

Plan of Talk

- Review of topics covered so far.
- Block Ciphers
 - DES
 - AES
- Stream Ciphers
- Symmetric key cryptography in Practice

Recap of Secure use of symmetric encryption:

- Two main requirements:
 - A strong encryption algorithm
 - Secret key known only to sender / receiver
- We cannot rely on the obscurity of the cryptosystem. We assume that the attacker knows everything about cryptosystems except the key used in cryptography.
- A secure channel is available for key distribution.

Symmetric key Cryptography

- Unconditional security or Perfect security.
 - Also referred sometimes as information theoretic security.
 - Example: One-time pads.
 - Computational Security.
 - Used in Practice.
 - Examples: DES, AES, Stream Ciphers etc.
-

Perfect Secrecy : One time pad:

- An encryption scheme has the property of *unconditional* security, if the cipher text generated by the algorithm does not reveal enough information to break the scheme even with access to unlimited amount computation.
- In other words, the adversary will not learn any knowledge to reverse the encryption from watching cipher text even with unlimited computing power.

One time pad

- Let X : input, y : output
- Perfect Security implies: $P_{X|Y}(x|y) = P_X(x)$
- One time Pad: This is a method to transmit a message M in $\{0,1\}^n$, an n bit string of binary numbers from a user say A (Alice) to B (Bob). Any intruder (Eve) should not get any information about M by watching the cipher text. A chooses a random key K in $\{0,1\}^n$ and adds it component wise to the message M . The transformed message is then transmitted on insecure channel where Eve can read the message. To decrypt the message Bob should have the copy of the key K which they would have exchanged on a secure channel.

One time pad example

- Let \oplus denote exclusive or symbol.
 - $0 \oplus 0 = 1 \oplus 1 = 0$;
 - $0 \oplus 1 = 1 \oplus 0 = 1$.
- Suppose A wishes to send a message $M=0110111$, and suppose they have previously established a shared secret key: $K = 1011011$.

The cipher text is formed by exclusive-oring the message with the key:

$$C = M \oplus K = 1101100.$$

Decryption is trivial: the message could be obtained by the same process, i.e. by addition of K to C .

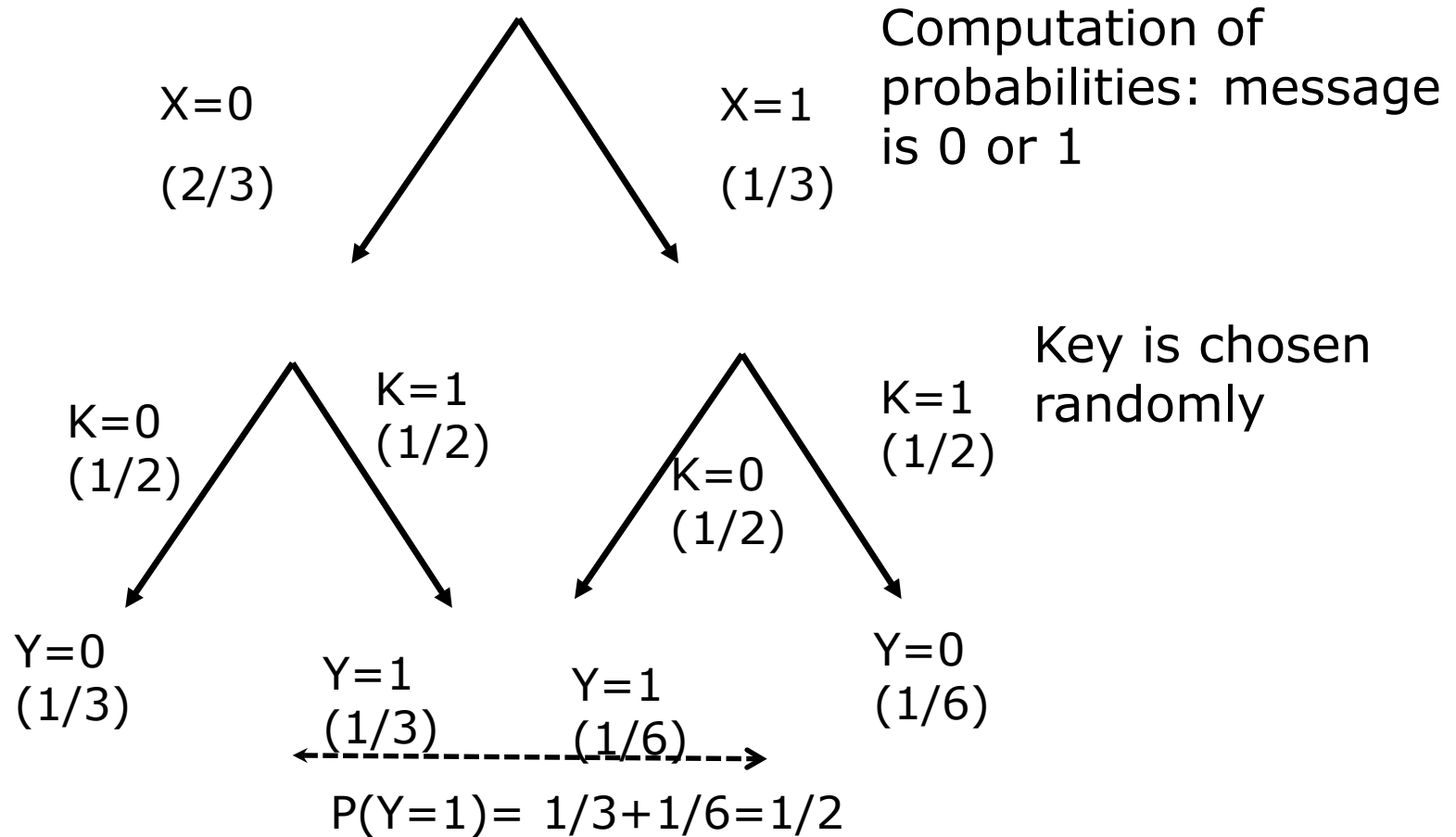
$$M = C \oplus K = 0110111.$$

- The one time pad offers perfect secrecy. Let us make it more precise what this means.
- Let us assume that the message space is binary (0 or 1) and key space is also binary. Assume that A chooses message 0 quarter of the time, i.e Probability that the message is 0 is equal to $1/4$, $P(M=0) = 1/4$.
Perfect secrecy means knowing this fact, any adversary (E) should not get more information by observing the cipher message ($C = M + K$).
i.e. The condition probability, $P(M = 0 \mid C = 1)$ should not be different from apriori probability $P(M=0)$.
- This means that seeing the cipher text C does not increase the adversaries knowledge about the message.

One time pad example

- Let message space be 0 or 1, i.e $X = 0$ or 1
 - Assume that the Adversary a priori knows that probability that $(X = 0)$ is $2/3$.
 - i.e, $P(X=0) = 2/3$, then $P(X=1) = 1/3$.
 - Suppose $Y = 1$ was observed at the output of the cipher
 - We want to prove $P(X=0|Y=1) = P(X=0)$
 - **This equivalent to : Seeing the cipher text does not increase the adversaries knowledge about the underlying message.**
-

$$P(X=0|Y=1) = P(X=0 \wedge Y=1) / P(Y=1) = (2/3 * 1/2) / (1/2) = 2/3 = P(X=0)$$



Main General Result

- When X and Y are long sequences of 1's and 0's of length n .
 - Theorem: $P(X=m|Y=c) = P(X=m)$.
 - Proof depends critically on the fact that K is generated according to uniform distribution,
 - i.e, $P(K=k_1) = 1/2^n$,
-

Implications of perfect secrecy

- Messages may be biased; could be observed by the adversaries.
 - Encryption transformation should distribute messages to cipher space fairly uniformly irrespective of known apriory statistics of the messages.
 - One-time pad analysis tells us that if we choose a random secret key pad at least the size as the message, we can achieve the perfect secrecy.
 - However, one-time pad is not practical.
-

Two time pad is bad!

- One time pad is not practical. It demands a key as long as the message.
- If one time pad used twice, it leaks statistics of the plain text. Germans made this mistake in the war times!.
- $C_1 = M_1 + K$; $C_2 = M_2 + K$; then
- $C_1 + C_2 = M_1 + M_2 + K + K = M_1 + M_2$.
- This means that you need a new key for every message.
- This idea is used in attacking Vegenere cipher (same key-pad is added many times)

Computationally Secure Ciphers

- We want a cipher function which is easy to encrypt,
 - But hard to invert if the key is unknown.
 - It is difficult to precisely define hardness of the inversion function .
 - An upper limit is set for computational complexity of inverting the function.
 - This definition is derived from practical pragmatism. Let us see what are the practical needs.
 - An encryption scheme is computationally secure if
 - the cost of breaking the cipher exceeds the value of encrypted information, and the time required to break the cipher exceeds the useful lifetime of the information.
-

Computationally Secure Ciphers

- How to define strongness of a computationally secure cipher?
 - No precise definition exists.
- Roughly speaking, if the key length n , the complexity of inverting the encryption without the key should be exponential in n .
- For example, 128 bit key cipher should provide 2^{128} work security.
- How to decide minimum key length?
- At least, key should be large enough to withstand a brute force search attack.
- A cipher with a n length key should not have any apparent weakness which helps to break the cipher with less than 2^n computation work.
- A cipher should be secure to withstand the cryptanalytic methods described earlier (Chosen Plaintext Attack(CPA), Chosen Ciphertext Attack(CPA) etc).



Blockciphers and stream ciphers

- These are two major kinds of ciphers, which differ in the way the plaintexts are encrypted.
- Block Cipher: A block cipher takes a fixed length plain text message block (for example, 64 or 128 bits) and a key, and produces a cipher text block of the same length as the original message.
- DES (56), Triple-DES (168), IDEA (128), Blowfish() and AES (128)
- Stream Cipher: Takes a key of fixed size and generates a key stream in a pseudo random fashion with large period; this key stream is then combined with the plain text message stream on a bit by bit basis to form a cipher text stream.
- RC4, A5, BlueTooth cipher etc.

Block Cipher

- Encrypts blocks of n characters/bits of plain text simultaneously outputting blocks of cipher texts.
- Same key is used for many different message blocks.
- Fundamental building blocks for many cryptographical functions.
- Examples include hash functions, pseudorandom generators, message authentication codes etc.
- **Confusion and diffusion principles:**
- **Diffusion** dissipates statistical structure of plaintext over bulk of ciphertext.
- **Confusion** makes relationship between ciphertext and key as complex as possible.
- Generally diffusion is created by permutations and confusion is created by substitution.

Product Ciphers and Fiestel Ciphers

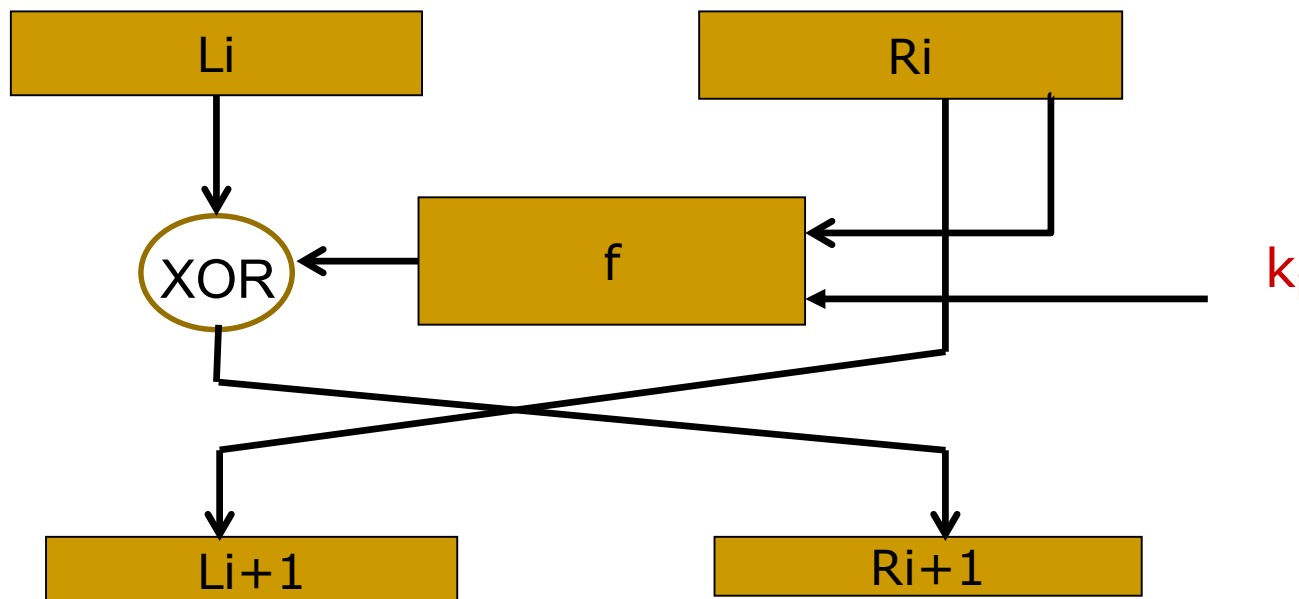
- A **product cipher** combines two or more transformations so that resulting cipher is more secure than the individual components by making use of confusion and diffusion principles.
- A **substitution-permutation cipher** is a product cipher made up of number of stages each involving substitution and permutation. The operations of substitution and permutation are responsible for effecting the confusion and diffusion respectively.
- An **iterated block cipher** is a block cipher involving sequential repetition of an iterated function called a round function.

Iterated Block Ciphers

- The parameters of iterated block ciphers are
 - r: number of rounds;
 - n: block length; k: bit-size of key, K from which r subkeys (round keys) k_i 's are derived.
- **Fiestel** ciphers are iterative ciphers; they repeat a given operation several times in rounds.

Iterated Block Ciphers Cont.

- For such a cipher, the input key is used to produce round keys k_1, k_2, \dots, k_r . The message is initially divided into two parts, namely left and right halves, L and R . For each of r rounds, the following operations are executed.



After r rounds, the final left and right halves are swapped and concatenated to form the cipher text.

The design of a good function f is partly ``ART" and partly ``SCIENCE".

Data Encryption Standard (DES)

- IBM's 1974 submission for a standard.
 - A Feistel cipher
 - Block size: $n = 64$,
 - keysize = $k = 56$ bits.

The key is specified with 64 bits containing 8 bits of parity.
Number of rounds = 16.

Strengthening DES:

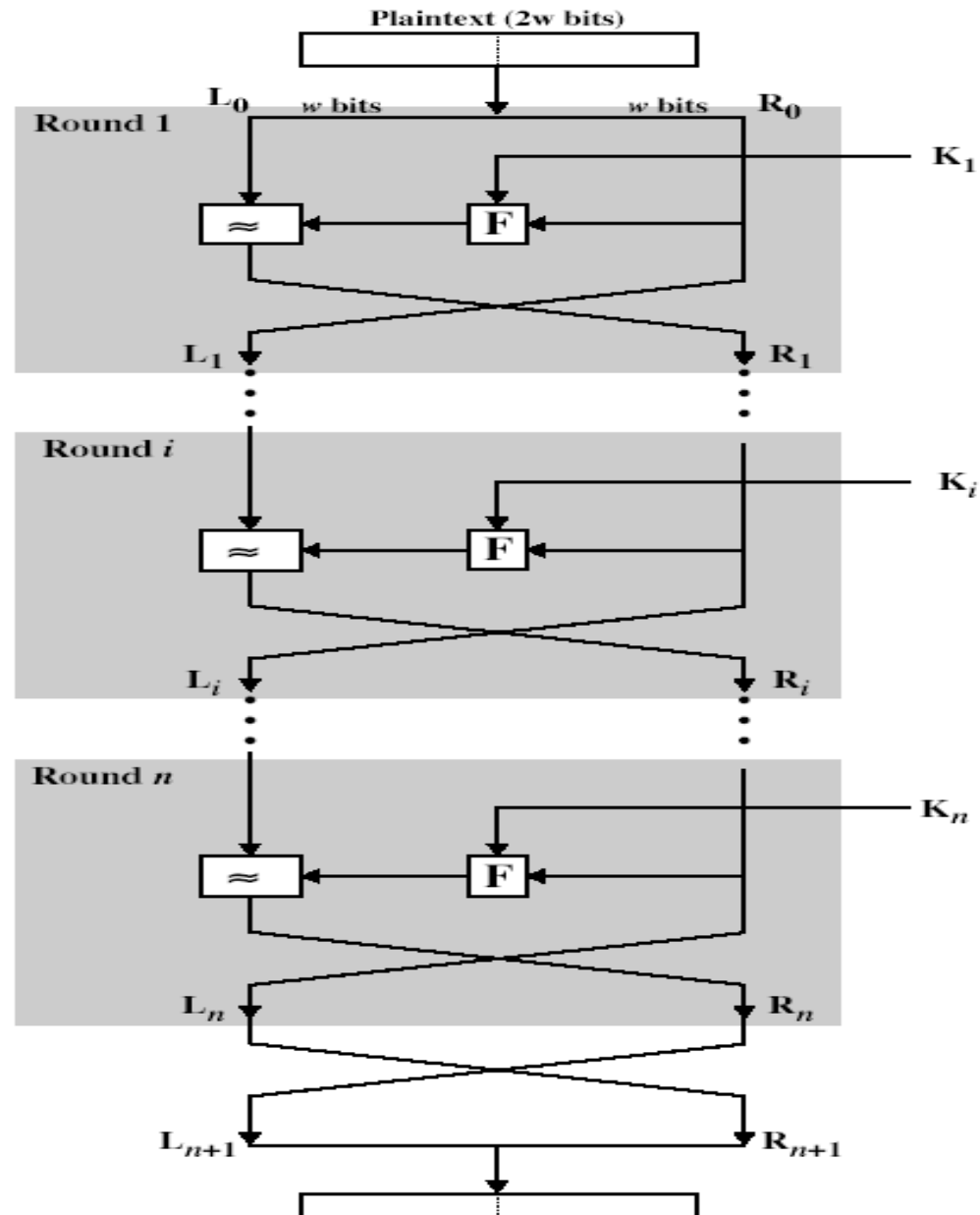
DESX: Apart from 56 bit key K , choose two new 64 bit keys K_I and K_O , then we encrypt

$$C = K_O \text{ XOR } \text{DES}(K, M \text{ XOR } K_I)$$

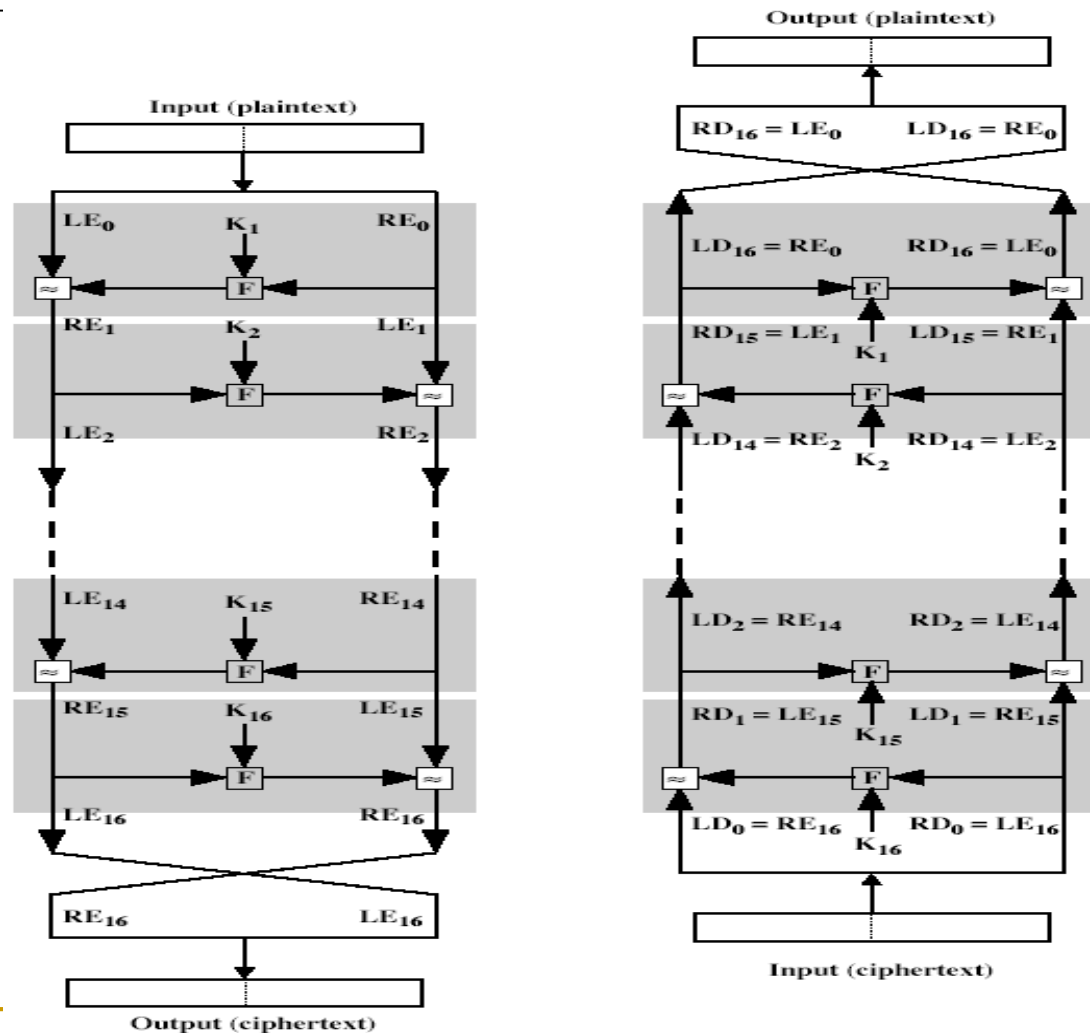
This method increases effective key length to $199-t$, where t is a quantity related to adversaries cryptanalytic assumptions where the adversary is able to collect 2^t matching input-output pairs.

- Read the textbook for more details on DES.

Feistel Cipher Structure



Feistel Cipher Decryption



Strengths, properties and attacks for DES

- Each bit of cipher text depends on all bits of the key and all bits of the plain text.

No statistical relationship between plain and cipher visible.

Altering a key bit or a plain text bit should alter each cipher bit with probability close to half.

Altering a cipher bit should result in unpredictable change in plain text block.

Cryptanalysis of DES

- Empirically it is found that DES is safe.
- Exhaustive search -- Brute force. 2^{64} computations.

Differential cryptanalysis

- Chosen plain text attack,
- Not realistic -complexity 2^{47} computations.

Linear cryptanalysis

- Complexity : 2^{43} computations.
- The new standard for encryption now is AES which has key space $\geq 2^{128}$.



Advanced Encryption Standard: Origins

- Reasons for a replacement for DES:
 - have theoretical attacks that can break it
 - have demonstrated exhaustive key search attacks
 - Can use Triple-DES – but slow, has small blocks
 - US NIST issued call for ciphers in 1997
 - 15 candidates accepted in Jun 98
 - 5 were shortlisted in Aug-99
 - Rijndael was selected as the AES in Oct-2000
 - issued as FIPS PUB 197 standard in Nov-2001
-



AES Requirements

- Private key symmetric block cipher
 - 128-bit data, 128/192/256-bit keys
 - stronger & faster than Triple-DES
 - Active life of 20-30 years (+ archival use)
 - Provide full specification & design details
 - Both C & Java implementations
 - NIST have released all submissions & unclassified analyses
-



AES Evaluation Criteria

■ Initial criteria:

- ❑ security – effort for practical cryptanalysis
- ❑ cost – in terms of computational efficiency
- ❑ algorithm & implementation characteristics

■ Final criteria

- ❑ general security
 - ❑ ease of software & hardware implementation
 - ❑ implementation attacks
 - ❑ flexibility (in en/decrypt, keying, other factors)
-



AES Shortlist

- after testing and evaluation, shortlist in Aug-99:
 - MARS (IBM) - complex, fast, high security margin
 - RC6 (USA) - v. simple, v. fast, low security margin
 - Rijndael (Belgium) - clean, fast, good security margin
 - Serpent (Euro) - slow, clean, v. high security margin
 - Twofish (USA) - complex, v. fast, high security margin
 - then subject to further analysis & comment
 - saw contrast between algorithms with
 - few complex rounds verses many simple rounds
 - which refined existing ciphers verses new proposals
-



The AES Cipher - Rijndael

- designed by Rijmen-Daemen in Belgium
 - has 128/192/256 bit keys, 128 bit data
 - 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
-



Rijndael

- data block of 4 columns of 4 bytes is state
 - key is expanded to array of words
 - has 9/11/13 rounds in which state undergoes:
 - byte substitution (1 S-box used on every byte)
 - shift rows (permute bytes between groups/columns)
 - mix columns (subs using matrix multiply of groups)
 - add round key (XOR state with key material)
 - view as alternating XOR key & scramble data bytes
 - initial XOR key material & incomplete last round
 - with fast XOR & table lookup implementation
 - Read Chapter 5 for more information. We do not study the design of AES in this subject.
-

Ways to make block encryption more complex and more secure

- Increase n , the block length and vary E , encryption function.
By varying the operating modes of encryption.
- There are five main operating modes of block encryption.
Refer Chapter 6 of the text by William Stallings.

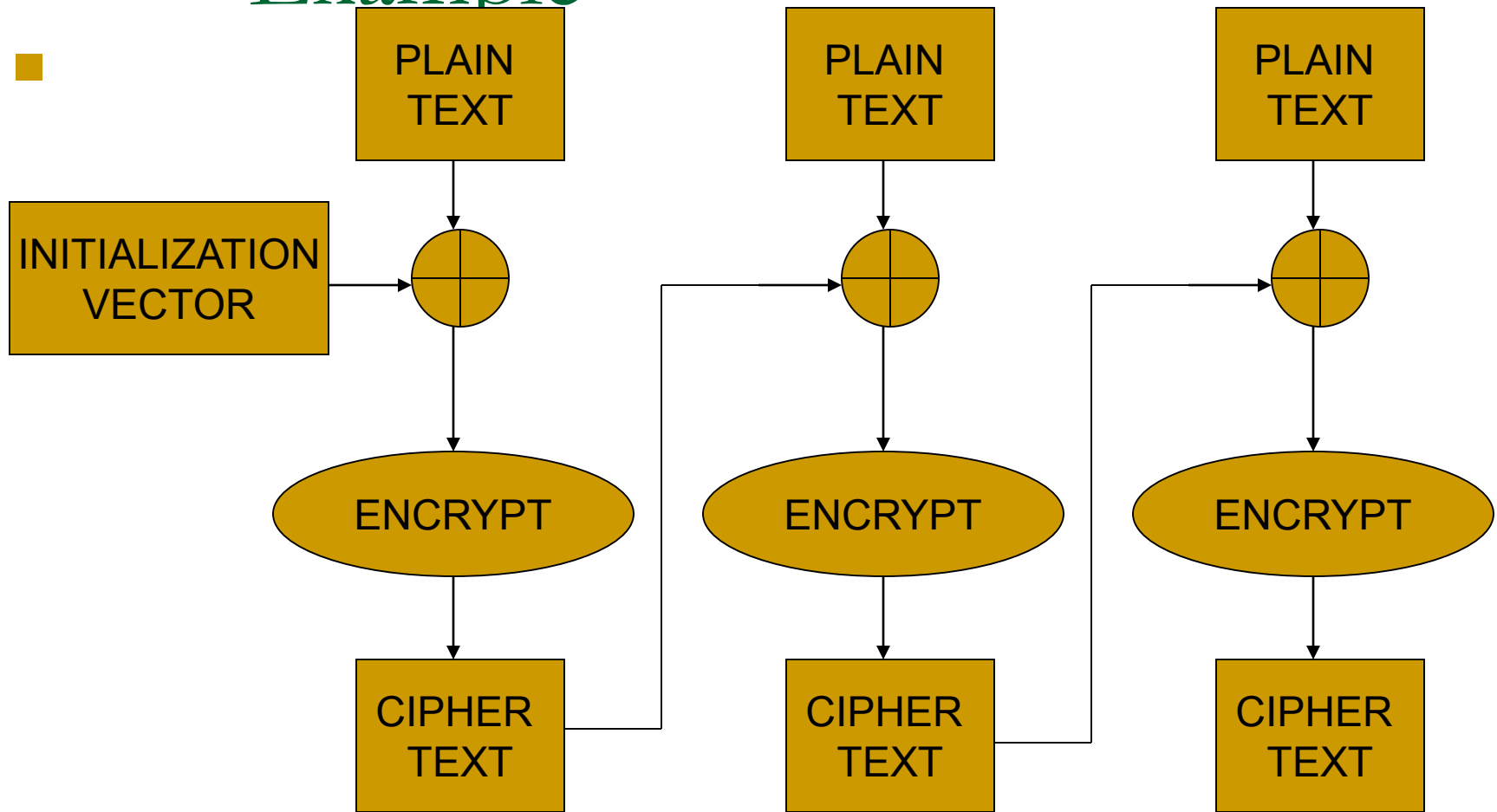


Block Cipher Modes

- ECB (electronic codebook) mode simply encrypts each 64-bit block of plaintext one after another under the same 56-bit DES key.
- In CBC (cipher block chaining) mode, each 64-bit plaintext block is bitwise XORed with the previous ciphertext block before being encrypted with the DES key.
- CFB (cipher feedback) mode allows one to use DES with block lengths less than 64 bits.
- The OFB mode essentially allows DES to be used as a stream cipher.

Shared Encryption Modes:

Example



Stream Ciphers

- Recently more popular with adhoc wireless network.
- Fast and easy to implement
- Typical examples: Bluetooth, A3, A5.
- Increasingly being used as main building blocks to secure information flow between short range wireless links, mobile phones and other portable devices.