



# Deploying a New Hash Algorithm

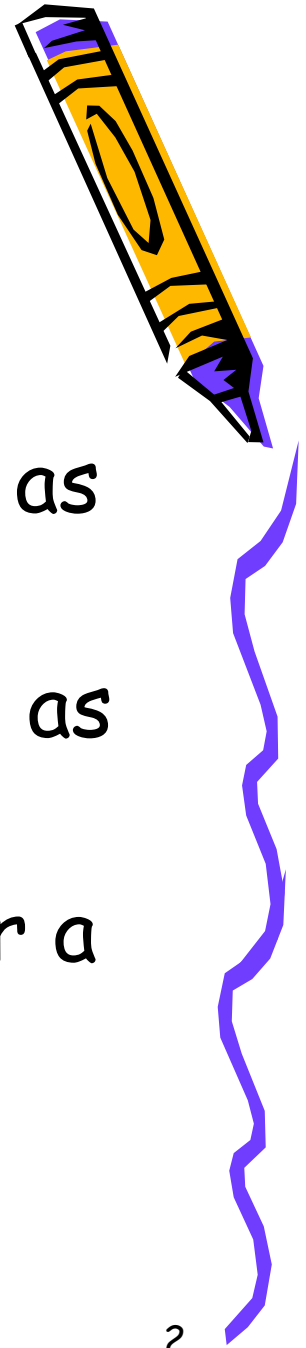
Presented

By

Archana Viswanath

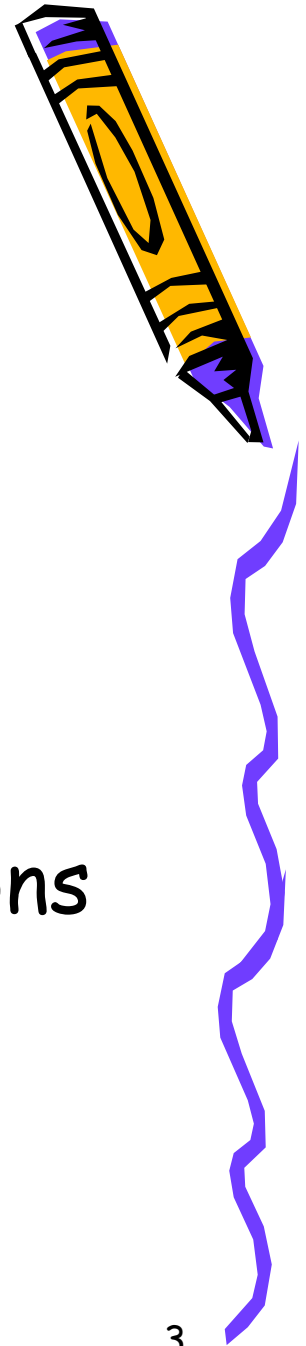
# function?

- Hash function
  - takes a message of any length as input
  - produces a fixed length string as output
  - termed as a **message digest** or a **digital fingerprint**.



# What is a Cryptographic Hash function?

- Cryptographic Hash function
  - hash function with certain additional security properties
  - used as a primitive in various information security applications
  - Provides authentication and message integrity

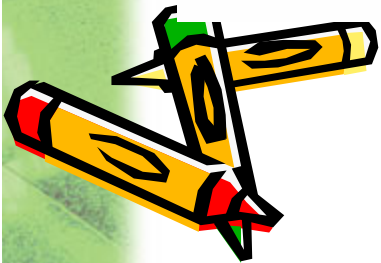
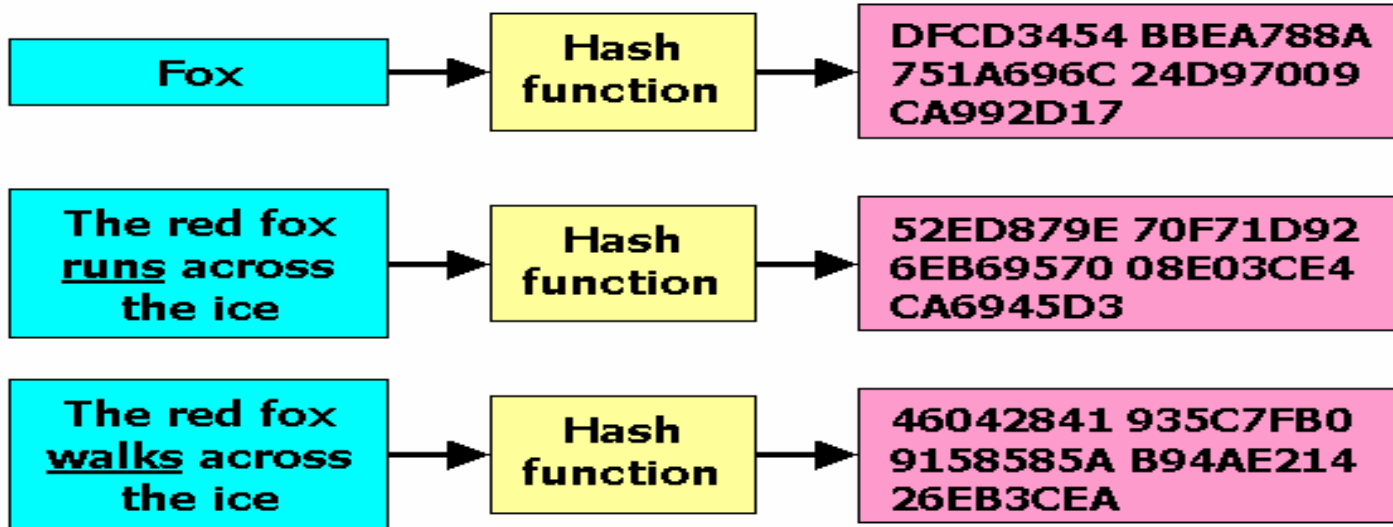


# Working of a Hash Function



**Input**

**Hash sum**



# Prerequisites for a Cryptographic Hash function



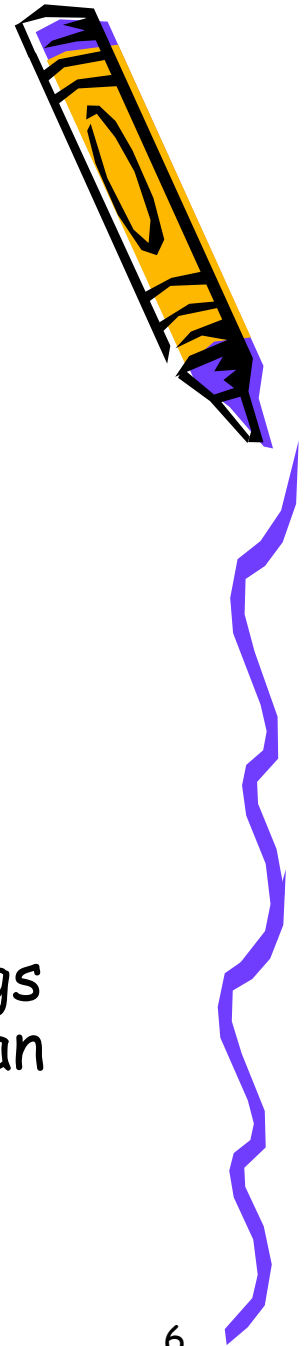
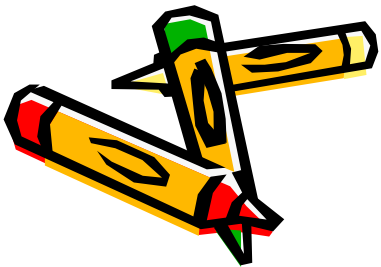
- **Preimage resistant**- given  $h$  it should be hard to find any  $m$  such that  $h = \text{hash}(m)$ .
- **Second preimage resistant**: given an input  $m_1$ , it should be hard to find another input,  $m_2$  (not equal to  $m_1$ ) such that  $\text{hash}(m_1) = \text{hash}(m_2)$ . This property is implied by collision-resistance.
- **Collision-resistant**: given  $\text{hash}(m_1)$ , it should be hard to find a message  $m_2$  such that  $\text{hash}(m_1) = \text{hash}(m_2)$ .



# Cryptographic Hash function considered insecure?

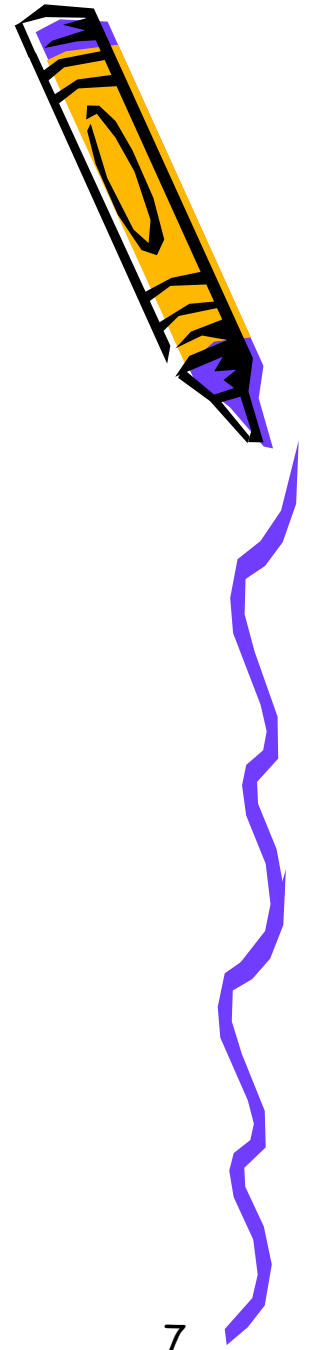
- Finding a (previously unseen) message that matches a given digest
- Finding "collisions", wherein two different messages have the same message digest.

An attacker who can do either of these things might, for example, use them to substitute an unauthorized message for an authorized one



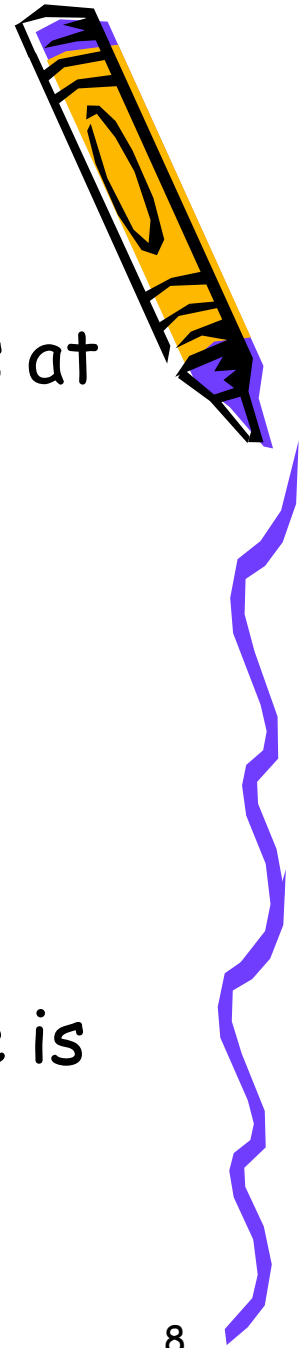
# Uses of Hash Functions

- Digital Signature
- HMAC
- Pseudo-Random functions
- Data Fingerprinting



# Overview of Hash Transition Problem

- We have to deploy new hash functions at some point soon.
- It is a special case of the protocol transition problem.
- Certificates rely on hashes
- Goal: maintain security while new code is deployed



# Protocol Transitional Environment

## Three types of Agents

- Old - Agents which only speak the older version
- Switch-hitting - Agents which can speak both versions
- New - Agents which can only speak the new version.



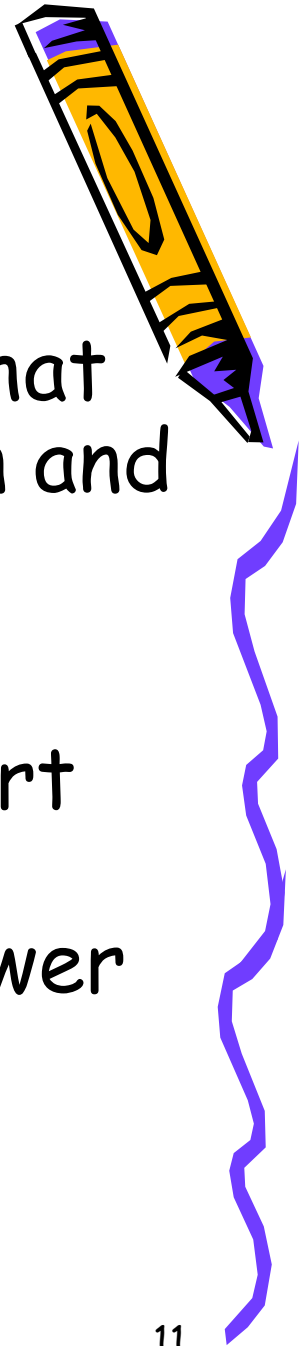
# Backward Compatibility

- Old agents and switch-hitting units should be able to communicate using the older version.
- **General approach** - deploy as many switch-hitting implementations and then transition
- **Interactive protocols** - Switch-hitting recognize that it is speaking to an old peer and fall back.
- **Non-Interactive protocols** - transmit messages with old implementation.



# Newest Common Version

- **Interactive Protocols** - detect that both peers speak the new version and use it.
- **Non-Interactive Protocols** - Start speaking older version and then advertise the support for the newer version.



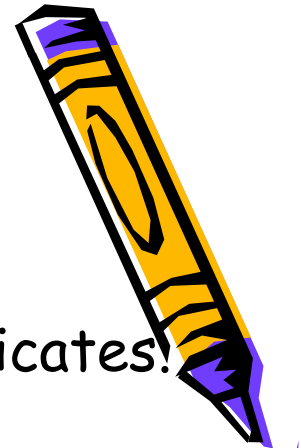
# Downgrade Protection

- Downgrade Attack(SSLv2) - Use MAC



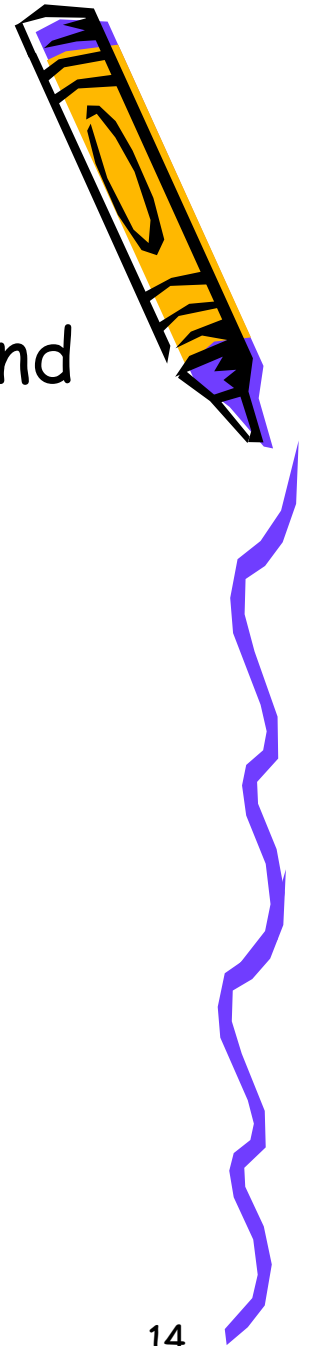
# Credentials versus Implementation

- Peers public keys are authenticated using certificates.
- Upgrade of protocol is uncoupled from certificate issuance.
- Possibility - User has a security protocol implementation which understands SHA-256 but a certificate which was digested with SHA-1—or indeed, one certificate digested with SHA-1 and one with SHA-256.
- Certificate and protocol version capabilities need to be dealt with separately.



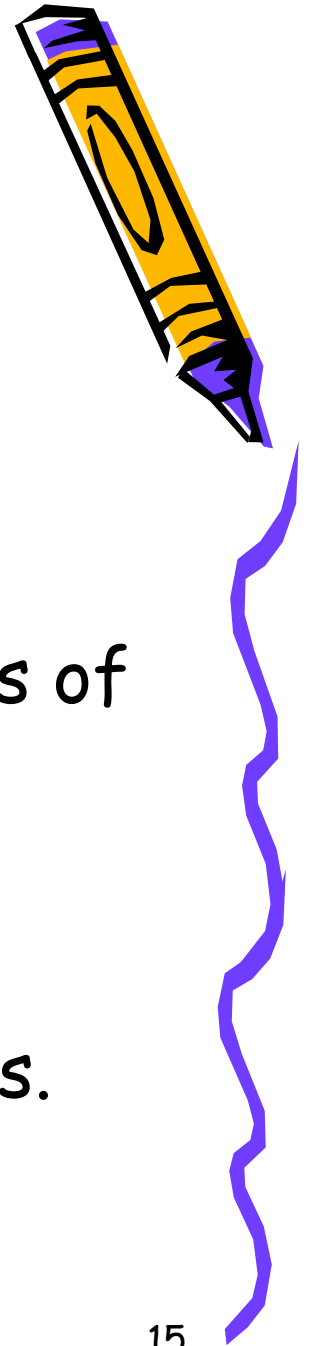
# Internet Mail Extensions

- It is a standard message encryption and authentication protocol.
- Functioning
  - Public Key Cryptography for key establishment
  - Symmetric Cryptography for bulk encryption
  - Digital Signatures for message authentication/non-repudiation.



# S/MIME - Types of Clients

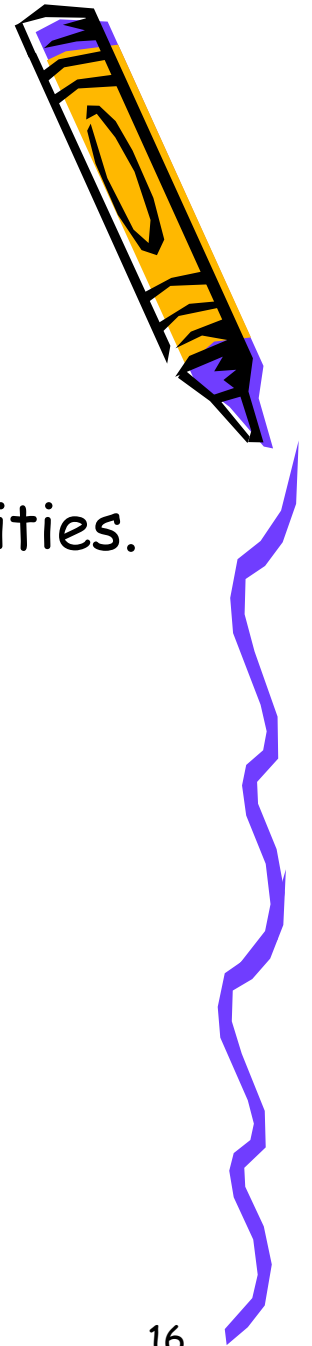
- Old clients.
- Switch-hitting clients with only old certificates.
- Switch-hitting clients with both types of certificate.
- Switch-hitting clients with new certificates.
- New clients with only new certificates.



# S/MIME - Initial Message

Sending without a recipient certificate

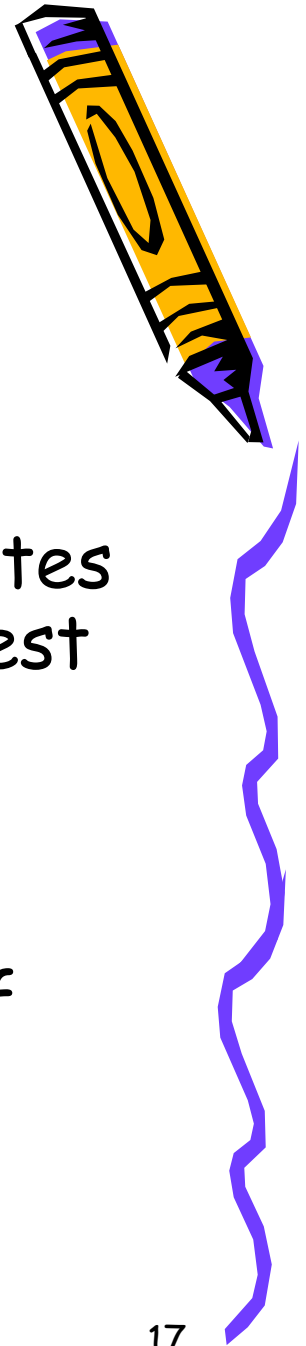
- No public key to encrypt
- No information about the recipient's capabilities.
- Choice of Certificate?
  - If sender has only one certificate
  - If sender has two certificates
- Choice of Digest Algorithm?



# S/MIME - Initial Message

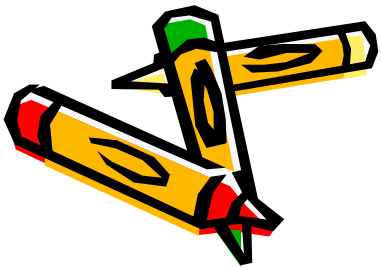
Sending with the recipient's certificate

- If the recipient has multiple certificates then the one created with the strongest algorithm should be used.
- *SMIMECapabilities* - indicates which algorithm the recipient was capable of using.



# Diffie-Hellman Key Agreement

- RFC 2631 [Res99] specified a method for Diffie-Hellman (DH) key agreement in which SHA-1 is used as a PRF (pseudo-random function) to compute encryption keys from the DH shared secret.



# Take Away

- TIME TO MOVE ON!!!

