Block vs Stream Ciphers

• block ciphers treats messages as blocks to be then en/decrypted separately.

• stream ciphers process messages a bit or byte at a time when en/decrypting—e.g., Vigenere

• many current ciphers are block ciphers- most major network-based cryptographic applications
Block Cipher Principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of $2^{64}$ entries for a 64-bit block
- instead create from smaller building blocks
- using idea of a product cipher
- It has complex structure compared to public-key algorithms

Motivation for Feistel Structure

![Diagram of Feistel Structure](image)

*Figure 3.4 General n-bit→n-bit Block Substitution (shown with n=4)*
Claude Shannon and Substitution-Permutation Ciphers

- In 1949 Claude Shannon introduced the idea of Substitution-Permutation (S-P) networks
  - Modern substitution-transposition product cipher
- These form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations we have seen before:
  - Substitution (S-box)
  - Permutation (P-box)
- Provide confusion and diffusion of message

Confusion and Diffusion

- Cipher needs to completely obscure statistical properties of original message
- A one-time pad does this
- More practically, Shannon suggested combining elements to obtain:
  - Diffusion – dissipates statistical structure of plaintext over bulk of ciphertext (each plaintext bit affects the value of many ciphertext bits)
  - Confusion – makes relationship between ciphertext and key as complex as possible – use complex substitution algorithm
Feistel Cipher Structure

- Horst Feistel proposed the **Feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon’s substitution-permutation network concept
Feistel Cipher Design Principles

- **block size**
  - increasing block provides more security, but reduces the en/decryption speed

- **key size**
  - larger size \(\rightarrow\) greater security, makes exhaustive key searching harder, but may slow cipher (common 64, 128)

- **number of rounds**
  - More rounds \(\rightarrow\) more security. (Typical 16 rounds)

- **subkey generation**
  - greater complexity makes cryptanalysis harder, but slows cipher

- **round function**
  - greater complexity can make analysis harder, but slows cipher

- **fast software en/decryption & ease of analysis**
  - are more recent concerns for practical use and testing
Feistel Cipher Decryption

• Use the same encryption algorithm with:
• The ciphertext as the input,
• The round keys are applied in reverse order:
  Use Kn in the first round, and K1 in the 16th round.

Data Encryption Standard (DES)

• most widely used block cipher in the world
• adopted in 1977 by NBS (now NIST) as FIPS PUB 46
• encrypts 64-bit data using 56-bit key
• IBM developed Lucifer cipher
  • by team led by Feistel
  • used 64-bit data blocks with 128-bit key
• in 1973 NBS issued request for proposals for a national cipher standard
• IBM submitted their revised Lucifer which was eventually accepted as the DES
DES Design Controversy

• although DES standard is public
• was considerable controversy over design
  • in choice of 56-bit key (vs Lucifer 128-bit)
  • and because design criteria were classified
• subsequent events and public analysis show in fact design was appropriate
• DES has become widely used, especially in financial applications

DES Encryption
Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- see text Table 3.2
- example:

\[
\text{IP}(675a6967 \ 5e5a6b5a) = (ffb2194d \ 004df6fb)
\]

DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

\[
L_i = R_{i-1} \\
R_i = L_{i-1} \text{ xor } F(R_{i-1}, K_i)
\]

- takes 32-bit R half and 48-bit subkey and:
  - expands R to 48-bits using perm E
  - adds to subkey
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes this using 32-bit perm P
Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (row bits) select one row
  - inner bits 2-5 (col bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  \[ S(18 \ 09 \ 12 \ 3d \ 11 \ 17 \ 38 \ 39) = 5fd25e03 \]
DES Key Schedule

- forms subkeys used in each round
- consists of:
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - selecting 24-bits from each half
    - permuting them by PC2 for use in function f,
    - rotating each half separately either 1 or 2 places depending on the key rotation schedule K

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again
- using subkeys in reverse order (SK16 ... SK1)
- note that IP undoes final FP step of encryption
- 1st round with SK16 undoes 16th encrypt round
- ....
- 16th round with SK1 undoes 1st encrypt round
- then final FP undoes initial encryption IP
- thus recovering original data value
Avalanche Effect

- A small change in the plaintext or the key should result in significant change in the ciphertext. It is a desirable property of encryption algorithm.
- where a change of one input or key bit results in changing approx half output bits
- making attempts to “home-in” by guessing keys impossible
- DES exhibits strong avalanche effect

Strength of DES – Key Size, DES Nature

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- recent advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
- now considering alternatives to DES
- DES Algorithm Nature: The main concern was about the S-Boxes. No body discovered the weakness in them
Strength of DES – Timing Attacks

- Attacks the actual implementation of the cipher
- Observes how long it takes to decrypt a ciphertext using a certain implementation.
- Uses the fact that calculations can take varying times depending on the value of the applied inputs.
- Noticing the Hamming weight (# of 1’s).
- DES is resistant to the timing attacks

Differential Cryptanalysis

- one of the most significant recent (public) advances in cryptanalysis
- published in 1990
- powerful method to analyse block ciphers
- used to analyse most current block ciphers with varying degrees of success
- DES reasonably resistant to it
Differential Cryptanalysis

- Finding the key by a chosen plaintext attack.
- A statistical attack against Feistel ciphers
- Design of S-P networks has output of function $f$ influenced by both input & key
- Hence cannot trace values back through cipher without knowing values of the key

Differential Cryptanalysis Compares Pairs of Encryptions

- With a known difference in the input
- Searching for a known difference in output
- When same subkeys are used
Linear Cryptanalysis

- another recent development
- also a statistical method
- must be iterated over rounds, with decreasing probabilities
- developed by Matsui et al in early 90's
- based on finding linear approximations
- can attack DES with $2^{47}$ known plaintexts, still in practise infeasible

Block Cipher Design Principles

- basic principles still like Feistel in 1970's
- number of rounds
  - more is better, exhaustive search best attack
- function f:
  - provides "confusion", is nonlinear, avalanche
- key schedule
  - complex subkey creation, key avalanche
Modes of Operation

- block ciphers encrypt fixed size blocks
- eg. DES encrypts 64-bit blocks, with 56-bit key
- need way to use in practise, given usually have arbitrary amount of information to encrypt
- Four standard modes were defined for DES
- Extended to five later, and they can be used with other block ciphers: 3DES and AES.

Electronic Codebook Book (ECB)

- message is broken into independent blocks which are encrypted
- each block is a value which is substituted, like a codebook, hence name
- each block is encrypted independently from the other blocks
  \[ C_i = DES_{k1} (P_i) \]
- uses: secure transmission of single values
Advantages and Limitations of ECB

- repetitions in message may show in ciphertext
  - if aligned with message block
  - with messages that change very little, which become a codebook analysis problem
- weakness due to encrypted message blocks being independent
- main use is sending a few blocks of data
Cipher Block Chaining (CBC)

- message is broken into blocks
- but these are linked together in the encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process
  \[ C_i = \text{DES}_K (P_i \ XOR \ C_{i-1}) \]
  \[ C_{-1} = \text{IV} \]
- uses: bulk data encryption, authentication
Advantages and Limitations of CBC

- each ciphertext block depends on all message blocks
- thus a change in the message affects all ciphertext blocks after the change as well as the original block
- need Initial Value (IV) known to sender & receiver
  - however if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate
  - hence either IV must be a fixed value or it must be sent encrypted in ECB mode before rest of message

Cipher FeedBack (CFB)

- message is treated as a stream of bits
- added to the output of the block cipher
- result is feedback for next stage (hence name)
- standard allows any number of bit (1, 8 or 64 or whatever) to be feedback
  - denoted CFB-1, CFB-8, CFB-64 etc
- is most efficient to use all 64 bits (CFB-64)
  \[ C_i = P_i \ XOR \ DES_{k_1}(C_{i-1}) \]
  \[ C_{-1} = IV \]
- uses: stream data encryption, authentication
Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- limitation is need to stall while do block encryption after every n-bits
- errors propagate for several blocks after the error
Output FeedBack (OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

\[ C_i = P_i \ XOR \ O_i \]
\[ O_i = DES_{k_1}(O_{i-1}) \]
\[ O_{-1} = IV \]
Advantages and Limitations of OFB

- used when error feedback a problem or where need to encryptions before message is available
- superficially similar to CFB
- but feedback is from the output of cipher and is independent of message
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- originally specified with m-bit feedback in the standards
- subsequent research has shown that only OFB-64 should ever be used

Counter (CTR)

- a “new” mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different counter value for every plaintext block (never reused)

\[ C_i = P_i \ XOR \ O_i \]
\[ O_i = DES_{K_1}(i) \]

- uses: high-speed network encryptions
Counter (CTR)

Advantages and Limitations of CTR

- **efficiency**
  - can do parallel encryptions
- random access to encrypted data blocks
- provable security (good as other modes)
- but must ensure never reuse key/counter values, otherwise could break (cf OFB)
Summary

• have considered:
  • block cipher design principles
  • DES
    • details
    • strength
  • Differential Cryptanalysis
  • Modes of Operation
    • ECB, CBC, CFB, OFB, CTR