Week 3

Block Ciphers

THE UNIVERSITY OF SYDNEY

DES and Block Ciphers

- A block cipher is a pair of encryption/decryption algorithms (E, D) operating on blocks of a fixed length B.

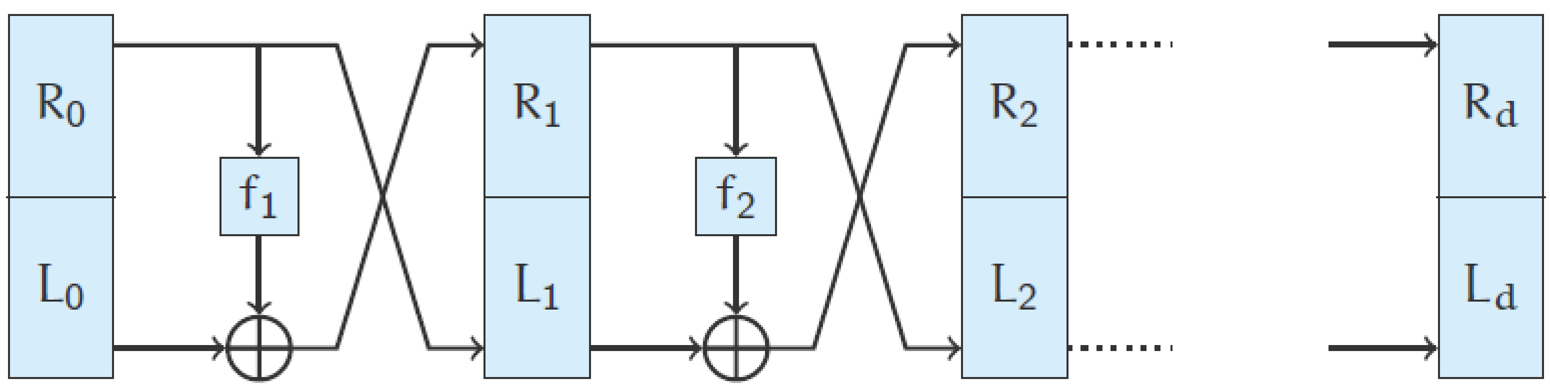
- Both algorithms take a K-bit key k, and for any block b:
$$D_k(E_k(b)) = b$$

- Data Encryption Standard (DES) is a block cipher operating on 64-bit blocks, using a 56-bit key.

Feistel Networks

- Recall: a block cipher $E_k : \{0,1\}^n \rightarrow \{0,1\}^n$ must be **invertible.**
  - Hard: coming up with a cryptographically secure invertible function.
  - Easier: coming up with pseudorandom functions (such as hashes).

- Feistel Network
  - Given d pseudorandom functions $f_1, \ldots, f_d$, where each f_i maps n bits to n bits, a Feistel Network combines these functions into a secure invertible function F, mapping $2n$ bits to $2n$ bits.

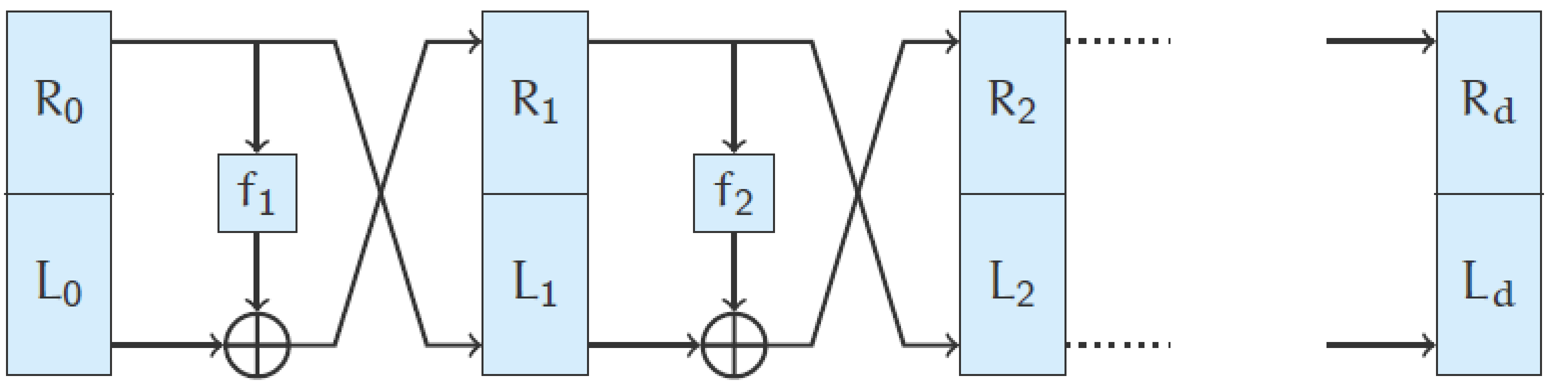


Feistel Networks

- Algebraically, we have d functions $f_1, \ldots, f_d$, and split our initial input into two: (L_0, R_0). The Feistel network is then:

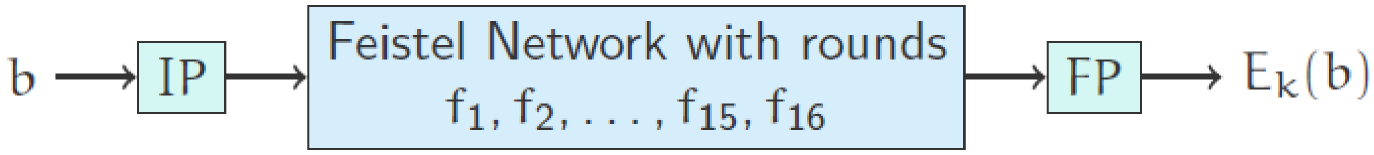
$$
\begin{aligned}
L_i &= R_{i-1} \\
R_i &= f_i(R_{i-1}) \oplus L_{i-1}
\end{aligned}
$$

  - The whole network is invertible because each step is invertible.
  - In fact, the inverse network is identical, but with the function order reversed: $f_d, f_{d-1}, \ldots, f_2, f_1$.
  - When used in a cipher, the functions $f_1, \ldots, f_d$ are called round functions.



DES Internals

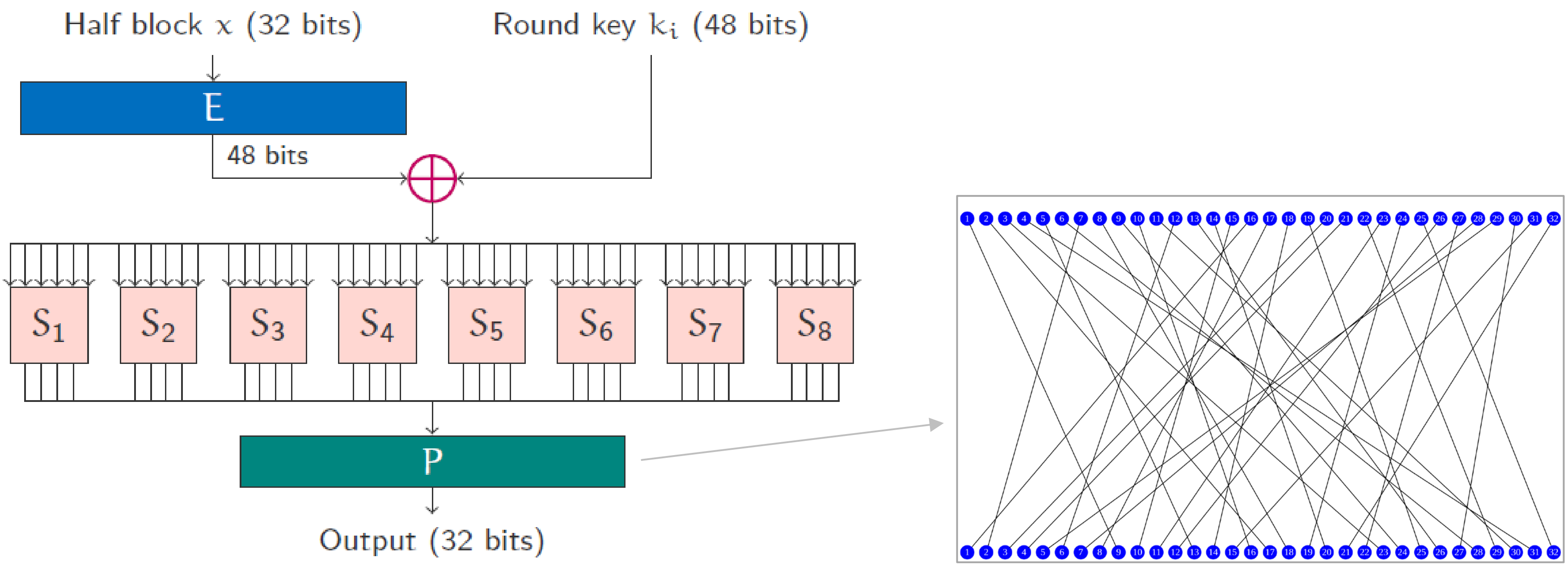
- DES takes a 56-bit key k and a 64-bit block b to be encrypted.
  - An initial permutation (IP) is applied to the block.
  - The block is then fed into a 16-round Feistel network, with round functions $f_i(x) = F(x, k_i)$, where $\mathrm{k_i} \in \{k_1, \ldots, k_{16}\}$ is the key schedule derived from k, and F is the DES round function.
  - Each round key k_i is 48-bits
  - A final permutation (FP) is then applied to the output.



The overall structure of DES

DES Round Function

- The round function $f_i(x) = F(x, k_i)$ consists of:
  - E, an expansion permutation widening x from 32 to 48 bits.
  - S_j, the substitution boxes (S-boxes) which collapse 6 bits to 4 bits.
  - P, a fixed permutation (P-box).
  - https://en.wikipedia.org/wiki/DES_supplementary_material



Diffusion and Confusion

- Diffusion and confusion are desirable properties of ciphers to prevent statistic-based cryptanalysis
- **Diffusion** is the dissipation of statistical information in the plaintext.
  - Flipping a bit in the plaintext should result in half the bits of the ciphertext changing.
  - Flipping a bit in the ciphertext should result in half the bits of the plaintext changing.
  - Related to permutation ciphers.
- **Confusion** is making the relationship between the key and ciphertext as complicated as possible.
  - Each bit of the ciphertext should depend on multiple bits of the key.
  - Even if an attacker gathers many (plaintext, ciphertext) pairs encrypted under the same key, they should not be able to deduce the key.
  - Related to substitution ciphers.

DES S-Boxes and P-Boxes

- When highly nonlinear S-boxes are combined with good P-boxes, both the properties of confusion and diffusion arise.
  - Having linear S-boxes would make the whole of DES a linear function.
  - Having P-boxes not spreading bits around enough would allow DES to be broken down into smaller independent subproblems.

- Qualitatively, "good" S and P-boxes "work together":
  - S-boxes will be highly nonlinear, and flipping an input bit should result in half the output bits flipping.
  - The P-box following this up should distribute those bits evenly across S-boxes in the next round.

Block Cipher Padding and Modes of Operation

THE UNIVERSITY OF SYDNEY

Block Cipher Padding

- We need to pad the plaintext when it does not fit neatly into blocks.
  - Done by appending a pre-defined sequence of bits to “fill” the final block.

- It’s important that the bits used for padding are chosen carefully, as they have cryptographic implications. Some acceptable padding functions:
  - ANSI X9.23 ANSI X9.23 cipher block chaining - IBM Documentation
  - PKCS#5 and PKCS#7 https://www.ibm.com/docs/en/zos/2.4.0?topic=rules-pkcs-padding-method

Block Cipher Modes of Operation

- Once a key k is chosen and loaded into a block cipher, E_k only operates on single blocks of data.
  - Block size usually small (16-byte blocks for AES)
  - Message to be sent usually large (web page + assets ≈ 500kB)
  - Need a way to repeatedly apply the cipher with the same key to a large message.

- A mode of operation describes how a block cipher is repeatedly applied to encrypt a message. Each mode of operation has its own advantages and disadvantages.

- By using different modes of operation, messages of an arbitrary length can be split into blocks and encrypted using a block cipher.

Evaluating Block Ciphers & Modes

- To evaluate a cipher and a mode of operation, examine:
  1. **Key Size:** Upper bound on security, but longer keys add costs (generation, storage, etc.)
  2. **Block Size:** Larger is better to reduce overheads, but is more costly.
  3. **Estimated Security Level:** Confidence grows the more it is analysed, known failure cases for certain modes
  4. **Throughput:** How fast can it be encrypted/decrypted? Can it be pre-computed? Can it be parallelised?
  5. **Error Propagation:** What happens as a result of bit errors or bit loss?

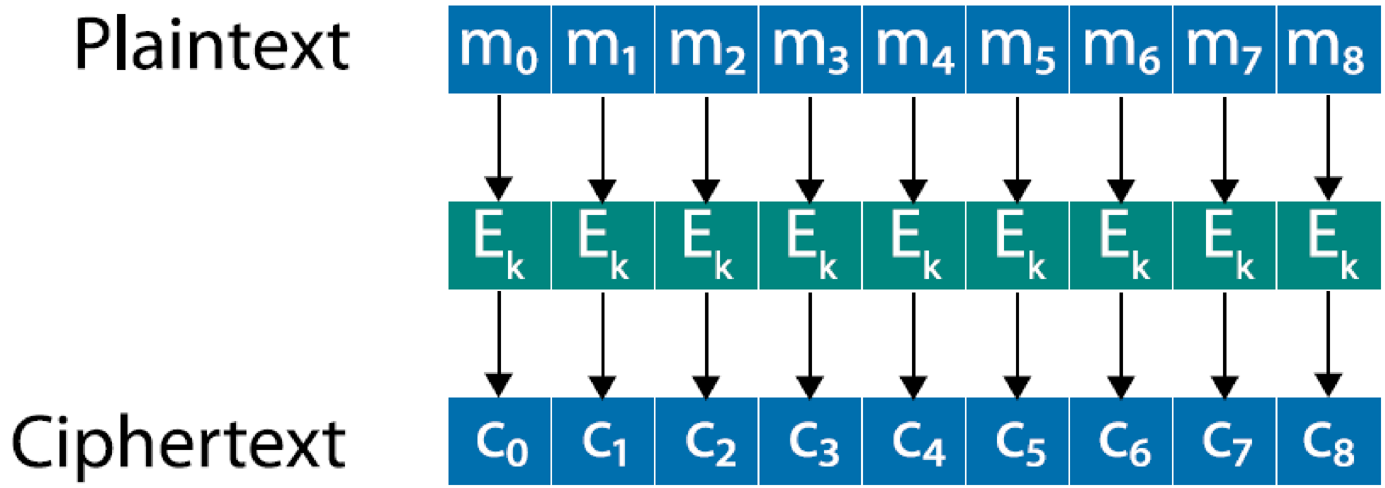
- Points #1 and #2 are relevant only to the cipher. Points #3, #4, and #5 are relevant to both the cipher and mode of operation.

Evaluating Block Ciphers & Modes

- We will consider 5 different modes of operation:
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Output Feedback (OFB)
  - Counter Mode (CTR)
  - Galois/Counter Mode (GCM)

Electronic Code Book (ECB) Mode

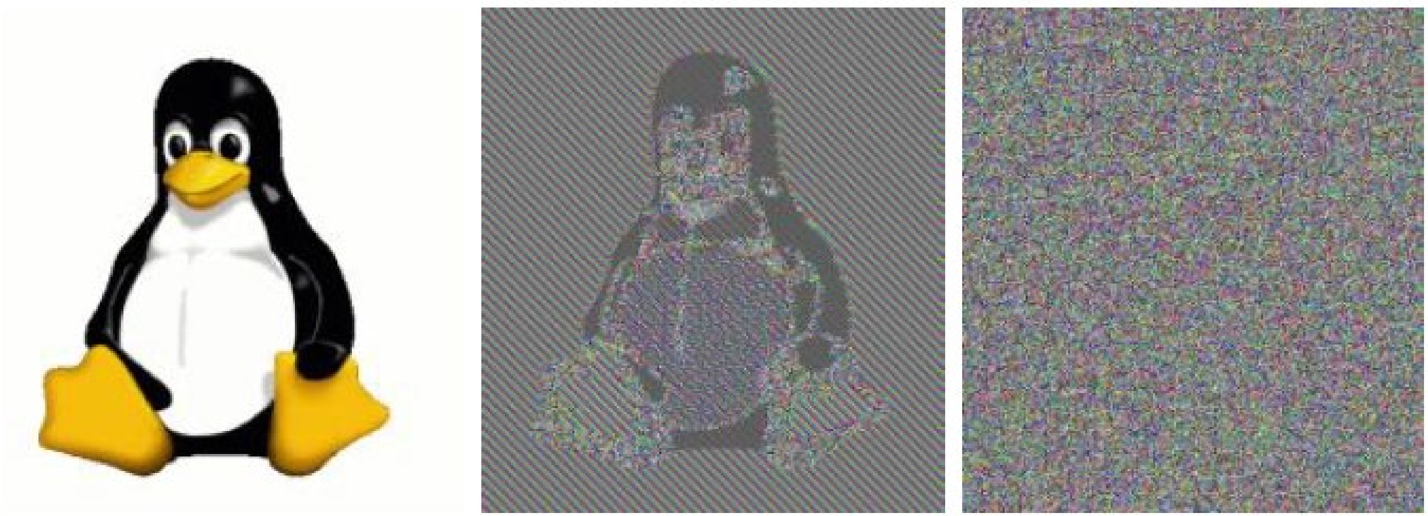
– Electronic Code Book (ECB) encrypts each block separately. Simple to implement but vulnerable to dictionary and frequency attacks.



– Problem with ECB is that it is a substitution cipher, with blocks instead of letters!

ECB Properties

- Identical plaintext blocks result in identical ciphertext blocks
  - Since blocks are enciphered independently, a reordering of ciphertext blocks results in reordering of plaintext blocks.

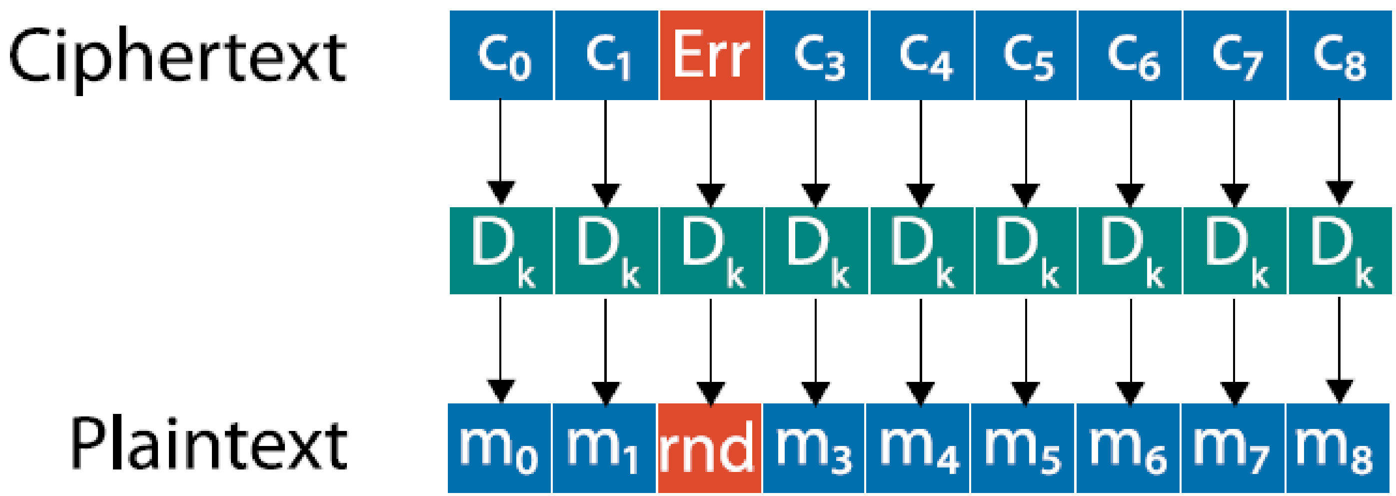


(a) Original Image (b) ECB mode (c) Other mode

Encryption of Tux[1] image.

ECB Properties

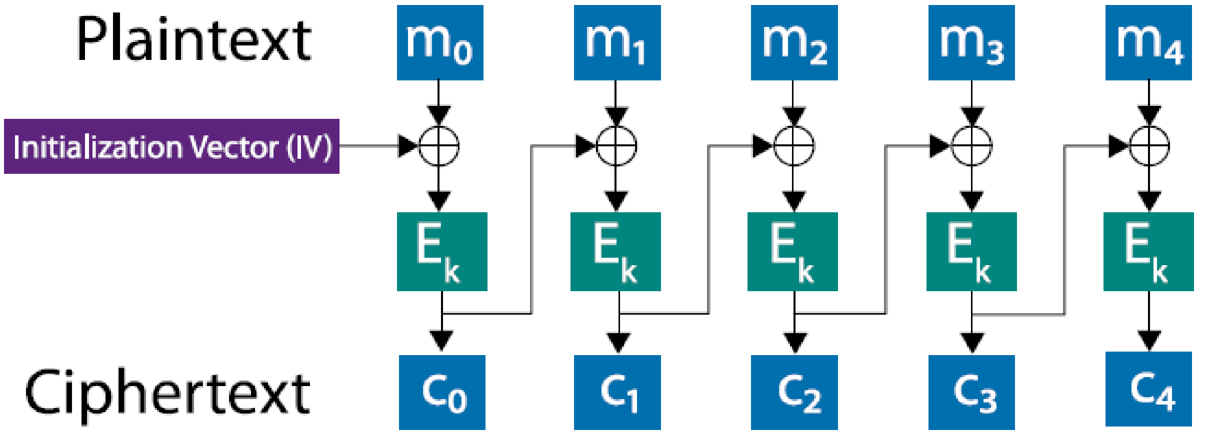
– Local error propagation: Bit errors only impact the decoding of the corrupted block (block will result in gibberish)



Error propagation in ECB

Cipher Block Chaining (CBC) Mode

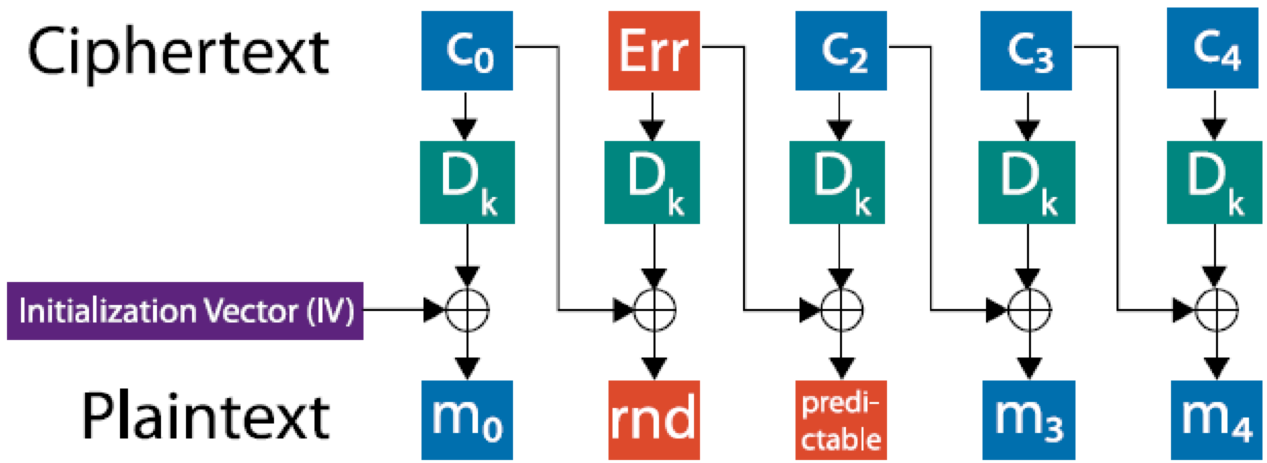
- In Cipher Block Chaining (CBC) blocks are chained together using XOR.
- The Initialisation Vector (IV) is a **random** value that ensures the same plaintext and key does not produce the same ciphertext and IV does not need to be kept a secret.



CBC Mode Encryption

CBC Decryption

- Ciphertext errors only affect two plaintext blocks, one in a predictable way.
- Encryption must be done sequentially.
- Decryption can be random-access and is fully parallelisable.



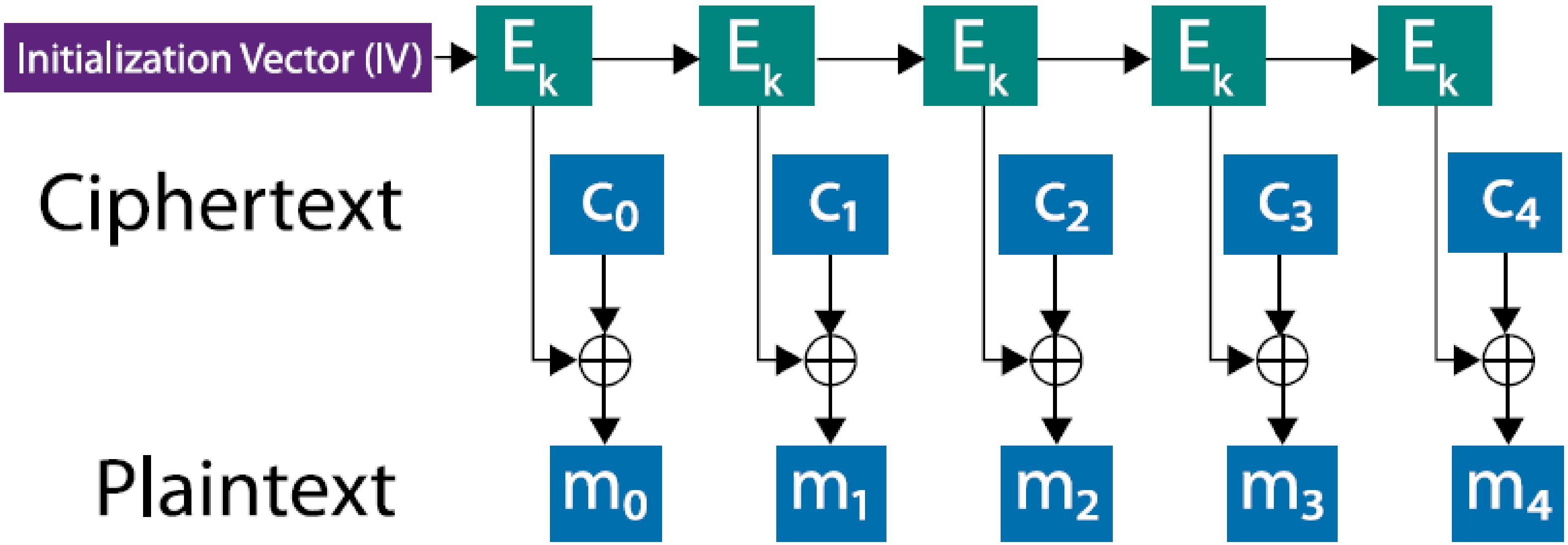
CBC Decryption

CBC Properties

- Identical plaintexts result in identical ciphertexts when the same plaintext is enciphered using the same key and IV.
  - Changing at least one of $[k, IV, m_0]$ addresses this.
- Rearrangement of ciphertext blocks affects decryption, as ciphertext part c_j depends on all of $[m_0, m_1, \cdots, m_j]$.
- **Error propagation:**
  - Bit error in ciphertext c_j affects deciphering current and next blocks c_j and c_{j+1}. Recovered block m'_j typically results in random bits.
  - Bit errors in recovered block m'_{j+1} are precisely where c_j was in error.
  - Attacker can cause **predictable** bit changes in m_{j+1} by altering c_j.
- **Bit recovery:**
  - CBC is self-synchronising in that if a bit error occurs in c_j but not c_{j+1}, then c_{j+2} correctly decrypts to m_{j+2}. Only two blocks affected by error.

Output Feedback (OFB) Mode

- Output Feedback Mode (OFB) turns a block cipher into a synchronous stream cipher.
- The Initialisation Vector (IV) and the key are used to generate a keystream.

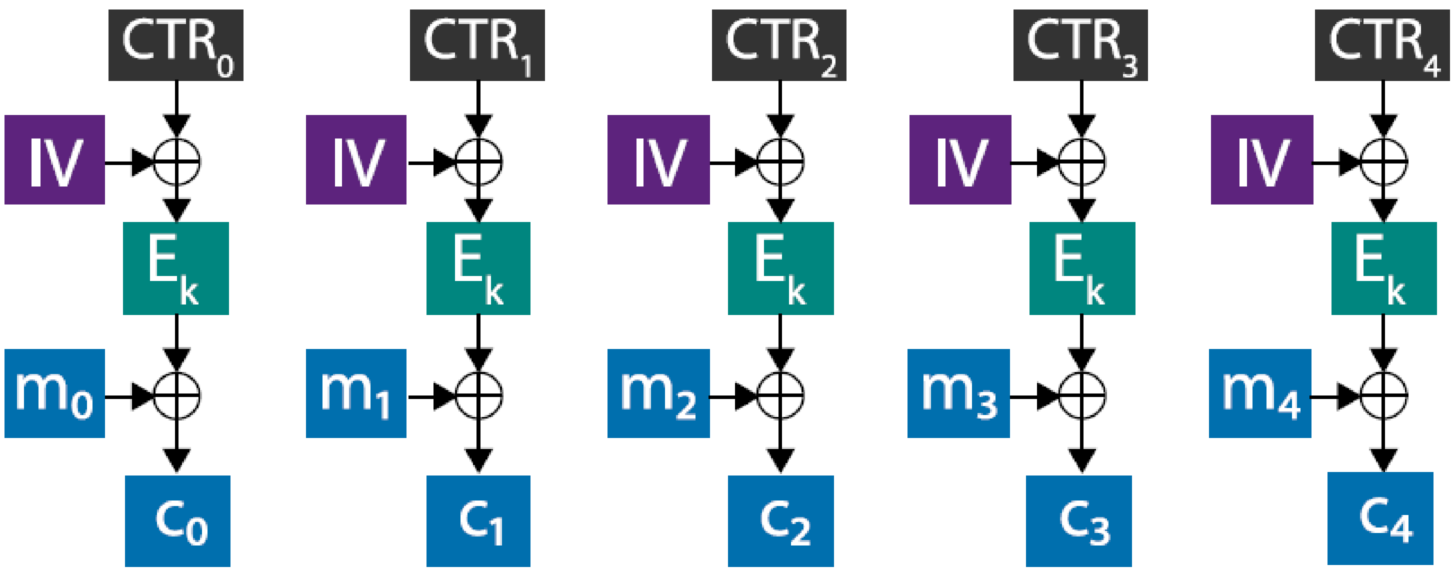


OFB Properties

- Identical plaintext results in identical ciphertext when the same plaintext is enciphered using the same key and IV.
- **Chaining Dependencies:** (Same as a stream cipher) The key stream is plaintext independent.
- **Error propagation:** (Same as a stream cipher) Bit errors in ciphertext blocks cause errors in the same position in the plaintext.
- **Error recovery:** (Same as a stream cipher) Recovers from bit errors, but not bit loss (misalignment of key stream)
- **Throughput:** Key stream may be calculated independently — e.g. pre-computed — before encryption/decryption become parallelisable.
- IV must change. Otherwise, it becomes a two time pad.

Counter (CTR) Mode

- Counter Mode (CTR) modifies the IV for each block using a predictable counter function, turning the block cipher into a stream cipher.
- The counter can be any function (e.g. a PRNG), but it is commonly just an incrementing integer.



CTR Mode Encryption

CTR Properties

- Identical plaintext results in identical ciphertext when the same plaintext is enciphered using the same key and IV.
- **Chaining Dependencies:** (Same as a stream cipher) The key stream is plaintext independent.
- **Error propagation:** (Same as a stream cipher) Bit errors in ciphertext blocks cause errors in the same position in the plaintext.
- **Error recovery:** (Same as a stream cipher) Recovers from bit errors, but not bit loss (misalignment of key stream)
- **Throughput:** Both encryption and decryption can be randomly accessed and/or parallelised: the best we could hope for.
- IV must change. Otherwise, it becomes a two time pad.

- *OFB and CTR share similar properties, because they both make the block cipher act as a stream cipher.*

Galois/Counter Mode

- Galois/Counter Mode (GCM) mode is not strictly a cipher mode of operation since it also provides a mechanism to verify the integrity of data: assurance the ciphertext has not been tampered with.
  - An extension of CTR mode.
  - While encryption happens, the ciphertext blocks are combined into something like a MAC.
  - Unlike HMAC, is parallelisable (you can't combine two HMACs into one larger one).
  - Used for low-latency, high-throughput dedicated hardware applications (network packets).

Attacks on DES

DES Keys

– Given one plaintext/ciphertext pair (m, c), there is a high probability that only one key will satisfy:

$$c = DES(m, k)$$

– Consider DES as a collection of permutations: $\pi(1) \ldots \pi(256)$. If $\pi(i)$ are independent permutations then for all (m, k):

$$\begin{aligned}\Pr[\exists k_1 \neq k : DES(m, k_1) = DES(m, k)] &= 2^{-56} \\ &= 1.39 \times 10^{-17} \\ &= 0.00000000000000139\%\end{aligned}$$

– Thus, given one (m, c) pair, the key is (almost definitely) uniquely determined. The problem is to find k.

Attacks on DES

- Known Plaintext Exhaustive Key Search
  - Strong n-bit block cipher, j-bit key, the key can be recovered on average in 2^{j-1} brute force attempts, given a small number ($< (j+4)/n$) of plaintext/ciphertext pairs
  - For DES, $j = 56$, n $= 64$ so exhaustive key search is expected to yield the key in 2^{55} operations.

Increasing Attacks on DES

- DES is a Feistel Network, so due to the complementation property:

$$DES(\neg m, \neg k) \ = \ \neg DES(m, k)$$

- This property can be exploited with a Chosen Plaintext Attack
  - Let the actual key be k and the proposed key be K. If the encryption of a message m and its complement $\neg m$ can be obtained, brute force attempts can be reduced as follows:

```
if c1 = DES(m, k) and c2 = DES(¬m, k) then
    if DES(m, K) ≠ c1 OR c2‾ then
        K ≠ k or ¬k
    end if
end if
```

$$\overline{c_2} = \neg c_2 = \mathrm{DES}(m, \neg k)$$

- Therefore, the search space is halved.

DES Enhancements

2DES, 3DES, DESX

2DES

- DES Double Encryption with 2 keys (2DES)

$$\mathrm{2DES}_{k_1,k_2}(m) = E_{k_1}(E_{k_2}(m))$$

  - Vulnerable to known plaintext meet-in-the-middle attack.

- Example:
  - for a fixed message, m, create a table of all possible ciphertext with each 56-bit encryption keys:

$$E_k(m) \text{ for all } k \text{ in } \{0,1\}^{56}$$

  - Then, for $c = \mathrm{E}_{k_1,k_2}(m)$, try to decrypt:

$$D_k(c) \text{ for all } k \text{ in } \{0,1\}^{56}$$

  - Until $D_k(c)$ appears in the table, since $D_{k_1}(c) = E_{k_2}(m)$.

- This means that 2DES can be broken in 2^{56} operations on average, using 2^{56} memory slots. (A time-space trade-off!). This is not good when the key is 112-bits (56 + 56).

3DES

- Two-key Triple DES (3DES) - DES 3 times, 2 keys. (112 bits)

$$\text{3DES}_{k_1,k_2}(m) \; = \; E_{k_1}(D_{k_2}(E_{k_1}(m)))$$

- A Common Vulnerabilities and Exposures (CVE) released in 2016 discovered a major security flow in DES/3DES.

- NIST deprecated DES/3DES for new applications in 2017, and for all applications at the end of 2023.

DESX

- A modification of DES to avoid exhaustive key search is DESX.

  $k_1 = 56$ bits (DES Key)

  $k_2 = 64$ bits (Whitening Key)

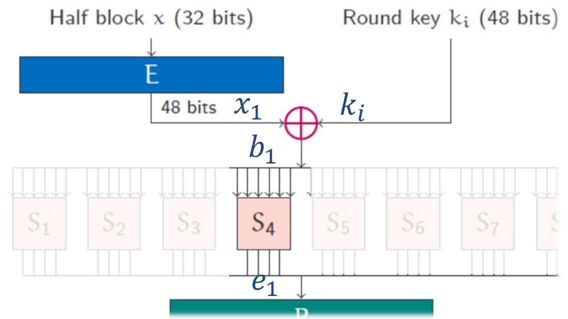
  $k_3 = 64$ bits

- $DESX_{k_1,k_2,k_3}(m) = k_3 \oplus E_{k_1}(m \oplus k_2)$

- The **whitening key** gives greater resilience to brute force attacks.

DES Cryptanalysis

Differential Cryptanalysis



- Better than brute-force approach for attacking DES.

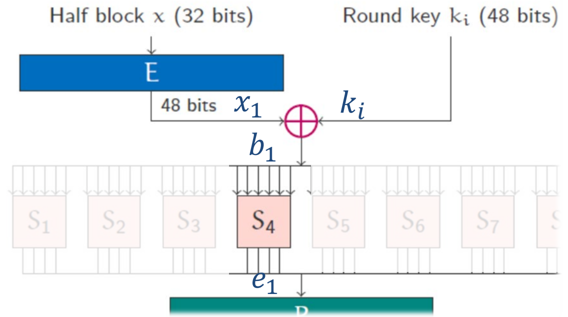
- Utilises (plaintext, ciphertext) pairs with Chosen Plaintext Attack (CPA). Involves looking at the XOR of two texts.

- Consider any s-box function $F(x, k_i)$ with key k_i and define the difference measure between two different inputs to the s-box (b_1 and b_2) as:

$$\begin{aligned} \Delta &= b_1 \oplus b_2 \\ &= (x_1 \oplus k_i) \oplus (x_2 \oplus k_i) \\ &= x_1 \oplus x_2 \end{aligned}$$

  where x_1 and x_2 are the corresponding outputs of the Expander Function.

- The input XOR ($b_1 \oplus b_2$) does not depend on the key, but the output XOR ($e_1 \oplus e_2$) will still depend on the key.



Example

- Define the set Δ(b) consisting of input ordered pairs (b_1, b_2) with 6 bits, where |Δ(b)| = 2^6 = 64.

$$\Delta(b) = \{(b_1, b_2) \text{ in } \{0,1\}^6 | \, b_1 \oplus b_2 = b\}$$

- If $b = b_1 \oplus b_2 = 110100$, then consider the first S-Box pairs to be:

$$\Delta(b) = \{(000000, 110100), (000001, 110101), \ldots (111111, 001011)\}$$

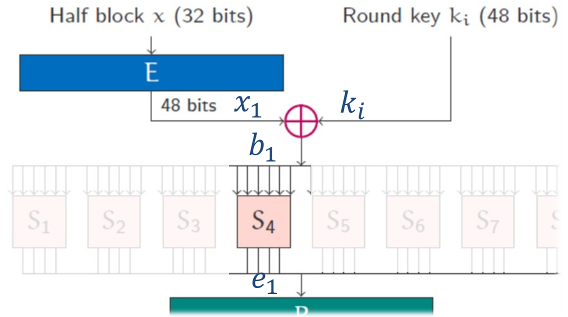
- For all 64 pairs in $\Delta(b)$, the distribution of output XORs $(e_1 \oplus e_2)$ is

$(e_1 \oplus e_2)$:	0000	0001	...	1111
Count	0	8		6

- So, if $(b_1 \oplus b_2) = 110100$ and $(e_1 \oplus e_2) = 0001$, then (b_1, b_2) must be one of the eight possible pairs, ∴ b_1 is one of 16 possible values.

- Since x_1 is derived from the known plaintext, the 6 bits of the key used in $x_1 \oplus ki = b_1$ are one of the 16 possible values. This is repeated with different Δ to make deductions about the key!

Example



- Define the set Δ(b) consisting of input ordered pairs (b_1, b_2) with 6 bits, where $|\Delta(b)| = 2^6 = 64$.

$$\Delta(b) = \{(b_1, b_2) \text{ in } \{0,1\}^6 \mid b_1 \oplus b_2 = b\}$$

Without knowing b, b_1 has 64 possibilities and b_2 has 64 possibilities causing a 64 x 64 brute-force search space. Because of b, b_1 and b_2 can only have 64 possibilities

- If $b = b_1 \oplus b_2 = 110100$, then consider the first S-Box pairs to be:

$$\Delta(b) = \{(000000, 110100), (000001, 110101), \ldots (111111, 001011)\}$$

- For all 64 pairs in $\Delta(b)$, the distribution of output XORs $(e_1 \oplus e_2)$ is

$(e_1 \oplus e_2)$:	0000	0001	…	1111
Count	0	8	…	6

Because of $e_1 \oplus e_2$ constraint, b_1 and b_2 can only have 8 possibilities. Since b_1 and b_2 are interchangable, each of b_1 and b_2 has 16 possibilities

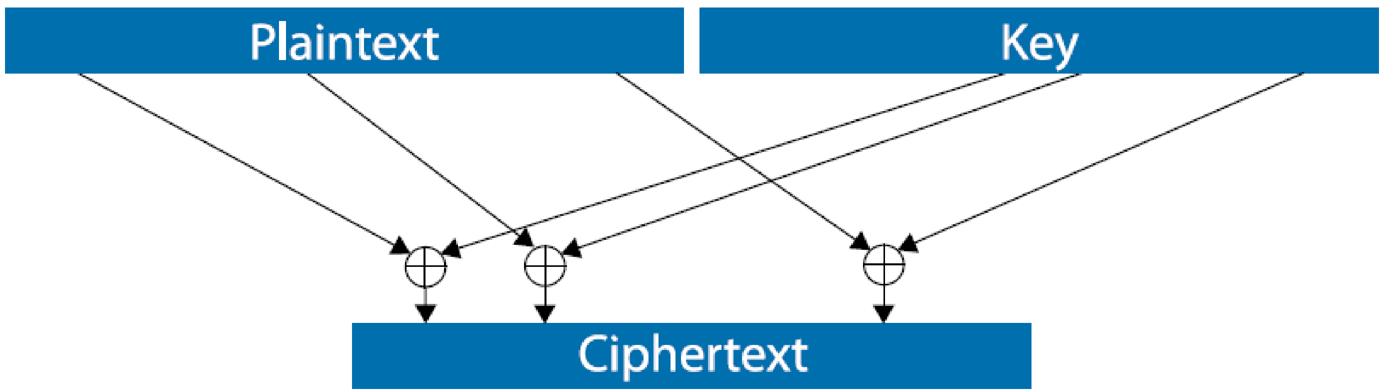
- So, if $(b_1 \oplus b_2) = 110100$ and $(e_1 \oplus e_2) = 0001$, then (b_1, b_2) must be one of the eight possible pairs, ∴ b_1 is one of 16 possible values.

- Since x_1 is derived from the known plaintext, the 6 bits of the key used in $x_1 \oplus ki = b_1$ are one of the 16 possible values. This is repeated with different Δ to make deductions about the key!

Since x_1 is known, 16 possibilities for b_1 are a result of the 16 possibilities for k_i

Linear Cryptanalysis

– Consider the ciphertext derived by combining certain bits from plaintext and key:



– The cipher can be easily broken because it is **linear**. For example:

$$c[1] = p[4] \oplus p[17] \oplus k[5] \oplus k[3]$$

– Therefore, $k[3] \oplus k[5] = c[1] \oplus p[4] \oplus p[17]$

Linear Cryptanalysis

- The purpose of linear cryptanalysis is to find the following "effective" linear expression for a given cipher algorithm

$$p[i_1, \dots, i_u] \oplus c[j_1, \dots, j_v] = k[s_1, \dots, s_w]$$

where $i_1, \dots, i_u, j_1, \dots, j_v$ and $s_1, \dots, s_w$ are fixed bit locations.

- An attacker wants the above expression to hold with probability, $\rho \neq 0.5$ for randomly given plaintext P and corresponding cipher text C. If $|\rho - 0.5|$ is large, then the attacker can accurately guess $k[s_1, \dots, s_w]$.

- Optimally, for a break, $|\rho - 0.5| = 0.5$ (i.e., $\rho = 0$ or 1) whilst a perfect cipher should have $\rho = 0.5$.

DES Strength against attacks

– Attack vs Complexity

Attack	Messages		Requirements	
	Known	Chosen	Storage	Processing
Exhaustive Precomputation	-	1	2^{56}	1
Exhaustive Search	1	-	Neg.	2^{55}
Linear Cryptanalysis	2^{43}(85%)	-	Texts	2^{43}
	2^{38}(10%)	-	Texts	2^{50}
Differential Cryptanalysis	-	2^{47}	Texts	2^{47}
	2^{55}	-	Texts	2^{55}

Replacing DES

Advanced Encryption Standard (AES)

THE UNIVERSITY OF SYDNEY

Introducing AES

- In 1997 NIST announced that a competition would be held to choose a new cipher to replace the outdated DES cipher, this to be was named the Advanced Encryption Standard – AES.

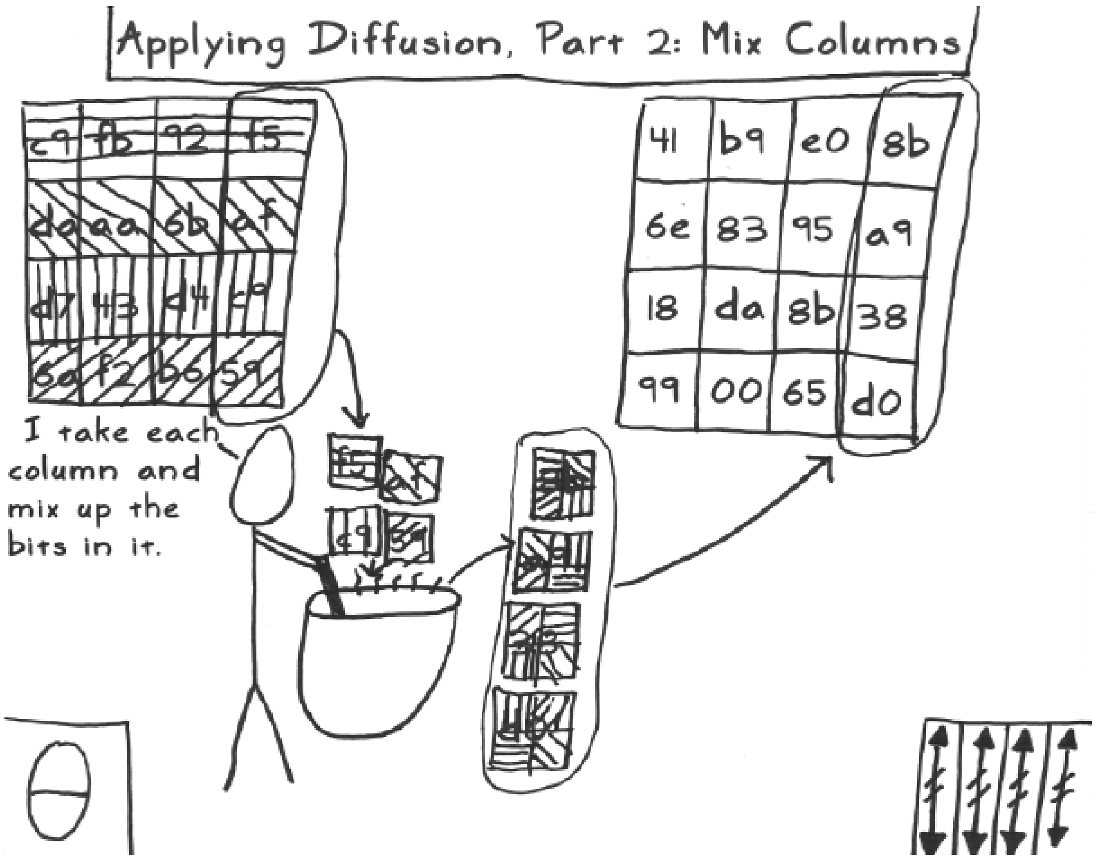
- Of the 15 international contenders, they chose Rijndael as the AES.
  - Block cipher
  - Operates on 128-bit blocks
  - Key length is variable: 128/192/256-bit keys
  - It is an SP-network (substitution-permutation network) which uses a single non-linear S-box which acts on a byte input to give a byte output (a 256-byte lookup table)
  - Construction gives tight differential and linear bounds

AES

- The number of rounds are variable:
  - 10 rounds – 128-bit keys
  - 12 rounds – 192-bit keys
  - 14 rounds – 256-bit keys
- Rounds have a 50% margin of safety based on current known attacks. Potential attacks (which require an enormous number of plaintext/ciphertext pairs) are possible on:
  - Only 6 rounds for 128-bit keys
  - Only 7 rounds for 192-bit keys
  - Only 9 rounds for 256-bit keys
- Safety against possible attacks believed to currently be ≈ 100%

AES

– http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html



Questions

- Why is it important to have properties of diffusion and confusion in the block cipher?

- What is the purpose of the Initialisation Vector (IV) in block cipher modes?

- What are two cryptanalysis techniques applied to DES and their main differences?

THE UNIVERSITY OF
SYDNEY