KF School of Computing and Information Sciences
Florida International University

CNT 4403
Computing and Network Security

Cryptography – Hash Functions

Dr. Kemal Akkaya

E-mail: *kakkaya@fiu.edu*

Hash Functions

- A *hash function H*: $\{0,1\}^* \rightarrow \{0,1\}^n$ maps (condenses) a variable-length bitstring to a bitstring of a *fixed* length.
- Given an input $x \in \{0,1\}^*$, the output $h = H(x)$ is called the ***hash value*** or ***message digest*** of x, and x is called a ***preimage*** of h.
- Applications:
  - Hash long messages for signing
  - Authentication protocols
  - Stream & Block ciphers
  - Message Authentication Codes (MACs)
  - Checksums

Hash Function Properties

- **One-wayness**
  - Given h=H(M) for random M, attacker cannot find M
  - It is computationally infeasible to do so
- **Second Preimage Resistance**
  - Given random M, attacker cannot find M' such that H(M)=H(M')
- **Collision Resistance**
  - Attacker cannot find M, M' such that H(M)=H(M')

Popular Hash Algorithms

- **MD5 (Rivest)**
  - 128-bit output; not secure anymore (collision attacks)
- **SHA-1 (NIST-NSA)**
  - US gov std; 160-bit output
  - Not secure anymore; phase out by 2030
  - SHA-2 (256, 512 bits) secure
- **RIPEMD-160**
  - Euro. RIPE project;160-bit
- **NIST SHA-3 competition**
  - 51 submissions (2008); 14 semi-finalists (2009); 5 finalists (2010); winner “Keccak” (2011)
  - NIST announced SHA-3 in 2015

MD5

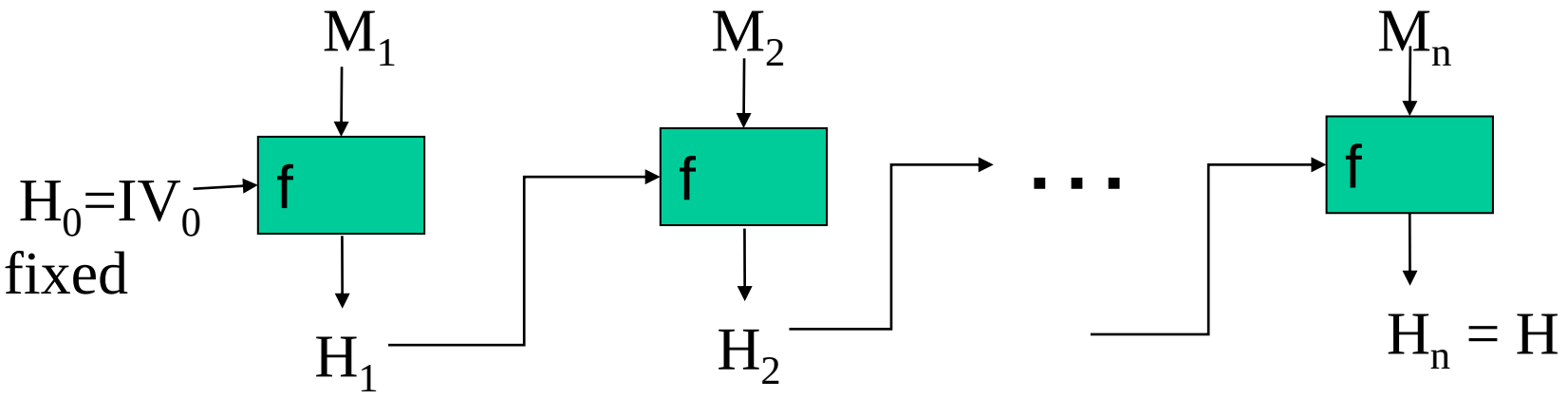
- **Most commonly used present-day message digest (MD) algorithm is the 128 bit MD5 algorithm**
  - Developed by Ron Rivest of the MIT Laboratory for Computer Science and RSA Data Security, Inc. in 1992
- **MD5 is an algorithm which:**
  - Takes an input of any length
  - Processes the input by 512 bits of blocks
  - Outputs a message digest of a fixed length (128-bit, 32 characters)
    - ✓ made up from only hexadecimal characters
  - MD5 uses the same algorithm every time
    - ✓ Hence it will always generate the same message digest for the same string (data).
- **Often used to generate a checksum of whole files**
  - especially apparent in the open source world

MD5 Details

- Processes 512-bit blocks of the message
- There are 4 32-bit registers a, b, c and d. These are initially loaded with IV_0 and carry the hash values from one 512-bit block to the next
- It works in an iterative (chaining) process:

$$H_{i+1} = f(H_i, M_{i+1}) \quad IV_0 = H_0$$

where M_i is a 512 bit block.



M_i 512 bits

H_i 128 bits

More on MD5

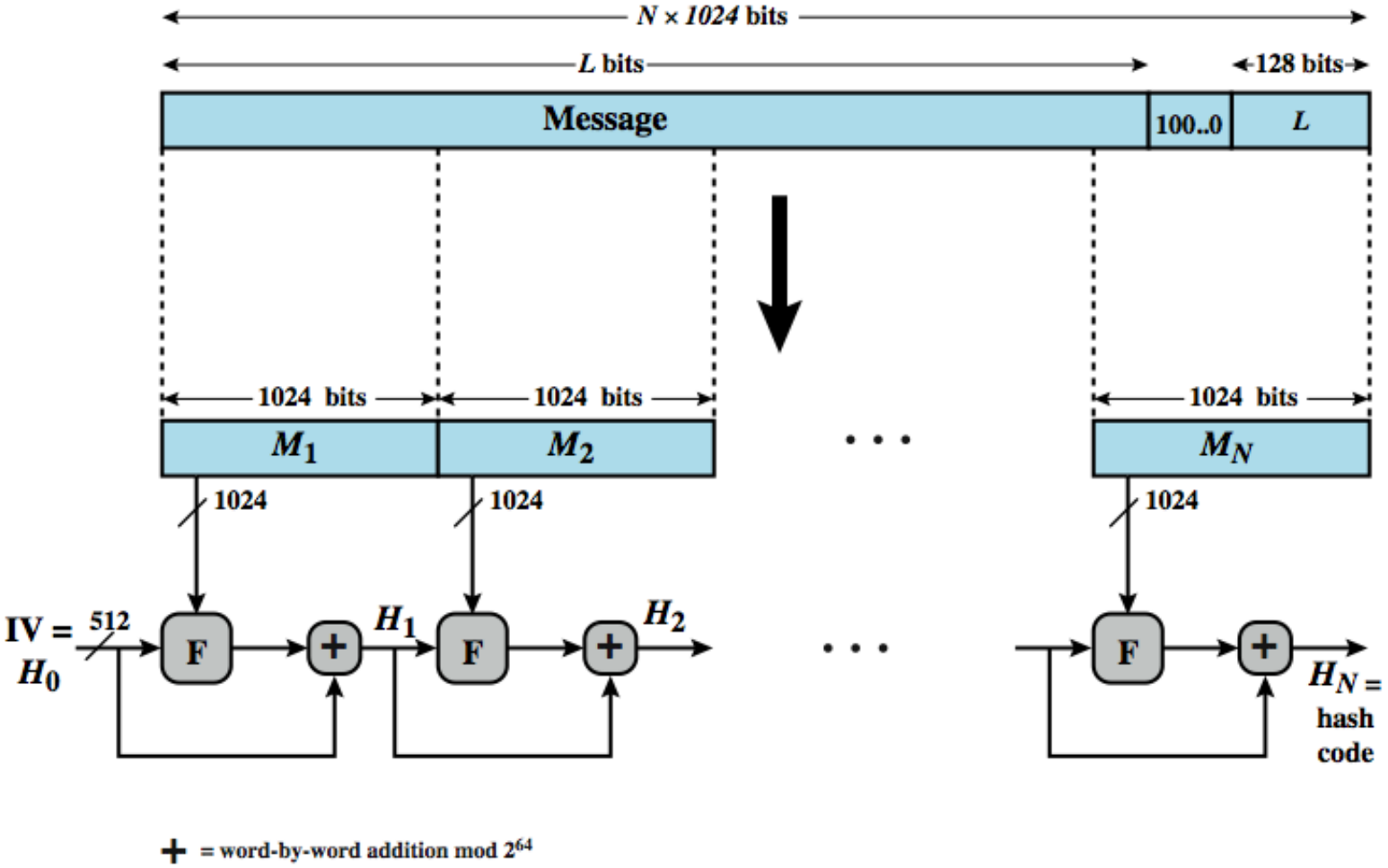
- **MD5 hashes have the advantage of generating completely different looking hashes from seemingly similar inputs. For example:**
  - The MD5 hash of bleh is 4eb20288afaed97e82bde371260db8d8
  - The MD5 hash of Bleh is dcc9f5ac2af04cedb008c04d5f9636b5
  - The MD5 hash of Blehlo is 7ed84db56d34b98757f884bf864b6448

- **There's no known way of getting from the MD5 hash to the originally inputted string**
  - Only known way of getting the original string is by brute force cracking
  - Birthday attack – finding any two inputs with the same hash value
- **MD5 has been compromised**
  - 2007: Two X.509 certificates are found with the same hash.
  - 2008: SSL *root certificates* are forged with MD5.

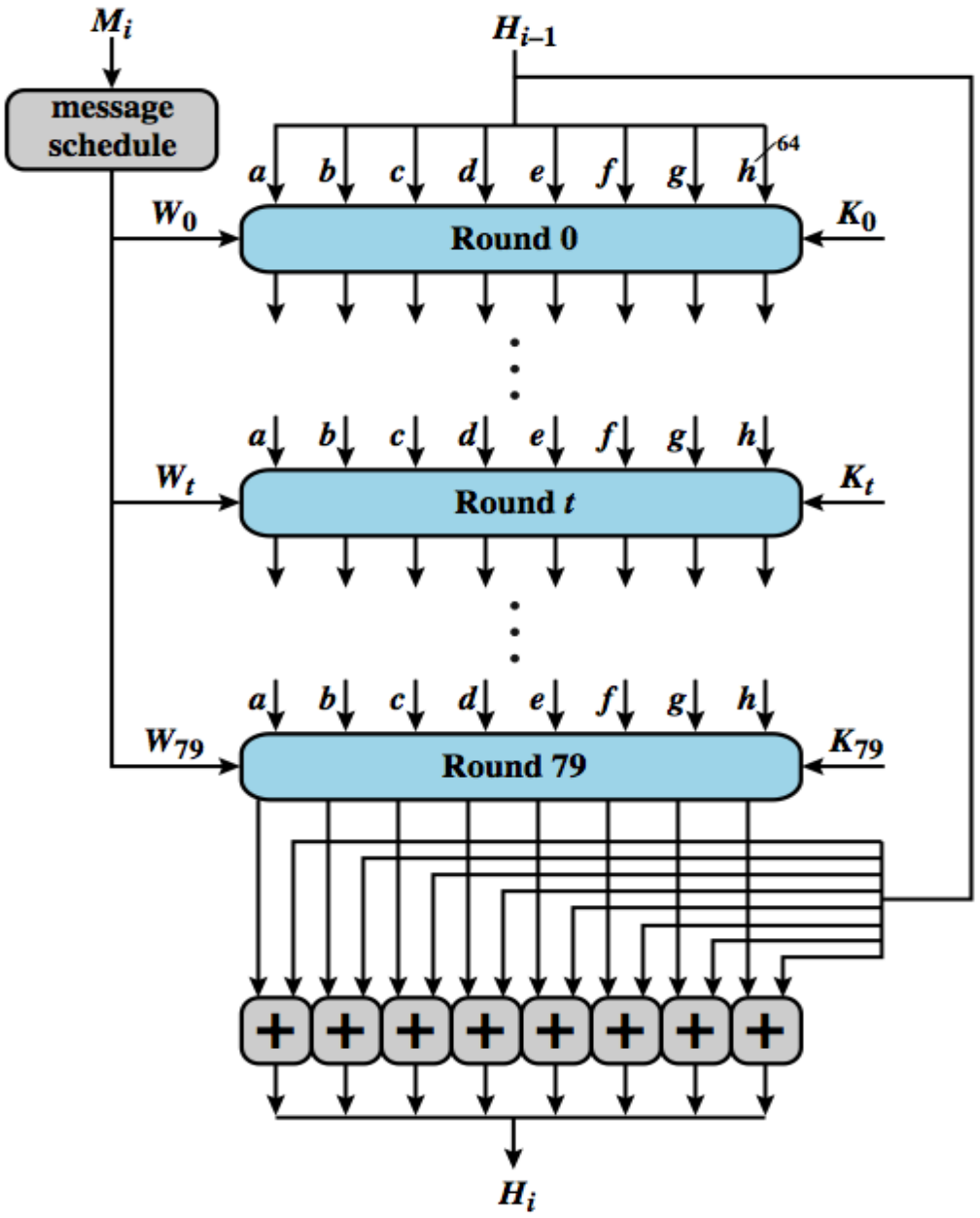
SHA - Secure Hash Functions

- **SHA originally developed by NIST/NSA in 1993**
- **Revised in 1995 as SHA-1**
  - US standard for use with DSA signature scheme
  - Processes 512 bits blocks
  - Produces 160-bit hash values
- **NIST issued revised FIPS 180-2 in 2002**
  - adds 3 additional versions of SHA
    - ✓SHA-256, SHA-384, SHA-512
    - ✓with 256/384/512-bit hash values
  - same basic structure as SHA-1 but greater security
- **NIST phased out SHA-1 use**
- **SHA-3 in use in 2015**

SHA-512 Structure



SHA-512 Function (F)



Use of Hashes

Message Authentication Codes

- **Main application of cryptographic hash functions is *message authentication codes* (MAC)**
- **Provides authentication**
  - Hash functions are generally faster
  - Code for crypto hash functions are widely available
- **MAC = Hash(Key || Message) (original proposal)**
  - The sender shares a secret key K_a with the receiver for message authentication.
  - The sender computes the MAC of a message M as follows: $MAC_{K_a}(M) = H(M \| K_a)$. The message-MAC pair is then transmitted to the receiver.
  - The receiver authenticates M by recalculating the MAC and comparing it with the received MAC.
    - ✓ If the two MACs match, the receiver is assured that the message comes from the alleged sender (**authentication**) and has not been altered during transmission (**integrity**)
  - Cannot provide **non-repudiation (why?)**

Various forms of MACs

- **How to do it best?**
- **prefix: $MAC_K(x) = H(K \| x)$**
  - not secure; extension attack.
- **suffix: $MAC_K(x) = H(x \| K)$**
  - mostly ok; problematic if H is not collision resistant.
- **envelope: $MAC_K(x) = H(K_1 \| x \| K_2)$**
- **HMAC: $MAC_K(x) = H(K_2 \| H(K_1 \| x))$**
  - provably secure; popular in Internet standards.
- **UMAC: (Rogaway et al., 1999)**
  - Extremely fast; adjustable security-speed tradeoff.
  - UMAC-30 is about 10x faster than HMAC-MD5.

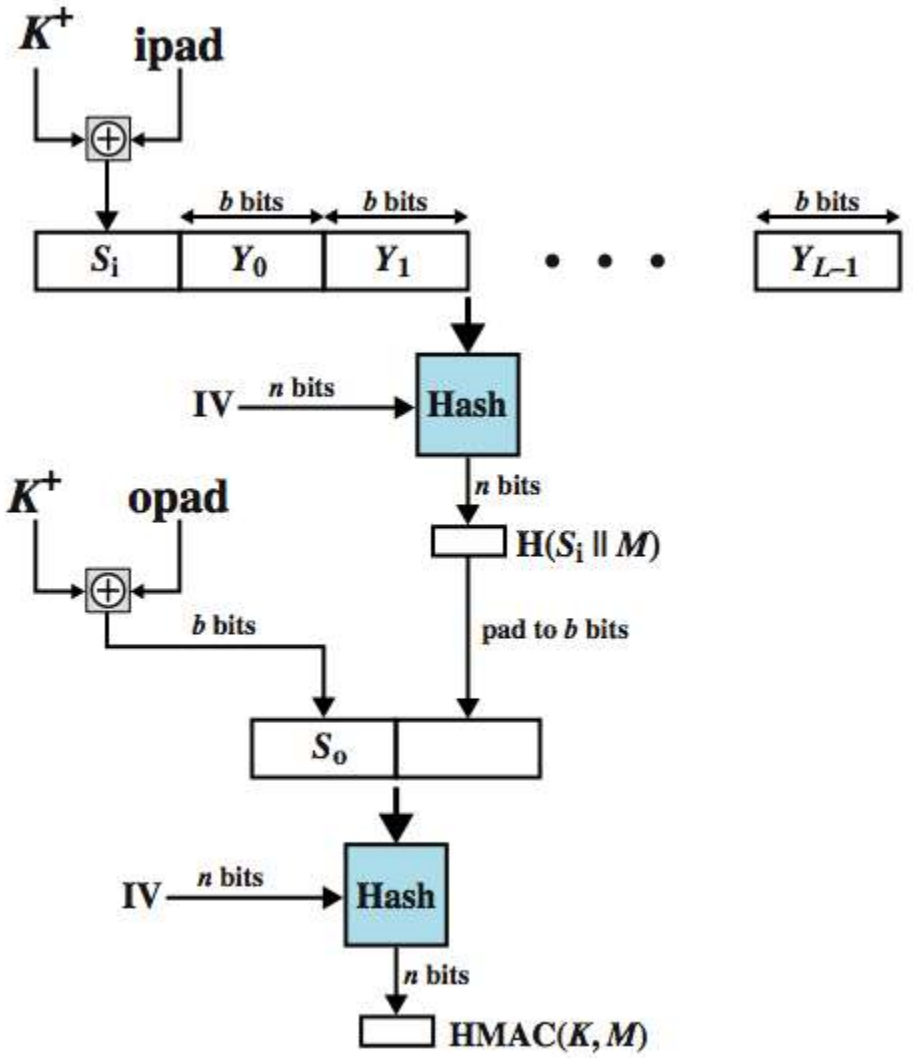
HMAC

- **Specified as Internet standard RFC2104**
  - Used in IPsec, TLS & SET
- **Uses hash function on the message:**

$$\mathrm{HMAC}_K(m) = h\Big((K \oplus \mathrm{opad}) \| h((K \oplus \mathrm{ipad}) \| m)\Big)$$

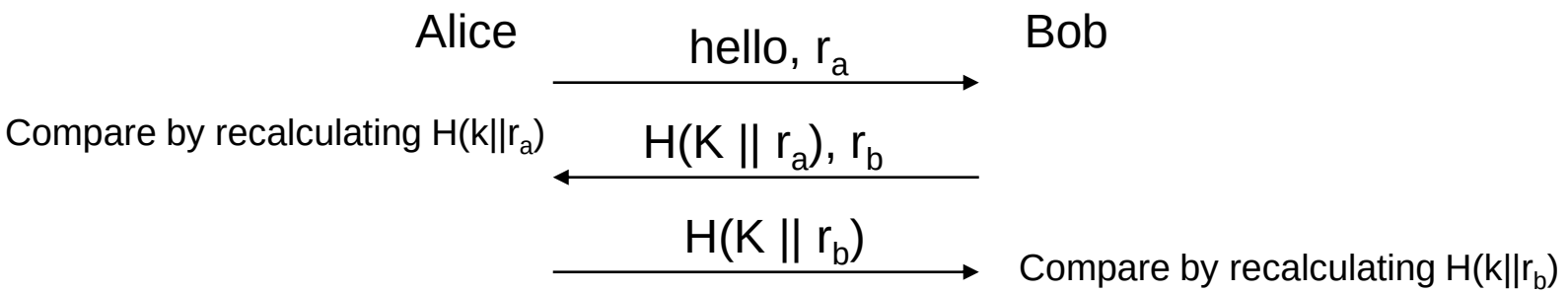
- **K is the key padded out to size**
- **opad, ipad are specified padding constants**
  - ipad =00110110 (36 in hex) repeated b/8 times
  - opad=01011100 (5C in hex) repeated b/8 times
    - ✓ b is the number of bits in a block
- **Any hash function can be used**
  - E.g., SHA-3
- **HMAC proven secure if embedded hash function has reasonable cryptographic strength**

HMAC Structure



Hash as Authentication Protocol

- **Challenge-response authentication instead of a password-based protocol:**



- **Hash is used instead of block cipher encryption $E_K(r_a)$, $E_K(r_b)$, & decryption.**