Message Authentication and Hash Functions

Message Authentication

- Message authentication is concerned with:
  - Protecting the integrity of a message
  - Validating identity of originator
  - Non-repudiation of origin (dispute resolution)
- Will consider the security requirements
- Three alternative functions used:
  - **Message encryption**
  - **Message Authentication Code (MAC)**
  - **Hash functions**

Message Authentication Requirements

In communications across a network, the following attacks could be identified:

- **Disclosure:** Release of message contents
- **Traffic analysis:** Discovery of the pattern of traffic between parties
- **Masquerade:** Insertion of messages into the network from a fraudulent source
- **Content modification:** Modification of the contents of a message

Message Authentication Requirements

- **Sequence modification:** Modification to a sequence of messages between parties
- **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 Function

- Message Encryption
- Message Authentication Code (MAC)
- Hash Function

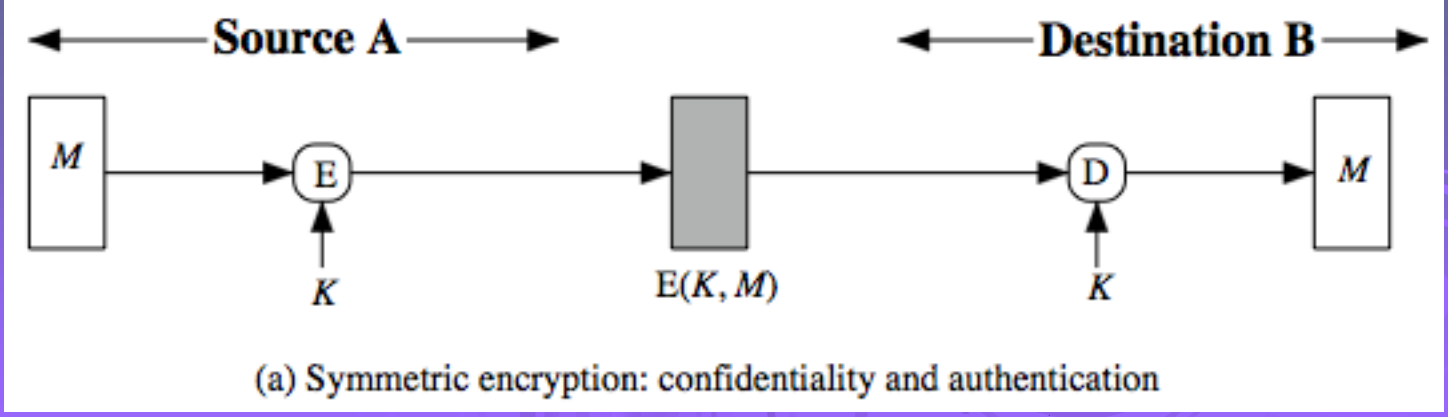
Message Encryption

- Message encryption by itself also provides a measure of authentication
- The analysis differs for symmetric and public key encryption schemes.

Symmetric Message Encryption

- Encryption can also provides authentication
- If symmetric encryption is used then:

- Receiver know sender must have created it
- Since only sender and receiver use key
- If message has suitable structure, redundancy or a checksum to detect any changes



(a) Symmetric encryption: confidentiality and authentication

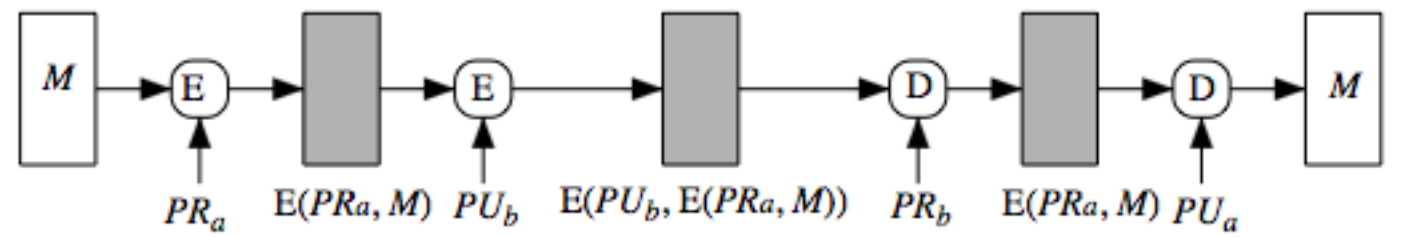
Public-key Message Encryption

- If public-key encryption is used:

- Anyone potentially knows public-key

However if

- Sender signs message using their private-key
- Then encrypts with recipients public key
- Have both secrecy and authentication
- Again need to recognize corrupted messages
- But at cost of two public-key uses on message



(d) Public-key encryption: confidentiality, authentication, and signature

Message Authentication Codes (MAC)

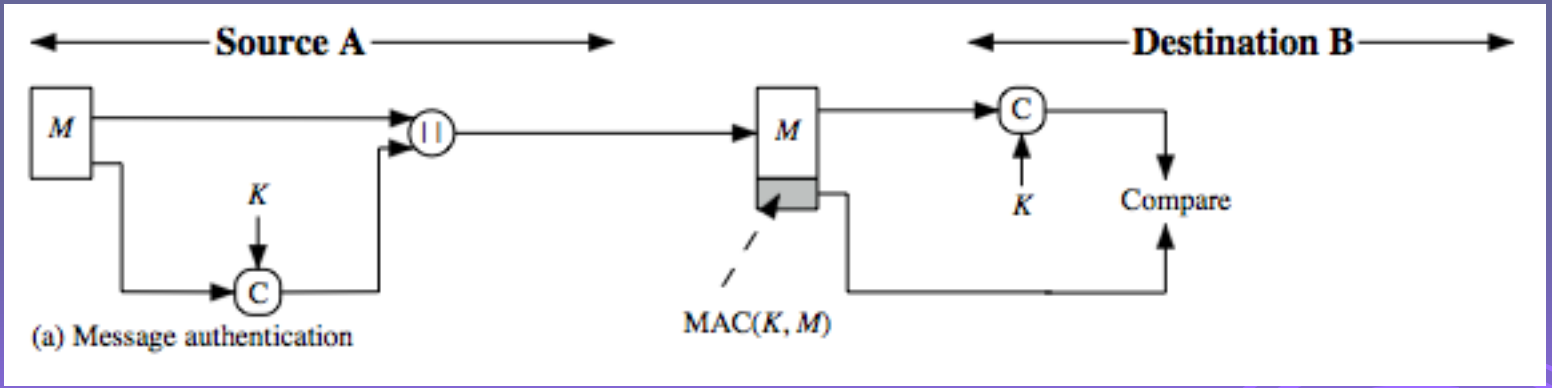
- Why use a MAC?
  - sometimes only authentication is needed
  - sometimes need authentication to persist longer than the encryption (eg. archival use)
- provides authentication
- Can combine use of MAC with encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before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

Message Authentication Code (MAC)

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

Message Authentication Code

- A small fixed-sized block of data
- Generated from message + secret key
- MAC = C(K,M)
- Appended to message when sent



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
- Is 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:
  1. knowing a message and MAC, is infeasible to find another message with same MAC
  2. MACs should be uniformly distributed
  3. 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≤M≤64) of final block
- But final MAC is now too small for security

Hash Functions

- Want a cryptographic hash function
  - Computationally infeasible to find data mapping to specific hash (one-way property)
  - Computationally infeasible to find two data to same hash (collision-free property)

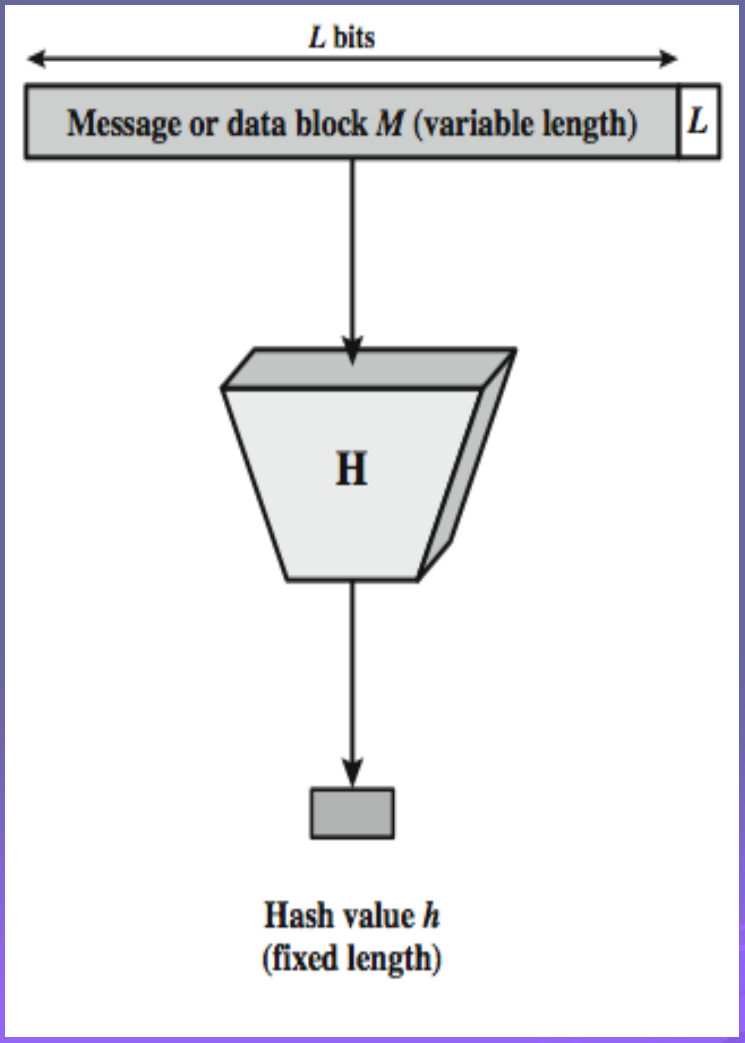
Hash Functions

- Condenses arbitrary message to fixed size

  `h = H(M)`

- Usually assume that the hash function is public and not keyed
  - cf. MAC which is keyed
- Hash used to detect changes to message
- Can use in various ways with message
- Most often to create a digital signature

Cryptographic Hash Function



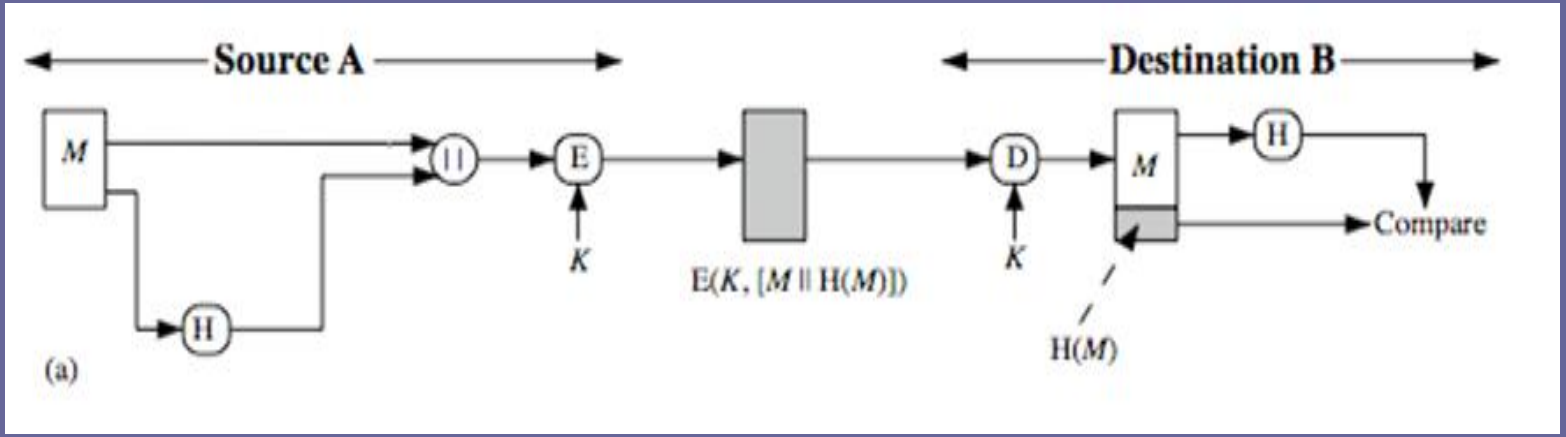
- Figure depicts the general operation of a cryptographic hash function.
- The input is padded out to an integer multiple of some fixed length (e.g., 1024 bits) and **the padding includes the value of the length of the original message in bits.**
- The length field is a security measure to increase the difficulty for an attacker to produce an alternative message with the same hash value.

Hash Functions & Message Authentication

- Message authentication is a mechanism or service used to verify the integrity of a message, by assuring that the data received are exactly as sent.

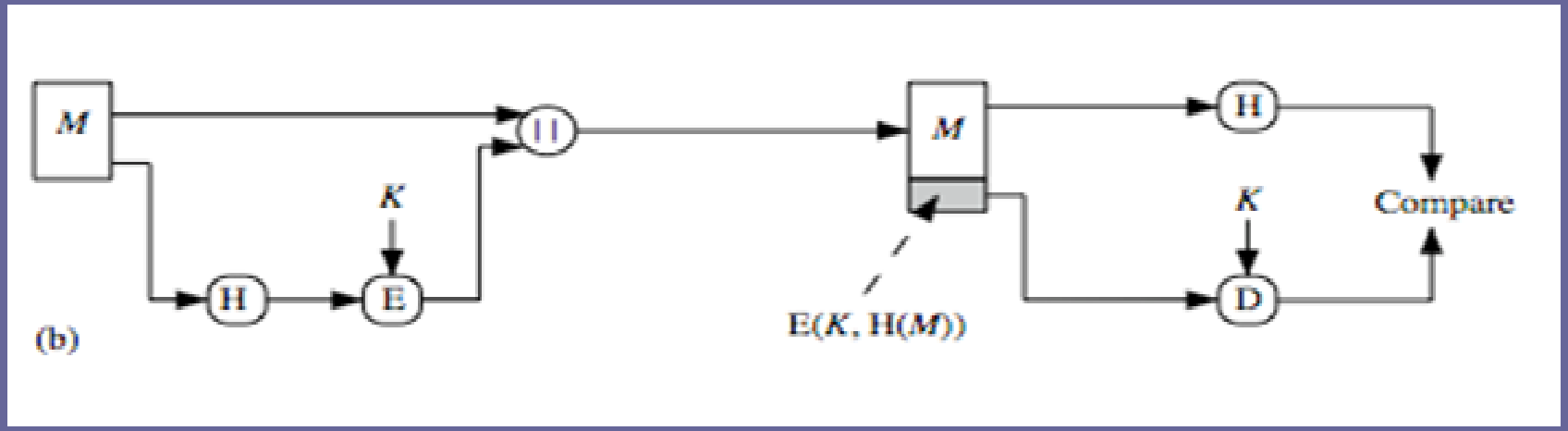
- Figure A, B, C and D illustrates a variety of ways in which a hash code can be used to provide message authentication, as follows:

A



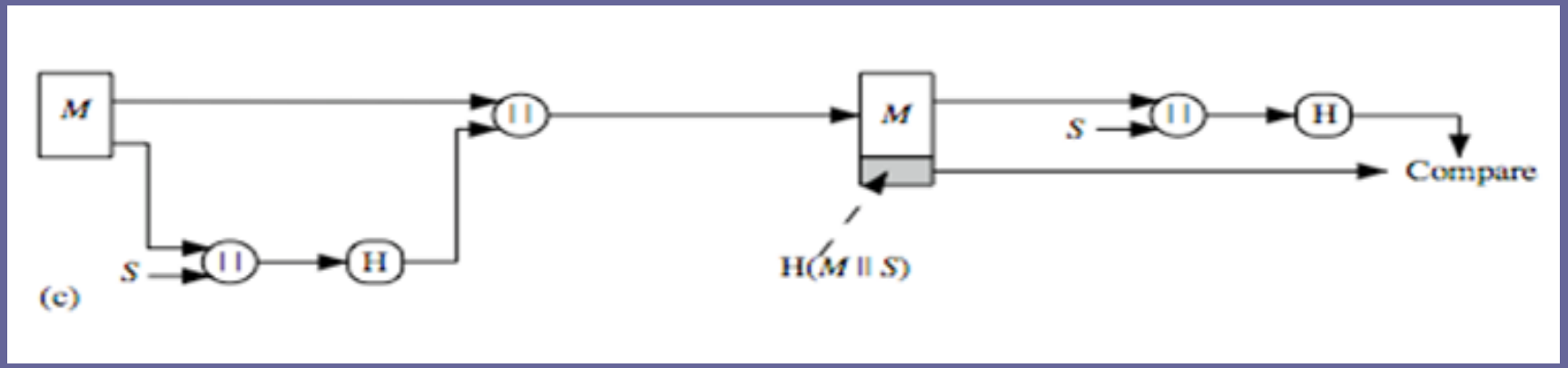
- The message plus concatenated hash code is encrypted using symmetric encryption. Since only A and B share the secret key, the message must have come from A and has not been altered.
- The hash code provides the structure or redundancy required to achieve authentication.

B



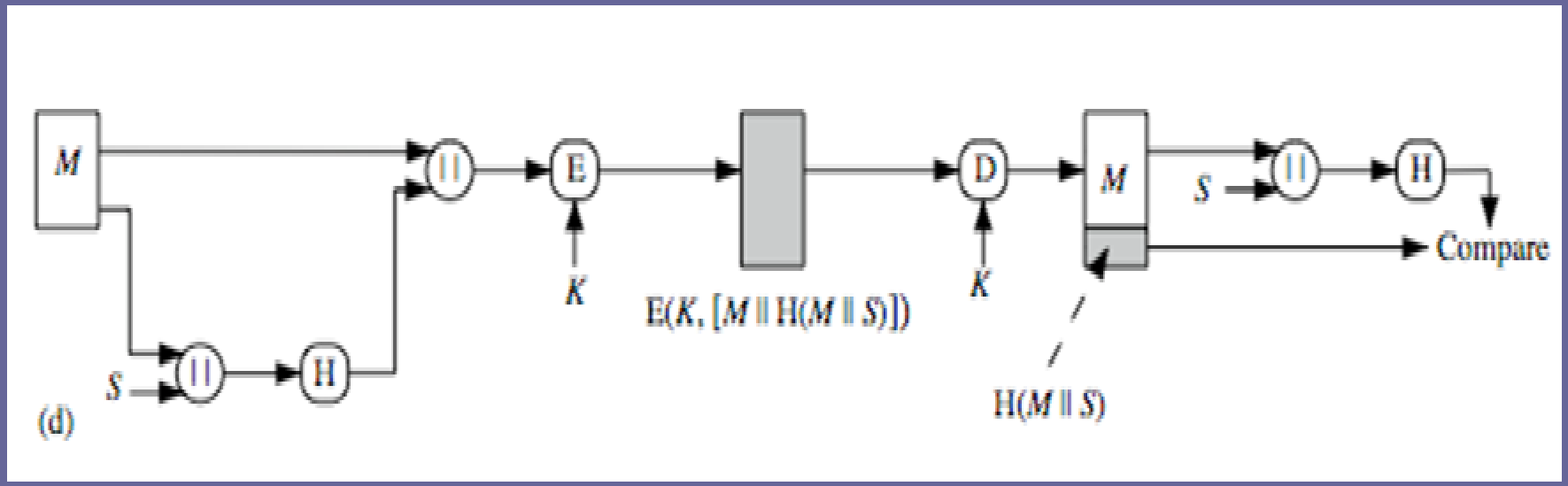
- Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications not requiring confidentiality.

C



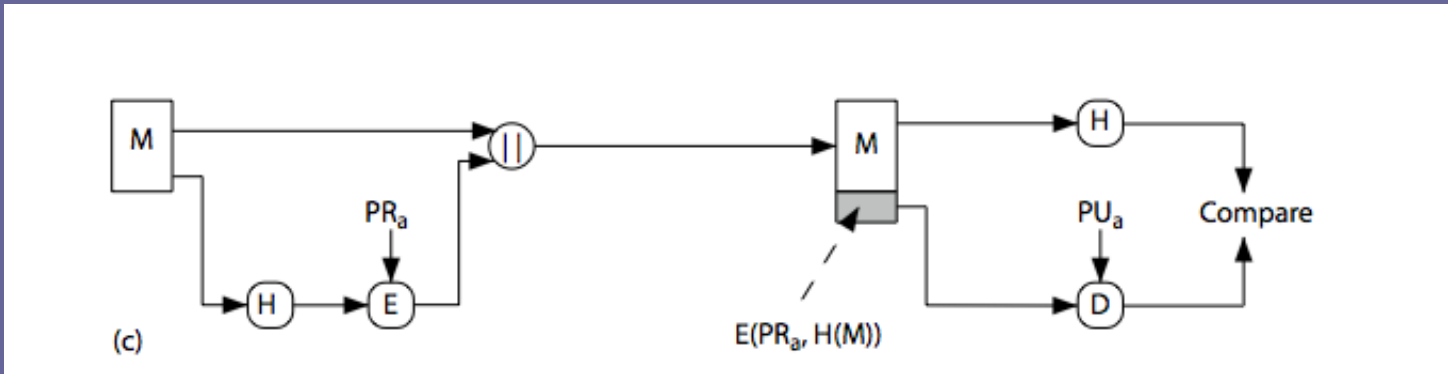
- Shows the use of a hash function but no encryption for message authentication.
- The technique assumes that the two communicating parties share a common secret value S.
- A computes the hash value over the concatenation of M and S and appends the resulting hash value to M.
- Because B possesses S, it can re-compute the hash value to verify.
- Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.

D



- Confidentiality can be added to the approach of (c) by encrypting the entire message plus the hash code.

Hash Functions & Digital Signatures



Requirements for Hash Functions

1. Can be applied to any sized message `M`
2. Produces fixed-length output `h`
3. Is easy to compute `h=H(M)` for any message `M`
4. Given `h` is infeasible to find `x`

   `H(x)=h`

   - one-way property

1. Given `x` is infeasible to find `y`

   `H(y)=H(x)`

   - weak collision resistance

1. Is infeasible to find any `x,y`

   `H(y)=H(x)`

   - strong collision resistance

Simple Hash Functions

- Are 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)

Birthday Attacks

- Might think a 64-bit hash is secure
- But by **Birthday Paradox** is not
- **Birthday attack** works thus:
  - 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 is that need to use larger MAC/hash

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.

Figure 11.9 A Letter in 2^{37} Variations [DAVI89]

Block Ciphers as Hash Functions

- Can use block ciphers as hash functions
  - 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 “meet-in-the-middle” attack
- Other variants also susceptible to attack

Hash Functions & MAC Security

- Bike 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 keyspace (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