Math 134: Cryptography

Lecture 17: the Data Encryption Standard

Eoin Mackall

February 19, 2025

University of California, Santa Cruz

Table of contents

The Data Encryption Standard (DES)

The Data Encryption Standard (DES)

In 1973, the National Bureau of Standards (now called NIST – the National Institute of Standards and Technology), issued a public request for a cryptographic algorithm to become a national standard.

The Data Encryption Standard (DES)

In 1973, the National Bureau of Standards (now called NIST – the National Institute of Standards and Technology), issued a public request for a cryptographic algorithm to become a national standard.

IBM submitted an algorithm called LUCIFER in 1974. The National Bureau of Standards sent this algorithm to the National Security Agecny (NSA), which made some changes.

The Data Encryption Standard (DES)

In 1973, the National Bureau of Standards (now called NIST – the National Institute of Standards and Technology), issued a public request for a cryptographic algorithm to become a national standard.

IBM submitted an algorithm called LUCIFER in 1974. The National Bureau of Standards sent this algorithm to the National Security Agecny (NSA), which made some changes.

The NSA then returned an algorithm which was essentially the Data Encryption Standard (DES) algorithm.

The Data Encryption Standard (DES)

The design decisions for DES were mostly mysterious until the 90's, when IBM released some details of the design criteria.

The Data Encryption Standard (DES)

The design decisions for DES were mostly mysterious until the 90's, when IBM released some details of the design criteria.

Overtime, DES became technologically dated. In 2000, NIST replaced the standard with a modified version called the Advanced Encrpytion Standard (AES).

The Data Encryption Standard (DES)

The Data Encryption Standard

The Data Encryption Standard (DES)

The Data Encryption Standard

- DES is a symmetric-key encryption algorithm

The Data Encryption Standard

- DES is a symmetric-key encryption algorithm
- DES uses a 56-bit encryption key. This means there are around $2^{56} \approx 7.206 \cdot 10^{16}$ elements in the keyspace.

The Data Encryption Standard (DES)

The Data Encryption Standard

- DES is a symmetric-key encryption algorithm
- DES uses a 56-bit encryption key. This means there are around $2^{56} \approx 7.206 \cdot 10^{16}$ elements in the keyspace.
- DES operates as a block cipher, transforming blocks of 64-bit plaintext into ciphertext

A simplified DES-type algorithm

A simplified DES-type algorithm

We first present a simplified version of DES with essentially all of the same underlying properties.

A simplified DES-type algorithm

We first present a simplified version of DES with essentially all of the same underlying properties.

In this simplification, we will operate on plaintext messages consisting of 12 bits (instead of the 64 used in DES).

A simplified DES-type algorithm

We first present a simplified version of DES with essentially all of the same underlying properties.

In this simplification, we will operate on plaintext messages consisting of 12 bits (instead of the 64 used in DES).

To encrypt a 12-bit plaintext message m, we first break m into left-right components L_0, R_0 so that $m = L_0R_0$.

A simplified DES-type algorithm

We first present a simplified version of DES with essentially all of the same underlying properties.

In this simplification, we will operate on plaintext messages consisting of 12 bits (instead of the 64 used in DES).

To encrypt a 12-bit plaintext message m, we first break m into left-right components L_0, R_0 so that $m = L_0 R_0$.

Example

Let's say the plaintext message to be sent is $m = 101000100110$.

A simplified DES-type algorithm

We first present a simplified version of DES with essentially all of the same underlying properties.

In this simplification, we will operate on plaintext messages consisting of 12 bits (instead of the 64 used in DES).

To encrypt a 12-bit plaintext message m, we first break m into left-right components L_0, R_0 so that $m = L_0R_0$.

Example

Let's say the plaintext message to be sent is $m = 101000100110$. Then

$$m = 101000100110 = \color{blue}{101000}\color{red}{100110}$$

so we set $L_0 = \color{blue}{101000}$ and $R_0 = \color{red}{100110}$.

A simplified DES-type algorithm

In this simplified version of DES, the key K has 9 bits.

A simplified DES-type algorithm

In this simplified version of DES, the key K has 9 bits.

The algorithm works by successively producing 12-bit strings m_i which are broken up into blocks $m_i = L_i R_i$ from the previous 12-bit string $m_{i-1} = L_{i-1} R_{i-1}$ and from 8-bits K_i of the key K.

A simplified DES-type algorithm

In this simplified version of DES, the key K has 9 bits.

The algorithm works by successively producing 12-bit strings m_i which are broken up into blocks $m_i = L_i R_i$ from the previous 12-bit string $m_{i-1} = L_{i-1} R_{i-1}$ and from 8-bits K_i of the key K.

We start with $m = m_0 = L_0 R_0$. We end with ciphertext $c = m_n = L_n R_n$ after some fixed number of rounds $n > 1$.

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

Example

Let's say $K = 111010011$.

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

Example

Let's say $K = 111010011$.Then:

$$K_1 = 11101001$$

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

Example

Let's say $K = 111010011$.Then:

$$K_1 = 11101001 \quad K_2 = 11010011$$

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

Example

Let's say $K = 111010011$.Then:

$$K_1 = 11101001 \quad K_2 = 11010011 \quad K_3 = 1010011\textcolor{red}{1}$$

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

Example

Let's say $K = 111010011$.Then:

$$K_1 = 11101001 \quad K_2 = 11010011 \quad K_3 = 1010011\textcolor{red}{1} \quad K_4 = 010011\textcolor{red}{11}$$

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.

Example

Let's say $K = 111010011$.Then:

$$K_1 = 11101001 \quad K_2 = 11010011 \quad K_3 = 1010011\color{red}{1} \quad K_4 = 010011\color{red}{11}$$

and so on.

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.
2. We use a function f which takes two inputs, a 6-bit input and an 8-bit input, and which outputs 6-bits.

(Simplified) DES Encryption (from m_{i-1} to m_i)

Let's say we have a 12-bit string $m_{i-1} = L_{i-1}R_{i-1}$. How do we produce the next string $m_i = L_iR_i$?

1. To start, we set K_i to be the 8-bit string gotten from K starting from the ith position.
2. We use a function f which takes two inputs, a 6-bit input and an 8-bit input, and which outputs 6-bits.
3. The 12-bit string $m_i = L_iR_i$ is gotten by setting $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$ where $\oplus$ is bit-wise addition modulo 2 (also called XOR).

(Simplified) DES Encryption (the function f)

Almost any function f can be used in this process, but results will vary. This one is similar to what's used in DES:

Almost any function f can be used in this process, but results will vary. This one is similar to what's used in DES:

The function f

We have two inputs: a 6-bit R_{i-1} input and an 8-bit K_i input.

(Simplified) DES Encryption (the function f)

Almost any function f can be used in this process, but results will vary. This one is similar to what's used in DES:

The function f

We have two inputs: a 6-bit R_{i-1} input and an 8-bit K_i input.

1. We first expand $R_{i-1} = b_1 b_2 b_3 b_4 b_5 b_6$ to 8-bits $b_1 b_2 b_4 b_3 b_4 b_3 b_5 b_6$.

(Simplified) DES Encryption (the function f)

Almost any function f can be used in this process, but results will vary. This one is similar to what's used in DES:

The function f

We have two inputs: a 6-bit R_{i-1} input and an 8-bit K_i input.

1. We first expand $R_{i-1} = b_1 b_2 b_3 b_4 b_5 b_6$ to 8-bits $b_1 b_2 b_4 b_3 b_4 b_3 b_5 b_6$.
2. We then compute $E_i = b_1 b_2 b_4 b_3 b_4 b_3 b_5 b_6 \oplus K_i$.

(Simplified) DES Encryption (the function f)

Almost any function f can be used in this process, but results will vary. This one is similar to what's used in DES:

The function f

We have two inputs: a 6-bit R_{i-1} input and an 8-bit K_i input.

1. We first expand $R_{i-1} = b_1b_2b_3b_4b_5b_6$ to 8-bits $b_1b_2b_4b_3b_4b_3b_5b_6$.
2. We then compute $E_i = b_1b_2b_4b_3b_4b_3b_5b_6 \oplus K_i$.
3. Now we use the S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

The function f

3. Now we use the S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

4. The first four bits of $E_i = a_1 a_2 a_3 a_4 ...$ indicate how to get the first three bits of $f(R_{i-1}, K_i)$ using S_1.

(Simplified) DES Encryption (the function f)

The function f

3. Now we use the S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

4. The first four bits of $E_i = a_1a_2a_3a_4\ldots$ indicate how to get the first three bits of $f(R_{i-1}, K_i)$ using S_1.
   The first bit a_1 specifies the row of S_1 to use (0 for the top, 1 for the bottom).

(Simplified) DES Encryption (the function f)

The function f

3. Now we use the S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

4. The first four bits of $E_i = a_1a_2a_3a_4\ldots$ indicate how to get the first three bits of $f(R_{i-1}, K_i)$ using S_1.
   The first bit a_1 specifies the row of S_1 to use (0 for the top, 1 for the bottom). The following three bits $a_2a_3a_4$ (interpreted as an integer) determine the column.

The function f

3. Now we use the S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

4. The first four bits of $E_i = a_1 a_2 a_3 a_4 \ldots$ indicate how to get the first three bits of $f(R_{i-1}, K_i)$ using S_1.
   The first bit a_1 specifies the row of S_1 to use (0 for the top, 1 for the bottom). The following three bits $a_2 a_3 a_4$ (interpreted as an integer) determine the column. The three bits $c_1 c_2 c_3$ in the corresponding spot are the output.

(Simplified) DES Encryption (the function f)

The function f

3. Now we use the S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

4. The first four bits of $E_i = a_1a_2a_3a_4...$ indicate how to get the first three bits of $f(R_{i-1}, K_i)$ using S_1.
   The first bit a_1 specifies the row of S_1 to use (0 for the top, 1 for the bottom). The following three bits $a_2a_3a_4$ (interpreted as an integer) determine the column. The three bits $c_1c_2c_3$ in the corresponding spot are the output.
5. The last four bits of $E_i = ...a_5a_6a_7a_8$ are used to get the last three bits $c_4c_5c_6$ of $f(R_{i-1}, K_i)$ similarly, using S_2.

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Let's use $m = 101000100110 = 101000100110 = L_0 R_0$ and $K = 111010011$ to find $m_1 = L_1 R_1$.

(Simplified) DES Encryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Let's use $m = 101000100110 = 101000100110 = L_0R_0$ and $K = 111010011$ to find $m_1 = L_1R_1$.

Note $K_1 = 11101001$. We have $L_1 = R_0 = 100110$. To find R_1 we need to evaluate $f(R_0, K_1)$.

(Simplified) DES Encryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Let's use $m = 101000100110 = \color{blue}{101000}\color{red}{100110} = \color{blue}{L_0}\color{red}{R_0}$ and $K = 111010011$ to find $m_1 = L_1 R_1$.

Note $K_1 = 11101001$. We have $L_1 = \color{red}{R_0} = \color{red}{100110}$. To find R_1 we need to evaluate $f(R_0, K_1)$.

We expand R_0 to 10101010 and compute

$$E_1 = 10101010 \oplus K_1 = 10101010 \oplus 11101001 = 01000011.$$

(Simplified) DES Encryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Note $K_1 = 11101001$. We have $L_1 = R_0 = 100110$. To find R_1 we need to evaluate $f(R_0, K_1)$.

We expand R_0 to 10101010 and compute

$$E_1 = 10101010 \oplus K_1 = 10101010 \oplus 11101001 = 01000011.$$

Then we use 0100 and S_1 to find 011 and 0011 and S_2 to find 101.

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Note $K_1 = 11101001$. We have $L_1 = R_0 = 100110$. To find R_1 we need to evaluate $f(R_0, K_1)$.

We expand R_0 to 10101010 and compute

$$E_1 = 10101010 \oplus K_1 = 10101010 \oplus 11101001 = 01000011.$$

Then we use 0100 and S_1 to find 011 and 0011 and S_2 to find 101. Hence $R_1 = L_0 \oplus 011101 = 101000 \oplus 011101 = 110101$ and

$$m_1 = L_1 R_1 = 100110110101.$$

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$?

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

1. We first swap the two halves of m_n to get $R_n L_n$.

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

1. We first swap the two halves of m_n to get $R_n L_n$.
2. By construction, we know that $L_n = R_{n-1}$.

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

1. We first swap the two halves of m_n to get $R_n L_n$.
2. By construction, we know that $L_n = R_{n-1}$.
3. We also know $R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$.

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

1. We first swap the two halves of m_n to get $R_n L_n$.
2. By construction, we know that $L_n = R_{n-1}$.
3. We also know $R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$. We calculate

$$
\begin{aligned}
R_n \oplus f(L_n, K_n) &= R_n \oplus f(R_{n-1}, K_n) \\
&= L_{n-1} \oplus f(R_{n-1}, K_n) \oplus f(R_{n-1}, K_n) = L_{n-1}
\end{aligned}
$$

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

1. We first swap the two halves of m_n to get $R_n L_n$.
2. By construction, we know that $L_n = R_{n-1}$.
3. We also know $R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$. We calculate

$$\begin{aligned} R_n \oplus f(L_n, K_n) &= R_n \oplus f(R_{n-1}, K_n) \\ &= L_{n-1} \oplus f(R_{n-1}, K_n) \oplus f(R_{n-1}, K_n) = L_{n-1} \end{aligned}$$

4. We combine to get $m_{n-1} = L_{n-1} R_{n-1}$.

(Simplified) DES Decryption

How does one decrypt from a 12-bit string $m_n = L_n R_n$? We show how one can recover $m_{n-1} = L_{n-1} R_{n-1}$ from m_n and the key K.

1. We first swap the two halves of m_n to get $R_n L_n$.
2. By construction, we know that $L_n = R_{n-1}$.
3. We also know $R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$. We calculate

$$\begin{aligned} R_n \oplus f(L_n, K_n) &= R_n \oplus f(R_{n-1}, K_n) \\ &= L_{n-1} \oplus f(R_{n-1}, K_n) \oplus f(R_{n-1}, K_n) = L_{n-1} \end{aligned}$$

4. We combine to get $m_{n-1} = L_{n-1} R_{n-1}$.

Remark

Aside from the swap, we effectively run DES encryption again using the keys $K_n, K_{n-1}, \ldots, K_2, K_1$.

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Suppose we use the same key $K = 111010011$ and we want to go from the message $m_1 = 100110110101$ back to m_0.

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Suppose we use the same key $K = 111010011$ and we want to go from the message $m_1 = 100110110101$ back to m_0.

We first write $m_1 = L_1R_1 = 100110110101$ and swap to get

$$R_1L_1 = 110101100110.$$

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

Suppose we use the same key $K = 111010011$ and we want to go from the message $m_1 = 100110110101$ back to m_0.

We first write $m_1 = L_1 R_1 = 100110110101$ and swap to get

$$R_1 L_1 = 110101100110.$$

We know that $L_1 = R_0 = 100110$. We calculate

$$L_0 = R_1 \oplus f(L_1, K_1).$$

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

We first write $m_1 = L_1 R_1 = 100110110101$ and swap to get

$$R_1 L_1 = 110101100110.$$

We know that $L_1 = R_0 = 100110$. We calculate

$$L_0 = R_1 \oplus f(L_1, K_1).$$

The 6-bit string L_1 is expanded to 10101010. Hence
$E_1 = 10101010 \oplus K_1 = 10101010 \oplus 11101001 = 01000011.$

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

We first write $m_1 = L_1R_1 = 100110110101$ and swap to get

$$R_1L_1 = 110101100110.$$

We know that $L_1 = R_0 = 100110$. We calculate

$$L_0 = R_1 \oplus f(L_1, K_1) = 110101 \oplus 011101.$$

The 6-bit string L_1 is expanded to 10101010. Hence
$E_1 = 10101010 \oplus K_1 = 10101010 \oplus 11101001 = 01000011.$

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

We first write $m_1 = L_1 R_1 = 100110110101$ and swap to get

$$R_1 L_1 = 110101100110.$$

We know that $L_1 = R_0 = 100110$. We calculate

$$L_0 = R_1 \oplus f(L_1, K_1) = 110101 \oplus 011101 = 101000.$$

The 6-bit string L_1 is expanded to 10101010. Hence
$E_1 = 10101010 \oplus K_1 = 10101010 \oplus 11101001 = 01000011.$

(Simplified) DES Decryption

S-boxes

$$S_1 = \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

Example

We first write $m_1 = L_1 R_1 = 100110110101$ and swap to get

$$R_1 L_1 = 110101100110.$$

We know that $L_1 = R_0 = 100110$. We calculate

$$L_0 = R_1 \oplus f(L_1, K_1) = 110101 \oplus 011101 = 101000.$$

The previous string was then $m_0 = L_0 R_0 = 101000100110$.