**Informally describe the main security properties of digital signatures, message authentication code and cryptographic hash functions.**

* **Digital signature**

1. **Confidentiality**: Information is kept secret from all but authorized parties.

2. **Integrity**: Messages have not been modified in transit.

3. **Message Authentication**: The sender of a message is authentic.

An alternative term is data origin authentication.

4. **Nonrepudiation**: The sender of a message can not deny the creation of the message.

5. **Identification/entity authentication**: Establish and verify the identity of an entity,

e.g., a person, a computer or a credit card.

6. **Access control**: Restrict access to the resources to privileged entities.

7. **Availability**: Assures that the electronic system is reliably available.

8. **Auditing**: Provide evidence about security-relevant activities, e.g.,

by keeping logs about certain events.

9. **Physical security**: Provide protection against physical tampering and/or responses

to physical tampering attempts.

10. **Anonymity**: Provide protection against discovery and misuse of identity

* **Message authentication code**

1. **Cryptographic checksum** A MAC generates a cryptographically secure authentication tag

for a given message.

1. **Symmetric** MACs are based on secret symmetric keys. The signing and verifying

parties must share a secret key.

3. **Arbitrary message size** MACs accept messages of arbitrary length.

4. **Fixed output length** MACs generate fixed-size authentication tags.

5. **Message integrity** MACs provide message integrity: Any manipulations

of a message during transit will be detected by the receiver.

6. **Message authentication** The receiving party is assured of the origin of the message.

7. **No nonrepudiation** Since MACs are based on symmetric principles,

they do not provide nonrepudiation.

* **cryptographic hash functions**

1. **Arbitrary message size** h(x) can be applied to messages x of any size.

2. **Fixed output length** h(x) produces a hash value z of fixed length.

3. **Efficiency** h(x) is relatively easy to compute.

4. **Preimage resistance** For a given output z, it is impossible to find any input x

such that h(x) = z, i.e, h(x) is one-way.

5. **Second preimage resistance** Given x1, and thus h(x1), it is computationally infeasible

to find any x2 such that h(x1) = h(x2).

6. **Collision resistance** It is computationally infeasible to find any pairs x1 = x2

such that h(x1) = h(x2).

**Recall and explain some specific recent examples of security problems caused by the use of weak or poorly implemented cryptography.**

Understanding Cryptography: Page 322: Attacks Against Secret Prefix MACs

“extend block attach”

Understanding Cryptography: Page 299: Birthday Attack

“Hash Collision”: 概率论