Cryptography II

Dr Aftab Ali

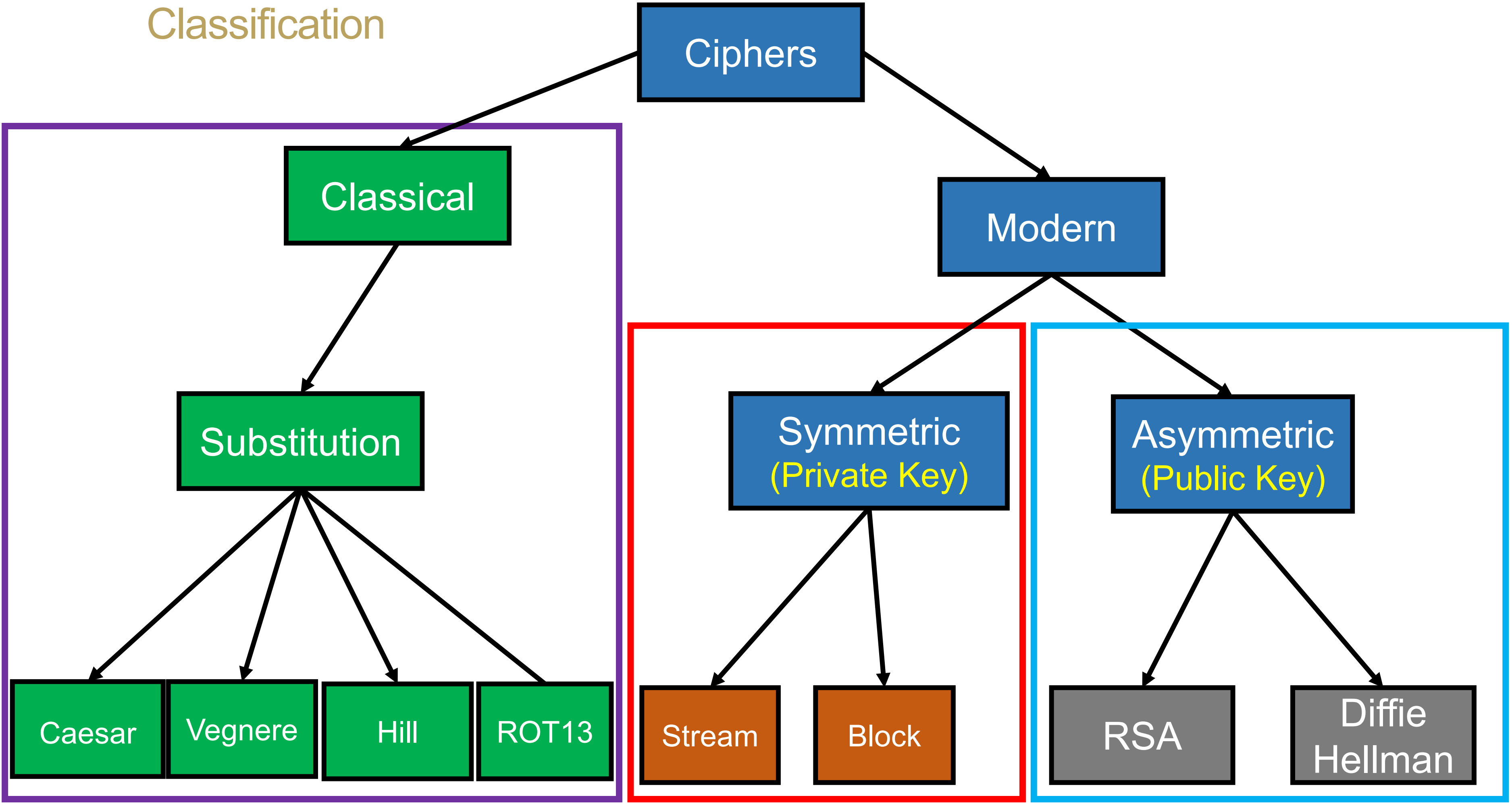
COM398

Introduction

- Cryptography: Definition
- Data Security
- Historical Background
- Symmetric Encryption examples
    - DES
    - AES
- Randomness
- Asymmetric Encryption
    - RSA
    - DH
- Trust in Cryptography

What we have covered

Classification



Symmetric Encryption
Data Encryption Standard (DES)

Until recently was the most widely used encryption scheme

- FIPS PUB 46
- Referred to as the Data Encryption Algorithm (DEA)
- Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block
- It is the best studied cipher.
- DES is a Feistel Cipher.

Symmetric Encryption
Data Encryption Standard (DES)

Strength concerns:

- Concerns about the algorithm itself
  - DES is the most studied encryption algorithm in existence

  – some detractors but no serious flaws

- Concerns about the use of a 56-bit key
  - The speed of commercial off-the-shelf processors makes this key length woefully inadequate

Symmetric Encryption
Data Encryption Standard (DES)

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

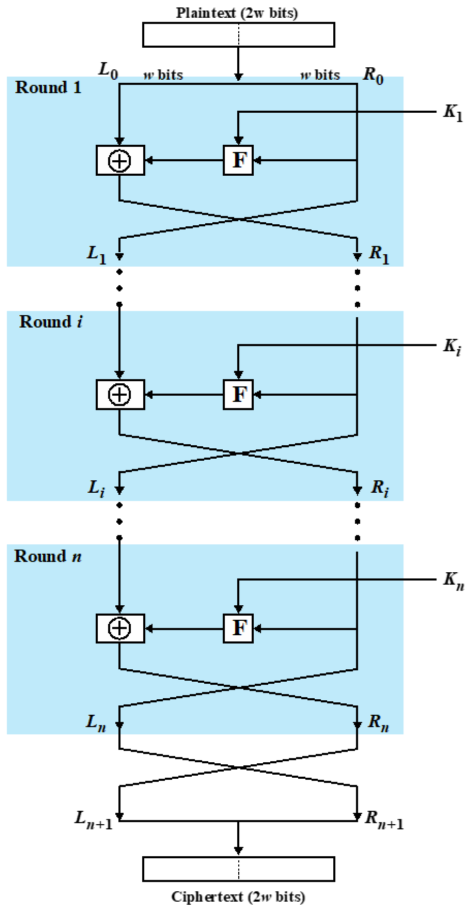
Average Time Required for Exhaustive Key Search

Symmetric Encryption

Feistel Network

- Many encryption algorithms incorporate a structure first proposed by Horst Feistel of IBM (1973); a **Feistel Network**.

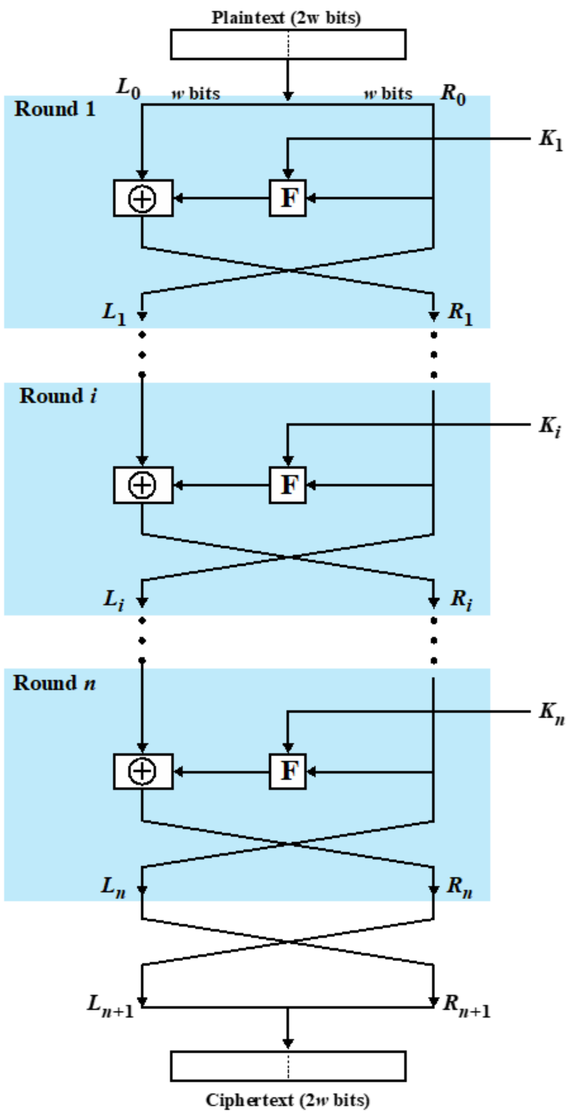
- This includes the DES standard.



Symmetric Encryption

Feistel Network

- The inputs to the encryption algorithm are a plaintext block of length 2w bits and a key K .
- The plaintext block is divided into two halves, L^0 and R^0.
- The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block.
- Each round i has as inputs L^{i-1} and R^{i-1}, derived from the previous round, as well as a subkey K^i, derived from the overall K .
- In general, the subkeys K^i are different from K and from each other and are generated from the key by a subkey generation algorithm.



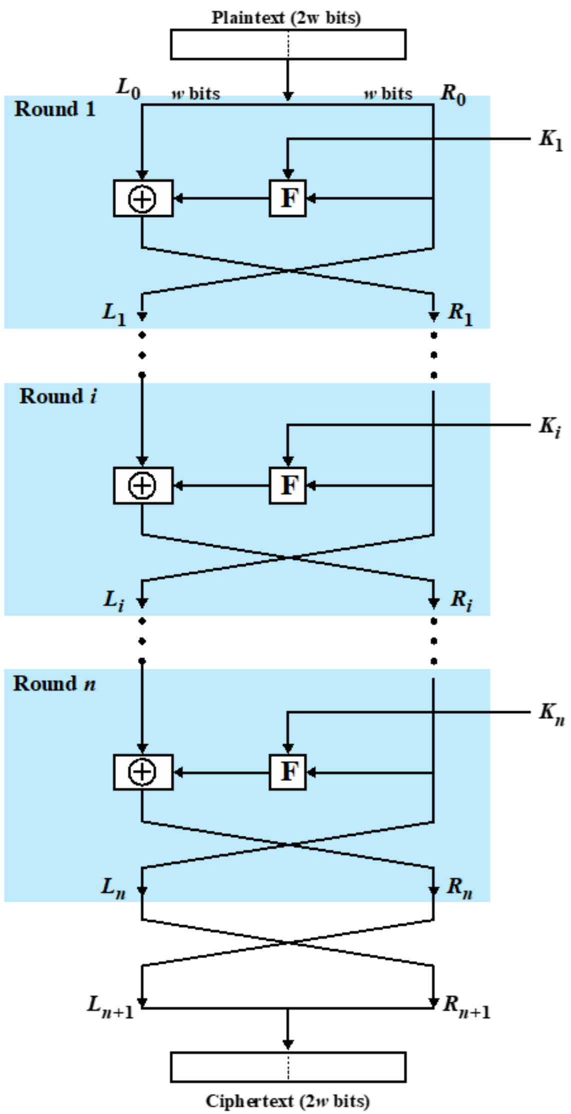
Symmetric Encryption

Feistel Network

- All rounds have the same structure. A substitution is performed on the left half of the data.

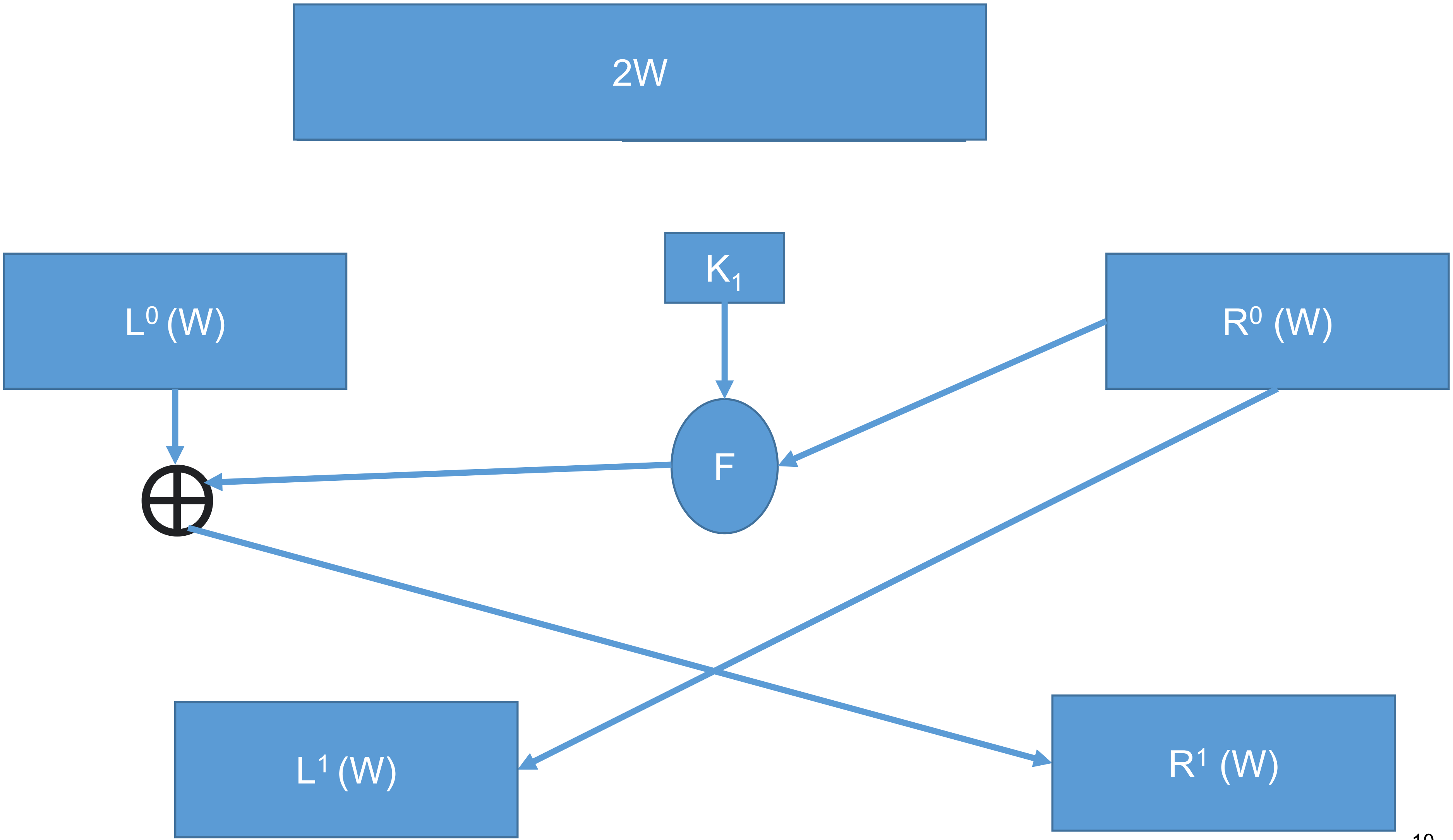
- This is done by applying a round function F to the right half of the data and then taking the exclusive-OR (XOR) of the output of that function and the left half of the data.
- The round function has the same general structure for each round but is parameterized by the round subkey K^i .

- Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.



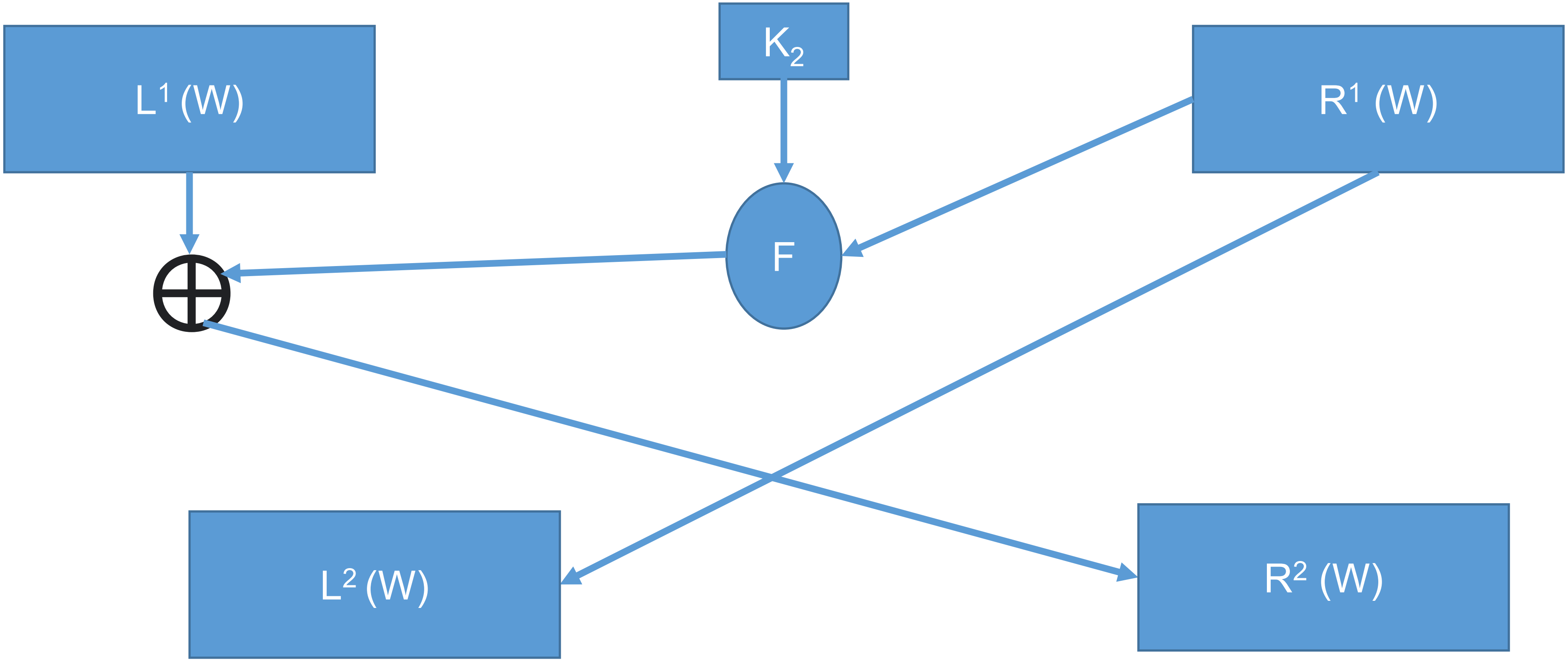
Symmetric Encryption

Feistel Network



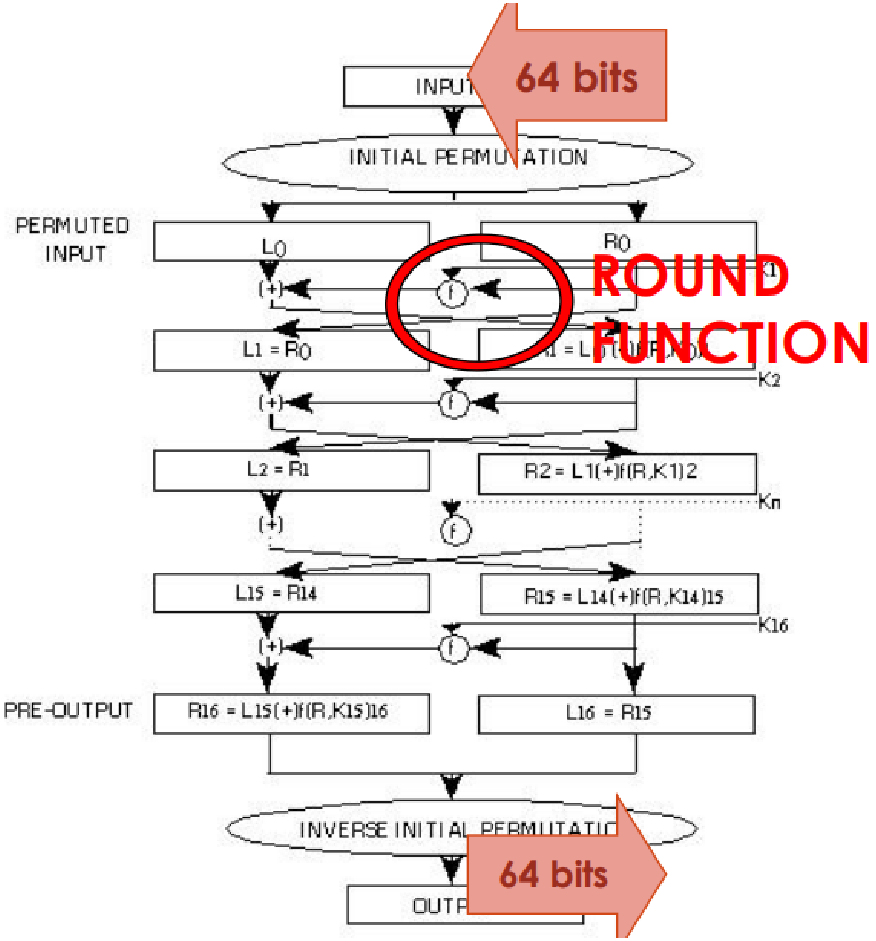
Symmetric Encryption

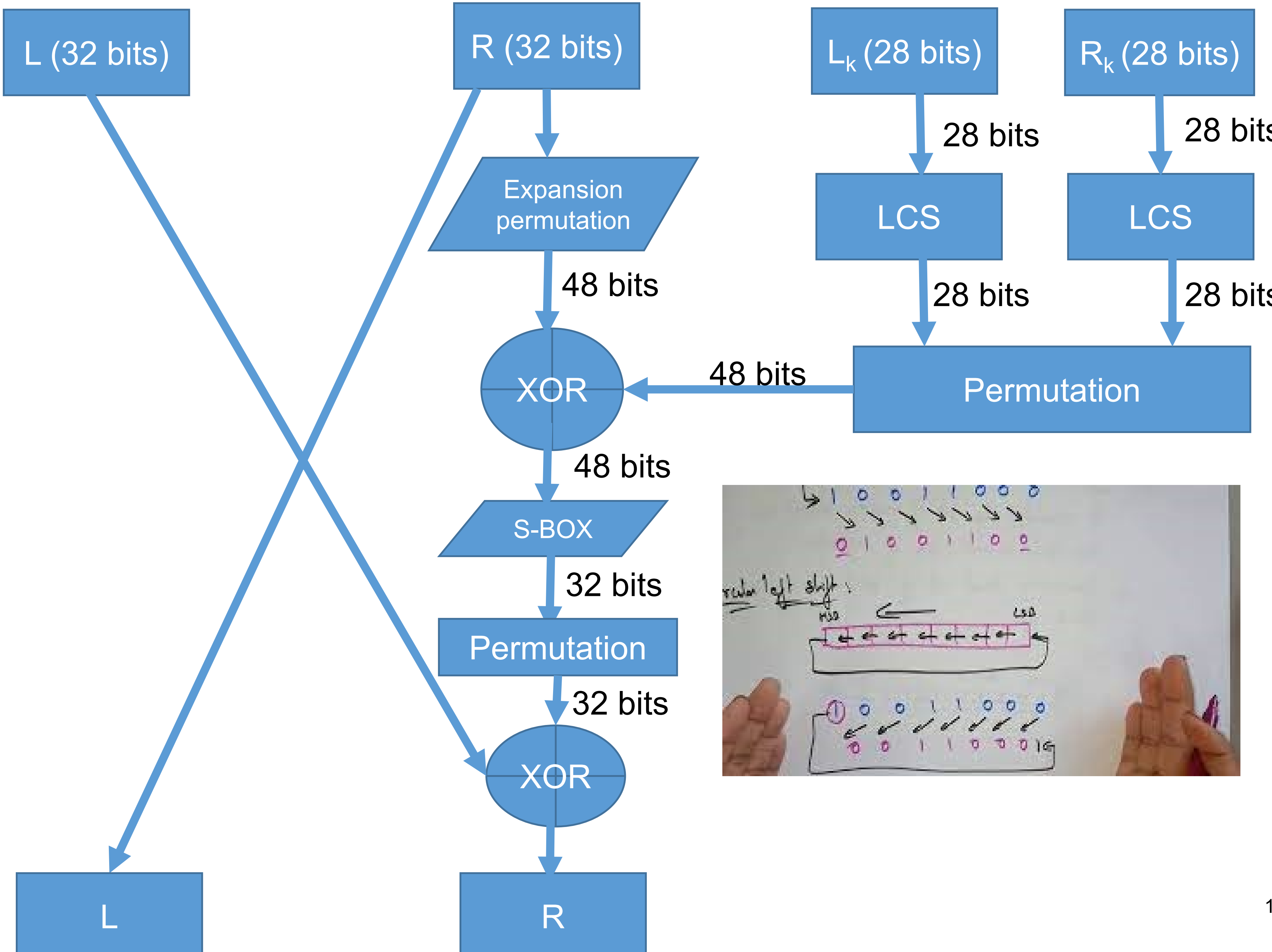
Feistel Network



DES

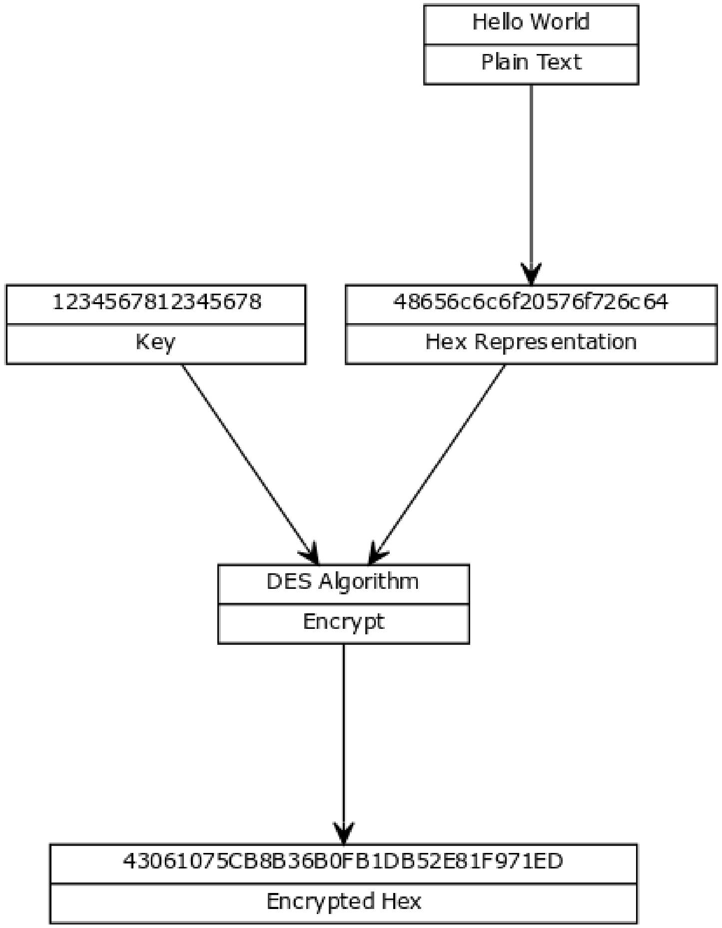
- 16 Rounds.
- In each round we apply:
  - Expansion Box.
  - Substitution Box.
  - XOR with the round key.




L (32 bits)
R (32 bits)
L_k (28 bits)
R_k (28 bits)
28 bits
28 bits
Expansion permutation
LCS
LCS
48 bits
28 bits
28 bits
XOR
48 bits
Permutation
48 bits
S-BOX
32 bits
Permutation
32 bits
XOR
L
R
left shift :
MSB
LSB

Symmetric Encryption

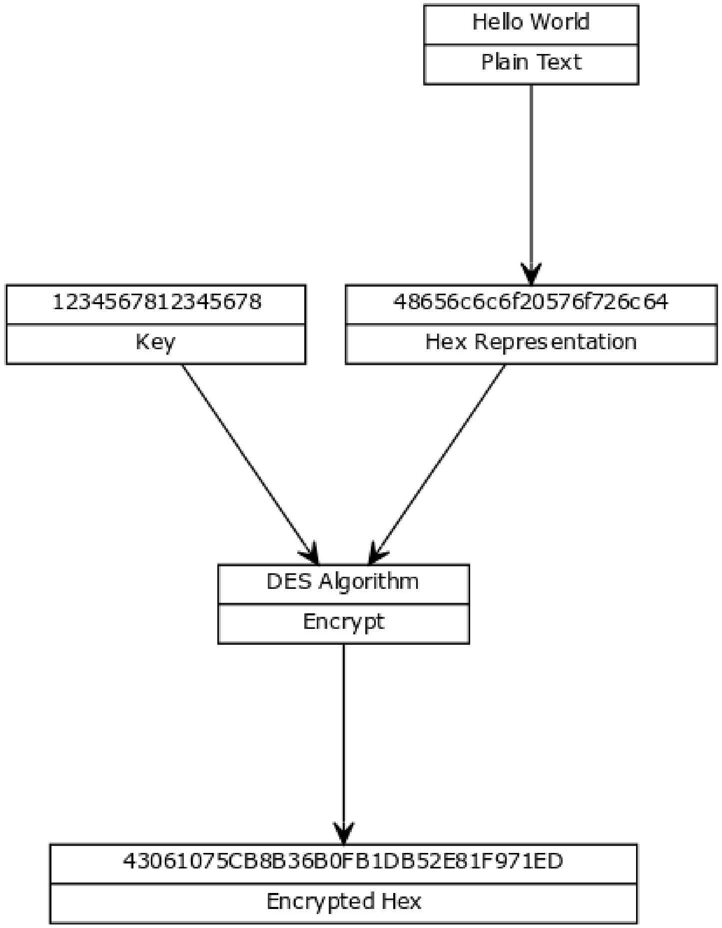
Data Encryption Standard (DES)



Variable Changes
Guess the outcome if we change the following

1. **Plaintext:** Hello world
2. **Key Change:** 00000000*

Symmetric Encryption
Data Encryption Standard (DES)

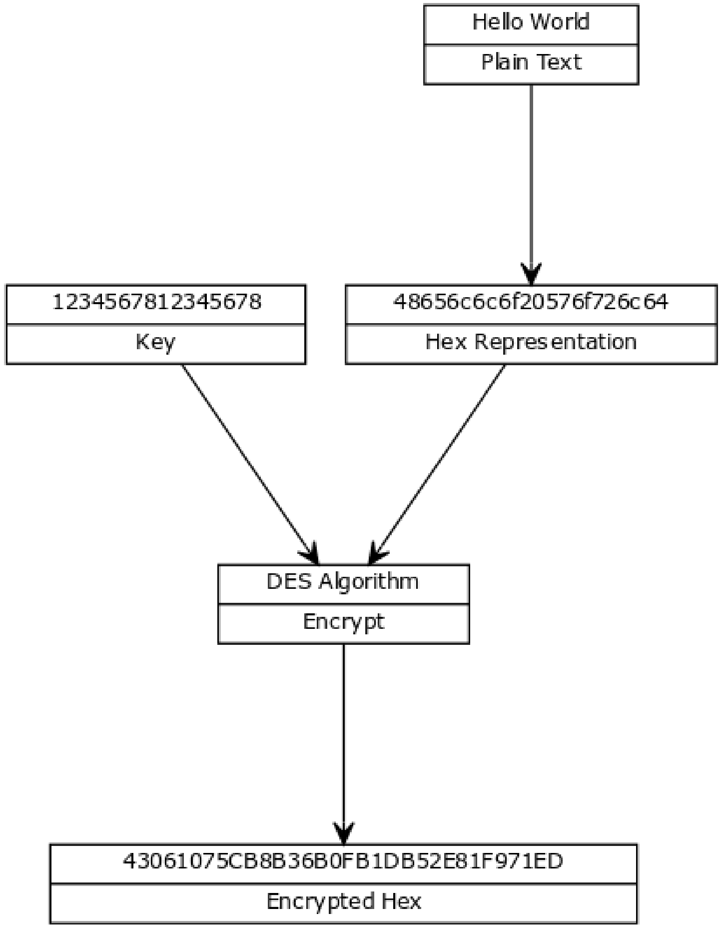


Variable Changes

Plaintext: Hello world
Hex: 48656c6c6f20776f726c64
Output:
80508BC5A5F89985FB1DB52E8
1F971ED

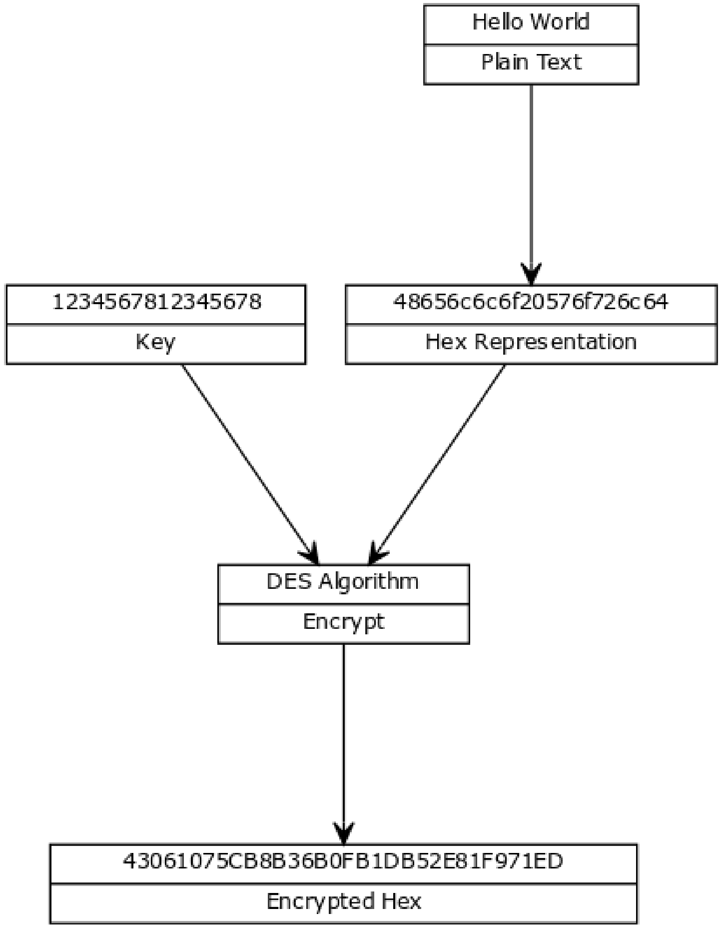
Key Change: 0000000000000000
Output:
EADE4DB3AFA20320011F38638
1D8E123

Symmetric Encryption
Data Encryption Standard (DES)



Why is the Encrypted Hex much longer?

Symmetric Encryption
Data Encryption Standard (DES)



Why is the Encrypted Hex much longer?

Why is this important?

Symmetric Encryption
Data Encryption Standard (DES)

Brute-force attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
- On average half of all possible keys must be tried to achieve success
- Number of Keys are dictated by the key size.

Symmetric Encryption

Triple DES (3DES)

- Repeats basic DES algorithm three times using either two or three unique keys

- First standardized for use in financial applications in ANSI standard X9.17 in 1985

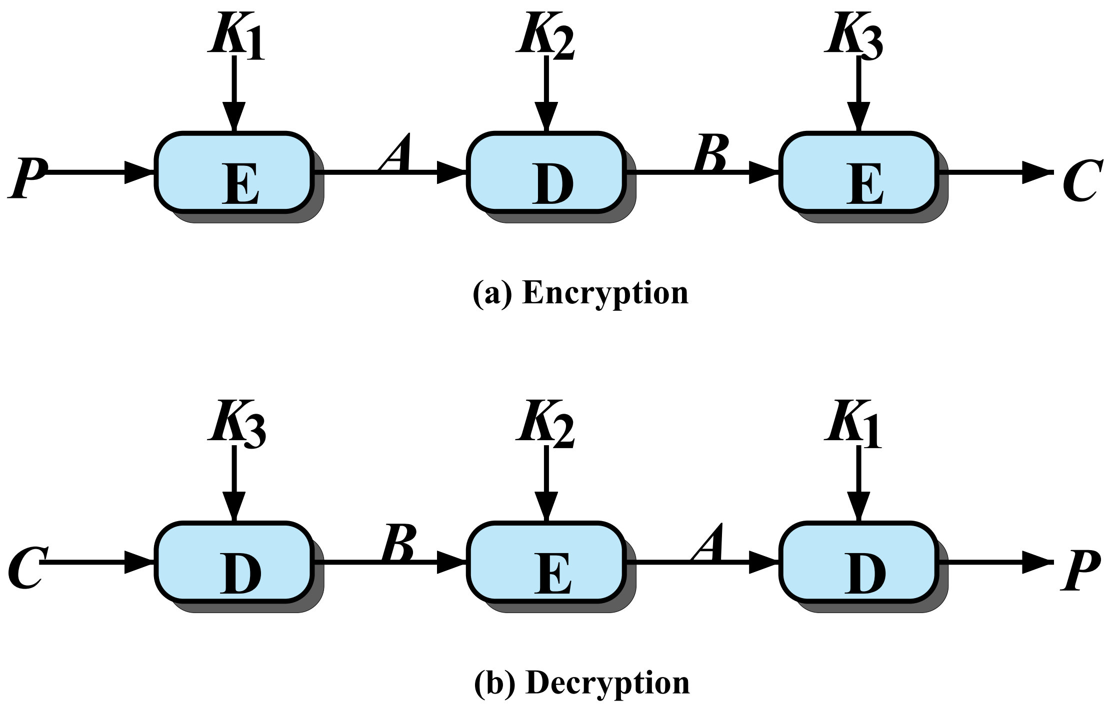
Attractions:

- 168-bit key length overcomes the vulnerability to brute-force attack of DES
- Underlying encryption algorithm is the same as in DES

Drawbacks:

- Algorithm is sluggish in software
- Uses a 64-bit block size

Symmetric Encryption
Triple DES (3DES)



Symmetric Encryption

Advanced Encryption Standard (AES)

Needed a replacement for 3DES

- 3DES was not reasonable for long term use

NIST called for proposals for a new AES in 1997

- Should have a security strength equal to or better than 3DES
- Significantly improved efficiency
- Symmetric block cipher
- 128 bit data and 128/192/256 bit keys

Selected Rijndael in November 2001

- Published as FIPS 197

AES

An **iterative** rather than **Feistel** cipher

- Processes data as block of 4 columns of 4 bytes
- Operates on entire data block in every round

Designed to be:

- Resistant against known attacks
- Speed and code compactness on many CPUs
- Design simplicity

Data block viewed as 4-by-4 table of bytes

Such a table is called the **current state**

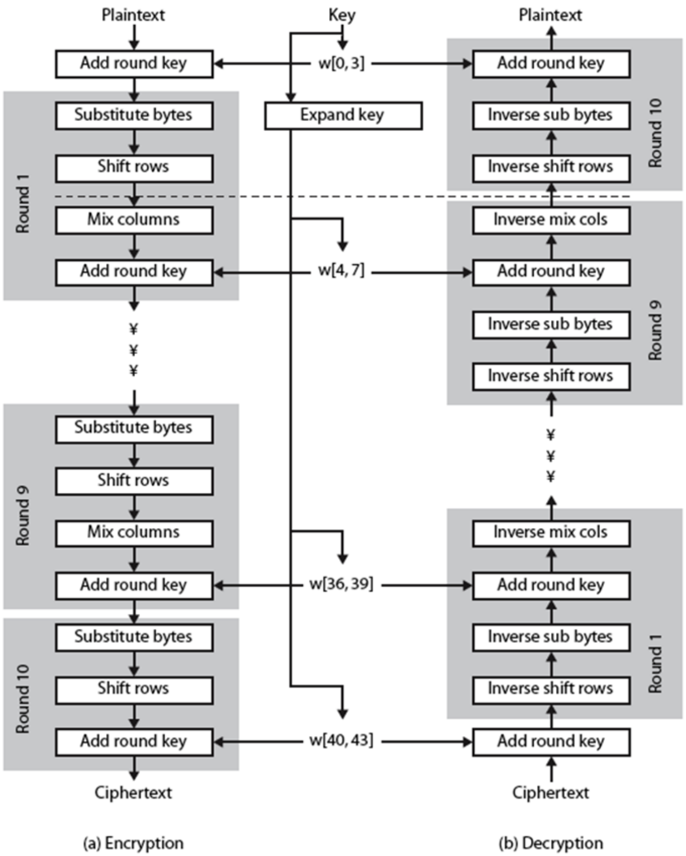
Key is expanded to array of words

Has 10 rounds in which state the following transformations (called `layers'):

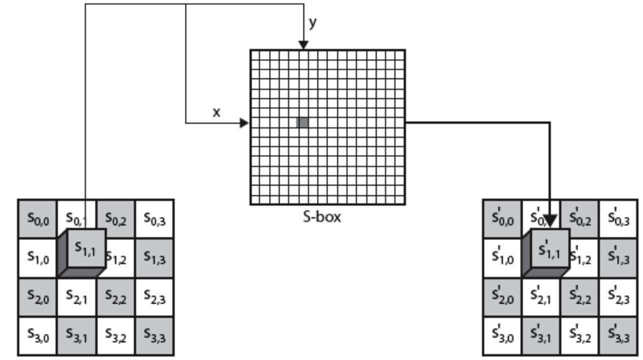
- BS- byte substitution (1 S-box used on every byte)
- SR- shift rows (permute bytes between groups/columns)
- MC- mix columns (uses matrix multiplication in GF(256))
- ARK- add round key (XOR state with round key)

First and last round are a little different

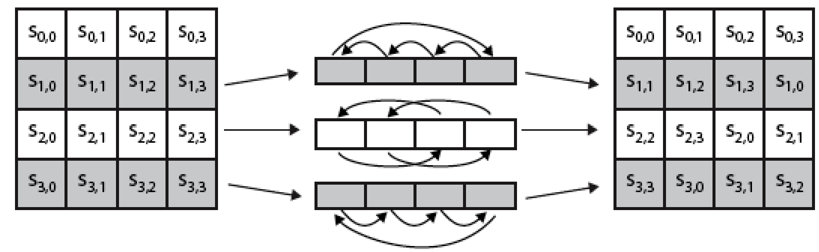
AES



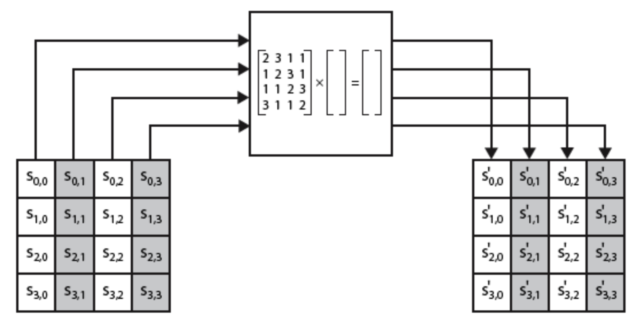
BS



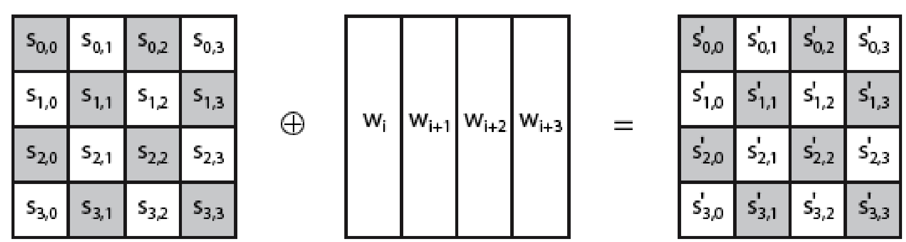
SR



MC



ARK



Random Numbers
Use

Random numbers are essential to many aspects of cryptography, including:

- Keys for public-key algorithms
- Stream key for symmetric stream cipher
- Symmetric key for use as a temporary session key or in creating a digital envelope
- Handshaking to prevent replay attacks
- Session key

Random Numbers

Criteria

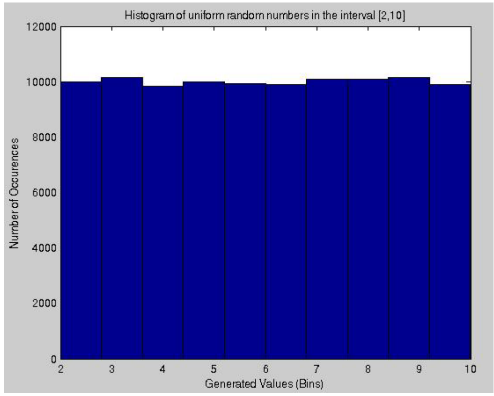
Criteria for a good Random Number Generator

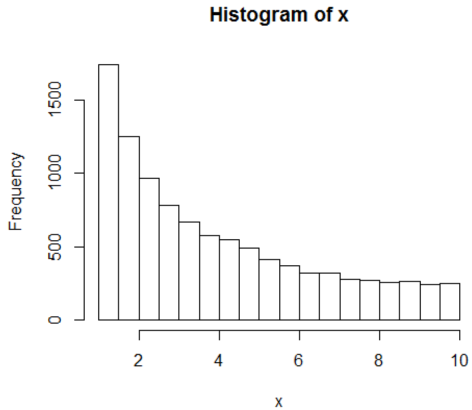
Uniform distribution of Random Numbers:

- Frequency of occurrence of each of the numbers should be approximately the same

Independence of Random Numbers:

- No one value in the sequence can be inferred from the others





Random Numbers
Criteria

Criteria for a good Random Number Generator

Unpredictability

- Each number is statistically independent of other numbers in the sequence
- Opponent should not be able to predict future elements of the sequence on the basis of earlier elements

Random Numbers

Random vs Pseudorandom

Cryptographic applications typically make use of algorithmic techniques for random number generation

- Algorithms are deterministic and therefore produce sequences of numbers that are not statistically random

Pseudorandom numbers are:

- Sequences produced that satisfy statistical randomness tests
- Likely to be predictable

True random number generator (TRNG):

- Uses a nondeterministic source to produce randomness
- Most operate by measuring unpredictable natural processes
  - e.g. radiation, gas discharge, leaky capacitors
- Increasingly provided on modern processors

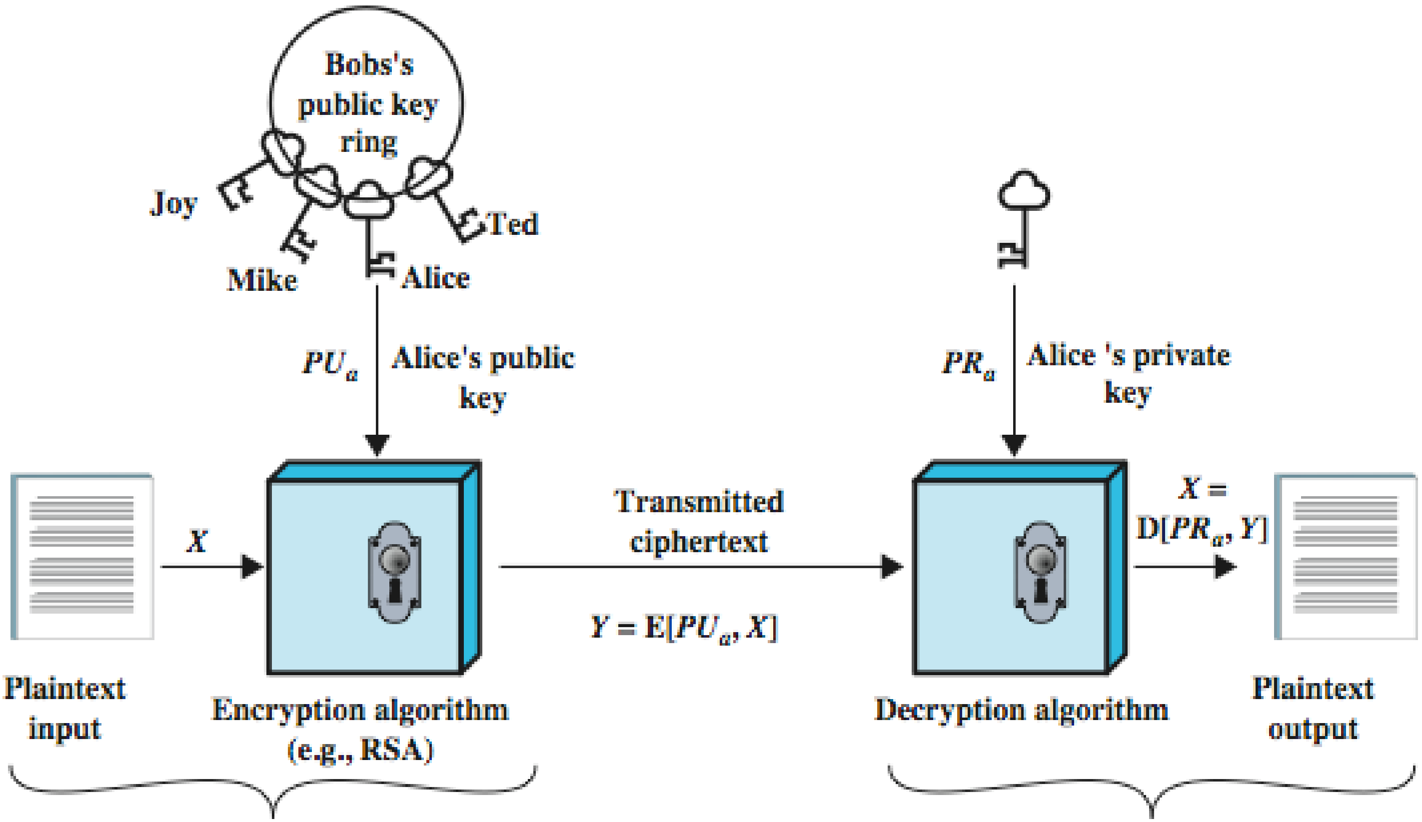
Public-Key Cryptography

- Probably most significant advancement in the 3000 year history of cryptography
- Uses two keys – a public & a private key
- Asymmetric since parties are not equal
- Uses clever application of number theoretic concepts to function
- complements rather than replaces private key crypto

Why Public-key

- Developed to address two key issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender
- Public invention due to Whitfield Diffie & Martin Hellman at Stanford Uni in 1976
  - known earlier in classified community

Public-Key Cryptography



(a) Encryption with public key

Public-Key Cryptography

In public key cryptography, if we know:

- the encryption algorithm **and**
- the key

to determine the ciphertext then how is it possible that we cannot work out what the plaintext (decryption key) is from this information?

One-way functions

- A **one-way function** is "easy" to compute and "difficult" to reverse.
- It is easy to take two prime numbers and multiply them together.
- If the numbers are small, we can do this in our heads, on a piece of paper.
- What if numbers get bigger and bigger?
- Multiplication of two prime numbers is **believed** to be a one-way function.
- The process of **exponentiation** means raising numbers to a power.
- Raising **2** to the power **3**, normally denoted 2^3 just means multiplying **2** by itself **3** times. In other words:
  - $2^3 = (2x2x2)$
  - $a^b = a \times a \times a \times \ldots \times a$
- **Modular exponentiation** means computing a^b mod some other number **n**.
  - a^b mod n.
- In other words, given a number **a** and a prime number **n**, the function
  - $f(b) = a^b \bmod n$

  Is a one-way function

RSA

- The **RSA** algorithm was the first practical implementation of public key encryption.
- It is named after the three researchers Ron Rivest, Adi Shamir and Len Adleman
- Let **n** be the product of two large primes **p** and **q** **(i.e. n=pxq)**
    - at least 512 bits.
- Select a special number **e**
    - greater than 1 and less than (**p**-1)(**q**-1).
    - The precise mathematical property that **e** must have is that there must be no numbers that divide neatly into **e** and into (**p**-1)(**q**-1), except for 1.
- Publish the pair of numbers (**n**,**e**)
- Compute the private key **d** from **p**, **q** and **e**
- The private key **d** is computed to be the unique inverse of **e** modulo (**p**-1)(**q**-1).
- In other words, **d** is the unique number less than (**p**-1)(**q**-1) that when multiplied by **e** gives you 1 modulo (**p**-1)(**q**-1).
    - $ed = 1 \bmod (p-1)(q-1)$

Public key: (e, n)

Private key: d

RSA

Encryption

Given a message M, $0 < M < n$ $M \in Z_n - \{0\}$

use public key (e, n)

compute $C = M^e \bmod n$ $C \in Z_n - \{0\}$

Decryption

Given a ciphertext C, use private key (d)

Compute $C^d \bmod n = (M^e \bmod n)^d \bmod n = M^{ed} \bmod n = M$

$$\text{Plaintext: } M \xrightarrow{C = M^e \bmod (n=pq)} \text{Ciphertext: } C$$

$$\text{Plaintext: } M \xleftarrow{C^d \bmod n} \text{Ciphertext: } C$$

From n, difficult to figure out p,q

From (n,e), difficult to figure d.

From (n,e) and C, difficult to figure out M s.t. $C = M^e$

Setting up RSA: example

Step 1: Let p = **47** and q = **59**. Thus n = **47** x **59** = **2773**

Step 2: Select e = **17**

e<n such that gcd(e,ϕ)=1

Step 3: Publish (n,e) = (**2773**, **17**)

Step 4: ϕ=(p-1) x (q-1) = **46** x **58** = **2668**

Use the Euclidean Algorithm to compute the modular inverse of **17** modulo **2668**. The result is d = **157**

<< Check: **17** x **157** = 2669 = 1(mod **2668**) >>

Public key is (**2773**,**17**)

Private key is **157**

Encryption and decryption

The encryption process to obtain the ciphertext C from plaintext M is very simple:

$$C = M^e \bmod n$$

The decryption process is also simple:

$$M = C^d \bmod n$$

Encryption and decryption: example

Public key is (2773,17)
Private key is 157

Plaintext block represented as a number: M = 31

Encryption using Public Key: $C = 31^{17} \pmod{2773}$

$= 587$

Decryption using Private Key: $M = 587^{157} \pmod{2773}$

$= 31$

Diffie–Hellman (DH) key exchange

- The **Diffie–Hellman (DH) key exchange** technique was first defined in their seminal paper in 1976.
- DH key exchange has the following important properties:
  - The resulting shared secret cannot be computed by either of the parties without the cooperation of the other.
  - A third party observing all the messages transmitted during DH key exchange cannot deduce the resulting shared secret at the end of the protocol.

Principle behind DH

- Assume that Alice and Bob are the parties who wish to establish a shared secret, and let their public and private keys in the public key cipher system be denoted by (PA , SA) and (PB , SB) respectively.
- The basic principle behind Diffie–Hellman key exchange is as follows:

1. Alice and Bob exchange their public keys PA and PB.
2. Alice computes F(SA , PB)

   $K = (PB)^{S_A} \bmod q$

3. Bob computes F(SB, PA)

   $K = (PA)^{S_B} \bmod q$

4. The special property of the public key cipher system, and the choice of the function F, are such that F(SA , PB) = F(SB, PA). If this is the case then Alice and Bob now share a secret.
5. This shared secret can easily be converted by some public means into a bitstring suitable for use as, for example, a DES key.

DH Example

- Let's say Alice and Bob agree on a prime modulo and a generator

  3 mod 17

Alice

- Now Alice select a private random number say 15

  3^{15} mod 17= 6

Bob

- Now Bob select a private random number say 13

  3^{13} mod 17= 12

- Now Alice take Bob public number 12

  12^{15} mod 17= 10

- Now Bob take Alice public number 6

  6^{13} mod 17=10

Man-in-the-middle attack

Trust in Encryption
Cryptanalysis

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

Trust in Encryption

Cryptanalysis

Type of Attack	Known to Cryptanalyst
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

Trust in Encryption

Computationally Secure Encryption Schemes

Criteria for a good Random Number Generator

Unpredictability

- Each number is statistically independent of other numbers in the sequence
- Opponent should not be able to predict future elements of the sequence on the basis of earlier elements

Encryption is computationally secure if:

- Cost of breaking cipher exceeds value of information.
- Time required to break cipher exceeds the useful lifetime of the information.

Usually very difficult to estimate the amount of effort required to break.

Can estimate time/cost of a brute-force attack.