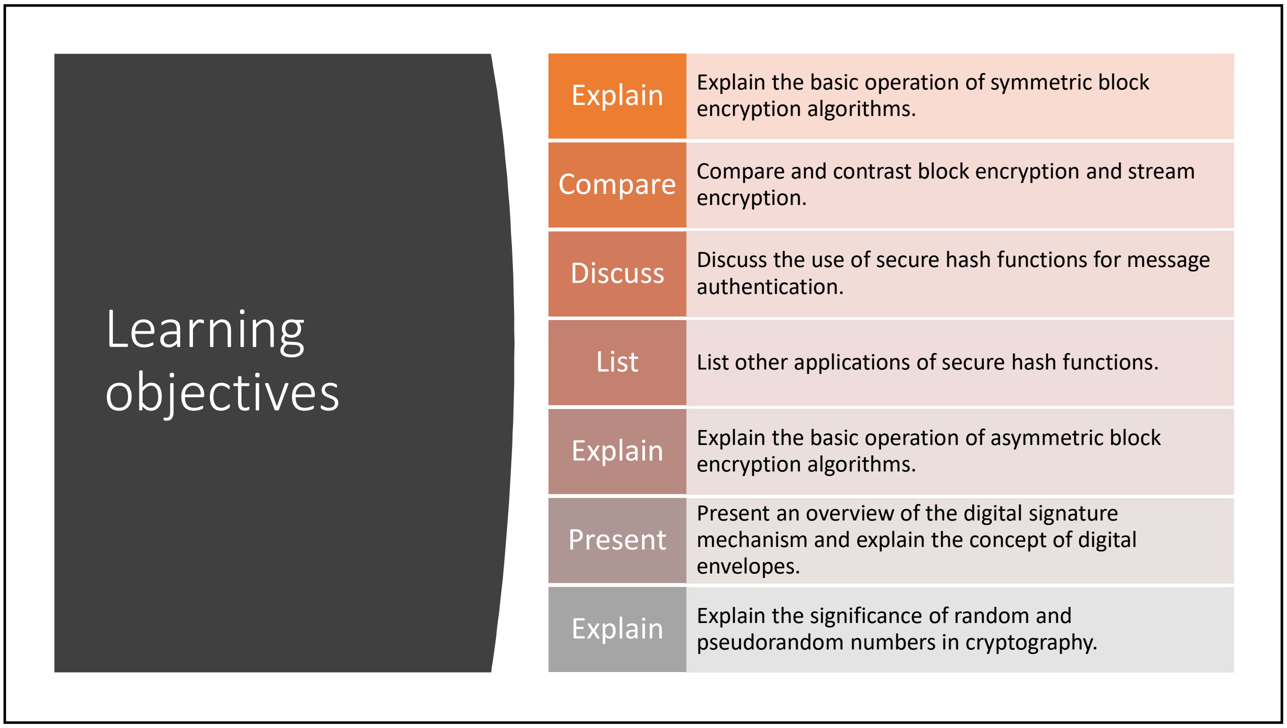
Computer Security – Principles and Practice (Pearson, fourth edition)

W. Stallings, L. Brown

* These slides are an adaptation of the original slides of the authors of the book

Cryptographic tools

1



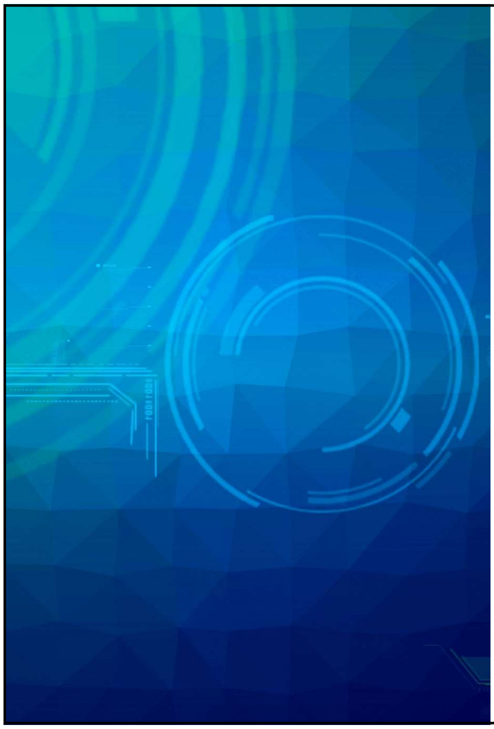
Learning objectives

Explain	Explain the basic operation of symmetric block encryption algorithms.
Compare	Compare and contrast block encryption and stream encryption.
Discuss	Discuss the use of secure hash functions for message authentication.
List	List other applications of secure hash functions.
Explain	Explain the basic operation of asymmetric block encryption algorithms.
Present	Present an overview of the digital signature mechanism and explain the concept of digital envelopes.
Explain	Explain the significance of random and pseudorandom numbers in cryptography.

2

Symmetric encryption

3

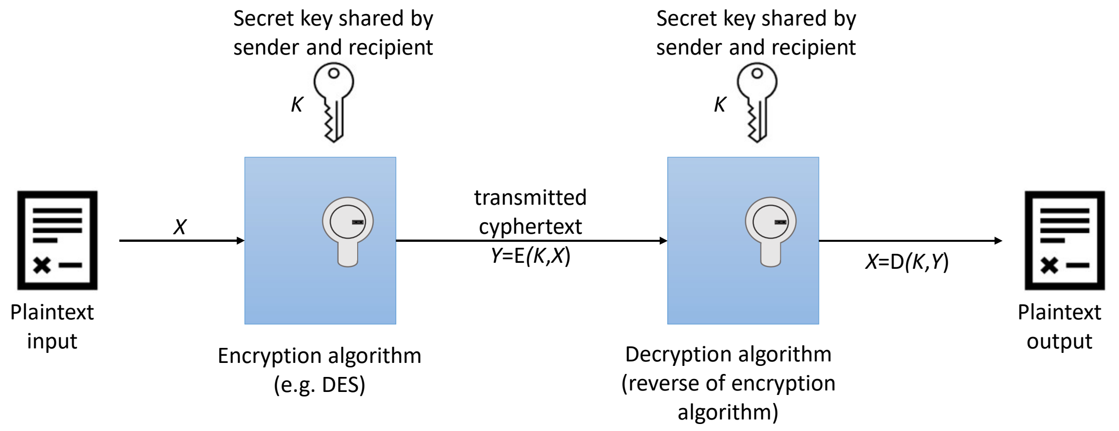


Symmetric Encryption

- The universal technique for providing confidentiality for transmitted or stored data
- Also referred to as conventional encryption or single-key encryption
- Two requirements for secure use:
  - Need a strong encryption algorithm
  - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure

4

Simplified model of symmetric Encryption



Secret key shared by sender and recipient

K

Plaintext input

X

Encryption algorithm (e.g. DES)

transmitted cyphertext *Y*=E(*K*,*X*)

Secret key shared by sender and recipient

K

Decryption algorithm (reverse of encryption algorithm)

X=D(*K*,*Y*)

Plaintext output

Attacking symmetric encryption

Cryptoanalytic attacks

- Rely on:
  - Nature of the algorithm
  - Some knowledge of the general characteristics of the plaintext
  - Some sample plaintext-ciphertext pairs
- Exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
  - If successful, all future and past messages encrypted with that key are compromised

Brute force attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
  - On average half of all possible keys must be tried to achieve success

Comparison of Three Popular Symmetric Encryption Algorithms

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES: Data Encryption Standard; AES: Advanced Encryption Standard

7

Data Encryption Standard (DES)

Until recently was the most widely used encryption scheme

- FIPS PUB 46 (January 1977)
- Referred to as the Data Encryption Algorithm (DEA)
- Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block

Strength concerns:

- Concerns about the algorithm itself
  - DES is the most studied encryption algorithm in existence
- Concerns about the use of a 56-bit key
  - The speed of commercial off-the-shelf processors makes this key length woefully inadequate

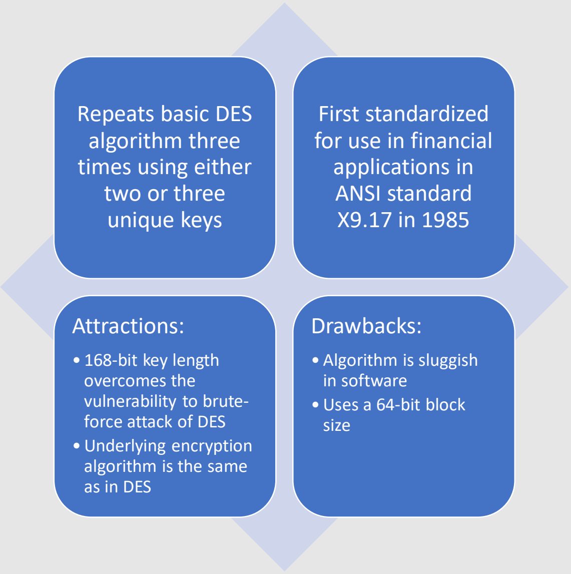
8

Average time required for exhaustive key search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/sec	Time Required at 10^{13} decryptions/sec
56	DES	2^{56} ~ 7.2 x 10^{16}	2^{45} µs ~ 1.14 years	1 hours
128	AES	2^{128} ~ 3.4 x 10^{38}	2^{117} µs ~ 5.4 x 10^{21} years	5.4 x 10^{17} years
168	Triple DES	2^{168} ~ 3.7 x 10^{50}	2^{157} µs ~ 5.9 x 10^{33} years	5.9 x 10^{29} years
192	AES	2^{192} ~ 6.3 x 10^{57}	2^{181} µs ~ 9.9 x 10^{40} years	9.9 x 10^{36} years
256	AES	2^{256} ~ 1.2 x 10^{77}	2^{245} µs ~ 1.8 x 10^{60} years	1.8 x 10^{56} years

(average case)

9



Repeats basic DES algorithm three times using either two or three unique keys

First standardized for use in financial applications in ANSI standard X9.17 in 1985

Attractions:

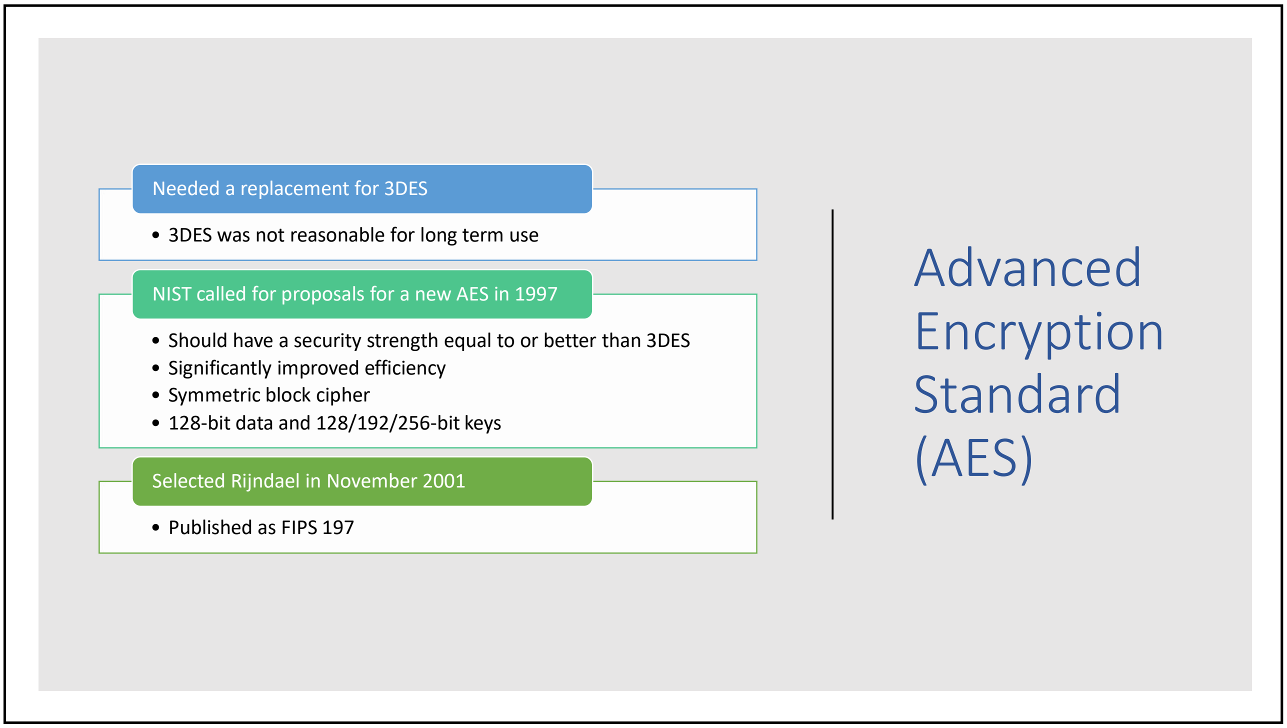
- 168-bit key length overcomes the vulnerability to brute-force attack of DES
- Underlying encryption algorithm is the same as in DES

Drawbacks:

- Algorithm is sluggish in software
- Uses a 64-bit block size

Triple DES (3DES)

10



Advanced Encryption Standard (AES)

Needed a replacement for 3DES

- 3DES was not reasonable for long term use

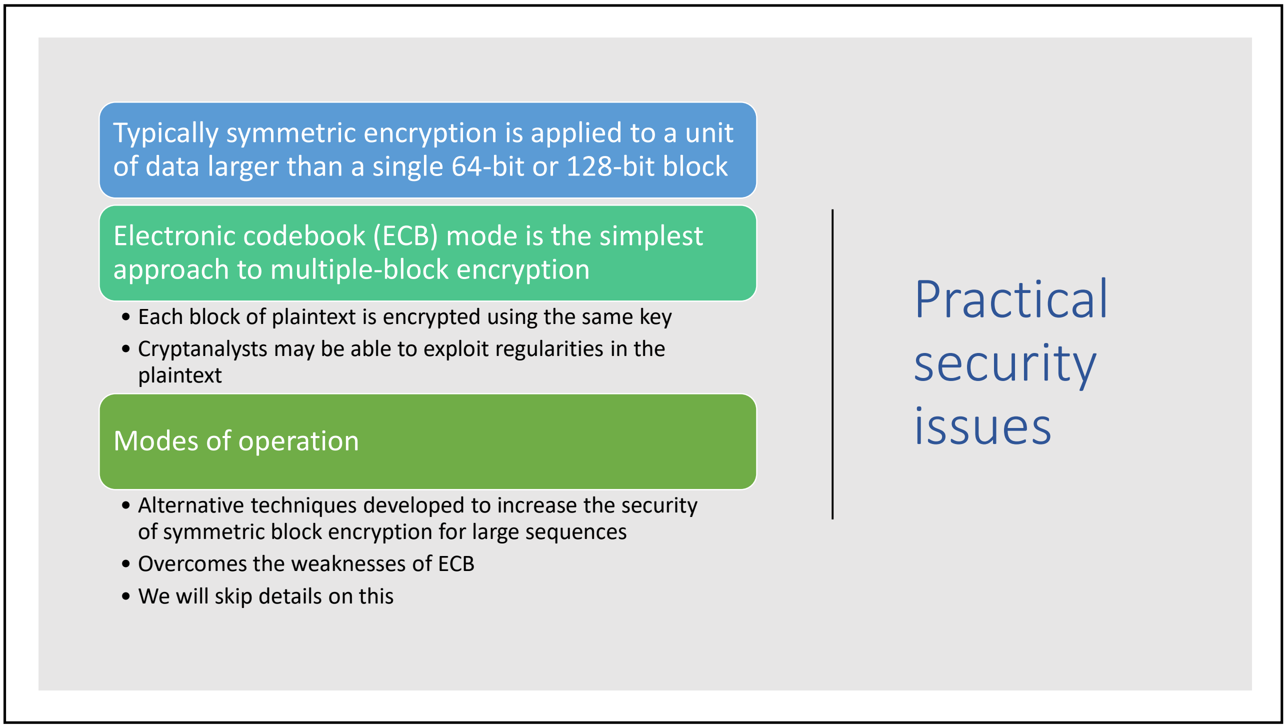
NIST called for proposals for a new AES in 1997

- Should have a security strength equal to or better than 3DES
- Significantly improved efficiency
- Symmetric block cipher
- 128-bit data and 128/192/256-bit keys

Selected Rijndael in November 2001

- Published as FIPS 197

11



Practical security issues

Typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block

Electronic codebook (ECB) mode is the simplest approach to multiple-block encryption

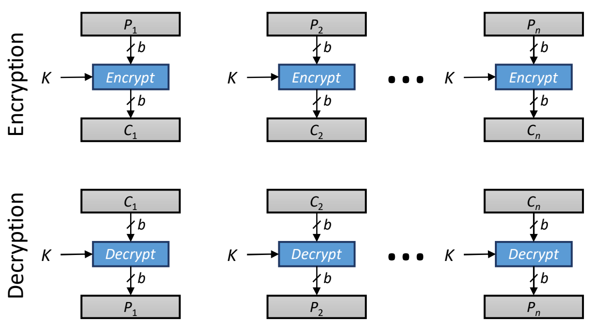
- Each block of plaintext is encrypted using the same key
- Cryptanalysts may be able to exploit regularities in the plaintext

Modes of operation

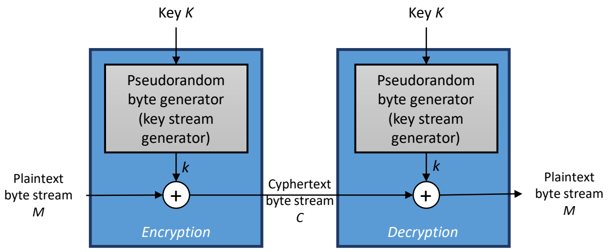
- Alternative techniques developed to increase the security of symmetric block encryption for large sequences
- Overcomes the weaknesses of ECB
- We will skip details on this

12

Types of symmetric encryption



Block cipher encryption (electronic codebook mode)



Stream encryption

13

Block cyphers

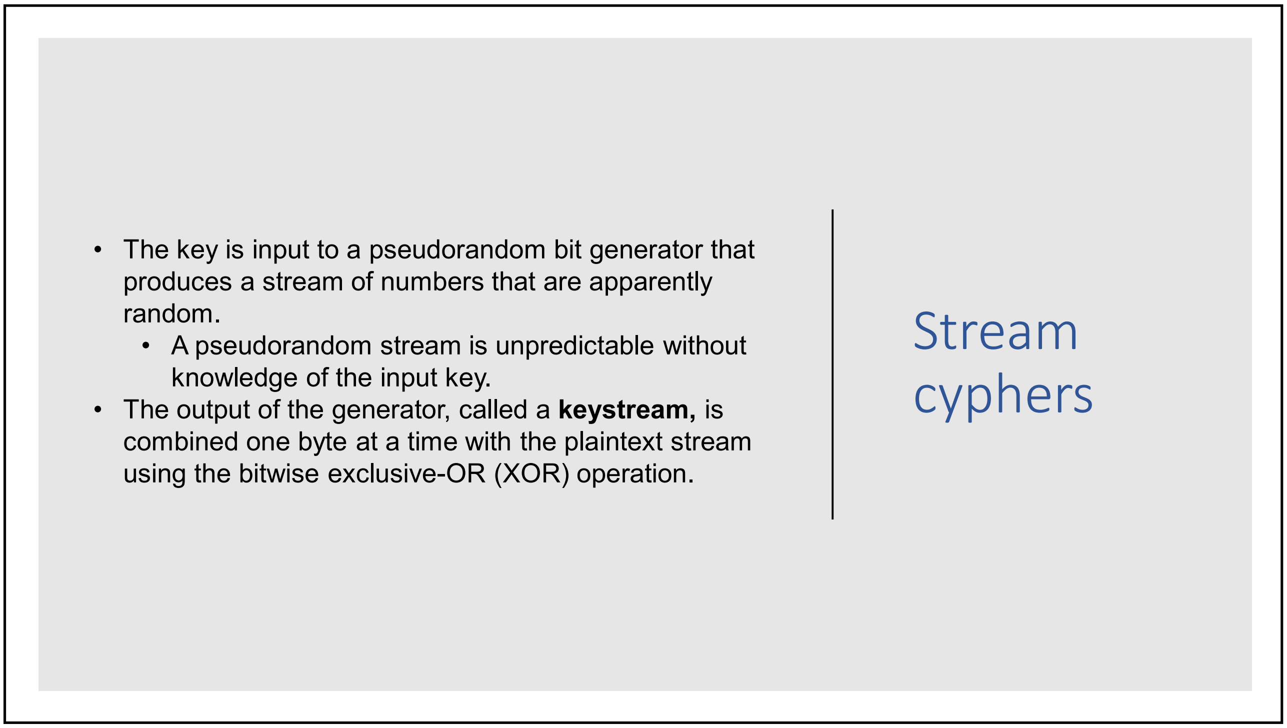
With the Electronic Codebook (ECB) mode:

- A plaintext of length *nb is divided into n b-bit blocks (P_1, P_2,….,P_n).*
- Each block is encrypted using the same algorithm and the same encryption key, to produce a sequence of *n b-bit blocks of ciphertext (C_1, C_2,….,C_n).*

Security concerns:

- A cryptanalyst may exploit regularities in the plaintext to ease the task of decryption.
- For example, if it is known that the message always starts out with certain predefined fields, then the cryptanalyst may have a number of known plaintext-ciphertext pairs to work with.

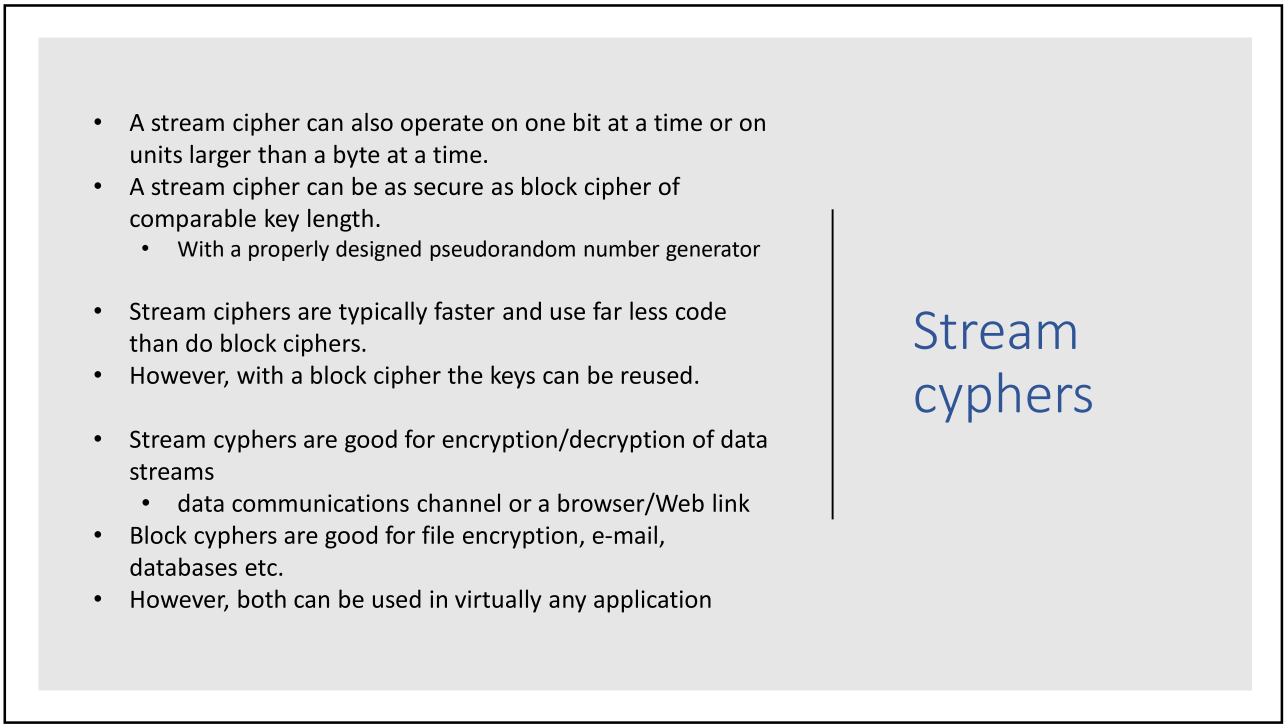
14



Stream cyphers

- The key is input to a pseudorandom bit generator that produces a stream of numbers that are apparently random.
  - A pseudorandom stream is unpredictable without knowledge of the input key.
- The output of the generator, called a **keystream,** is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation.

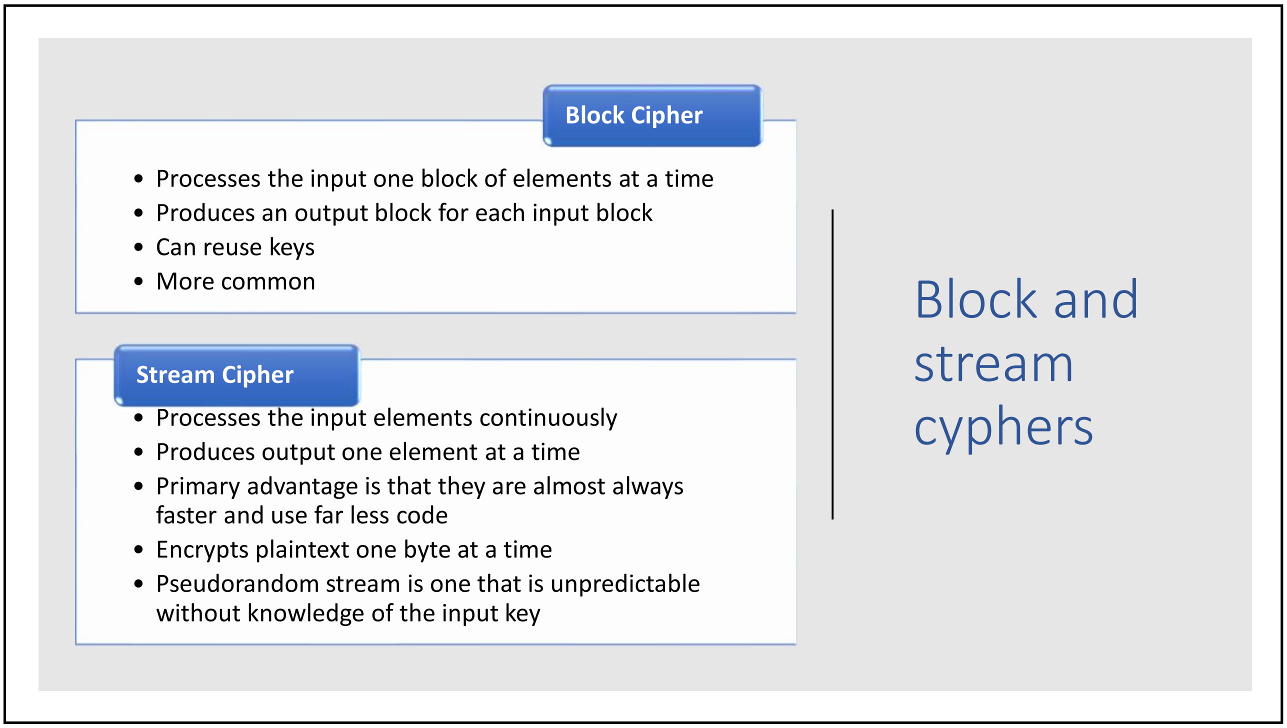
15



Stream cyphers

- A stream cipher can also operate on one bit at a time or on units larger than a byte at a time.
- A stream cipher can be as secure as block cipher of comparable key length.
  - With a properly designed pseudorandom number generator
- Stream ciphers are typically faster and use far less code than do block ciphers.
- However, with a block cipher the keys can be reused.
- Stream cyphers are good for encryption/decryption of data streams
  - data communications channel or a browser/Web link
- Block cyphers are good for file encryption, e-mail, databases etc.
- However, both can be used in virtually any application

16



Block and stream cyphers

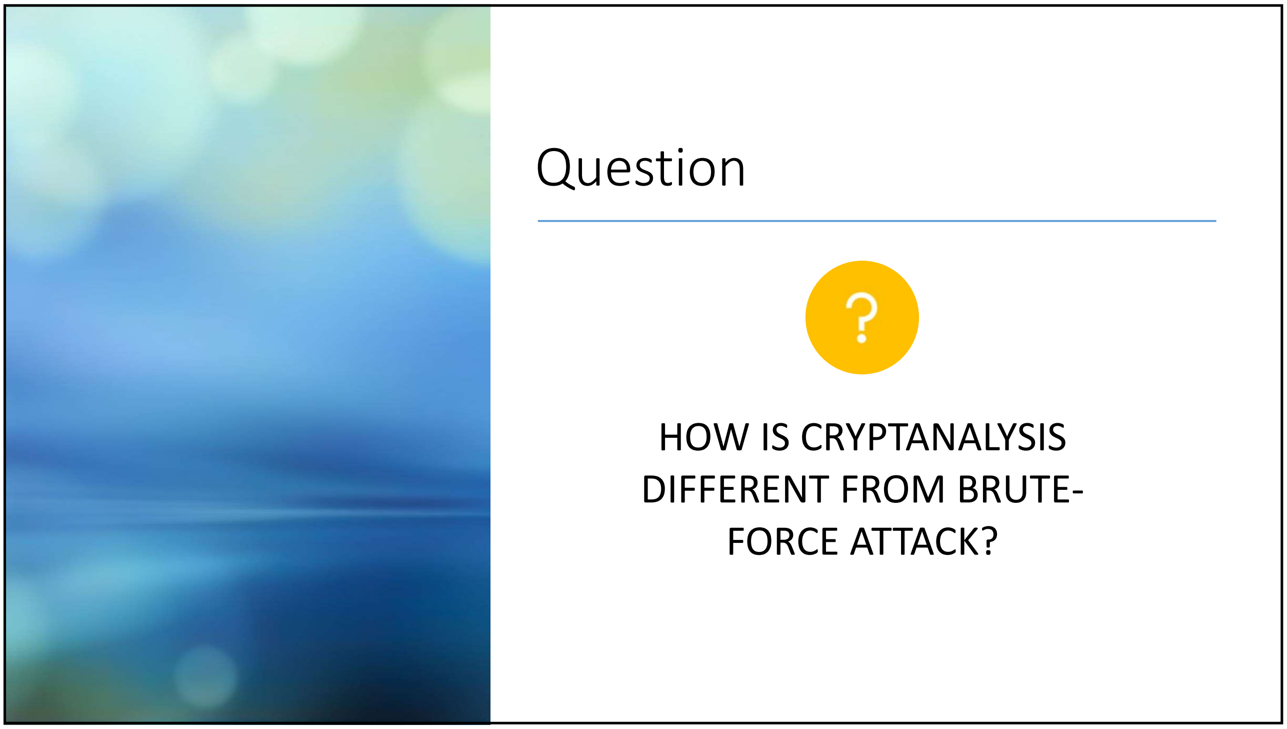
Block Cipher

- Processes the input one block of elements at a time
- Produces an output block for each input block
- Can reuse keys
- More common

Stream Cipher

- Processes the input elements continuously
- Produces output one element at a time
- Primary advantage is that they are almost always faster and use far less code
- Encrypts plaintext one byte at a time
- Pseudorandom stream is one that is unpredictable without knowledge of the input key

17

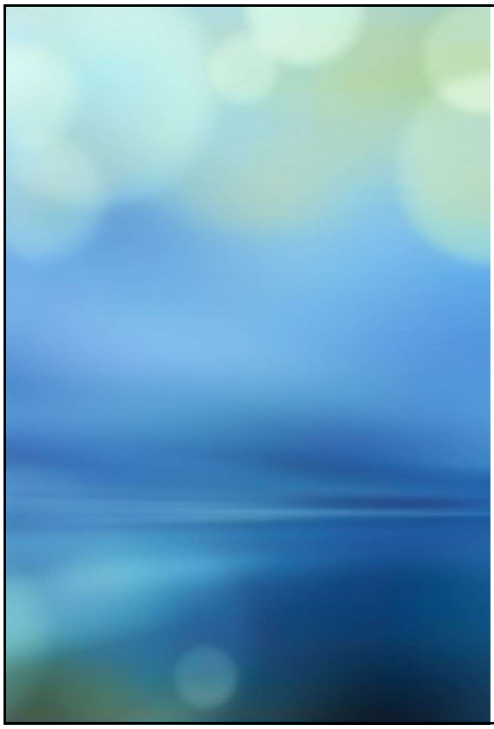


Question

?

HOW IS CRYPTANALYSIS DIFFERENT FROM BRUTE-FORCE ATTACK?

18

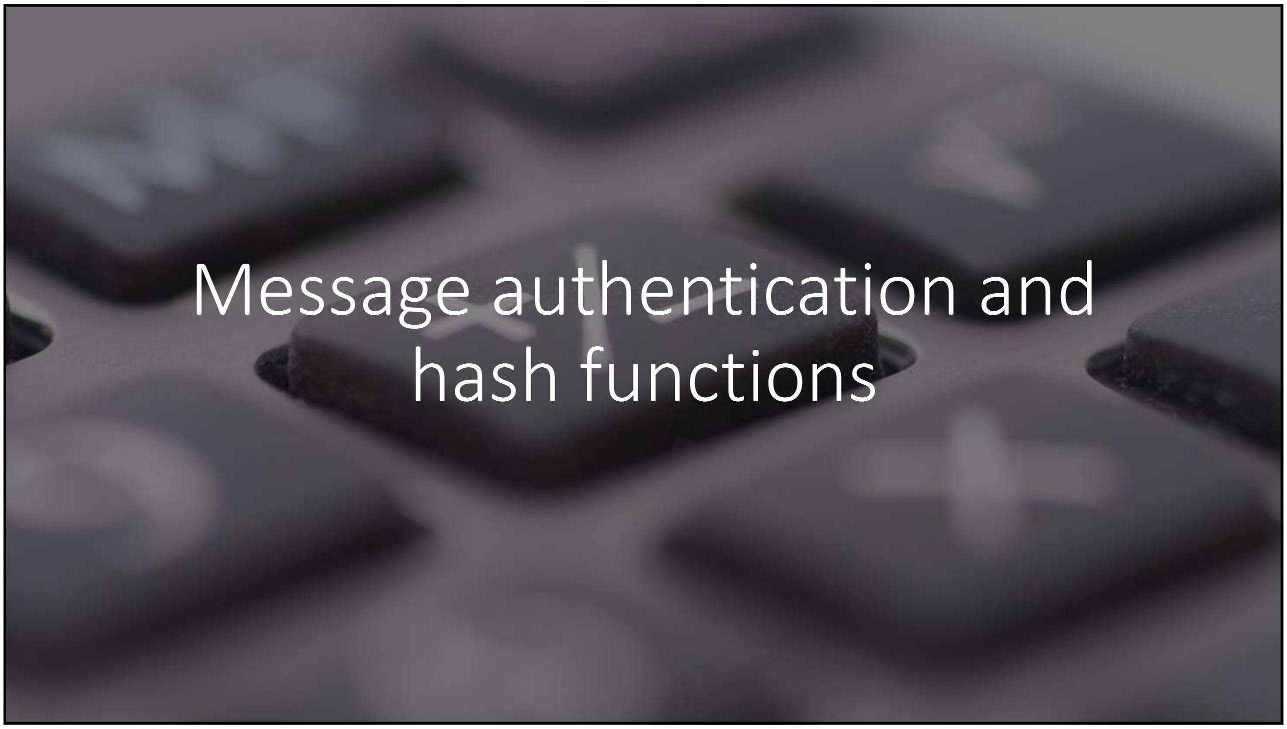


Question

?

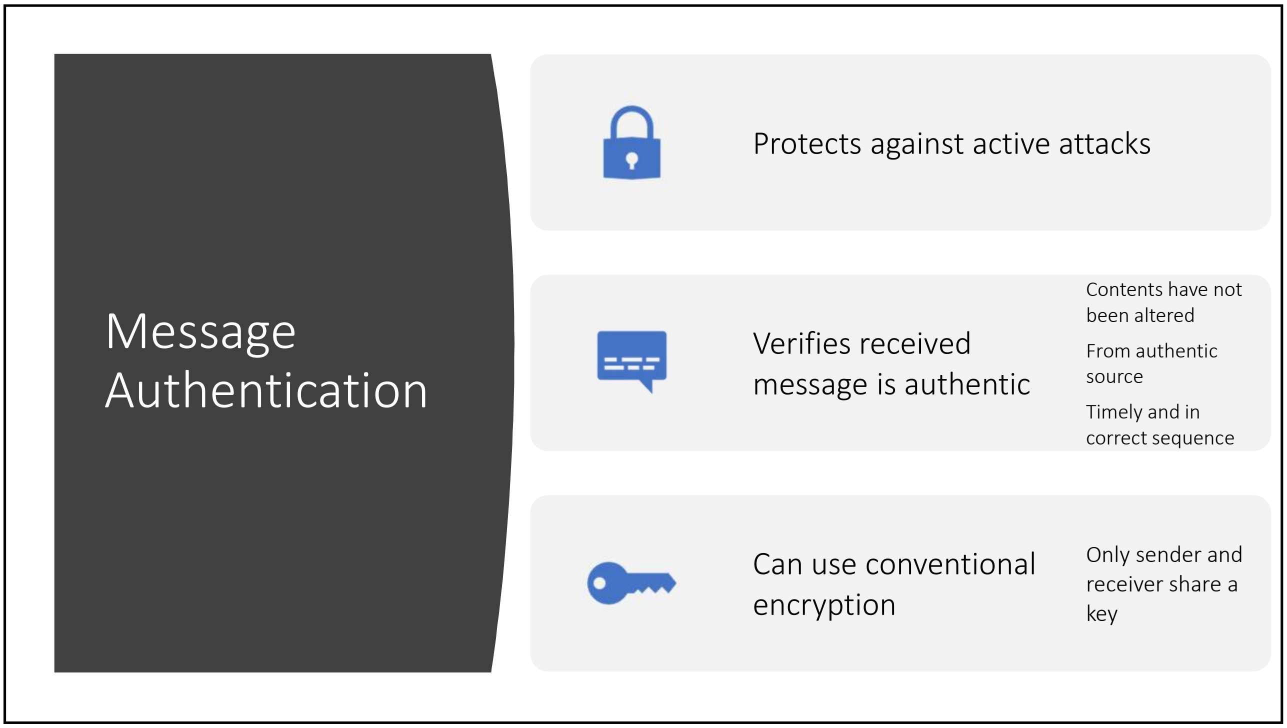
WHAT ARE THE (TWO) PRINCIPAL REQUIREMENTS FOR THE SECURE USE OF SYMMETRIC ENCRYPTION?

19



Message authentication and hash functions

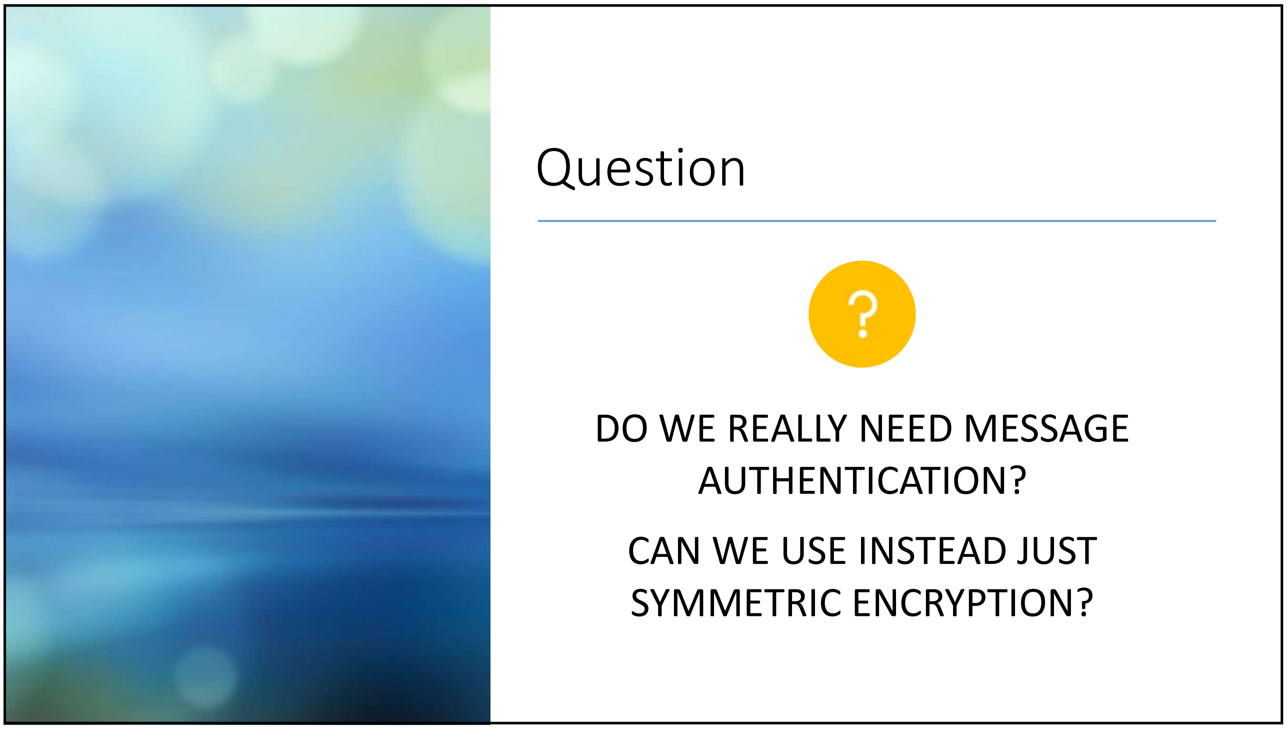
20



Message Authentication

- Protects against active attacks
- Verifies received message is authentic
  - Contents have not been altered
  - From authentic source
  - Timely and in correct sequence
- Can use conventional encryption
  - Only sender and receiver share a key

21



Question

DO WE REALLY NEED MESSAGE AUTHENTICATION?

CAN WE USE INSTEAD JUST SYMMETRIC ENCRYPTION?

22

Message authentication without confidentiality

- Message encryption by itself does not provide a secure form of authentication
- It is possible to combine authentication and confidentiality in a single algorithm by encrypting a message plus its authentication tag
- Typically, message authentication is provided as a separate function from message encryption
- Situations in which message authentication without confidentiality may be preferable include:
    1. There are a number of applications in which the same message is broadcast to a number of destinations
    2. An exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages
    3. Authentication of a computer program in plaintext is an attractive service
- Thus, there is a place for both authentication and encryption in meeting security requirements

23

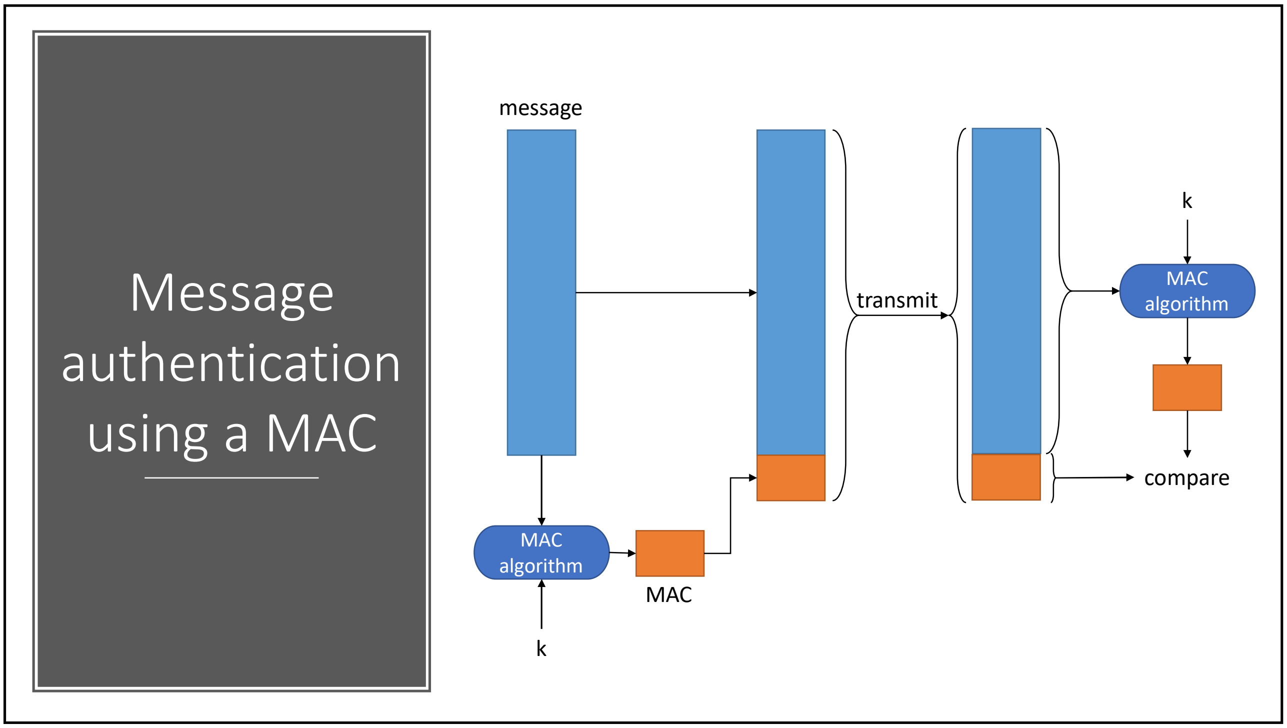
Message Authentication Code

- A small block of data (the message authentication code – MAC) is appended to the message to be authenticated
    - MAC generated by means of a secret key:

$$MAC = Func(Key, Message)$$

    - The secret key is shared between the two communicating parties

24



Message authentication using a MAC

message

transmit

k

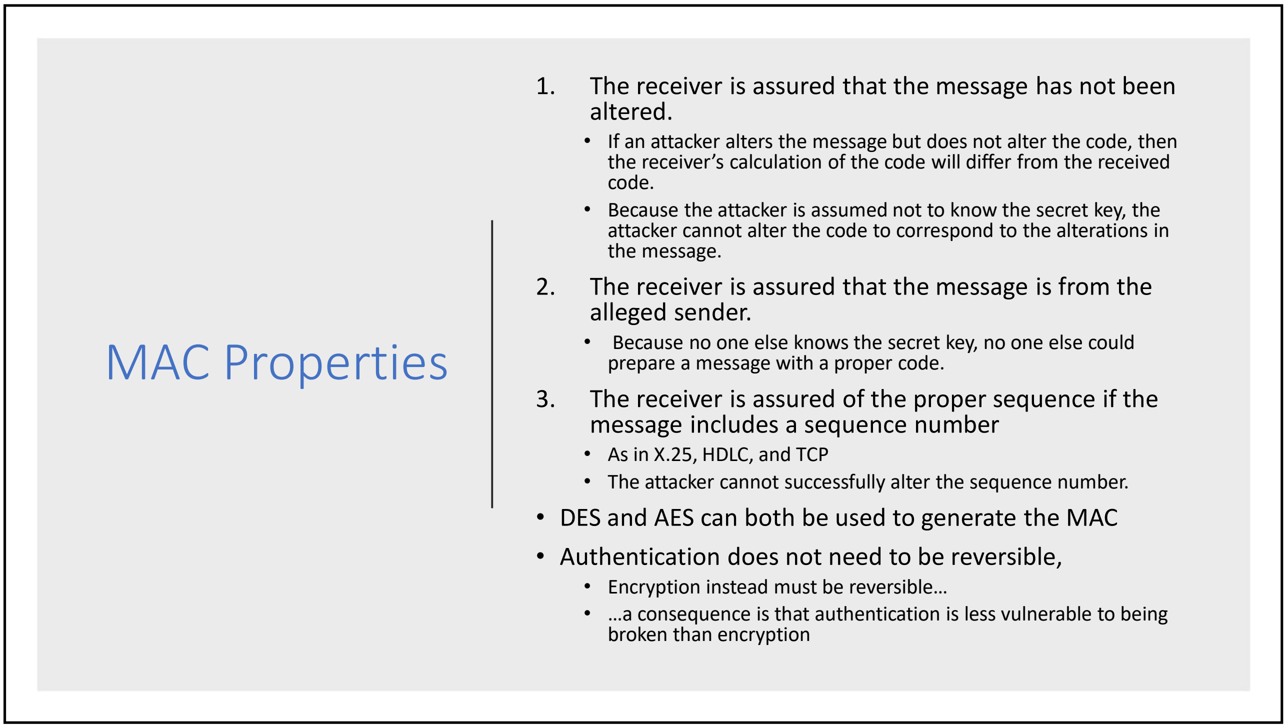
MAC algorithm

compare

MAC algorithm

MAC

k



MAC Properties

1. The receiver is assured that the message has not been altered.
   - If an attacker alters the message but does not alter the code, then the receiver’s calculation of the code will differ from the received code.
   - Because the attacker is assumed not to know the secret key, the attacker cannot alter the code to correspond to the alterations in the message.
2. The receiver is assured that the message is from the alleged sender.
   - Because no one else knows the secret key, no one else could prepare a message with a proper code.
3. The receiver is assured of the proper sequence if the message includes a sequence number
   - As in X.25, HDLC, and TCP
   - The attacker cannot successfully alter the sequence number.

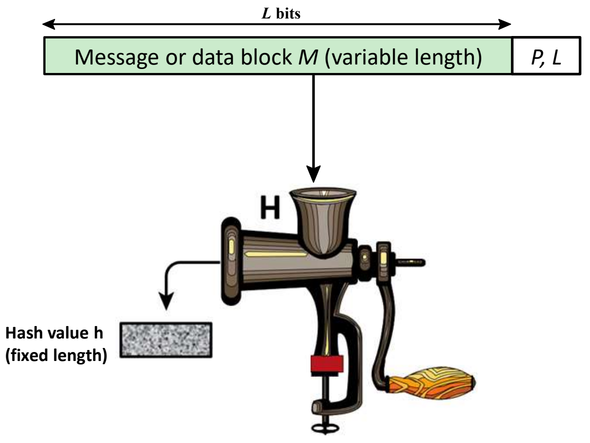
- DES and AES can both be used to generate the MAC
- Authentication does not need to be reversible,
  - Encryption instead must be reversible…
  - …a consequence is that authentication is less vulnerable to being broken than encryption

One-way hash functions

Properties of a one-way hash function $H(\cdot)$:

1. It can be applied to data blocks of any size
2. It produces a fixed length output
3. It is easy to compute (making both software or hardware implementations practical)
4. It is **one way**: it is computationally infeasible to find x such that $H(x) = h$
5. It is **weak collision resistant**: given x it is computationally infeasible to find y such that $H(x) = H(y)$
6. It is **collision resistant**: it is computationally infeasible to find a pair x, y such that $H(x) = H(y)$

27



P, L = padding plus length field

Hash function $h = H(M)$

28

Message authentication with one-way hash functions

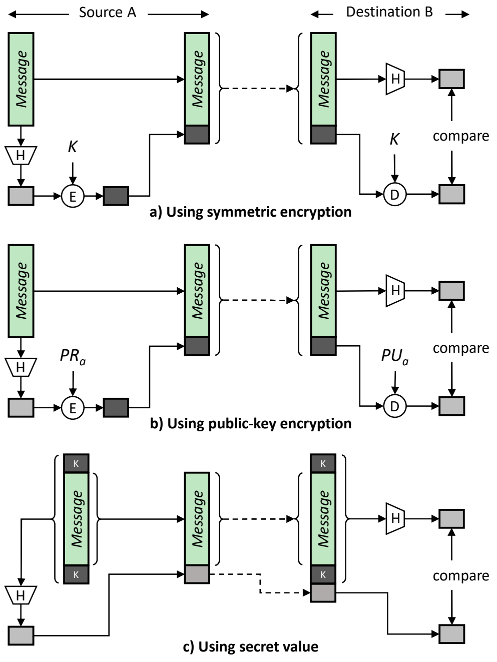
The hash function takes a variable-size message M and produces a fixed-size message digest $H(M)$

- Typically, the message is padded out to an integer multiple of some fixed length (e.g., 1024 bits)
  - 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.

- Does not need a secret key!!

29

Alternative ways of authenticating a message

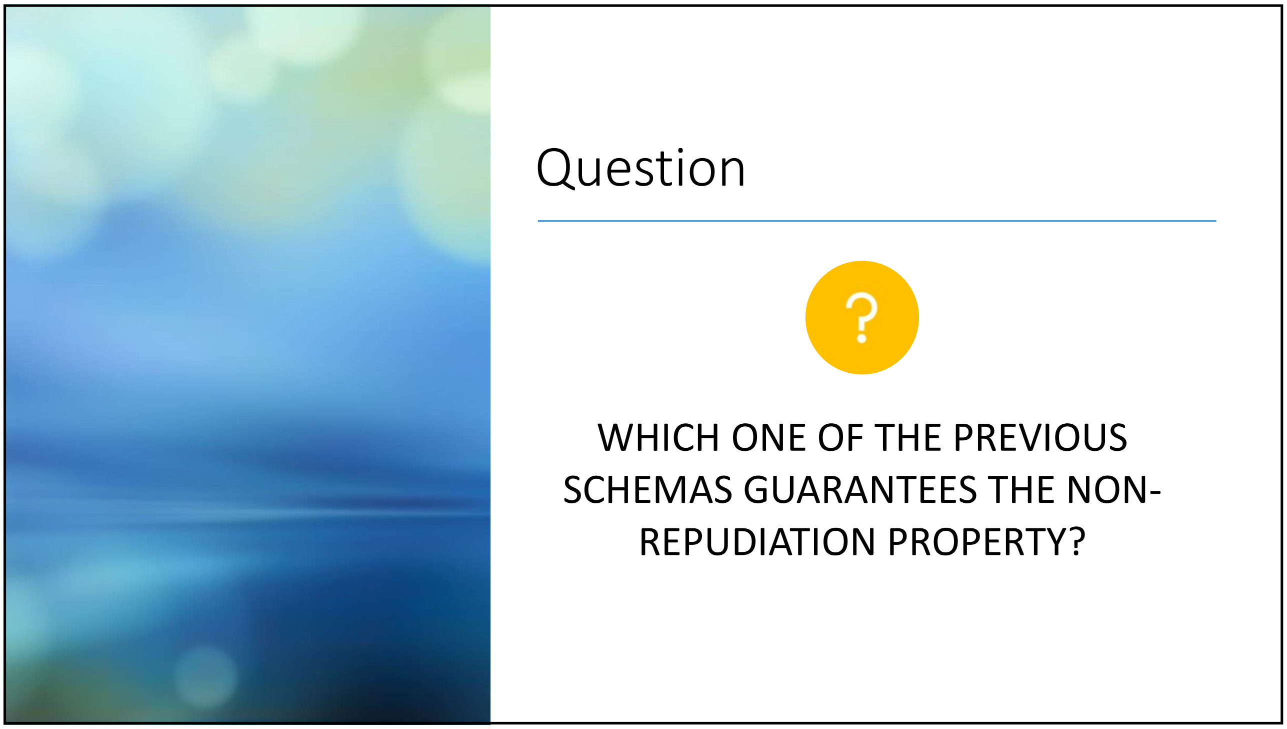


a) Using symmetric encryption

b) Using public-key encryption

c) Using secret value

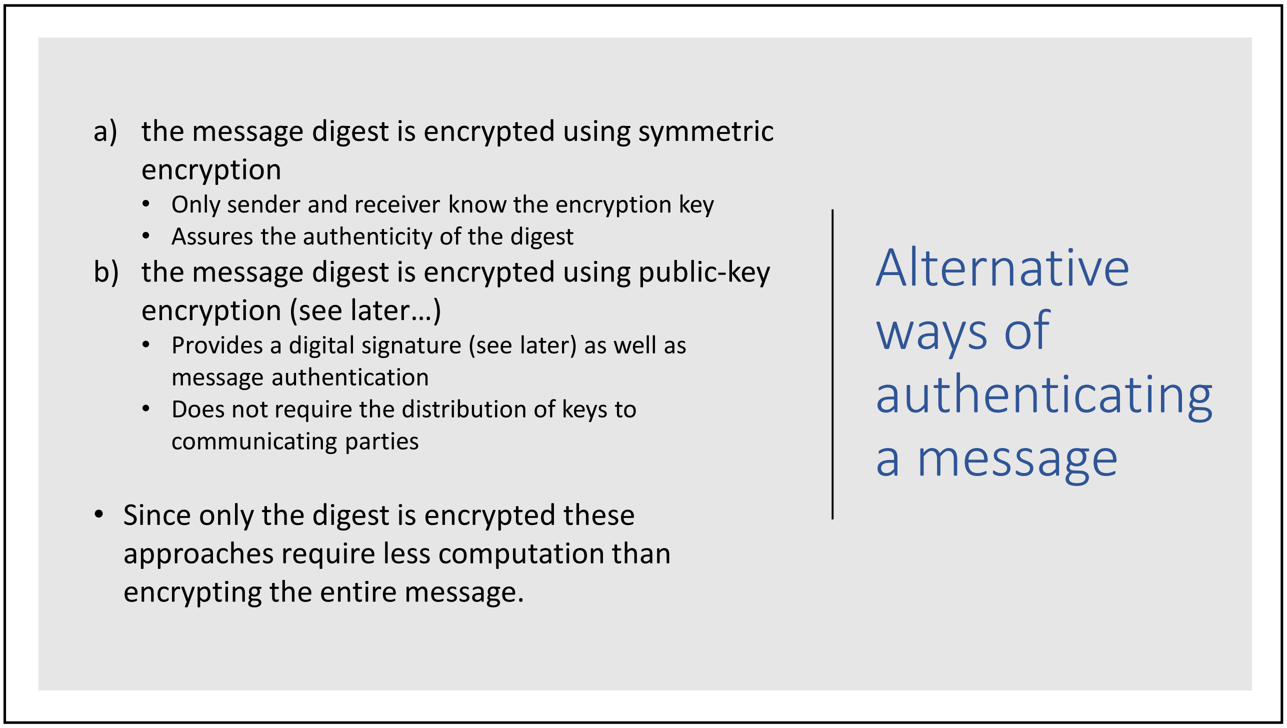
30



Question

WHICH ONE OF THE PREVIOUS SCHEMAS GUARANTEES THE NON-REPUDIATION PROPERTY?

31



Alternative ways of authenticating a message

a) the message digest is encrypted using symmetric encryption
   - Only sender and receiver know the encryption key
   - Assures the authenticity of the digest

b) the message digest is encrypted using public-key encryption (see later…)
   - Provides a digital signature (see later) as well as message authentication
   - Does not require the distribution of keys to communicating parties

- Since only the digest is encrypted these approaches require less computation than encrypting the entire message.

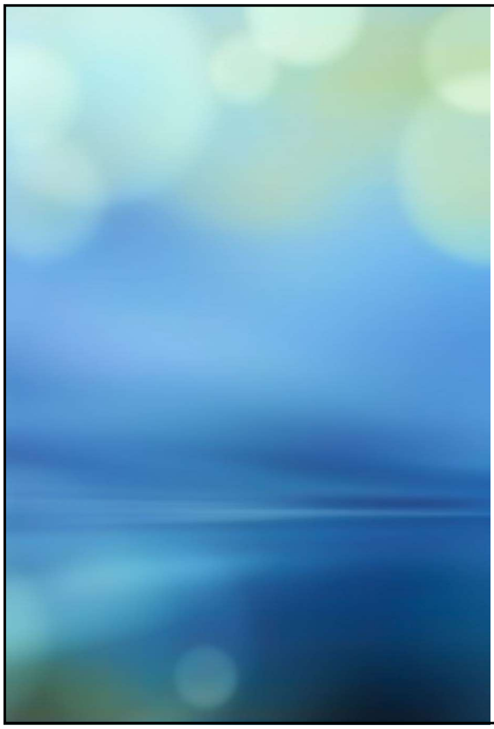
32

Alternative ways of authenticating a message (with no cryptography)

The method (c) instead avoids encryption!

- Advantages:
  - Software encryption is rather slow (even if the message is short, there may be several messages…)
  - Hardware encryption has costs and it is optimized for large data sizes
  - The encryption algorithm may be patented (and thus subject to royalties)
- Method (c) based on the keyed hash MAC technique:
  - Assumes that two communicating parties, say A and B, share a common secret key *K*.
  - As long as *K* is secret there's no way for the attacker to modify the message or to generate a false message.
  - Using *K* at the beginning and at the end makes the scheme more secure.

33



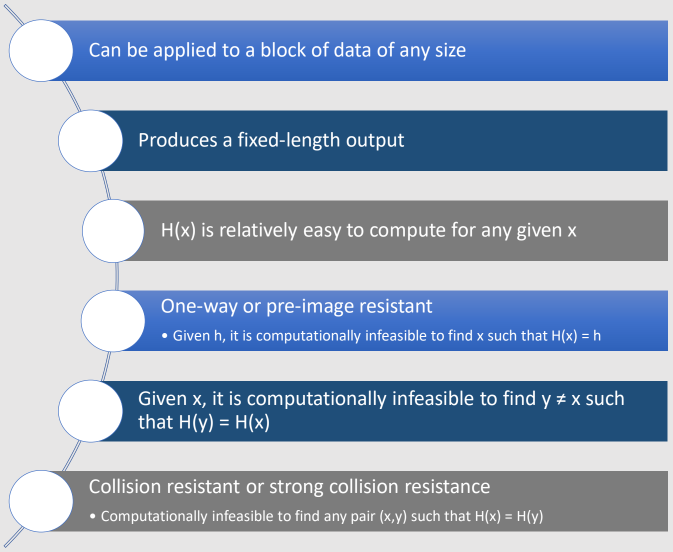
Question

?

WHY DOES THE HASH FUNCTION NEED TO BE ONE-WAY?

One-way: it is computationally infeasible to find x such that $H(x) = h$

34



- Can be applied to a block of data of any size
- Produces a fixed-length output
- H(x) is relatively easy to compute for any given x
- One-way or pre-image resistant
  - Given h, it is computationally infeasible to find x such that H(x) = h
- Given x, it is computationally infeasible to find y ≠ x such that H(y) = H(x)
- Collision resistant or strong collision resistance
  - Computationally infeasible to find any pair (x,y) such that H(x) = H(y)

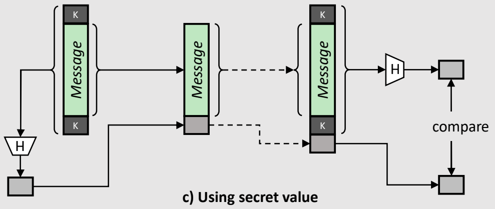
for message authentication, a hash function must have these properties

35

4) One-way or pre-image resistant:

- Given h, it is computationally infeasible to find x such that H(x) = h

In case (c), if this was not true it would be possible to find the secret key k!



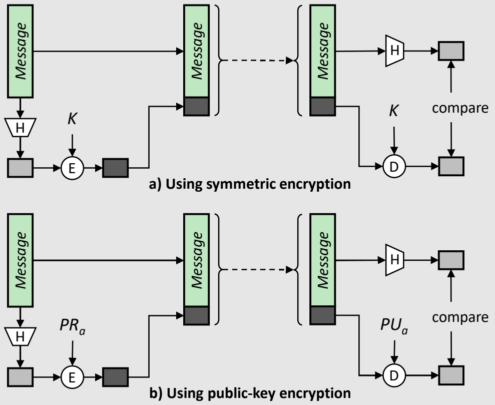
c) Using secret value

for message authentication, a hash function must have these properties

36

5) Given x, it is computationally infeasible to find y ≠ x such that H(y) = H(x)

Necessary to prevent forgery in cases (a), (b) and (c)



a) Using symmetric encryption

b) Using public-key encryption

for message authentication, a hash function must have these properties

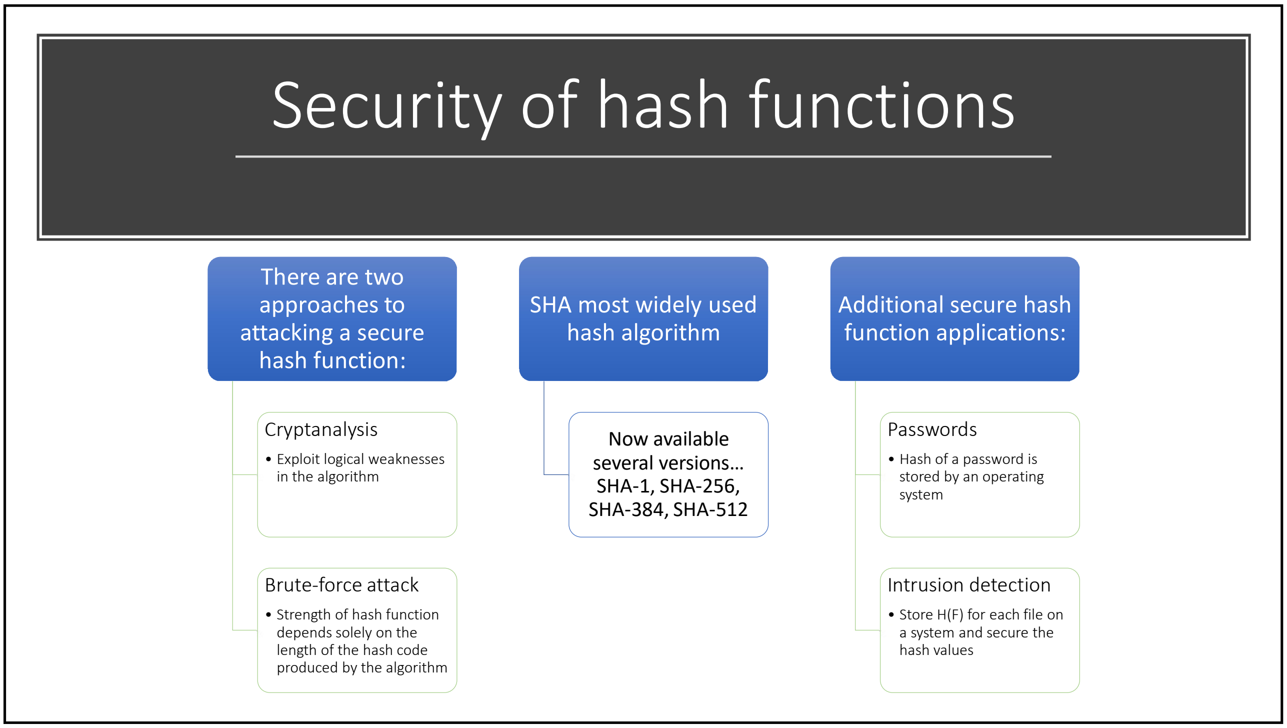
37

6) it is computationally infeasible to find a pair x, y such that $H(x) = H(y)$

- protects against an attack in which one party generates a message for another party to sign.
- For example:
  - suppose Bob gets to write an “I Owe You” message, send it to Alice, and she signs it.
  - Bob forges two messages with the same hash,
  - one of which requires Alice to pay a small amount …
  - and one that requires a large payment.
  - Alice signs the first message and Bob is then able to claim that the second message is authentic.

for message authentication, a hash function must have these properties

38



Security of hash functions

There are two approaches to attacking a secure hash function:

Cryptanalysis

- Exploit logical weaknesses in the algorithm

Brute-force attack

- Strength of hash function depends solely on the length of the hash code produced by the algorithm

SHA most widely used hash algorithm

Now available several versions… SHA-1, SHA-256, SHA-384, SHA-512

Additional secure hash function applications:

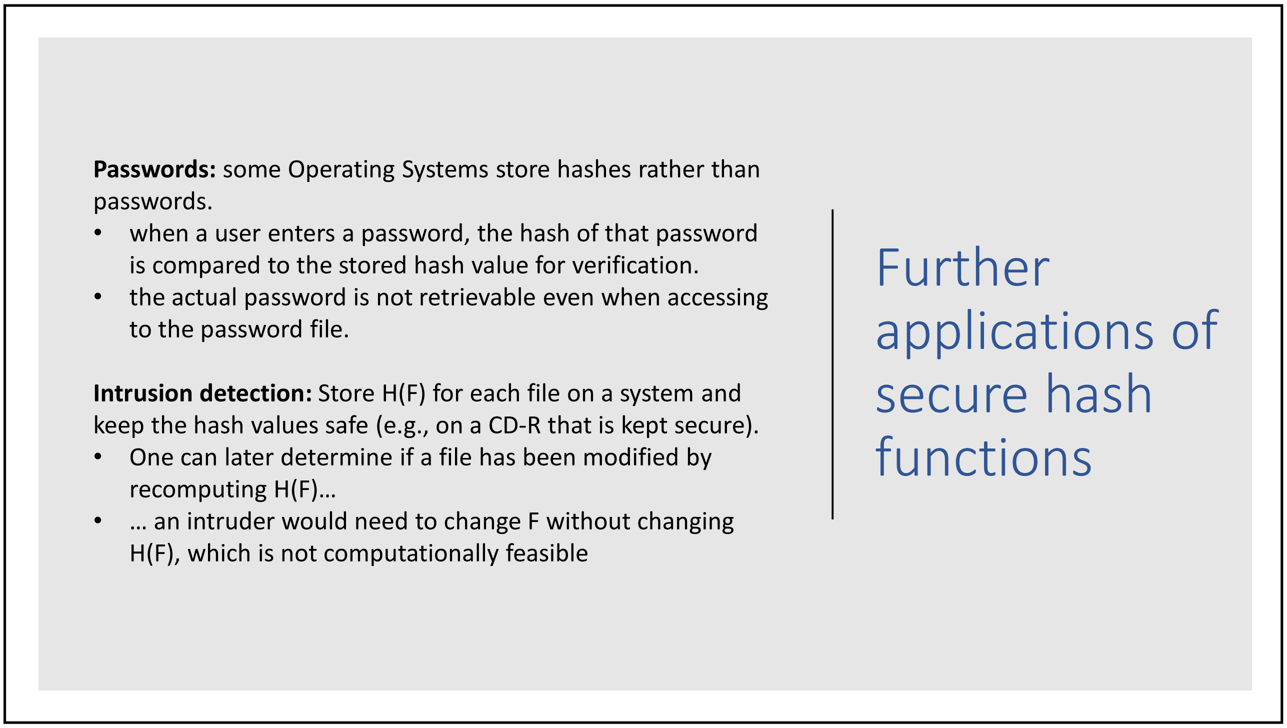
Passwords

- Hash of a password is stored by an operating system

Intrusion detection

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

39



Further applications of secure hash functions

Passwords: some Operating Systems store hashes rather than passwords.

- when a user enters a password, the hash of that password is compared to the stored hash value for verification.
- the actual password is not retrievable even when accessing to the password file.

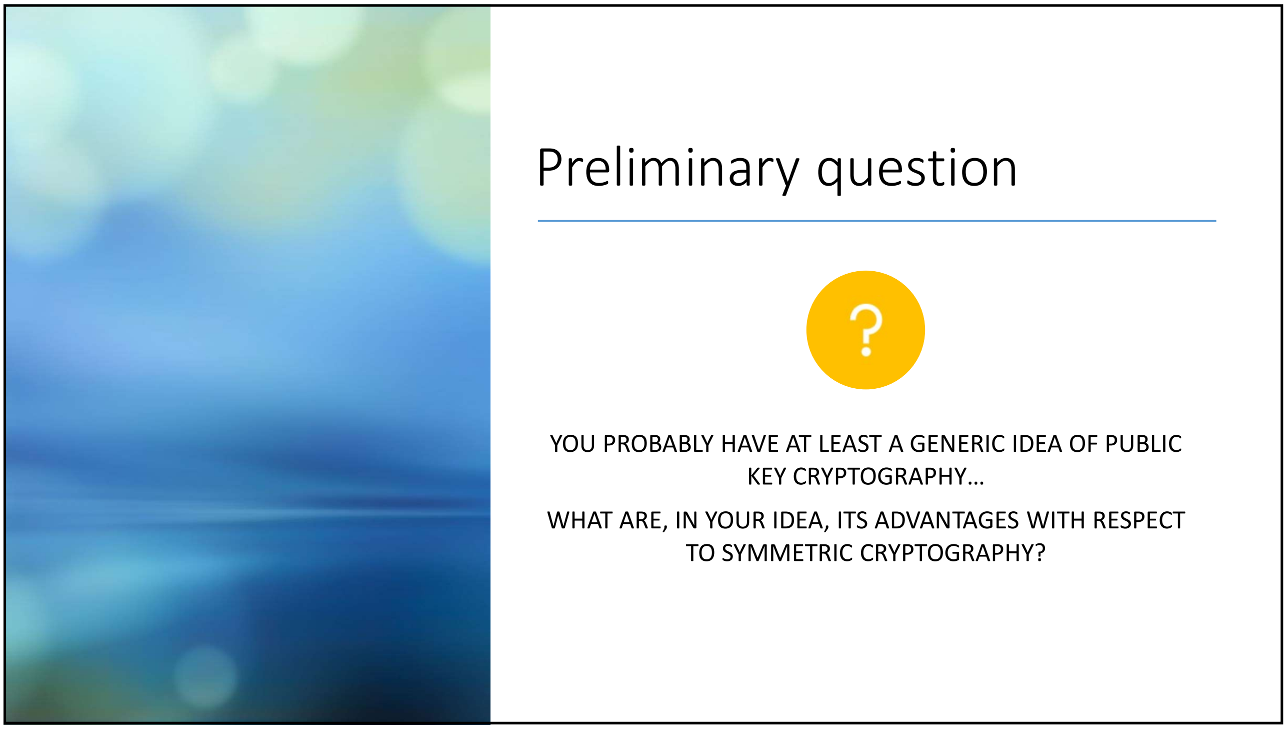
Intrusion detection: Store H(F) for each file on a system and keep the hash values safe (e.g., on a CD-R that is kept secure).

- 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), which is not computationally feasible

40

Public key encryption

41



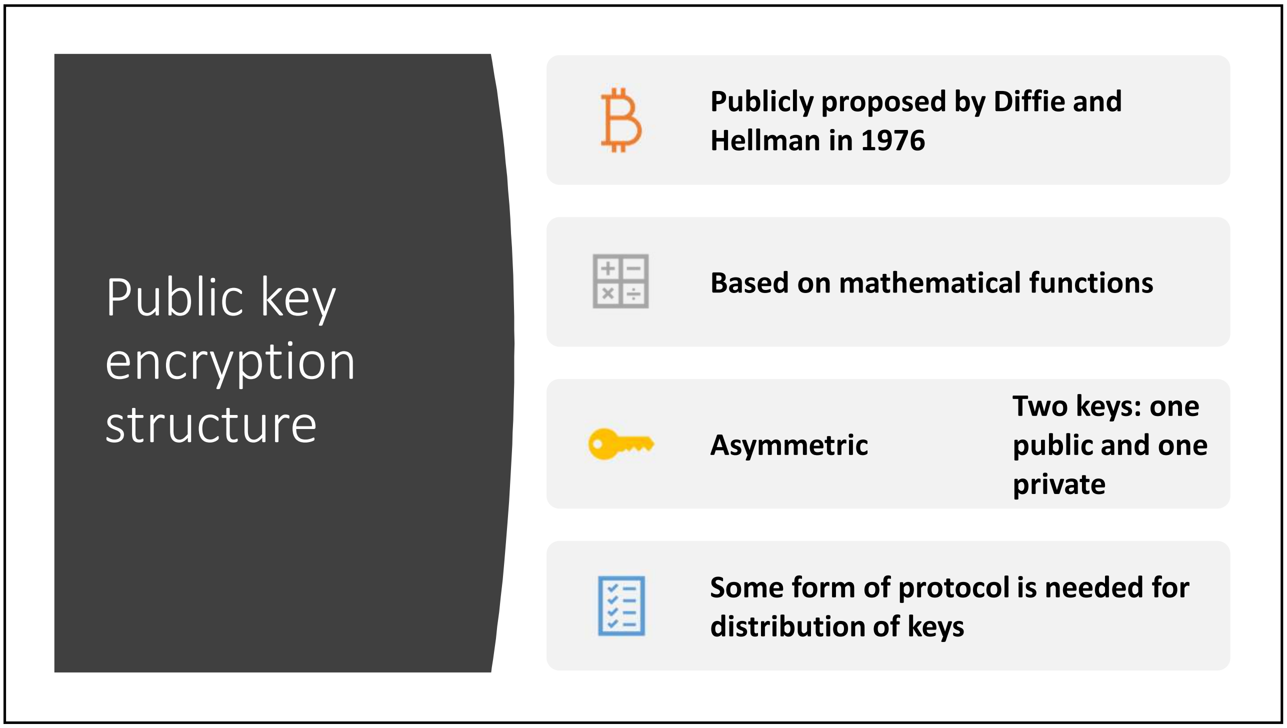
Preliminary question

?

YOU PROBABLY HAVE AT LEAST A GENERIC IDEA OF PUBLIC KEY CRYPTOGRAPHY…

WHAT ARE, IN YOUR IDEA, ITS ADVANTAGES WITH RESPECT TO SYMMETRIC CRYPTOGRAPHY?

42



Public key encryption structure

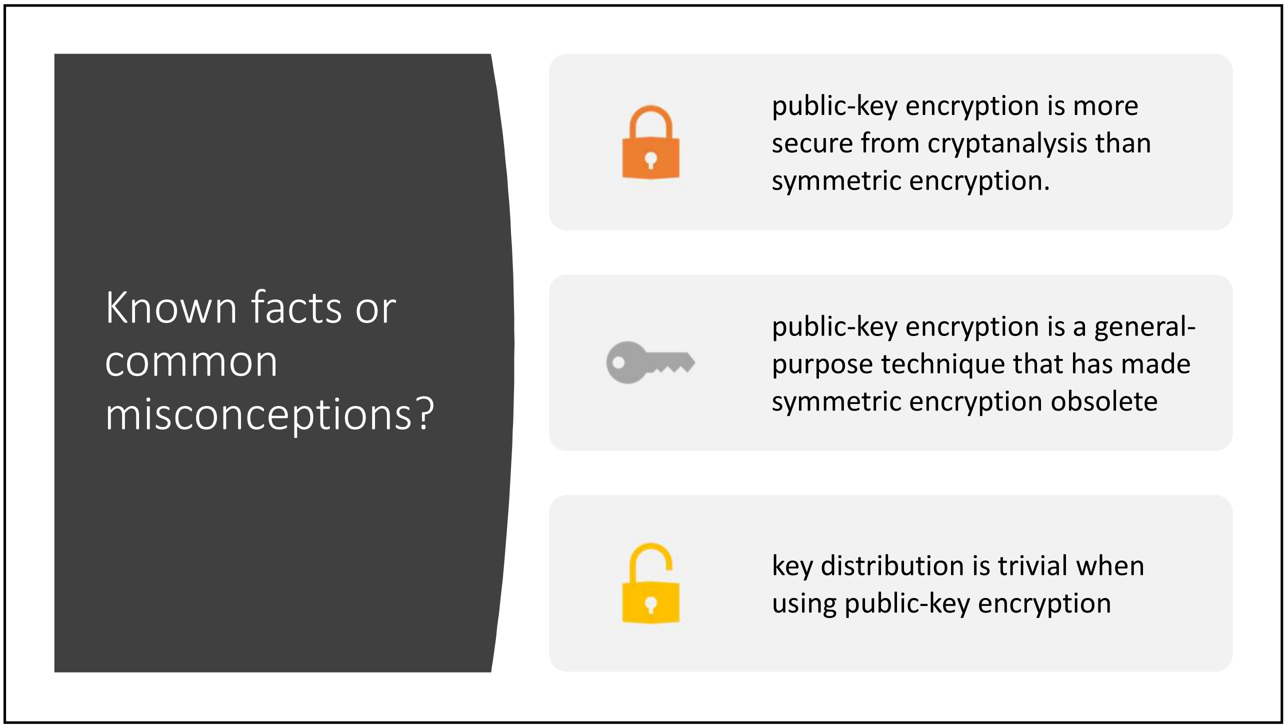
Publicly proposed by Diffie and Hellman in 1976

Based on mathematical functions

Asymmetric — **Two keys: one public and one private**

Some form of protocol is needed for distribution of keys

43



Known facts or common misconceptions?

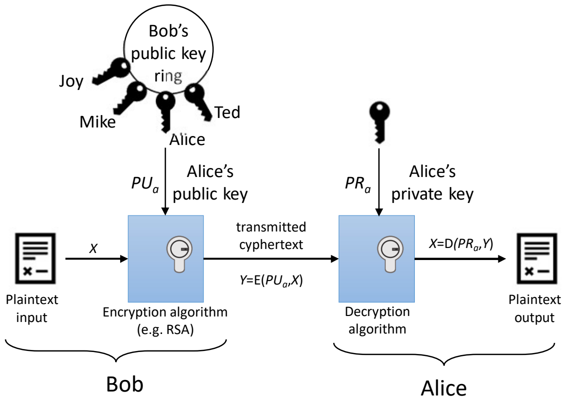
public-key encryption is more secure from cryptanalysis than symmetric encryption.

public-key encryption is a general-purpose technique that has made symmetric encryption obsolete

key distribution is trivial when using public-key encryption

44

A public-key encryption scheme

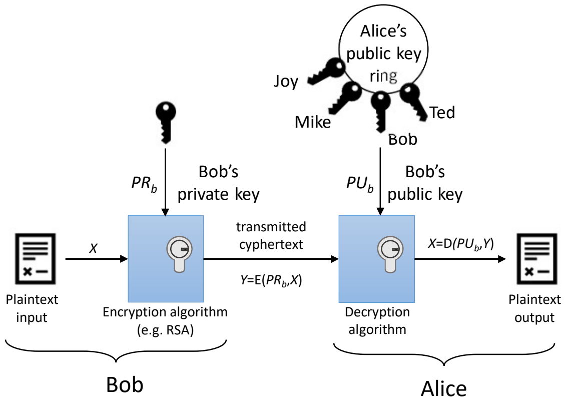


Encryption with public key

- Plaintext
  - Readable message or data that is fed into the algorithm as input
- Encryption algorithm
  - Performs transformations on the plaintext
- Public and private key
  - Pair of keys, one for encryption, one for decryption
- Ciphertext
  - Scrambled message produced as output
- Decryption key
  - Produces the original plaintext

45

Another public-key encryption scheme

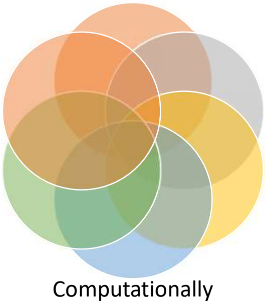


Encryption with private key

- User encrypts data using his or her own private key
- Anyone who knows the corresponding public key will be able to decrypt the message

46

Requirements for Public-Key Cryptosystems



Computationally easy to create key pairs

Computationally easy for sender knowing public key to encrypt messages

Computationally easy for receiver knowing private key to decrypt ciphertext

Computationally infeasible for opponent to determine private key from public key

Computationally infeasible for opponent to otherwise recover original message

Useful if either key can be used for each role

Asymmetric encryption algorithms

RSA (Rivest, Shamir, Adleman)

- Developed in 1977
- Most widely accepted and implemented approach to public-key encryption
- Block cipher in which the plaintext and ciphertext are integers between 0 and *n*-1 for some *n*.

Diffie-Hellman key exchange algorithm

- Enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages
- Limited to the exchange of the keys

Digital Signature Standard (DSS)

- Provides only a digital signature function with SHA-1
- Cannot be used for encryption or key exchange

Elliptic curve cryptography (ECC)

- Security like RSA, but with much smaller keys

Applications of public key cryptosystems

Algorithm	Digital signature	Symmetric key distribution	Encryption of secret keys
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	yes	yes	Yes

49

Digital Signatures

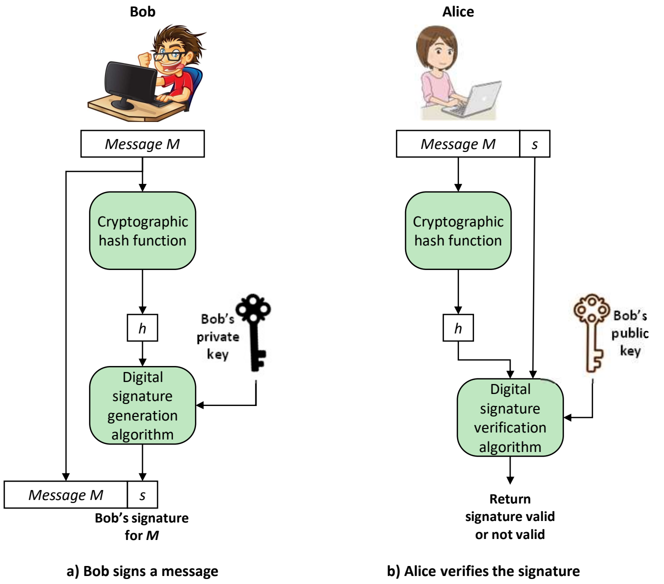
- NIST FIPS PUB 186-4 defines a digital signature as:

  > **”The result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity and signatory non-repudiation.”**

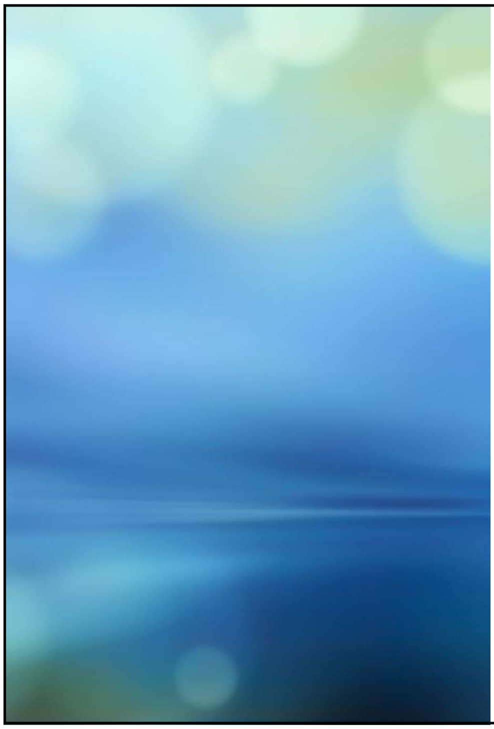
- Thus, a digital signature is a data-dependent bit pattern, generated by an agent as a function of a file, message, or other form of data block
- FIPS 186-4 specifies the use of one of three digital signature algorithms:
  - Digital Signature Algorithm (DSA)
  - RSA Digital Signature Algorithm
  - Elliptic Curve Digital Signature Algorithm (ECDSA)

50

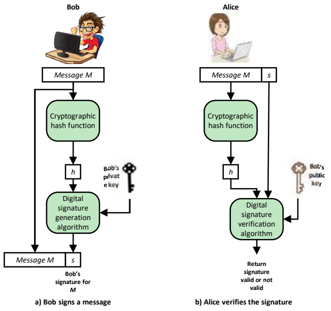
Essential elements of a digital signature process


Bob
Alice
Message M
Message M
s
Cryptographic hash function
Cryptographic hash function
h
h
Bob's private key
Bob's public key
Digital signature generation algorithm
Digital signature verification algorithm
Message M
s
Bob's signature for M
Return signature valid or not valid
a) Bob signs a message
b) Alice verifies the signature

51



Questions


Bob
Alice
Message M
Message M
s
Cryptographic hash function
Cryptographic hash function
h
h
Bob's privat e key
Bob's public key
Digital signature generation algorithm
Digital signature verification algorithm
Message M
s
Bob's signature for M
Return signature valid or not valid
a) Bob signs a message
b) Alice verifies the signature

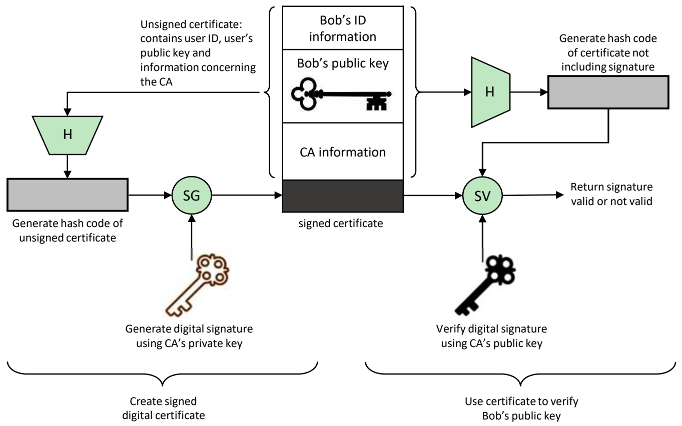
THE PREVIOUS SCHEMA:

1. provides confidentiality?
2. is Alice sure the message is from Bob?
3. is Alice sure the message has not been altered?

52

Public key certificate use

… ok, you built your public/private key pair and now?
How do you make sure your announcement of public key is not forged?


Unsigned certificate: contains user ID, user's public key and information concerning the CA
Bob's ID information
Bob's public key
CA information
H
Generate hash code of unsigned certificate
SG
signed certificate
Generate digital signature using CA's private key
Create signed digital certificate
H
Generate hash code of certificate not including signature
SV
Return signature valid or not valid
Verify digital signature using CA's public key
Use certificate to verify Bob's public key

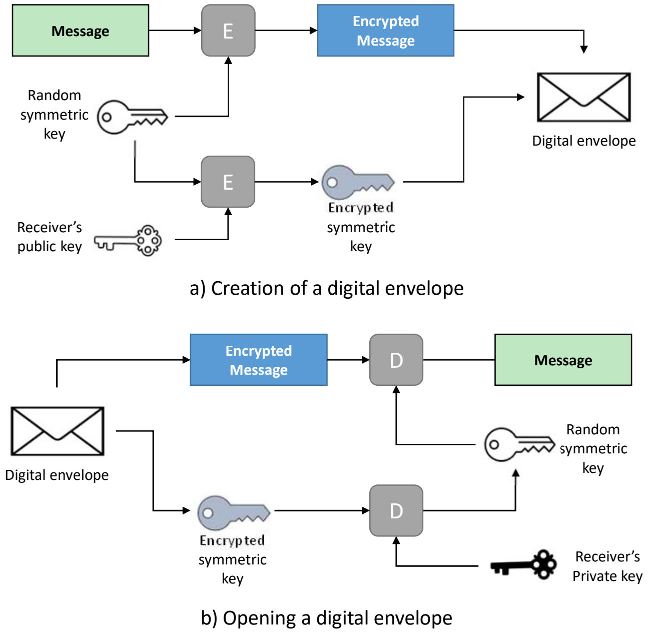
CA: certificate authority that is trusted by the user community

- government agency, financial institution, …

53

Digital envelopes

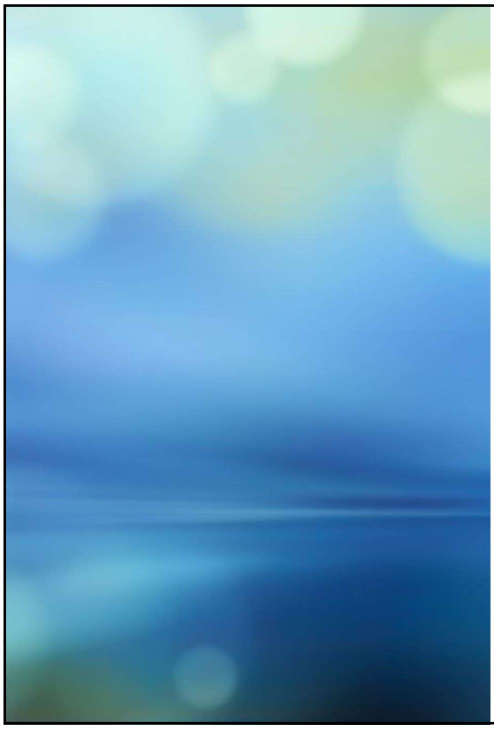
A way to encrypt a message without needing to first arrange for sender and receiver to have the same secret key



a) Creation of a digital envelope

b) Opening a digital envelope

54



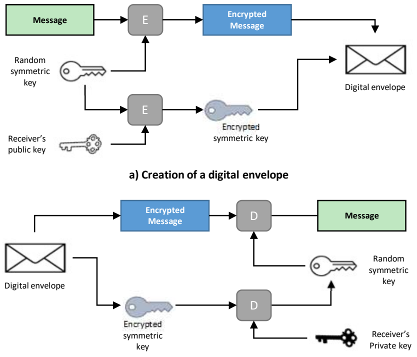
Problem

THE PREVIOUS SCHEMA (DIGITAL ENVELOPE) DOES NOT GUARANTEE THE AUTHENTICATION OF THE SENDER.

CAN YOU IMPROVE THE SCHEMA TO INCLUDE AUTHENTICATION?

55

Add authentication to digital envelopes



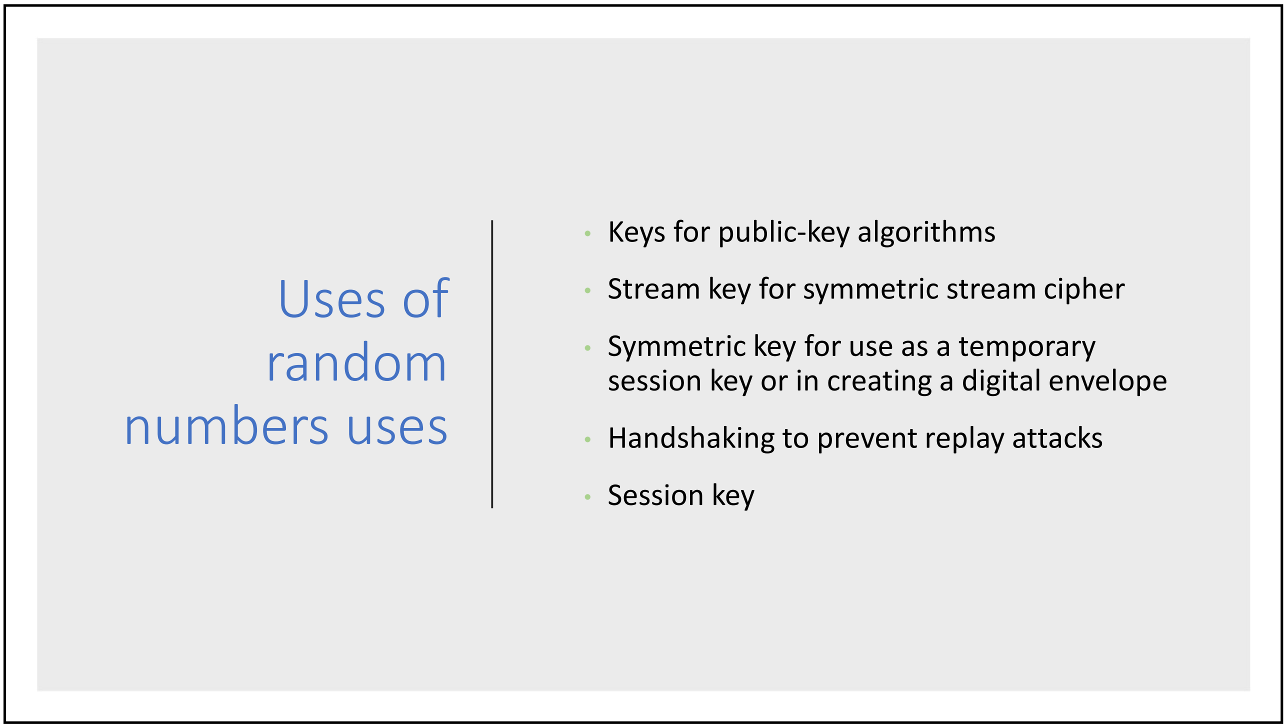
a) Creation of a digital envelope

b) Opening a digital envelope

56

Random numbers

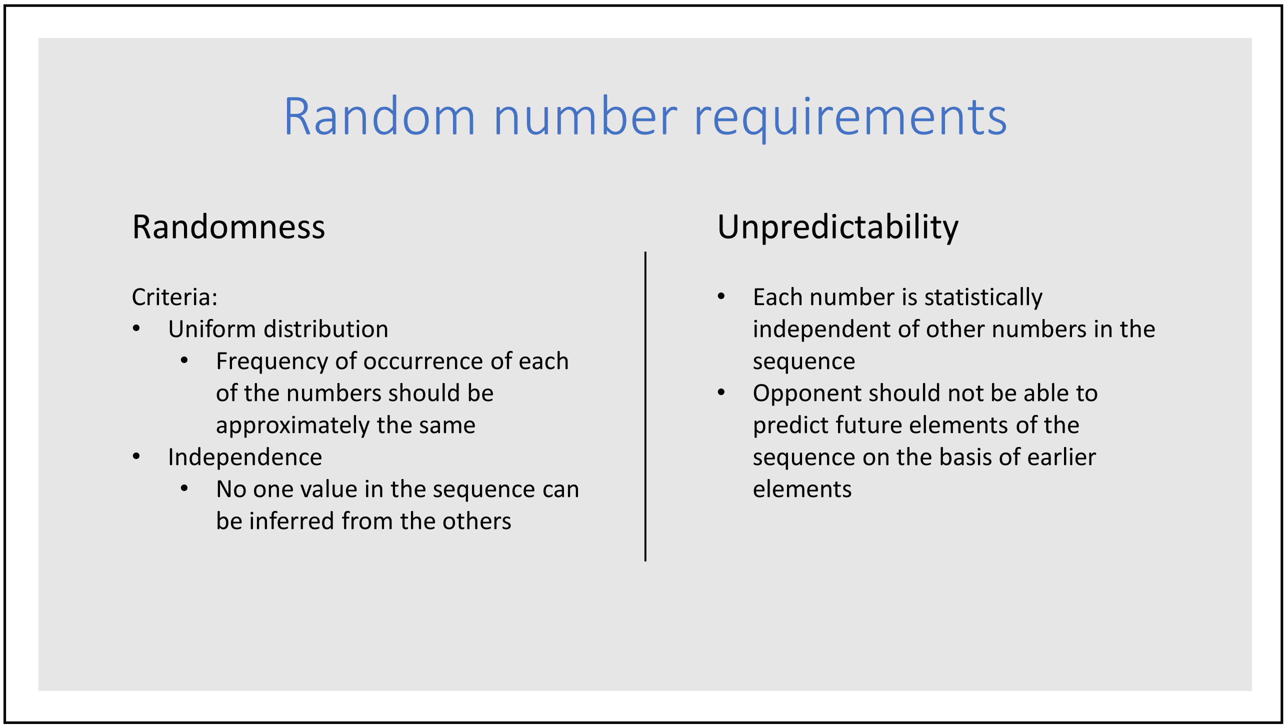
59



Uses of random numbers uses

- Keys for public-key algorithms
- Stream key for symmetric stream cipher
- Symmetric key for use as a temporary session key or in creating a digital envelope
- Handshaking to prevent replay attacks
- Session key

60



Random number requirements

Randomness

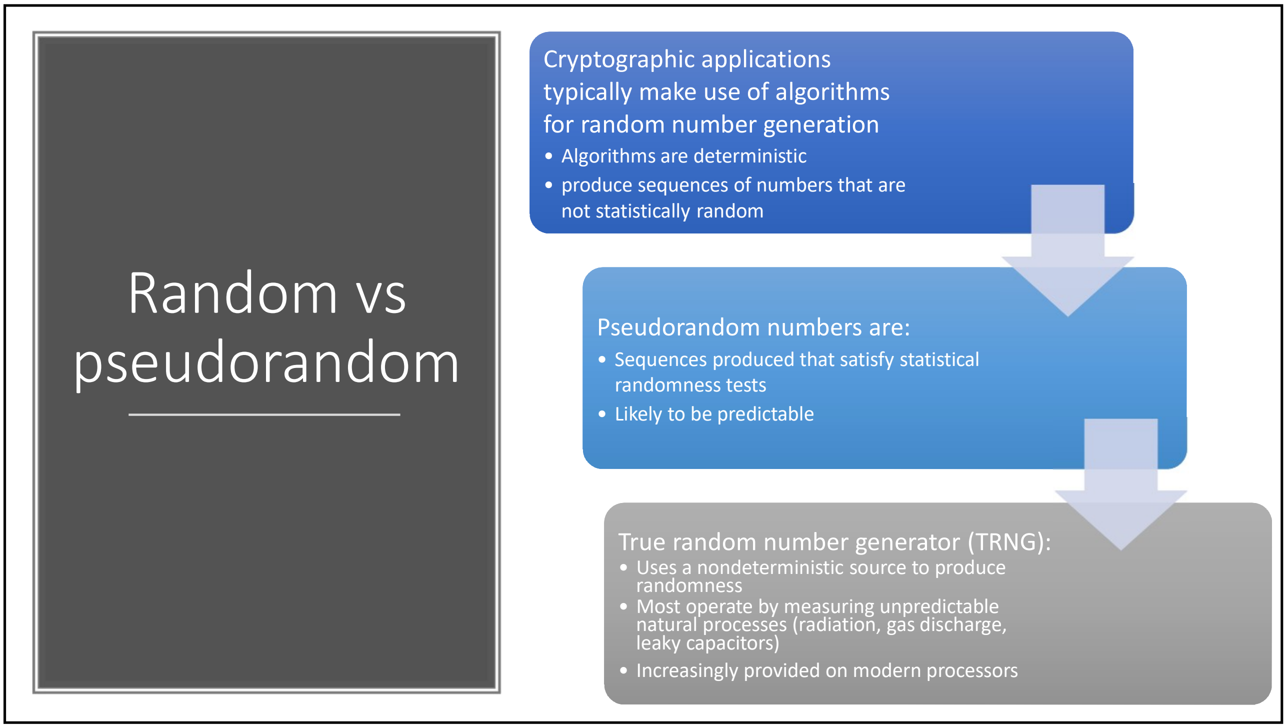
Criteria:

- Uniform distribution
  - Frequency of occurrence of each of the numbers should be approximately the same
- Independence
  - No one value in the sequence can be inferred from the others

Unpredictability

- Each number is statistically independent of other numbers in the sequence
- Opponent should not be able to predict future elements of the sequence on the basis of earlier elements

61



Random vs pseudorandom

Cryptographic applications typically make use of algorithms for random number generation

- Algorithms are deterministic
- produce sequences of numbers that are not statistically random

Pseudorandom numbers are:

- Sequences produced that satisfy statistical randomness tests
- Likely to be predictable

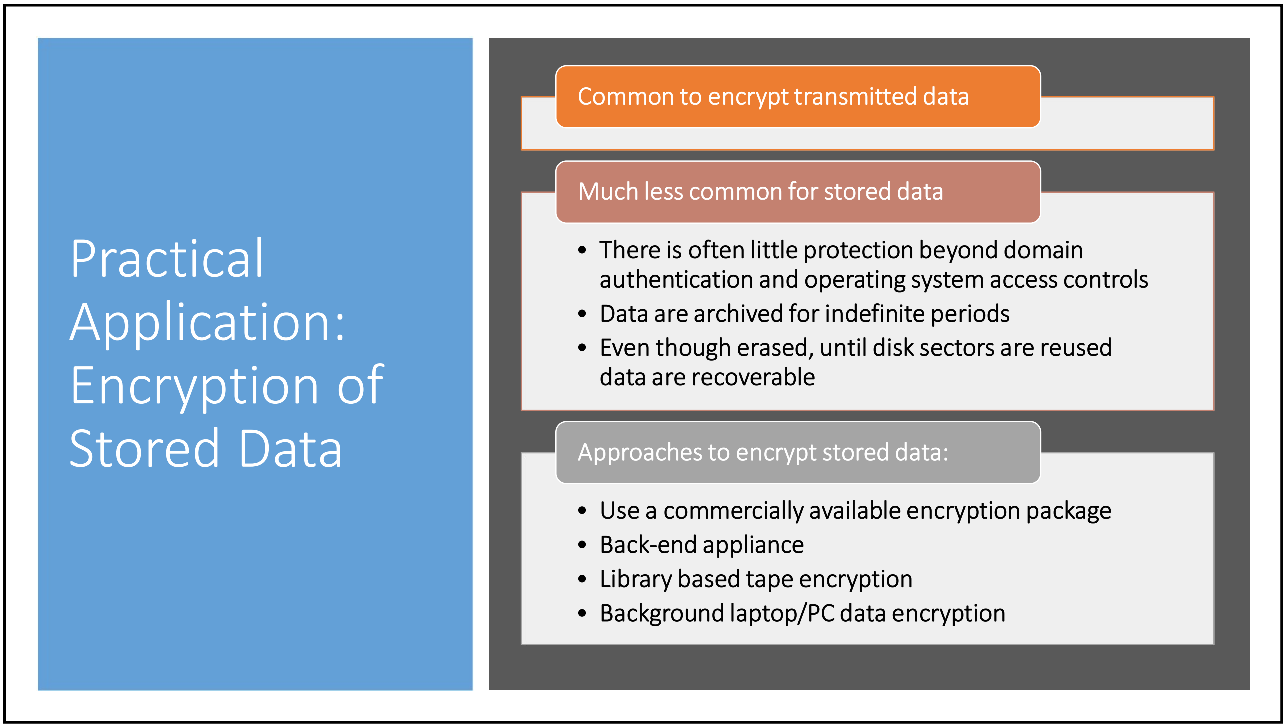
True random number generator (TRNG):

- Uses a nondeterministic source to produce randomness
- Most operate by measuring unpredictable natural processes (radiation, gas discharge, leaky capacitors)
- Increasingly provided on modern processors

62

Conclusions

63



Practical Application: Encryption of Stored Data

Common to encrypt transmitted data

Much less common for stored data

- There is often little protection beyond domain authentication and operating system access controls
- Data are archived for indefinite periods
- Even though erased, until disk sectors are reused data are recoverable

Approaches to encrypt stored data:

- Use a commercially available encryption package
- Back-end appliance
- Library based tape encryption
- Background laptop/PC data encryption

64

Summary

- Confidentiality with symmetric encryption
  - Symmetric encryption
  - Symmetric block encryption algorithms
  - Stream ciphers
- Message authentication and hash functions
  - Authentication using symmetric encryption
  - Authentication without message encryption
  - Secure hash functions
  - Other applications of hash functions
- Public-key encryption
  - Structure
  - Applications & Requirements
  - Asymmetric encryption
- Digital signatures and key management
  - Digital signature
  - Public-key certificates
  - Symmetric key exchange using public-key encryption
  - Digital envelopes
- Random and pseudorandom numbers
  - The use of random numbers
  - Random versus pseudorandom
- Practical Application: Encryption of Stored Data

65

Exercise 1

Draw an attack tree for gaining access to a (physical) office.

You have 10 minutes to make it before I'll show my solution
In these minutes I'm available to answer to your questions.

66

Exercise 1 - Draw an attack tree for gaining access to an office

Exercise 2

$H(\cdot)$ is a cryptographic hash function that maps a message of an arbitrary bit length on to a 20-bit hash value.

a) How many random messages would be required to find two different messages M and M' such that $H(M) = H(M')$?

b) What is the probability that none of n randomly generated messages collide?

You have 10 minutes to make it before I'll show my solution
In these minutes I'm available to answer to your questions.

Solution 2a

$H(\cdot)$ is a cryptographic hash function that maps a message of an arbitrary bit length on to a 20-bit hash value.
How many random messages would be required to find two different messages M and M' such that $H(M) = H(M')$?

There are ______________ different hash values

Hence, in the very worst case, it will be necessary ___________messages

Say that you have generated n different messages without success (i.e. each producing a different value of the hash).

The probability that the $n + 1$ generated message
collide with one of the previous n messages (provided they all have different hashes) is: ______

When $n = 2^{10}$ what this probability is: _____________

70

Solution 2a

$H(\cdot)$ is a cryptographic hash function that maps a message of an arbitrary bit length on to a 20-bit hash value.
What is the probability that none of n randomly generated messages collide?

[redacted]

71

Solution 2b

$H(\cdot)$ is a cryptographic hash function that maps a message of an arbitrary bit length on to a 20-bit hash value.
What is the probability that none of n randomly generated messages collide?

Hint: by induction.

Let $p(n)$ be the probability that there's no collision among n randomly generated messages. Hence:

- $p(1) =$______________________
- $p(2) =$______________________
- $p(n) =$______________________

72

Solution 2b

$p(n)$

[redacted]

73

Exercise 3

Alice and Bob are organizing a dinner. Alice is cooking at home and Bob is buying food. To enforce the security in her communications, Alice is appending to her messages a MAC.
However, Alice doesn’t expect that her messages can be overheard and tampered by Eve.
Alice sends Bob the message M=“buy a gallon of water”, with the message authentication code MAC(K,M).

1. Eve intercepts the message and sends to Bob a copy of the same message. How much water will Bob buy? Explain why.
2. Eve intercepts the message and changes “water” with “wine”… will Alice and Bob get drunk tonight?
3. Bob likes very much beer, so he buys beer instead of water and he pretends that the message was M’=“buy a gallon of beer”. Is Alice able to prove that the message has been forged by Bob?

Solution 3

Alice sends Bob the message M=“buy a gallon of water”, with the message authentication code MAC(K,M).

a)

b)

c)

Exercise 4

Same as before, but this time Alice is using a digital signature:
Alice and Bob are organizing a dinner. Alice is cooking at home and Bob is buying food. To enforce the security in her communications, Alice is appending to her messages her digital signature computed with her private key KA.
However, Alice doesn't expect that her messages can be overheard and tampered by Eve.
Alice sends Bob the message M="buy a gallon of water", with the digital signature DS(M,KA).

1. Eve intercepts the message and sends to Bob a copy of the same message. How much water will Bob buy? Explain why.
2. Eve intercepts the message and changes "water" with "wine"… will Alice and Bob get drunk tonight?
3. Bob likes very much beer, so he buys beer instead of water and he pretends that the message was M'="buy a gallon of beer". Is Alice able to prove that the message has/has not been forged by Bob?

76

Alice sends Bob the message M="buy a gallon of water", with the digital signature DS(M,KA).

a)

b)

Solution 4

c)

77