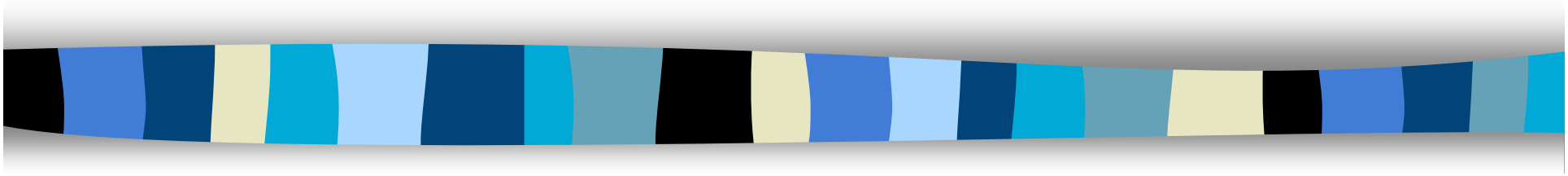
Cryptography CS 555

Lecture 7



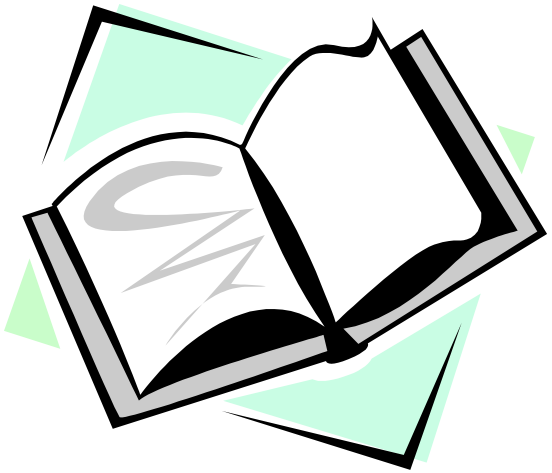
Department of Computer Sciences
Purdue University

Lecture Outline

- Hash functions
- Birthday paradox
- MD5
- SHA1
- RIPMD-160

Recommended Reading

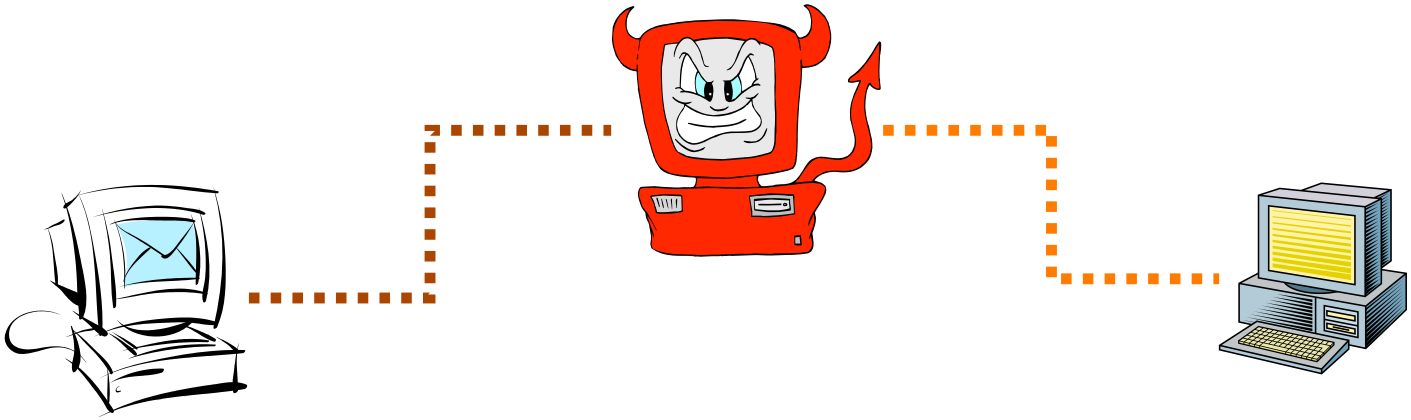
- Stallings, Chapter 12: 12.1, 12.2 and 12.3
- Stinson, Chapter 4: 4.1, 4.2 and 4.3



Security Services

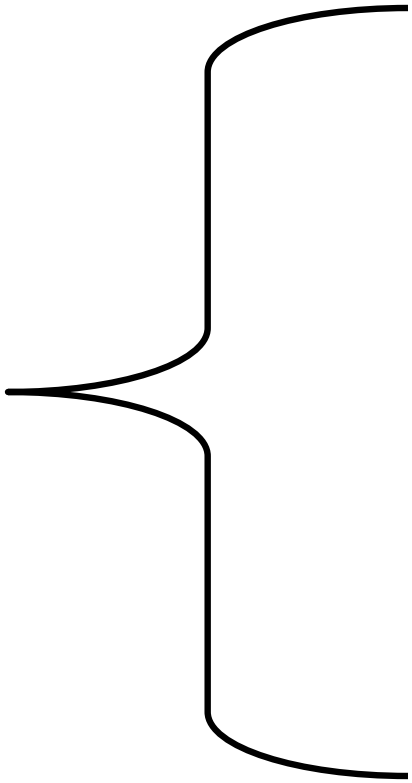
- ✓ Confidentiality
- Integrity
- Authentication
- Non-repudiation
- Access control
- Availability

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.

Begin Math



Functions

Definition

Given two sets, X and Y, a function f : X → Y, a function f from set X to set Y, is a relation which uniquely associates members of set X with members of set Y.

Terminology

X is called domain

Y is called range or codomain.

For y = f(x) where x ∈ X and y ∈ Y, y is called the image of x and x is called the preimage of y.

The number of function from X to Y is $|Y|^{|X|}$

Image of a Function

Definition

Given a function f : X → Y, then Im(f), called image of f, is the set of all y ∈ that have at least one preimage.

Examples

f:R → R, $f(x) = x^2$

Im(f) = [0, inf)

Injections and Surjections

Definition

Given a function f : X → Y, then:

f is a one-to-one function (or injection) if each element in Y is the image of at most one element in X (I.e. $x1 \neq x2 \Rightarrow f(x1) \neq f(x2)$).
Example: f(x) = x

f is a onto function (or surjection) if Im(f) =Y, for each y in Y there exists x in X s.t. f(x) = y.
Example: f(x) = x - 2

Bijection and Inverse of a Function

Definition

f is a bijection if f is one-to-one and onto.

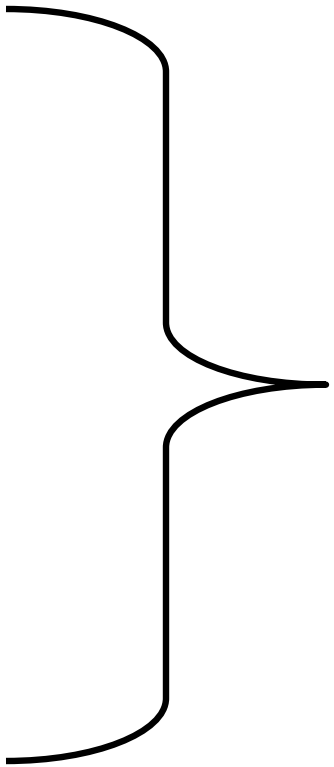
Definition

Given a function $f : X \rightarrow Y$, its inverse $f^{-1}(x)$ is defined by $f(f^{-1}(x)) = f^{-1}(f(x)) = x$

Theorem

The inverse function of a function f exists if and only if f is a bijection.

End Math



Hash Functions

- Map a message of variable length n bits to fingerprint of fixed length m bits, with m < n (output referred as message digest).
- A hash is a many-to-one function, so collisions can happen.
- Two fundamental properties: compression and easy to compute.
- In general, the hash function is public.
- Hash functions can be used to detect changes to message.

Requirements for Hash Functions

Given a function h:X → Y, then we say that h has:

- preimage resistance (one-way):

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

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

  if given x ∈ X it is computationally infeasible to find a value x' ∈ X, x'≠x s.t. h(x') = h(x)

- collision resistance (strong collision resistance):

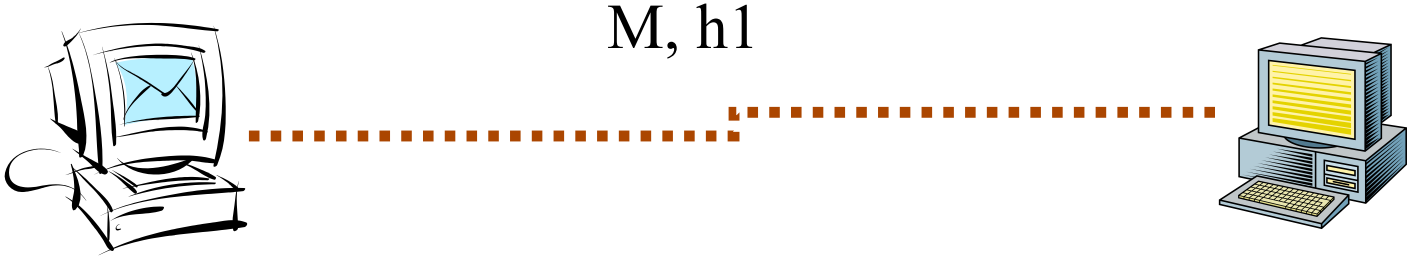
  if it is computationally infeasible to find any two distinct values x',x ∈ X, s.t. h(x') = h(x)

Classification

- MDC (manipulation detection codes) or MIC (message integrity codes)
  - Do not use a key
  - One-Way Hash Functions (OWHFs)
  - Collision Resistant Hash Functions (CRHFs)
- MAC (message authentication codes)
  - Use also a key
  - both authentication and integrity

Data Integrity with Hash Functions

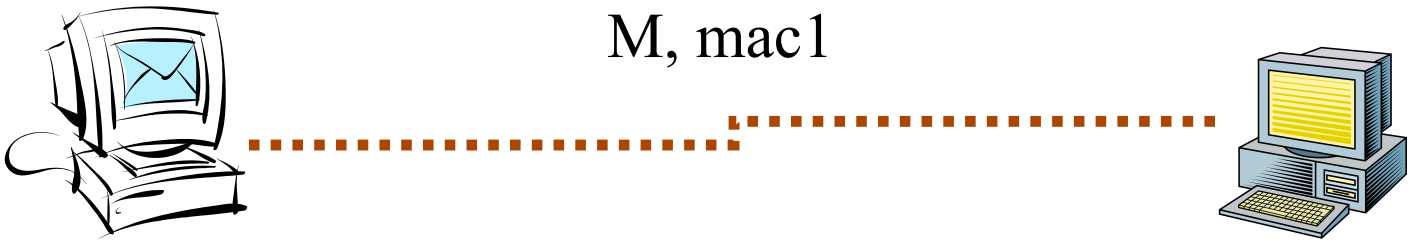
- Hash function is public:
  - Sender computes the value h1 = h(M) and sends it along with the message
  - Receiver computes h2 = h(M)
  - Checks if h1 = h2 ?
  - Yes accept the message, no reject the message



Source Authentication with MACs

- Hash function is public and the key shared between the sender and the receiver is secret
- The output of MAC can not be produced without knowing the secret key
  - Sender computes mac1 = MAC(M, H, K) and sends it along with the message M
  - Receiver computes mac2 = MAC(M, H, K) and checks if mac1 = mac2 ? Yes: accept the message, no: reject

    Because the mac1 could have been generated only by someone that knew the secret key K, this mechanism provides also data source authentication.



Birthday Paradox

- Given a group of people, the minimum number of people such that two will share the same birthday with probability > 0.5 is only 23.
- What does this have to do with cryptography anyway?
- PATIENCE …

General Problem

- Given a random variable that is an integer with uniform distribution between 1 and n and a selection of k instances, k<n of the random variable, what is the probability P(n, k) that there is at least one duplicate?

- Solution:

$$P(n,k) = 1 - \frac{n!}{(n-k)!n^k}$$

Solution (cont.)

$$P(n,k) = 1 - \frac{n!}{(n-k)!\,n^k} = 1 - [(1-\frac{1}{n})\cdot(1-\frac{2}{n})\cdot\ldots\cdot(1-\frac{k-1}{n})]$$

$$1 - x \le e^{-x} \quad \text{implies} \quad P(n,k) = 1 - \frac{n!}{(n-k)!\,n^k} > 1 - e^{-\frac{k\cdot(k-1)}{2n}}$$

$$P(n,k) > 0.5 \quad \text{implies} \quad \frac{1}{2} = 1 - e^{-\frac{k\cdot(k-1)}{2n}}$$

For large k, $(k-1)k \approx k^2$, we obtain

$$k = \sqrt{2\ln 2n} = 1.18\sqrt{n} \approx \sqrt{n}$$

Why 23?

For the birthday problem, n = 365

$$k \approx \sqrt{n} \approx \sqrt{365} \approx 22.54 \approx 23$$

Going Back to Hash Functions…

Hash functions map n bits to m bits, m < n. Given h(x), if h is applied to k random inputs, what is the minimum k such that there exists x' such that h(x') = h(x) with probability > 0.5?

In other words: how many hash computations needs an attacker to try to break the 2-nd pre-image resistance condition?

Answer: $k = 2^{(m-1)}$

Moreover …

Given a hash function h that maps n bits to m bits, m < n. Apply h to k random inputs to produce the set X and apply again h to k additional random inputs to produce the set Y. What is the minimum k such that I found a match between the two sets with probability > 0.5?

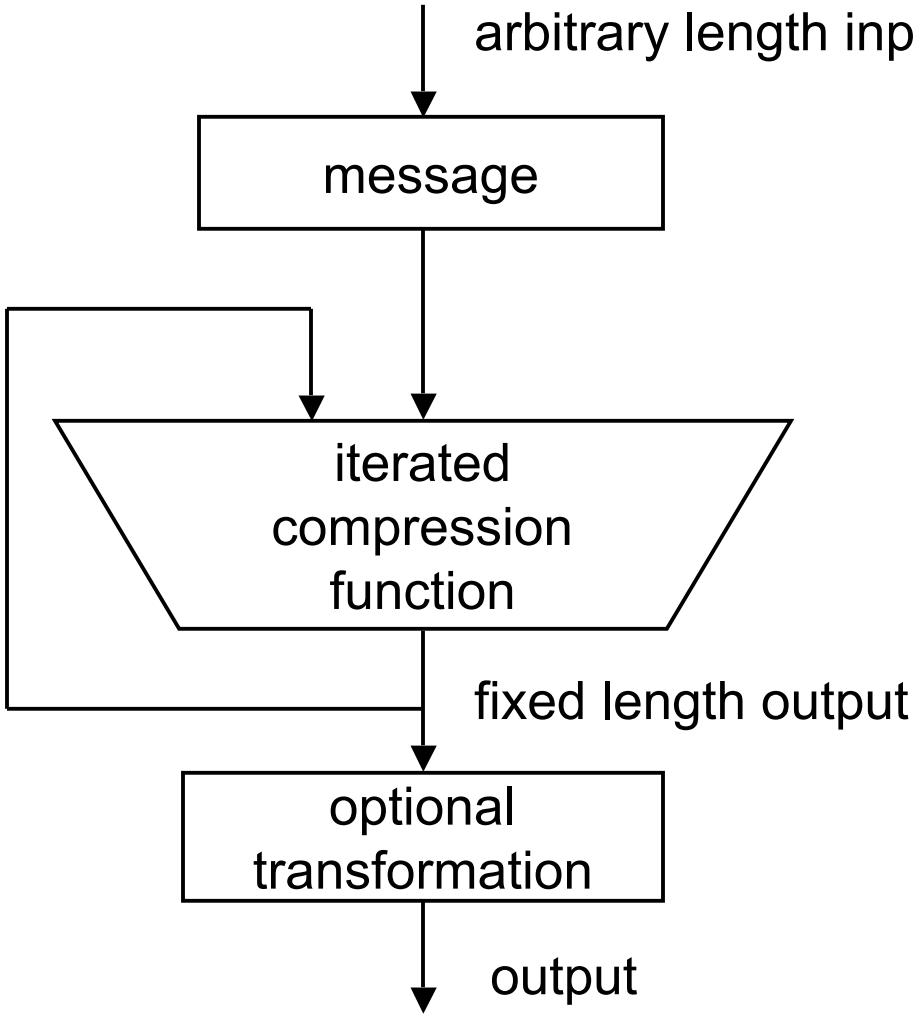
In other words: how much work does an attacker to do to find a collision (break the collision resistance property)?

Answer: $k = 2^{m/2}$

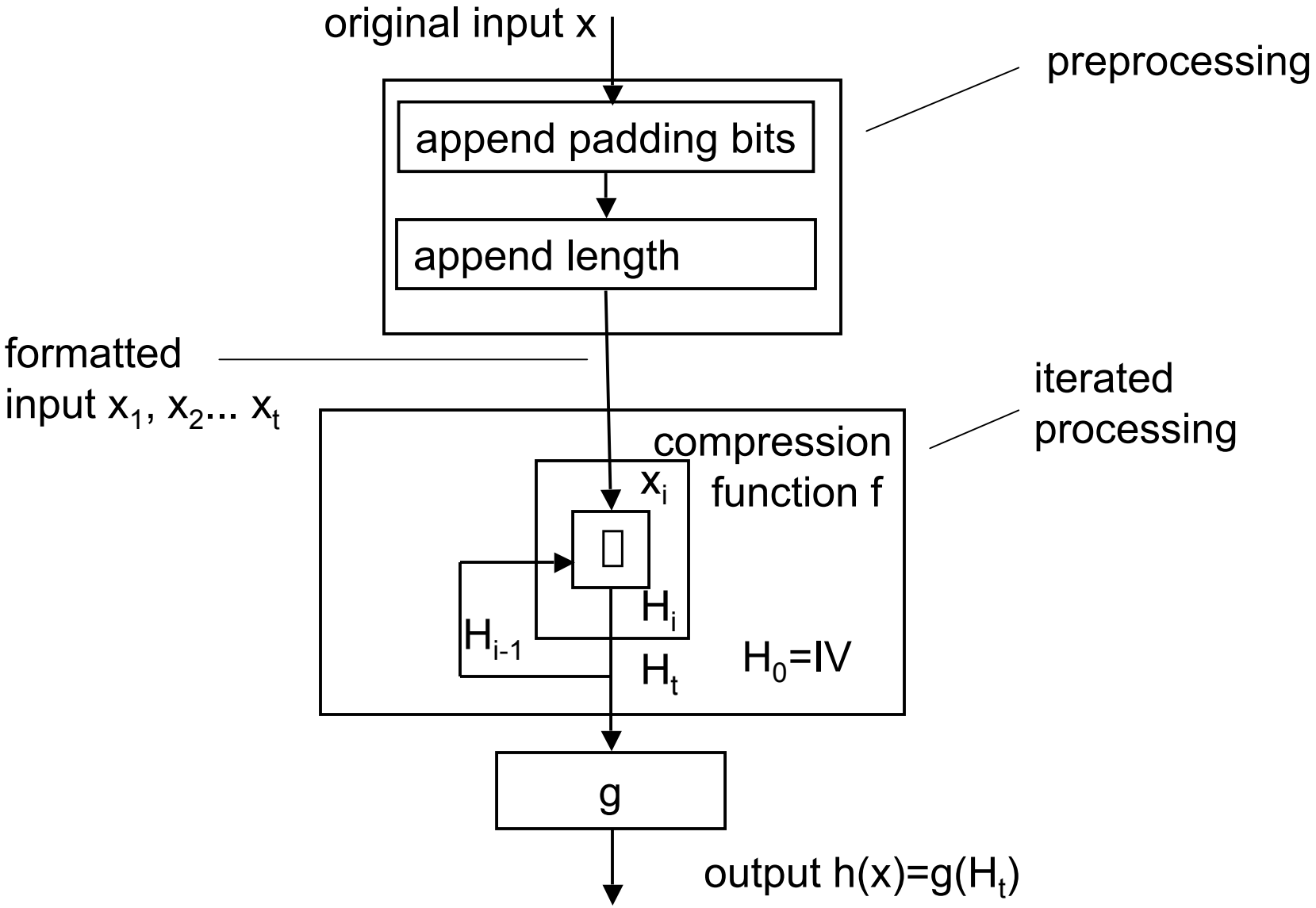
Birthday Attacks

- Attacks runs in $O(2^{m/2})$ and works against all the unkeyed hash function
- Steps:
  - Attacker generates $2^{m/2}$ variations of a valid message all with essentially the same meaning
  - Attacker 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)

Model for Iterated Hash Functions



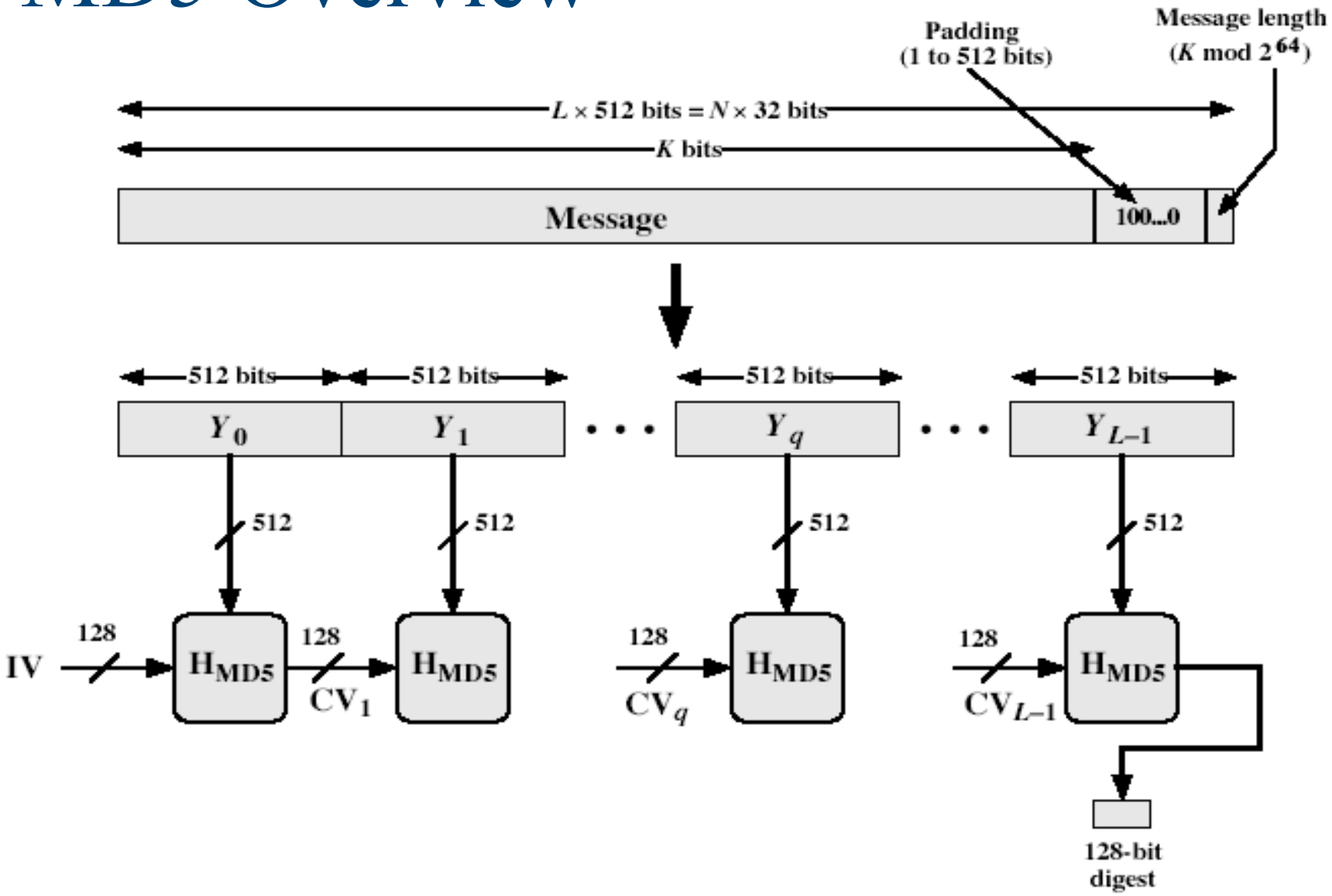
Details


original input x
preprocessing
append padding bits
append length
formatted
input x_1, x_2... x_t
iterated
processing
compression
function f
x_i
H_i
H_{i-1}
H_t
H_0=IV
g
output h(x)=g(H_t)

MD2, MD4 and MD5

- Family of one-way hash functions (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 L ≡ 448 mod 512
- Append a 64-bit (treated as unsigned int) representing the length of the message before padding (actually length mod $2^{64)}$
- Initialize the 4-word (128-bit) buffer (A,B,C,D)

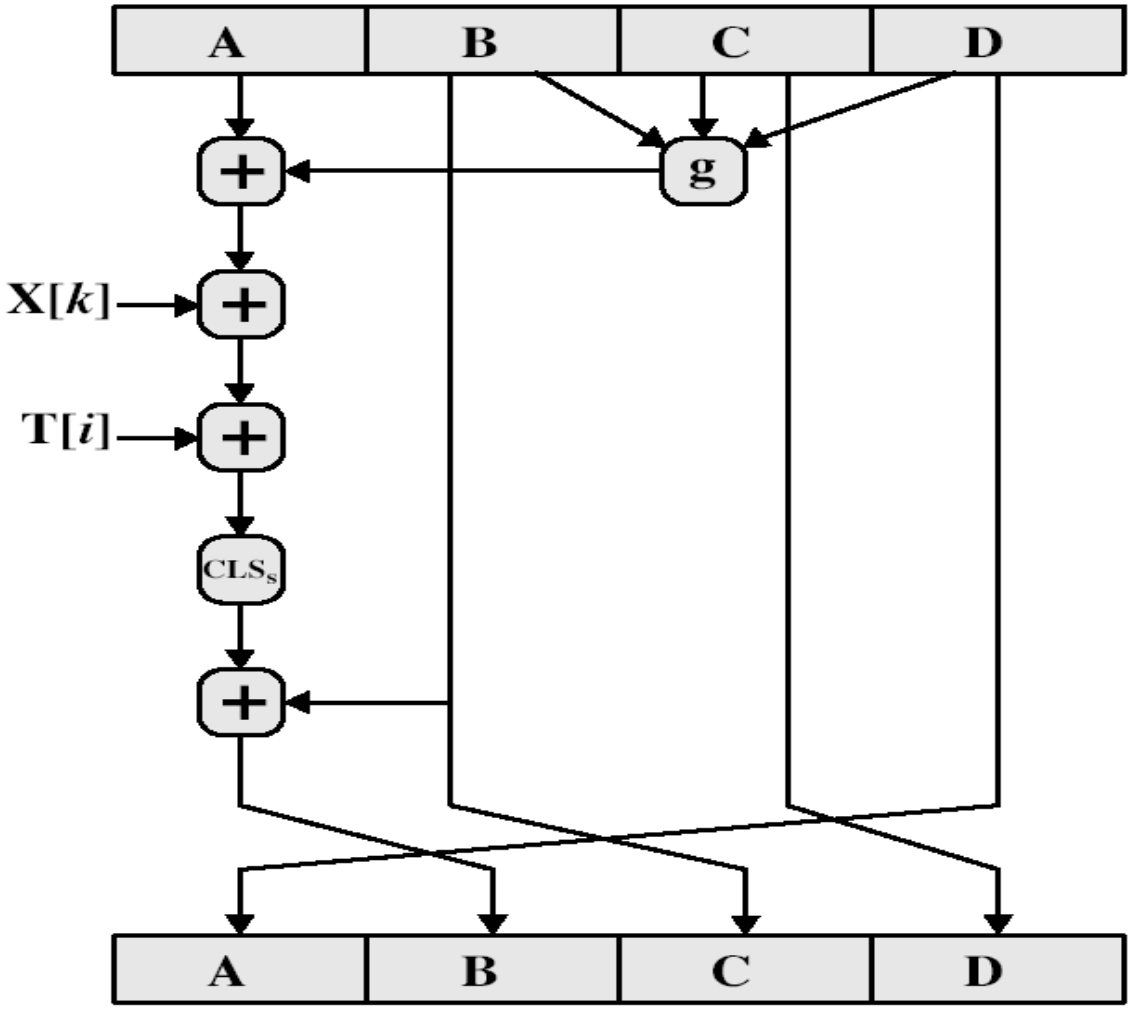
  A = 01 23 45 67

  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 bit operations each

MD5 Compression Function (Single Step



MD5 Compression Function

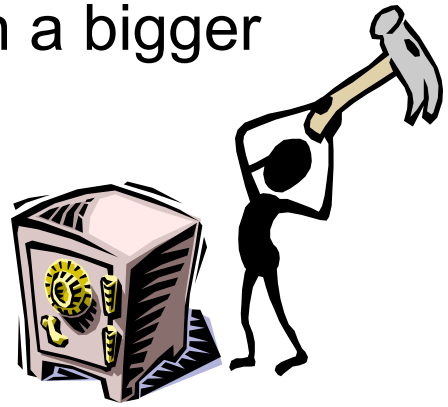
- Each round has 16 steps of the form:

  ```
  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□c) (neg(b) □d)
- T[i] is a constant value derived from sin function
- X[k] derived from a 512-block of the message

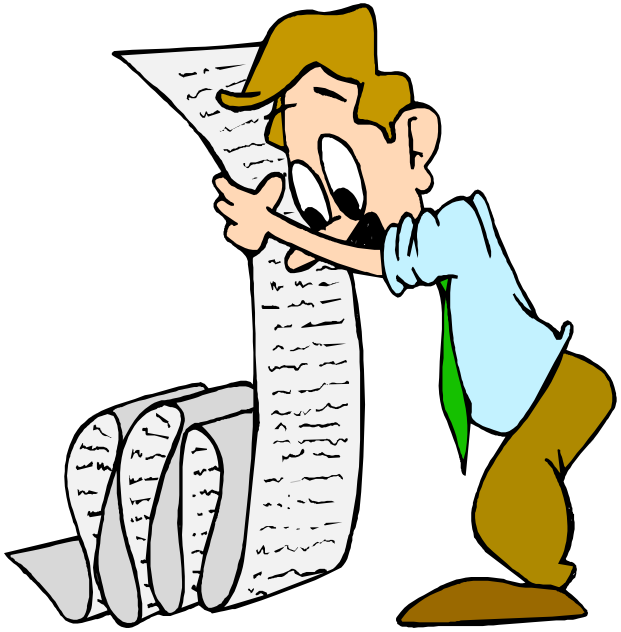
MD5 Cryptanalysis

- 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, unable to extend to full MD5
  - Dobbertin (1996) created collisions on MD5 compression function only on the first block, but initial constants prevent exploit on a bigger message
- Brute force biggest concern:
  - $2^{m/2}$, since m =128 , 2^{64}



Summary

- Hash functions produce a digest of the message, map n bits to m bits, with m < n
- Hash functions requirements are one-way, weak-collision resistance and strong collision resistance
- Brute force attacks, finds a collision in $O(2^{m/2})$



Next Lectures..

- September 18:
  - Digital rights management: Prof. Atallah
- September 23:
  - Hw2 due in class, at the beginning of the lecture.
  - Hw1 will be returned
  - HMAC; Stallings 12.4 Stinson 4.4 and 4.5