e-mails.


Application
SSL
Transport
Network
Link

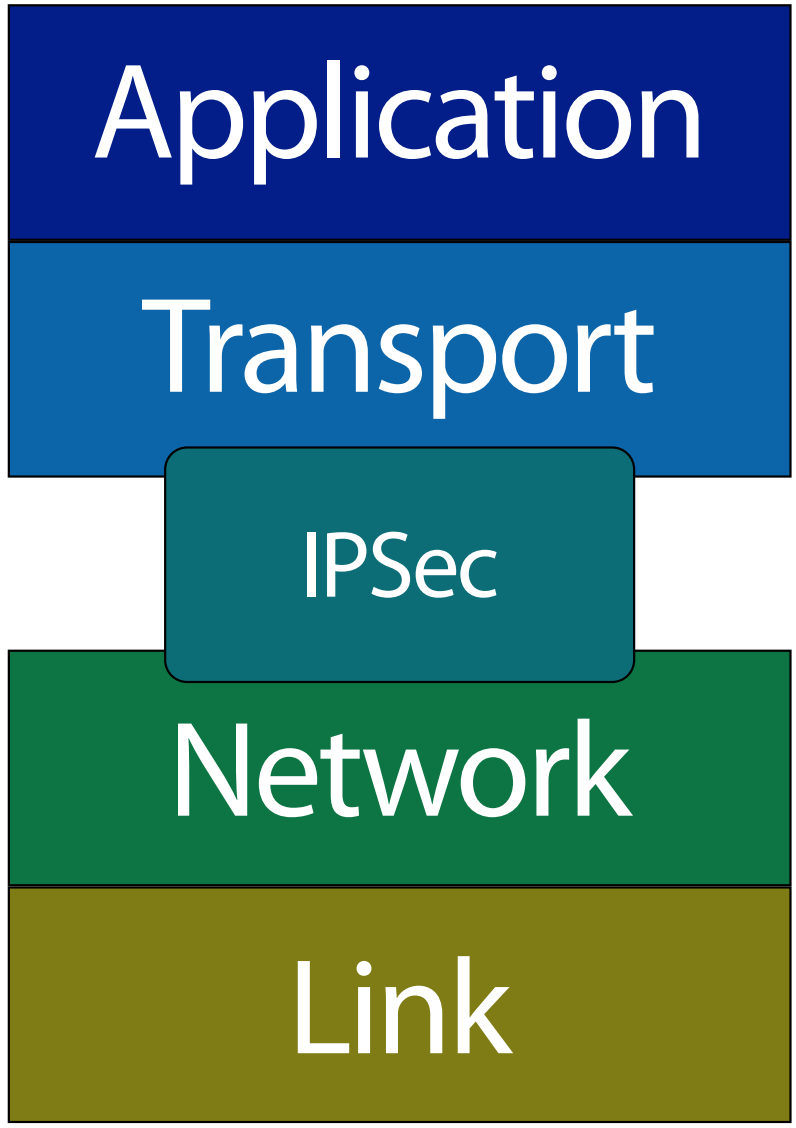
1. Establishing a shared master key between *A* and *B*

2. Derive encryption keys E_A, E_B and MAC keys M_A, M_B from the master key.

3. Transfer data m with sequence number s as m'

$$h = H(m + M_B + s)$$

$$m' = E_B(m, h)$$


Application
Transport
IPSec
Network
Link

IPSec is a suite of protocols

Authentication Header (AH) protocol: source authentication and data integrity.

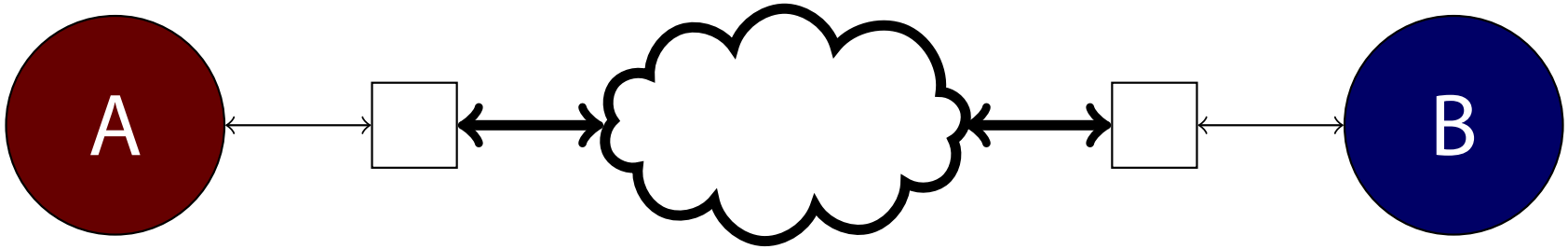
Encapsulation Security Payload (ESP) protocol: source authentication, data integrity and confidentiality.

Security Association (SA): a one-way logical connection, consisting of:

- an SA ID (Security Parameter Index)
- src and dst IP addresses
- type and key for encryption
- type and key for integrity check

$$m = E_B(\text{IP datagram} + \text{padding})$$
$$h = H(i + n + m, M_B)$$

VPN: Virtual Private Network



VPN: Virtual Private Network

VPN can run over either IPSec or SSL. We will discuss the pros and cons of these approaches in Problem Set 6.