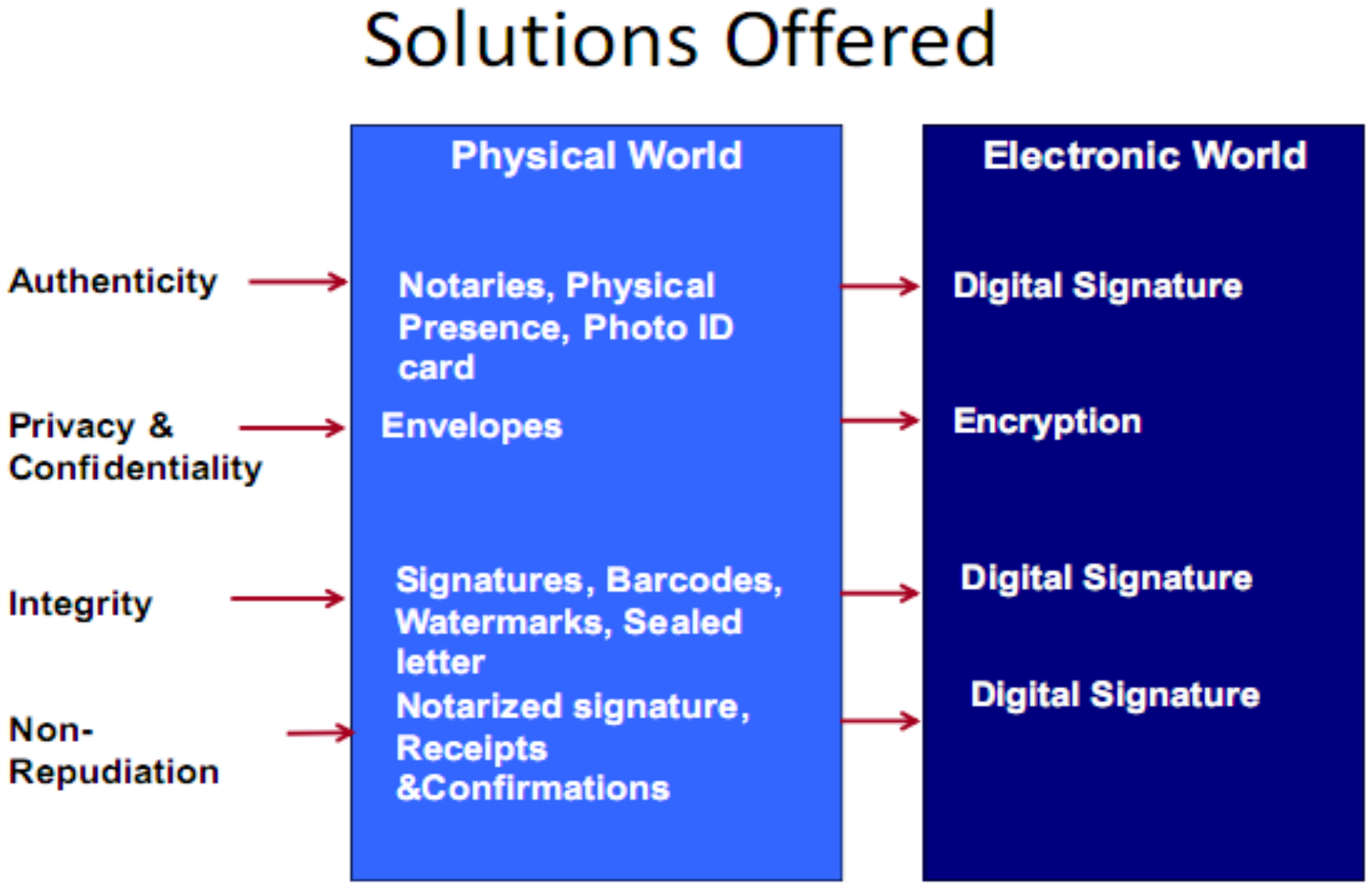
What is a digital signature?

- A digital signature is a kind of ID.
- You can use it on the Internet to identify yourself in a secure manner.
- This is extremely useful in areas such as electronic commerce.
- For instance, when making a credit card purchase on the Internet, you can use your digital signature to "sign" that purchase.
- This helps to ensure that only you can make purchases with your credit card number.

Digital Signature

- A mathematical scheme for demonstrating the reliability of a digital message in terms of
  - **Authentication**
    - a valid signature (generated via user's private key) shows that the message was sent by that user
  - **Integrity**
    - message has not been altered during transmission
  - **Non-Repudiation**
    - an entity that has signed some information cannot at a later time deny having signed it
- Can be accomplished using
  - RSA
  - MAC (Message Authentication Code)

Digital Signatures



Digital Signature

- Digital signature can be used in all electronic communications
  - Web, e-mail, e-commerce
- It is an electronic stamp or seal that append to the document.
- Ensure the document being unchanged during transmission.

Digital signatures provide Message Authentication

- Secure communication includes
  - Privacy/Secrecy/Confidentiality
  - Authenticity
  - Integrity
  - Non-repudiation
- Digital signatures provide last three of these security services.
- Message authentication and integrity check plays an important role in a variety of applications:
  - Internet protocols
  - Network management
  - Wherever undetected manipulation of messages can have disastrous effects

Digitized Written Signature??

- Simply taking a digital picture of a written signature does not provide adequate security.
- Such a digitized written signature could easily be copied from one electronic document to another with no way to determine whether it is legitimate.
- Electronic signatures, on the other hand, are unique to the message being signed and will not verify if they are copied to another document.

Digital signatures are used just like handwritten signatures

- Digital signatures are used just like handwritten signatures.
- When you add them to a document, you are "signing" that document as a way of endorsing or agreeing with what the document says.
- Unlike handwritten signatures, digital signatures are used only with computers. They are electronic signatures that can be used to sign electronic documents, like word processing files or spreadsheets.

Importance of Digital Signatures

- Digital Signatures are a central component of modern cryptographic systems.
- In analogy to handwritten signatures on paper documents digital signatures are used to guarantee the authenticity of electronic documents.
- Thus they play an important role for example in secure and reliable systems for electronic commerce.

Requirements for a Digital Signature

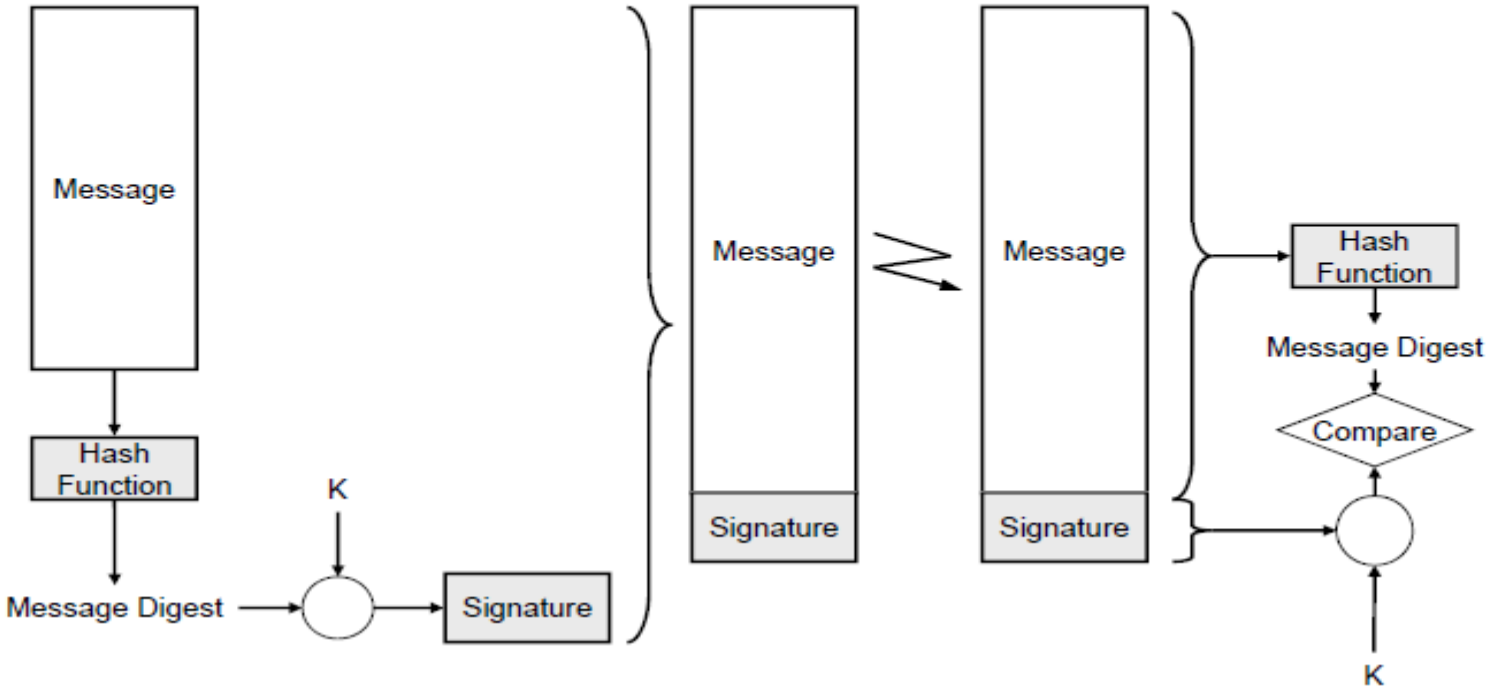
1. The signature must be a bit pattern that depends on the message being signed
2. The signature must use some information unique to the sender, to prevent both forgery and denial.
3. It must be relatively easy to produce digital signature.
4. It must be relatively easy to recognize and verify the digital signature.
5. It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
6. It must be practical to retain a copy of the digital signature in storage.

(a) Conventional (symmetric) Encryption
$A \rightarrow B: E_K[M]$ •Provides confidentiality —Only A and B share K •Provides a degree of authentication —Could come only from A —Has not been altered in transit —Requires some formatting/redundancy •Does not provide signature —Receiver could forge message —Sender could deny message
(b) Public-Key (asymmetric) Encryption
$A \rightarrow B: E_{KU_b}[M]$ •Provides confidentiality —Only B has KR_b to decrypt •Provides no authentication —Any party could use KU_b to encrypt message and claim to be A
$A \rightarrow B: E_{KR_a}[M]$ •Provides authentication and signature —Only A has KR_a to encrypt —Has not been altered in transit —Requires some formatting/redundancy —Any party can use KU_a to verify signature
$A \rightarrow B: E_{KUb}[E_{KRa}(M)]$ •Provides confidentiality because of KU_b •Provides authentication and signature because of KR_a

Message Authentication Using Plain Encryption

24

Digital Signatures with Conventional Encryption and Hash Functions



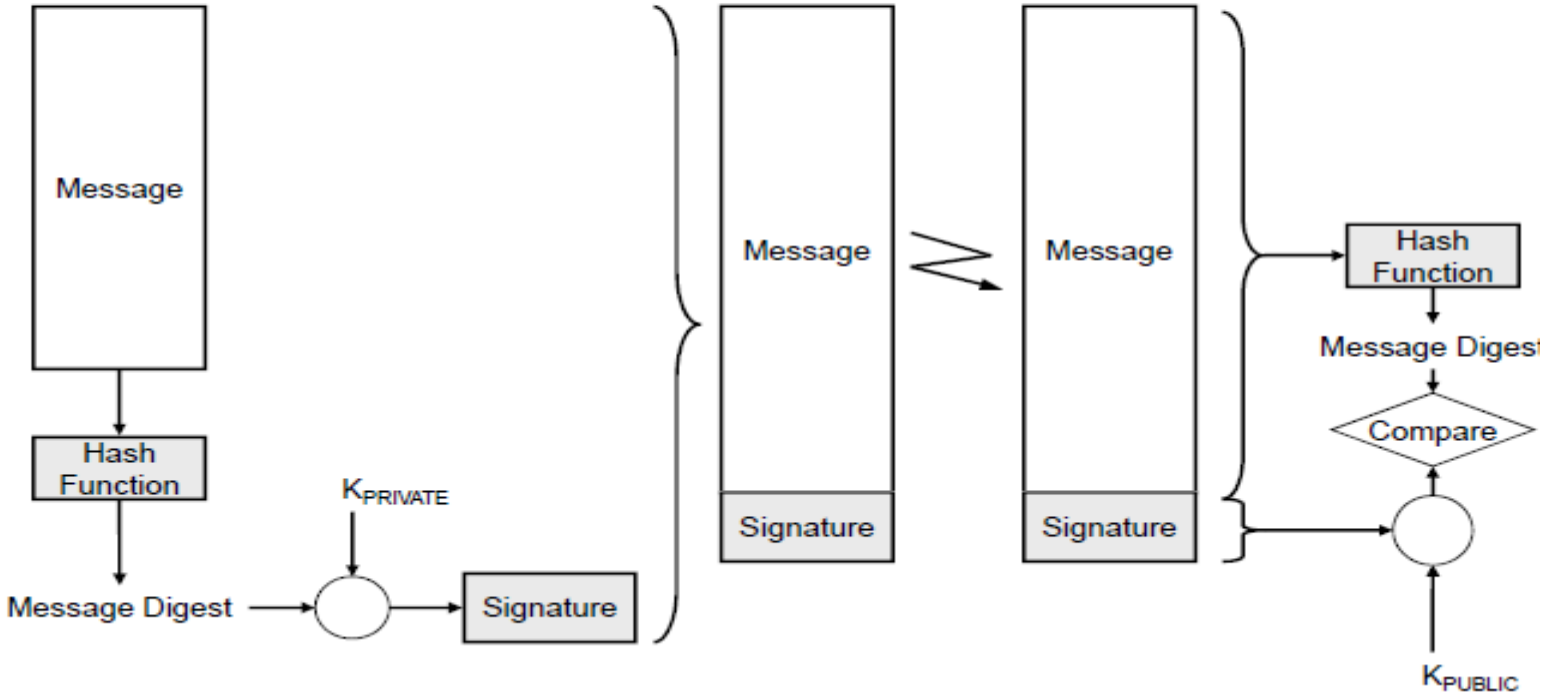
Public Key Electronic Signatures

- Another type of electronic signature is implemented using public key cryptography.
- Data is electronically signed by applying the originator's private key to the data.
- To increase the speed of the process, the private key is applied to a shorter form of the data, called a "hash" or "message digest," rather than to the entire set of data.
- The resulting digital signature can be stored or transmitted along with the data.

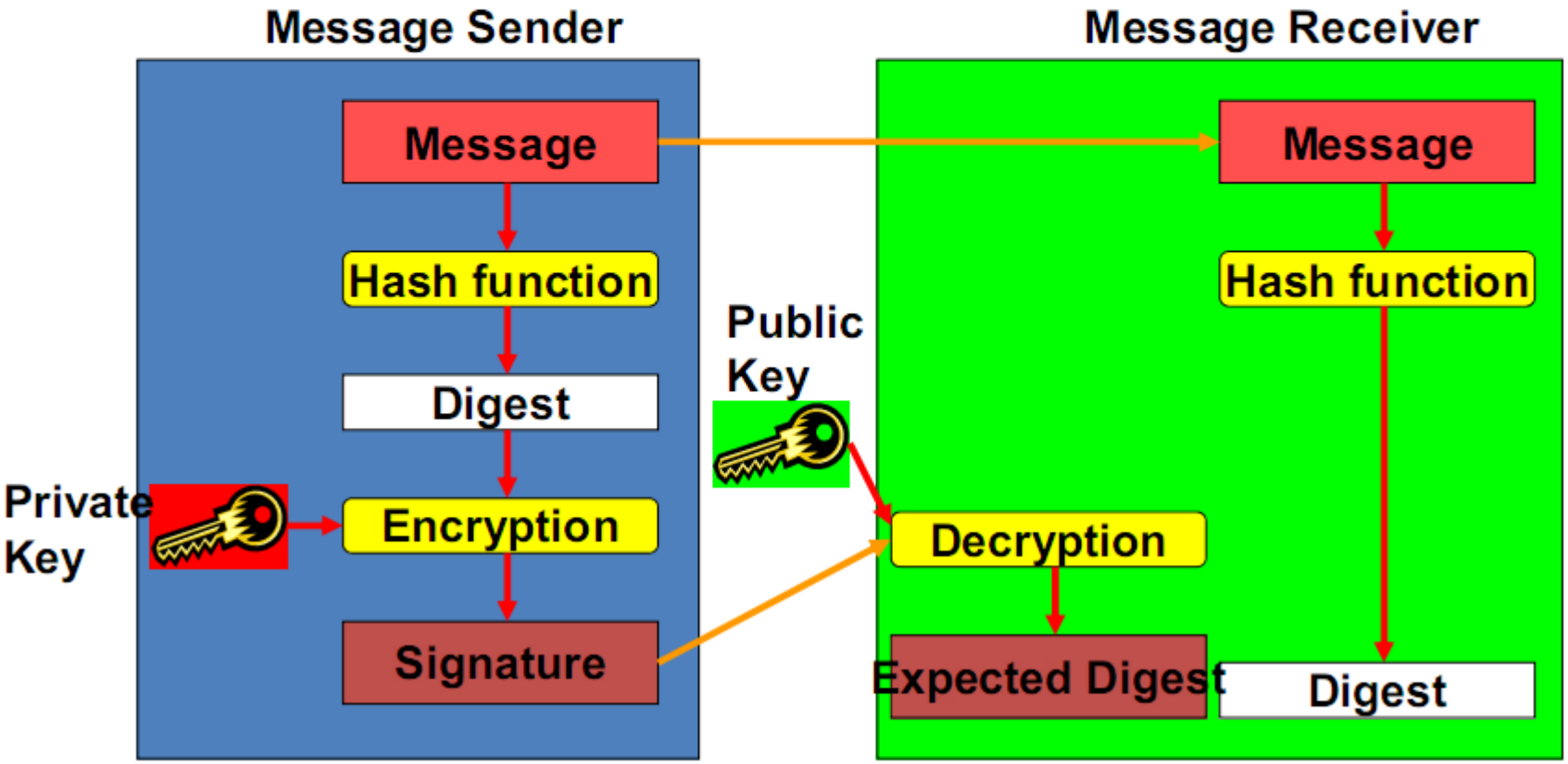
Public Key Electronic Signatures

- The signature can be verified by any party using the public key of the signer.
- This feature is very useful, for example, when distributing signed copies of virus-free software. Any recipient can verify that the program remains virus-free.
- If the signature verifies properly, then the verifier has confidence that the data was not modified after being signed and that the owner of the public key was the signer.

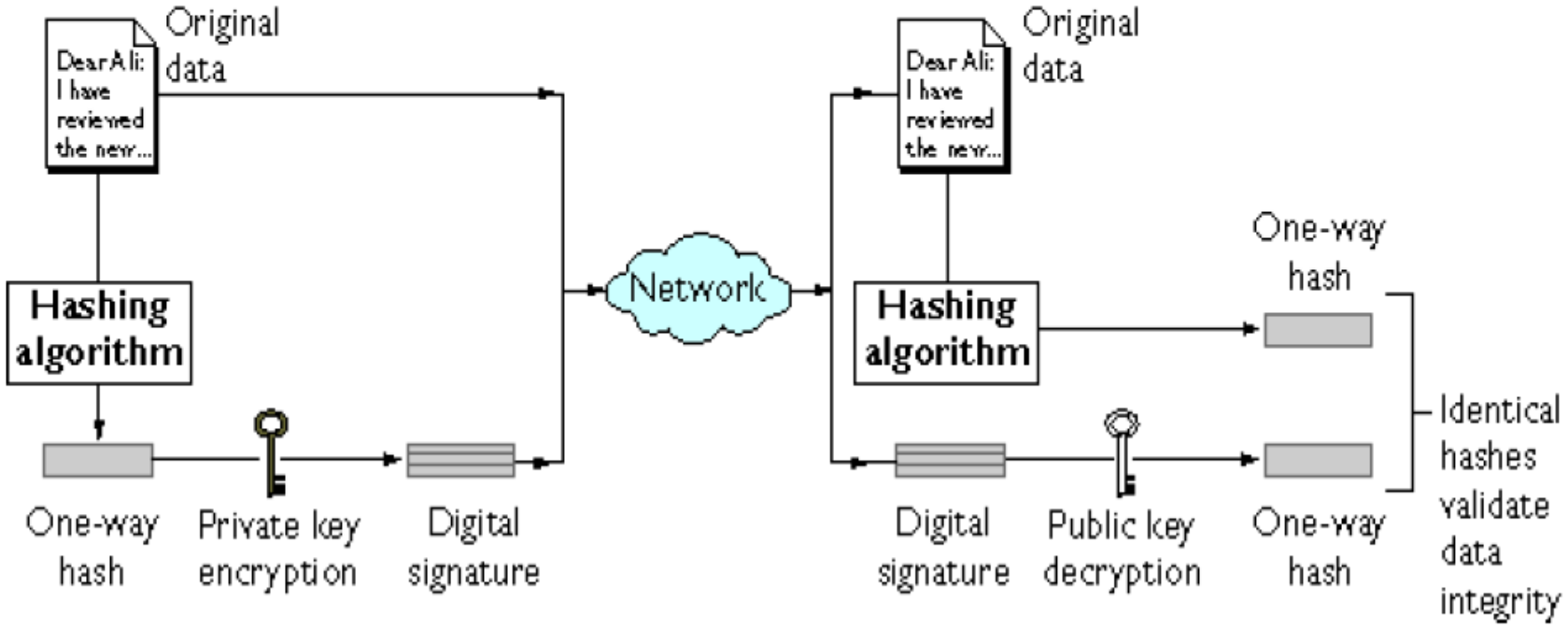
Digital Signatures with Hash Functions and Public Key Encryption



Digital Signature Generation and Verification



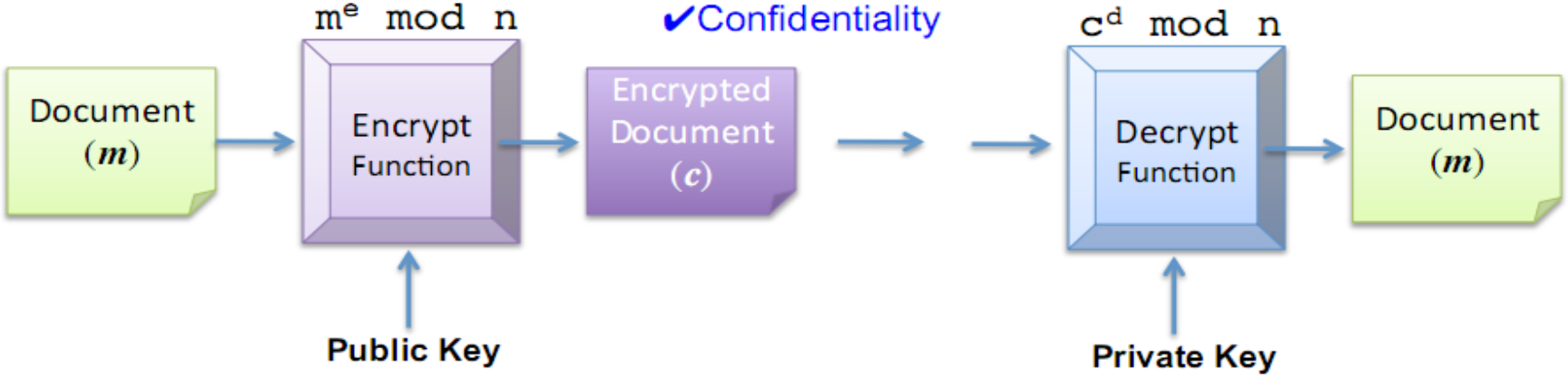
Steps in making a digital signature



RSA Digital Signature

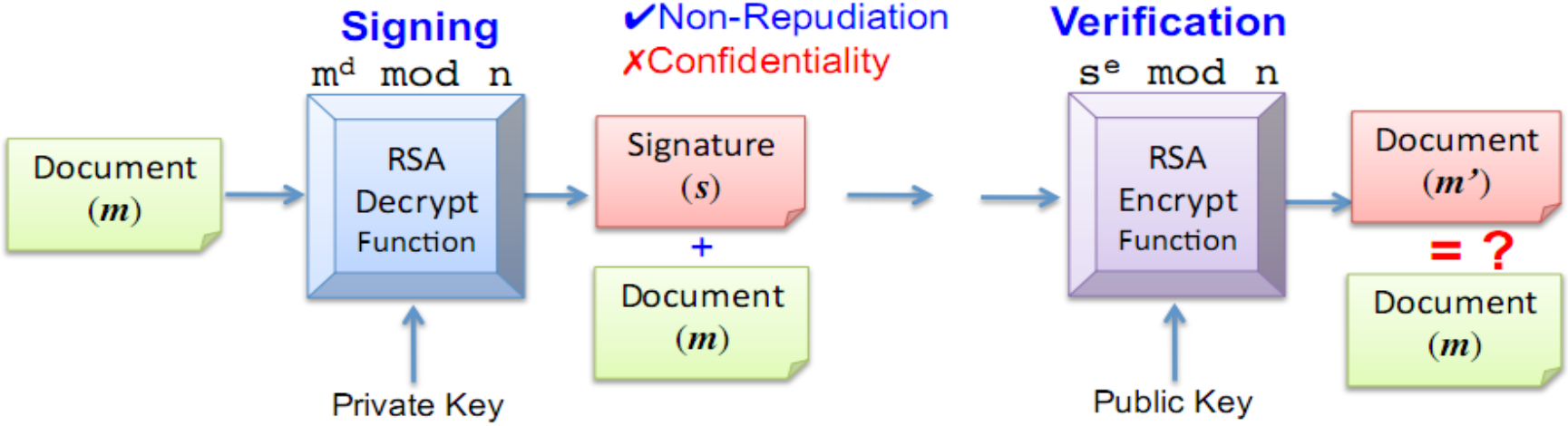
Encryption:

✗ Authenticity
✗ Integrity
✔Confidentiality



Digital Signature:

✔Authenticity
✔Integrity
✔Non-Repudiation
✗Confidentiality



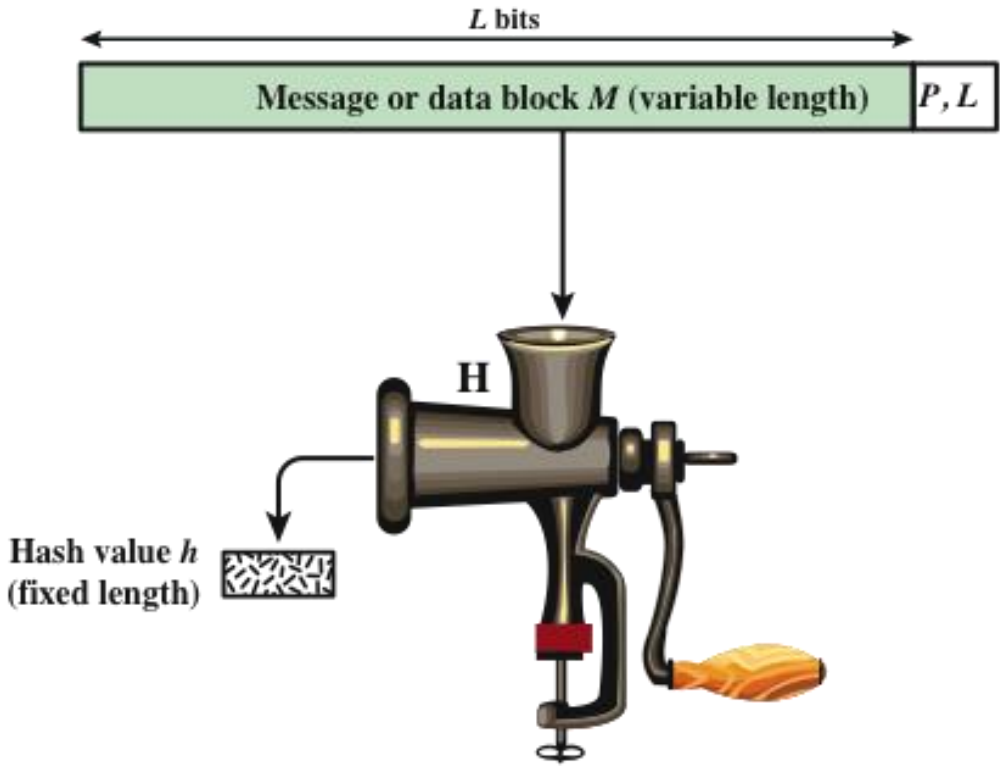
Hash Functions

- A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value
  - $h = H(M)$
  - Principal object is data integrity
- Cryptographic hash function
  - An algorithm for which it is computationally infeasible to find either:

    (a) a data object that maps to a pre-specified hash result (the one-way property)

    (b) two data objects that map to the same hash result (the collision-free property)

Figure 11.1 Cryptographic Hash Function h = uppercase h left parenthesis m right parenthesis



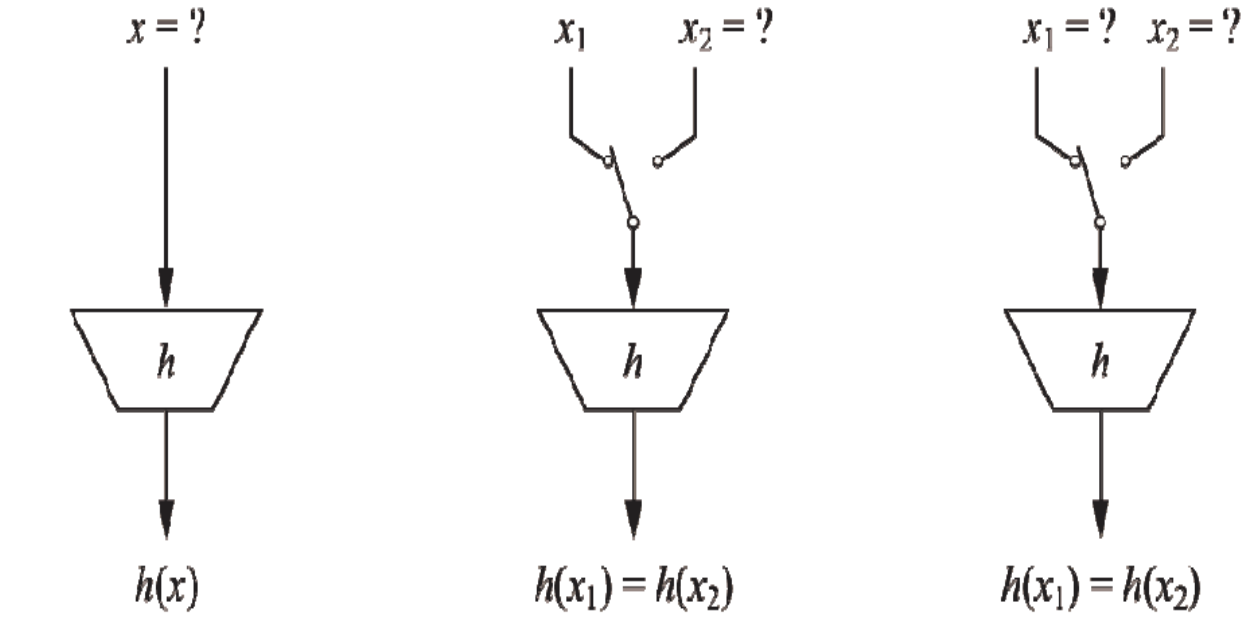
P, L = padding plus length field

Properties of CHF

- Easy to compute the ***digest***
- Infeasible to modify a message without changing the ***digest***
- Preimage resistance: Infeasible to generate a message from a given ***digest***
- Second preimage resistance: Given an input m_1 to the hash function it is infeasible to find another different input m_2 with the same ***digest***
- Collision resistance: It is infeasible to find any two different inputs to the hash function (m_1 and m_2)with the same *digest*

Merkle–Damgård construction method can be used to build collision-resistant cryptographic hash functions
(Uses collision-resistant one-way compression functions)

- The three security properties of hash functions



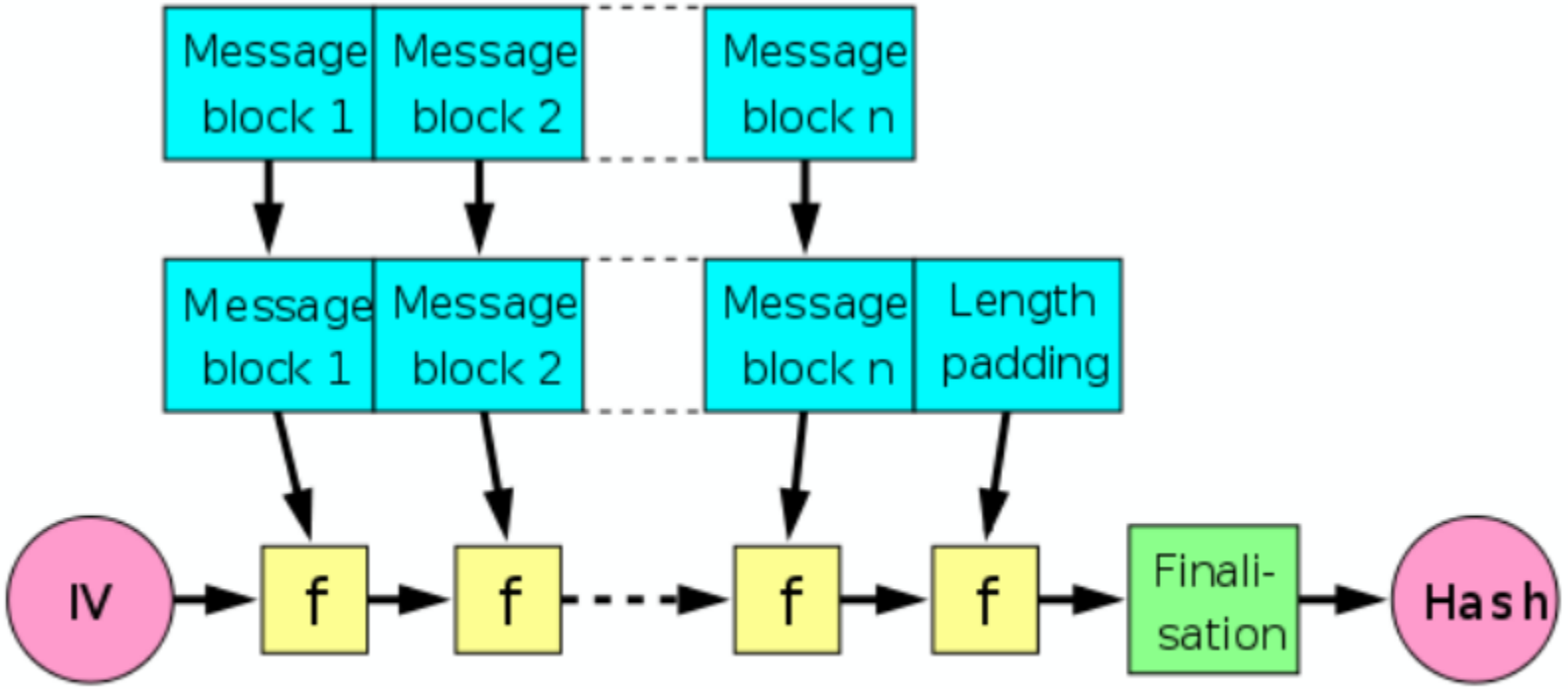
preimage resistance

second preimage resistance

collision resistance

Merkle–Damgård construction

- Theorem: Any compression function f which is collision resistant can be extended to a collision resistant hash function h (taking arbitrary length inputs).



Compression Function $f: \{0,1\}^{n+b} \rightarrow \{0,1\}^n$

Merkle–Damgård construction

- Construct a Cryptographic Hash Function

  $h:\{0,1\}^* \rightarrow \{0,1\}^n$

  from a compression function

  $f:\{0,1\}^{n+b} \rightarrow \{0,1\}^n$

- For $m \in \{0,1\}^*$ add padding to m so that the padded message length $|m'|$ is a multiple of b
- Let padded message $m' = m_1 m_2 m_3 \ldots m_k$ with each m_i of length b
  - Padding = 10...0$|m|$, where $|m|$ is the length of m
- Let $v_0 = IV$ and $v_i = f(v_{i-1} \,||\, m_i)$ for $1 \le i \le k$
- The hash value $h(m) = v_k$

Secure Hash Algorithm (SHA)

- SHA was originally designed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993
- Was revised in 1995 as SHA-1
- Based on the hash function MD4 and its design closely models MD4
- Produces 160-bit hash values
- In 2002 NIST produced a revised version of the standard that defined three new versions of SHA with hash value lengths of 256, 384, and 512
  - Collectively known as SHA-2

Table 11.3 Comparison of SHA Parameters

Algorithm	Message Size	Block Size	Word Size	Message Digest Size
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

Note: All sizes are measured in bits.

Figure 11.9 Message Digest Generation Using SHA-512

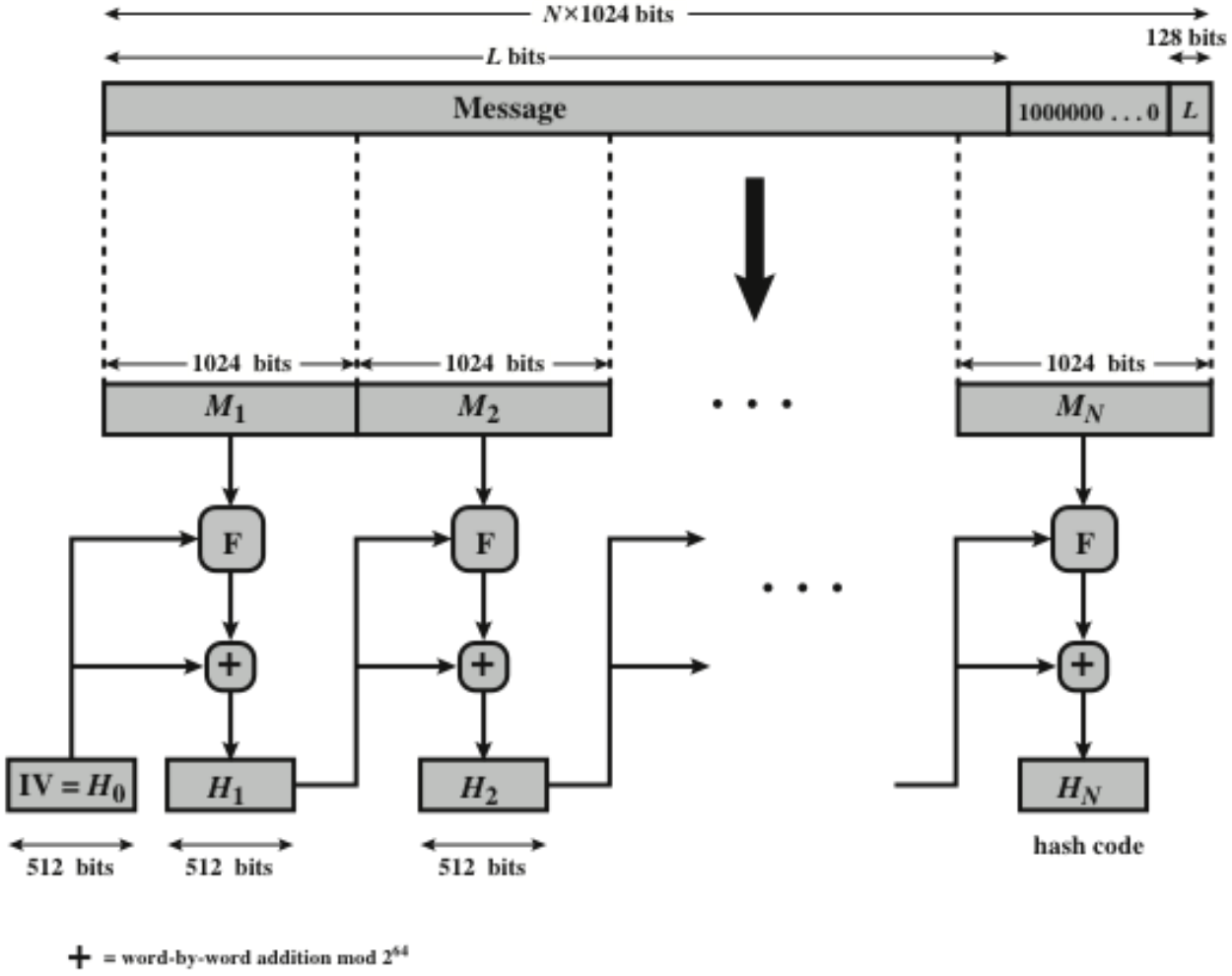
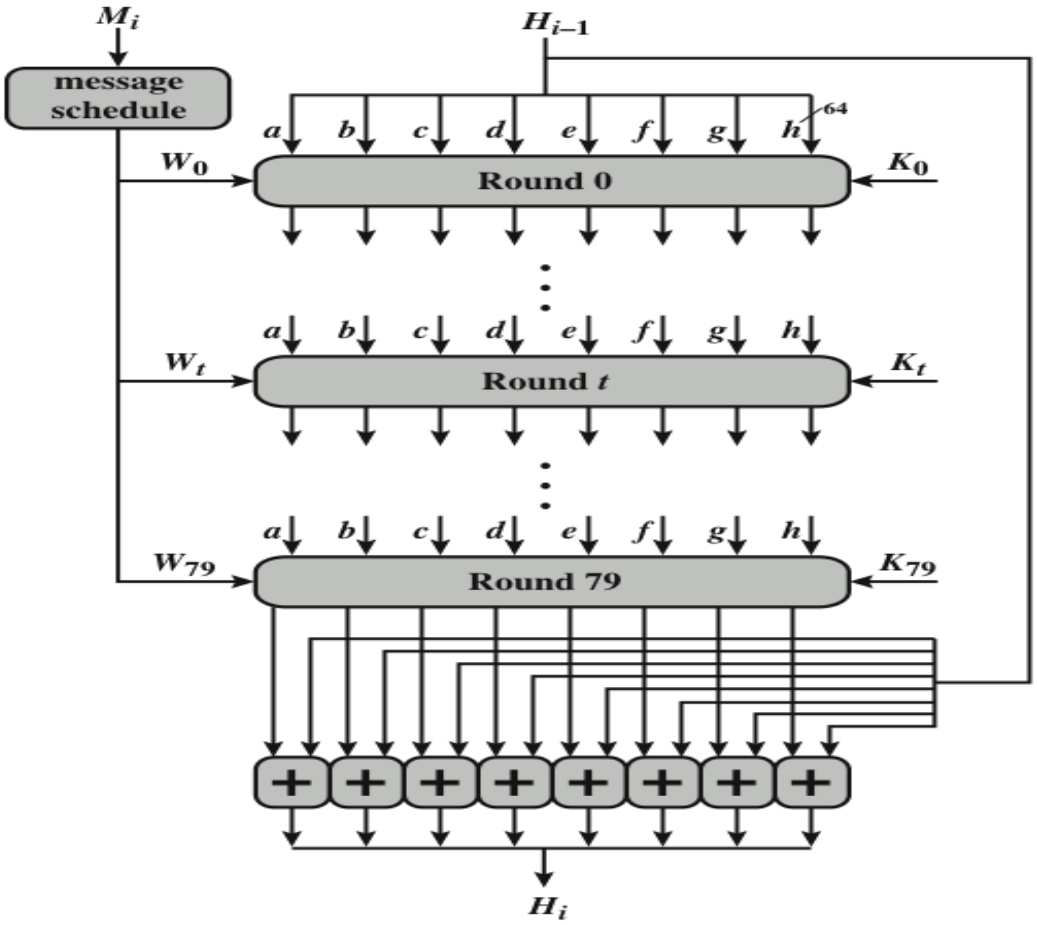
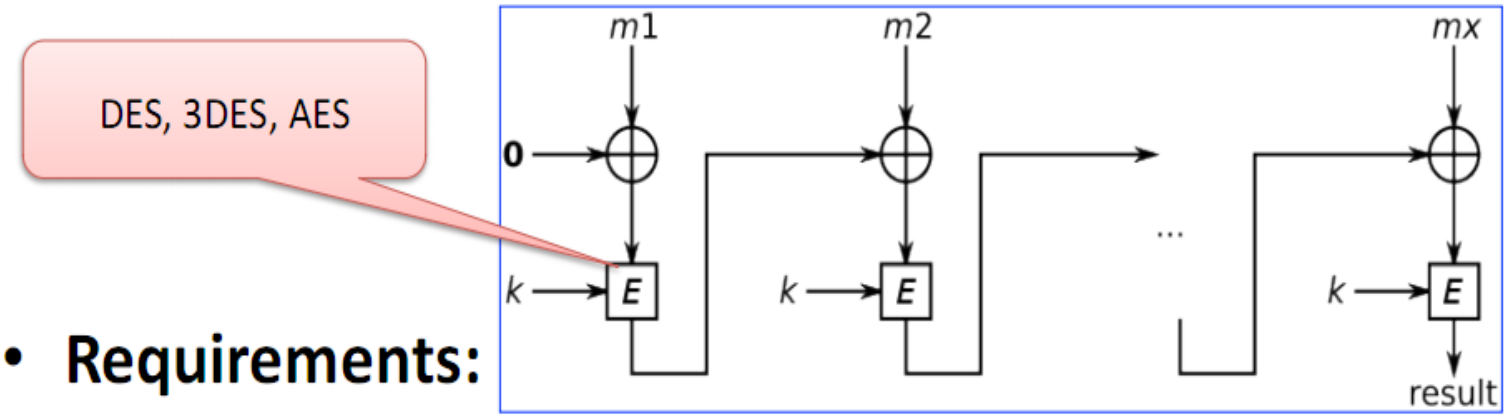


Figure 11.10 SHA-512 Processing of a Single 1024-Bit Block



MAC

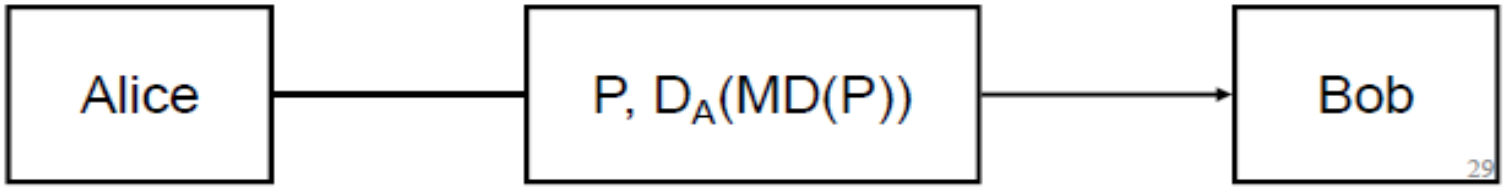
- A MAC is a cryptographic checksum $MAC=C_K(M)$
- Any block cipher chaining mode can be used to generate MAC by making the final block as a MAC



- **Requirements:**
  - Potentially many messages have same MAC but finding these is computationally infeasible
  - MAC's should be uniformly distributed (resistant against against brute force attack)
  - MAC should depend equally on all bits of the message

Why not base MACs on Encryption

- Message authentication codes based on encryption functions are a bad idea because:
  - Inefficiency of encryption algorithm in software
  - US export restrictions
- Solution
  - Use Message Digests or Use a one-way hash function to create a fixed size finger print of the variable sized message.



Hashed MAC (HMAC)

- Any cryptographic hash function, such as MD5 or SHA-1, can be used in the calculation of an HMAC; the resulting MAC algorithm is termed **HMAC-MD5** or **HMAC-SHA1** accordingly.
- The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output length in bits, and on the size and quality of the cryptographic key.
- HMAC-SHA-1 and HMAC-MD5 are used within the IPSec and TLS protocols.

HMAC Description

- $H(\cdot)$ be a cryptographic hash function (SHA1 or MD5)
- K be a secret key
- m be the message to be authenticated
- $\|$ denote concatenation
- $\oplus$ denote exclusive or (XOR)
- opad → 00110110 outer padding (0x5c..., one-block-long hexadecimal constant)
- ipad → 01011100 inner padding (0x36..., one-block-long hexadecimal constant)

Both ipad and opad flip half of the bits

HMAC(K,m) is mathematically defined as

$$H((K \oplus opad) \| H((K \oplus ipad) \| m))$$