# What Is The Output Of The First Round Of The Des Algorithm When The Plaintext And The Key Are Both All

**What Is The Output Of The First Round Of The Des Algorithm When The Plaintext And The Key Are Both All**

Understanding the Data Encryption Standard (DES) is crucial for grasping how data security is maintained in various systems. One fundamental aspect of DES involves analyzing how the algorithm transforms plaintext into ciphertext through multiple rounds of complex operations. Specifically, examining the output after the first round when both the plaintext and the key are composed entirely of 'all' (assuming this means all bits set to 1) can provide insight into the core mechanics of DES encryption.

In this comprehensive article, we will explore the inner workings of DES, focusing on the initial round's transformation. We will delve into the initial permutation, key schedule, the expansion function, substitution boxes (S-boxes), permutation functions, and how these components interact when both the plaintext and the key are all ones. Our goal is to provide a detailed, step-by-step analysis suitable for students, security professionals, and enthusiasts aiming to deepen their understanding of DES operations.

---

## Understanding the Data Encryption Standard (DES)

Before analyzing the specific scenario, it's essential to understand the structure and process of DES encryption.

### Overview of DES

- Block Size: DES encrypts data in 64-bit blocks.
- Key Size: The key used is 64 bits, but only 56 bits are effectively used; the remaining 8 bits are parity bits.
- Rounds: The algorithm consists of 16 rounds of processing.
- Operations per Round:
- Expansion of the right half from 32 to 48 bits.
- XOR with the round key.
- Substitution through S-boxes.

- Permutation (P-box).
- XOR with the left half.

## Initial and Final Permutations

- Initial Permutation (IP): Rearranges the bits of the plaintext before processing.
- Final Permutation (IP-1): Reverses the initial permutation after all rounds.

Understanding these permutations is key to following how the bit transformations occur during encryption.

---

# Scenario: Plaintext and Key Are All Bits Set to 1

Let's clarify what this means:

- Plaintext: 64 bits, all set to 1 (binary: 111...111).
- Key: 64 bits, all set to 1 (binary: 111...111).

This scenario provides a simplified case to analyze how DES processes uniform input data.

---

# Step-by-Step Analysis of the First Round

The first round of DES involves several steps:

1. Initial Permutation of Plaintext
2. Splitting the permuted plaintext into Left and Right halves
3. Key Schedule: Generating Round Key
4. Expansion of the Right Half
5. XOR with Round Key
6. Substitution via S-boxes
7. Permutation (P-box)
8. XOR with Left Half to produce new Right Half
9. Swapping halves (if applicable)

Let's analyze each step.

---

## 1. Initial Permutation (IP)

The initial permutation rearranges the bits according to a fixed table. Since the plaintext is all ones, the permutation will produce a rearranged sequence of all ones, just in a different order. The permutation table is known and fixed.

Result: After IP, the 64 bits are still all ones, but their positions are permuted.

---

## 2. Splitting into Left and Right Halves

The permuted 64-bit block is split into two 32-bit halves:

- Left (L0): 32 bits
- Right (R0): 32 bits

Since the permutation preserved the number of bits and all bits are 1, both halves are sequences of 32 ones.

Result:

- L0: 32 ones
- R0: 32 ones

---

## 3. Key Schedule: Generating the Round Key

- The 64-bit key undergoes parity bit dropping (removing every 8th bit), resulting in a 56-bit key.
- The 56-bit key is split into two 28-bit halves.
- These halves are shifted according to a schedule to generate round subkeys.
- For simplicity, since the key is all ones, the parity bits are also ones, and after dropping, the key halves are sequences of ones.

Result:

- Round key (K1): 48 bits, all ones (after compression permutation).

---

## 4. Expansion of the Right Half (E-box)

- The 32-bit R0 is expanded to 48 bits using the expansion table.
- Since R0 is all ones, the expanded R0 will be a sequence of 48 ones.

Result:

- Expanded R0: 48 ones

---

## 5. XOR with Round Key

- The 48-bit expanded R0 is XORed with the 48-bit round key.
- Both are all ones, so:

XOR of 1 and 1 = 0

Result:

- After XOR: 48 zeros

---

## 6. Substitution via S-boxes

- The 48-bit result is divided into 8 groups of 6 bits each.
- Each group of 6 bits is input to an S-box, which outputs 4 bits.
- The S-boxes are fixed lookup tables.

Given that the input to each S-box is 6 zeros, we need to determine the output for input '000000'.

S-box behavior:

- The S-box takes the outer two bits to determine the row.
- The inner four bits determine the column.
- For '000000':
- Row bits: first and last bits: 0 and 0 → row = 0
- Column bits: middle four bits: 0000 → column = 0

Lookup in each S-box at position [row=0][column=0].

Most S-boxes are designed such that the output for input '000000' is a specific 4-bit value, often 14 (1110) or 0, depending on the specific S-box.

Result:

- Each S-box outputs a 4-bit value based on input '000000'.
- For simplicity, assuming S-box 1 outputs '1110' (14), and similar for others, but actual values depend on S-box design.

Let's assume each S-box outputs '1110' for input '000000' (this is a common pattern in some S-boxes).

- Total output: 8 4 bits = 32 bits, each '1110'.

---

## 7. Permutation (P-box)

- The 32 bits are permuted according to the P-box table.
- Since all bits are '1110', after permutation, the pattern of bits remains similar, just reordered.

Result:

- P-box output: sequence of 32 bits, each either '1' or '0', based on permutation.

---

## 8. XOR with Left Half (L0)

- The P-box output is XORed with L0 (which is all ones).
- XOR with '1' flips the bit.

- Since P-box output is a mixture of '1's and '0's, the XOR will invert each bit accordingly.

Result:

- New R1 = L0 XOR P-box output

---

## 9. Swapping Halves (L1 and R1)

- For the first round, the new left half L1 is R0.
- The new right half R1 is the XOR result obtained above.

Summary of first round output:

- Left: R0 (all ones)
- Right: XOR of L0 and P-box output

---

# Summary of the First Round Output When Plaintext and Key Are All Ones

- The initial permutation preserves the all-ones pattern.
- The right half after expansion remains all ones.
- XOR with the round key results in all zeros.
- S-box substitution for input '000000' yields a fixed 4-bit output per S-box, often '1110'.
- After the P-box permutation, bits are rearranged but remain the same in content.
- XOR with the left half (all ones) inverts bits, leading to the final right half of the first round being a pattern of zeros and ones depending on the P-box output.

---

# Implications and Conclusions

This analysis demonstrates how the DES algorithm processes uniform input data:

- When both plaintext and key are all ones, the initial steps of DES produce a predictable pattern of zeros and ones after the first round.
- The key operation—XOR with the round key—transforms the data into zeros, which then undergo substitution and permutation.
- The specific output depends heavily on the S-box design and permutation tables.

Practical Considerations:

- Such uniform inputs are rare in real-world data, but analyzing them helps understand the algorithm's behavior.
- The pattern of zeros and ones after the first round can be used as a baseline for cryptanalysis or understanding the diffusion properties of DES.

---

# Conclusion

Understanding the output of the first round of DES when both plaintext and key are set to all ones requires a detailed step-by-step examination of each operation. The process involves permutations, expansions, XOR operations, substitutions via S

# Frequently Asked Questions

## What is the output of the first round of the DES algorithm when both the plaintext and the key are all ones?

The output of the first round will be the result of applying the expansion, XOR with the key, substitution through S-boxes, and permutation to the initial plaintext, all with the key bits set to all ones. This specific output depends on the initial permutation and the S-box mappings, but generally, it produces a fixed, deterministic 32-bit value based on these inputs.

## Does setting both plaintext and key to all ones simplify the first round of DES?

Yes, setting both plaintext and key to all ones simplifies the initial XOR operation and can make the resulting intermediate values more predictable, but the overall complexity of DES ensures that the output still appears highly nonlinear and secure.

## How does the initial permutation (IP) affect the first round output when plaintext is all ones?

The initial permutation rearranges bits of the plaintext; with all ones, it results in the permuted block still being all ones, so the permutation doesn't alter the uniformity but sets the stage for subsequent operations.

## What role does the key's all-one value play in the first round of DES?

The all-one key contributes to the XOR operation with the expanded plaintext, producing a specific pattern in the round's input to the S-boxes, which influences the substitution output and ultimately the round's result.

## Can the output of the first round be predicted if both plaintext and key are all ones?

While the operations are deterministic, predicting the exact output requires knowledge of the permutation tables, S-box mappings, and the internal structure of DES. With all ones, the pattern is predictable, but the exact output depends on these fixed components.

## Is the first round output of DES when both plaintext and key are all ones considered secure or predictable?

The output becomes predictable in the sense that it follows a fixed pattern based on the inputs, but DES's security relies on the complexity of multiple rounds; a single round's output alone isn't indicative of overall security.

# Additional Resources

What Is The Output Of The First Round Of The DES Algorithm When The Plaintext And The Key Are Both All

---

Introduction

The Data Encryption Standard (DES) has long been a cornerstone in the field of cryptography, historically serving as a primary symmetric-key encryption algorithm for securing sensitive data. Developed in the early 1970s and officially adopted as a federal standard in 1977, DES's structure relies on a series of iterative rounds—each applying complex permutations and substitutions—to transform plaintext into ciphertext.

Understanding the behavior of DES at each stage, particularly during its initial round, is crucial for

cryptanalysts, security researchers, and students seeking to grasp the intricacies of block cipher operations. A specific scenario of interest involves analyzing the output of the first DES round when both the plaintext and the key are composed entirely of binary ones—i.e., all bits set to 1. This scenario provides a controlled environment to observe how the algorithm's initial transformations respond to such uniform inputs, offering insights into its diffusion and confusion mechanisms.

This article delves into the detailed step-by-step process of the DES first round under these conditions, presenting the underlying transformations, the impact of initial permutations, key scheduling, expansion, substitution, and final permutation. The goal is to provide a comprehensive, technical understanding suitable for academics, security professionals, and enthusiasts alike.

---

## Background: The DES Structure and Its First Round

### Overview of DES Architecture

DES operates on 64-bit blocks, using a 56-bit key (with 8 parity bits). Its structure involves:

- Initial Permutation (IP)
- 16 rounds of Feistel operations
- Final Permutation (IP-1)

Each round applies a function $f(R, K)$ to the right half of the data, then XORs the result with the left half, swapping halves afterward.

### Focus on the First Round

In the first round:

- The plaintext undergoes an initial permutation (IP).
- The permuted data is split into left and right halves.
- The right half is expanded, XORed with the first round key, substituted via S-boxes, permuted, and then combined with the left half.

Given that the plaintext and the key are both all ones, the examination of the first round's output reveals how the DES's initial transformations handle uniform bit patterns.

---

## Assumptions and Initial Conditions

For this analysis, we assume:

- Plaintext: 64 bits, all set to 1 (binary 1).
- Key: 64 bits, all set to 1 (binary 1). Since DES uses a 56-bit key, the parity bits are included but do not affect core operations; all bits are 1.

These assumptions simplify calculations and highlight the cipher's behavior under uniform input conditions.

---

Step-by-Step Breakdown

1. Initial Permutation (IP)

The first step permutes the plaintext bits according to a fixed table, rearranging bits. The permutation table (standard IP table) rearranges bits from the original positions to new positions.

Result: With all bits as 1, the permuted block remains all ones. The permutation merely changes the order but not the value of bits; since all bits are 1, the permuted block remains 64 ones.

Implication: The initial permutation does not alter the content; it only permutes positions.

2. Splitting into Left and Right Halves

Post-IP, the 64-bit block is divided into:

- Left half (L0): bits 1–32
- Right half (R0): bits 33–64

Since all bits are 1, both halves are 32 bits of all ones.

3. Key Schedule: Generating the First Round Key

DES derives round keys through:

- Compression permutation (PC-1)
- Left shifts
- Compression permutation (PC-2)

Given an all-ones key, the key schedule produces a round key of 48 bits with a specific pattern.

Key considerations:

- Parity bits are ignored in the key schedule; all bits are 1.

- The key bits are permuted and shifted identically, resulting in a round key with all bits set to 1 (assuming the key bits are all 1s and shifts do not introduce zeros).

Result: The first round key, $K_1$, is a 48-bit value with all bits equal to 1.

---

4. The Function $f(R, K)$ in the First Round

The Feistel function involves several steps:

a. Expansion (E-Box)

- Expands 32 bits to 48 bits via a fixed expansion table.
- With R0 as all ones, the expansion repeats certain bits according to the table.

Result: The expanded 48-bit block is all ones because the input is all ones; the expansion table only repeats bits, so the output remains all ones.

b. XOR with Round Key

- The expanded R0 (all ones) is XORed with the round key $K_1$ (all ones).

XOR Operation:

- 1 XOR 1 = 0

Result: The XORed output is a 48-bit block of all zeros.

c. Substitution via S-Boxes

- The 48-bit XOR output is divided into 8 groups of 6 bits each, passed through respective S-boxes.
- Each S-box maps 6 bits to 4 bits.

Behavior with all zeros input:

- For each 6-bit input, the S-box lookup depends on the input value.
- Since all bits are zero, each 6-bit group is 000000 binary (decimal 0).

S-Box Output:

- The first S-box (S1) maps 000000 to a specific 4-bit output.
- Similarly, all other S-boxes process 000000.

Standard S-Box Values:

- Each S-box has a predefined lookup table.
- For input 000000 (row=0, column=0), the output is the value at position [row=0][column=0].

Example:

- S1: maps 000000 to 14 (binary 1110)
- S2: maps 000000 to 0 (binary 0000)
- S3: maps 000000 to 15 (binary 1111)
- S4: maps 000000 to 8 (binary 1000)
- S5: maps 000000 to 13 (binary 1101)
- S6: maps 000000 to 1 (binary 0001)
- S7: maps 000000 to 12 (binary 1100)
- S8: maps 000000 to 7 (binary 0111)

Note: Actual S-box outputs depend on standard tables, but the key point is that the input is always 000000, and the outputs are fixed per S-box.

Result: Concatenate the 8 4-bit outputs to form a 32-bit block.

d. Permutation (P-Box)

- The 32-bit S-box output is permuted according to a fixed permutation table.
- Since the input bits are fixed (from the S-box outputs), the permutation rearranges bits but does not change their values.

Result: The permuted 32-bit block, which is then XORed with the left half.

---

5. Finalizing the First Round Output

- The output of the function \(f(R, K)\) (the permuted S-box outputs) is XORed with the left half (which is all ones).

XOR Operation:

- 1 XOR 1 = 0

Result:

- The new right half becomes the previous left half (all ones).

- The new left half becomes the result of the XOR (all zeros).

This completes the first round, with the output being the concatenation of the new left and right halves:

- Left: all zeros
- Right: all ones

---

6. Final Permutation (Post-Round)

- After completing all rounds, the output undergoes a final permutation, which is the inverse of the initial permutation (IP-1).
- Since the data consists of all zeros and ones, the permutation rearranges bits but does not change the pattern of the data.

---

Summary of the First Round Output

Given the initial conditions:

- Plaintext: all bits set to 1
- Key: all bits set to 1

The first round produces:

- Left half: all zeros (32 bits)
- Right half: all ones (32 bits)

The combined 64-bit block after the first round is: 32 zeros followed by 32 ones

---

Implications and Observations

Diffusion and Confusion

The scenario illustrates that, even with uniform inputs, DES's structure introduces non-trivial transformations. The expansion and S-box substitutions, driven by the key, produce a mixture of bits that deviate from the initial uniformity.

Sensitivity to Input Patterns

While the initial input pattern was uniform, the first round's output reflects the cipher's non-linear components (S-boxes) and permutations, which serve to diffuse the bits across the data block.

Security Context

This controlled case underscores the importance of key and plaintext randomness. Uniform inputs tend to produce predictable patterns, but the non-linear S-box transformations ensure that subsequent rounds introduce sufficient complexity.

---

Conclusion

Analyzing the output of the first DES round when both plaintext and key are all ones reveals that:

- The initial permutation preserves the uniformity.
- The expansion and XOR with the

## What Is The Output Of The First Round Of The Des Algorithm When The Plaintext And The Key Are Both All

What Is The Output Of The First Round Of The Des Algorithm When The Plaintext And The Key Are Both All

## Related Articles


- Which Accurately Describes Reproduction In Gymnosperms? Select Four Options.Some Require Heat For Seed
- Using Logicworks, Design The Addition And Subtraction Circuits For An Arithmetic Logic Unit Capable Of
- What Is The Name Of The Protein That Combines With Cyclins To Exert Local Control Of The Cell Cycle?A)

Back to Home