At cats’ green on the Sunday he took the message from the inside of the pillar and added Peter Moran's name to the two names already printed there in the "Brontosaurus" code. The message now read: “Leviathan to Dragon: Martin Hillman, Trevor Allan, Peter Moran: observe and tail.” What was the good of it John hardly knew. He felt better, he felt that at last he had made an attack on Peter Moran instead of waiting passively and effecting no retaliation. Besides, what was the use of being in possession of the key to the codes if he never took advantage of it?

―Talking to Strange Men, Ruth Rendell
Message Authentication

• protecting message content (ie secrecy) by encrypting the message
• now consider
  – how to protect message integrity (ie protection from modification)
  – confirming the identity of the sender
• then three alternative functions used:
  – message encryption (the ciphertext itself is the authenticator)
  – message authentication code (MAC)
  – hash function
Security Attacks

- disclosure of message contents
- traffic analysis (discover the pattern)
- Masquerade (insert a msg from a fraudulent source)
- content modification
- sequence modification (insert, delete, reorder)
- timing modification (delay or replay)
- source repudiation (denial of a transmission)
- destination repudiation (denial of a receipt)
Message Encryption

• message encryption by itself also provides a measure of authentication
• if symmetric encryption is used then:
  – receiver know sender must have created it
  – since only sender and receiver now key used
  – know content cannot of been altered
  – if message has suitable structure, redundancy or a checksum to detect any changes
Message 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 recipients public key
    • have both secrecy and authentication
  – again need to recognize corrupted messages
  – but at cost of two public-key uses on message
Figure 11.1 Basic Uses of Message Encryption

(a) Symmetric encryption: confidentiality and authentication

(b) Public-key encryption: confidentiality

(c) Public-key encryption: authentication and signature

(d) Public-key encryption: confidentiality, authentication, and signature
Message Authentication Code (MAC)

• generated by an MAC function $C$ that creates a small fixed-sized block
  – depending on both message $M$ and a shared secret key $K$, $\text{MAC}=C_K(M)$
  – MAC is appended to the message $M$

• 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) Message authentication

\[ C \]

\[ C_K(M) \]

\[ K \]

\[ K \]

Compare
Message Authentication Codes

• can also use encryption for secrecy
  – generally use separate keys for each
  – can compute MAC either before or after encryption
  – is generally regarded as better done before

• why use a MAC?
  – MAC is much less expensive than en/decryption
  – sometimes only authentication is needed
  – One end with a heavy load, check MAC selectively
MAC Properties

• a MAC is a cryptographic checksum

\[ \text{MAC} = C_K (M) \]

– 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
  • 100-bit M, and 20-bit MAC
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
Figure 11.6  Data Authentication Algorithm (FIPS PUB 113)
Hash Functions

• condenses arbitrary message to fixed size
• usually assume that the hash function is public and not keyed
  – cf. MAC which is keyed
• used to detect changes to message
• can use in various ways with message
• most often to create a digital signature
Hash Functions & Digital Signatures
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
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 $x$ s.t. $H(x) = h$
   - one-way property
5. given $x$ is infeasible to find $y$ s.t. $H(y) = H(x)$
   - weak collision resistance
6. is infeasible to find any $x, y$ s.t. $H(y) = H(x)$
   - 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 to produce a given hash
• need a stronger cryptographic function (next chapter)
<table>
<thead>
<tr>
<th></th>
<th>bit 1</th>
<th>bit 2</th>
<th>...</th>
<th>bit n</th>
</tr>
</thead>
<tbody>
<tr>
<td>block 1</td>
<td>$b_{11}$</td>
<td>$b_{21}$</td>
<td>...</td>
<td>$b_{n1}$</td>
</tr>
<tr>
<td>block 2</td>
<td>$b_{12}$</td>
<td>$b_{22}$</td>
<td>...</td>
<td>$b_{n2}$</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>block m</td>
<td>$b_{1m}$</td>
<td>$b_{2m}$</td>
<td>...</td>
<td>$b_{nm}$</td>
</tr>
<tr>
<td>hash code</td>
<td>$C_1$</td>
<td>$C_2$</td>
<td>...</td>
<td>$C_n$</td>
</tr>
</tbody>
</table>

**Figure 11.7** Simple Hash Function Using Bitwise XOR
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 MACs
Dear Anthony,

[This letter is] to introduce [you to] Mr. Alfred [P.]

Barton, the [newly appointed] chief jewellery buyer for [our]
Northern [European] area. He will take over [the]
responsibility for all our interests in watches and jewellery
in the [area, region]. Please [afford] him every help he [may need]
to seek out the most modern lines for the top end of the market. He is empowered to receive on our behalf samples of the latest [watch and jewellery] products, up to a limit of ten thousand dollars. He will hold a signed copy of this letter as proof of identity. An order with his signature, which is appended, authorizes you to charge the cost to this company at the head office address. We expect that our volume of orders will increase in the next year and hope that the new appointment will prove advantageous to both our companies.

Figure 11.9 A Letter in $2^3$ 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)
  – due to direct birthday attack and variants
Hash Functions & MAC Security

- like block ciphers have:
- **brute-force** attacks exploiting
  - strong collision resistance hash have cost $2^{m/2}$
    - 128-bit hash looks vulnerable, 160-bits better
  - MACs with known message-MAC pairs
    - can either attack keyspace (cf key search) or MAC
      - $\text{Min}(2^k, 2^n)$
    - at least 128-bit MAC and 128-bit key 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
  - $\text{CV}_i = f[\text{CV}_{i-1}, M_i]; \ H(M) = \text{CV}_N$
  - typically focus on collisions in function $f$
  - like block ciphers is often composed of rounds
  - attacks exploit properties of round functions
Summary

• have considered:
  – message authentication using
  – message encryption
  – MACs
  – hash functions
  – general approach & security