Justin Ciocoi

Oct. 9, 2023

CSCI 360 Textbook Notes

More About Block Ciphers

Introduction

- While block ciphers are the basis for both DES and AES, and can be used in encryption algorithms, they can also be utilized as building blocks through which various other cryptographic functions could be accomplished
- This chapter will focus on supplementary information on block ciphers, especially:
    - The most important *modes of operation*
    - The *security pitfalls* of these different modes
    - The principles of *key whitening*
    - *Meet-in-the-middle attacks*
    - Why *double encryption* is not a good idea
    - *Triple encryption*

5.1: Encryption with Block Ciphers: Modes of Operation

- Realistically, one would want to encrypt more than 8 or 16 bytes of plain-text at a time
- There are several ways of encrypting longer plain-text passages, some of which include:
    - **Electronic Code-Book Mode (ECB)**
    - **Cipher Block Chaining Mode (CBC)**
    - **Cipher Feedback Mode (CFB)**
    - **Output Feedback Mode (OFB)**
    - **Counter Mode (CTR)**

- All five of these modes have the same goal of encrypting data and providing confidentiality to a message sent between two parties
- In practice, however, authentication, or making sure the sender is who they say they are, is also important and the **Galois Counter Mode** exists to provide this authentication
- **5.1.1: Electronic Code-book Mode (ECB)**
    - ECB Mode is the most straightforward method of encryption
    - Let $e_k(x_i)$ denote the encryption of a plain-text block, x_i, with key k
    - Let ${e_k}^{-1}(y_i)$ denote the decryption of cipher-text block y_i with key k
    - Assume the block size encrypted by the block cipher is b bits
    - Below is a diagram representing the fundamental idea of ECB Mode

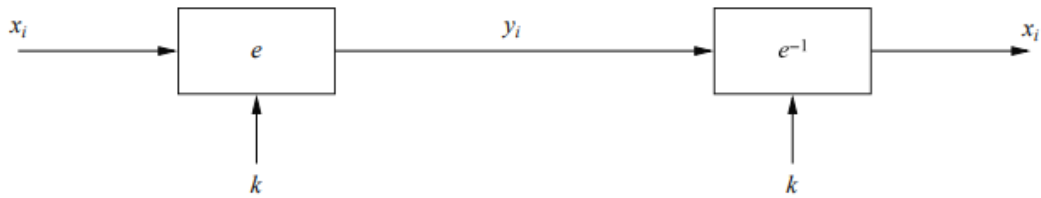


Fig. 5.1 Encryption and decryption in ECB mode

    - ***Formal Definition of ECB Mode***
        - Let $e()$ be a block cipher of block size b, and let x_i and y_i be bit strings of length b
        - **Encryption:** $y_i = e_k(x_i), i \geq 1$
        - **Decryption:** $x_i = e_k^{-1}(y_i) = e_k^{-1}(e_k(x_i)), i \geq 1$
    - ECB Mode's biggest weakness is that it encrypts data in a deterministic manner
    - To illustrate this weakness, let us consider the case of a substitution attack against an electronic bank transfer
    - Let us assume that a bank transfer is defined by 5 fields, each of which is 16 bytes long

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount $

Fig. 5.2 Example for a substitution attack against ECB encryption

- The attacker could theoretically take the following steps
  - The attacker opens two bank accounts, one at bank Am and one at bank B
  - The attacker then taps the encrypted line of communication between banks
  - Then, the attacker sends $1.00 transfers between accounts repeatedly, and takes note of the repeating blocks of cipher-text
  - Now, the attacker can replace the cipher-text field for the receiving account to the cipher-text corresponding to their account, making all transfers enter his account
  - Finally, the attacker withdraws the money and quickly moves to a country with a relaxed attitude on the extradition of white-collar criminals
- The weakness of ECB Mode can also be shown through the encryption of bitmaps using ECB Mode, which is illustrated very clearly in the below image

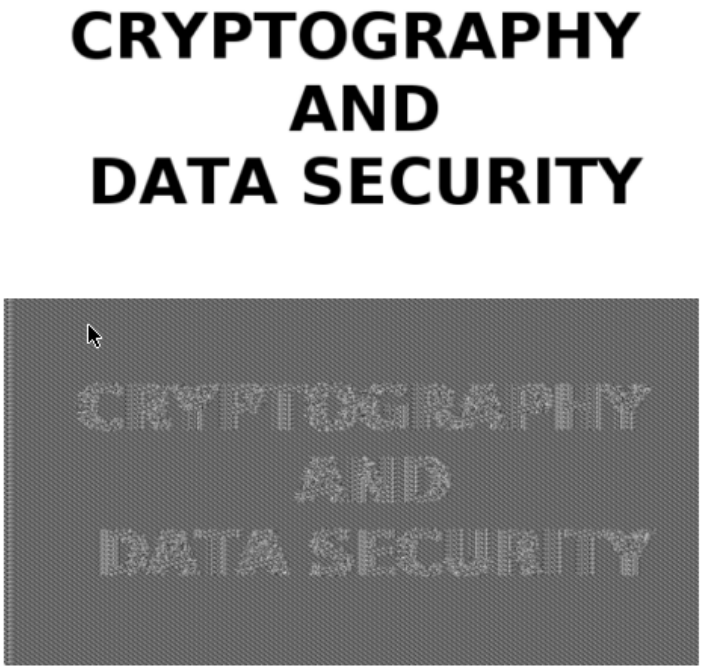
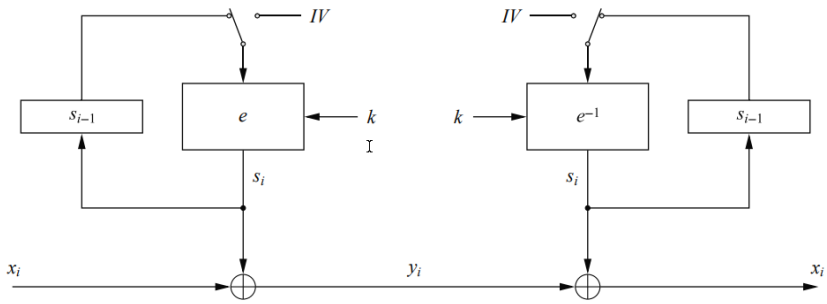


Fig. 5.3 Image and encrypted image using AES with 256-bit key in ECB mode

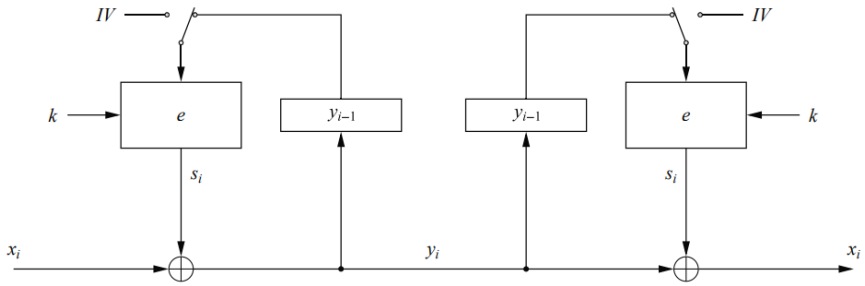
- **5.1.2: Cipher Block Chaining Mode (CBC)**
  - In CBC Mode, the encryption of each block is "*chained*" together such that cipher-text y_i depends not only on plain-text block x_i, but on all other previous plain-text blocks as well
  - ***Formal Definition of CBC Mode***
    - Let $e()$ be a block cipher of block size b; let x_i and y_i be bit strings of length b; let IV be a *nonce (number used once)* of length b
    - **Encryption (First Block):** $y_1 = e_k(x_1 \oplus IV)$
    - **Encryption (General Block):** $y_i = e_k(x_i \oplus y_{i-1})$, for $i \geq 2$
    - **Decryption (First Block):** $x_1 = e_k^{-1}(y_1) \oplus (IV)$
    - **Decryption (General Block):** $x_i = e_k^{-1}(y_i) \oplus y_{i-1}$, for $i \geq 2$
  - To verify correctness, we can do the following
    - *For the first block:*
  - $d(y_i) = e_k^{-1}(y_i) \oplus y_{i-1} = e_k^{-1}(e_k(x_i \oplus y_{i-1})) \oplus y_{i-1} = (x_i \oplus y_{i-1}) \oplus y_{i-1} = x_i$
    - *For the general block:*
  - $d(y_1) = e_k^{-1}(y_1) \oplus IV = e_k^{-1}(e_k(x_1 \oplus IV)) \oplus IV = (x_1 \oplus IV) \oplus IV = x_1$
  - If a new IV is chosen each time a CBC cipher is used, then the encryption function becomes non-deterministic
  - The IV does not have to be kept secret, only the key
  - However, some advanced attacks have emerged that do provide function in keeping the IV secret
  - For the previous substitution attack example, this attack should be unsuccessful because the attacker would not be able to recognize any patterns in the transmitted cipher-text
- **Output Feedback Mode (OFB)**
  - OFB starts with encrypting an IV with a block cipher

- The next block of *key stream bits* is computed by feeding the previous cipher output into the block cipher and encrypting it, after which the process is repeated
- This is a *synchronous* stream cipher as the key stream does not depend on either the plain- or cipher-text
- The encryption and decryption in OFB Mode are essentially the same operation and are depicted in the diagram below



- **Fig. 5.5** Encryption and decryption in OFB mode
- ***Formal Definition of OFB Mode***
  - Let $e()$ be a block cipher of size b; let x_i and y_i and s_i be bit strings of length b; let IV be a *nonce* of length b
  - **Encryption (First Block):** $s_1 = e_k(IV)$ and $y_1 = s_1 \oplus x_1$
  - **Encryption (General Block):** $s_i = e_k(s_{i-1})$ and $y_i = s_i \oplus x_i$, for $i \geq 2$
  - **Decryption (First Block):** $s_1 = e_k(IV)$ and $x_1 = s_1 \oplus y_i$
  - **Decryption (General Block):** $s_i = e_k(s_{i-1})$ and $x_i = s_i \oplus y_i$
- Due to the use of an IV the encryption provided by OFB Mode is, like CBC Mode, non-deterministic

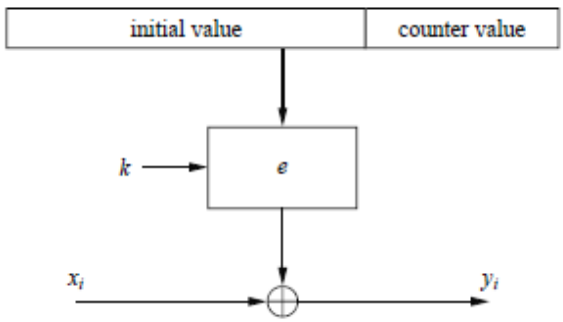
- **5.1.4: Cipher Feedback Mode (CFB)**
  - CFB Mode largely resembles OFB Mode, however instead of the block cipher's output being fed back, the previous cipher text is fed back to generate the next b key stream bits
  - As in OFB, encryption and decryption are essentially the same operation
  - Below is a representation of encryption and decryption in CFB Mode



- **Fig. 5.6** Encryption and decryption in CFB mode

- ***Formal Definition of CFB Mode***
  - Let $e()$ be a block cipher of block size b; let x_i and y_i be bit strings of length b; let IV be a *nonce* of length b
  - **Encryption (First Block):** $y_1 = e_k(IV) \oplus x_1$
  - **Encryption (General Block):** $y_i = e_k(y_{i-1}) \oplus x_i$, for $i \geq 2$
  - **Decryption (First Block):** $x_1 = e_k(IV) \oplus y_1$
  - **Decryption (General Block):** $x_i = e_k(y_{i-1}) \oplus y_i$, for $i \geq 2$
- Similarly to OFB Mode, the use of an IV makes this method of encryption non-deterministic

- **5.1.5: Counter Mode (CTR)**
  - Similar to OFB and CFB Modes, however the keystream is derived from the input of a counter that assumes a different value each time the block cipher computes a new key stream block



- **Fig. 5.7** Encryption and decryption in counter mode

- ***Formal Definition of CTR***
  - Let $e()$ be a block cipher of size b; let x_i and y_i be bit strings of length b

        - The concatenation of the initialization value and counter, CTR_i, is denoted by $(IV||CTR_i)$ and is a bit string of length b
        - **Encryption:** $y_i = e_k(IV||CTR_i) \oplus x_i$, for $i \geq 1$
        - **Decryption:** $x_i = e_k(IV||CTR_i) \oplus y_i$, for $i \geq 1$
    - This model is desirable in cases largely due to the fact that it can be *parallelized* since there is no feedback needed

- **5.1.6: Galois Counter Mode (GCM)**
    - ***Formal Definition of GCM***
        - Let $e()$ be a block cipher of block size 128 bits; let x be the plaintext consisting of $x_1, ..., x_n$; let AAD be the *additional authenticated data*
        - **Encryption**
            - Derive a counter value, CTR_0 from the IV and compute $CTR_1 = CTR_0 + 1$
            - Compute ciphertext: $y_i = e_k(CTR_i) \oplus x_i$, for $i \geq 1$
        - **Decryption**
            - Generate authentication subkey $H = e_k(0)$
            - Compute $g_0 = AAD * H$ (Galois Field Multiplication)
            - Compute $g_i = (g_{i-1} \oplus y_i) * H$, for $1 \leq i < n$ (Galois Field Multiplication)
            - Final authentication tag $T = (g_n * H) \oplus e_k(CTR_0)$
    - Below is a representation of the Galois Field Counter

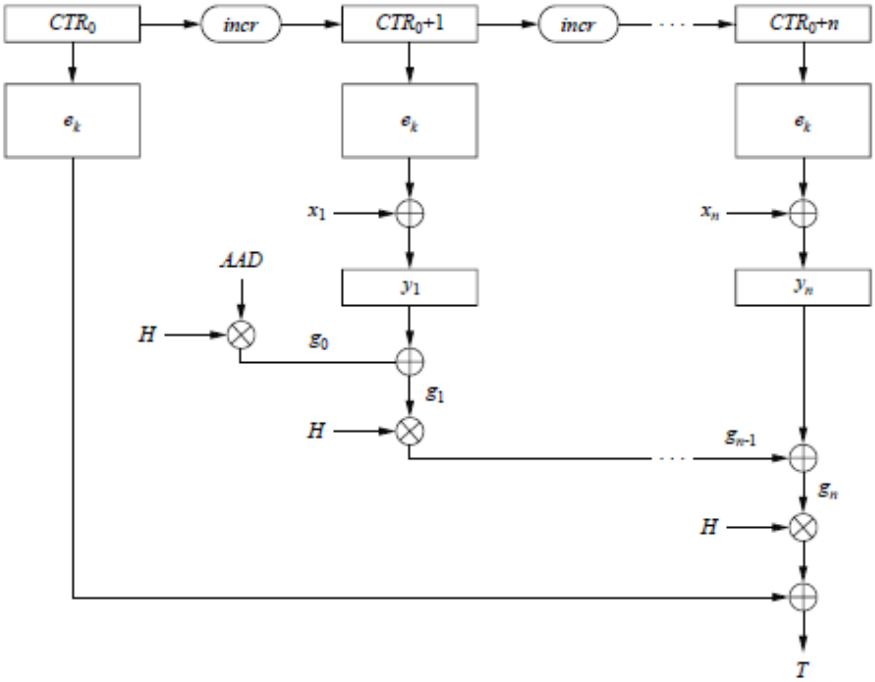


Fig. 5.8 Basic authenticated encryption in Galois Counter mode

Exhaustive Key Search Revisited

- A brute force attack is capable of producing *false positive* results
- Because of this, multiple plain- cipher-text pairs are needed to brute-force attack most modern ciphers
- The length of the respective plain-text required to break a cipher using a brute-force attack is known as the *unicity distance*
- ***Theorem 5.2.1***
  - Given a block cipher with a key length of κ bits and block size n bits as well as t plain- and cipher-text pairs, $(x_1, y_1), \ldots, (x_n, y_n)$, the expected number of false keys which encrypt all plain-texts to the corresponding cipher-texts is:
    - $2^{\kappa - tn}$