

# Cryptographic Hash Functions

Raj Jain

Washington University in Saint Louis

Saint Louis, MO 63130

[Jain@cse.wustl.edu](mailto:Jain@cse.wustl.edu)

Audio/Video recordings of this lecture are available at:

<http://www.cse.wustl.edu/~jain/cse571-17/>

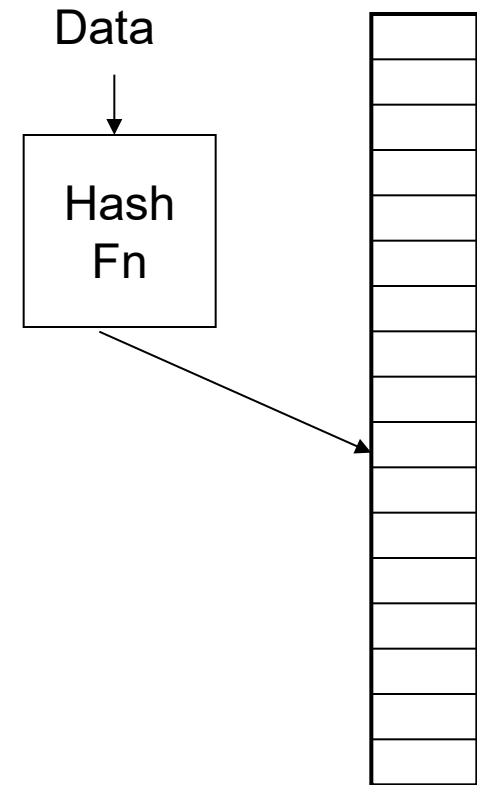


1. Cryptographic Hash Functions
2. Applications of Crypto Hash Functions
3. Birthday Problem
4. Secure Hash Algorithm (SHA)

These slides are based partly on Lawrie Brown's slides supplied with William Stallings's book "Cryptography and Network Security: Principles and Practice," 7<sup>th</sup> Ed, 2017.

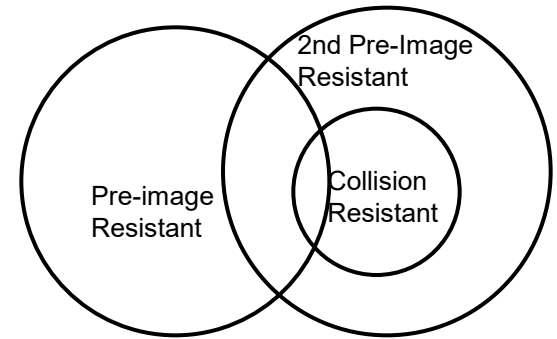
# Hash Function

- ❑ Hash tables used in data searches
  - ❑ The hash function should
    1. Take variable size input
    2. Produce fixed output size (Size of the table)
    3. Be easy to compute
    4. Be pseudorandom so that it distributes uniformly over the table  $\mathcal{P}$
- Minimizes collisions



# Cryptographic Hash Functions

1. Variable Size Input
2. Fixed output size
3. Efficient computation
4. Pseudorandom
5. Pre-image Resistant = one-way  
It is not possible to find any  $M$ , given  $h$ .
6. 2<sup>nd</sup> Pre-image Resistant = Weak Collision Resistant  
It is not possible to find  $y$ , such that  $h(y)=h(x)$
7. Strong Collision Resistant: It is not possible to find any two  $x$  and  $y$ , such that  $h(y)=h(x)$



# Examples of Crypto Hash Functions

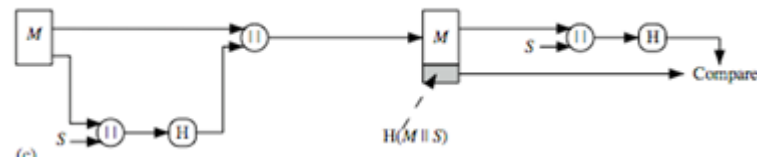
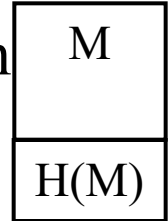
- ❑ MD4 = Message Digest 4 [RFC 1320] - 32b operations
- ❑ MD5 = Message Digest 5 [RFC 1321] - 32b operations
- ❑ SHA = Secure hash algorithm [NIST]
- ❑ SHA-1 = Updated SHA
- ❑ SHA-2 = SHA-224, SHA-256, SHA-384, SHA-512  
SHA-512 uses 64-bit operations

# Applications of Crypto Hash Fn

1. **Message Authentication** = Integrity+Source Repudiation

MD5 has is used to check if a file has been modified.

Use a secret value before hashing so that no one else can modify M and hash



Can encrypt Message, hash, or both for confidentiality

2. **Digital Signatures**: Encrypt hash with private key

3. **Password storage**: Hash of the user's password is compared with that in the storage. Hackers can not get password from storage.



4. **Pseudorandom number generation**: Hash an IV, Hash the hash, ..., repeat

# Birthday Problem

- What is the probability that two people have the same birthday (day and month)

K	Total	Different
2	$365^2$	$365 \times 364$
3	$365^3$	$365 \times 364 \times 363$
		...
k	$365^k$	$365 \times 364 \times 363 \times \dots \times (365 - k + 1)$

$$\begin{aligned} P(\text{No common day}) &= \frac{365 \times 364 \times 363 \times \dots \times (365 - k + 1)}{365^k} \\ &= \frac{365!}{365^k (365 - k)!} \end{aligned}$$

# Birthday Problem (Cont)

- ❑ With 22 people in a room, there is better than 50% chance that two people have a common birthday
- ❑ With 40 people in a room there is almost 90% chance that two people have a common birthday
- ❑ If there  $k$  people, there are  $k(k-1)/2$  pairs

$$P(\text{1 pair having common birthday}) = \frac{k(k-1)}{2 \times 365}$$

$$k \geq \sqrt{365} \Rightarrow P > 0.5$$

- ❑ In general,  $n$  possibilities
  - ▷ ❑  $n$  trials to find a collision

k	P
2	.01
3	.02
4	.03
...	...
19	.41
20	.44
21	.48
22	.51
23	.54
...	...
38	.88
39	.89
40	.90

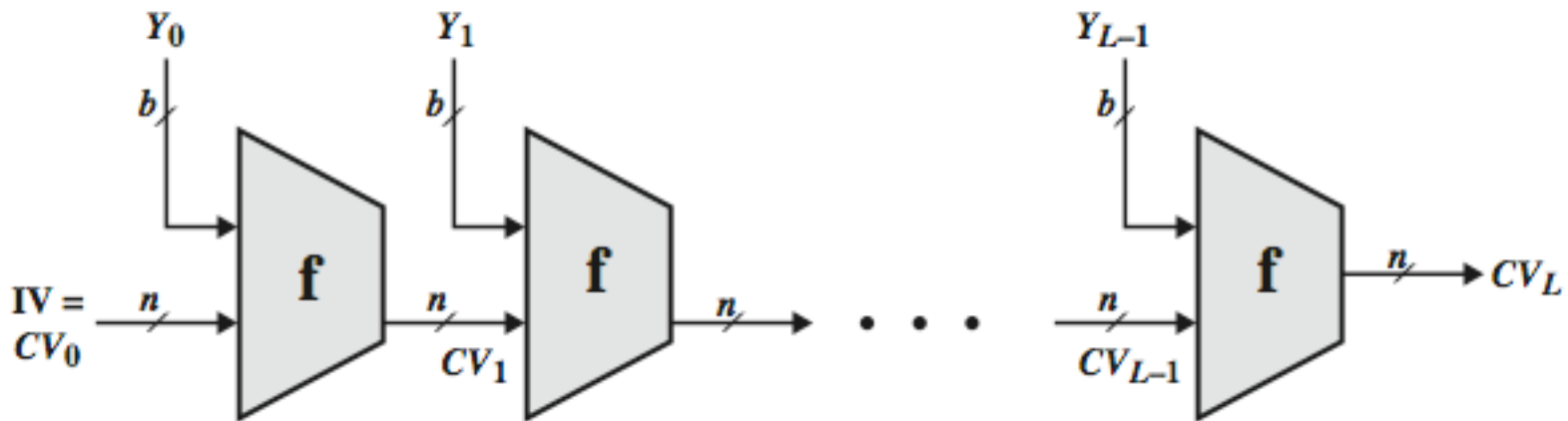


# Probability of Hash Collisions

- ❑ Arbitrary length message  $\mapsto$  Fixed length hash
  - $\mapsto$  Many messages will map to the same hash
- ❑ Given 1000 bit messages  $\mapsto 2^{1000}$  messages
- ❑ 128 bit hash  $\mapsto 2^{128}$  possible hashes
  - $\mapsto 2^{1000}/2^{128} = 2^{872}$  messages/hash value
- ❑ n-bit hash  $\mapsto$  Need avg  $2^{n/2}$  tries to find two messages with same hash
- ❑ 64 bit hash  $\mapsto 2^{32}$  tries (feasible)
- ❑ 128 bit hash  $\mapsto 2^{64}$  tries (not feasible)

# Hash Function Cryptanalysis

- ❑ Hash functions use iterative structure
  - Process message in blocks
- ❑ Compression function **f** takes previous output and next block to produce next output
- ❑ If compression function is collision resistant, the entire structure is collision resistant [Merkle 89]



# Block Ciphers as Hash Functions

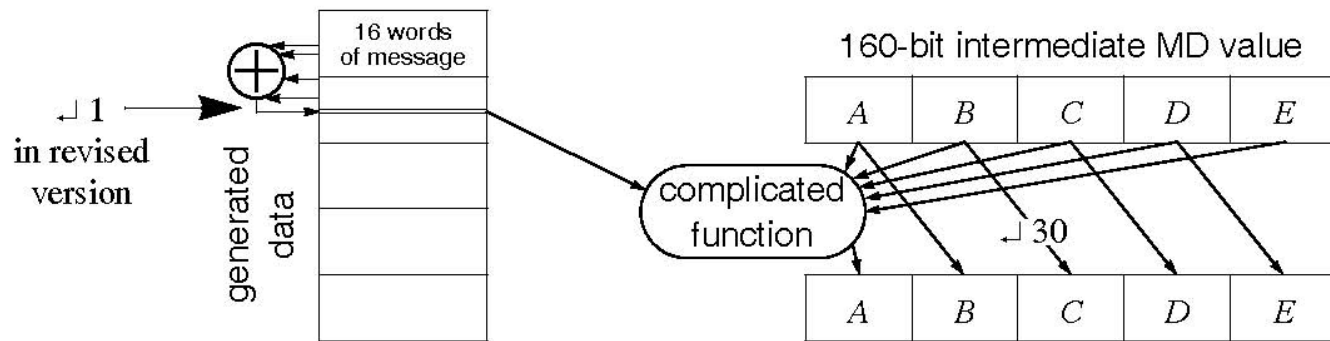
- ❑ Can use block ciphers as hash functions
  - Using  $H_0=0$  and zero-pad of final block
  - Compute:  $H_i = E_{M_i} [H_{i-1}]$
  - And use final block as the hash value
  - Similar to CBC but without a key
- ❑ Resulting hash is too small (64-bit)
  - Both due to direct birthday attack
  - And to “meet-in-the-middle” attack
- ❑ Other variants also susceptible to attack

# Secure Hash Algorithm (SHA)

- ❑ Successor to and similar to MD5 (by Ron Rivest)
- ❑ SHA-0: FIPS PUB 180, 1993. Withdrawn shortly after publ.
- ❑ SHA-1: FIPS PUB 180-1, 1995. 160 bit hash
- ❑ SHA-2: FIPS PUB 180-2, 2002
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512
- ❑ SHA-1 is used in TLS, SSL, PGP, SSH, S/MIME, and IPsec
  - Required by law in US Govt applications
  - Used in Digital Signature Standard
- ❑ Pseudo-codes for SHA algorithms are available.
- ❑ NIST certifies implementations.

# SHA-1 Algorithm

- ❑ 160 bit hash using 512 bit blocks and 32 bit operations
- ❑ Five passes (4 in MD5 and 3 in MD4) of 16 operations each
- ❑ Maximum message size is  $2^{64}$  bit
- ❑ 512 bits are expanded to  $5 \times 512$  bits:
  - $n^{\text{th}}$  word = xor of  $n-3$ ,  $n-8$ ,  $n-14$ , and  $n-16$
- ❑ In SHA-1 these words are rotated left by one bit before xor
- ❑ Total 80 words:  $W_0, \dots, W_{79}$



# SHA-2

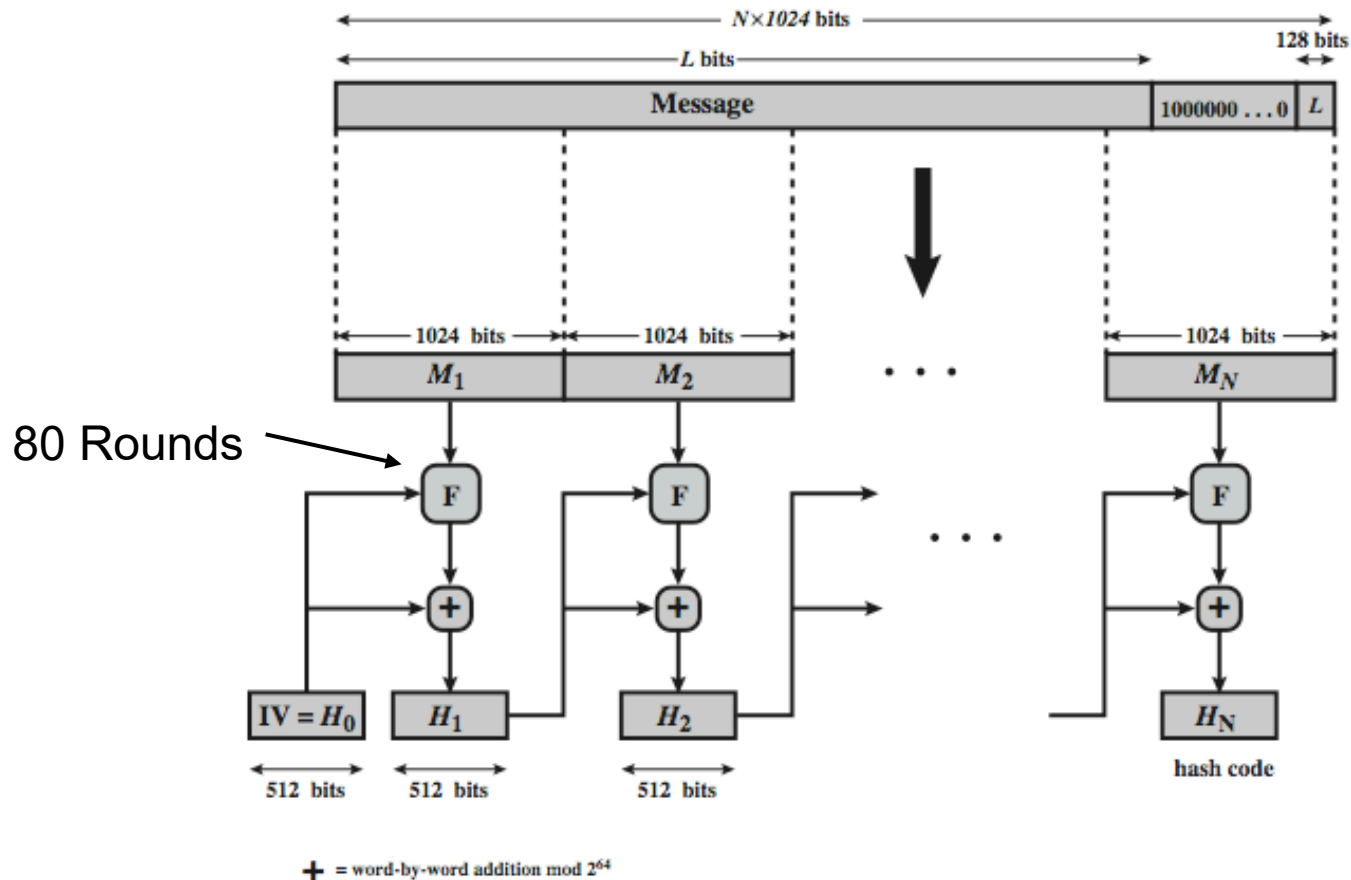
- ❑ SHA-256 uses 32-bit operations
- ❑ SHA-512 uses 64-bit operations
- ❑ Use different shift amounts and additive constants
- ❑ SHA-224 and SHA-384 are simply truncated versions of SHA-256 and SHA-512 using different initial values.
- ❑ SHA-224 matches the key length of two-key triple-DES

Algorithm	Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collision
SHA-0	160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rotl	Yes
SHA-1	160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rotl	$2^{63}$ attack
SHA-256/224	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr,rotr	None yet
SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+,and,or,xor,shr,rotr	None yet

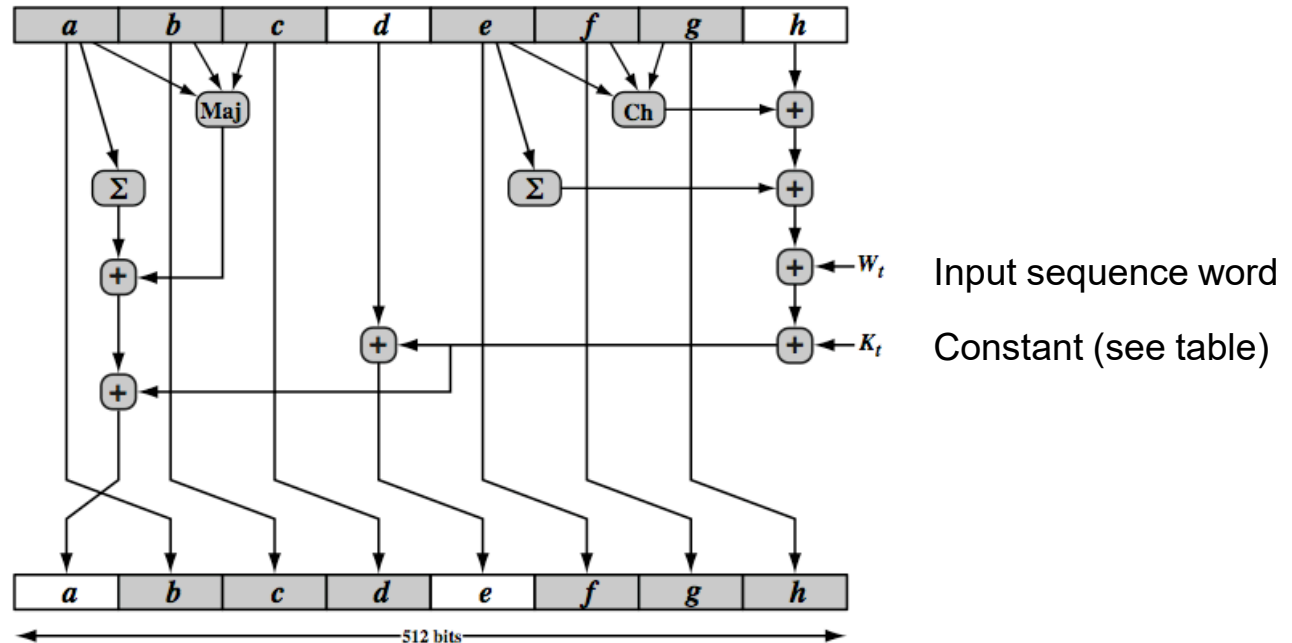
[Source: Wikipedia]

# SHA-512 Overview

- 1. Append padding bits
- 2. Append length



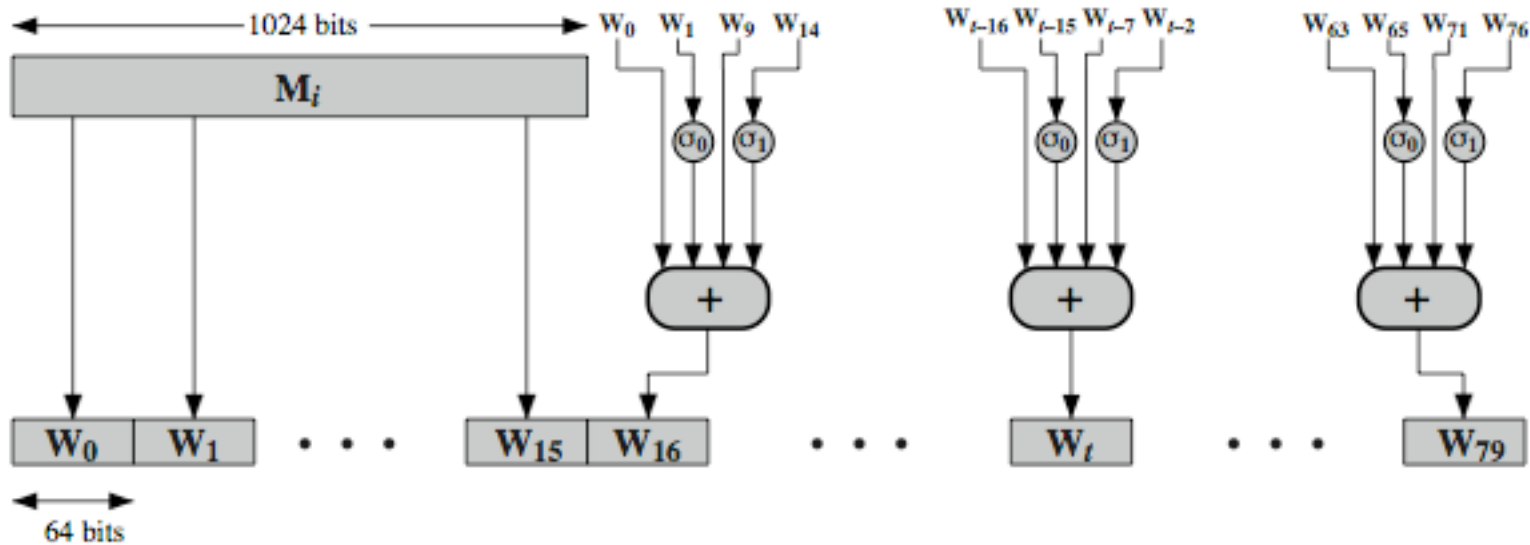
# SHA-512 Round Function



- ❑ Conditional fn  $Ch(e, f, g)$ : if  $e$  then  $f$  else  $g$   
 $= (e \text{ AND } f) \square (\text{Not } e \text{ and } g)$
- ❑ Majority Fn  $Maj(a, b, c)$ : True if 2 of 3 args are true  
 $= (a \text{ AND } b) \square (a \text{ AND } c) \square (b \text{ AND } c)$



# 80-Word Input Sequence



- ❑  $W_t = S_1(W_{t-2}) + W_{t-7} + S_0(W_{t-15}) + W_{t-16}$
- ❑  $S_0(x) = \text{ROTR}^1(x) + \text{ROTR}^8(x) + \text{SHR}^7(x)$
- ❑  $S_1(x) = \text{ROTR}^{19}(x) + \text{ROTR}^{61}(x) + \text{SHR}^6(x)$
- ❑  $\text{ROTR}^n(x)$  = rotate right by  $n$  bits
- ❑  $\text{SHR}^n(x)$  = Left shift  $n$  bits with padding by 0's on the right
- ❑  $+$  = Addition modulo  $2^{64}$

# SHA-3

- ❑ SHA-2 (esp. SHA-512) seems secure
  - Shares same structure and mathematical operations as predecessors so have concern
- ❑ NIST announced in 2007 a competition for the SHA-3
  - Has had 3 rounds of narrowing down the selections
  - Five algorithms advanced to the third (and final) round in December 2010
  - Final selection announced in 2012

Ref: [http://en.wikipedia.org/wiki/NIST\\_hash\\_function\\_competition](http://en.wikipedia.org/wiki/NIST_hash_function_competition)

# SHA-3 Requirements

- ❑ Replace SHA-2 with SHA-3 in any use
  - So use same hash sizes
- ❑ Preserve the online nature of SHA-2
  - So must process small blocks (512 / 1024 bits)
- ❑ Evaluation criteria
  - Security close to theoretical max for hash sizes
  - Cost in time & memory
  - Characteristics: such as flexibility & simplicity

# Summary



1. Hash functions are used to get a digest of a message  
Must take variable size input, produce fixed size pseudorandom output, be efficient to compute
2. Cryptographic hash functions should be preimage resistant, 2<sup>nd</sup> preimage resistant, and collision resistant
3. Cryptographic hashes are used for message authentication, digital signatures, password storage
4. SHA-1 produces 160 bit output, SHA-224, SHA-256, SHA-384, and SHA-512 produce 224, 256, 384, and 512 bit outputs. All consist of 80 rounds.
5. SHA-3 designed but not yet required.

# Homework 11

- ❑ Compute the following hash function:

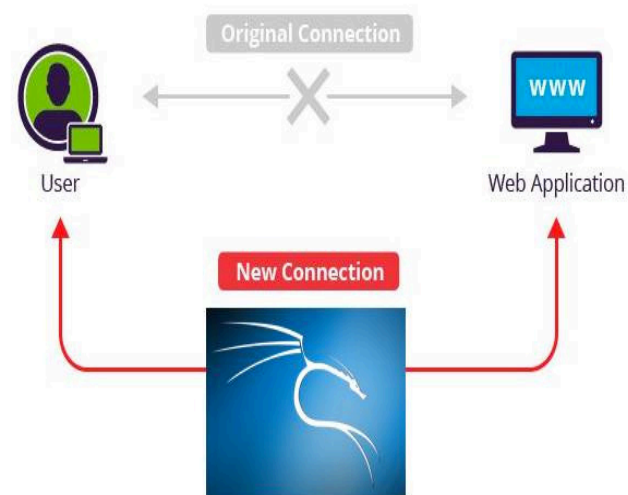
$$h = \left( 7 + \sum_{i=1}^k (m_i)^2 \right) \bmod 251$$

for a 4-byte message  $M = \{m_1, m_2, m_3, m_4\} = \{128, 252, 33, 19\}$   
All are decimal numbers.

- ❑ Check if the hash function is:
  - A. Collision Resistant
  - B. Pre-image resistant
  - C. Second Pre-image Resistant
- ❑ Show counter examples for any property that is not satisfied.

# Lab 11: Man-in-the-Middle Attack

- ❑ In this lab, you will perform man-in-the-middle-attack using Kali Linux
- ❑ Please do it on your network at home and don't do it when connected to WUSTL network
- ❑ You need to have two computers or a computer for Kali and your phone/iPad as a target



# Lab 11 (Cont)

- ❑ Login to Kali Linux from live USB
- ❑ Install bettercap program to Kali
  - *apt-get install bettercap*
- ❑ Explore *bettercap* options using help option
- ❑ Know your target's IP address
  - Use ipconfig/ifconfig on laptops
  - iPhone/iPad: Settings →WiFi → touch (i) symbol on the network name
  - Use an app on other mobiles
- ❑ Run *bettercap* sniffer with the right options to explore devices in your network. These are the devices that you can target. Some devices on your network may not be seen by bettercap. Those devices can not be targeted.
- ❑ Stop bettercap

# Lab 11 (Cont)

- ❑ Find the correct options in bettercap that define the target, interface and parsers
- ❑ Run the following command
  - *bettercap -X1 YOUR\_TARGET\_IP -X2 YOUR\_INTERFACE --proxy -X3 POST*
  - Here, X1 is the option for target, X2 is the option for the interface, and X3 is the option for parsers
- ❑ On your target, access <https://login.libproxy.wustl.edu/login>
  - Note that this is https
- ❑ Login with your username and a random password
- ❑ Capture Kali output. Submit a screen shot of the output along with all the commands you used in Kali



# Acronyms

- ❑ CBC           Cipher Block Chaining
- ❑ DES           Data Encryption Standard
- ❑ FIPS           Federal Information Processing Standard
- ❑ IPsec          IP Security
- ❑ IV             Initial Value
- ❑ MD4           Message Digest 4
- ❑ MD5           Message Digest 5
- ❑ MIME          Multi-Purpose Internet Mail Encapsulation
- ❑ NIST          National Institute of Science and Technology
- ❑ PGP           Pretty Good Privacy
- ❑ PR            Private
- ❑ PUB           Publication
- ❑ RFC           Request for Comment
- ❑ ROTR          Rotate Right
- ❑ SHA           Secure Hash Algorithm
- ❑ SHR           Shift Left

# Acronyms (Cont)

- ❑ SSH            Secure Hash
- ❑ SSL            Secure Socket Layer
- ❑ TLS            Transport Layer Security

# Scan This to Download These Slides



Raj Jain

<http://rajjain.com>

# Related Modules



CSE571S: Network Security (Spring 2017),  
<http://www.cse.wustl.edu/~jain/cse571-17/index.html>

CSE473S: Introduction to Computer Networks (Fall 2016),  
<http://www.cse.wustl.edu/~jain/cse473-16/index.html>



Wireless and Mobile Networking (Spring 2016),  
<http://www.cse.wustl.edu/~jain/cse574-16/index.html>

CSE571S: Network Security (Fall 2014),  
<http://www.cse.wustl.edu/~jain/cse571-14/index.html>



Audio/Video Recordings and Podcasts of  
Professor Raj Jain's Lectures,  
<https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw>