

## Symmetric Key Cryptosystems

- Based on the sender and the receiver of a message knowing and using
  - The same (secret) key
  - Cryptosystem
- Sender uses the secret key to encrypt the message
  - Receiver uses the same secret key to decrypt the message
- Key management
  - Main problem is the sender and receiver to agree on a secret key without anyone else finding out
  - One of the fundamental issues that has to be addressed in symmetric key cryptosystems
  - In a network of  $n$  nodes the total number of different symmetric keys needs to be at least  $\frac{n(n-1)}{2}$  (assuming that each pair of nodes share a common key)
- Examples of symmetric key algorithms are
  - DES, Triple DES, AES (Rijndael), IDEA

1

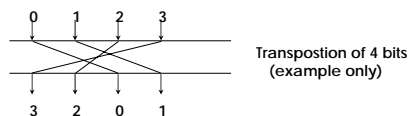
## Data Encryption Standard (DES)

- In January 1977 a standard encryption method was adopted by the U.S. government
  - Its origins lay in an internal IBM project codenamed *Lucifer* to develop a cryptographic algorithm
- Though the algorithm used is complex
  - It is easily implemented in hardware
  - Software implementations are also widely available
- DES is a *block cipher*
  - Operates on a single chunk of data at a time
    - \* 64 bits (8 bytes)
  - Produces a 64 bit output
- The key length is 56 bits
  - Has outlived its usefulness, too small key size and block size
  - Survives in the form of 3DES (three DES encryptions in sequence)
  - This fixes the small key size problem, but not the small block size

2

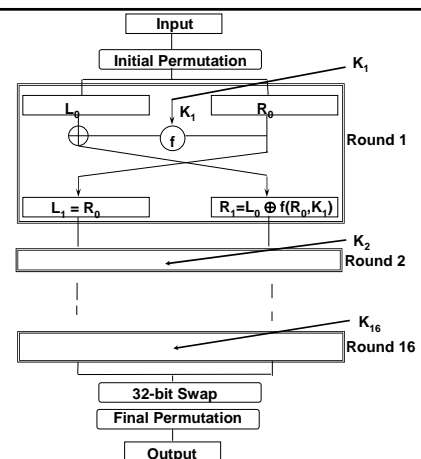
## DES Operation

- The algorithm has 19 distinct stages
- The first stage reorders the bits of the 64-bit input block by applying a fixed permutation (nobody seems to know why, no cryptographic effect)



- The last stage is the exact inverse of this permutation
- The stage penultimate to the last one
  - Exchanges the leftmost 32 bits with the rightmost 32 bits
- The remaining 16 stages are called *rounds*
  - Functionally identical but take as an input a quantity computed from the round key and the round number
  - Round key: 48 bits from the 56-bit key, different at each round

3



4

## Cracking DES

- 56 bits is a short key
- *Brute force attack* (try every key)
  - $2^{56}$  encryptions to try all keys
  - Special chips can check 4 million keys/second
  - \$1 million dollar DES cracking machine could break it in a few hours
- June 18 '97, DESCHALL group
  - Used spare Internet CPUs
  - 4 months, 18 quadrillion keys (25%), 78,000 computers

5

## Stream Encipherment

- The above method of operating DES is called Electronic Code Book (ECB) mode
- It does not protect against replay attacks and other attacks that occur on streams of information
- For data communications applications
  - Would like each fragment to depend on what had preceded it
  - One simple way to convert DES into a stream cipher is to XOR plaintext block N with ciphertext block N-1 before it is encrypted - Cipher Block Chaining (CBC) mode

6

## Advanced Encryption Standard (AES)

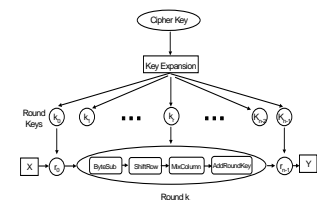
- In 1997 the U.S. government announced a call for proposals to develop a new Advanced Encryption Standard
  - After a long process 5 algorithms were selected as finalists (out of 15)
- Rijndael was the algorithm that was eventually chosen as the new AES
  - Symmetric cipher with variable key and block sizes of 128, 192 and 256 bits
  - Support for fast encryption and decryption in software (700 Mbps)
  - Can be implemented efficiently in small 8-bit devices e.g. smartcards
  - Serpent (AES finalist) much more secure, the most conservative finalist, about one-third the speed of Rijndael

\* Designed for security instead of elegance and efficiency

7

## AES Overall Structure

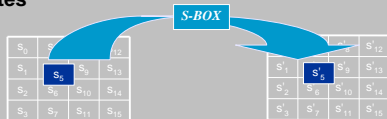
- The cipher consists of between 10 or 14 rounds ( $N_r$ )
  - Depending on the key length ( $N_k$ ) and the block length ( $N_b$ )
- A plaintext block  $X$  undergoes  $n$  rounds of operations to produce an output block  $Y$ 
  - Each operation is based on the value of the  $n$ th round key
- The round keys are derived from the cipher key
  - By first expanding the key and then selecting parts of the expanded key for each round



8

## AES Round (1/2)

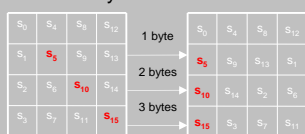
### SubBytes



state array

### ShiftRows

state array



9

## AES Round (2/2)

### MixColumns

coeff.s matrix

state array

$$\begin{bmatrix} s'_0 & s'_4 & s'_8 & s'_{12} \\ s'_1 & s'_5 & s'_9 & s'_{13} \\ s'_2 & s'_6 & s'_{10} & s'_{14} \\ s'_3 & s'_7 & s'_{11} & s'_{15} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \otimes \begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{bmatrix}$$

### AddRoundKey

state array

round key

$$\begin{bmatrix} s'_0 & s'_4 & s'_8 & s'_{12} \\ s'_1 & s'_5 & s'_9 & s'_{13} \\ s'_2 & s'_6 & s'_{10} & s'_{14} \\ s'_3 & s'_7 & s'_{11} & s'_{15} \end{bmatrix} = \begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{bmatrix} \oplus \begin{bmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{bmatrix}$$

10

## AES Criticism

- Simple algebraic structure
  - Possible to write an AES encryption as a formula over the finite field with 256 elements
  - Not an attack but a representation
  - However, it is a new *avenue* of attack
- For 128-bit keys the best attack we know of covers 70% of the cipher
- AES relies on the hope that future attacks will not give large improvements

11

## Snake Oil

- Snake oil from commercial enterprises is quite common in this area
  - Usually they claim to have developed a revolutionary new algorithm
- Why should you not try to develop your own encryption algorithm?
- How should you debunk such claims?

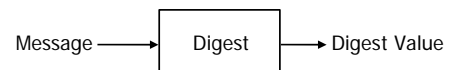
12

### Message Digest Functions (1/3)

- Also known as cryptographic hashes
- Non-reversible functions
- Take an arbitrary size message and mangle it into a fixed size digest
- It should be impossible to find two messages with the same MD, or come up with a message with a given MD

13

### Message Digest Functions (2/3)



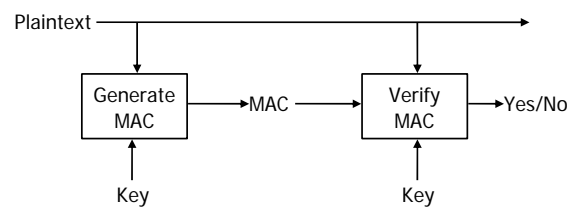
14

### Message Digest Functions (3/3)

- MD2, MD4, and MD5 used to be most popular; SHA-1 next most popular (until a few months ago)
- All produce 128 bit digests
- MD4 and MD2 were recently ``broken`` and MD5 has significant weaknesses
- SHA-1 was proposed by the U.S. government; it produces a 160 bit digest
- Practical collision search attacks very likely to exist according to very recent research
- Message digests are not difficult to design, but most are not secure
- So, for the time being use SHA-256

15

### Secret Key Integrity Protection

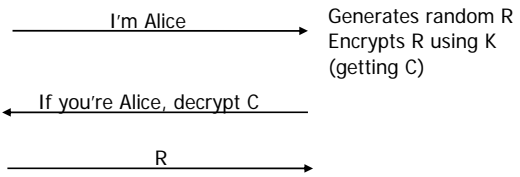


16

## Challenge / Response Authentication

Alice (knows K)

Bob (knows K)



17