Introduction to Cryptography CS 355

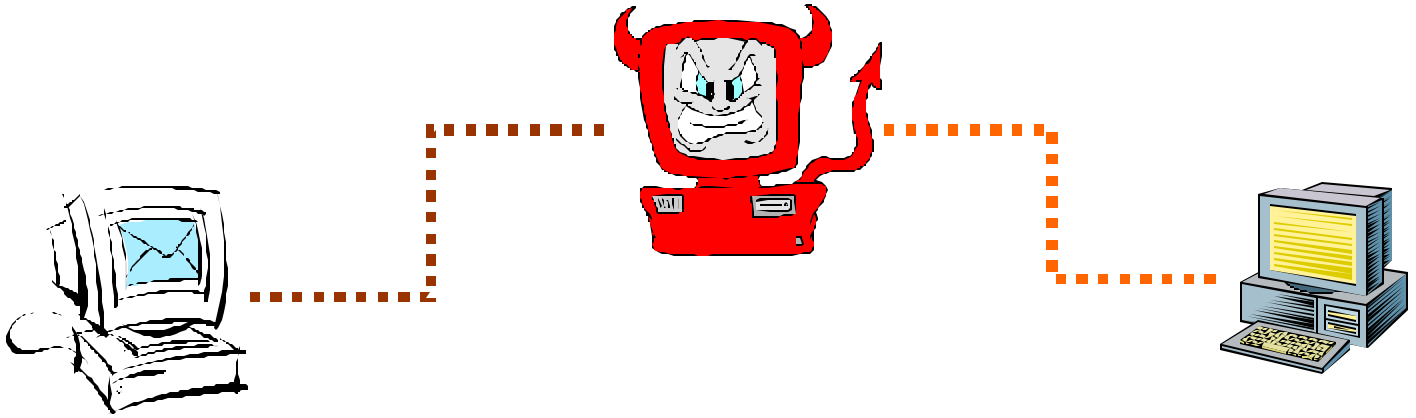
Lecture 27

Cryptographic Hash Functions

Lecture Outline

- Hash functions
- Security properties
- MD5 & SHA1

Data Integrity and Source Authentication



- Encryption does not protect data from modification by another party.
- Need a way to ensure that data arrives at destination in its original form as sent by the sender and it is coming from an authenticated source.

Cryptographic Hash Functions

- A hash function maps a message of an arbitrary length to a m-bit output
  - output known as the fingerprint or the message digest
  - if the message digest is transmitted securely, then changes to the message can be detected
- A hash is a many-to-one function, so collisions can happen.

Security Requirements for Cryptographic Hash Functions

Given a function $h:X \rightarrow Y$, then we say that h is:

- preimage resistant (one-way):

  if given $y \in Y$ it is computationally infeasible to find a value $x \in X$ s.t. $h(x) = y$

- 2-nd preimage resistant (weak collision resistant):

  if given $x \in X$ it is computationally infeasible to find a value $x' \in X$, s.t. $x' \neq x$ and $h(x') = h(x)$

- collision resistant (strong collision resistant):

  if it is computationally infeasible to find two distinct values $x', x \in X$, s.t. $h(x') = h(x)$

Uses of hash functions

- Message authentication
- Software integrity
- One-time Passwords
- Digital signature
- Timestamping

Bruteforce Attacks on Hash Functions

- Attacking one-wayness
  - Goal: given $h:X \rightarrow Y$, $y \in Y$, find x such that $h(x)=y$
  - Algorithm: pick a random set X_0 of q values in X,
    for each $x \in X_0$, return x if $h(x)=y$,
    after all q values have been evaluated, return fail
  - When h is a random instance of all functions mapping X to Y, the average-case success probability is
    $$e = 1 - \left(1 - \frac{1}{|Y|}\right)^q \approx \frac{q}{|Y|}$$
  - Let $|Y|=2^m$, to get ε to be close to 0.5, $q \approx 2^{m-1}$

Bruteforce Attacks on Hash Functions

- Attacking collision resistance
  - Goal: given h, find x, x' such that h(x)=h(x')
  - Algorithm: pick a random set X_0 of q values in X
    for each $x \in X_0$, computes $y_x=h(x)$
    if $y_x=y_{x'}$ for some $x' \neq x$ then return (x,x') else fail
  - The average success probability is $1-e^{-\frac{q(q-1)}{2|Y|}}$
  - Let $|Y|=2^m$, to get ε to be close to 0.5, $q \approx 2^{m/2}$
  - This is known as the birthday attack.

Choosing the length of Hash outputs

- Because of the birthday attack, the length of hash outputs in general should double the key length of block ciphers
  - SHA-256, SHA-384, SHA-512 to match the new key lengths (128,192,256) in AES

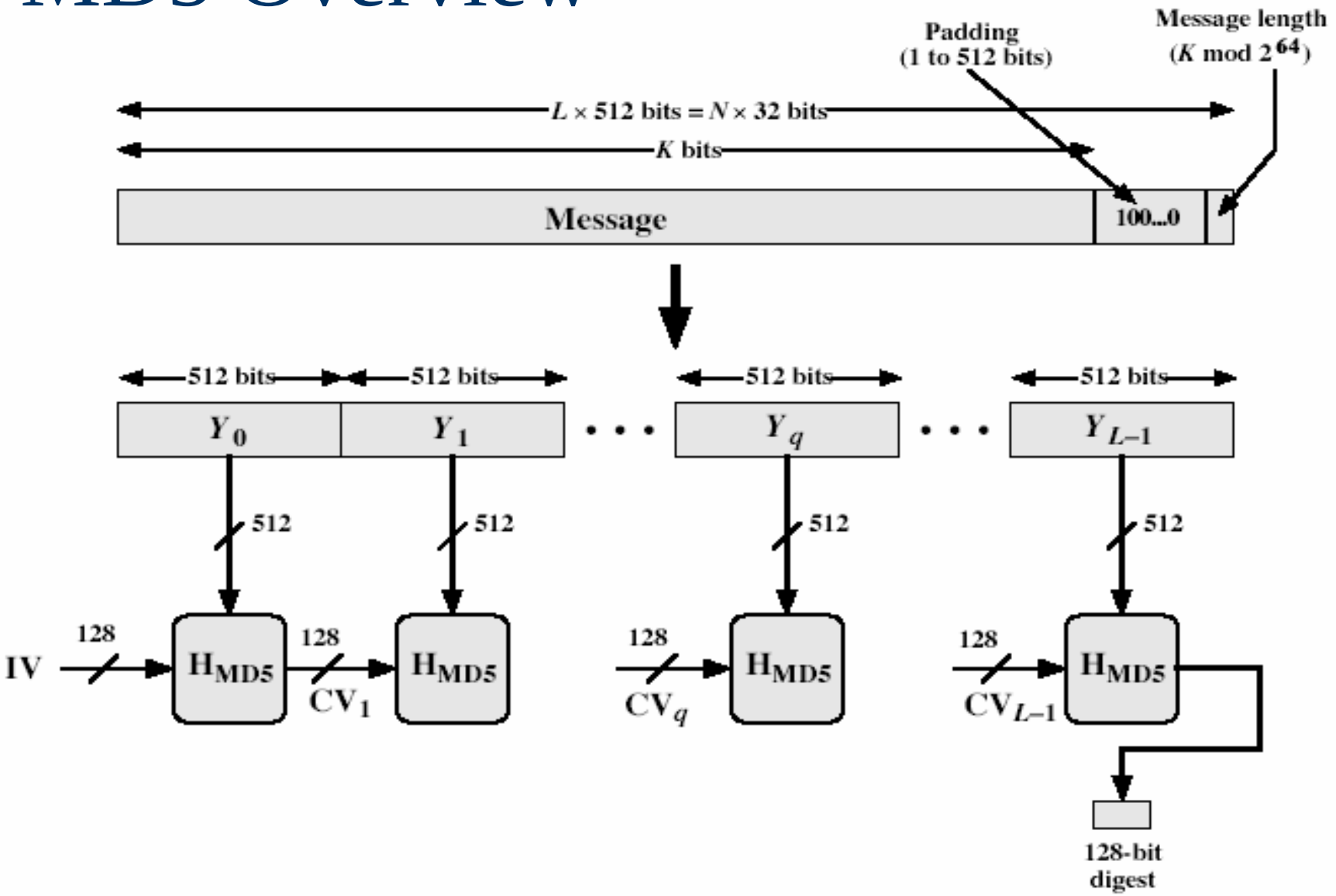
Iterative Construction of Hash Functions

- A hash function needs to map a message of an arbitrary length to a m-bit output
  - h: $\{0,1\}^* \rightarrow \{0,1\}^m$
- The iterative construction
  - use a compression function that takes a fixed-length input string and output a shorter string
    - f:$\{0,1\}^{m+t} \rightarrow \{0,1\}^m$
  - a message is divided into fixed length blocks and processed block by block

MD2, MD4 and MD5

- Family of cryptographic hash functions designed by Ron Rivest
- MD2: produces a 128-bit hash value, perceived as slower and less secure than MD4 and MD5
- MD4: produces a 128-bit hash of the message, using bit operations on 32-bit operands for fast implementation, specified as Internet standard RFC1320
- MD5: produces a 128-bit output, specified as Internet standard in RFC1321; till relatively recently was widely used.

MD5 Overview



MD5 Details

- The message is padded (1 followed by 0s) such that its length ≡ 448 mod 512
- Append a 64-bit (treated as unsigned int) representing the length of the message (before padding)
- Initialize the 4-word (128-bit) buffer (A,B,C,D)

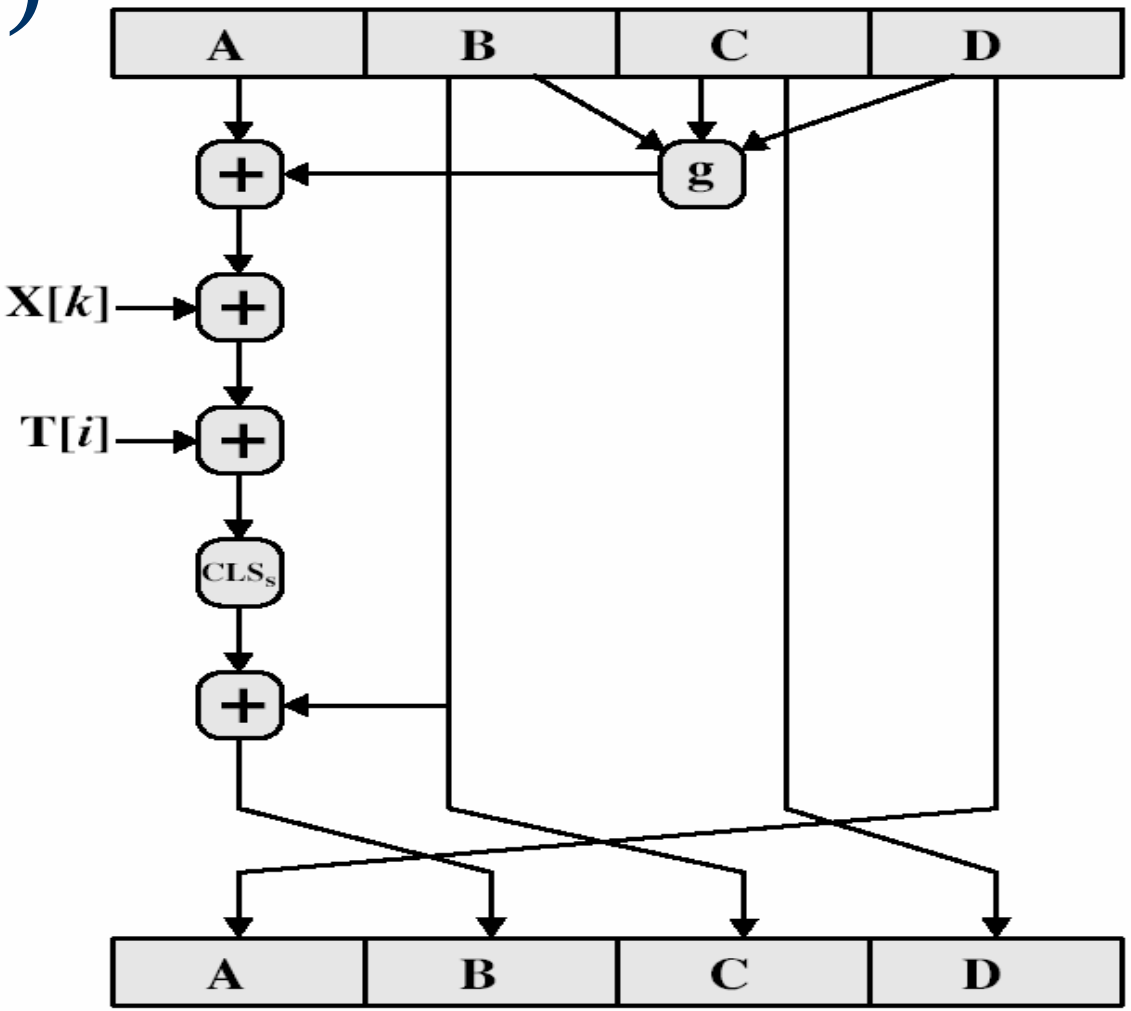
  A = 01 23 45 67

  B = 89 AB CD EF

  C = FE DC BA 98

  D = 76 54 32 10
- The message is processed in 16-word (512-bit) chunks, using 4 rounds of 16 steps each

MD5 Compression Function (Single Step)



MD5 Compression Function

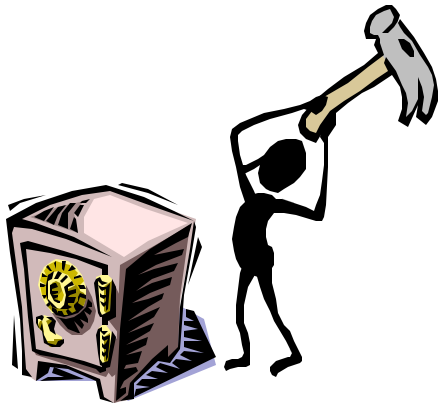
- Each round has 16 steps of the form:

  ```
  a = b+((a+g(b,c,d)+X[k]+T[i])<<<s)
  ```

- a,b,c,d are the 4 words of the buffer, but used in varying permutations
- 4 rounds, each round has 16 steps
- g(b,c,d) is a different nonlinear function in each round (F,G,H,I);
  - Example: round 1 g(b,c,d)=(b$\wedge$c)$\vee$(neg(b) $\wedge$d)
- T[i] is a constant value derived from sin function
- X[k] derived from a 512-block of the message

MD5 Cryptanalysis

- Known attacks:
  - Berson (1992): for a single-round MD5, he used differential cryptanalysis to find two messages producing the same hash. Attack does not work for 4-round MD5.
  - Boer & Bosselaers(1993): found a pseudo collision (same message, two different IV's)
  - Dobbertin (1996) created collisions on MD5 compression function with a chosen IV
  - Wang, Feng, Lai, Yu (2004) found collisions of MD5
    - works on any IV
    - easy to find multiple collisions



SHA1 (Secure Hash Algorithm)

- SHA was designed by NIST and is the US federal standard for hash functions, specified in FIPS-180 (1993).
- SHA-1, revised version of SHA, specified in FIPS-180-1 (1995) use with Secure Hash Algorithm).
- It produces 160-bit hash values.
- NIST have issued a revision FIPS 180-2 that adds 3 additional hash algorithms: SHA-256, SHA-384, SHA-512, designed for compatibility with increased security provided by AES.

SHA1 Overview

- As in MD5 message is padded such as its length is a multiple of 512 bits
- Initialize a 5-word (160-bit) buffer

  Word A: 67 45 23 01

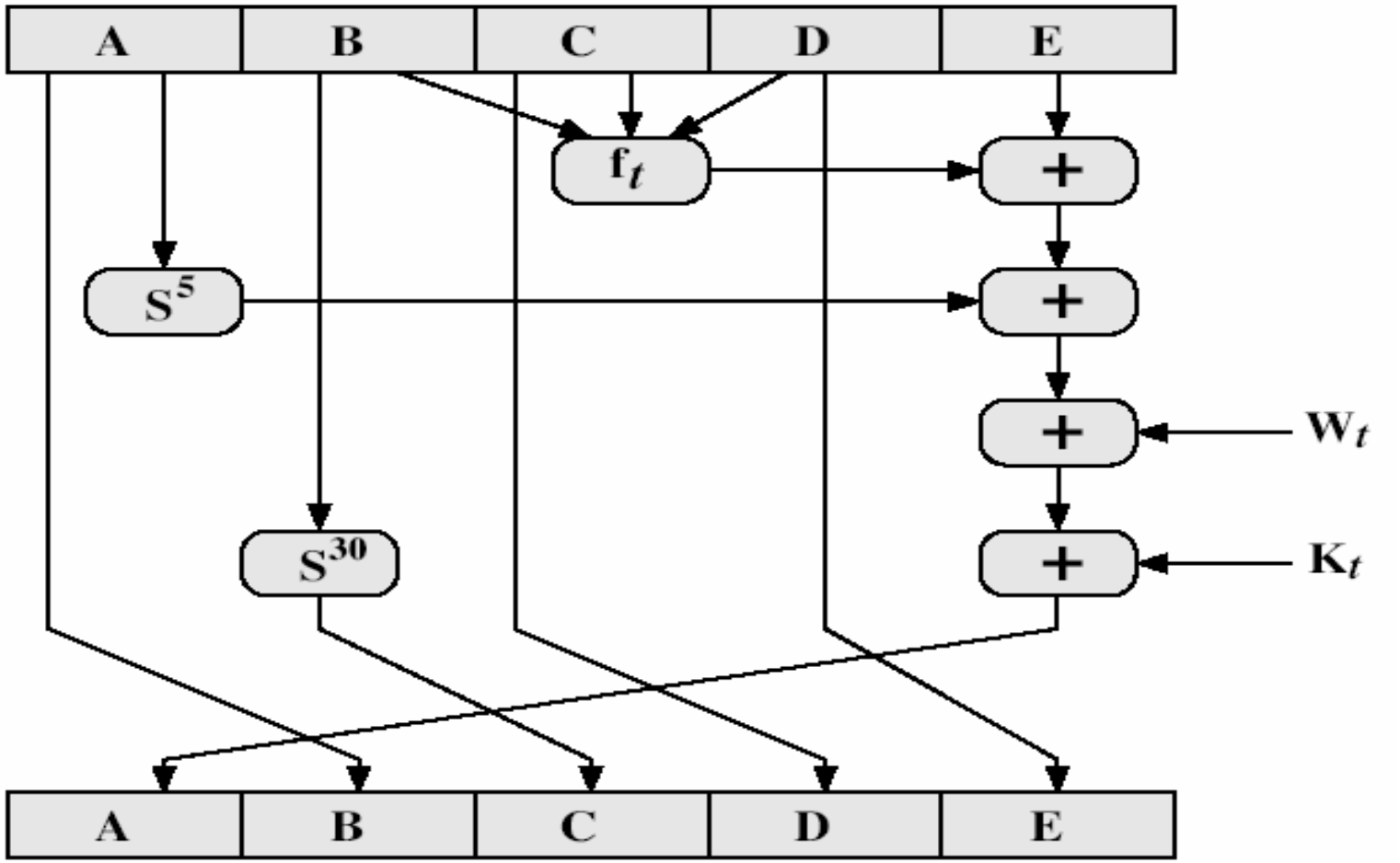
  Word B: EF CD AB 89

  Word C: 98 BA DC FE

  Word D: 10 32 54 76

  Word E: C3 D2 E1 F0
- Message is processed in 16-word (512-bit) chunks:
  - expand 16 words into 80 words by mixing & shifting
  - use 4 rounds of 20 operations on message block and buffer

SHA-1 Compression Function (Single Step)



SHA-1 Compression Function

- Each round consists of 20 steps, updates the buffer as follows:

  (A,B,C,D,E) <-
  (E+f(t,B,C,D)+(A<<5)+W_t+K_t),A,(B<<30),C,D)
- t is the step number
- f(t,B,C,D) is a non-linear function for round
- W_t is derived from the message block
  - $W_t=S^1(W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3})$
- K_t is a constant value derived from the sin function
- S^k is circular left shift by k bits

SHA-1 Cryptanalysis

- SHA1 shuffles and mixes them using rotates & XOR's to form a more complex input that makes finding collisions more difficult.
- Brute force attack is harder (160 vs 128 bits for MD5)
- Wang, Yin, and Yu (2005) found ways to find collisions using no more than 2^{69} hash evaluations

Coming Attractions …

- Integrity/Authenticity
- Hash functions