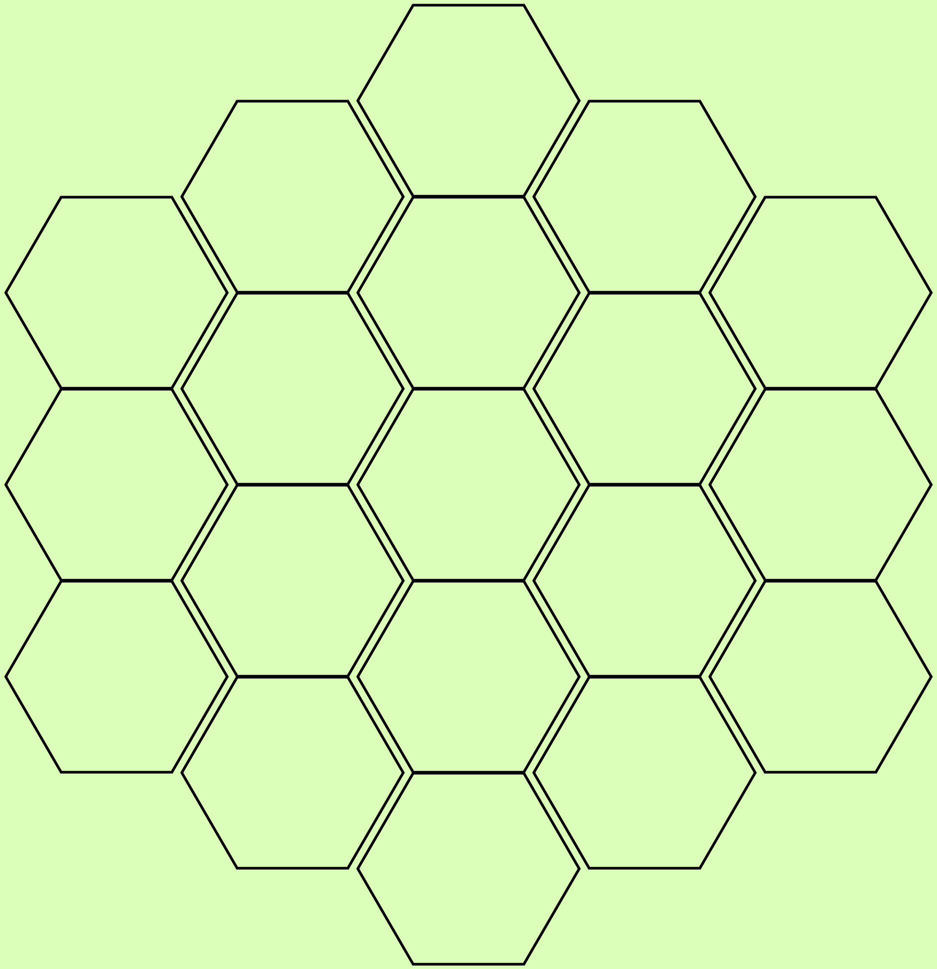
A Compact Authentication & Key Distribution Protocol
Based on a Broadcast Control Channel

Yuliang Zheng
Monash University
Melbourne, Australia
Email: yzheng@fcit.monash.edu.au

Outline of the Talk

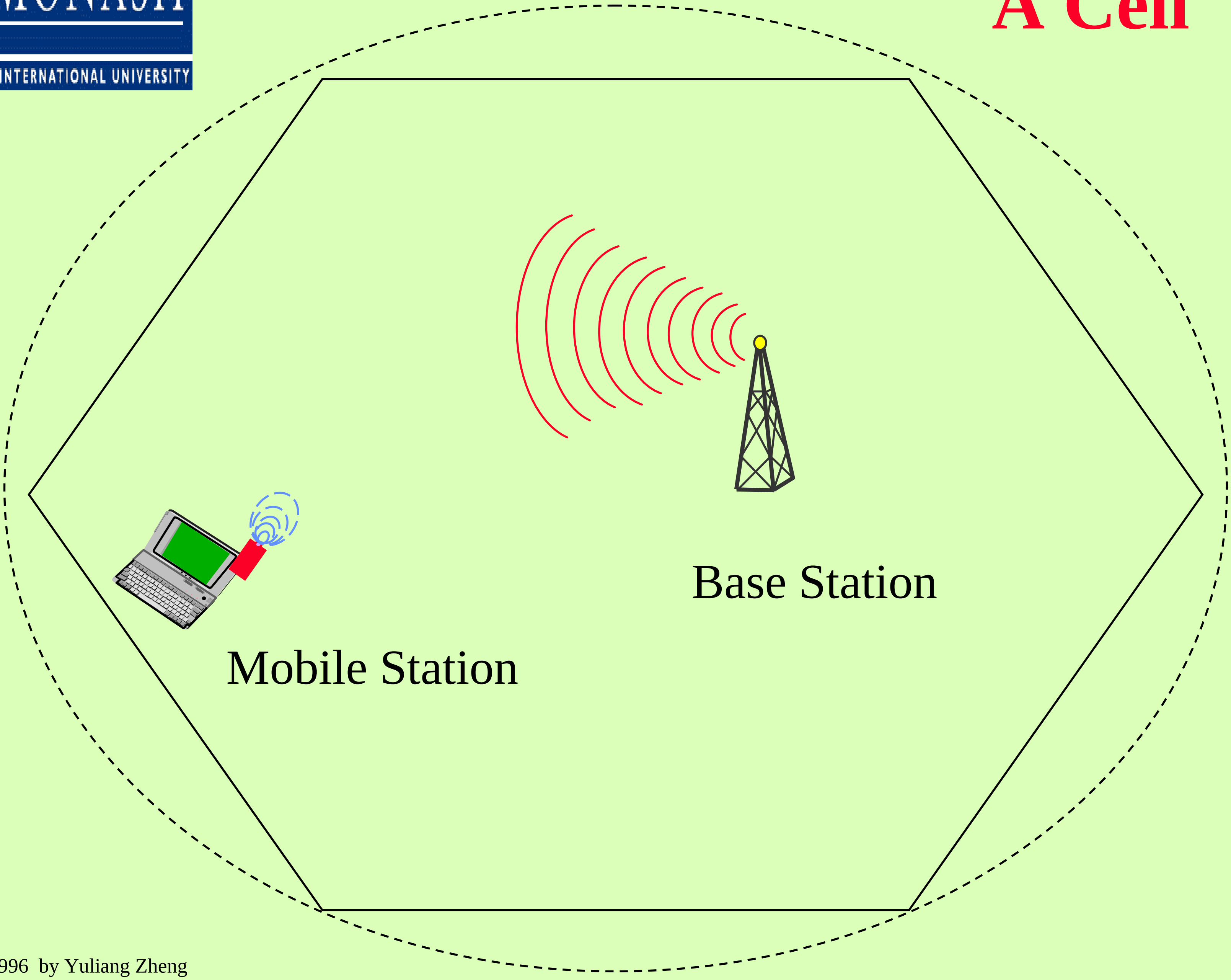
- Security issues in mobile computing
- Encryption and digital signature
- Identifying 2 problems with Beller, Chang & Yacobi's 5-step protocol (1993)
- Introducing a new 1.5 step protocol
- conclusion

Cells in Mobile Comp & Comm



neighbouring cells use different frequencies

A Cell



Issues in Mobile Computing

- Confidentiality of data
- Identification of a mobile user
- authentication of a base station
- prevention of insider attacks
- hand-over of authentication info.
- anonymity of a mobile station
- comp. and comm. cost for achieving the above

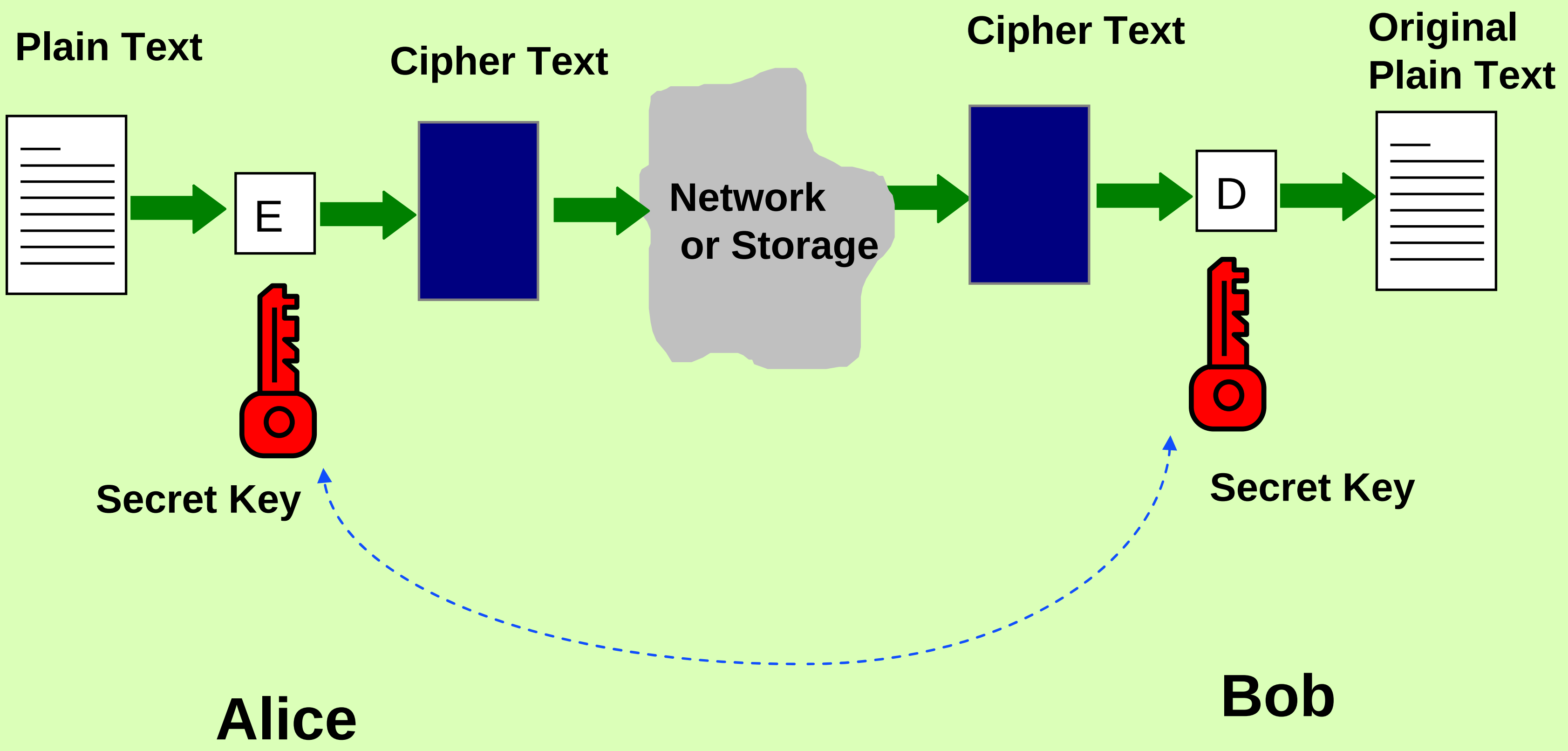
Issues in Mobile Comp (cnt'd)

- light weight of a mobile station
  - small batteries
  - low computing power
  - can only carry out relatively simple computing tasks !
- some contradict one another !
  - low computing power <---> high-level confidentiality and integrity
  - identification <---> anonymity

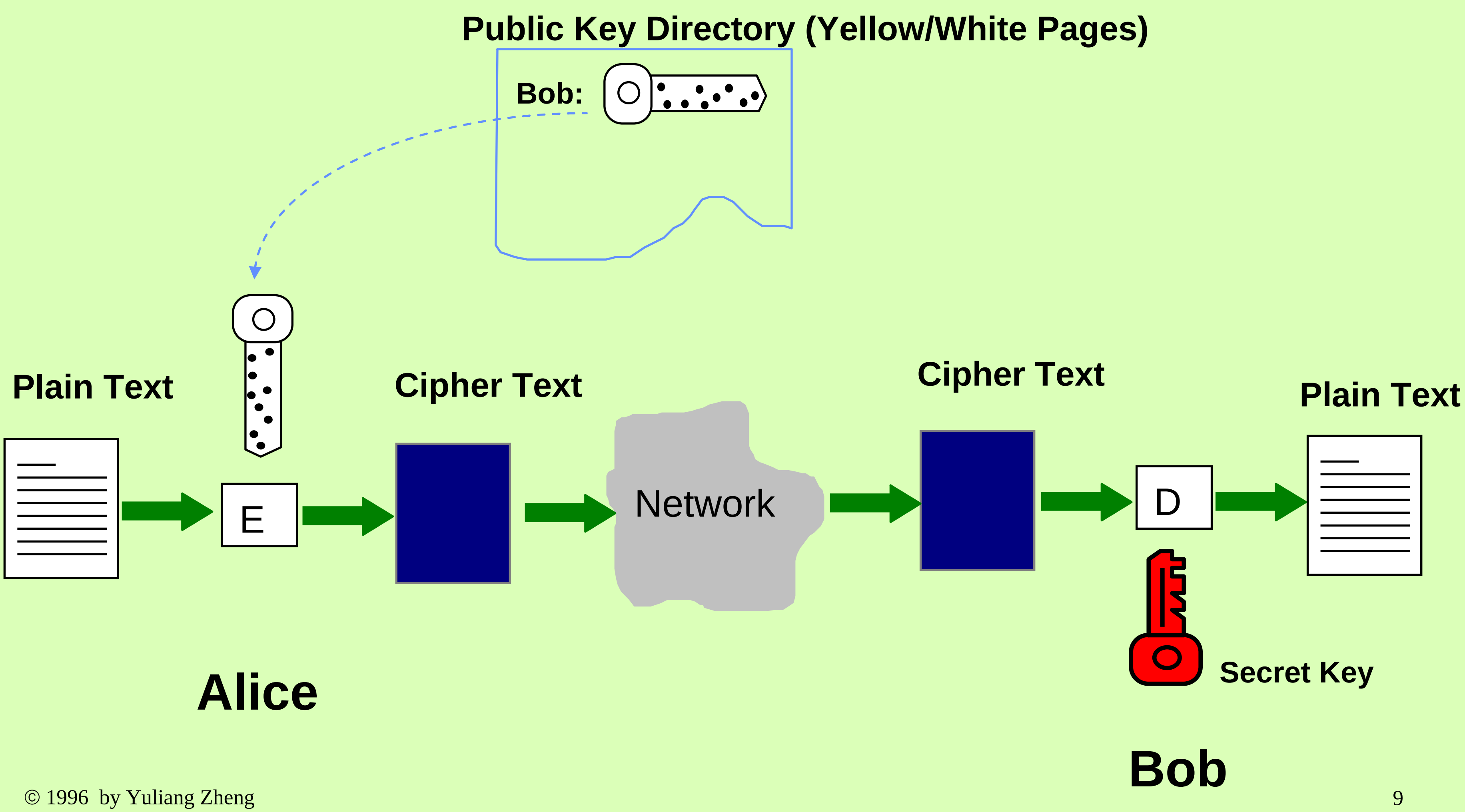
Two Major Issues

- Authentication of
  - a Mobile by a Base
  - a Base by a Mobile
- Key distribution (for securing communications contents)

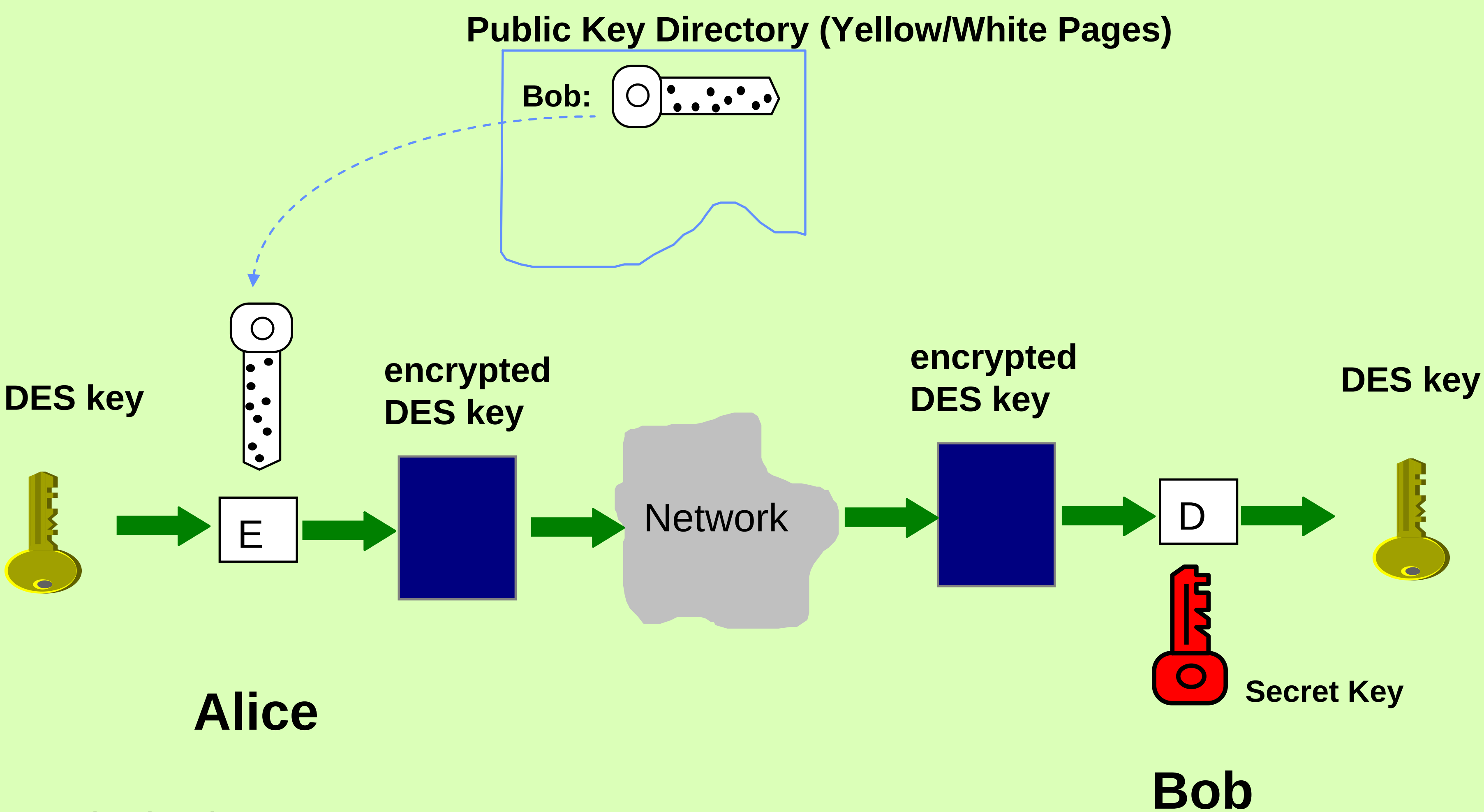
Private key cipher



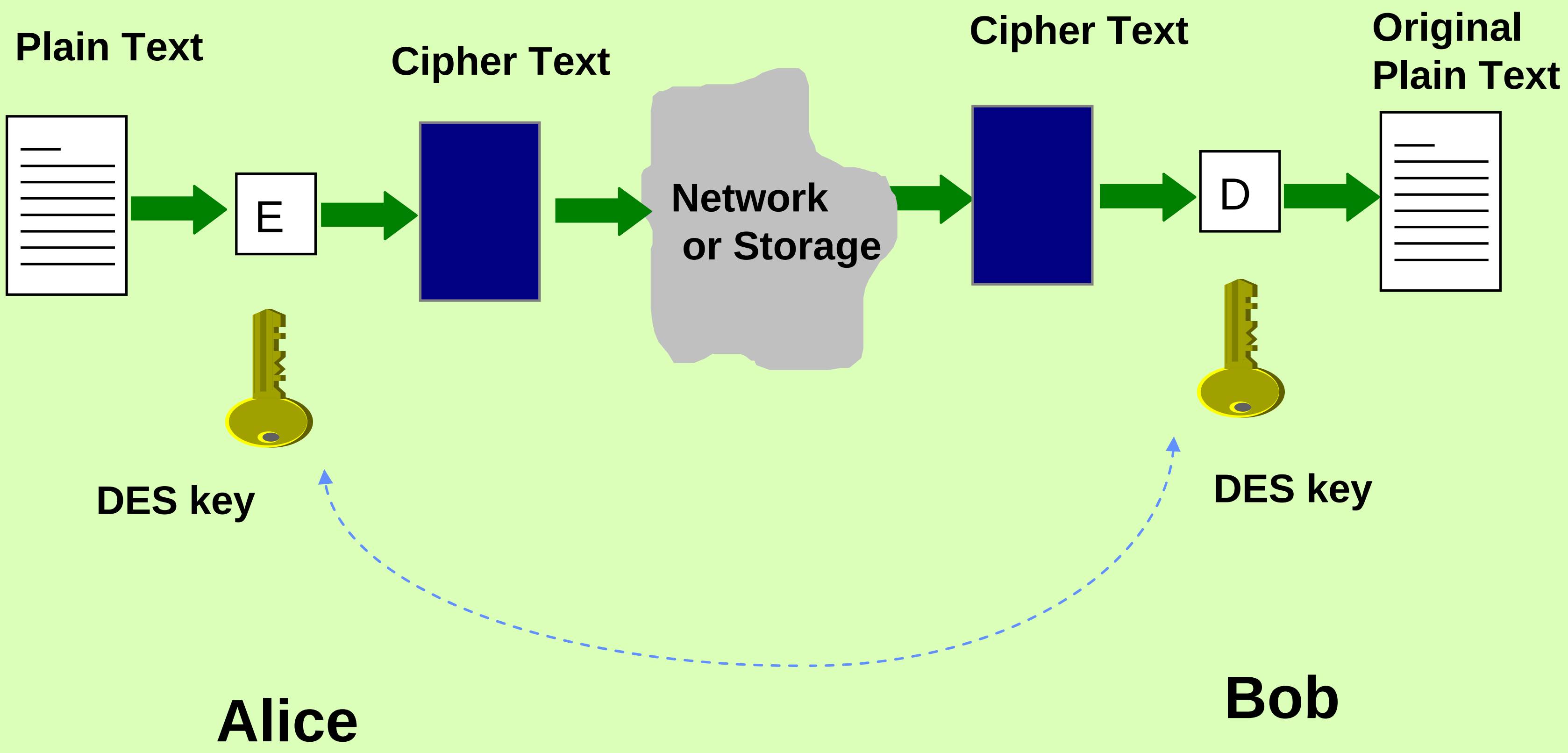
Public Key Cryptosystem



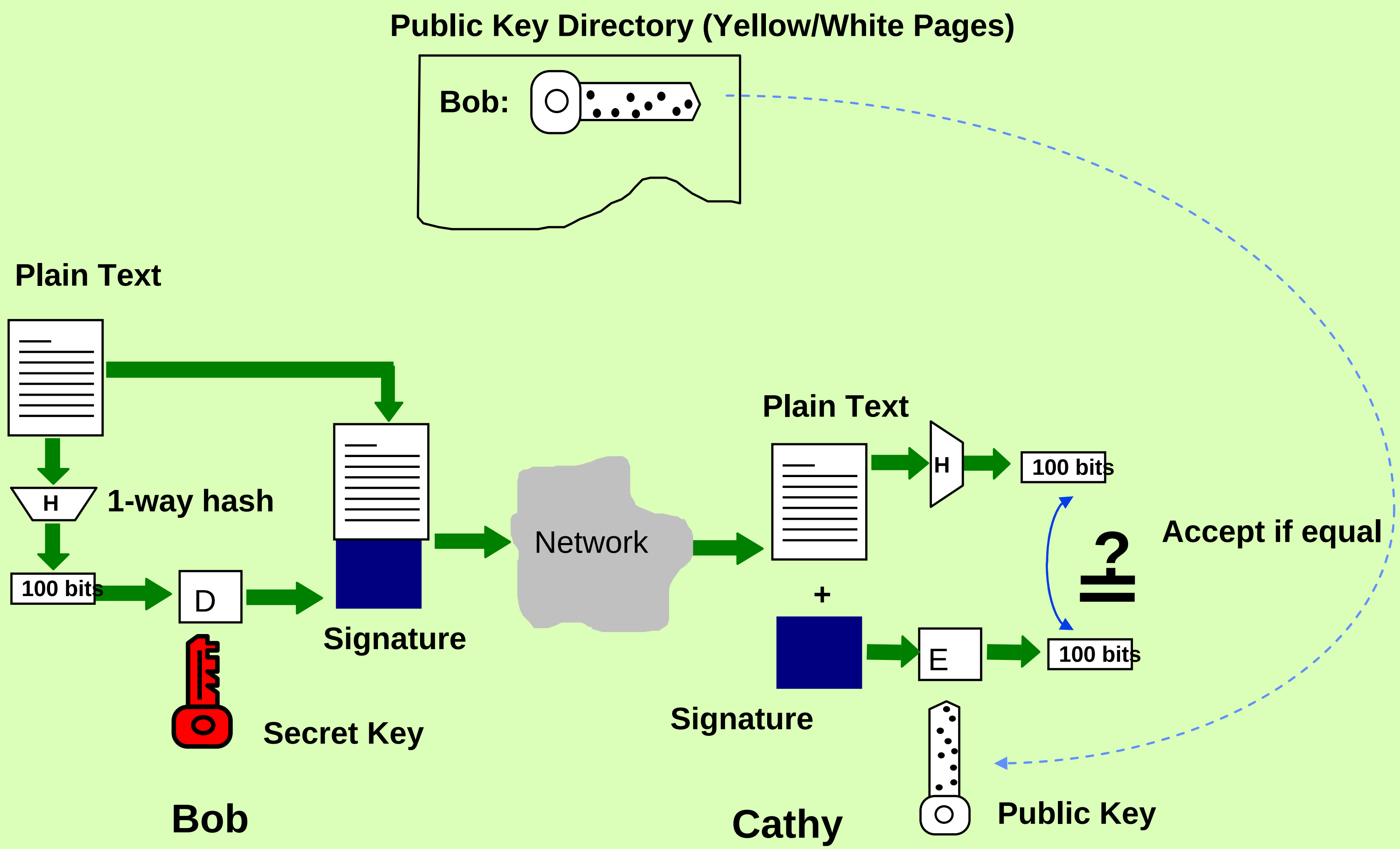
Hybrid Cryptosystem (1)



Hybrid cryptosystem (2)



Digital Signature (for long doc)



Notable Protocols for Mobile Comp & Comm

- GSM, 1990
- Cellular Digital Packet Data (CDPD) in USA, 1994
- Aziz-Diffie, 1994
- Molva-Samfat-Tsudik, 1994
- Asokan, 1994
- Herzberg et al, 1994
- Samfat-Molva-Asokan, 1995
- Mu-Varadharajan, 1996
- Beller-Chang-Yacobi, 1993

Beller, Chang & Yacobi Protocol (or BCY protocol)

- Based on two hard problems:
  - discrete logarithm on finite fields
  - factorisation of integers (Rabin's digital signature)
- Assumes the existence of a certification authority CA (or authentication centre)

4 Types of Parameters in BCY

- Public to all
- for Certification Authority
- for a base station b
- for a mobile station m

Parameters public to all

- **N: a large prime**
- **g: a generator for GF(N)***
- **1-way hash function: *hash***

Parameters for Cert. Auth.

- secret data: 2 large primes
- public data: their product N_{ca}

Parameters for Base b

- **secret data:**
  - **2 large primes**

    S_b
- **public data:**
  - **the product of the 2 primes:** N_b
  - $P_b \equiv g^{S_b} \pmod{N}$

$$sig_{ca,b} \equiv \sqrt{hash(b, N_b, P_b)} \pmod{N_{ca}}$$

Parameters for Mobile m

- **secret data:**

$$S_m$$

- **public data:**
  - $P_m \equiv g^{S_m} \pmod{N}$

$$sig_{ca,m} \equiv \sqrt{hash(m, P_m)} \pmod{N_{ca}}$$

5 Steps in BCY Protocol

Mobile m

Base b

Hi →

$$b, N_b, P_b, sig_{ca,b}$$ ←

$$e_2 \equiv x^2 (\bmod N_b), e_3 = DES_x(m, P_m, sig_{ca,m})$$ →

$$DES_{sk}(b)$$ ←

$$DES_{sk}(m)$$ →

5 Steps in BCY Protocol (cnt'd)

Mobile m — Hi → Base b

This is my certificate. ←

I checked your cert. It's OK. Here is my cert. encrypted using DES. The key for DES is sealed using your public key. →

I checked your cert. It's OK. Here is my name sealed using DES with the new session key ←

I can recover your name. Here is my name sealed using DES with the same session key →

2 Problems with BCY Protocol

- **5 steps --- very inefficient !**
- **vulnerable to replay attacks !**

Why all the 5 Steps are needed

- **An attacker can obtain a mobile station's public data**
  - **m, i.e. the ID of the mobile**

$$P_m \equiv g^{S_m} \pmod N$$

$$sig_{ca,m} \equiv \sqrt{hash(m, P_m)} \pmod{N_{ca}}$$

- **He will then be able to successfully masquerade Mobile m, and pass Steps 1, 2 and 3 !**

Why all the 5 Steps are needed (cnt'd)

- Although it's very unlikely that the attacker can derive the valid session key sk,
  Steps 4 & 5 are absolutely necessary for the genuine Mobile and Base to confirm the consistency of their session keys.

Replay Attacks
--- Potentially More Serious

- Consider an attacker malicious towards Mobile m
  - Records the 5 steps between Mobile m and Base b.
  - Some time later, initiates a communication session with Base b.
  - Replays the data previously sent by m to b
  - Passes all the 5 steps !!!

Cause financial loss to Mobile m

- Assume that the 5 steps are followed by a destination address encrypted using the session key. Now, as the attacker malicious towards Mobile m does not have the session key, he cannot choose a destination address as he wishes.
- But he can simply send a random ciphertext to Base b.

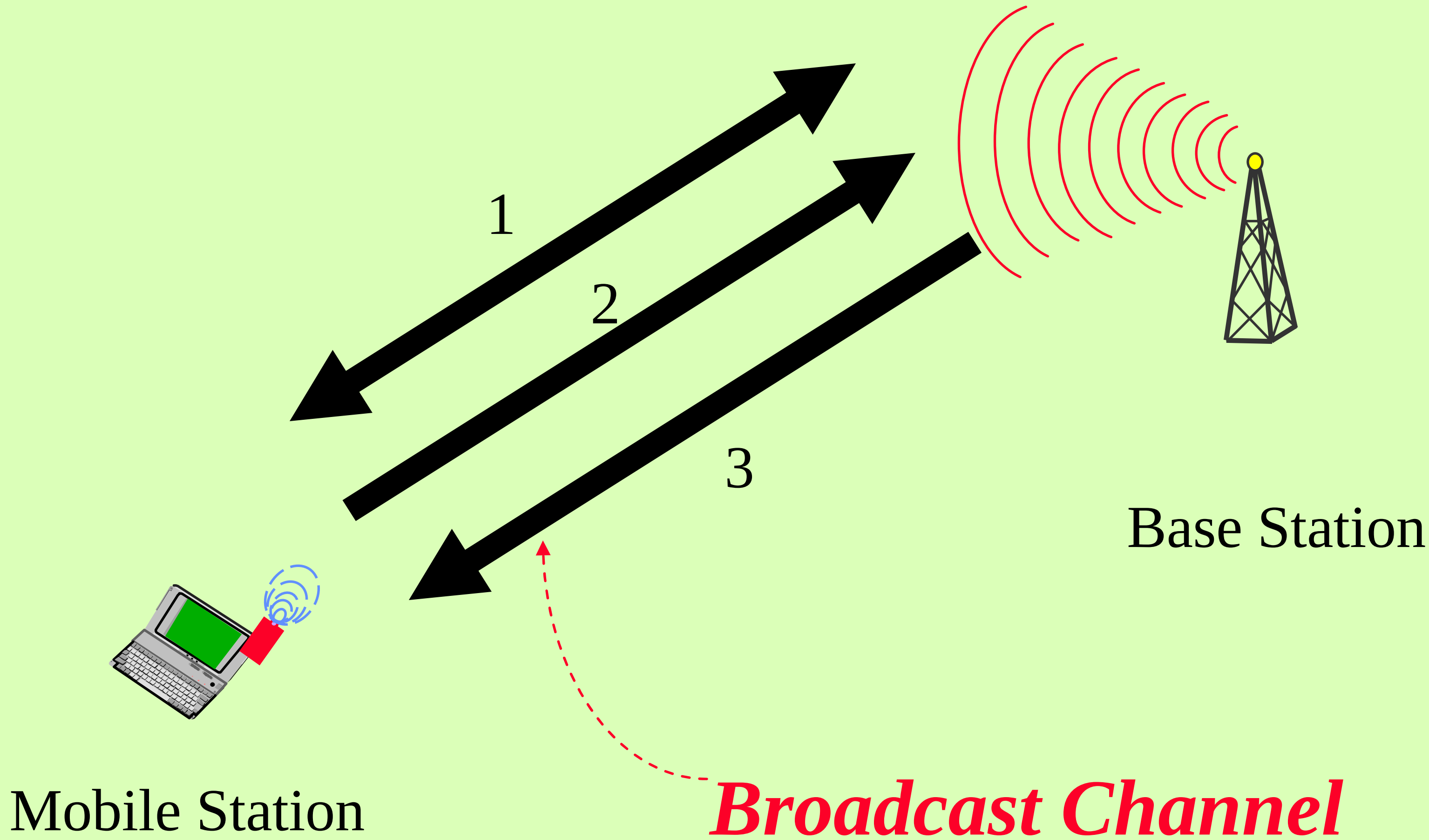
Cause financial loss to Mobile m

- Now, the attacker waits to see if a connection between him and another address decrypted from his random ciphertext can be established.
  If it is established (AND the address happens to be, say, a fax number), the attacker may be able to send, unidirectionally, a large amount of data to the address !

A New Proposal with 1.5 Steps

- **Main Ideas:**
  - **Using a broadcast control channel**
  - **Using**
    - **Certification Authority (such as X.500 directory services), and**
    - **public key cryptography.**

3 Types of Channels



Functions of a Broadcast (Control) Channel

- For the network to propagate to mobile stations various types of control information:
  - synch parameters
  - available services
  - current network time
  - Base station ID
  - etc
- (Each mobile station keeps on monitoring the BCC)

Goods & Bads of Broadcast

- Bads
  - Everybody can hear
    - Encryption is required to provide confidentiality
- Goods
  - Everybody can hear !
    - Base:
      - needs to say once only (no need to repeat)
      - can propagate certificate, time and even nonce
    - Mobile: can choose to ignore if not interested

4 Types of Parameters

- Public to all
- for Certification Centre
- for a base station b
- for a mobile station m

Parameters public to all

- p: a large prime
- q: a large prime factor of p-1
- g: has order q mod p
- 1-way hash function: *hash*

- Use DSS (digital signature standard), but others (e.g. Schnorr’s) are OK too.

Parameters for Cert. Auth.

- **secret data:** x_{ca}
- **public data:** $y_{ca} \equiv g^{x_{ca}} \pmod{p}$

Parameters for Base b

- **secret data:** x_b
- **public data:**
  - $y_b \equiv g^{x_b} \pmod p$
  - $sig_{ca,b} \equiv (r_b, s_b)$

where

$$r_b \equiv (g^{k_b} \bmod p) \pmod q$$

$$s_b \equiv (h(M_b) + x_{ca} \cdot r_b) / k_b \pmod q$$

- **M_b=(b, y_b, expire date, ...)**

Parameters for Mobile m

- **secret data:** x_m
- **public data:**
  - $y_m \equiv g^{x_m} \pmod p$
  - $sig_{ca,m} \equiv (r_m, s_m)$

where $r_m \equiv (g^{k_m} \bmod p) \pmod q$

$$s_m \equiv (h(M_m) + x_{ca} \cdot r_m) / k_m \pmod q$$

- **M_m=(m, y_m, expire date, ...)**

0.5 Step: Base --> Mobile

- **Base b** uses part of the capacity of a Broadcast Control Channel (BCC) to propagate, regularly, the following info to all mobile stations in the cell:

$$b, y_b, sig_{ca,b}, current\_time / nonce, etc$$

- **Note**: Information on Certification Authority may also be broadcast, at a less frequent rate.

0.5 Step: Base --> Mobile (cnt'd)

- **When Mobile m roams into the cell of Base b, or a user switches it on, it records, *at the background*, the following info in the BCC:**
  - **the certificate information,**
  - **current_time / nonce**
  - **etc**
- **Mobile m then checks the authenticity of the certificate, *at the background*.**

Why we say it's "0.5" Steps

- It can be done
  - **at the background**, and
  - well before an actual session is started, and
  - once only for a cell (or less, depending the certificate verification strategy chosen)

Base <--- Mobile

- **When Mobile wishes to initiate a communication session with Base b, it sends the following to Base b:**

$$(c_1, c_2)$$

How (c_1, c_2) are defined

- (c_1, c_2) **are defined as**

$$c_1 \equiv g^x (\text{mod } p) \text{ for random } x$$

$$c_2 = DES_k(sk, t / n, m, y_m, sig_{ca,m}, \ldots, sig_m)$$

$$k = y_b{}^x \text{ mod } p$$

$$sig_m = \text{ Mobile m's signature on}$$

$$(sk, t / n, m, y_m, sig_{ca,m}, \ldots)$$

Checking by Base

- **operations by Base b upon receiving c_1 and c_2 from Mobile b:**

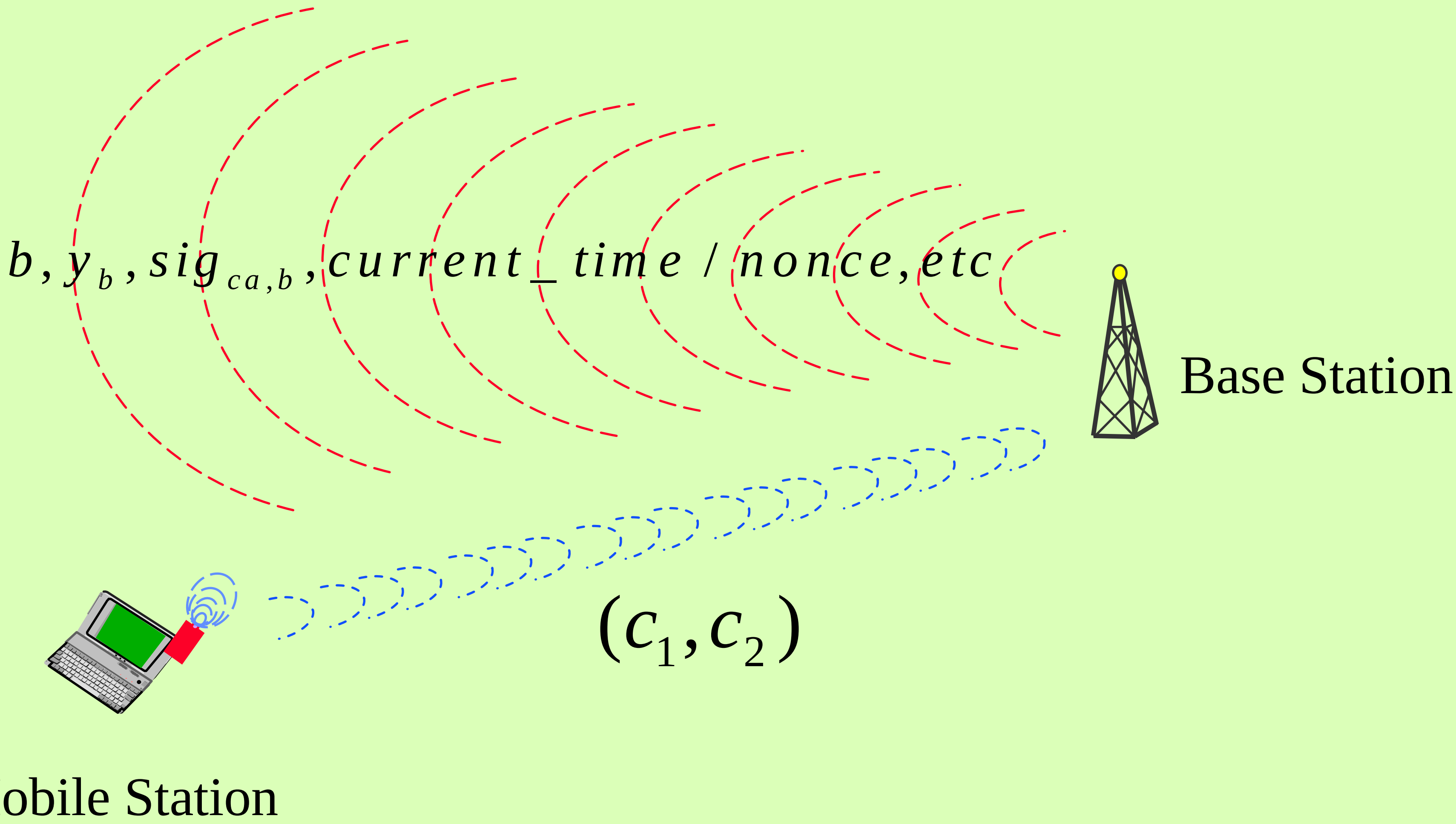
$$k = c_1^{x_b} \bmod p$$

  - **Decrypting c_2 by the use of k**
  - **verifying the freshness of time-stamp t, or nonce**
  - **verifying the certification authority's signature**
  - **verifying Mobile's signature**

Checking by Base (cnt'd)

- **Base accepts K as a valid session key only all the checkings are OK**

0.5 + 1 Steps


$b, y_b, sig_{ca,b}, current\_time / nonce, etc$
Base Station
(c_1, c_2)
Mobile Station

Properties of the 1.5 Protocol

- Consistency of session keys is guaranteed.
- As time-stamp/nonce is involved, replay attacks are avoided.
- only 1.5 steps ---> efficient !
- Anonymity of Mobile against an on-looker
- Pre-computation by Mobile is possible

Properties of the 1.5 Protocol (cnt'd)

- Masquerade of Mobile, even by the base station, is prevented
- Currently under investigation ---
  - Applicable to distributed computing
    - broadcast is inherent in virtually all current LANs or WANs, especially in those based on Ethernet technology

Possible improvements

- Let Certification Authority use a signature with light-weight verification (such as Rabin)
- Let Base sign time-stamp or nonce
- Simplifying the protocol (?)
- Security proof
  - formal proof (based on logic), OR
  - exact security initiated by Bellare & Rogaway

Other issues under consideration

- **Strategies for pre-computation by Mobile m**
  - **to shorten the time to establish a connection**
- **Information transfer assocaited with roaming**

Summary

- Identified 2 problems with Beller, Chang & Yacobi's protocol
  - 5 steps --- inefficient
  - re-play attacks possible
- A new 1.5 step protocol

And finally ...

Q & C ?