Network Security

Asim Rasheed

Where we are …

- Introduction to network security
- Vulnerabilities in IP
- I. CRYPTOGRAPHY
  - Symmetric Encryption and Message Confidentiality
  - Public-Key Cryptography and Message Authentication
- II. NETWORK SECURITY APPLICATIONS
  - Authentication Applications (Kerberos, X.509)
  - Electronic Mail Security (PGP, S/MIME)
  - IP Security (IPSec, AH, ESP, IKE)
  - Web Security (SSL, TLS, SET)
- III. SYSTEM SECURITY
  - Intruders and intrusion detection
  - Malicious Software (viruses)
  - Firewalls and trusted systems

Message Authentication and Hash Functions

Message Authentication

- Message authentication
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)
- Security requirements
- Alternative functions used
  - message encryption
  - Message Authentication Code (MAC)
  - hash function

Authentication Requirements

- Disclosure

  Release of message contents to any person or process not possessing the appropriate cryptographic key

- Traffic analysis

  Discovery of traffic pattern between parties

- Masquerade

  Insertion of messages into the network from a fraudulent source

- Content modification

  Change of message contents e.g., insertion, deletion, transposition and modification

Authentication Requirements

- Sequence modification

  Modification to a sequence of messages between parties e.g., insertion, deletion, reordering

- Timing modification

  Delay or replay of messages

- Source repudiation

  Denial of transmission of message by source

- Destination repudiation

  Denial of receipt of message by destination

Authentication Functions

- Functions that may be used to produce an authenticator are

  – Message encryption

  – Message Authentication Code (MAC)

  – Hash function

Message Encryption

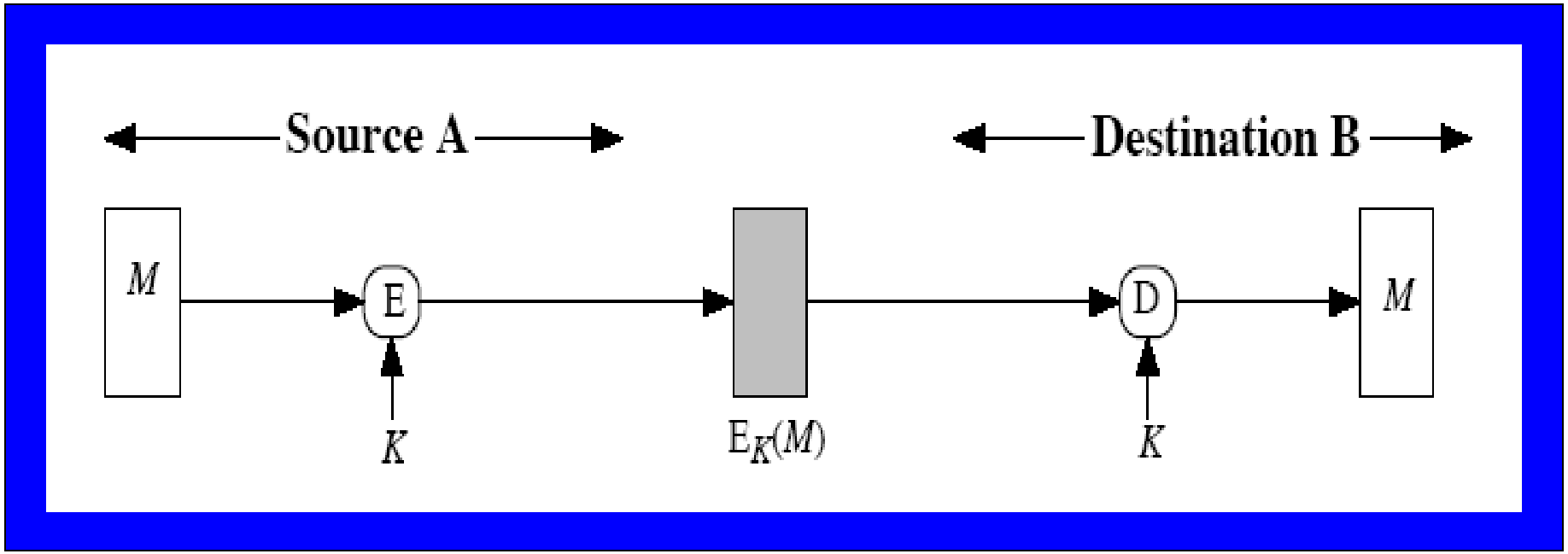
- Message encryption, by itself also provides a measure of authentication
- If symmetric encryption is used then:

–receiver knows sender must have created it

–since only sender and receiver know the key used

–if message has suitable structure, redundancy or a checksum to detect any changes

Symmetric Encryption Confidentiality and Authentication



Confidentiality and Authentication

Symmetric Encryption

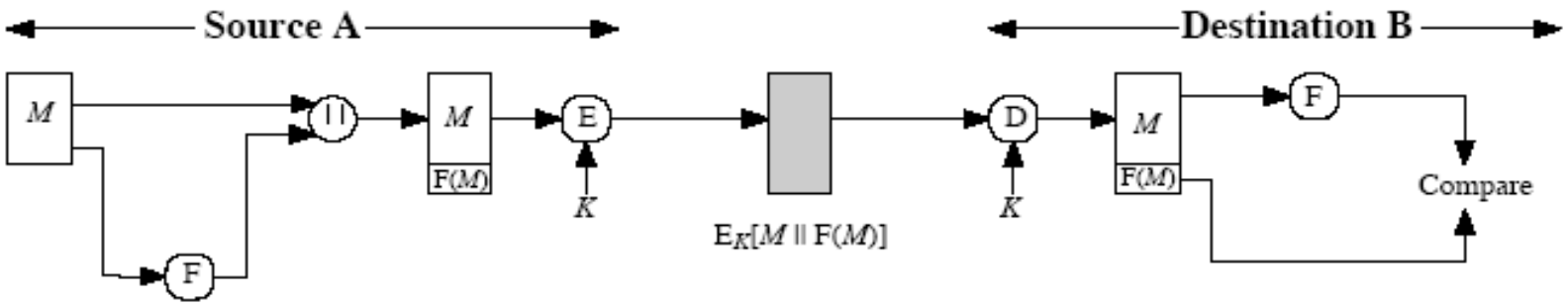
- It may be difficult to determine automatically if incoming cipher-text decrypts to intelligible plaintext.
- Thus, an opponent could achieve a certain level of disruption simply by issuing messages by purporting to come from a legitimate user
- **Solution**

–Force plain text to have some structure

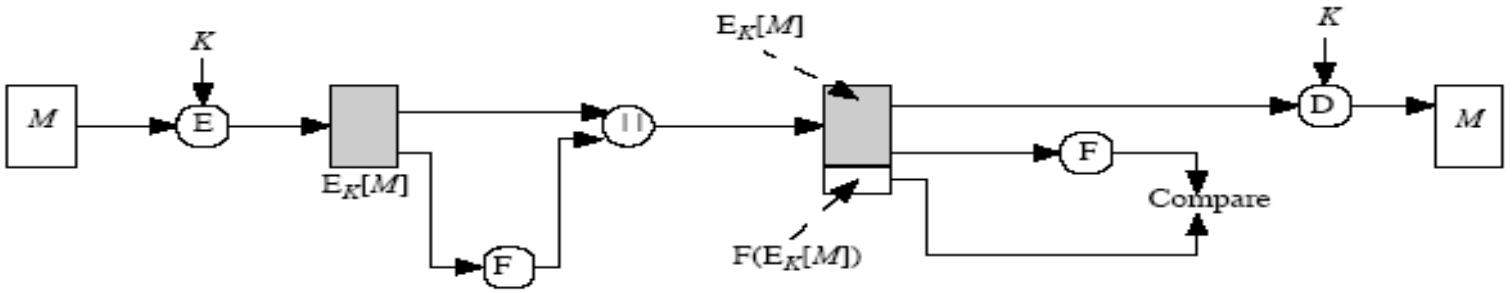
–Easily recognizable but cannot be replicated with recourse to the encryption function

–E.g., append an **error detecting code** known as FCS or checksum to each message before encryption

Symmetric Encryption Error Control



Internal error control



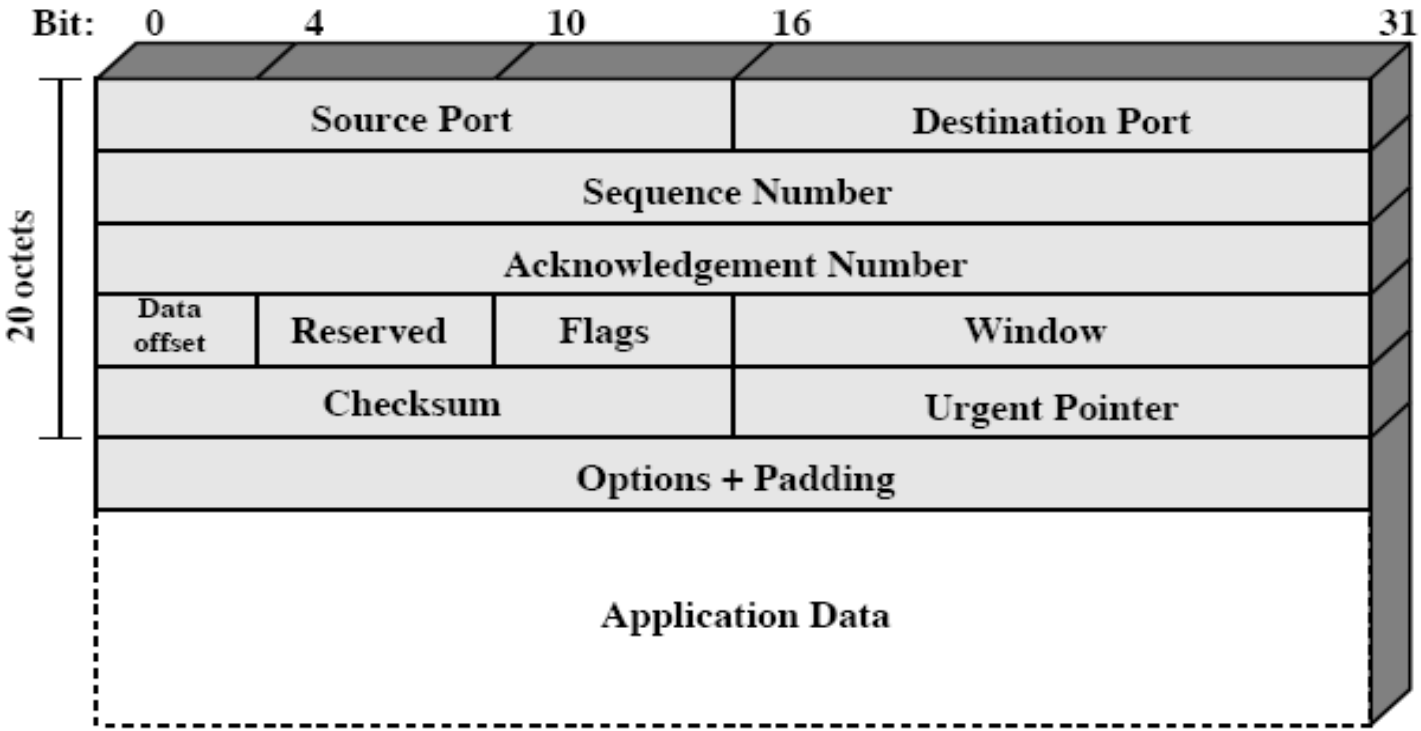
External error control

Symmetric Encryption Error Control

- An error control is just one example
- Any sort of structuring added to the transmitted message serves to strengthen the authentication capability
- Structures provided by the use of communications architecture consisting of layered protocols can be used.
- E.g., TCP/IP protocol architecture

–Because successive TCP segments on a given connection are numbered sequentially, encryption assures that an opponent does not delay, mis-order, or delete any segments

Structures to Strengthen Authentication



TCP Segment

Public Key Encryption

- If **public-key encryption** is used:

–encryption provides no confidence of sender

–since anyone potentially knows public-key

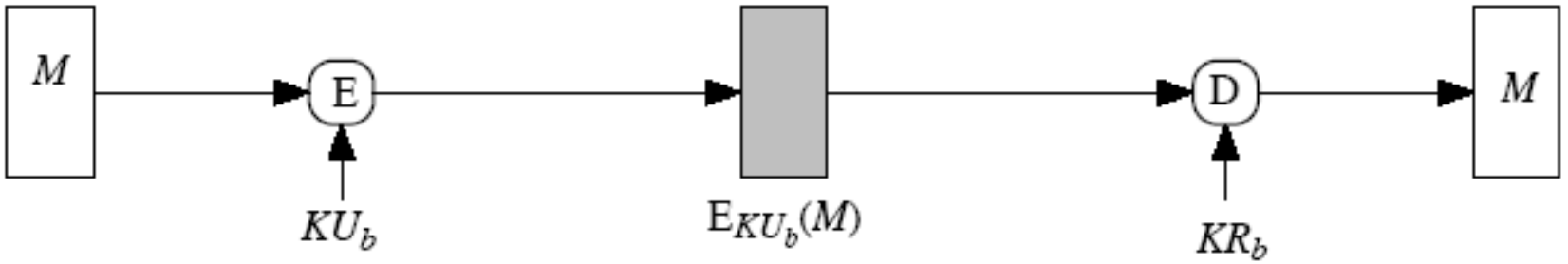
–however if

- sender signs message using their private-key
- then encrypts with recipient's public key
- Will now have both **secrecy** and **authentication**

–again need to recognize corrupted messages

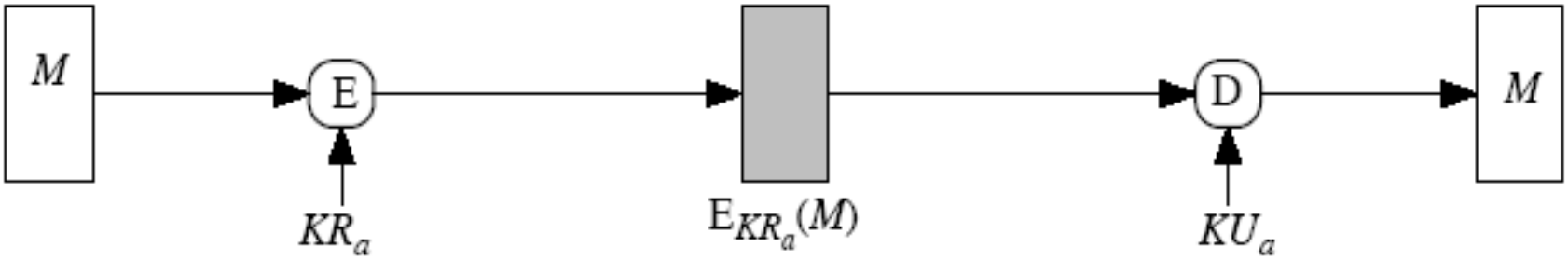
–but at the cost of two public-key uses on message

Public Key Encryption Confidentiality



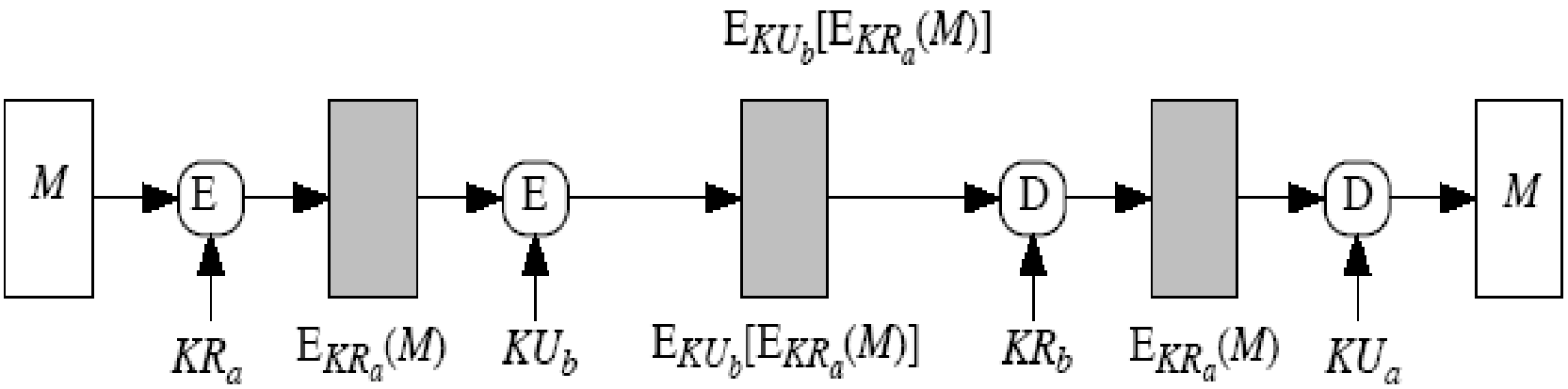
Confidentiality

Public Key Encryption Authentication and Signatures



Authentication and Signatures

Public Key Encryption Confidentiality, Authentication and Signatures

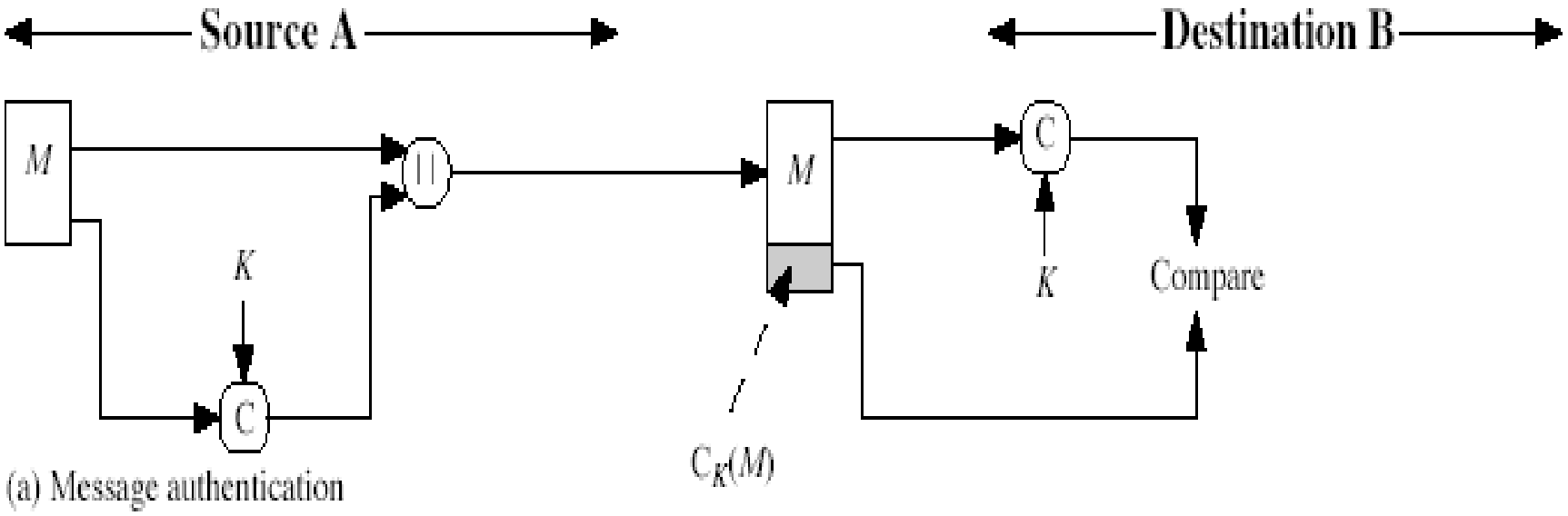


Confidentiality, Authentication and Signatures

Message Authentication Code (MAC)

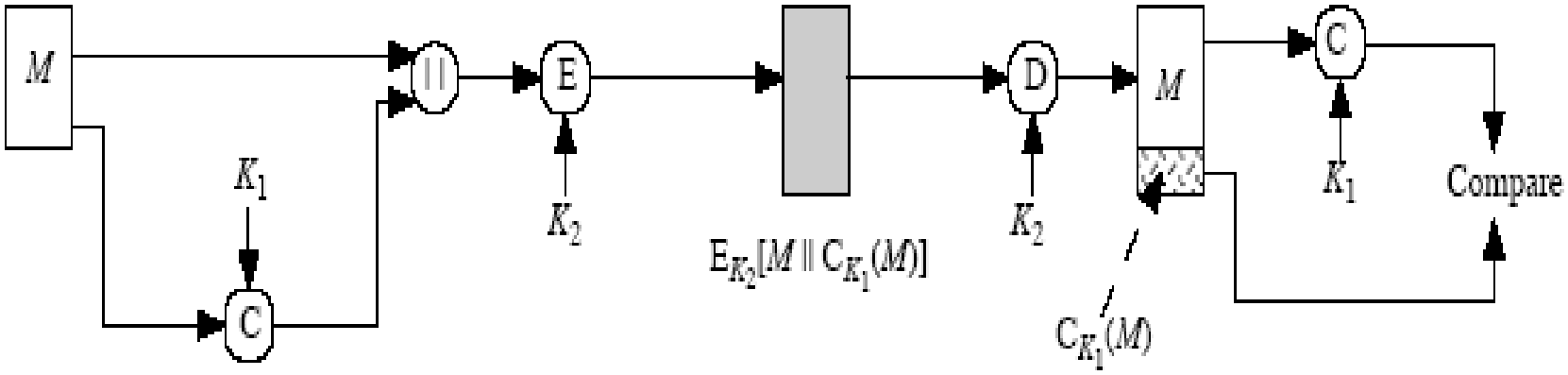
- Generated by an algorithm that creates a small fixed-sized block
  - –depending on both message and some key
  - –like encryption though need not be reversible, generally a many to one function
- Appended to message as a **signature**
- Receiver performs same computation on message and checks whether it matches the MAC
- Provides assurance that message is unaltered and comes from sender

Basic Uses of Message Authentication Code



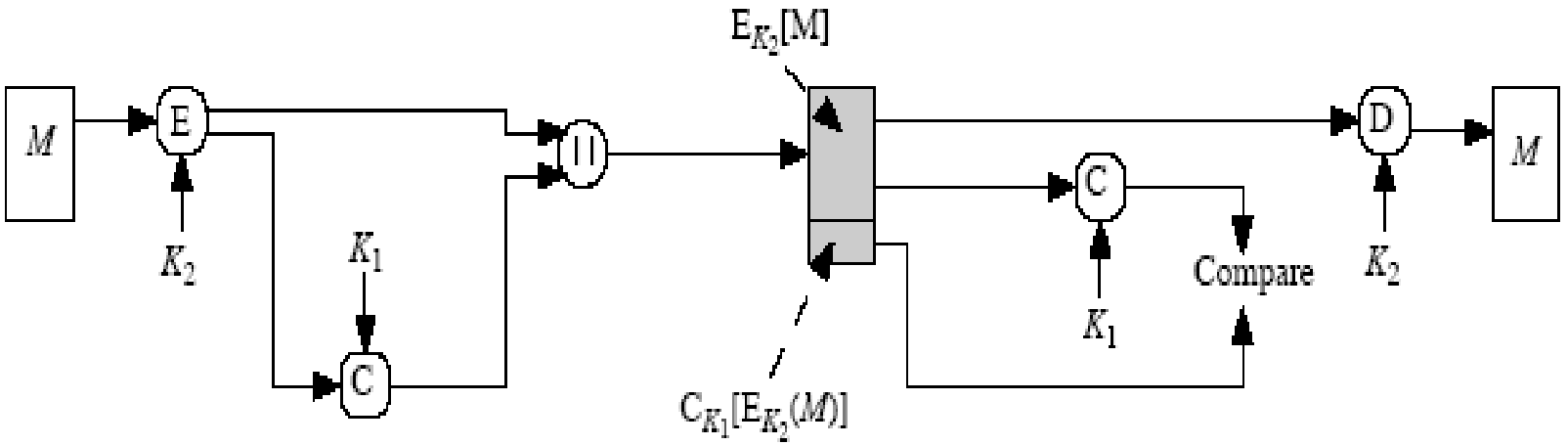
Message Authentication

Basic Uses of Message Authentication Code



Message authentication and confidentiality; authentication tied to plaintext

Basic Uses of Message Authentication Code



Message authentication and confidentiality; authentication tied to cipher-text

Message Authentication Codes

- As shown the MAC provides authentication
- Can also use encryption for secrecy
  - –generally uses separate keys for each
  - –can compute MAC either before or after encryption
  - –is generally regarded as better done before
- Why use a MAC?
  - –sometimes only authentication is needed
  - –sometimes need authentication to persist longer than the encryption (eg. archival use)
- MAC is not a digital signature

Hash Functions

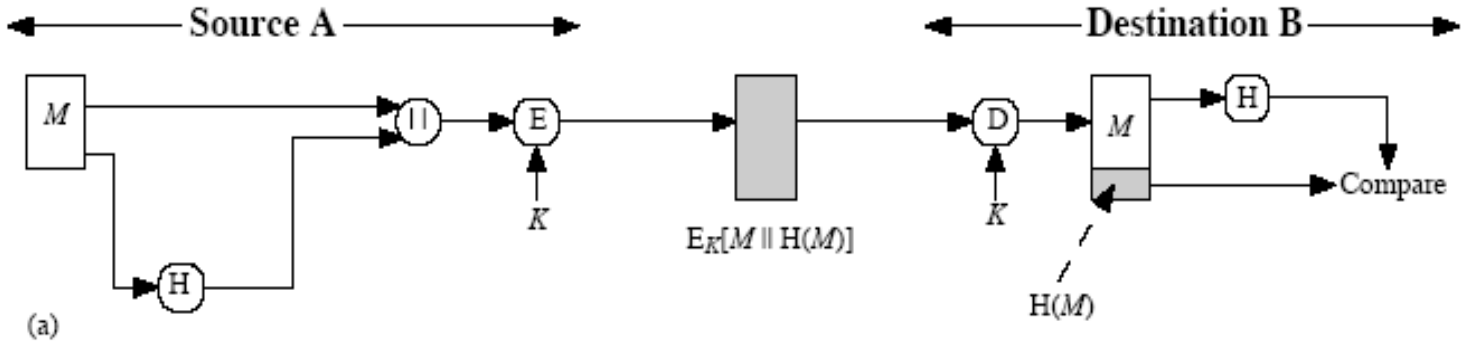
- A variation on MAC is one way hash function
- Condenses arbitrary message to fixed size
- Usually assume that the hash function is public and not keyed
  - cf. MAC which is keyed
- Hash code known as Message Digest or Hash value
- Hash used to detect changes to message
- Can be used in various ways with message
- Most often used to create a digital signature

Hash Functions

- Hash codes can be used to provide message authentication as:

a. Message plus the concatenated hash code is encrypted using symmetric encryption

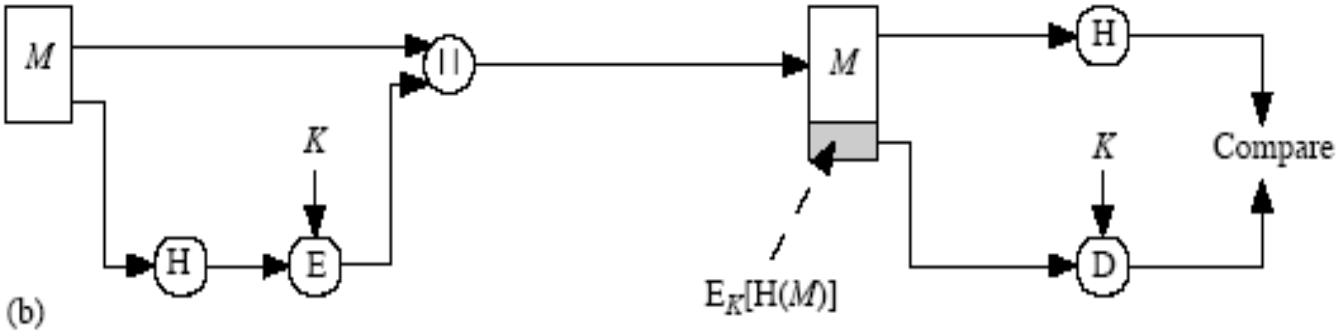
– Because encryption is applied to the entire message plus hash code, confidentiality is also provided



Hash Functions Contd.

b. Only the hash code is encrypted, using symmetric encryption

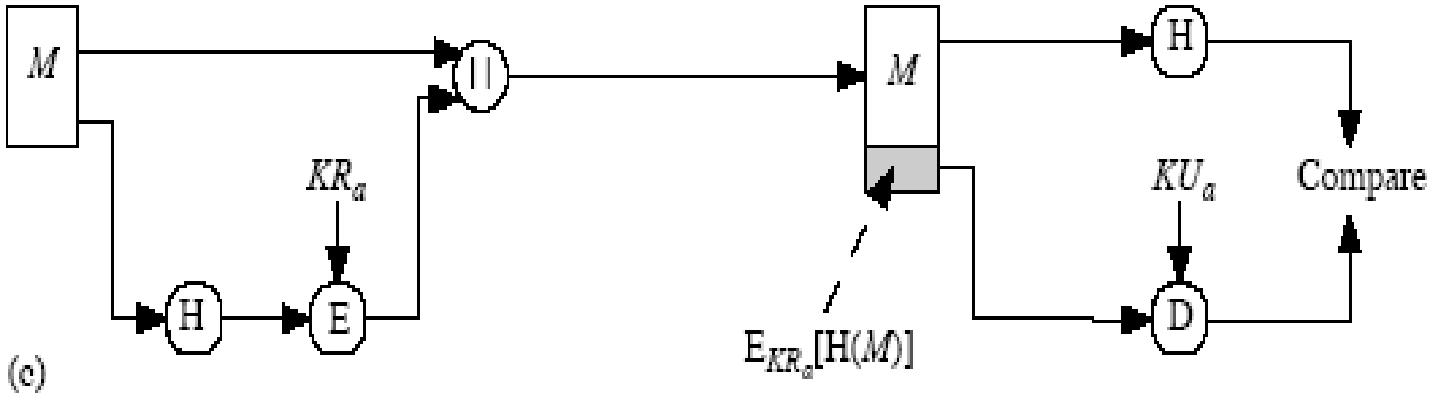
–Reduces processing burden for those application which don't require confidentiality



Hash Functions Contd.

c. Only hash code is encrypted using the public key encryption and sender's private key

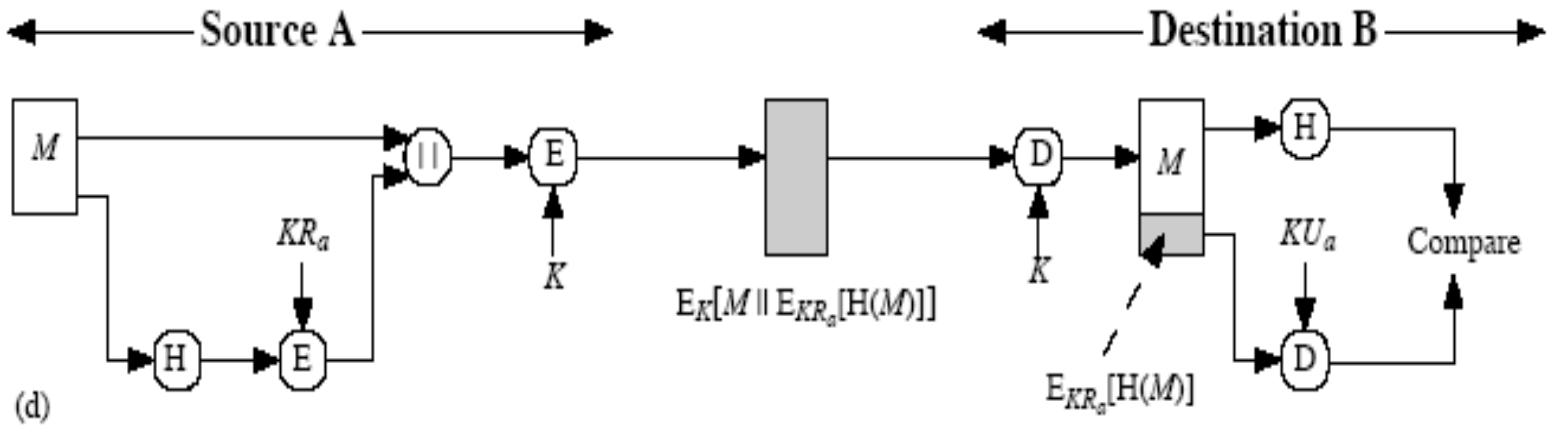
–Provides authentication and digital signatures



Hash Functions …contd

d. If confidentiality plus digital signatures are required then message plus public key encrypted hash code can be encrypted using a symmetric encryption key.

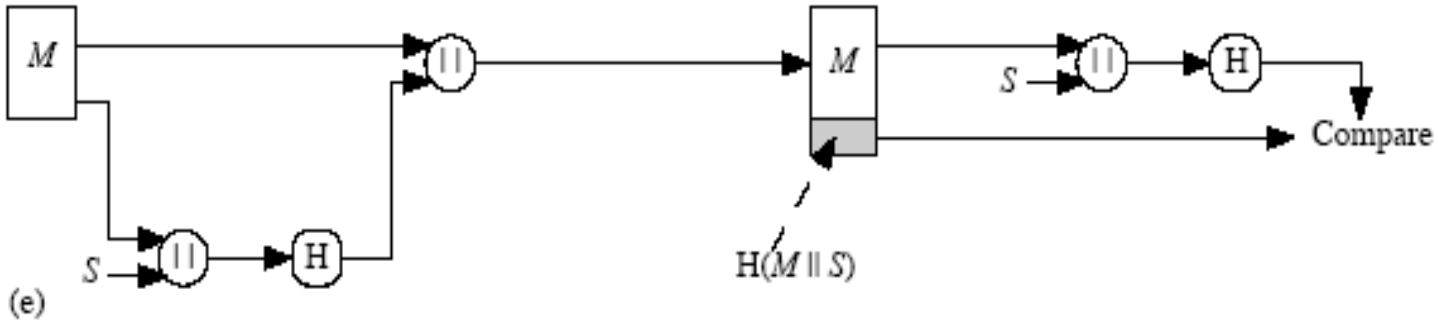
–This is the most common technique



Hash Functions …contd

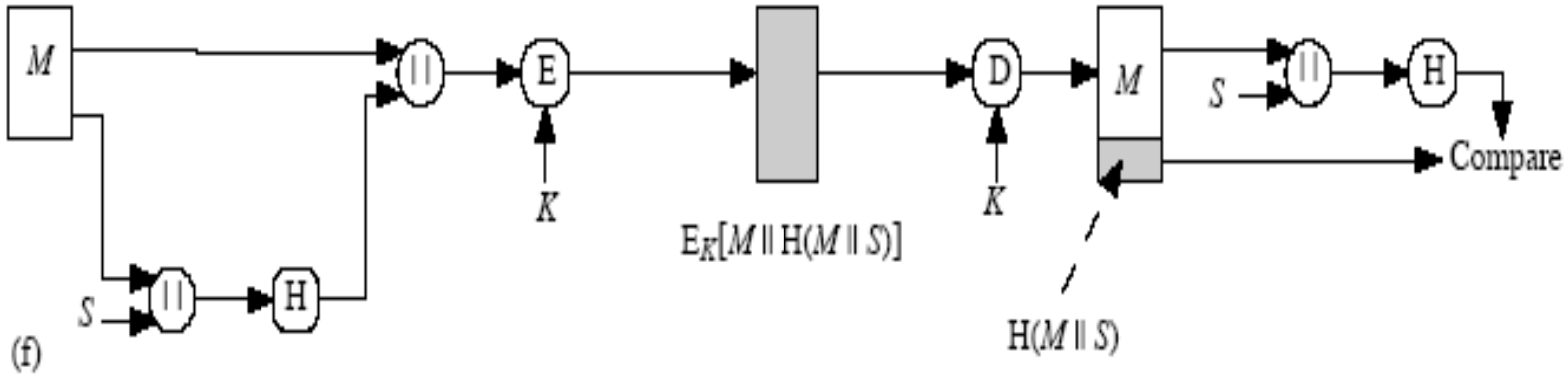
e. Uses hash function but no encryption for message authentication

–Assumes that the two parties share a common secret value S



Hash Functions …contd

f. Confidentiality can be added to the last approach by encrypting the entire message plus the hash code



Hash Functions …contd

- When confidentiality is not required, (b) and (c) have an advantage over those that encrypt the entire message
- Several reasons to avoid encryption
  - Encryption software is quite slow
  - Encryption hardware costs are not negligible
  - Encryption hardware is optimized towards large data sizes
  - Encryption algorithms may be covered by patents e.g., RSA
  - Encryption algorithms are subject to U.S. export control

MAC Properties

- A MAC is a cryptographic checksum

$$MAC = C_K(M)$$

– condenses a variable-length message M

– using a secret key K

– to a fixed-sized authenticator

- A many-to-one function

– potentially many messages have same MAC

– but finding these needs to be very difficult

Requirements for MACs

- Taking into account the types of attacks
- Need the MAC to satisfy the following:

–knowing a message and MAC, is infeasible to find another message with same MAC

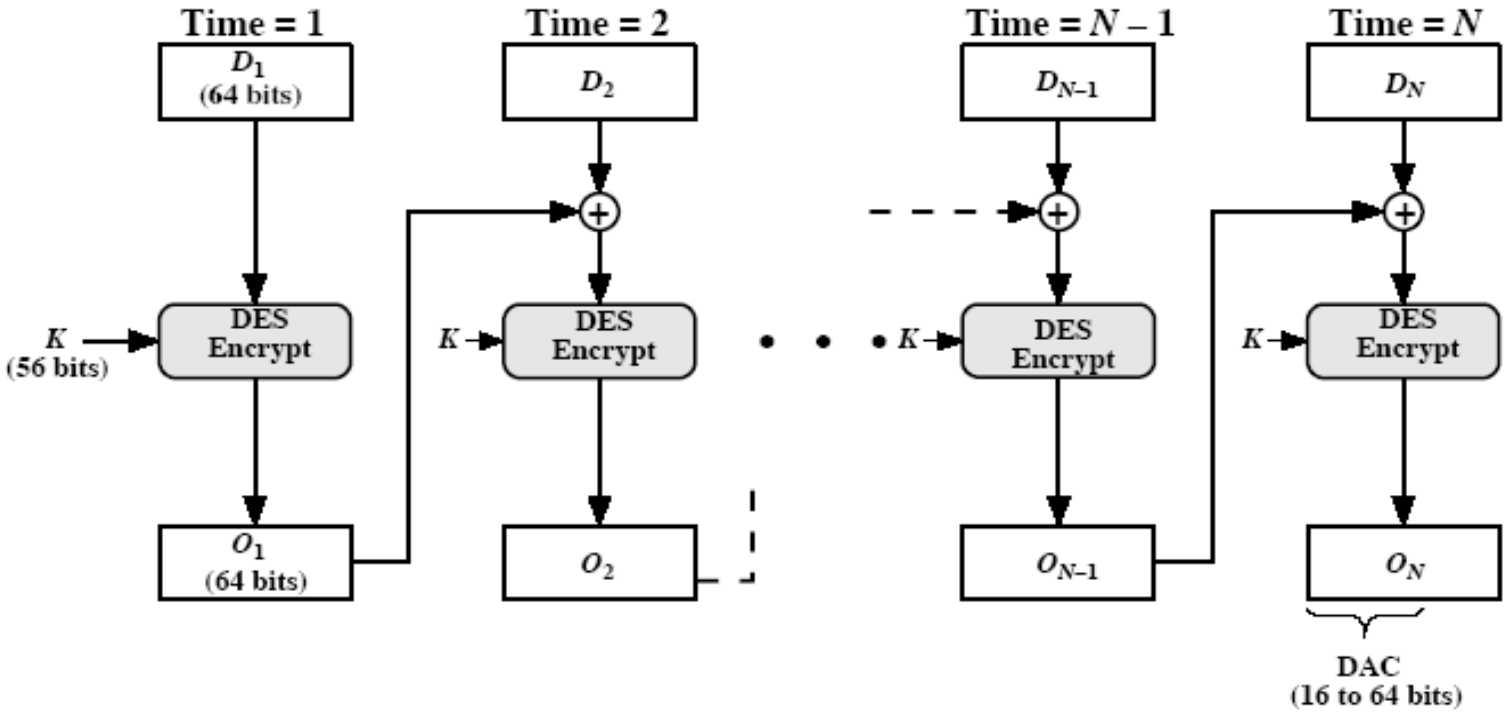
–MACs should be uniformly distributed

–MAC should depend equally on all bits of the message

Using Symmetric Ciphers for MACs

- Can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
  - or the leftmost M bits ($16 \leq M \leq 64$) of final block
- But final MAC is now too small for security

Data Authentication Algorithm



Hash Function Properties

- A Hash Function produces a fingerprint of some file/message/data

$$h = H(M)$$

–condenses a variable-length message M

–to a fixed-sized fingerprint

- Assumed to be public
- Typically quite complex

Requirements for Hash Functions

- Can be applied to any sized message M
- Produces fixed-length output h
- Is easy to compute h=H(M) for any message M
- Given h is infeasible to find x s.t. H(x)=h
  - one-way property
- Given x is infeasible to find y s.t. H(y)=H(x)
  - weak collision resistance
- Is infeasible to find any x,y s.t. H(y)=H(x)
  - strong collision resistance

Simple Hash Functions

- Several proposals for simple functions
- Based on XOR of message blocks
- Not secure since can manipulate any message and either not change hash or change hash also
- Need a stronger cryptographic function (next chapter)

Simple Hash Functions

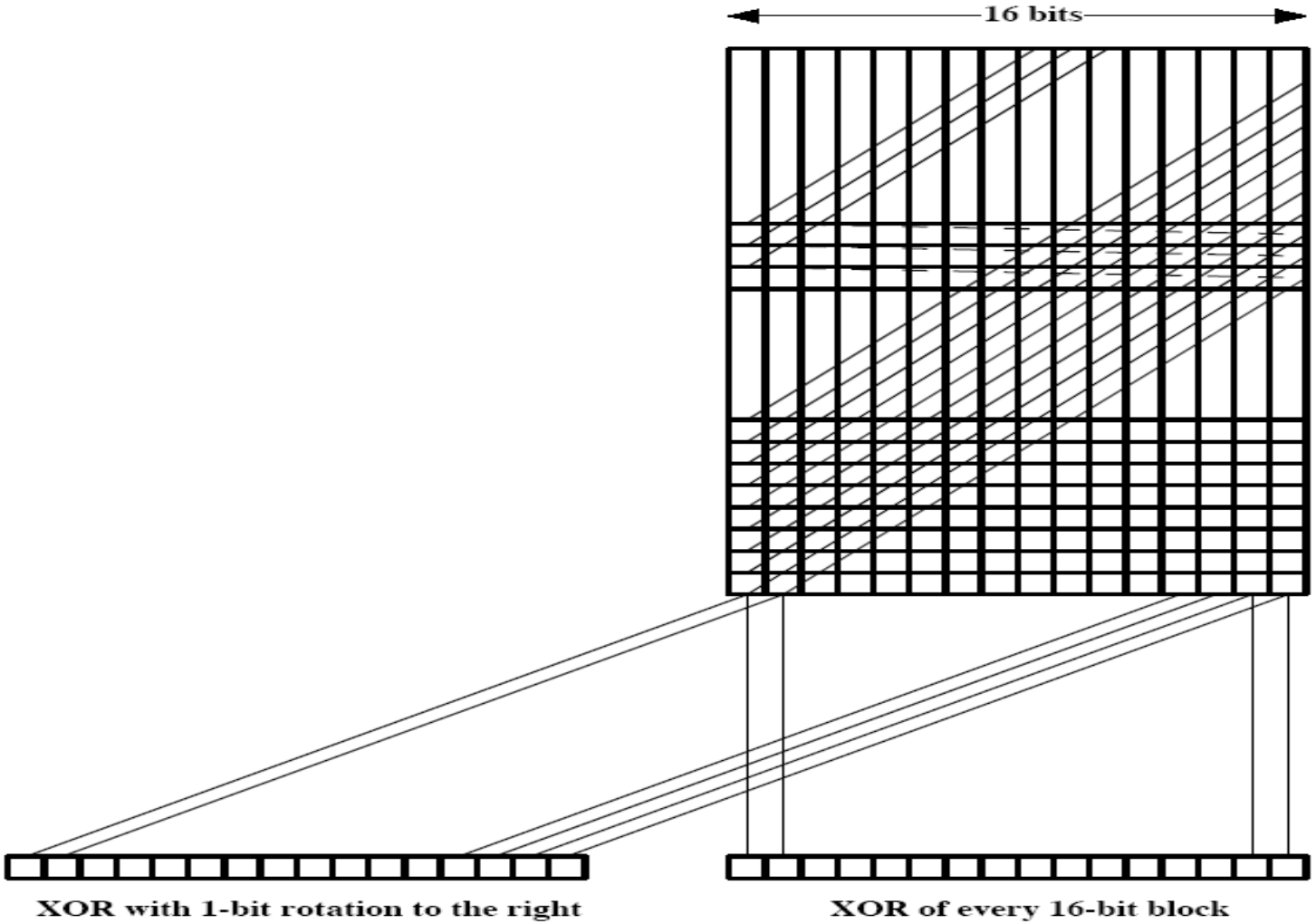
	bit 1	bit 2	• • •	bit n
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block m	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

A simple Hash Function using Bit-wise XOR

Simple Hash Functions

- A simple way to improve performance is to perform one-bit circular shift, on the hash value after each block is processed
- Initially set the n-bit hash value to zero
- Process each successive n-bit block of data as follows
  - Rotate the current hash value to the left by one bit
  - XOR the block into the hash value

Two Simple Hash Functions



Birthday Attacks

- One might think a 64-bit hash is secure
- But by Birthday Paradox is not
- Birthday attack works as

–opponent generates $2^{m/2}$ variations of a valid message all with essentially the same meaning

–opponent also generates $2^{m/2}$ variations of a desired fraudulent message

–two sets of messages are compared to find pair with same hash (probability > 0.5 by birthday paradox)

–have user sign the valid message, then substitute the forgery which will have a valid signature

- Conclusion: need to use larger MACs

Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}

Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}

Northern {European / Europe} {area / division} . He {will take / has taken} over {the / --}

responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}

in the {area / region} . Please {afford / give} him {every / all the} help he {may need / needs}

to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the

market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the

{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}

of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}

as proof of identity. An order with his signature, which is {appended / attached}

{authorizes / allows} you to charge the cost to this company at the {above / head office}

address. We {fully / --} expect that our {level / volume} of orders will increase in

the {following / next} year and {trust / hope} that the new appointment will {be / prove}

{advantageous / an advantage} to both our companies.

Block Ciphers as Hash Functions

- Can use block ciphers as hash functions but without secret key
  - –using H_0=0 and zero-pad of final block
  - –compute: $H_i = E_{M_i}[H_{i-1}]$
  - –and use final block as the hash value
  - –similar to CBC but without a key
- Resulting hash is too small (64-bit)
  - –both due to direct birthday attack
  - –and to "man-in-the-middle" attack
- Other variants also susceptible to attack
  - –Some form of birthday attack will succeed against any hash scheme involving CBC without a secret key unless
    - either the resulting hash code is small enough
    - or a larger hash code can be decomposed into independent subcodes

Block Ciphers as Hash Functions

- Other variants also susceptible to attack

–Some form of birthday attack will succeed against any hash scheme involving CBC without a secret key unless

- either the resulting hash code is small enough
- Or a larger hash code can be decomposed into independent sub-codes

Hash Functions & MAC Security

- Like block ciphers have:
- Brute-force attacks exploiting

–strong collision resistance hash have cost $2^{m/2}$

- have proposal for h/w MD5 cracker
- 128-bit hash looks vulnerable, 160-bits better

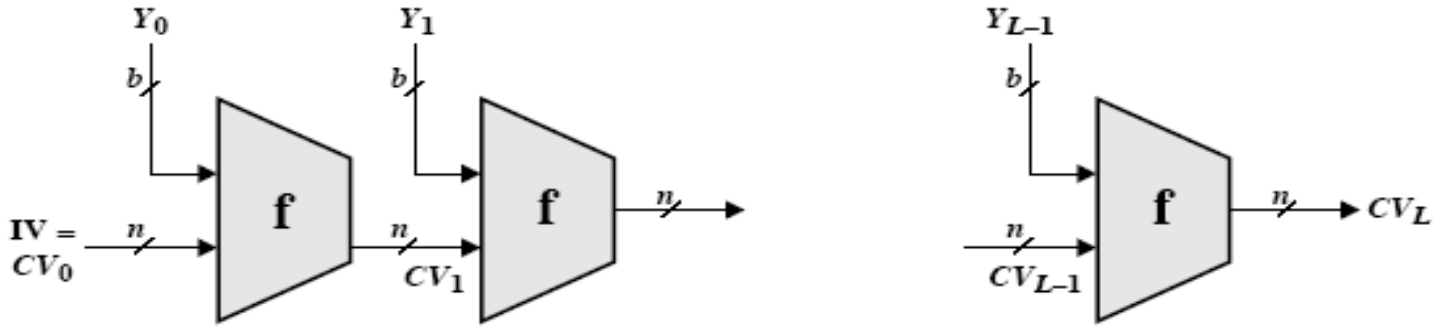
–MACs with known message-MAC pairs

- can either attack key-space (cf key search) or MAC
- at least 128-bit MAC is needed for security

Hash Functions & MAC Security

- Cryptanalytic attacks exploit structure
  - like block ciphers want brute-force attacks to be the best alternative
- Have a number of analytic attacks on iterated hash functions
  - $CV_i = f[CV_{i-1}, M_i]$; $H(M)=CV_N$
  - typically focus on collisions in function f
  - like block ciphers is often composed of rounds
  - attacks exploit properties of round functions

General Structure of Secure Hash Code



IV = Initial value
CV = chaining variable
Y_i = ith input block
f = compression algorithm
L = number of input blocks
n = length of hash code
b = length of input block

Summary

- Message authentication using
- Message encryption
- MACs
- Hash functions
- General approach & security

Any question ?