Privacy and Network Security

Lecture 4 – Applied Cryptography Part 1

Learning Objectives

- After this lecture you will be able to:
  - Explain basics of cryptography
  - Discuss random number generators
  - Discuss symmetric encryption algorithms
  - Discuss different attack models to encryption mechanisms

Chapter 3, 4, 6, 7, 8 of Cryptography and Network Security

Random and Pseudorandom Numbers

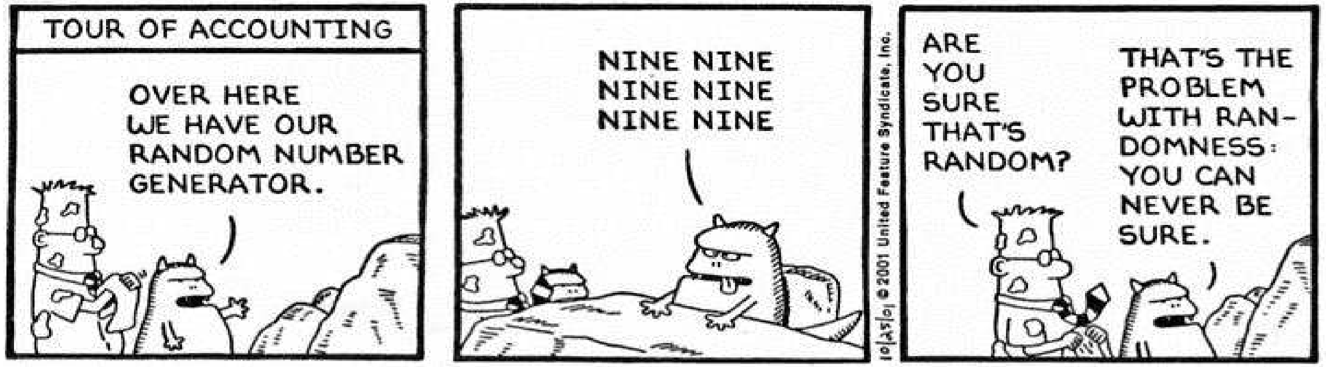
A number of **network security** algorithms based on cryptography make use of **random numbers**

- Generation of keys for **public-key** algorithms, such as RSA encryption.
- Generation of a stream key for symmetric **stream cipher**
- Generation of a symmetric key for use as a **temporary session key** in a number of networking applications, such as Transport Layer Security, Wi-Fi, e-mail security, and IP security
- A number of **key distribution** scenarios, such as Kerberos, random numbers are used for handshaking to prevent replay attacks.

Requirements of Random Numbers

1. Randomness
   - Criteria for validating that a sequence of numbers is random:
     - **Uniform distribution:** The distribution of bits in the sequence should be uniform; that is, the frequency of occurrence of ones and zeros should be approximately the same.
     - **Independence:** No one subsequence in the sequence can be inferred from the others.

2. Unpredictability
   - An attacker should not be able to predict future elements of the sequence on the basis of earlier elements



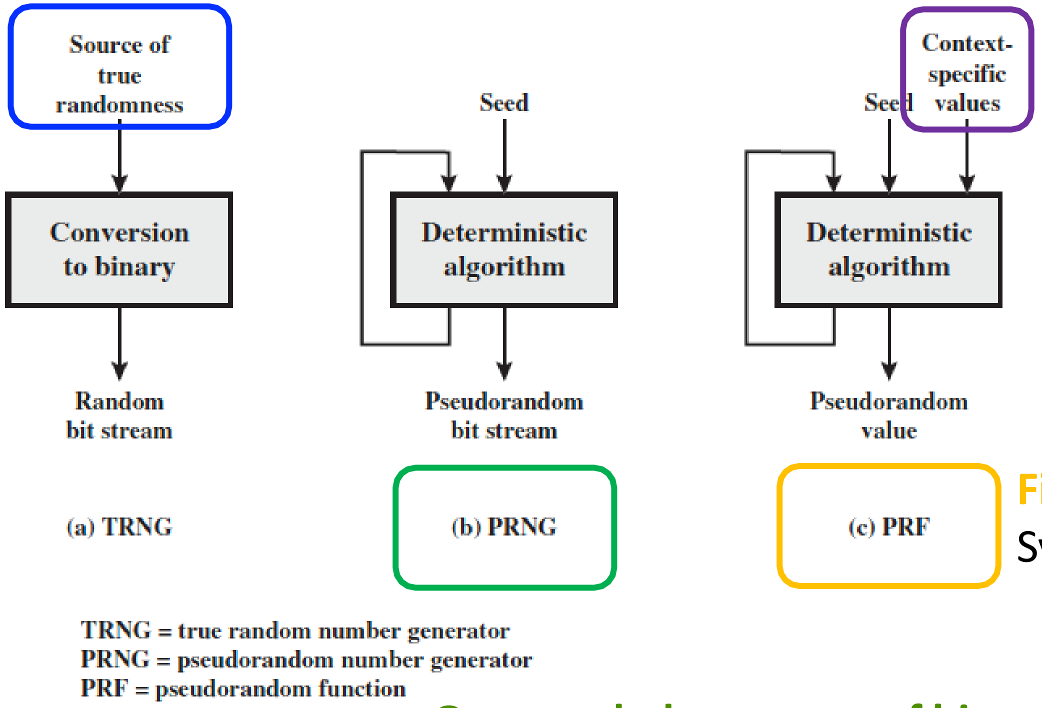
Pseudorandom Numbers

- Using **deterministic** algorithms we can generate a sequence of numbers that are **NOT** statistically random but they will pass reasonable tests of randomness

Entropy source

Taken from physical environment of the computer, e.g., mouse, keyboard, …

Conversion of analogue to binary



User ID, Application ID, …

Fixed length stream of bits

Symmetric encryption key

Open ended sequence of bits

Symmetric stream cipher

Terms related to encryption

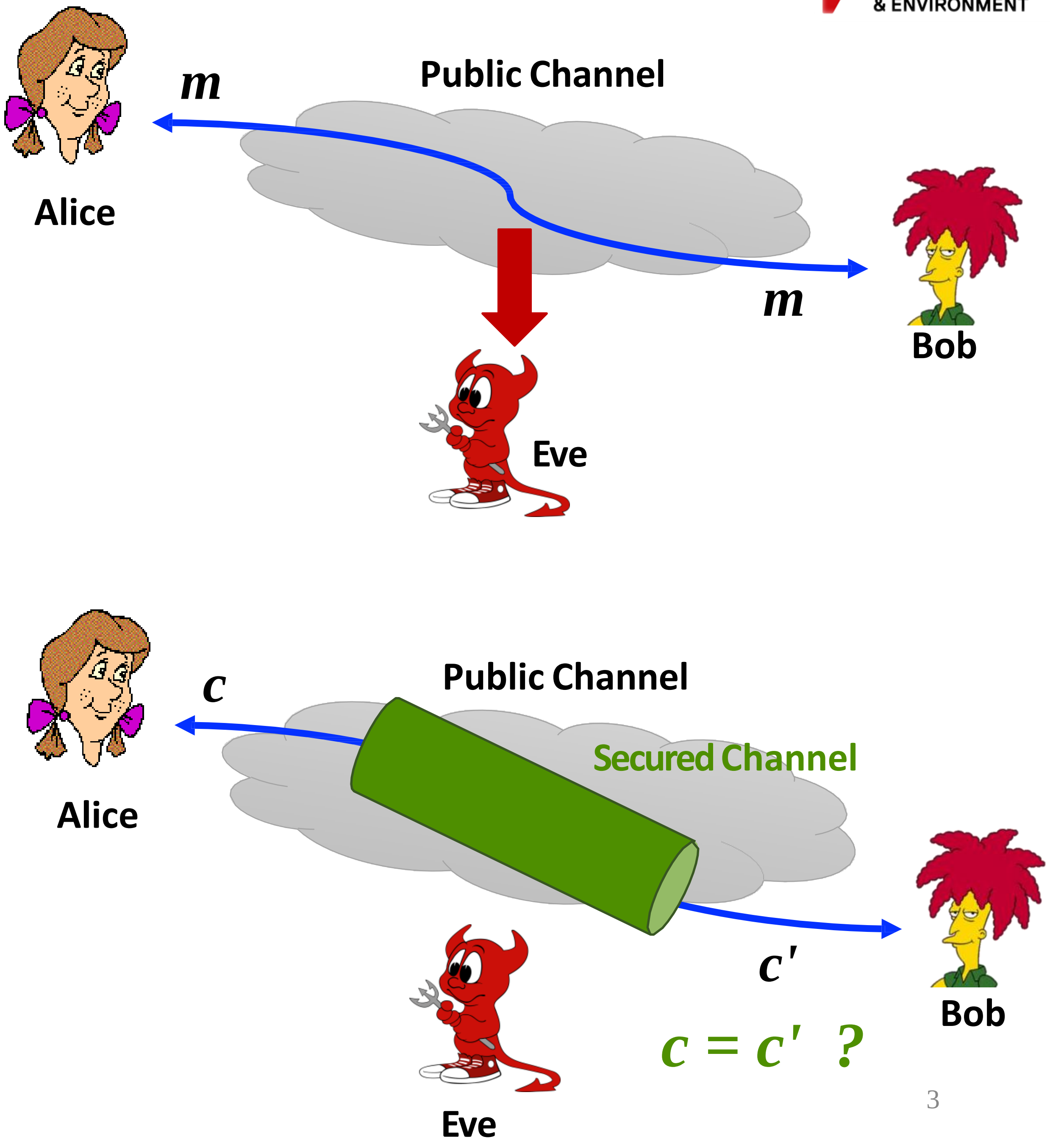
- Plaintext: an original message
- Ciphertext: the coded message, coded from plaintext
- Encryption or enciphering: the process of converting from plaintext to ciphertext
- Decryption or deciphering: the process of restoring the plaintext from the ciphertext

- The many schemes used for encryption constitute the area of study known as **cryptography**.
- Such a scheme is known as a **cryptographic system or a cipher**.
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code."
- The areas of cryptography and cryptanalysis together are called **cryptology**.

University of Salford MANCHESTER

SCHOOL OF SCIENCE, ENGINEERING & ENVIRONMENT

- **Actors:** Alice, Bob, Eve
- Elements of the **Cryptosystem:**
  - Plaintext: m
  - Ciphertext: c
  - Set of keys: K
    - *Encryption key:* k_e
    - *Decryption key :* k_d
  - Encryption function: E
    - $E(m, k_e) = c$
  - Decryption function: D
    - $D(c, k_d) = m$



Classification

1. **The type of operations used to transform plaintext to ciphertext**
   - **Substitution:** Each element is replaced with another element (letter, number, symbol)
   - **Transposition:** Elements of the text are rearranged
2. **The keys used**
   - **Symmetric:** Sender and receiver have the same key (single-key, secret-key)
   - **Asymmetric:** Sender and receiver each use a different key (public-key)

3. The way in which the plaintext is processed

- **Block cipher:** Processes one block of data at a time
- **Stream cipher:** Processes the input continuously

Requirements

1. Sender and receiver must have **a copy of the key(s)** and must keep it secure
2. Strong encryption algorithm - The attacker should be unable to decrypt the ciphertext or discover the key even if she knows a number of plaintexts and the corresponding ciphertext
3. No information should be lost! All operations should be reversible.

Security of symmetric encryption depends on the **secrecy of the key**, **NOT** the secrecy of the algorithm

University of Salford MANCHESTER

SCHOOL OF SCIENCE, ENGINEERING & ENVIRONMENT

Cryptanalysis

The process of attempting to discover the plaintext or the key

Type of Attack	Known to Cryptanalyst
Ciphertext only	• Encryption algorithm • Ciphertext to be decoded
Known plaintext	• Encryption algorithm • Ciphertext to be decoded • One or more plaintext–ciphertext pairs formed with the secret key
Chosen plaintext	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	• Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Bruteforce attack!
Background knowledge needed, e.g. what kind of plaintext you are looking for

Certain plaintext pattern!

Security of encryption

Security can be described with regard to how well a cipher can resist certain types of attacks.

CPA game and IND-CPA:

- Defender: have an oracle (a black box to outsiders that gives answers to queries)
- Adversary: keep sending messages $m1$, $m2$ … mN
- Defender: the oracle will send back $E(m1)$,$E(m2)$, … , $E(mN)$.
- Adversary: send $m1$' , $m2$' (they could be one of the $m1$, $m2$ … mN) and the oracle encrypts one of them, mb', (picks b randomly, 0 or 1) and sends the encryption to the adversary

- Goal: adversary distinguishes whether $m1'$ or $m2'$ was encrypted with a certain advantage.

No CPA scheme can be deterministic, since the adversary could already have the encryption of $m1'$ or $m2'$.

$$\Pr[b' = b] \leq 1/2 + \varepsilon$$

ε *is the advantage here*

How Secure is a System?

- How secure is a system against cryptanalysis when the enemy has **unlimited time and manpower available** for the analysis of intercepted cryptograms?

- **Computationally Secure**
- **Perfect Secrecy**

Computionally Secure

With regard to computational power

- The cost of breaking the cipher exceeds the value of the encrypted information.
- The time required to break the cipher exceeds the useful lifetime of the information.

Perfect Secrecy

- A cipher (K, M, C) has <u>perfect secrecy</u> if
  - Given plaintexts $m_0, m_1 \in M \rightarrow |m_0| = |m_1|$
  - Given ciphertext c
  - Given the secret key k $\xleftarrow{R} K$

- c does not reveal anything about the plaintext
- c does not reveal anything about m_0, m_1
- No attack is possible on c

- **Perfect secrecy requires:** $|\mathcal{K}| \geq |M|$

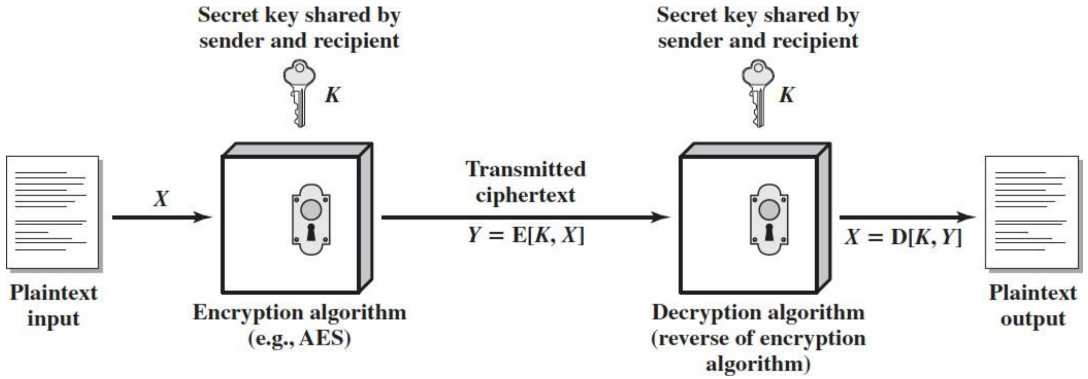
- Shannon’s definition:
    - A secure cipher should have confusion and diffusion properties:
    - **Confusion:** the relationship between the key and ciphertext is obscured
    - **Diffusion:** the statistical structure of the plaintext which leads to its redundancy is “dissipated”, i.e., the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext
        - Adversary must do more work to find statistical properties

Shannon, “Communication Theory of Secrecy Systems”, http://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf

Symmetric Encryption

Five components:

- **Plaintext (m):** The original message or data
- **Secret key (k):** Input to the algorithm.
- **Ciphertext (c):** The scrambled message produced as output. For a given message, two different keys will produce two different ciphertexts.
- **Encryption algorithm E(m,k):** Performs various substitutions and transformations on the plaintext.
- **Decryption algorithm D(c,k):** The encryption algorithm run in **reverse**.



Building blocks

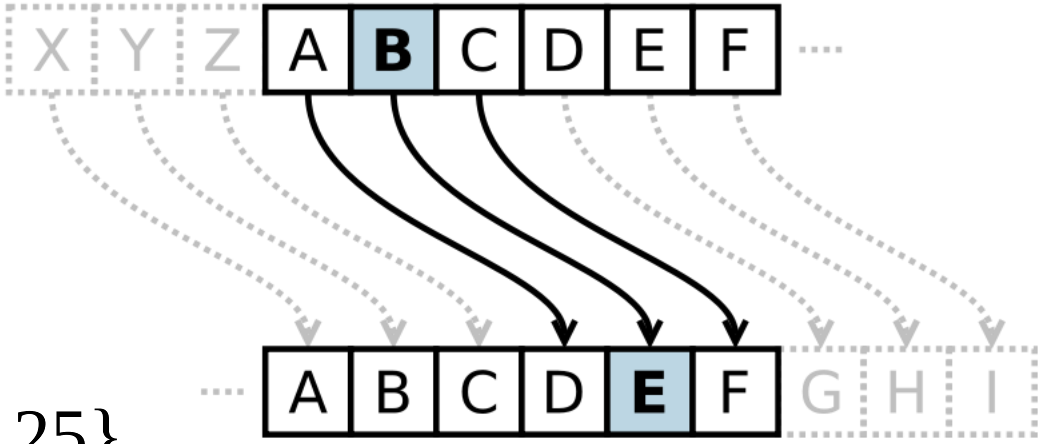
The two basic building blocks of all encryption techniques are substitution and transposition.

Substitution is replacing the letters of plaintext by other letters or by numbers or symbols.

Transposition or permutation is changing the locations of the letters in the plaintext so that they are rearranged into a different sequence.

Ceasar Cipher Cryptosystem

- The key k is the offset that shifts the alphabet
  - Encrypt using
    - $c_i = (m_i + k) mod 26$
  - Decrypt using
    - $m_i = (c_i - k) mod 26$



- Plaintext space: M
- Key space: K = {an integer between 0 and 25}
- |M| = |K|

Susbstitution Cipher

- **Does Ceasar provide confusion and diffusion?**
- Confusion
  - **Yes,** no relationship between key and ciphertext

- Diffusion
  - **NO!** Changing one symbol in the plaintext has a very predictable result
  - Only changes one symbol in the output

- **How to break Ceasar cipher?**

  - Brute-force?
  - Statistical analysis?

Attacks to Substitution Ciphers

1. **Brute force attack**
2. **Statistical attack**

- In every language
  - Symbols occur with different probabilities
- Frequency analysis
  - Looks at how often each is seen in a sample
  - Match frequency in ciphertext to frequency in plaintext
  - Gives a short list of possible mappings

Based on a sample of 40,000 words

Letter	Frequency
E	12.02
T	9.10
A	8.12
O	7.68
I	7.31
N	6.95
S	6.28
R	6.02
H	5.92
D	4.32
L	3.98
U	2.88
C	2.71
M	2.61
F	2.30
Y	2.11
W	2.09
G	2.03
P	1.82
B	1.49
V	1.11
K	0.69
X	0.17
Q	0.11
J	0.10
Z	0.07

http://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html

Note on Substitution

- An S-Box (substitution box) is
  - A table used for a table-lookup type of substitution mechanism
  - An m x n substitution cipher
  - Invertible if $m = n$

Leftmost

Rightmost

	00	01	10	11
0	011	101	111	100
1	000	010	001	110

	00	01	10	11
0	00	10	01	11
1	10	00	11	01

Example of Substitution

- Invertible if same input and output size
  - If the input to the left box is **001**, the output is **101**
  - The input **101** in the right table creates the output **001**
  - The two tables are inverses of each other

Encryption S-Box

	00	01	10	11
0	011	101	111	100
1	000	010	001	110

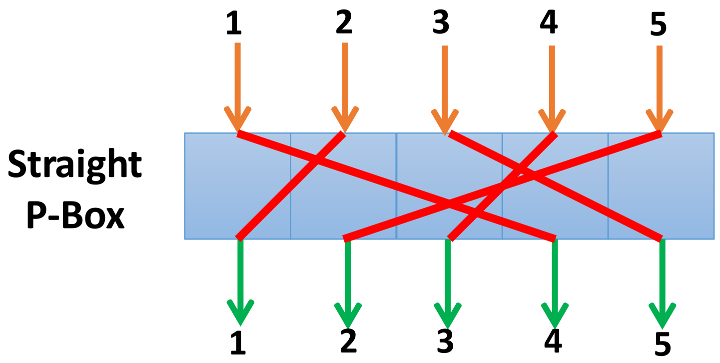
Decryption S-Box

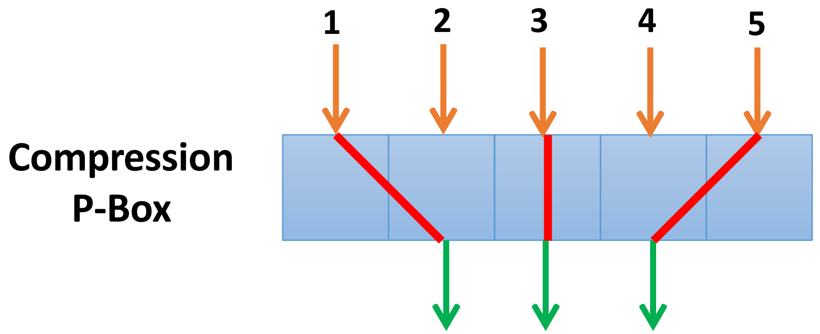
	00	01	10	11
0	100	110	101	000
1	011	001	111	010

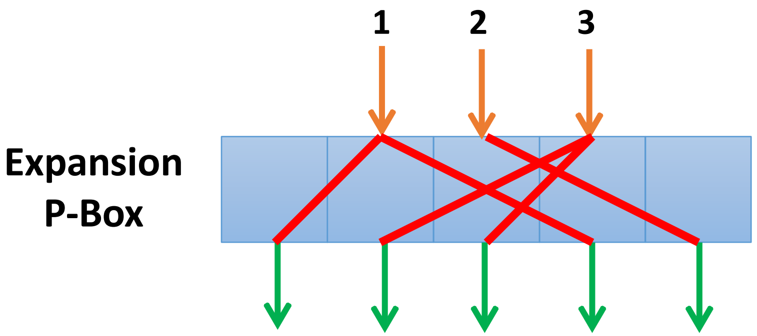
Try it:
Encrypt 000 and 110 into ciphertext, c1 and c2; use first bit to decide row and next two bits to decide column
Calculate the sum of the ciphertext, i.e., c = c1 ⊕ c2
Convert c1, c2, and the sum c back to plaintext

Note on Permutation

- A P-Box (permutation box) is
  - A method of bit-shuffling used to permute or transpose bits across S-boxes inputs
  - The traditional transposition cipher for characters

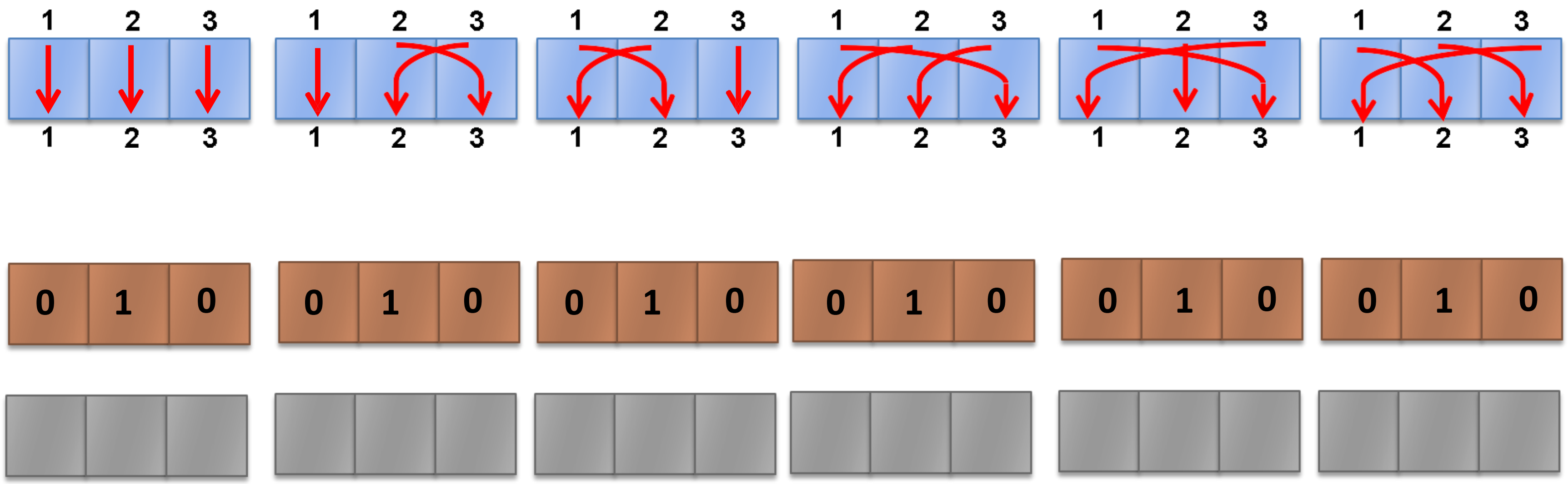






Example of Permutation

- Straight 3x3 P-Box (permutation box)
  - 6 possible mappings
  - Same number of inputs and outputs



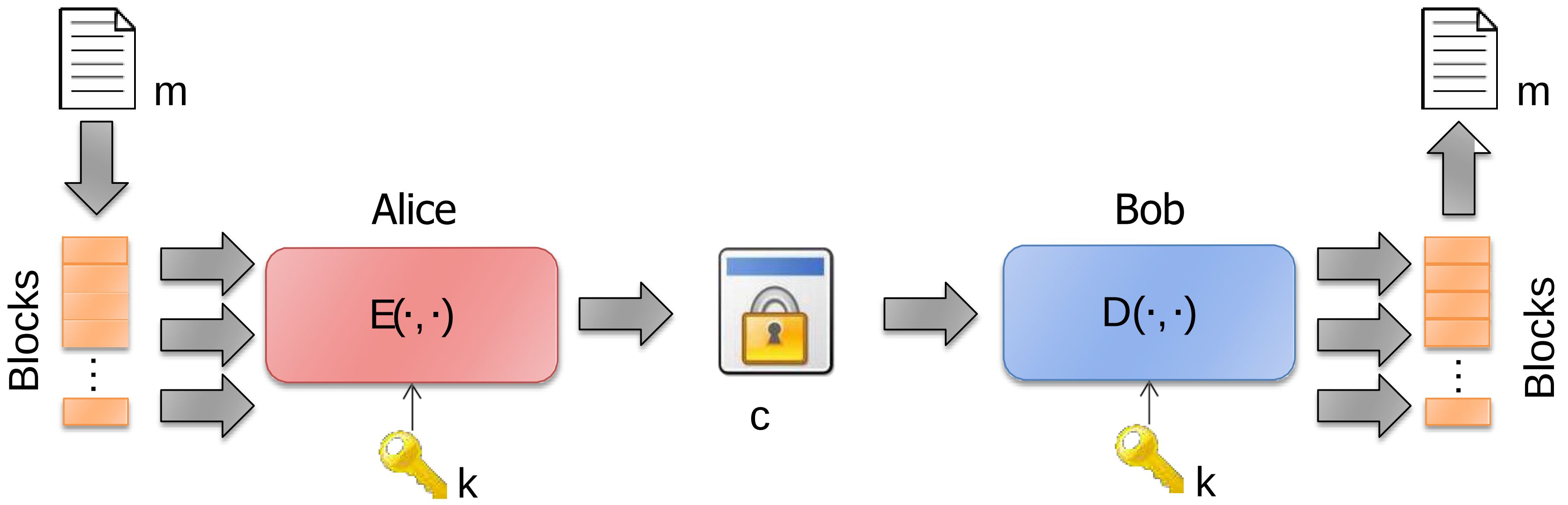
University of Salford MANCHESTER

SCHOOL OF SCIENCE, ENGINEERING & ENVIRONMENT

Block Cipher

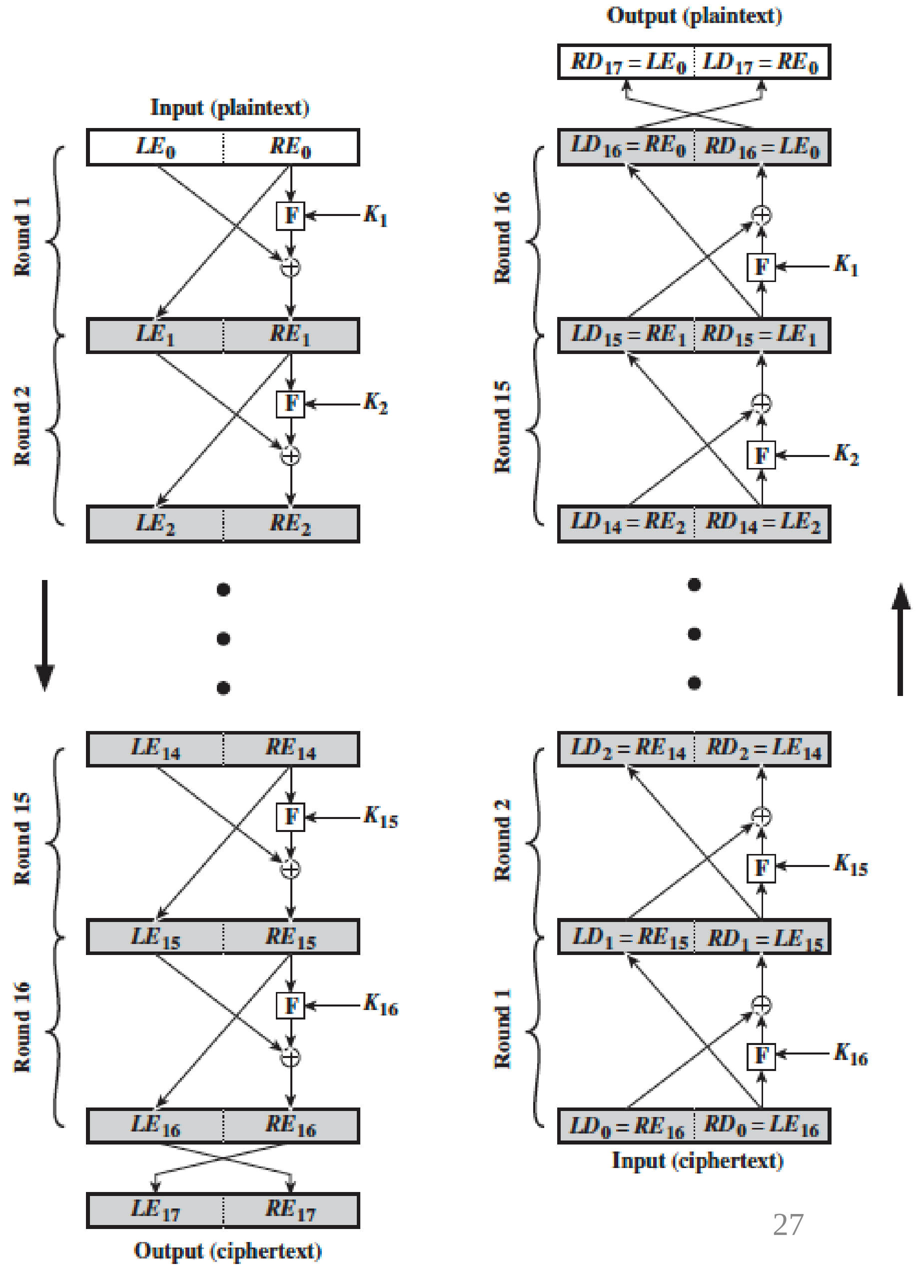
Block Cipher

- Takes one block (plaintext) and transforms it into a block of the same length using a provided secret key
- Decrypts by applying the reverse transformation to the ciphertext block using the same secret key
- Encrypt/Decrypt blocks of data of fixed length (e.g. 128bits)



Feistel Cipher Structure

- **Rounds**
  - **Each round uses the output of the previous round as input**
  - **Each round has the same structure**
- **In each round:**
  - **Substitution: F function with subkey and right half, then xor with left half**
  - **Permutation: switch left half and right half**
  - **Using subkey derived from key**
  - **Decryption is the reverse of encryption**

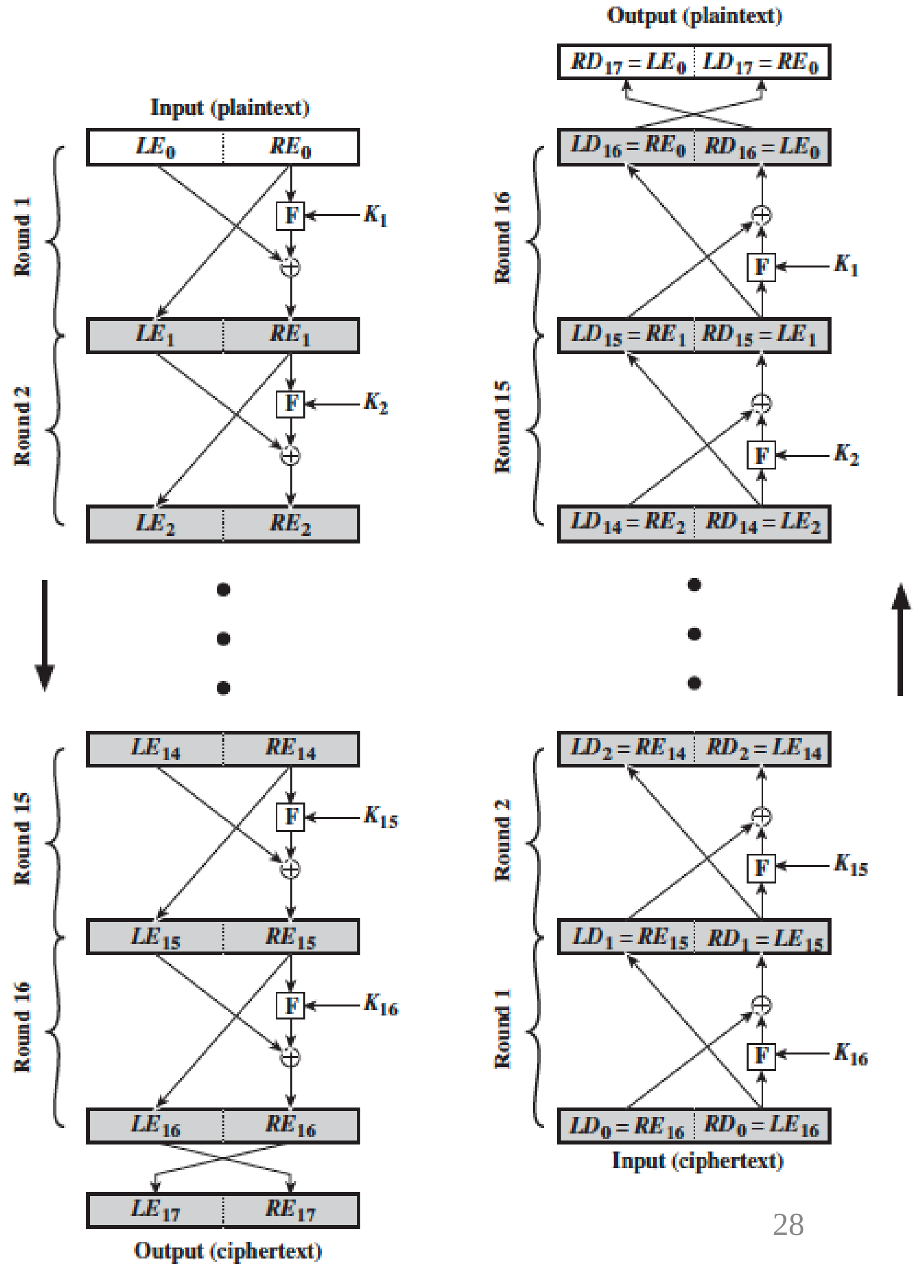


University of Salford MANCHESTER

SCHOOL OF SCIENCE, ENGINEERING & ENVIRONMENT

Feistel Cipher Structure

- **Block size:** larger block sizes, greater security; reduced encryption/decryption speed.
- **Key size:** larger key size, greater security; may decrease encryption/decryption speed.
- **Number of rounds:** increasing number improves security. A typical size is 16 rounds.
- **Subkey generation algorithm:** greater complexity, greater difficulty of cryptanalysis.
  - **Sub keys are different from K and from each other**
- **Round function:** greater complexity, greater resistance to cryptanalysis.

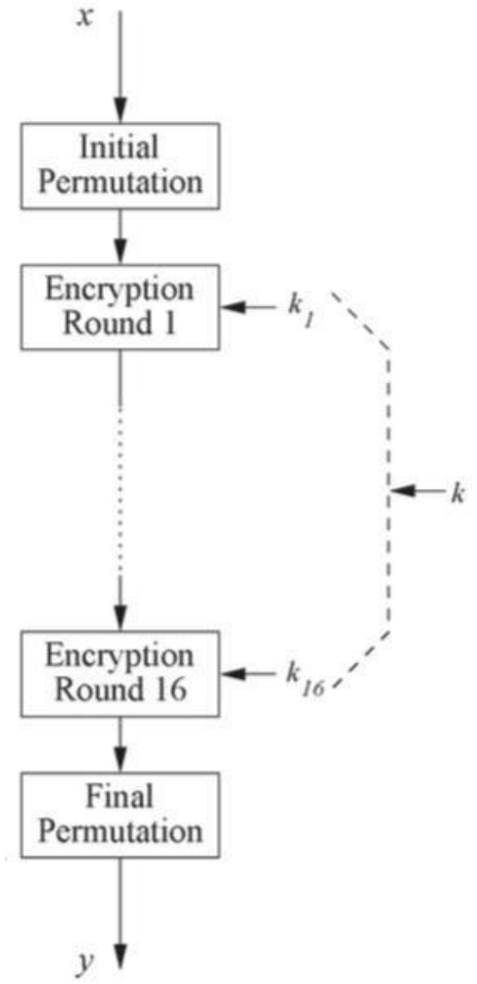


Most Common Block Ciphers

- Fixed key and block length
  - **DES**: $m = 64\,bits, \quad k = 56\,bits$
  - **3-DES**: $m = 64\,bits, \quad k = 168\,bits$
  - **AES**: $m = 128\,bits, \; k = 128/192/256\,bits$

Data Encryption Standard (DES)

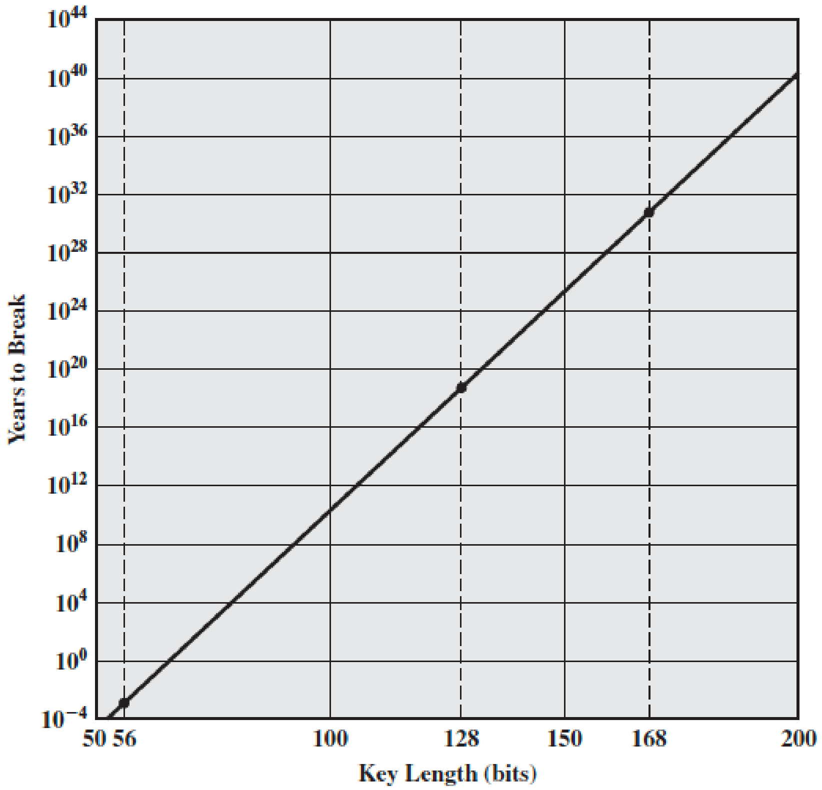
- DES was designed by IBM in 1977
- Plaintext of 64-bits blocks processed with 56 bits keys
  - longer plaintexts are processed in 64-bit blocks
- Based on the permutation mechanisms and XOR over keys
  - 16 rounds of identical operation
  - Different subkeys in each round derived from the main key by permutation



DES details: https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf

Data Encryption Standard (DES)

- Strength of DES:
  - **Algorithm**
    - No known flow so far
  - **Key size**
    - 2^{56} keys to search!
    - It was broken in 1998 in less than 3 days with 'DES Cracker' machine which was built for less than $250,000



Time to break a DES-style system assuming 106 decryptions/microseconds

If the only form of attack to an encryption algorithm is **brute force,** then the way to counter such attacks is **using longer keys**.

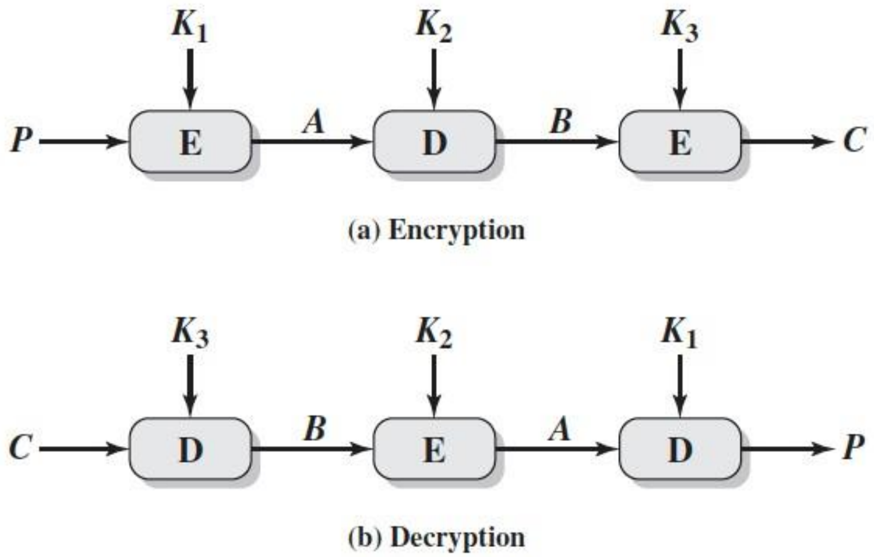
DES Cracker: https://web.archive.org/web/20130622022127/https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/

Triple DES

- 3DES first standardized for use in financial applications in ANSI standard X9.17 in 1985
- Plaintext of 64-bits blocks, three keys and three executions of the DES algorithm: encrypt-decrypt-encrypt (EDE) sequence
- 3DES has a 168-bits key length. FIPS 46-3 allows for the use of two keys, where K1=K3; this provides for a key length of 112 bits.

- Strength of 3DES:
  - Algorithm
    - As strong as DES
  - Key size
    - Not easy to bruteforce.

Challenge?
- **Performance (Speed & implementation)**
- **64-bit block size not efficient!**



FIPS: Federal Information Processing Standard

Advanced Encryption Standard (AES)

- Introduced in 2001 by NIST to replace DES
  - Features:
    - Be publicly defined
    - Be a symmetric block cipher
    - Have a key length that can be increased if needed
    - Be implementable both in hardware and software
    - Evaluation Criteria: security, computational efficiency, memory requirement, flexibility
  - Proposed by two Belgian cryptographers: Dr. Joan Daemen and Dr.Vincent Rijmen.

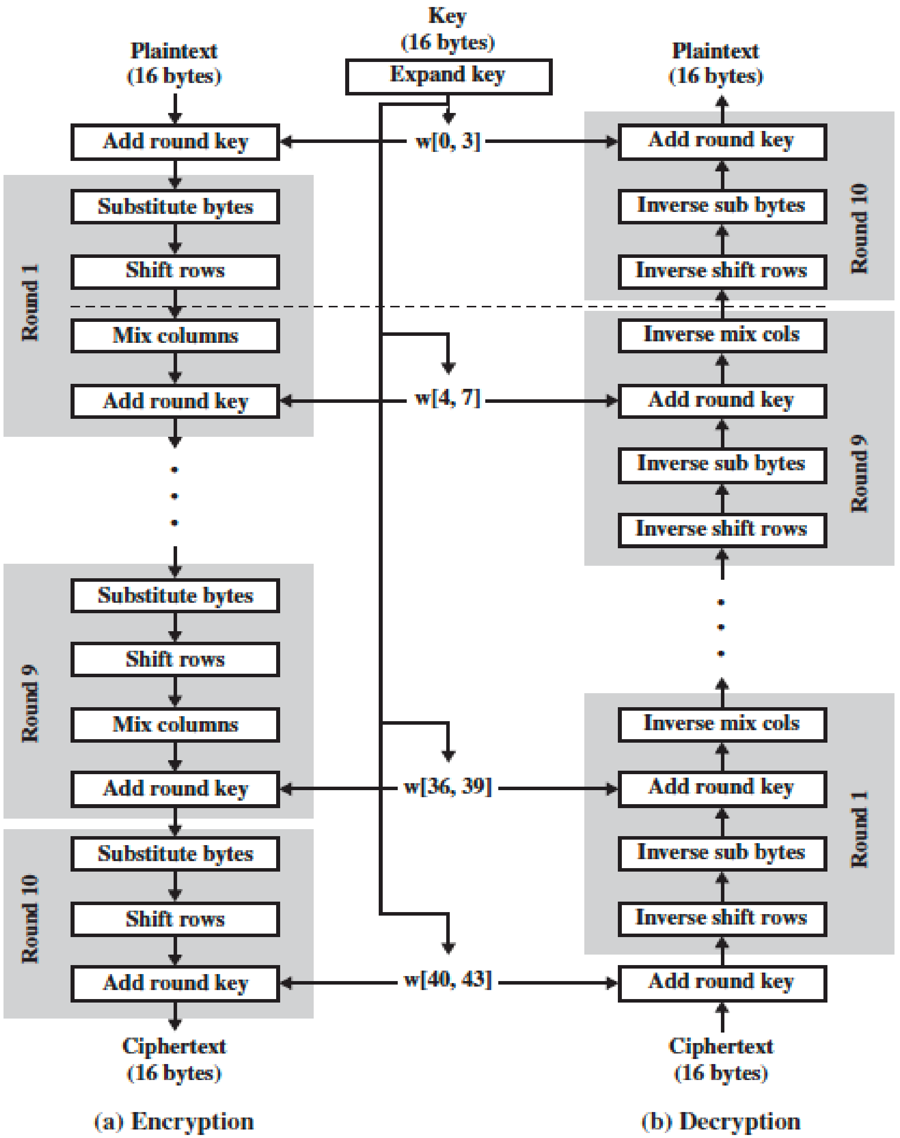
- Plaintext of 128-bits blocks, and a key of length 128, 192, or 256 bits

NIST: National Institute of Standards and Technology

AES Structure

- Plaintext input is a square matrix of bytes
  - This block is copied into a **state** array which is modified in each stage
  - The first four bytes of a 128-bit plaintext input occupy the first column of the matrix
  - State is copied to an output matrix at the end

- Key is depicted as a square matrix of bytes
  - Expanded into an array of 44 words of 32 bits
  - The first four bytes of the key occupy the first column of the ***w*** matrix
  - Four distinct words serve as a round key for each round.



University of Salford MANCHESTER

SCHOOL OF SCIENCE, ENGINEERING & ENVIRONMENT

AES Details

- In both AES and DES encryption and decryption is done in rounds
  - In AES the number of rounds depends on the key length
- AES is **NOT** a Feistel cipher, it is an algebraic cipher
  - Data blocks are processed in parallel during each round using substitutions and permutations
- Four different steps are used:

1. **Substitute bytes:** Using a table, referred to as an S-box, to perform a byte-by-byte substitution of the block.

Confusion

2. **Shift rows:** A simple permutation that is performed row by row.
3. **Mix columns:** A substitution that alters each byte in a column as a function of all of the bytes in the column.

Diffusion

4. **Add round key:** A simple bitwise XOR of the current block with a portion of the expanded key.

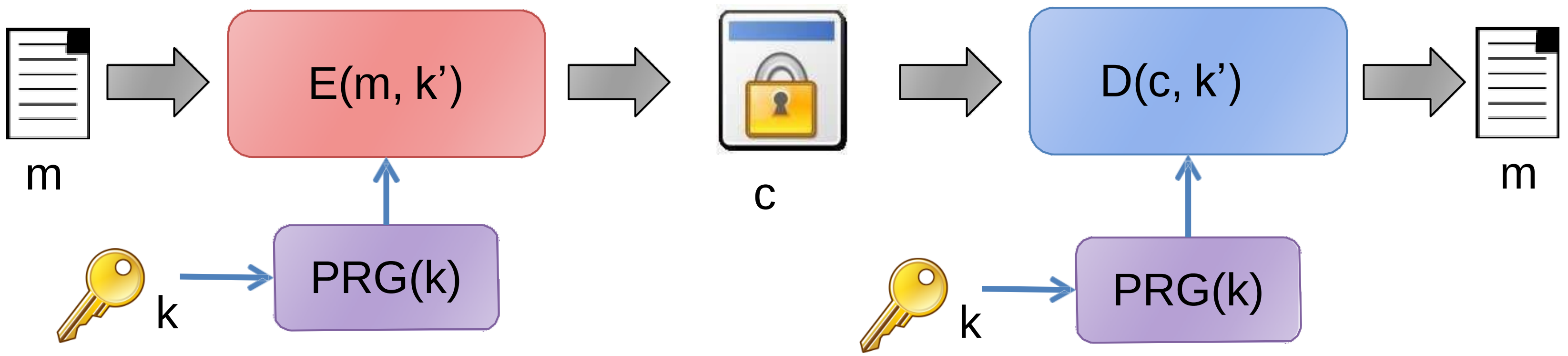
Stream Cipher

Stream Cipher

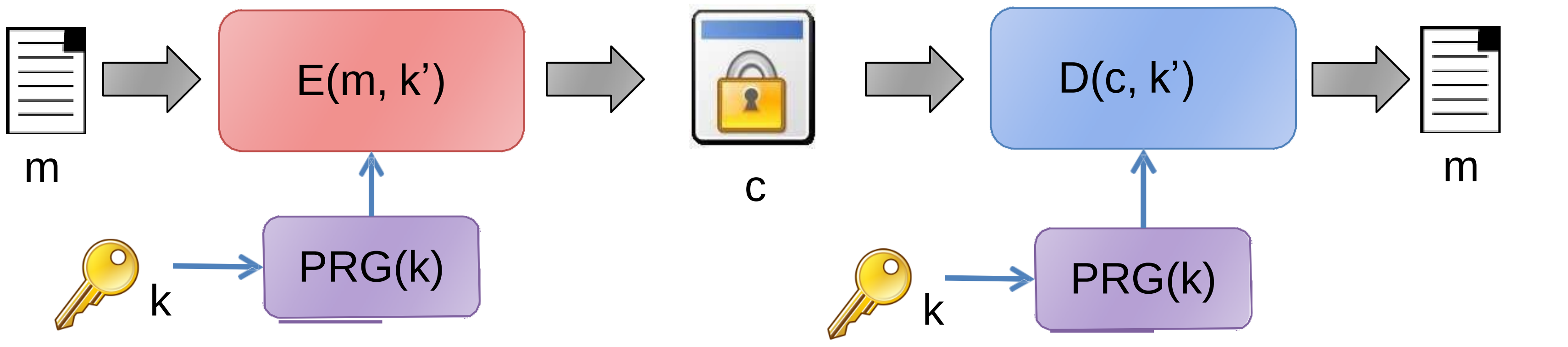
- Briefly
  - Replace <u>random</u> key with **<u>pseudo-random</u>**
  - Exploits PRG to replace the key with a **<u>key stream</u>**
  - One truly random key used as **<u>seed</u>**

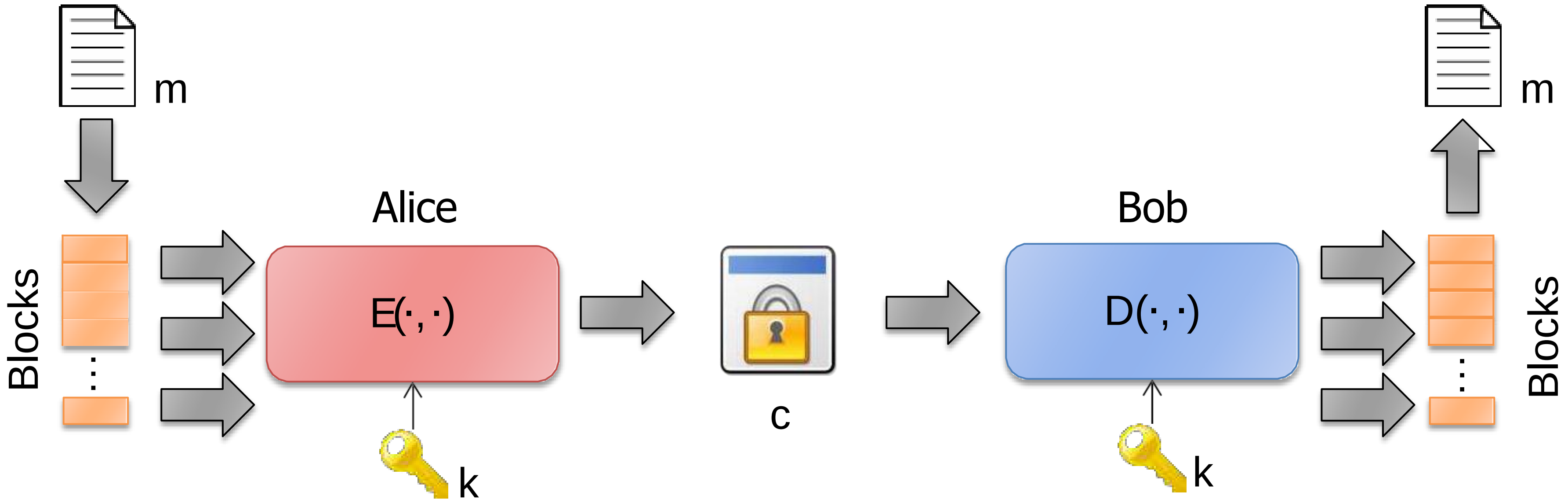
Key needs to be sufficiently long, at least 128 bits is desirable!

  - c= $E(k, m) = PRG(k) \oplus m$
  - $D(k, c) = PRG(k) \oplus c$



Block vs. Stream Cipher





Stream ciphers are faster and use less code

With a proper PRG, Stream ciphers can be as secure as block ciphers

In block cipher you can reuse keys, but in stream cipher you cannot!

Where to use **stream cipher** and where **block cipher?**

RC4 Algorithm

- A stream cipher designed in 1987 by Ron Rivest for RSA Security
- A variable key-size (1 to 256 bytes) stream cipher with byte-oriented operations
- **Simple and fast**; used in many protocols
  - Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards for communication between browser and web server
  - IEEE 802.11 WLAN – WEP and WPA
- Uses random permutation

Table 2.3 Speed Comparisons of Symmetric Ciphers on a Pentium II

Cipher	Key Length	Speed (Mbps)
DES	56	9
3DES	168	3
RC2	Variable	0.9
RC4	Variable	45

William Stallings, "Network Security Essentials: Applications and Standards"

University of Salford MANCHESTER

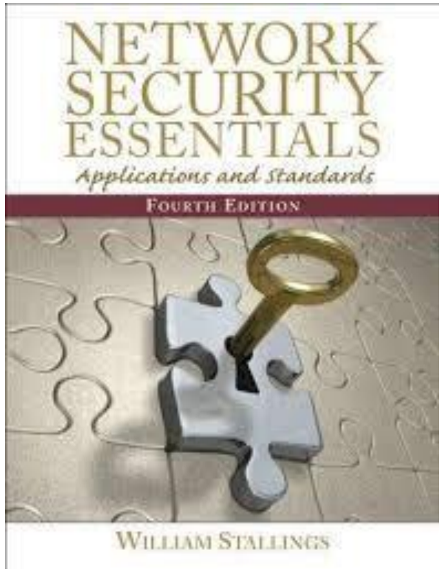
SCHOOL OF SCIENCE, ENGINEERING & ENVIRONMENT

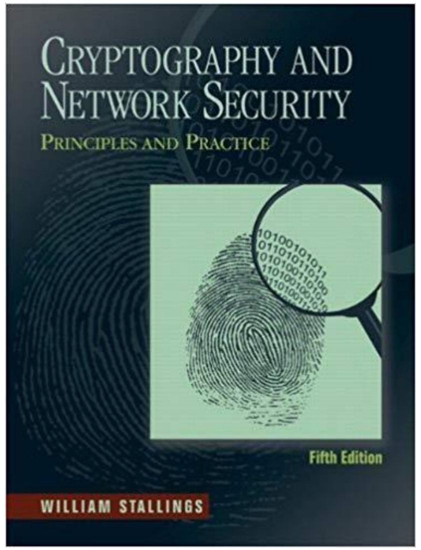
RC4 Security

- There are a number of security attacks to RC4
  - The keystream generated by RC4 is biased
  - The first few bytes are strongly non-random
  - …..

References

- William Stallings, “Network Security Essentials: Applications and Standards”
- William Stallings, “Cryptography and Network Security: Principles and Practice”





Introduction to Python

- General python tutorial: https://www.w3schools.com/python/
- Introduction to common crypto packages/tools/functions https://cryptohack.org/
- Workshop will be coding practice using python package pycryptodome manual: https://pycryptodome.readthedocs.io/en/latest/
- Please install python version >= 3.8 on your machine, if you wish to bring your machine to do the exercises in workshop