

Advanced data Security

Part I:

Cryptography and Network Security

Chapter6:

Cryptographic Hash Functions, MAC and Digital Signature

Prepared by:

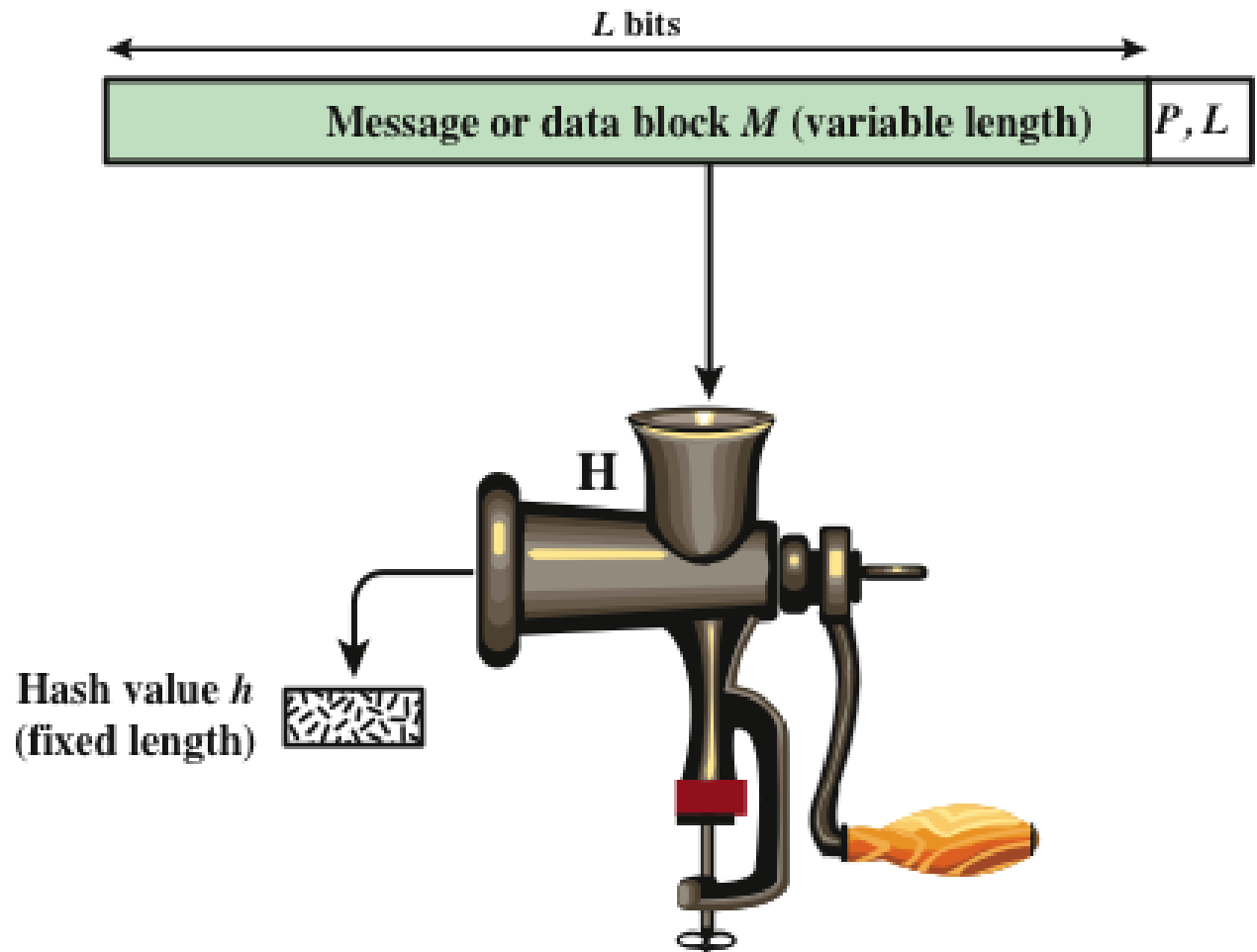
Dr. Hamssa Hasrouny

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)

the **input** is **padding** out to an **integer multiple of some fixed length (e.g., 1024 bits)**, and the padding includes the value of the length of the original message in bits.

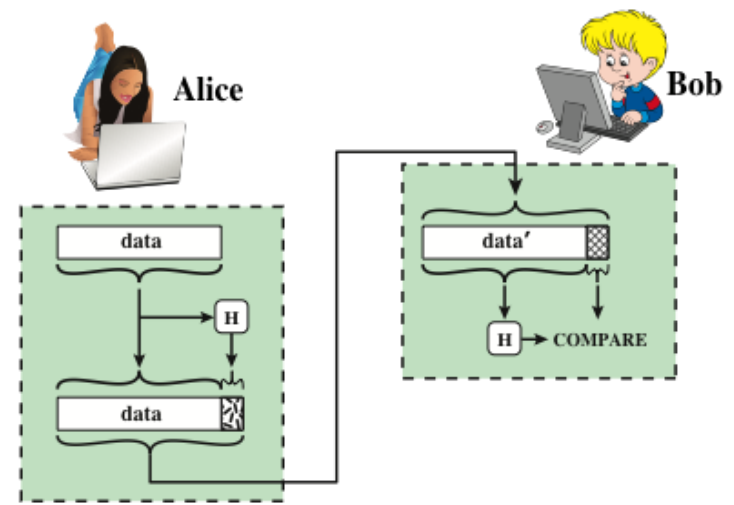
The length field is a security measure to increase the difficulty for an attacker to produce an alternative message with the same hash value.



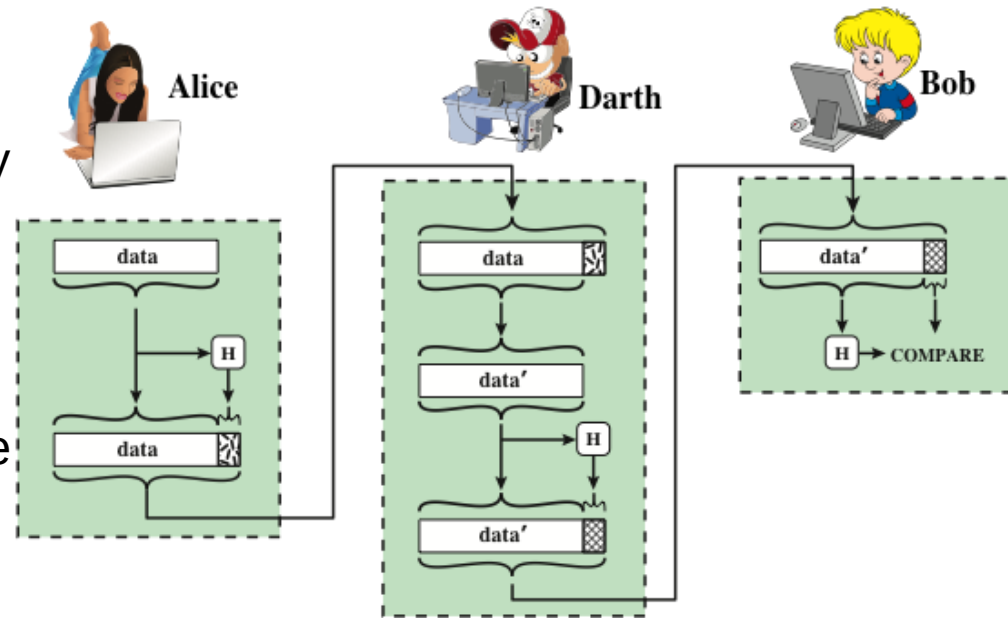
P, L = padding plus length field

Figure 11.1 Cryptographic Hash Function; $h = H(M)$

- Message authentication assures that data received are exactly as sent.
- the hash function value is referred to as a message digest .
- The sender computes a hash value and transmits both the hash value and the message.
- The receiver performs the same hash calculation on the message bits and compares this value with the incoming hash value.
- If there is a mismatch, the receiver knows that the message (or possibly the hash value) has been altered (Figure 11.2a).
- The **hash function must be transmitted in a secure fashion.**
- if an adversary alters or replaces the message, it is not feasible for adversary to alter the hash value to fool the receiver.
- **To prevent this attack, the hash value generated by Alice must be protected.**



(a) Use of hash function to check data integrity



(b) Man-in-the-middle attack

Figure 11.2 Attack Against Hash Function

a) achieve authentication, confidentiality, data integrity.

b) do not require confidentiality

c) share a common secret S , not sent, an opponent cannot modify an intercepted message and cannot generate a false message.

d) Confidentiality added to (c).

When confidentiality is not required, method (b) has an advantage over methods (a) and (d), which encrypts the entire message, less computation is required.

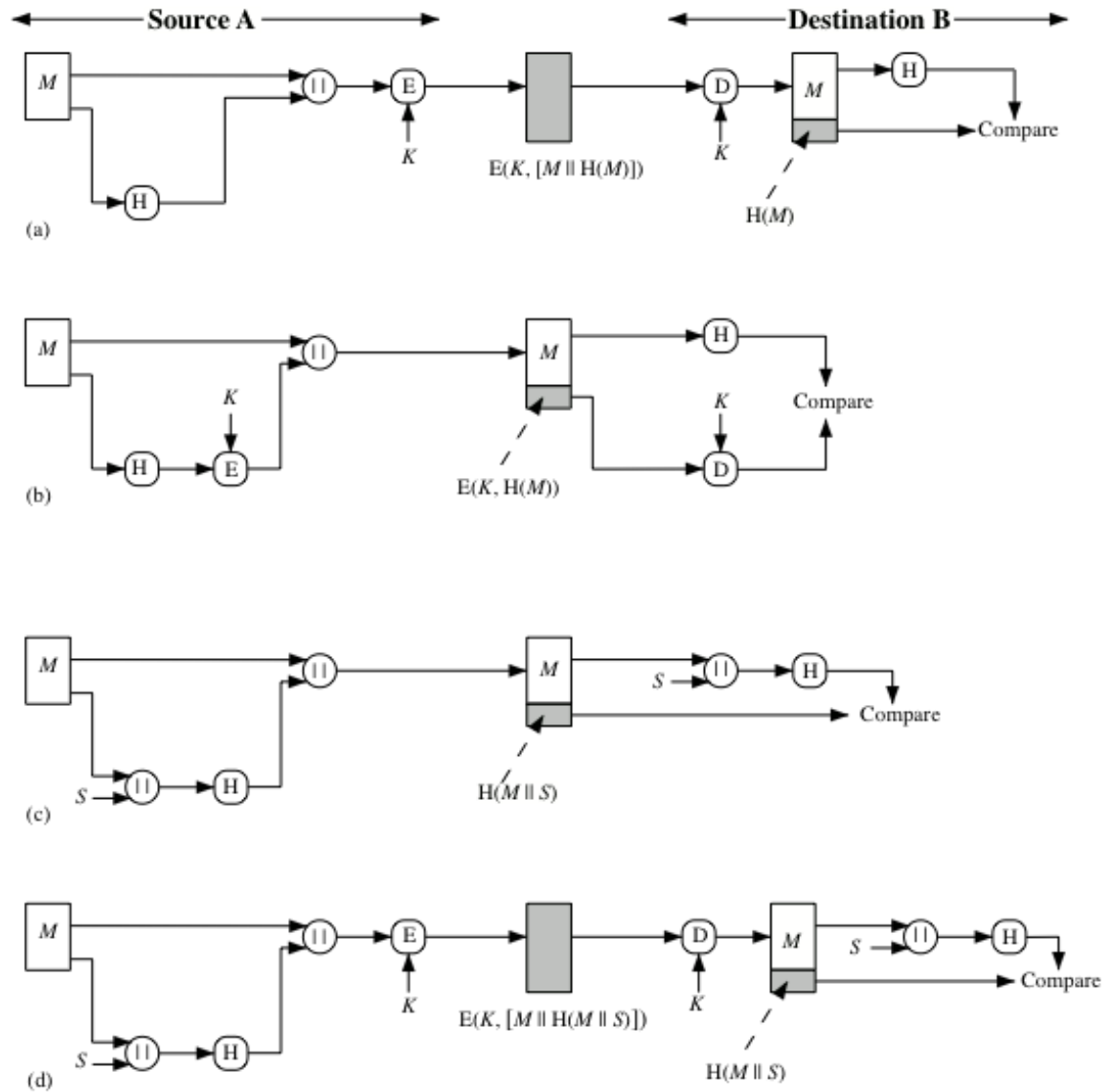


Figure 11.3 Simplified Examples of the Use of a Hash Function for Message Authentication

Justification for encryption usage:

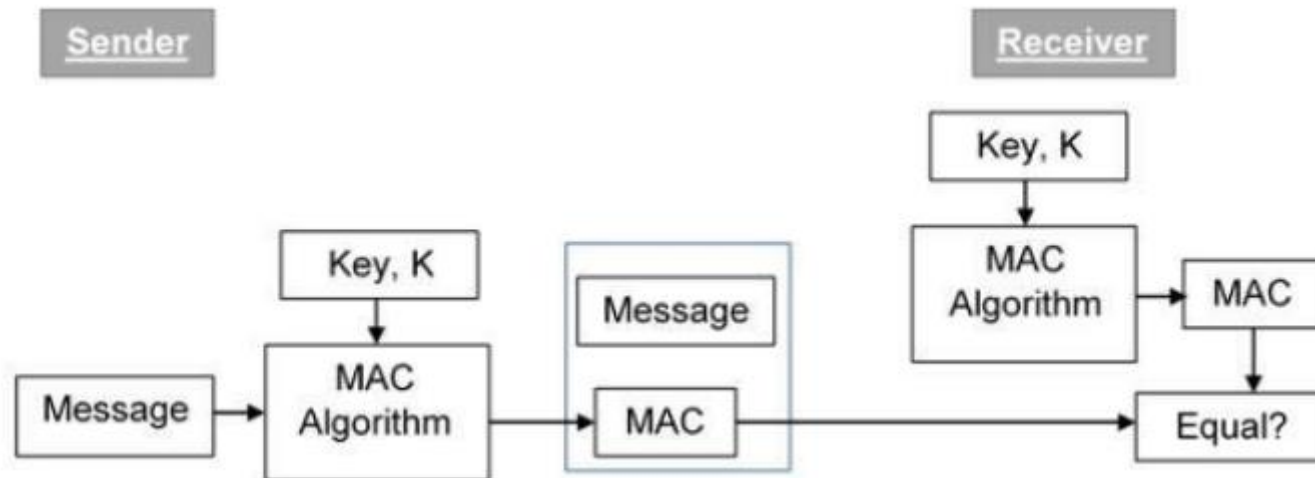
- **Encryption software is relatively slow.** Even though the amount of data to be encrypted per message is small, there may be a steady stream of messages into and out of a system.
- **Encryption hardware costs are not negligible.** Low-cost chip implementations of DES are available, but the cost adds up if all nodes in a network must have this capability.
- Encryption hardware is optimized **toward large data sizes**. For small blocks of data, a high proportion of the time is spent in initialization/invocation **overhead**.
- Encryption algorithms may be covered by patents, and there is a cost associated with licensing their use.

Message Authentication Code (MAC)

- Also known as a *keyed hash function*
- Typically used between two parties that **share a secret key** to authenticate information exchanged between those parties

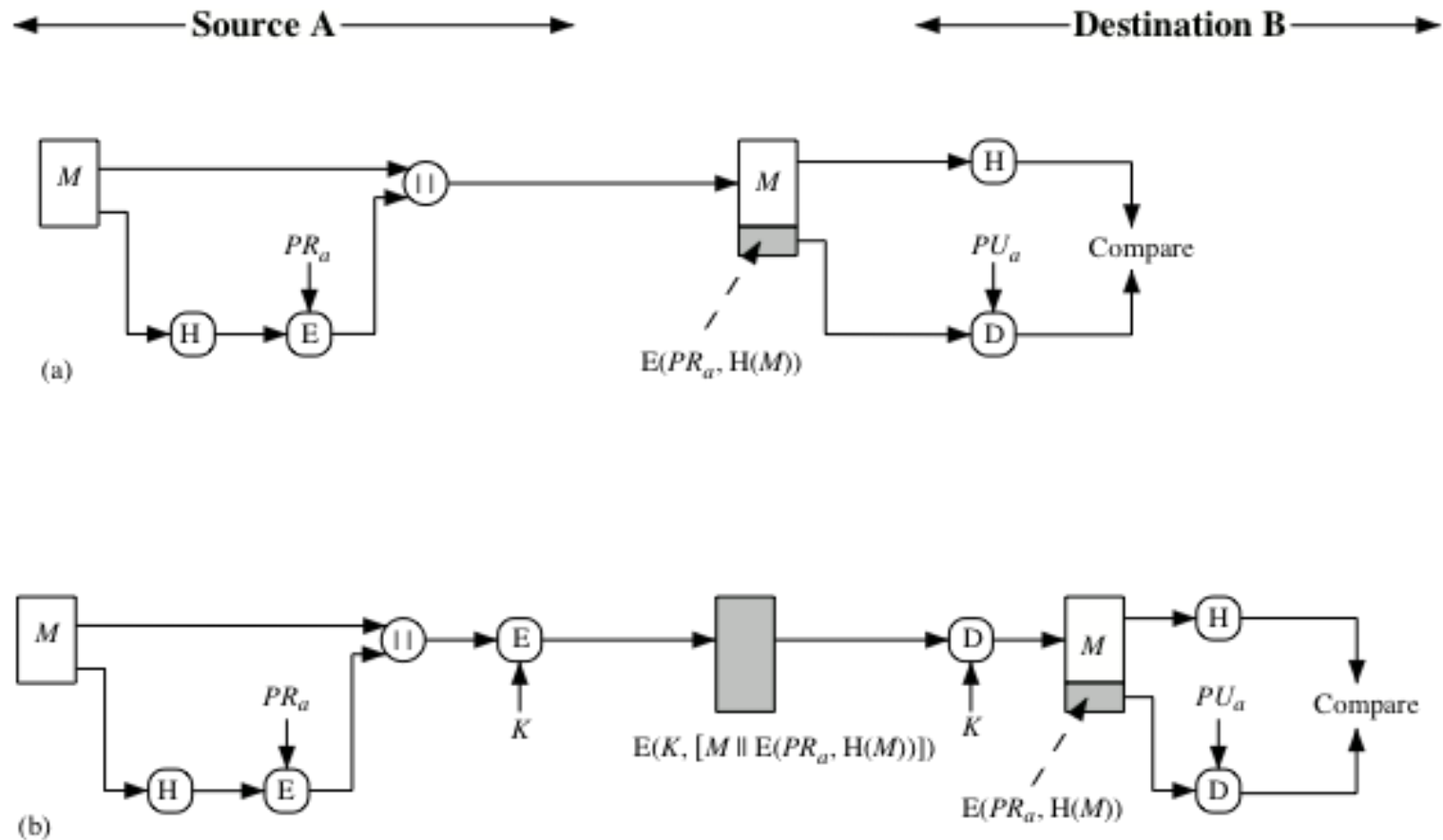
Takes as input a secret key and a data block and produces a hash value (MAC) which is associated with the protected message

- If the integrity of the message needs to be checked, the MAC function can be applied to the message and the result compared with the associated MAC value
- An attacker who alters the message will be unable to alter the associated MAC value without knowledge of the secret key



Digital Signature

- Operation is similar to that of the MAC
- The hash value of a message is encrypted with a user's private key
- Anyone who knows the user's public key can **verify the integrity of the message**
- An attacker who wishes to alter the message **would need to know the user's private key**



- a. The hash code is encrypted, using public-key encryption with the sender's private key. it provides **authentication** and **digital signature**, because **only the sender could have produced the encrypted hash code**.
- b. If **confidentiality** and a digital signature is desired, then the message plus the **private-key-encrypted hash code can be encrypted using a symmetric secret key**.

Figure 11.4 Simplified Examples of Digital Signatures

Other Hash Function Uses

Commonly used to create a one-way password file

When a user enters a password, the hash of that password is compared to the stored hash value for verification

This approach to password protection is used by most operating systems

Can be used for intrusion and virus detection

Store $H(F)$ for each file on a system and secure the hash values

One can later determine if a file has been modified by recomputing $H(F)$

An intruder would need to change F without changing $H(F)$

Can be used to construct a pseudorandom function (PRF) or a pseudorandom number generator (PRNG)

A common application for a hash-based PRF is for the generation of symmetric keys

Requirements and Security

Before proceeding, we need to define two terms.

Preimage

- x is the preimage of h for a hash value $h = H(x)$
- Is a data block whose hash function, using the function H , is h
- Because H is a many-to-one mapping, for any given hash value h , there will in general be multiple preimages

Collision

- Occurs if we have $x \neq y$ and $H(x) = H(y)$
- Because we are using hash functions for data integrity, collisions are clearly undesirable



Table 11.1

Accepted requirements for a Cryptographic Hash function H

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h , it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.
Pseudorandomness	Output of H meets standard tests for pseudorandomness

(Table can be found on page 323 in textbook.)

Table 11.2

Hash Function Resistance Properties Required for Various Data Integrity Applications

	Preimage Resistant	Second Preimage Resistant	Collision Resistant
Hash + digital signature	yes	yes	yes*
Intrusion detection and virus detection		yes	
Hash + symmetric encryption			
One-way password file	yes		
MAC	yes	yes	yes*

* Resistance required if attacker is able to mount a chosen message attack

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Rounds	Operations	Security (bits)
MD5 (as reference)		128	128 (4 × 32)	512	Unlimited ^[37]	64	And, Xor, Rot, Add (mod 2 ³²), Or	<64 (collisions found)
SHA-0		160	160 (5 × 32)	512	2 ⁶⁴ − 1	80	And, Xor, Rot, Add (mod 2 ³²), Or	<80 (collisions found)
SHA-1		160	160 (5 × 32)	512	2 ⁶⁴ − 1	80		<80 (theoretical attack ^[38])
SHA-2	SHA-224	224	256 (8 × 32)	512	2 ⁶⁴ − 1	64	And, Xor, Rot, Add (mod 2 ³²), Or, Shr	112
	SHA-256	256						128
	SHA-384	384	512 (8 × 64)	1024	2 ¹²⁸ − 1	80	And, Xor, Rot, Add (mod 2 ⁶⁴), Or, Shr	192
	SHA-512	512						256
SHA-3	SHA-512/224	224						112
	SHA-512/256	256						128
	SHA3-224	224	1600 (5 × 5 × 64)	1152	Unlimited ^[39]	24 ^[40]	And, Xor, Rot, Not	112
	SHA3-256	256		1088				128
	SHA3-384	384		832				192
	SHA3-512	512		576				256
	SHAKE128	d (arbitrary)		1344				min(d/2, 128)
	SHAKE256	d (arbitrary)		1088				min(d/2, 256)

Attacks on Hash Functions

there are two categories of attacks on hash functions:
brute-force attacks and cryptanalysis

Brute-Force Attacks

- Does not depend on the specific algorithm, only depends on **bit length**
- In the case of a hash function, attack depends only on the bit length of the hash value
- Method is to pick values at random and try each one until a collision occurs

Cryptanalysis

- An attack based on **weaknesses** in a particular cryptographic algorithm
- Seek to exploit some property of the algorithm to perform some attack other than an exhaustive search

Table 11.3

Comparison of SHA Parameters

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

Note: All sizes are measured in bits.