L93 - Secret-Key Cryptography

Secret-Key Cryptography

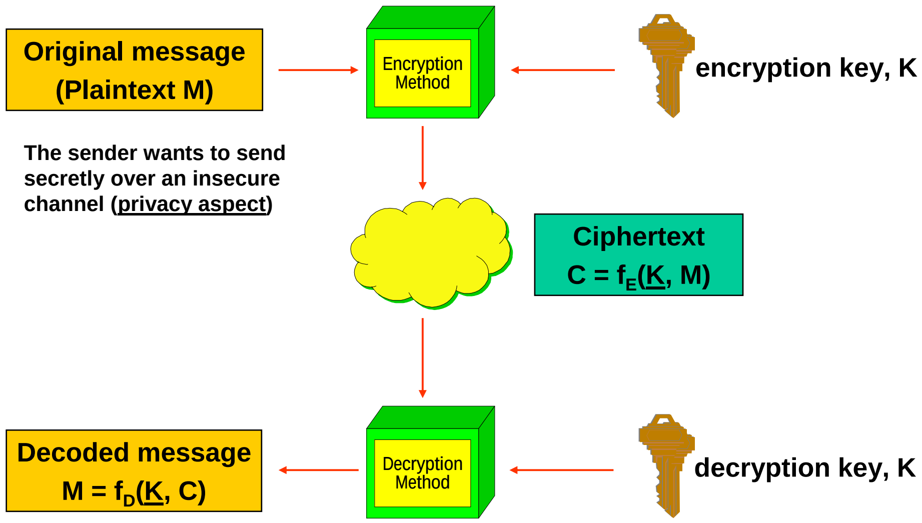
DES, 3DES, IDEA, AES

Agenda

- Introduction
- DES
- 3DES
- DES-Modes
- IDEA
- RC4
- AES

 Secret-Key, v4.3 2

Secret-Key Technique



Secret-Key Techniques

- **Examples**
  - Data Encryption Standard (DES, 56bit)
  - Multiple Encryption DES (3DES, 112bit)
  - International Data Encryption Algorithm (IDEA, 128bit)
  - RC4, RC5
  - Advanced Encryption Standard (AES, 128/168/256 bit))
- **Encrypting large messages**
  - Electronic Code Block (ECB)
  - Cipher Block Chaining (CBC)
  - Output Feedback Mode (OFB)
  - Cipher Feedback Mode (CFB)

Agenda

- Introduction
- DES
- 3DES
- DES-Modes
- IDEA
- RC4
- AES

DES

- **History**
  - designed and developed by IBM
  - published 1977 by NIST (National Institute of Standards and Technology) as official standard for unclassified information
    - lot of US government regulations refer to DES
  - widely adopted by the industry for use in security products
- **Scrutinized by cryptanalysts**
  - for 25 years with no significant flaw found
- **Simple logical operations**
  - can be easily implemented in hardware
  - very high speed, up to gigabit/s (!) with special chips

DES Algorithm 1

- **Description of the DES algorithm**
  - a sequence of permutations and substitutions based on the encryption key
  - 64-bit block cipher
    - encrypts 64-bit of plaintext resulting in 64-bit of ciphertext
  - 56-bit key
    - the same key is used for encryption and decryption
  - steps
    - initial and final permutation
      - has nothing to do with security
      - make DES less efficient to implement in SW
      - in software implementations sometimes ignored

 Secret-Key, v4.3 7

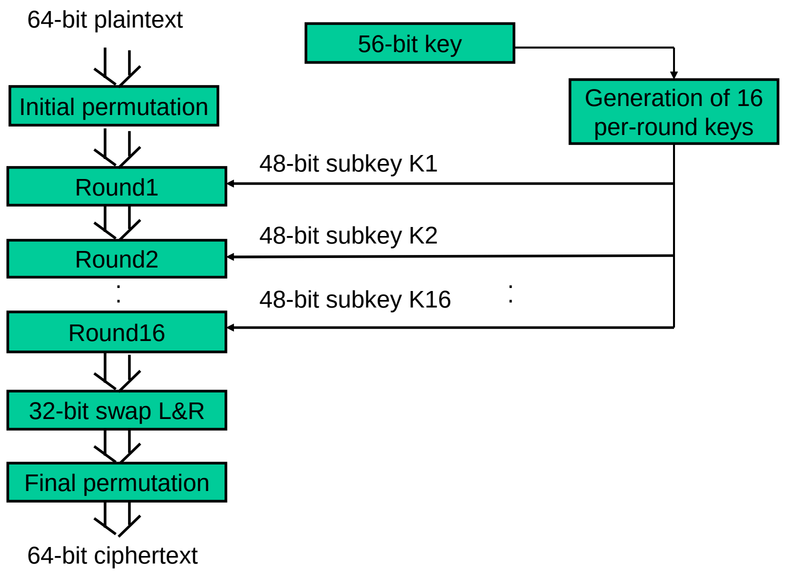
DES Algorithm 2

- steps (cont.)
  - key transformation
    - initial permutation of 56-bit key, then partitioning in two 28-bit units, every unit is rotated left by the number of round
    - subkey Ki (i = number of round) is derived applying final permutation
    - resulting in 16 subkeys K1 - K16
  - for every round the corresponding subkey is used (K1, K2, … K16)
  - round
    - 32 input left, 32 input right
    - input right becomes output left
    - output right is EXORed of input left and a function of input right and subkey Ki
    - complexity lies in this function (expansion permutation, EXORed with Ki, given to S-box substitutions, final P-box permutation)
- decryption done by same procedure

  subkeys must be used in reverse order (K16, K15, …. K1)

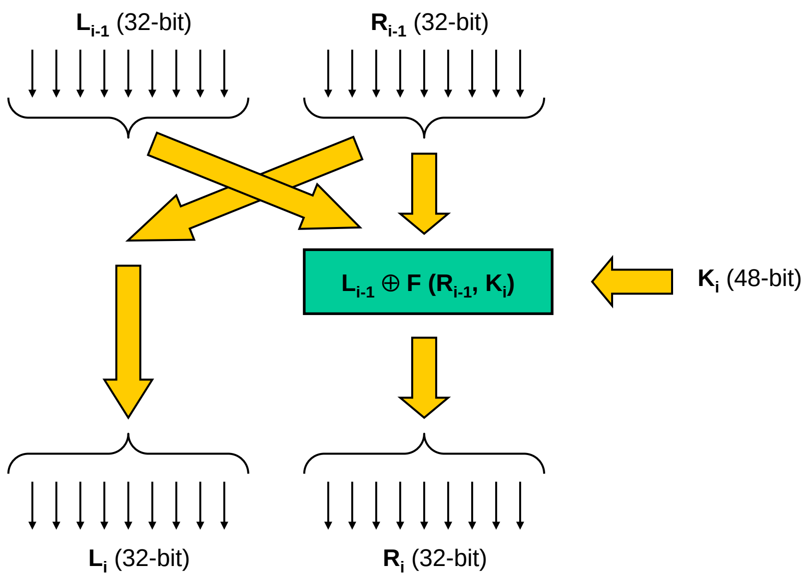
 Secret-Key, v4.3 8

DES Algorithm 3



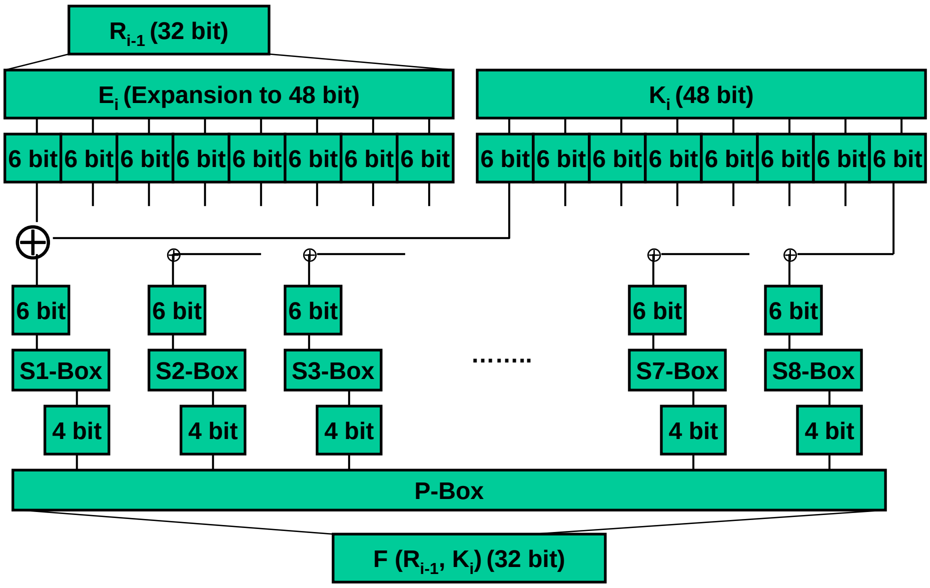
 Secret-Key, v4.3 9

DES Round (i)



 Secret-Key, v4.3 10

DES Round (i) Function F



Security of DES 1

- **Standardization and Design**
  - originally IBM specified key length 128 bit
  - after invitation to discuss this matter with NSA (National Security Agency) it was reduced to 56 bit
  - design process (especially S-boxes) was kept secret
  - there are some “rumors” about these facts
- **Cryptanalyst**
  - tried out a lot of methods to break it
  - actually in most cases only brute-force is the danger
- **Conclusion:**
  - the algorithm is very good and still considered to be very robust, but the key length is not

Security of DES 2

- **Key length issues**
  - originally 56 bit
  - in 1977 Diffie and Hellmann designed a machine to break DES by brute-force attack
    - estimated cost 20Mill $, successful break in 12 hours
  - cost / time to break depending on key-length in 1996
    - 40-bit (10Mill$ / 0.02 sec, 10k$ / 12 min, 400$ / 5 hours)
    - 56-bit (10Mill$ / 21 min, 10k$ / 556 days, 400$ / 38 years)
    - 168-bit (10Mill$ / 10^{17} years, 10k$ / 10^{19} years, 400$ / too long)
  - in 1998 EFF built a special-purpose engine
    - DES Cracker for 250k$ finding key in 4.5 days
  - in 1996 minimal recommended key length was 90 bits to provide security through 2016, in 2000 128 bit is considered as good key length

Agenda

- **Introduction**
- **DES**
- **<u>3DES</u>**
- **DES-Modes**
- **IDEA**
- **RC4**
- **AES**

How to improve DES

- **Increase key length to 112 bits**
  - 2^{112} ($5x10^{33}$) possible keys to try out by brute-force attack instead of 2^{56} ($7x10^{16}$)
  - seems to be sufficient for the next 100 million years
- **Ideas to implement this**
  - by running DES twice with two different 56 bit keys
    - but Cryptanalyst developed a method that makes double encryption suspect and it turned out, that double encryption is not much more secure than single encryption
  - Triple Encryption (3DES, 112 bit)
    - three stages: first DES encrypt with K1 (56bit), then DES decrypt with K2 (56bit) and finally encrypt with K1 again (EDE) hence slower than single DES, 2 keys (112bit) are seen as save enough,
    - EDE allows backward compatibility with single DES when K1 = K2

Agenda

- **Introduction**
- **DES**
- **3DES**
- **DES-Modes**
- **IDEA**
- **RC4**
- **AES**

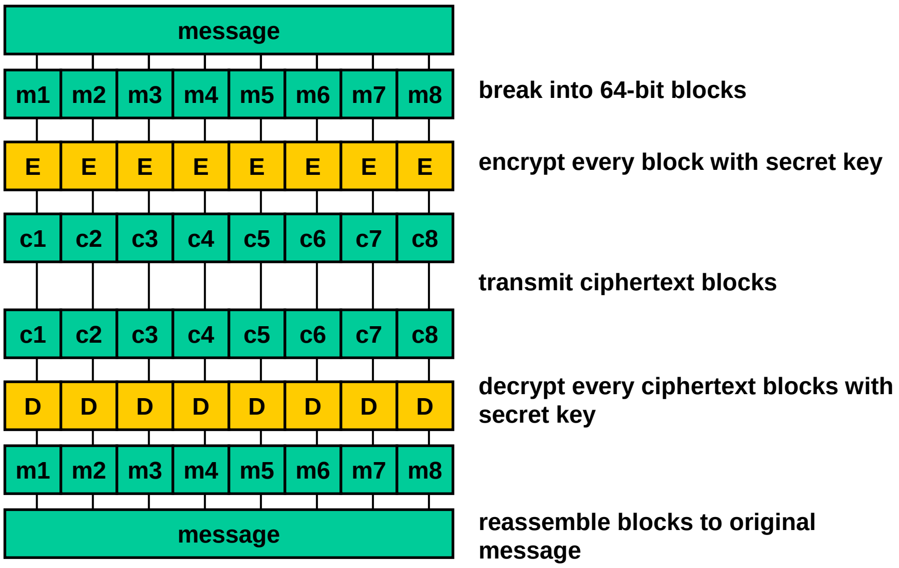
DES Modes Overview

- **DES**
  - is basically a mono alphabetic substitution cipher using 64-bit character that means whenever the same 64-bit plaintext is encrypted the same 64-bit ciphertext will result
- **For encryption of larger messages than 64-bits**
  - block cipher
  - ECB - Electronic Codebook Mode
  - CBC - Cipher Block Chaining
- **For encryption of messages less than 64-bits**
  - stream cipher
  - CFB - Cipher Feedback
  - OFB - Output Feedback

DES Mode - ECB 1

- **ECB - Electronic Codebook Mode**
  - message is broken into 64-bit blocks, padding the last one to full 64-bits
  - every block is encrypted with the secret key

DES Mode - ECB 2



DES Mode - ECB 3

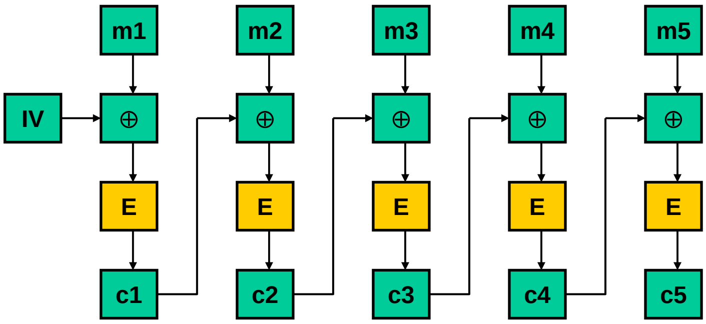
- **ECB - Electronic Codebook Mode (cont.)**
  - problems which do not show up in the single-block case
    - if message contains two identical blocks the ciphertext will be identical
    - this can exploited by a cryptanalyst to help breaking DES
    - this can be misused by an eavesdropper by gaining information from repeated blocks
    - this can be misused by an active intruder by rearranging blocks or modifying blocks to his own advantage
      - remove, repeat (replay attack), or interchange blocks at will
    - vulnerable to insertion, replay and dictionary attack

DES Mode - CBC 1

- **CBC - Cipher Block Chaining**
  - a method avoiding some of the problems of ECB
    - by avoiding that two identical blocks of plaintext will result in the same ciphertext
    - this makes cryptanalysis for breaking DES more difficult
  - basic idea: introduce random numbers into ECB
  - problem: how to get the same numbers for decryption
  - solution: add a feedback
    - plaintext is EXORed with the previous ciphertext block before encryption,
    - initialization vector (IV) for the first block
      - random data to avoid block replay
    - IV must be given to the receiver before starting decryption

 Secret-Key, v4.3 21

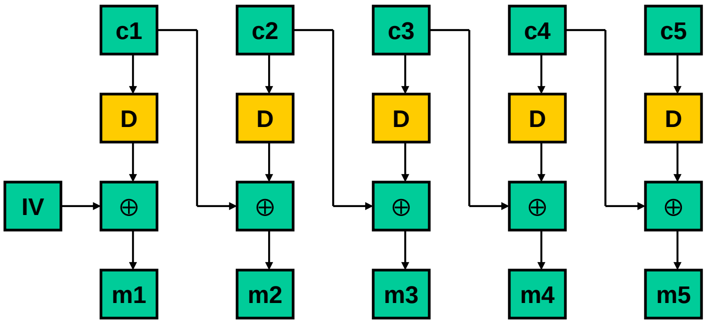
DES Mode - CBC 2



Encryption with CBC

 Secret-Key, v4.3 22

DES Mode - CBC 3



Decryption with CBC

DES Mode - CBC 4

- **CBC - Cipher Block Chaining (cont.)**
  - does not eliminate the problem of someone modifying the message in transit
    - eavesdropper can no longer seen repeated values
      - this makes cryptanalysis more difficult
    - eavesdropper can no longer simple copy or move ciphertext blocks
      - but he can still modify the ciphertext by altering ciphertext bits
    - modification lead to change in the next block
    - modification lead to garbage in the same block
      - but what if not recognized or controlled by a program when decrypted
  - general solution
    - include a 64-bit CRC at the end before encryption and check the CRC at the receiver site
    - but still there is certain probability of undetectable bit changes

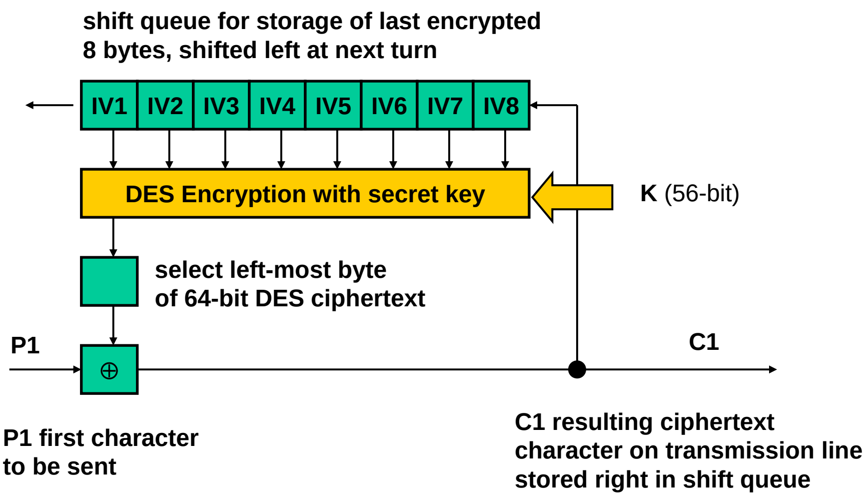
DES Modes - Block versus Stream

- **Cipher block chaining**
  - has the disadvantage of requiring an entire 64-bit block to arrive before decryption can begin
  - unsuitable for usage with interactive terminals
    - people type lines shorter 8 characters, stop and wait for response
- **Stream ciphers**
  - are able to perform byte-by-byte encryption
  - DES algorithm act as random number generator
    - pseudorandom stream controlled by a key
    - EXORing plaintext with pseudorandom stream
    - pseudorandom stream bits are based on previous ciphertext
    - application of one-time pad
  - Cipher Feedback (CFB), Output Feedback (OFB)

DES Mode - CFB 1

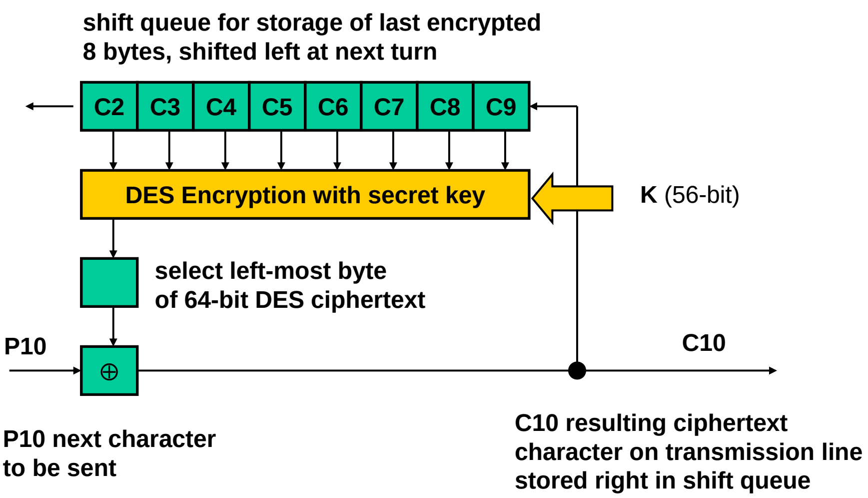
- **CFB - Cipher Feedback**
  - generation of keystream
    - 64-bit block as a shift queue
      - remembers last 8 bytes already sent in ciphertext
    - queue is encrypted by DES block cipher producing a 64-bit ciphertext
    - only leftmost byte of 64-bit ciphertext is used as a keystream generator for EXORing with plaintext byte
    - resulting 8-bit of ciphertext is sent on the transmission line and put back into the queue
      - the oldest byte will leave the queue
  - only the encryption function is used from block cipher
    - at decryption we have to EXOR with the same values!
  - initialization vector (IV) is needed, must be unique
    - start value of shift register

DES Mode - CFB 2



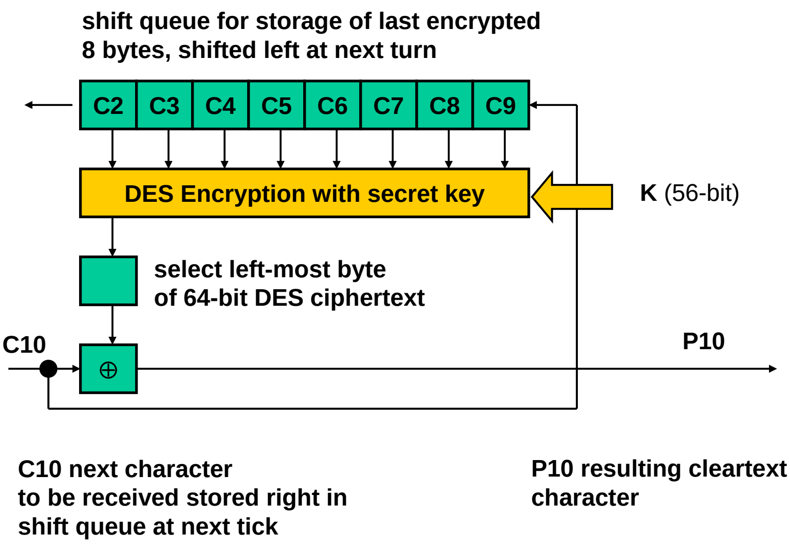
© 2005, D.I. Manfred Lindner Secret-Key, v4.3 27

DES Mode - CFB 3



© 2005, D.I. Manfred Lindner Secret-Key, v4.3 28

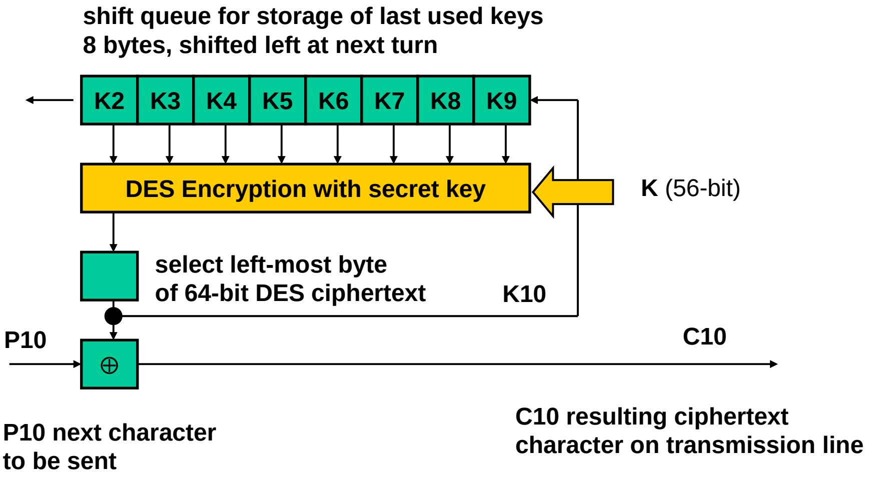
DES Mode - CFB 4



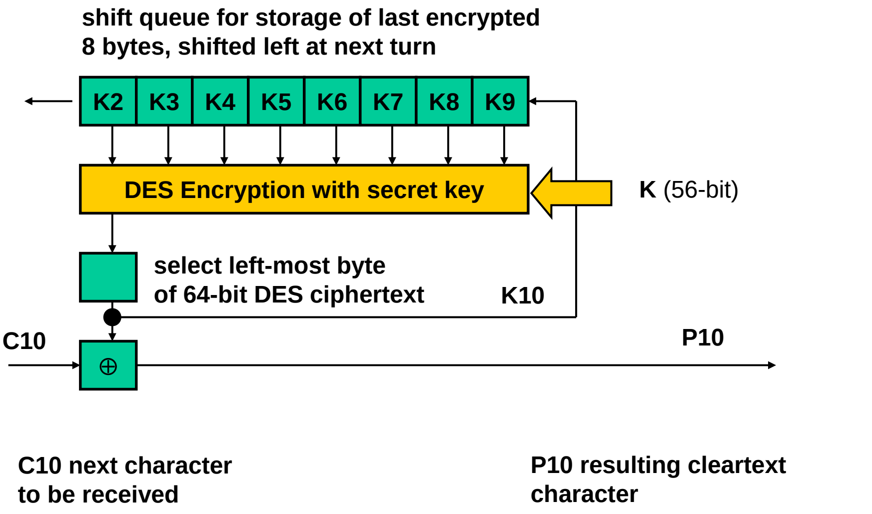
DES Mode - OFB 1

- **OFB - Output Feedback**
  - with CFB a single bit error on the line will influence the decryption of 8 bytes
    - as long as bad byte is stored in shift queue
    - but effect is localized and will not ruin the rest of the message
  - if this is not acceptable then OFB
    - internal feedback of keys into the keystream generator
    - otherwise similar to CFB
  - initialization vector (IV) is needed, must be unique
    - start value of shift register
  - with OFB a single bit error on the line will influence only one bit in the resulting plaintext
  - but OFB is less secure than other modes
    - keystream reuse attack (same key, IV used twice)

DES Mode - OFB 2



DES Mode - OFB 3



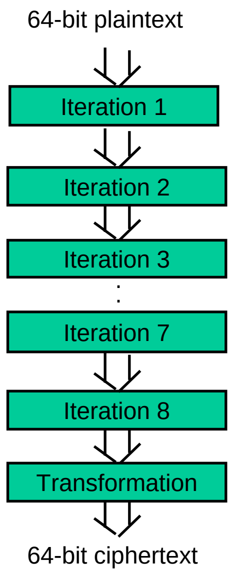
Agenda

- Introduction
- DES
- 3DES
- DES-Modes
- IDEA
- RC4
- AES

IDEA 1

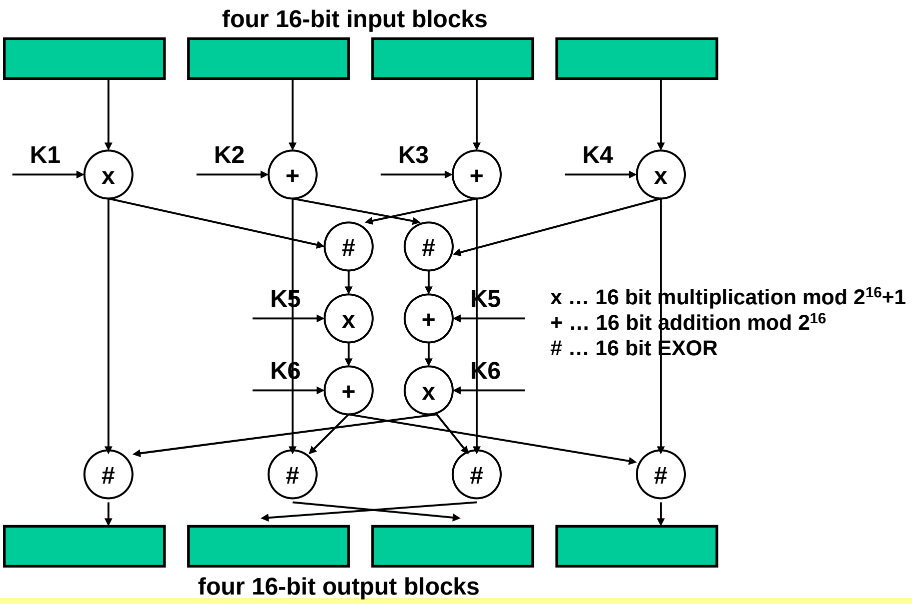
- **history**
  - 1990, IPES - Improved Proposed Encryption Standard
  - 1993, IDEA - International Data Encryption Algorithm
- **best block cipher available until AES**
- **operations**
  - 16 bit EXOR, addition modulo 2^{16}, multiplication modulo $2^{16}+1$ (prime), 8 rounds mangling
  - 64-bit data block, 4 sub-blocks
  - 128-bit key, 52 generated subkeys of 16 bits each
    - 6 keys for each iteration, 4 for final transformation
  - encryption and decryption uses the same algorithm
    - reversed and slightly modified subkeys

IDEA 2



 Secret-Key, v4.3 35

IDEA 1. Iteration (Subkeys K1 … K6) 3



 Secret-Key, v4.3 36

IDEA 4

- twice the speed as DES
- free of NSA guidance
- no real weaknesses found up to now
- 128 bit key length
  - breaking IDEA by exhaustive search (brute-force) requires currently unbelievable computing resources
- patented
  - but no license fee for non-commercial use
- part of PGP
  - Pretty Good Privacy
- can be used in DES - CBC and other DES modes

Agenda

- Introduction
- DES
- 3DES
- DES-Modes
- IDEA
- <u>RC4</u>
- AES

RC4

- **developed by Ron Rivest in 1987 for RSADSI**
- **secret algorithm for a long time**
  - RSADSI still treats it as a trade secret
  - the name is trademarked
- **compatible program was released on Usenet in September 1994**
- **variable key size stream cipher**
  - works in OFB mode
    - the keystream is independent of the plaintext
  - 8x8 S-box
    - slowly evolves with use
  - highly non-linear
  - RSADSI claims that it is immune to differential and linear cryptanalysis

Agenda

- **Introduction**
- **DES**
- **3DES**
- **DES-Modes**
- **IDEA**
- **RC4**
- **AES**

AES 1

- **Advanced Encryption Standard (AES)**
  - NIST sponsored a contest for new proposals which should replace DES and TripleDES in 1997
  - contest request
    - algorithm for a symmetric block cipher
    - the full design must be public
    - key lengths 128, 192, 256 bits must be supported
    - both SW and HW implementations must be possible
    - the algorithm must be public or licensed on nondiscriminatory terms
  - finalists of these contest were
    - Rijndael (from Joan Daemon, Vincent Rijmen, 86 votes)
    - Serpent (59 votes)
    - Twofish (team Bruce Schneier, 31 votes)
    - RC6 (from RSA lab, 23 votes)
    - Mars (IBM, 13 votes)

 Secret-Key, v4.3 41

AES 2

- **Advanced Encryption Standard (AES)**
  - Rijndael algorithm was chosen as the new standard
- **Rijndael:**
  - supports key length and block sizes from 128 bits to 256 bits in steps of 32
  - AES selects 128 bit block length and key lengths 128, 192, 256
  - 128 bit key length gives a key space of 3×10^{38} keys
  - is based on Galois field theory
  - substitution and permutation in several rounds (10 rounds for 128 bit keys)
  - all operations involve entire bytes (SW friendly)
  - only one S-box is used, XOR function and rotation is used
  - matrix multiplication using finite Galois field $GF(2^8)$
  - 2 GHZ machine should be able to do 700Mbit/s encryption

 Secret-Key, v4.3 42

Secret-Key Algorithm Comparison

• Blowfish	1-448 bits,	old and slow
• DES	56 bits,	too weak to use now
• IDEA	128 bits,	good, but patented
• RC4	1-2048 bits,	caution, some keys are weak
• RC5	128-256 bits,	good, but patented
• Rijndael	128-256 bits,	best choice
• Serpent	128-256 bits,	very strong
• TripleDES	112-168 bits,	second best choice
• Twofish	128-256 bits,	very strong, widely used

 Secret-Key, v4.3 43

Additional Information

- **TCP-IP Tutorial**
  - IBM Redbook
    - www.redbooks.ibm.com/pubs/pdfs/redbooks/gg243376.pdf
  - Chapter 21.1.1
  - Chapter 21.1.2
- **Internet Protocol Journal**
  - Volume 4 – Issue 2
    - www.cisco.com/ipj/
  - Article „ Goodbye DES, Welcome AES“

 Secret-Key, v4.3 44