Network Security

COMP 30650: Networks and Internet Systems

Dr. Gavin McArdle
Email: gavin.mcardle@ucd.ie
Office: A1.09 Computer Science

RECAP

- **Application Layer**
- **The Bigger Picture**
  - New Nodes switched on
  - Access content on web server
- **Intro to Security**
  - Keep messages secret and unaltered
    - Confidential
    - Authentic
    - Integrity

TODAY'S PLAN

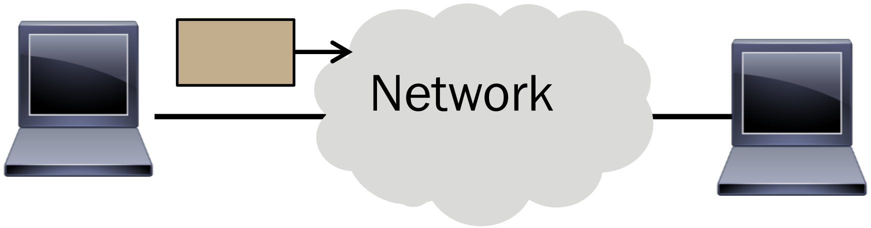
- Confidentiality
  - Encryption
  - Symmetric and Public Key Encryption
- Authentication
  - Digital Signatures
    - Hashes
- Preventing Replays

ENCRYPTION AND CONFIDENTIALITY

Encrypting information to provide confidentiality

Confidentiality: The state of keeping or being kept secret or private.

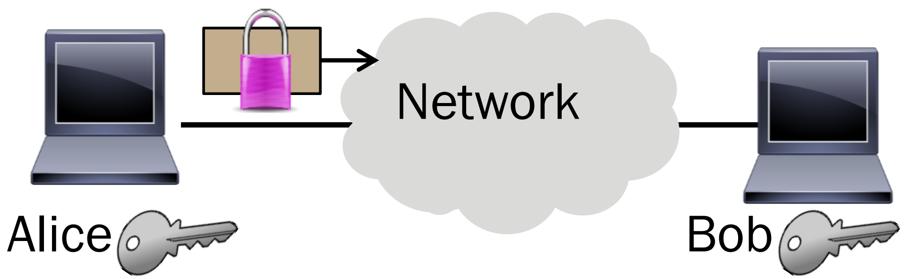
Encryption: The process of encoding a message or information in such a way that only authorized parties can access it.



ENCRYPTION AND CONFIDENTIALITY

Encrypting information to provide confidentiality

- <u>Symmetric</u> and <u>public key</u> encryption
- Treat crypto functions as black boxes
  - Function/algorithm which takes a <u>*key*</u> and the data and produces an encrypted message.



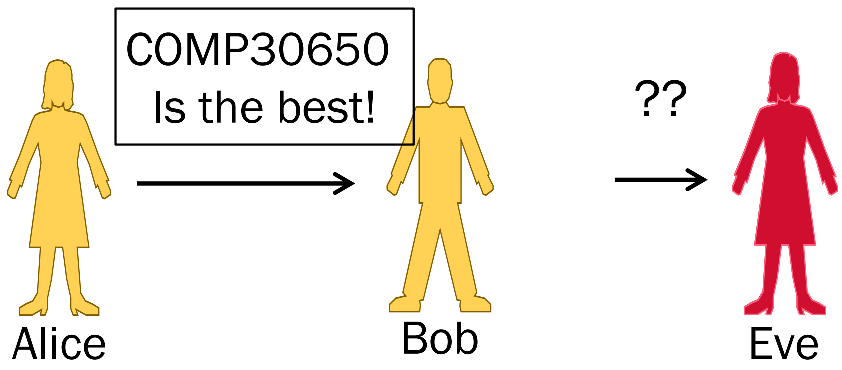
GOAL AND THREAT MODEL

Goal is to send a private message from Alice to Bob

- This is called confidentiality

Threat is Eve will read the message

- Eve is a passive adversary (observes)

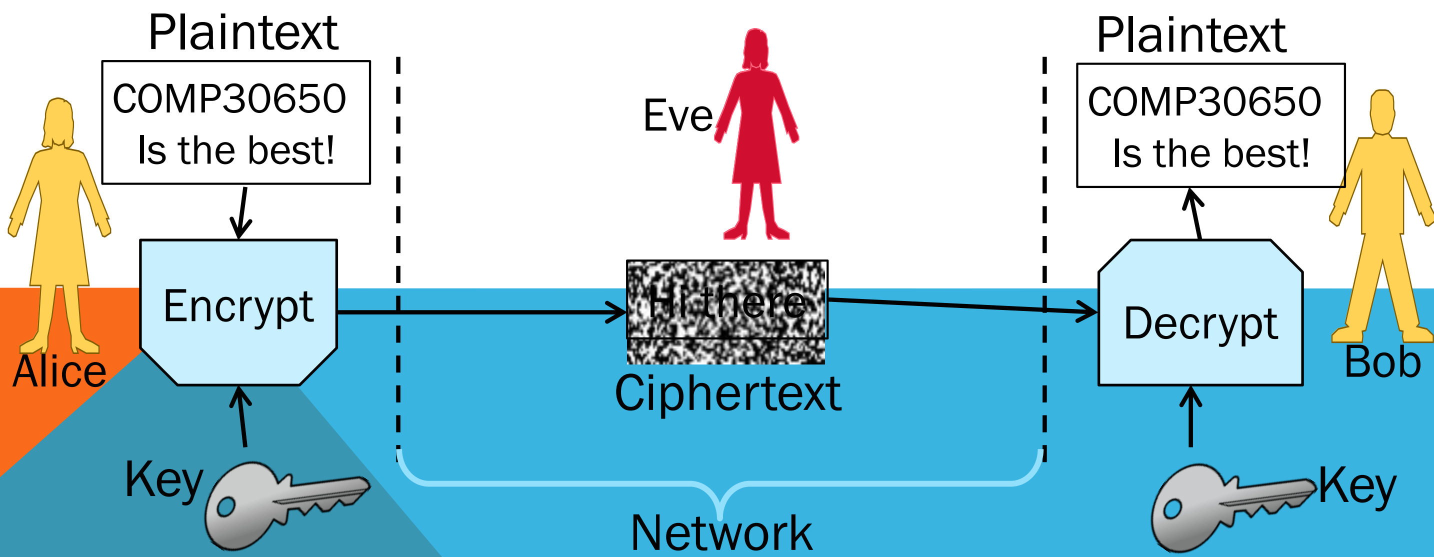


ENCRYPTION/DECRYPTION MODEL

Alice encrypts private message (<u>plaintext</u>) using key which results in a <u>ciphertext</u>.

Eve sees ciphertext but can't relate it to private message

Bob decrypts using key to obtain the private message



ENCRYPTION/DECRYPTION

Encryption is a reversible mapping

- Ciphertext is confused/jumbled plaintext

Assume attacker knows algorithm

- Security does not rely on its secrecy

Algorithm is parameterized by keys

- Security relies on key secrecy

ENCRYPTION/DECRYPTION

Encryption is a reversible mapping

- Ciphertext is confused/jumbled plaintext

Assume attacker knows algorithm

- Security does not rely on its secrecy

Algorithm is parameterized by keys

- Security relies on key secrecy
- Must be distributed!!

ENCRYPTION/DECRYPTION

Two main kinds of encryption:

1. Symmetric key encryption », e.g., AES (Advanced Encryption Standard)

- Alice and Bob share same secret key
- Encryption is a mangling box based on data and key.
- Decryption is a mangling box based on data and key.

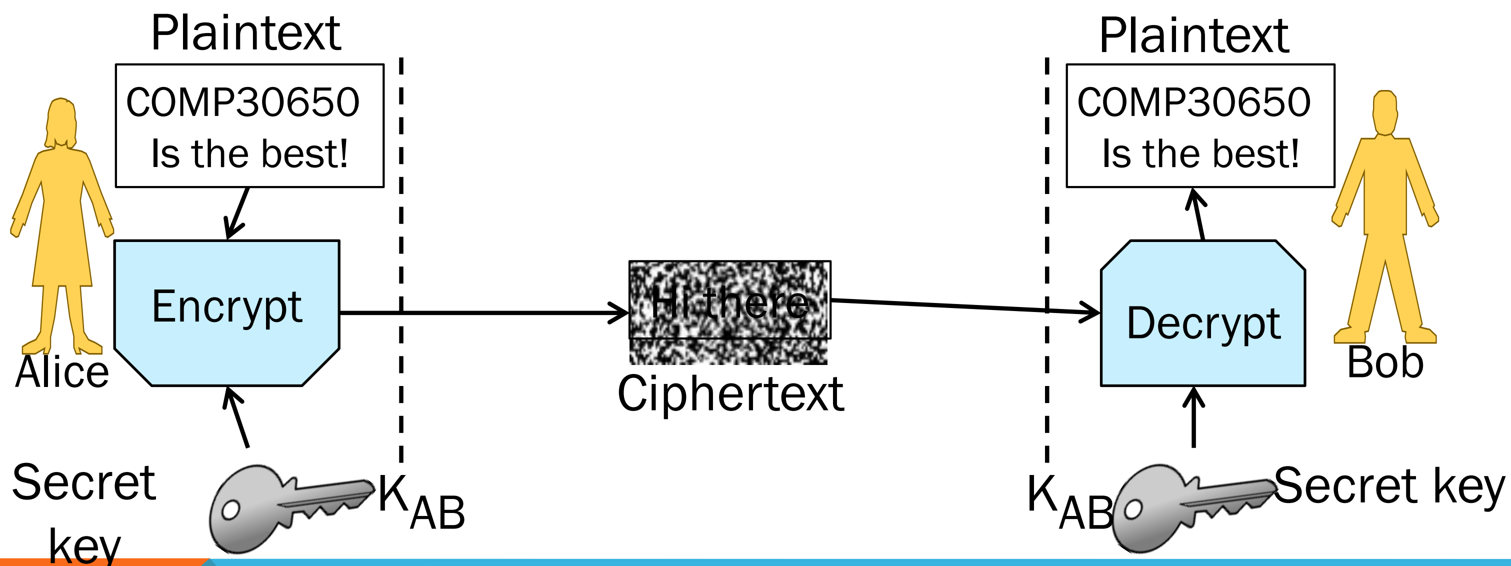
2. Public key encryption », e.g., RSA (Rivest–Shamir–Adleman)

- Alice and Bob each have a key in two parts: a public part (widely known), and a private part (only owner knows)
- Encryption is based on mathematics (e.g., RSA is based on difficulty of factoring)

SYMMETRIC (SECRET KEY) ENCRYPTION

Alice and Bob have the same secret key, K_{AB}

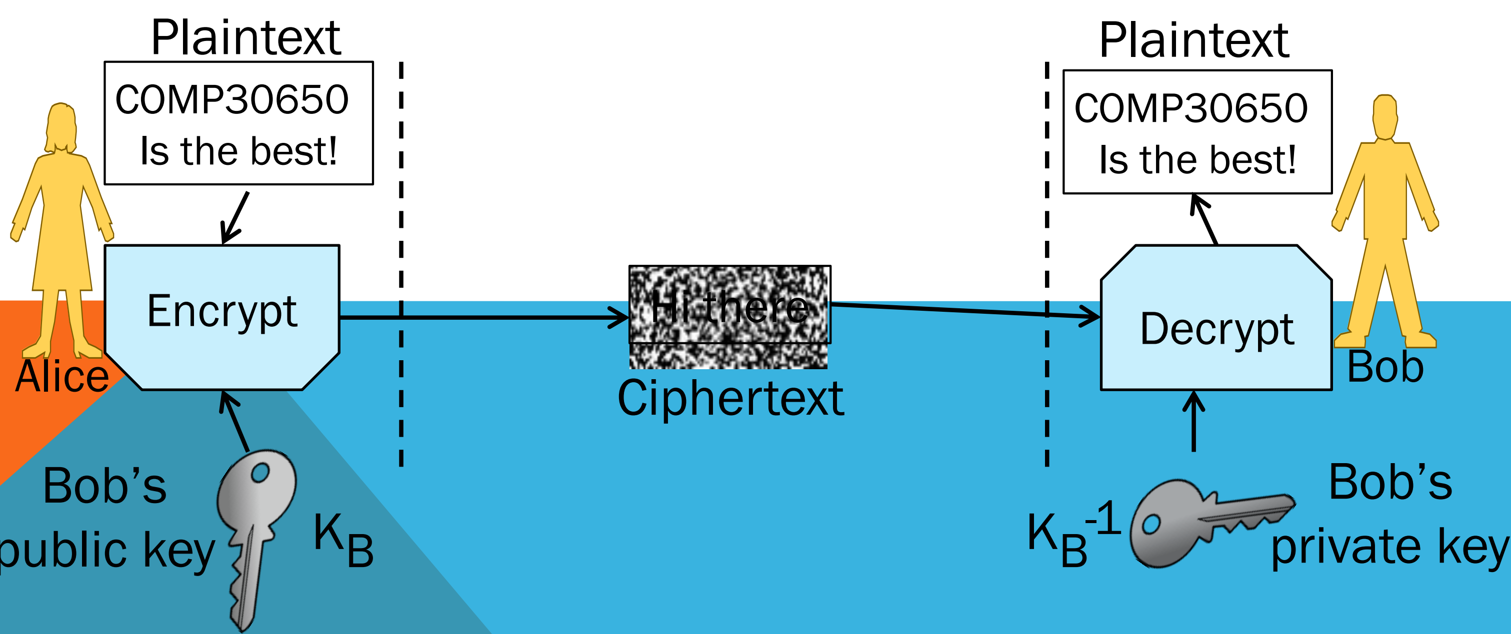
- Anyone with the secret key can encrypt/decrypt



PUBLIC KEY (ASYMMETRIC) ENCRYPTION

Alice and Bob each have public/private key pair (K_B/ K_B^{-1})

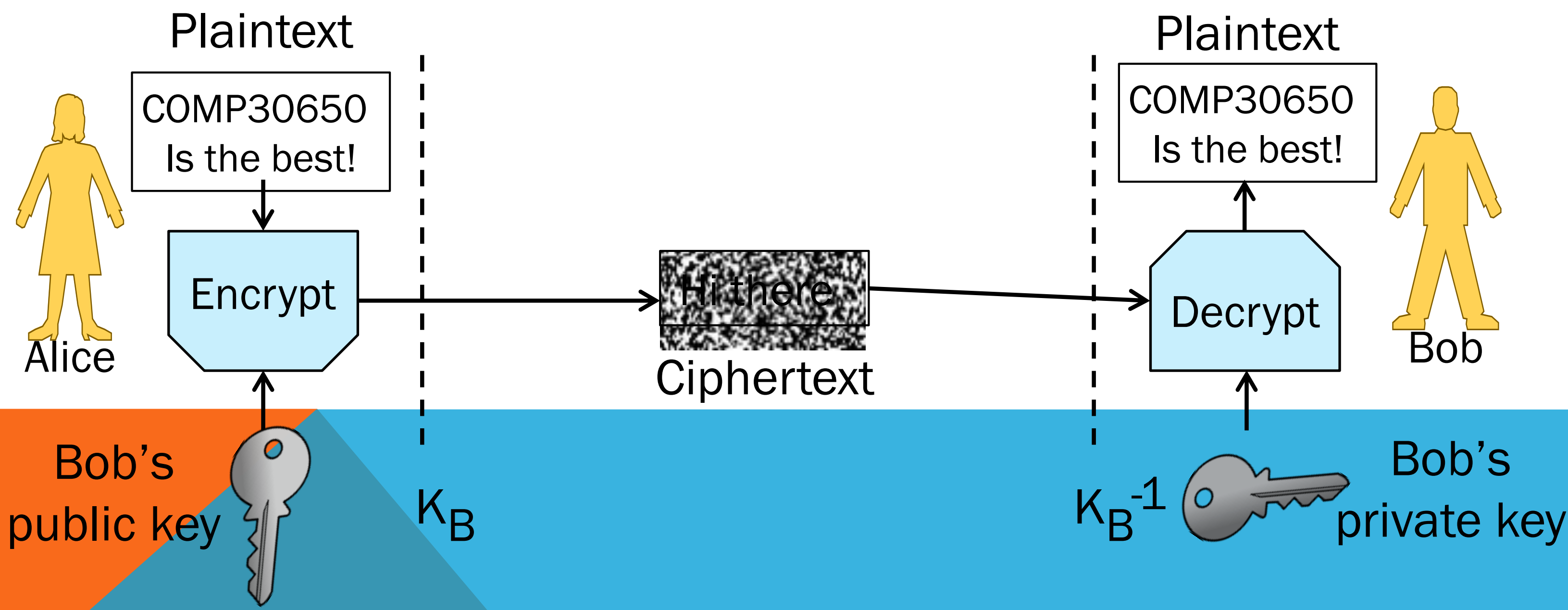
- Public keys are well-known, private keys are secret to owner
- Private key is used to decrypt messaged which are encrypted with the public key.



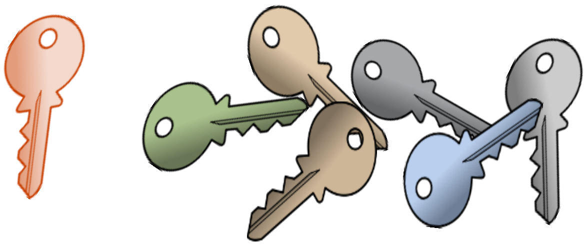
PUBLIC KEY ENCRYPTION

Alice encrypts with Bob's public key K_B; anyone can do this

Bob decrypts with his private key K_B^{-1}; only he can do this



KEY DISTRIBUTION



This is a big problem on a network!

- Often want to talk to new parties

Symmetric encryption is problematic

- Have to first set up shared secret

Public key idea has other difficulties

- Need trusted directory service
  - Is Bob’s public key really Bob’s public key?
- Certificates help with this

SYMMETRIC VS. PUBLIC KEY

Have complementary properties

- Want the best of both!

Property	Symmetric	Public Key
Key Distribution	Hard–share secret per pair of users	Easier– publish public key per user
Runtime Performance	Fast– good for high data rate	Slow– few, small, messages

WINNING COMBINATION

Alice uses public key encryption to send Bob a small private message

- It's a key!

Alice and Bob send large messages with symmetric encryption

- Using the key they now share

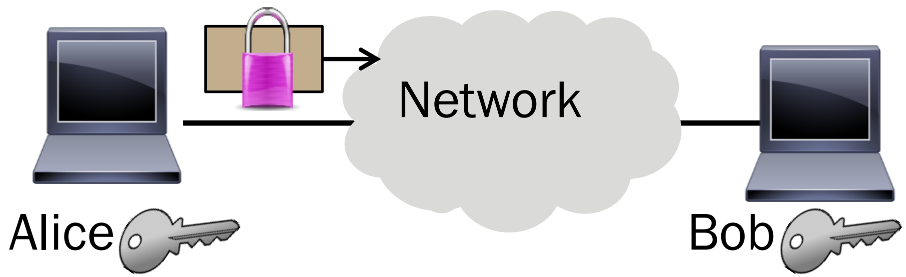
The key is called a session key

- Generated for short-term use

AUTHENTICATION AND INTEGRITY

Encrypting information to provide authenticity (=correct sender) and integrity (=unaltered)

- Confidentiality isn't enough



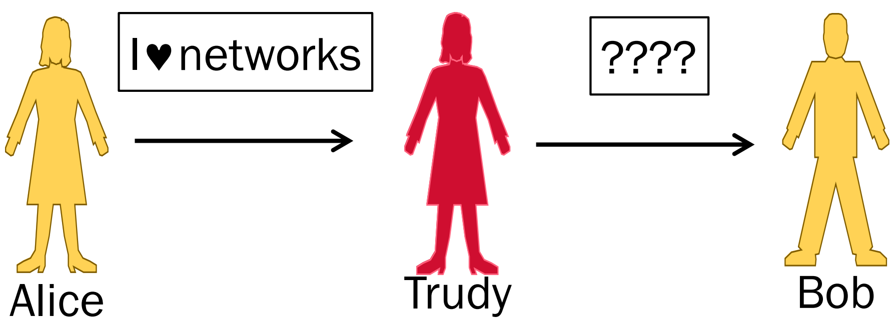
GOAL AND THREAT MODEL

Goal is to let Bob verify the message came from Alice and is unchanged

- This is called integrity/authenticity

Threat is Trudy will tamper with messages

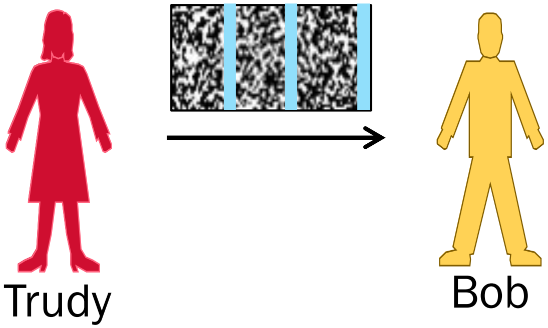
- Trudy is an active adversary (interferes)



ENCRYPTION IS NOT ENOUGH

What will happen if Trudy flips some of Alice's message bits?

- Bob will decrypt it and get the message back but its likely to be garbage….

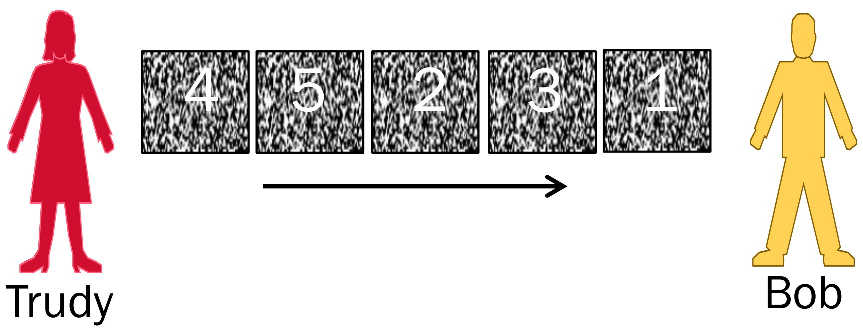


ENCRYPTION IS NOT ENOUGH

Typically encrypt blocks of data

What if Trudy reorders message?

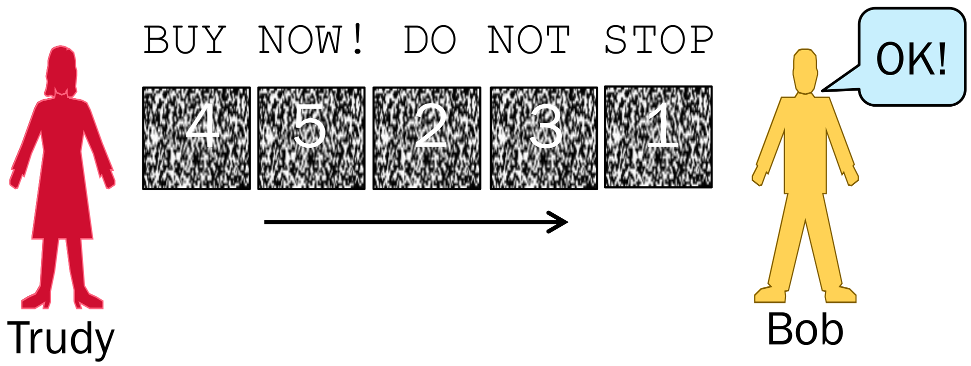
- Bob will decrypt, and …



ENCRYPTION ISSUES

What if Trudy reorders message?

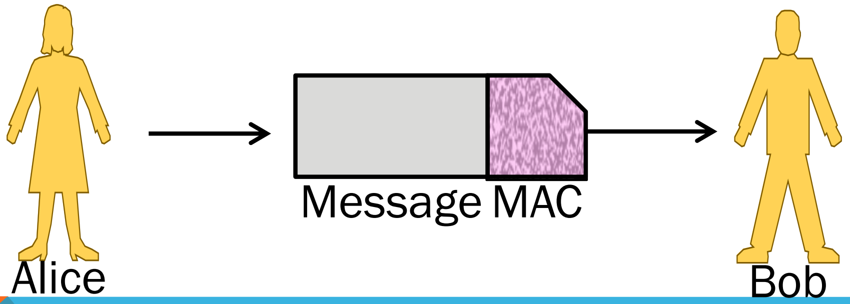
- Bob will receive altered message



MAC (MESSAGE AUTHENTICATION CODE)

MAC is a small token to validate the integrity/authenticity of a message

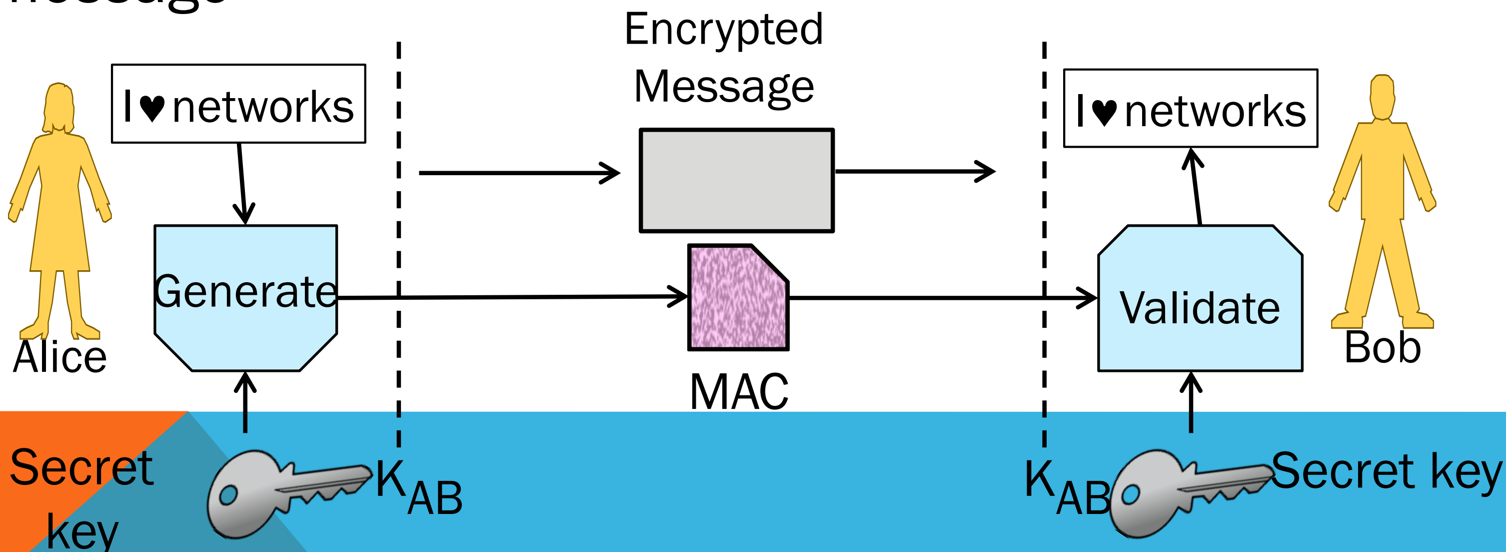
- Send the MAC along with message
- Validate MAC, process the message
- Example: HMAC scheme
- Detect Changes in the cipher text/message.



MAC

A symmetric encryption operation – key is shared

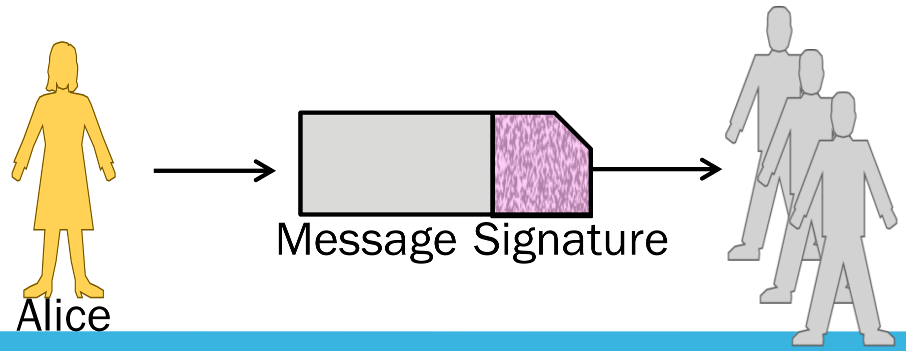
- Lets Bob validate unaltered message came from Alice
- Doesn't let Bob convince Charlie that Alice sent the message



DIGITAL SIGNATURE

Signature validates the integrity/ authenticity of a message

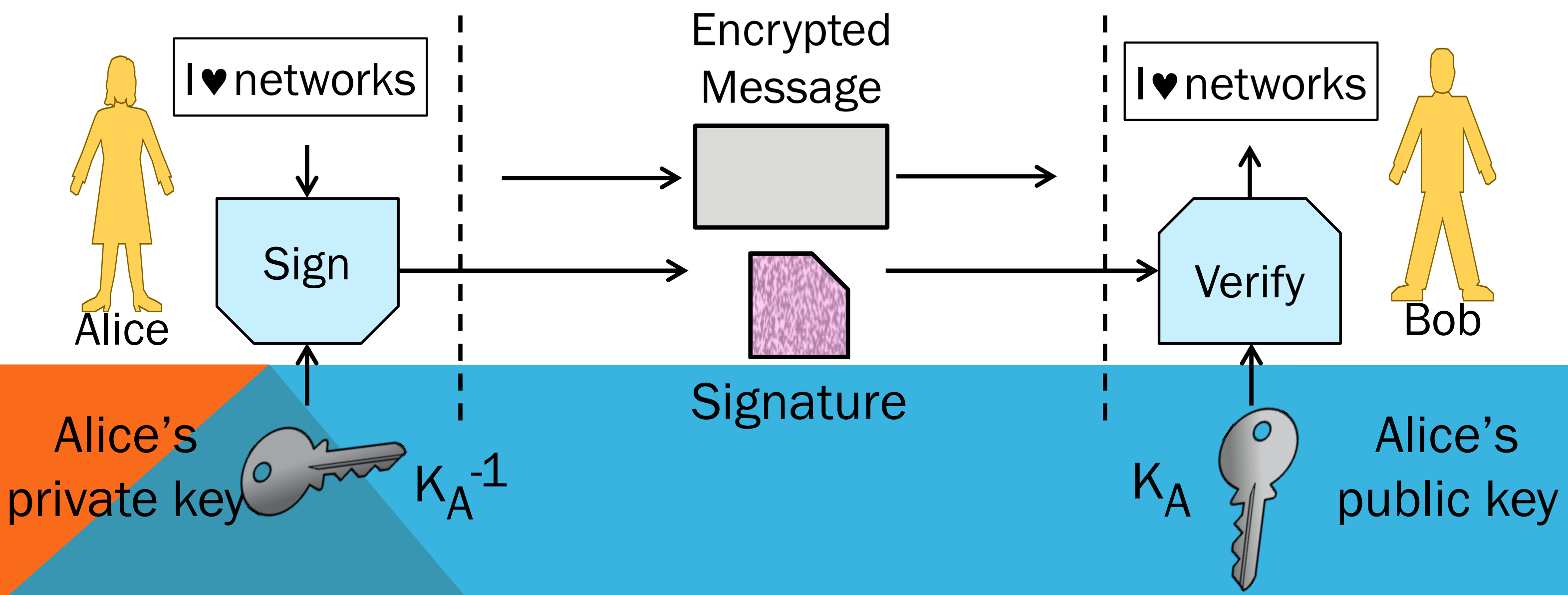
- Send it along with the message
- Lets all parties validate
- Example: RSA signatures



DIGITAL SIGNATURE (2)

Public key operation – public/private key parts

- Alice signs with private key, K_A^{-1}, Bob verifies with public key, K_A
- Does let Bob convince Charlie that Alice sent the message



SPEEDING UP SIGNATURES

Same tension as for confidentiality:

- Public key has keying advantages
- But it has slow performance!

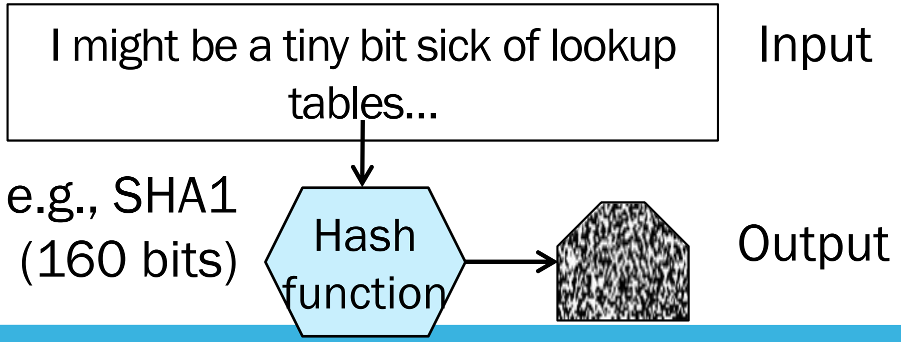
Use a technique to speed it up

- <u>Message digest</u> stands for message
- Sign the digest instead of full message

MESSAGE DIGEST OR CRYPTOGRAPHIC HASH

Digest/Hash is a secure checksum

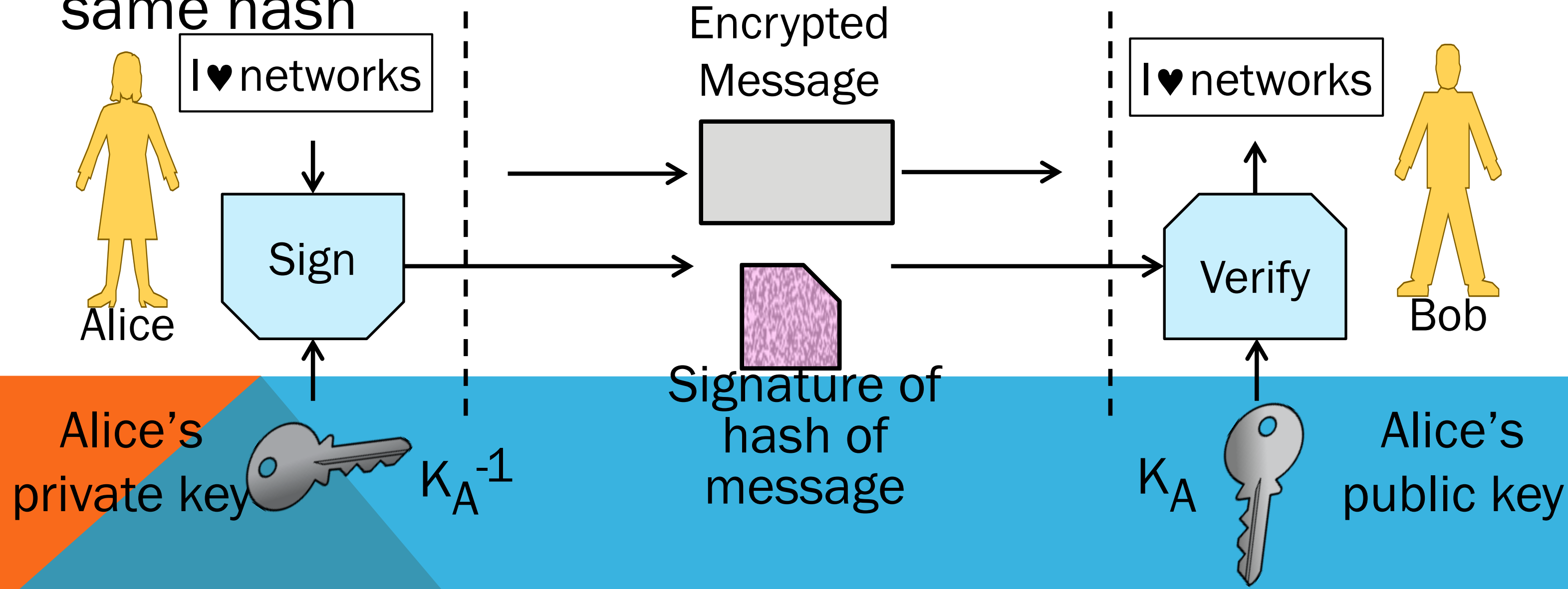
- Deterministically mangles bits to pseudo-random output (like CRC)
- Can't find messages with same hash
- Acts as a fixed-length descriptor of message – very useful!



SPEEDING UP SIGNATURES

Conceptually as before except sign the hash of message

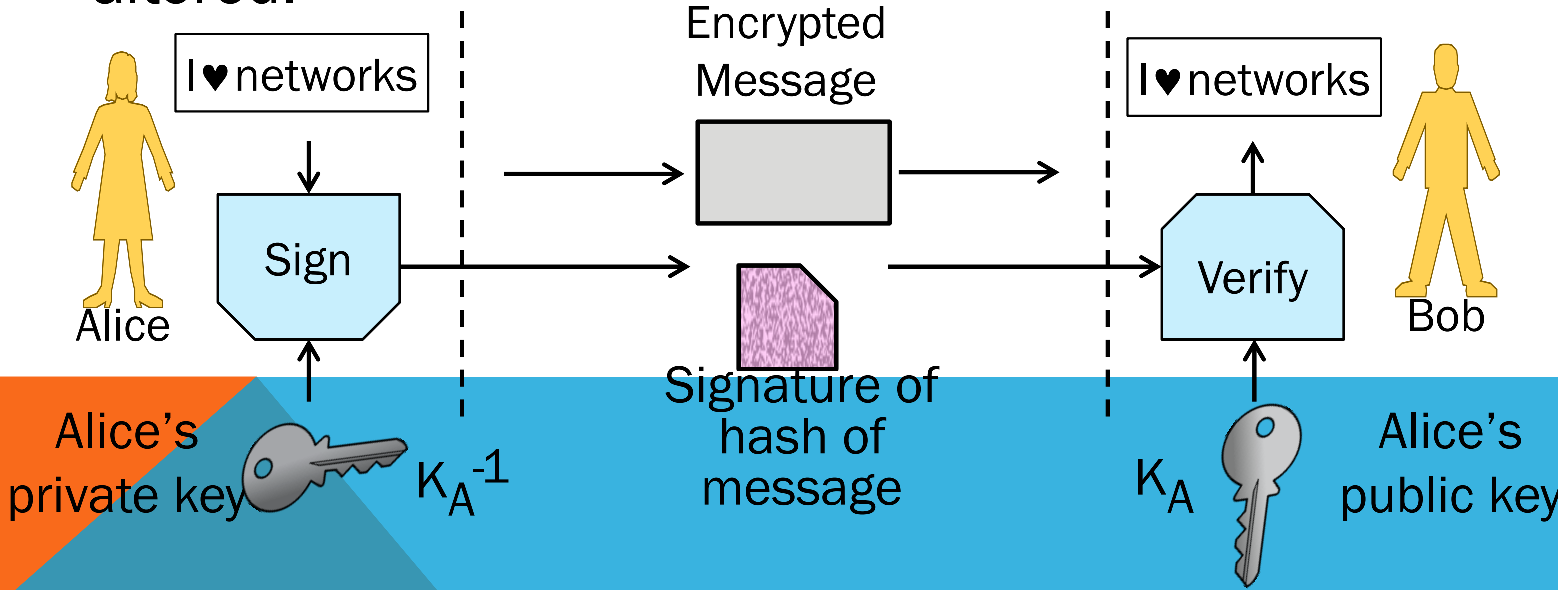
- Hash is fast to compute, so it speeds up overall operation
- Hash stands for message as can't find another with same hash



SPEEDING UP SIGNATURES

Sending: Hash is generated from the message and signed with Alice's private key and sent along with the encrypted message.

Receiving: Bob generates the hash from the message and applies Alice's public key to verify it has not been altered.



PREVENTING REPLAYS

We normally want more than confidentiality, integrity, and authenticity for secure messages!

- Want to be sure message is fresh

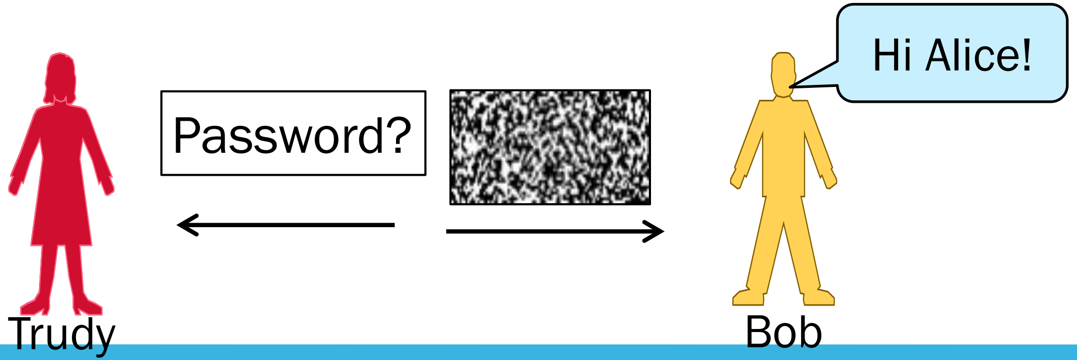
Don’t want to mistake old message for a new one – a <u>replay</u>

- Acting on it again may cause trouble

PREVENTING REPLAYS

Replay attack:

- Trudy records Alice's messages to Bob
- Trudy later replays them (unread) to Bob; she pretends to be Alice



PREVENTING REPLAYS

To prevent replays, include proof of freshness in messages

- Use a timestamp, or nonce