Asymmetric Key Encryption

- Important to know who should know which key(s)

- In general:
  - Sender encrypts with recipient’s public key
  - Recipient decrypts with its private key

Matrix of Keys

Key details	***A* should know**	***B* should know**
A's private key	Yes	No
A's public key	Yes	Yes
B's private key	No	Yes
B's public key	Yes	Yes

Fig 4.1

Asymmetric Key Cryptography

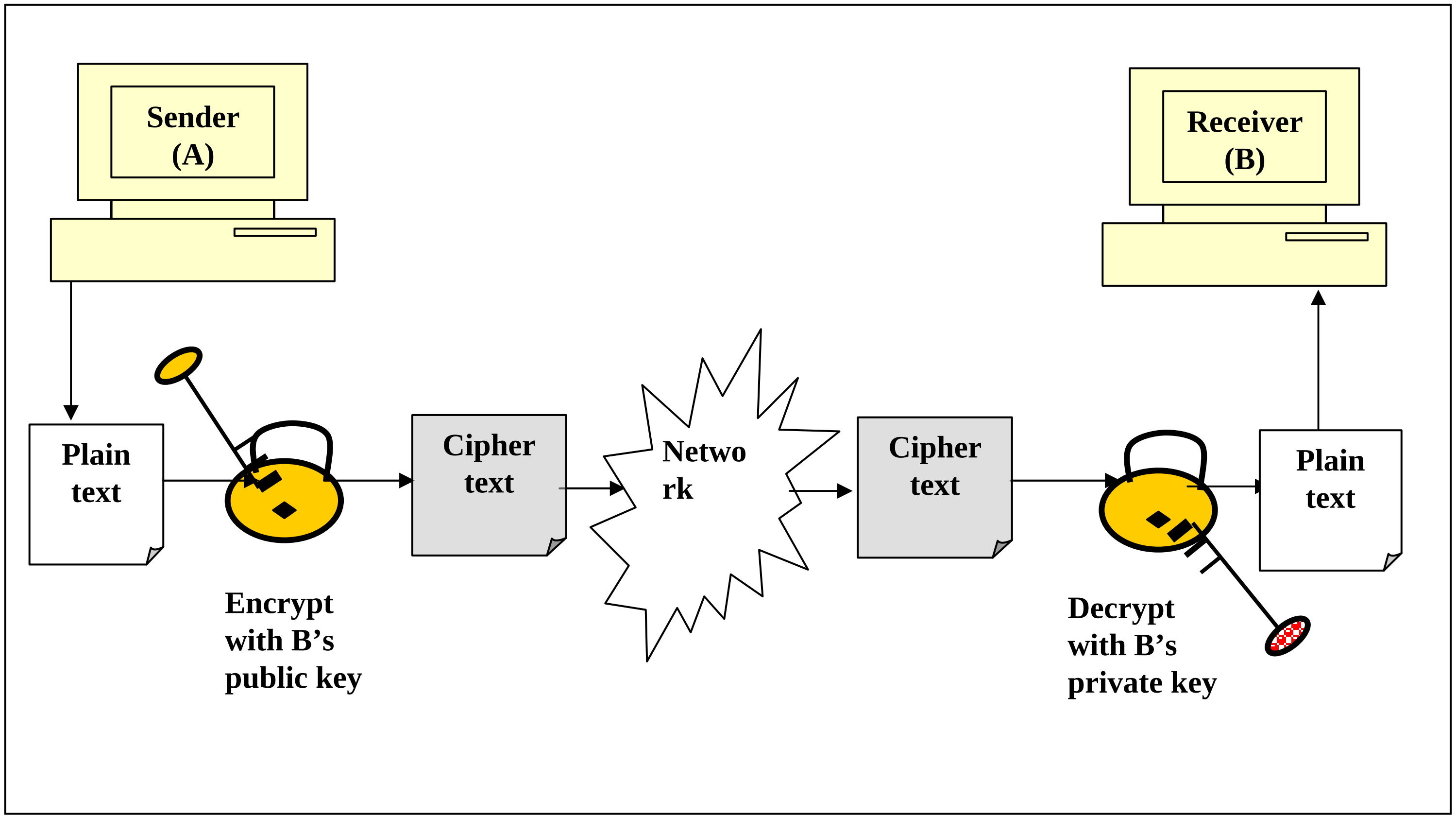

Fig 4.2

Asymmetric Key Example

- Consider a bank and its customers
- Customers encrypt their messages with bank’s public key
- Bank decrypts messages with its private key

Asymmetric Key Cryptography Example

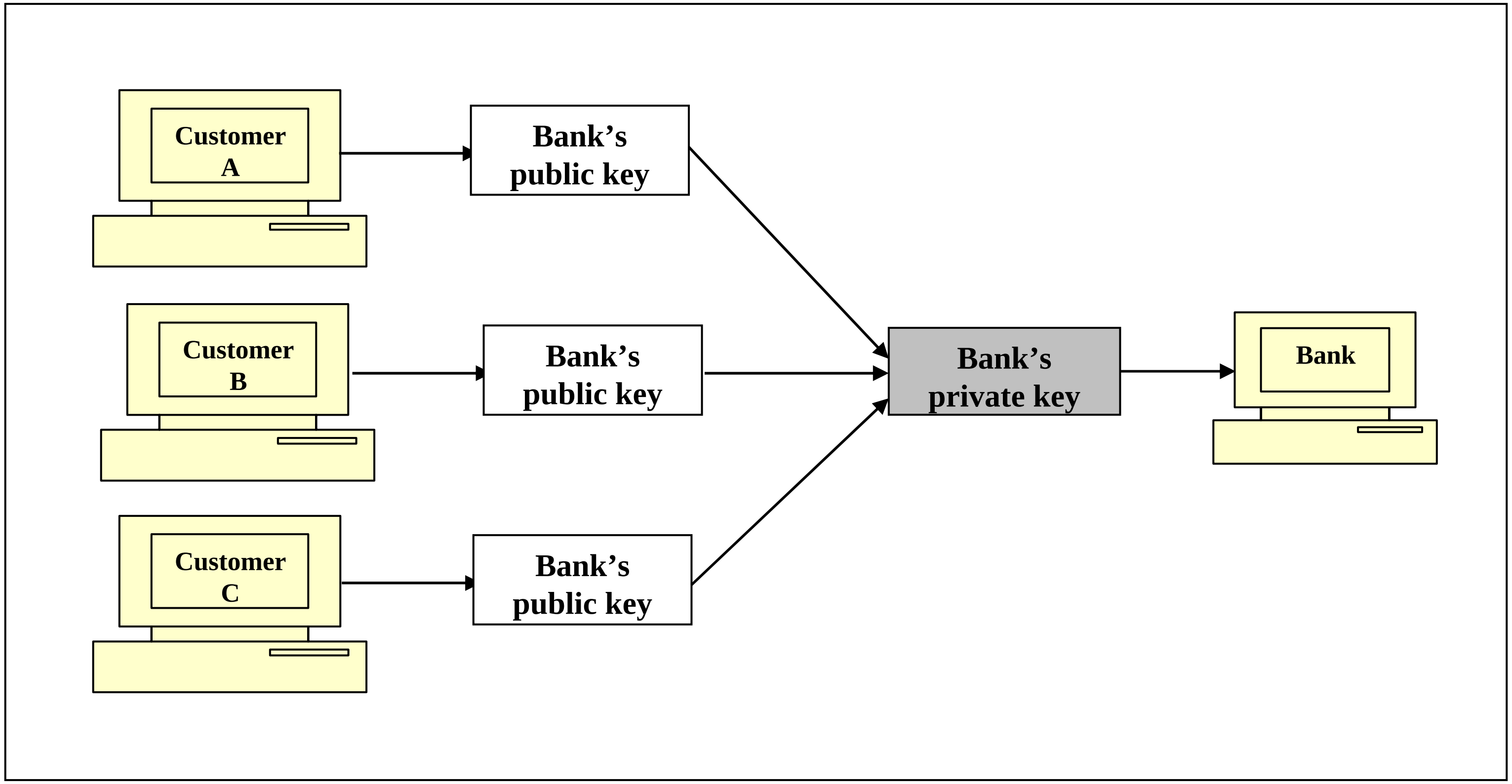

Fig 4.3

RSA Algorithm

- World’s most popular Asymmetric Key Encryption algorithm
- Based on the theory of Prime Numbers
- Example Follows

RSA Algorithm

1. Choose two large prime numbers P and Q.

2. Calculate N = P x Q.

3. Select the public key (i.e. the encryption key) E such that it is not a factor of (P – 1) and (Q – 1).

4. Select the private key (i.e. the decryption key) D such that the following equation is true:
(D x E) mod (P – 1) x (Q – 1) = 1

5. For encryption, calculate the cipher text CT from the plain text PT as follows:
$CT = PT^E \bmod N$

6. Send CT as the cipher text to the receiver.

7. For decryption, calculate the plain text PT from the cipher text CT as follows:
$PT = CT^D \bmod N$

Fig 4.4

Example of RSA Algorithm

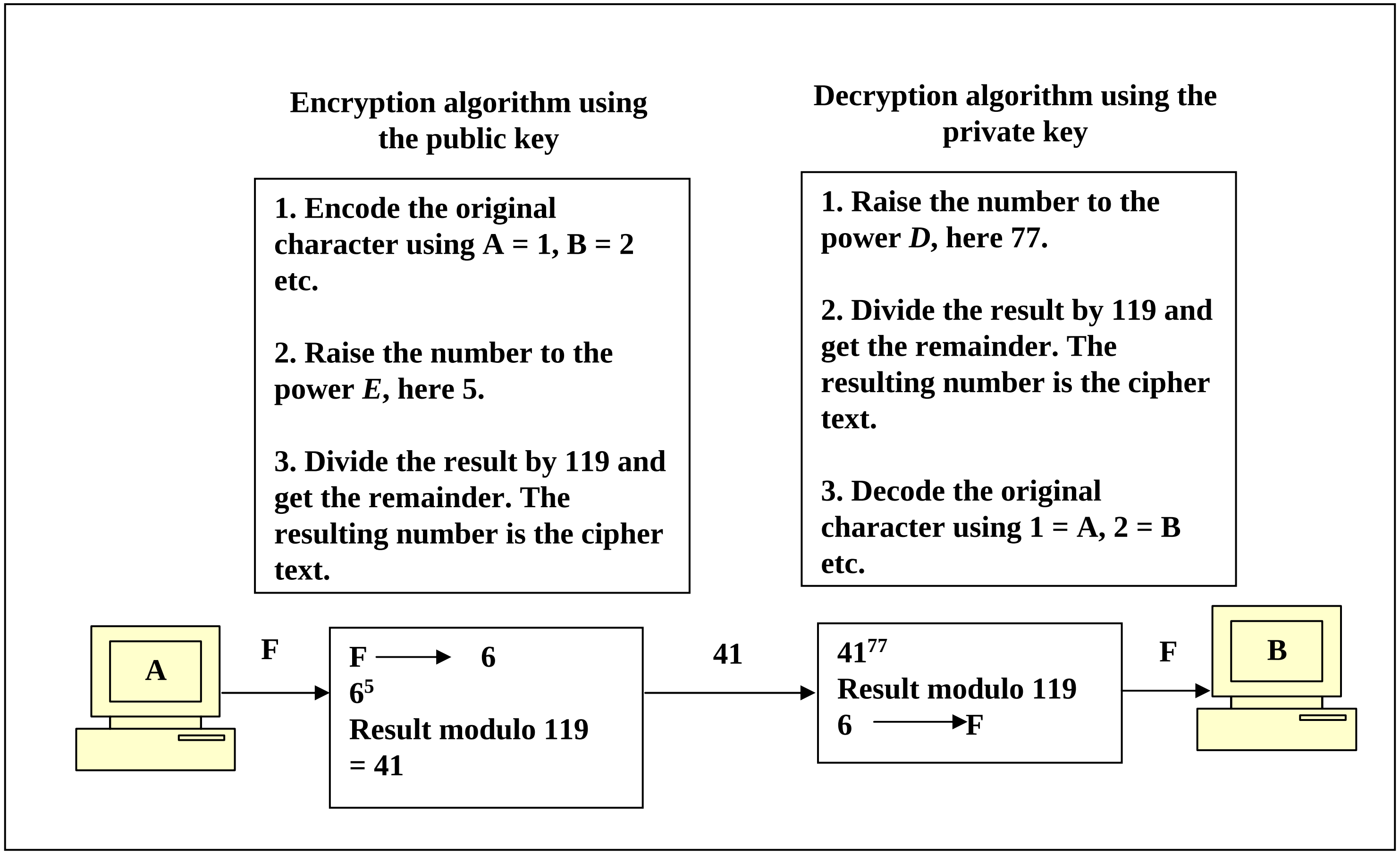

Fig 4.6

Symmetric v/s Asymmetric

Characteristic	**Symmetric Key Cryptography**	**Asymmetric Key Cryptography**
Key used for encryption / decryption	Same key is used for encryption and decryption	One key used for encryption and another, different key is used for decryption
Speed of encryption / decryption	Very fast	Slower
Size of resulting encrypted text	Usually same as or less than the original clear text size	More than the original clear text size
Key agreement / exchange	A big problem	No problem at all
Number of keys required as compared to the number of participants in the message exchange	Equals about the square of the number of participants, so scalability is an issue	Same as the number of participants, so scales up quite well
Usage	Mainly for encryption and decryption (confidentiality), cannot be used for digital signatures (integrity and non-repudiation checks)	Can be used for encryption and decryption (confidentiality) as well as for digital signatures (integrity and non-repudiation checks)

Fig 4.7

Digital Signature Concept

- Sender encrypts message or its fingerprint with its private key
- Guarantees that only the sender could have created this message
- Basis for Non-repudiation

Basis for Digital Signatures

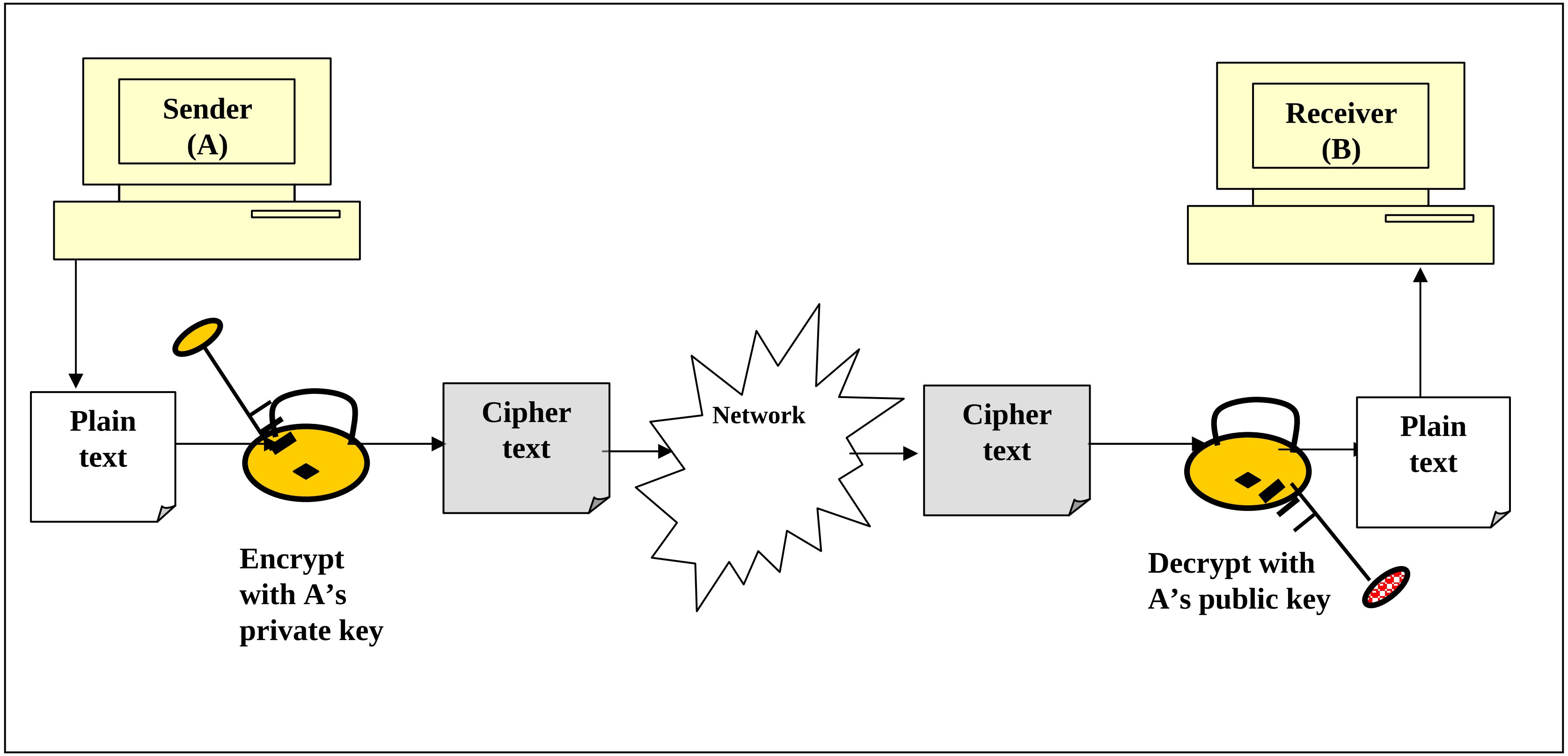

Fig 4.16

Message Digest Concept

- Also called as *Hash*
- Unique representation of a message
- Similar to finger print of a human

Message Digest Idea

- **Original number is 7391743**

Operation	**Result**
Multiply 7 by 3	21
Discard first digit	1
Multiply 1 by 9	9
Multiply 9 by 1	9
Multiply 9 by 7	63
Discard first digit	3
Multiply 3 by 4	12
Discard first digit	2
Multiply 2 by 3	6

- **Message digest is 6**

Fig 4.18

Message Digest Concept

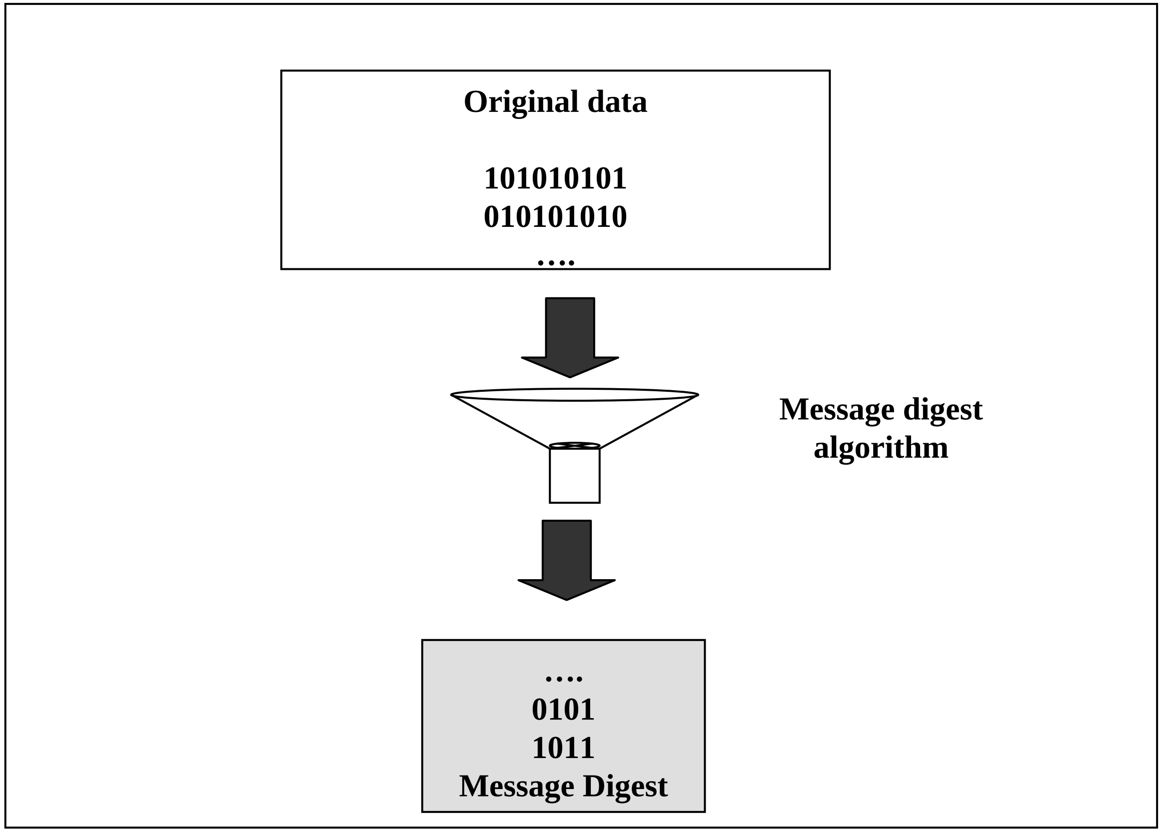

Fig 4.19

Message Digest Demands - 1

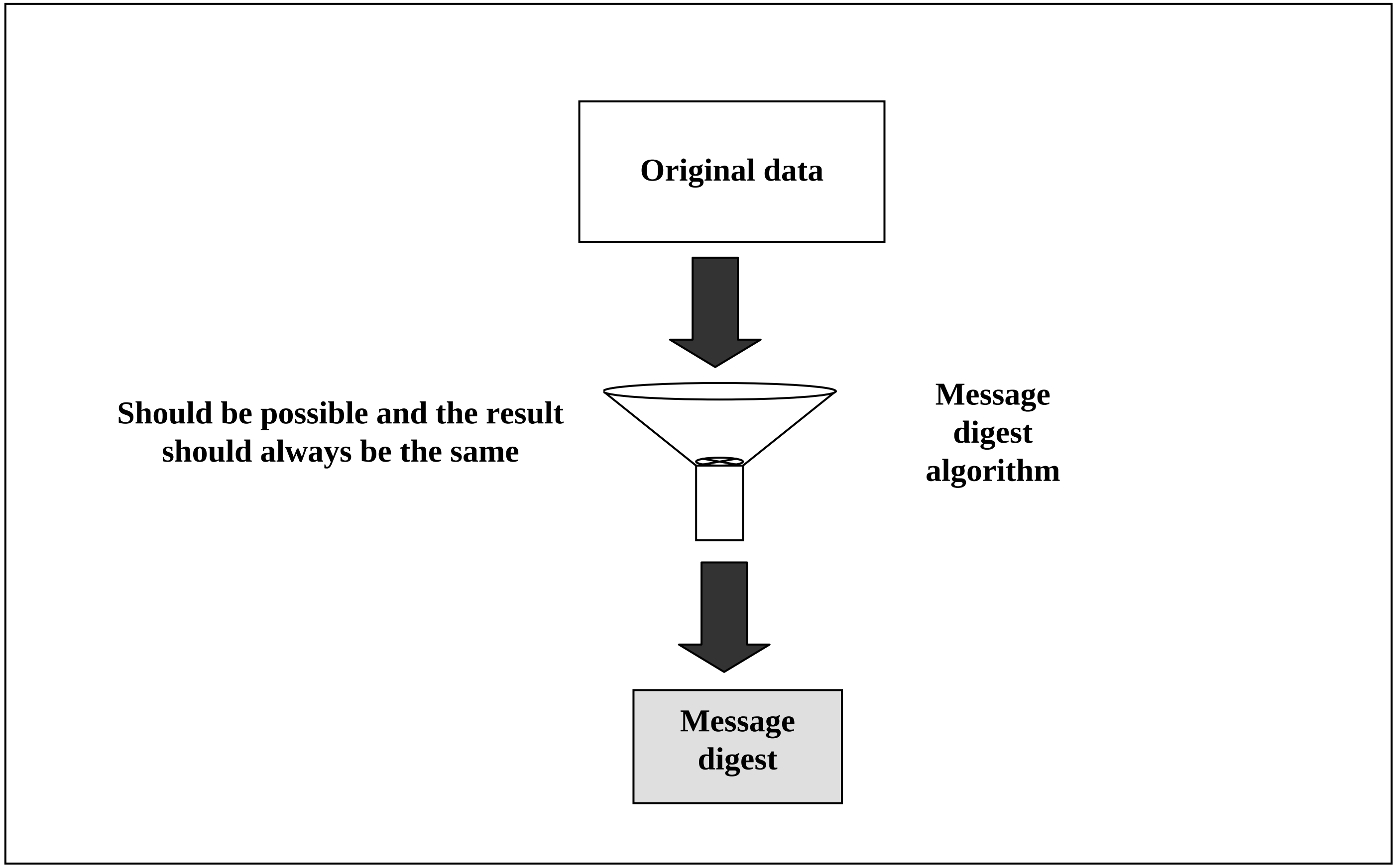

Fig 4.20

Message Digest Demands - 2

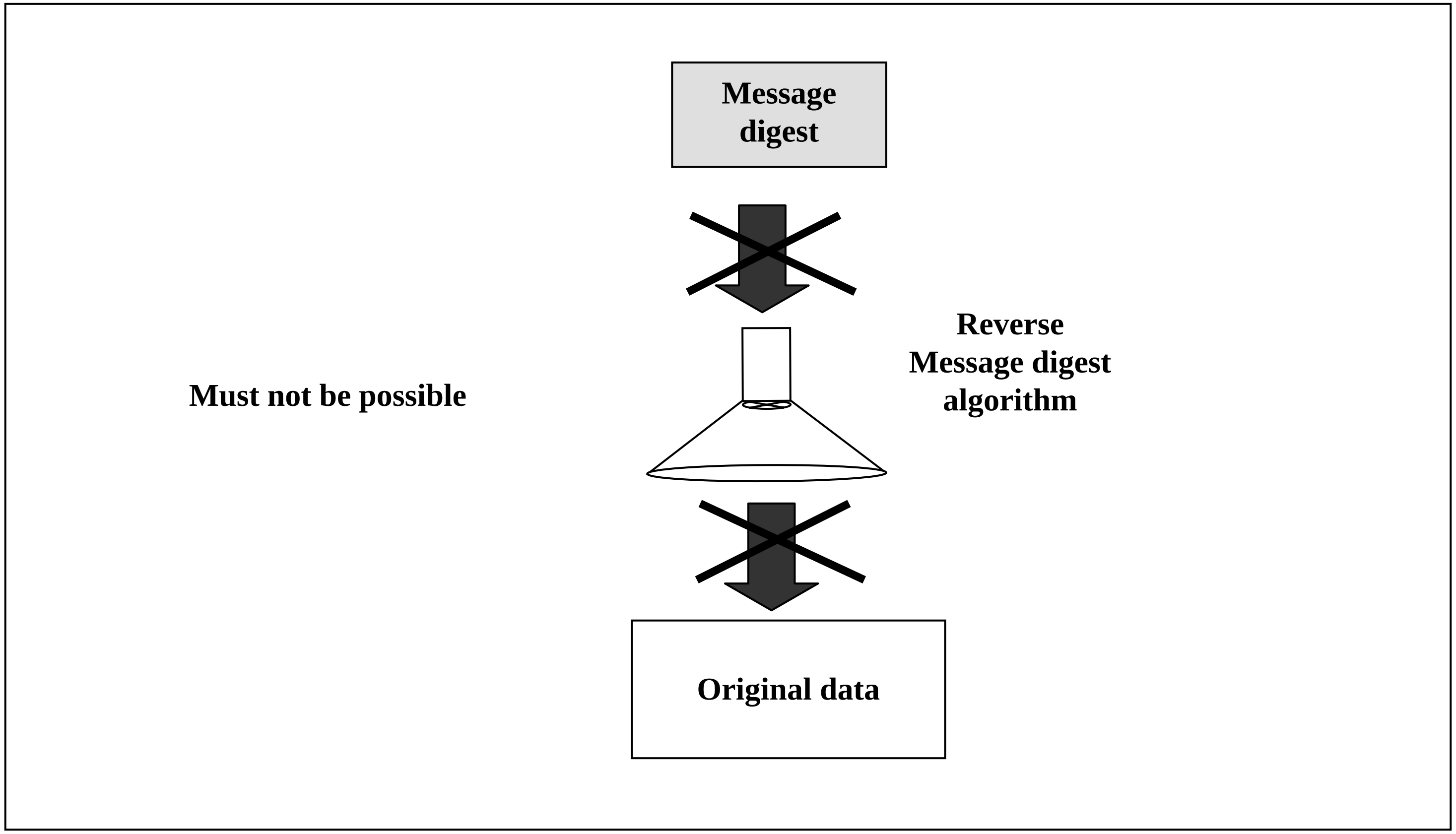

Fig 4.21

Message Digest Demands - 3

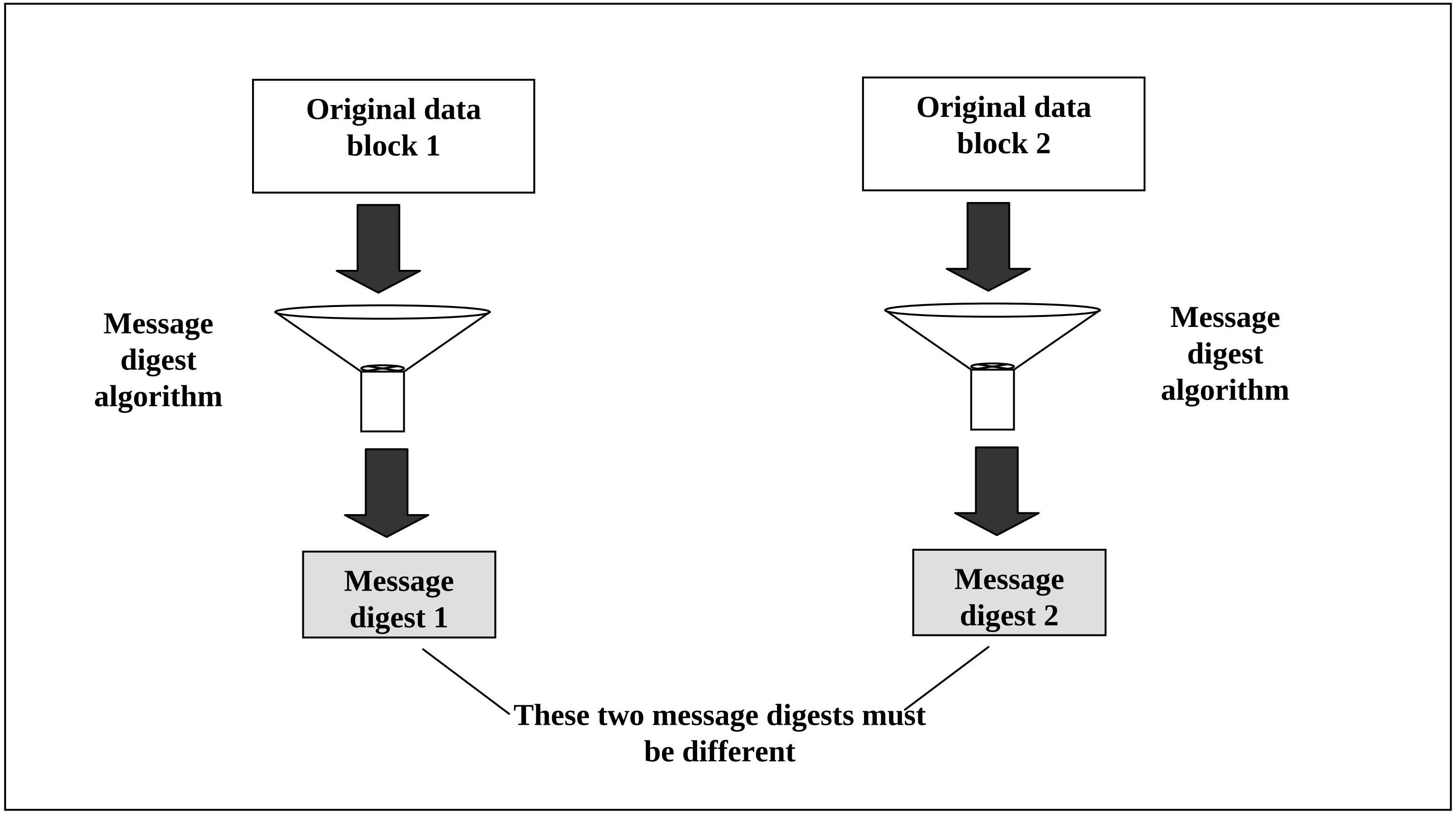

Fig 4.22

Message Digest Differences

- Even if the original messages differ minutely, message digests differ dramatically

- Basis for the guarantee of uniqueness

Message Digest Example

Message

Please pay the newspaper bill today

Message digest

306706092A864886F70D010705A05A305802010030090605 2B0E03021A050

0303206092A864886F70D010701A0250423506C656173652070617920746865206E65777370617065722062696C6C20746F646179041479630AC8041BAA1C40747F2FC29D881AEF92299B

Message

Please pay the newspaper bill tomorrow

Message digest

306A06092A864886F70D010705A05D305B020100300906052B0E03021A0500303506092A864886F70D010701A0280426506C6561736520706179207468 65206E65777370617065722062696C6C20746F6D6F72726F7704146EEC2E0DB9570A5AF6CEB631CE057AE830A87C5B

Fig 4.23

Message Digest Algorithms

- Basic principle: Take the original message, and reduce it to a smaller *fingerprint*
- Examples: MD5, SHA-1
- SHA-1 is considered stronger

One MD5 Operation

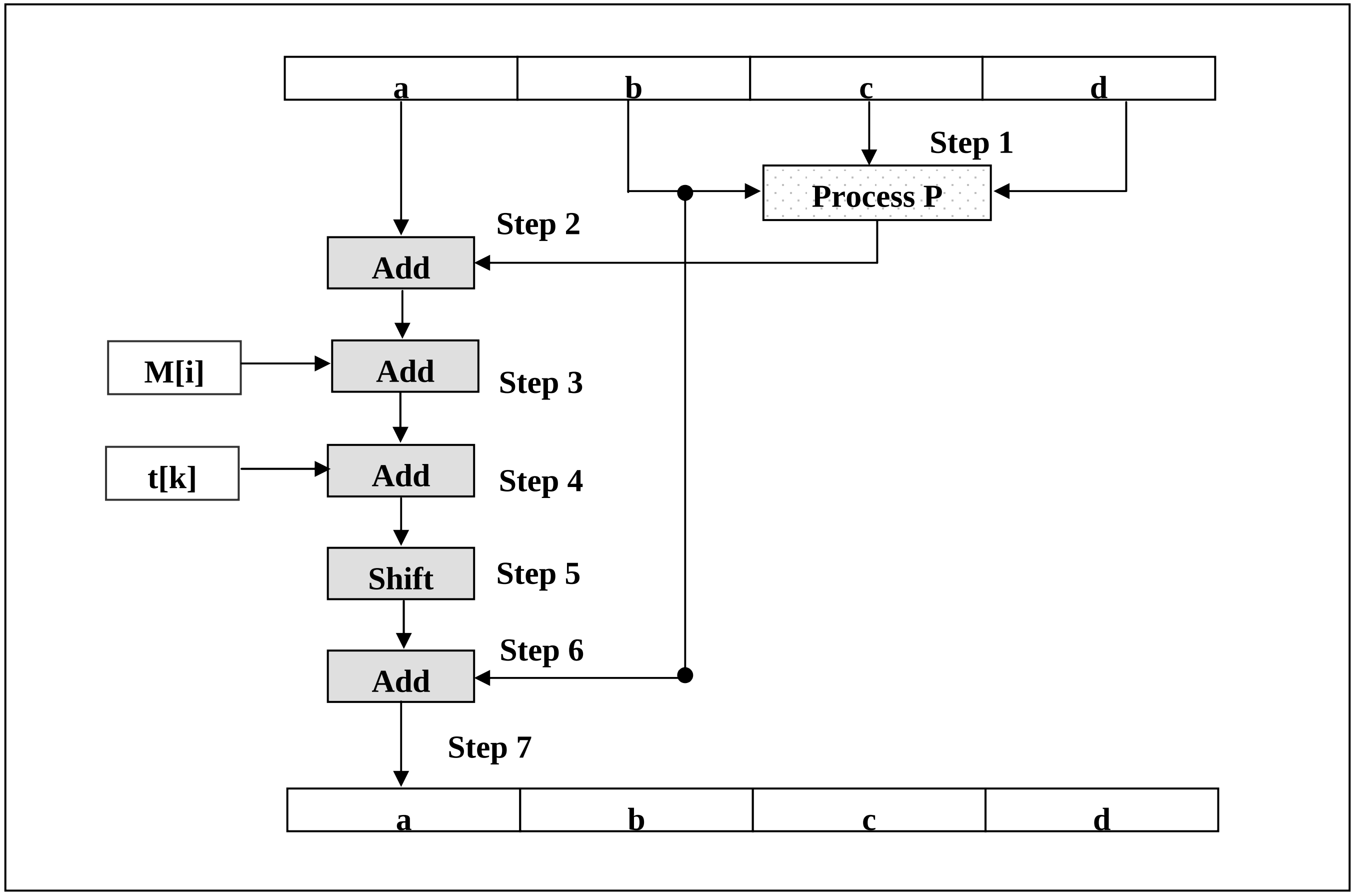

Fig 4.33

Single SHA-1 Iteration

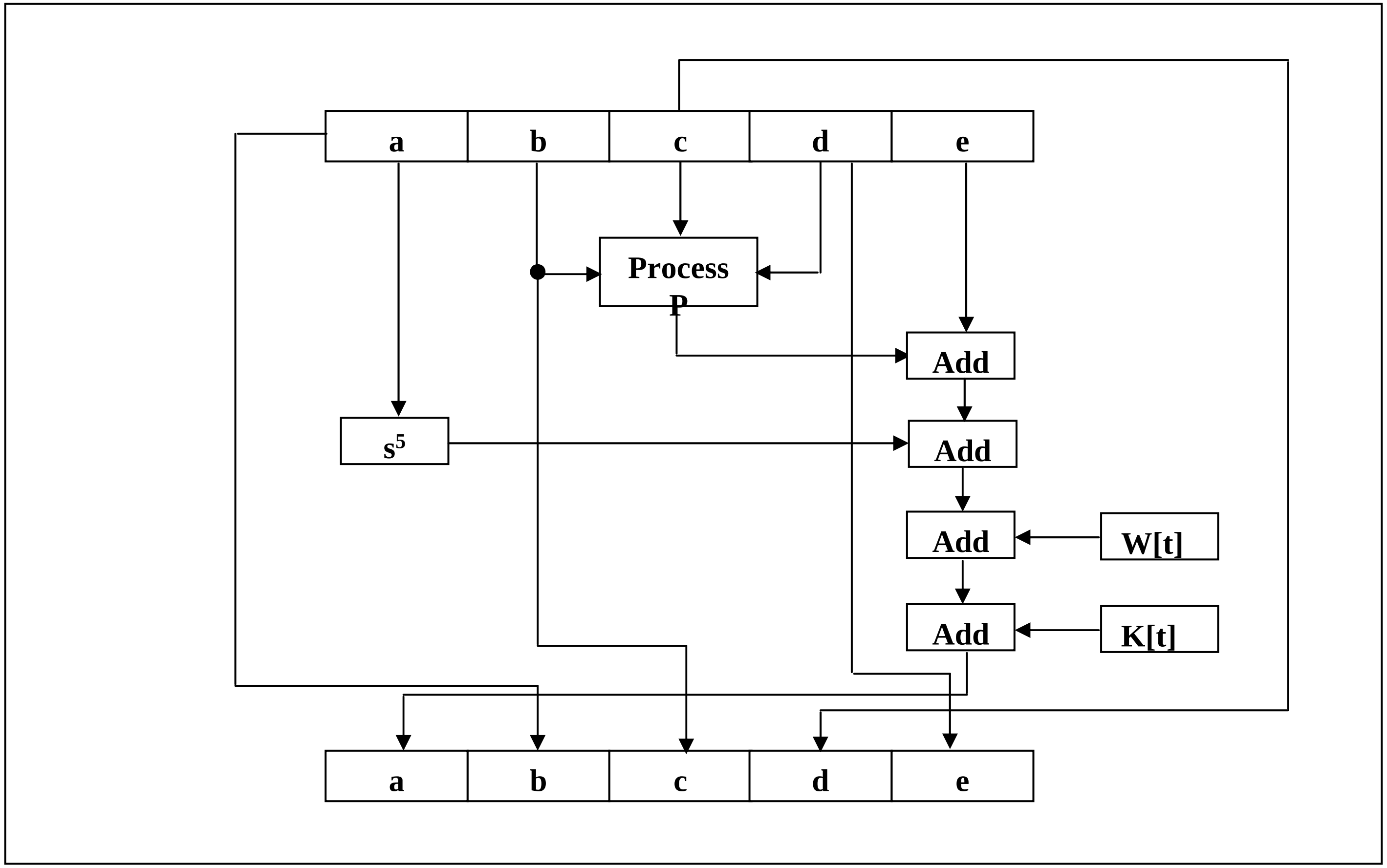

Fig 4.39

Comparison of MD5 and SHA-1

Point of discussion	**MD5**	**SHA**
Message digest length in bits	128	160
Attack to try and find the original message given a message digest	Requires 2^{128} operations to break in	Requires 2^{160} operations to break in, therefore more secure
Attack to try and find two messages producing the same message digest	Requires 2^{64} operations to break in	Requires 2^{80} operations to break in
Successful attacks so far	There have been reported attempts to some extent (as we discussed earlier)	No such claims so far
Speed	Faster (64 iterations, and 128-bit buffer)	Slower (80 iterations, and 160-bit buffer)
Software implementation	Simple, does not need any large programs or complex tables	Simple, does not need any large programs or complex tables

Fig 4.42

Message Authentication Code (MAC)

- Similar to message digest
- In addition, also involves encryption and decryption
- Sender and receiver must know a shared secret key

Message Authentication Code (MAC)

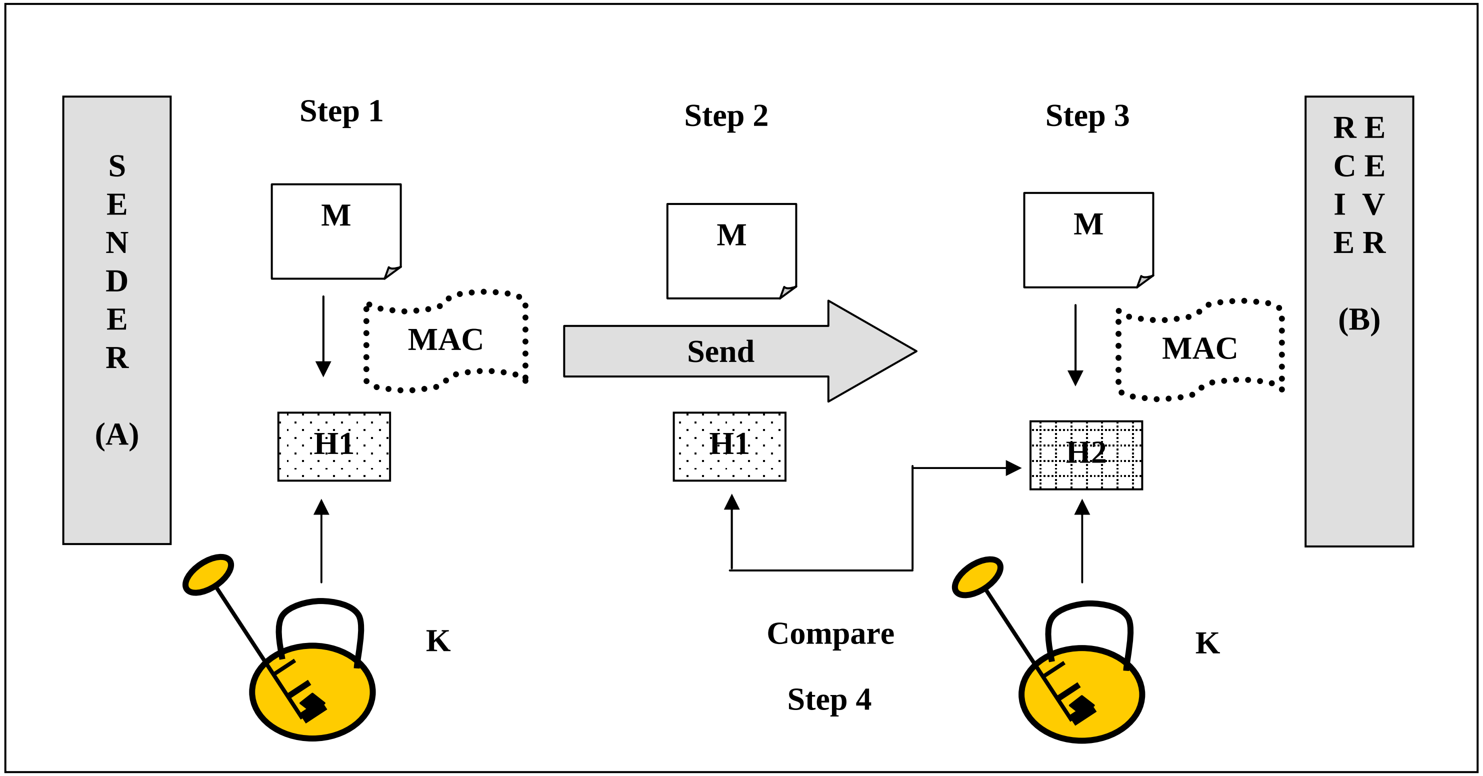

Fig 4.43

HMAC Concept

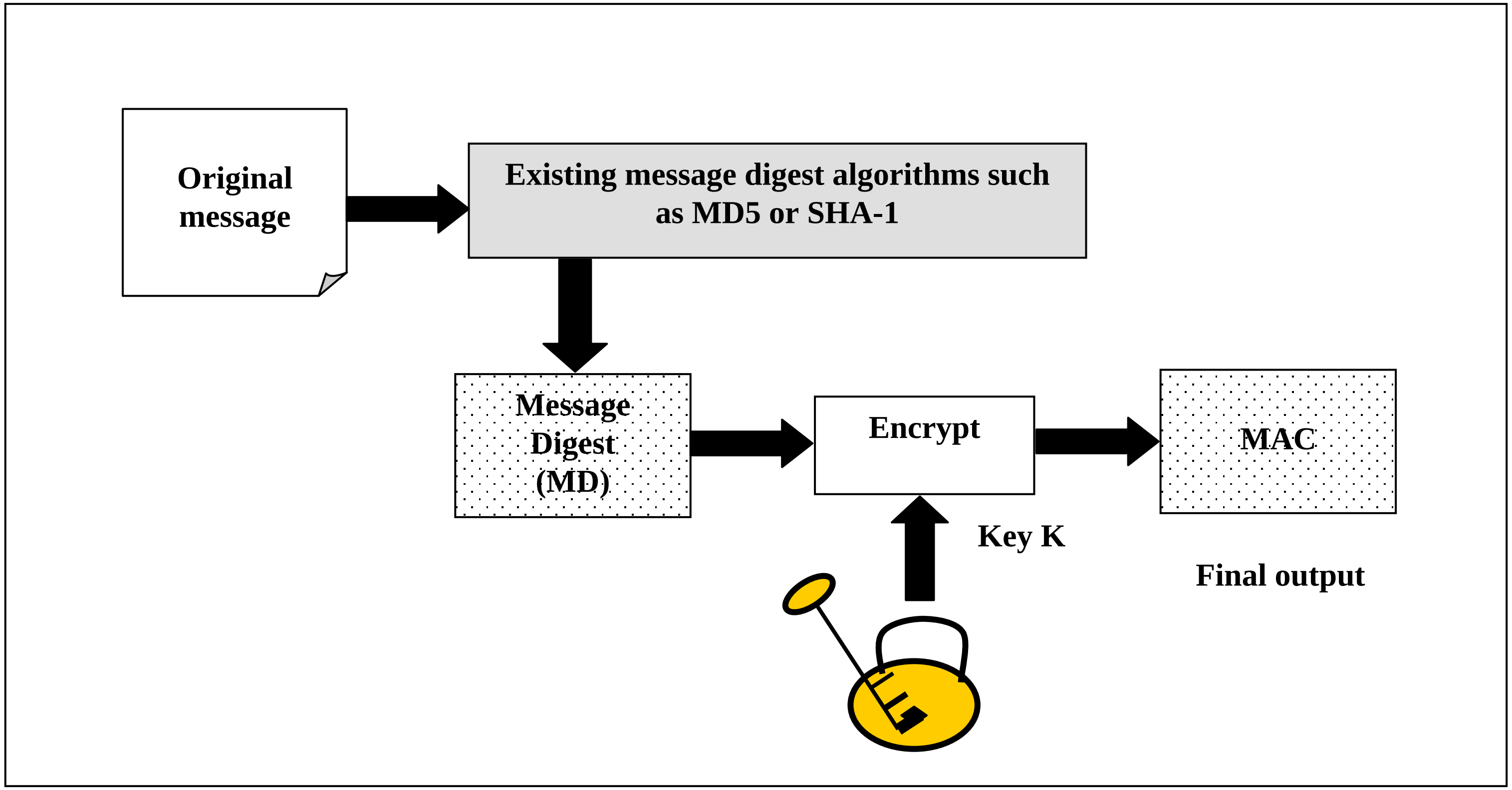

Fig 4.44

Complete HMAC Operation

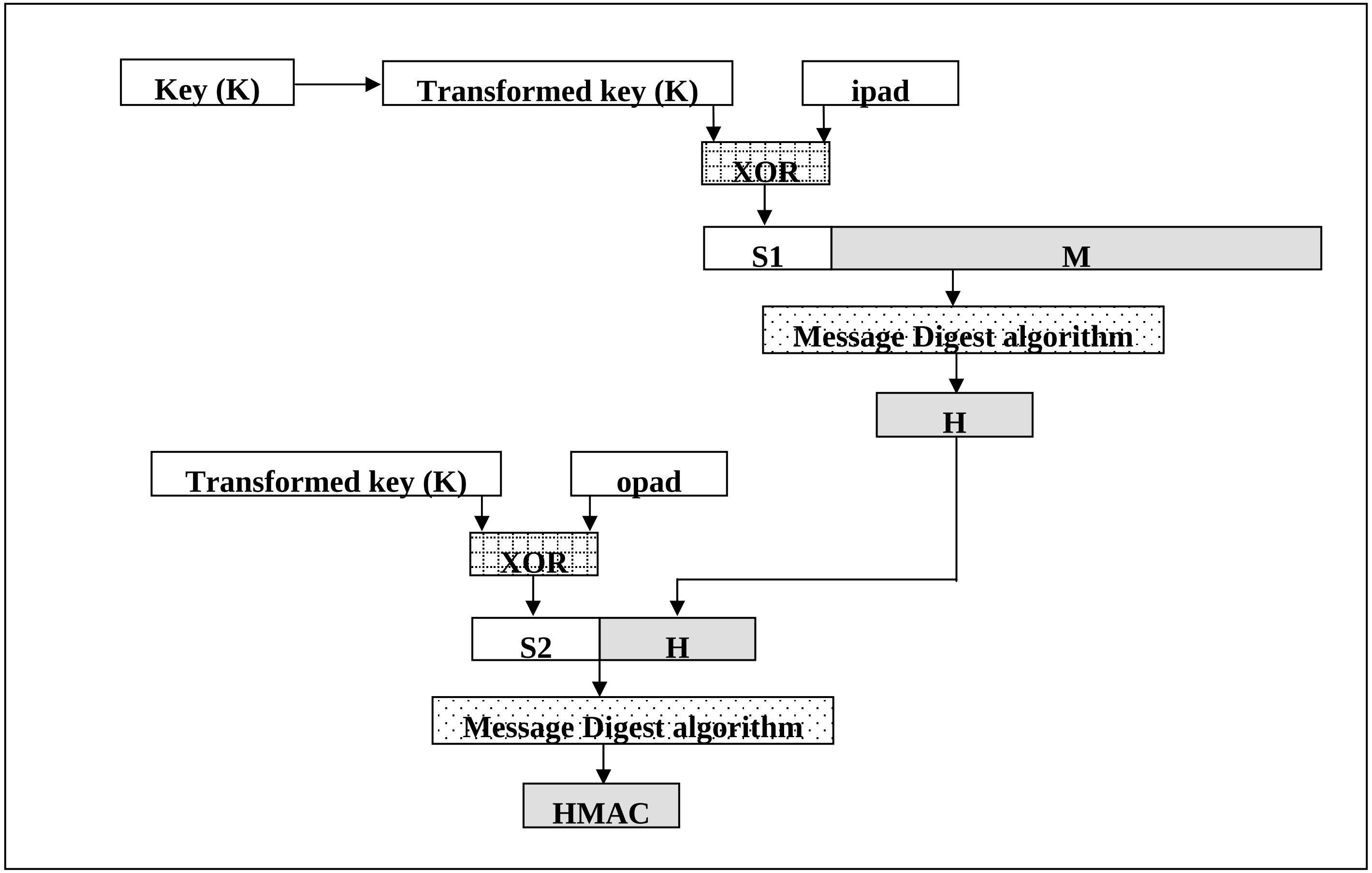

Fig 4.52