Week 4
Symmetric Cryptography

THE UNIVERSITY OF SYDNEY

Dr. Thilini Dahanayaka

School of Computer Science, The University of Sydney

Agenda

- Cryptography - Definitions
- Historical Ciphers
- Stream and Block Ciphers
- DES and 3DES
- Advanced Encryption Standard (AES)
- Cipher Block Modes
- Pseudorandom Number Generation

Math to Know

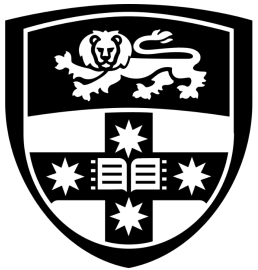
- Hopefully you have read the lecture notes provided on the require math background. Following are the key ideas you need to know.
    - XOR function
    - Modular Arithmetic
    - Extended Euclidean Algorithm
    - Primitive Roots under modulo
    - Discrete Logarithm Problem
    - Groups/Rings/Fields and their variants
    - Different types of finite fields and operations within those.
    - Broader applicability of those math concepts in different aspects of cryptography.

Even if you haven't done it today, please ensure that you follow the given notes on math. Those are assessable. You will also need them throughout this unit.

Recommended Reading

- Cryptography & Network Security - 7th Edition by William Stallings
  - **Chapter 1** – Computer network and security concepts
  - **Chapter 2** – Introduction to Number Theory
  - **Chapter 3** – Classical Encryption Techniques
  - **Chapter 4** – Block Ciphers and Data Encryption Standards
  - **Chapter 5** – Finite Fields
  - **Chapter 6** – Advanced Encryption Standards
  - **Chapter 7** – Block Cipher Operation
  - **Chapter 8** – Random bit Generation and Stream Ciphers

Now that's a lot of chapters !! All you need to know are in the lectures and we are not using all the content of those chapters. However, reading those will help you to better understand the concepts.



THE UNIVERSITY OF
SYDNEY

1. Cryptography - Definitions

Cryptography

- An indispensable tool
  - Secure protocols and systems use cryptography for:
    - Data confidentiality (encryption)
    - Data integrity (signatures, message authentication codes)
    - Authentication & data origin verification (signatures, message authentication codes)
    - Non-repudiation (signatures)

- Fundamental understanding of the principles of cryptography is required for many careers in Computing.

- In this unit, we introduce it more from a functional point of view, rather than mathematical/formal.

Cryptography

- A double-edged sword: complex and many subtleties
- Inconsiderate application of cryptography: likely insecure system
  - Worse, an insecure system that seems secure to its developers and users
  - The attacker doesn't know it is supposed to be secure and will take it apart
  - Historical example GSM A5/1 algorithm
- Implementing cryptography requires tremendous experience and knowledge of the peculiarities of hardware and programming languages
  - Do not cook (implement) your own crypto and use it for real
  - Recognized way to become proficient is to practice, find a mentor, and discuss with other implementers → long career path with no shortcuts!

Cryptography

Cryptography: The science of securely transmitting data, spatially and temporally.

- **Spatially:** Over a physical distance
- **Temporally:** Data stays secure a long time into the future

- In this unit, we are commonly concerned with:
  - Confidentiality
  - Integrity
  - Authentication

- There are other applications of cryptography:
  - Non-repudiation
  - Privacy
  - Anonymity

Means to Achieve These

- **Confidentiality:** Encryption
  - Symmetric encryption
  - Asymmetric (public-key) encryption
- **Integrity:**
  - Symmetric and asymmetric ways to achieve this
  - Message Authentication Codes and Signatures
  - Hash functions
- **Authentication:** Protocols

Two Forms of Cryptography

Symmetric Cryptography: The key for encryption and decryption is the same ($k_e = k_d$), or at least $k_d \leftarrow k_e$ (i.e., can derive k_d from k_e)

- Also known as **shared-key cryptography**
- Based on algorithms to shuffle symbols around and map them to others

Asymmetric Cryptography: The keys for encryption and decryption are different ($k_e \neq k_d$), and it is infeasible to compute k_d from k_e

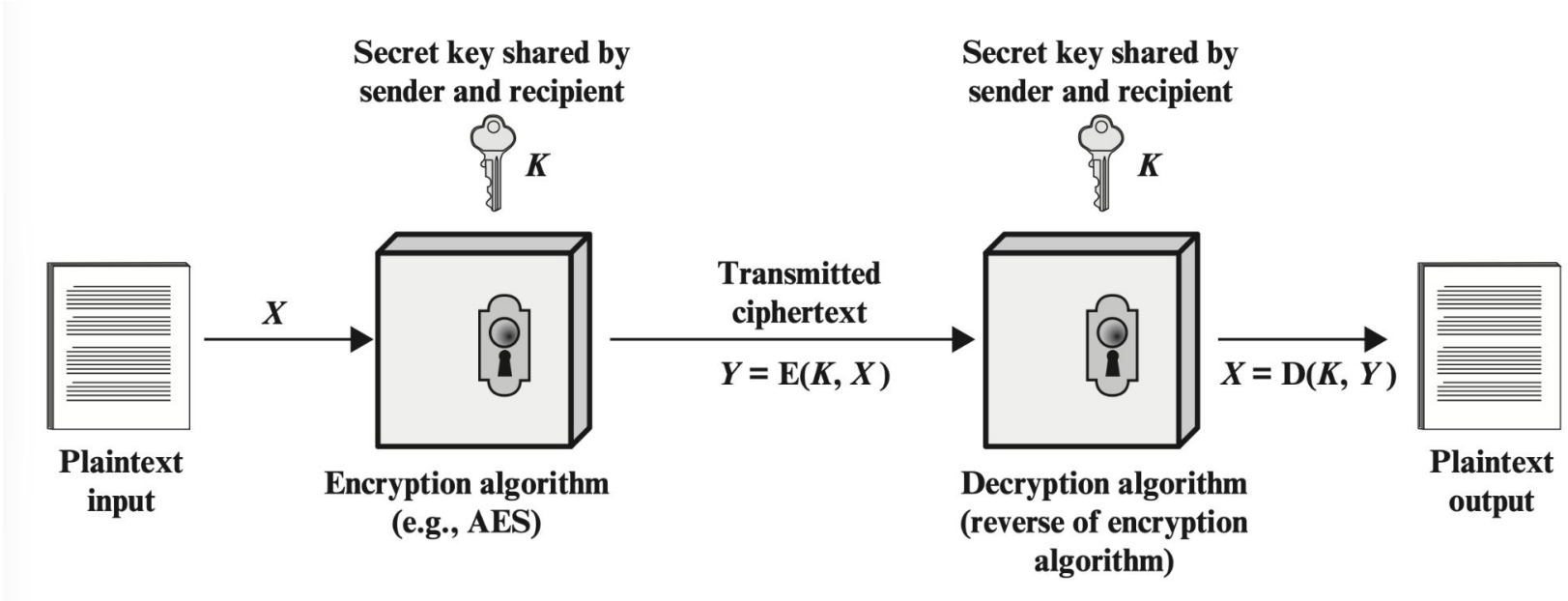
- Also known as **public key cryptography**
- Based on the hardness of some mathematical problems

Kerkhoff’s Principle

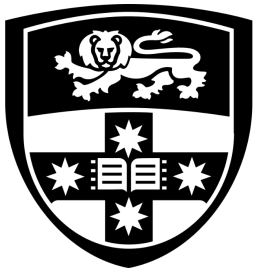
Kerkhoff’s Principle: The security of a given cryptographic mechanism must depend only on the security of the cryptographic keys used, but never on the mechanism or anything else, except the keys, being a secret

- Cryptographic mechanisms are usually publicly disclosed - often products of public competitions.
- Many eyes argument: the weaknesses of a mechanism are found faster if many experts analyze it.
- This principle has proved valuable time and again.
- There is a sad history of cryptographic designers abandoning the principle
  - Famous case: A5/1 and A5/2 for GSM. Initially secret, later found weak.
  - CSS encryption for DVDs (found weak)

Symmetric Cipher Model



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

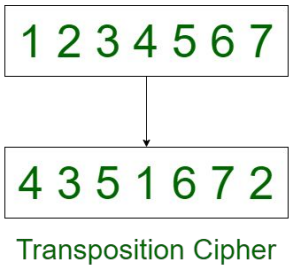


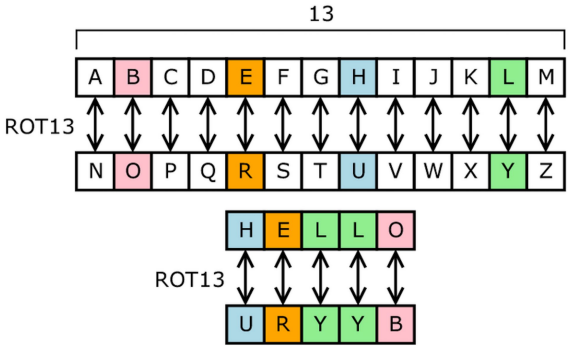
THE UNIVERSITY OF
SYDNEY

2. Historical Ciphers

Traditional Concepts

- Transposition and substitution
  - Transposition shifts the positions of symbols according to a rule
  - Substitution replaces symbols with others, according to a rule
  - Rule usually parameterized by key
  - Transposition and substitution can both be part of the same cipher





Substitution Techniques - Caesar Cipher

- Special case of monoalphabetic substitution – The simplest form of substitution cipher. A symbol from the plaintext alphabet is always mapped to the same symbol of the ciphertext alphabet

$$
\begin{aligned}
r_1 &\rightarrow s_5 \\
r_2 &\rightarrow s_2 \\
r_3 &\rightarrow s_9 \\
\ldots &\rightarrow \ldots \\
r_n &\rightarrow s_{17}
\end{aligned}
$$

Substitution Techniques - Caesar Cipher

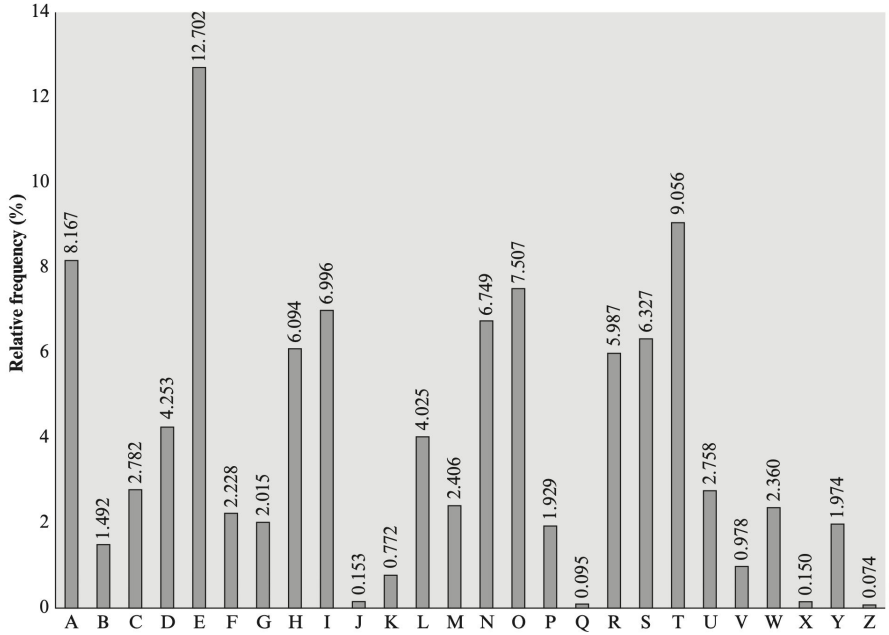
- Every letter is simply shifted by n positions, e.g., n = 3:

  $$a \rightarrow d, b \rightarrow e, c \rightarrow f, \ldots$$

- Roman historian Suetonius claimed this cipher was used by Gaius Iulius Caesar, but others used similar methods before

- Security by today's standards is very low, but the level of education thousands of years ago may have made it effective enough.

Problem - Frequency Analysis



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

- Every natural language has an associated characteristic frequency at which letters occur.
- This is preserved in the ciphertext. Makes it easy to deduce the mapping

Substitution Techniques - Vigenère Cipher

- Consider the plaintext "HELLO" and the keyword "KEY".
- For simplicity, we'll use an alphabet *A* of 26 letters (*N*=26) with numerical values assigned as *A*=0, *B*=1, ..., *Z*=25.
- **Given:**
  - Plaintext *P*= "HELLO" $\longrightarrow P_i$={7,4,11,11,14}
  - Keyword *K*= "KEY" $\longrightarrow K_j$={10,4,24}, where $|K|$=3
- **Encryption Process:**
  - To encrypt each letter P_i using the formula $C_i=(P_i+K_{i \bmod |K|}) \bmod 26$
- **Decryption Process:**
  - To decrypt each letter C_i using the formula $P_i=(C_i-K_{i \bmod |K|}+26) \bmod 26$

Substitution Techniques - Vigenère Cipher

Encryption Process:

To encrypt each letter P_i using the formula $C_i = (P_i + K_{i \bmod |K|}) \bmod 26$:

1. For $P_0 = H = 7$:
   - $C_0 = (7 + 10) \bmod 26 = 17$ (which corresponds to "R")
2. For $P_1 = E = 4$:
   - $C_1 = (4 + 4) \bmod 26 = 8$ (which corresponds to "I")
3. For $P_2 = L = 11$:
   - $C_2 = (11 + 24) \bmod 26 = 9$ (which corresponds to "J")
4. For $P_3 = L = 11$, note that we cycle back to the start of the keyword:
   - $C_3 = (11 + 10) \bmod 26 = 21$ (which corresponds to "V")
5. For $P_4 = O = 14$:
   - $C_4 = (14 + 4) \bmod 26 = 18$ (which corresponds to "S")

So, the ciphertext C for the plaintext "HELLO" encrypted with the keyword "KEY" is "RIJVS".

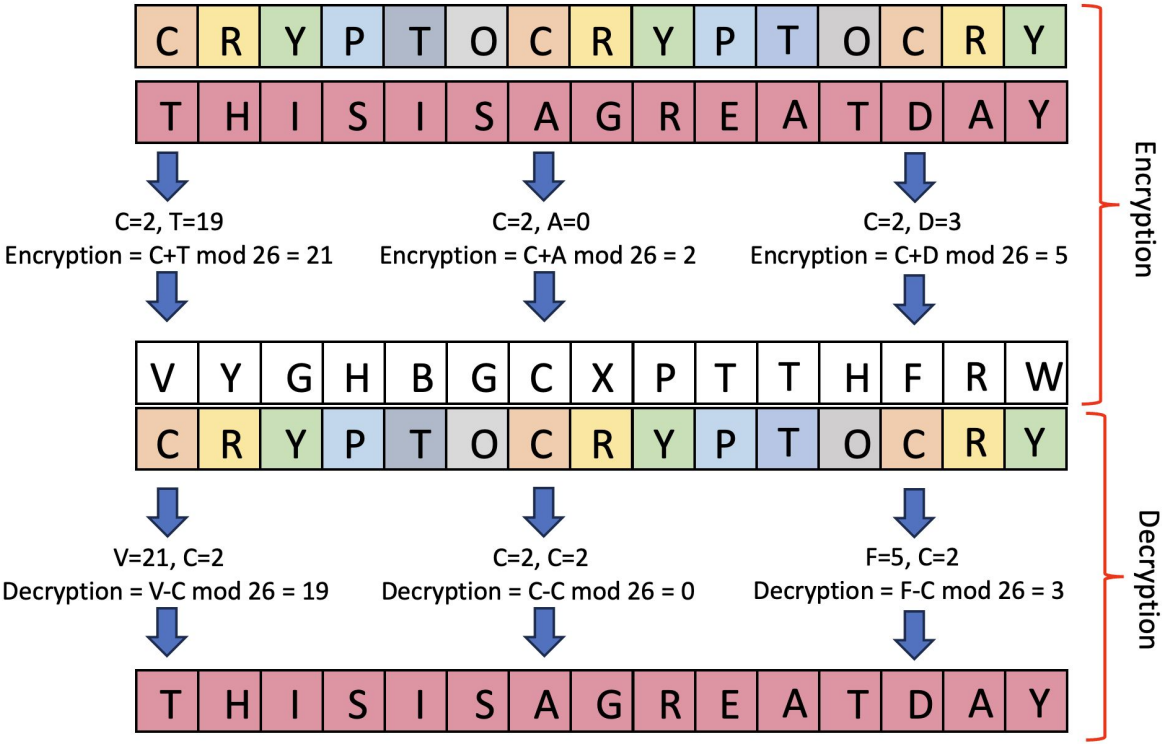
Decryption Process:

To decrypt each letter C_i using the formula $P_i = (C_i - K_{i \bmod |K|} + 26) \bmod 26$:

1. For $C_0 = R = 17$:
   - $P_0 = (17 - 10 + 26) \bmod 26 = 7$ (which corresponds to "H")
2. For $C_1 = I = 8$:
   - $P_1 = (8 - 4 + 26) \bmod 26 = 4$ (which corresponds to "E")
3. For $C_2 = J = 9$:
   - $P_2 = (9 - 24 + 26) \bmod 26 = 11$ (which corresponds to "L")
4. For $C_3 = V = 21$:
   - $P_3 = (21 - 10 + 26) \bmod 26 = 11$ (which corresponds to "L")
5. For $C_4 = S = 18$:
   - $P_4 = (18 - 4 + 26) \bmod 26 = 14$ (which corresponds to "O")

This decrypts the ciphertext "RIJVS" back to the original plaintext "HELLO".

Substitution Techniques - Vigenère Cipher



Breaking Vigenère Cipher

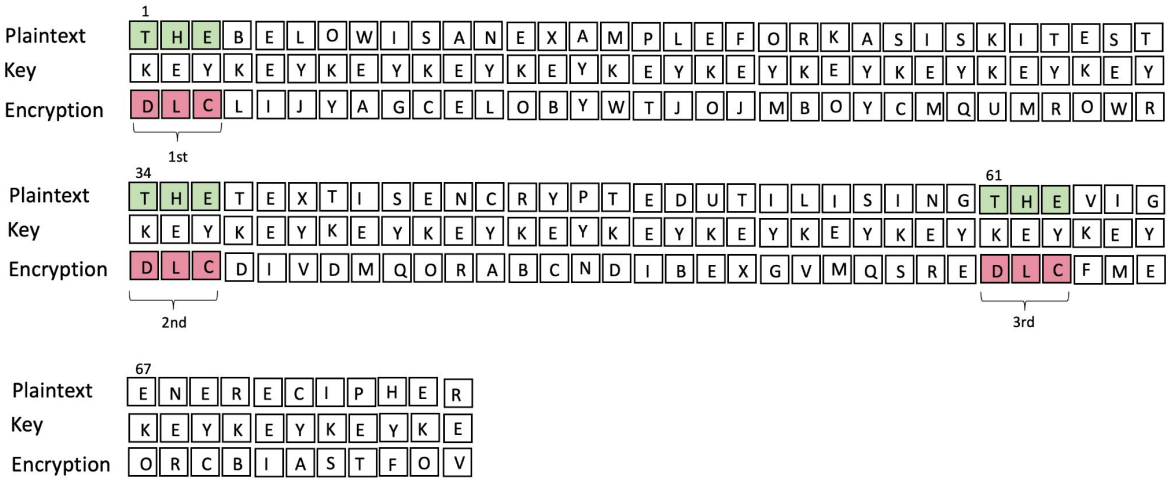
- The Vigenère cipher encrypts plaintext by applying a series of Caesar ciphers based on the letters of a keyword. For each letter of the keyword, it uses a different shift value (i.e., a different Caesar cipher).
- When analyzed separately, each monoalphabetic substitution (each Caesar cipher) used in the Vigenère cipher is relatively easy to break due to its simplicity.
- Several statistical tests exist to estimate d, e.g., Friedman test or Kasiski test

Breaking Vigenère Cipher

- Kasiski test is intuitive:
  - When the plaintext contains the same sequences of symbols at intervals of length k, then these are mapped to the same sequences of symbols in the ciphertext.
  - Search for such repetitions in ciphertext, note the intervals.
  - Greatest common divisor of all intervals is likely the key length
  - Example: https://crypto.interactive-maths.com/kasiski-analysis-breaking-the-code.html

Breaking Vigenère Cipher

- Assume our key is “KEY”
- And our plain text is “THE BELOW IS AN EXAMPLE FOR KASISKI TEST. THE TEXT IS ENCRYPTED UTILISING THE VIGENERE CIPHER.”



- The distance between the 1st and 2nd repetition is 33. Therefore the key size must be one of 1, 3, 11 and 33, which are factors of 33.
- The distance between the 2nd and 3rd repetition is 27. Therefore the key size must be one of 1, 3, 9 and 27.
- The common factors are 1 and 3. We can ignore 1 as it is a very weak key that can be brute-forced.
- So the possible key size is 3.

Transposition Techniques: Rail Fence Cipher

- The plaintext is written down as a sequence of diagonals and then read off as a sequence of rows
- Example: “MEET ME AFTER THE TOGA PARTY” with a rail fence of depth 2

M E M A T R H T G P R Y
E T E F E T E O A A T

The encrypted message is: MEMATRHTGPRYETEFETEOAAT

From Classic to Modern

- Polyalphabetic ciphers addressed an observed shortcoming: frequency analyses (in many forms) can yield the key
- Shannon identified two design goals for modern ciphers:
  - **Diffusion:** must blur statistically significant characteristics in plaintext
    - Every ciphertext symbol must depend on many plaintext symbols
    - On average, a bit flip in plaintext should change half the ciphertext bits
  - **Confusion:** must blur dependency between ciphertext and key
    - Each symbol in ciphertext must depend on several symbols in key

One Time Pad (OTP)

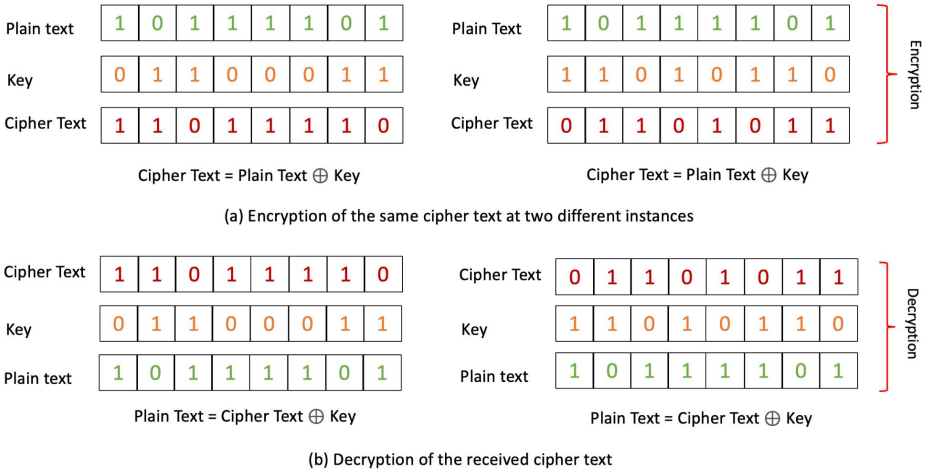
This is a symmetric cryptographic scheme. Not be confused with the OTP (One Time Password) that is used in Multi-Factor-Authentication (MFA).

Perfect security: given a ciphertext, any possible plaintext is equally likely the correct one

- One-Time Pad o: random sequence of bits with $|o| \geq |p|$, i.e., at least as long as the plaintext
- Ciphertext: $c = o \oplus p$ (XOR)
- OTP is perfectly secure if, and only if:
  - o is truly random (more later!)
  - $|o| \geq p$
  - Never used twice, not even in part

- OTPs have been used on occasion, but only in extremely high-value situations (e.g., red telephone). Why? Conditions on o are usually **much too hard** to achieve - OTP is impractical

One Time Pad (OTP) Example



- OTP is not secure if the key is repeated.
  - Let's consider a One-Time Pad that uses XOR.
  - And consider two plain texts P_1 and P_2 encrypted by the same key K
  - So we get $C_1 = P_1 \oplus K$ and $C_2 = P_2 \oplus K$
  - Now assume the attacker gets hold of the two cipher texts, C_1 and C_2. The attacker can build a new string P where $P = C_1 \oplus C_2$, which can be simplified.

    $P = C_1 \oplus C_2$

    $P = (P_1 \oplus K) \oplus (P_2 \oplus K)$

    $P = P_1 \oplus P_2$ (because $K \oplus K$ is all zeros)

    This also means that $P_1 = P \oplus P_2$ and $P_2 = P \oplus P_1$
  - Using this information, the attacker can try "crib dragging". The idea is to guess a word that can appear in one of the plan texts. For example, if it is English, and the text is sufficiently large, it is reasonable to assume the word "the" will be there somewhere in the plain text. So the idea is to try "the" in all possible positions in one of the plain texts, XOR them with P and see whether some readable can be obtained. The process continues like this.

http://travisdazell.blogspot.com/2012/11/many-time-pad-attack-crib-drag.html

Attacks on Cryptography

- A cryptographic system is **compromised**
  - If an adversary can obtain plaintexts for certain ciphertexts, or
  - An adversary can obtain decryption key k_d when given k_e

- **Cryptanalysis**: decrypting without knowledge of key

- Several forms of attacks define the strength of the adversary:
  - **Ciphertext-only** - Attacker may only see ciphertexts
  - **Known plaintext** - Attacker may see some plaintexts and corresponding ciphertexts
  - **Chosen plaintext** - Attacker may choose a plaintext and see the corresponding ciphertext
  - **Chosen ciphertext** - Attacker may choose ciphertext and decrypted plaintext

3. Stream vs. Block Ciphers

Stream Ciphers

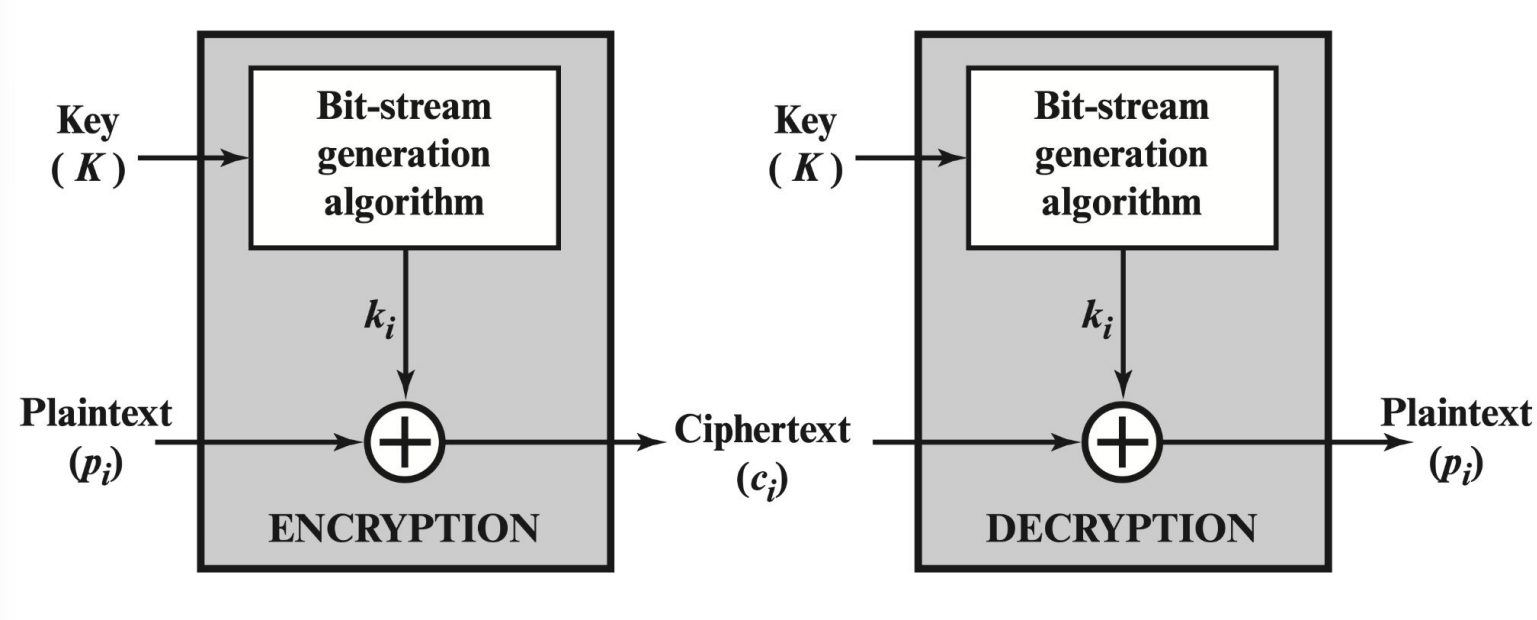
- Pseudo-random functions to generate keystream from seed

Idea: generate a key stream from the secret key

- Apply some f (k) to obtain a stream b = $b_0\ b_1\ b_2\ \ldots\ b_n$ of bits
- Apply a combination function on plaintext p = $p_0\ p_1\ p_2\ \ldots\ p_n$ and $b_0\ b_1\ b_2\ \ldots\ b_n$ - commonly $\oplus$ (XOR)
- Ciphertext is then simply: c = b $\oplus$ p
- Challenge: keystream must be as 'unpredictable' as possible without knowledge of k

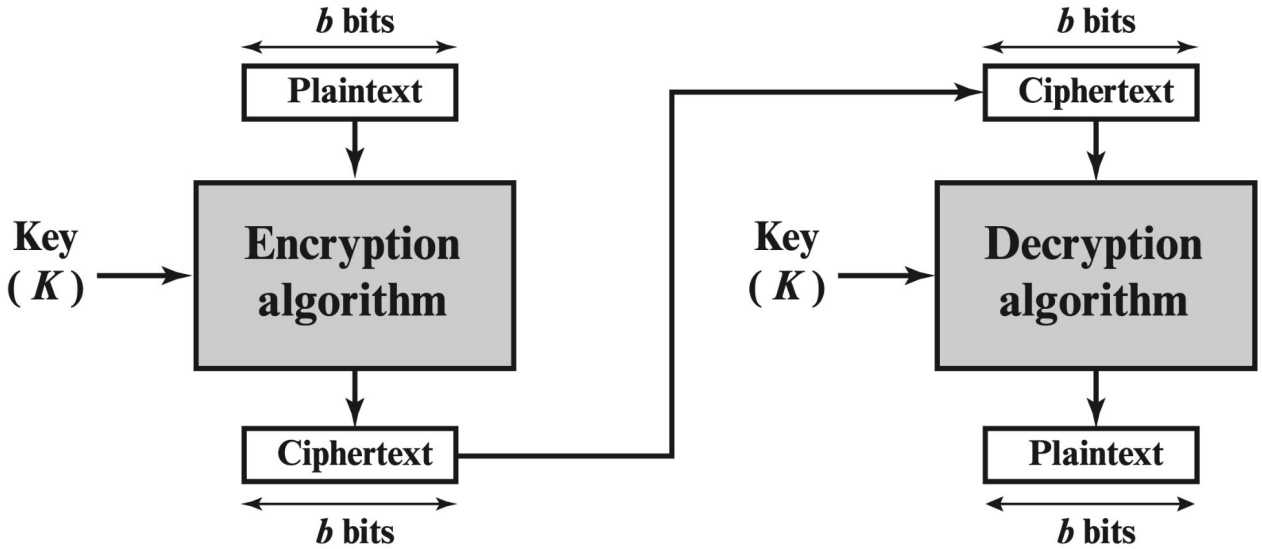
Rephrasing this: it must be infeasible to determine whether the key stream is random or deterministic - the latter would imply there is a pattern!

Stream Ciphers



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

Block Ciphers



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

Block Ciphers

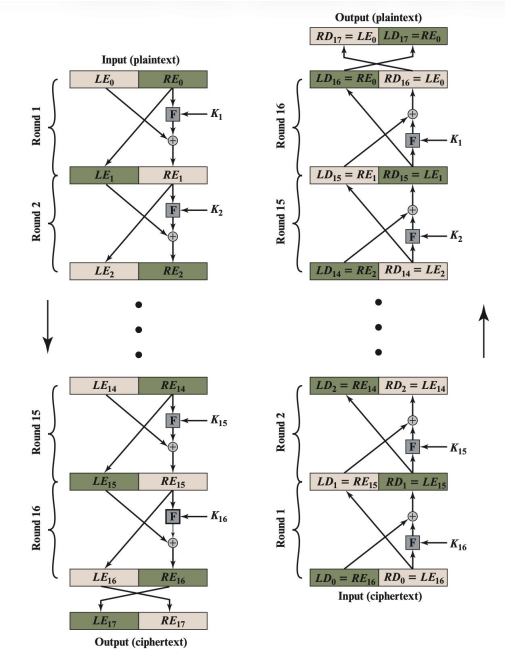
- Substitution implemented via Substitution box **(S-boxes)**
- Permutation implemented via Permutation box **(P-boxes)**

  → We will discuss S-boxes and P-boxes in DES

- Multiple rounds of executing the algorithm to improve **diffusion and confusion**
- Rounds arranged in so-called ‘networks’

Example 1: Feistel Cipher

- Define the steps of a block cipher
- Multiple rounds
- All rounds in the Feistel cipher have the same structure:
  - **Substitution:** is performed on the left half of the data
  - **Permutation:** is performed that consists of the interchange of the two halves of the data
- Used in many block ciphers: DES, Blowfish, Twofish, RC6



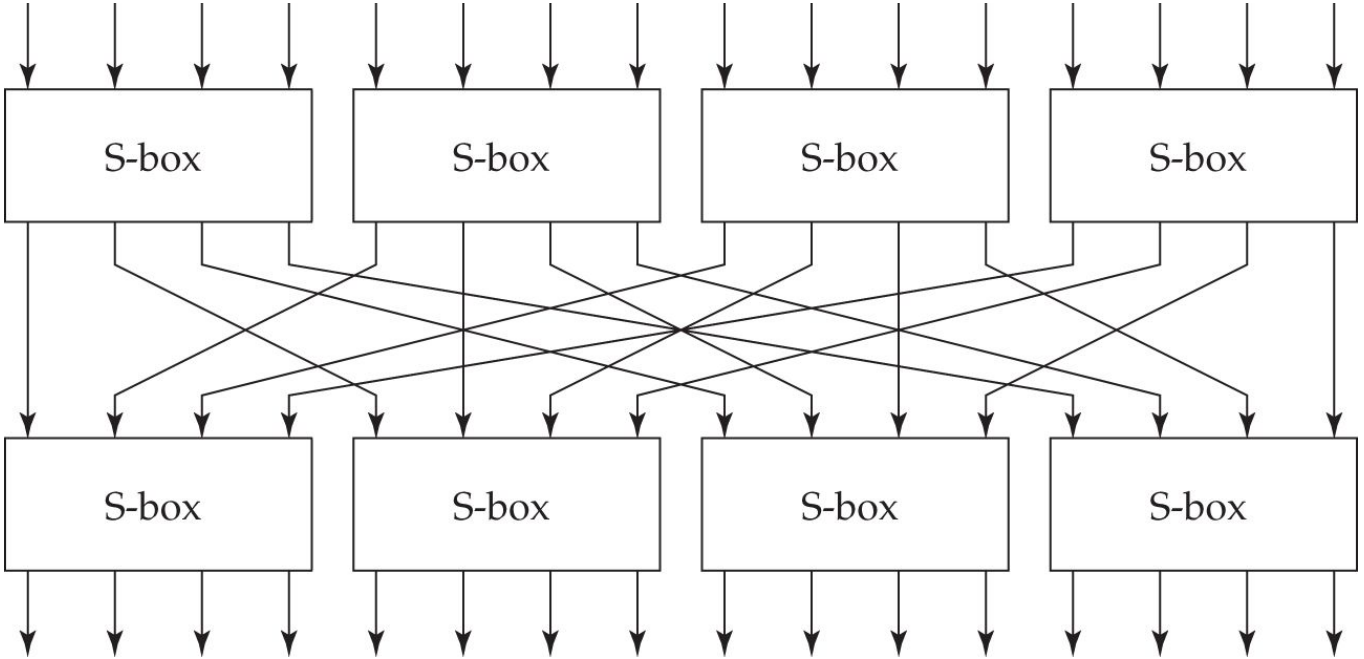
Source: Cryptography and Network Security (Seventh Edition - William Stallings)

Feistel Cipher: Encryption and Decryption

Source: Cryptography and Network Security (Seventh Edition - William Stallings)

Example 2: Substitution-Permutation Network (SPN)

- Alternative network structure.



4. DES and 3DES

Data Encryption Standard (DES)

- Published in 1977, standardized in 1979
- Key in each round: **subkey** derived from the actual key
- Total of 8 S-boxes
- Outputs from **S-boxes** are sent through one **P-box** to achieve good spread over S-boxes **in the next round** (and a bit more that is not important for our purposes)
- It can be shown that the number of rounds, 16, is indeed necessary to achieve security

S-Boxes

- Look-up tables: map of block of bits to and output block of bits
  - Mapping is carefully chosen and vetted
- How to make sure that decryption is possible?
  - Either S-boxes themselves are invertible, or
  - Their arrangement in a cipher is invertible as a whole
- Changing one bit in input of S-box will map to an output block where, on average, half of all bits are flipped (diffusion)
- We give an example from one of the most famous standards: Data Encryption Standard (DES)
  - Note: DES was secure in its day; today the key length is too short
  - State-of-the-art: Advanced Encryption Standard (AES)
  - Also uses S-boxes, but differently

S-Box S[0] from DES

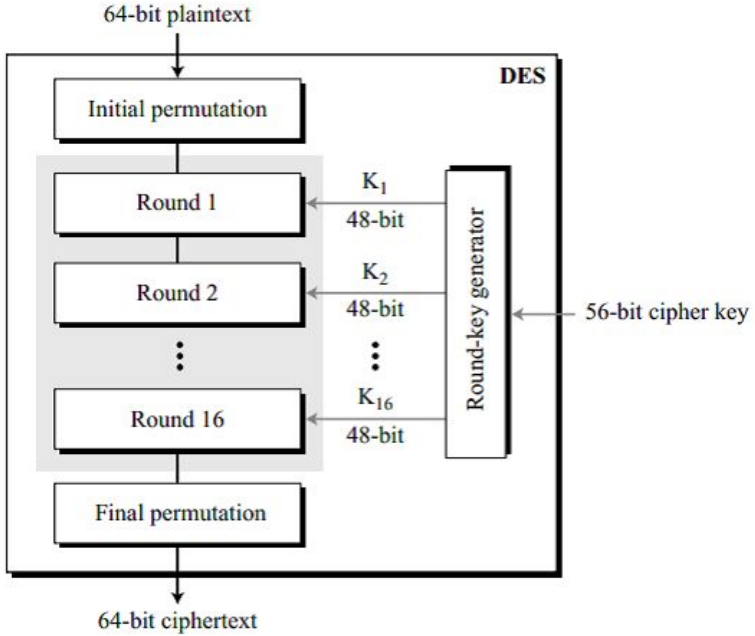
Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

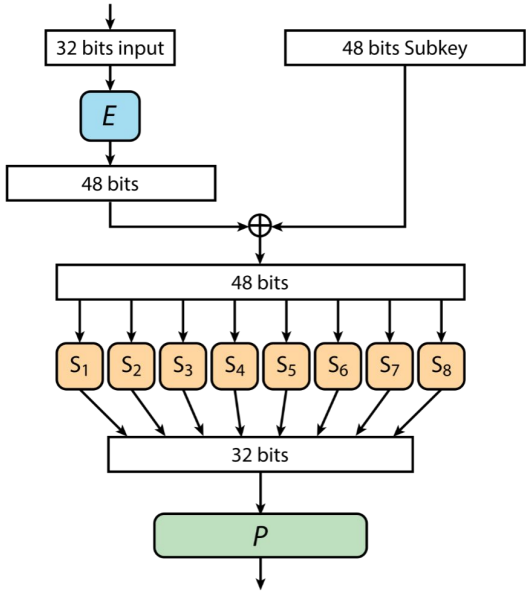
- **Used like this:**
  - Input string = $(b_0, b_1, b_2, b_3, b_4, b_5)$
  - b_0 and b_5 give row; (b_1, b_2, b_3, b_4) give column
  - Interpret the number in that cell as binary
- **Example:**
  - Input = (1, 1, 0, 1, 1, 1) $\rightarrow (110111)_2$
  - Row: $(11)_2 = (3)_{10}$; Column: $(1011)_2 = (11)_{10}$
  - Output: $(14)_{10} = (1110)_2 \rightarrow$ **1110**

P-Boxes

- Transposition is defined in P-boxes
- These are simple index permutations: re-orderings
- Often, P-boxes are used together with S-boxes
  - Create chain: S-boxes $\longrightarrow$ P-box $\longrightarrow$ S-boxes, etc.
  - P-box takes the output from S-boxes
  - P-box distributes bits of **one** S-box on to as many of the following S-boxes as possible

DES Encryption and Decryption





https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm

Source: Wikipedia

The Strength of DES vs. 3DES vs. AES

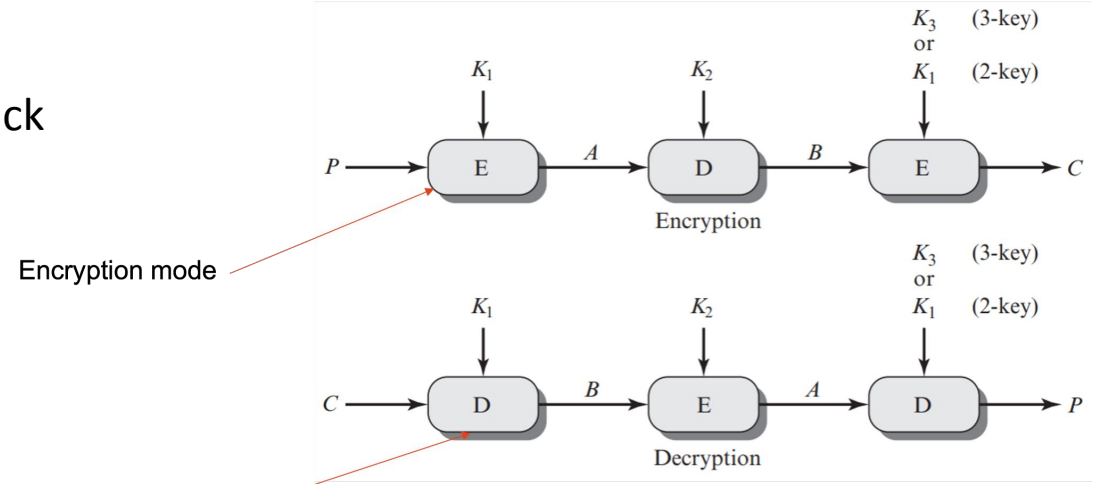
- 56-bit keys:
  - 2^{56} possible keys = 7.2 x 10^{16} keys
  - However, a rate of 1 billion = 10^9 keys combinations per second is reasonable for today's multicore computers
  - Supercomputer technology can run a rate of 10^{13} encryptions per second

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^9 years	6.3×10^6 years

Source: Cryptography and Network Security (Seventh Edition - William Stallings)

3DES Encryption and Decryption

- Purpose: DES replacement
- 168-bit key, no brute-force attack
- Has 2 or 3 keys



Block Cipher Design Principles

- **Number of rounds:** The greater the number of rounds, the more difficult it is to perform cryptanalysis, even with a weak **function F** - provides the element of confusion in a Feistel cipher

- **Design of function F:**
  - Nonlinear
  - Good avalanche properties algorithm (strict avalanche criterion (SAC))
  - Bit independence criterion (BIC)
  - SAC and BIC appear to strengthen the effectiveness of the **confusion function.**

- **Key Schedule Algorithm:** Should guarantee key/ciphertext SAC and BIC

5. Advanced Encryption Standard (AES)

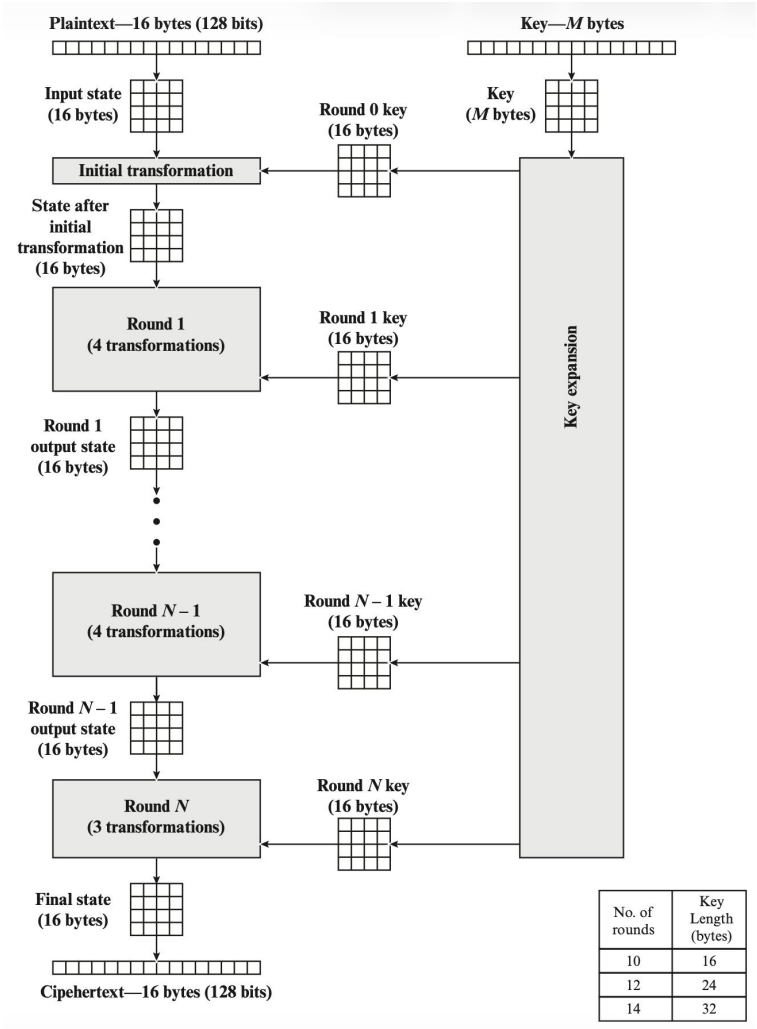
Finite Field Arithmetic

- AES uses arithmetic in the finite field GF(2^8) with the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$
- Multiplication of two bytes is defined as multiplication in the finite field GF(2^8)
- Addition of two bytes is defined as the bitwise XOR operation.
- Example: $A = (a_7\ a_6 \ldots a_1\ a_0)$ and $B = (b_7\ b_6 \ldots b_1\ b_0)$.
  - The sum $A + B = (c_7\ c_6 \ldots c_1\ c_0)$, where $c_i = a_i$ XOR b_i

AES

- Current de-facto block cipher on the Internet
- Chosen in an open competition to be the successor of DES
- Operates at key lengths of 128, 192, and 256 bit
- Uses S-Boxes and P-Boxes
- The S-Boxes actually describe a mathematical function that is believed to achieve excellent diffusion and confusion
- Cryptanalysis attempts aimed to describe this function in easier terms, but so far have all failed
- **Fun fact:** AES is not a cipher. The **Rijndael cipher** has been selected as the Advanced Encryption Standard (AES).

AES Structure



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

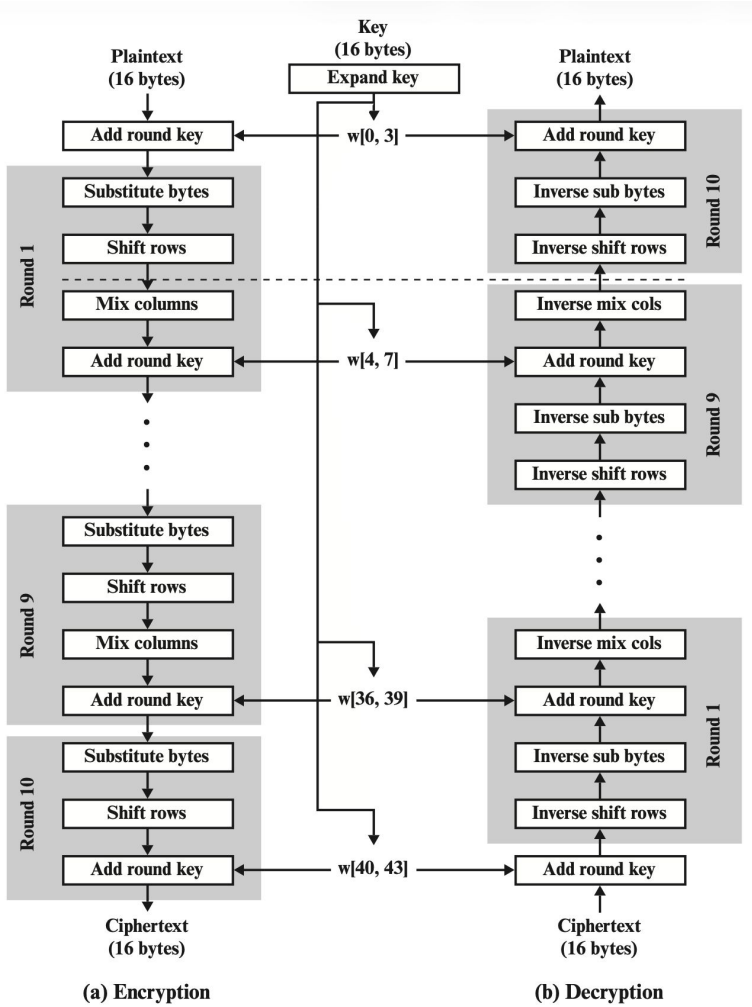
AES Encryption

- The key is expanded into an array of 44, 32-bit words **w**[i] using the **key expansion algorithm** – takes 16-byte key (4 words) and produces a linear array of 176-byte (44 words)

- Round key is 4 distinct words for AES-128, 6 words for AES-192, and 8 words for AES-256

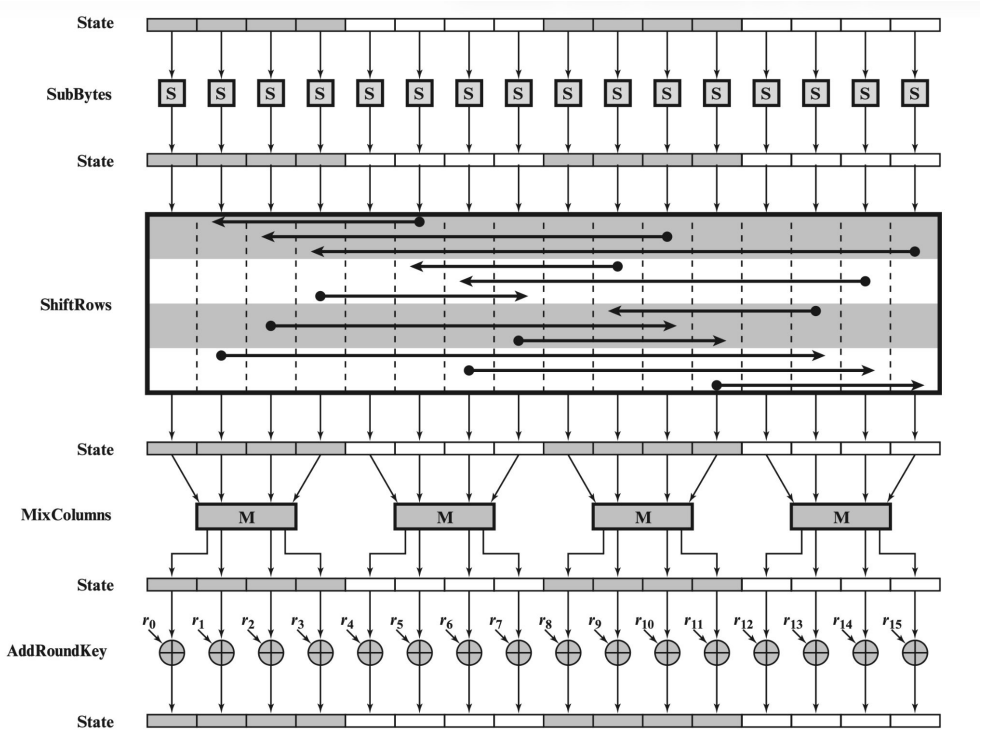
- Four different stages are used, one of permutation and three of substitution:
  - **Substitute bytes**: Uses an S-box to perform a byte-by-byte substitution of the block.
  - **ShiftRows**: A simple permutation.
  - **MixColumns**: A substitution that makes use of arithmetic over GF(2^8).
  - **AddRoundKey**: A simple bitwise XOR of the current block with a portion of the expanded key.

AES Encryption and Decryption



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

AES Encryption Round

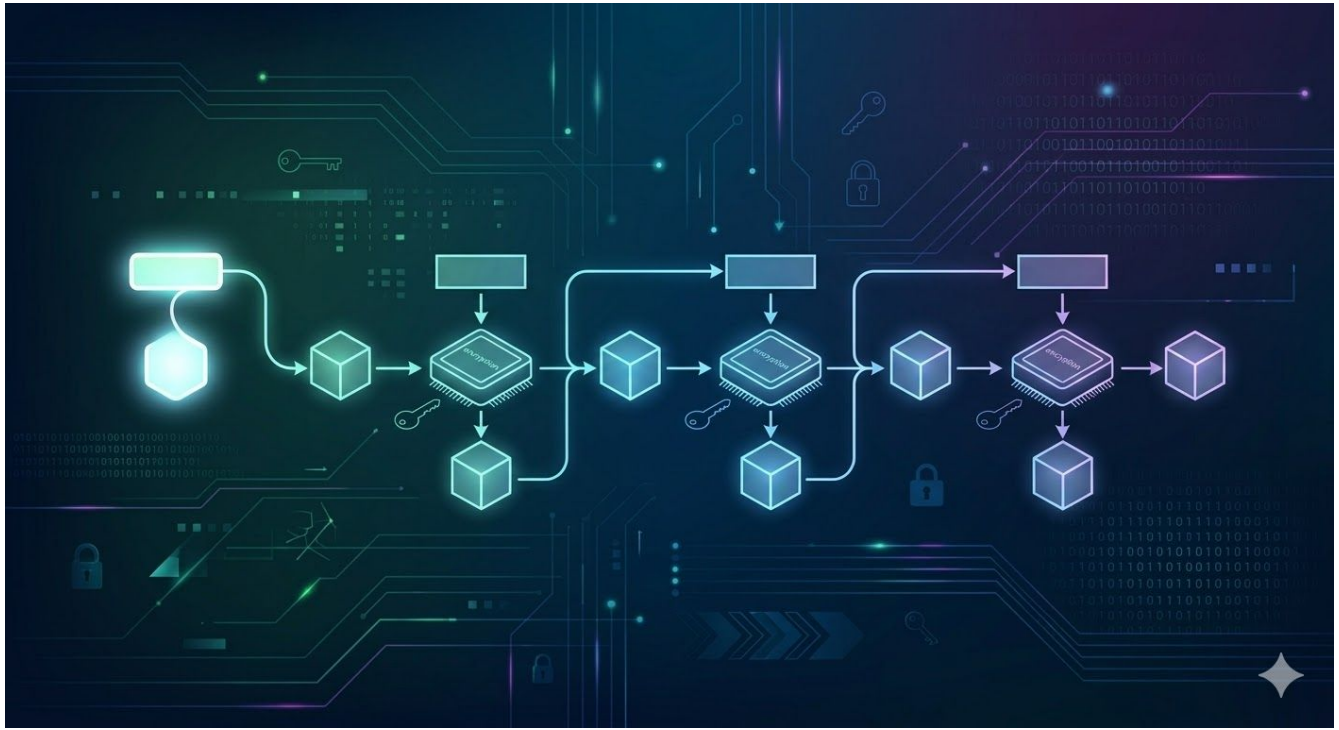


Source: Cryptography and Network Security (Seventh Edition - William Stallings)

Watch this Later

https://www.youtube.com/watch?v=gP4PqVGudtg

6. Block Modes

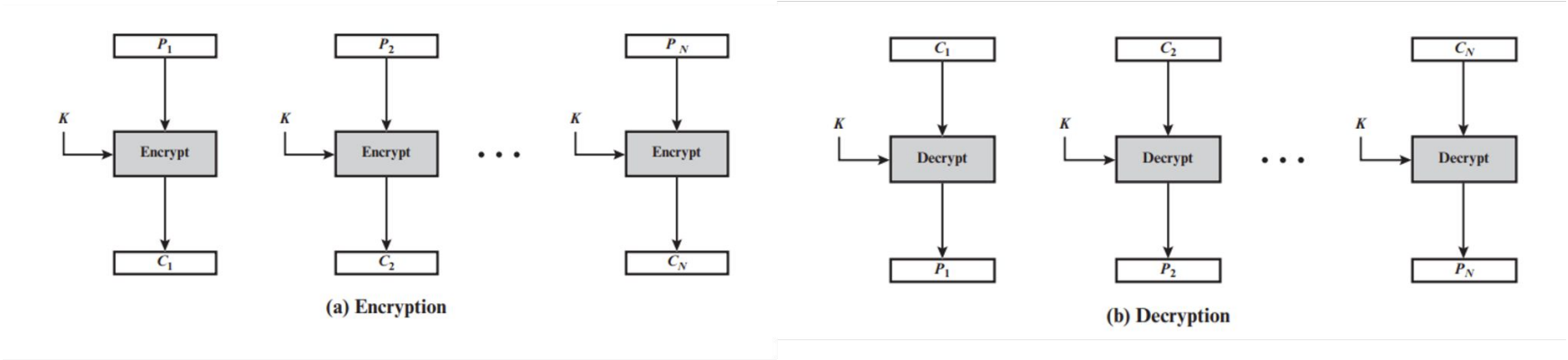


Block Modes for 3DES and AES

- Ciphers must handle messages of arbitrary length
- Solution: split messages in block and process them according to a **cipher block mode**.
- These modes of operation can introduce new security problems if not designed and **used** properly.
- **Classic block modes** only encrypt data
  - E.g., Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter (CTR), …
- Modern modes provide **authenticated encryption (AE, AEAD)**
  - Combine encryption and integrity protection
  - E.g., Galois Counter Mode (GCM), Counter-with-CBC-MAC (CCM) mode, …

Electronic Code Book Mode - ECB

- Block-wise: $c_i = Enc_k(m_i)$



Source: Cryptography and Network Security (Seventh Edition - William Stallings)

The Problem with ECB

Figure: Before ECB encryption.

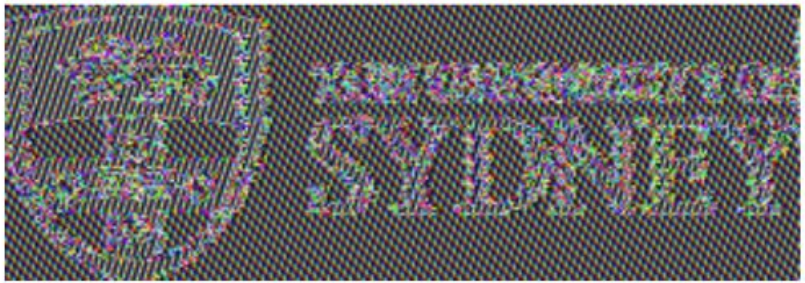
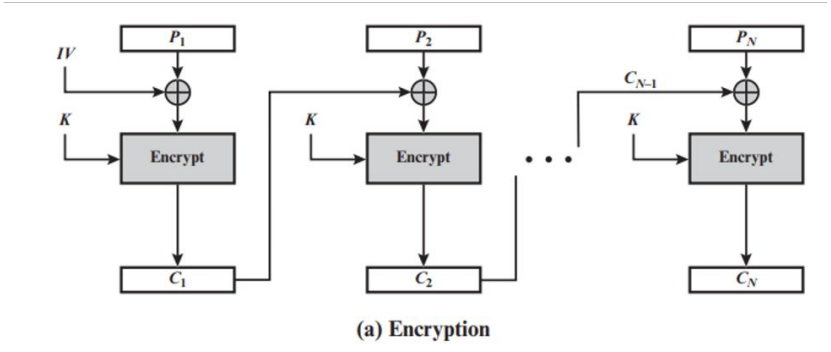


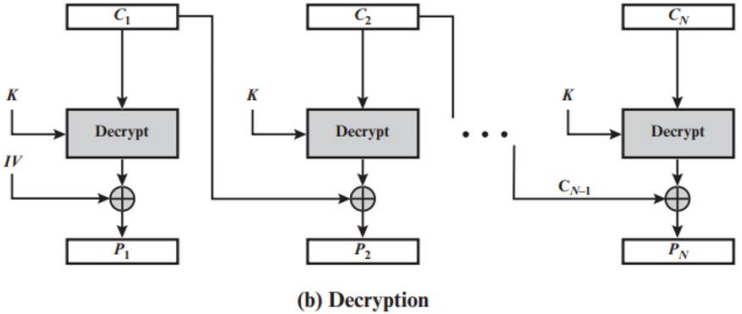
Figure: After ECB encryption.

Simple message: good for teaching. Beyond that, do not use it

Cipher Block Chaining (CBC)

- CBC encrypt: $c_i = Enc_k(c_{i-1} \oplus m_i)$
- CBC decrypt: $m_i = Dec_k(c_i) \oplus c_{i-1}$





Source: Cryptography and Network Security (Seventh Edition - William Stallings)

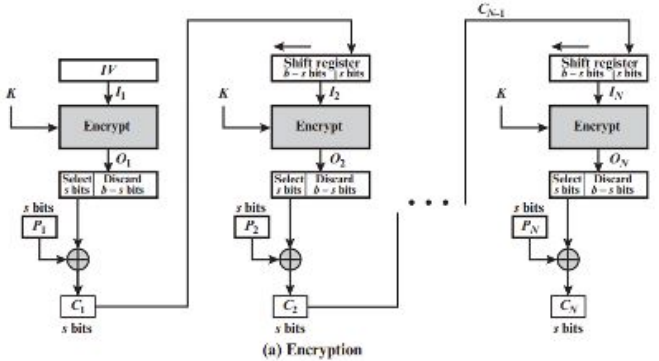
After using CBC

THE UNIVERSITY OF SYDNEY

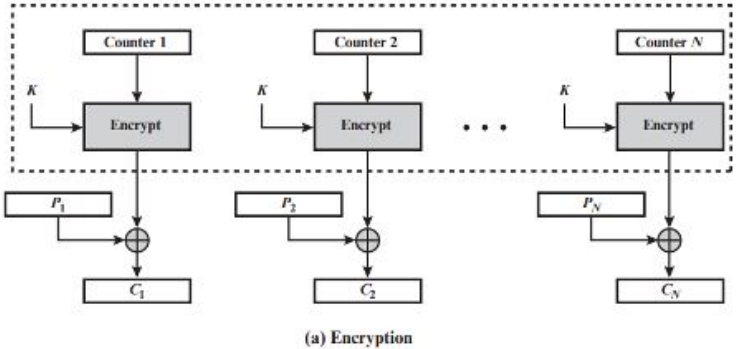
Figure: Before ECB encryption.

Figure: After CBC encryption.

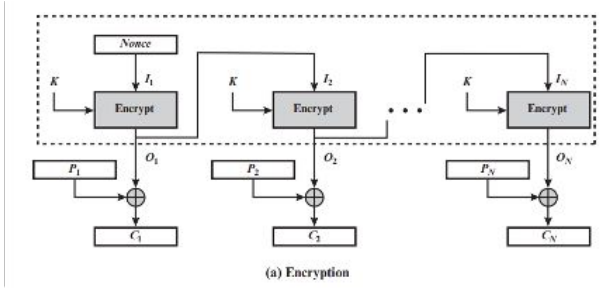
Other Block Modes



Cipher Feedback Mode (CFB)



Counter Mode (CTR)



Output Feedback Mode (OFB)

7. Pseudorandom Number Generation

The Use of Random Numbers

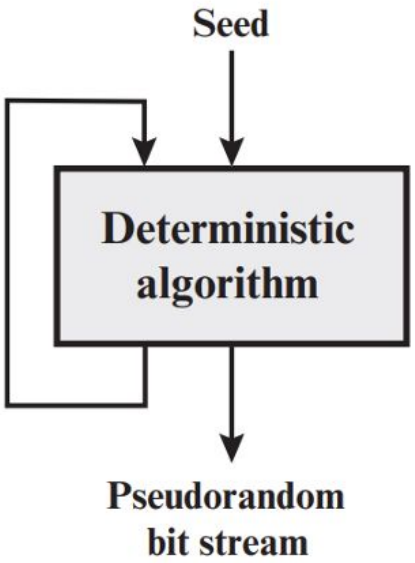
- Key distribution and reciprocal authentication schemes
- Session key generation
- Generation of keys for the RSA public-key encryption algorithm (next week’s lecture)
- Generation of a bit stream for symmetric stream encryption (mentioned above)

CSPRNG

Cryptographically Secure Pseudo-Random Number Generator

- A Pseudorandom Number Generator **(PRNG)** outputs a **deterministic** sequence of numbers
  - Takes a seed value as the start value
  - Output sequence depends on the seed
- A PRNG is cryptographically secure, i.e., a **CSPRNG**, if:
  - It is computationally infeasible to brute-force the seed value (try all possible values) to correctly predict the output, and
  - It is computationally infeasible to distinguish the output sequence from true randomness
- Only option for the attacker is, in other words, to know the **seed**

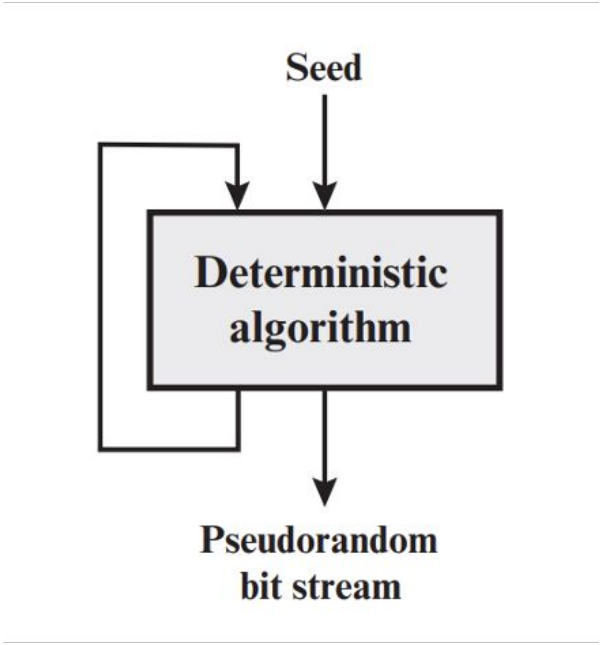
PRNG

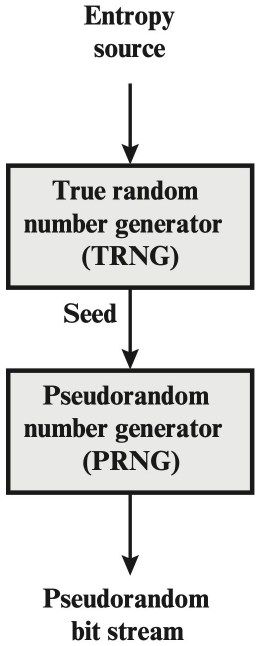


So, how to generate "Seed"?

Source: Cryptography and Network Security (Seventh Edition - William Stallings)

PRNG



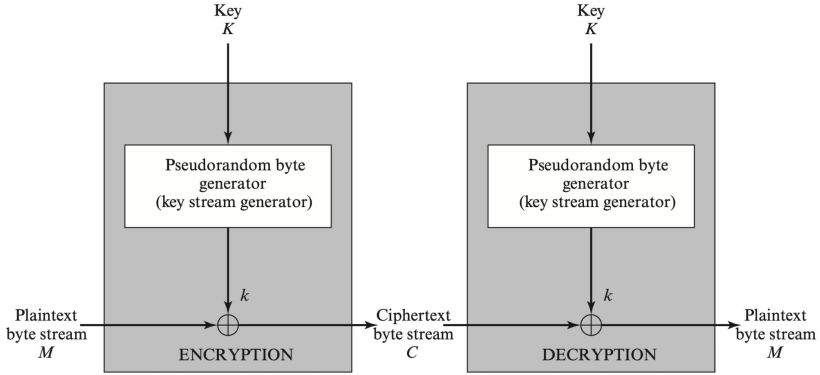


Source: Cryptography and Network Security (Seventh Edition - William Stallings)

The use of PRNG in Stream Ciphers

ChaCha20 and Salsa20

- Family of related ciphers by DJ Bernstein
- Current de-facto standard for stream ciphers on the Internet
- Basis of CSPRNG in OpenBSD and Linux



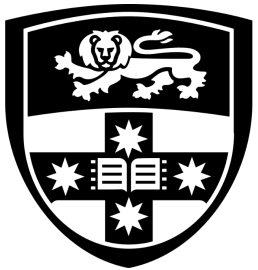
RC4

- Used to be de-facto standard for TLS/SSL
- Feasible breakage in 2013
- **Avoid today**

Recap

We discussed:

- Basic terminology: Plaintext, Ciphertext and Key
- Kerckhoffs' Principle
- Historical ciphers
- Traditional ideas of symmetric encryption - substitution and transposition
- Information theoretic view - confusion and diffusion
- One Time Pad
- Block Ciphers and Stream Ciphers
- S-Boxes, P-boxes, and Feistel Networks
- DES and AES
- Cipher Block Modes
- Pseudorandom number generator



THE UNIVERSITY OF
SYDNEY