# Cryptography Part III: Hash Functions

## *Cryptographic Hardware for Embedded Systems*

## *ECE 3170*

Fall 2025

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

# Reading Assignment

- Please read chapter 18 of the course textbook by Schneier

# One-Way Hash Function Definition by Schneier

- Given a message *M* of arbitrary length, a hash function *H* generates a fixed-length output *h*
  - $h = H(M)$
- A hash function is one-way if it satisfies the following
  i. Given *M* and *H*, it is easy to compute *h*
  ii. Given *h* and *H*, it is hard to compute *M*
  iii. Given *M* and *H*, it is hard to compute *M'* such that $H(M') = H(M)$
- A one-way hash function can be used to provide a "signature" of *M*
  - Note that property iii above makes it hard for an adversary to change the message but not the one-way hash value
  - Property iii above is also generalized as *collision resistance*

# Use Case Scenario

- Message Authentication
  - Recall that in this course we take “message authentication” to mean identity authentication + message integrity (both!)
  - Financial example: bank deposit
  - An adversary should not be able to, for example, alter a message containing a withdrawal (or any other bank transaction, for that matter) and do so *undetected*
  - In general, hash functions form the basis of message signatures
    - Recall as mentioned in the lecture “Authentication I” that an old-fashioned signature is a known way to provide authentication! We thus need a method of digital signing

# MD5 Overview

- Authored by Ronald Rivest, Professor of Electrical Engineering and Computer Science at MIT
  - Co-author of the asymmetric RSA cryptographic algorithm in 1977
  - Invented MD5 in 1991
- Goal: message integrity
- Keyless

# MD5 Overview (continued)

- Message is divided into blocks
- MD5 block input size: 512 bits
- MD5 block output size: 128 bits
- Keyless
- Message length: up to $2^{64}$ blocks

# MD5 Input

- The overall message (plaintext) must be a multiple of 512 bits
- The last 512-bit block may only contain 448 bits or less of the plaintext
  - Note that 448 = 512-64
  - The last 64 bits represents the message length
  - If less than 448 bits of plaintext are available for the last block, pad with a one followed by as many zeros as are necessary

# MD5 Has Four Rounds

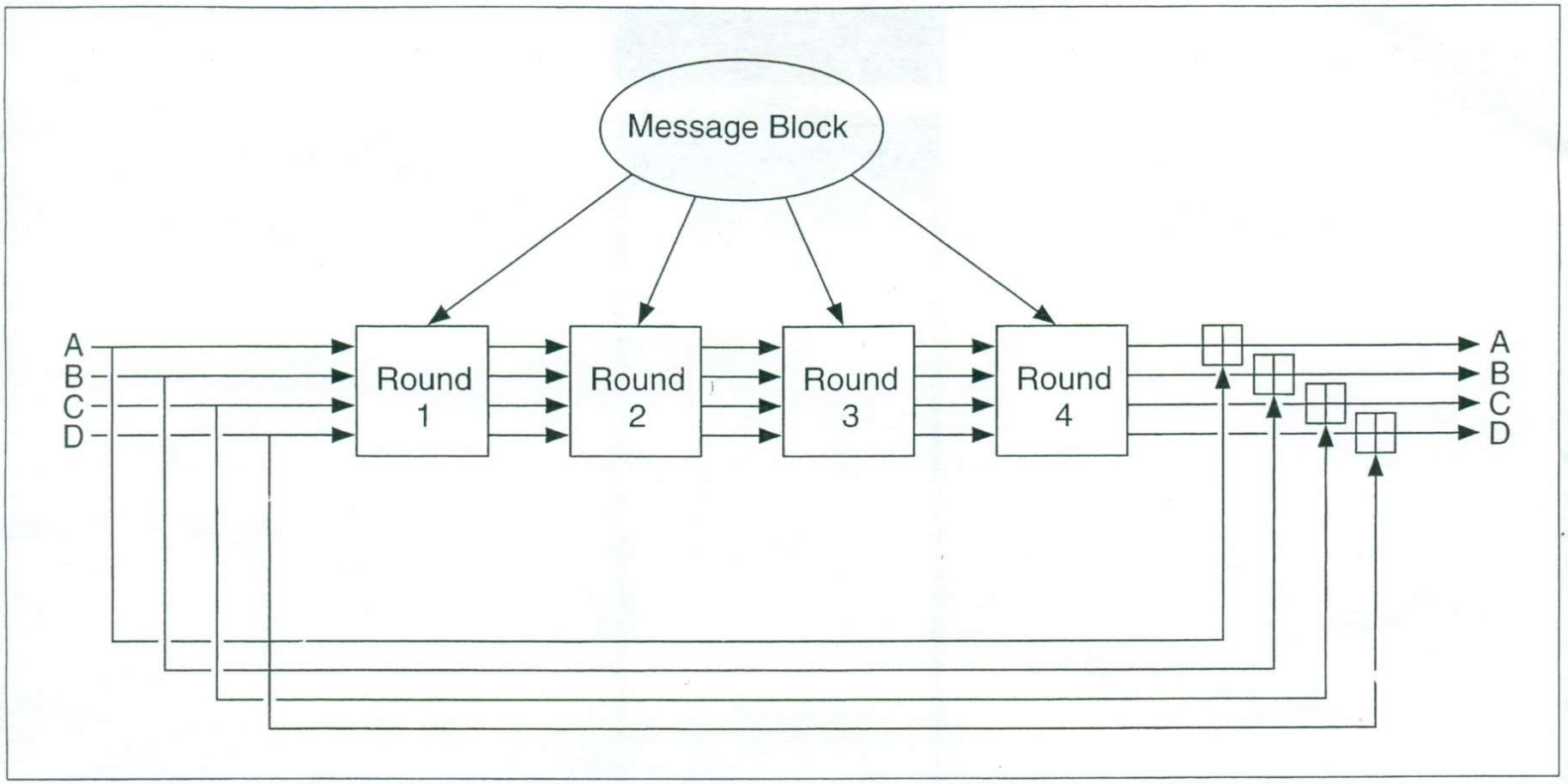


*Figure 18.5 MD5 main loop.*

# A, B, C and D

- Four 32-bit variables are used in MD5
  - Referred to in the literature as *chaining variables*
- Initial values
  - A = 0x01234567
  - B = 0x89abcdef
  - C = 0xfedcba98
  - D = 0x76543210

# MD5 Operation

- Initial values
  - a = A = 0x01234567
  - b = B = 0x89abcdef
  - c = C = 0xfedcba98
  - d = D = 0x76543210
- Each MD5 is a series of MD5 operations

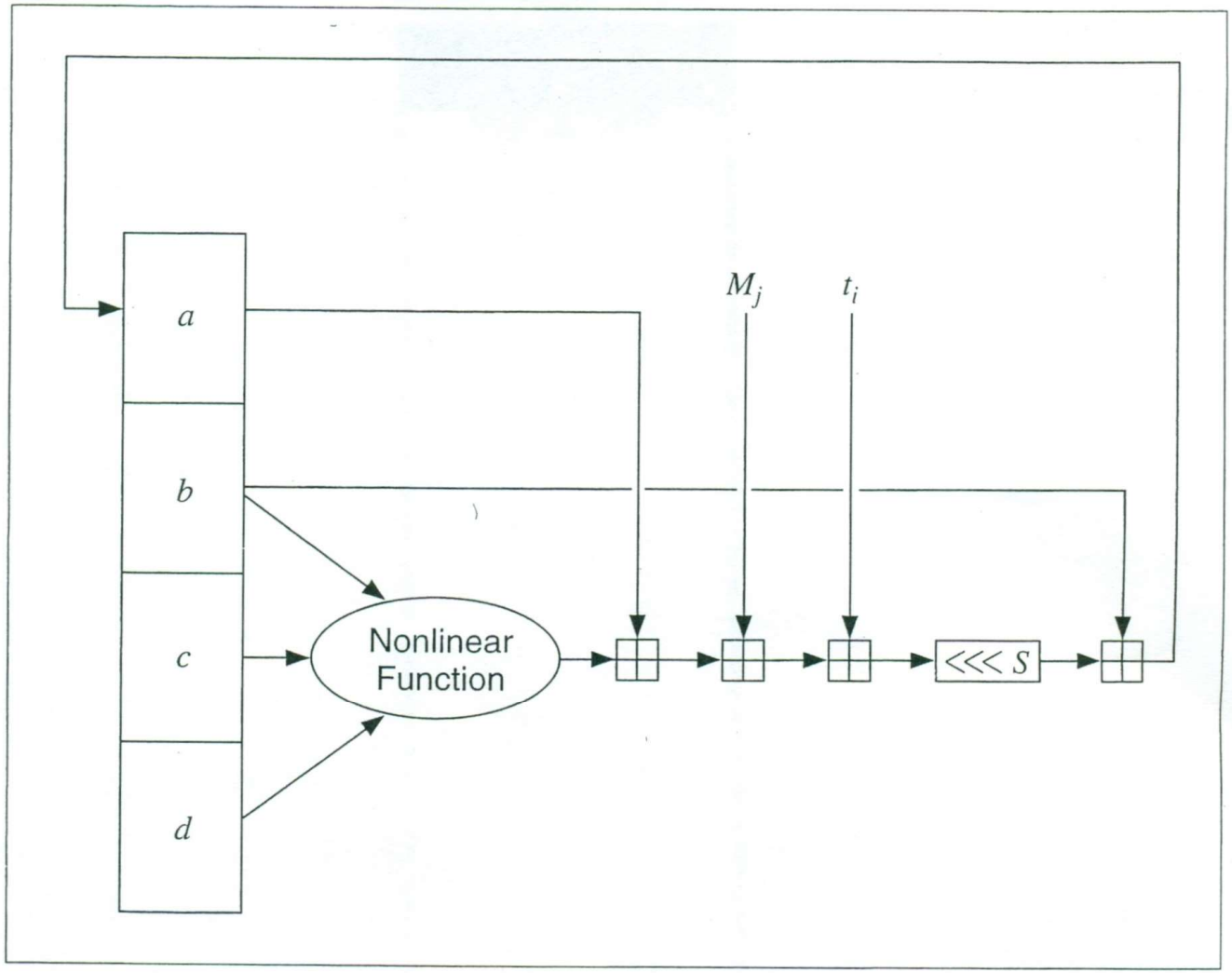


*Figure 18.6 One MD5 operation.*

# MD5 First Round

- FF(a,b,c,d,$M_j$,s,$t_i$) is defined as follows
  - a = b + ((a + F(b,c,d) + $M_j$ + $t_i$) <<< s)
  - where the following definitions hold
    - F(X,Y,Z) = (X OR Y) AND ((notX) OR Z)
    - $M_j$ represents the $j^{th}$ sub-block
      - 512 bit block divided into 16 sub-blocks each with 32 bits
    - For the 64 steps – 16 per round – $t_i$ is the integer part of $2^{32}$*abs(sin(*i*)) where *i* is in radians
      - Note that each $t_i$ is a constant
    - <<< s represents a left circular shift of s bits

Round 1:

FF (*a*, *b*, *c*, *d*, $M_0$, 7, 0xd76aa478)
FF (*d*, *a*, *b*, *c*, $M_1$, 12, 0xe8c7b756)
FF (*c*, *d*, *a*, *b*, $M_2$, 17, 0x242070db)
FF (*b*, *c*, *d*, *a*, $M_3$, 22, 0xc1bdceee)
FF (*a*, *b*, *c*, *d*, $M_4$, 7, 0xf57c0faf)
FF (*d*, *a*, *b*, *c*, $M_5$, 12, 0x4787c62a)
FF (*c*, *d*, *a*, *b*, $M_6$, 17, 0xa8304613)
FF (*b*, *c*, *d*, *a*, $M_7$, 22, 0xfd469501)
FF (*a*, *b*, *c*, *d*, $M_8$, 7, 0x698098d8)
FF (*d*, *a*, *b*, *c*, $M_9$, 12, 0x8b44f7af)
FF (*c*, *d*, *a*, *b*, $M_{10}$, 17, 0xffff5bb1)
FF (*b*, *c*, *d*, *a*, $M_{11}$, 22, 0x895cd7be)
FF (*a*, *b*, *c*, *d*, $M_{12}$, 7, 0x6b901122)
FF (*d*, *a*, *b*, *c*, $M_{13}$, 12, 0xfd987193)
FF (*c*, *d*, *a*, *b*, $M_{14}$, 17, 0xa679438e)
FF (*b*, *c*, *d*, *a*, $M_{15}$, 22, 0x49b40821)

# MD5 Second, Third and Fourth Rounds

- MD5 second, third and fourth rounds are very similar to the first
- Definitions of GG and G (second round), HH and H (third round), and II and I (fourth round) are provided by Schneier

# MD5 Security Analysis

- Each step adds in the result of the previous step
  - Avalanche effect, i.e., the dependency of the output bits on the input bit spreads faster
- Left circular shifts also increase the avalanche effect
  - Diffusion

# MD5 Hacks

- Believed to be collision resistant for many years…
  - "…in 2004 a team of Chinese cryptanalysts presented a new method finding collisions in MD5…" pg. 249 of Katz & Lindell
  - Flame malware attack discovered in 2012 used MD5 signatures to falsify a certificate claiming that code was from a legitimate company…

# Collision Experiment on Hash Functions

- Note that as defined a hash function *H* maps a larger number of bits *M* into a smaller number of bits *h*
  - Therefore it is impossible to always generate a unique *h*
  - *H* may also be called or referred to as a compression function
- Collision-finding experiment
  - Adversary *A* finds a collision if *A* can find *M* and *M'* such that $H(M') = H(M)$
  - If it is infeasible for *A* to find a collision, we say that *H* is *collision resistant*

# Weaker Notions of Security

- *Target-collision resistance*
  - Given a uniformly random *M*, it is infeasible for an adversary to find *M´* such that *H*(*M'*) = *H*(*M*)
  - Note1: this is also referred to in the literature as *second preimage resistance*
  - Note2: collision resistance (see previous page) implies *target-collision resistance*, i.e., second preimage resistance
- *Preimage resistance*
  - Given a uniformly random *h*, it is infeasible for an adversary to find *M* such that *H*(*M*) = *h*
  - Note that second preimage resistance implies

NOTE: PLEASE SEE LECTURE RECORDING FOR ANY CONTENT WHICH MAY (OR MAY NOT!) HAVE BEEN ADDED TO THE EMPTY PAGES