CS 161 Final Cheat Sheet

Kerchoff's Principle

You should not rely on the secrecy of the algorithm/protocol and or keysize, as wall as the possible plain text for security because eventually the adversary will figure them out.

Mono-Alphabetic Ciphers: 1 to 1 mapping of characters to symbols

- Subsitution
  - Shift or Caesar's Cipher $E_k(m) \leftarrow m + k(\texttt{mod } N)$ $D_k(c) \leftarrow c - k(\texttt{mod } N)$
  - Affine Cipher: $E_k(m) \leftarrow k_! m + k_2(\texttt{mod } N)$ $D_k(c) \leftarrow k_!^{-1}(c - k(\texttt{mod } N)$
  - Substitution Ciphers have an extreme vulnerability to frequency attacks.

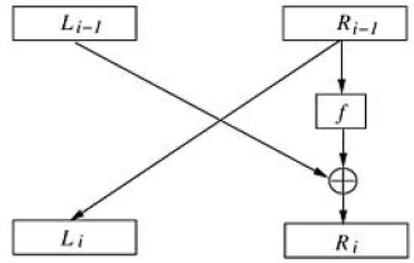
Poly-Alphabetic Ciphers

- Vigenere Cipher: Shift by a repeated key
- Book Cipher (Beale Cipher) key is hidded in a passage of a set book.
- Vernam Cipher
  - Message is m bits and the key is n bits.
  - Bitwise xor the message and the key, if m is greater than n, then use the key multiple times.
- One-Time Pad
  - Same idea as the Vernam Cipher except we use a key that is the same length or greater than the length of the message, then discard it after each use.
- Transposition/Permutation Cipher
  - Break the message into n bit blocks, then on each block perfom the same permutation
  - Despite being polyalphabit, the cipher is still vulnerable to frequency attacks. Because the original patterns are still basically present. You can attack by checking anagrams.

Data Encryption Standard (DES)

DES is a block cipher in which messages are divided into data blocks of a fixed length and each block is treated as one message either in M or in C. The DES encryping and decryption algorithms take as an input a 64-bit plaintext or ciphertext message and a 56-bit key, and output a 64-bit ciphertext or plaintext message. DES is done in 3 steps:

1. Apply a fixed "initial permutation" IP to the input block. $(L_0, R_0) \leftarrow IP(\texttt{Input Block})$ This step has no apparent cryptographic significance.
2. Iterate the following 16 rounds of operations (Feistel Cipher)



- the function is nonlinear and is considered a Substitution Cipher
- the move from $L_i \rightarrow R_{i-1}$ is a Transposition cipher
- Vernam cipher is used at the xor
- k is a 48 bit subsection of the 56 bit, "round key"

Single DES

- vulnerable to brute force or exaustive key search attacks

Triple DES

Triple DES uses an encryption-decryption-encryption scheme,
$c \leftarrow E_{k_1}(D_{k_2}(E_{k_1}(m)))$
$m \leftarrow D_{k_1}(E_{k_2}(D_{k_1}(m)))$
This scheme enlarges the keyspace while maintaining backward compatibility with single DES if $k_1 = k_2$

Advanced Encryption Standard (AES)

AES is a block cipher with variable block size and variable keysize. (block size can be 128, 192, 256 bit)
AES has 4 states:

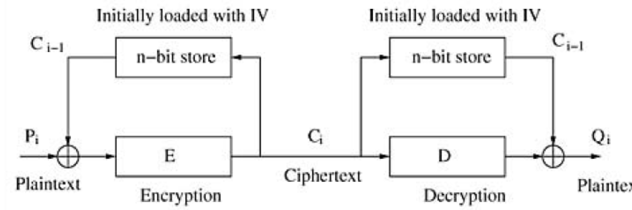
1. Sub Bytes State: nonlinear substitution on each byte
2. Shift Rows State: Transposition rearranges the order of elements in each row
3. Mix Columns State: Polynomial multiplication after converting column to polynomial.
4. Add Round Key State: adds elements of round key to the state, basically bitwise "OR"

Decryption is the inverse of these steps.

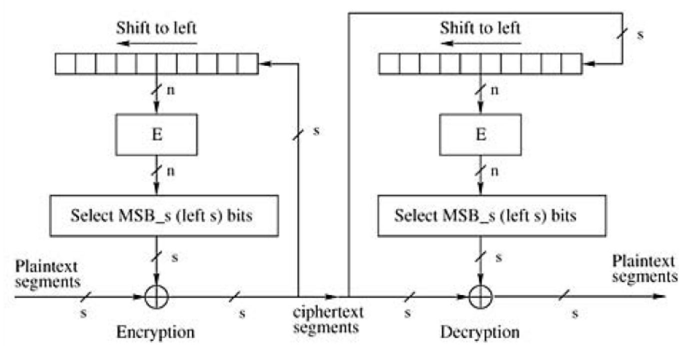
Confidentiality Modes of Operation

Different modes of operation have been devised on top of an underlying block cipher algorithm

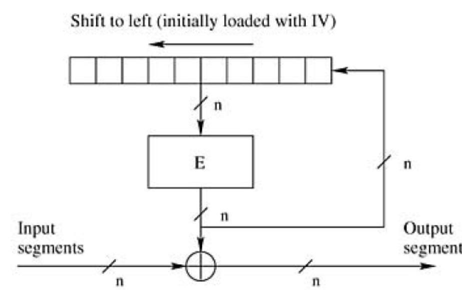
- Electronic Codebook (ECB) Mode This mode encrypts and decrypts every block seperately. It is deterministic and leaves patterns in the cipher text. (for example images.)
- Cipher Block Chaining (CBC) Mode
  - This is the most common mode of operation. In this mode the output is a sequence of n-bit cipher blocks which are chained together so that each cipher block is dependent on all the previous data blocks.
  - Decryption can be done in parallel
  - CBC cannot prived data integrity protection.
  - If the CBC claims data integrity protection, Eve can use (Bomb Oracle Attack) a Decryption Oracle to figure out the padding scheme and eventually the last byte of the cipher text.



- Cipher Feedback (CFB) Mode
  - CFB mode of opration features feeding successive cipher segments which are output from the mode back as input to the underlying block cipher algorithm.
  - CFB requires an IV as the initial n-bit input block



- Output Feedback (OFB) Mode
  - The OFB mode feeds successive output blocks from the underlying block cipher back to it.
  - The feedback blocks form a string of bits which used as the key stream of the Vernam cipher.



- Counter (CTR) Mode
  - The CTR mode features feeding the underlying block cipher algorithm with a counter value which counts up from an initial value. With a counter counting up, the underlying block cipher algorithm outputs successive blocks to form a string of bits. This string of bits is used as the key stream of the vernam cipher, that is, the key stream is XOR-ed with the plaintext blocks. $C_i \leftarrow P_i \oplus E(Ctr_i, i = 1, 2, \ldots, m$ $P_i \leftarrow C_i \oplus E(Ctr_i, i = 1, 2, \ldots, m$

Bomb Oracle Attack

Asymmetric Cryptography

Oneway Trapdoor Function

- Asymmetric crypto system, Public Key Cryptography
- $D \rightarrow R$ is oneway, it is easy to evaluate $\forall x \in D$ and difficult to invert for all values in R.

Textbook Encryption Algorithms

- All or Nothing Secrecy: Given Cipher Text the attacker must not be able to get any information about the plain text
- Passive Attacker: The attacker doesn't modify or manipulate ciphertexts they also don't ask for encryption or Decryption services.

Diffie-Hellman Key Exchange Protocol

`Common Input` $(p, g) : p$ `is a large prime, g is a generator element in` F_p^*

`Output` `An element in` F_p^* `shared between Alice Bob.`

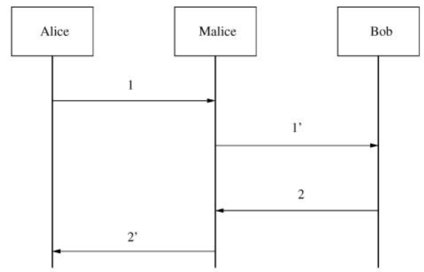
1. Alice picks $a \in U(1, p-1)$; computes $g_a \leftarrow g^a (\text{mod } p)$; sends g_a to Bob.
2. Bob picks $b \in U(1, p-1)$; computes $g_b \leftarrow g^b (\text{mod } p)$; sends g_b to Alice.
3. Alice computes $k \leftarrow g_b^a (\text{mod } p)$
4. Bob computes $k \leftarrow g_a^b (\text{mod } p)$

Alice and Bob both compute the same key,

$$k = g^{ba} (\text{mod } p) = g^{ab} (\text{mod } p)$$

P is a public 2048 bit prime number.

Man in the Middle Attack on Diffie-Helman



1. Alice picks a $\in_u [1, p-1)$, computes $g_a \leftarrow g^a (\text{mod } p)$ she sends g_a to Malice("bob");
2. (1') Malice("Alice") computes $g_m \leftarrow g^m (\text{mod })$ for some $m \in [1, p-1)$; he sends g_m to Bob;
3. (2) Bob picks $b \in_U [1, p-1)$, computes $g_b \leftarrow g^b (\text{mod } p)$; he sends g_b Malice("Alice");
4. (2') Malice("Bob") sends to Alice: g_m;
5. (3) Alice computes $k_1 \leftarrow g_m^a (\text{mod } p)$;
6. (4) Bob computes $k_2 \leftarrow g_m^b (\text{mod } p)$;

Diffie-Helman and the Discrete Logarithm Problem

- Computational Diffie-Hellman Problem
- Discrete Logarithm Problem