CS 5173/4173 Computer Security

Midterm review

Covered Topics

- Lectures 1 - 5
  - Basic Security Concepts
  - Introduction to Cryptography
  - DES, AES
  - Modes of Block Cipher Operations
  - Double DES and Triple DES
  - Cryptographic Hash Functions

Type of Questions

- Multiple choices (single-select)
- Short answer questions (about important concepts mentioned in the class)
- Open-ended questions

Introduction to Cryptography

- Basic Security Concepts
  - Confidentiality, integrity, availability
- Introduction to Cryptography
  - Secret key cryptography
    - Sender and receiver share the same key
    - Applications
      - Communication over insecure channel, Secure storage, Authentication, Integrity check

Introduction to Cryptography

- Introduction to Cryptography
  - Public key cryptography
    - Public key: publicly known
    - Private key: kept secret by owner
    - Encryption/decryption mode
      - How the keys are used?
    - Digital signature mode
      - How the keys are used?
    - Application: Secure communication, secure storage, authentication, digital signature, key exchange

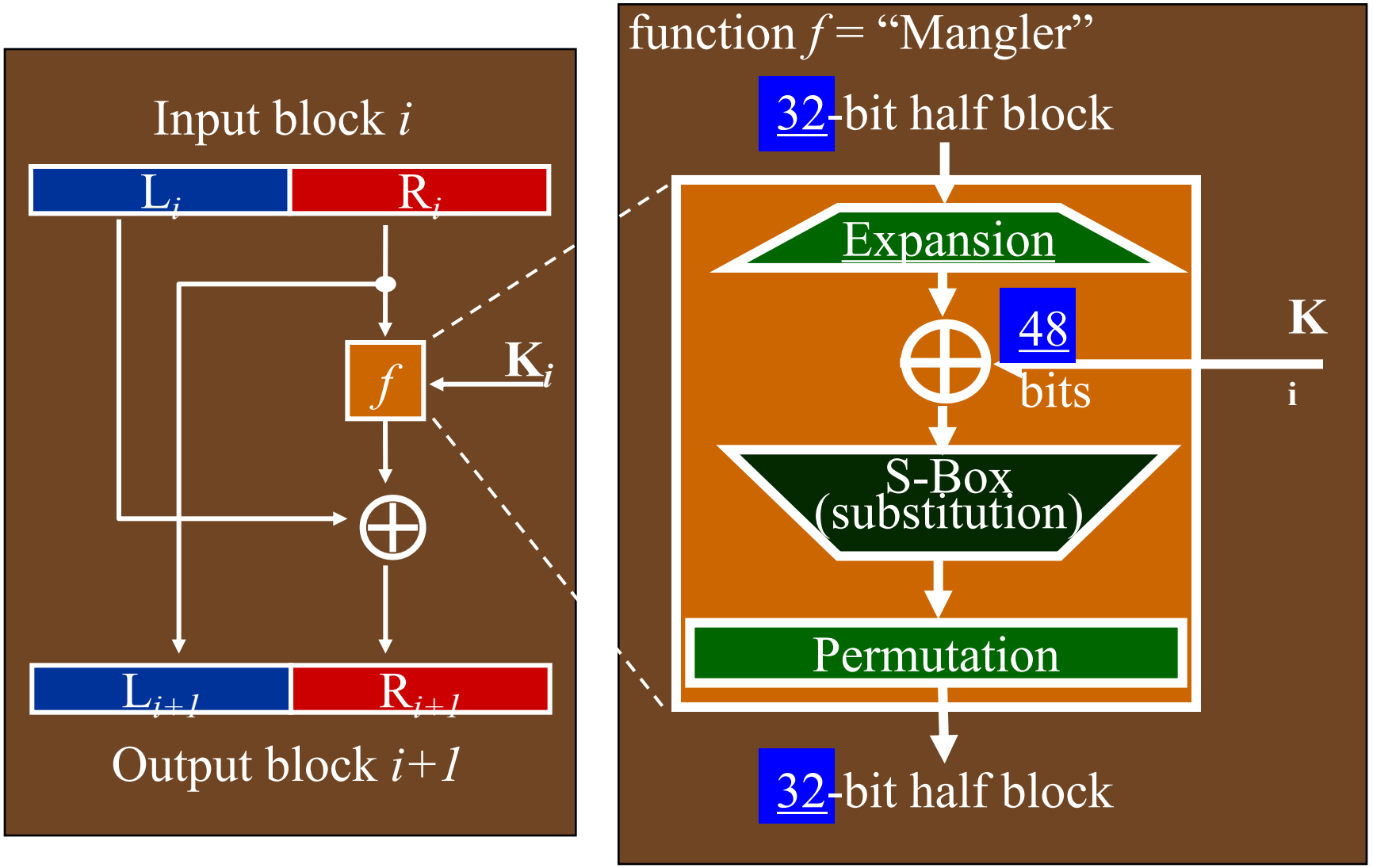
Introduction to Cryptography

- Introduction to Cryptography
  - Hash function
    - Map a message of arbitrary length to a fixed-length short message
  - Desirable properties
    - Performance, one-way, weak collision free, strong collision free

DES

- DES
  - Parameters
    - Block size (input/output 64 bits)
    - key size (56 bits)
    - number of rounds (16 rounds)
    - subkey generalization algorithm
    - round function

DES Round: *f* (Mangler) Function



Modes of Block Cipher Operations

- ECB (Electronic Code Book)
- CBC (Cipher Block Chaining Mode)
- OFB (Output Feedback Mode)
- CFB (Cipher Feedback Mode)
- CTR (Counter Mode)

Modes of Block Cipher Operations

- Properties of Each Mode
  - Chaining dependencies
  - Error propagation
  - Error recovery

Double DES and Triple DES

- You need to understand how double and triple DES works
  - Double DES C=Ek2(Ek1(P))
  - Triple DES C = Ek1(Dk2(Ek1(P))
  - Meet-in-the-middle attacks
  - Operation modes using Triple DES

The Meet-in-the-Middle Attack

1. Choose a plaintext P and generate ciphertext C, using double-DES with $\mathcal{K}1+\mathcal{K}2$
2. Then…
   a. encrypt P using single-DES for all possible 2^{56} values K_1 to generate all possible single-DES ciphertexts for P:
      $X_1, X_2, \ldots, X_{2^{56}}$;
      store these in a table indexed by ciphertex values
   b. decrypt C using single-DES for all possible 2^{56} values K_2 to generate all possible single-DES plaintexts for C:
      $Y_1, Y_2, \ldots, Y_{2^{56}}$;
      for each value, check the table

Steps … (Cont'd)

3. Meet-in-the-middle:
   - Each match ($X_i = Y_j$) reveals a *candidate key pair* K_i+K_j
   - There are 2^{112} pairs but there are only 2^{64} X's
4. On average, how many pairs have identical X and Y?
   - For any pair (X, Y), the probability that X = Y is $1/ 2^{64}$
   - There are 2^{112} pairs.
   - The average number of pairs that result in identical X and Y is $2^{112} / 2^{64} = 2^{48}$

Steps … (Cont'd)

5. The attacker uses a second pair of plaintext and ciphertext to try the 2^{48} Key pairs

– There are 2^{48} pairs and there are 2^{64} X's (Y's)

— The average number of pairs that result in identical X and Y is $2^{48} / 2^{64} = 2^{-16}$

— The expected number of survived candidate key pairs is less than 1. Tfter examine two pairs of plaintext and ciphtertext, the attacker identifies the key

Hash Function

- Also known as
  - Message digest
  - One-way transformation
  - One-way function
  - Hash
- Length of $H(m)$ much shorter then length of m
- Usually fixed lengths: 128 or 160 bits

Desirable Properties of Hash Functions

- Consider a hash function H
  - Performance: Easy to compute H(*m*)
  - One-way property: Given H(*m*) but not *m*, it's computationally infeasible to find *m*
  - Weak collision resistance (free): Given H(*m*), it's computationally infeasible to find *m'* such that H(*m'*) = H(*m*).
  - Strong collision resistance (free): Computationally infeasible to find m_1, m_2 such that H(m_1) = H(m_2)

Length of Hash Image

- Question
  - Why do we have 128 bits or 160 bits in the output of a hash function?
  - If it is too long
    - Unnecessary overhead
  - If it is too short
    - Loss of strong collision free property
    - Birthday paradox

Birthday Paradox (Cont'd)

- Implication for hash function H of length m
  - The hash value of an arbitrary input message is randomly distributed between 1 and 2^m
  - What is the least value of *k* such that
    - If we hash k messages, the probability that at least two of them have the same hash is larger than 0.5?

$$k \approx \sqrt{n} = \sqrt{2^m} = 2^{m/2}$$

  - Birthday attack
    - Choose m $\geq$ 128

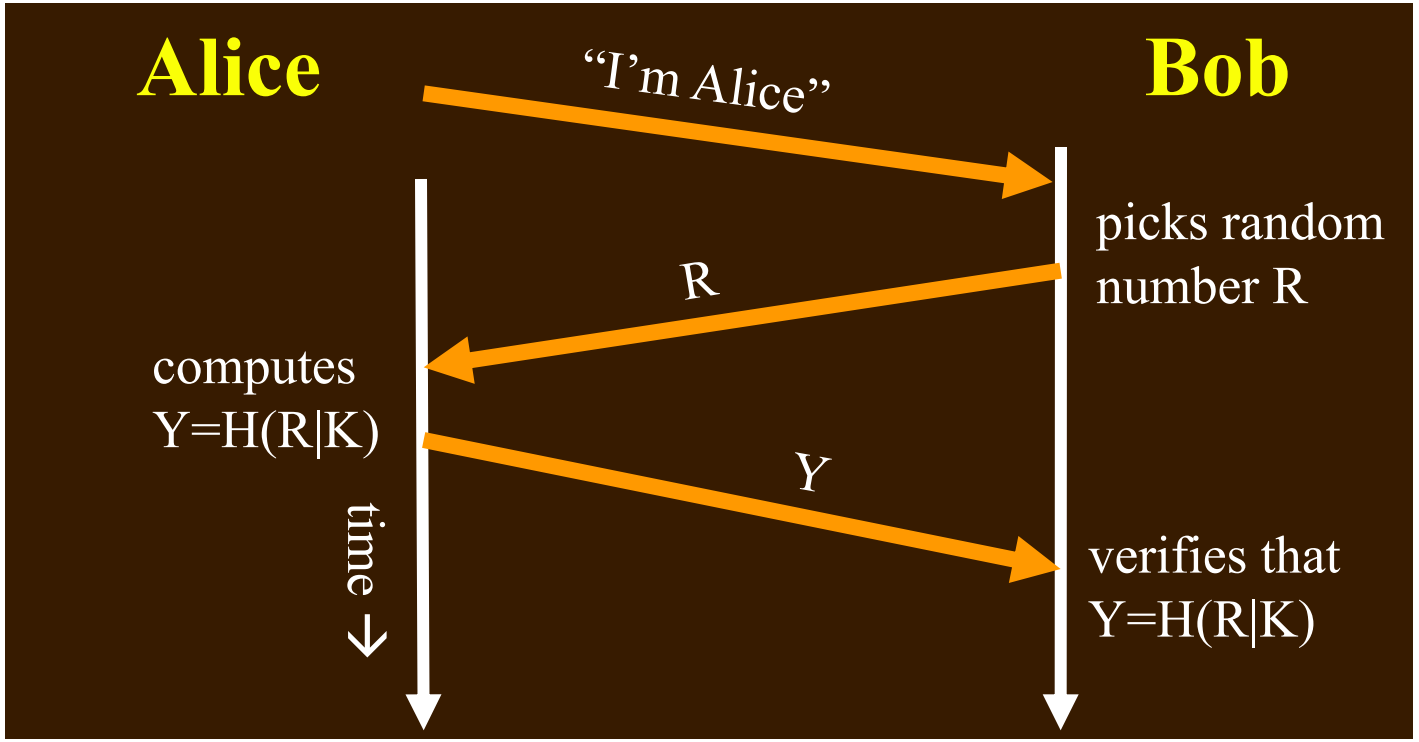
Application: File Authentication

- Want to detect if a file has been changed by someone after it was stored
- Method
  - Compute a hash H(F) of file F
  - Store H(F) separately from F
  - Can tell at any later time if F has been changed by computing H(F') and comparing to stored H(F)

- Why not just store a duplicate copy of F???

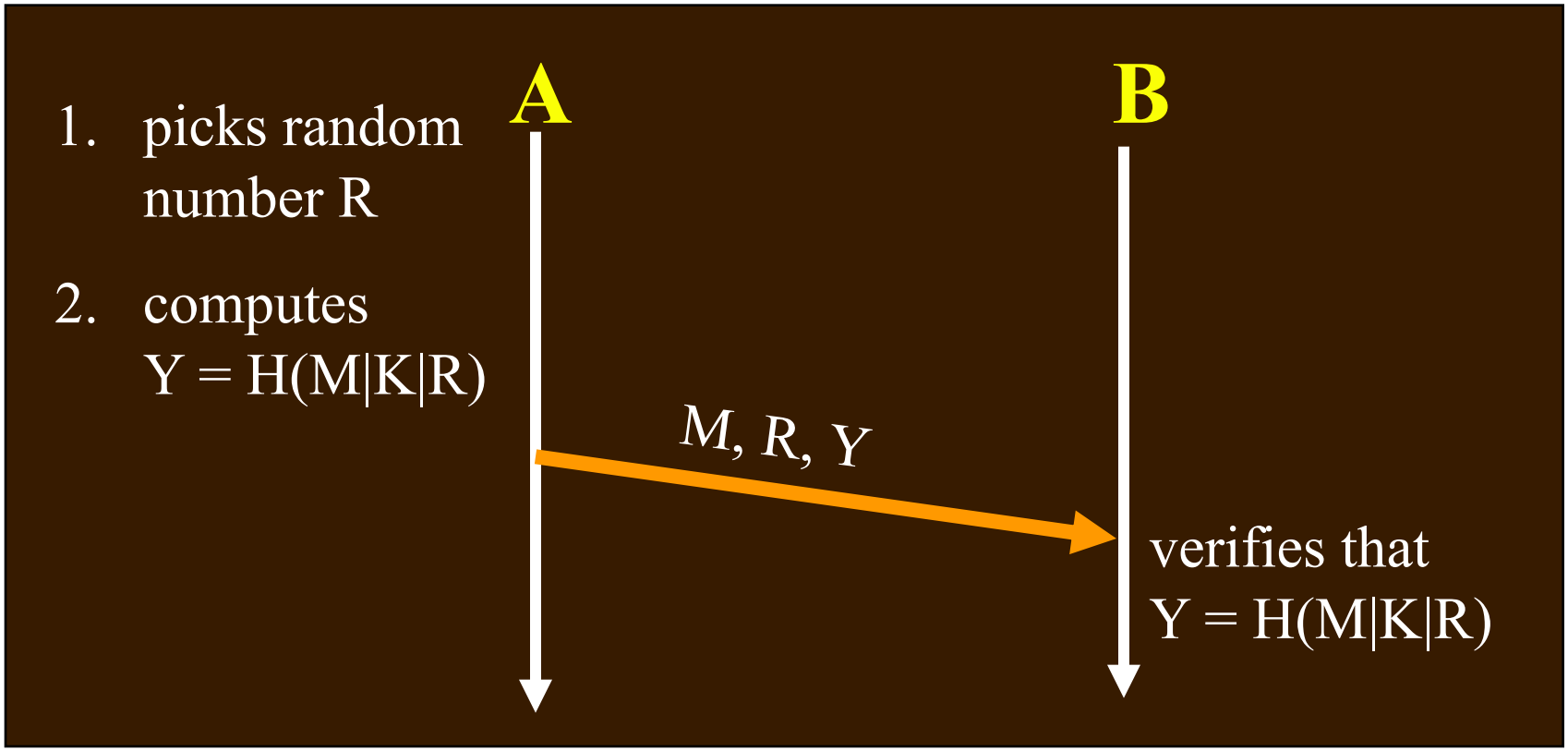
Application: User Authentication

- Alice wants to authenticate herself to Bob
  - assuming they already share a secret key K
- Protocol:



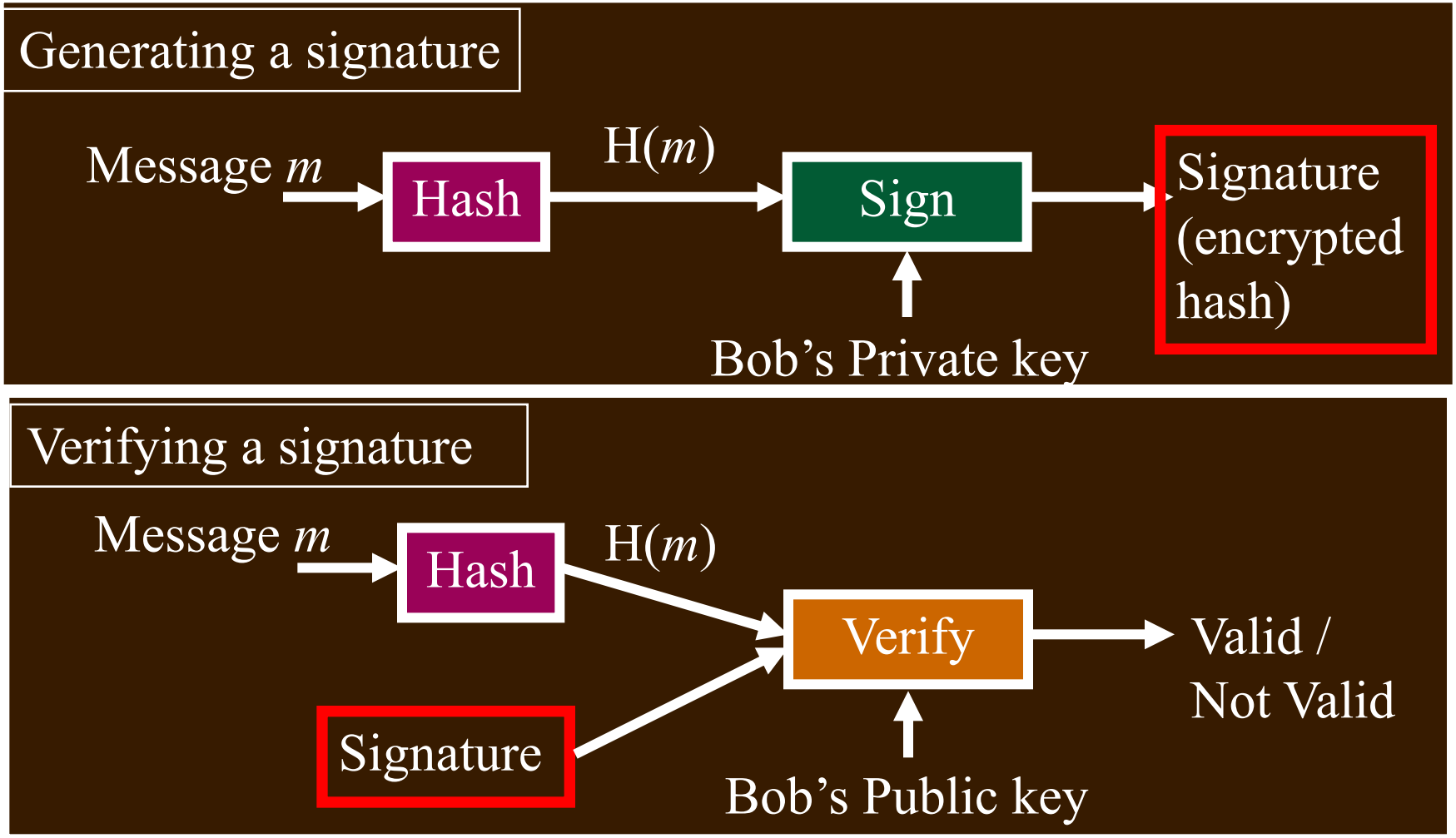
Application: Message Integrity

- A wishes to authenticate (but not encrypt) a message M (and A, B share secret key K)



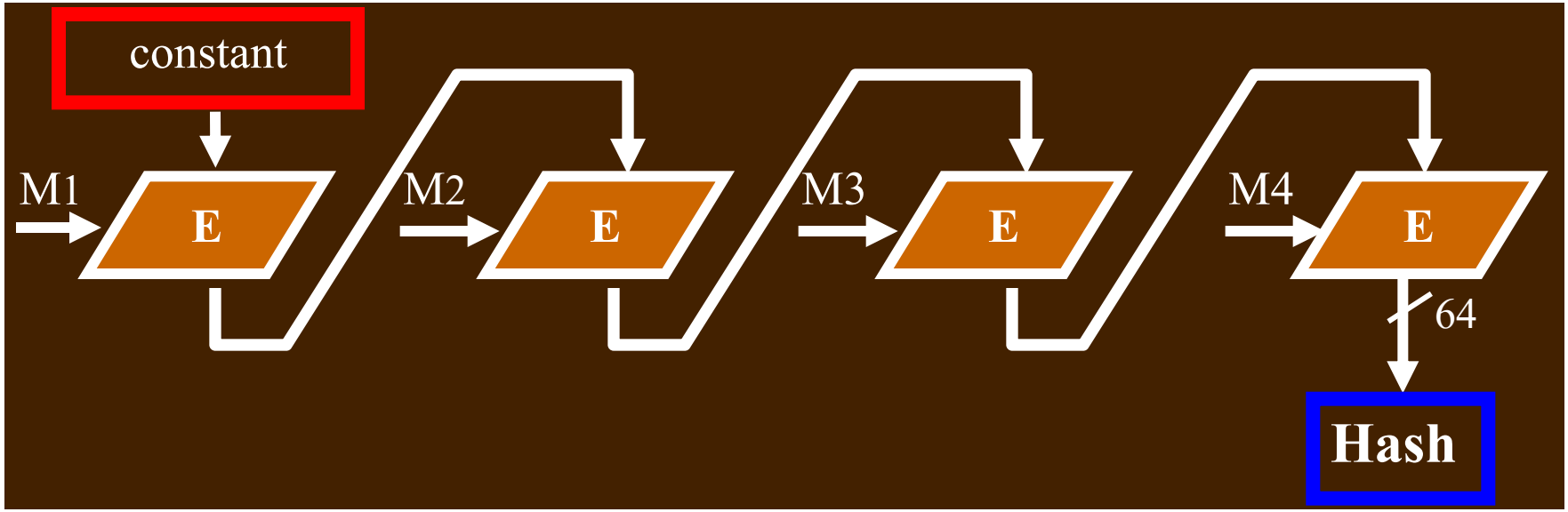
- Why is R needed?
- Why is K needed?

Application: Digital Signatures



- Only one party (Bob) knows the private key

Is Encryption a Good Hash Function?



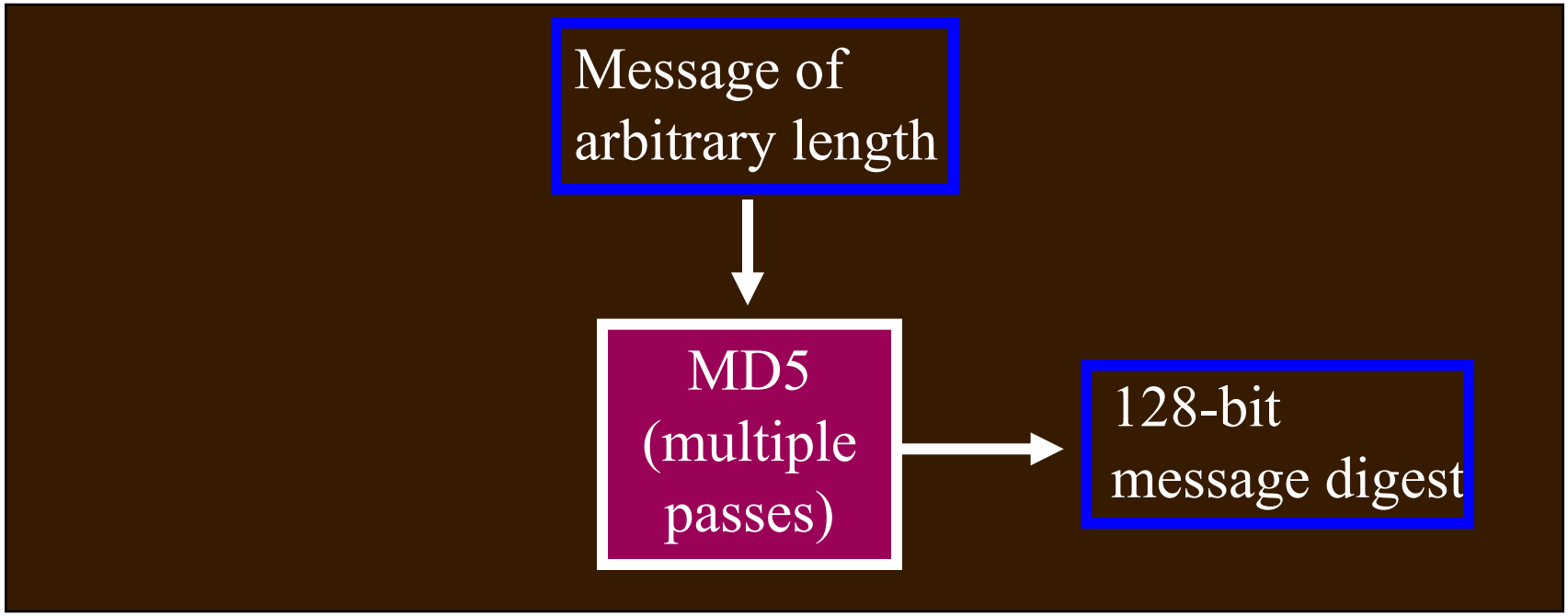
- Building hash using block chaining techniques
  - Encryption block size may be too short (DES=64)
    - Birthday attack
  - Expensive in terms of computation time

Modern Hash Functions

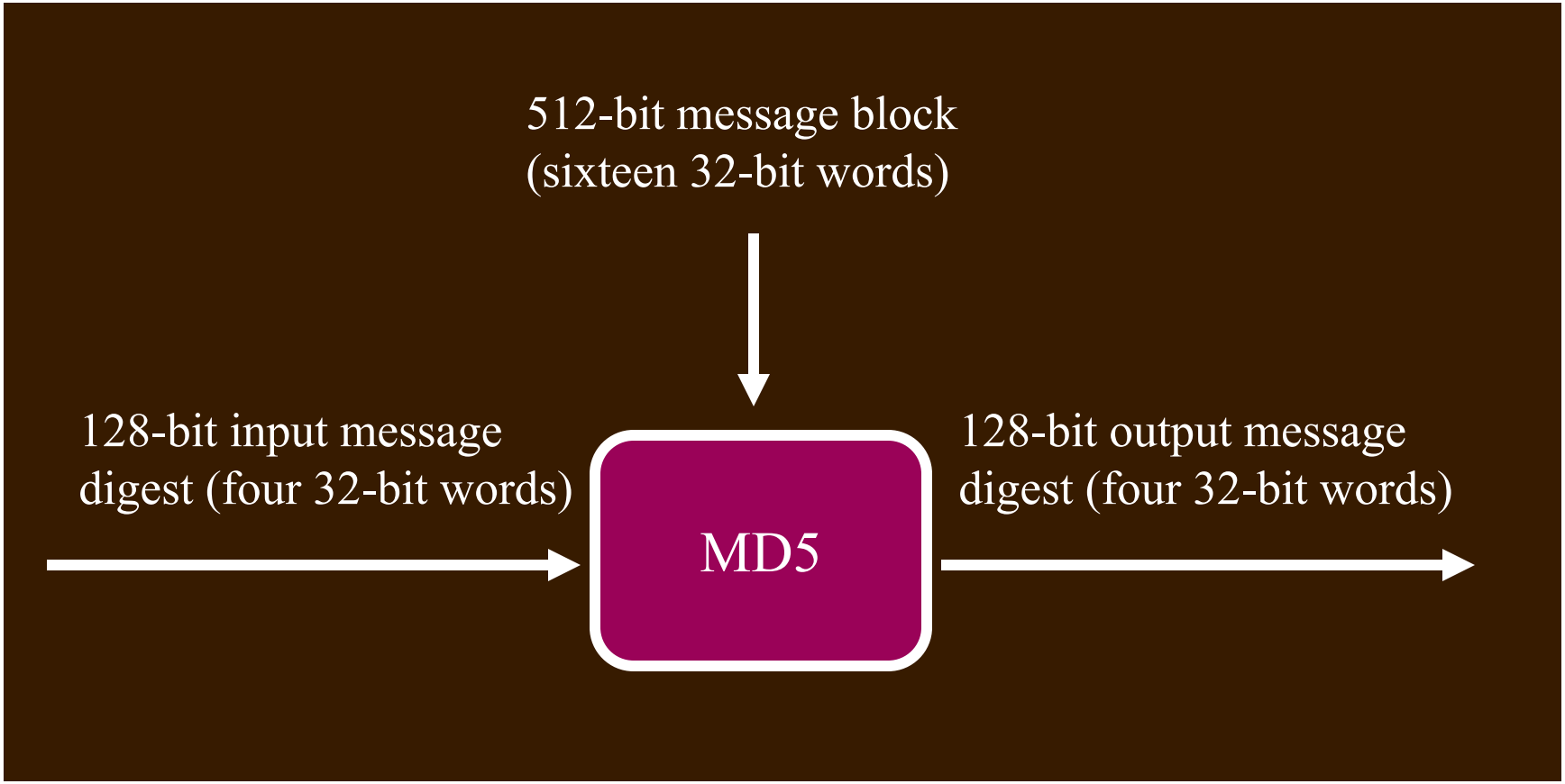
- MD5
  - Previous versions (i.e., MD2, MD4) have weaknesses.
  - Broken; collisions published in August 2004
  - Previous versions are too weak to be used for serious applications
- SHA (Secure Hash Algorithm)
  - Weaknesses were found
- SHA-1
  - Collisions in 2^{69} hash operations, much less than the birthday attack of 2^{80} operations
- SHA-256, SHA-384, …

MD5: Message Digest Version 5

- MD5 at a glance

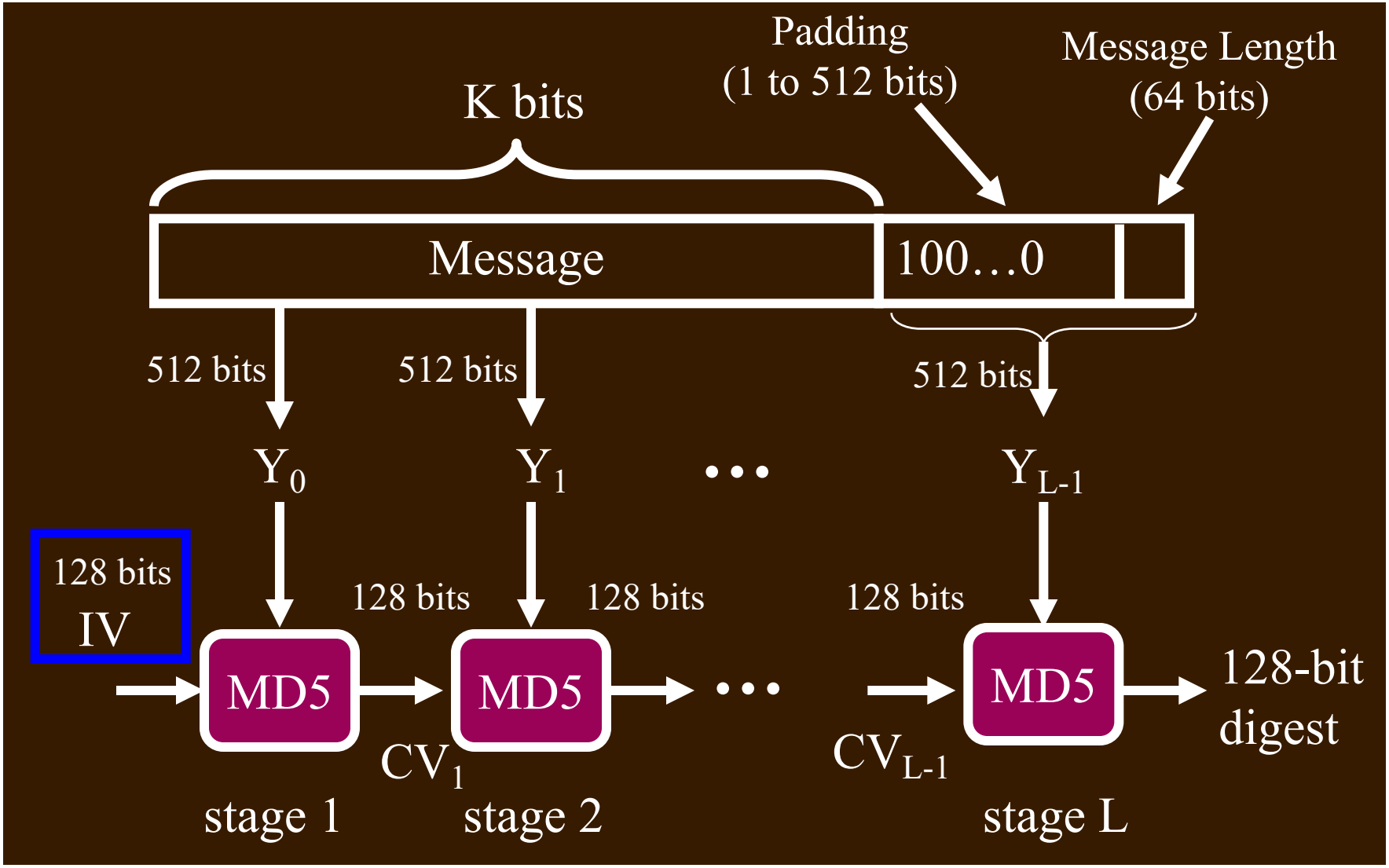


Processing of A Single Block



Called a compression function

MD5: A High-Level View



Padding

- There is always padding for MD5, and padded messages must be multiples of 512 bits
- To original message M, add padding bits "10…0"
  - enough 0's so that resulting total length is 64 bits less than a multiple of 512 bits
- Append L (original length of M), represented in 64 bits, to the padded message

- Footnote: the bytes of each 32-bit word are stored in little-endian order (LSB to MSB)

Secure Hash Algorithm (SHA)

- Developed by NIST, specified in the Secure Hash Standard, 1993
- SHA is specified as the hash algorithm in the Digital Signature Standard (DSS)
- SHA-1: revised (1995) version of SHA

SHA-1 Parameters

- Input message must be < 2^{64} bits
- Input message is processed in 512-bit blocks, with the same padding as MD5
- Message digest output is 160 bits long
  - Referred to as five 32-bit words **A, B, C, D, E**
  - IV: **A** = 0x67452301, **B** = 0xEFCDAB89, **C** = 0x98BADCFE, **D** = 0x10325476, **E** = 0xC3D2E1F0