

# **Computer Security: Principles and Practice**

## **Chapter 20 – Public-Key Cryptography and Message Authentication**

First Edition

by William Stallings and Lawrie Brown

Lecture slides by Lawrie Brown

# Public-Key Cryptography and Message Authentication

- now look at technical detail concerning:
  - secure hash functions and HMAC
  - RSA & Diffie-Hellman Public-Key Algorithms

# Simple Hash Functions

- a one-way or secure hash function used in message authentication, digital signatures
- all hash functions process input a block at a time in an iterative fashion
- one of simplest hash functions is the bit-by-bit exclusive-OR (XOR) of each block

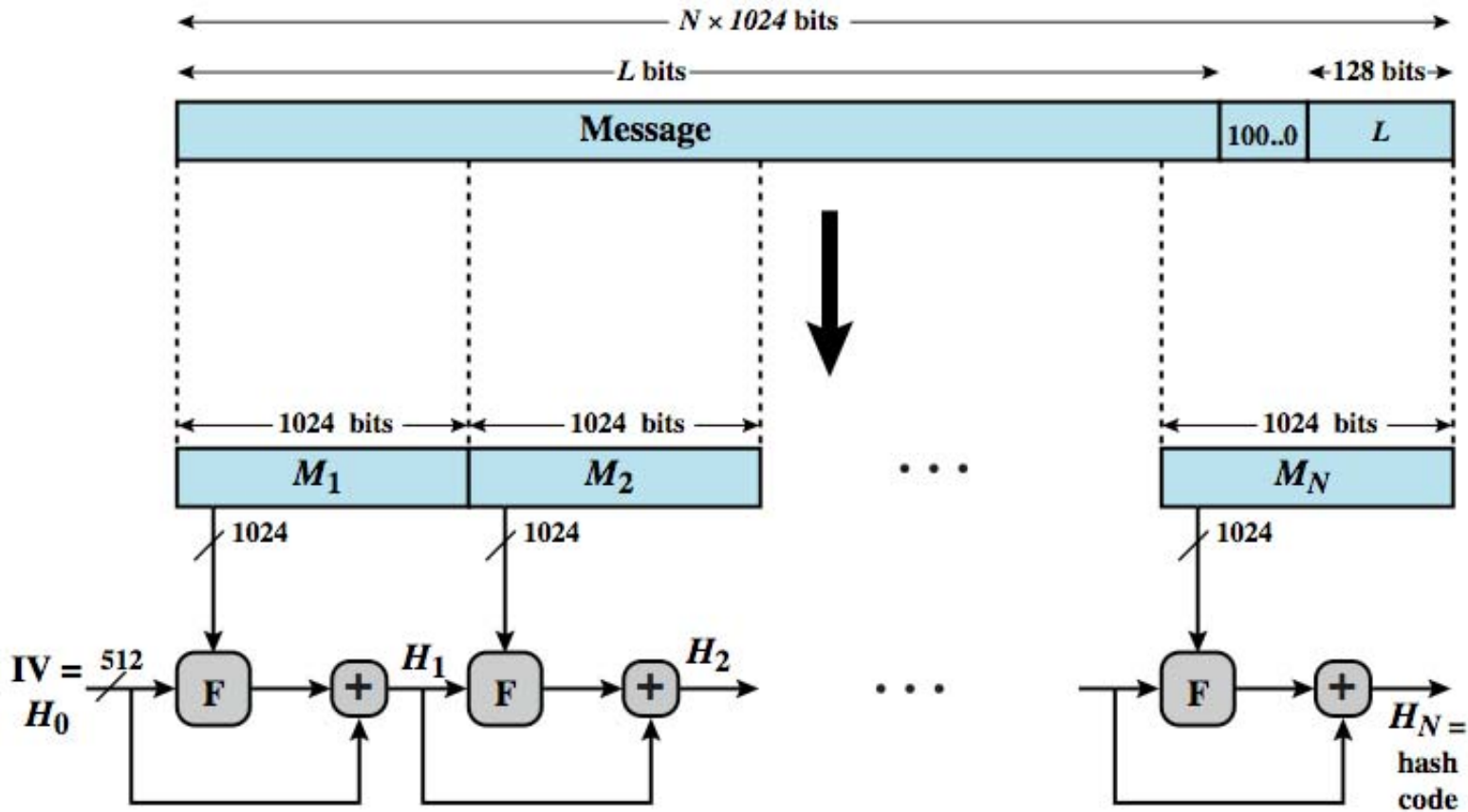
$$C_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{im}$$

- effective data integrity check on random data
- less effective on more predictable data
- virtually useless for data security

# SHA Secure Hash Functions

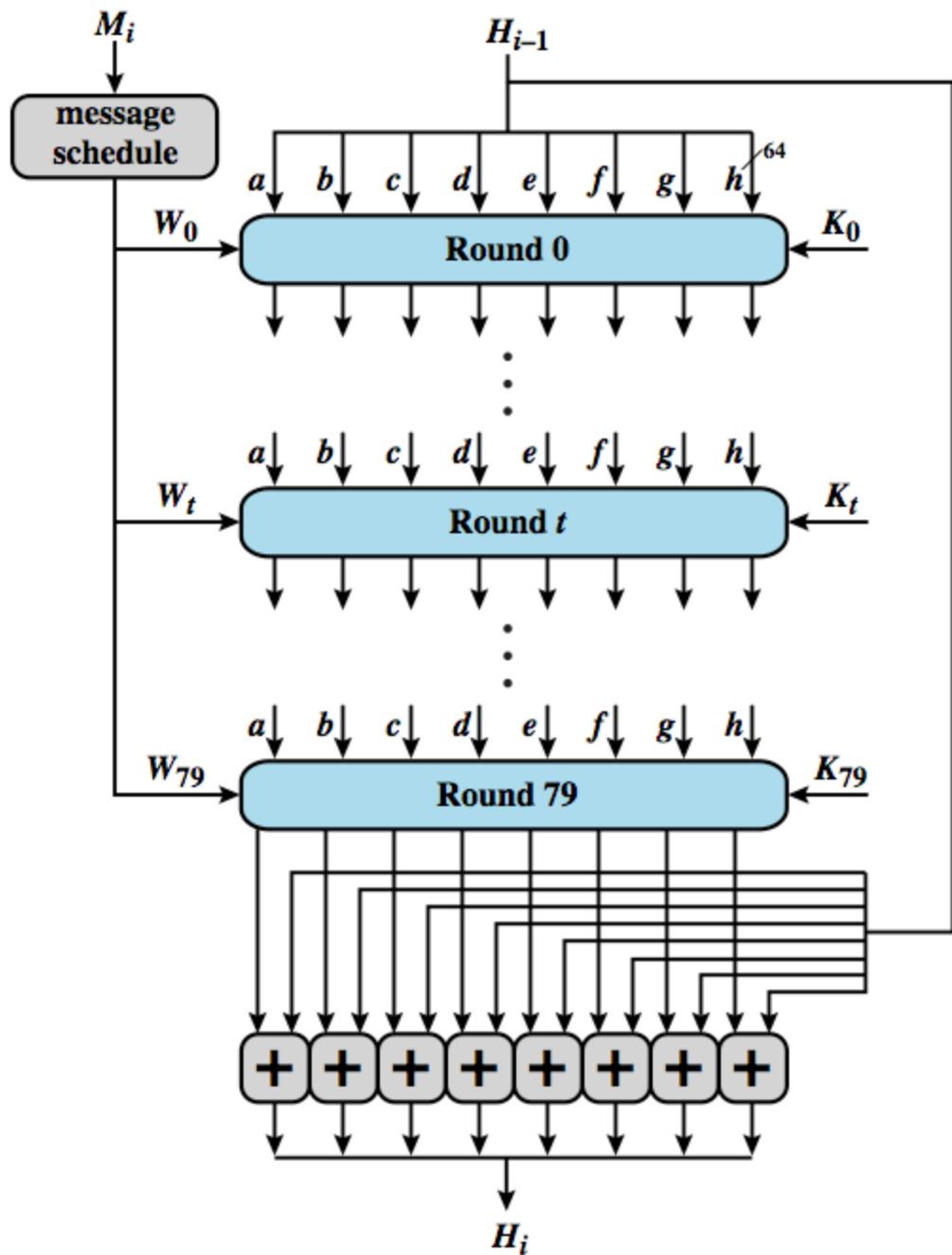
- SHA originally developed by NIST/NSA in 1993
- was revised in 1995 as SHA-1
  - US standard for use with DSA signature scheme
  - standard is FIPS 180-1 1995, also Internet RFC3174
  - produces 160-bit hash values
- NIST issued revised FIPS 180-2 in 2002
  - adds 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
  - with 256/384/512-bit hash values
  - same basic structure as SHA-1 but greater security
- NIST intend to phase out SHA-1 use

# SHA-512 Structure



$+$  = word-by-word addition mod  $2^{64}$

# SHA-512 Round



# Other Secure Hash Functions

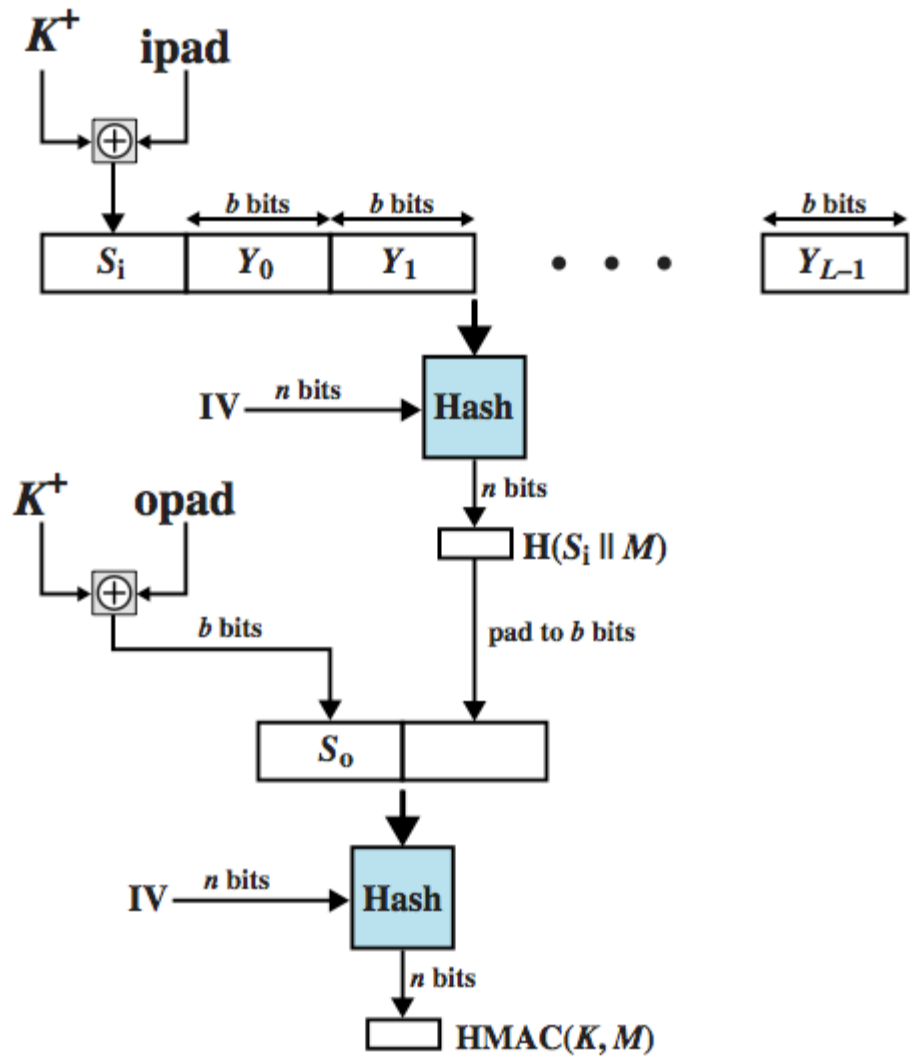
- most based on iterated hash function design
  - if compression function is collision resistant
  - so is resultant iterated hash function
- MD5 (RFC1321)
  - was a widely used hash developed by Ron Rivest
  - produces 128-bit hash, now too small
  - also have cryptanalytic concerns
- Whirlpool (NESSIE endorsed hash)
  - developed by Vincent Rijmen & Paulo Barreto
  - compression function is AES derived W block cipher
  - produces 512-bit hash

# HMAC

- interest a MAC using a cryptographic hash
  - due to speed and code availability
- must incorporate key into use of hash alg
- HMAC (RFC2104) widely supported
  - used in IPsec, TLS & SET
- HMAC treats hash as “black box”
- HMAC proven secure if embedded hash function has reasonable cryptographic strength



# HMAC Structure



# Security of HMAC

- security based on underlying hash strength
- have prob given time and no msg-MAC's
- either attacker computes output even with random secret IV
  - brute force key  $O(2^n)$ , or use birthday attack
- or attacker finds collisions in hash function even when IV is random and secret
  - ie. find  $M$  and  $M'$  such that  $H(M) = H(M')$
  - birthday attack  $O(2^{n/2})$
  - MD5 secure in HMAC since only observe

# RSA Public-Key Encryption

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key alg
- uses exponentiation of integers modulo a prime
- encrypt:  $C = M^e \bmod n$
- *decrypt*:  $M = C^d \bmod n = (M^e)^d \bmod n = M$
- both sender and receiver know values of  $n$  and  $e$
- only receiver knows value of  $d$
- public-key encryption algorithm with
  - public key  $PU = \{e, n\}$  & private key  $PR = \{d, n\}$ .

# RSA Algorithm

## Key Generation

Select $p, q$	$p$ and $q$ both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer $e$	$\text{gcd}(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate $d$	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

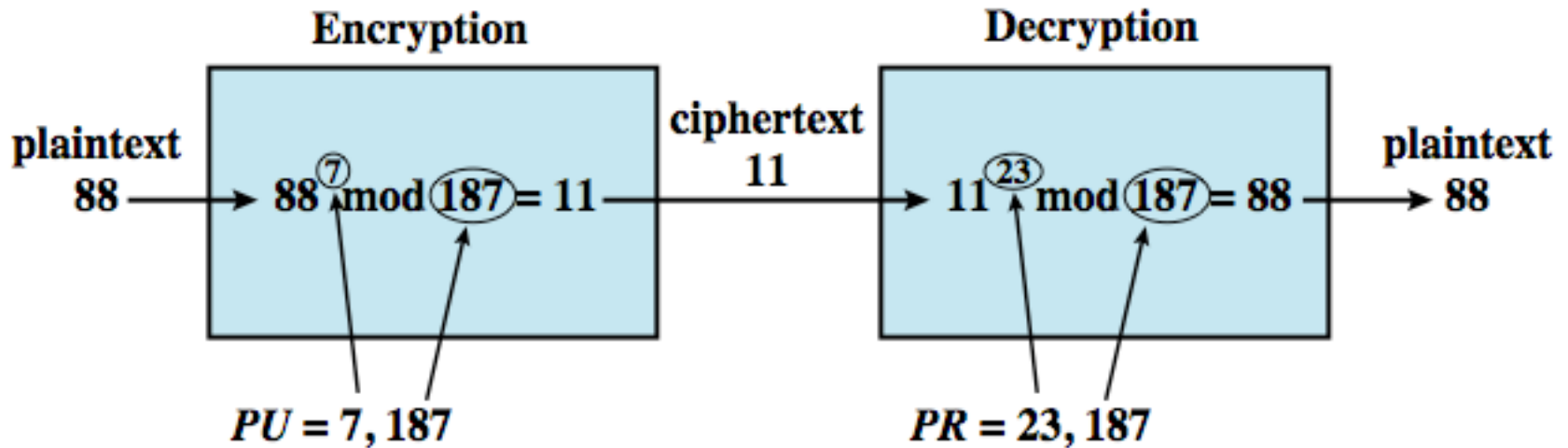
## Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

## Decryption

Ciphertext:	$C$
Plaintext:	$M = C^d \pmod{n}$

# RSA Example



# Attacks on RSA

- brute force
  - trying all possible private keys
  - use larger key, but then slower
- mathematical attacks (factoring  $n$ )
  - see improving algorithms (QS, GNFS, SNFS)
  - currently 1024-2048-bit keys seem secure
- timing attacks (on implementation)
  - use - constant time, random delays, blinding
- chosen ciphertext attacks (on RSA props)

# Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that Williamson (UK CESG) secretly proposed the concept in 1970
- practical method to exchange a secret key
- used in a number of commercial products
- security relies on difficulty of computing discrete logarithms

# Diffie-Hellman Algorithm

## Global Public Elements

$q$  prime number  
 $\alpha$   $\alpha < q$  and  $\alpha$  a primitive root of  $q$

## User A Key Generation

Select private  $X_A$   $X_A < q$   
Calculate public  $Y_A$   $Y_A = \alpha^{X_A} \text{ mod } q$

## User B Key Generation

Select private  $X_B$   $X_B < q$   
Calculate public  $Y_B$   $Y_B = \alpha^{X_B} \text{ mod } q$

## Generation of Secret Key by User A

$$K = (Y_B)^{X_A} \text{ mod } q$$

## Generation of Secret Key by User B

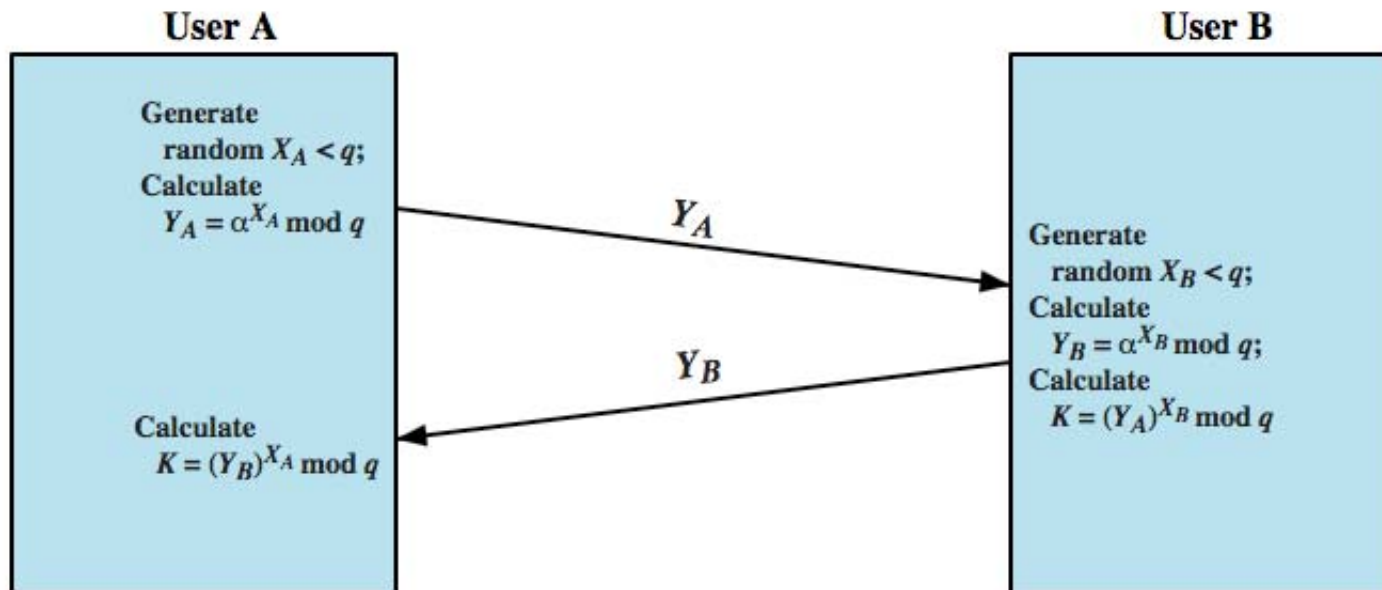
$$K = (Y_A)^{X_B} \text{ mod } q$$



# Diffie-Hellman Example

- have
  - prime number  $q = 353$
  - primitive root  $\alpha = 3$
- A and B each compute their public keys
  - A computes  $Y_A = 3^{97} \bmod 353 = 40$
  - B computes  $Y_B = 3^{233} \bmod 353 = 248$
- then exchange and compute secret key:
  - for A:  $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$
  - for B:  $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$
- attacker must solve:
  - $3^a \bmod 353 = 40$  which is hard
  - desired answer is 97, then compute key as B does

# Key Exchange Protocols



# Man-in-the-Middle Attack

➤ attack is:

1. Darth generates private keys  $X_{D1}$  &  $X_{D2}$ , and their public keys  $Y_{D1}$  &  $Y_{D2}$
2. Alice transmits  $Y_A$  to Bob
3. Darth intercepts  $Y_A$  and transmits  $Y_{D1}$  to Bob. Darth also calculates  $K2$
4. Bob receives  $Y_{D1}$  and calculates  $K1$
5. Bob transmits  $X_A$  to Alice
6. Darth intercepts  $X_A$  and transmits  $Y_{D2}$  to Alice. Darth calculates  $K1$
7. Alice receives  $Y_{D2}$  and calculates  $K2$

➤ all subsequent communications compromised

# Other Public-Key Algorithms

- Digital Signature Standard (DSS)
  - FIPS PUB 186 from 1991, revised 1993 & 96
  - uses SHA-1 in a new digital signature alg
  - cannot be used for encryption
- elliptic curve cryptography (ECC)
  - equal security for smaller bit size than RSA
  - seen in standards such as IEEE P1363
  - still very new, but promising
  - based on a mathematical construct known as the elliptic curve (difficult to explain)

# Summary

- discussed technical detail concerning:
  - secure hash functions and HMAC
  - RSA & Diffie-Hellman Public-Key Algorithms