Applied Cryptography and Computer Security

CSE 664 Spring 2017

Lecture 5: Symmetric Encryption II

Department of Computer Science and Engineering
University at Buffalo

Symmetric Encryption

- Recall types of attacks against an encryption scheme
  - ciphertext only
  - known plaintext
  - chosen plaintext
  - chosen ciphertext
- In this lecture, we
  - move towards security against more powerful adversaries
  - learn about block ciphers

Security Against Chosen-Plaintext Attacks

- **In chosen-plaintext attack (CPA), adversary $\mathcal{A}$ is allowed to ask for encryptions of messages of its choice**
  - **it is now active and adaptive**
- **$\mathcal{A}$ is given black-box access to encryption oracle and can query it on different messages**
  - **notation $\mathcal{A}^{\mathcal{O}(\cdot)}$ means $\mathcal{A}$ has oracle access to algorithm $\mathcal{O}$**
- **As before, $\mathcal{A}$ is asked to distinguish between encryptions of messages of its choice**
- **Is this model too strong?**

CPA Security

- **CPA indistinguishability experiment** $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\mathcal{E}}(n)$

  1. **random key** k **is generated by** $\mathsf{Gen}(1^n)$
  2. $\mathcal{A}$ **is given** 1^n **and ability to query** $\mathsf{Enc}_k(\cdot)$**, and chooses two messages** m_0, m_1 **of the same length**
  3. **random bit** $b \leftarrow \{0, 1\}$ **is chosen, challenge ciphertext** $c \leftarrow \mathsf{Enc}_k(m_b)$ **is computed and given to** $\mathcal{A}$
  4. $\mathcal{A}$ **can use** $\mathsf{Enc}_k(\cdot)$ **and eventually outputs bit** b'
  5. **experiment outputs 1 if** $b' = b$ **(**$\mathcal{A}$ **wins) and 0 otherwise**

- $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ **has indistinguishable encryptions under the chosen-plaintext attack (CPA-secure) if for all PPT** $\mathcal{A}$

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\mathcal{E}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

CPA Security

- How come adversary is allowed to query Enc_k on a message and later use that message for the challenge?
- How does this notion of security compare to the indistinguishability against eavesdroppers?
- How about security for multiple encryptions?
  - good news! no need for other definitions
  - then really long messages can be treated as several fixed-length messages

Towards CPA-Secure Encryption

- **We are going to use a new building block: pseudorandom functions**
  - **just like pseudorandomness of one string doesn't make sense, we'll consider a distribution (or class) of functions**
  - **we'll look at keyed functions** $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$
    - **the first argument is the key** k **and second argument is the input** x
  - **once the key is fixed, the function** $F_k : \{0,1\}^n \to \{0,1\}^n$ **is fixed**
- **Pseudorandom property is now defined as**
  - **a computationally limited adversary cannot distinguish behavior of a pseudorandom function** F_k **(for a randomly chosen and secret** k**) from a function** f **chosen at random**

Towards CPA-Secure Encryption

- f is one of all possible functions that map n-bit inputs to n-bit outputs
  - each function can be specified as a lookup table
  - if f is chosen at random, outputs $f(x)$ and $f(y)$ are uniformly distributed and independent
- Pseudorandomness property of F_k no longer holds if
  - key k is known or not chosen at random
  - adversary is not bounded by polynomial (in n) time

Towards CPA-Secure Encryption

- **Definition: An efficient function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a pseudorandom function if any PPT distinguisher D cannot tell apart outputs of F_k and f, i.e.,**

  $$|\mathsf{Pr}[D^{F_k(\cdot)}(1^n) = 1] - \mathsf{Pr}[D^{f(\cdot)}(1^n) = 1]| \le \mathsf{negl}(n)$$

  for a uniformly chosen function $f : \{0,1\}^n \to \{0,1\}^n$ and uniformly chosen key $k \leftarrow \{0,1\}^n$

- **Pseudorandom functions are useful for different purposes in cryptography**
  - **we start with CPA-secure encryption schemes**

CPA-Secure Encryption

- **Intuitively, F_k enciphers its input (message?) rather well**
  - **the problem is that $F_k(m)$ is deterministic, not sufficient**
  - **how do we randomize encryption?**
- **Solution for CPA-secure encryption**
  - Gen**: on input 1^n, choose $k \overset{R}{\leftarrow} \{0,1\}^n$**
  - Enc**: on input key $k \in \{0,1\}^n$ and message $m \in \{0,1\}^n$, choose $r \overset{R}{\leftarrow} \{0,1\}^n$ and output ciphertext $c := (r, F_k(r) \oplus m)$**
  - Dec**: on input key $k \in \{0,1\}^n$ and ciphertext $c = (c_1, c_2)$, output message $m = F_k(c_1) \oplus c_2$**

CPA-Secure Encryption

- **Theorem:** Given that F is a pseudorandom function, the above construction is a CPA-secure encryption scheme for n-bit messages

- Proof idea:

  1. Suppose that random function f is used in place of F_k. Proof the construction secure.
  2. Replace f with F_k and show that any negligible advantage in breaking indistinguishability has to come from the use of F_k.

CPA-Secure Encryption in Practice

- **Block ciphers used in practice are keyed permutations**
  - **can we use them in place of pseudorandom functions and still get the proper level of security?**
- **Define pseudorandom permutation similar to pseudorandom functions**
  - **efficient, negligible advantage in distinguishing from a random permutation**
- **Claim: a pseudorandom permutation is also a pseudorandom function**
  - **probability of collision in a pseudorandom function is negligible**
- **We also want to be able to invert pseudorandom permutation F_k**
  - **i.e., block cipher decryption algorithm**

CPA-Secure Encryption in Practice

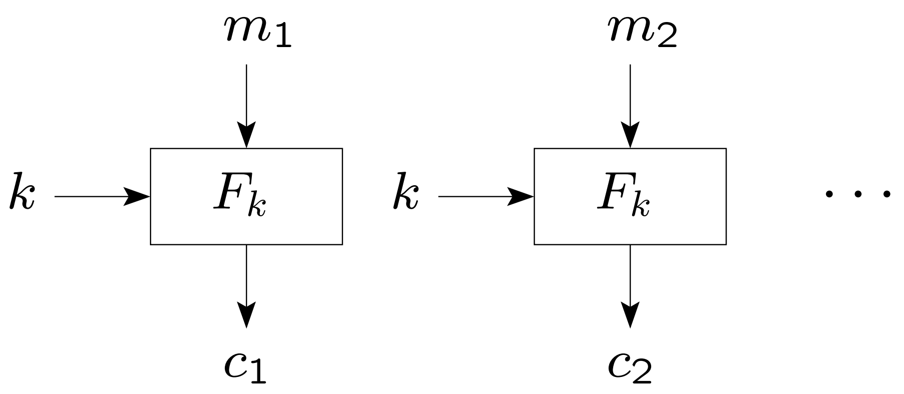
- How about messages of sizes other than n?
  - shorter messages
  - really long messages
- Short messages
  - unambiguously pad the message to be n bits
  - often can append a "1" followed by the necessary number of "0"s
- Messages longer than n
  - partition message into blocks of size n: $m = m_1 m_2 ... m_\ell$
  - encrypting each block separately results in doubling message length
  - modes of encryption with less expansion exist

Encryption Modes

- **Encryption modes indicate how messages longer than one block are encrypted and decrypted**
- **4 modes** of operation were standardized in 1980 for Digital Encryption Standard (DES)
  - can be used with any block cipher
  - electronic codebook mode (ECB), cipher feedback mode (CFB), cipher block chaining mode (CBC), and output feedback mode (OFB)
- **5 modes** were specified with the current standard Advanced Encryption Standard (AES) in 2001
  - the 4 above and counter mode

Encryption Modes

- **Electronic Codebook (ECB) mode**
  - **divide the message m into blocks $m_1 m_2 ... m_\ell$ of size n each**
  - **encipher each block separately: for $i = 1, \ldots, \ell$, $c_i = F_k(m_i)$**
  - **the resulting ciphertext is $c = c_1 c_2 ... c_\ell$**

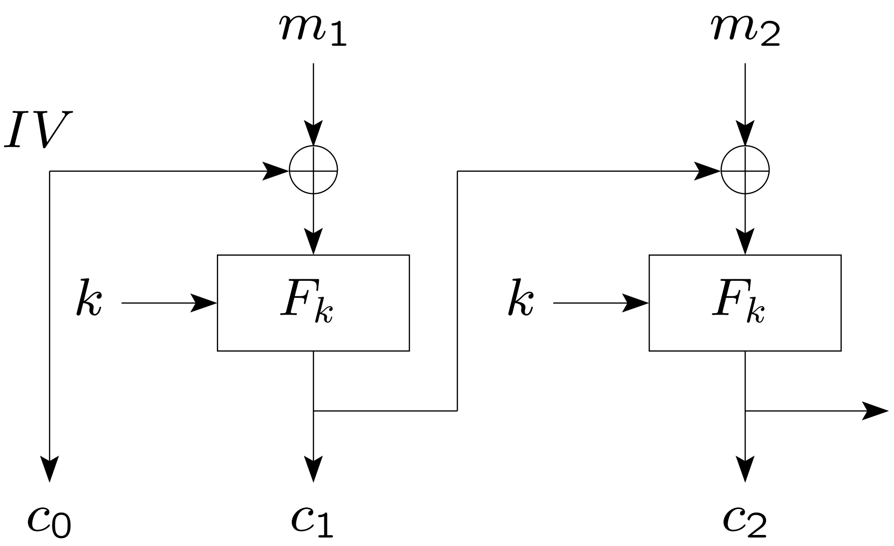


Encryption Modes

- **Properties of ECB mode:**
  - identical plaintext blocks result in identical ciphertexts (under the same key)
  - each block can be decrypted independently
  - errors in a ciphertext block do not affect other blocks
- **Is it secure?**

Encryption Modes

- **Cipher Block Chaining (CBC) mode**
  - **set** $c_0 = IV \overset{R}{\leftarrow} \{0,1\}^n$ **(initialization vector)**
  - **encryption: for** $i = 1, \ldots, \ell$**,** $c_i = F_k(m_i \oplus c_{i-1})$
  - **decryption: for** $i = 1, \ldots, \ell$**,** $m_i = c_{i-1} \oplus F_k^{-1}(c_i)$



Encryption Modes

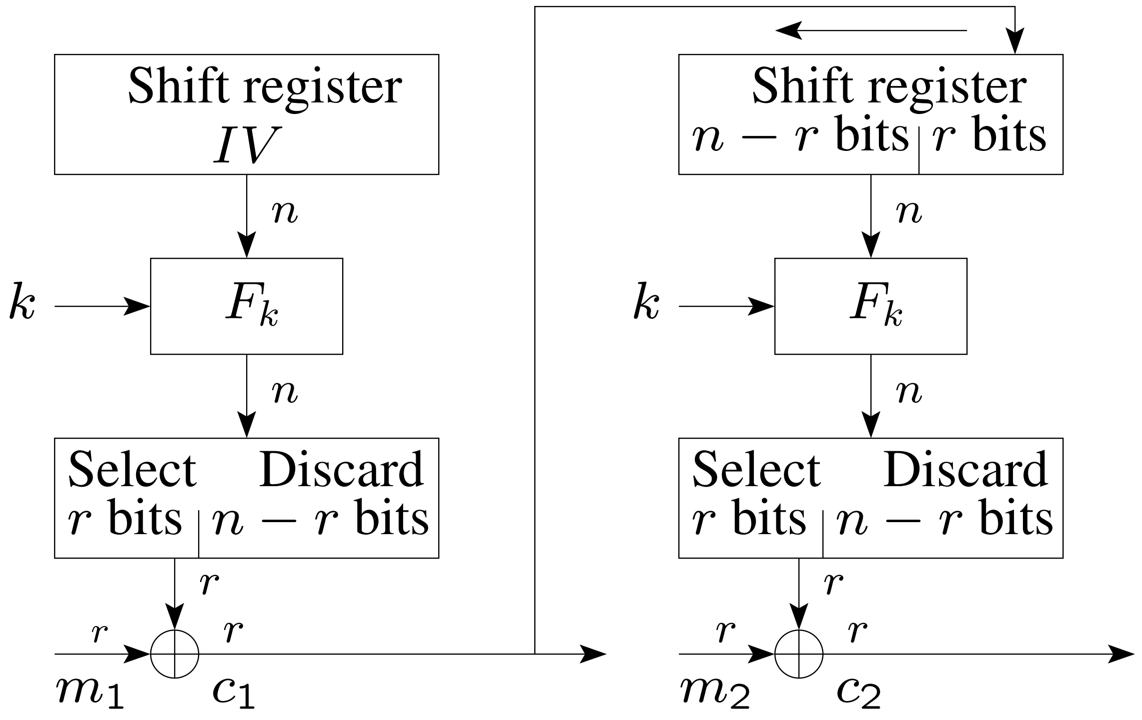
- **Properties of CBC mode:**
  - **if F is a pseudorandom permutation, this mode is CPA-secure**
  - **a ciphertext block depends on all preceding plaintext blocks**
  - **sequential encryption, cannot use parallel hardware**
  - **IV must be random and communicated intact**
    - **if the IV is not random, security quickly degrades**
    - **if someone can fool the receiver into using a different IV, security issues arise**

Encryption Modes

- **Cipher Feedback (CFB) mode**
  - **the message is XORed with the encryption of the feedback from the previous block**
  - **set initial input** $I_1 = IV$
  - **encryption:** $c_i = F_k(I_i) \oplus m_i$; $I_{i+1} = c_i$
  - **decryption:** $m_i = c_i \oplus F_k(I_i)$
- **This mode allows the block cipher to be used as a stream cipher**
  - **if our application requires that plaintext units shorter than the block are transmitted without delay, we can use this mode**
  - **the message is transmitted in** r**-bit units (**r **is often 8 or 1)**

Encryption Modes

- **Cipher Feedback (CFB) mode**
  - **input: key k, n-bit IV, r-bit plaintext blocks $m_1, \ldots$**
  - **output: r-bit ciphertext blocks $c_1, \ldots$**

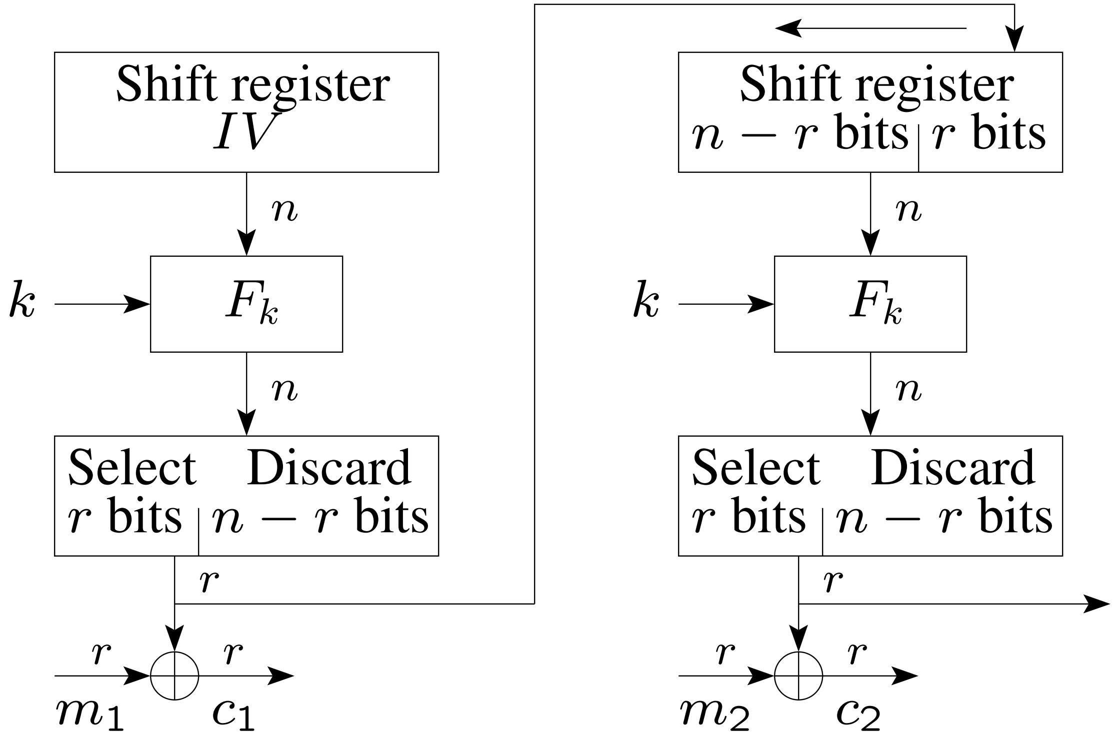


Encryption Modes

- **Properties of CFB mode:**
  - the mode is CPA-secure
  - similar to CBC, a ciphertext block depends on all previous plaintext blocks
  - decreased throughput when used on small units
    - one encryption operation is applied per r bits, not per n bits

Encryption Modes

- **Output Feedback (OFB) mode**
  - **similar to CFB, but the feedback is from encryption output and is independent of the message**

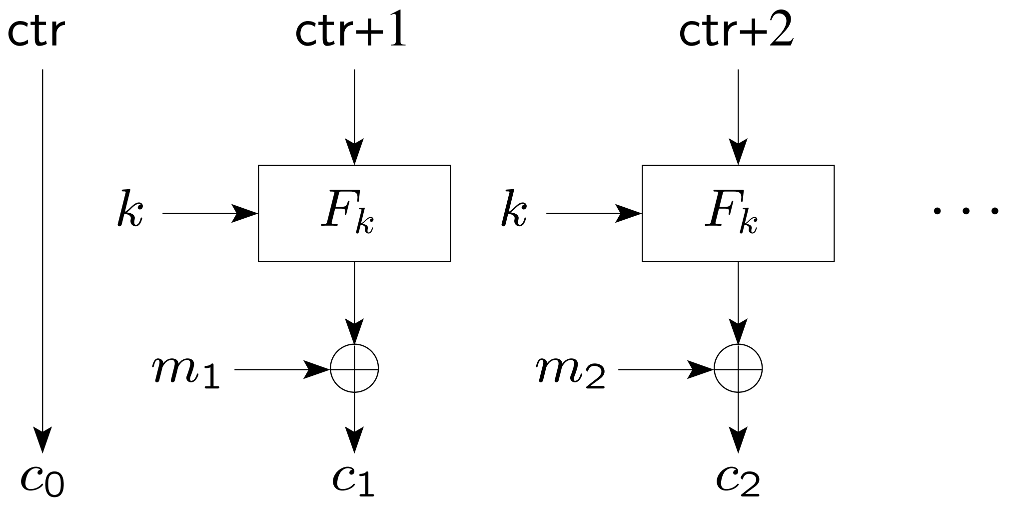


Encryption Modes

- **Output Feedback (OFB) mode:**
  - **n-bit feedback is recommended**
  - **using fewer bits for the feedback reduces the size of the cycle**
- **Properties of OFB:**
  - **the mode is CPA-secure**
  - **the key stream is plaintext-independent**
  - **OFB may be used for applications in which all error propagation must be avoided**
  - **similar to CFB, throughput is decreased for $r < n$, but the key stream can be precomputed**

Encryption Modes

- **Counter (CRT) mode**
  - **a counter is encrypted and XORed with a plaintext block**
  - **no feedback into the encryption function**
  - **initially set** $\mathsf{ctr} = IV \overset{R}{\leftarrow} \{0,1\}^n$



Encryption Modes

- **Counter (CRT) mode**
  - **encryption: for $i = 1, \ldots, \ell$, $c_i = F_k(\mathsf{ctr} + i) \oplus m_i$**
  - **decryption: for $i = 1, \ldots, \ell$, $m_i = F_k(\mathsf{ctr} + i) \oplus c_i$**
- **Properties:**
  - **ciphertext can have the same length as the plaintext**
  - **we just truncate the value and transmit it**

Encryption Modes

- Advantages of counter mode
  - Hardware and software efficiency: multiple blocks can be encrypted or decrypted in parallel
  - Preprocessing: encryption can be done in advance; the rest is only XOR
  - Random access: ith block of plaintext or ciphertext can be processed independently of others
  - Security: at least as secure as other modes (i.e., CPA-secure)
  - Simplicity: doesn't require decryption or decryption key scheduling
- But what happens if the counter is reused?

Practical Remarks

- Use good randomness
  - true randomness for long-term secrets
  - cryptographically strong pseudo-random number generator in other cases
- Stick to exact specification of a CPA-secure encryption mode
  - ECB mode has historical significance only
- Both the size of the key and block size must be sufficiently large

Message Integrity

- The above modes in general don't protect transmitted ciphertexts from tampering
  - some modes are easier to tamper with than others
  - none achieve "proper" integrity protection
- A separate integrity or message authentication mechanism should be used to ensure that the message arrives intact

Summary

- Block ciphers vs stream ciphers
  - which type is preferred?
- Notions of security for symmetric encryption
- What is next?
  - practical constructions for block ciphers
  - past and current standards