Wireless Network Security Overview

Irfan Ali

Objective

- *Understand the threats against mobile/wireless networks*
- *Identify the components of designing a secure network architecture*
- *Learn specifics of LTE security*

Why security?

- *"Network" is a resource that requires protection against a wide range of threats*
  - Service theft
  - Denial-of service attack on the whole network or a selected victim
  - Privacy violation (information theft)
  - Leveraging the network to launch attacks on other parties
  - …

“Mobile/Wireless” Challenge

- *Physical security has very limited applicability*
  - Heavy dependency on cryptographic security
- *Mobility means involved parties are changing frequently*
- *Mobile devices are constrained*
  - Low computational power, basic user, basic UI.
- *Performance is important*
  - Real-time traffic needs to be respected.

What is Security?

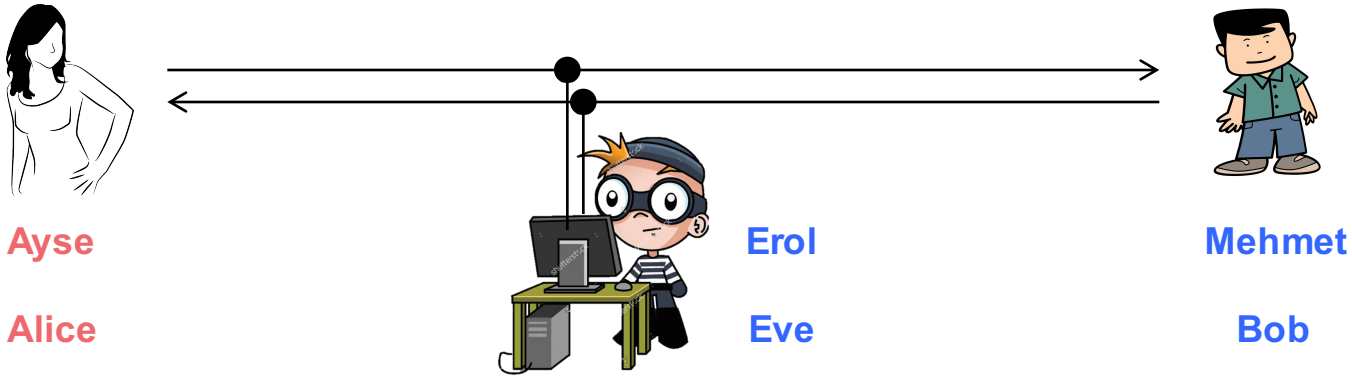
- *Security is about achieving some **goals** when there are **adversaries** present.*
- *There are three components of security:*
  - **Policy**: Statement of what needs to be achieved
  - **Threat model**: what can the adversary do?
  - **Mechanisms**: How to protect against the threat.

Goals/Policy

- *Entity Authentication*
  - Verify who you are. Typically the you in the "identity", eg. IMSI or username: bob@yahoo.com or web server: google.com
- *Message Authentication or Data Origin Authentication*
  - Ensuring that the received data came from claimed sender
- *Message Integrity Protection*
  - No one change the message without the receiver noticing this
- *Confidentiality*
  - Only source and intended receiver can understand the message
- *Authorization*
  - Allowing access only to resources (eg. files, services) you are entitled to access.
- *Privacy*
  - Only authorized people can see my identity.
- *Availability (System Level)*
  - Should prevent malicious users from blocking legitimate users from accessing the resources
- *Non-repudiation*
  - Provide undeniable evidence that message is from sender. Stronger form of data origin authentication.

Threat Model

- *Man-in-the-middle attack*



- Listen
- Delay
- Delete
- Modify
- Create new message
- Replay message
- Change the sequence of messages

Mechanisms

- *System, software or hardware to counteract the threat model and achieve goals.*
- *Cryptography!*

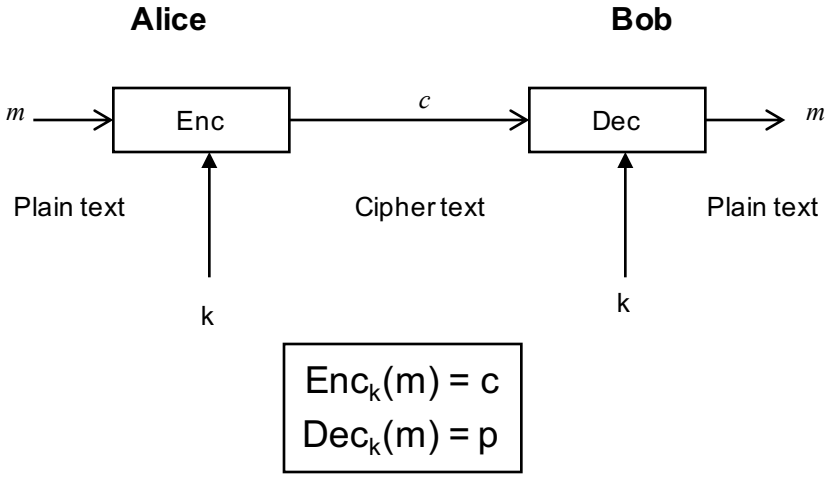
Cryptography

- *Mathematical techniques related to aspects of security such as*
  - Entity Authentication
  - Message Authentication
  - Confidentiality
  - Data Integrity
- *In the next few slides, we will look at mechanisms for*
  - Confidentiality, and
  - Message Integrity protection
  - Message authentication or Data Origin Authentication

Key Cryptographic Methods

- *Two cryptographic Methods:*
  - **Symmetric key**: uses same key at both ends (shared key)
    - Also called Private key scheme
  - **Asymmetric key**: uses two different keys (public and secret keys)
    - Also called Public Key Scheme
- *Another tool used with the above is:*
  - **Hash function**: One way transformation, used for digital signature generation.

Private Key Encryption



- A *private-key encryption scheme* is defined by a message space **M** and algorithms (Gen, Enc, Dec):
  - Gen (key-generation algorithm): generates k
  - Enc (encryption algorithm): takes key k and message $m \in \mathbf{M}$ as input; outputs ciphertext c.

    $c \leftarrow Enc_k(m)$
  - Dec (decryption algorithm): takes key k and ciphertext c as input; outputs m.

    $m := Dec_k(c)$

Perfect Secrecy

Perfect secrecy (informal)

- "Regardless of any *prior* info. the attacker has about the plaintext, the ciphertext should leak no *additional* information about the plaintext"
  - (Ciphertext-only attack, one ciphertext)

Perfect Secrecy

Perfect secrecy (informal)

- Attacker's information about the plaintext = attacker-known *distribution* of the plaintext
  - Perfect secrecy means that observing the ciphertext should not change the attacker's knowledge about the distribution of the plaintext

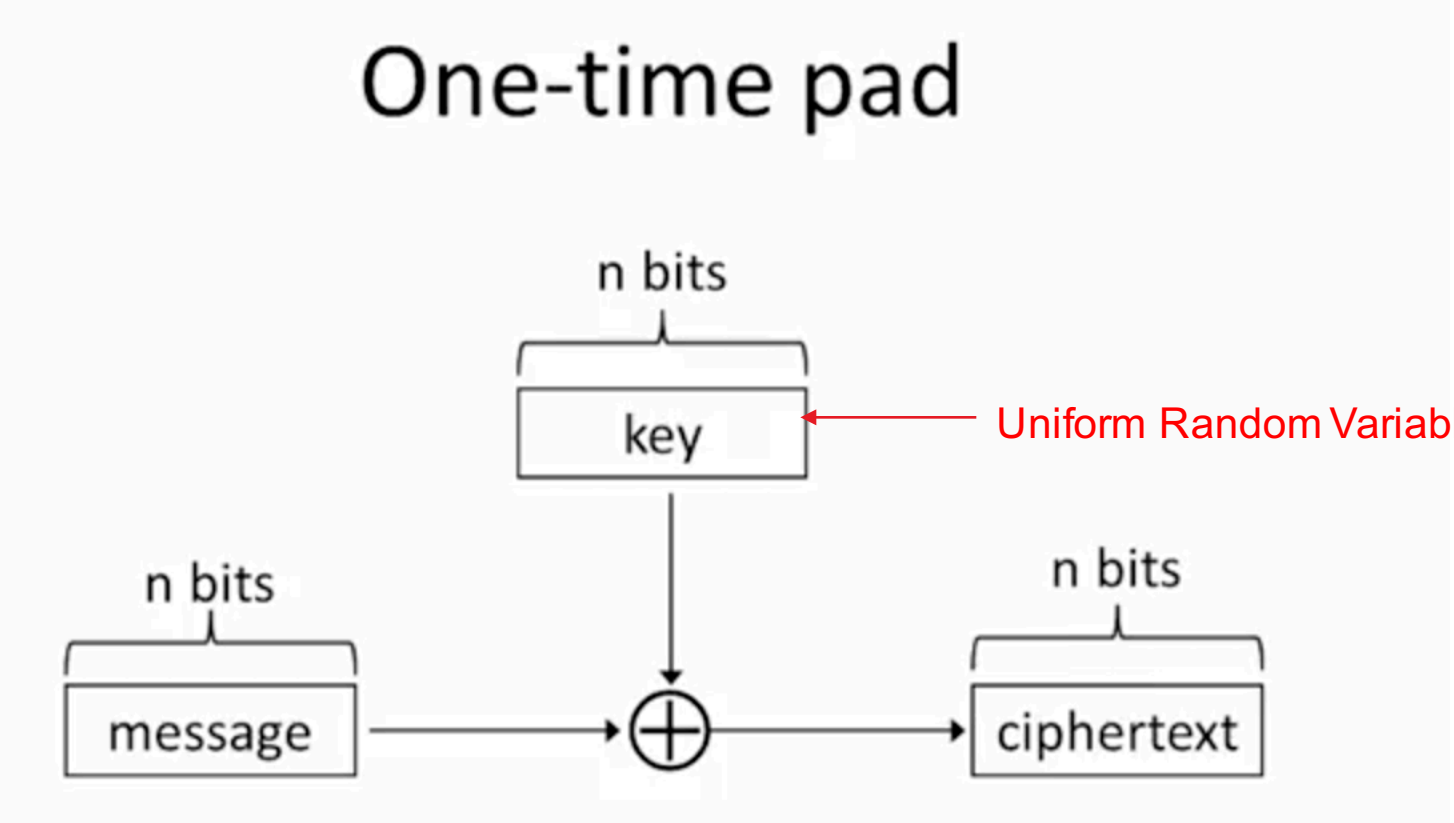
Perfect Secrecy

Perfect secrecy (formal)

- Encryption scheme (Gen, Enc, Dec) with message space **M** and ciphertext space **C** is *perfectly secret* if for every distribution over **M**, every $m \in \mathbf{M}$, and every $c \in \mathbf{C}$ with $Pr[C=c] > 0$, it holds that

$$Pr[M = m \mid C = c] = Pr[M = m].$$

One-Time Pad


One-time pad
n bits
key
Uniform Random Variable
n bits
message
n bits
ciphertext

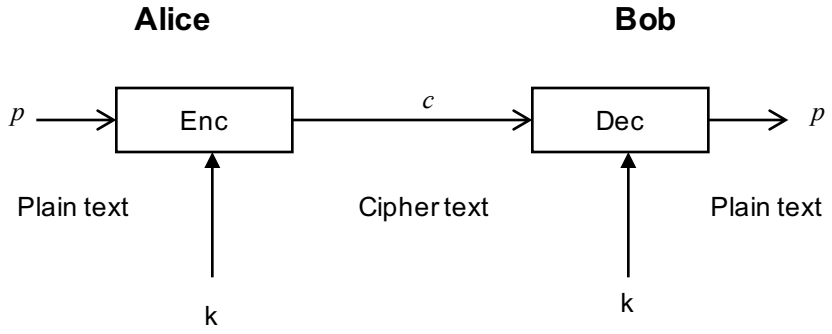
Shannon's Perfect Secrecy Theorem

- *Let (Gen, Enc, Dec) be an encryption scheme over message space M for which |M| = |K| = |C|. This scheme is perfectly secret if and only if:*
  1. Every key k belonging to K is chosen with equal probability 1/|K| by algorithm Gen
  2. For every m belonging to M and every c belonging to C, there exists a single key k belonging to K such that $Enc_k(m) = c$

The bad news …

Thm: perfect secrecy $\Rightarrow$ $|\mathcal{K}| \geq |\mathcal{M}|$

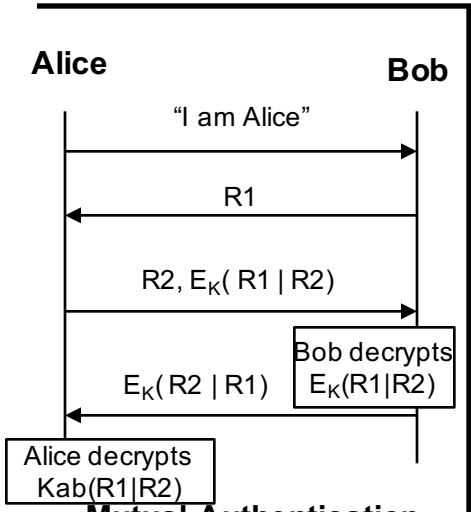
Symmetric Key Cryptography: Encryption and Message Authentication



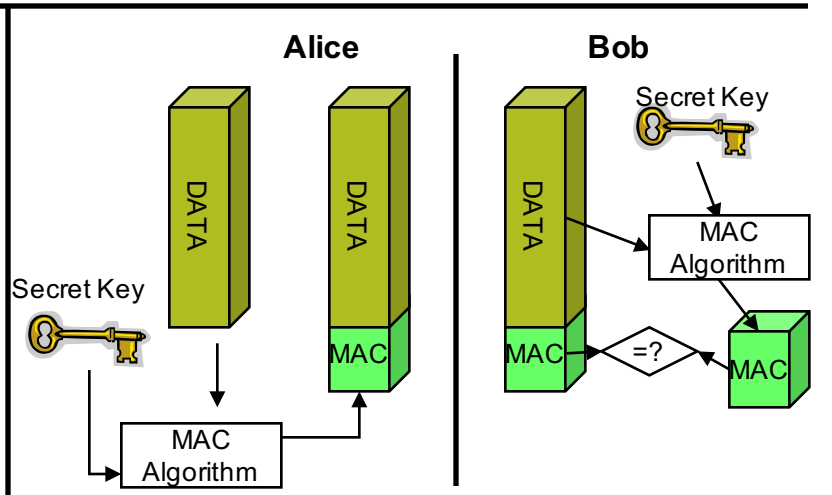
$Enc_k(p) = c$

$Dec_k(c) = p$

- *Secret key does not allow for "non-repudiation", i.e for the sending party to not-deny that it has sent the message*
  - In secret key, Alice (sender) can say that Bob (destination) create the encrypted message on his own using the common known algorithm and key.
  - For non-repudiation to be provided, we need to prove that no one except the sender could have generated the message, which is not possible when symmetric key is used.



Mutual Authentication with Shared Secret Key



Message Authentication with Shared Secret Key

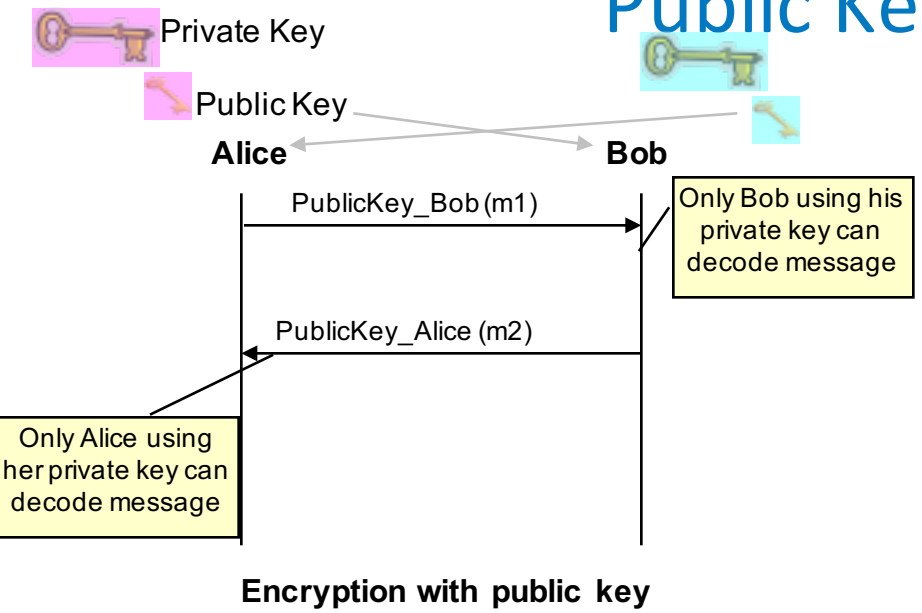
Symmetric Key Cryptography: Limitation

- *Sharing of keys needs to be done out-of-band*
- *Does not scale.*
- *Cannot provide non-repudiation.*

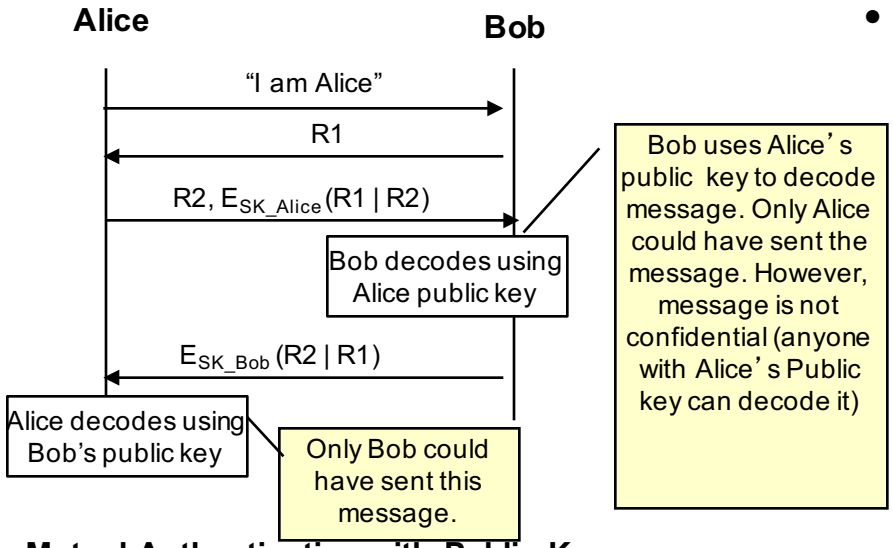
Public Key Cryptography

- *There are a pair of Keys (PK, SK):*
  - Public key PK, which is known to everyone
  - Secret key SK, which is known only to the source
  - You cannot derive the PK from SK and vice-versa
  - Plain text encoded with PK can be decoded with SK
    - $Enc_{PK}(p) = c$; $Dec_{SK}(c) = p$
  - Plain text encoded with SK can be decoded with PK
    - $Enc_{SK}(p) = c$; $Dec_{PK}(c) = p$
  - Sign/Verify
    - $Sign_{SK}(m) = s$; $Verify_{PK}(m,s) = OK?$

Public Key Cryptography



Encryption with public key

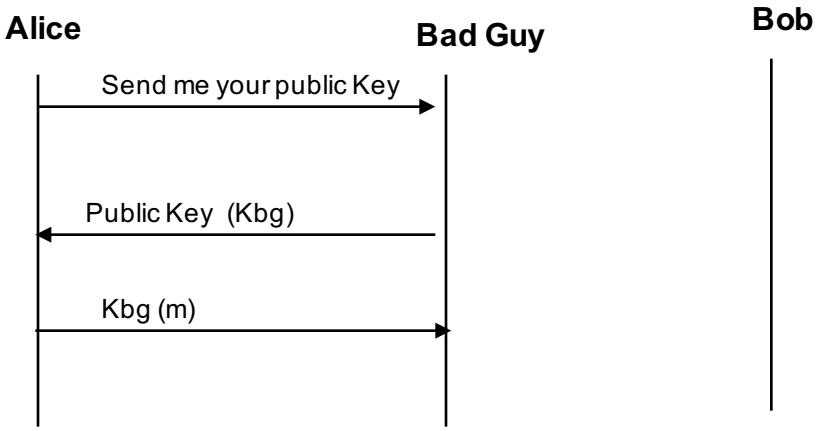


Mutual Authentication with Public Key

- *Uses two keys*
  - Public Key
  - Secret Key – known to **only one** entity
- *Confidentiality:*
  - Public key used for encryption and private for decryption
- *Authentication:*
  - Private key used for encryption and public key for decryption
- Based on the fact that message encrypted with key k1 can be decrypted only with key k2 and k2 cannot be derived from k1.
  - Well-known algorithm is the Rivest Shamir and Alderman (RSA) algorithm, based on premise that it is extremely difficult to factor the product of two large prime numbers. Secret key are the two large prime numbers and public key is the product of the prime numbers

Public Key is typically not used for encrypting long messages. Typically used to secure transfer of relatively short symmetric keys, which are then used for symmetric key "bulk" encryption.

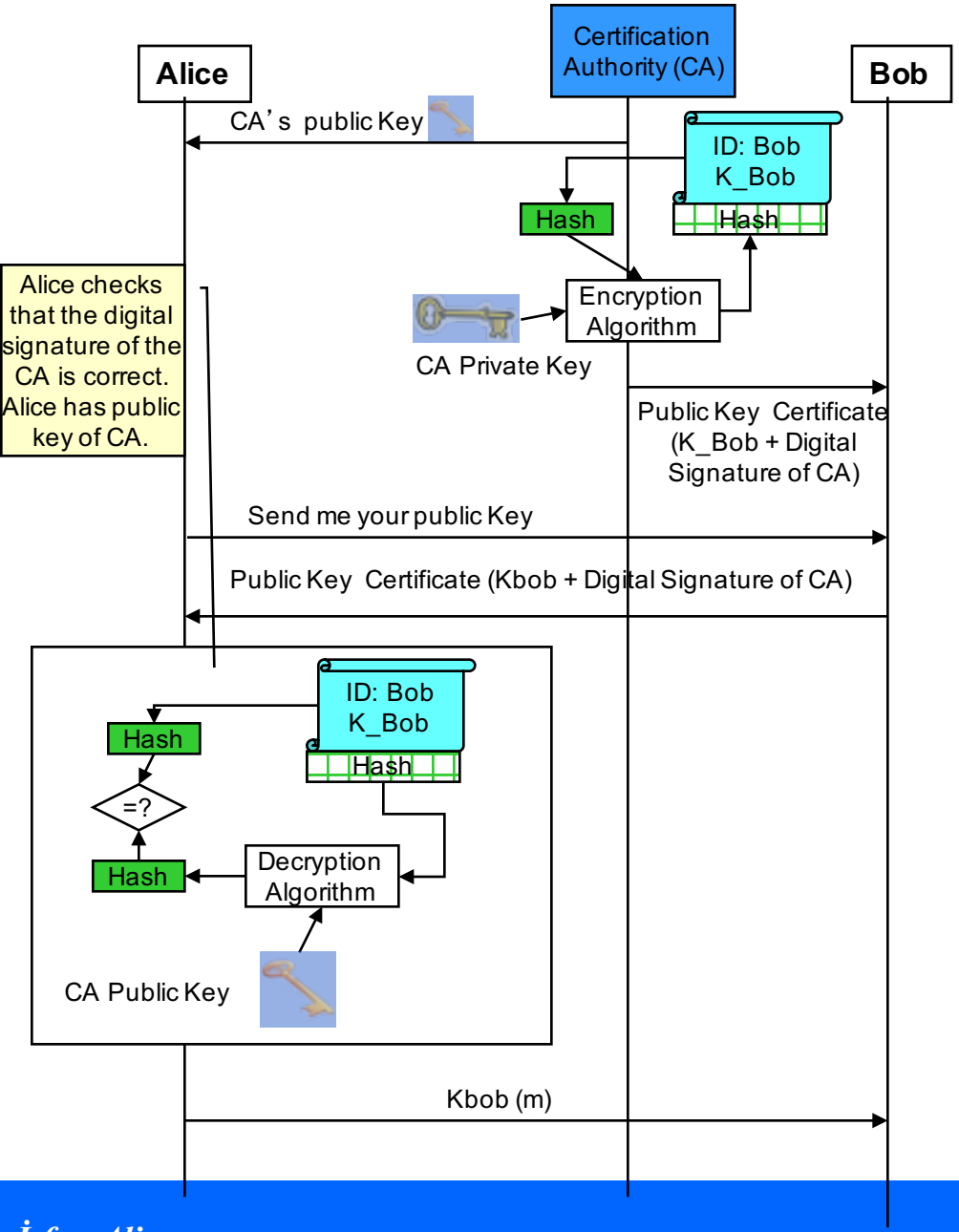
Public Key Cryptography: Obtaining public key -1



Bad Guy impersonating Bob

- *Impersonation Attack:*
  - Alice has no means of associating a Public Key with Bob.

Public Key Cryptography: Obtaining public key -2



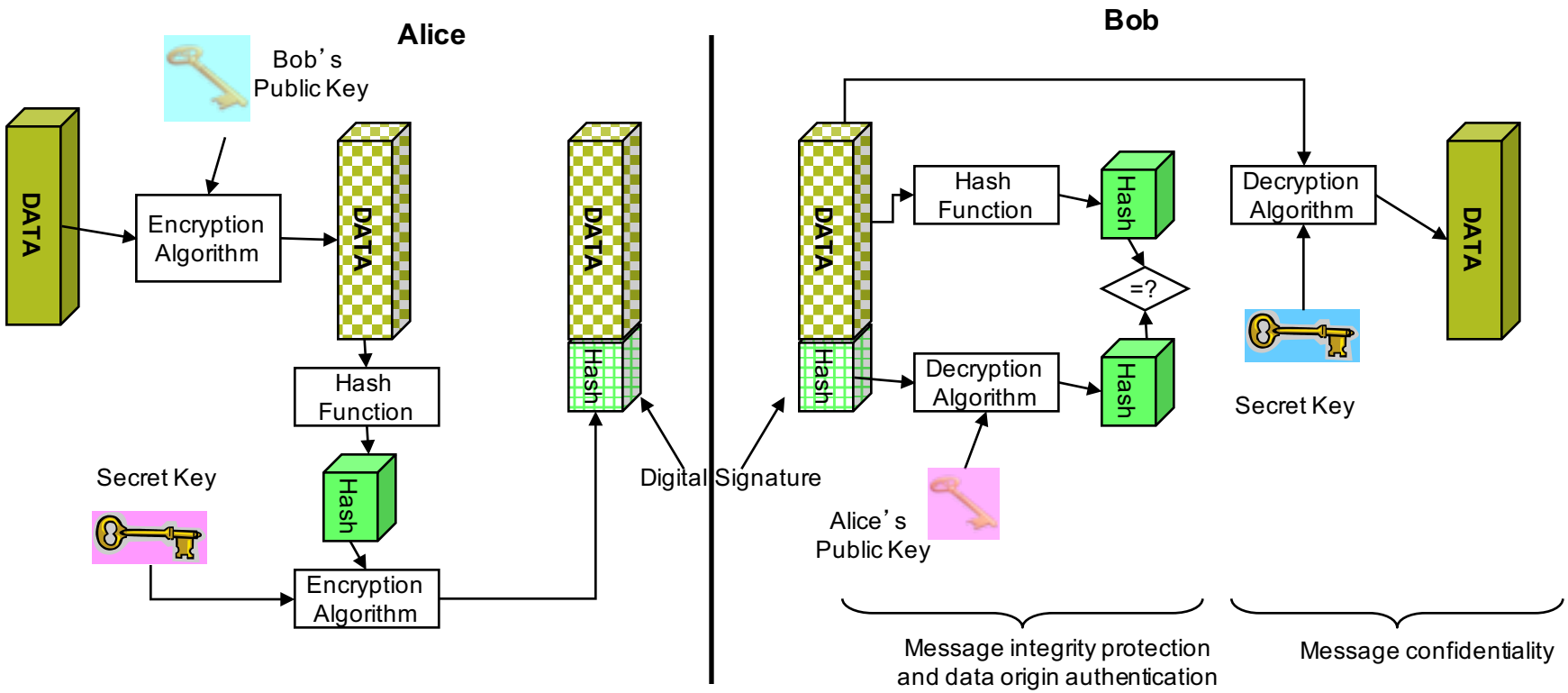
- *Impersonation Attack can be avoided by using public Key Certificates.*
  - Digitally signed statement by a trusted entity – a Certificate Authority (CA) – publishing some information about another entity (eg Bob).
  - Certificate is signed by CA's private key
  - Certificate binds an identifier to a public key
  - X.509 is a widely used standard for public key certificates. This is an ITU standard.
- *Since Bob and Alice may belong to different administrative domains, they may have different CA. To solve this problem a heirarchy of CAs has been defined.*

One-way Hash function

Message Integrity checking

- *A one-way Hash function takes an arbitrarily long input message and creates a fixed length, pseudo-random output called hash.*
  - Knowing the hash, difficult to find out the message that produced the hash
  - Two messages almost never generate the same hash.
- *Algorithms:*
  - Message Digest 5 (MD5): Creates 128 bit message digest (hash)
  - Secure Hash Algorithm (SHA-1): Creates 160 bit message digest (hash)
- *Unkeyed and keyed Hash Functions*
  - MD5 and SHA-1 are unkeyed hash functions
  - Keyed Hash function: HMAC uses in addition a shared secret key.

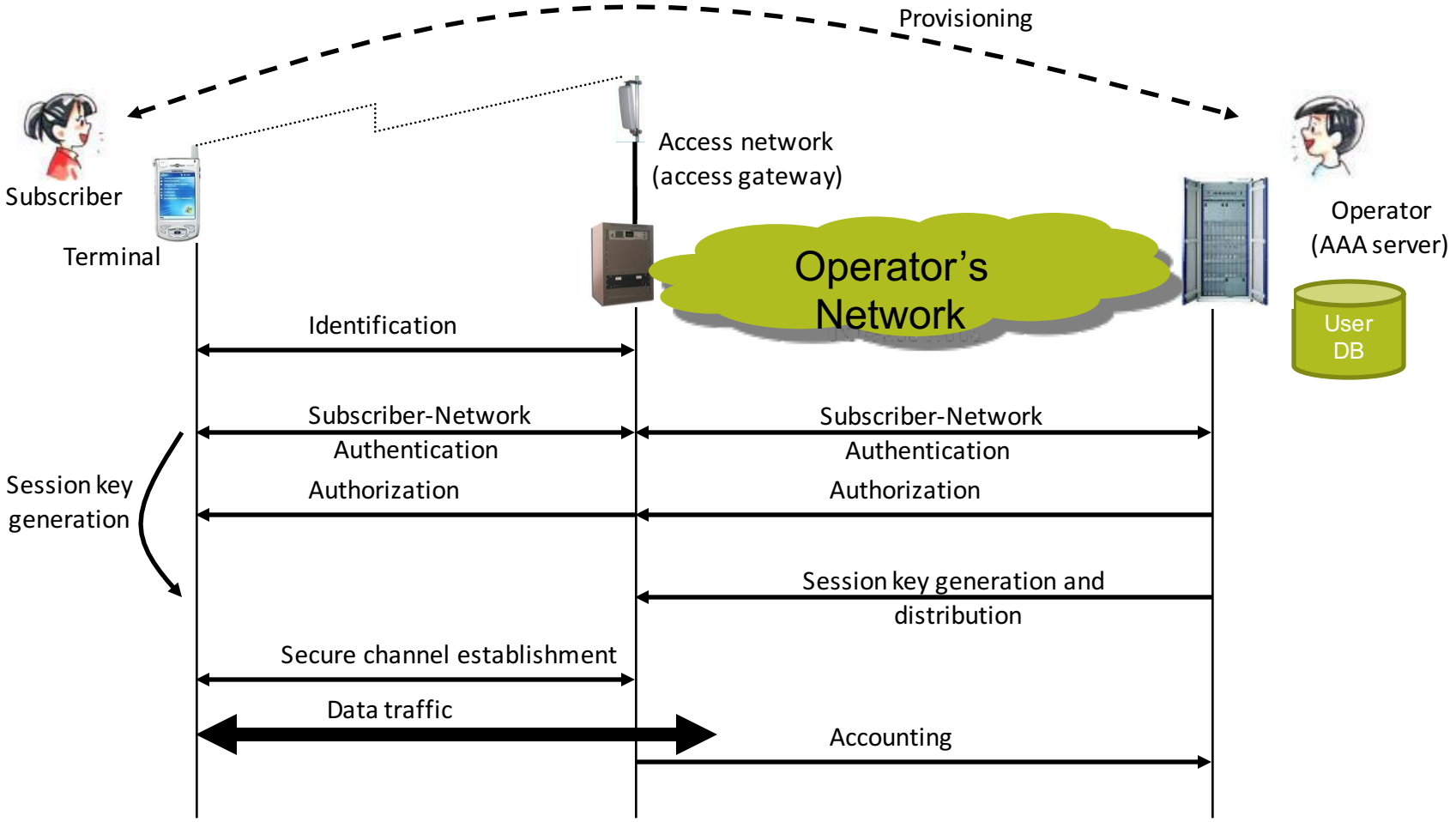
Public Key Cryptography: Overall



$E_{PK}(p) = c \;\; ; D_{SK}(c) = p$

$Sign_{SK}(m) = s \;\; ; \;\; Verify_{PK}(m,s) = OK?$

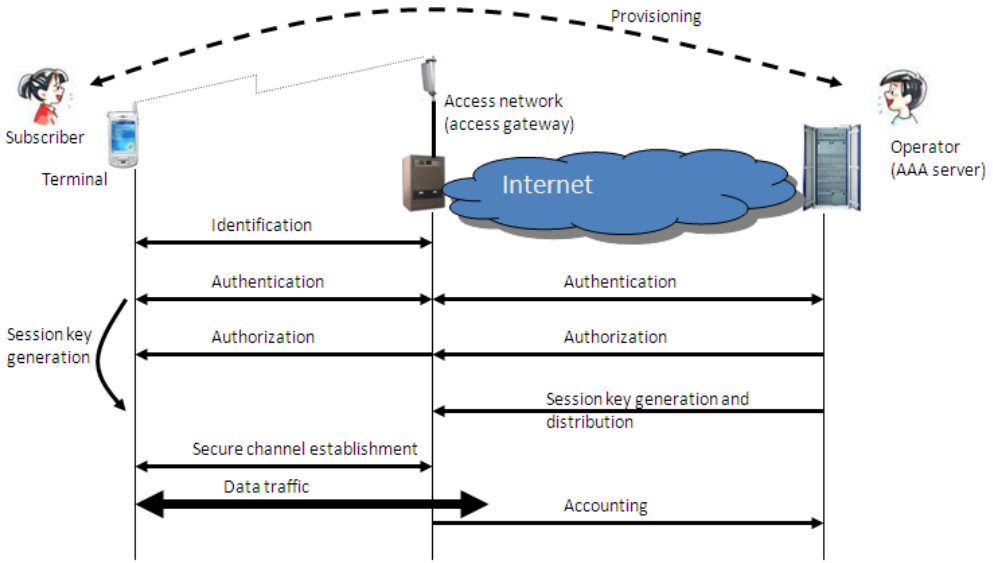
Template



AAA: Authentication, authorization, accounting

Components

- *Provisioning*
  - Static, mechanic, dynamic
- *Identification*
  - subscriber/ terminal
    - NAI, IMSI, MAC address
  - Access network/operator
    - MAC address, SSID, Operator ID



- *Subscriber-network authentication*
  - 3-party security model
  - 3GPP-AKA, EAP (RFC 3748) methods: EAP-TLS, EAP-AKA, MS-CHAPv2/EAP-TTLS
  - EAP/802.1X, EAP/RADIUS (RFC 2865)

TLS	Transport Layer Security
TTLS	Tunneled TLS
AKA	Authentication and Key Agreement
MS-CHAP	Microsoft Challenge-handshake Authentication Protocol
RADIUS	Remote Authentication Dial In User Service

Key Derivation

- *Subordinate keys are computed for purpose-specific use (e.g., data vs. signaling security).*