Applied Cryptography and Computer Security
CSE 664 Spring 2017

Lecture 6: Digital Encryption Standard (DES)

Department of Computer Science and Engineering
University at Buffalo

Lecture Outline

- **Previously we talked about:**
  - defining security for encryption
  - using theoretical models for encryption
- **In this lecture:**
  - move to practical constructions
  - learn about design principles of block ciphers
  - learn about how DES works

Symmetric Encryption

- **Block ciphers** are used in practice to implement pseudorandom permutations
  - it is desirable to base secure encryption on weaker assumptions than existence of pseudorandom permutations
- All block ciphers are heuristics
  - no proofs of security
- **Such algorithms are:**
  - normally very fast
  - can be used as primitives in more complex cryptographic protocols

Block Ciphers

- **The algorithm maps an n-bit plaintext block to an n-bit ciphertext block**
- **Most modern block ciphers are product ciphers**
  - **we sequentially apply more than one obfuscation technique to the message**
- **A common design for an algorithm is to proceed in iterations**
  - **one iteration is called a round**
  - **each round consists of similar operations**
  - **iterations are used to amplify obfuscation**

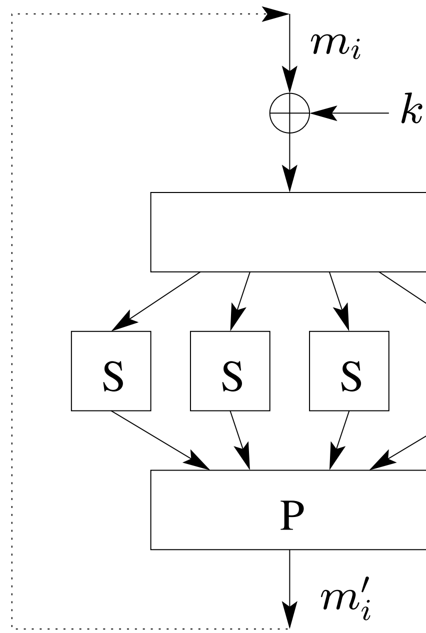
Design Principles of Block Ciphers

- **Specifying a random permutation on an n-bit block requires huge amount of storage**
  - **there are $(2^n)!$ permutations, each can be encoded in $\log(2^n!) \approx n2^n$ bits**
  - **how do we achieve similar effect with reasonable resources?**
- **Confusion-diffusion paradigm**
  - **split a block into small chunks**
  - **define a permutation on each chunk separately (confusion)**
  - **mix outputs from different chunks by rearranging bits (diffusion)**
  - **repeat to strengthen the result**

Design Principles of Block Ciphers

- **Substitution-permutation networks**
  - **since a permutation on a block can be specified as a lookup table, this is called substitution**
  - **instead of having substitutions defined by the key, such functions are fixed and applied to messages and keys**
  - **mixing algorithm is called mixing permutation**

Design Principles of Block Ciphers



- **For this type of algorithm to be reversible, each operation needs to be invertible**

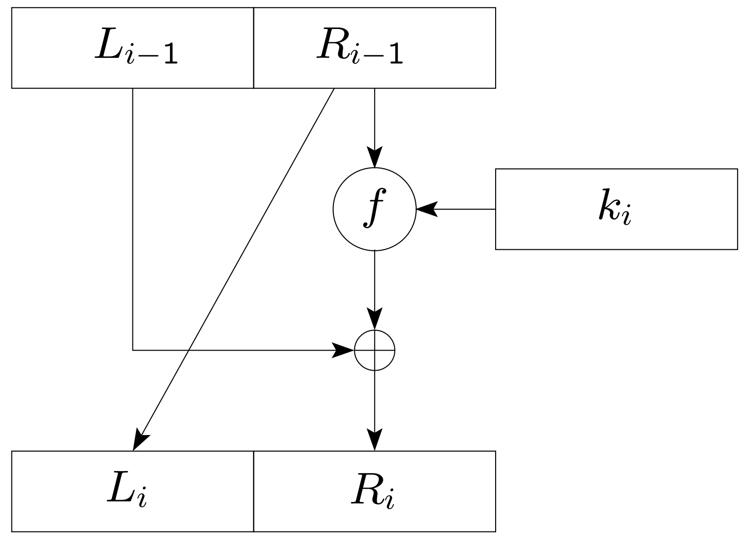
Design Principles of Block Ciphers

- **Let's denote one iteration or round by function g**
- **The initial state s_0 is the message m itself**
- **In round i:**
  - **g's input is round key k_i and state s_{i-1}**
  - **g's output is state s_i**
- **The ciphertext c is the final state s_{Nr}, where Nr is the number of rounds**
- **Decryption algorithm applies g^{-1} iteratively**
  - **the order of round keys is reversed**
  - **set $s_{Nr} = c$, compute $s_{i-1} = g^{-1}(k_i, s_i)$**

Design Principles of Block Ciphers

- **Another way to realize confusion-diffusion paradigm is through Feistel network**
  - **in Feistel network each state is divided into halves of the same length: L_i and R_i**
  - **in one round:**
    - $L_i = R_{i-1}$
    - $R_i = L_{i-1} \oplus f(k_i, R_{i-1})$

Design Principles of Block Ciphers



- **Are there any advantages over the previous design?**
  - **operations no longer need to be reversible, as the inverse of the algorithm is not used!**
  - **reverse one round's computation as $R_{i-1} = L_i$ and $L_{i-1} = R_i \oplus f(k_i, R_{i-1})$**

Design Principles of Block Ciphers

- In both types of networks, the substitution and permutation algorithms must be carefully designed
  - choosing random substitution/permutation strategies leads to significantly weaker ciphers
  - each bit difference in S-box input creates at least 2-bit difference in its output
  - mixing permutation ensures that difference in one S-box propagates to at least 2 S-boxes in next round

Block Ciphers

- **Larger key size** means greater security
  - for n-bit keys, brute force search takes $2^n/2$ time on average
- **More rounds** often provide better protection
  - the number of rounds must be large enough for proper mixing
- **Larger block size** offers increased security
  - security of a cipher also depends on the block length

Data Encryption Standard (DES)

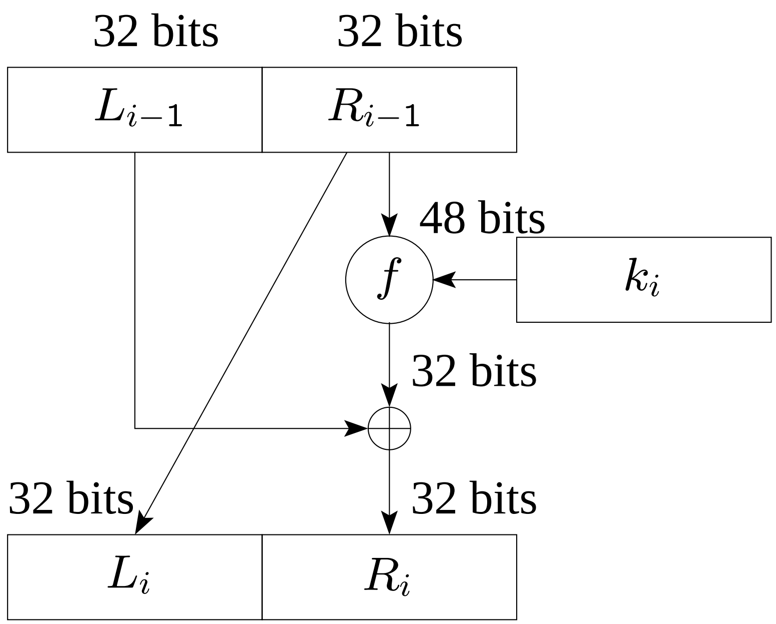
- **In 1973 NIST published a solicitation for cryptosystems**
- **NIST stands for the National Institute of Standards and Technology**
  - **previously the National Bureau of Standards**
  - **was founded in 1901 within technology administration of the US Commerce Department**
  - **develops and promotes standards and technology**
  - **cryptographic standards are published in the Federal Information Processing Standards (FIPS)**

DES

- **DES was developed by IBM as a modification of an earlier system Lucifer**
- **DES was adopted as a standard in 1977**
- **It was expected to be used as a standard for 10–15 years**
- **Was replaced only in 2001 with AES (Advanced Encryption Standard)**
- **DES characteristics:**
  - **key size is 56 bits**
  - **block size is 64 bits**
  - **number of rounds is 16**

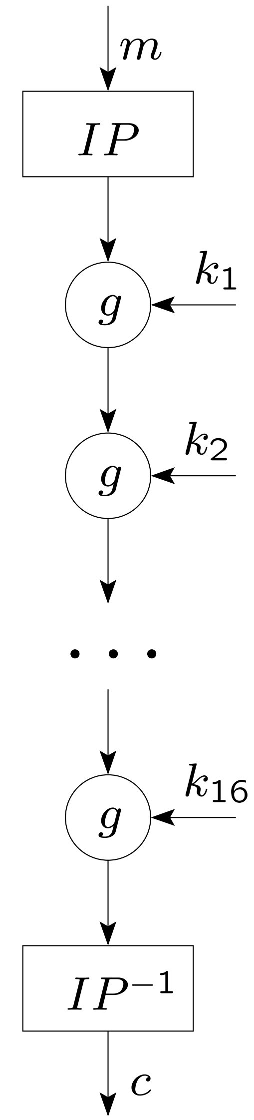
DES

- **DES uses Feistel network**
  - **Feistel network is used in many block ciphers such as DES, RC5, etc.**
  - **not used in AES**
  - **in DES, each L_i and R_i is 32 bits long; k_i is 48 bits long**



DES

- **DES has a fixed initial permutation IP prior to 16 rounds of encryption**
- **The inverse permutation IP^{-1} is applied at the end**

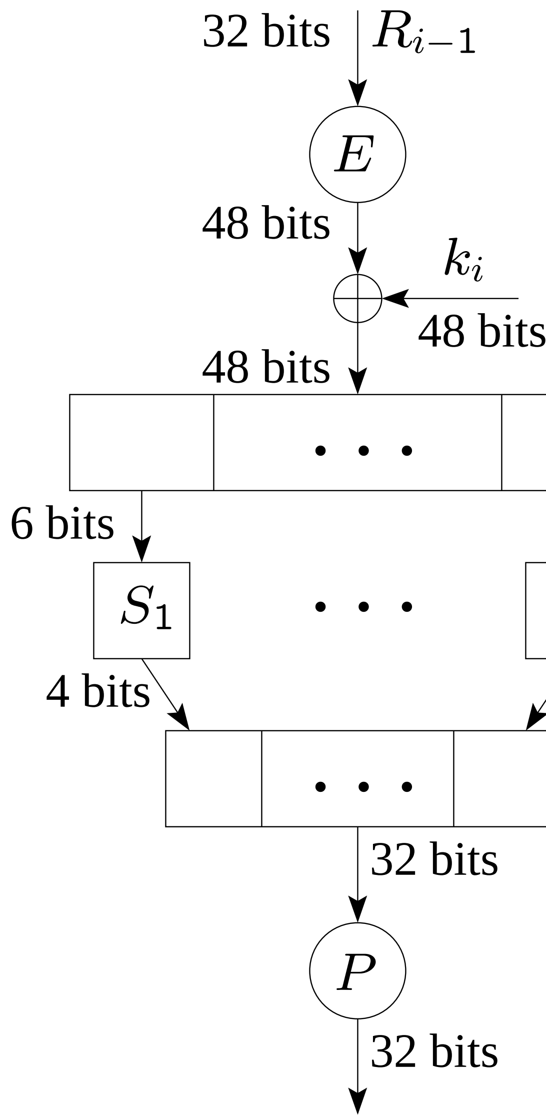


DES

- **The f function** $f(k_i, R_{i-1})$

  1. **first expands R_{i-1} from 32 to 48 bits (k_i is 48 bits long)**
  2. **XORs expanded R_{i-1} with k_i**
  3. **applies substitution to the result using S-boxes**
  4. **and finally permutes the value**

DES f Function



DES

- There are 8 S-boxes
  - S-boxes are the only non-linear elements in DES design
  - they are crucial for the security of the cipher
- **Example:** S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

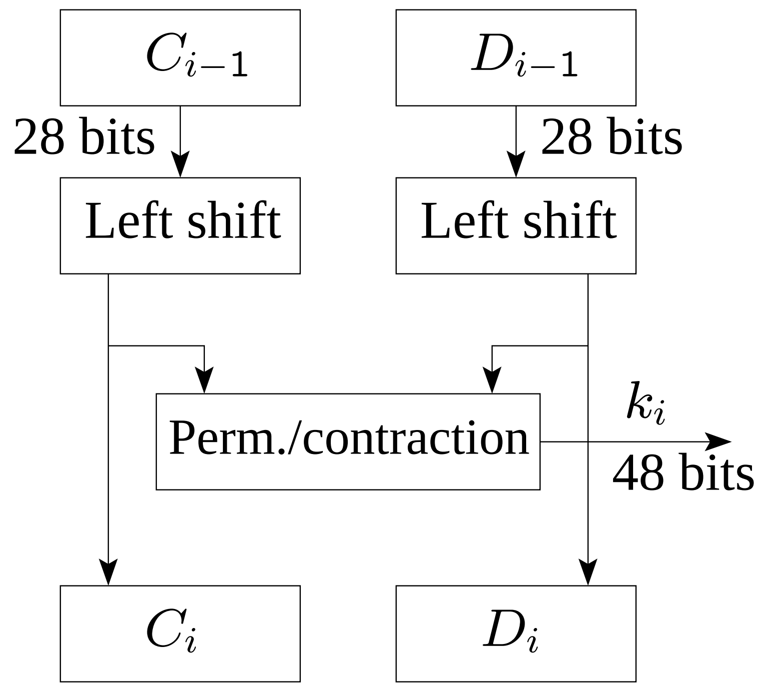
  - input to each S-box is 6 bits $b_1 b_2 b_3 b_4 b_5 b_6$
  - row = $b_1 b_6$, column = $b_2 b_3 b_4 b_5$
  - output is 4 bits

DES

- More about S-boxes..
  - a modified version of IBM's proposal was accepted as the standard
  - some of the design choices of S-boxes weren't public, which triggered criticism
  - in late 1980s – early 1990s differential cryptanalysis techniques were discovered
  - it was then revealed that DES S-boxes were designed to prevent such attacks
  - such cryptanalysis techniques were known almost 20 years before they were discovered by others

DES Key Schedule

- **Key computation consists of:**
  - **circular shift**
  - **permutation**
  - **contraction**



DES

- **Why does decryption work?**
  - **round function g is invertible**
    - **compute $L_{i-1} = R_i \oplus f(k_i, L_i)$**
    - **compute $R_{i-1} = L_i$**
  - **in the beginning apply IP and at the end apply IP^{-1}**
  - **round keys $k_{16}, \ldots, k_1$ and the f function are computed as before**

Attacks on DES

- **The key size is 56 bits, is this too small?**
- **Is the design non-linear enough to be hard to break?**
- **Are there cryptanalysis techniques that we can use against DES?**
  - **two techniques exist: linear and differential cryptanalysis**

DES Weak Keys

- **The master key k is used to generate 16 round keys**
- **Some keys result in the same round key to be generated in more than one round**
  - **this reduces complexity of the cipher**
- **Solution: check for weak keys at key generation**
- **DES has 4 weak keys:**
  - `0000000 0000000`
  - `0000000 FFFFFFF`
  - `FFFFFFF 0000000`
  - `FFFFFFF FFFFFFF`

Attacks on DES

- **Brute force attack**
  - try all possible 2^{56} keys
  - time-consuming, but no storage requirements
- It was conjectured in 1970s that a cracker machine could be built for $20 million
- In 1990s RSA Laboratories called several DES challenges
  - Challenge II-2 was announced in 1998
    - the winner was Electronic Frontier Foundation (EFF)
    - they built a DES Cracker machine for less than $250,000
    - it found the key in 56 hours and searched 88 billion keys per second

Attacks on DES

- RSA Laboratories called Challenge III in 1999
  - was solved in a record time
  - cooperative effort of the DES Cracker and a worldwide network of 100,000 computers
  - the key was found in 22 hours 15 minutes
  - over 245 billion keys were tested per second
  - http://www.distributed.net/des

Attacks on DES

- **Differential Cryptanalysis**
  - **complex technique that tracks the behavior of pairs of text blocks evolving along each round**
  - **set $\triangle_m = m_1 \oplus m_2$ and $\triangle_c = c_1 \oplus c_2$**
  - **distribution of $\triangle_c$'s, given $\triangle_m$'s, may reveal information about the key**
- **Not effective against DES**
  - **attack on 8-round DES requires 2^{38} known plaintext-ciphertext pairs**
  - **attack on 16-round DES requires 2^{47} chosen plaintext pairs**

Attacks on DES

- **Linear Cryptanalysis**
  - a slightly more recent technique (1993) based on finding linear approximations to describe DES transformations
  - the goal is to find some bits of the key
- **How does DES do?**
  - the attack has no practical implication on the cipher
  - attack on 8-round DES requires 2^{21} known plaintext pairs
  - attack on 16-round DES requires 2^{43} known plaintext pairs

Increasing Security of DES

- The best attack against DES is brute force search
  - DES uses a 56-bit key and this raised concerns
- One proposed solution is double DES
  - apply DES twice by using two different keys k_1 and k_2
  - encryption $c = E_{k_2}(E_{k_1}(m))$
  - decryption $m = D_{k_1}(D_{k_2}(c))$
- The resulting key is $2 \cdot 56 = 112$ bits, so it is more secure
  - is it really?

Meet-in-the-Middle Attack

- **The goal of the attack is, given pairs (m, c), find keys k_1 and k_2**

- **It is based on the observation that**

$$c = E_{k_2}(E_{k_1}(m)) \text{ and } E_{k_1}(m) = D_{k_2}(c)$$

- **Thus, the idea is to try all possible 2^{56} keys for k_1 and all possible keys for k_2 until a match is found**

Meet-in-the-Middle Attack

- $c = E_{k_2}(E_{k_1}(m))$ and $E_{k_1}(m) = D_{k_2}(c)$
- **Algorithm steps:**
  - encrypt m with all possible 2^{56} keys k_1
  - store all pairs $(k_1, E_{k_1}(m))$ sorted by $E_{k_1}(m)$
  - decrypt c with all possible 2^{56} keys k_2
  - for each decrypted result $D_{k_2}(c)$ check to see if there is a match $D_{k_2}(c) = E_{k_1}(m)$
  - when a match is found, verify the keys on another pair (m', c')
  - if the second pair matched, accept the keys k_1 and k_2
- The overall effort is on the order of 2^{56}

Meet-in-the-Middle Attack

- **Why do we need the second pair?**
  - **block size is 64 bits, so for a given m there are 2^{64} potential ciphertexts**
  - **with two 56-bit keys, there are 2^{112} potential double keys that can map m to c**
  - **for a single pair (m, c) the number of double keys (k_1, k_2) that produce $c = E_{k_2}(E_{k_2}(m))$ is $2^{112}/2^{64} = 2^{48}$**
  - **thus 2^{48} false alarms for a single pair are expected**
  - **with one more pair (m', c'), extra 64 bits of known text, the alarm rate goes down to $2^{48}/2^{64} = 1/2^{16}$**

Meet-in-the-Middle Attack

- **With two pairs (m, c) and (m', c') the correct keys k_1 and k_2 can be determined with probability $1 - 1/2^{16}$**
- **Known plaintext attack against double DES succeeds in about 2^{56} work as opposed to 2^{55} on average for DES**
  - **the 112-bit key provides the level of security similar to that of the 56-bit key**
- **Is there any hope to make DES stronger?**

Triple DES

- **Triple DES with two keys k_1 and k_2:**
  - **encryption** $c = E_{k_1}(D_{k_2}(E_{k_1}(m)))$
  - **decryption** $m = D_{k_1}(E_{k_2}(D_{k_1}(c)))$
- **Key space is $2 \cdot 56 = 112$ bits**
- **There is no know practical attack against 3DES with 2 keys**
  - **e.g., Merkle and Hellman attack requires 2^{56} chosen plaintext-ciphertext pairs and 2^{56} work**

Triple DES

- **Triple DES with three keys k_1, k_2, and k_3:**
  - **encryption** $c = E_{k_3}(D_{k_2}(E_{k_1}(m)))$
  - **decryption** $m = D_{k_1}(E_{k_2}(D_{k_3}(c)))$
- **Key space is $3 \cdot 56 = 168$ bits**
- **There is no known practical attack against it either**
- **Many applications that used DES switched to 3DES**
- **Can be made backward compatible by setting $k_1 = k_2$ or $k_3 = k_2$**

Summary of Attacks on DES

- DES
  - best attack: brute force search
  - 2^{55} work on average
  - no other requirements
- Double DES
  - best attack: meet-in-the-middle
  - requires 2 plaintext-ciphertext pairs
  - requires 2^{56} space and about 2^{56} work
- Triple DES
  - best practical attack: brute force search

Summary

- **DES is a block cipher with**
  - **key size of 56 bits**
  - **block size of 64 bits**
  - **Feistel structure**
  - **and 16 rounds**
- **DES was the de facto standard for over 20 years**
- **The best attack against DES is brute force search**
- **Triple DES can be used to improve resistance to such attacks**